

Proyecto	1	Página	1/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

Resumen

Se debe realizar un controlador para realizar la lectura y escritura del módulo RTC V3023. Los datos del sistema deben poder ser desplegados en un monitor LCD mediante el protocolo VGA. Por lo cual, se deben realizar controladores para la RTC y la VGA, además estos y el funcionamiento en general del sistema deben ser controlados por el microcontrolador Picoblaze de Xilinx. Asimismo, se debe poder ajustar los parametros de hora, fecha, y cronometro, en forma descendente, por medio de un teclado y utilizar la salida de audio disponible en la Nexys 4.

Introduccion

Este proyecto consiste en realizar un controlador de módulos RTC (Real Time Controller), específicamente para el módulo V3023. Este controlador será capaz de escribir y leer dicho módulo para obtener parámetros de reloj, cronómetro y alarma.

Asimismo, para poder desplegar la información relevante de los parámetros anteriores, se conectará un monitor LCD mediante el protocolo VGA. Por otro lado, para poder programar y dar instrucciones al circuito, se deberá usar un teclado, además de implementar la salida de audio del a Nexys 4 para notificar al usuario del fin de la alarma.

Todos los módulos mencionados deben ser controlados por el microcontrolador Picoblaze de Xilinx.

Finamente, el sistema realiza la correcta comunicación y lectura de la RTC, además de permitir al usuario visualizar y acceder a la misma de una forma eficiente.

1. Objetivos

- Diseñar un controlador de RTC que permita leerlo y programarlo mediante una interfaz de usuario que consiste en un teclado, un monitor comunicado a través del protocolo VGA y un microcontrolador.
- Implementar el control general del sistema mediante el uso de el microcontrolador Picoblaze.
- Investigar y conocer el lenguaje de programación del Picoblaze.
- Asegurar la sincronía de tiempos entre el Picoblaze, la RTC, y los demás módulos del circuito.
- Realizar la conversión del código del Picoblaze para su implementación en el lenguaje Verilog.
- Investigar el funcionamiento del módulo RTC y el protocolo de comunicación del mismo.
- Investigar el funcionamiento del módulo de teclado y su protocolo de comunicación.
- Diseñar un controlador para el módulo RTC, cuyo bus de datos y direcciones están multiplexados.
- Cumplir con las reglas de temporizado del sistema, en especial, con el protocolo de comunicación del módulo RTC.

Proyecto	1	Página	2/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

- Combinar el controlador de RTC con un controlador VGA, por medio del Picoblaze, para poder desplegar la información del módulo al usuario. Este módulo VGA será adaptado del proyecto anterior.
- Desarrollar un banco de pruebas (testbench) para poder emular el comportamiento del módulo RTC con la finalidad de comprobar el funcionamiento del circuito controlador.
- Desarrollar un banco de pruebas (testbench) para verificar el correcto trasiego de datos del teclado hacia el Picoblaze.
- Implementa la salida de audio mono incluida en la Nexys 4.

2. Descripción del sistema

El sistema se divide en 5 módulos, los cuales son teclado, Controlador de RTC, controlador de VGA, salida de audio y microcontrolador Picoblaze. El circuito está diseñado de manera que el Picoblaze funcione como controlador principal del sistema, es decir, se encarga de intercomunicar los demás módulos entre sí. Específicamente permite realizar el acceso a la RTC, por medio de su respectivo controlador, ya sea para lectura o la escritura de nuevos datos enviados por el teclado al microcontrolador. Además controla el envío de los datos que se despliegan en la VGA y activa o desactiva el módulo de audio.

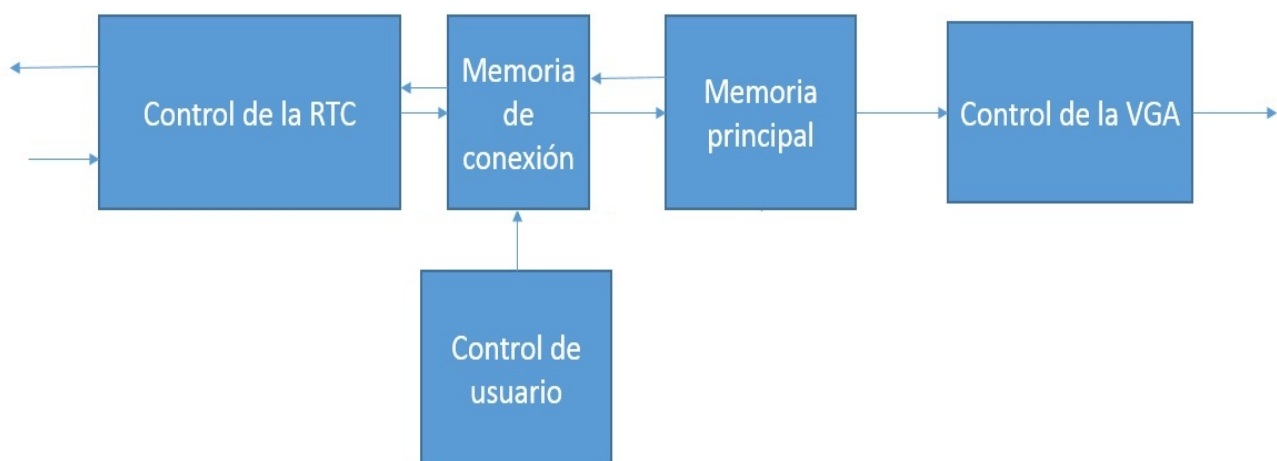


Figura 1: Diagrama de módulos principales del sistema.

2.1. Controlador de la pantalla

Para el periférico de despliegue de datos se ha solicitado emplear un monitor LCD controlado mediante el protocolo VGA (Video Graphics Array), con la finalidad de aprovechar el desarrollo del proyecto

Proyecto	1	Página	3/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

anterior y adaptando algunas cosas para poder incorporar datos dentro de la imagen que se desplegará en el monitor. Este diseño está basado en el mismo implementado para el proyecto anterior, con la salvedad de que, en este diseño, los registros no son actualizados de forma autónoma y son modificados por el microcontrolador PicoBlaze. Esto varía la salida del bus Adr en el diseño anterior y se cambia por el PortID.

2.1.1. Diagrama de primer nivel

Para efectos del circuito, este debe admitir datos de las variables del tiempo emitidas por el Picoblaze, que son emitidos por el PortID y el DataOUT del mismo microcontrolador. Para ello, se ha puesto un banco de 16 registros (11 aprovechados) direccionados mediante un bus de direcciones de 4 bits y 8 bits de datos. Asimismo, se deben colocar las entradas y salidas pertinentes del protocolo, que son las salidas de sincronía y las salidas de color. Asimismo, este circuito recibe la señal de IRQ procedente de la RTC para poder señalar el cúlmimo de la cuenta del timer.

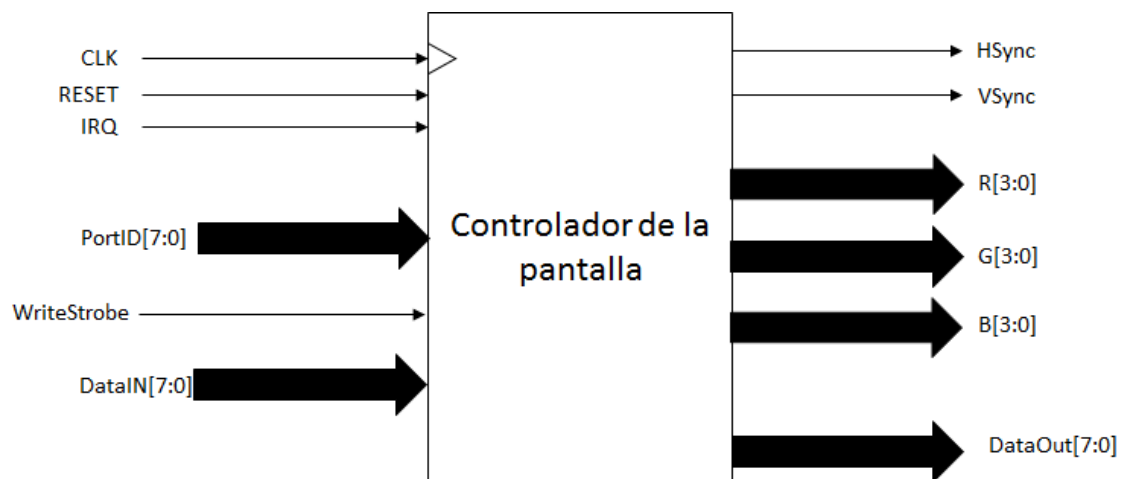


Figura 2: Diagrama de primer nivel VGA

A como es posible observar en la figura 2, existen dos buses que permiten obtener la información que será desplegada en la pantalla. El bus "PortID" permite direccionar el registro de la memoria de datos al registro que va a ser refrescado y el bus "DataIN" recopila la información del registro direccionado mediante "PortID". Esta comunicación es de solo lectura y solo será efectuada durante la sincronía vertical, que no contiene imagen para evitar glitches y cambios en el registro inesperados que vayan a afectar la nitidez de la imagen.

Asimismo, este circuito tiene un bus de salida "DataOUT" que es multiplexado hacia el Picoblaze para avisar de los períodos de sincronía vertical para poder actualizar la VGA.

Para concluir este apartado, la interfaz se definirá mediante imágenes para cargar una interfaz agradable al usuario, esto aprovechando la capacidad de síntesis de memoria ROM en el entorno de Xilinx.

2.1.2. Diagrama de segundo nivel

Adentrando en el diseño, se pueden identificar tres grandes bloques que componen el control de pantalla, donde es posible generar los datos, generar las señales de posición y sincronía y las señales de los colores.

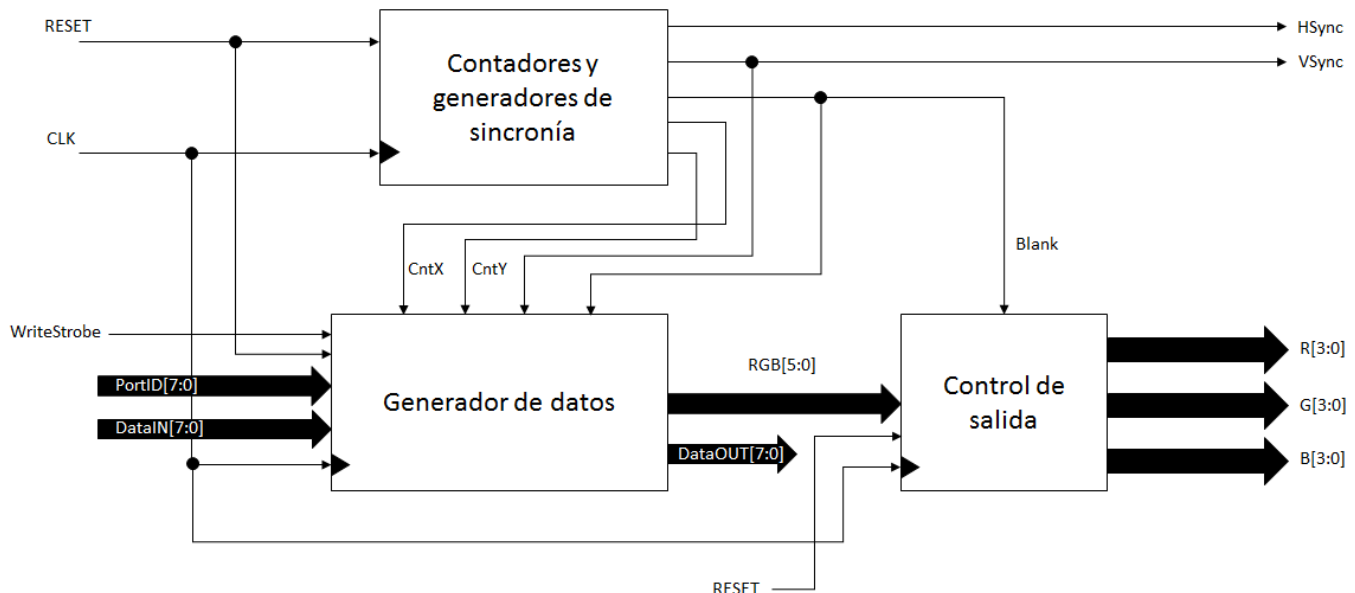


Figura 3: Diagrama de segundo nivel VGA

Como es posible apreciar en la figura 3, el primer bloque es el de "Contadores y generadores de sincronía", que permite generar las señales de sincronización del monitor LCD y, además, generar la posición de la pantalla en la que se ubica el cursor. Esto último es clave para la etapa de "Generador de datos", que decidirá los datos que serán pintados y las memorias que serán consultadas.

El bloque "Generador de datos" permitirá producir las señales de color primitivo (2 bits por color) de la imagen que será pintada en la pantalla con base a los datos que se registren mediante los buses "PortID" y "DataIN", cuyos datos se almacenarán en un buffer de memoria de video. Basado en los datos de este buffer, se determinará el dato que se ilustrará (número) mediante un posicionamiento en memoria. Los datos que saldrán en la dirección serán generados por un contador de 12 dígitos y su proceso de incremento estará condicionado por el VSync para poder actualizar el buffer en momentos de sincronía (Pasado hacia el microcontrolador), donde se no pintan pixeles. Por otro lado, las posiciones de memoria serán determinantes para consultar la memoria ROM que contendrá la interfaz y los números.

Por otro lado, el "Control de salida" permite decodificar el color en 2 bits en 4 bits mediante el duplicado. Asimismo, apaga los bits de salida en caso de entrar en etapa de sincronía, indicado mediante la entrada de "Blank". Esto evitará enviar datos en caso de ubicarse en un proceso de tiempo prohibido en la pantalla.

Proyecto	1	Página	5/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

2.1.3. Diagrama de tercer nivel

Profundizando aún más en el diseño, los bloques se pueden descomponer más haciendo evidente la presencia de una memoria ROM que contiene la información de la interfaz gráfica de usuario, que está subdividida en la memoria ROM de interfaz y la memoria ROM de los números, ya que todo el entorno es visual (véase figura 4).

Asimismo, el generador de datos está compuesto por un direccionador de memoria mediante punteros y un banco de registros que ayudan a tener los datos estables para su uso. Dependiendo de la posición del cursor en la pantalla (CntX) y (CntY) y los datos de los registros, es posible direccionar a un campo específico de la memoria ROM de números y hacer el intercambio entre ROM de interfaz y ROM de números.

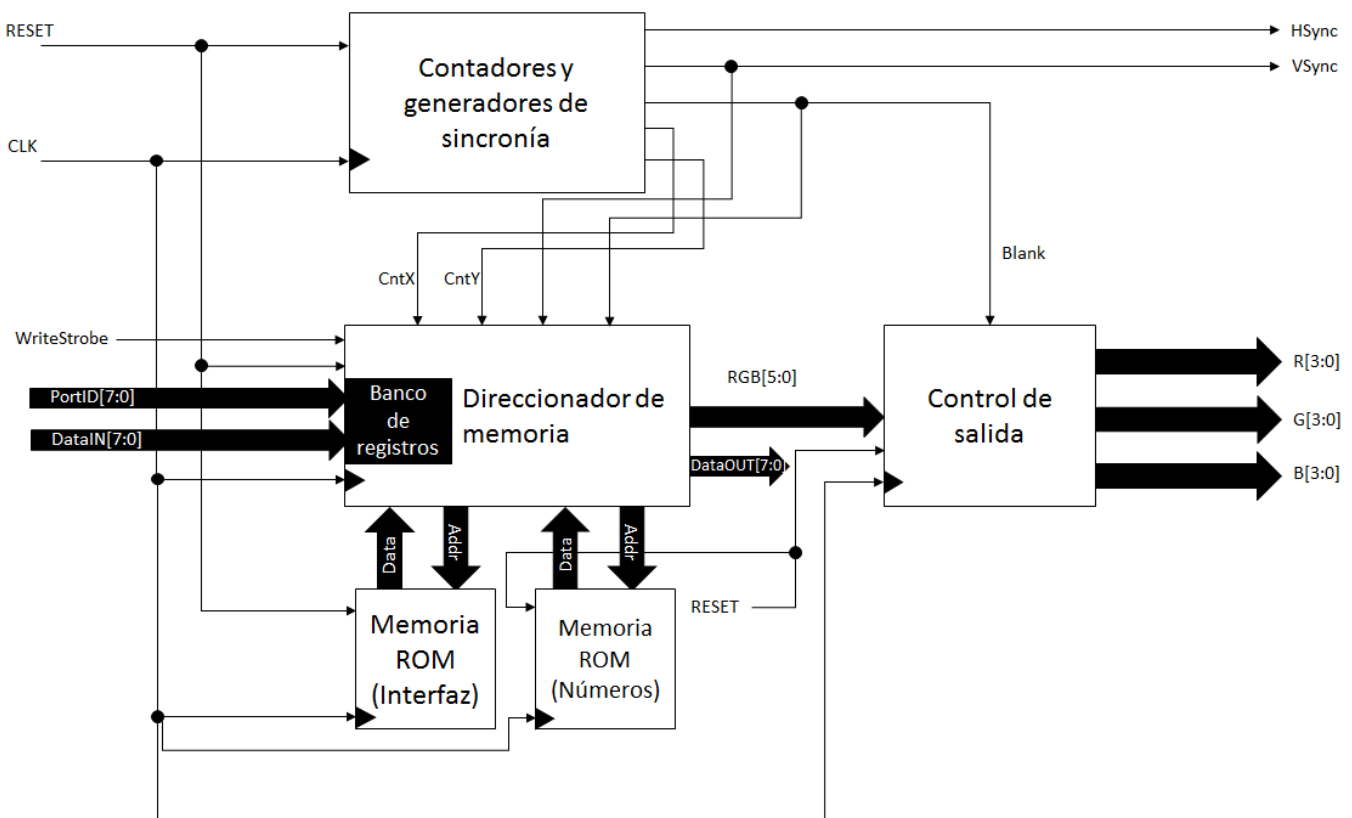


Figura 4: Diagrama de tercer nivel VGA

Por otro lado, se encuentran los contadores y generadores de sincronía, que están compuestos por dos contadores y un bloque combinacional. El primer contador gobierna el conteo horizontal (CntX) y, a su vez, regula la velocidad de conteo descartando los 2 primeros bits menos significativos, por lo cual, el contador es de 12 bits para abarcar las 800 líneas horizontales. Este contador envía una señal de conteo al contador vertical, quien gobierna la posición vertical (CntY). El bloque combinacional permite generar las señales VSync y HSync dependiendo de la magnitud del conteo CntX y CntY, así como, la señal de Blank, donde no debe existir señal de color por encima de 0V.

El Control de salida permite, a como se explicó en el apartado anterior, expandir los bits de color para poder generar una señal de 12 bits de color. Asimismo, es gobernado por el Blank que pone en ceros

Proyecto	1	Página	6/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

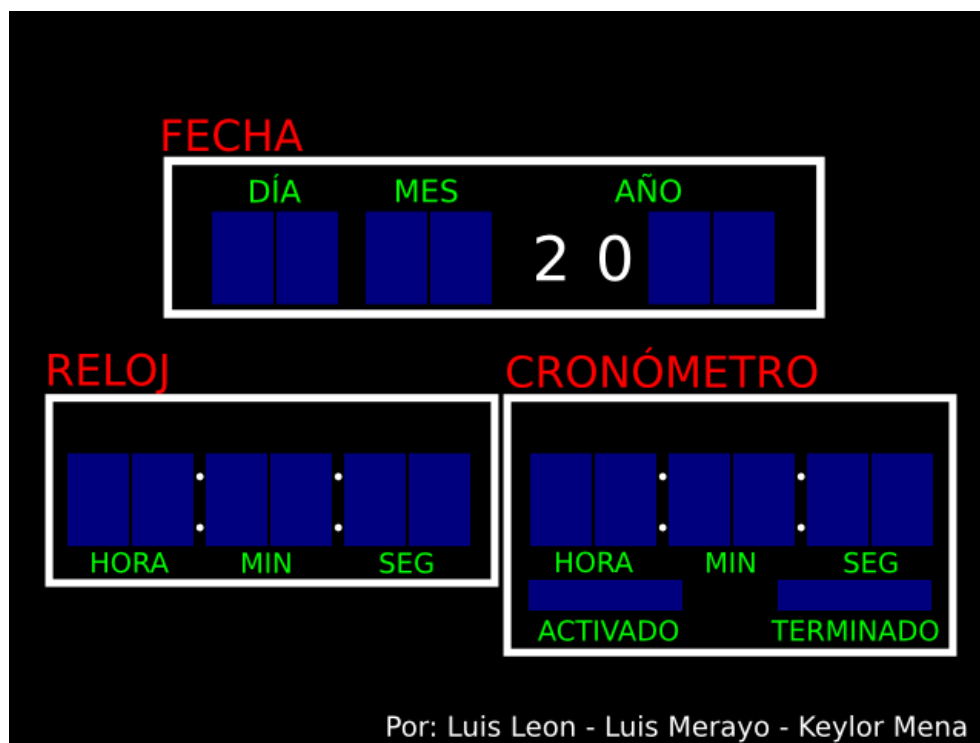


Figura 5: Interfaz de VGA

todos los bits de color en tiempos de zona prohibida.

Para finalizar, las memorias tienen un dimensionamiento de 307200 espacios de memoria para albergar la interfaz de usuario (640X480) (figura 5) y de 24000 espacios para albergar las imágenes de los números (figura 6). Los datos de estas memorias serán generadas por un script de Matlab y se cargarán mediante un archivo de texto con 6 bits.

Cabe destacar que, la interfaz tiene espacios azules para indicarle al sistema de punteros cuando debe hacer cambio de imagen por un número. Esto se hizo con la finalidad de ahorrar espacio en la Nexys y optimizar el código, ya que existieron problemas debido al rendimiento en tiempo de todo el bloque VGA.

2.2. Controlador de teclado

Una de las diferencias de este proyecto con respecto al anterior es la incorporación de un teclado con comunicación bajo el protocolo serial PS2. Este protocolo serial emplea un reloj generado por el periférico y una línea de datos que permite transmitir los datos en sincronía con el reloj del periférico.

Para efectos de uso, se han tomado en cuenta las siguientes teclas para poder usar, más adelante, el circuito:

Para las salidas del circuito, este guardará los valores de la selección del usuario hasta que haya un commit = 1 y un ReadStrobe que indique que los datos fueron leídos por el microcontrolador. Asimismo,



Figura 6: Números VGA

Tecla	Función
F1	Alterar fecha
F2	Alterar hora
F3	Alterar cronómetro
F10	Apagar alerta del cronómetro
F11	Activar cronómetro
F12	Detener cronómetro
Esc	Descartar y Salir
Enter	Guardar y Salir
Tab	Seleccionar Siguiente
0-9	Valores numéricos

hay una salida hacia los display de 7 segmentos para comprobar los datos.

2.2.1. Diagrama de primer nivel

El diseño básico es presentado en la figura 7, donde se indican las entradas y salidas del circuito. En estas, se presentan las direcciones del PortID procedente del Picoblaze, la salida de datos hacia el Picoblaze (DataOUT) y un indicador de lectura ReadStrobe, que permite comprobar al controlador de teclado que los datos fueron leídos y poder, así, borrar los registros del control. Por otro lado, es posible notar las entradas del teclado PS2 (PS2_CLK y PS2_DATA) y las salidas hacia los displays de 7 segmentos que permiten comprobar el funcionamiento del microcontrolador y del teclado.

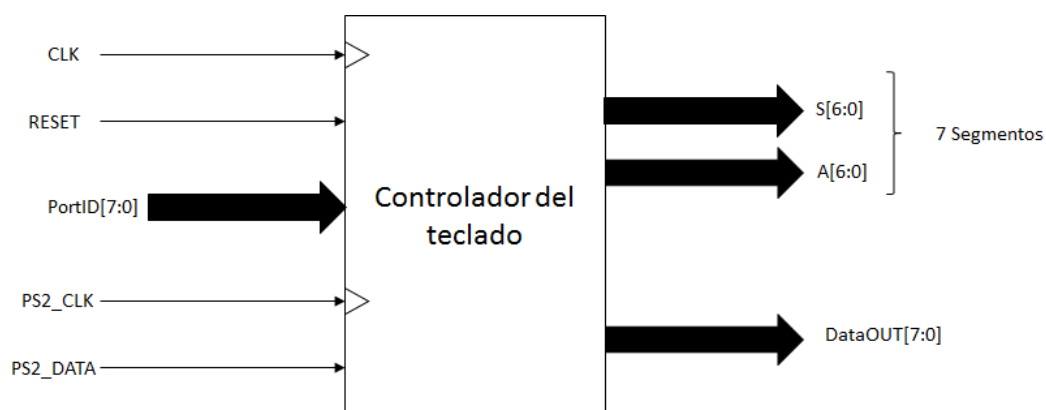


Figura 7: Diagrama de primer nivel KB

2.2.2. Diagrama de segundo nivel

Para la segunda iteración del diseño, se investigó acerca del funcionamiento del teclado y un código de ejemplo para un teclado que muestra los códigos de las teclas en los display de 7 segmentos [3]. Este código estaba diseñado para una Nexys 4 DDR, que, a como fue posible notar, es ligeramente diferente a la Nexys 4 que se emplea, actualmente.

Tomando en cuenta un diseño basado en el código de ejemplo, la iteración del diseño queda de acuerdo con la figura 8.

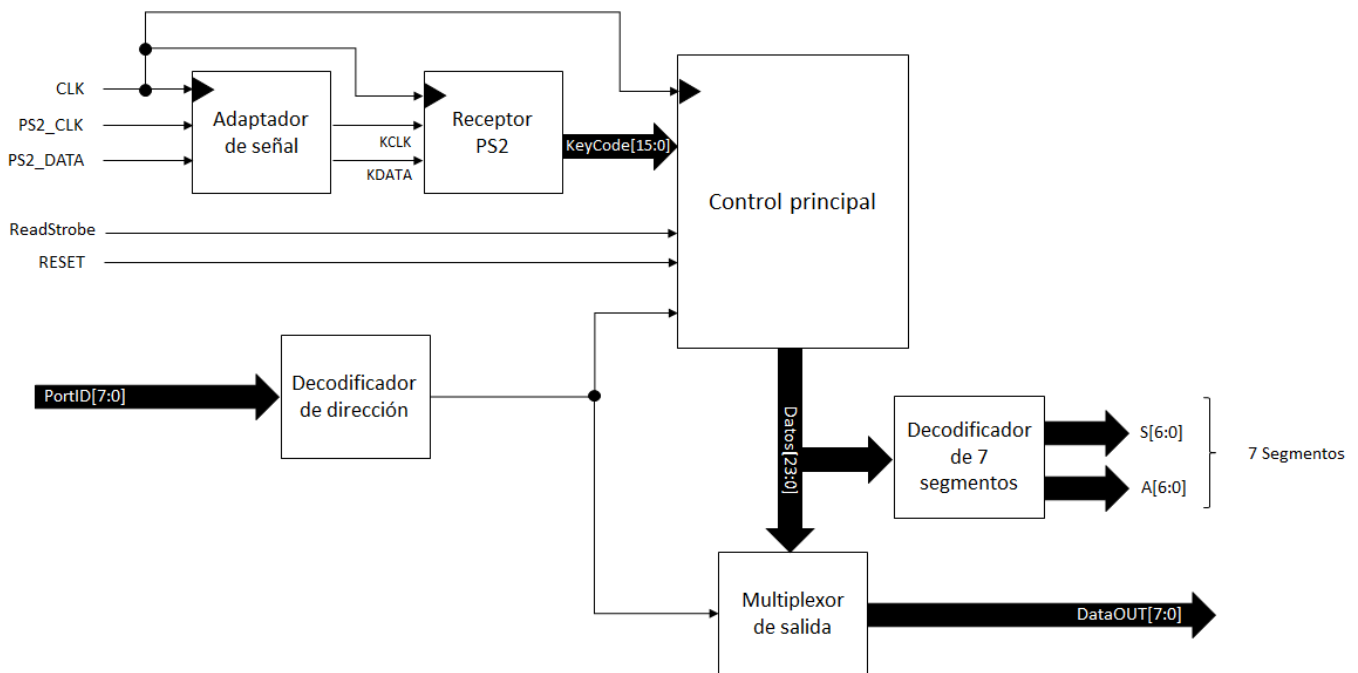


Figura 8: Diagrama de segundo nivel KB

Las etapas modificadas y basadas en el ejemplo son el Adaptador de señal, el Receptor PS2 y el Decodificador de 7 segmentos. La alteración de estas radica en las conexiones y el ancho de los buses de datos de I/O de cada segmento. Sin embargo, el Adaptador de señal se ha tomado intacto, ya que es esencial para el Receptor PS2 y evitar posibles rebotes causados por desconexiones o posibles fluctuaciones eléctricas.

El Receptor PS2 permite convertir el dato recibido bajo formato serial a un formato paralelo de 32 bits (16 bits aprovechables), que permiten tener en cuenta los valores de las teclas en una forma de pila para el Control de Teclado (KeyCode).

El Control principal se encarga de decodificar cada tecla, modificar los registros correctos y llevar en cuenta las direcciones que se van modificando bajo cada combinación en la secuencia de las teclas. Asimismo, vigila la lectura del Picoblaze para determinar cuando el dato ha salido leído y repetir una nueva rutina de lectura. Este control cuenta con 3 registros que contienen la dirección del dato que se va a alterar (si son segundos, minutos, horas...), el dato que se escribirá, gobernado principalmente por las teclas numéricas y el registro de commit, que permite guardar el dato en el Picoblaze.

El decodificador de dirección recibe el PortID del Picoblaze y permite direccionar al multiplexor (Multiplexor de salida) en la salida de los datos hacia el mismo microcontrolador (Seleccionar entre Dirección, Dato o Commit). Asimismo, lleva en cuenta cuando el Picoblaze hace lectura del registro commit, que alerta que el Control Principal debe hacer Reset.

Finalmente, el Decodificador de 7 segmentos es combinacional y permite decodificar los datos en Hexadecimal para proyectarlos en los displays de la placa Nexys 4.

2.2.3. Diagrama de tercer nivel

Para la última iteración del diseño modular del Controlador e Teclado, el diseño continúa la línea del segundo nivel, a excepción de que se hacen visibles los nombres de los registros que serán alterados por el ControlKB y serán entregados al Mux para dirigirlos al Picoblaze y el Decodificador de 7 segmentos. Además, se han colocado los nombres de los módulos como fueron implementados en HDL. Para observar el diseño, consulte la figura 9.

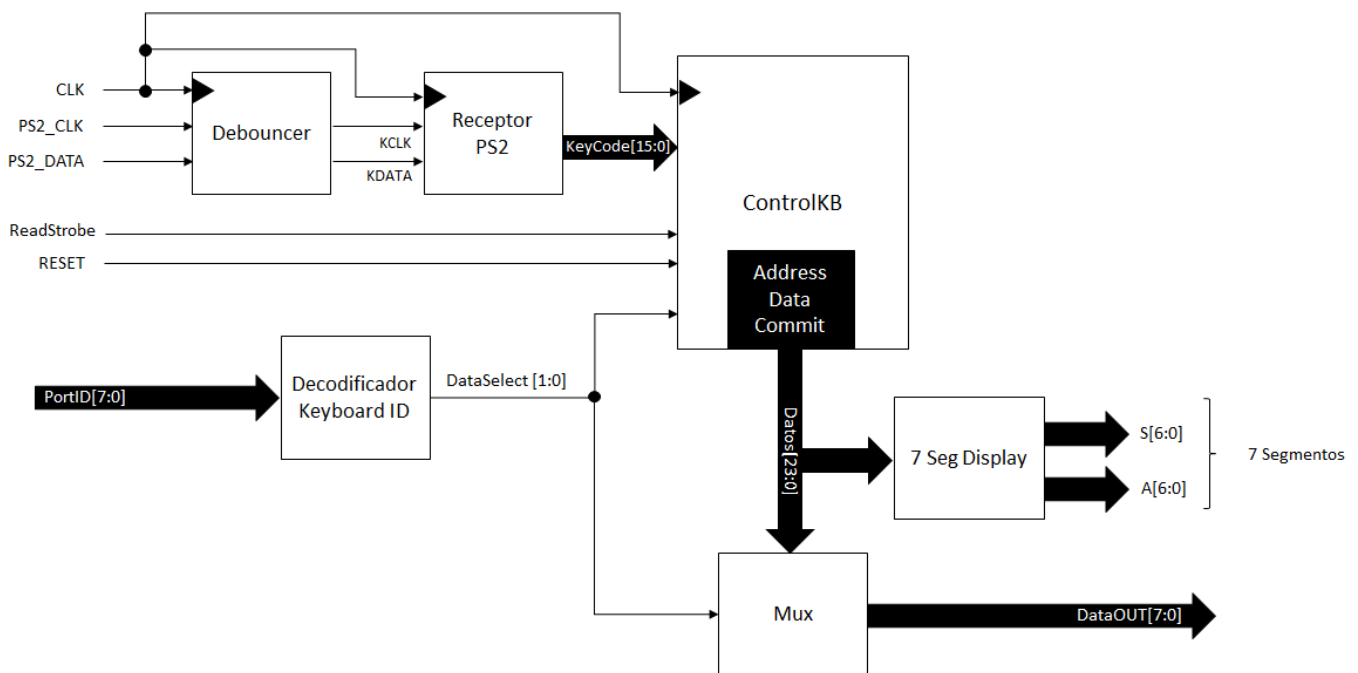


Figura 9: Diagrama de tercer nivel KB

2.3. Control de usuario

Para poder controlar el acceso del usuario, que se comunica por medio de 7 botones, 3 interruptores que indican que se quiere cambiar, el reloj, el timer o la alarma, y para moverse entre los registros de datos y aumentar o disminuir sus valores.

2.3.1. Diagrama de primer nivel

el control de usuario posee 3 interruptores y 4 botones para que el usuario elija los datos y que desea cambiar. Además posee entradas y salidas de memoria para poder alterar los registros y escribirlos en la rtc. Se puede notar esto en la figura 10.

2.3.2. Diagrama de segundo nivel

En este diagrama mostrado en la Fig. 11 se muestra como se pretende realizar el control de usuario, el cual consiste en tan solo dos bloques.



Figura 10: Diagrama de primer nivel del control de usuario.

Consiste un un control de acceso que controla el cambio de los valores de los registros y el control de sus direcciones en el registro. El registro estará en la memoria de coneccion esta memoria controla la actualización de los registros por medio de señales de control, esta es la señal de control, con esta se controla la salida memoria in y memoria out.

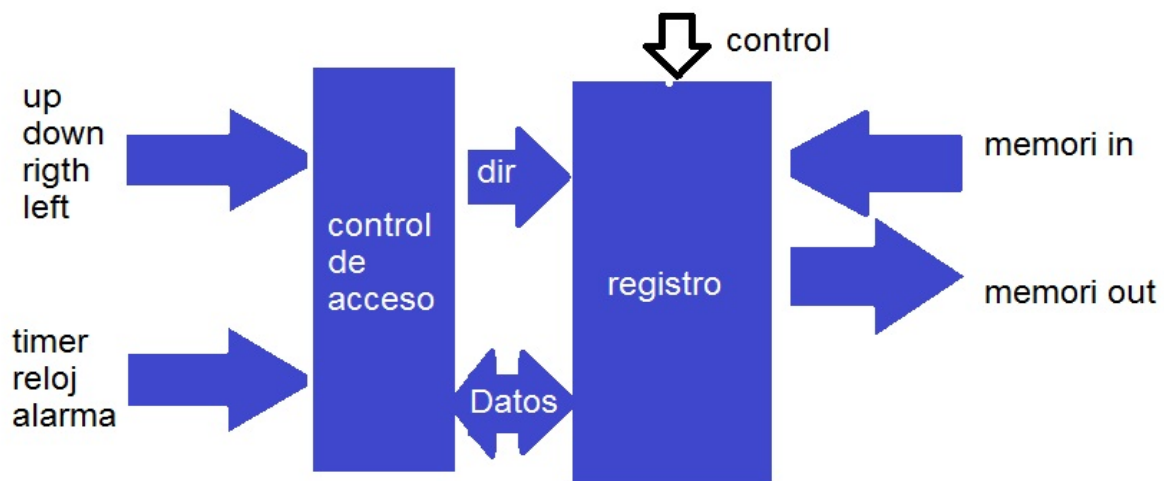


Figura 11: Diagrama de segundo nivel del Teclado.

2.4. Controlador del RTC

Para la implementación de esta interfaz que va a permitir la comunicación entre la FPGA y el RTC, se desarrolló 7 bloques principales divididos en una jerarquía de 3 niveles, se puede ver en la figura 12. Existen 3 bloques principales uno de inicializacion, un while true, que permite la lectura continua de los datos de la rtc, y una de programación que permite actualizar los cambios del control de usuario.

Ademas existen 2 bloques que permiten un bloque que permite leer y escribir datos, esta activa un control que esta basado en los tiempos de la figura 13 y la figura 14, como se puede notar existen muchas similitudes entre ambos ciclos, para esto llamaremos a esta diferencia ciclo” de esta manera podemos armar el cuadro de figura 15

Este diagrama muestra los cambios que deben ocurrir según los tiempos del timer dentro del modulo, al llegar el tiempo final saca una bandera indicando el final.

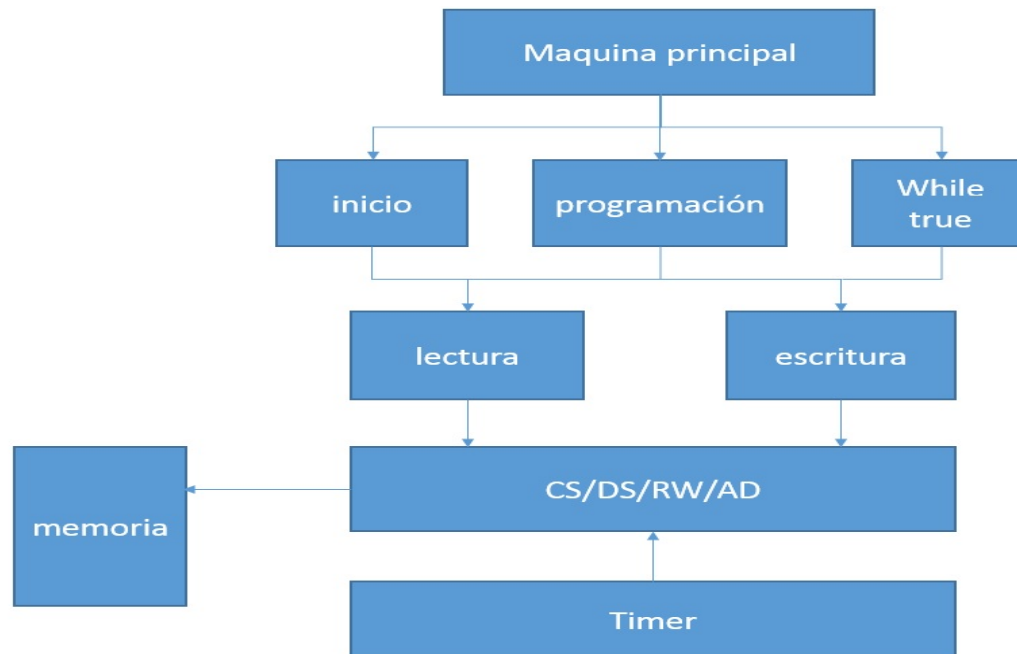


Figura 12: jerarquía de la RTC.

2.4.1. nivel 1 control RTC

Para este nivel se requiere la entrada y salida de datos al registro de memoria de conexión y tiene las salidas necesarias para controlar la RTC, esto se nota en la figura 16.

2.4.2. nivel 2 RTC

Esta etapa posee 4 bloques principales uno que controla la comunicación de la máquina principal de control y el bloque de la RTC. Esto se muestra en la figura 17 en este se ven los caminos posibles a tomar. Ya sea escritura y lectura y sus respectivos tiempos de espera en la escritura de la RTC.

Para esta máquina existen 2 caminos que se pueden tomar, dependiendo de si se va a escribir o leer en la RTC, además de tener un ciclo de espera de la lectura o escritura en la RTC. Y tiene 2 modos de lectura uno por si se va a leer la RTC o la memoria del bloque. Siguiendo la jerarquía, existen 2 máquinas, escritura y lectura, estas respetan los flujos de la figura 18 y 19, estas tienen las entradas de datos y dirección y la señal r y w respectivamente y tiene solo una salida de datos y r y w de esta manera la máquina controla que dato sale, si la dirección y o el dato, además tiene el bit de ciclo que determina si está en el ciclo de escritura o lectura, como se nota en los flujos, el bit de ciclo no depende de la máquina, sino de la parte del programa donde esta se encuentre.

Por último el control de salida responde al cuadro de la figura 15 a este le entra, el bit de activación que sale de la máquina de escritura o lectura, y entran los datos del ciclo y dato que salen dependiendo del tiempo; internamente este tiene un timer, con el fin de llevar el tiempo desde la activación, y dependiendo del tiempo que transcurre genera los cambios de la figura 15.

Ciclo de escritura

	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300
CS	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
AD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RW	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
data	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 13: Diagrama de tiempos completo del ciclo de escritura.

Ciclo de lectura

	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300
CS	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
AD	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DS	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RW	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
data	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 14: Diagrama de tiempos completo del ciclo de lectura.

Control de salida (ciclo , dato, r, w)

	CS	AD	DS	WR	Dato	R1	W1
0	1	ciclo	1	1	0	0	0
10	0	1	~ciclo	1	0	0	0
60	0	1	~ciclo	1	Dato	R	W
120	1	1	~ciclo	1	Dato	R	W
140	1	1	1	ciclo	Dato	R	W
150	1	ciclo	1	ciclo	0	0	0

Figura 15: cuadro de tiempos del control de salida.

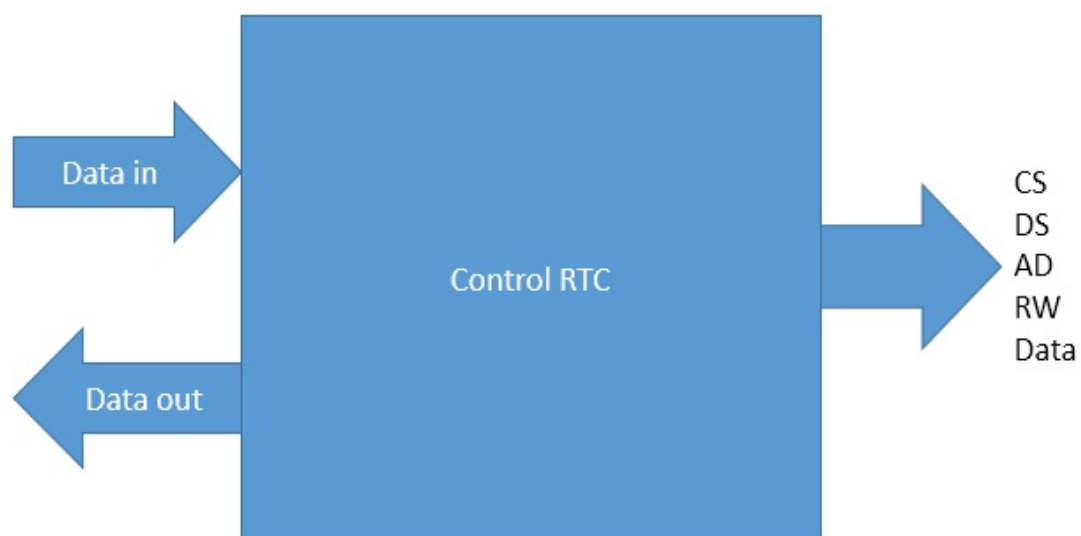


Figura 16: nivel 1 del RTC

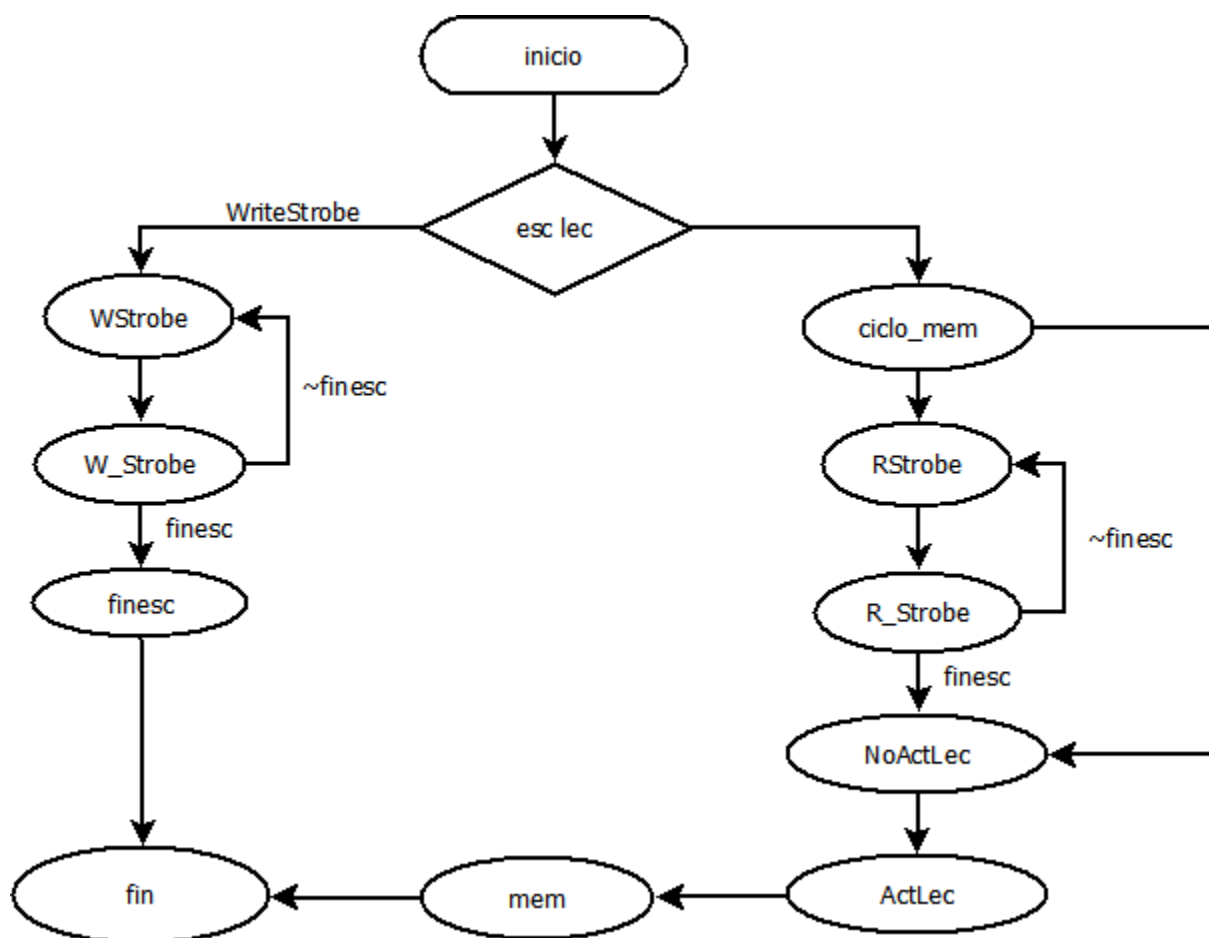


Figura 17: diagrama de la maquina de control de la RTC

Proyecto	1	Página	16/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

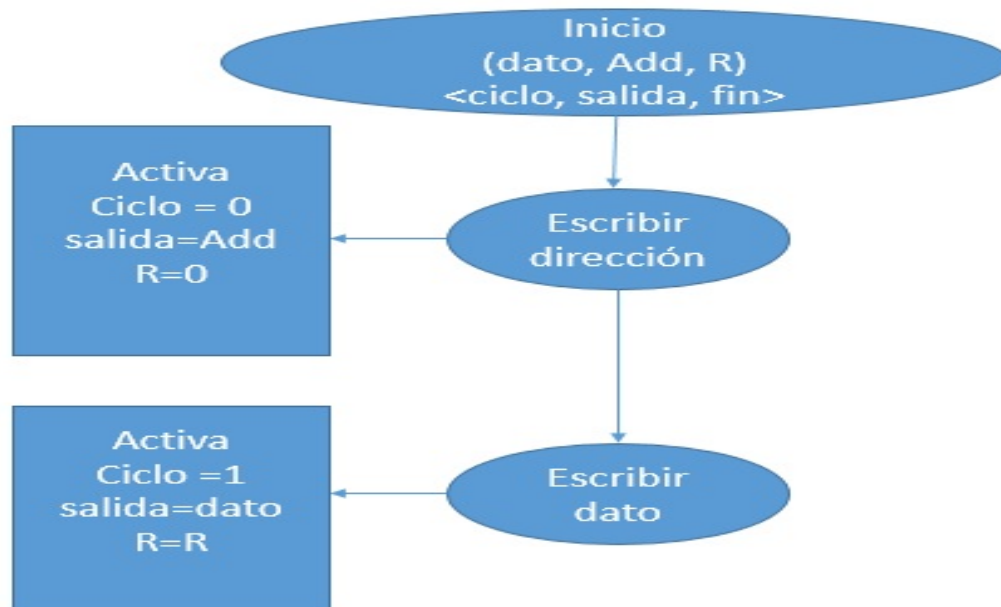


Figura 18: flujo de datos de la maquina de lectura

2.5. control principal

Una de las ideas principales del proyecto es controlar todo el sistema con una sola maquina realizada por un microprocesador. Para esta efecto se buscaron crear 5 ciclos, como se muestra de la figura 20, donde el primer ciclo es el de inicializacion falta diseño.

Después de este un ciclo de teclado, como se muestra en la figura 21, en este se usan los espacios de memoria y Id de la tabla 2, usando los espacios de los registros auxiliares se guardan valores para generar ciclos como se muestra en el diagrama de flujo.

Este flujo tiene 3 etapas principales, una donde guarda el valor del registro que desea cambiar y el valor del teclado. Luego se generan los cambios en la RTC, los procesos de arriba y abajo se hacen automáticamente y una vez que se hace un comit se escribe en la dirección y el dato que ingresaron previamente, el ultimo proceso es activar o desactivar el irq.

Después de este proceso sigue el ciclo de la RTC como se muestra en la figura 22, este genera ciclos donde activa la lectura en la RTC, y espera a que el dato este estable para poder leerlo.

Después de este proceso el sistema espera a que la VGA le solicite los datos luego este pasa al proceso de VGA como se muestra en la figura 23, este proceso se parece mucho al de escritura, donde se crea un ciclo donde se recorren todas las direcciones de escritura con la ayuda de un auxiliar y con ayuda del deco se determina el valor de dirección donde se guarda en la VGA.

Por facilidad se uso la misma dirección en la memoria de regitros que en el id.

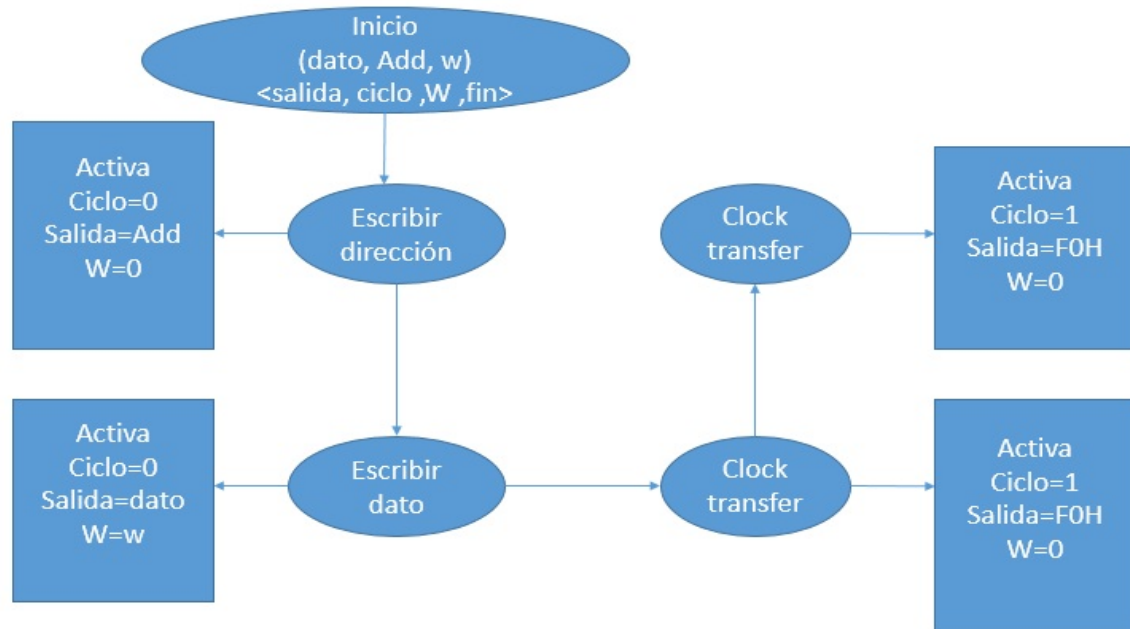
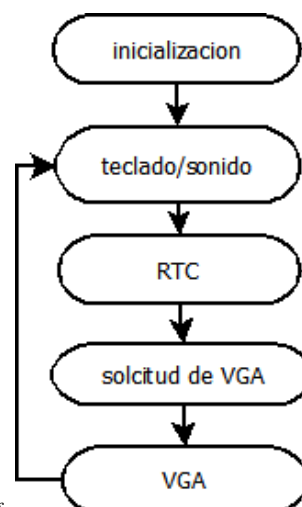


Figura 19: flujo de datos de la maquina de Escritura



De Flujo Principial.png

Figura 20: Diagrama de flujo principal

de teclado irq.png

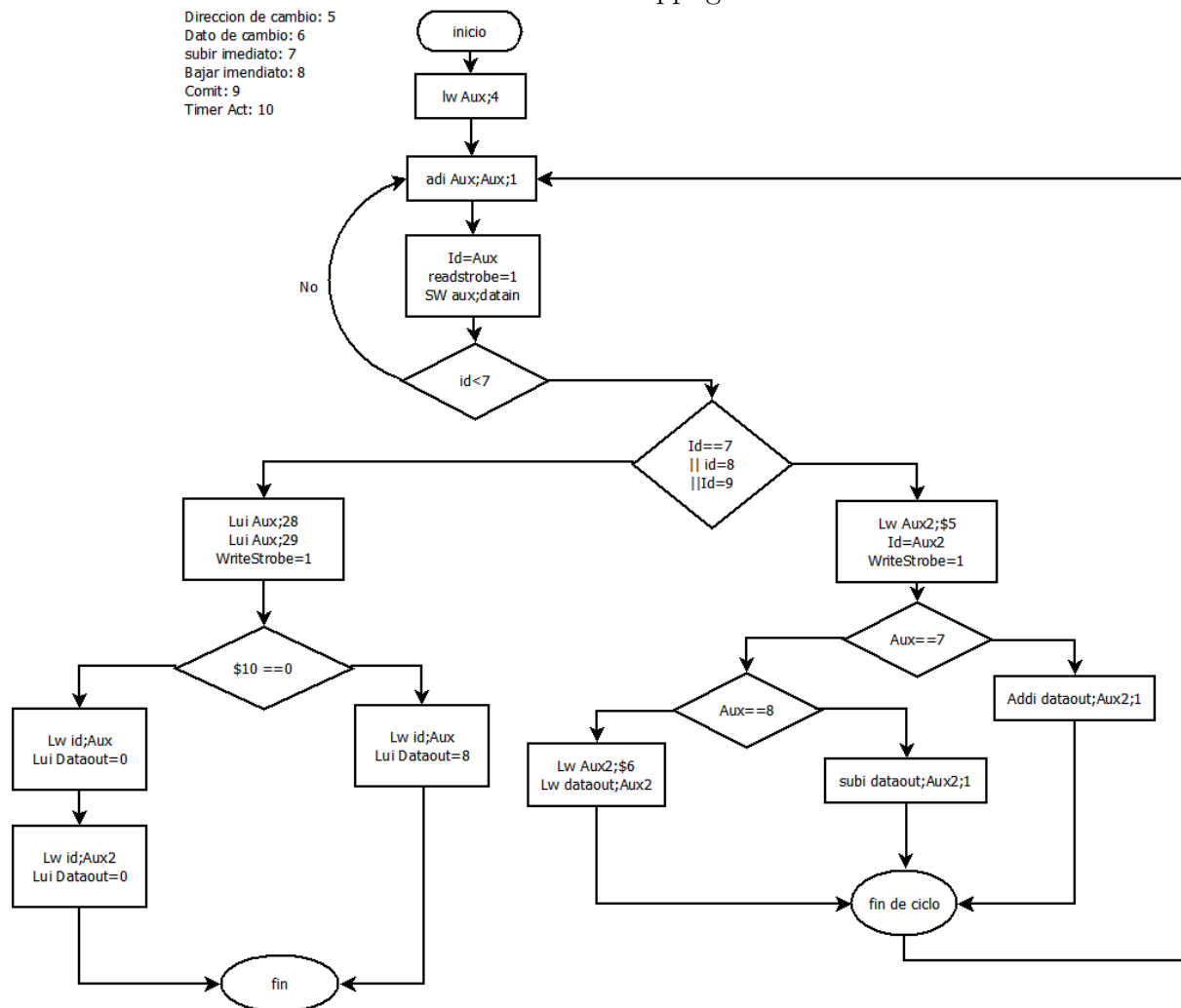
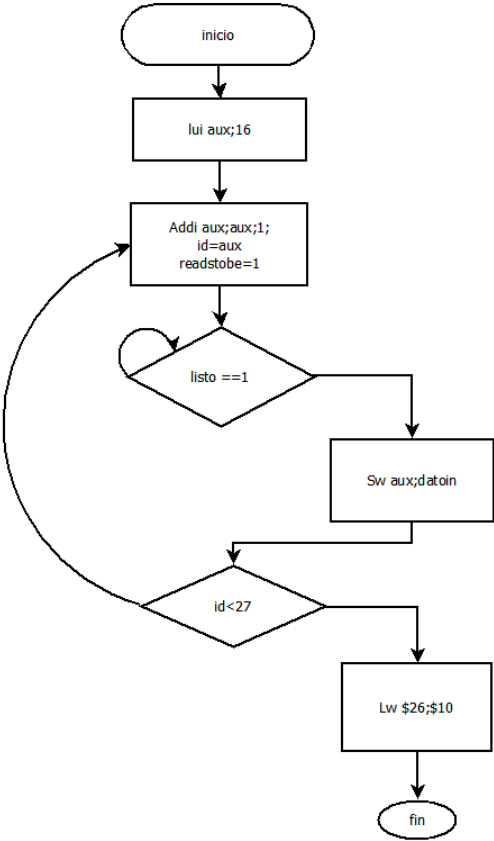


Figura 21: Diagrama de flujo principal

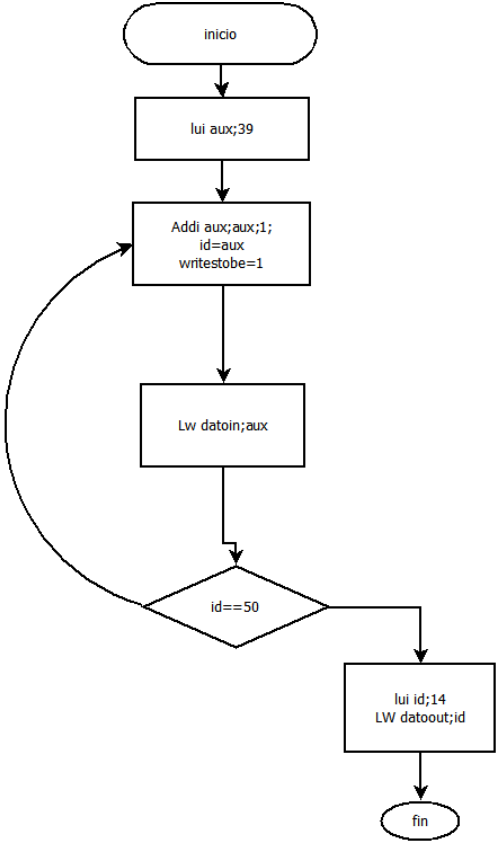
Variable	Deco	memoria
Segundos=	17	33
minutos=	18	34
horas=	19	35
Dias=	20	36
Meses=	21	37
Años=	22	38
SegundosT=	23	h41
minutosT=	24	h42
horasT=	25	h43
timerdisp	26	
timeract	27	
Staus1	28	0
Staus2	29	1



de flujo RTC.png

Figura 22: Diagrama de flujo principal

Variable	memoria
Segundos=	40
minutos=	41
horas=	42
Dias=	43
Meses=	44
Años=	45
SegundosT=	46
minutosT=	47
horasT=	48
timerdisp	49
timeract	50



de flujo VGA.png

Figura 23: Diagrama de flujo principal

Proyecto	1	Página	21/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

Cuadro 2: resumen del Deco

Id	CS	ADD
1	status 0	00
2	status 1	01
3	status 2	02
4	comit	F0
5	teclado	
6	teclado	
7	teclado	
14	sonido	
17	RTC	d33
18	RTC	d34
19	RTC	d35
20	RTC	d36
21	RTC	d37
22	RTC	d38
23	RTC	h41
24	RTC	h42
25	RTC	h43
26	RTC	
27	RTC	
28	RTC	
29	RTC	
33	AUX	
34	AUX	
35	AUX	
36	AUX	
40	VGA	Direccion
41	VGA	Dato
42	VGA	Cursor3
43	VGA	4
44	VGA	5
45	VGA	6
46	VGA	7
47	VGA	8
48	VGA	9
49	VGA	10
50	VGA	11

Proyecto	1	Página	22/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

3. Datos y resultados

3.1. Simulaciones

Para el diseño del controlador VGA, se tenía que verificar el funcionamiento de la misma con respecto a los datos consultados a la memoria de registros de la RTC. Para ello, se ha realizado una simulación para verificar la funcionalidad del controlador VGA. Para este caso, la simulación de comprobación consta de dos partes importantes: la simulación en ISim y la traducción de los datos brindados para graficarlos en un script de Matlab.

Previo a simular, se tomará el punto de decenas en el día, que se encuentra en el punto en Y 130, de acuerdo con la imagen ???. Seguidamente, se verificará el cambio en el script de Matlab en comparativa con los datos que ingresaron e la figura 30.

```

/*
  Parametros de punteros
*/

// Ubicación del punto (1,1)
parameter FechaY1 = 130;
parameter HoraY1 = 288;
parameter IndicadoresY1 = 370;
// Parametros en X
parameter DDayX = 132;
parameter UDayX = 174;
parameter DMonthX = 232;
parameter UMonthX = 274;
parameter DYearX = 416;
parameter UYearX = 458;
parameter DRHourX = 38;
parameter URHourX = 80;
parameter DRMinuteX = 128;
parameter URMinuteX = 170;
parameter DRSecondX = 218;
parameter URSecondX = 260;
parameter DCHourX = 339;
parameter UCHourX = 381;
parameter DCMinuteX = 429;
parameter UCMMinuteX = 471;
parameter DCSecondX = 519;
parameter UCSecondX = 561;
parameter ActivadoX = 338;
parameter FinalizadoX = 500;
// Ubicación del punto (1,H)
parameter FechaY2 = 190;
parameter HoraY2 = 348;
parameter IndicadoresY2 = 388;

```

Figura 24: Punteros de memoria

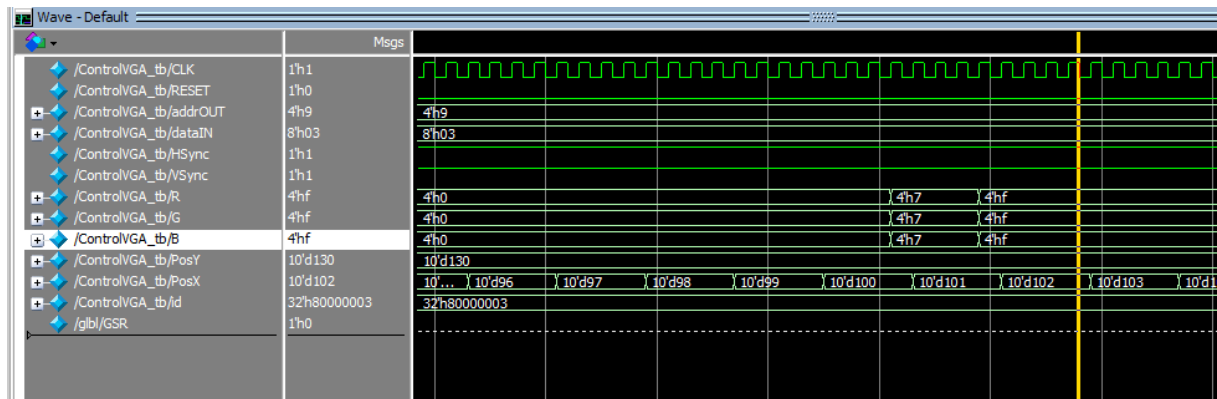


Figura 25: Transiciones de datos.

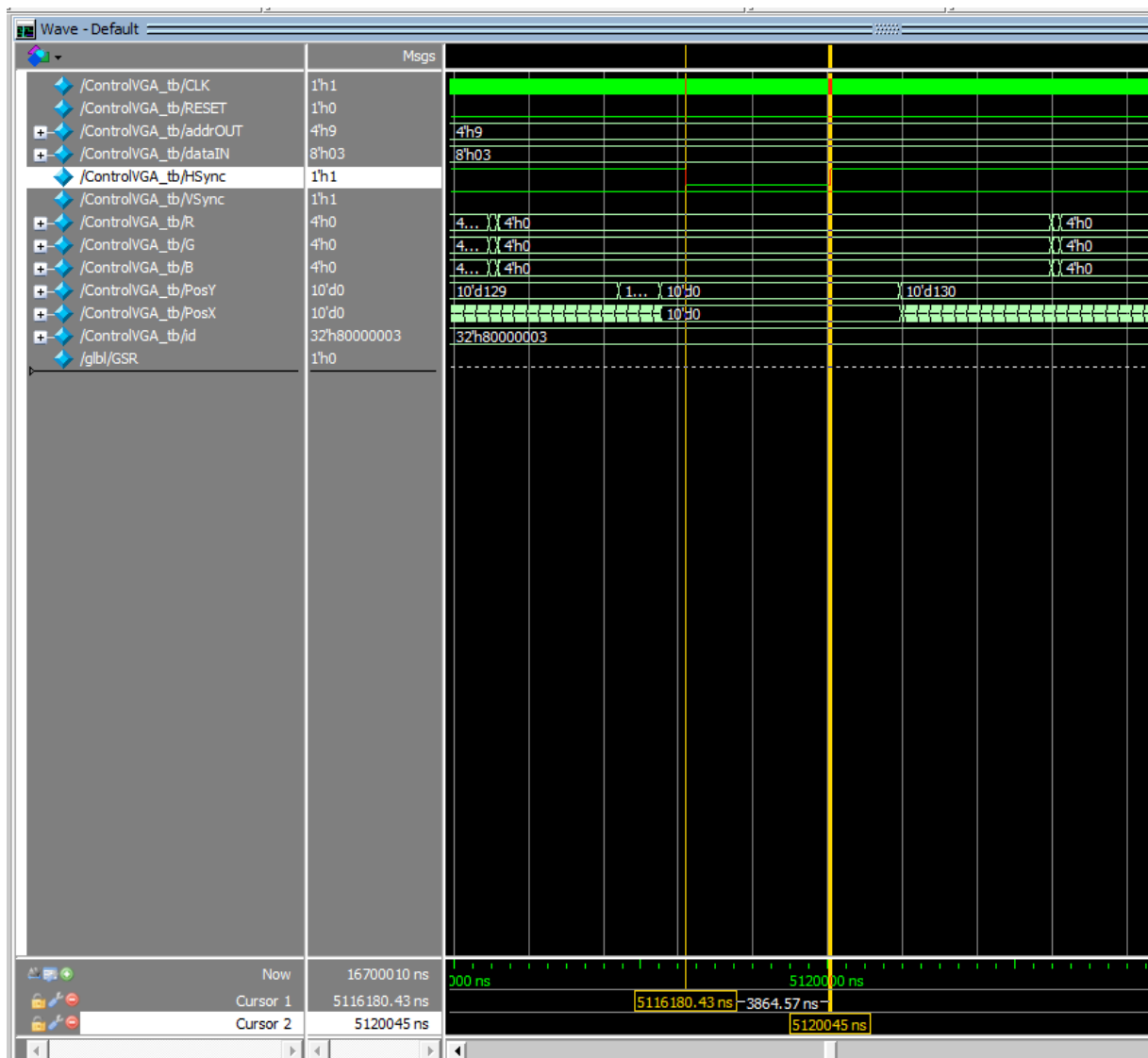


Figura 26: Tiempo de pulso HSync - Se cumple

Proyecto	1	Página	24/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

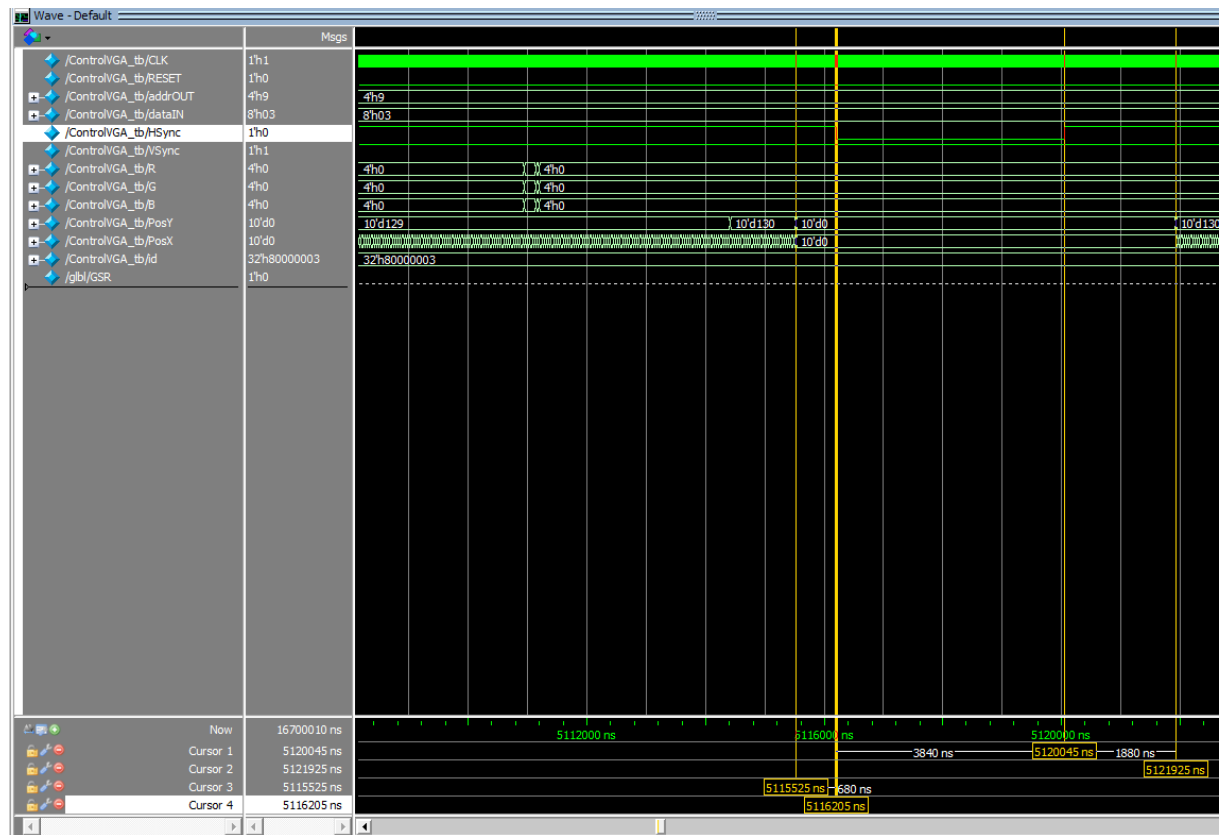


Figura 27: Tiempos en HSync - Porches

Para el teclado, se ha desarrollado un testbench para poner a prueba la funcionalidad del Controlador de Teclado de forma completa, con emulación de presión de tecla. Refiérase a la figura 31

En esta simulación, es posible apreciar los cambios en los registros Address, Data y Commit, que corresponden a los tres posibles registros que puede leer el picoblaze y seleccionarlos mediante el PortID. Además, el Keyboard_Output puede desplegar un reinicio a cero después de la activación del ReadStrobe.

3.2. Mediciones

Es importante considerar el consumo de potencia de la implementación del diseño en la FPGA y el área ocupada dentro del Artix7, chip de FPGA que trae integrada la FPGA Nexys 4. En la Tabla 3 es posible observar algunos parámetros de consumo de potencia (obtenidos por medio del XPower Analyzer) y los recursos brindados por la conclusión de la síntesis:

Se realizó la medicion de algunos parámetros del circuito, como lo son AD, CS, WR, RD. Además la señal de IRQ, la salida de audio, yel bus de datos bidireccional.

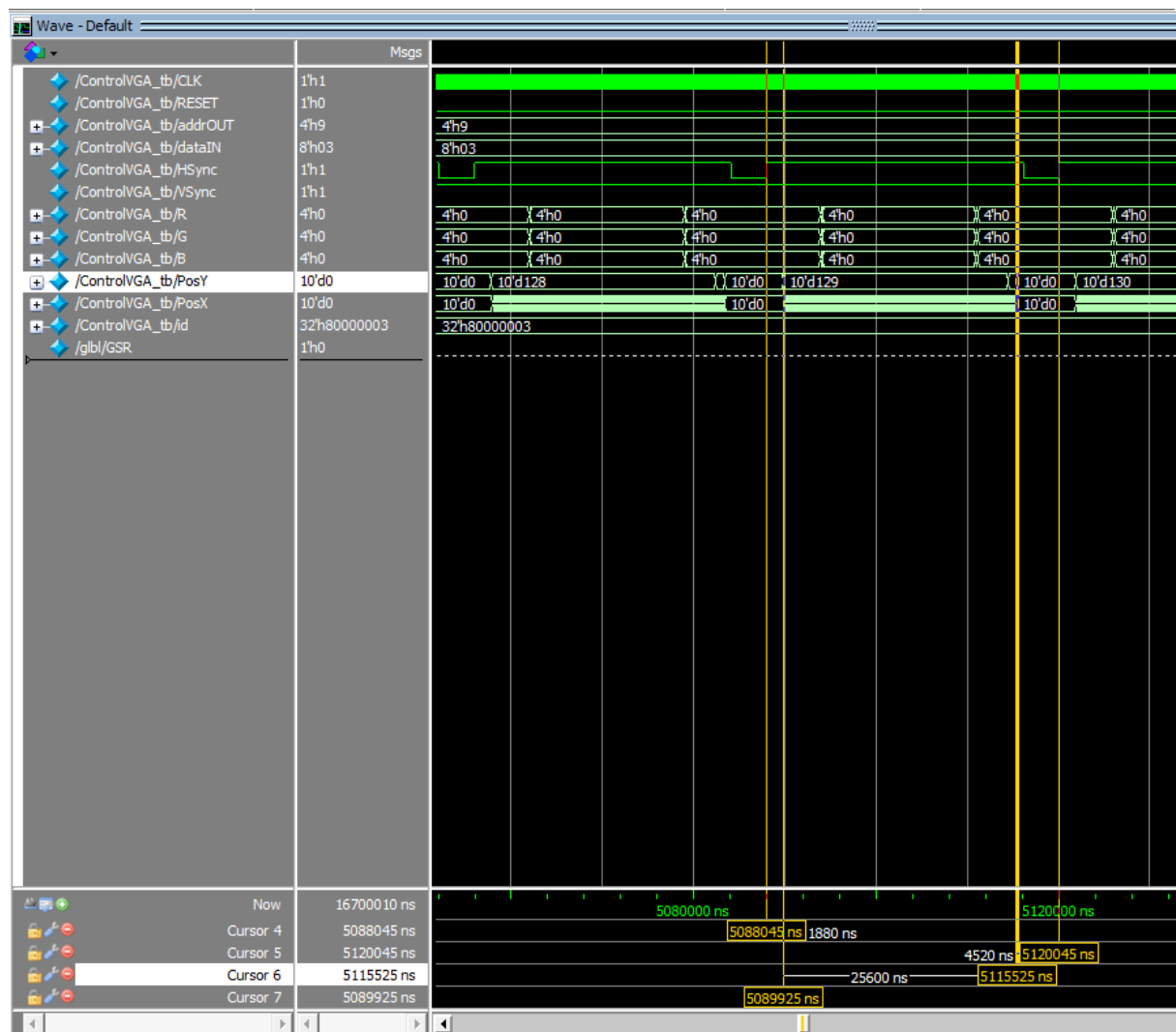


Figura 28: Tiempos en HSync en general

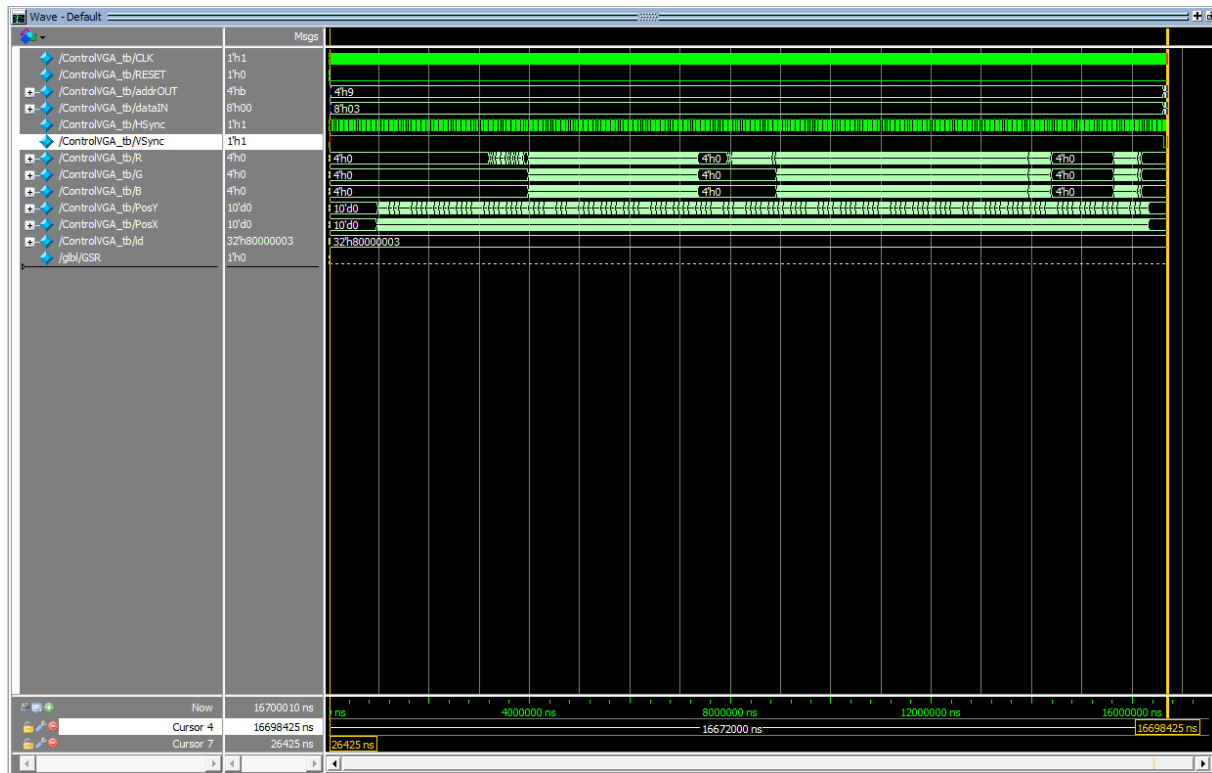


Figura 29: Tiempos en VSync

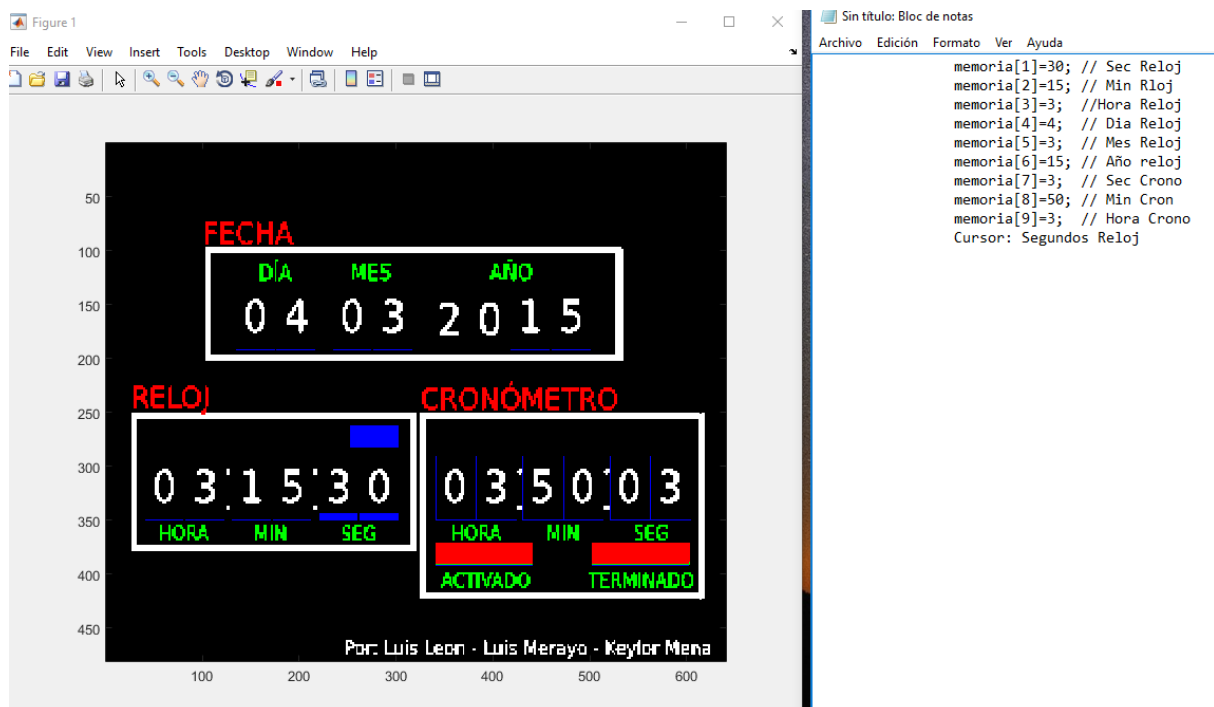


Figura 30: Tiempos en HSync en general

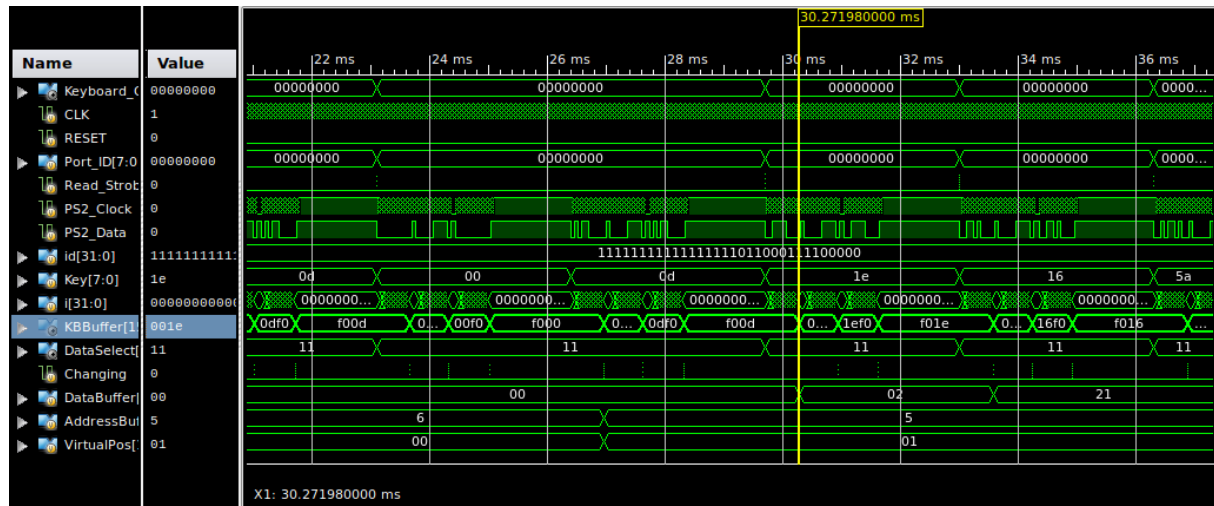


Figura 31: Simulación del teclado

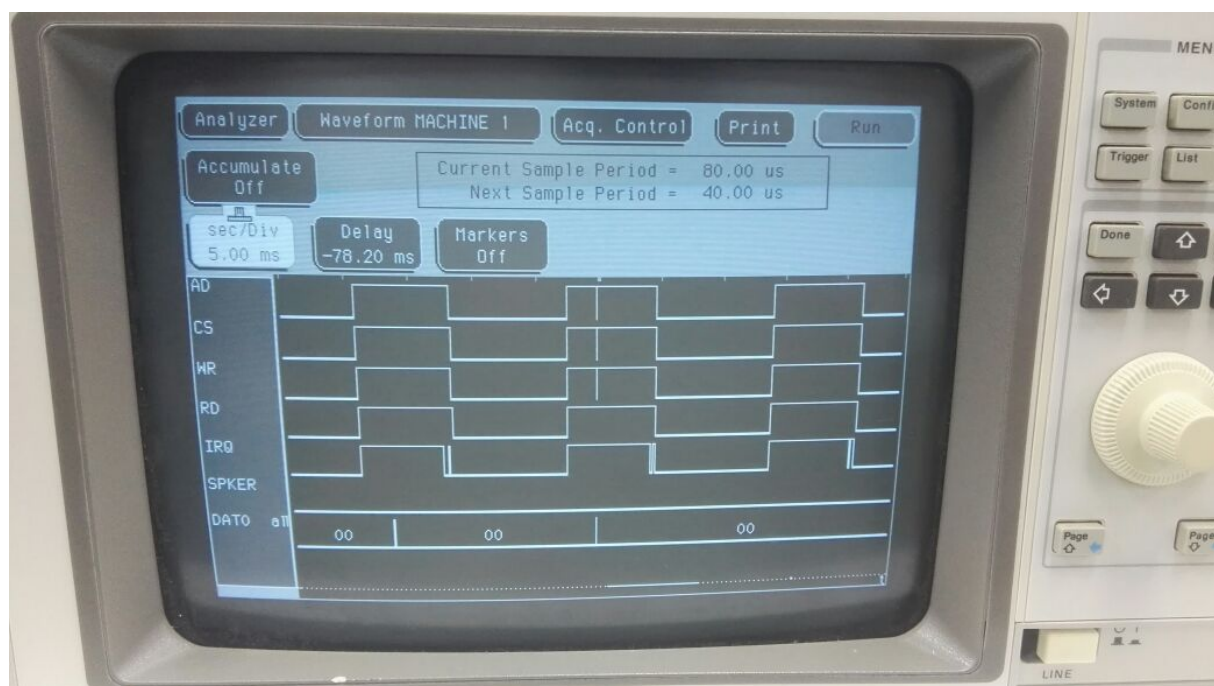


Figura 32: Proceso general de operación del circuito

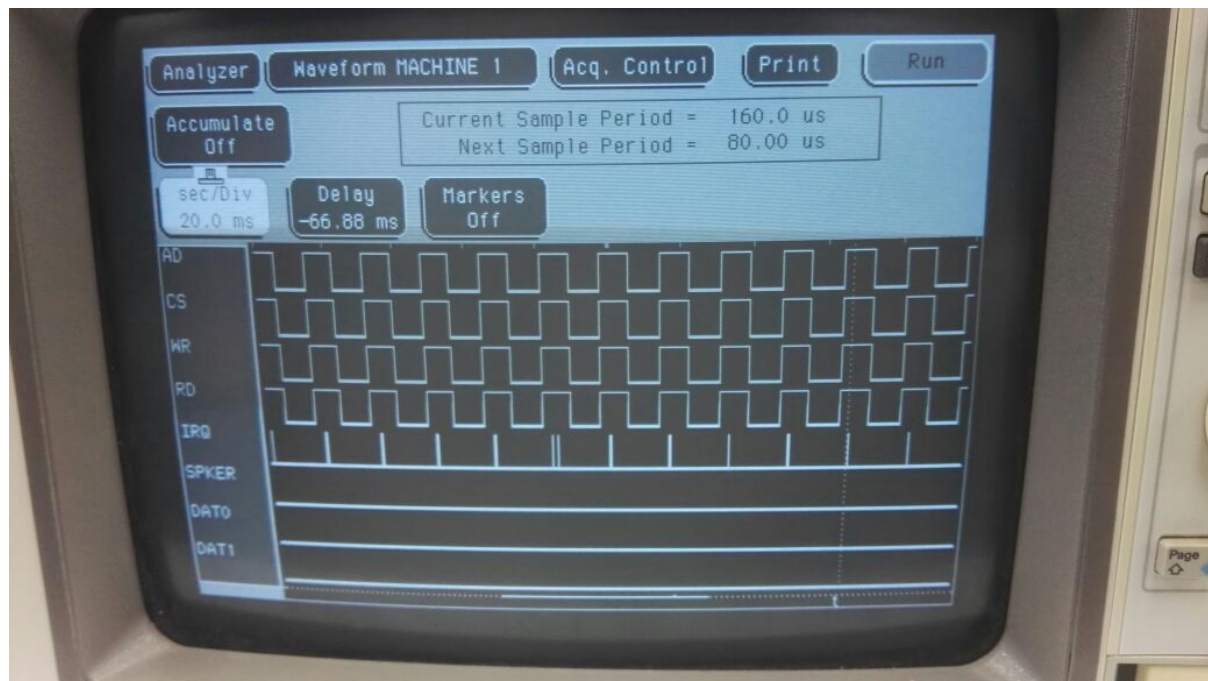


Figura 33: Señal IRQ activada

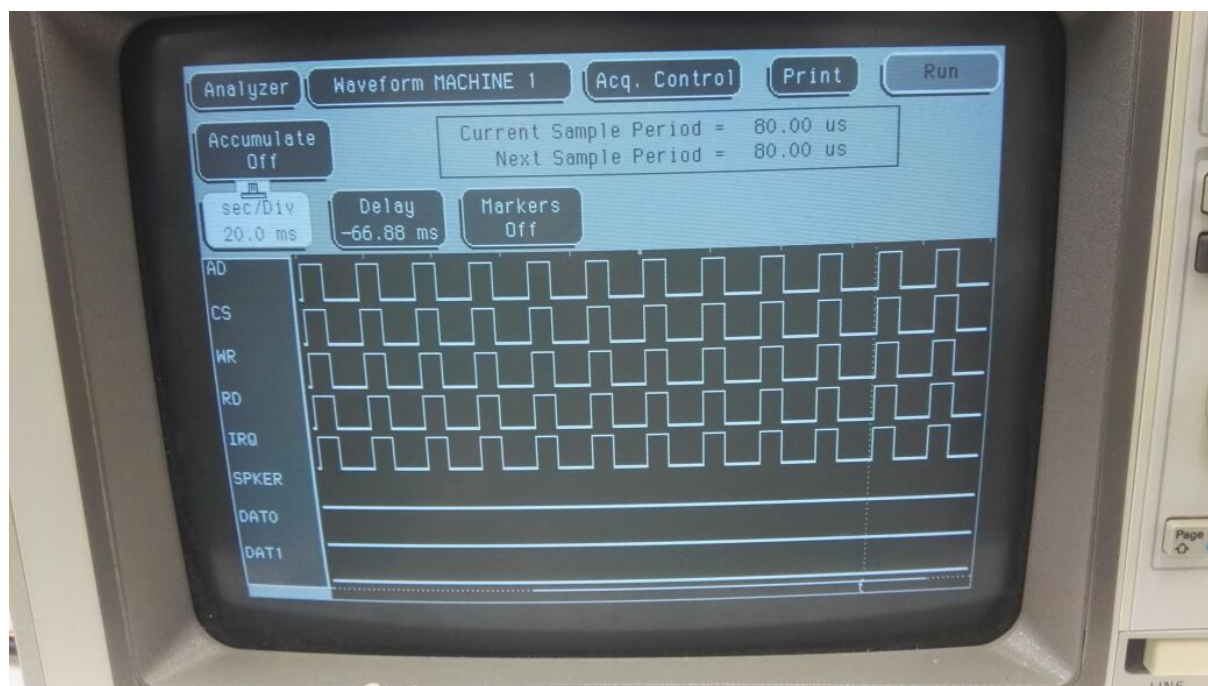


Figura 34: Señal IRQ desactivada y alarma desactivada

Cuadro 3: Consumo de recursos de la FPGA

Recurso	Consumo
Potencia	194mW
Slices ocupadas	1 %
Registros	¡1 %
IO	22 %
LUTs lógicos	1 %
LUT FF	1306
DSP48E1	1 %
Reloj Máximo	98.739MHz

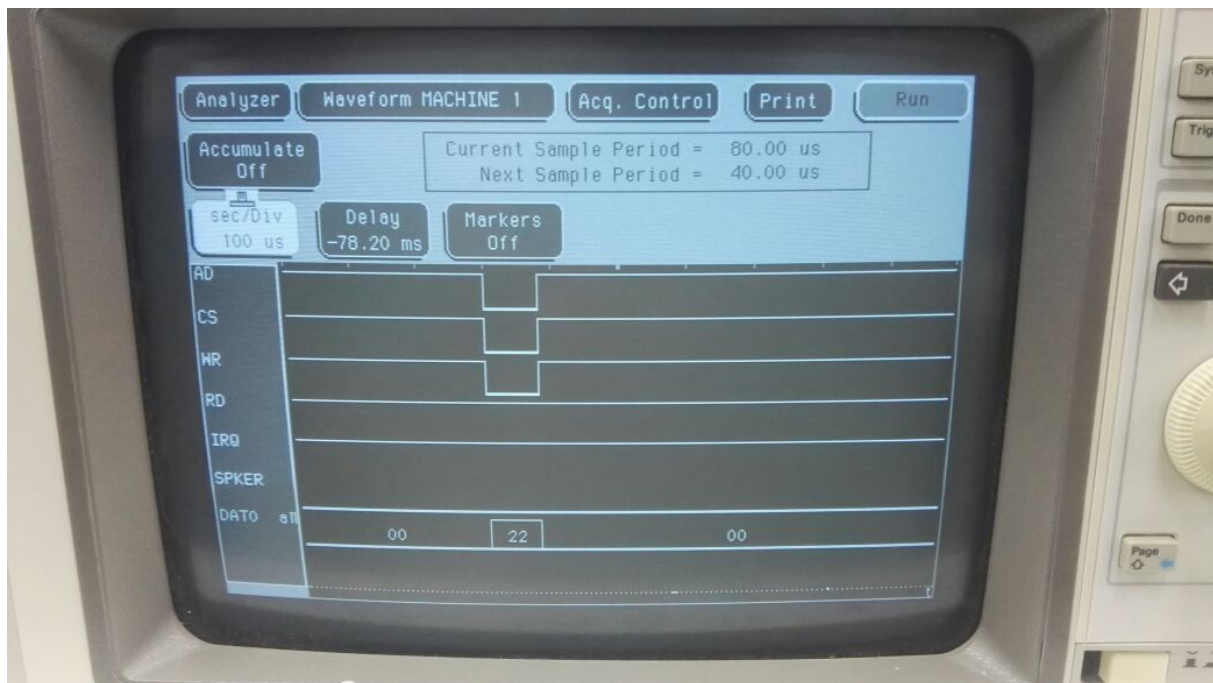


Figura 35: Escritura de dirección en la RTC

4. Análisis de datos y resultados

El funcionamiento del sistema esta en un optimo funcionamiento en la practica, aunque existen algunos defectos, por ejemplo:

- El tiempo no esta en una optima condición, en la tabla 3 se puede notar que el tiempo de ejecución esta en 98.739 MHz.
- En las mediciones realizadas en el analizador lógico podemos notar que las señales RD,CS y WR no tienen mucha separacion en sus comienzos y finales.

Pero gracias a la construcción del micro de una manera muy sistemática n la activación de sus señales este problema se minimizo.

Ademas, el segundo problema se redujeron sus efectos, debido que el sistema de salida de la RTC no trabaja tan cercano en sus tiempos críticos, sino se que se agrego un tiempo extra por cualquier error.

Proyecto	1	Página	30/32
Trabajo	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	Actualizado en:	6/11/2016
Grupo	8	Revisado en:	8/11/2016
Revisado por:	Alfonso Chacón Rodríguez	Diseñadores	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

5. Hoja de datos de unidades desarrolladas

Para las conexiones del diseño implementado en Verilog del circuito, se debe seguir con la siguiente guía de conexión, que en este caso, se muestran en las tablas 4 y 5. Es importante seguirla para evitar conflictos y cabe aclarar que se recomienda implementarla en Nexys 4, ya que esta tabla tiene el pinout establecida para dicha placa de FPGA.

6. Conclusiones y recomendaciones

6.1. Conclusiones

- Utiliza un microcontrolador como máquina de control facilitó el proceso de desarrollo de dicha etapa, debido a que se abstrae a nivel de software.
- Colocar un módulo externo de RTC aliviana la carga del microcontrolador en algunos sistemas, como por ejemplo, en dispositivos móviles, ya que se puede aprovechar el potencial del microprocesador para ejecutar tareas de mayor importancia.
- Desarrollar una unidad de control principal ayuda en la comunicación entre los diferentes periféricos, esto comunicando un identificador (PORT_ID) y un bus de datos centralizado.
- Colocar un teclado para configurar la información del sistema ayuda a tener una mejor experiencia de usuario, ya que se pueden escribir los números en lugar de estar seleccionándolos mediante dos botones.

6.2. Recomendaciones

- Se debe acondicionar mejor la comunicación entre la RTC y la Nexys 4, debido a fluctuaciones en la alimentación de este en el IRQ que ocasionaron problemas. Esto se solucionó de forma temporal mediante un circuito RC.
- La interfaz de la VGA debe realizarse de forma externa para mejorar la experiencia, ya que, si se importan imágenes mediante ROM en HDL, puede consumir gran cantidad de recursos que pueden afectar la velocidad máxima del sistema.
- Controlar los periféricos mediante circuitos modulares separados.

7. Bibliografía

[3] "Nexys4-DDR Keyboard Demo [Reference.Digilentinc]", *Reference.digilentinc.com*, 2016. [Online]. Available: <https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-keyboard-demo/start>. [Accessed: 12-Oct-2016].

Cuadro 4: Asignación de pines de las entradas y salidas

Señal	Designación	Pin	Función
clk	input	E3	Señal de reloj de la Nexys 4, es de 100MHz.
PS2_CLOCK	input	F4	Señal de reloj del teclado
PS2_DATA	input	B2	Datos en serie del teclado
datRTC[0]	inout	B13	Bit de datos y direcciones de entrada.
datRTC[1]	inout	F14	Bit de datos y direcciones de entrada.
datRTC[2]	inout	D17	Bit de datos y direcciones de entrada.
datRTC[3]	inout	E17	Bit de datos y direcciones de entrada.
datRTC[4]	inout	G13	Bit de datos y direcciones de entrada.
datRTC[5]	inout	C17	Bit de datos y direcciones de entrada.
datRTC[6]	inout	D18	Bit de datos y direcciones de entrada.
datRTC[7]	inout	E18	Bit de datos y direcciones de entrada.
AN[0]	output	N6	Ánodos de Display.
AN[1]	output	M6	Ánodos de Display.
AN[2]	output	M3	Ánodos de Display.
AN[3]	output	N5	Ánodos de Display.
AN[4]	output	N2	Ánodos de Display.
AN[5]	output	N4	Ánodos de Display.
AN[6]	output	L1	Ánodos de Display.
AN[7]	output	M1	Ánodos de Display.
SEG[0]	output	L3	Segmentos de Display.
SEG[1]	output	N1	Segmentos de Display.
SEG[2]	output	L5	Segmentos de Display.
SEG[3]	output	L4	Segmentos de Display.
SEG[4]	output	K3	Segmentos de Display.
SEG[5]	output	M2	Segmentos de Display.
SEG[6]	output	L6	Segmentos de Display.
R[0]	output	A3	Bit para color rojo.
R[1]	output	B4	Bit para color rojo.
R[2]	output	C5	Bit para color rojo.
R[3]	output	A4	Bit para color rojo.
G[0]	output	C6	Bit para color verde.
G[1]	output	A5	Bit para color verde.
G[2]	output	B6	Bit para color verde.
G[3]	output	A6	Bit para color verde.
B[0]	output	B7	Bit para color azul.
B[1]	output	C7	Bit para color azul.
B[2]	output	D7	Bit para color azul.
B[3]	output	D8	Bit para color azul.
HSync	output	B11	Sincronización horizontal.
VSync	output	B12	Sincronización vertical.

Cuadro 5: Continuación de asignación de pines de las entradas y salidas

Señal	Designación	Pin	Función
RD	output	V11	Señal para leer.
CS	output	P15	Señal de selección.
WR	output	V15	Señal para escribir.
AD	output	G14	Selecciona entre dirección y dato.
inirq	input	U11	Interrupción de RTC.
PWM_OUT	output	A11	Sonido.
inRESET	input	E16	Reset.