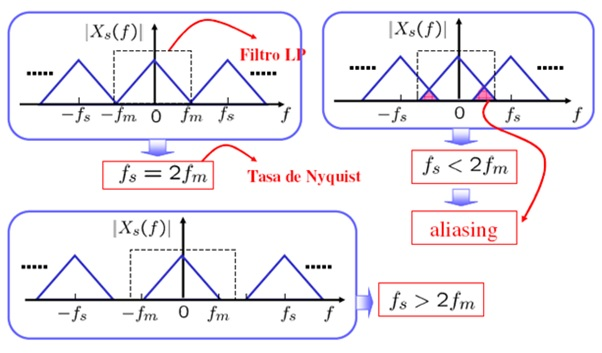
Introducción

Cuando nos encontramos trabajando con FPGAs, que son dispositivos generalmente digitales que cuentan con una gran capacidad de procesamiento debido a su forma de operación concurrente, podríamos vernos en la situación de querer adquirir señales analógicas por algún motivo. Hay algunos chips en el mercado que tienen implementadas entradas analógicas, pero las generaciones más viejas de FPGAs, cuentan únicamente con entradas y salidas digitales. Con la modulación Sigma Delta, podemos convertir estos pines digitales en entradas analógicas, utilizando muy pocos componentes externos a la FPGA. El objetivo del presente trabajo es introducir los conceptos básicos de funcionamiento, las ventajas de esta técnica, la implementación en múltiples variantes, y resultados tanto de simulación como experimentales.

Muestreo

En el proceso de muestreo, la señal continua es muestreada de forma espaciada uniformemente en intervalos de tiempo Ts. Las muestras x[n], de la señal x(t) pueden ser representadas como x[n]=x(n\*Ts). El efecto del muestreo en el dominio de la frecuencia, es replicar el espectro de la señal x(t) en los múltiplos de la frecuencia de muestreo fs=1/Ts.



El proceso de muestreo se puede observar en la figura para 3 casos. El primer caso la frecuencia de muestreo es la frecuencia de Nyquist donde fs=2fm. El segundo caso la frecuencia de muestreo es menor a 2fm, en este caso se produce un solapamiento del espectro que impide la reconstrucción de la señal, este fenómeno se llama aliasing. En el tercer caso, la frecuencia de muestreo fs es mayor a 2fm, en este caso donde la frecuencia de muestreo es mayor que la de Nyquist es conocido como sobremuestreo.

Cuantificación

Como la señal está siendo convertida al dominio digital, esta no puede ser representada con infinita precisión. El valor analógico de la señal debe ser representado por un número finito de niveles preestablecidos, que luego son codificados a palabras binarias cuyo largo determina la cantidad de bits del ADC, este proceso se llama cuantificación y es un proceso no reversible.

El proceso de cuantificación conlleva un problema importante que es la adición de un error o ruido de cuantificación, este error toma valores entre Q/2 y –Q/2 , donde Q es la altura del escalón de cuantificación. El valor de Q depende del rango dinámico de nuestro ADC y de la cantidad de bits que utilizamos para cuantificar que es n.

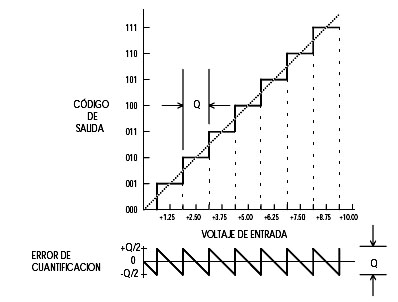
Relación señal a ruido

Bajo ciertas condiciones, podemos modelizar la cuantificación como un proceso lineal de adición de ruido blanco cuya función de probabilidad es

Podemos calcular la potencia de ruido calculando la varianza de p(e)

Entonces la relación señal a ruido está dada por:

La ecuación anterior nos indica que por cada bit extra de resolución incrementamos la relación señal a ruido en aproximadamente 6dB. Si fuera el caso de que la señal de entrada fuera un tono puro, Ps seria V^2/2, y suponiendo que el rango dinámico del adc es simétrico e igual a 2V la relación señal a ruido en ese caso seria

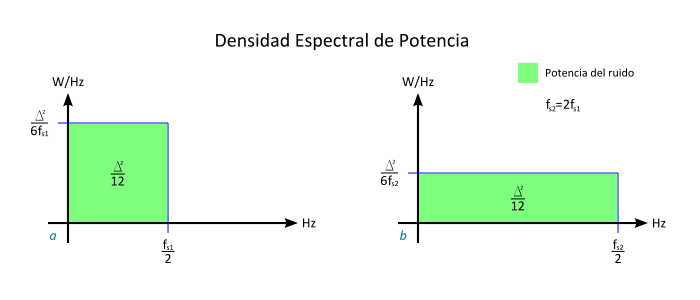


Sobremuestreo

En el proceso de muestreo, el teorema de Nyquist dice que para que una señal pueda ser reconstruida, es necesario que sea mayor a dos veces el ancho de banda de la señal a muestrear. Generalmente son muestreadas a una señal mucho mayor que la frecuencia de Nyquist. La relación entre la frecuencia de muestre real y la frecuencia de Nyquist se llama oversampling ratio (OSR) y es definido como:

El proceso de sobremuestreo mejora la relación señal a ruido, tomando muestras más rápido que la frecuencia de Nyquist. Lo importante de esto es que la cantidad de ruido que estamos inyectando en el proceso de cuantificación es exactamente la misma que la que inyectaríamos muestreando a la frecuencia de Nyquist, pero su distribución es diferente. Si continuamos modelando el ruido como ruido blanco y este está equitativamente distribuido en el espectro, la densidad espectral de potencia está dada por , realizando la integral en la banda de interés obtenemos que

Si tomo que OSR es una potencia de 2 (OSR = 2^r) obtengo que hay un aumento de 3 dB por cada vez que duplicamos la frecuencia de muestreo tomando como base la frecuencia de Nyquist.



Numero efectivo de Bits

La resolución de un ADC esta especificada por el número de bits de la palabra digital que representa al valor analógico, generalmente niveles para un adc de N bits. Sin embargo, todos los ADC reales introducen ruido y distorsión. El ENOB especifica la resolución del ADC ideal que tendría la misma resolución que el circuito real. La definición más frecuente del ENOB es

Donde SINAD (Signal-to-Noise and distortion ratio) es un parámetro que mide la calidad de la señal frente al ruido y la distorsión.

Donde P es la potencia de la señal, ruido y distorsión. Habitualmente este parámetro esta expresado en dB

Modulador Sigma-Delta

Los moduladores Sigma Delta convierten señales analógicas en señales PDM (Pulse Density Modulation). La señal es representada por una trama de 1 bits de una frecuencia R veces superior a la frecuencia teórica de Nyquist (donde R es el factor de sobremuestreo). Esta trama de bits es filtrada digitalmente y submuestreada para convertir la trama en palabras digitales que representen la señal analógica.

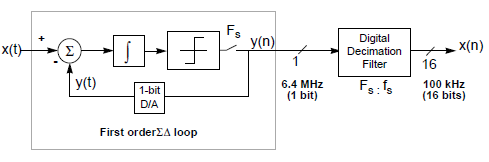


Diagrama en bloques de un modulador Sigma Delta de 1er orden.

Como se muestra en el diagrama, el modulador está conformado por un integrador, un cuantizador de 1 bit que funciona a una frecuencia de clock Fs determinada y un DAC de 1 bit.

Noise Shaping

Si usamos un modelo lineal para describir el modulador, podemos modelar el cuantizador de 1 bit como un proceso lineal en el cual se añade un ruido blanco E(z),tomamos la entrada X(z), la salida Y(z) y planteamos la ecuación de lazo cerrado del sistema.

Despejando la salida en función de la entrada y el ruido se llega a la siguiente ecuación

Si graficamos la Noise Transfer Fuction (NTF) en matlab, veremos que corresponde a una transferencia pasa altos. El efecto que produce esto es que desplaza el ruido hacia las altas frecuencias mejorando así la relación señal a ruido del modulador.

Relación señal a ruido modulador Sigma-Delta

La cantidad de ruido añadida sigue siendo la misma que se calculo en la sección (revisar), además, el sobremuestrear implica una mejora del SNR de 6.02 db por cada vez que duplicamos la frecuencia de muestreo. El hecho de que el ruido ahora este afectado por un filtro pasa altos, mejora aun más el SNR en la banda de paso de interés (-fb, fb), para calcular la SNR es necesario calcular la potencia de ruido en el ancho de banda de interés

Si OSR >> 1 entonces podemos realizar un desarrollo de Taylor de primer orden en torno al cero de sin(pi/OSR) donde obtendríamos que

Reemplazando en la ec

La anterior expresión indica la potencia de ruido en la banda de interés para un modulador sigma delta de primer orden. Con ella podemos calcular la relación señal a ruido

Si tomamos que OSR es una potencia de 2 (OSR=2^r) entonces obtenemos que

Si nos referimos al Numero efectivo de bits, obtenemos que la resolución de la señal mejora 1.5 bits efectivos cada vez que duplicamos la frecuencia de muestreo.

Filtro Decimador

Como se menciono anteriormente, la señal modulada contiene poco ruido de cuantizacion a bajas frecuencias, pero aumenta rápidamente hacia las frecuencias más altas. Si el ancho de banda Fb es mucho menor que Fs/2, podemos remover el ruido de las altas frecuencias utilizando un filtro digital pasabajos. Luego, la señal filtrada debe ser decimada para reducir la frecuencia de muestreo a la frecuencia de Nyquist 2Fb.

Una buena opción para estos filtros son los llamados filtros Sinc. Los filtros Sinc son filtros FIR que calculan la media móvil de la señal de entrada, por lo tanto, si esta es x(n), la salida y(n) esta dada por

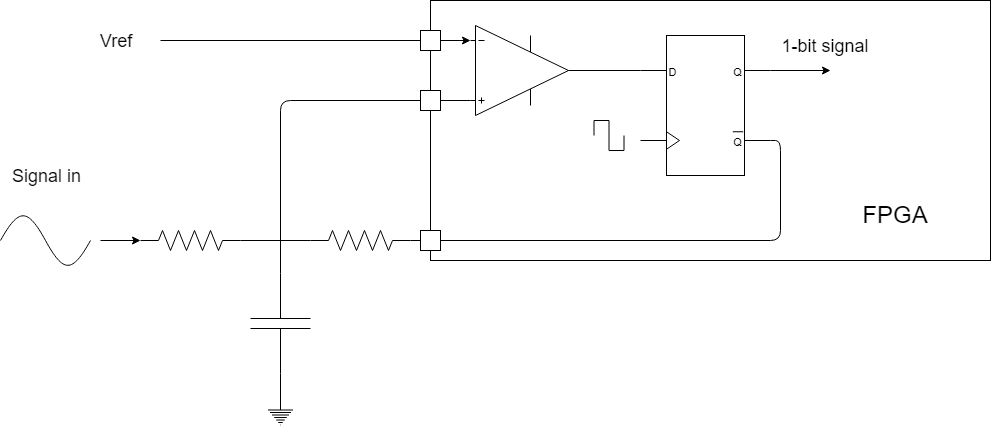
Su respuesta impulsiva está dada por:

Su transformada Z esta dada por

La ventaja de estos filtros estos filtros es que no requieren multiplicadores en su implementación, ya que si obviamos el término 1/N, todos sus coeficientes son 1. Esto lo hace realmente eficiente en cuestión de utilización de recursos dentro de un chip FPGA.

Implementación en FPGA

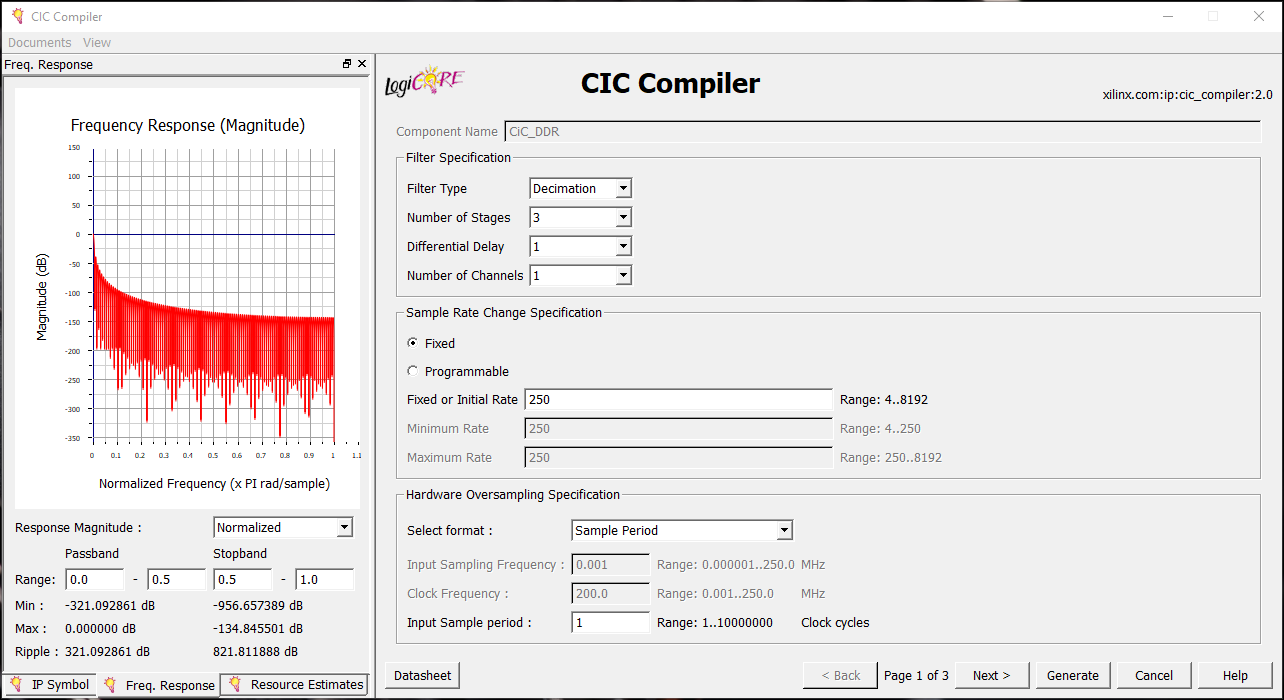
El kit utilizado es un Spartan 3 starter kit Board (UG130). Para la implementación del Modulador dentro de la FPGA, se implemento el circuito que se muestra en la Figura …. En este diagrama el integrador es reemplazado por una red RC, para el comparador utilizamos una entrada LVDS y como elemento de muestreo utilizamos un flip flop D.



El valor de Vref adoptado fue de 1.65V ya que este es el punto medio del voltaje de alimentación del chip FPGA.

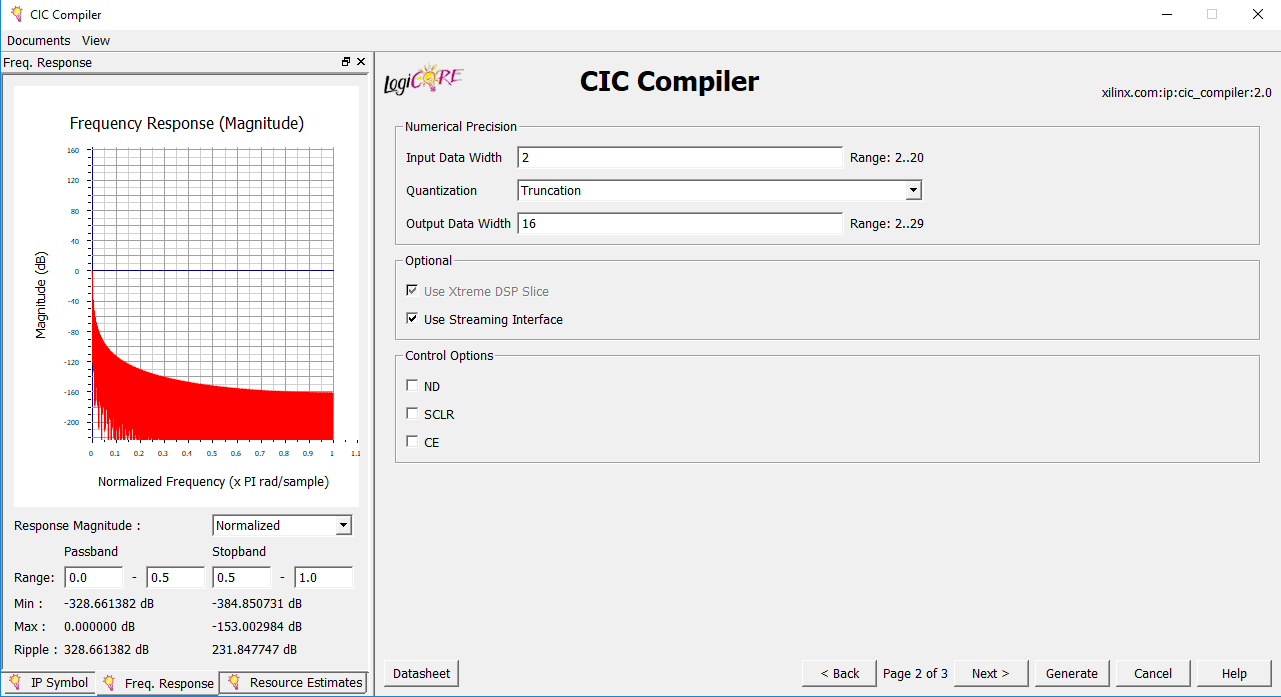
El filtro CIC fue descrito en HDL utilizando múltiples topologías, sin embargo ninguna de ella lograba pasar el análisis de timing de la herramienta de sintetizado, siendo su máxima frecuencia de operación de 100 Mhz.

Se debió recurrir al LogiCORE IP CIC Compiler V2.0 incluido en la herramienta de síntesis ISE 14.7, el cual proporciona un bloque de instanciación listo para implementar.



En la sección Sample Rate Change Specificaion seleccionamos la cantidad de decimaciones que serán realizadas a la señal. En la sección Hardware Oversampling Specification, fue seleccionado la opción Sample Period y la Input Sample period fue definida a 1, con esto le indicamos que por cada pulso de reloj el filtro recibirá una muestra para procesar.

En la página 2 del compilador, debemos especificar el ancho de palabra de entrada y de salida. El ancho mínimo de la entrada es de 2 bits, lo cual, teniendo en cuenta que el filtro procesa números positivos y negativos el rango mínimo de entrada es de -2 a 1. Como la salida del modulador son 1’s y 0’s, la adaptación que se debe hacer es simplemente constituir un vector std\_logic de ancho 2 donde el bit menos significativo se conecte a la salida del modulador y el más significativo se conecta directamente a 0. La palabra de salida se estableció en 16 bits, sin embargo, estos 16 bits no reflejan la resolución real del conversor A/D.



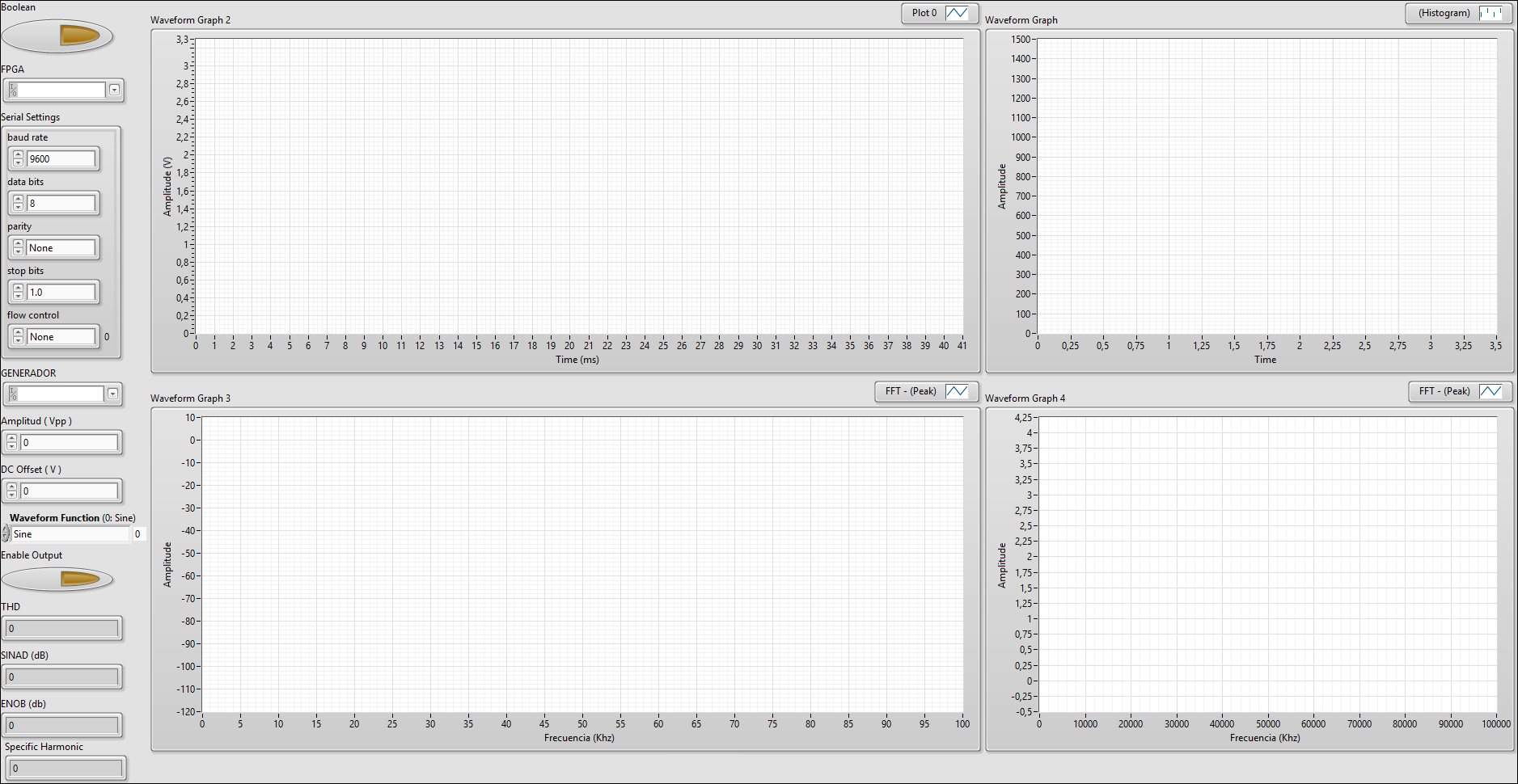
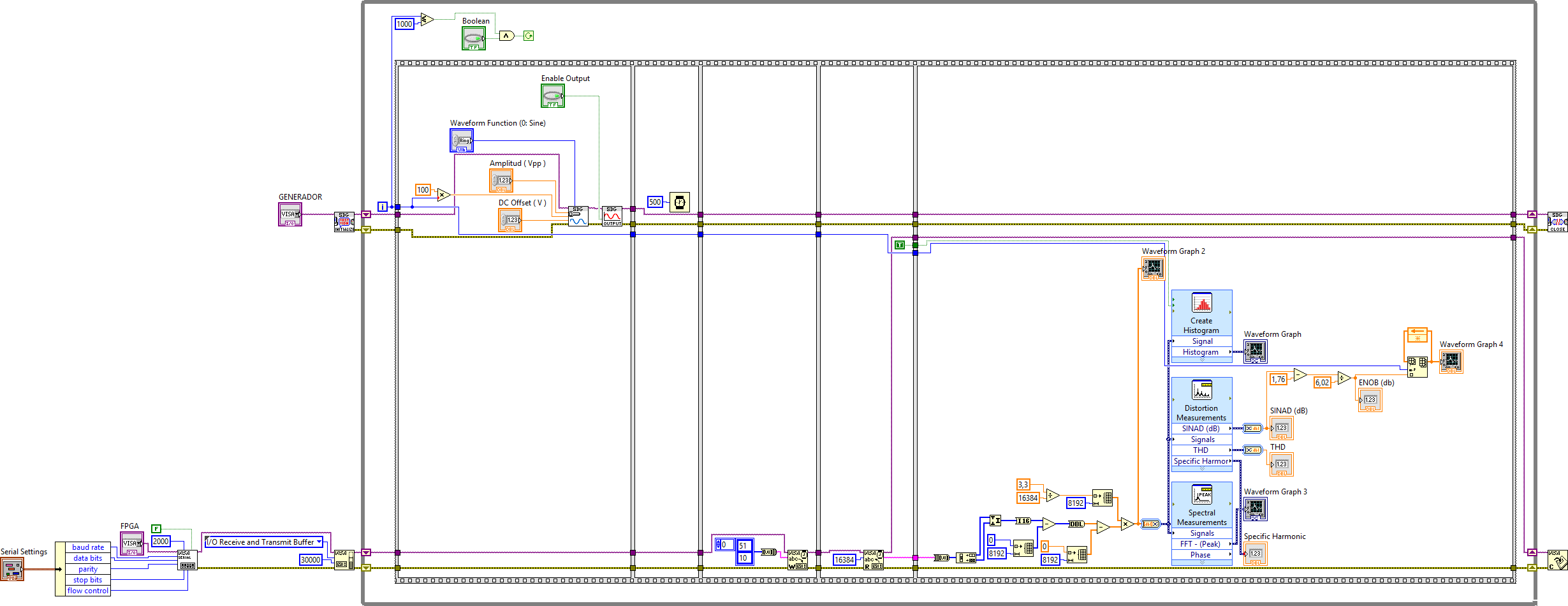
Para poder corroborar el funcionamiento del modulador se utilizo la herramienta embebida ScopeIO (Sagreras, Miguel Ángel) la cual permite visualizar señales que se encuentran dentro de la FPGA a través de un monitor conectado por VGA. El ScopeIO está preparado para funcionar con una interfaz Ethernet. Como el Kit utilizado no cuenta con esta conexión, fue necesario reemplazar el bloque de conexión Ethernet por uno de conexión serial RS232, además fue necesario modificar la interfaz realizada en Node WebKit para que esta pueda conectarse por serial.

Conexión a la PC

Para poder analizar la calidad de la señal, se utilizo un Instrumento Virtual creado en Labview el cual recibe la señal a través del puerto serie, se hubiera sido preferible utilizar Ethernet para lograr una mayor velocidad, pero el kit utilizado no cuenta con el conector ni la capa física.

Para realizar el envió se implemento en lenguaje VHDL un bloque que adquiera muestras que al recibir un comando especifico a través del puerto serie, reinicie un contador de 13 bits a cero cuya salida direccione una memoria ram dual port de 8192 posiciones por 16 bits por posición donde se almacenan los datos de salida del filtro CiC. Cuando el contador alcanza su cuenta máxima, este se auto-deshabilita quedando fijo en su máxima cuenta y donde dispara otro contador que direccionara el bloque de lectura para el envió de los datos a través de un bloque UART. Como la UART solo envía de a 8 bits, es necesario realizar 16384 envíos para despachar la memoria RAM completa, por ende el segundo contador tendrá 14 bits, de los cuales los 13 más altos direccionaran el puerto de lectura de la memoria RAM, mientras que el más bajo, seleccionara si se envía la parte alta o la parte baja de la palabra que se obtenga del bloque de memoria. Estos datos luego serán reconstruidos en el instrumento virtual.

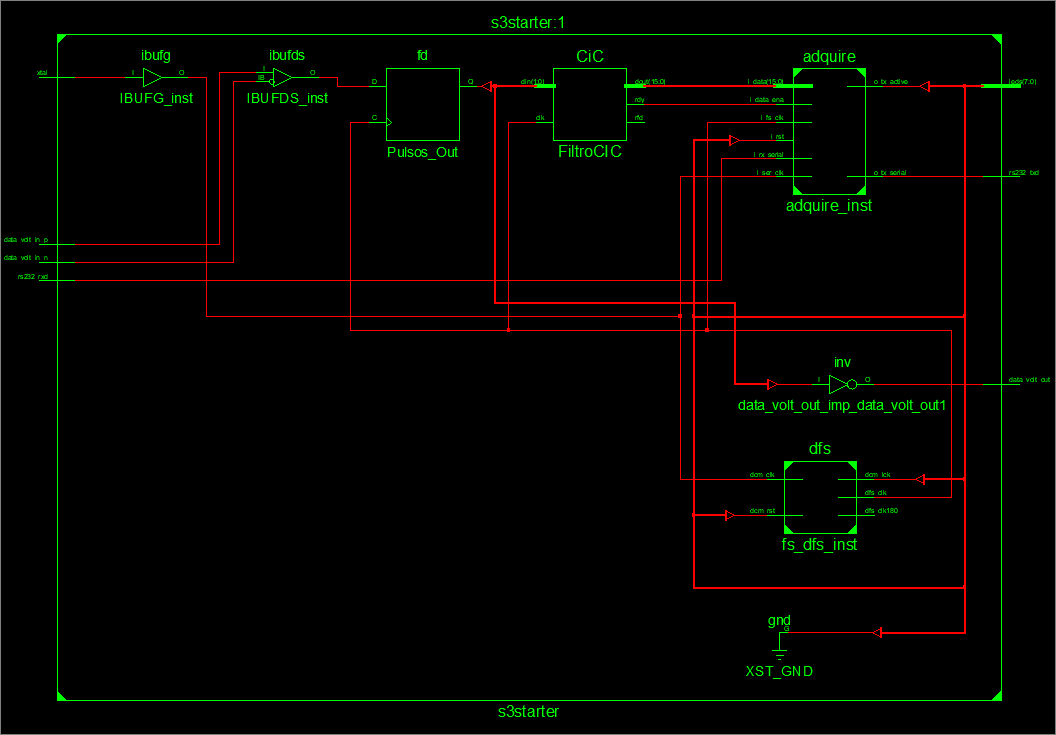
El diagrama en bloques del instrumento virtual es el que se muestra en la figura …, Para generar la señal se utilizo un generador de funciones arbitrarias SIGLENT SDF810 el cual se controla desde el instrumento virtual con las bibliotecas particulares del instrumento.



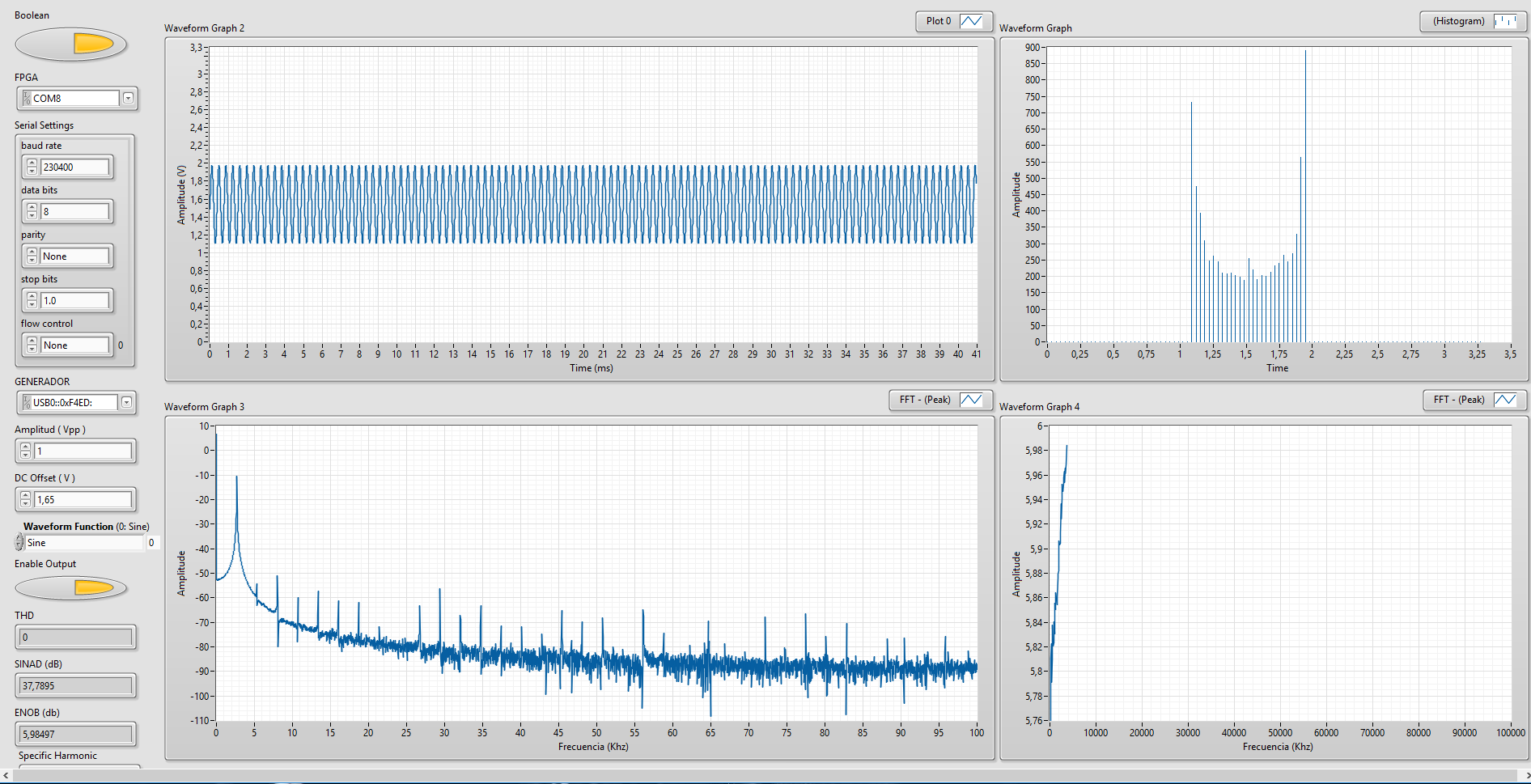
El instrumento, el cual automatiza la medición, realiza un barrido en frecuencia y captura los datos que luego son utilizados para calcular la cantidad efectiva de bits, la distorsión armónica total de la señal obtenida desde la FPGA, así como también graficar su respuesta en frecuencia y un histograma de las muestras recibidas.

Mediciones

El esquemático RTL del hardware descrito es el que se observa en la figura… .



En el instrumento se selecciona el puerto serie al que se conecta el kit FPGA y el puerto USB en el que se conecta el generador. El Baud Rate seleccionado fue el máximo permitido por LabView 230400 para que el tiempo de transferencia fuese el mínimo posible.



Para las mediciones primeramente se utilizo una frecuencia de 100Mhz con múltiples tasas de decimacion. La curva que se buscaba medir fue el número efectivo de bits, la cual resulta variante con la frecuencia, luego se elevo la frecuencia un 35% para ver como variaba el número efectivo de bits.

Las curvas obtenidas para para una frecuencia de clock de 100 Mhz

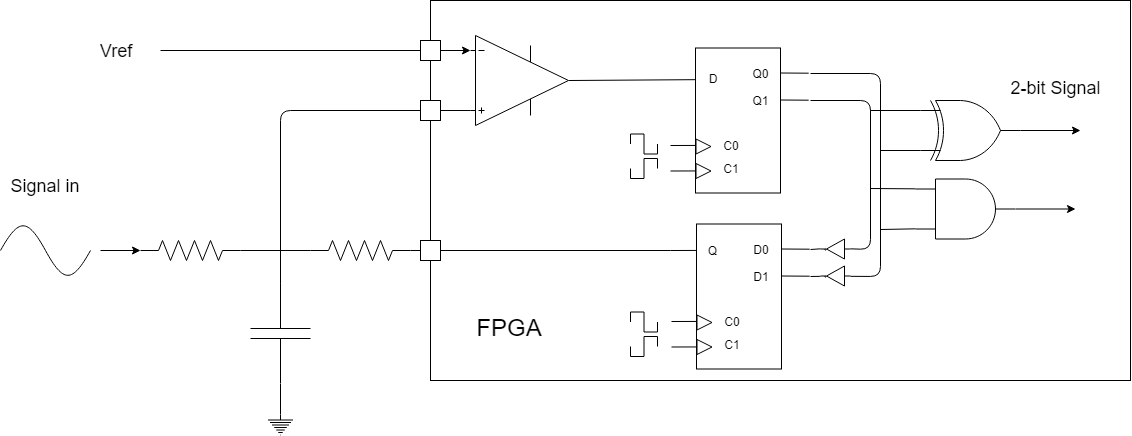
Donde R es la cantidad de decimaciones realizadas por el filtro CiC.

Como habíamos dicho antes, la cantidad de decimaciones realizadas han de ser seleccionadas de forma tal que sean iguales a la tasa de sobremuestreo. Tal como se vio de forma matemática, cuando la cantidad de decimaciones (tasa de sobremuestreo) aumenta, la cantidad de bits efectivos también lo hace. Sin embargo, los valores experimentales obtenidos son menores a los que se obtienen de forma teórica.

Sigma Delta con flip flop DDR

Una mejora que se planteo es realizar el sigma delta con Flip Flops DDR. Esto permitirá seguir trabajando a las mismas frecuencias con una señal de clock mas relajada, lo que implica una reducción del consumo de potencia del conversor. El esquema propuesto es el que se muestra en la Figura… .

Un inconveniente que surgió al intentar utilizar las instancias de estos flip flops es que el pad de entrada del Kit utilizado no permite realizar la conexión de una entrada LVDS a los flip flops DDR de la entrada. Por lo tanto se tuvieron que describir los flip flop DDR por comportamiento.



Otro detalle a tener en cuenta, es que previo al filtro se deberá realizar una decimacion extra, esto es necesario debido a que el filtro no funciona con flip flops DDR. Por lo tanto es necesario reducir la frecuencia de datos de la salida de los flip flop DDR a la mitad, esto se logra haciendo promedios de a dos muestras siguiendo el ritmo de alguno de los 2 relojes de los flip flops DDR.

Para el caso de los FFDDR se utilizo una frecuencia de reloj de 50Mhz lo que equivale a una FS de 100Mhz. La curva obtenida para este caso se ve en la figura…

.Si bien conseguiremos una pequeña reducción en el consumo de potencia, la perdida de números efectivos de Bits, particularmente a bajas frecuencias es considerable.

Lecciones aprendidas

El eje temático de la beca fueron los ADC con tecnología sigma delta. En primera instancia se realizo un estudio de su funcionamiento teórico, desde lo básico sobre teoría de muestreo, hasta las formulas de la relación señal a ruido del modulador lo cual complemento el aprendizaje durante mi formación académica.

En el área del procesamiento digital de señales, se realizo un estudio sobre los filtros CiC siendo estos los más recomendados para las implementaciones en FPGA. Se intento describirlos en VHDL utilizando la forma recursiva clásica multi-etapa, una arquitectura en descomposición polifásica, pero finalmente se tuvo que recurrir a los IP cores de Xilinx. La principal lección aprendida de esto, es que a veces es preferible recurrir a las herramientas que nos proporcionan los fabricantes para evitar perder el tiempo tratando de hacer algo que ya está hecho.

Cuando tuve que realizar la adaptación de la interfaz grafica del ScopeIO que está hecha en JavaScript, me vi forzado a aprender sobre este lenguaje de programación, el cual luego me sirvió para desarrollar un trabajo para una materia de sistemas embebidos.

Sobre las FPGAs, la lectura en detalle de los manuales de usuario de los chips me ayudo a mejorar mi comprensión sobre la arquitectura interna de los chips y los recursos internos con los que cuentan. Como instanciar sus componentes así como la utilización de las herramientas de síntesis y de simulación.