

Osciloscopio digital mediante FPGA

Daniel Fajardo Sebastián, Rubén Ramón Serrano, Felip Vicedo Roman

Dept. Física y Arquitectura de Computadores
Universidad Miguel Hernández
e-mail: danielfajardo@coitt.es, rubenramones@yahoo.es,
felip@umh.es

Resumen

En este artículo se describe la implementación de un Osciloscopio Digital mediante el uso de una FPGA. Ésta se usa para realizar todas las funciones combinatoriales y secuenciales necesarias para capturar la señal, tratarla y visualizarla.

Este Osciloscopio en el que continuamos trabajando es pequeño, fiable, sencillo, práctico y sobre todo barato. Esto es posible utilizando componentes comerciales, una FPGA y usando la pantalla compatible con VGA, ya que la visualización se realizará mediante un monitor externo convencional, como el que usan los PCs.

1. Introducción

Tanto un Ingeniero informático como un Ingeniero de Telecomunicaciones ha de tener conocimientos de Electrónica, pues hoy en día se abren un gran abanico de conocimientos y aplicaciones que directa o indirectamente están basados en ella; desde un programa de ordenador, hasta descolgar un móvil, tienen como base la electrónica.

Teniendo en cuenta la importancia de este área de conocimiento y según lo aprendido durante los años de estudiante, hemos considerado que la herramienta más instructiva para la enseñanza de esta asignatura es el Osciloscopio.

1.1. Motivación y Objetivos

A muchos estudiantes de cualquier carrera técnica me consta que les gustaría disponer de un Osciloscopio en casa, al igual que a muchos aficionados de la electrónica. El motivo por el que no disponemos de él es el precio. Un osciloscopio como los que disponemos en los laboratorios de Electrónica de la UMH puede superar los 1000 euros.

El objetivo principal de nuestro proyecto es conseguir realizar un osciloscopio sencillo, funcional, pequeño, capaz de muestrear a suficiente frecuencia y con la

precisión suficiente para lo requerido en nuestra titulación. Todo ello pensando que el coste del mismo sea inferior a 100 euros.

2. El Osciloscopio Digital

Frente al clásico osciloscopio analógico se impone cada vez más el Osciloscopio Digital hasta tal punto que ya prácticamente no se utilizan los analógicos. La principal razón de este fenómeno es la extraordinaria evolución de la Electrónica Digital. Cada vez los fabricantes de chips Digitales consiguen realizar chips más pequeños, con mayor número de transistores, capaces de funcionar cada vez a más velocidad y a menor coste.

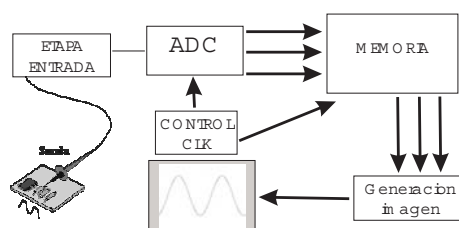


Figura 1: Esquema Osciloscopio Digital

El funcionamiento básico del Osciloscopio digital podemos verlo en la figura 1. Dispone, como cualquier equipo de instrumentación, de una etapa de Acondicionamiento de Señal. El convertidor A/D es el encargado de muestrear la señal analógica y de la cuantificación de su valor. Esta parte del Osciloscopio es especialmente crítica, ya que en función de la frecuencia de muestreo y el nivel de precisión del conversor (niveles de cuantificación), definirá las principales características del osciloscopio. Una vez muestreados se almacenan los valores en una memoria digital para su posterior procesado y visualización.

Los Osciloscopios Digitales, gracias a la lógica digital, pueden realizar cálculos complejos sobre los datos almacenados en la memoria. Esta es una gran ventaja de los Osciloscopios Digitales frente a los Analógicos; el procesado de los datos. No obstante, la principal virtud frente al clásico Osciloscopio se presenta a la hora de visualizar señales no repetitivas o aleatorias. Los Osciloscopios clásicos se basan en entradas periódicas para generar señales en diente de sierra, imprescindibles para la correcta visualización. En cambio en los digitales esto no ocurre. Lo que le proporciona la capacidad de visualizar cualquier tipo de entrada tanto, periódica como aleatoria.

3. Lógica Digital

En este apartado explicaremos las herramientas utilizadas para la implementación de la Lógica Digital necesaria para la fabricación de nuestro Osciloscopio

[3].

3.1. Todo en un solo chip (SoC: System on Chip)

Para nuestro diseño lógico como vemos en la figura 1, necesitamos implementar principalmente los siguientes bloques:

- Memoria: guardará los datos adquiridos por el conversor A/D.
- Módulo Trigger: Controla el comienzo de la adquisición de datos, según la pendiente y el nivel de la señal de entrada.
- Módulo de control e interfaz: Se encarga de la base de tiempos, (período de muestreo) y controla la interfaz de entrada (el ratón).
- Generador VGA: Se encarga de mostrar la señal por pantalla.

Cuando nos paramos a pensar los componentes que necesitamos para implementar todos estos módulos nos daríamos cuenta enseguida de la complejidad del proyecto.

Por este motivo hemos decidido implementar toda la lógica digital dentro de una FPGA. Con lo que conseguimos:

- Implementar toda la lógica en un sólo chip (SoC).
- Podemos diseñar nuestros módulos a medida, sin tener que restringirnos a los existentes en el mercado. Nosotros somos los responsables del funcionamiento de cada uno y podemos implementarlos tal y como necesitamos.
- Diseño flexible, ampliable y modificable. Tán sólo con reprogramar la FPGA podemos cambiar cualquier aspecto de nuestro diseño, tanto corregir errores como ampliar el diseño.
- Reducir Costes. Resulta más económico el desembolso de una única FPGA que el de muchos componentes que hagan la misma función.
- Mayor Productividad. Es más sencillo y rápido el diseño sobre nuestro chip que el diseño con dispositivos de diferentes fabricantes y con características que seguramente no sean exactamente las deseadas.
- Reducción del tamaño. La placa, donde están todos los chips necesarios será más reducida.

3.2. ¿Que es una FPGA?

Una FPGA (Field Programmable Gate Array) traducido literalmente es un array de puertas programables. Veamos en que consiste este array de puertas.

Una FPGA está organizada en:

- CLBs (Bloques Lógicos Configurables): Un array de celdas regularmente dispuestas sobre el chip, estos CLBs son programables.

- IOBs (Input Output Block): Es la conexión programable con los pines de entrada/salida dispuestos alrededor del chip.
- PSM (Programmable Switch Matrix): Bloques de interconexión, que bajo programación permiten conectar los CLB's entre sí o con IOBs.

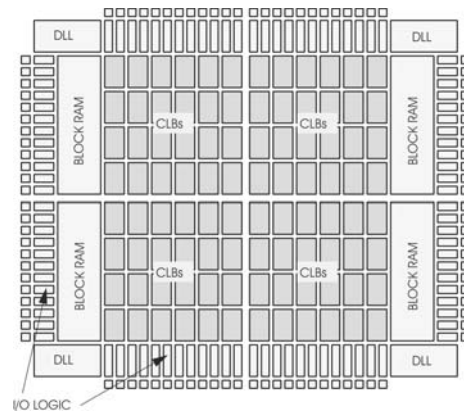


Figura 2: Composición de una FPGA: CLB's y IOBs

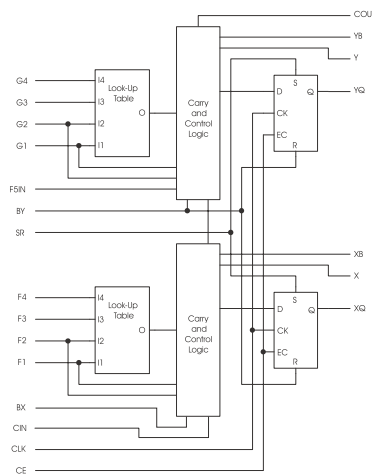


Figura 3: Composición interna de un Slice

Cada CLB está compuesto por celdas más pequeñas programables llamadas Slices. Cada Slice es capaz de contener cualquier función lógica digital de 5 variables, además también dispone de dos Flip-Flop tipo D.

2. Implementación: En esta fase el software implementa el archivo anterior para la FPGA o dispositivo lógico seleccionado. Particulariza el diseño configurando adecuadamente las CLBs, IOBs y PSMs.
3. Generación del *bitstream*. El *bitstream* es el archivo programable en la FPGA. Incluye las ordenes necesarias para hacer saber a la FPGA la configuración calculada en la fase anterior.

4. Implementación del osciloscopio

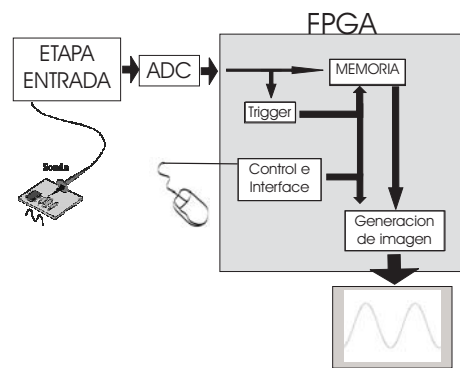


Figura 5: Diagrama de bloques de nuestro Osciloscopio

4.1. Material utilizado

Para el desarrollo del diseño del Osciloscopio Digital hemos usado los siguientes elementos:

- Conversor ADS831-E [5]. Este es un conversor que trabaja por aproximaciones sucesivas, proporciona a la salida ocho bits, con codificación binaria o en complemento a dos, según las necesidades. Lo comercializa la empresa Burr-Brown perteneciente a Texas Instrument, en un encapsulado SOIC.
- Filtro activo paso bajo, implementado con el THS4061.
- Limitador de tensión de entrada con la utilización de diodos zener.
- Fijador de tensión utilizando dos amplificadores operacionales 741.
- Selector de escalas, utilizando resistencias de diferentes valores, el THS4061 y un multiplexor analógico (AD8174) de gran ancho de banda.
- Xess XSA-100. Placa de desarrollo comercializada por Xess en la cual disponemos de:

- Xilinx Spartan II FPGA de 100 mil puertas lógicas equivalentes.
- Interfaz VGA. Dispone conector VGA y de conversor DAC de 2 bits por color (6 bits = 64 colores) mediante resistencias ponderadas.
- Conector PS/2. Mediante el cual se puede conectar un teclado o ratón (en nuestro caso un ratón).
- Reloj programable hasta 100MHz.
- Otros componentes que no han sido necesarios utilizar: 16 Mbyte SDRAM, 256 KByte Flash, 7-segment LED.

4.2. Tratamiento de señales

Un osciloscopio es un instrumento de visualización de señales. Puede mostrar cualquier tipo de señal, ya sean sísmicas, latidos del corazón, ondas electromagnéticas etc... siempre y cuando se le haya dotado de un transductor adecuado, dado que sólo es capaz de visualizar señales eléctricas. Las mediciones que un osciloscopio tiene que hacer han de ser muy precisas y es aquí donde el sistema de adquisición de datos tiene gran importancia, ya que cualquier error al principio del sistema provocaría un resultado que distaría mucho de la realidad.

Nuestro sistema acondicionador de señal tiene las siguientes características:

1. Permite que la señal tenga componente continua o no (acoplamiento CC y CA).
2. Tiene un selector de escala.

Para la implementación de la etapa de adquisición de datos se ha necesitado:

1. Un circuito fijador de nivel de tensión para que la señal entre correctamente al conversor analógico digital.
2. Un conversor analógico digital.
3. Todavía no se le ha acoplado un filtro para evitar que se produzca Aliasing, por lo que la señal de entrada no he de tener componente frecuencial por encima de $2*fs$.

4.3. Los diferentes circuitos utilizados

En primer lugar lo que hacemos es limitar el voltaje pico a pico máximo de la señal de entrada (en nuestro caso a 20 Vpp), para evitar posibles saturaciones en alguno de los componentes que la etapa utiliza mas tarde.

La señal debe pasar por un filtro que limite el ancho de banda total con el que podamos trabajar, de forma que no aparezca el denominado efecto Aliasing. Teóricamente el ancho de banda máximo en el que nos podemos mover viene determinado por el ADS831-E (conversor analógico-digital) que, por el criterio de Nyquist, nos fija el máximo a 40 MHz.

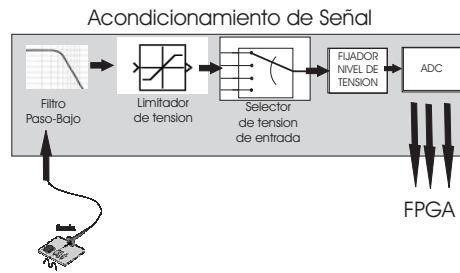


Figura 6: grafico2

Toda etapa de entrada de un osciloscopio debe permitir que el usuario pueda escoger entre acoplamiento en continua o alterna (CC ó CA), también debe poder cambiar el fondo de escala según las necesidades de la señal que quiera ver. Nuestro osciloscopio cumple tanto con el primer como con el segundo requisito (el usuario lo maneja todo desde el monitor a través del ratón). Se hace necesaria la utilización de un fijador de nivel para añadir un offset a la señal, que permita el correcto funcionamiento del convertor analógico digital, ya que este debe trabajar entre 1.5V y 3.5V.

El convertor que hemos utilizado (ADS831-E)[5], fue elegido principalmente porque su perfil se adaptaba bastante a lo que nosotros buscábamos:

1. Módico precio.
2. Gran ancho de banda.

Este es un convertor que trabaja por aproximaciones sucesivas. Lo comercializa la empresa Burr-Brown perteneciente a Texas Instrument, en un encapsulado SOIC.

4.4. Memoria Digital

La memoria de nuestro osciloscopio es uno de los puntos más críticos del proyecto. Esta memoria ha de ser capaz de escribir en ella los datos provenientes del convertor y de leer los datos para su visualización.

La mayoría de aplicaciones que han de mostrar imagen por pantalla suelen guardar la imagen a visualizar en una memoria que contiene la información del color (RGB) que tiene cada pixel. En nuestro caso podemos generar 64 colores ya que trabajamos con 6 bits digitales de color (2 bits por color). Si siguiéramos estas pautas en la implementación de la memoria para una pantalla VGA (480x640 pixels) necesitaríamos una memoria:

$$480 \times 640 \times 6 \frac{\text{bits RGB}}{\text{pixel}} = 1843200 \text{ bits} \approx 230 \text{ kbytes}$$

En cambio hemos decidido implementar esta memoria como si de un vector se tratara. Guardamos sólo los valores muestreados por el convertor A/D y los vamos guardando consecutivamente. De esta manera si guardamos un valor

de 8 bits (conversor A/D de 8 bits) para cada columna de nuestra pantalla VGA (640x480):

$$640 \times 8 \text{ bits} = 5120 \text{ bits} = 640 \text{ bytes}$$

La implementación de la misma se ha realizado mediante la herramienta *Xilinx CORE Generator* que nos ha generado una memoria de doble puerto sintetizada y optimizada por Xilinx para sus FPGAs.

A parte de la memoria en sí, hemos programado un módulo que gestiona la lectura y escritura de la misma. La escritura se realiza consecutivamente desde la dirección 0 hasta la 640 cuando el control del trigger nos lo indica. Esto hace que por nuestra VGA sea tan sencillo leer todos los datos que contiene cada vez que dibujamos una de las 480 líneas verticales que tiene nuestra pantalla.

4.5. Visualización

Tal y como hemos explicado anteriormente el principal objetivo del proyecto es el de conseguir un osciloscopio a bajo coste. Por ese motivo hemos pensado que la mejor opción es que la visualización se haga mediante un monitor VGA porque:

- Abaratar costes. Si no integramos una pantalla en el sistema el precio se reduce drásticamente.
- Pantallas VGA muy extendidas. Los PCs por ejemplo utilizan pantallas compatibles con VGA. Nuestro osciloscopio podrá ser visualizado en cualquiera de estas pantallas.
- Fácil implementación. Gracias a la potencia de las FPGAs la implementación de un módulo que genere las señales VGA es relativamente sencillo.

Las señales VGA se dividen en dos grupos:

- De sincronismo. Sincronismo horizontal y vertical. Indican a la pantalla la frecuencia a la que tienen que realizar el refresco de cada línea vertical o de cada imagen.
- RGB. Indican, mediante 3 señales analógicas, la intensidad del componente de Rojo, Verde y Azul de cada pixel. Para generarla se utiliza un conversor A/D de 2 bits por color, basado en una red de resistencias ponderada, lo cual nos proporciona un total de 64 colores.

4.6. Control Escalas

Cuando visualizamos una señal en la pantalla de un Osciloscopio, debemos ajustar la escala vertical y horizontal. También podemos querer desplazarla vertical u horizontalmente. Veamos a continuación cómo realizamos esto en nuestro diseño.

Cuando queremos ensanchar o comprimir la visualización de una señal, lo que hacemos es cambiar la escala del eje horizontal, para ello un osciloscopio

lo que hace es mostrar los puntos más dispersos o más juntos en la pantalla. Esto se realiza mediante un proceso denominado diezmado, para poder realizar este proceso además del mismo hay que implementar un filtro digital antes del diezmado.

Para controlar el eje de escala vertical lo podemos hacer por hardware o por software. El proceso por software es parecido al descrito anteriormente de la escala de tiempos. Consiste en hacer un "zoom" sobre el eje vertical. Lo único que tenemos que hacer es mostrar los puntos con más o menos separación vertical según la escala.

En cambio, por hardware estamos implementado un selector de escalas usando el amplificador operacional THS4061, fabricado por Texas Instrument que tiene un gran ancho de banda y se caracteriza por su gran rapidez, (también en encapsulado SOIC). Utilizando este integrado a modo de amplificador no inversor y usando una serie de resistencias de baja tolerancia como divisores de tensión conseguimos amplificar o atenuar la señal de manera directa, dicha señal será conectada a la entrada de un multiplexor analógico (AD8174) y mediante los impulsos eléctricos que envíe la FPGA se seleccionará la entrada óptima para poder ser visualizada por pantalla.

5. Resultados

Hemos conseguido implementar un osciloscopio digital de bajo coste, para visualizar las señales se usa una pantalla de ordenador y un ratón para controlar el funcionamiento completo del osciloscopio (trigger, tipo de acoplamiento, escalas etc...). El circuito se podría implementar en una placa de 10*10 cm. por lo que es un sistema perfectamente portable a cualquier lugar.

Referencias

- [1] Aldec
www.aldec.com
- [2] Lluís Terés, Yago Torroja, Serafín Olcoz, Eugenio Villar , *VHDL Lenguaje Estándar de Diseño Electrónico*, McGraw-Hill 1998. *Amplificadores Operacionales y Circuitos Integrados Lineales* James M. Fiore
- [3] René Rateau, *Osciloscopios funcionamiento y Utilización*, Paraninfo 1999.
- [4] Serafín Alfonso Pérez, Enrique Soto, Santiago Fernández, *Diseño de Sistemas Digitales con VHDL*, Thompson 2002.
- [5] Texas Instruments ADS831
<http://focus.ti.com/docs/prod/folders/print/ads831.html>
- [6] Xilinx
www.xilinx.com