



Universidad Tecnológica de la Mixteca

Electrónica Digital II

Profesor: M.C. Felipe Santiago Espinosa

Proyecto Final – Cronómetro digital con salida a un monitor VGA

Alumnos:

López Bollo Jonathan
Peralta Rosales Oscar Esaú
Ventura Mijangos Giovanni

Cronómetro digital con salida a un monitor VGA

I. Introducción

En la lógica digital existen diversos dispositivos con los cuales podemos trabajar. En el proyecto final para la materia de Electrónica Digital II se usa un dispositivo FPGA, en particular una tarjeta Nexys2 de Xilinx modelo Spartan 3E.

El FPGA es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada mediante un lenguaje de descripción especializado.

El proyecto consiste en un cronometro digital que podrá ser visible en la pantalla de un monitor, contará con tres funciones básicas, iniciar, pausar y reiniciar. Las herramientas de desarrollo utilizadas son:

- Active VHDL.
- ISE de Xilinx.
- Adept (Programador de la tarjeta).

II. Objetivo

Elaborar e implementar un cronómetro digital mediante una tarjeta Nexys2 Xilinx modelo Spartan 3E de seis dígitos, los dos primeros para los minutos, los siguientes dos para los segundos, y los últimos para las centésimas de segundo. La salida se enviará a una pantalla usando el puerto VGA. Las funciones básicas que tendrá serán inicio, pausa y reinicio accesibles mediante los botones “push” de la misma tarjeta.

III. Desarrollo

En la figura 1.1 se muestra el esquema general del cronómetro, en primer instancia se divide en dos partes, la primera resuelve el conteo del cronómetro, y la otra usa los datos del conteo para mandarlos a pantalla. El sistema cuenta con 7 elementos fundamentales dos divisores de frecuencia, dos contadores, un filtro, un modulo para generar las imágenes, y un modulo para controlar la salida a la pantalla.

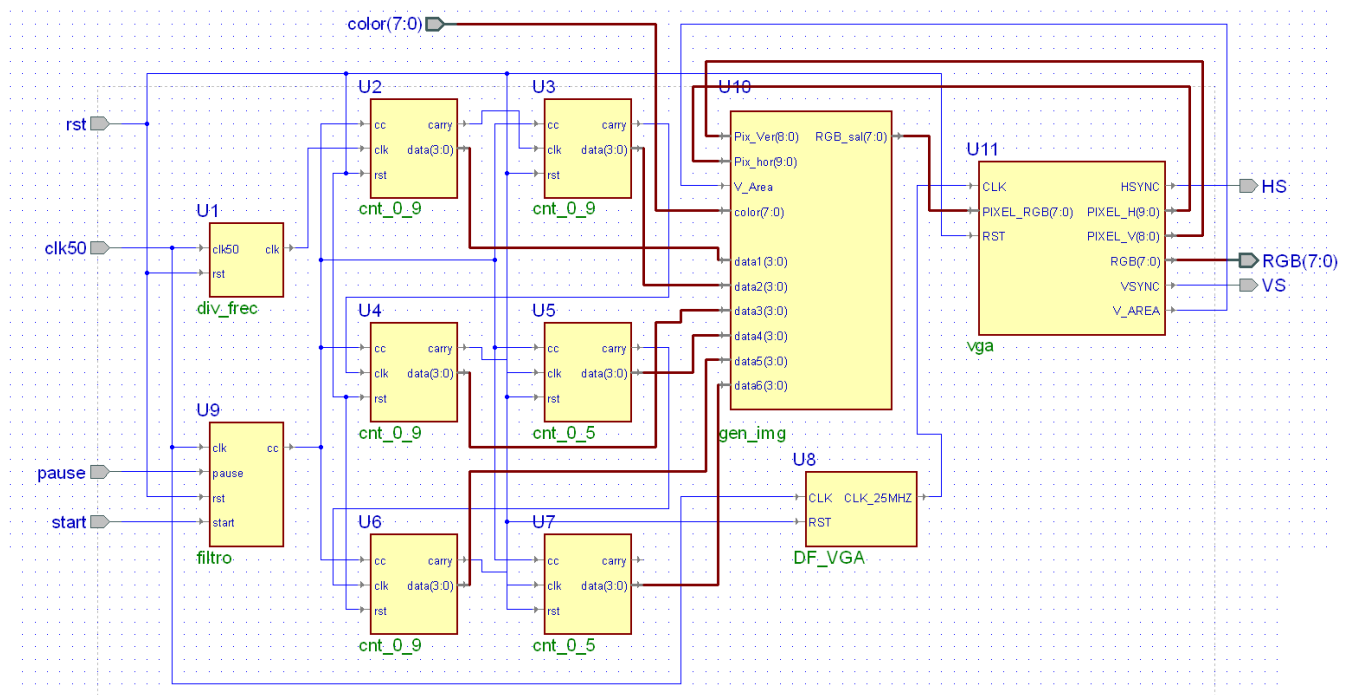


Fig. 1.1 Esquema general del cronómetro

- **div_freq** : Este módulo es el encargado de generar un pulso sincronizado cada centésima de segundo. El FPGA trabaja a una frecuencia de 50MHz generando un pulso de reloj cada 2×10^{-8} segundos, para poder sincronizar a centésimas de segundo es necesario que trabajemos a una frecuencia de 100Hz, por ello, si generamos un pulso cada 500,000 ciclos obtenemos esta frecuencia de trabajo. Así internamente el filtro cuenta con un vector de 20 bits usado como contador el cual se reinicia y genera un pulso cuando llega a 500,000.
- **DF_VGA** : Este divisor de frecuencia es ocupado para entregar una frecuencia de 25MHz, pues la pantalla será de una resolución de 640x480 según la tabla de especificaciones mostrada en la figura 2.

Formato	Reloj MHz	Horizontal (en píxeles)					Vertical (en líneas)				
		Vídeo activo	Porche delantero	Sincr.	Porche trasero	Total	Vídeo activo	Porche delantero	Sincr.	Porche trasero	Total
640x480@60Hz	25	640	16	96	48	800	480	9	2	29	520
640x480@60Hz	25,175	640	16	96	48	800	480	11	2	31	524
800x600@60Hz	40,000	800	40	128	88	1056	600	1	4	23	628
800x600@72Hz	50,000	800	56	120	64	1040	600	37	6	23	666
1024x768@60Hz	65,000	1024	24	136	160	1344	768	3	6	29	806
1024x768@75Hz	75,000	1024	24	136	144	1328	768	3	6	29	806

Fig. 2. Valores para diversas resoluciones de la pantalla.

- **Filtro** : Este módulo es el encargado de obtener las pulsaciones de los dos botones correspondientes a inicio y pausa, dependiendo del botón pulsado genera una señal para indicar a los contadores continuar o detenerse.
- **cnt_0_9** : Contador de 0 a 9 de 4 bits, a parte de las señales de clock (clk) y reset (rst) la señal “cc” proveniente del filtro le indica si continua o no. Dispara una señal de acarreo cuando llega a 9 para otro contador que la usa como reloj.
- **cnt_0_5** : Similar al contador anterior, solo que cuenta de 0 a 5.
- **vga** : Este módulo es el controlador de la pantalla, trabaja a 25MHz para la resolución de 640x480 pixeles, las señales PIXEL_H y PIXEL_V indican la fila y columna actual del pixel. La señal V_AREA nos indica si el pixel está en posición para poder ser pintado o esta en tiempo de sincronismo.
- **gen_img** : Se encarga de pintar un pixel de la pantalla según su posición. Recibir como señales 6 datos, uno por cada dígito del cronómetro, la posición de pixel (Pix_ver y Pix_hor), si el pixel está en posición para pintarse (V_Area) y colores provenientes de los switch del FPGA.

El componente principal para desplegar los dígitos es “gen_img”, el controlador “vga” nos indica la posición actual del pixel tanto horizontal como verticalmente. Podríamos considerar que el pixel sigue una trayectoria en la pantalla de izquierda a derecha y de arriba a abajo, así que se debe aprovechar el momento en que el pixel pase por la zona de la pantalla que nos interese colorear.

Siguiendo la idea de un display de 7 segmentos, se definen 7 líneas que pueden ser o no dibujadas dependiendo del dato actual. Cada una de estas líneas se dibujan a partir de de una coordenada usada como referencia.

En la figura 3 podemos identificar las variables que representan la posición del pixel actual (Pix_hor y Pix_Ver) y la posición de referencia a partir de la cual nos interesa dibujar (XHora1 y YHora1). Si el pixel se encuentra dentro un área generada a partir de XHora1 y YHora1 el pixel es

coloreado y a su vez la área generada es coloreada. Este fragmento de código es para colorar la línea de representa el segmento “a” de un display, por ello la segunda condición nos verifica que el número actual guardado en la señal “data” utilice dicho segmento.

```
-- a
if( Pix_hor > XHora1 + 35 and Pix_hor < XHora1 + 40 and Pix_ver > YHora1 + 5 and Pix_ver < YHora1 + 35 ) then
  if ( data = 9 or data = 8 or data = 7 or data = 4 or data = 3 or data = 2 or data = 1 or data = 0 ) then
    RGB_sal <= color;
  else
    RGB_sal <= NOT color;
  end if;
elsif ...
```

Fig. 3. Fragmento de código para dibujar la línea del segmento “a” de un display.

Cuando el pixel se encuentra en una zona de la pantalla que nos interesa, actualizamos el valor de nuestra referencia de dibujo y a su vez se actualiza el dato que es dibujado. En este caso son 6 zonas, una por cada dígito del cronómetro. En la figura 4 se muestra el segmento de código que realiza esta acción.

```
if ( Pix_hor > POSX and Pix_hor < POSX + 55 ) then
  XHora1 <= POSX; --115
  data <= data6;
elsif ( Pix_hor > POSX + 55 and Pix_hor < POSX + 110 ) then
  XHora1 <= POSX + 55;
  data <= data5;
elsif ...
```

Fig. 4. Fragmento de código la posición del área dibujable.

Así, por ejemplo, cuando el pixel está en la zona donde nos interesa dibujar el primer dígito de las horas, la referencia en XHora1 es actualiza a la posición en X donde inicia esa área, y la señal data es actualizada con el valor del contador que lleva ese dato, en este caso data6.

IV. Resultados

Resuelto el problema de cómo mostrar los dígitos en la pantalla, la implementación de este sistema no tuvo mayores complicaciones. En las figuras siguientes se pueden observar fotografías del sistema en funcionamiento.

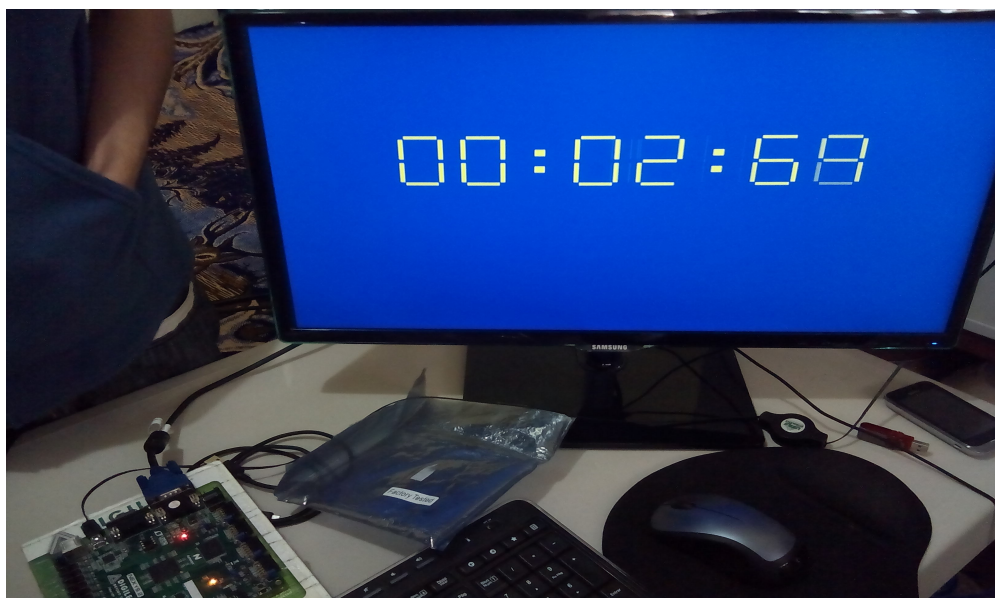


Fig. 5 y 6. Cronómetro en funcionamiento

V. Conclusiones

A lo largo de todo el curso de Electrónica Digital II se aprendió sobre la estructura interna del FPGA y como utilizarla para resolver problemas, en esta última parte se realizaron comunicaciones del FPGA con otros dispositivos externos a través del puerto VGA para la pantalla, PS/2 para el teclado y el mouse, y también por el puerto serial, para transmitir datos entre una computadora y el FPGA. En este proyecto solo usamos el puerto VGA.

Como estudiantes de la carrera de Ing. en Computación, aprendemos muchos lenguajes de programación, sin embargo el paradigma de la mayoría de ellos son similares, quizás VHDL fue un poco enredado al principio, estamos acostumbrados a ver casi todo de una manera casi absolutamente secuencial, e imaginarnos coordinar procesos que se ejecutan de manera paralela y saber que en las asignaciones sólo la última es realizada y las demás son descartadas, hizo que nuestras mentes pensarán de manera distinta a lo común. Además, conocer dispositivos como el FPGA sirvió para saber que no sólo una computadora es utilizada para resolver problemas, y que quizás un dispositivo o un conjunto de ellos programados para resolver un problema lo haga incluso más rápido que una computadora, pues son especializados para ello.