

<b>Proyecto</b>	1	<b>Página</b>	1/26
<b>Trabajo</b>	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	<b>Actualizado en:</b>	6/11/2016
<b>Grupo</b>	8	<b>Revisado en:</b>	8/11/2016
<b>Revisado por:</b>	Alfonso Chacón Rodríguez	<b>Diseñadores</b>	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. Ante ello, se debe realizar un controlador para el RTC y para la VGA. Asimismo, se deben poder ajustar la hora, activar la alarma y el cronómetro de forma descendente mediante botones e interruptores dispuestos en la FPGA 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 deberán usar los botones señalados en el instructivo y algunos interruptores.

Finalmente, el conjunto es un circuito que permita controlar el módulo y comunicar al usuario mediante los botones y el monitor LCD, donde él podrá recibir la información relevante y poder modificar dicha información.

## 1. Objetivos

- Diseñar un controlador de RTC que permita leerlo y programarlo mediante una interfaz de usuario consistente en botones incorporados dentro de la FPGA (Nexys 4) y un monitor comunicado a través del protocolo VGA.
- Investigar el funcionamiento del módulo RTC y el protocolo de comunicación del mismo.
- Diseñar un controlador para el módulo RTC, cuyo bus de datos y direcciones estén multiplexados.
- Cumplir con las reglas de temporizado del sistema, en especial, con el protocolo de comunicación del módulo RTC.
- Combinar el controlador de RTC con un controlador VGA 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.

## 2. Descripción del sistema

El sistema se puede dividir en cuatro subsistemas, para facilitar el diseño dividimos el sistema en 4 grandes partes, el controlador de la pantalla, el controlador para el RTC, el control de usuario y una memoria principal. Éstos subsistemas, pueden ser desarrollados de manera separada siempre que se tenga el cuidado necesario con los datos que comparten entre los bloques, para este efecto se desarrollo

<b>Proyecto</b>	1	<b>Página</b>	2/26
<b>Trabajo</b>	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	<b>Actualizado en:</b>	6/11/2016
<b>Grupo</b>	8	<b>Revisado en:</b>	8/11/2016
<b>Revisado por:</b>	Alfonso Chacón Rodríguez	<b>Diseñadores</b>	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

una memoria con 2 registros que se actualizan entre ellos al activar banderas. En la Fig. 1 se puede observar la composición general del sistema.

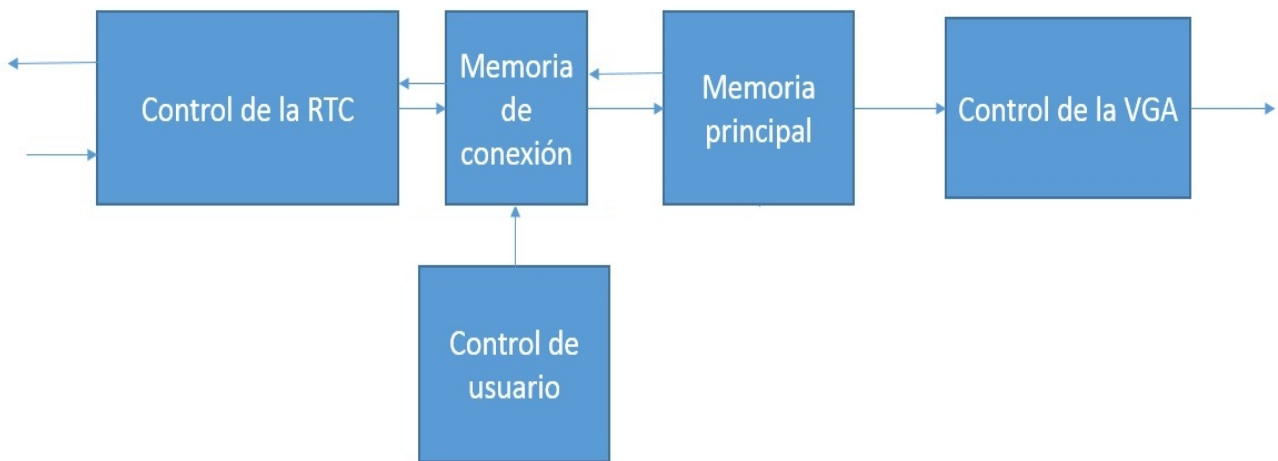


Figura 1: Diagrama de módulos principales del sistema.

## 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 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.

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

<b>Proyecto</b>	1	<b>Página</b>	3/26
<b>Trabajo</b>	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	<b>Actualizado en:</b>	6/11/2016
<b>Grupo</b>	8	<b>Revisado en:</b>	8/11/2016
<b>Revisado por:</b>	Alfonso Chacón Rodríguez	<b>Diseñadores</b>	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

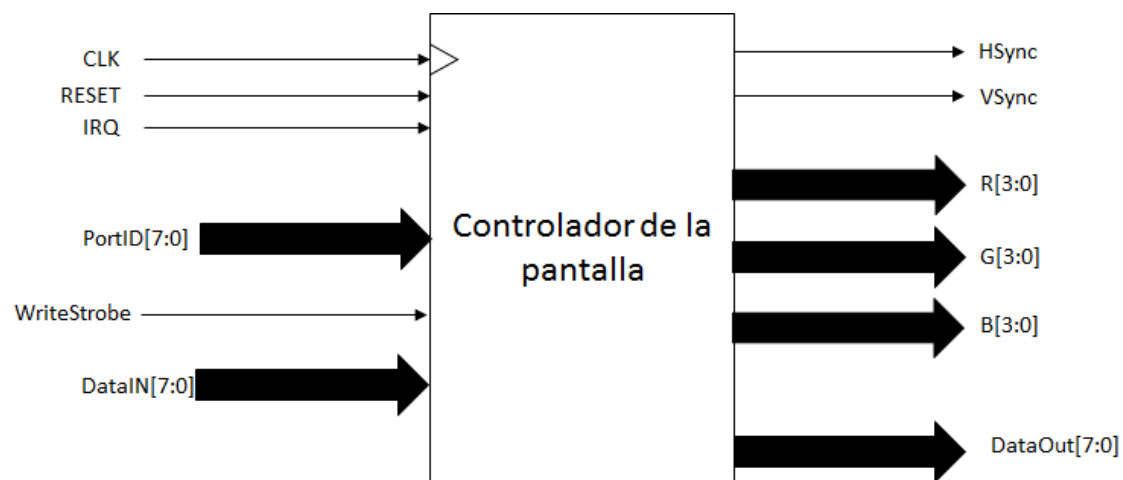


Figura 2: Diagrama de primer nivel VGA

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. 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.

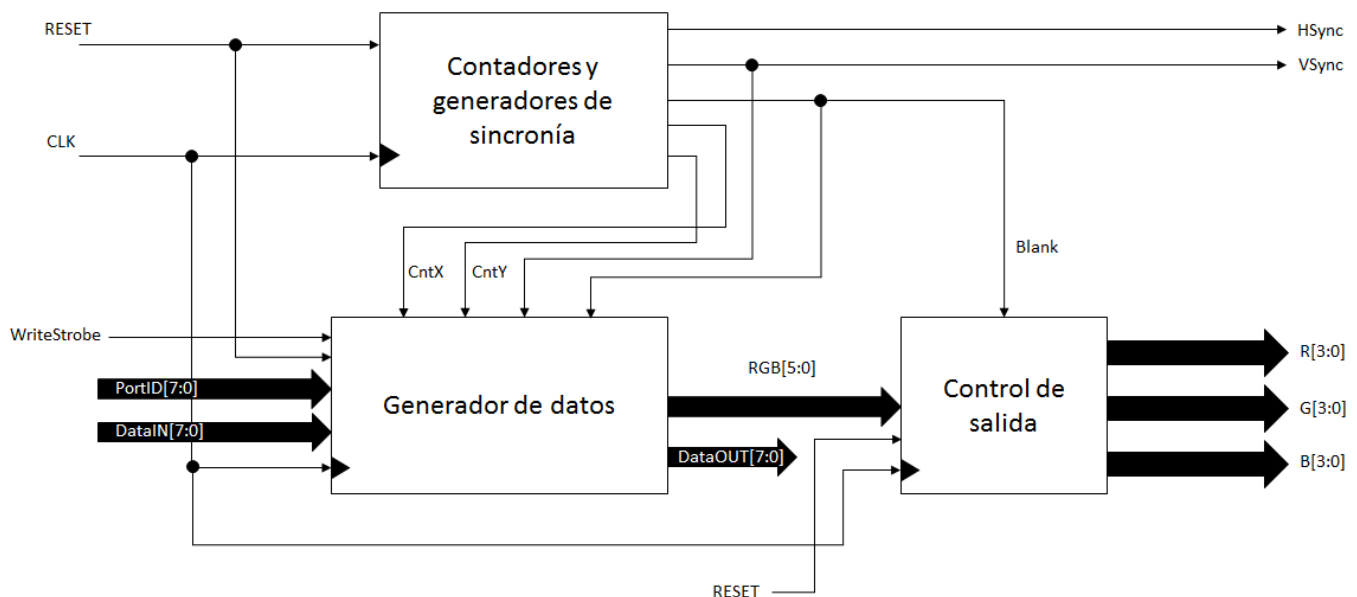


Figura 3: Diagrama de segundo nivel VGA

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.

### 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.

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.

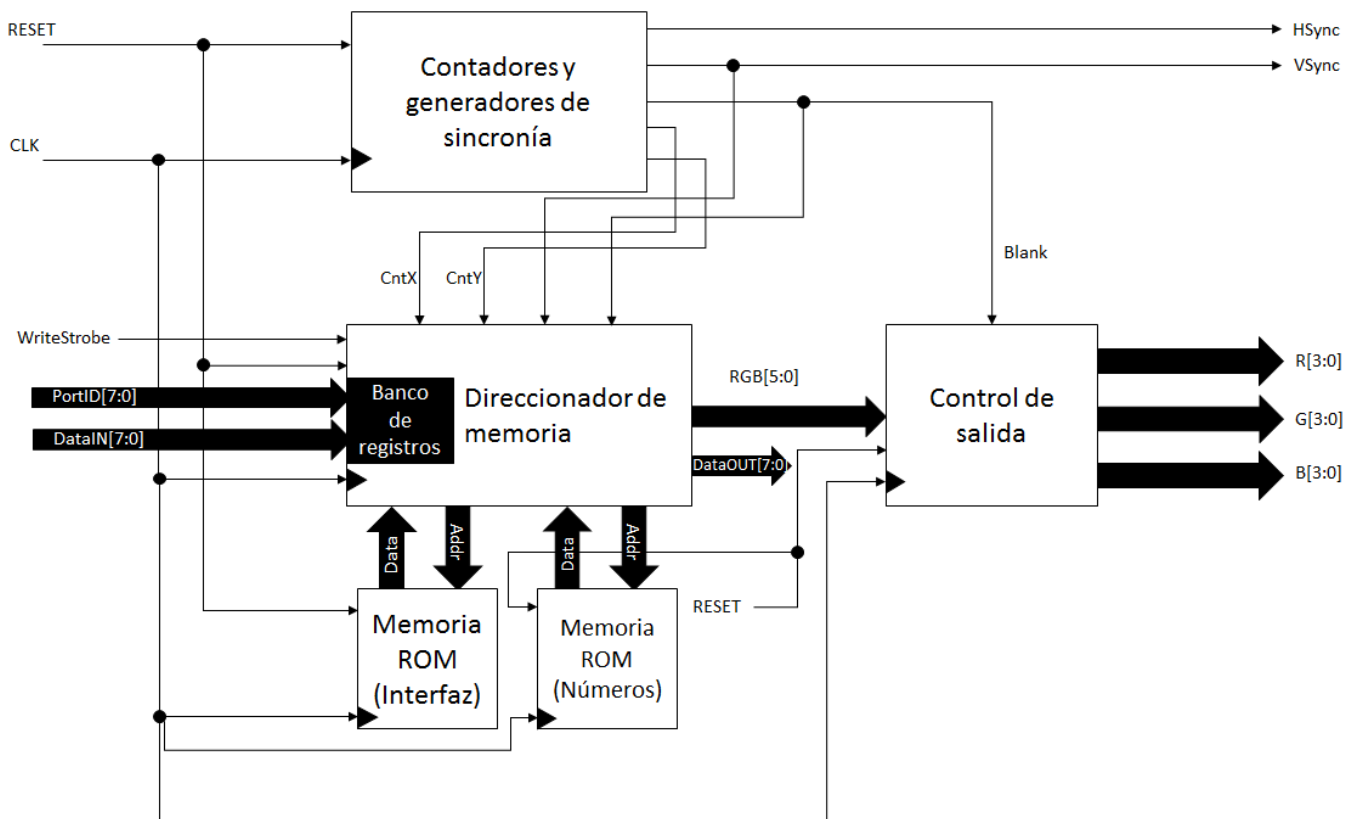


Figura 4: Diagrama de tercer nivel VGA

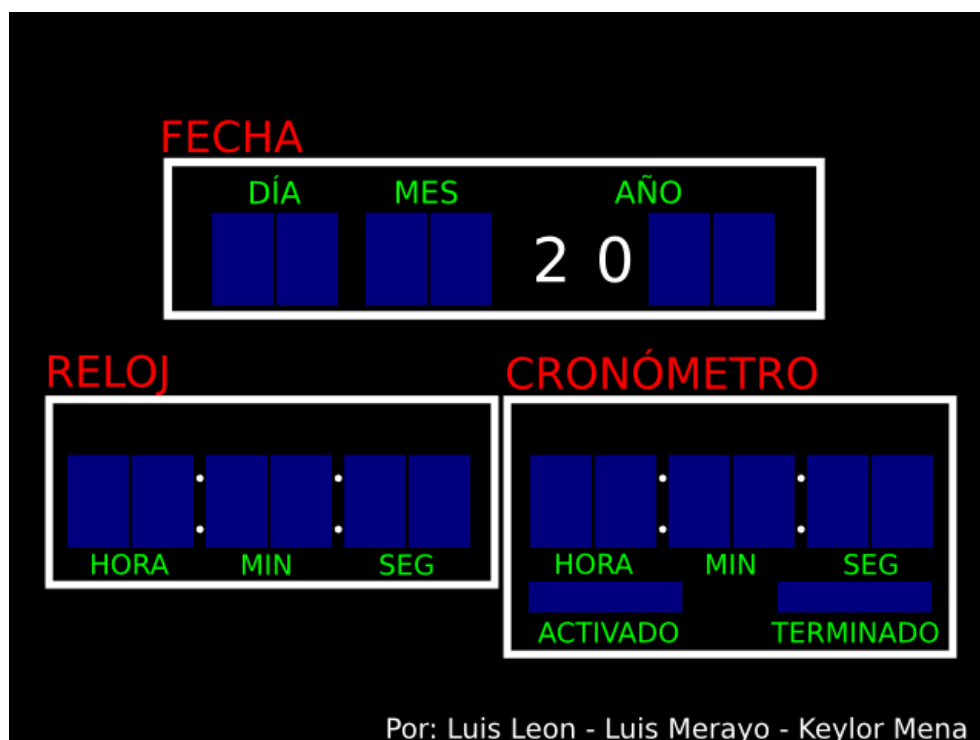


Figura 5: Interfaz de VGA

<b>Proyecto</b>	1	<b>Página</b>	6/26
<b>Trabajo</b>	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	<b>Actualizado en:</b>	6/11/2016
<b>Grupo</b>	8	<b>Revisado en:</b>	8/11/2016
<b>Revisado por:</b>	Alfonso Chacón Rodríguez	<b>Diseñadores</b>	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

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 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.



Figura 6: Números VGA

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.

<b>Proyecto</b>	1	<b>Página</b>	7/26
<b>Trabajo</b>	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	<b>Actualizado en:</b>	6/11/2016
<b>Grupo</b>	8	<b>Revisado en:</b>	8/11/2016
<b>Revisado por:</b>	Alfonso Chacón Rodríguez	<b>Diseñadores</b>	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

## 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:

Cuadro 1: Configuración de las teclas	
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

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, 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.

### 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.

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

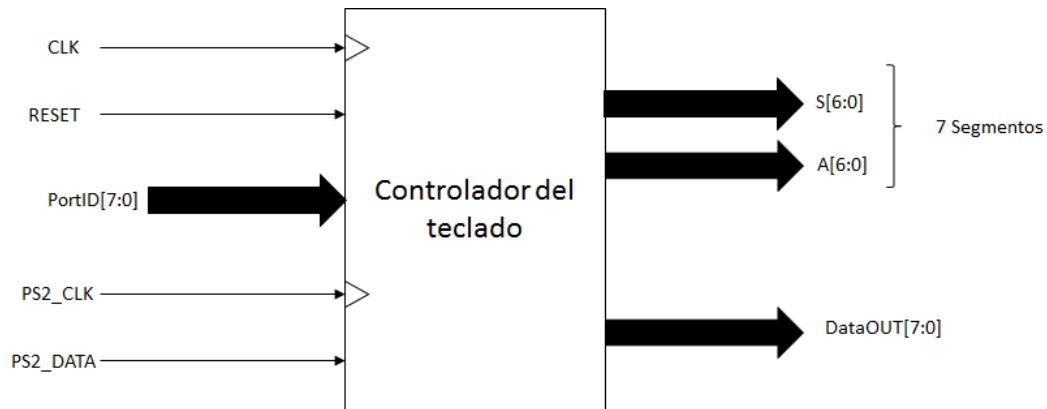


Figura 7: Diagrama de primer nivel KB

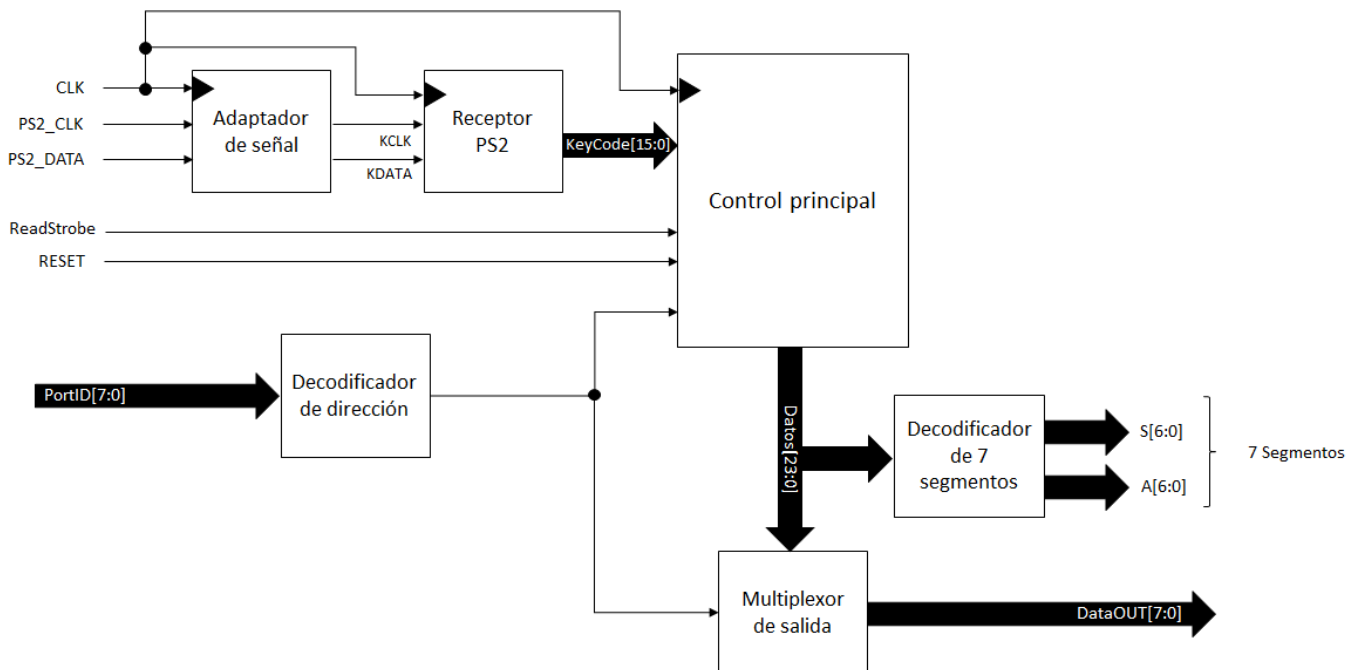


Figura 8: Diagrama de segundo nivel KB



<b>Proyecto</b>	1	<b>Página</b>	9/26
<b>Trabajo</b>	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	<b>Actualizado en:</b>	6/11/2016
<b>Grupo</b>	8	<b>Revisado en:</b>	8/11/2016
<b>Revisado por:</b>	Alfonso Chacón Rodríguez	<b>Diseñadores</b>	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

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.

## 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.

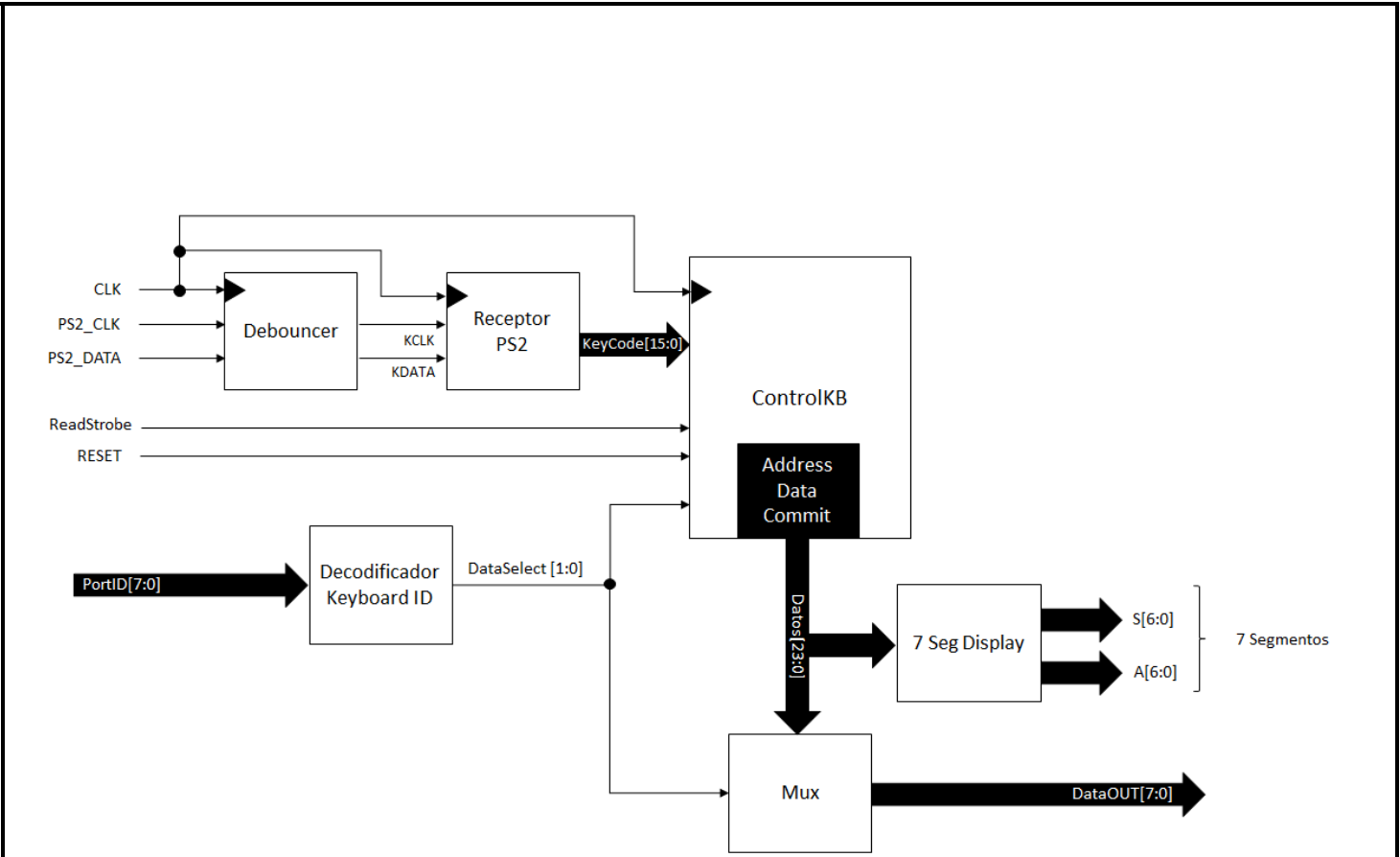


Figura 9: Diagrama de tercer nivel KB



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

<b>Proyecto</b>	1	<b>Página</b>	11/26
<b>Trabajo</b>	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	<b>Actualizado en:</b>	6/11/2016
<b>Grupo</b>	8	<b>Revisado en:</b>	8/11/2016
<b>Revisado por:</b>	Alfonso Chacón Rodríguez	<b>Diseñadores</b>	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

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.

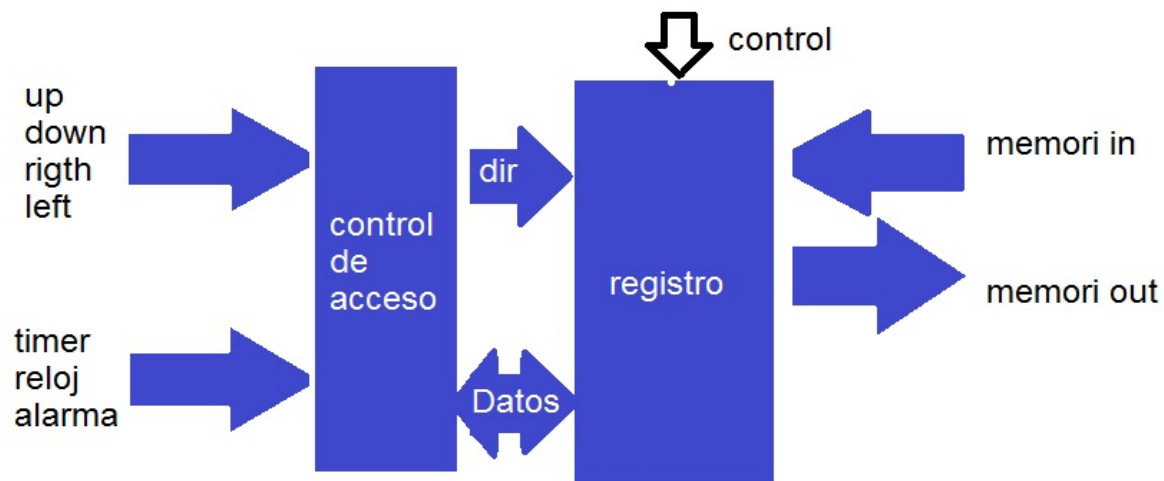


Figura 11: Diagrama de segundo nivel del Teclado.

## 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 inicialización, 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.

### 2.4.1. nivel 1 control RTC

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

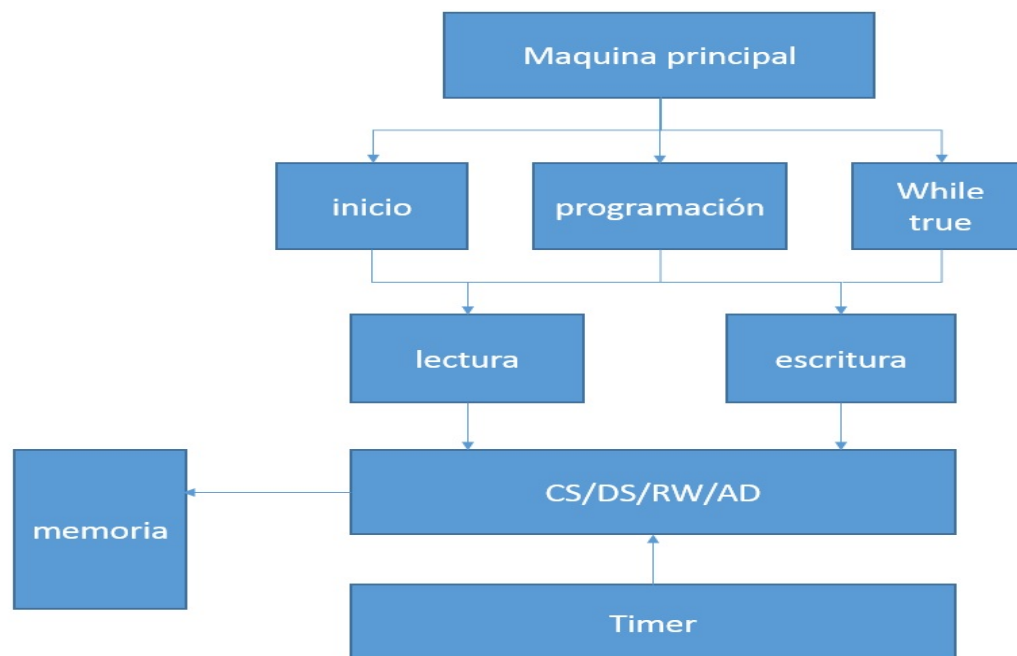


Figura 12: jerarquía de la RTC.

### 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.

<b>Proyecto</b>	1	<b>Página</b>	14/26
<b>Trabajo</b>	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	<b>Actualizado en:</b>	6/11/2016
<b>Grupo</b>	8	<b>Revisado en:</b>	8/11/2016
<b>Revisado por:</b>	Alfonso Chacón Rodríguez	<b>Diseñadores</b>	Keylor Mena Venegas Luis Leon Vega Luis Merayo Gatica

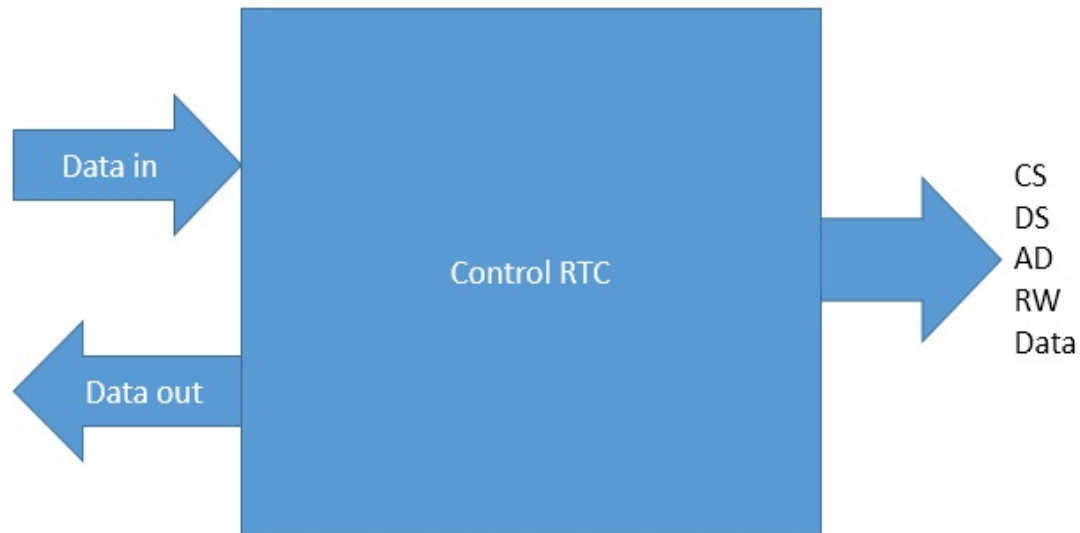


Figura 16: nivel 1 del RTC

#### 2.4.2. nivel 2 RTC

Esta etapa posee 4 bloques principales uno que controla la comunicación de la maquina 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 maquina 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 maquinas, 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 maquina controla que dato sale, si la dirección y o el dato, además tiene el bit de ciclo que determina si esta en el ciclo de escritura o lectura, como se nota en los flujos, el bit de ciclo no depende de la maquina, sino de la parte del programa donde esta se encuentre.

Por ultimo el control de salida responde al cuadro de la figura 15 a este le entra, el bit de activación que sale de la maquina 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.

#### 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 este efecto se buscaron crear 5 ciclos, como se muestra de la figura 20, donde el primer ciclo es el de inicialización falta diseño.

Después de este un ciclo de teclado, como se muestra en la figura 21, en este se usan los espacios

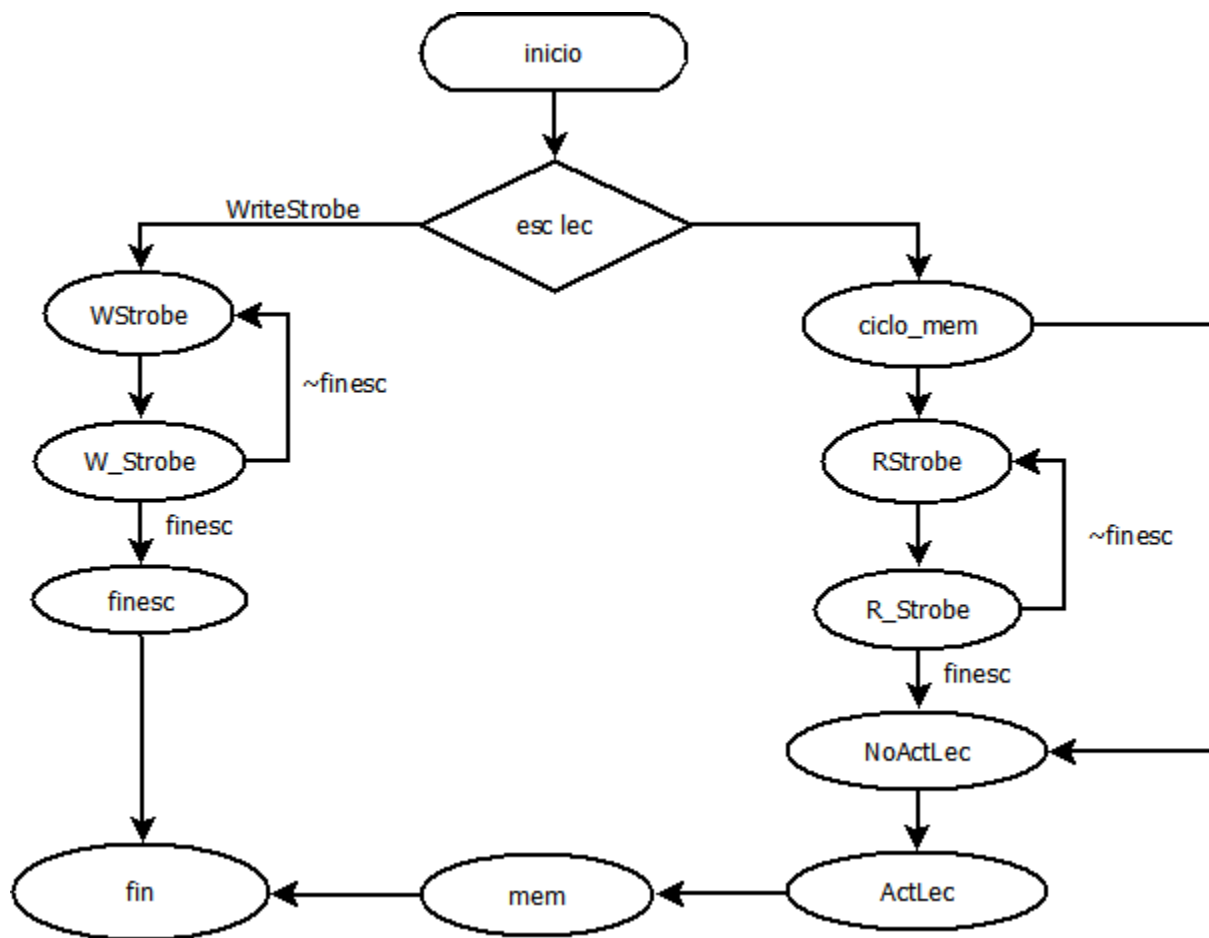


Figura 17: diagrama de la maquina de control de la RTC

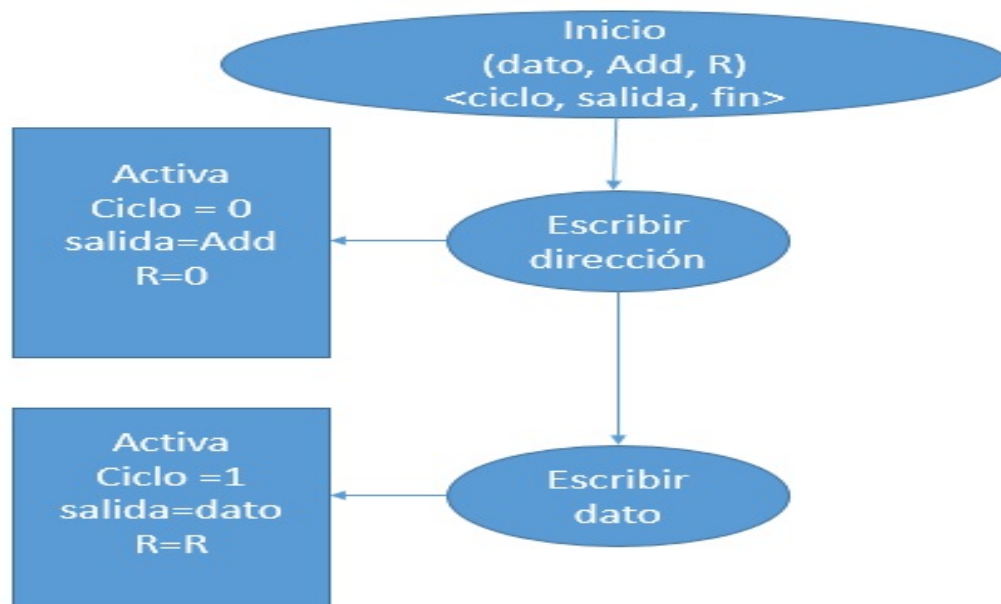


Figura 18: flujo de datos de la maquina de lectura

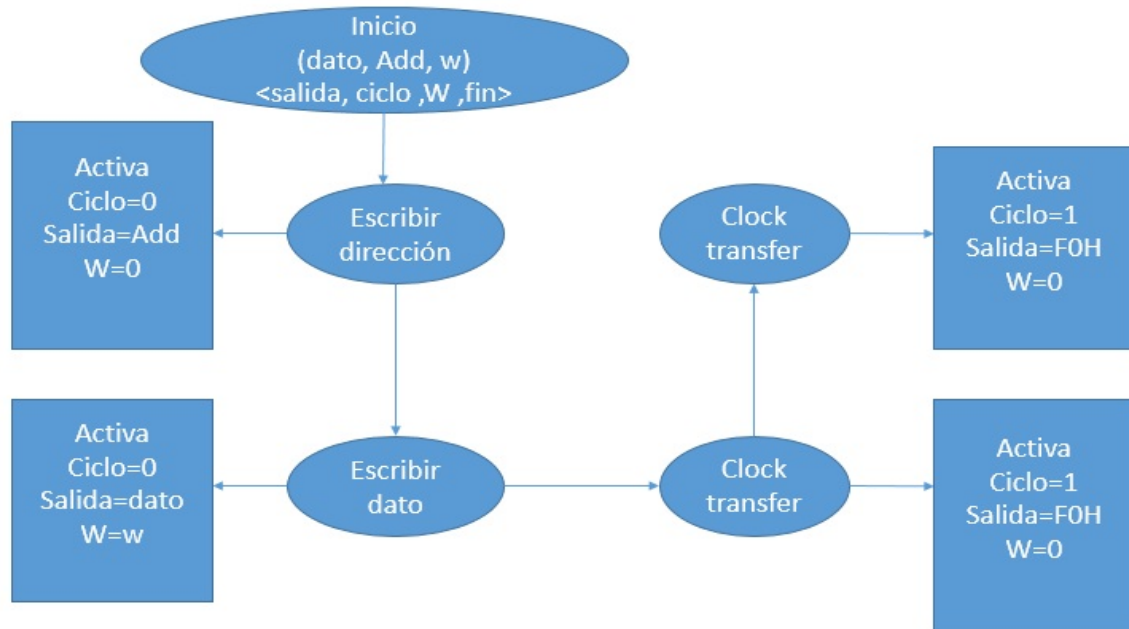
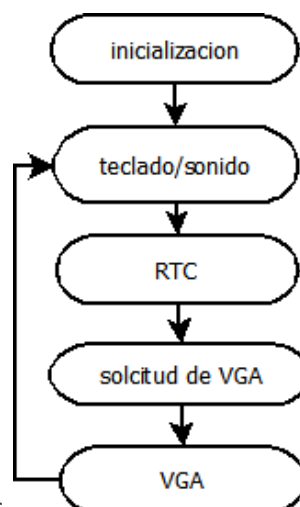


Figura 19: flujo de datos de la maquina de Escritura



De Flujo Principial.png

Figura 20: Diagrama de flujo principal



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.

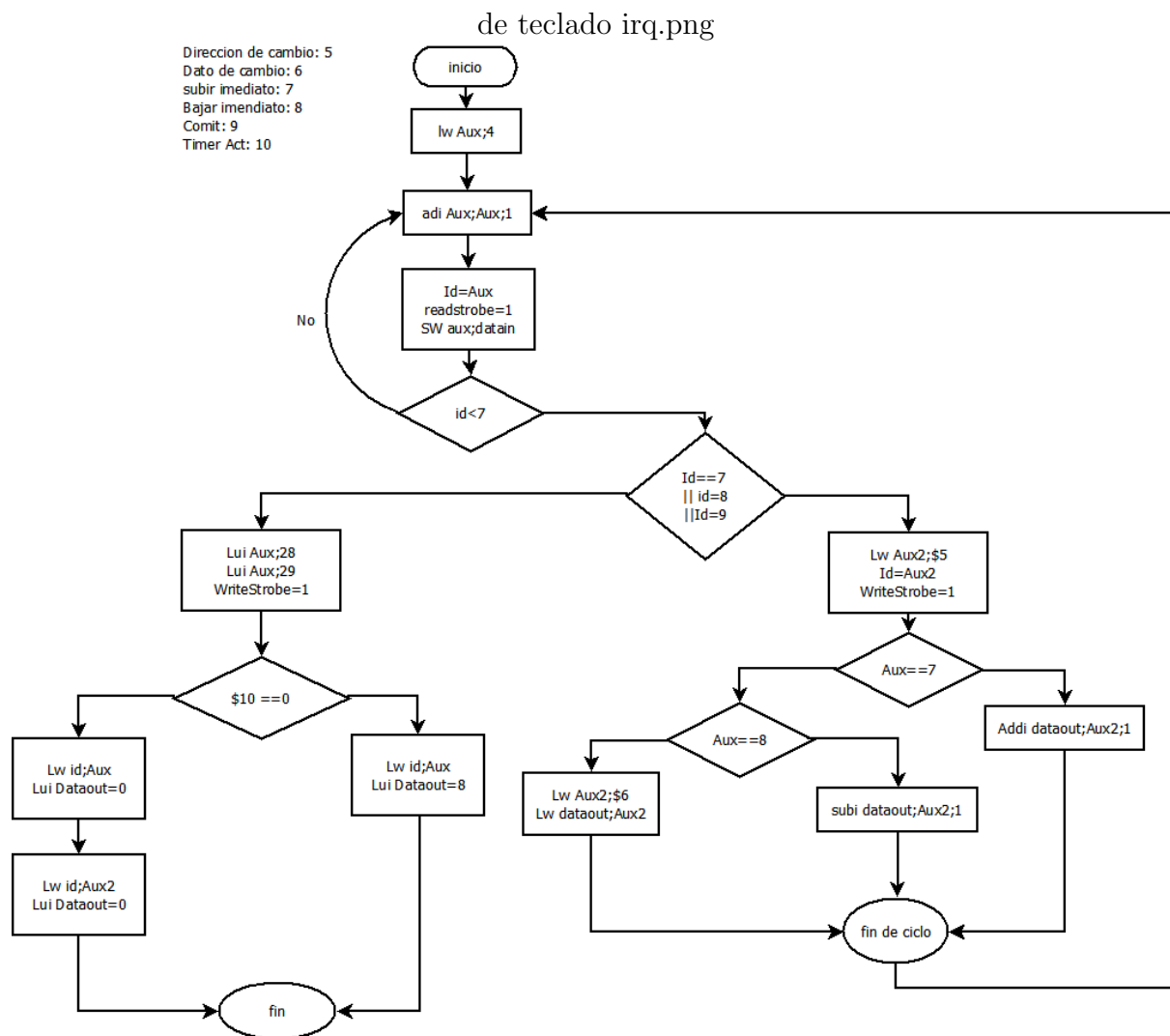
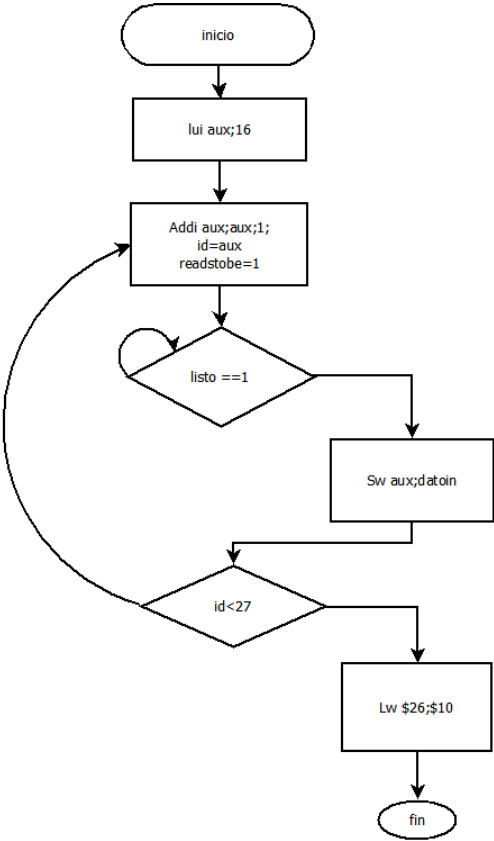


Figura 21: Diagrama de flujo principal

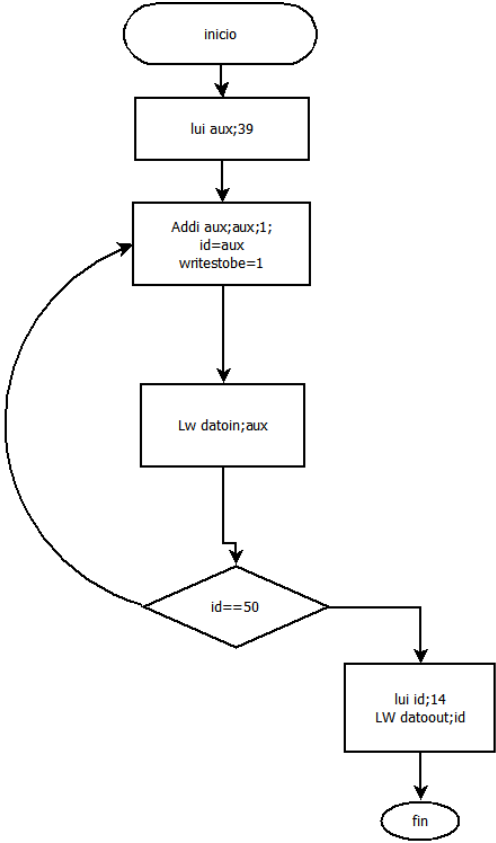
Variable	Deco	memoria
Segundos=	17	33
minutos=	18	34
horas=	19	35
Dias=	20	36
Meses=	21	37
Años=	22	38
SegundosT=	23	h41
minutosT=	24	h42
horasT=	25	h43
timerdisp	26	
timeract	27	
Staus1	28	0
Staus2	29	1



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

<b>Proyecto</b>	1	<b>Página</b>	20/26
<b>Trabajo</b>	Desarrollo de un reloj en tiempo real con Nexys 4 y V3023	<b>Actualizado en:</b>	6/11/2016
<b>Grupo</b>	8	<b>Revisado en:</b>	8/11/2016
<b>Revisado por:</b>	Alfonso Chacón Rodríguez	<b>Diseñadores</b>	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	21/26
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

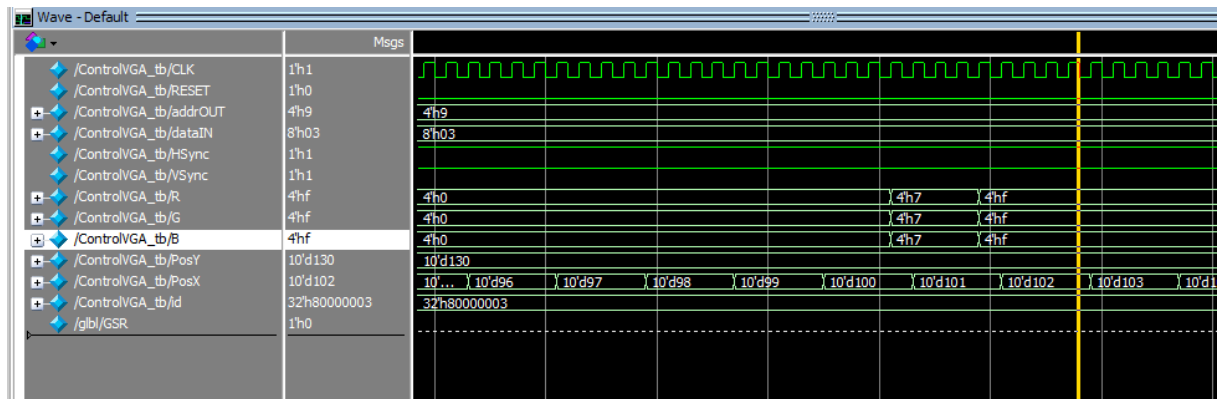


Figura 25: Transiciones de datos.

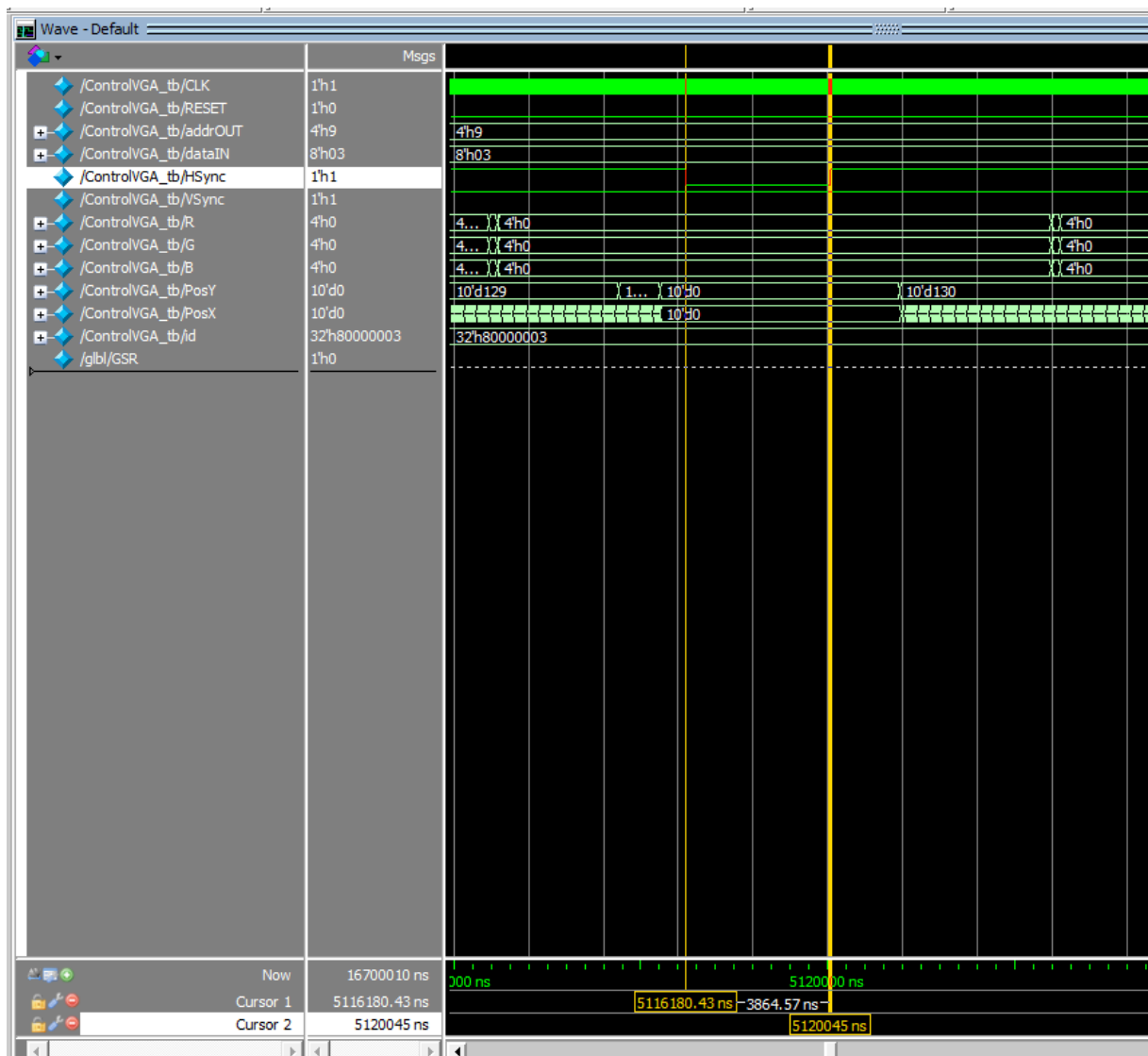


Figura 26: Tiempo de pulso HSync - Se cumple

Proyecto	1	Página	23/26
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

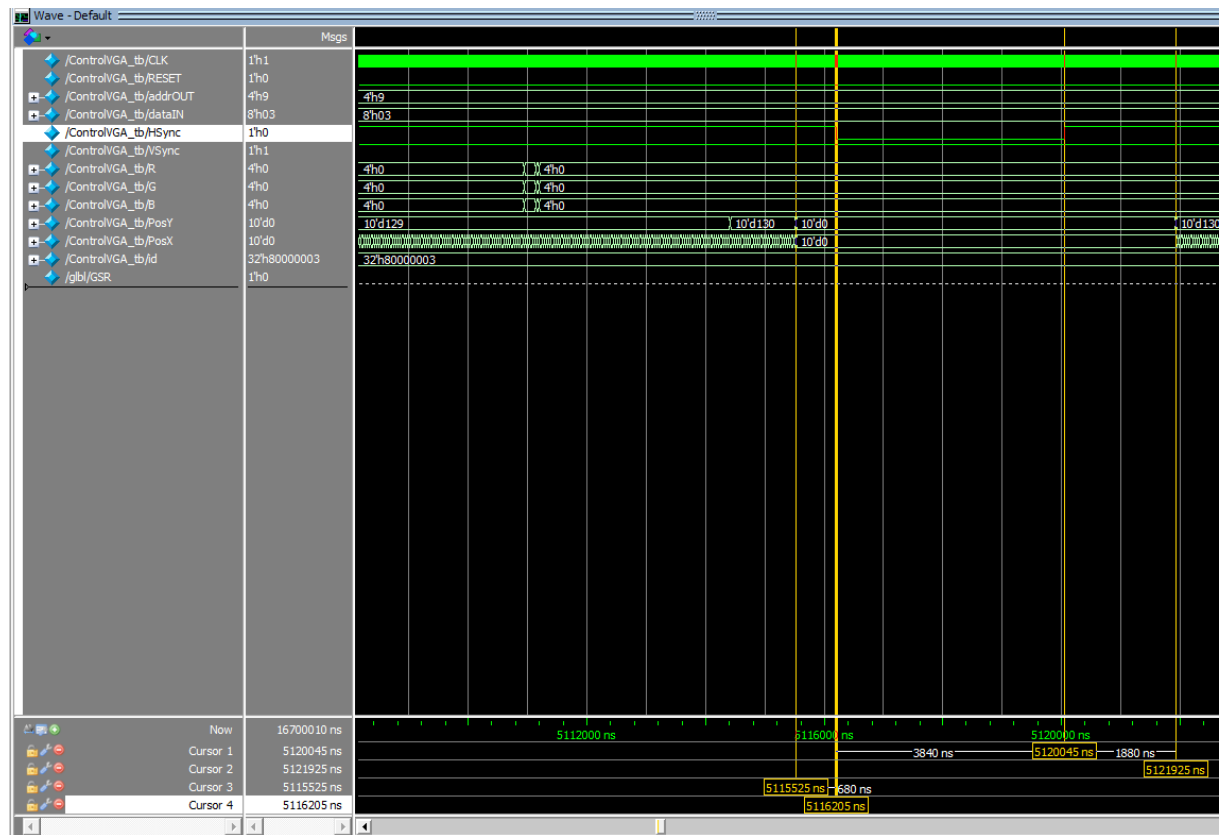


Figura 27: Tiempos en HSync - Porches

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:

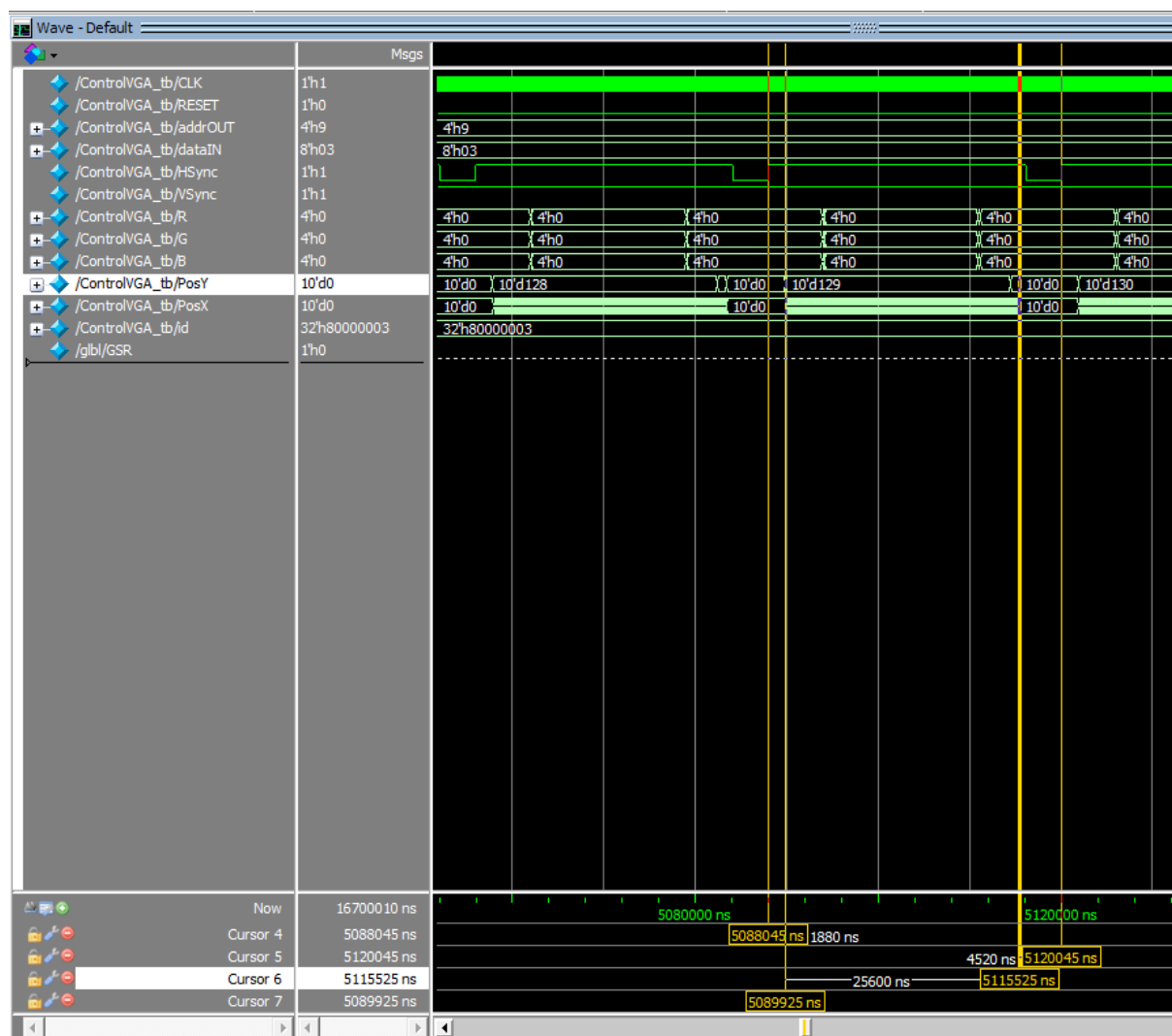


Figura 28: Tiempos en HSync en general



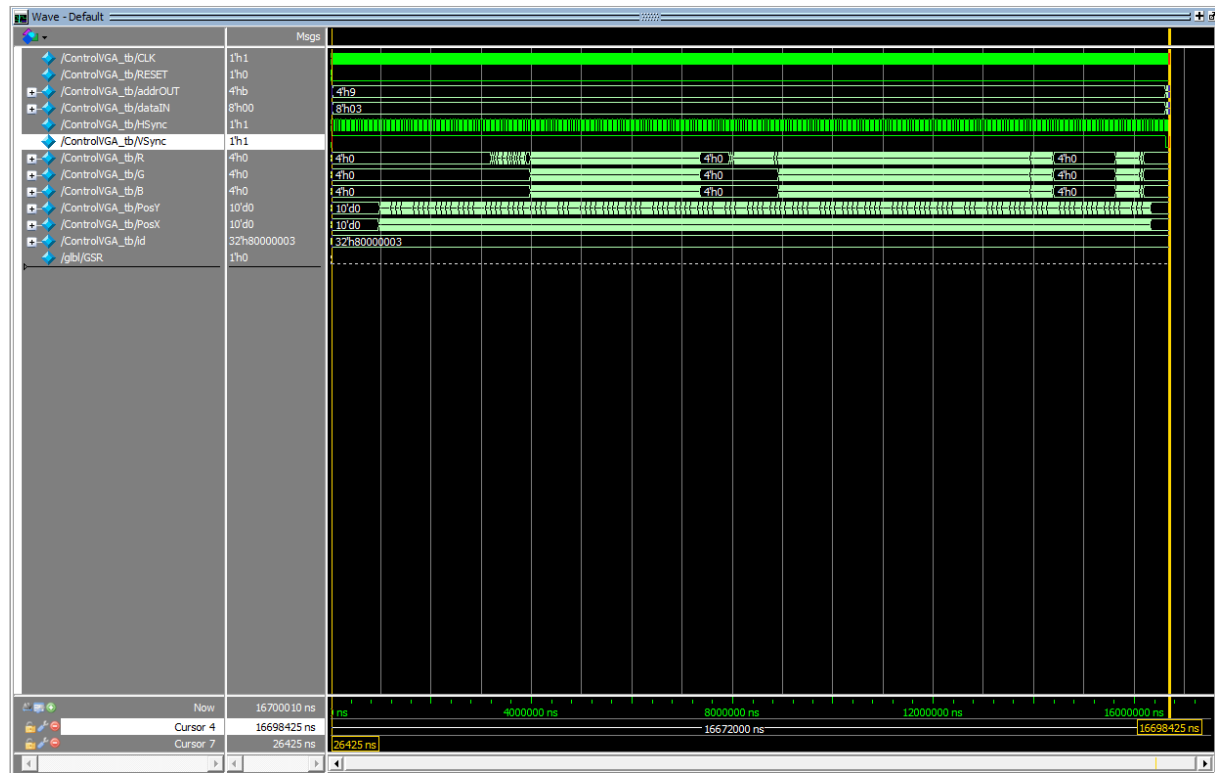


Figura 29: Tiempos en VSync

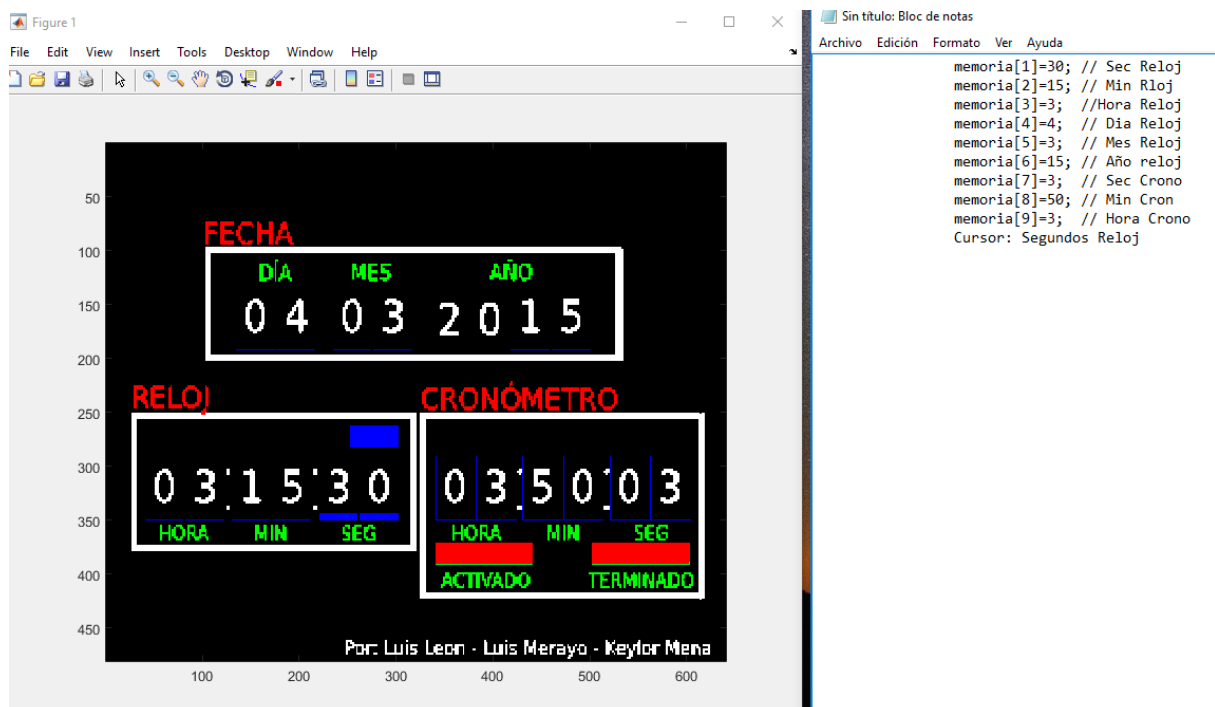


Figura 30: Tiempos en HSync en general

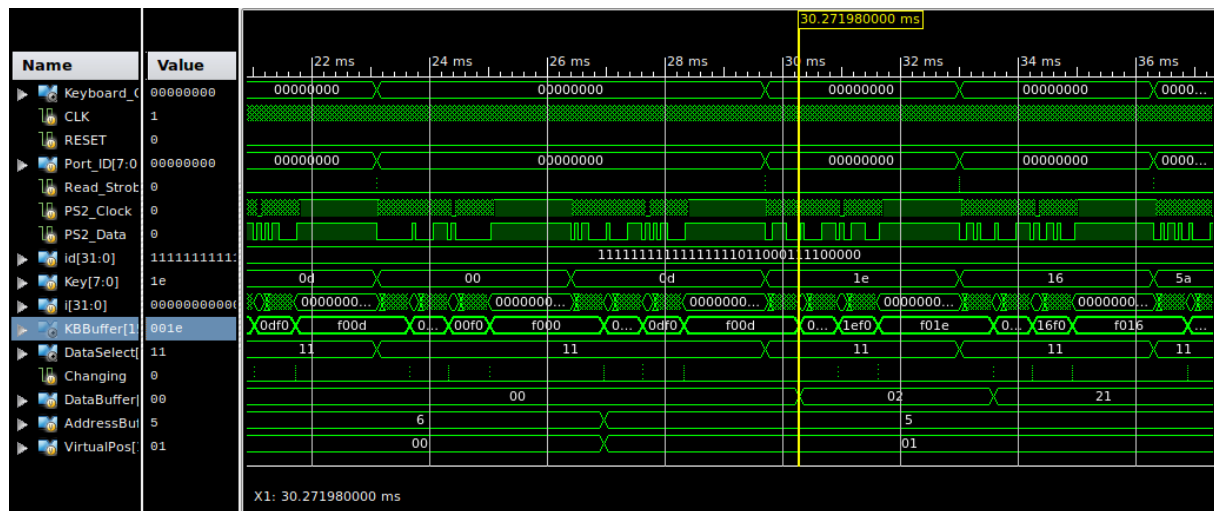


Figura 31: Simulación del teclado

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

## 4. Análisis de datos y resultados

## 5. Hoja de datos de unidades desarrolladas

## 6. Conclusiones y recomendaciones

### 6.1. Conclusiones

### 6.2. Recomendaciones

## 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].