

|                      |                                  |                        |   |
|----------------------|----------------------------------|------------------------|---|
| <b>Proyecto</b>      | 1                                | <b>Página</b>          | 1/26  |
| <b>Trabajo</b>       | Desarrollo de un controlador VGA | <b>Actualizado en:</b> | 27/08/2016  |
| <b>Grupo</b>         | 1                                | <b>Revisado en:</b>    | 30/08/2016  |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez         | <b>Diseñadores</b>     | Keylor Mena Venegas<br>Luis Leon Vega<br>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 controlador VGA | <b>Actualizado en:</b> | 27/08/2016          |
|                      | VGA                              | <b>Revisado en:</b>    | 30/08/2016          |
| <b>Grupo</b>         | 1                                | <b>Diseñadores</b>     | Keylor Mena Venegas |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez         |                        | 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.

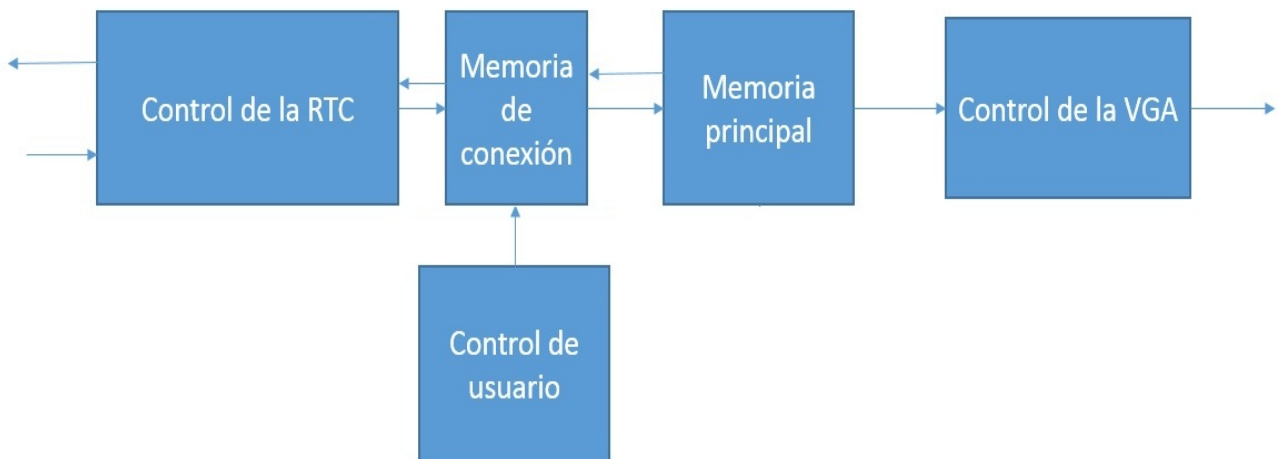


Figura 1: Diagrama de módulos principales del sistema.

## 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.

### 2.1.1. Diagrama de primer nivel

Para efectos del circuito, este debe tener acceso a los datos procedentes del buffer de memoria que contendrá la información que emite el RTC. 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.

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 ".AdrMem" permite direccionar el registro de la memoria de datos al registro que va a ser refrescado y el bus "DataMem" recopila la información del registro direccionado mediante ".AdrMem". 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.

|               |                                  |                 |                     |
|---------------|----------------------------------|-----------------|---------------------|
| Proyecto      | 1                                | Página          | 3/26                |
| Trabajo       | Desarrollo de un controlador VGA | Actualizado en: | 27/08/2016          |
|               | VGA                              | Revisado en:    | 30/08/2016          |
| Grupo         | 1                                | Diseñadores     | Keylor Mena Venegas |
| Revisado por: | Alfonso Chacón Rodríguez         |                 | Luis Leon Vega      |
|               |                                  |                 | Luis Merayo Gatica  |

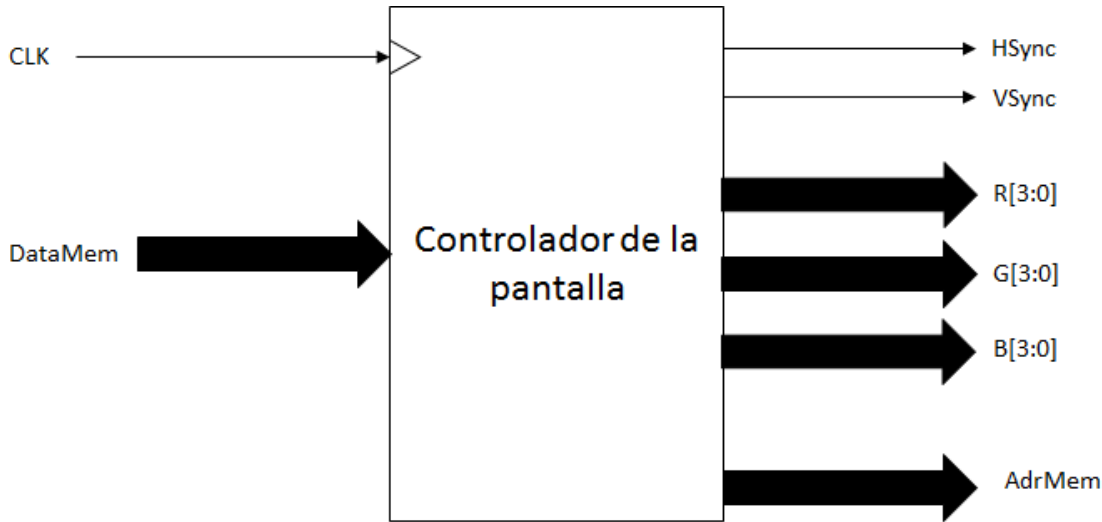


Figura 2: Diagrama de primer nivel VGA

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.

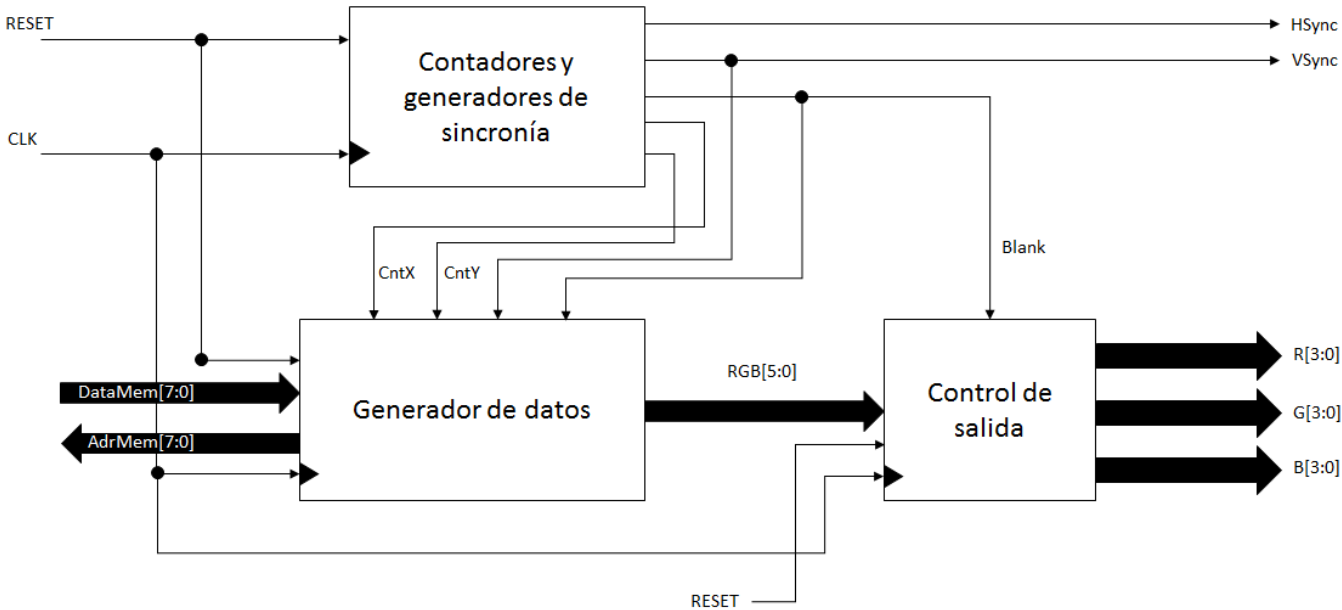


Figura 3: Diagrama de primer nivel VGA

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

|                      |                                  |                        |   |
|----------------------|----------------------------------|------------------------|---|
| <b>Proyecto</b>      | 1                                | <b>Página</b>          | 4/26  |
| <b>Trabajo</b>       | Desarrollo de un controlador VGA | <b>Actualizado en:</b> | 27/08/2016  |
| <b>Grupo</b>         | 1                                | <b>Revisado en:</b>    | 30/08/2016  |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez         | <b>Diseñadores</b>     | Keylor Mena Venegas<br>Luis Leon Vega<br>Luis Merayo Gatica |

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 .AdrMemz "DataMem", 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, 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.

### 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.

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

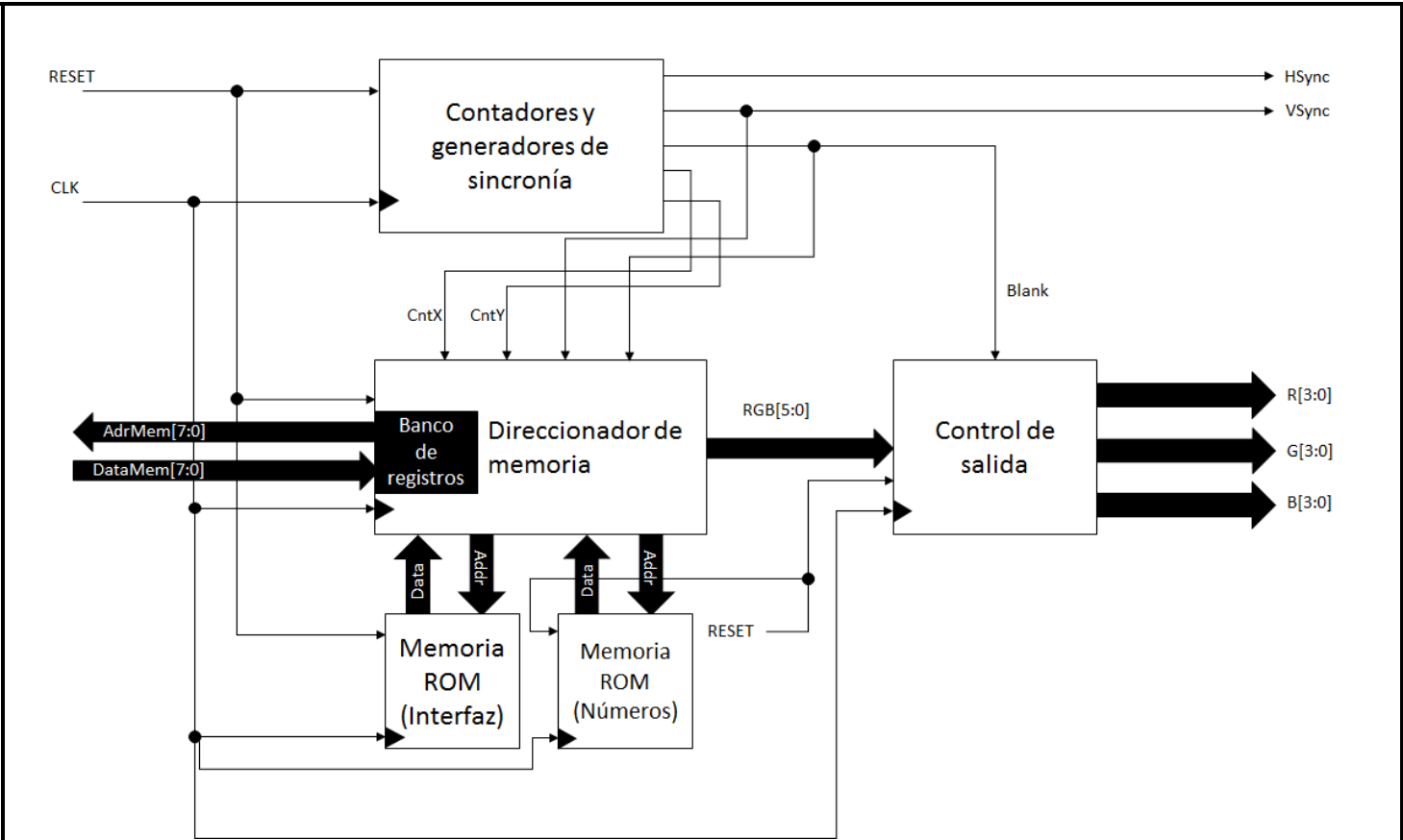


Figura 4: Diagrama de primer nivel VGA

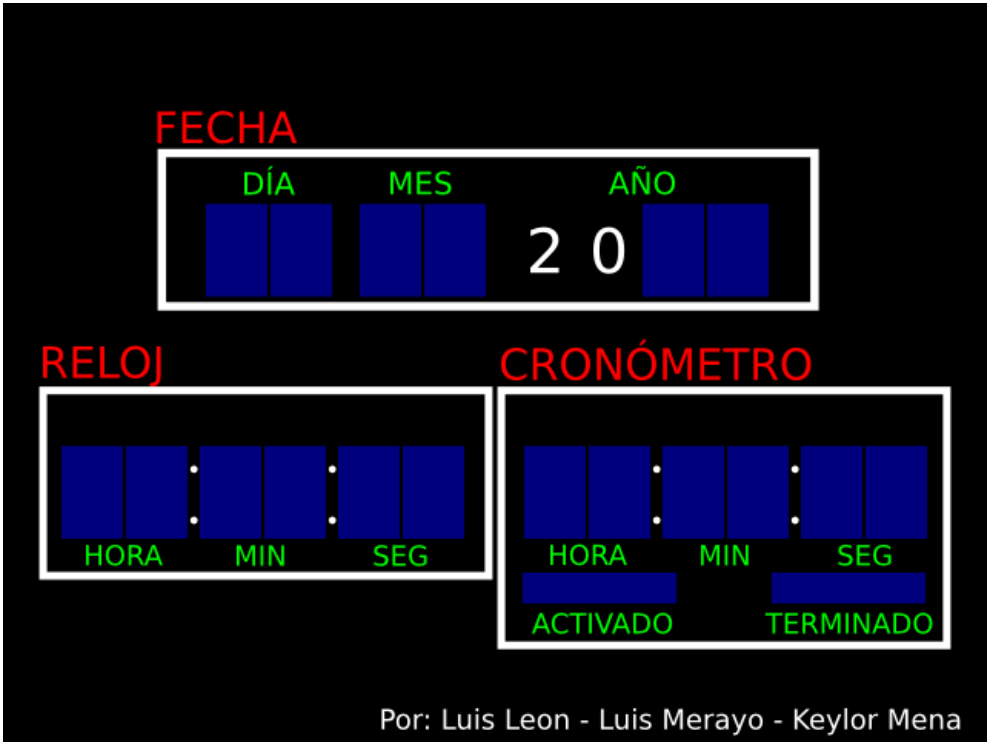


Figura 5: Diagrama de primer nivel VGA

|                      |                                  |                        |   |
|----------------------|----------------------------------|------------------------|---|
| <b>Proyecto</b>      | 1                                | <b>Página</b>          | 6/26  |
| <b>Trabajo</b>       | Desarrollo de un controlador VGA | <b>Actualizado en:</b> | 27/08/2016  |
| <b>Grupo</b>         | 1                                | <b>Revisado en:</b>    | 30/08/2016  |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez         | <b>Diseñadores</b>     | Keylor Mena Venegas<br>Luis Leon Vega<br>Luis Merayo Gatica |

mediante un archivo de texto con 6 bits.



Figura 6: Diagrama de primer nivel 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.

## 2.2. Control de usuario

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

|               |                                  |                 |   |
|---------------|----------------------------------|-----------------|---|
| Proyecto      | 1                                | Página          | 7/26  |
| Trabajo       | Desarrollo de un controlador VGA | Actualizado en: | 27/08/2016  |
| Grupo         | 1                                | Revisado en:    | 30/08/2016  |
| Revisado por: | Alfonso Chacón Rodríguez         | Diseñadores     | Keylor Mena Venegas<br>Luis Leon Vega<br>Luis Merayo Gatica |

disminuir sus valores y 3 interruptores que [ermiten realizar estos cambios.

### 2.2.1. Diagrama de primer nivel

el control de usuario posee 3 interruptores y 3 botones para que el usuario elija los datos y que desea cambiar. Ademas posee entradas y salidas de memoria para poder alterar los registros y escribirlos en la rtc. Se puede notar esto en la figura 7.



Figura 7: Diagrama de primer nivel del control de usuario.

### 2.2.2. Diagrama de segundo nivel

En este diagrama mostrado en la Fig. 8 se muestra como se pretende realizar el control de usuario, el cual consiste en tan solo dos bloques.

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 almacenara los cambios que se desean realizar en la memoria y una etapa de salida que controla los momentos donde se escribe en la rtc los cambios que se desean en la rtc.

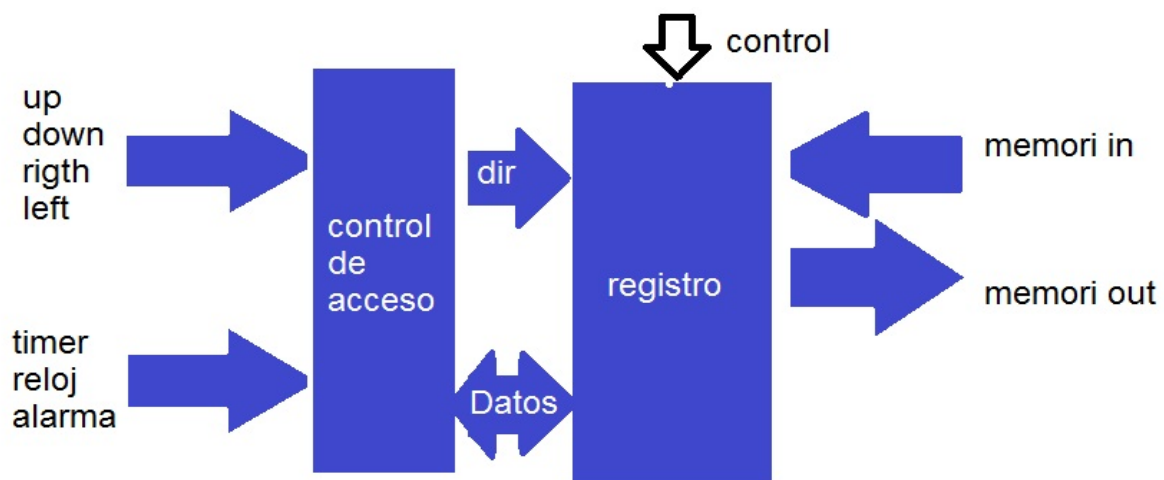


Figura 8: Diagrama de segundo nivel del Teclado.

|                      |                              |                        |                     |
|----------------------|------------------------------|------------------------|---------------------|
| <b>Proyecto</b>      | 1                            | <b>Página</b>          | 8/26                |
| <b>Trabajo</b>       | Desarrollo de un controlador | <b>Actualizado en:</b> | 27/08/2016          |
|                      | VGA                          | <b>Revisado en:</b>    | 30/08/2016          |
| <b>Grupo</b>         | 1                            | <b>Diseñadores</b>     | Keylor Mena Venegas |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez     |                        | Luis Leon Vega      |
|                      |                              |                        | Luis Merayo Gatica  |

## 2.3. 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 9. 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, este consta del punto visto anteriormente 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 10 y la figura 11, 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 12

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.

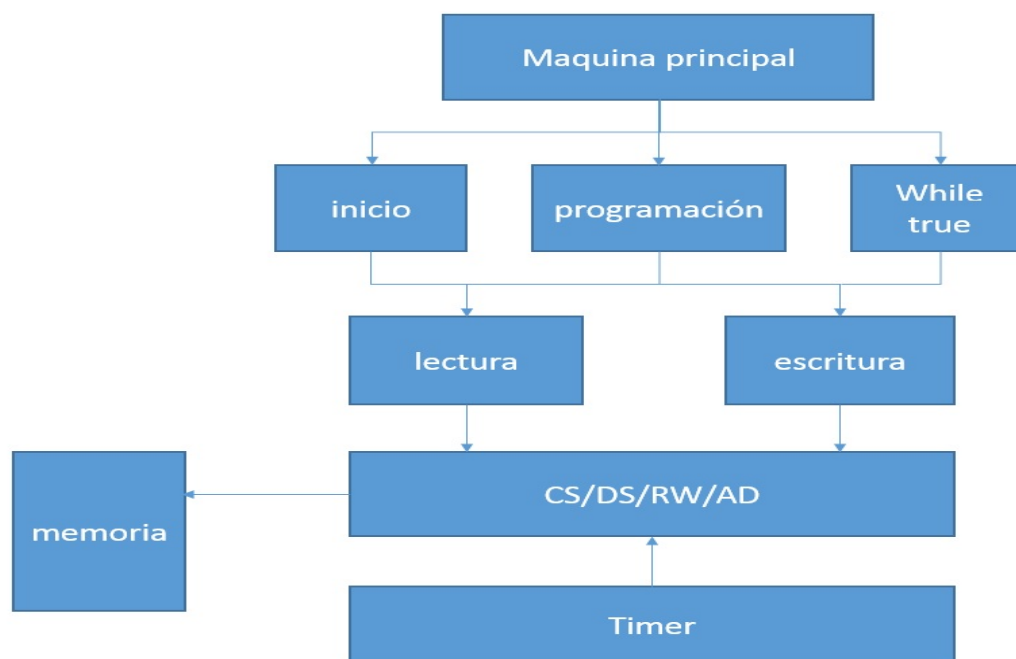


Figura 9: jerarquía de la RTC.

### 2.3.1. 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 13.

### 2.3.2. nivel 2 RTC

Como se menciona antes existen 6 maquina de estados agrupados en 3 jerarquías.



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 10: 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 11: 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 12: cuadro de tiempos del control de salida.

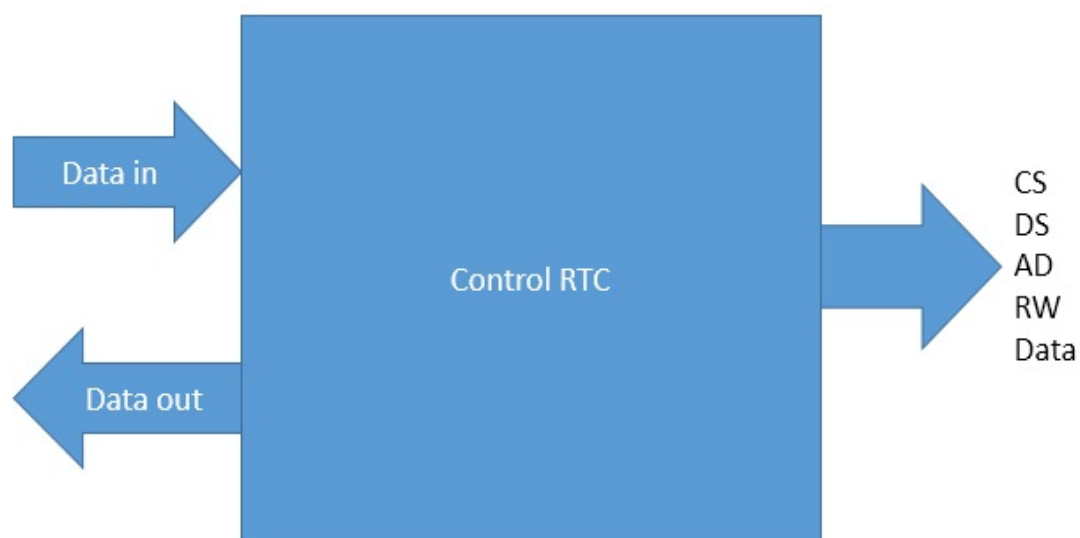


Figura 13: nivel 1 del RTC

|                      |                                     |                        |   |
|----------------------|-------------------------------------|------------------------|---|
| <b>Proyecto</b>      | 1                                   | <b>Página</b>          | 11/26   |
| <b>Trabajo</b>       | Desarrollo de un controlador<br>VGA | <b>Actualizado en:</b> | 27/08/2016  |
| <b>Grupo</b>         | 1                                   | <b>Revisado en:</b>    | 30/08/2016  |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez            | <b>Diseñadores</b>     | Keylor Mena Venegas<br>Luis Leon Vega<br>Luis Merayo Gatica |

La primera maquina de estados es la de inicializacion, esta tiene un bit de entrada para su inicializacion. Ademas posee salidas de datos y dirección y bits W y R, que siempre están en cero y se conectan a las maquinas de lectura y escritura respectivamente, esto responde al flujo de la figura 14.

La maquina encargada del proceso while true responde al flujo de la figura, esta tiene un bit para la iniciación de la maquina y un bit que indica que finalizo el proceso, de igual manera un bit que indica que la maquina de estados siguiente termino su proceso para que esta salte entre estados y igual que la maquina pasada tiene las señales W y R y la salida de datos y dirección.

Siguiendo la jerarquía, existen 2 maquinas, escritura y lectura, estas respetan los flujos de la figura 15 y 16, 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 las maquina controla que dato sale, si la direccion y o el dato, ademas 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 12 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 12.

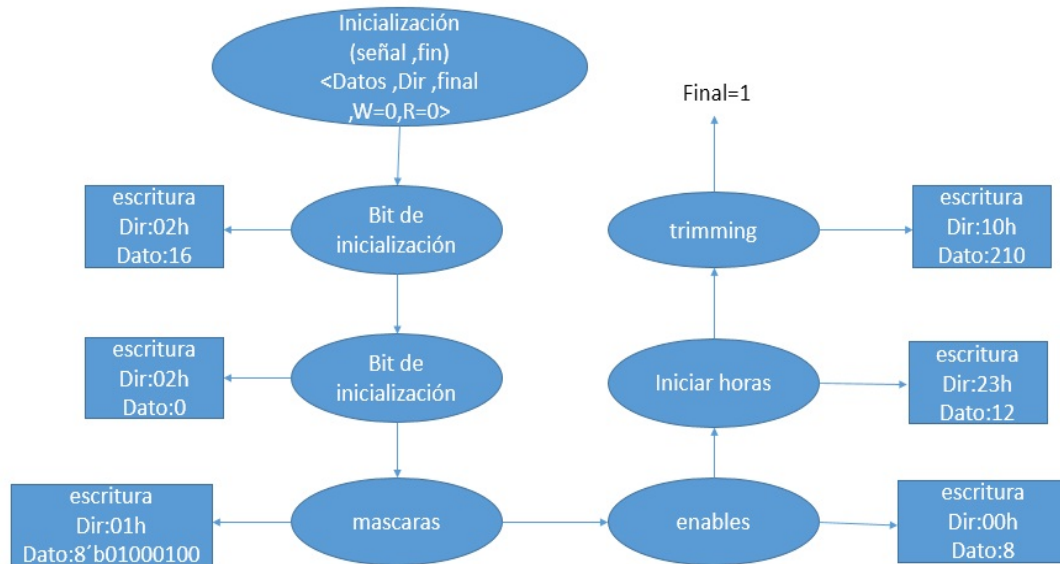


Figura 14: flujo de datos de la maquina de estados de inicializacion

## 2.4. Control general de la RTC

Debido a que existen muchas maquinas con diferentes jerarquia se ocupa diseñar una maquina de estado genral que controle y vigile el funcionamiento entre la maquinas de estados. Esta maquina posee

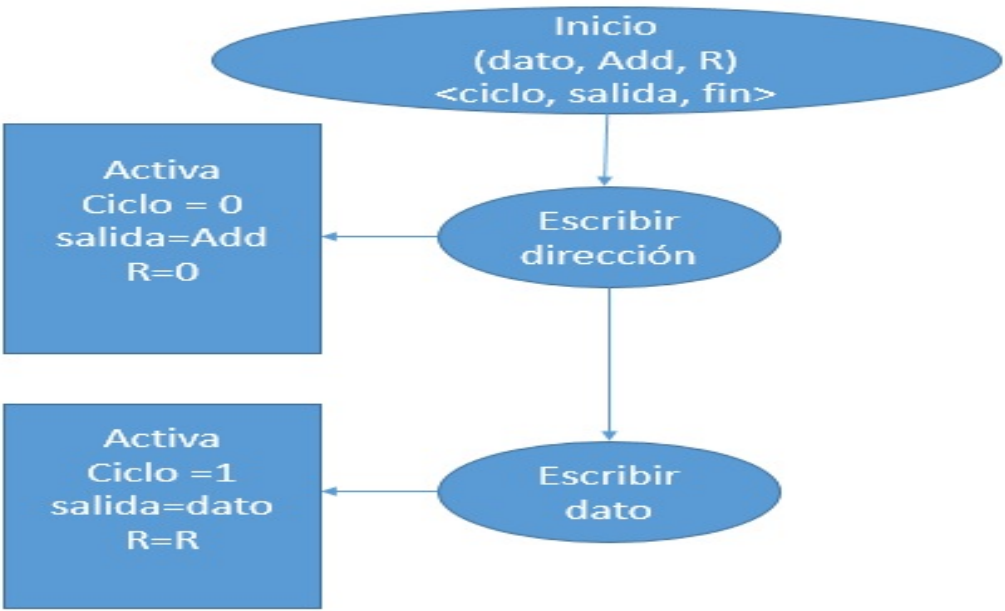


Figura 15: flujo de datos de la maquina de lectura

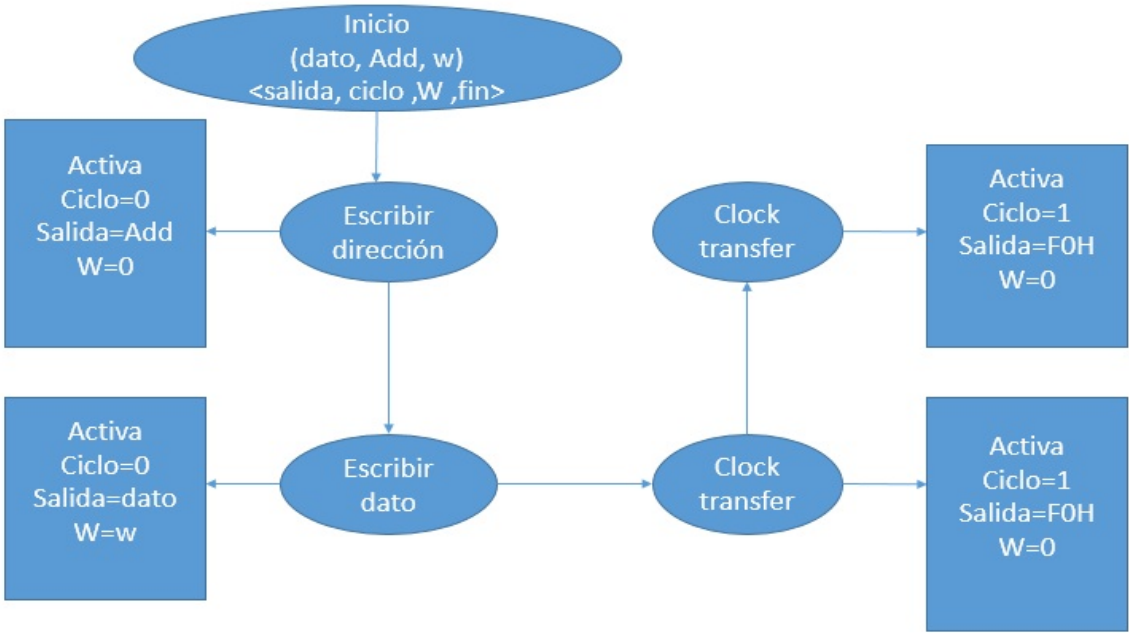


Figura 16: flujo de datos de la maquina de Escritura

|                      |                                     |                        |   |
|----------------------|-------------------------------------|------------------------|---|
| <b>Proyecto</b>      | 1                                   | <b>Página</b>          | 13/26   |
| <b>Trabajo</b>       | Desarrollo de un controlador<br>VGA | <b>Actualizado en:</b> | 27/08/2016  |
| <b>Grupo</b>         | 1                                   | <b>Revisado en:</b>    | 30/08/2016  |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez            | <b>Diseñadores</b>     | Keylor Mena Venegas<br>Luis Leon Vega<br>Luis Merayo Gatica |

5 niveles la inicializacion de la RTC, la inicializacion de la RTC en cero, el inicio de lectura continua, la solicitud de escritura y la escritura de los cambios en la memoria.

#### **2.4.1. inicializacion**

Este consta del primer estado del control de la RTC, este estado consta de los parametros de e instrucciones que deban introducirse en la RTC para que esta comience a correr el tiempo, ademas de la escriura de la mascarar en los estados 0 y 1 de la RTC, que nos permiten poner las banderas del timer y iniciar con el cronometro detenido. Aparte de ser el estado inicial de toda la RTC, este es el estodo donde este vuelve cada vez que es reinicia.

#### **2.4.2. inicializacion de la memoria en cero**

El objetivo de este estado es que todos los datos de la memoria inicie en valores conocidos cada vez que se reinicia o inicia la RTC. Para este efecto se aprovechara el efecto de escritura en la rtc que tiene el control de usuario. Como el control de usuario escribe lo que este en la memoria, mas un cambio en la RTC, en la RTC. Si no existe cambios en la memoria en la RTC se escribirán los valore iniciales de la memoria, como esta inicia en cero, la RTC iniciara a contar desde cero.

#### **2.4.3. lectura continua**

Este bloque es el que lee los valores que se encuentra cambiando en la RTC, este es el modulo que actualiza constantemente los valores de la RTC en la memoria dentro de la FPGA. Este estado genera una pausa mientras la memoria se estabiliza para mea estados.

#### **2.4.4. solicitud de control de usuario**

Para controlar la escritura en el control de usuario no se escribirá dentro de la RTC todo el tiempo, solo cuando termine el ciclo de lectura continua y el usurario este realizando cambios, este es un estado transitorio, en este estado puede pasar al estado de lectura continua o a la escritura de cambios

#### **2.4.5. escritura de cambios**

Esta etapa activa el control de usuario para poder alterar los valores de la RTC con los valores que el usuario desea alterar. Esta etap solo se accede cuando el usurio desea realizar cambios en la RTC.

### **3. Datos y resultados**

#### **3.1. Simulaciones**

Se realizó la simulacion post-síntesis de todos los módulos del proyecto, de manera que se pueda realizar una verificacion de su funcionamiento, y del cumplimiento de los tiempos de escritura y lectura en la RTC.

##### **3.1.1. Simulaciones VGA**

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

|                      |                                  |                        |                     |
|----------------------|----------------------------------|------------------------|---------------------|
| <b>Proyecto</b>      | 1                                | <b>Página</b>          | 14/26               |
| <b>Trabajo</b>       | Desarrollo de un controlador VGA | <b>Actualizado en:</b> | 27/08/2016          |
|                      | VGA                              | <b>Revisado en:</b>    | 30/08/2016          |
| <b>Grupo</b>         | 1                                | <b>Diseñadores</b>     | Keylor Mena Venegas |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez         |                        | Luis Leon Vega      |
|                      |                                  |                        | Luis Merayo Gatica  |

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 23.

```

/*
  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 DCMMinuteX = 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 17: Punteros de memoria

### 3.1.2. Simulaciones RTC

## 3.2. Tiempos RTC simulacion post-síntesis

### 3.3. 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 1 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:

**Proyecto** 1  
**Trabajo** Desarrollo de un controlador  
 VGA  
**Grupo** 1  
**Revisado por:** Alfonso Chacón Rodríguez

**Página** 15/26  
**Actualizado en:** 27/08/2016  
**Revisado en:** 30/08/2016  
**Diseñadores** Keylor Mena Venegas  
 Luis Leon Vega  
 Luis Merayo Gatica

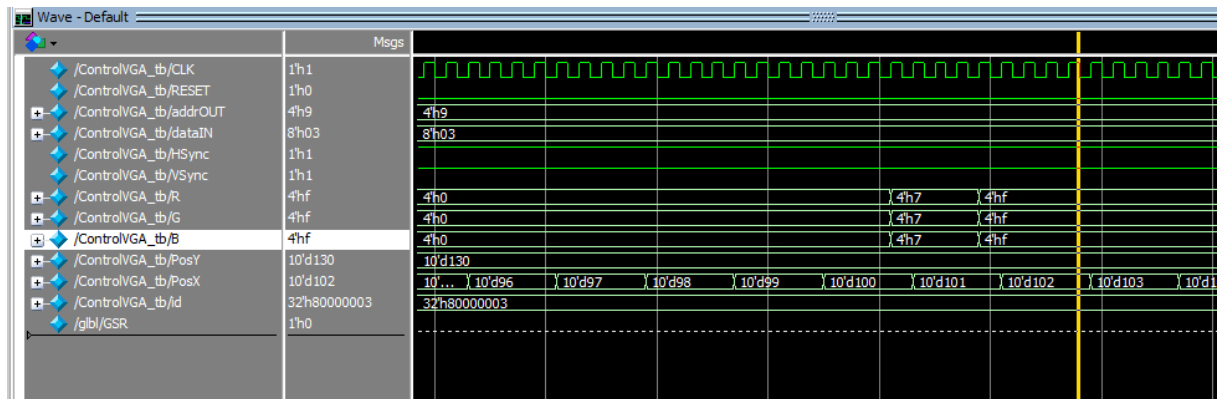


Figura 18: Transiciones de datos.

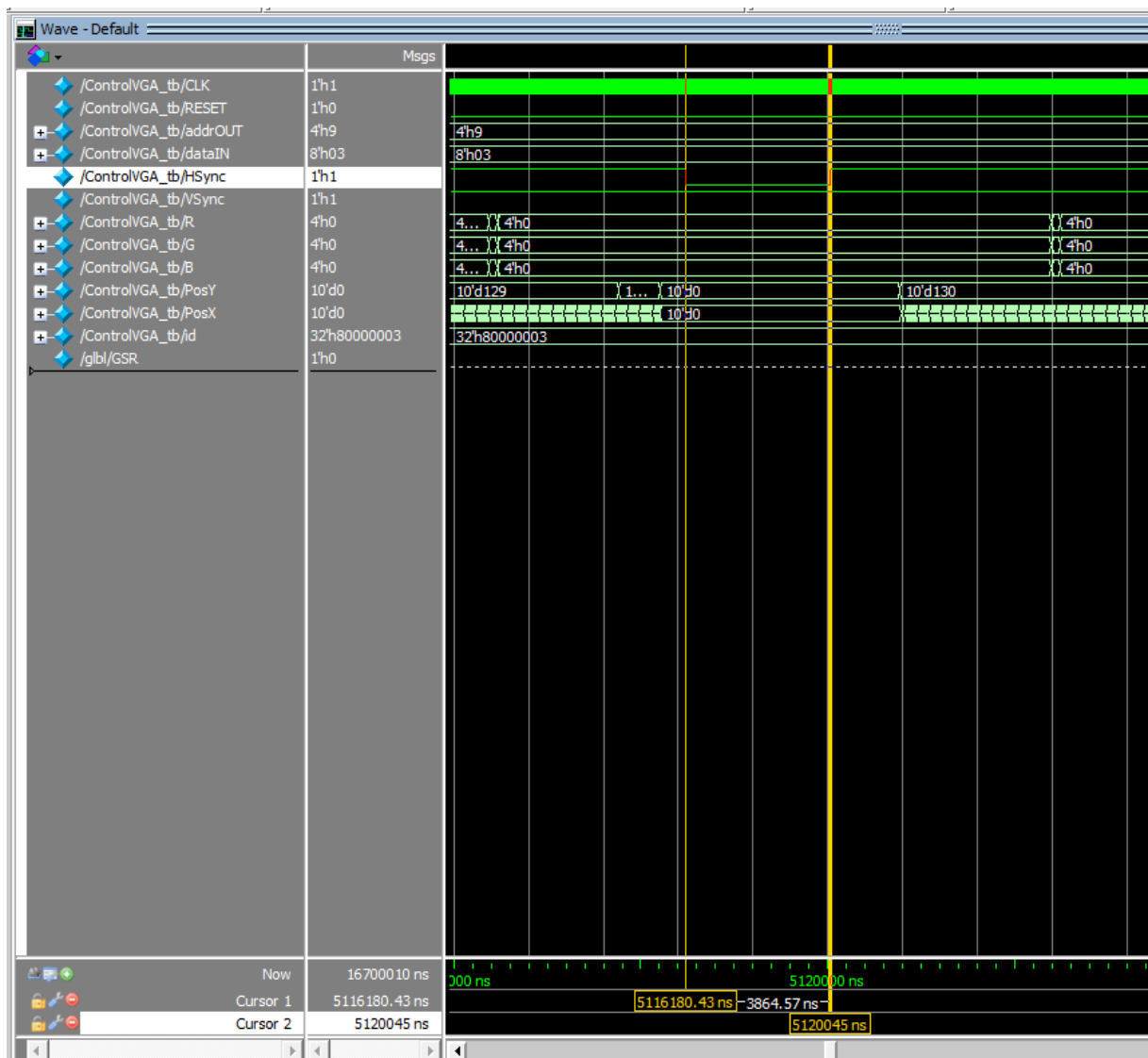


Figura 19: Tiempo de pulso HSync - Se cumple

**Proyecto** 1  
**Trabajo** Desarrollo de un controlador VGA  
**Grupo** 1  
**Revisado por:** Alfonso Chacón Rodríguez

**Página** 16/26  
**Actualizado en:** 27/08/2016  
**Revisado en:** 30/08/2016  
**Diseñadores** Keylor Mena Venegas  
 Luis Leon Vega  
 Luis Merayo Gatica

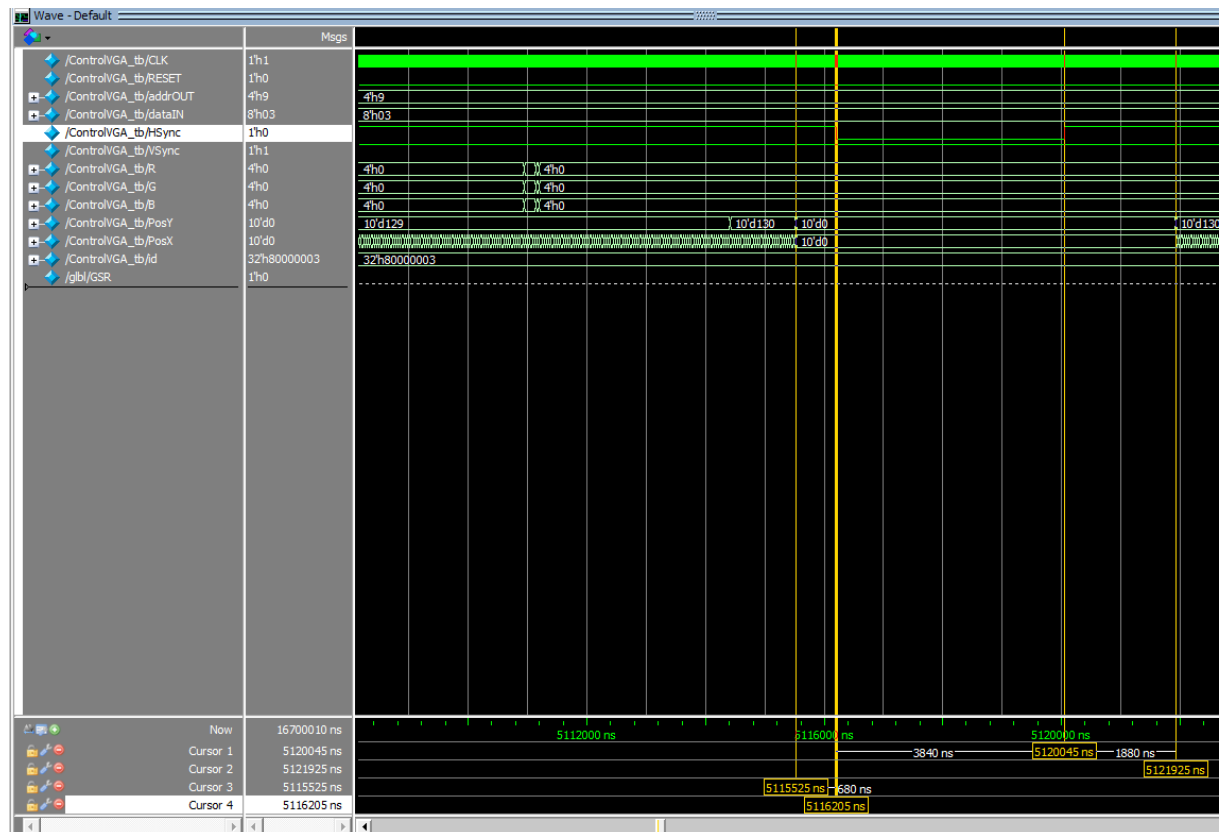


Figura 20: Tiempos en HSync - Porches

Cuadro 1: Consumo de recursos de la FPGA

| Recurso         | Consumo    |
|-----------------|------------|
| Potencia        | 89mW       |
| Slices ocupadas | 1 %        |
| Registros       | ¡1 %       |
| MUX             | 3 %        |
| IO              | 14 %       |
| LUTs lógicos    | 1 %        |
| LUT FF          | 1419       |
| DSP48E1         | 1 %        |
| Reloj Máximo    | 111.592MHz |



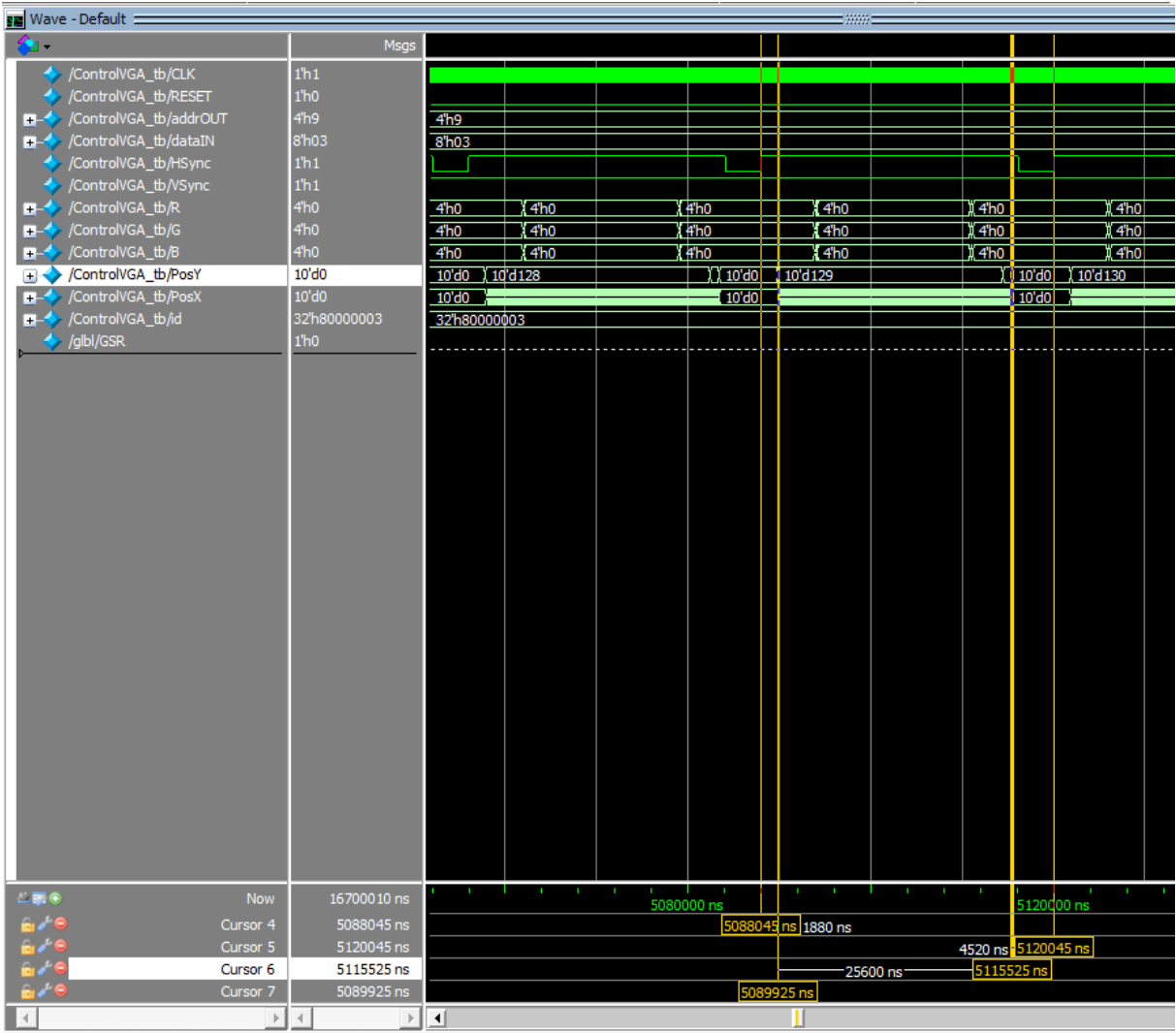


Figura 21: Tiempos en HSync en general

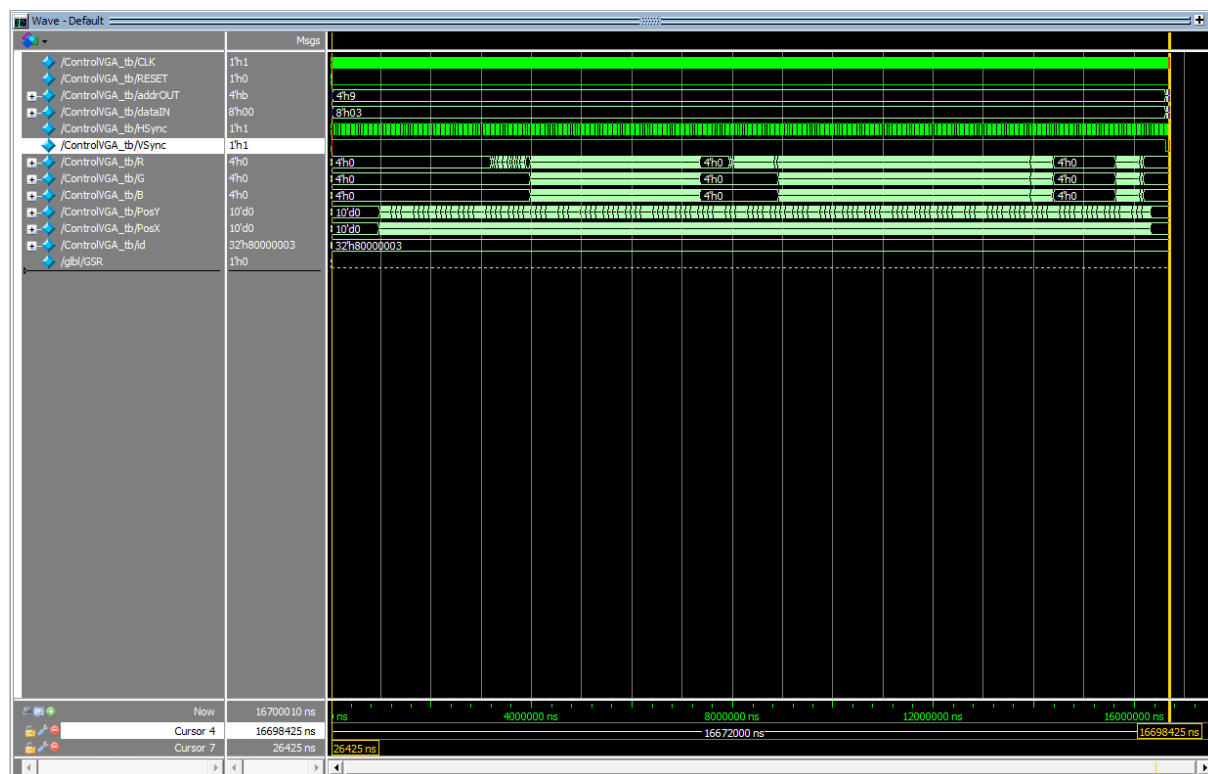


Figura 22: Tiempos en VSync

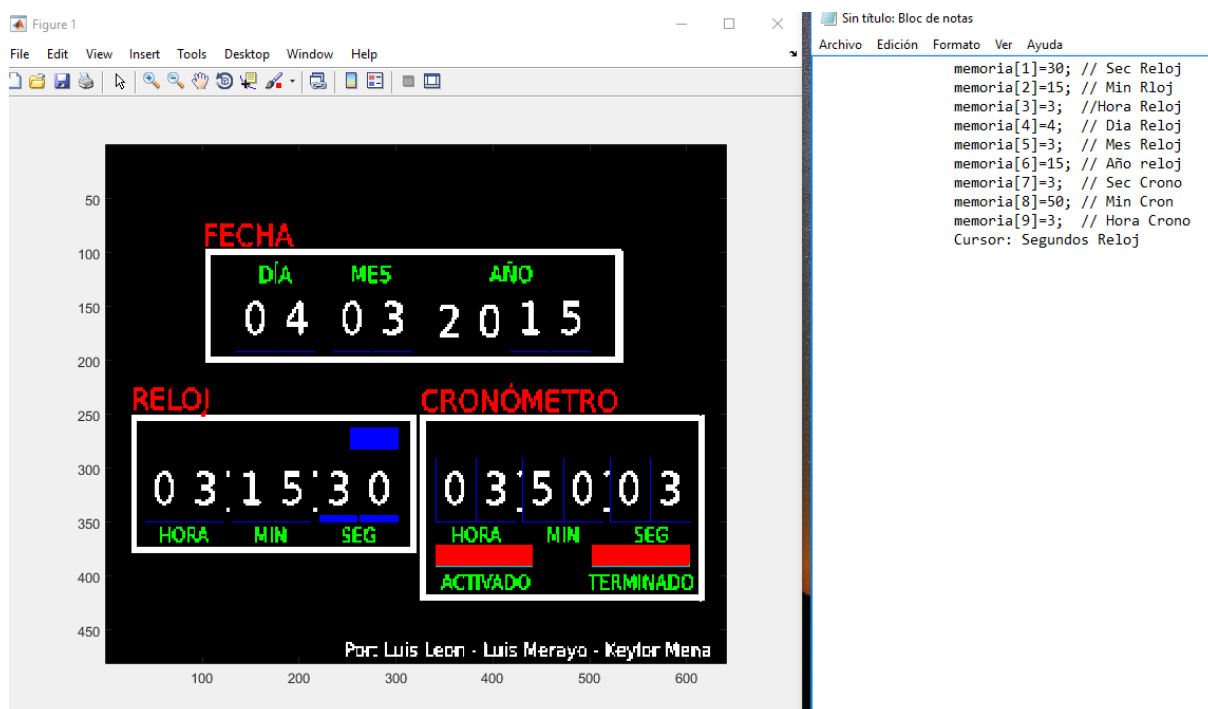


Figura 23: Tiempos en HSync en general

**Proyecto** 1  
**Trabajo** Desarrollo de un controlador  
 VGA  
**Grupo** 1  
**Revisado por:** Alfonso Chacón Rodríguez

**Página** 19/26  
**Actualizado en:** 27/08/2016  
**Revisado en:** 30/08/2016  
**Diseñadores** Keylor Mena Venegas  
 Luis Leon Vega  
 Luis Merayo Gatica

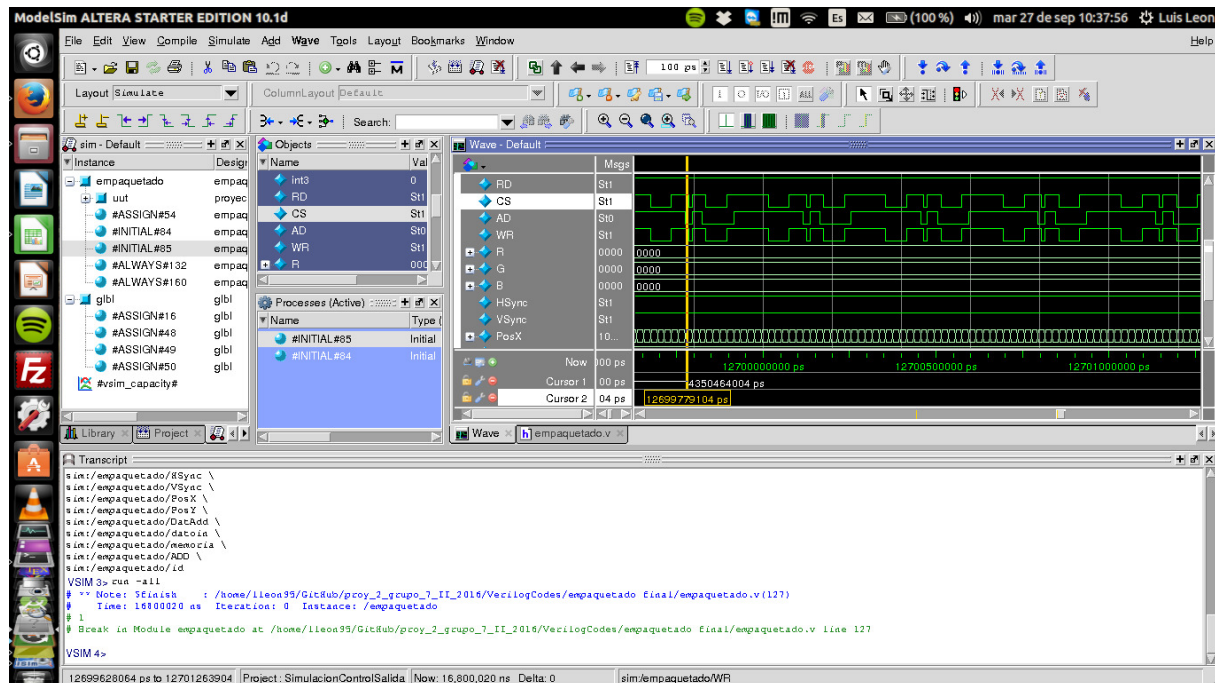


Figura 24: Captura de simulación post-síntesis

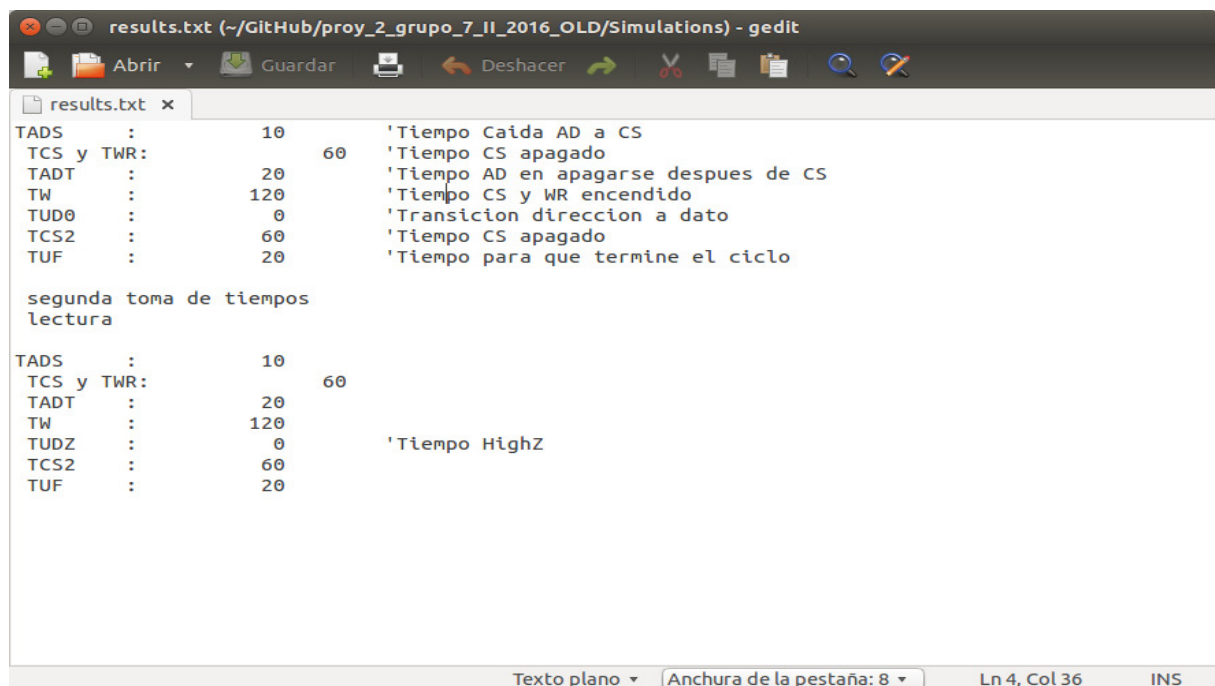


Figura 25: Tiempos de la simulación post-síntesis

Proyecto 1  
Trabajo Desarrollo de un controlador  
VGA  
Grupo 1  
Revisado por: Alfonso Chacón Rodríguez

Página 20/26  
Actualizado en: 27/08/2016  
Revisado en: 30/08/2016  
Diseñadores Keylor Mena Venegas  
Luis Leon Vega  
Luis Merayo Gatica

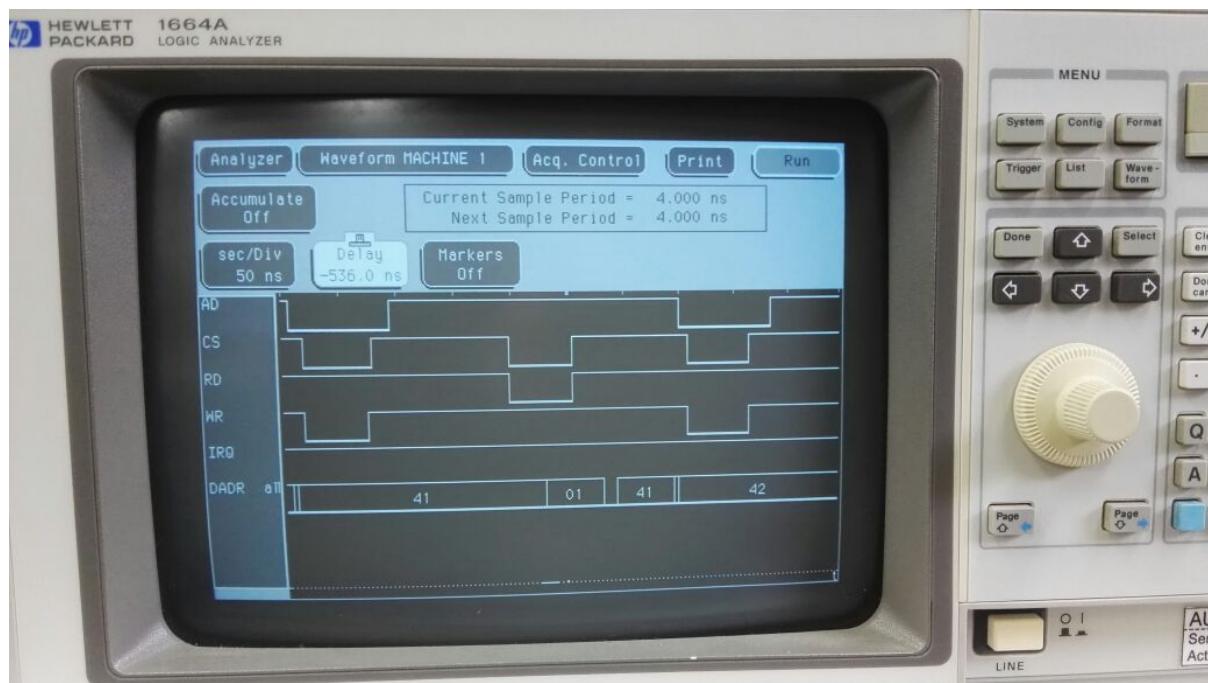


Figura 26: Captura 1 del analizador lógico

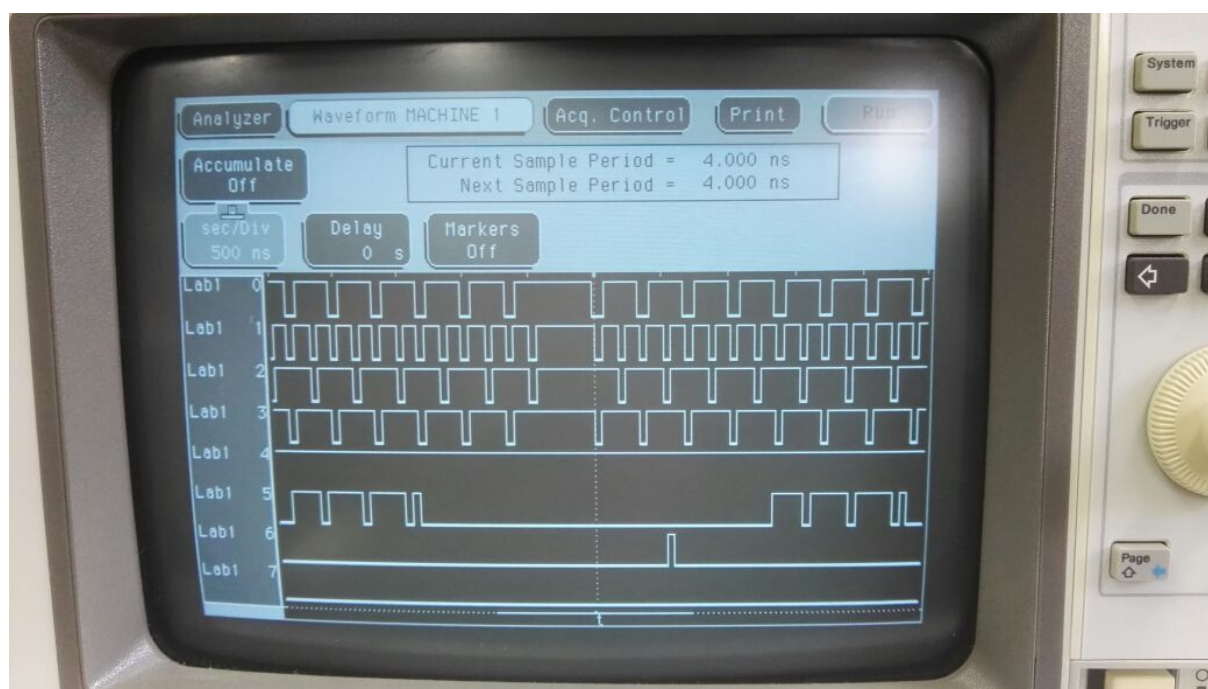


Figura 27: Captura 2 del analizador lógico



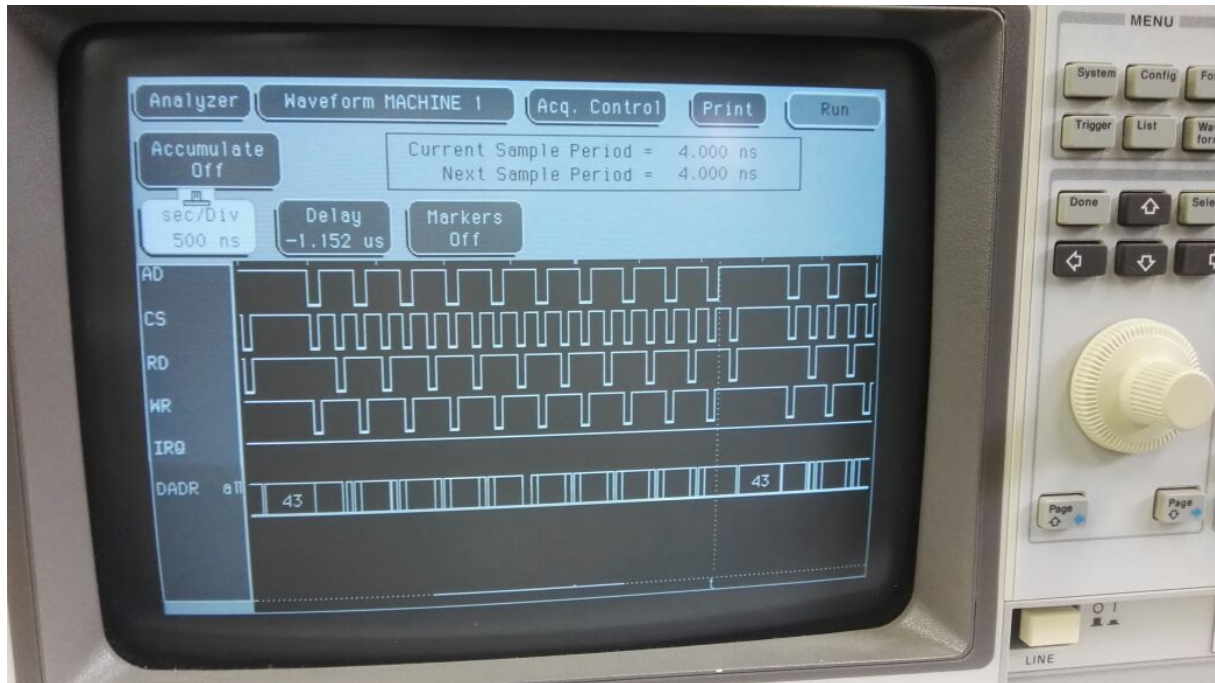


Figura 28: Captura 3 del analizador lógico

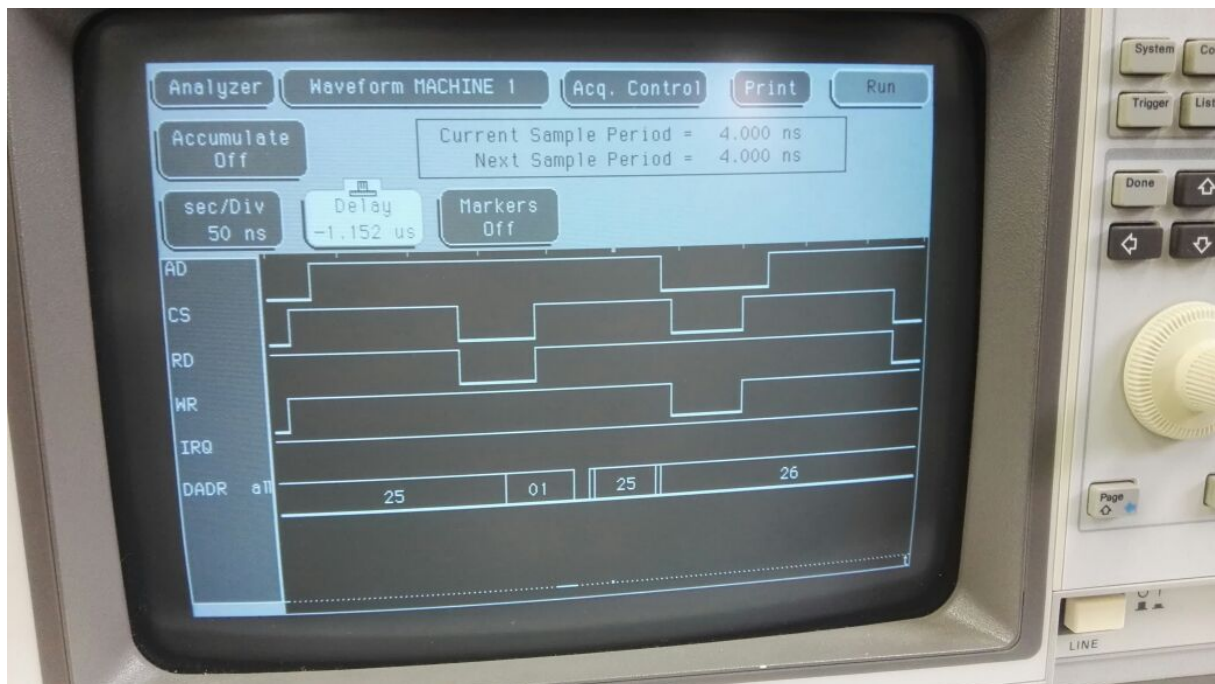


Figura 29: Captura 4 del analizador lógico

Proyecto 1  
Trabajo Desarrollo de un controlador VGA  
Grupo 1  
Revisado por: Alfonso Chacón Rodríguez

Página 22/26  
Actualizado en: 27/08/2016  
Revisado en: 30/08/2016  
Diseñadores Keylor Mena Venegas  
Luis Leon Vega  
Luis Merayo Gatica

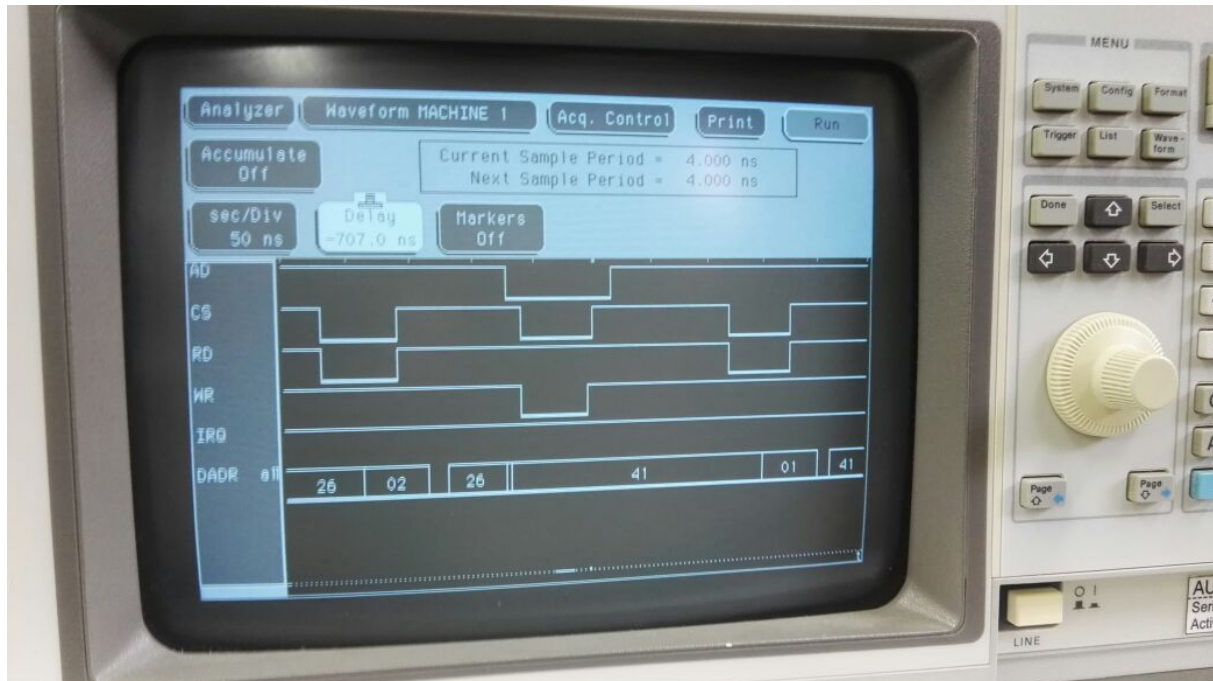


Figura 30: Captura 5 del analizador lógico

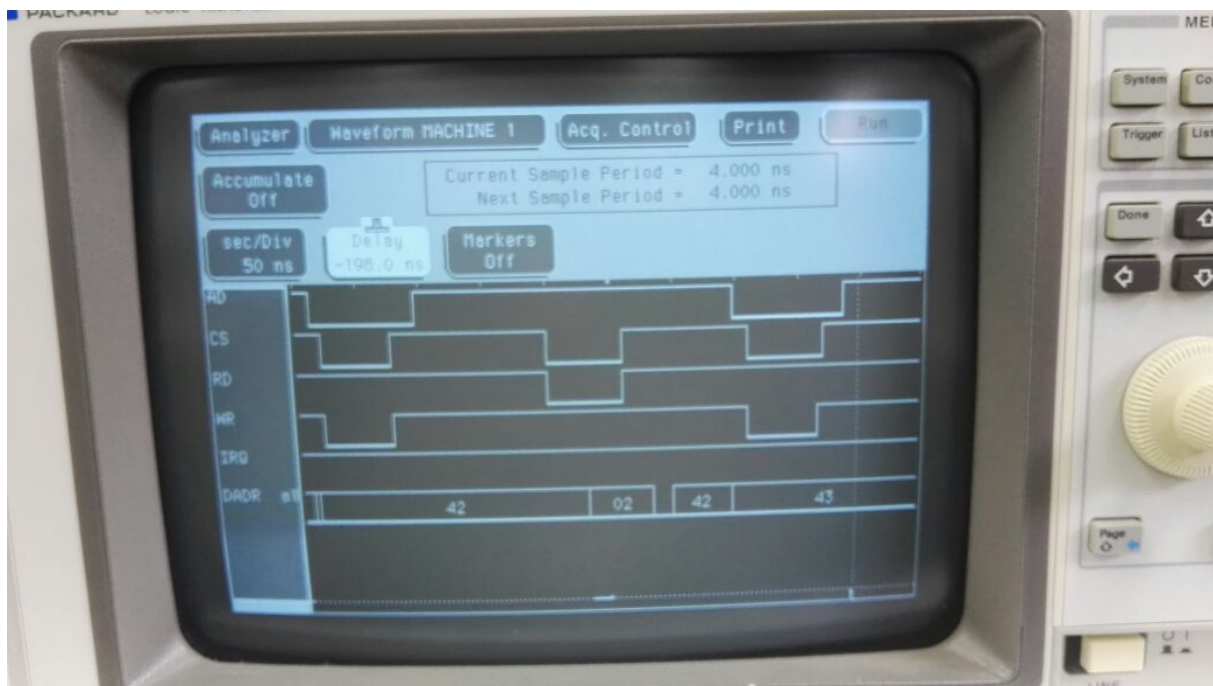


Figura 31: Captura 6 del analizador lógico

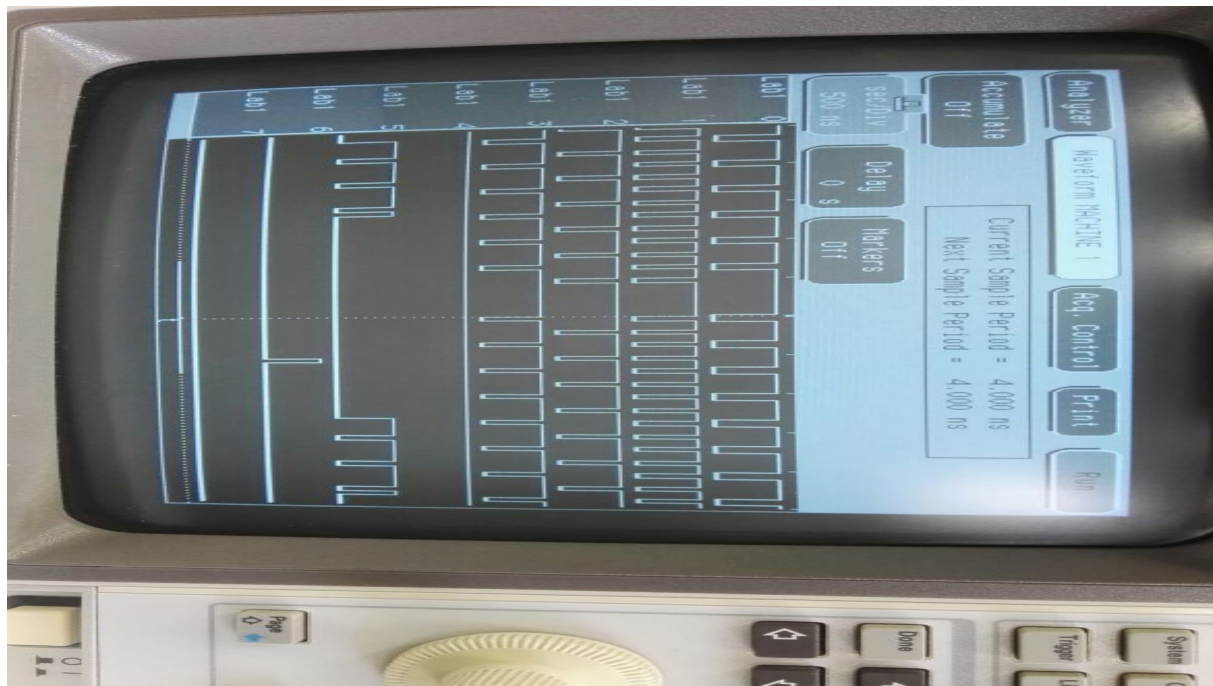


Figura 32: Captura 7 del analizador lógico

### 3.4. Mediciones en analizador lógico

## 4. Análisis de datos y resultados

### 4.1. Análisis de la RTC

La comunicación de la RTC con el circuito, como se mencionó anteriormente, se realizó por medio de maquinas de estado, y directamente con la de control de salida, cuya funcion es activar las señales, como CS, AD, RD, y WR las cuales permiten leer y escribir del dispositivo. Según las hoja de datos del V3023[1], estas cuatro señales deben respetar tiempos mínimos de acción para validar cualquier procedimiento, de no ser asi podria interpretarse un dato no válido.

De la simulación post-síntesis, figura 24 , se obtuvieron los tiempos que dura cada una de las señales en el proceso de lectura y escritura, ambos procesos, figura 25, tienen una duración de 300ns en total, y todas las señales respetan lo mínimo segun el datasheet, cabe destacar que la duración resultó ser un poco mas extensa en el momento de programarlo en la FPGA, sin embargo eso no causó problemas, ya que mientras sean mayores a los valores minimos es aceptable.

En la figura 24, además se verificó que la señal AD se activara y desativara un tiempo mínimo de 5ns y 10ns antes y despues, respectivamente, de las demás señales, con el fin de establecer si se va a escribir o leer una dirección o un dato.

El control de usuario, el cual contenía además el control de programacion de la RTC, al ser un módulo que combinaba muchos procesos, tenia muchos errores, lo cual no permitía la correcta escritura en la RTC, por lo cual se decidió separarlos.

|                      |                                  |                        |   |
|----------------------|----------------------------------|------------------------|---|
| <b>Proyecto</b>      | 1                                | <b>Página</b>          | 24/26   |
| <b>Trabajo</b>       | Desarrollo de un controlador VGA | <b>Actualizado en:</b> | 27/08/2016  |
| <b>Grupo</b>         | 1                                | <b>Revisado en:</b>    | 30/08/2016  |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez         | <b>Diseñadores</b>     | Keylor Mena Venegas<br>Luis Leon Vega<br>Luis Merayo Gatica |

A partir de lo anterior el control de la RTC, en general, se realizó de una manera estructurada, con ayudada el método de la factorización de máquinas de estado, lo cual permite una búsqueda de errores más rapida, y la modificacion de aspectos del diseño sin afectarlo en su totalidad.

## 4.2. Análisis del consumo de recursos

Como se aprecia en el Cuadro 1, el reloj máximo al cual puede correr el programa es de 111,592 MHz, lo cual es lo suficientemente eficiente y optimizado, para correr a la frecuencia que se utilizo de 100 MHz. Anterior a este rendimiento se encontro el problema de que la frecuencia maxima era de aproximadamente 89 MHz, la causa fue el diseño del controlador de VGA, el cual disponia de muchas operaciones lógicas, las cuales por lo general generan retrasos considerables en las señales y rutas de datos muy extensas en tiempo, por lo cual, para corregir el error, se rediseñaron ciertas secciones del código de VGA.

La potencia consumida por el circuito es de 89mW, por lo que si se considera la cantidad de Flip Flops utilizados, así como el método que se utilizó para crear la parte visual de la VGA,es decir cargar imagenes a la FPGA para dibujar los numeros, y la cantidad de maquina de estado, el consumo de es relativamente alto.

## 4.3. Análisis de la VGA

Para el controlador de VGA, los gráficos se desplegaron de acuerdo con las simulaciones anteriores. Asimismo, no existieron problemas de sincronía con excepción de inestabilidad en algunos registros, sin embargo, esto se debe a intercalamiento de datos en la RTC, donde se leen datos erróneos.

Con algunas observaciones, estos datos erróneos eran las direcciones de consulta de la RTC, por lo cual, es un problema con el módulo de control de la RTC.

Por otro lado, existieron problemas con el reloj máximo tolerado por la implementación, que dieron como resultados 89MHz aproximadamente en comparación de los 100MHz con los que se iba a alimentar el reloj del circuito. Sin embargo, se optimizó la lógica de los punteros de memoria y se logró incrementar a 111MHz

## 4.4. Análisis de las mediciones con analizador lógico

En las figuras 26,27,28,29,30,31,32 se puede notar los tiempos de las señales CS, AD, RD, y WR de la RTC, además del canal bidireccional de datos, en donde se aprecia el cambio entre dirección y dato. En este se puede bservar que existe una sobreposición de datos y direcciones, esto se debe al corto periodo de tiempo que tiene la RTC para responder a este cambio. Est causa inestabiidad en los datos, los cuales se visulaizan en la pantalla VGA.

# 5. Hoja de datos de unidades desarrolladas

## 5.1. Distribución de pines en la Nexys 4

NET R[0]"LOC = A3;  
NET R[1]"LOC = B4;  
NET R[2]"LOC = C5;



|                      |                                     |                        |   |
|----------------------|-------------------------------------|------------------------|---|
| <b>Proyecto</b>      | 1                                   | <b>Página</b>          | 25/26   |
| <b>Trabajo</b>       | Desarrollo de un controlador<br>VGA | <b>Actualizado en:</b> | 27/08/2016  |
| <b>Grupo</b>         | 1                                   | <b>Revisado en:</b>    | 30/08/2016  |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez            | <b>Diseñadores</b>     | Keylor Mena Venegas<br>Luis Leon Vega<br>Luis Merayo Gatica |

```

NET R[3]"LOC = A4;
NET "G[0]"LOC = C6;
NET "G[1]"LOC = A5;
NET "G[2]"LOC = B6;
NET "G[3]"LOC = A6;
NET "B[0]"LOC = B7;
NET "B[1]"LOC = C7;
NET "B[2]"LOC = D7;
NET "B[3]"LOC = D8;
NET clk"LOC = E3;
NET "HSync"LOC = B11;
NET "VSync"LOC = B12;
NET "DatAdd[0]"LOC = B13;
NET "DatAdd[1]"LOC = F14;
NET "DatAdd[2]"LOC = D17;
NET "DatAdd[3]"LOC = E17;
NET "DatAdd[4]"LOC = G13;
NET "DatAdd[5]"LOC = C17;
NET "DatAdd[6]"LOC = D18;
NET "DatAdd[7]"LOC = E18;
NET .AD"LOC = G14;
NET ÇS"LOC = P15;
NET RD"LOC = V11;
NET "WR"LOC = V15;
NET ïnt1"LOC = P4;
NET ïnt2"LOC = P3;
NET ïnt3"LOC = R3;
NET "Down"LOC = V10;
NET "Left"LOC = T16;
NET reset"LOC = E16;
NET Rig"LOC = R10;
NET Üp"LOC = F15;

```

Todos los pines tienen un IOSTANDARD de LVCMOS33.

## 6. Conclusiones y recomendaciones

### 6.1. Conclusiones

- En circuitos con una gran cantidad de procesos de control, la factorización de maquinas de estado permite una estructuración eficiente.
- El conocimiento de los tiempos en el diseño dentro de una FPGA, permite dar una idea de la eficiencia de un diseño.
- Respetar los tiempos mínimos de escritura y lectura en una RTC asegura al diseñador que se realizan ambos procesos de manera correcta, sin generar metaestabilidad.

|                      |                                     |                        |   |
|----------------------|-------------------------------------|------------------------|---|
| <b>Proyecto</b>      | 1                                   | <b>Página</b>          | 26/26   |
| <b>Trabajo</b>       | Desarrollo de un controlador<br>VGA | <b>Actualizado en:</b> | 27/08/2016  |
| <b>Grupo</b>         | 1                                   | <b>Revisado en:</b>    | 30/08/2016  |
| <b>Revisado por:</b> | Alfonso Chacón Rodríguez            | <b>Diseñadores</b>     | Keylor Mena Venegas<br>Luis Leon Vega<br>Luis Merayo Gatica |

- Es posible mejorar la interfaz del usuario para hacerla más amigable precargando un archivo de imagen convertida en binario en una ROM.
- El coste que tiene incorporar imágenes al diseño compromete recursos de la FPGA por lo que se debe optimizar y/o usar una GPU.
- Es posible visualizar mediante un testbench y un software para procesar imágenes en resultado que se podrá ver en la pantalla.

## 6.2. Recomendaciones

- Usar una lógica optimizada para poder reducir el consumo de recursos en punteros de memoria.
- Usar una GPU externa para procesar las imágenes y no comprometer recursos de la FPGA.
- Segmentar el control de la RTC por medio de la factorización de máquinas de estado.
- Seguir paso por paso los procesos de inicialización de la RTC según se muestra en el datasheet.
- Para mejorar la visualiación de los datos en la pantalla VGA se debe aumentar el tiempo de la transición entre direcciones y datos.

## 7. Bibliografía

- [1] EM Microelectronic, *Very Low Power 8-Bit 32 kHz RTC Module with Digital Trimming, User RAM and High Level Integration*, Swatch Group Electronic systems, 2015.
- [2] Digilent, *Nexys 4 FPGA Board Reference Manual*, Digilent, 2013.