Resumen

Se deberá crear un algoritmo de control de reloj de tiempo real (RTC V3023) el cual será programado en el micro-controlador PicoBlaze de Xilinx según su set de instrucciones y arquitectura. Se implementará la metodología del diseño modular y aplicando conceptos de diseño RTL y Top-Down.El reto del proyecto no solo es reemplazar las máquinas de estado de control general del proyecto pasado, sino también aprender a interconectar el micro-controlador con los sistemas periféricos mediante una interfaz de control eficaz. Además se debe respetar el protocolo del teclado USB-PS2 y los tiempos de escritura y lectura del RTC. La información obtenida de este debe ser desplegada en una pantalla VGA usando el controlador existente y para el caso de la alarma del Timer se debe producir un sonido de alerta.

Introducción

Un RTC V3023 (Real Time Controller) es un dispositivo Electrónico que mide tiempo real y lo muestra en una SRAM en formato de (años,meses,días,horas,minutos,segundos) cada uno de 8 bits en BCD. También tiene una rutina de programación para ajustar la hora y fecha. Además el RTC posee funciones secundarias como una alarma y un temporizador. Este último será utilizado en este proyecto. Los datos obtenidos del RTC serán desplegados en una Pantalla VGA usando de base el controlador del proyecto pasado. Utilizar la plataforma Nexys 4 para el desarrollo de este proyecto resulta útil porque la placa provee al diseñador las herramientas necesarias para una implementación eficiente del sistema. El PicoBlaze es un micro-controlador compacto de arquitectura 8-bit diseñado para FPGAs de Xilinx mediante descripción HDL a nivel de celdas (soft core). El PicoBlaze es bastante eficiente y ocupa poco espacio relativamente. Su comportamiento es programado en lenguaje ensamblador mediante un set de aproximadamente 100 instrucciones. Las señales desplegadas por el circuito controlador deben cumplir los estándares establecidos, a continuación se muestran los tiempos de señales requeridas y la memoria RAM donde se escriben y leen datos del V3023:

Timing Characteristics (standard temperature range)
V_{DD}= 5.0 ±10%, V_{SS} = 0V and T_A=-40 to +85°C

Parameter	Symbol	Test Conditions	Min.	Typ.	Max.	Unit
Chip select duration, write cycle	t_{CS}		50			ns
Write pulse duration	t_{WR}		50			ns
Time between two transfers	t_W		100			ns
RAM access time (note 1)	t_{ACC}	C_{LOAD} = 50pF		50	60	ns
Data valid to Hi-impedance (note 2)	t_{DF}		10	30	40	ns
Write data settle time (note 3)	t_{DW}		50			ns
Data hold time (note 4)	t_{DH}		10			ns
Advance write time	t_{ADW}		10			ns
$\overline{PF}$ response delay	t_{PF}				100	ns
Rise time (all timing waveform signals)	t_R				200	ns
Fall time (all timing waveform signals)	t_F				200	ns
$\overline{CS}$ delay after $\overline{A}$ /D (note 5)	$t_{\overline{A}/Ds}$		5			ns
$\overline{CS}$ delay to $\overline{A}$ /D	$t_{\overline{A}/Dt}$		10			ns

Figura 1: Tiempos de RTC V3023

RAM Map

Address Dec	Address Hex	Parameter	Range
Data Space			
Status			
00	00	status 0	
01	01	status 1	
02	02	status 2	
Special purpose			
16	10	digital trimming	0-255
Clock			
32	20	1/100 second	00-99
33	21	seconds	00-59
34	22	minutes	00-59
35	23	hours (note 1)	00-23
36	24	date	01-31
37	25	month	01-12
38	26	year	00-99
39	27	week day	01-07
40	28	week number	00-53
Alarm			
48	30	1/100 second	00-99
49	31	seconds	00-59
50	32	minutes	00-59
51	33	hours (note 1 & 2)	00-23
52	34	date	01-31
Timer			
64	40	1/100 second	00-99
65	41	seconds	00-59
66	42	minutes	00-59
67	43	hours	00-23

Figura 2: Distribución de RAM en RTC V3023

Objetivos

1. Objetivo General

- Desarrollar habilidades básicas en el diseño de sistemas digitales avanzados aprovechando las capacidades de diseño y verificación de las herramientas EDA para diseñar sobre microcontrolador programable en FPGA por medio de HDLs y lenguaje ensamblador.

2. Objetivos Específicos

- Proponer un diseño a nivel de bloques de un sistema que junto con un algoritmo del microcontrolador permita controlar el dispositivo RTC V3023, específicamente la obtención de fecha/hora, la programación de la misma y de un temporizador.El diseño deberá incorporar las buenas prácticas del diseño Top-Down.
- Mejorar el sistema de despliegue de caracteres y símbolos en una pantalla VGA compatible de 640 por 480 pixeles para visualizar los datos extraídos y los que se van a programar. También mejorar la interfaz de usuario al darle la posibilidad de introducir los datos mediante un teclado de protocolo PS2.
- Desarrollar los códigos del microcontrolador y los periféricos en ensamblador y Verilog respectivamente en el Xilinx ISE que implementa el diseño del primer objetivo, y verificarlo a nivel de simulación, con resultados de simulación post-síntesis en Modelsim, a una velocidad de reloj de al menos 100MHz.
- Comprobar el funcionamiento correcto de la unidad sobre una FPGA Nexys 4.

3. Objetivos de aprendizaje

- En el transcurso de este proyecto se pondrán a prueba los valores de honestidad por el respeto a la información, la disciplina en el trabajo, la objetividad en el trato a los compañeros, el liderazgo y el orden en lograr cumplir con las tareas.

- Para la realización de este proyecto el estudiante deberá aprender a especificar correctamente sistemas digitales por medio de HDLs y el uso de herramientas EDA.
- Para la realización de este proyecto el estudiante debe saber utilizar la metodología de diseño Top-Down aplicada a sistemas digitales, y aplicar los procedimientos adecuados de medición de señales eléctricas digitales.

Descripción del sistema

1. Diseño de Bloques RTL

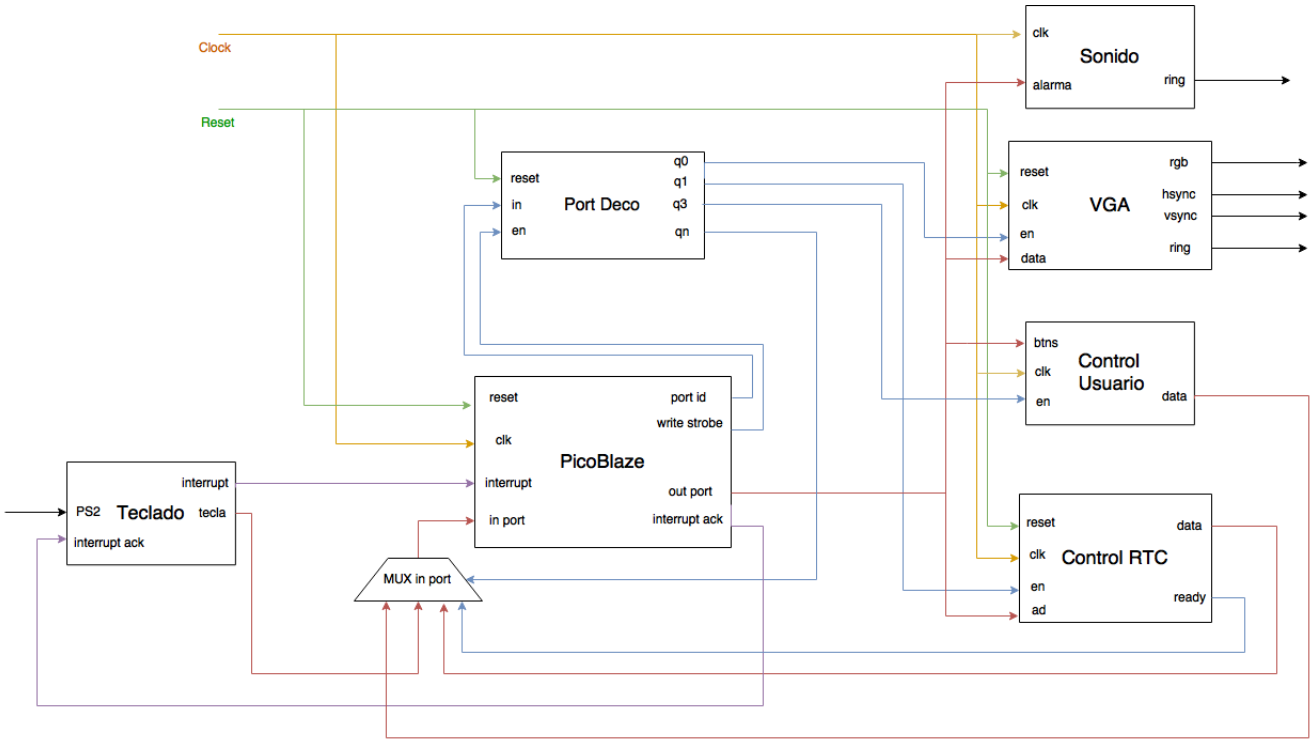

Figura 3: Diagrama RTL de Control del RTC V3023 con PicoBlaze

Como casi todo sistema digital, el Controlador RTC implementado con PicoBlaze se divide en una ruta de Datos y otra de señales de control. En la Figura 3 se muestra el diseño a nivel de bloques RTL de ruta de datos del sistema y a continuación se describirán los módulos pertinentes.

Teclado: Este módulo se encarga de recibir los datos introducidos por el usuario a través de un teclado USB-PS2 y mantenerlos de ser necesario hasta que el microcontrolador los revise. Se compone de cuatro pequeños bloques: Un registro SiPo (Serial in, Parallel out) para obtener el código de la tecla y estabilizarlo, luego un circuito combinacional que de salida tiene las señales que activan o no estos códigos. Luego un circuito secuencial que hace que dichas señales solo permanezcan en alto por 10ns para evitar que la máquina de control de usuario malinterprete la cantidad de pulsos. Por último está un módulo de flip flops SR que mantienen las señales de programar hasta que el PicoBlaze las consulte.

Multiplexor In Port: Este módulo representa la selección de puerto de entrada que realiza el microcontrolador para obtener datos del resto del circuito. El multiplexor tiene una salida de 8 bits y 3 canales generales de 8 bits como entrada: Datos del Teclado, Control RTC y Control de usuario. La entrada de selección es la señal

Port ID que provee el PicoBlaze.

PicoBlaze: El PicoBlaze es el bloque central ya que es el microcontrolador que contiene el algoritmo ensamblado y se encarga de guiar el flujo de datos y generar salidas de control necesarias. Como entradas posee un puerto de entrada de 8 bits, una entrada de clock, una de interrupción y una de reset. Como salidas posee un puerto de salida de 8 bits, la señal de selección de puerto, una señal de escritura (write strobe) para indicar que el dato de salida es estable y una señal de apagado de interrupción.

Decodificador de Puerto: Este módulo representa un circuito combinacional cuya entrada es el número de puerto de 8 bits y sus salidas son señales de habilitación de los demás módulos, así aunque todos comparten los datos de salida del PicoBlaze, solo el módulo habilitado puede recibirlos. También tiene una entrada de reset que pone todas sus salidas en bajo y una entrada enable que corresponde al write strobe del microcontrolador.

Control RTC: Este módulo representa un circuito secuencial (máquinas de estado) que controlan los procesos de escritura y lectura del RTC. El diseño de estos toman en cuenta los tiempos a respetar presentados en la Introducción, además poseen como salidas las señales de control WR, RD, CS, AD y un registro de 8 bits cuyas salidas van al Multiplexor In Port.

Registro Data Usuario: El módulo de registros se encarga de guardar los datos a escribir. Funciona como una interfaz entre el usuario (teclado) y el microcontrolador. Este banco de información puede ser manipulado por el usuario para modificar datos de fecha, mes, años, hora, minutos y segundos con el fin de programar una hora o el temporizador del sistema. Cuando el usuario haya terminado de establecer los datos deseados, se graba toda la información en los registros de puerto de entrada. Cuando el programa ensamblado se encuentra en la etapa de programar el reloj o el temporizados, el PicoBlaze lee los datos de este banco de registros. También puede controlar cuando leer las salidas de un módulo de resta necesario para implementar la cuenta en reversa del temporizador.

Controlador VGA: Este módulo consiste básicamente en el circuito utilizado para el proyecto anterior. Un sincronizador que toma el reloj del sistema y lo adecua a la frecuencia de trabajo determinada para el monitor VGA y un generador de caracteres, que toma los datos de memoria y registros para desplegarlos en pantalla. En este caso se debe imprimir la información de fecha, hora, temporizador, una señal (ring) cuando se detenga el temporizador y un pequeño resumen de los controles del teclado.

Sonido: Este bloque se encarga de reproducir una melodía cuando la señal de entrada alarma se activa, su salida va a las salidas analógicas de la Nexys 4 para posteriormente ser amplificada por parlantes.

2. Programa Ensamblado en Microcontrolador

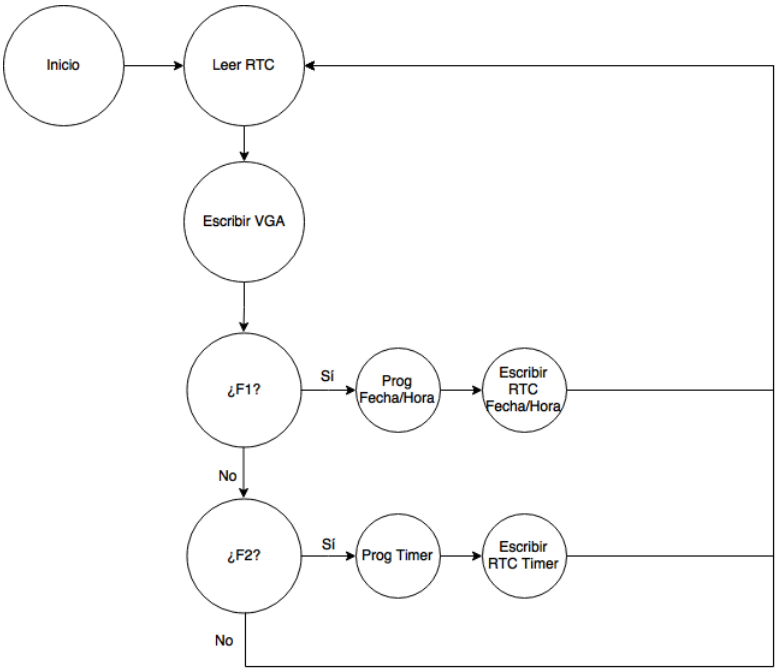

Figura 4: Diagrama de Algoritmo de control principal

Algoritmo de control principal: Como lo dice su nombre, se encarga de controlar el funcionamiento general del sistema. Esta diseñado de tal manera que pueda ser codificado en lenguaje ensamblador y así implementarlo en el PicoBlaze. Básicamente es una rutina que revisa la señales de entrada y las compara para tomar decisiones. También controla el flujo de datos, por ejemplo: selecciona los datos que deben ser desplegados en la VGA según la etapa en que se encuentre. También posee rutinas que no aparecen en el diagrama y por razones de espacio se representan como etapas, pero se describen a continuación:

- Inicio: Rutina en la que el microcontrolador lleva a cabo el proceso de inicialización del RTC que indica el fabricante incluyendo la activación del IRQ flag necesaria para utilizar el temporizador.
- Leer RTC: En esta rutina el sistema lee el RTC y despliega la información en pantalla VGA, solo es interrumpido por las señales de botón de programar fecha/hora o el temporizador en cuyo caso salta a la rutina de la programación seleccionada, de no ser así vuelve a Leer RTC.
- Prog Fecha/Hora: Rutina en que se espera que el que el usuario programa la fecha y la hora que quiera usando las flechas del teclado y una vez que termina presiona enter para saltar a escribir los datos.
- Escribir RTC F/H: Una vez confirmados los datos de la fecha y la hora, esta rutina los introduce en el RTC.
- Prog Temp: En este el usuario programa el temporizador usando el método anterior. Luego salta a la rutina que escribe el tiempo de temporizado en el RTC.
- Escribir RTC Timer: Una vez confirmados los datos del temporizador, esta rutina los introduce en el RTC. Además reinicia la señal de IRQ para controlar el ring.

3. Diseño de Máquinas de Estado

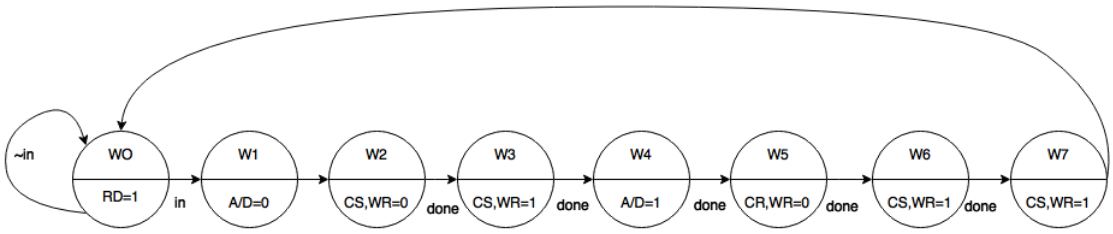

Figura 5: Control de escritura en el RTC V3023

Control de escritura: Para escribir sobre la RAM del RTC, es necesario manipular las señales de control A/D, WR, RD y CS adecuadamente, cumpliendo con los tiempos de activación de cada una en el proceso de escritura, según indica las hojas de datos. Para lograr el cometido, se diseñó la máquina de estados que representada en la figura 11. Esta máquina de estados se encuentra en su estado inicial hasta que el microcontrolador permita su activación. Por cada estado se van activando y desactivando las salidas de control al RTC antes mencionadas, con el fin de cumplir con el protocolo establecido. Por otro lado, este sistema respeta los tiempos de acceso al RTC en modo escritura; según la hoja de datos, hay señales que deben variar con un lapso mínimo de hasta 100 ns; como el sistema general trabaja a una frecuencia de 100MH, el siguiente pulso positivo del reloj no puede ser el que mueva la máquina de escritura. En los estados pertinentes, existen salidas que activan un temporizador, el cual brinda una señal de çuenta-terminadaçuando el tiempo pre-establecido por el mismo estado se haya cumplido. Dicha señal habilita entonces el cambio de estado en la máquina y el movimiento de la señales de control. A modo general, consta de 8 estados que se mueven según el reloj y el temporizador. Se recorre completamente esta máquina de estados por cada ciclo de escritura del sistema general.

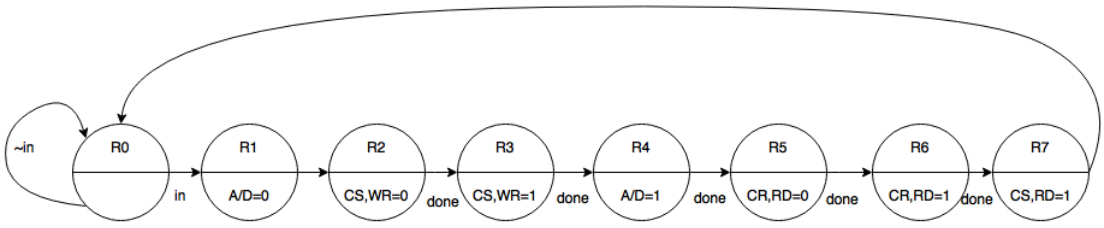

Figura 6: Control de lectura desde el RTC V3023

Control de lectura: Este control del RTC se encarga de verificar los ciclos de lectura en el chip; el diagrama es mostrado en la figura 12. Su comportamiento es similar al del Control de Escritura, pues del mismo modo, debe respetar tiempos de acceso de lectura del RTC predefinidos por el fabricante. De esta manera, las salidas de los estados son la activación y desactivación de dichos controles a los tiempos requeridos. En este módulo, como el anterior, se utiliza un temporizador al que se le define el tiempo a contar según el estado donde se encuentre el sistema. Una finalizado el ciclo del temporizador, la máquina de estados continúa con su ciclo de escritura. Esta máquina de estados se ejecuta constantemente en el sistema excepto en el periodo de inicialización y el de programación de fecha y hora. Consta de 8 estados y por cada ciclo de lectura que se realiza en el sistema general, se recorre esta máquina en su totalidad

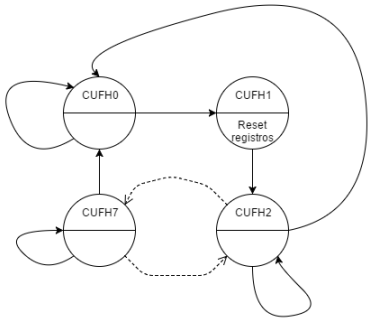

Figura 7: Control de usuario para modificar registros de Fecha/Hora

Máquinas de Estado de Control de Usuario para programar Fecha/Hora: Esta máquina se ejecuta cada vez que el usuario desea modificar la fecha y hora; recorre un banco de registros con la información de fecha que el usuario desea programar, se puede intercambiar entre estados de forma bidireccional de forma que el usuario pueda variar como él desea el registro a programar. Además, a partir de cualquier estado se puede volver al estado inicial. Cada inicio de ciclo de la máquina, se resetean los valores del registros para que el usuario puede modificarlos desde 0. Una vez finalizada la programación, se ejecuta la máquina para Programar Fecha/Hora. En la figura 9 se puede observar el diagrama de esta máquina.

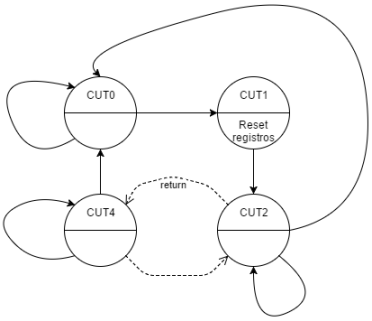

Figura 8: Control de usuario para modificar registros de Temporizador

Máquinas de Estado de Control de Usuario para programar Temporizador: Esta máquina de estados funciona igual que la anterior descrita, la única diferencia consiste en que no se graba en el RTC lo que el usuario programe, la resta de 24 horas menos lo programado por el usuario, mediante el uso de un decodificador y un restador. Esto pues el timer del RTC cuenta hacia arriba y se desea desplegar una cuenta hacia abajo. La máquina se observa en la figura 10.

4. Para el controlador VGA

Para desplegar la información del reloj, así como las posibles teclas a utilizar dentro del sistema para ingresar datos del teclado se muestra la siguiente imagen, esta como plantilla base para comenzar a desarrollarla a nivel de hardware.

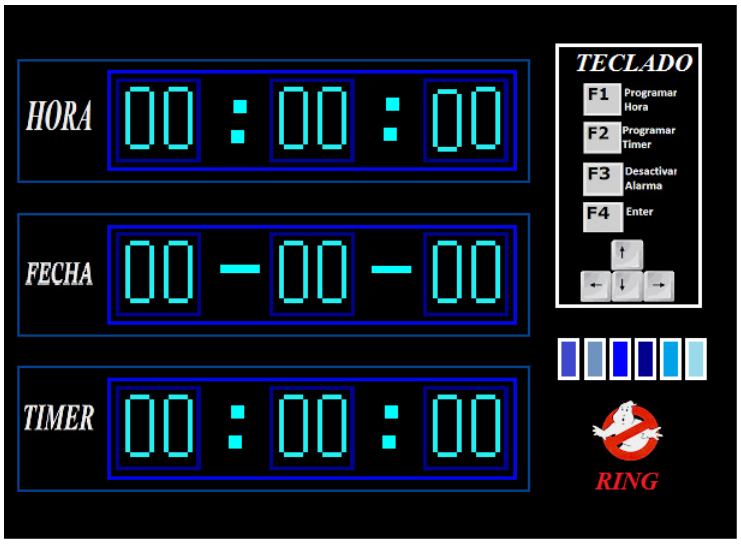

Figura 9: Interfaz VGA propuesta

Diagrama de primer nivel

Figura 10: Diagrama de primer nivel del controlador VGA

El circuito genera las señales de sincronización del circuito que son llamadas “hsync” y “vsync”, ambas son conectadas a la terminal VGA de la Nexys 2 respectiva, que es conectada al monitor para así controlar el recorrido horizontal y vertical del cañón de electrones. Posteriormente se crea un circuito que genere imágenes y patrones específicos según la posición del pixel; para este laboratorio se pide una interfaz de reloj en el monitor, en dicha interfaz se debe imprimir: hora y fecha actual, así como un cronometro con alarma. Las entradas del sistema son:

- 9 bytes provenientes del multiplexor desde la salida del rtc, cada uno contiene la información de: horas,minutos,segundos, día, mes y años. También los bytes restantes contienen la información del cronometro en: segundos, minutos y horas.
- Un botón de reset para limpiar la pantalla, y el CLK o señal de reloj para que el sistema trabaje a la velocidad establecida por el VGA.

Y como salidas del sistema, la señal RGB para indicar el color del pixel a imprimir, las señales de Horizontal sync y Vertical sync, para sincronizar los ciclos de impresión.

Diagrama de segundo nivel

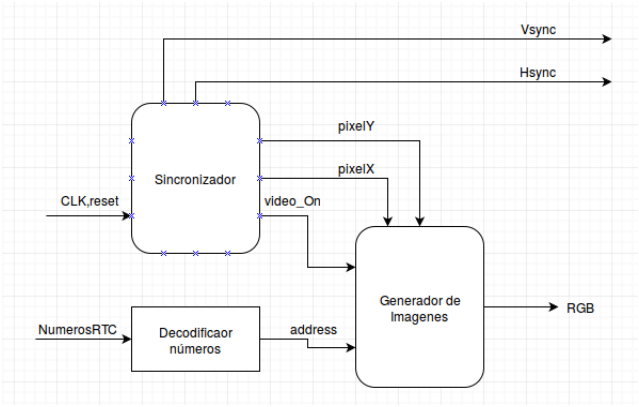

Figura 11: Diagrama de segundo nivel del controlador VGA

En la figura 14 se puede observar el diagrama de segundo nivel de diseño propuesto para el controlador, este mismo posee dos grandes etapas principales, el circuito de sincronía (para controlar el cañón del haz de electrones) y el generador de imagen (con la lógica necesaria para imprimir patrones en el monitor).

El circuito de sincronía se encarga de recibir una señal de reloj y adecuarla a los estándares VGA de la frecuencia de señal horizontal y vertical. Al mismo tiempo lleva una cuenta de estas señales para saber las coordenadas X y Y del píxel que se está imprimiendo, para luego enviarlas al Generador de Imagen. También envía la Señal de Impresión que se encarga de des habilitar la selección de color en los bordes de la pantalla donde no se despliega información. Por último, recibe una señal de Reset la cual controla la señal de impresión para des habilitar la selección de color en toda la pantalla.

El Generador de imagen tiene como función decidir el color del píxel que se va a pintar, dependiendo de las coordenadas X y Y de este, puede ser el color del fondo, los números provenientes del RTC o los diseños extra y decoraciones en la pantalla. Para esto, se usan multiplexores, decodificadores y además un módulo de memoria para recordar el patrón de las Letras y números que se deben desplegar.

Diagrama de tercer nivel del controlador

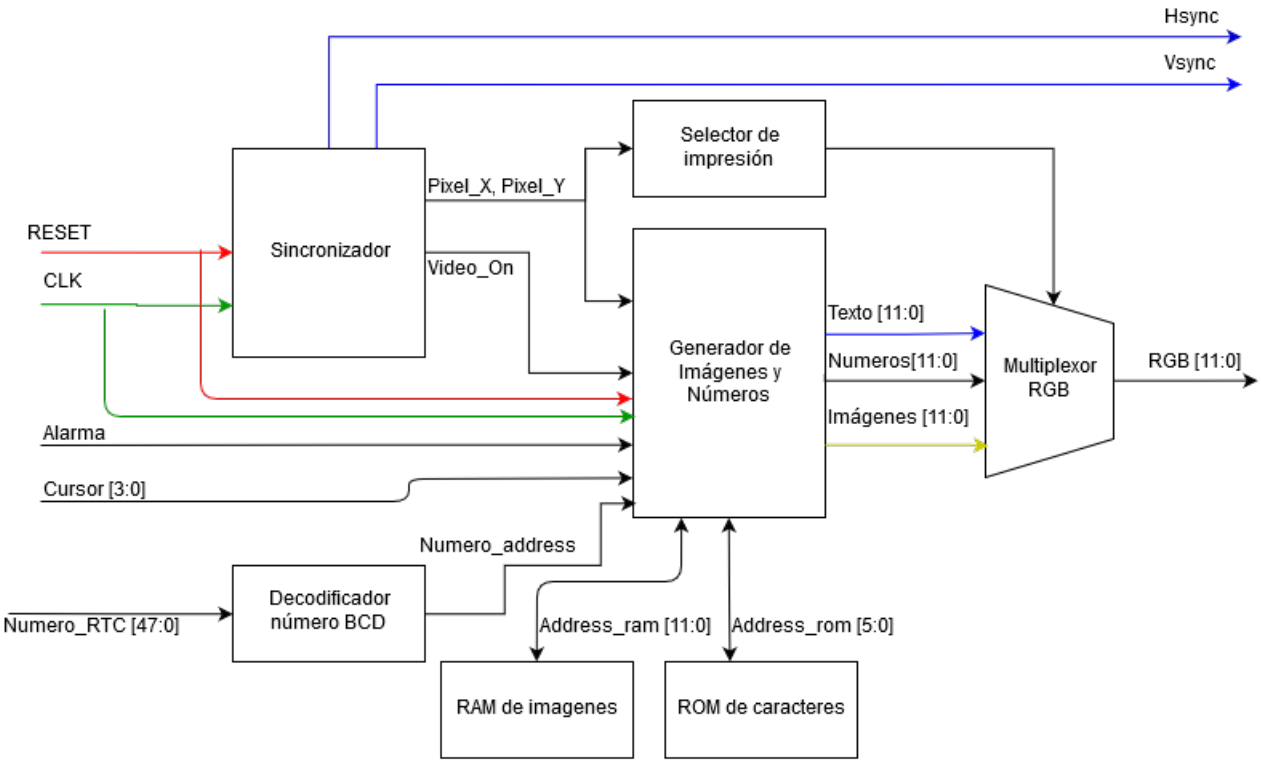

Figura 12: Diagrama de tercer nivel del controlador

El diagrama de tercer nivel del controlador VGA se observa en la figura anterior. Para realizar este control de la pantalla VGA se tienen los módulos que se detallan a continuación: Sincronizador: bloque encargado de generar las señales de sincronía del cañón de electrones del monitor, las cuales son Hsync y Vsync, sincronización horizontal y vertical respectivamente. Para seguir las normas del diseño sincrónico, una sola señal de reloj, dentro de este módulo se generan varios contadores para generar señales de activación a 25MHz, que es la tasa de velocidad de los pixeles, y así cumplir con las reglas de diseño. Además, para conocer la posición actual del haz en la pantalla se generan como salidas extra $\text{Pixel}_Y y Pixel_X, quesonbusesdedatosconelvalordelacoordenadaactualenelmonitor.Ta$

Generador de imágenes y números: este módulo es sumamente extenso ya que aquí se generan los valores binarios para formar un patrón de pixeles legible en la pantalla, a nivel general se encarga de tomar los datos de las memorias ram y rom para imprimirlos en la coordenada de pantalla correspondiente, así como los patrones generados por módulos menores con detalles para la pantalla como: animaciones, bordes decorativos. Recibe varias señales de entrada como lo son: $\text{video}_o n, parasaberdóndesepuedeimprimir; Pixel_Y y Pixel_X, paracolocarlasimágenes; a$

Selector de impresión: bloque encargado de seleccionar que se debe imprimir en la pantalla, tiene guardados en registros los parámetros generales de coordenadas definidos por el diseñador correspondientes para cada tipo de impresión, ya sea una imagen, texto o números.

Decodificador número BCD: este decodificador se encarga de generar la dirección para la memoria ROM correspondiente con el carácter numérico a imprimir en pantalla, la salida de este debe ser entrada del módulo generador de imagen, debido a que este es el que decide que patrón imprimir.

Rom de caracteres: memoria con los patrones binarios correspondientes a un número o letra, estas direcciones

son accesadas por parte del módulo generador de imagen, de acuerdo al valor que recibe desde el decodificador BCD.

Ram de imágenes: memoria ram con todos los patrones de pixeles para formar las imágenes que se observaran en la pantalla. Cuando el valor de $Pixel_Y$ y $Pixel_X$ coincide con el establecido por los parámetros de impresión, se accesa dicha m

Multiplexor RGB: la salida final con el pixel a imprimir se genera en este módulo, selecciona de acuerdo al dato que recibe del módulo seleccionador, y así decide cual bus de datos RGB se debe imprimir.

5. Para el Controlador del Teclado

En este proyecto se utilizó un teclado PS/2 para el ingreso de datos al sistema. Antes de detallar los niveles de diseño propuestos se debe comprender primero el funcionamiento del teclado. Este mismo posee dos señales: una de reloj y otra de dato; la transmisión de datos se realiza de manera serial, la siguiente imagen muestra de mejor manera como se transmiten estos datos en formato serie.

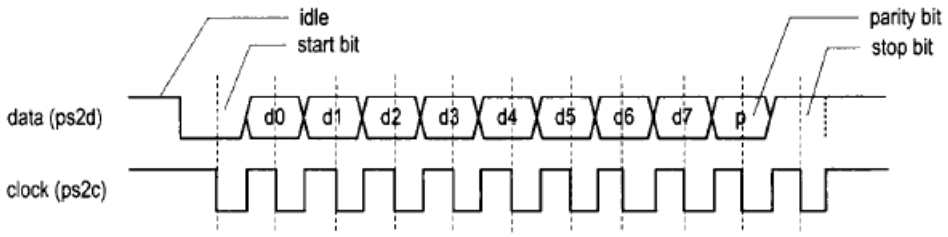

Figura 13: Protocolo de comunicación PS/2

Ambas señales al no existir interrupción alguna por parte de una tecla presionada se encuentran en valor lógico alto, cuando se presiona una tecla el valor del bit 'data' cambia de valor lógico, en este caso a un valor de 0, para indicar que la trama comienza, luego se envía un byte con la codificación correspondiente a la tecla para luego finalizar con un bit de paridad y seguido de uno de fin, todo esto al ritmo del reloj del teclado. Es importante considerar que las lecturas de datos se realizan en los flancos negativos del reloj. Otro punto importante a tomar en cuenta es que las entradas provenientes del teclado son de 'colector abierto', lo que significa que el valor lógico alto en realidad se trata de una alta impedancia, por lo que se debe colocar un resistor de Pull-Up en ambos pines, dato y reloj.

Diagrama de primer nivel

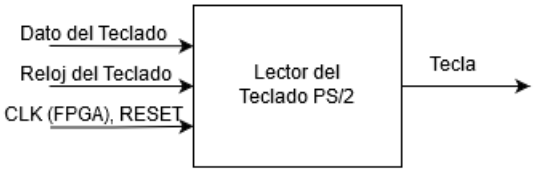

Figura 14: Diagrama de primer nivel del lector de teclado

Para la lectura de datos provenientes del teclado tipo PS/2 se propone el diagrama anterior de manera general. Las señales dato de teclado y reloj de teclado son propias de un teclado tipo PS/2, y la señal clk y reset son propias de sistema, en este caso una placa con una fpga modelo Nexys4.

Este módulo se encarga de recibir los datos para posteriormente enviar la tecla correspondiente al Picoblaze para su debido trato, este genera una señal llamada tecla que tomar un valor lógico alto cuando una tecla esta presionada, siempre y cuando esta sea alguna de las utilizadas dentro del sistema, por ejemplo, la tecla 'Enter' si genera un cambio mientras que la tecla F1 se ignora.

Diagrama de segundo nivel

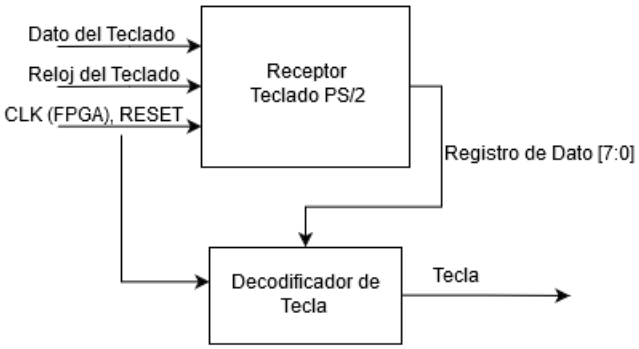

Figura 15: Diagrama de segundo nivel del lector de teclado

En este nivel se constituye por dos módulos los cuales se describen a continuación:

Receptor PS/2: módulo encargado de recibir los datos del teclado para colocarlos en un registro correspondiente, este registro es de 8bits, realmente es una máquina de estados que se encarga de leer datos en cada flanco negativo del reloj del teclado, cada uno de estos flancos coloca el dato correspondiente en el registro, cuando se llene el registro con el valor actualizado se genera una señal de control indicando que se acabó el proceso de lectura.

Decodificador de Tecla: recibe como entrada el registro del módulo receptor PS/2 para decodificar este valor de acuerdo a las teclas que se utilizan por parte del picoblaze. Las teclas propuestas a utilizarse son: Q, W, E, ENTER y las todas las flechas. Cuando el byte contenga la codificación correspondiente a una de estas teclas se envía una señal al Picoblaze indicando la letra correspondiente que fue presionada. Esta señal toma un valor lógico alto solo si la tecla esta presiona. Por lo tanto, en el bus de salida Tecla, hay 8 bits cada uno corresponde a cada una de las teclas que se utilizan, las cuales fueron detalladas anteriormente.

Datos y Resultados

1. Simulaciones

Picoblaze y Control General del RTC:

Como se explicó anteriormente, existen protocolos pre-establecidos de acceso en modo escritura y lectura para el RTC. Si el programador desea interactuar con el reloj, debe necesariamente respetar los tiempos definidos por el fabricante en dicho protocolo. Estas son las funciones más importantes del sistema, pues en ellas se

basan las demás rutinas, por esta razón se realizó una simulación de estas subrutinas y así poder comprobar el correcto funcionamiento en cuanto el cumplimiento a tiempos mínimos de acceso y la manipulación de señales de control del RTC en modo lectura y escritura. Para este caso, no se muestran imágenes de dichas simulaciones pues ya se presentaron en el proyecto anterior, entonces, se toma por un hecho que el tema de tiempos críticos se cumple debidamente. Del mismo modo, se mantuvo el código para la sección de control de Usuario por botones (ahora teclas). Para las simulaciones de dicha etapa hay que referirse nuevamente al informe del proyecto anterior a este.

Es importante comprobar el funcionamiento adecuado del Picoblaze en la escritura, lectura y manipulación de los registros en general. Como lo presentado son simulaciones post-síntesis, se mostrará el movimiento de registros en las máquinas de inicialización y stand-by, donde se observa claramente la escritura y toma de direcciones y datos desde el Picoblaze hasta el RTC y viceversa si corresponde.

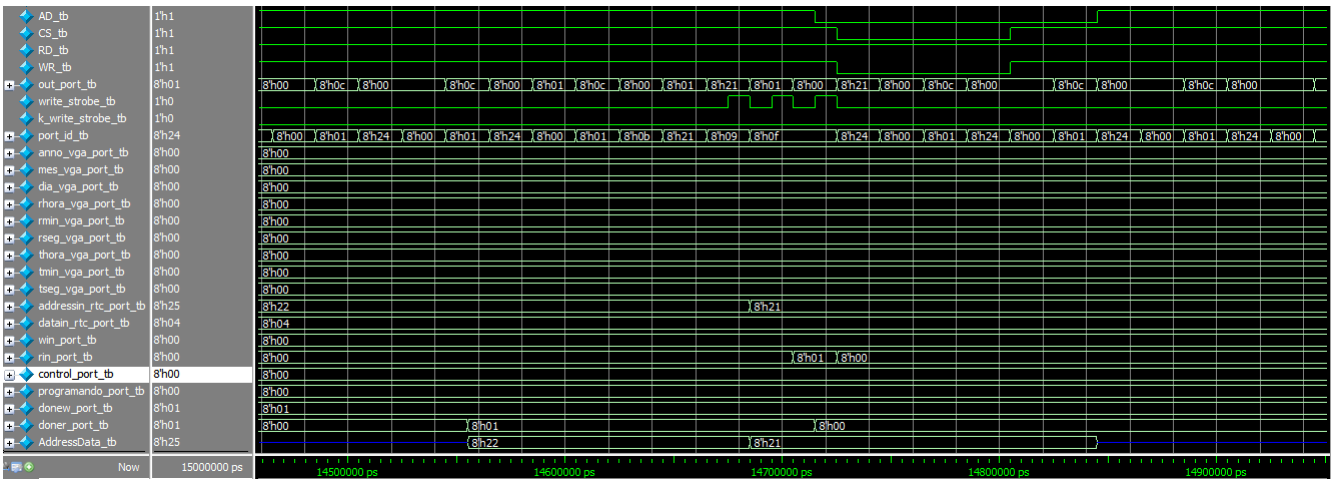

Figura 16: Movimiento de Registro en PicoBlaze al leer del RTC

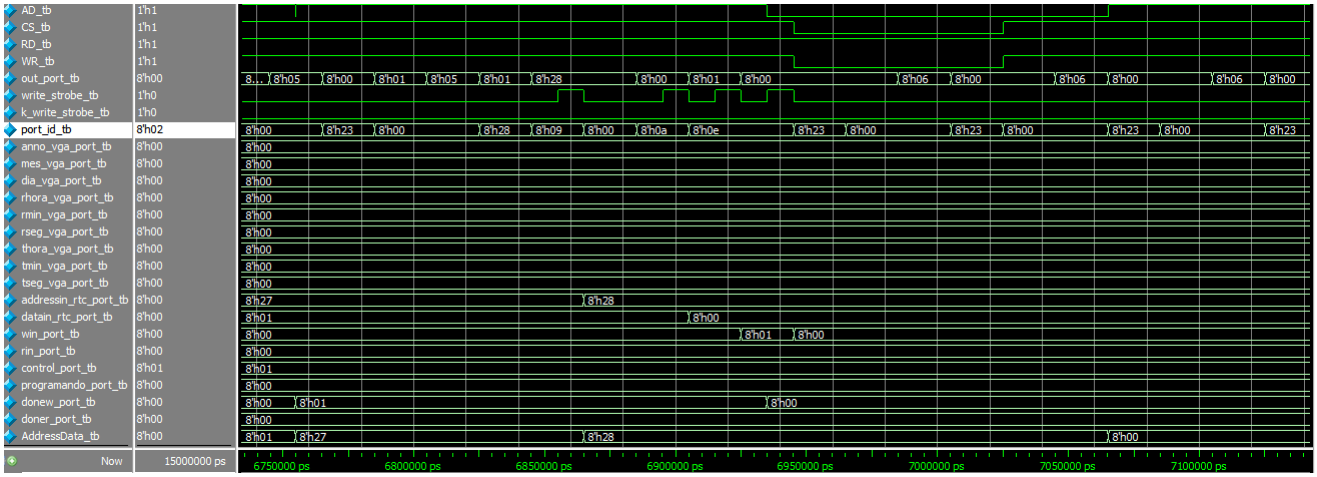

Figura 17: Movimiento de Registro en PicoBlaze al escribir en el RTC

Controlador de Teclado:

Es importante comprender el protocolo de comunicación PS2 y asegurar el correcto funcionamiento para crear una interfaz física con el usuario estable. Los siguientes diagramas de tiempos corresponden a las simulaciones

realizadas para comprobar el correcto funcionamiento del teclado, primero se tiene la simulación de la recepción de datos y carga en el registro correspondiente, posteriormente se tiene la simulación de la decodificación de este dato recibido. En la recepción de datos interesa observar las señales rx done tick y dout, las cuales respectivamente corresponde a la señal de finalización de carga de datos y el registro con los datos obtenidos, se puede observar que cuando la señal rx done tick pasa a un valor de 1, se carga el nuevo dato en dout, en este caso un 15 en formato hexadecimal.

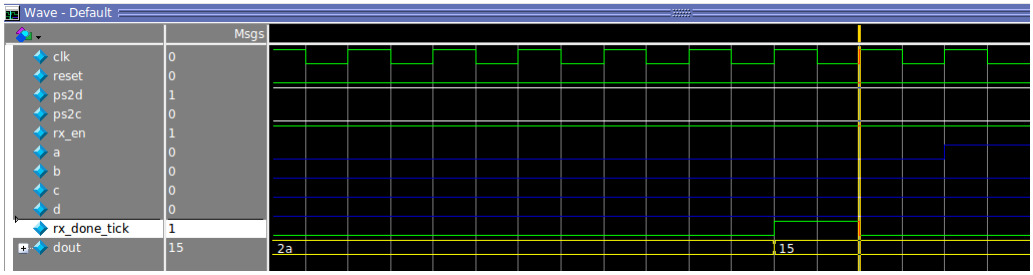

Figura 18: Recepción de datos del teclado

En la decodificación de tecla, un valor de 15 en el registro corresponde a la tecla 'Q', en este caso el bit 'a' cambia a un 1 lógico cuando se detecta el valor de 15 en el registro dout.

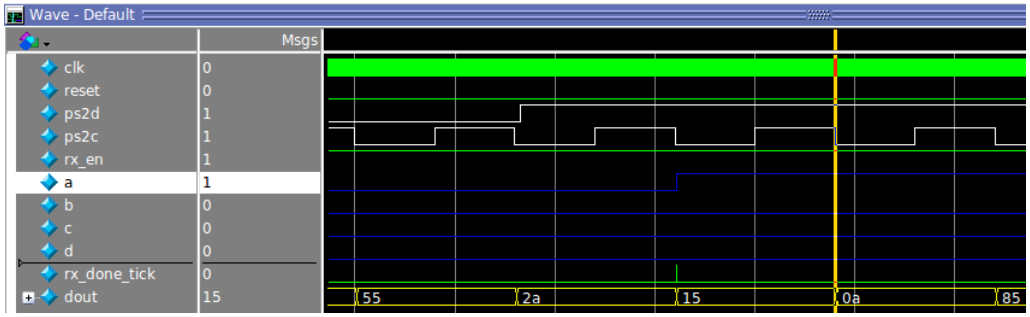

Figura 19: Decodificación de datos del teclado

Controlador VGA:

Debido al proceso largo, complicado y tedioso de revisar las combinaciones de color para los bits rojo, azul y verde para saber si se están logrando las impresiones adecuadas, se aprovecho la virtud que posee el lenguaje Verilog para realizar pruebas y simulaciones. Mediante un testbench que genera un archivo de texto con la codificación hexadecimal que posee el bus RGB en cada pixel a lo largo del recorrido de la pantalla. El archivo de texto se puede manejar mediante lenguajes de alto nivel como: python,c++ o matlab para lograr crear una imagen de 640x480 pixeles a partir de este archivo. Los resultados obtenidos utilizando Python se muestran en la figura 19.

La simulación de la sincronización con los tiempos del protocolo VGA para un correcto despliegue de información en la pantalla se asumen debido al primer proyectodel curso.

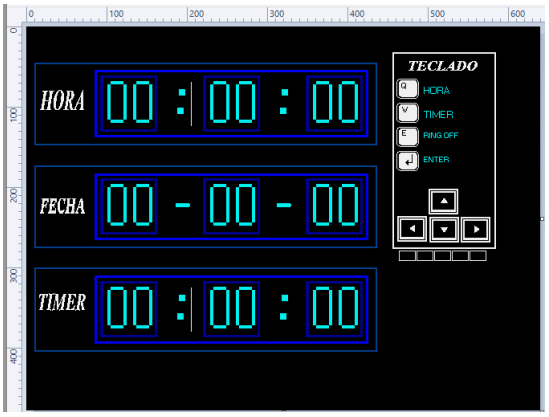

Figura 20: Simulación de la pantalla VGA

Los resultados obtenidos en ModelSim para los bits de salida para la señal 'rgb' concuerdan en su totalidad con lo esperado a nivel de diseño.

2. Mediciones

Para este apartado se van a tomar en cuenta el resultado del análisis del consumo de potencia del dispositivo y consumo de recursos dado que en el proyecto anterior quedo claro con las mediciones realizadas que se cumplía con el control de tiempos críticos para el adecuado acceso al RTC.
Para verificar la correcta sincronización entre la actualización de datos del registro y la lectura de estos mismo por el controlador VGA, basta con ver la interfaz en el monitor y comprobar la correcta impresión de los datos, así como la estabilidad de la imagen.

Reporte de área utilizada

Slices	Cantidad usada	Total Disponible
Slice registers	842	126800
Slice LUT's	4050	63400
Pares de LUT's flip flops	4293	na
IOB's	33	210

Cuadro 1: Consumo de área para slices lógicos

Primitivas y cajas negras	Cantidad
Flip Flops	842
FSMs	7
IO buffers	32
Clock buffers	1
Multiplexores	343
Comparadores	269
RAMs	36
Sumadores y Restadores	16

Cuadro 2: Estadísticas de consumo de área para primitivas y cajas negras

Reporte de consumo de potencia

Total (mW)	Dinámica (mW)	Quiescente (mW)
88	0	88

Cuadro 3: Potencia total consumida

Reporte de tiempo

Reloj	Frecuencia Máxima
CLK	171.89 Mhz

Cuadro 4: Máxima frecuencia de trabajo obtenida para el sistema

Señal	Delay (ns)
Display	6.635
Reset	5.139
Escritura	1.816

Cuadro 5: Peores delays obtenidos

Análisis de datos y resultados

Nuevamente, para el controlador VGA se necesitaba adecuar el sistema para que el circuito de sincronización cumpliera con los tiempos establecidos por el protocolo estudiado el proyecto anterior. La diferencia es que en este caso, el sistema está diseñado sobre la plataforma Nexys 4, quien provee un reloj a una frecuencia de 100 MHz. De igual manera, si no se sincroniza correctamente, existe riesgo de retraso de señales o pérdida de información. La solución se basó igualmente en habilitar una señal que pausara y reanudara la cuenta del reloj. Como era de esperarse, al realizar las mediciones el resultado era el correcto.

Para el controlador VGA, su construcción se basó en los ejemplos descritos en el libro "FPGA Pritotyping by Verilog Examples", de Pong Chu [1], tanto para la sincronización, como para la creación de la interfaz. En este proyecto se utilizaron algunas imágenes para el desarrollo de la interfaz. Estas fueron hechas a partir de gestor

de imágenes en su mayoría. Al tomar imágenes de la web y ubicarlos en algún sector dela VGA, se generaban problemas de deformación de imágenes y decoloración; al hacerlas en el gestor, se eliminaron dichos problemas y se obtuvo el resultado deseado en la interfaz.

De la misma forma, el módulo receptor de teclas fue basado en los códigos descritos en el Pong Chu [1], a partir de este se hicieron las modificaciones pertinentes, sujetas al funcionamiento del circuito general.

El tiempo de ejecución de las instrucciones en el Picoblaze era un tema a tomar en cuenta, pues podía generar problemas a nivel de implementación cuando se controlara los tiempos de acceso. Por ejemplo, en el Picoblaze, dos instrucciones para colocar una dirección o recibir datos (en el caso de la lectura) se ejecutan en 40 ns, mientras todo un ciclo para poder leer correctamente en el RTC tomaba más de 200 ns. Para solucionar este problema, se creó un ciclo luego de cada instrucción, en este se leía el dato indicador de final de lectura y se comparaba, si este estaba activo, un cambio de estado ocurría y se rompía dicho ciclo. Las diferencias entre las frecuencias de relojes utilizados para la VGA y el resto de sistema en general permite tomar este tipo de alternativas para solucionar problemas.

Por último, se presentaron problemas de volumen enla implementación de volumen; al colocar la salida en la salida de audio de la Nexys 4, se obtenía una señal con bajo voltaje, por tanto, era difícil escuchar. Este problema es probablemente debido a errores en la generación de señales compatible con dicha salida.No obstante, se solucionó a la hora de presentación del proyecto utilizando una salida del PMOD XDAC brindada por la plataforma utilizada, así, la salida.

Hojas de datos de unidades desarrolladas

Se utiliza el reloj de 100 MHz proporcionado por la Nexys 4, en la figura 23 se muestra la distribución de pines en la FPGA, asignado según la hoja de datos de esta plataforma, así como los pines respectivos en el RTC. Las salidas en la FPGA son entradas en la RTC.

Señal	Designación	Pin FPGA	Función	Pin RTC
clock	input	E3	Reloj	N/A
reset	input	P4	Reseteo del Sistema	N/A
ps2d	input	B2	Dato serial del Teclado	N/A
ps2c	input	F15	Reloj del Teclado	N/A
rx_en	input	V10	Enable del teclado	N/A
AddressDatao [0]	inout	G14	Datos y direcciones	2
AddressDatao [1]	inout	P15	Datos y direcciones	3
AddressDatao [2]	inout	V11	Datos y direcciones	5
AddressDatao [3]	inout	V15	Datos y direcciones	6
AddressDatao [4]	inout	B13	Datos y direcciones	23
AddressDatao [5]	inout	F14	Datos y direcciones	25
AddressDatao [6]	inout	D17	Datos y direcciones	26
AddressDatao [7]	inout	E17	Datos y direcciones	27
IRQ	input	V10	Enable del teclado	8
AD	output	K16	Control escritura y lectura en RTC	7
CS	output	C17	Control escritura y lectura en RTC	20
RD	output	E18	Control escritura y lectura en RTC	22
WR	output	D18	Control escritura y lectura en RTC	21
speaker	output	A13	Salida Analógica de Sonido	N/A
rgb [0]	output	D8	Bit para color	N/A
rgb [1]	output	D7	Bit para color	N/A
rgb [2]	output	C7	Bit para color	N/A
rgb [3]	output	B7	Bit para color	N/A
rgb [4]	output	A6	Bit para color	N/A
rgb [5]	output	B6	Bit para color	N/A
rgb [6]	output	A5	Bit para color	N/A
rgb [7]	output	C6	Bit para color	N/A
rgb [8]	output	A4	Bit para color	N/A
rgb [9]	output	C5	Bit para color	N/A
rgb [10]	output	B4	Bit para color	N/A
rgb [11]	output	A3	Bit para color	N/A
hsync	output	B11	Sincronización horizontal	N/A
vsync	output	B12	Sincronización vertical	N/A

Figura 21: Asignación de pines de entradas y salidas en la FPGA

Conclusiones y recomendaciones

1. Conclusiones

- La utilización de un microcontrolador como el Picoblaze facilita la descripción del comportamiento del circuito digital, ya que ejecuta las instrucciones en secuencia y no en paralelo, además centraliza el control y el flujo de datos de los periféricos.
- La correcta descripción de hardware en lenguaje Verilog es esencial para el buen funcionamiento de la interfaz entre PicoBlaze y los puertos de los periféricos.
- Para sincronizar el proceso de control y el despliegue de información, es importante analizar los tiempos de ejecución y propagación de señales del sistema.
- Al utilizar herramientas de software como Xilinx ISE, Modelsim y de hardware como FPGA se simplifica el diseño y la síntesis del prototipo del sistema digital.

2. Recomendaciones

- Es importante estudiar y comprender el funcionamiento del Real Time Clock (V3023) y los requerimientos del protocolo para que el sistema sea totalmente compatible y asegurar un buen funcionamiento.
- Diferenciar los bloques combinacionales y los secuenciales para describirlos correctamente según especifica el Hardware Description Language.
- Nombrar los puertos de los periféricos en el programa de ensamblador para facilitar la escritura e interpretación del código.
- Usar lenguajes de alto nivel para facilitar la simulación de los diseños implementados de la interfaz de usuario y ahorrar el tiempo de Síntesis e Implementación en la FPGA.
- Utilizar imágenes codificadas en mapas de bits logra la obtención de una interfaz VGA mas
- Realizar simulaciones de cada módulo por aparte y constantemente, esto facilita la identificación de errores de diseño.
- Llevar buen control de las versiones de los archivos y programas que se van modificando, esto ayuda a la coordinación del grupo de trabajo y la evaluación del profesor. En este proyecto se utilizó Github como registro.

Referencias

[1] Chu, Pong P. (2008) FPGA prototyping by Verilog examples. Xilinx SpartanTM-3 Version, John Wiley Sons, Inc.