

**Escuela de IngenieríÂa en Electronica**

**Laboratorio de Diseño de Sistemas Digitales**

**Bitácora**

**Proyecto:**

Control y programación RTC con Nexys3

**Profesor:**

Alfonso Chacón RodríÂuez

**Estudiantes:**

Keylor Mena Venegas

Luis Leon Vega

Luis Merayo Gatica

**Periodo**

II Semestre, 2016

## ***Descripción del problema***

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 Nexys3.

## ***Introducción al proyecto***

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.

## ***Objetivo General***

Diseñar un controlador de RTC que permita leerlo y programarlo mediante una interfaz de usuario consistente en botones incorporados dentro de la FPGA (Nexys3) y un monitor comunicado a través del protocolo VGA.

## ***Objetivos Específicos***

- 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.

## *Control de eventos*

**Fecha: 30 de Agosto**

**Integrantes:** Luis Leon, Luis Merayo, Keylor Mena

**Hora:** 13:00 - 15:00

**Actividad:**

El profesor explicó el instructivo y expresó las necesidades que deben ser cubiertas durante todo el proyecto. Quedó claro en que se debe desarrollar unidades de control para el proyecto (que se pueden hacer con FSM o Máquinas de Estados Finita) y desarrollar un controlador para obtener datos del RTC y para programarlo.

El reto de este último surge, principalmente, de comunicar el circuito al RTC mediante un bus multiplexado de direcciones y datos (A/D Bus). Se debe crear un control que permita escribir y leer de forma controlada y, con ello, que se pueda seleccionar el hardware para cada caso.

En este día, se hizo una reunión grupal para decidir las tareas de cada miembro, lo cual estableció:

- Luis Leon: Control general del circuito - Controlador VGA
- Luis Merayo: Control del RTC
- Keylor Mena: Encargado de testbench.

No se estableció ninguna aproximación al diseño del proyecto para dar tiempo de que cada miembro pueda estudiar el instructivo y el RTC con calma y de forma individual. Asimismo, se quedó que el día de mañana, esta tarea será efectuada al igual que la distribución de todas las tareas que puede implicar el proyecto.

Sin embargo, se ha decidido usar el control VGA que se ha diseñado en el proyecto anterior para agilizar la tarea del nuevo diseño de proyecto, asumiendo de que se conocen todas las partes de este controlador. Por otro lado, para efectos de la comprensión del nuevo proyecto, se asistió a la tutoría el día anterior, brindando la facilidad de que uno de los miembros conozca el sistema de antemano. Para ello será recomendado ver el enlace de interés al final de esta entrada.

**Enlaces de interés:**

- Documento de la tutoría: <http://bit.ly/2bEI3RZ>
- Datasheet del RTC: <http://bit.ly/2cpNGJa>

**Fecha: 31 de Agosto**

**Integrantes:** Luis Leon

**Hora:** 7:30 - 8:00

**Actividad:**

Se incorporaron los apartados de descripción del problema, introducción al proyecto y los objetivos dentro del documento de control de eventos (bitácora). Esto para seguir el formato de la plantilla proporcionada por el profesor.

**Fecha: 1 de Septiembre**

**Integrantes:** Keylor Mena, Luis Leon, Luis Merayo

**Hora:** 8:00-9:00

**Actividad:**

Se realizó una reunión de todos los miembros del grupo, donde se plasmaron las ideas del control del RTC y se establecieron las comunicaciones entre los módulos principales, donde se destacan

1. Control VGA
2. Control de Usuario
3. Control de Memoria
4. Control de RTC

Como parte del protocolo, se estableció una comunicación entre el control principal (de Memoria) con el control RTC, ilustrado en la siguiente imagen de la Figura 1

Por otro lado, se hizo un acercamiento al diagrama de flujo del módulo de RTC, mostrado en la Figura 2

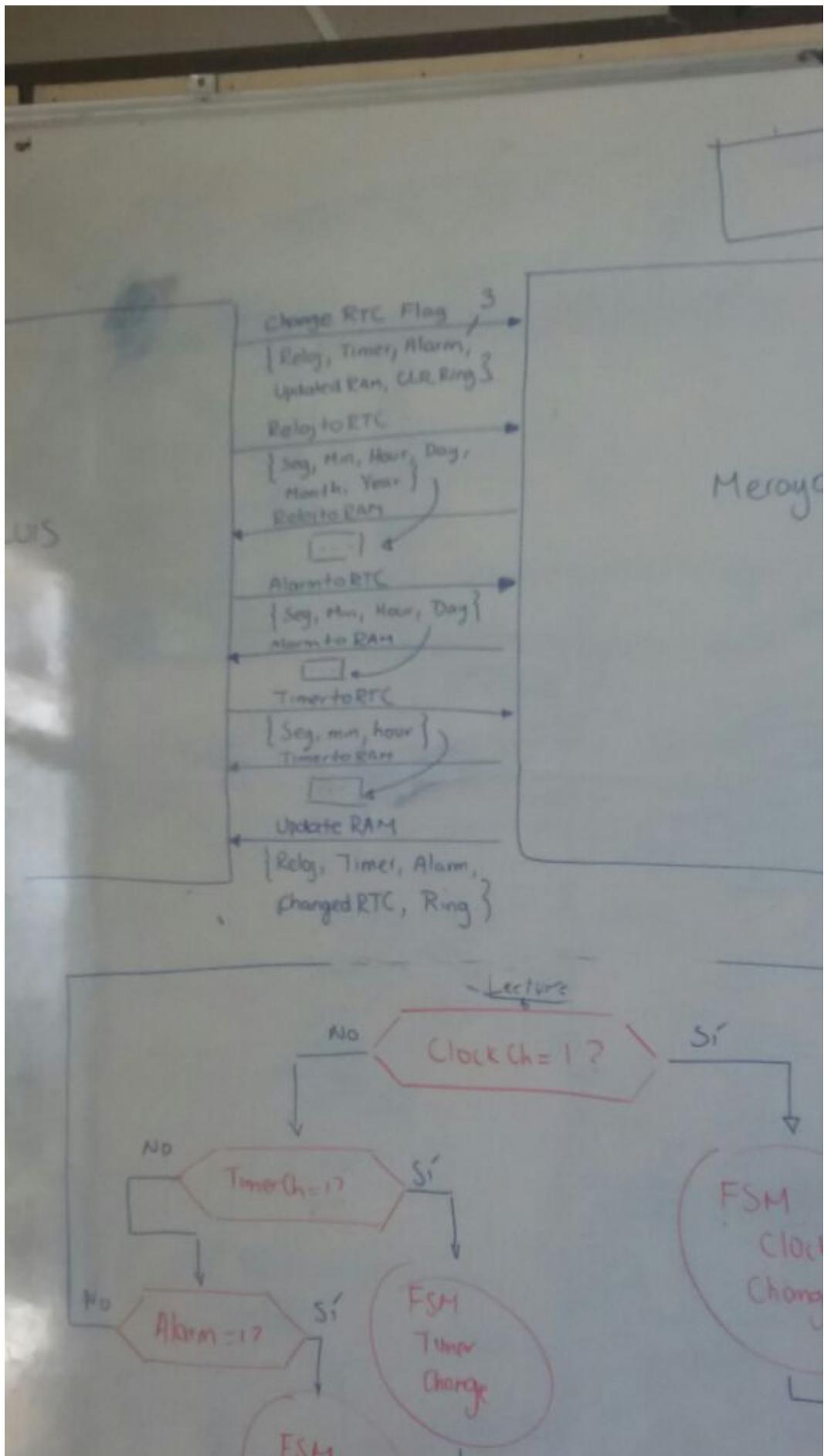


Figura 1: Diagrama de protocolo

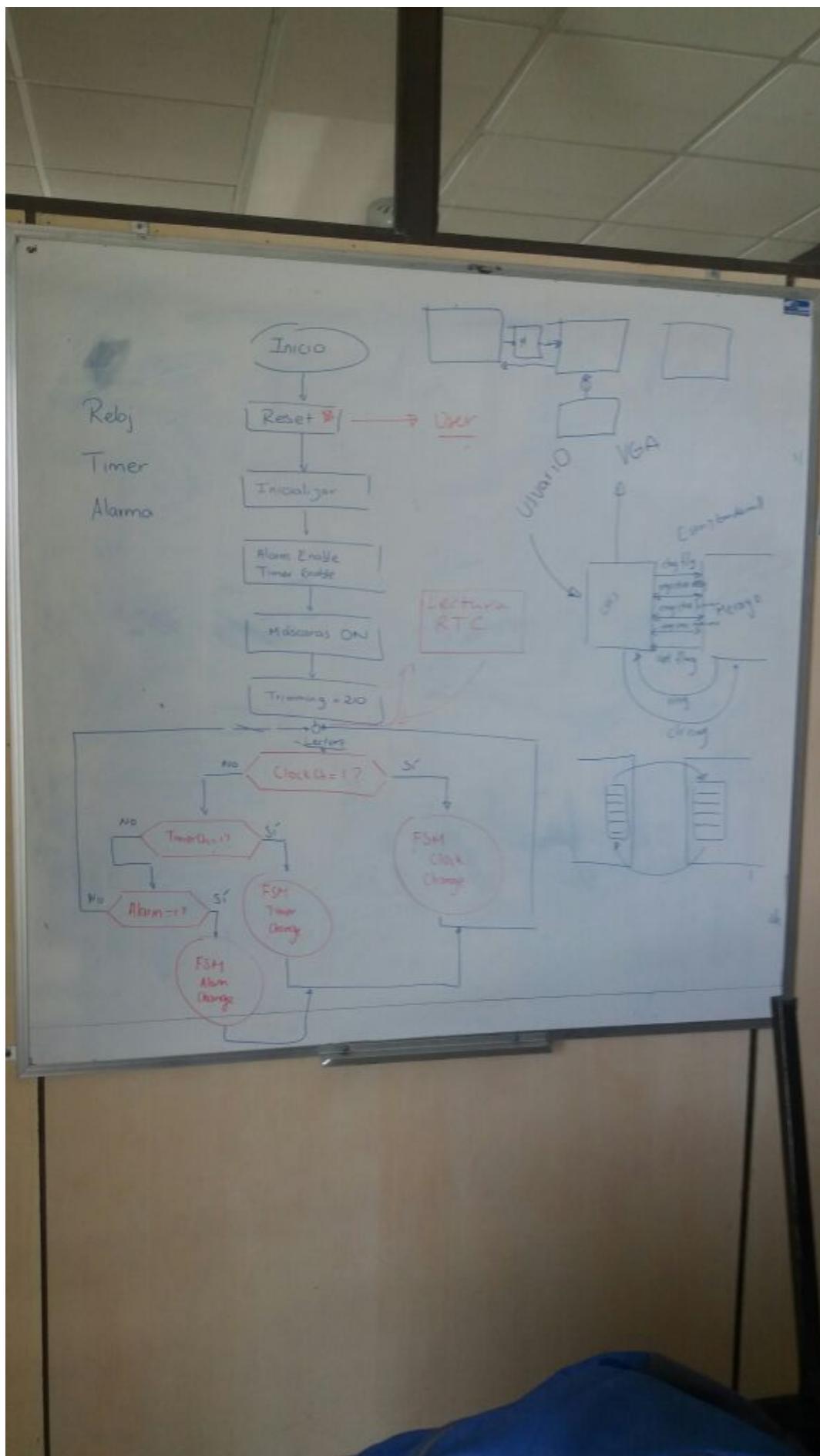


Figura 2: Diagrama de flujo para el control del RTC

**Fecha:** 2 de Septiembre

**Integrantes:** Keylor Mena

**Hora:** —

**Actividad:**

ACTUALIZACION REQUERIDA

**Fecha: 3 de Septiembre**

**Integrantes:** Luis Leon

**Hora:** 13:00 - 16:00

**Actividad:**

Se hizo el diseño del sistema, que incorpora la etapa de control de acceso a la memoria por parte del control de usuario y el RTC. Además, se incorporó el módulo VGA como parte del control de memoria. Esto se puede ver en la figura 3

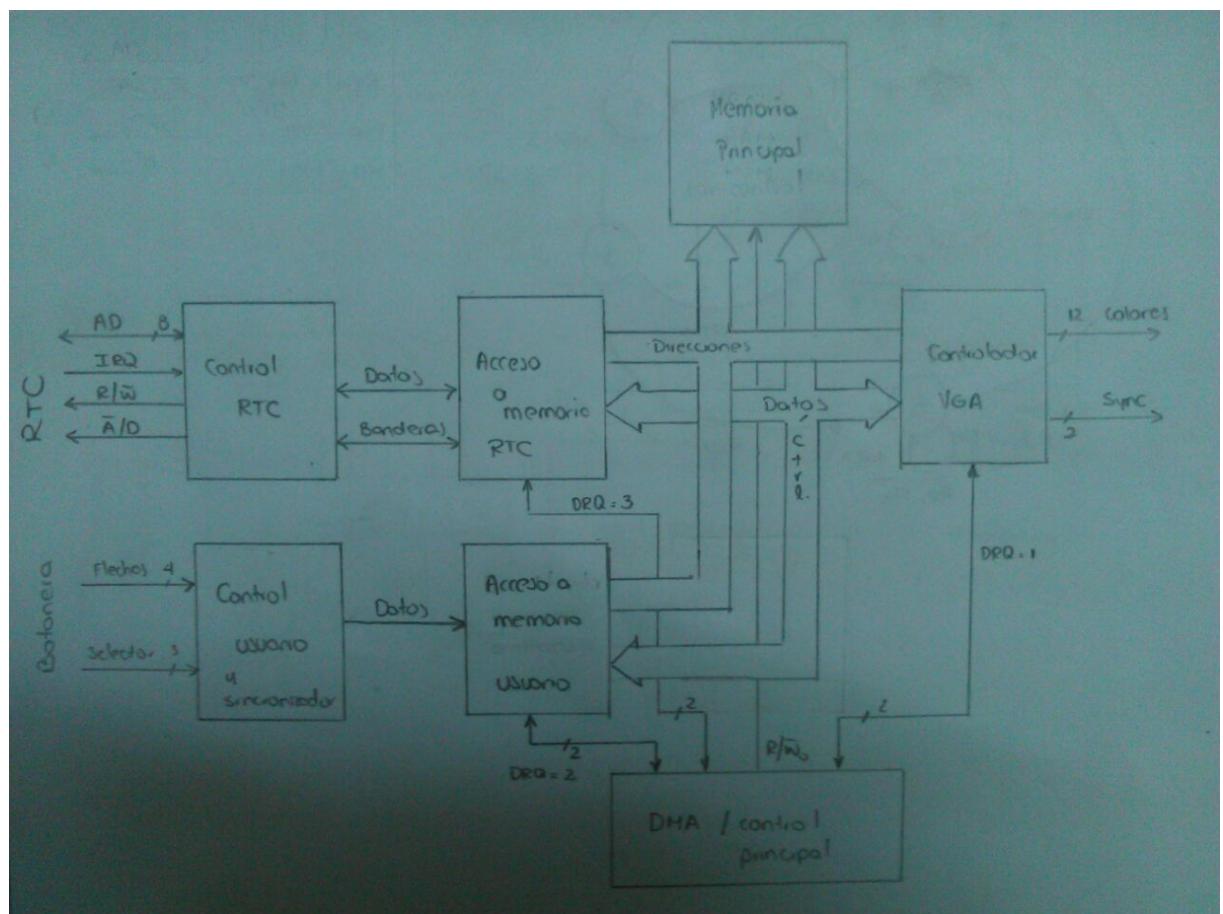


Figura 3: Diagrama general del sistema

En cuanto al acceso a la memoria, esta se regirá por una FSM (máquina de estados), cuyos diseños se presentan en las figuras 4 y 5.

**Enlaces de interés:** <http://bit.ly/2c3durR>

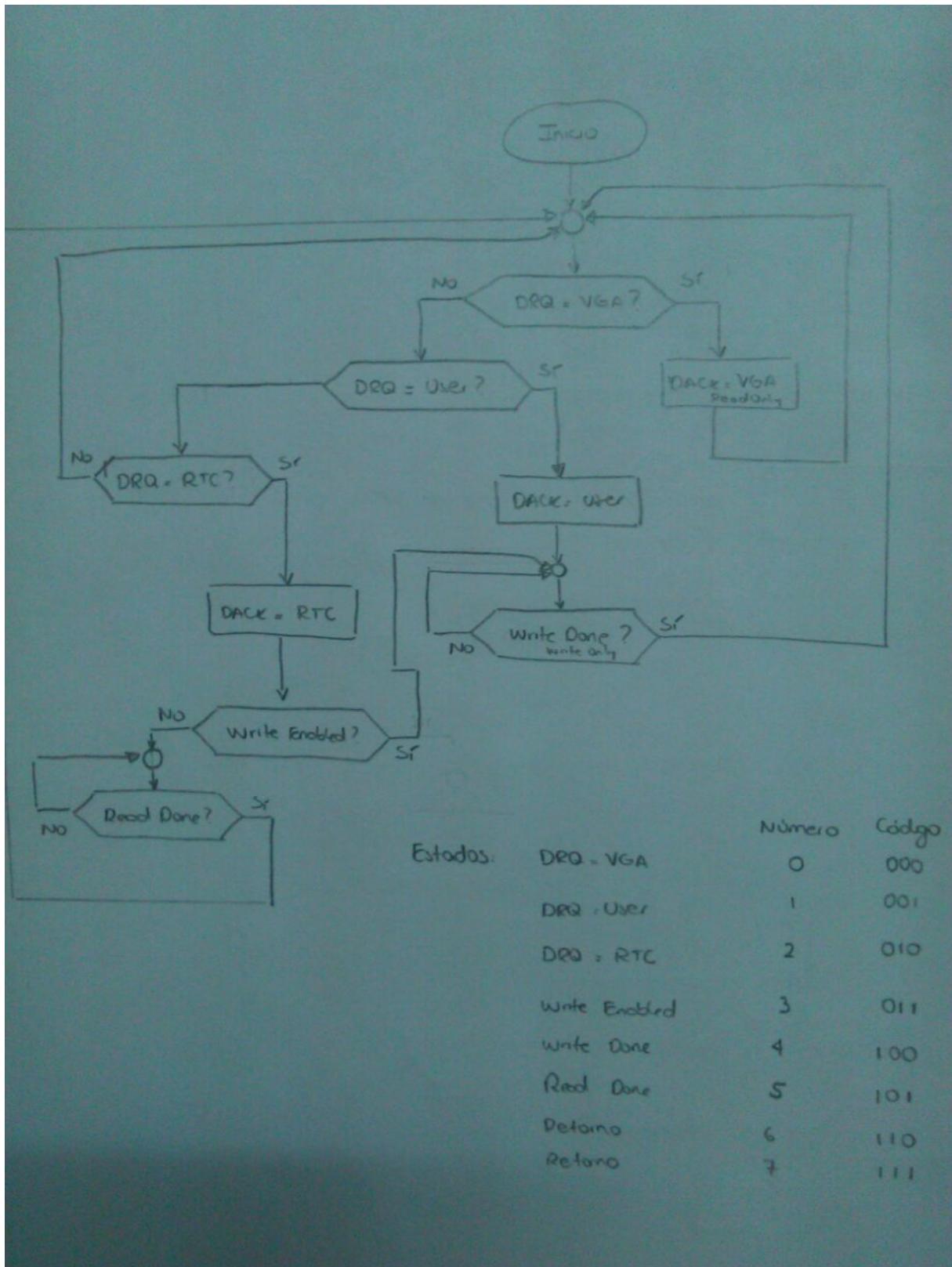


Figura 4: Diagrama de flujo de la máquina de estados principal

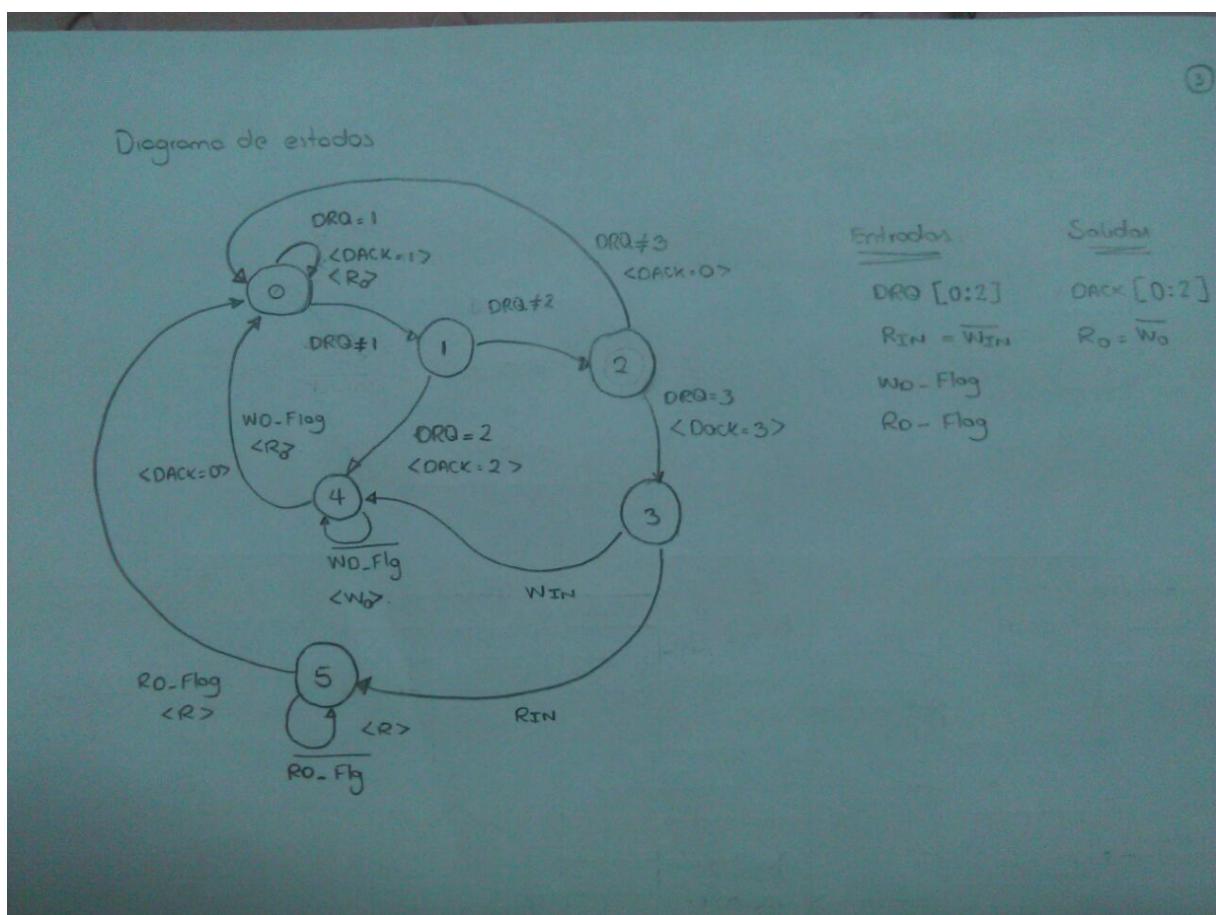


Figura 5: Diagrama de estados de la máquina de estados principal

**Fecha:** 4 de Septiembre

**Integrantes:** keylor Mena

**Hora:** 13:00 - 16:00

**Actividad:**

Se programo el nuevo registro dual con actualizacion controlada, este permite aislar etapas principales teniendo registros individuales entre ellas y actualizando estos con señales o banderas, se llamo memoria DRULC(Dual Register Update Log Controller), una manera de apreciar se representa en la figura6



Figura 6: Diagrama del control RTC

**Fecha: 5 de Septiembre****Integrantes:** Luis Leon**Hora:** 14:30 - 17:00**Actividad:**

Se hicieron los diagramas de bloques del control VGA y del control de usuario. Estos diagramas contienen unidades de control y otras subunidades que deben ser descritas, por ello, se hace un pequeño índice de los elaborado el día de hoy.

## ■ Control VGA:

Diagrama de bloques (Figura 7)

Diagrama de flujo (Figura 8)

Diagrama de bloques direccionamiento (Figura 9)

Diagrama de flujo direccionamiento (Figura 10)

## ■ Control de Usuario:

Diagrama de bloques (Figura 11)

Diagrama de flujo (Figura 12)

Finalmente, por algunos problemas de salud, se ha decidido colaborar en la parte de implementación del código en Verilog y al diseño del control VGA. Se expondrá a los compañeros la situación y, por otro lado, se aclararán algunos aspectos de memoria que deben ser discutidos para que haya armonía en la implementación.

**Fecha: 3 de Septiembre****Integrantes:** Luis Merayo**Hora:** 21:00 - 22:00**Actividad:**

Actividad: Se realizó un primer diseño del control del RTC, por lo que se resaltó la ruta de datos e inicialmente se crearon tres máquinas de estado, una de escritura, otra de lectura, y la principal, la cual controla los procedimientos iniciales luego de un reset. También para controlar los tiempos se colocó un contador. Esto se puede ver en la figura 13

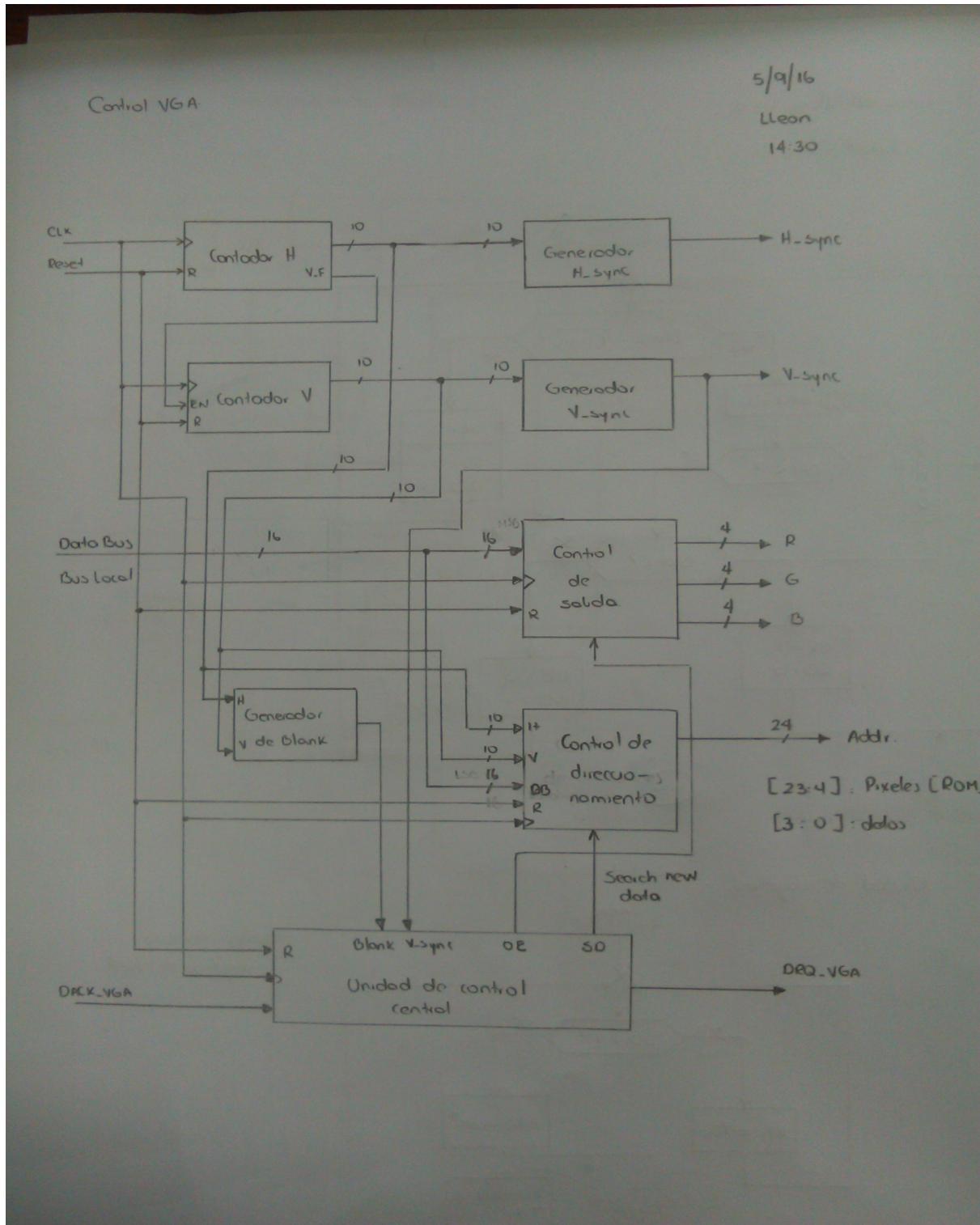


Figura 7: Diagrama de bloques del Control VGA

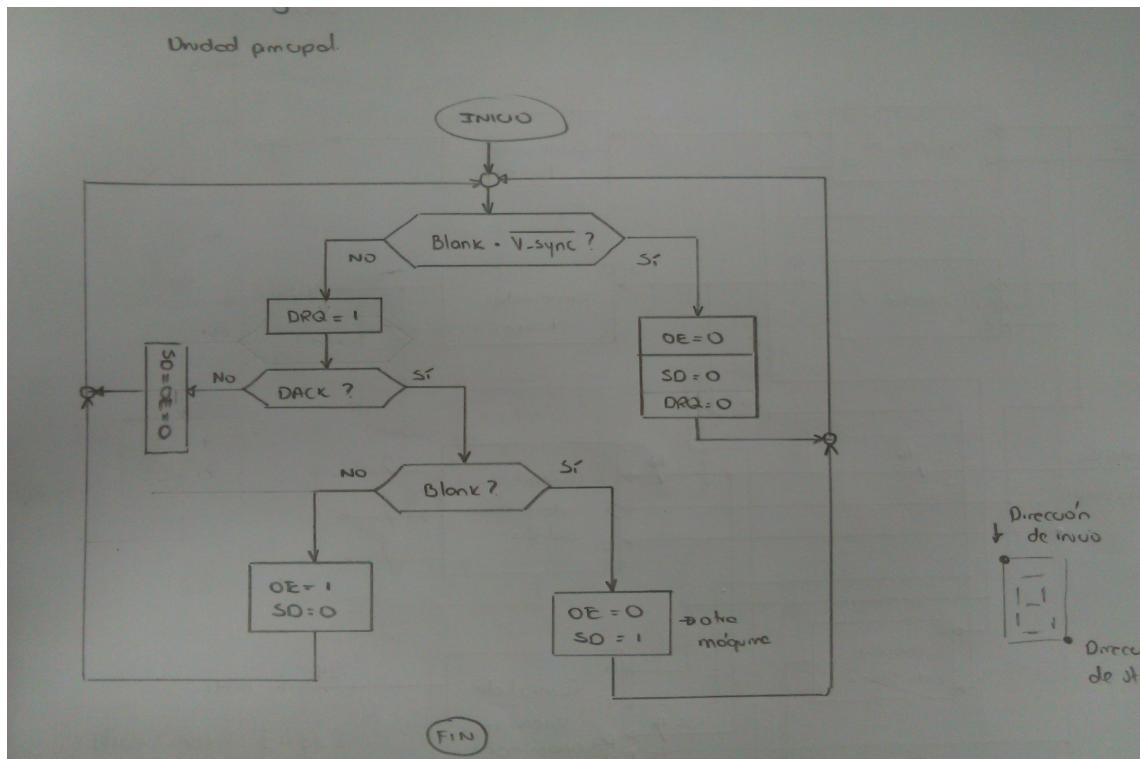


Figura 8: Diagrama de Flujo del Control VGA

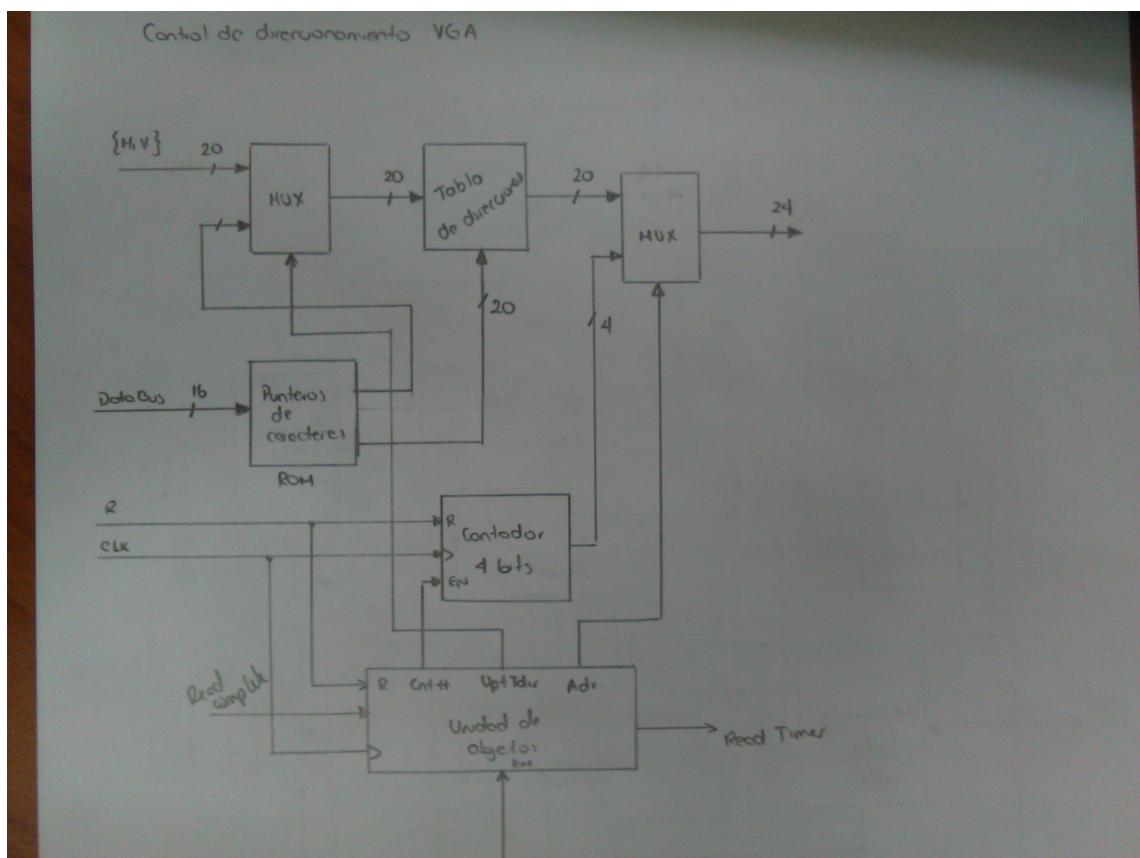


Figura 9: Diagrama de Bloques del Control VGA - Sección de Direccionamiento

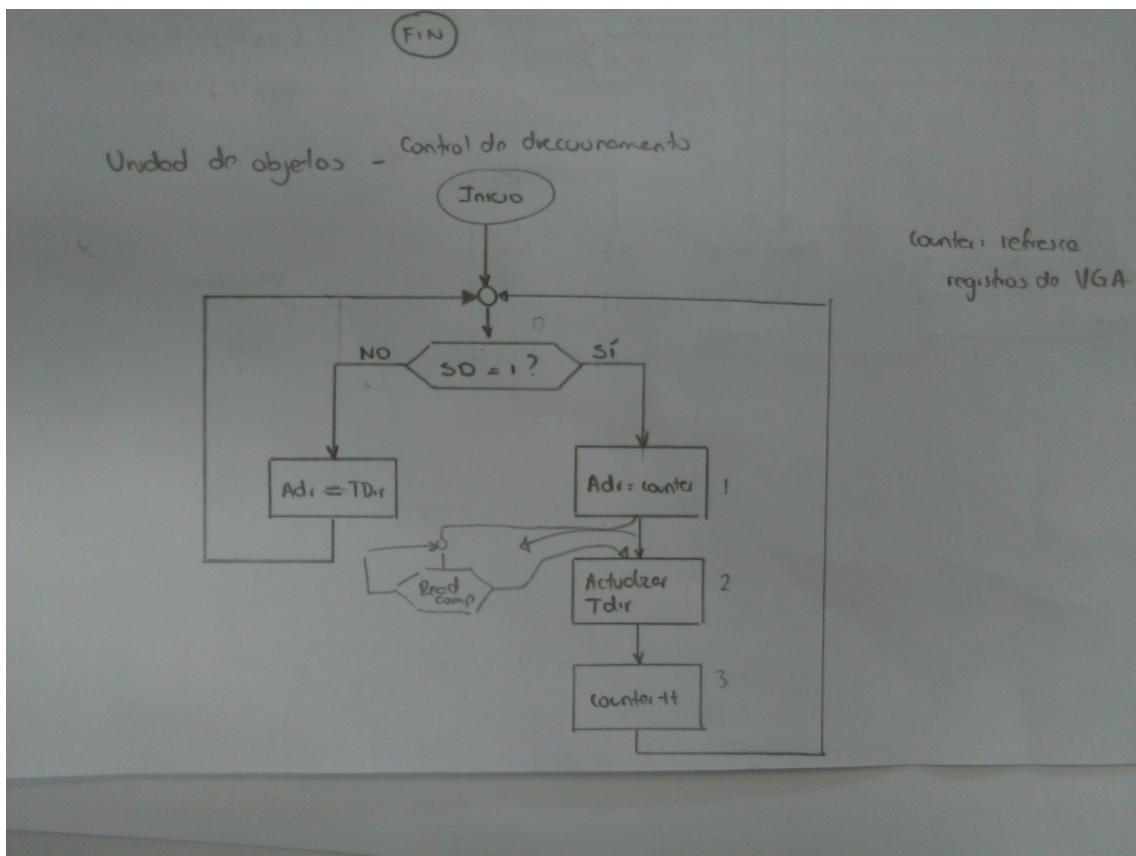


Figura 10: Diagrama de Bloques del Control VGA - Sección de Direccionamiento

**Fecha: 6 de Septiembre**

**Integrantes:** Keylor Mena y Luis Merayo

**Hora:** 7:00 - 9:00

**Actividad:**

se realizo un diseño la maquina de estados del control RTC usando el método de factorizacion. por lo cual tenemos la jerarquía de la figura 14 el flujo de cada maquina de estados estan en las figuras 15,16,17.

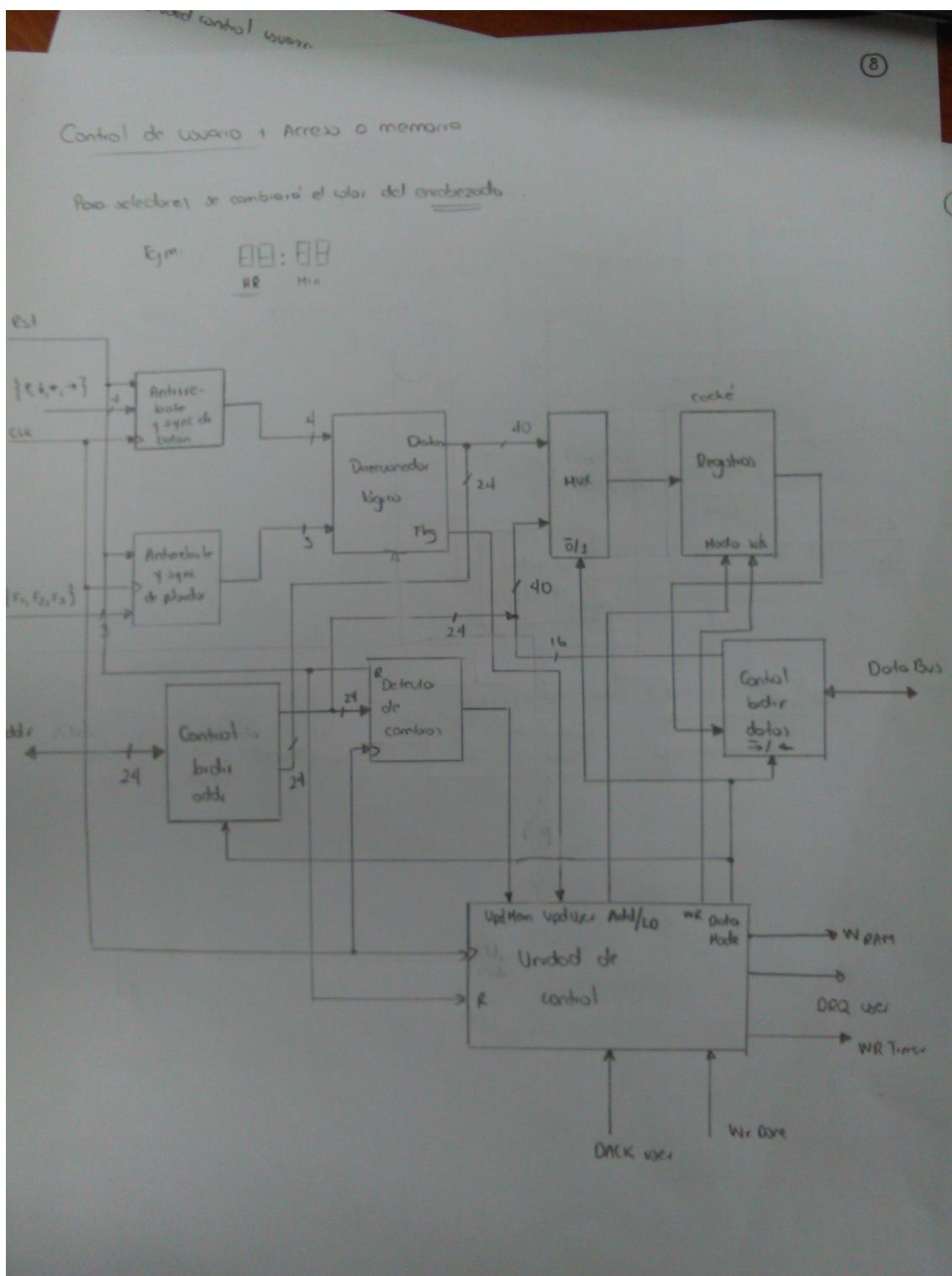


Figura 11: Diagrama de Bloques del Control de Usuario

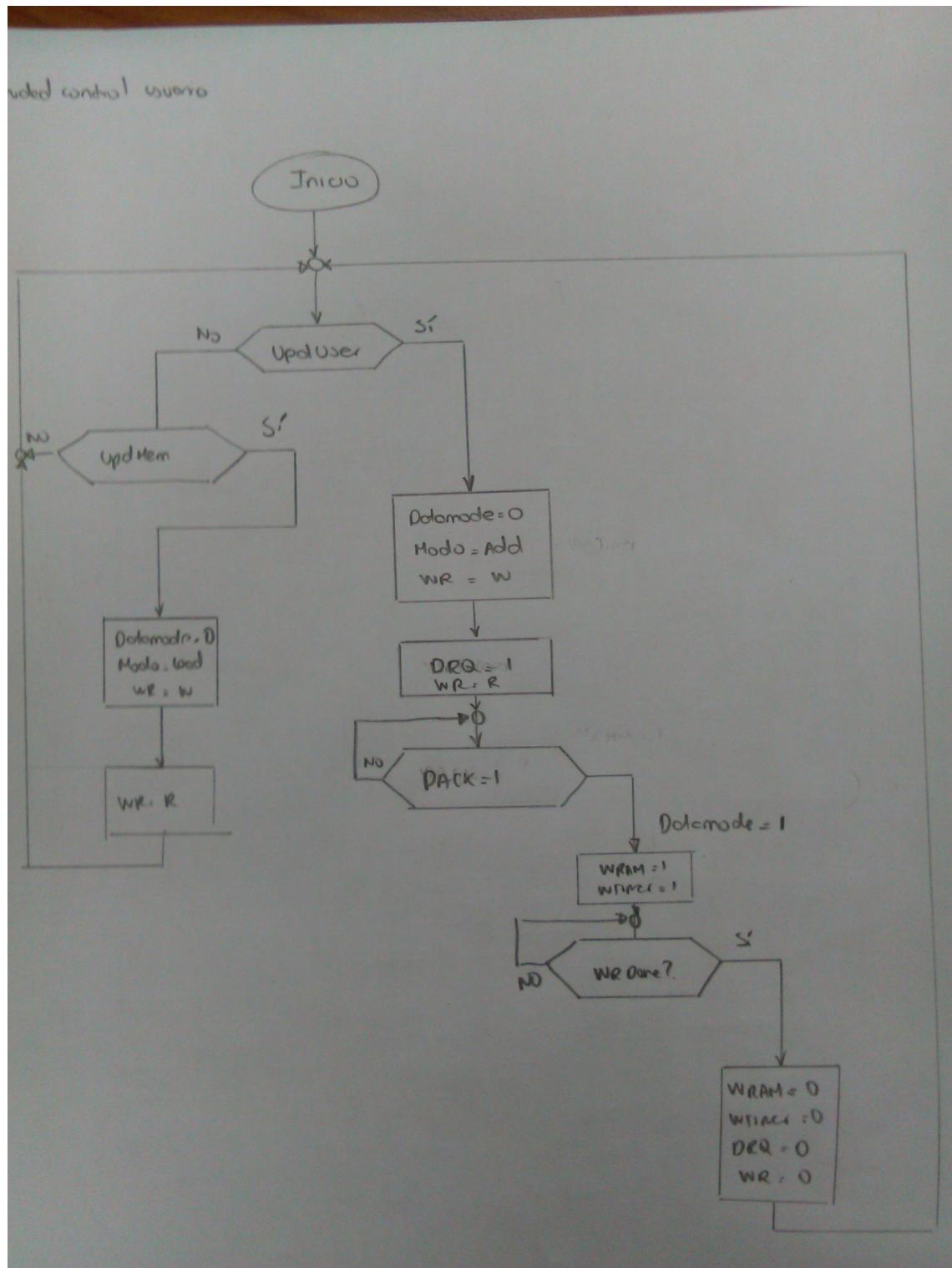


Figura 12: Diagrama de Flujo del Control de Usuario

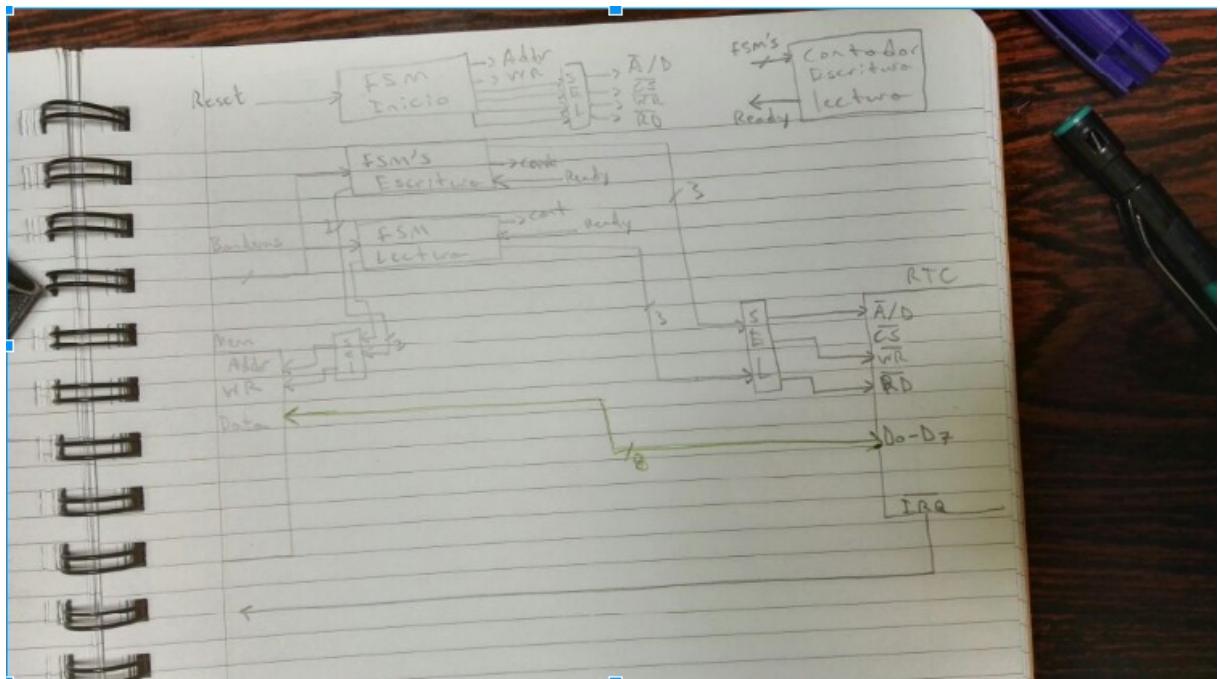


Figura 13: Diagrama del control RTC

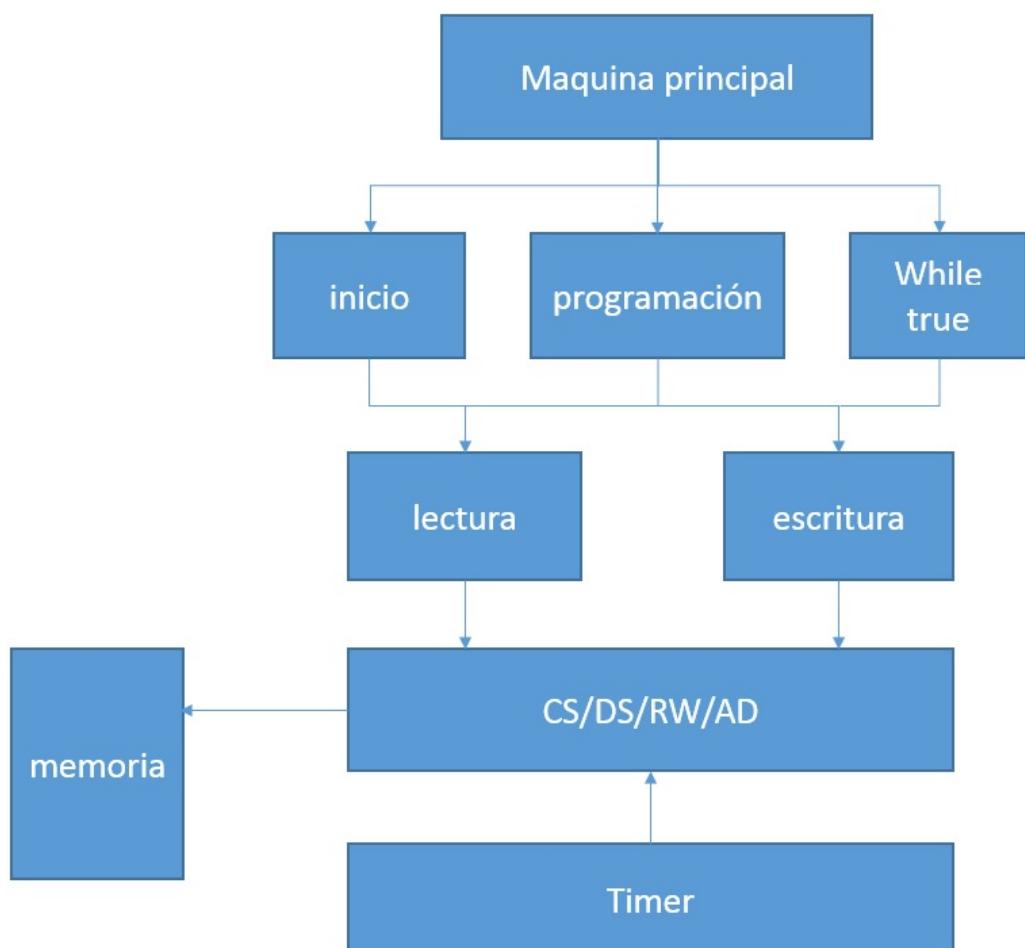


Figura 14: jerquia de la RTC

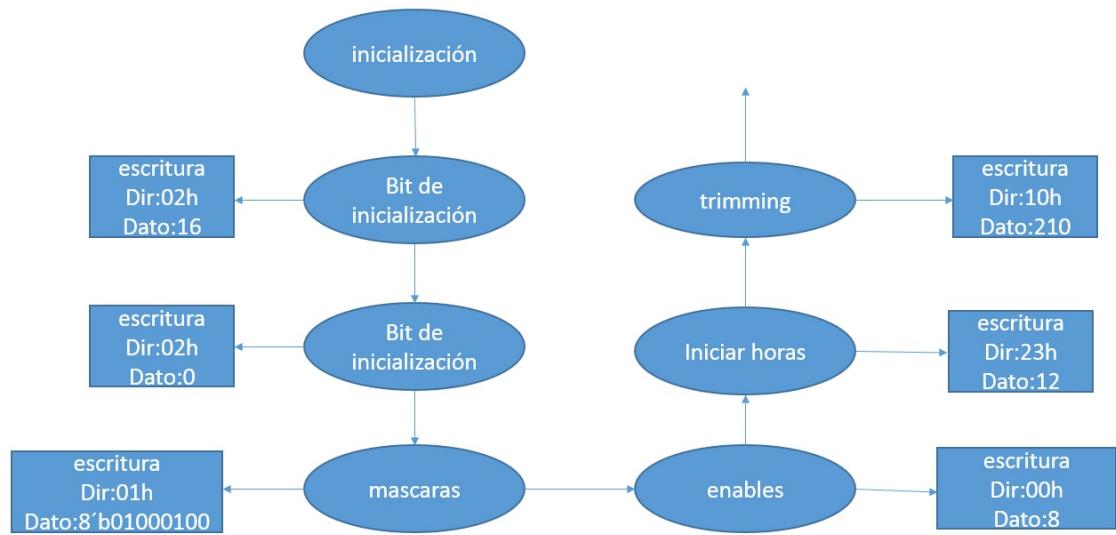


Figura 15: flujo del proceso de inicialización

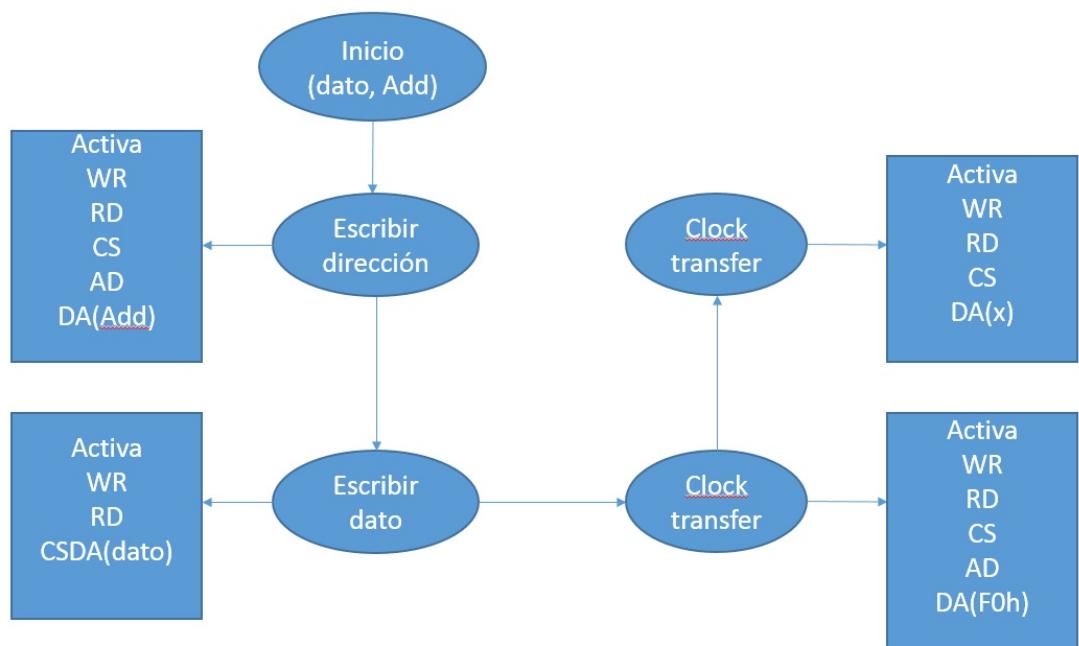


Figura 16: flujo del proceso de escritura

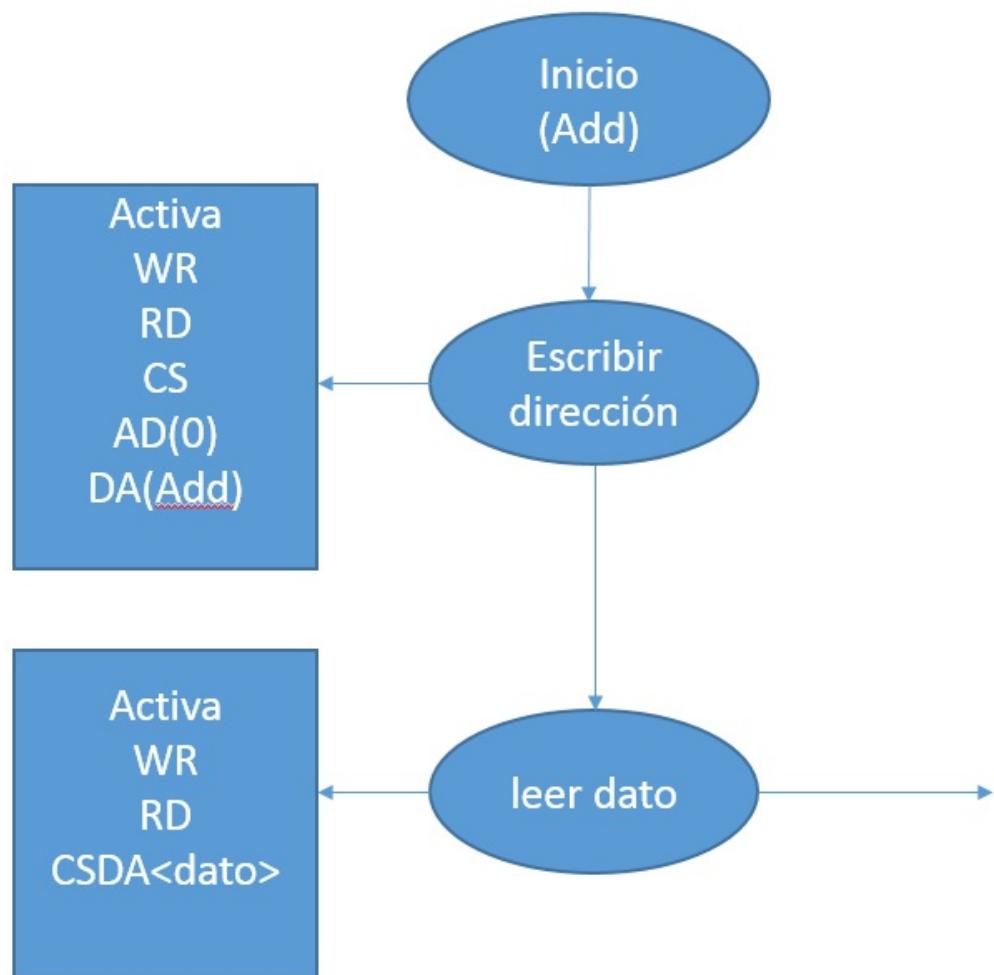


Figura 17: flujo del proceso de lectura

**Fecha: 11 de Septiembre****Integrantes:** Luis Leon**Hora:** Variada**Actividad:**

Se implementó el control VGA empleando una ROM para la plantilla. Sin embargo, esta implementación empleó muchos recursos valiosos de la VGA, destacando que se usaban 75 % bloques LUT, lo que obligó a hacer un replanteamiento del bloque.

**Fecha: 12 de Septiembre****Integrantes:** Luis Leon**Hora:** 14:00-21:00**Actividad:**

Se asistió a la tutoría del laboratorio de Diseño de Sistemas Digitales, donde se adquirió la idea de comprimir la imagen para disminuir el uso de la memoria ROM y del problema descrito el día anterior. Asimismo, se recomendó usar colores fundamentales para evitar un uso excesivo de los recursos de la FPGA.

Por otro lado, se inició con hacer una nueva interfaz usando un programa de dibujo vectorial, que en este caso se llama Inkscape, que es de uso libre. La ventaja que supone usar un diagrama de dibujo vectorial es que se puede escalar sin perder calidad de la imagen, a comparación del uso de mapa de bits (bitmap). La interfaz se encuentra a en la Figura 18. Esta interfaz será combinada con otras plantillas que tendrán los indicadores y los números. El cambio entre plantillas será efectuado mediante punteros de memoria.

Finalmente, se ha escrito el script de Matlab para poder convertir la imagen en un archivo de texto para importarla a la ROM mediante la síntesis y se elaboró el diagrama de bloques del control VGA nuevamente, el cual se ilustra en la Figura 19.

**Enlaces de interés:**

- Carpeta de scripts de Matlab: <http://bit.ly/2cz3gi0>

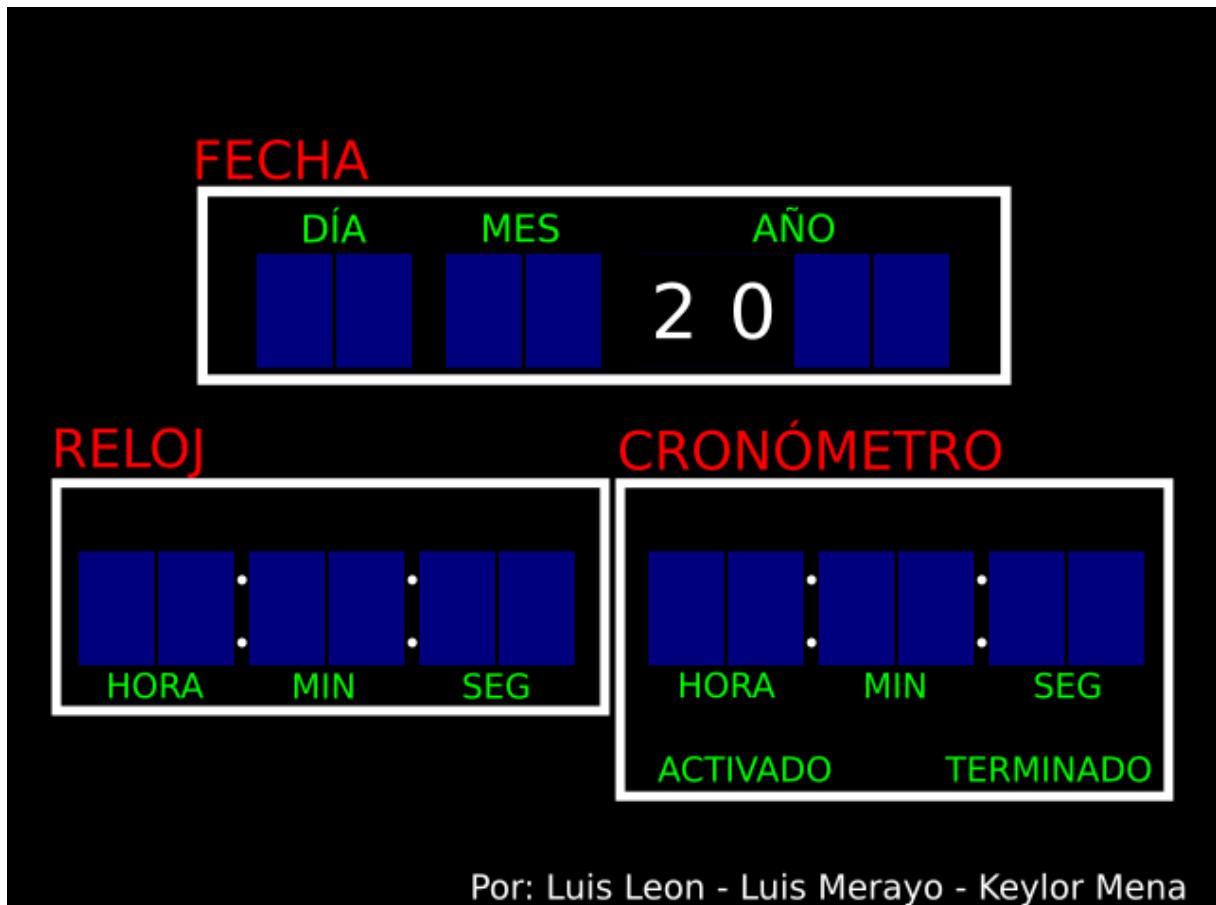


Figura 18: Plantilla de interfaz de usuario

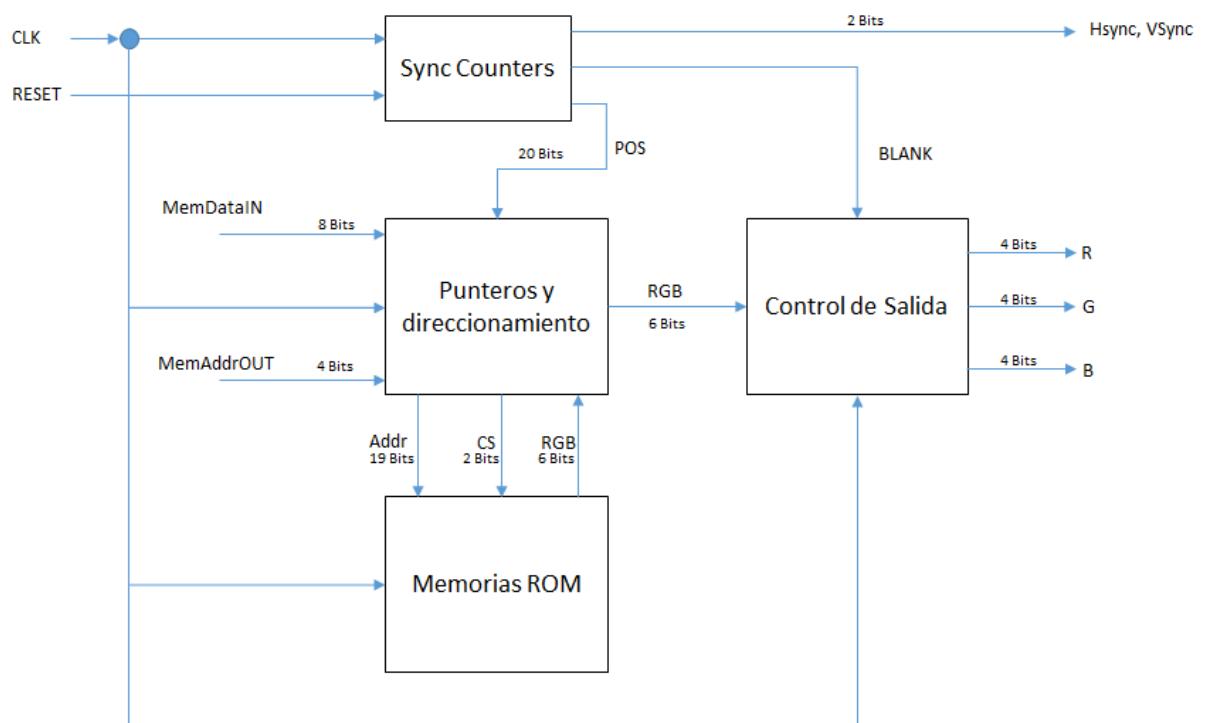


Figura 19: Diseño de bloques (2) del control VGA

**Fecha: 13 de Septiembre****Integrantes:** Luis Leon**Hora:** 7:00 - 14:00**Actividad:**

Se comenzó con la implementación del control VGA en HDL. Se crearon los diversos bloques y quedaron listos para someterlos a prueba. El día de mañana se realizarán pruebas intensivas. Asimismo, se elaboró el script de Matlab para poder decodificar los resultados del testbench en la imagen correspondiente. Para ello, se estableció un formato, el cual es \_XXX\_YYY\_CCC, donde X es la posición en el eje X, Y en el eje Y y C es el color en medio byte de colores RGB.

**Enlaces de interés:**

- Carpeta de scripts de Matlab: <http://bit.ly/2cz3gi0>

**Fecha: 14 de Septiembre****Integrantes:** Luis Leon**Hora:** 7:00 - 10:00 y 18:00 - 21:00**Actividad:**

Se realizaron las pruebas del módulo, encontrando problemas con los punteros y un ligero problema con la memoria emulada en el testbench, lo que provocaba que la imagen se vea como se plasma en la Figura 20. Asimismo, se encontró un serio deterioro de la imagen producto de la ampliación de la imagen, por lo cual, se decidió implementarla en la escala normal como fue diseñada (640X480). Esto, de forma consciente de que se utilizarán más recursos, pero con la ventaja de que esto elevará la calidad del trabajo. Al final, el resultado obtenido se puede apreciar en la Figura 21. Con esto, el trabajo sobre el módulo de la VGA queda concluido y listo para que los demás compañeros puedan ejecutar su trabajo.

**Fecha: 15 de Septiembre****Integrantes:** Keylor Mena**Hora:** 12:00 - 17:00**Actividad:**

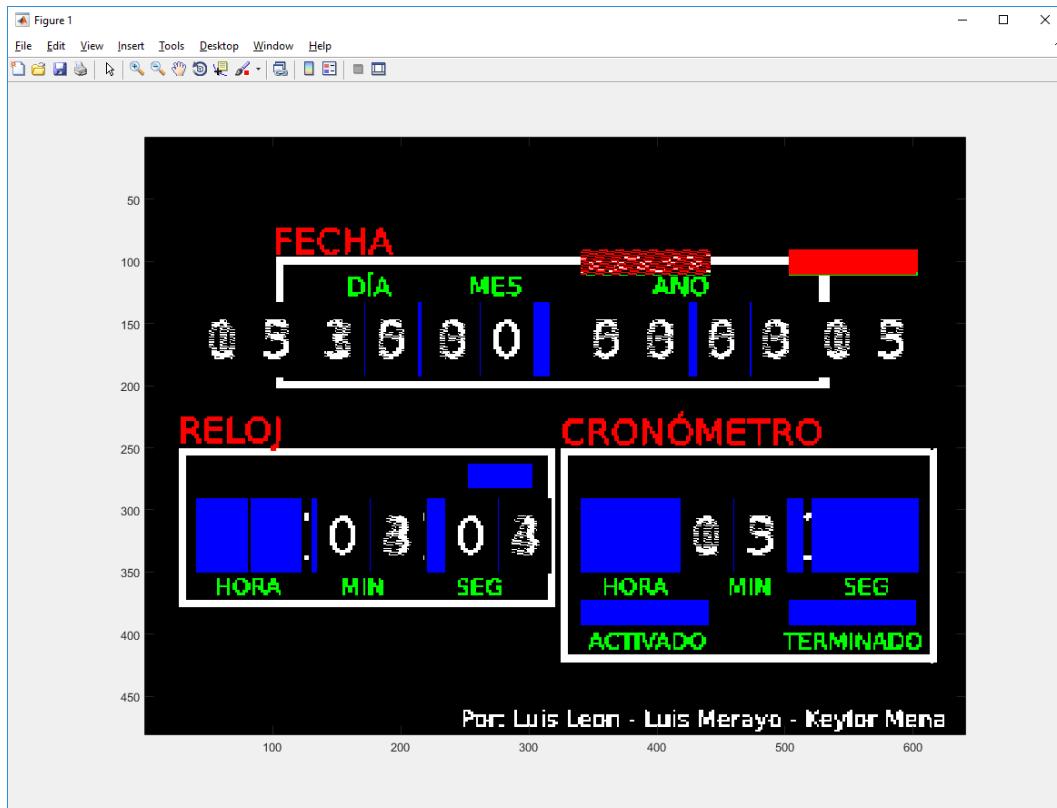


Figura 20: Problemas con la memoria VGA



Figura 21: Resultado exitoso de la prueba de la VGA

Se realizo el primer intento de la union de los modulos internos de la rtc Sin exito alguno.

### **Fecha: 17 de Septiembre**

**Integrantes:** Keylor Mena

**Hora:** 12:00 - 17:00

**Actividad:**

Se realizo el segundo intento de la union de los modulos internos de la rtc con mejoras significantes a lo anterior pero sin funcionar

### **Fecha: 19 de Septiembre**

**Integrantes:** Keylor Mena y luis Leon

**Hora:** 12:00 - 17:00

**Actividad:**

Se realizo el tercer intento de la union de los modulos internos de la rtc Los modulos unidos en el rtc

### **Fecha: 24 de Septiembre**

**Integrantes:** Luis Leon y Luis Merayo

**Hora:** 10:00-12:00 am

**Actividad:**

Se corrigieron la activación de los triestados , de acuerdo con los ejemplos brindados por el ISE, y se eliminaron warnings que había en la VGA.

Ademas, se encontraron errores con los modulos del control RTC, ya que el programa notificaba que existian modulos desconectados, y se procedio a corregir los problemas.

**Fecha: 23 de Septiembre**

## Integrantes: Todos

**Hora:** 7:00 - 10:00 y 18:00 - 21:00

## Actividad:

Se realizó la unión de los módulos de la RTC, el control de usuario y la VGA, se realizó la simulación post síntesis 22, y se verificó que los tiempos estuvieran correctos.

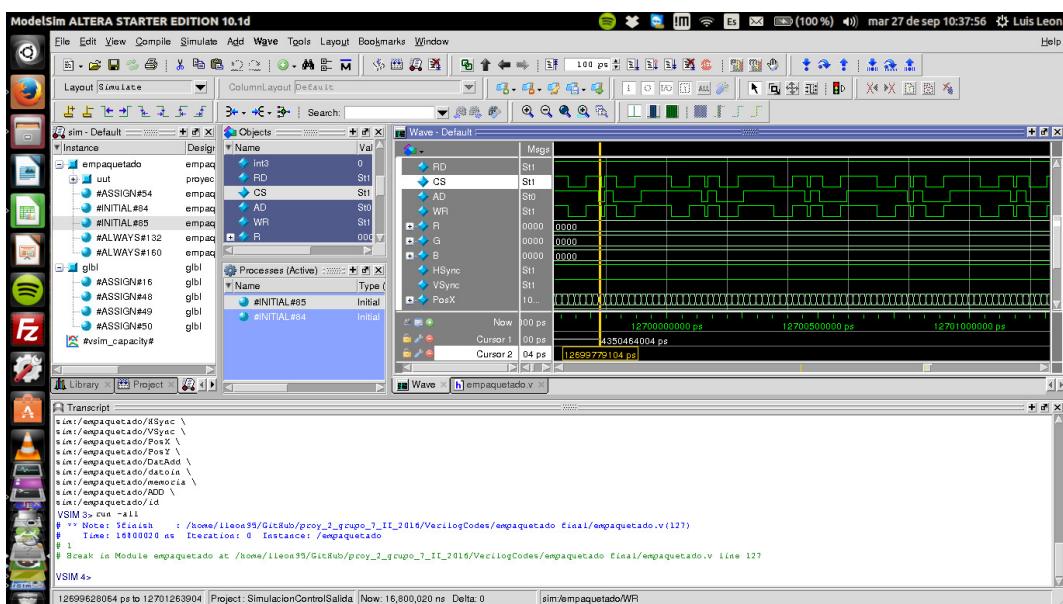


Figura 22: Tablas de tiempos de escritura y lectura

**Fecha: 26 de Septiembre**

**Integrantes:** Luis Leon y Luis Merayo

**Hora:** 9:00-12:00 am

**Actividad:**

Se realizaron pruebas en la nexys 4 y el monitor con los modulos del proyecto conectados completamente, cabe destacar que el proyecto no funciono, y la pantalla muestra numeros aleatorias, pero no hubo cambio en ninguno de ellos.

**Fecha: 27 de Septiembre**

**Integrantes:** Luis Leon y Luis Merayo

**Hora:** 10:00-12:00 am

**Actividad:**

Se corrigieron la activación de los triestados , de acuerdo con los ejemplos brindados por el ISE, y se eliminaron warnings que había en la VGA.

Ademas, se encontraron errores con,os modulos del control RTC, ya que el programa notificaba que existian modulos desconectados, y se procedio a corregir los problemas.

**Fecha: 28 de Septiembre**

**Integrantes:** Luis Leon

**Hora:** Durante varios periodos del día

**Actividad:**

Se reimplementó el sistema de punteros del módulo de la VGA con la finalidad de reducir el periodo de propagación de las señales, especialmente, las entradas con referencia al reloj hacia la salida.

Esta nueva implementación optimizó de 89MHz que se tenía como reloj máximo a 111.430MHz, realizando una importante mejora para correr el proyecto con el reloj incorporado dentro de la Nexys 4.

Dentro del cambio principal en este sistema de punteros es que se aprovechan los campos de memoria ROM de interfaz para poder identificar los inicios de los punteros para calcular la sustitución de la salida VGA, haciendo selección de memoria.

**Fecha: 28 de Septiembre**

**Integrantes:** todos

**Hora:** 10:00-12:00 am

**Actividad:**

Se logro leer los segundos de la RTC, pero los demas datos no se leen. Ademas el bloque de control de usuario no ha funcionado, ya que cuando se activa su respectivo switch de activacion, la lectura de segundos se detiene permanentemente.

**Fecha: 29 de Septiembre**

**Integrantes:** todos

**Hora:** 7:00 am-4:30 pm

**Actividad:**

Se rediseñó el control de usuario, debido a las fallas del anterior, ahora se conforma de una maquina de estados de punteros de memoria, dos registros para incrementar y disminuir los valores del clk , y una maquinas de estados de escritura.

**Fecha: 29 de Septiembre**

**Integrantes:** todos

**Hora:** 6:00 -10:30 pm

**Actividad:**

La escritura y la lectura funcionan, no obstante la programacion del timer no se ha podido realizar. Ademas la pantalla muestra pequeños glitches.