Escuela de IngenieríÂa en Electronica

Laboratorio de Diseño de Sistemas Digitales

Bitácora

Proyecto:

Control y programación RTC con Nexys3

Profesor:

Alfonso Chacón RodríÂguez

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: 9 de Noviembre

Integrantes: todos

Hora: 20:00 -22:30 pm

Actividad:

Se diseño el primer intento de aproximarse a un diagrama de bloques de nivel 2. Esto se puede notar en la figura 1, en este se puede notar 5 bloques principales, uno de ellos es el microprocesador echo con el picoblaze. Ademas podemos notar que este tiene como entrada las señales PosX y PosY de la VGA, de esta manera se controla la lectura de la RTC cuando la VGA se encuentra pintando en algunos lugares de la pantalla, Ademas la entrada IRQ controla cuando la etapa de sonido funciona.
La memoria alimenta con los datos que debe pintar la VGA, estos datos vienen de la RTC directamente cuando se encuentra actualizando los datos. ademas que posee un espacio para la señal IRQ y el teclado.
El teclado introduce a la memoria los datos que el usuario desea cambiar para que se muestre inmediatamente en la VGA. Ademas una vez que el usuario desea introducir el cambio en la RTC, este bloque se comunica con el controlador RTC para introducir el cambio.

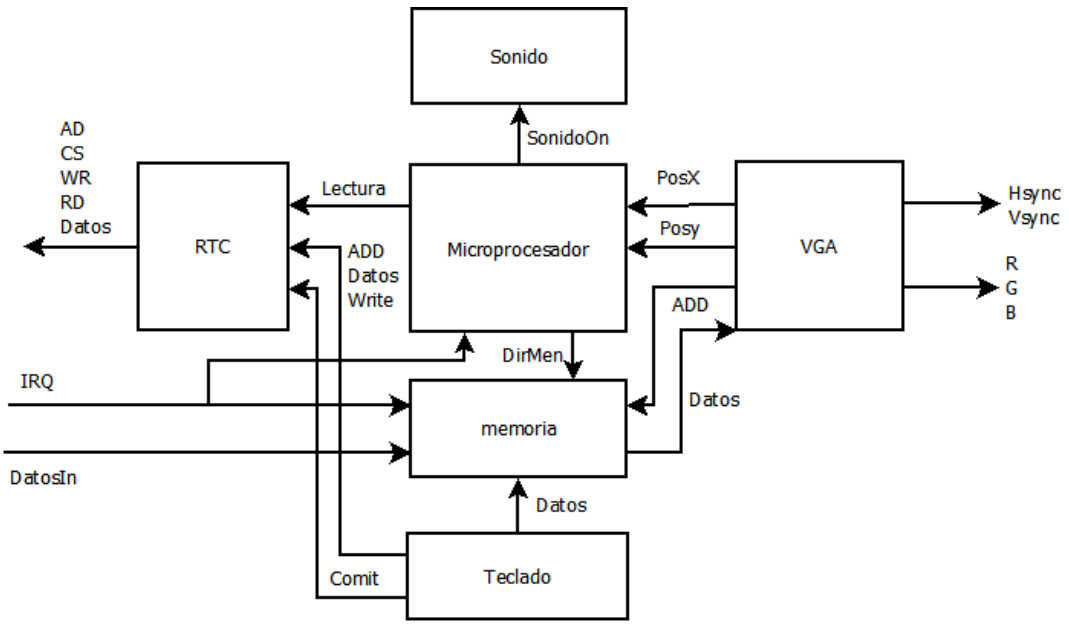


Figura 1: Diagrama de bloques nivel 2 primer intento

Fecha: 10 de Noviembre

Integrantes: Luis Leon

Hora: 14:00 - 16:00

Actividad:

Algunos de los miembros del equipo asistieron a la tutoría de hoy, donde se aclaró el funcionamiento del microcontrolador PicoBlaze para FPGA. Esto ha originado algunos cambios en el diseño del día de ayer, al saber que el funcionamiento del Picoblaze es con base a un puerto I/O controlado por una memoria (físico representado en memoria). Esto ha generado el cambio en el diseño, por lo cual, se debe referir a la figura 2.

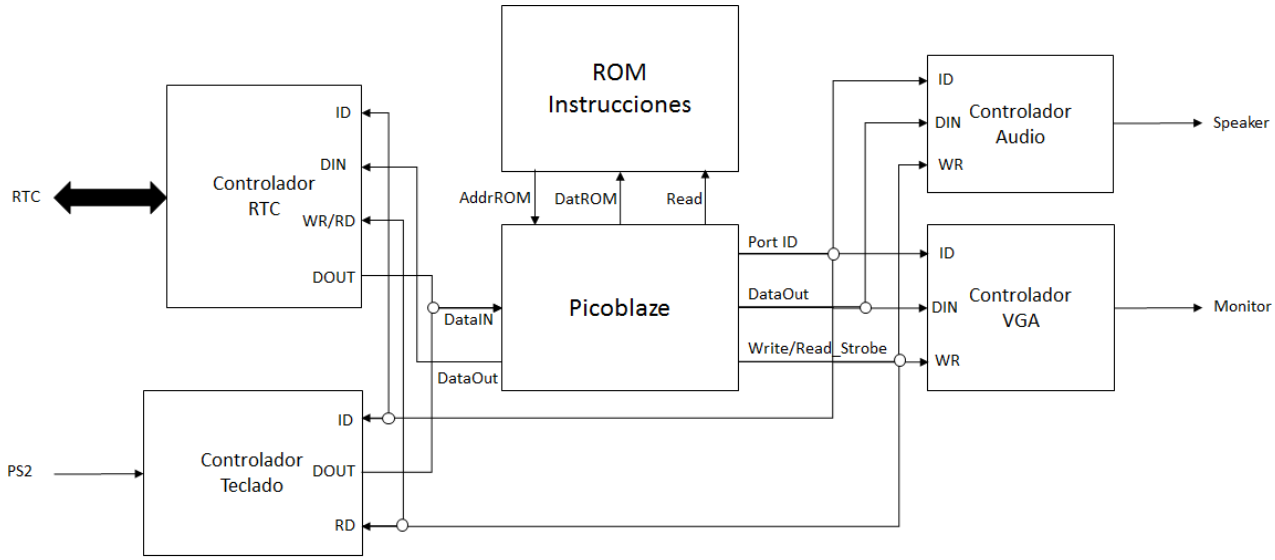


Figura 2: Diagrama de bloques nivel 2 segundo intento

Por otro lado, el mapa de memoria mediante el cual, se controlarán todos los periféricos es representado en la figura 3. Este mapa contempla que:

- La VGA contiene un banco de registros estabilizados que consultará dependiendo de la posición del cursor. Estos registros son accedidos por el Port ID 1 y modificados por el Port ID 2.
- El teclado brinda el código de la tecla presionada. Luego de que se hace el ReadStrobe en el controlador de teclado, este código se limpia y espera la próxima tecla o, bien, que se levante la tecla (Evento KeyUp).
- El controlador de audio solo tiene un registro que le habilita para que suene el speaker o no.
- El controlador de RTC tiene todos los registros que son posible cambiar. Cuando se hace un WriteStrobe, se habilita el ciclo de escritura en la RTC en el dato dado por la dirección del ID (Port ID). Cuando se hace ReadStrobe, se puede ejecutar una lectura o algo similar en el próximo ciclo.

- Dependiendo del orden de magnitud, se habilitará la lectura de datos. Ejemplo: Si el Bit 5 está activo, es un dato que va a la RTC, si el Bit 4 está en 1 pero el Bit 5 no, entonces va a audio y así sucesivamente, de acuerdo al mapa.

de Memoria.png

Módulo	Dirección B10	Dato	Comentario
VGA	1	Dirección Registro	Previo a cambiar un dato, se debe dar la dirección
	2	Dato Registro	Luego de la dirección se da el dato
Teclado	5	Tecla pulsada	Después del Read_Strobe, el dato se vuelve 0
Audio	9	Activación	Indica si va a sonar o no
RTC	17	Segundos	Reloj - Se lee mediante Read_Strobe y se activa el ciclo de escritura con Write_Strobe
	18	Minutos	
	19	Horas	
	20	Días	
	21	Meses	
	22	Años	
	23	Segundos	Cronómetro - Se lee mediante Read_Strobe y se activa el ciclo de escritura con Write_Strobe
	24	Minutos	
	25	Horas	
	26	Timer Activado	
	27	Timer Disparado	

Figura 3: Mapa de memoria

Fecha: 9 de Octubre

Integrantes: todos

Hora: 11:00 -13:00 pm

Actividad:

Se reviso los cambios relizados el dia anterior por todos los compañeros, en esta se aprobó por todos el diseño a nivel de bloques, pero se realizo cambio al banco de registro de la figura 3 donde se elimino los bancos de registros ioports de la rtc dejando un solo registro que controla las direcciones en las que el debe escribir, ademas de crear un bus de entrada de datos
Los registro de la rtc se pasaron a la memoria scratch para guardar los datos de la rtc una vez que lea. Se puede notar mas facil en la figura 4

Integrantes: todos

Hora: 15:00 -18:00 pm

Actividad:

Tomando en cuenta algunos datos y consejos que el profe acoto al proyecto se decidió integrar un deco a la salida del id. De esta manera se puede tener un control mas fácil en cada modulo creando un CS en cada modulo y que este sea el que active

de Memoria2.png

Modulo	Dirección	Dato
VGA	1	Dirección de registro
	2	Dato de registro
Teclado	5	Dato de entrada del teclado
Audio	9	Indica si tiene que salir el audio
RTC	17	Salida de direcciones al modulo de control de la RTC
Dato_in	18	Dato de entrada directamente de la RTC
Auxiliar	33	Datos auxiliares de activación de registros o modulos

Figura 4: Mapa de memoria

los procesos, como se puede notar en la figura 5.
Otro aspecto a considerar es que se puede tomar este deco para generar las direcciones de la RTC y ahorrar procesos del microprocesador generando datos y direcciones en un solo acceso. Pensando en este aspecto de debe considerar tener la memoria scratch con un direccionamiento de igual manera para el id de cada dato como se muestra en la columna de "Dirección B10"de la figura 3.
De esta manera solo se ocupa una variable o un registro para accesar a la memoria scratch y generar el valor de dirección en la RTC

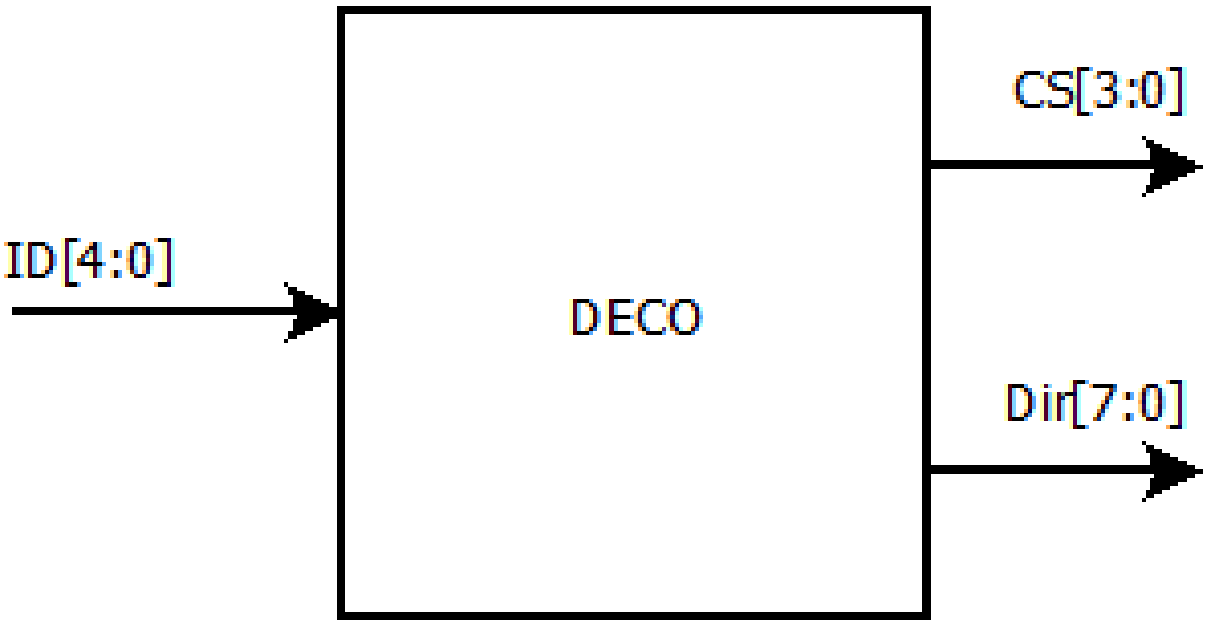


Figura 5: Deco de ID

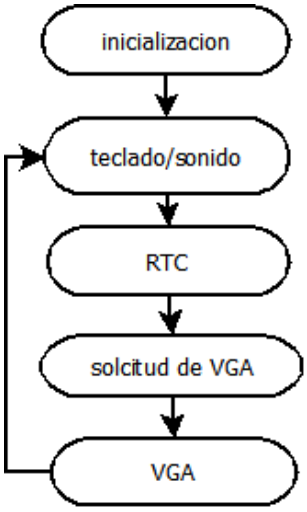
Fecha: 12 de Octubre

Integrantes: todos

Hora: 16:00 -18:00 pm

Actividad:

Después de un análisis de los avances del día anterior se procedió a diseñar un primer intento de un diagrama de flujo de el micro procesador.
Para esta efecto se buscaron crear 5 ciclos, como se muestra de la figura 6, donde el primer ciclo es el de inicializacion falta diseño.
Después de este un ciclo de teclado, como se muestra en la figura 7, en este se



De Flujo Princiapal.png

Figura 6: Diagrama de flujo principal

usan los espacios de memoria y Id de la tabla 1, 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 8, 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 9, 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.

Fecha: 13 de Octubre

de teclado irq.png

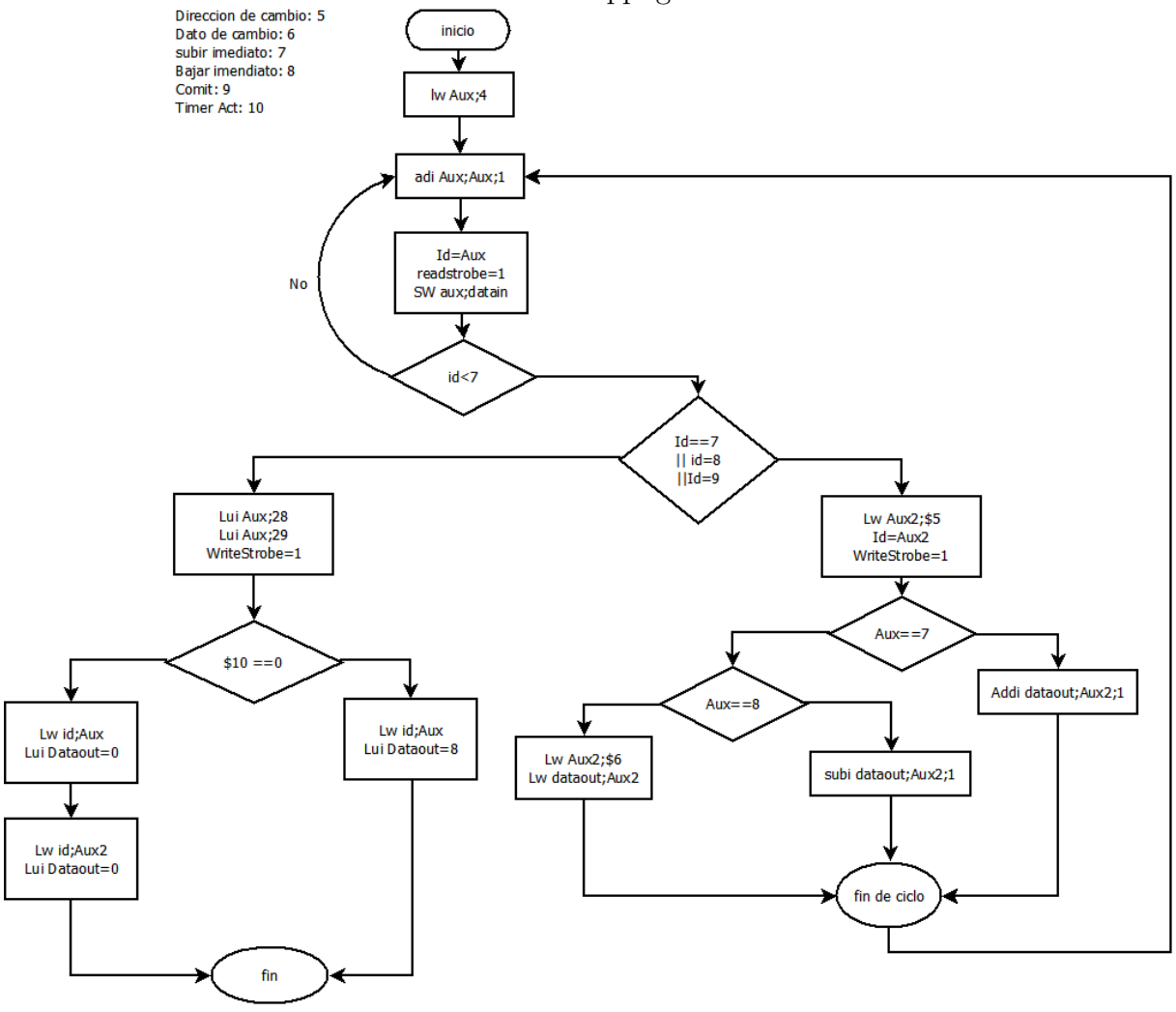
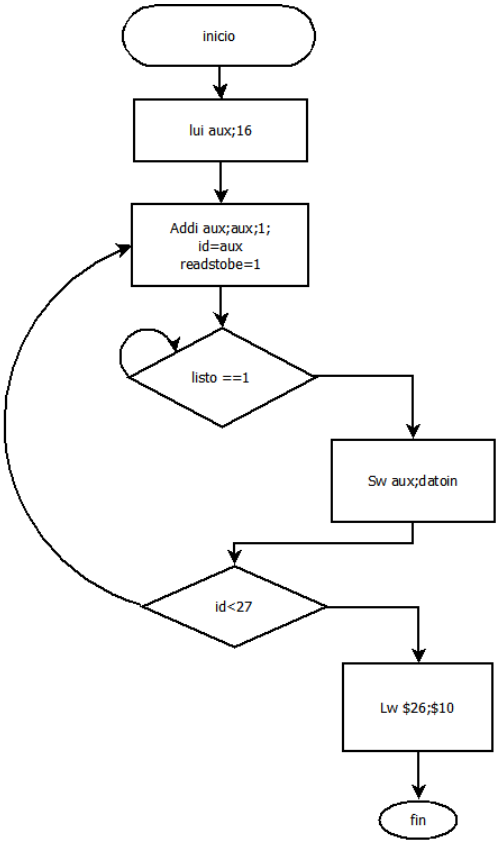


Figura 7: Diagrama de flujo principal

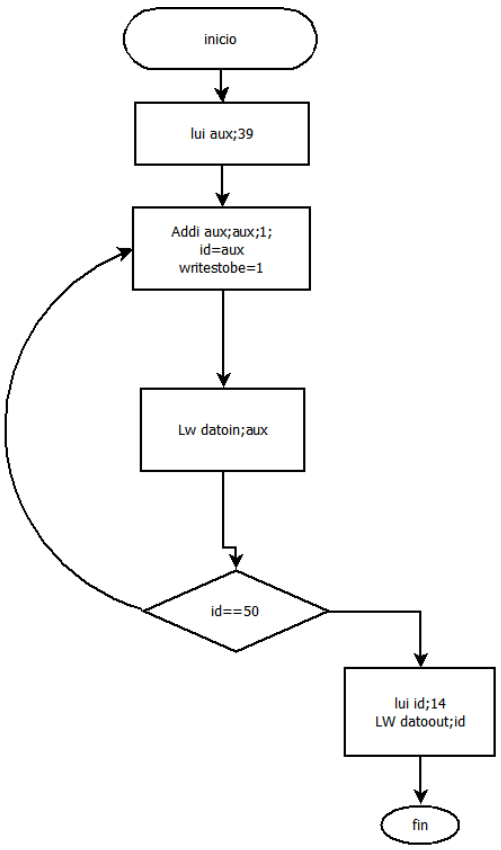
Variable	Deco	memoria
Segundos=	17	33
minutos=	18	34
horas=	19	35
Dias=	20	36
Meses=	21	37
Años=	22	38
SegundosT=	23	h41
minutosT=	24	h42
horasT=	25	h43
timerdisp	26	
timeract	27	
Staus1	28	0
Staus2	29	1



de flujo RTC.png

Figura 8: 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 9: Diagrama de flujo principal

Tabla 1: resumen del Deco

Id	CS	ADD
0	MemoriaEstable	
1	Escritura	
2	VSync	
5	teclado	
6	teclado	
7	teclado	
8	teclado	
9	teclado	
10	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

Tabla 2: resumen del Deco

Id	CS	ADD
2	VSync	
5	teclado	Direccion
6	teclado	Dato
7	teclado	Commit
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

Integrantes: Todos

Hora: 13:30 -16:10 pm — 19:30-21:30

Actividad:

Con respecto al teclado:
Se ha probado un ejemplo disponible para Nexys 4 DDR, el cual, consiste en un módulo para obtener datos procedentes del teclado de forma unidireccional, mostrando todos los códigos recibidos mediante el protocolo serial PS/2. En este se ha descubierto el siguiente comportamiento:

- Proceso 1:
  Al presionar la tecla (evento KeyDown), se recibe el código de la tecla que fue presionada (Scan Code).
- Proceso 2:
  Al mantener la tecla (evento KeyPress), no ocurre nada y el código de la tecla se mantiene como el último recibido.
- Proceso 3:
  Al soltar la tecla (evento KeyUp), se recibe el código de liberación (F0) seguido del código de la tecla que fue liberada.

Conociendo el procedimiento, se expondrá un ejemplo para aclarar el funcionamiento de este ejemplo. Suponer que se presiona la tecla Enter, que tiene un Scan Code 5A.
Al no presionar la tecla aún, en el display: **00 00 00 00**
Al presionar la tecla, en el display: **00 00 00 5A**
Al mantener la tecla, en el display: **00 00 00 5A**
Al soltar la tecla, en el display: **00 5A F0 5A**

Finalmente, al terminar el día, se implementaron los bloques más significativos, excluyendo al control, en Verilog. Eso con base al diagrama 10. Mañana se realizará el control.

Con respecto al audio:

Con respecto al picoblaze:

Enlaces de interés:

- Ejemplo del teclado en Digilent: https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-keyboard-demo/start

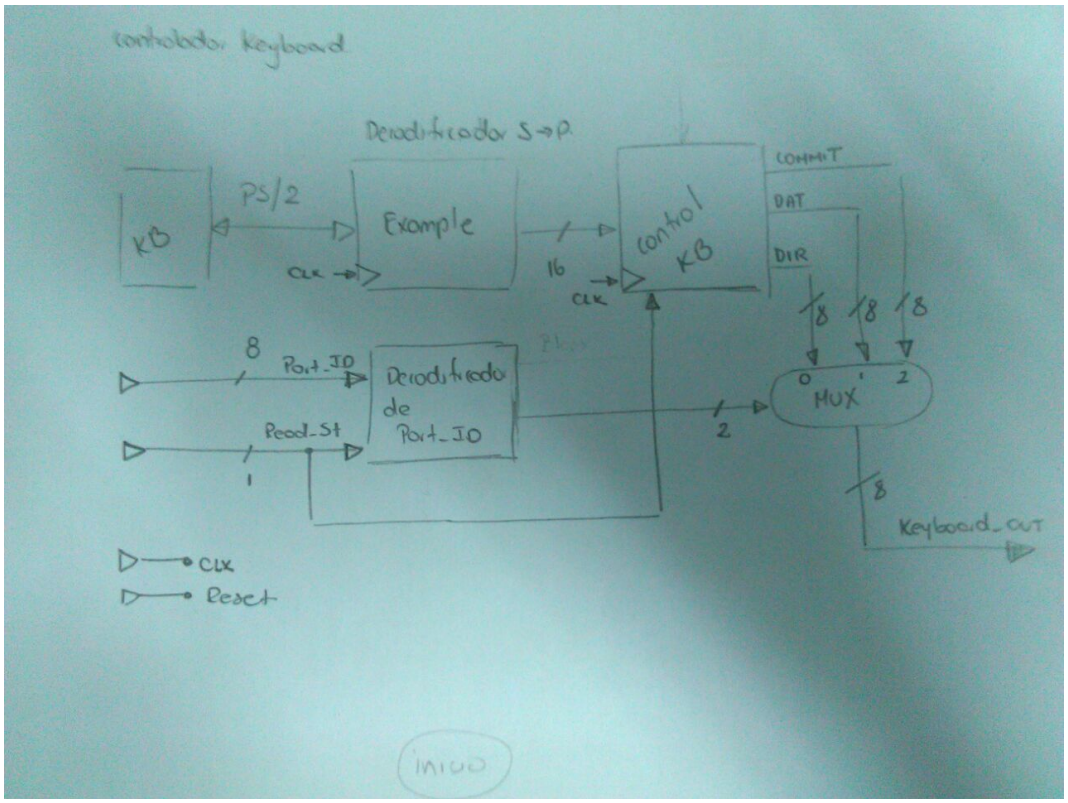


Figura 10: Diagrama de bloques del controlador de teclado

Fecha: 14 de Octubre

Integrantes: Luis Leon

Hora: 7:00 - 9:20

Actividad:

Se realizó el módulo de control del controlador de teclado, señalado en el diagrama 10 como Çontrol KB". Se ha comentado el equipo de trabajo de la conclusión de dicho controlador y quedará pendiente la simulación mediante un testbench, donde se deberá representar el comportamiento del teclado bajo el protocolo PS/2.

Fecha: 15 de Octubre

Integrantes: Luis Leon

Hora: 12:30 - 16:30

Actividad:

Se realizó el modelo comportamental del teclado PS/2 para poder probar el controlador de teclado mediante el testbench. Se han hecho algunas correcciones y se ha tomado prevista de casos repetitivos de teclas que no se habían tomado en cuenta. Al finalizar el proceso de trabajo, el controlador está listo para ser implementado en el sistema principal y debidamente probado. En la figura 11 se muestra el comportamiento de la simulación gráfica, donde se aprecia los cambios en los registros al pulsar y liberar las teclas. Por otro lado, en la figura 12 se muestra una tabla generada mediante el script del testbench para corroborar los resultados.
Se recuerda que el funcionamiento consta de tres etapas: la consulta de la dirección que se ha cambiado (en memoria), el dato con el nuevo valor y verificación de commit.

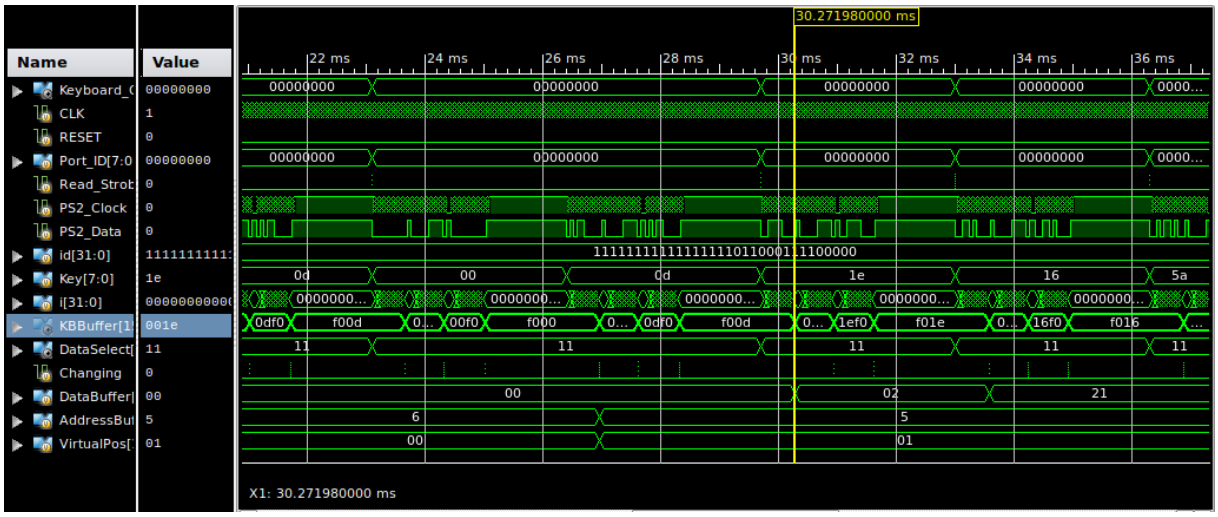


Figura 11: Diagrama de bloques del controlador de teclado

Read_Strobe	PORT_ID	Keyboard_Output	Key
1	05	00	00
0	06	00	00
1	06	00	00
0	07	00	00
1	07	00	00
0	07	00	00
1	05	0b	78
0	06	0b	78
1	06	01	78
0	07	01	78
1	07	01	78
0	07	00	78
1	05	06	05
0	06	06	05
1	06	00	05
0	07	00	05
1	07	00	05
0	07	00	05
1	05	06	1e
0	06	06	1e
1	06	02	1e
0	07	02	1e
1	07	00	1e
0	07	00	1e
1	05	06	16
0	06	06	16
1	06	21	16
0	07	21	16
1	07	00	16
0	07	00	16
1	05	06	5a
0	06	06	5a
1	06	21	5a
0	07	21	5a
1	07	01	5a

Figura 12: Diagrama de bloques del controlador de teclado