

| | | | |
|----------------------|-------------------------------------|------------------------|---|
| Proyecto | 1 | Página | 1/13 |
| 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 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

| | | | |
|----------------------|----------------------------------|------------------------|---|
| Proyecto | 1 | Página | 2/13 |
| 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 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.

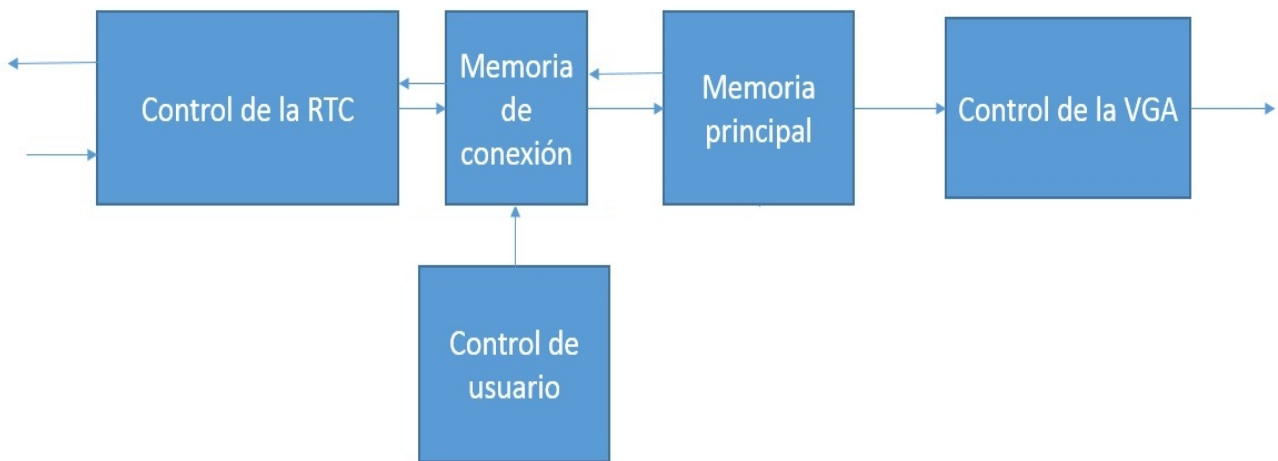


Figura 1: Diagrama de módulos principales del sistema.

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.

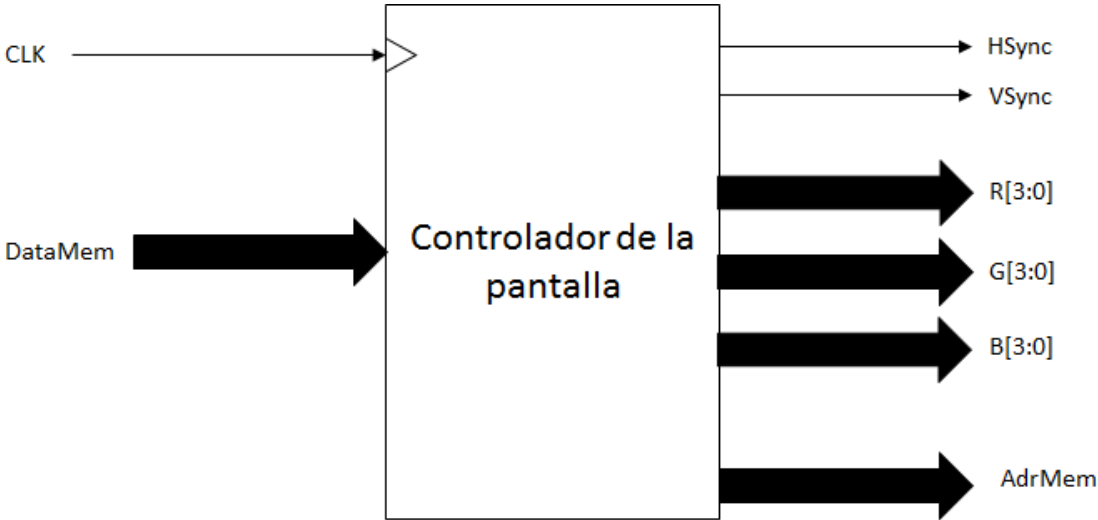


Figura 2: Diagrama de primer nivel VGA

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.

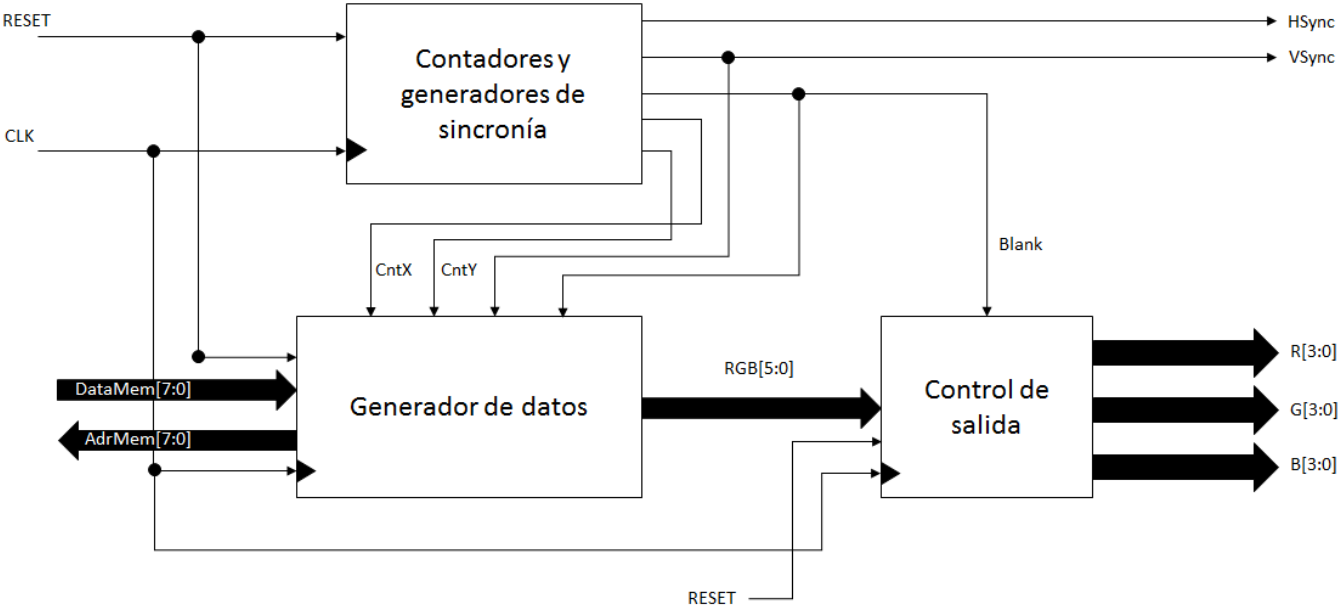


Figura 3: Diagrama de primer nivel VGA

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

| | | | |
|----------------------|----------------------------------|------------------------|---|
| Proyecto | 1 | Página | 4/13 |
| 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 Luis Leon Vega 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

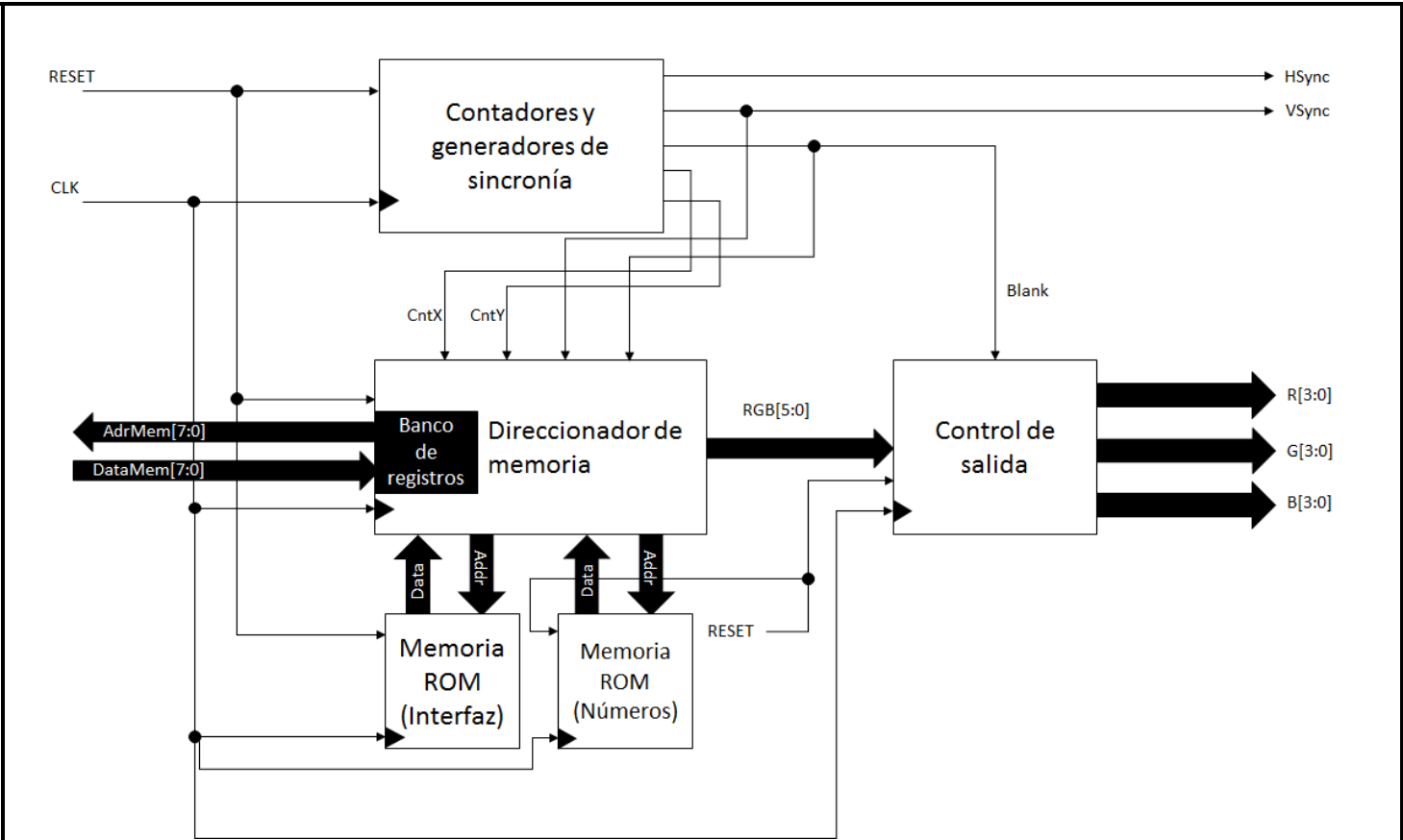


Figura 4: Diagrama de primer nivel VGA

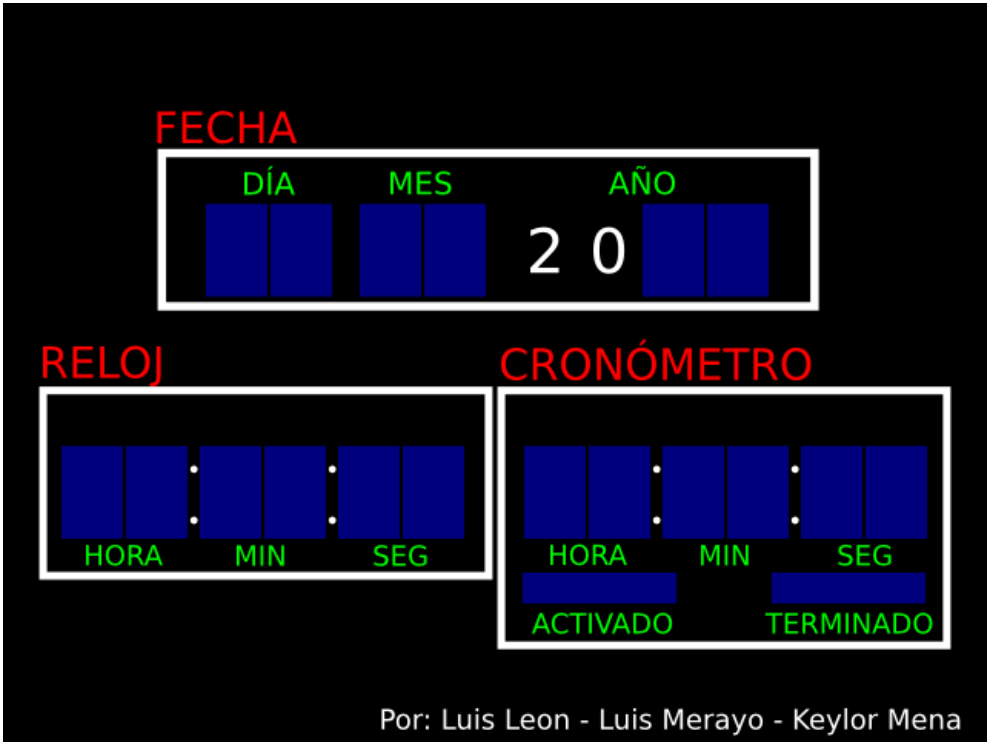


Figura 5: Diagrama de primer nivel VGA

| | | | |
|----------------------|----------------------------------|------------------------|---|
| Proyecto | 1 | Página | 6/13 |
| 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 Luis Leon Vega 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 7 botones, 3 interruptores que indican que se quiere cambiar, el reloj, el timer o la alarma, y para moverse entre los registros de

| | | | |
|---------------|------------------------------|-----------------|---------------------|
| Proyecto | 1 | Página | 7/13 |
| Trabajo | Desarrollo de un controlador | 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 |

datos y aumentar o disminuir sus valores.

2.2.1. 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 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 en 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 conexión, 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.

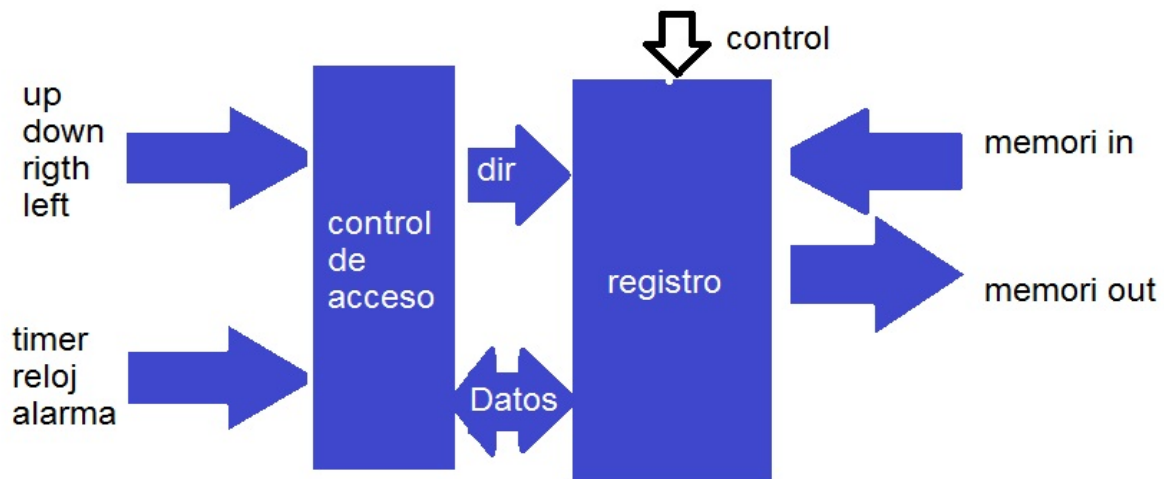


Figura 8: Diagrama de segundo nivel del Teclado.

| | | | |
|----------------------|-------------------------------------|------------------------|---|
| Proyecto | 1 | Página | 8/13 |
| 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 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.

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.

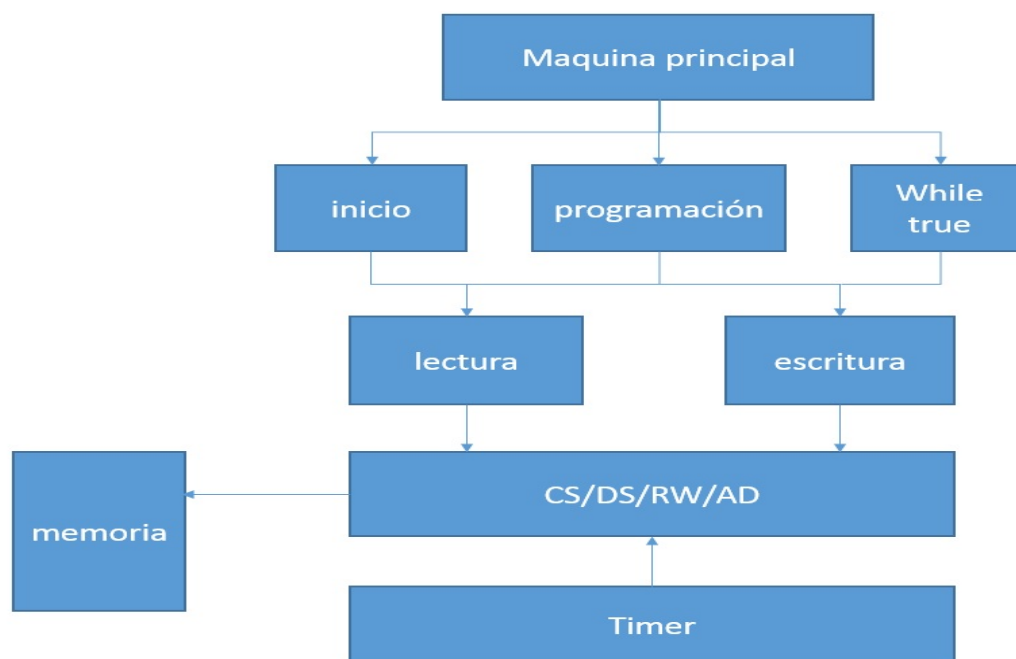


Figura 9: jerarquía de la RTC.

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.

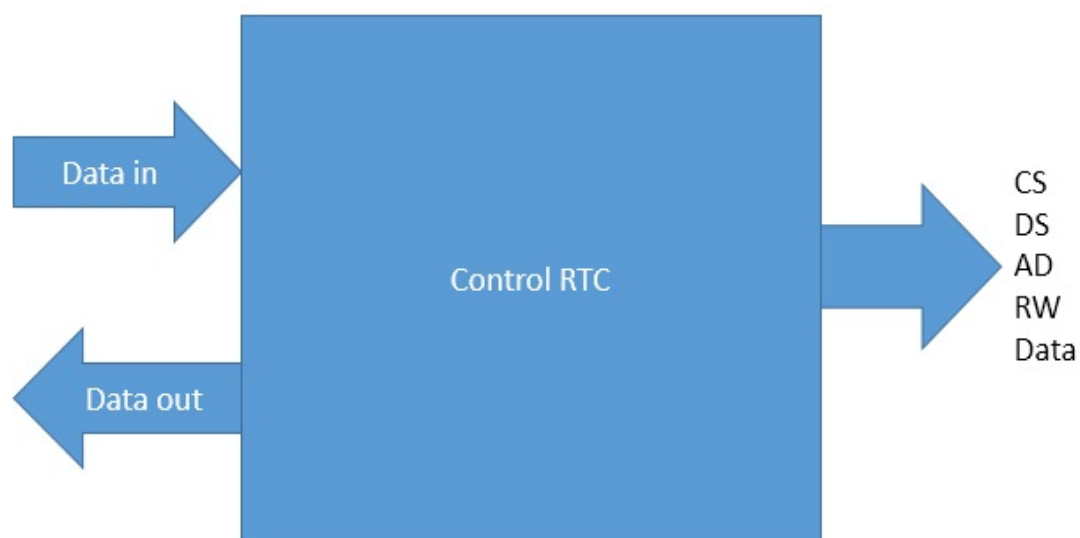


Figura 13: nivel 1 del RTC

| | | | |
|----------------------|----------------------------------|------------------------|---------------------|
| Proyecto | 1 | Página | 11/13 |
| 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 |

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.

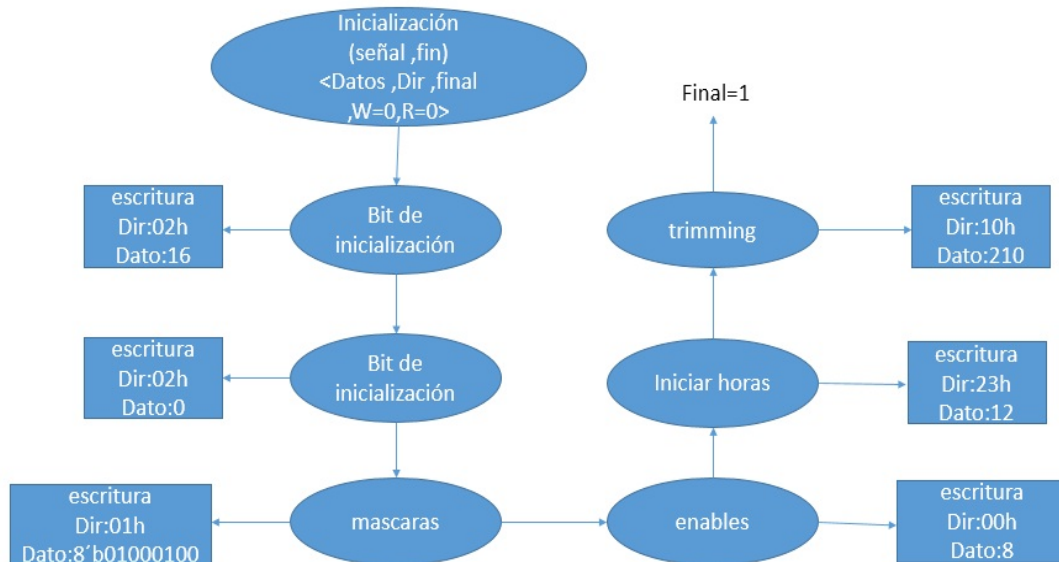


Figura 14: flujo de datos de la maquina de estados de inicializacion

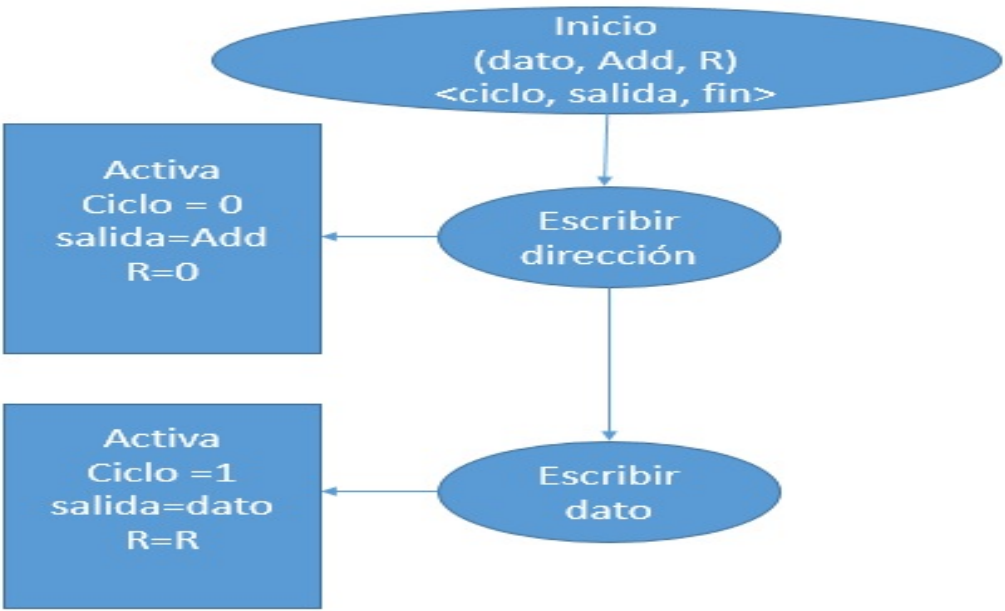


Figura 15: flujo de datos de la maquina de lectura

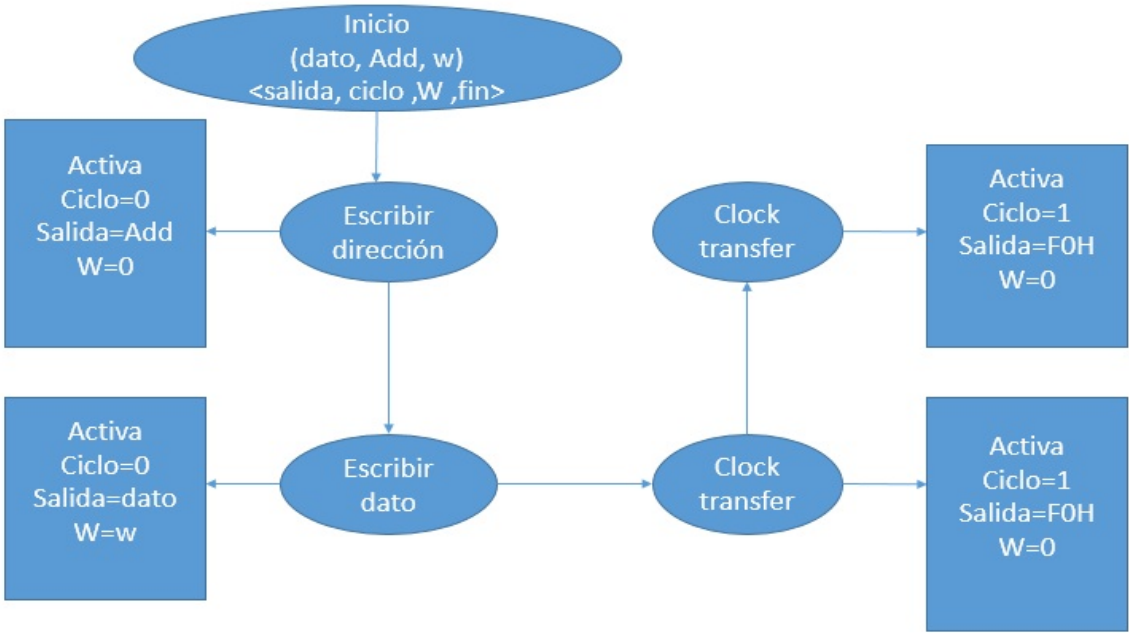


Figura 16: flujo de datos de la maquina de Escritura

| | | | |
|---------------|------------------------------|-----------------|---------------------|
| Proyecto | 1 | Página | 13/13 |
| Trabajo | Desarrollo de un controlador | 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 |

2.4. ROM de Instrucciones

3. Datos y resultados

3.1. Simulaciones

4. Análisis de datos y resultados

5. Hoja de datos de unidades desarrolladas

6. Conclusiones y recomendaciones

6.1. Conclusiones

6.2. Recomendaciones