

# **Analizador Lógico de 100MHz utilizando FPGA**

Oscar de Oliveira Madeira, Gabriel Eirea, Juan Pablo Oliver, Julio Pérez Aclé.

Instituto de Ingeniería Eléctrica  
Universidad de la República  
Casilla de Correos 30  
Montevideo, Uruguay.

Tel: (598 2) 711 0974 - Fax: (598 2) 711 7435

E-mail: oscar@iie.edu.uy, geirea@iie.edu.uy, jpo@iie.edu.uy, julio@iie.edu.uy

## **Abstract**

This paper describes the design of a logic analyzer with 48 channels, 16K samples per channel and up to 100MHz of acquisition frequency. The user interface is implemented with a computer connected via a serial or parallel port.

An FPGA is used for the sample acquisition control. This module controls the trigger condition detection, the FIFO memory used for sample storage, and the connection with a microcontroller which performs the communications with the computer. The selected chip was the EPF6016 of the new Altera FLEX6000 family, and the development software used was Max+Plus II from the same company.

The architecture, the design criteria used and the obtained results are described.

## **Resumen**

En este trabajo se describe el diseño de un analizador lógico con 48 canales, 16K muestras por canal y que puede operar con frecuencias de adquisición de hasta 100MHz. La interfaz de usuario está realizada sobre un computador que se comunica en forma serie o paralelo con el analizador.

Para el control de la adquisición de las muestras, se utiliza una FPGA, que se encarga de la detección de la condición de disparo, del manejo de las memorias FIFO que se usan para el almacenamiento de las muestras, y de la interconexión con el microcontrolador que realiza la comunicación con el computador. El chip elegido fue el EPF6016 de la nueva familia FLEX6000 de Altera, y el sistema de desarrollo utilizado fue el Max+Plus II de la misma compañía.

Se detallan la arquitectura y los criterios de diseño utilizados así como los resultados obtenidos.

## **Introducción**

El presente trabajo se enmarca dentro de un proyecto conjunto entre el Instituto de Ingeniería Eléctrica de la Universidad de la República y la empresa IMAT, con el objetivo de desarrollar un analizador lógico de bajo costo, para ser utilizado en diversas aplicaciones, por ejemplo en laboratorios de enseñanza.

Dicho analizador consta de una placa que adquiere los datos y que está conectada a un computador en forma serial o paralela, el cual se encargará del procesamiento y despliegue de los datos adquiridos, así como de la interfaz con el usuario. Si bien en el resto del trabajo nos referiremos al computador como un PC, el mismo puede ser cualquier computador con interfaz serie o paralelo.

Las especificaciones iniciales del proyecto eran las siguientes: 48 canales, frecuencia de adquisición programable hasta 100MHz, más de 8K muestras y pretrigger programable.

En este trabajo se describe el diseño de la placa de adquisición de señales, y en particular el controlador de adquisición que fue desarrollado en una FPGA por los autores del trabajo, dentro de las actividades del Grupo de Electrónica Aplicada del Instituto de Ingeniería Eléctrica.

## Descripción general

El funcionamiento del analizador es el clásico: el usuario programa una condición de disparo, la profundidad del pretrigger y la frecuencia de muestreo. Una vez que la condición de disparo se satisface se completa la adquisición y se transfieren los datos al computador para su análisis y visualización.

El diagrama de bloques del analizador lógico completo se muestra en la Figura 1. Si bien se hará una breve descripción de cada uno de los bloques nos concentraremos en la etapa de control y en la selección del tipo de memorias a utilizar.

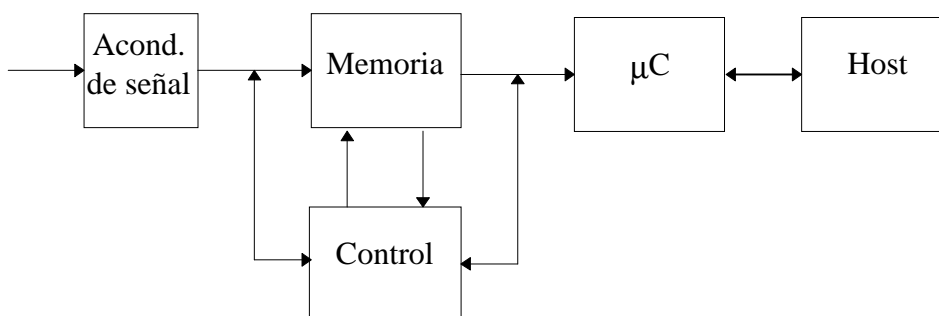


Figura 1 - Diagrama en bloques del analizador lógico.

Como puede apreciarse en dicho diagrama, el analizador está formado por los siguientes bloques:

- Acondicionamiento de señales
- Memoria para almacenar los datos adquiridos
- Lógica de Control
- Comunicaciones

La etapa de acondicionamiento de señales se resolvió utilizando buffers triestado CMOS rápidos, con el fin de detectar los niveles de la señal de entrada y no cargar el circuito a testear. No fue parte de este proyecto el estudio de los cables a utilizar así como la adaptación de las señales de entrada.

Para la memoria de datos se plantearon dos posibles soluciones: usar memorias SRAM o memorias FIFO. La primer solución implica que el controlador de adquisición debería generar el direccionamiento de la memoria SRAM, lo cual se hace muy difícil trabajando a 100MHz. El hecho de que las memorias FIFO tengan los punteros de escritura y lectura generados internamente, y que sólo se deba manejar las señales de control de escritura y lectura, soluciona el problema antes mencionado. Para poder brindar la prestación de tener un pretrigger programable, se decidió el uso de FIFOs con flags programables. Teniendo en cuenta estos aspectos, se eligió utilizar las FIFOs CY7C4261-10 de Cypress, que funcionan en forma síncrona con un reloj de hasta 100MHz, poseen una capacidad de 16Kx9 y tienen dos flags programables (Almost Empty y Almost Full).

El bloque de control se encarga de brindar el reloj de muestreo, detectar la condición de disparo, controlar la escritura y lectura de los datos en las FIFOs, programar las banderas de las FIFOs e interactuar con el microcontrolador. Este bloque está realizado sobre una FPGA y se analizará en detalle en el apartado siguiente.

La comunicación entre la placa adquisidora de datos y el PC se realiza a través de un µC en forma serie o paralela. Este bloque se encarga de realizar la transferencia de los datos adquiridos de la memoria FIFO al PC, y programar los diferentes parámetros de adquisición desde el PC al FPGA. Si bien esta comunicación podría ser controlada por lógica dentro de la FPGA, se decidió usar el µC debido a que ya se tenía desarrollado el software de comunicación, tanto del lado del µC como del PC, por lo que se ahorra mucho tiempo en el desarrollo total. La versión actual del software está desarrollada bajo LINUX, pero puede migrarse a Windows, o incluso a cualquier otro host que admita una comunicación serie o paralela. El PC se encarga del procesamiento de los datos, su despliegue y toda la interfaz con el usuario.

Un esquema con los componentes y las señales de la placa adquisidora se muestra en la Figura 2.

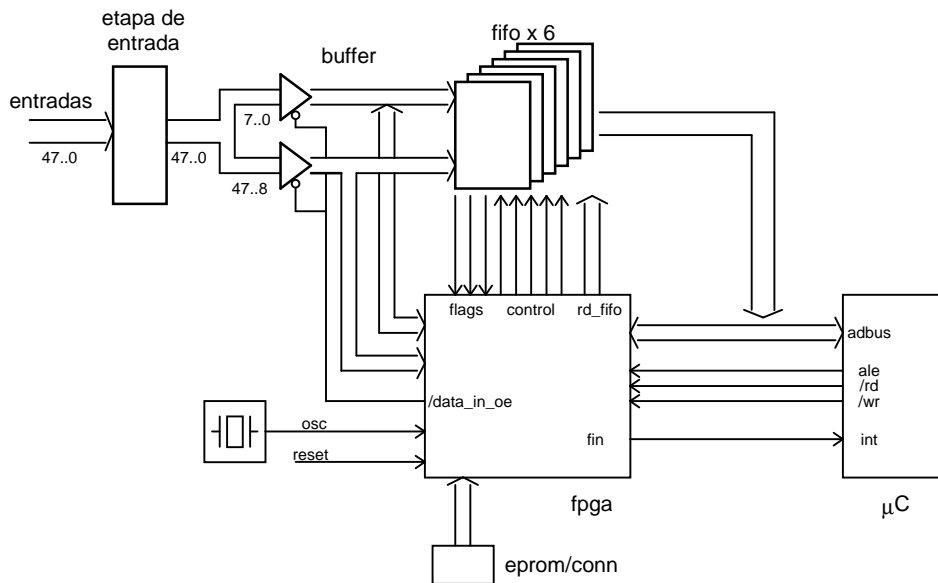


Figura 2 - Esquema de la placa adquisidora de datos.

### Control de adquisición

El control de adquisición es la parte más importante de la placa de este analizador, y está diseñada sobre una FPGA EPF6016TC144-2 de ALTERA. Sus funciones son: detectar la condición de disparo, generar la señal de reloj para la adquisición, controlar las memorias FIFO para el almacenamiento de los datos y la programación de los flags, y realizar la interconexión con el microcontrolador para la programación de los parámetros de adquisición elegidos por el usuario y la lectura de los datos una vez finalizada la adquisición.

Su diseño se realizó en forma modular para facilitar las posibles mejoras a ser realizadas en el futuro. En la Figura 3 se puede ver el diagrama en bloques. A continuación describiremos brevemente los criterios utilizados para la elección del chip y el diseño de cada uno de los bloques.

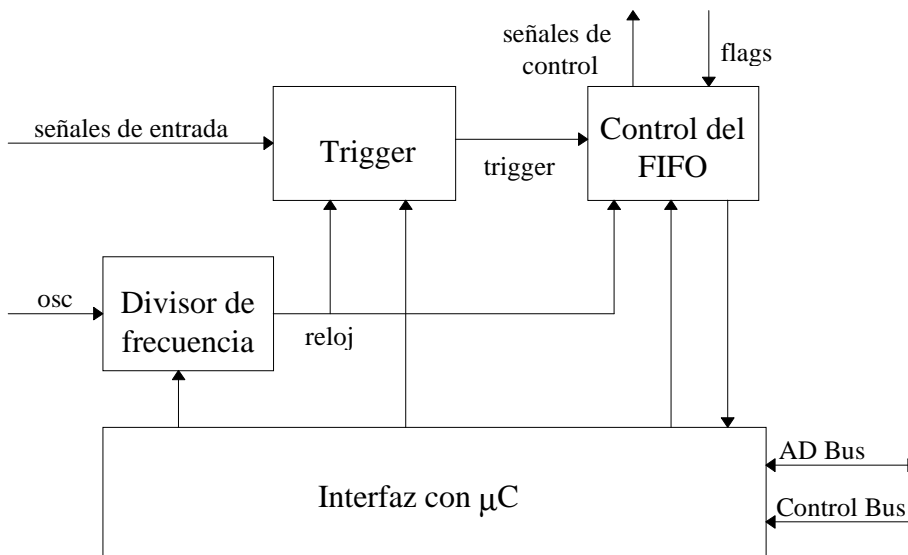


Figura 3 - Diagrama en bloques del control de adquisición.

## Elección del chip

La elección del chip a utilizar en el diseño, surgió del análisis de varios factores. Siempre se buscó la solución final dentro de los chips de ALTERA, debido a que se dispone del software de desarrollo y experiencia en su empleo.

Debido a la cantidad de registros a usar y a la frecuencia de funcionamiento, la elección final debería estar en las familias basadas en tecnología RAM (FLEX 8000, 10K y 6000). La frecuencia de funcionamiento deseada, determinó también que se debería trabajar con los integrados de más alta velocidad (-2), los cuales, al momento de comenzar el proyecto solo se encontraban disponibles en la familia 8000 (dentro de la familia 10K se hicieron disponibles en el cuarto trimestre del año y a un costo muy alto).

Con el surgimiento de la familia FLEX6000, se volvió a analizar la mejor solución, decidiéndonos por la EPF6016TC144-2, que cumple con las especificaciones requeridas, a un precio mucho menor que las otras familias. Con este chip, queda suficiente capacidad disponible como para las futuras mejoras previstas.

## Trigger

El bloque Trigger es el encargado de detectar la condición de disparo del analizador. Dicha condición de disparo es programable. Para esto, se dispone de dos registros de 48 bits: Registro de Datos y Registro de Máscaras. Ambos registros se programan desde el microcontrolador escribiendo en ellos como si fueran seis registros de 8 bits. Se dispone además de un Registro de Modo, programable por el microcontrolador, y que determina el modo de disparo. En esta primer versión están disponibles solamente dos modos:

- **MODO 0** - se dispara cuando los datos de entrada son iguales al programado en el Registro de Datos
- **MODO 1** - se dispara cuando son iguales los bits no enmascarados de los datos de entrada y del Registro de Datos. Los bits enmascarados se determinan por el contenido del Registro de Máscaras.

Para el diseño del comparador de 48 bits, se decidió el uso de una estructura pipeline, que garantizara un retardo determinístico desde que se establece la condición de disparo hasta que se genera la señal *trigger*. La estructura elegida se muestra en la Figura 4. Como se puede apreciar en ella, los datos de entrada se almacenan en un registro y a continuación se opera con los datos en forma sincronizada en 4 niveles. De esta forma se garantiza que la señal *trigger* se activa siempre 5 períodos de reloj luego de establecida la condición de disparo en la entrada.

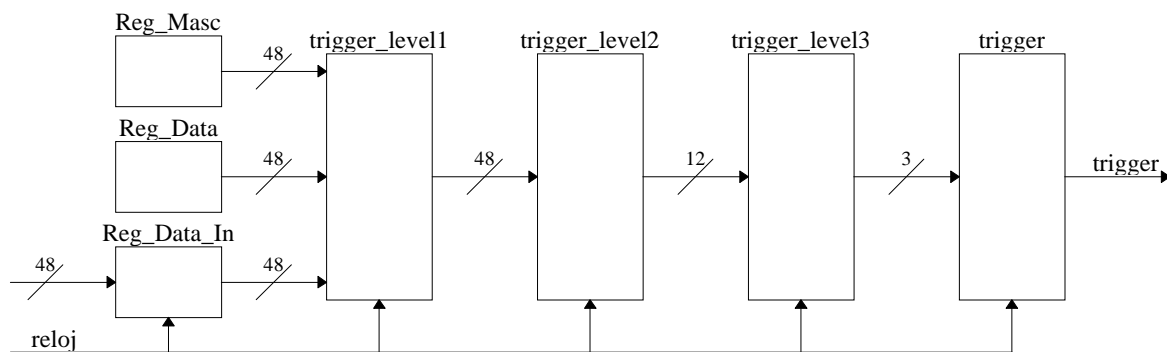


Figura 4 - Estructura pipeline del comparador de datos.

## Control de las memorias FIFO

Este bloque tiene la función de generar las señales de control de las memorias FIFO, para manejar la adquisición de datos y su posterior lectura desde el  $\mu$ C.

Para su comunicación con el  $\mu$ C posee un Registro de Comandos y un Registro de Status. Para la funcionalidad de pretrigger programable posee un Registro de Pretrigger, que puede ser escrito desde el  $\mu$ C.

Los comandos que puede recibir desde el  $\mu$ C son los siguientes:

- **Reset:** inicializa todas las máquinas de estado.

- **Reset FIFO:** inicializa las FIFOs.
- **Armar:** inicia una adquisición.
- **Leer dato:** avanza el puntero de lectura de las FIFOs, para que se pueda leer la próxima muestra.

El control de las FIFOs se realiza mediante las siguientes cuatro máquinas de estado:

- **adquirir:** a la llegada de un comando Armar, comienza con la adquisición. Tiene como entradas la señal proveniente del bloque trigger, y todos los flags de las FIFOs. Como salida tiene las señales de control de las FIFOs y una señal *fin* que se usa para interrumpir al  $\mu$ C cuando se terminó la adquisición.
- **resetear FIFOs:** se encarga de inicializar las FIFOs cuando se recibe un comando Reset FIFO.
- **programar pretrigger:** se encarga de programar el flag de las FIFOs que se usa para detectar la condición de pretrigger. Esta acción se ejecuta automáticamente cuando se realiza una escritura sobre el Registro de Pretrigger.
- **leer dato:** avanza el puntero de lectura de las FIFOs a la llegada del comando Leer dato, permitiendo que el  $\mu$ C pueda leer la siguiente muestra.

## Divisor de frecuencia

Este bloque es el que se encarga de generar la señal de reloj del sistema, a partir de un oscilador de 100MHz. La frecuencia es programable para poder obtener diferentes frecuencias de muestreo, pudiendo de esta manera tener ventanas con diferentes tiempos de muestreo.

El bloque divisor de frecuencia consta de divisores por dos y por cinco, obteniendo así una señal reloj de frecuencia 100, 50, 25, 10, 5, 2.5, 1MHz y 500, 250, 100 KHz. Los divisores por dos se hicieron con flip-flops tipo T (ripple counters), mientras que los divisores por cinco se hicieron con un contador de anillo de forma de minimizar la lógica y adaptarse mejor a la arquitectura de la FPGA elegida.

## Interfaz con el $\mu$ C

La interconexión entre el microcontrolador y el control de adquisición se realiza a través de este bloque. El microcontrolador ve al control de adquisición como un dispositivo mapeado en memoria externa, sobre el cual puede leer o escribir. Cada uno de los registros de configuración del control de adquisición puede ser escrito directamente por el  $\mu$ C como una dirección de memoria. De la misma manera, el  $\mu$ C puede leer el Registro de Status. La lectura de los datos adquiridos también se realiza como una lectura de memoria, para lo cual se tiene asociadas seis direcciones consecutivas (una para cada memoria FIFO).

La Interfaz con el  $\mu$ C recibe como entradas el bus de direcciones y datos multiplexado, las señales /WR, /RD y ALE, y a partir de ellas demultiplexa el bus, decodifica las diferentes direcciones y hace llegar los datos y las señales de escritura (o lectura) a cada uno de los registros.

Mientras se están adquiriendo los datos, están deshabilitadas todas las escrituras sobre los registros de configuración y las lecturas de los datos almacenados en las FIFO. Sí se puede leer el Registro de Status a los efectos de que el  $\mu$ C pueda saber en que estado está la adquisición.

## Resultados obtenidos

El analizador lógico diseñado tiene las siguientes prestaciones:

- Canales: 48
- Frecuencia de muestreo: 100, 50, 25, 10, 5, 2.5, 1MHz y 500, 250, 100 KHz.
- Capacidad de muestras: 16K.
- Ventanas de muestreo: desde 163.84 $\mu$ s hasta 163.84ms.
- Pretrigger: programable en pasos de 64 muestras.

Se han realizado extensas simulaciones sobre el bloque de control de adquisición. A la fecha se está desarrollando el circuito impreso y el resto del sistema. Se espera realizar las pruebas sobre el primer prototipo en los próximos meses.

Las simulaciones hechas, probaron el correcto funcionamiento del control de adquisición, tanto a nivel lógico como a nivel funcional trabajando a 100MHz. Los estudios de tiempos realizados estiman la frecuencia máxima de funcionamiento en 115MHz.

Respecto a la capacidad del chip elegido, el diseño actual utiliza el 29% de las celdas lógicas y el 66% de los pines de I/O. La capacidad ociosa del chip era uno de los parámetros manejados en el momento de su elección, ya que nos permitirá incorporar las mejoras expuestas en el próximo apartado.

En cuanto al costo de la placa diseñada, la mayor parte se debe a la memoria necesaria para la adquisición. El uso de memorias FIFO en el presente diseño, permite trabajar con memorias de diferente capacidad (desde 64 hasta 32K muestras) y diferente velocidad (30 a 100MHz), sin modificación alguna en la placa, debido a que el encapsulado y conexionado siempre es el mismo. Esta flexibilidad, permite tener la misma placa a un costo mucho menor en aplicaciones donde los requerimientos en velocidad y profundidad de memoria sean menores.

### ***Futuras mejoras***

En un futuro próximo, se prevé realizar algunas mejoras al diseño y dotar al sistema de algunas prestaciones adicionales, como ser:

- Cantidad de muestras programable en pasos de 64 muestras
- Incorporar más modos de trigger, por ejemplo flanco de determinada señal, combinación de eventos, etc.
- Generación de señales de estímulo para el circuito bajo prueba en forma coordinada con la adquisición.
- Eliminar el  $\mu C$  y hacer que la FPGA se encargue de la comunicación con el PC.

Cabe destacar, que las dos primeras prestaciones detalladas, se pueden incorporar sin realizar ningún cambio en el hardware de la placa, simplemente modificando el diseño de la lógica de la FPGA.

### ***Conclusiones***

En este trabajo se presentó el diseño e implementación de una placa adquirente de datos, para ser utilizada como parte de un analizador lógico que funciona conjuntamente con un computador que realiza la interfaz con el usuario.

La solución elegida para su diseño, brinda una gran flexibilidad en cuanto a prestaciones, velocidad y capacidad de muestras, lo que permite tener un analizador de muy bajo costo que lo hace muy interesante en diversas aplicaciones, especialmente en laboratorios de enseñanza.

La elección del uso de lógica programable nos permitió el desarrollo en un tiempo muy corto y la posibilidad de implementar mejoras en el futuro.

### ***Agradecimientos***

Los autores del presente trabajo agradecen a la empresa IMAT, y en especial a su director José María Acosta, por su iniciativa para realizar este desarrollo en forma conjunta.

### ***Referencias y Bibliografía***

1997 Altera Digital Library  
1995 Cypress Data Book