

Capítulo 6

Implementación del Conversor Digital/Analógico

6.1. Requerimientos Generales

Si bien la placa de desarrollo no cuenta con un circuito integrado para realizar la conversión digital/analógica, en este capítulo se presenta cómo es posible fabricar un DAC utilizando la lógica programable del Zynq junto con circuitería electrónica externa. Básicamente, esta función es realizada por un sistema electrónico cuya entrada es un valor digital indicando una proporción entre la tensión de salida y un voltaje de referencia K . En principio, se estableció que el conversor D/A a diseñar deberá generar fielmente la señal reconstruida por el algoritmo de Sensado Compresivo. Por ello, se plantearon las siguientes especificaciones:

1. Relación señal a ruido de por lo menos 60 dB (1000 veces). Este parámetro deberá contemplar tanto el error de cuantización¹ como cualquier producto generado a causa de distorsión no lineal.
2. Ancho de banda comprendido en el rango completo del espectro audible humano. Aproximadamente hasta 15 ~ 20 KHz.
3. Utilizar aritmética de punto fijo en la lógica programable.
4. No exceder los 5 Mhz de frecuencia de salida en los conectores PMOD, para prevenir la utilización de los puertos de alta velocidad.
5. Evitar aumentar la complejidad del circuito analógico externo al SoC siempre que sea posible.

¹Error producido por aproximar muestras al valor más cercano dentro un conjunto predefinido. Es modelado como ruido agregado a la señal ideal.

Asumiendo que la señal y el ruido de cuantización no están correlacionados², la relación *señal a ruido de cuantización* SQNR se puede estimar como [28]:

$$SQNR = 6,02N + 1,76 \text{ [dB]} \quad (6.1)$$

La ecuación (6.1) nos permite encontrar la cantidad de niveles de voltaje discernibles como 2^N , con N la longitud de las muestras digitales³ en bits. La resolución está determinada por $\frac{K}{2^N}$, donde el DAC entrega valores de tensión entre 0 y K volts. Para cumplir la primera especificación, utilizamos la ecuación (6.1) y se obtiene la *cantidad de bits efectivos* o ENOB:

$$ENOB = \left\lceil \frac{60 \text{ [dB]} - 1,76 \text{ [dB]}}{6,02 \text{ [dB]}} \right\rceil = 10 \text{ bits} \quad (6.2)$$

Una de las primeras opciones consideradas para la generación de la señal de voltaje fue la *Modulación por Ancho de Pulso* (PWM). En esta técnica de conversión, los valores discretos de amplitud de la señal digital $x[n]$ modulan al ciclo de trabajo $\delta = \frac{K\tau}{T}$ a una tasa igual a la frecuencia de muestreo $f_s = \frac{1}{T}$ (ver Figura 6.1). Si $x[n]$ presenta su amplitud A normalizada entre $[-1, 1]$, entonces cada T segundos el tiempo de encendido τ se actualiza como $\tau = T(\frac{A+1}{2})$. Además, realizando la expansión en series de Fourier para la forma de onda en este caso y considerando simetría par, se obtiene [29]:

$$f(t) = A_0 + \sum_{n=1}^{\infty} [A_n \cos(\frac{2n\pi t}{T}) + B_n \sin(\frac{2n\pi t}{T})] \quad (6.3)$$

con:

$$A_0 = K\delta, A_n = K \frac{1}{n\pi} [\sin(n\pi\delta) - \sin(2n\pi(1 - \frac{\delta}{2}))], B_n = 0 \quad (6.4)$$

Por lo tanto, de las ecuaciones (6.3) y (6.4) se infiere que realizando un filtrado pasabajos, es posible conservar la componente de baja frecuencia A_0 y obtener la versión analógica de la señal $x(t)$; ya que esta contiene la información de $x[n]$ contenida en el valor medio instantáneo. Los términos A_n deberán ser eliminados ya que su presencia degradan la SNR. Una forma de realizar este sistema se muestra en la Figura 6.2, en la cual el modulador PWM se implementa en lógica programable y mediante la utilización de un pin PMOD se obtiene la señal de tensión luego de un filtro analógico. En este caso, la resolución N en bits del DAC está determinado por la frecuencia de reloj f_{clk} de la siguiente manera:

$$N = \log_2\left(\frac{f_{clk}}{f_s}\right), N \in \mathbb{N} \quad (6.5)$$

²Esta premisa no se cumple cuando una señal y su frecuencia de muestreo están relacionadas armónicamente.

³Se asume enteros sin signo.

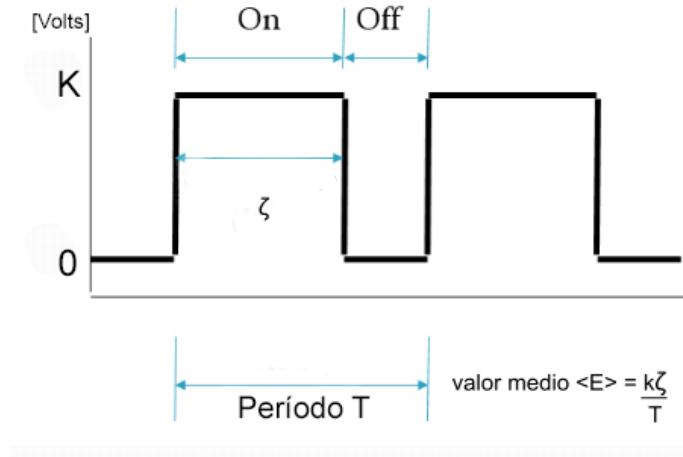


Figura 6.1: Señal PWM

Sin embargo, realizar este método presenta una serie de desventajas. En primer lugar, existe una relación de compromiso entre f_{clk} y el numero de bits. Si consideramos que la frecuencia de muestreo f_s es igual a dos veces el ancho de banda AB deseado, se cumple que la frecuencia de clock f_{clk} es:

$$f_{clk} = 2^{(N+1)} AB \quad (6.6)$$

Para $N = 10$ y $AB = 20 \text{ KHz}$ se necesita una frecuencia de reloj $f_{clk} = 40,96 \text{ MHz}$, por lo que no es posible cumplir con el quinto item de las especificaciones. Si relajamos la condición del ancho de banda a 5 KHz y utilizando 9 bits para la representación, entonces $f_{clk} = 5,12 \text{ MHz}$. En segundo lugar, la componente fundamental de la señal PWM estará a una octava de la frecuencia de corte del filtro pasabajos. En ese caso, si se desea por lo menos una atenuación de 40 dB a f_s se necesitará un filtro de séptimo orden (140 dB por década de atenuación) por lo que su diseño y fabricación se vuelve complejo. Una solución a este inconveniente sería disminuir la relación $\frac{AB}{f_s}$ para aliviar los requerimientos del filtro, aunque en ese caso el ancho de banda queda bastante comprometido. A su vez, según se especificó en el Capítulo 4, debido al diodo de protección y resistencia serie de 200Ω los conectores PMOD presentan una frecuencia máxima de salida igual a 24 MHz . Entonces, no es posible aumentar f_{clk} por encima de este límite. Volviendo al caso de $AB = 5 \text{ KHz}$, el filtro debería tener una pendiente de por lo menos 60 dB por octava. Queda entonces fundamentado que las especificaciones establecidas al comienzo de este capítulo no pueden ser cumplidas con la técnica de PWM. Es por ello que se investigaron otras alternativas.

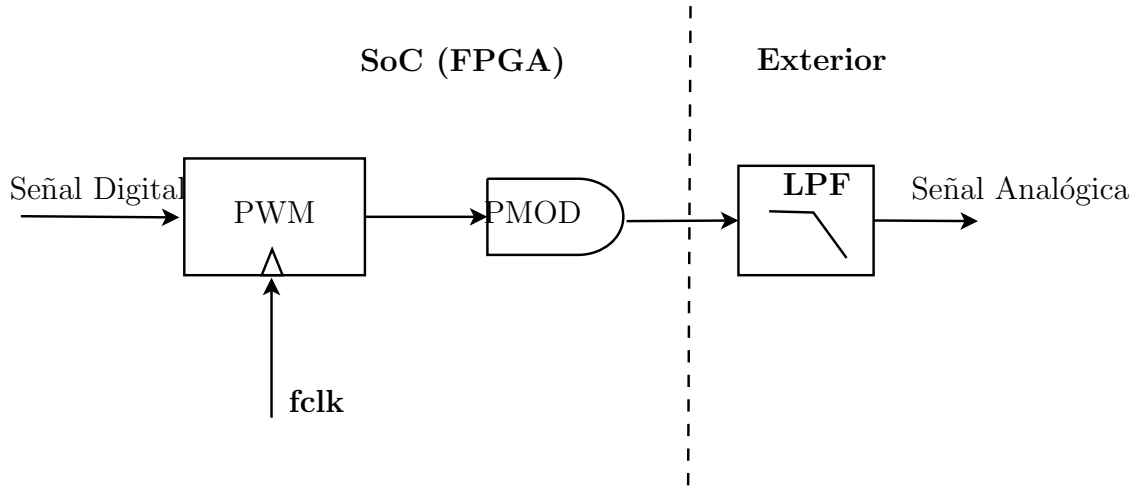


Figura 6.2: Primera propuesta de DAC con PWM.

6.2. Modulador Delta - Sigma

6.2.1. Delta Sigma de Primer Orden

Considerando al DAC como una planta a controlar, en la Figura (6.3) se muestra la esencia del modulador $\Delta\Sigma$. La señal digital $R(z)$ que se desea convertir al dominio analógico es comparada con la salida del conversor D/A generando la señal de Error $E_r(z)$, ingresando al controlador $G_c(z)$. Luego, la señal de control $U(z)$ ingresa a la planta generando la señal en tiempo continuo $Y(s)$. Esta configuración resulta conveniente, ya que la mayor parte del sistema es digital, lo que propicia alta integración en el SoC, no presenta derivas térmicas y es de bajo costo. Además, como se verá en breve, permite obtener excelentes SNR gracias al *Modelado de Ruido*. Inherentemente, los DAC de N bits pueden presentar transferencias alineales como cuantización no lineal si no son calibrados [30]. Sin embargo, utilizando sólo un bit en la conversión estos problemas son evitados. Por eso, el diagrama en bloques mostrado en la Figura 6.4 es una realización práctica en donde la realimentación es en el dominio discreto y la salida presenta sólo dos niveles posibles. Linealizando el modelo de control y considerando al error de cuantización como una perturbación que se suma a la señal analógica, el sistema resultante es el de la Figura (6.5). La transferencia respecto a la señal a convertir es:

$$T_R(z) = \frac{G_c(z)}{1 + G_c(z)} \quad (6.7)$$

Mientras que respecto al error $E(z)$ resulta:

$$T_E(z) = \frac{1}{1 + G_c(z)} \quad (6.8)$$

Usualmente, se utiliza como controlador un bloque integrador⁴. Esto permite gene-

⁴De aquí proviene el nombre *sigma*, haciendo referencia al proceso de acumulación de la señal de error. La denominación *delta* surge de la diferencia entre la entrada y la salida.

rar la señal $U(z)$ en función de los valores previos y actuales de $E_r(z)$. La integración discreta se implementa como:

$$G_c(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1} \quad (6.9)$$

Por lo que, reemplazando (6.9) en (6.7) y (6.8) se obtiene:

$$T_R(z) = \frac{Y(z)}{R(z)} = \frac{z^2 - z}{2z^2 - 3z + 1} = \frac{0,5z}{z - 0,5} \quad (6.10)$$

$$T_E(z) = \frac{Y(z)}{E_r(z)} = \frac{z - 1}{2z - 1} = \frac{0,5(z - 1)}{z - 0,5} \quad (6.11)$$

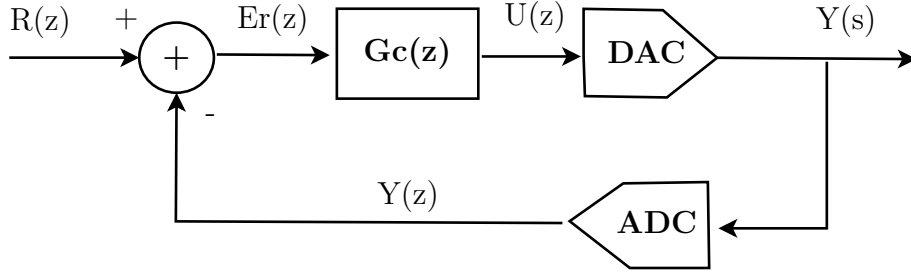


Figura 6.3: Lazo de control para la conversión D/A

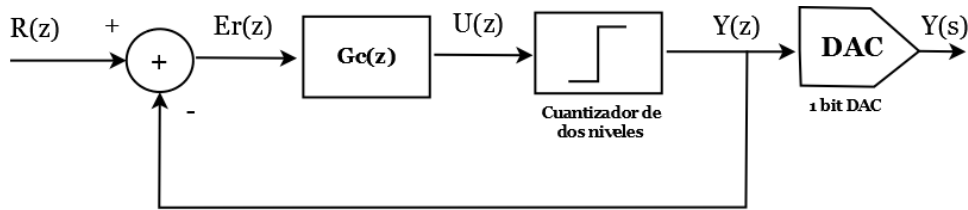


Figura 6.4: Modelo real del modulador Delta Sigma.

Graficando las transferencias (6.10) y (6.11) (Figura 6.6) se puede apreciar que en las bajas frecuencias el lazo no atenúa el contenido espectral de la señal de entrada pero sí rechaza al ruido de cuantización. Esto último es de gran importancia, ya que a pesar que el DAC de un bit presenta demasiado error de conversión (sólo puede generar dos valores discretos) gracias al modulador $\Delta\Sigma$ es posible obtener SRN comparables a convertidores multibits. La técnica de aplicar el filtrado pasa-altos de la Figura 6.6b a la cuantización se denomina *Modelado de Ruido*. Aunque, para

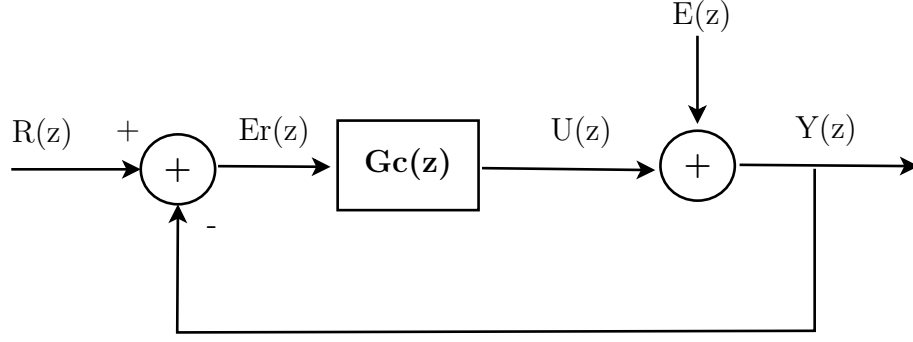


Figura 6.5: Modelo lineal del modulador Delta Sigma.

aprovechar este beneficio, el ancho de banda útil de la señal $R(z)$ debe ser mucho menor que $\frac{f_s}{2}$. Es por ello que se requiere utilizar una etapa de sobremuestreo previo a ingresar al modulador. Se define el factor de sobremuestreo $M \in \mathbb{N}$ de la siguiente manera:

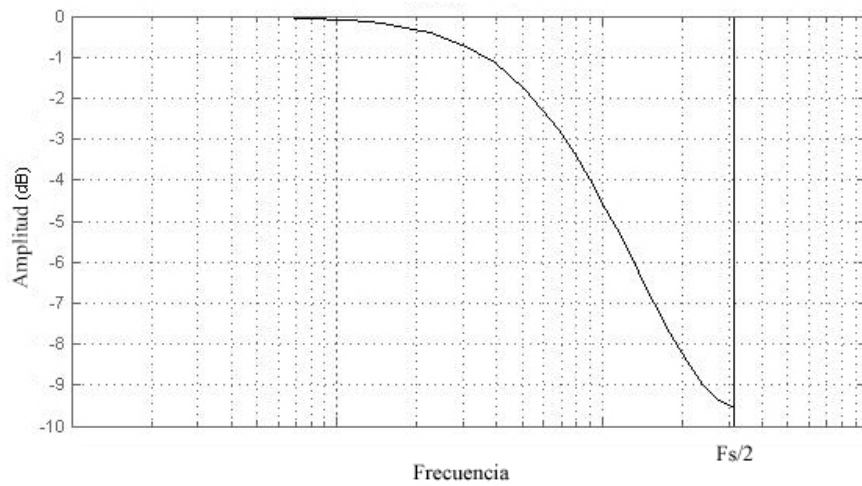
$$M = \frac{f_s}{2AB} \quad (6.12)$$

Entonces, volviendo a las especificaciones requeridas para el DAC, la gráfica de la Figura 6.7 establece que para $M = 32$, la relación SNR es casi 60 dB en un modulador de segundo orden⁵. Implementando al conversor $\Delta\Sigma$ en lógica programable, la frecuencia de reloj requerida sera igual a la frecuencia de muestreo. Es decir, se deberá generar una muestra de salida por cada ciclo de reloj. La siguiente ecuación de diseño resulta entonces:

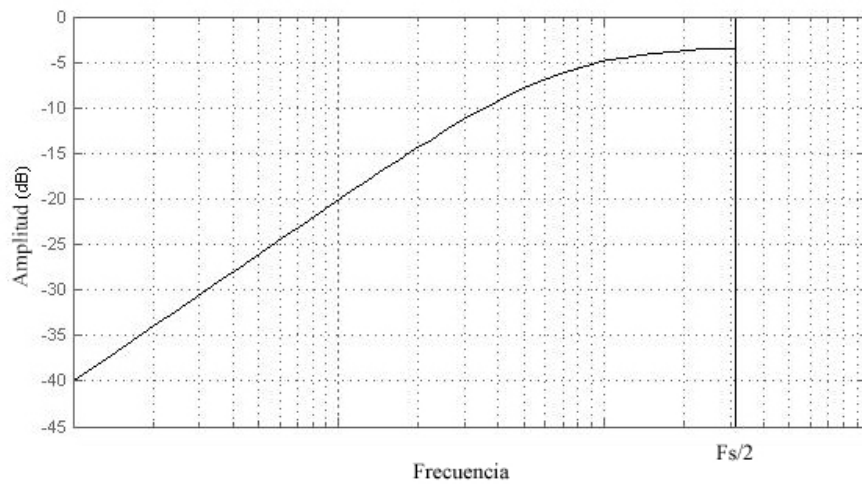
$$f_{clk} = f_s = 2AB.M \quad (6.13)$$

Siendo así, se opta por realizar un conversor de orden dos ya que se requiere un modulador de orden superior para alcanzar mejores SNR sin aumentar demasiado el factor M . A su vez, no se recomienda utilizar un único lazo ya que esto genera oscilaciones de baja frecuencia para entradas DC ([30] y [31]).

⁵El orden denota la cantidad de lazos de realimentación.



(a) Transferencia de lazo respecto a la señal de entrada



(b) Transferencia de lazo respecto al error de cuantización

Figura 6.6: Transferencia de lazos a frecuencia normalizada $F_s/2 = 1$.

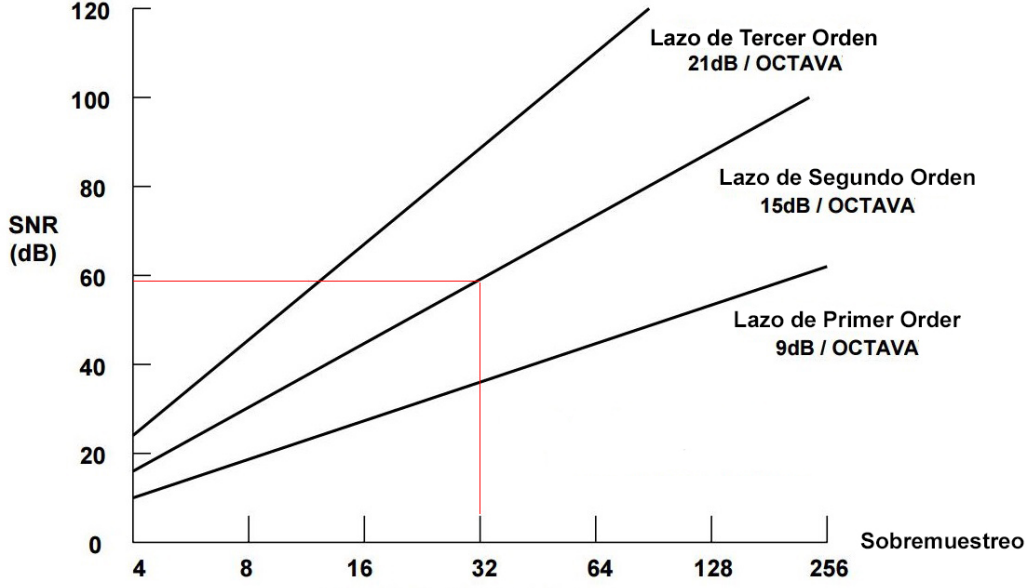


Figura 6.7: Relación entre el orden de un modulador $\Delta\Sigma$ y la cantidad de sobremuestreo necesario para conseguir un SNR determinado [32]. Se muestra también la atenuación del ruido de cuantización en dB por octava.

6.2.2. Delta Sigma de Segundo Orden

La topología del sistema de segundo orden junto con sus filtros de lazo fueron obtenidos de [33], en donde se realiza su diseño para receptores de banda ultra ancha. Se requiere que la amplitud de la señal digital a la entrada del modulador esté acotada en el intervalo $[-0.9, 0.9]$ para evitar problemas de inestabilidad ya que se trata de un sistema no lineal. A la salida, la señal $x_p[m]$ adopta el valor -1 si la entrada al bloque $sgn()$ es negativa, en caso contrario vale 1. El diagrama en bloques se presenta en la Figura 6.9. Linealizando como en el caso anterior se obtiene la transferencia total:

$$TLC(z) = \frac{X_p(z)}{X(z)} = \frac{1/4}{(z - p_1)(z - p_2)} \quad (6.14)$$

Con $p_1 = 0,75 + 0,433j$ y $p_2 = 0,75 - 0,433j$. El sistema es estable ya que $|p_1| = |p_2| \leq 1$; los polos se encuentran dentro del círculo unitario. Para analizar qué sucede con el margen de fase M_Φ y ganancia M_g primero se debe plantear la transferencia de lazo abierto:

$$GH(z) = \frac{1/4}{(z - 1)(z - 0,5)} \quad (6.15)$$

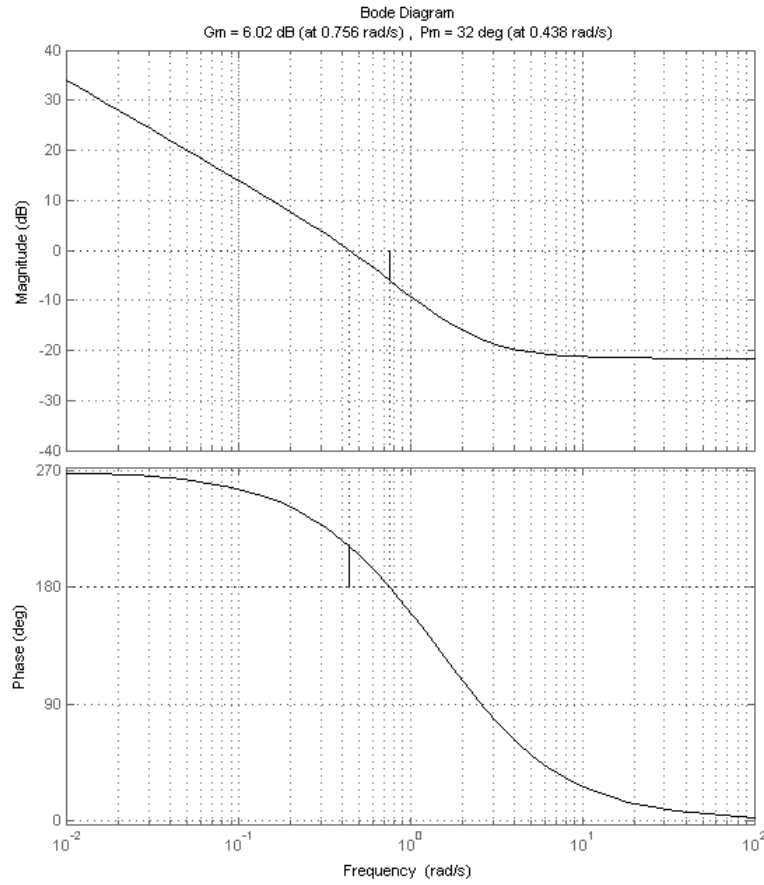


Figura 6.8: Diagrama de Bode para la obtención del margen de fase y ganancia en el modulador de orden dos.

Luego, se aplica la *transformación bilineal* con $f_s = 1$ a (6.16) a fin de realizar el análisis de estabilidad en el dominio continuo:

$$GH(s) = 0,0833 \frac{(s - 2)^2}{s(s + 0,67)} \quad (6.16)$$

Armando el diagrama de Bode de la Figura 6.52 se haya que $M_\Phi = 32^\circ$ y $M_g = 6$ dB. Ambos márgenes son positivos, por lo tanto el sistema es estable. En el caso discreto no hay riesgo que se produzca una situación de inestabilidad debido a variaciones en estos valores ya que no existen derivas en los parámetros del sistema.

Re-expresando al modulador en función de bloques integradores y retardos unitarios, con el fin de efectuar el diseño en lógica programable, se obtiene la estructura de la Figura 6.10. Se decidió utilizar representación en punto fijo de 24 bits complemento a dos mediante el formato $Q_{7,17}$, donde 17 bits son fraccionarios, 6 bits representan la

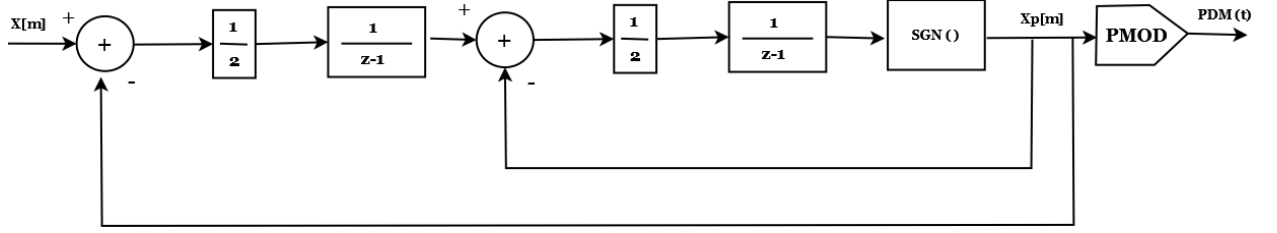


Figura 6.9: Modulador $\Delta\Sigma$ de segundo orden.

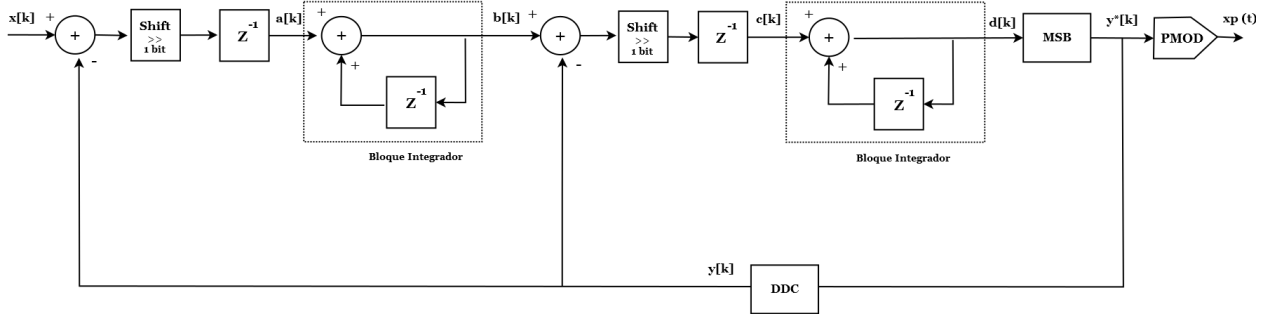


Figura 6.10: Diagrama detallado del modulador realizado en FPGA. Se muestra también la ubicación de las variables de estado $a[k]$, $b[k]$, $c[k]$ y $d[k]$.

parte entera y un bit es utilizado para el signo. La resolución es $2^{-17} = 7.6294 \times 10^{-6}$ y el rango de valores posibles está comprendido en el intervalo $[-64; 63,99999]$. La razón por la que escogió 6 bits de parte entera es para evitar el desborde aritmético en los nodos integrales del modulador. La multiplicación por $\frac{1}{2}$ se realizó mediante un registro desplazamiento con el fin de utilizar LUTs en vez de bloques DSP. La función signo es producida por el bloque MSB, cuya salida es el bit de signo del dato de entrada. Luego, el convertidor DDC genera como respuesta el valor 131071 (0.9999 en formato $Q_{7,17}$) cuando el bit de entrada vale 1 y -131072 (-1 en formato $Q_{7,17}$) si es 0. El siguiente paso fue plantear cuatro variables de estado y la expresión de la salida:

$$\begin{aligned}
 a[k] &= \frac{1}{2}x[k-1] - \frac{1}{2}y[k-1] \\
 b[k] &= a[k] + b[k-1] \\
 c[k] &= \frac{1}{2}b[k-1] - \frac{1}{2}y[k-1] \\
 d[k] &= c[k] + d[k-1] \\
 y[k] &= \text{sgn}(d[k])
 \end{aligned} \tag{6.17}$$

Por lo tanto, se necesitaron 10 registros para almacenar datos: ocho para las variables de estado (valor actual y anterior), dos para recordar la entrada y salida del instante anterior. En cada flanco positivo de la señal de reloj se genera un bit de salida y se actualizan los estados. El código VHDL del modulador se encuentra en la sección A.3.3 del apéndice.

6.3. Diseño completo del DAC

El vector $x[n]$ generado en el sistema reconstructor de Sensado Compresivo es enviado a una etapa de interpolación con un factor $M = 32$. Después, se aplica saturación de manera preventiva para acotar la amplitud entre $[-0.9, 0.9]$; ya que el filtrado realizado en el sobremuestreo digital puede generar transitorios de amplitud mayores a ese rango. El modulador $\Delta\Sigma$ es el siguiente en la cadena de procesamiento, generando un stream de bits PDM. El pin del conector PMOD actúa como un DAC de 1 bit, generando una tensión de 3.3 Volts para un valor lógico en alto y 0 Volts en bajo. Finalmente, los pulsos de tensión ingresan a un filtro pasabajos Bessel de cuarto orden y frecuencia de corte igual a 20 KHz. Se usó este tipo de filtro ya que presenta respuesta de fase lineal dentro de la banda de paso. Con esto se elude la distorsión de fase lo cual provoca que las envolventes de las señales no conserven su forma original.

El reloj del PL en el Zynq 7000 funciona con una frecuencia global de 100 MHz. Por cuestiones de diseño del interpolador, se optó por dividir esta señal por un factor de 64. Esto genera un f_{clk} para el DAC de 1.56 MHz y utilizando la ecuación (6.13) el ancho de banda para la señal analógica $x(t)$ resulta 24.41 KHz.

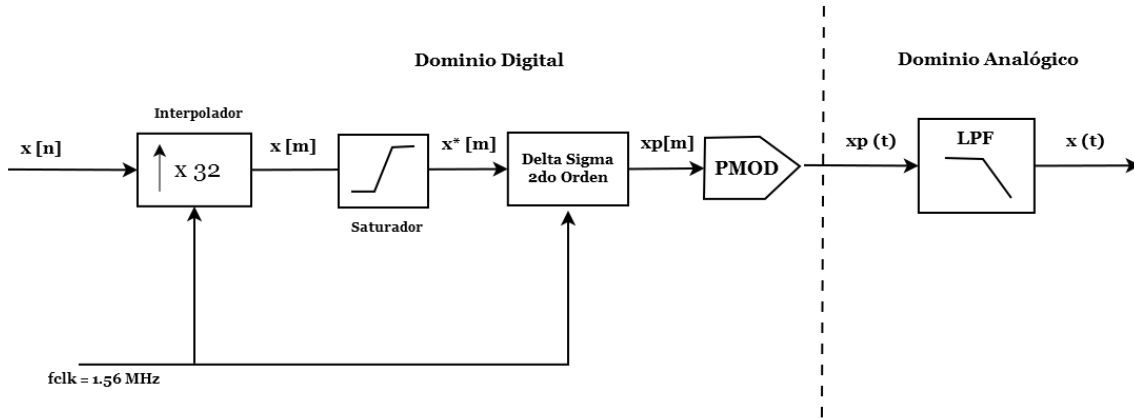


Figura 6.11: Diagrama en bloques del sistema de conversión D/A desarrollado.

En el Cuadro 6.1 se efectúa una comparación entre el DAC PWM propuesto al comienzo junto con el diseño $\Delta\Sigma$ realizado. Se puede apreciar que en el sistema de segundo orden se emplea una frecuencia de reloj tres veces menor, se logra cinco veces mayor ancho de banda y un aumento de la SNR de 20 dB. Sin embargo, la implementación en lógica programable es más compleja ya que se necesita de interpolación.

	PWM	$\Delta\Sigma$
f_{clk}	5.12 MHz	1.5625 MHz
AB	5 Khz	24.41 KHz
Filtro	Séptimo Orden	Cuarto Orden
SNR	$\approx 40dB$	$\approx 60dB$
ENOB	6 bits	10 bits
Interpolación	No	Si

Cuadro 6.1: Comparación entre DAC PWM y $\Delta\Sigma$

6.3.1. Modelo de Simulink

Con el fin de verificar el correcto funcionamiento del modulador y sus especificaciones técnicas, se armó un modelo de simulación (Figura 6.12) utilizando la herramienta Simulink[®] incluida en la versión R2013a de Matlab[®]. Este paso fue hecho antes de la programación del FPGA en el SoC. Primero, se ingresa al sistema con una sinusoidal discreta (bloque *Señal Digital*) de amplitud 0,7 pico y frecuencia igual a 5 Khz. El periodo de muestreo empleado es $T_s = 640$ ns. Después, se ubicaron varios cuantizadores con intervalos $q = 2^{-17}$ para simular la precisión del formato $Q_{7,17}$. Los bloques *SGN* Y *DDC* realizan las funciones detalladas en el diseño del $\Delta\Sigma$ de segundo orden. Las señales medidas con el elemento *Scope* que se presentan en la Figura 6.13 fueron la entrada (6.13a), la señal PDM generada por el DAC de 1 bit (6.13b) y la versión analógica a la salida (6.13c). Además, la Figura 6.14 muestra las señales internas a los lazos del sistema. La primera de ellas (6.14a) es la diferencia entre la sinusoidal de entrada y la salida PDM. Luego, se produce la segunda señal de error (6.14b) y finalmente ingresa la forma de onda (6.14c) al DAC.

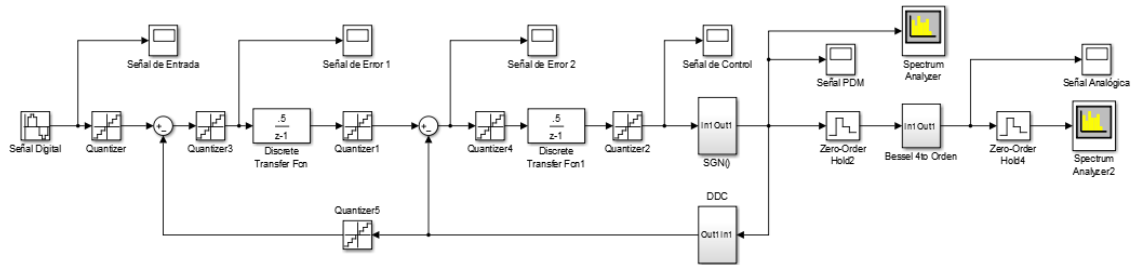
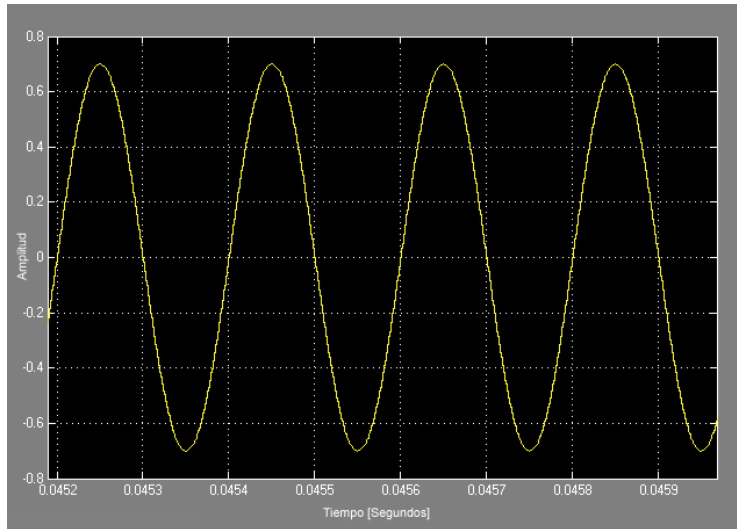
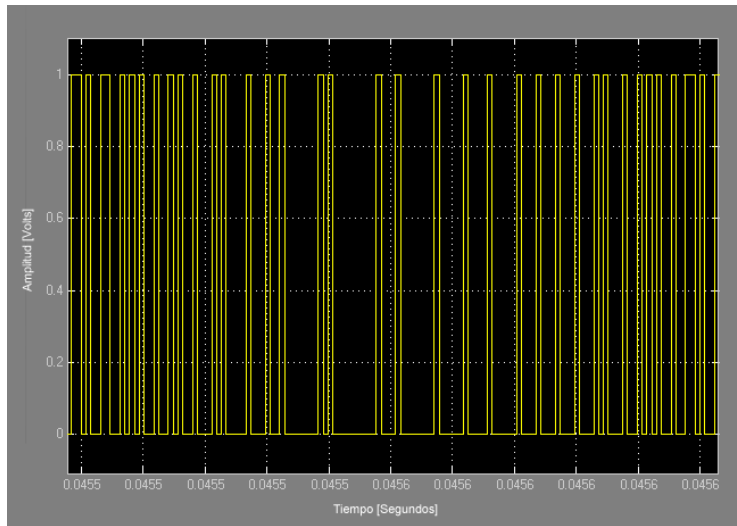


Figura 6.12: Modelo de simulación realizado en Simulink para el convertor Delta-Sigma.

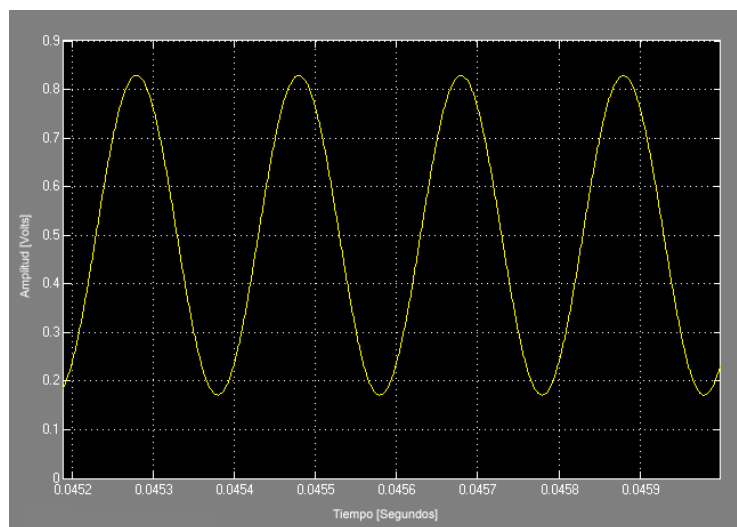
Efectivamente, en la Figura 6.15a se aprecia el efecto de modelado de ruido en el modulador. El ruido, que es tratado independientemente a la señal de entrada por el sistema, presenta la forma de transferencia pasa-altos. En la zona de bajas frecuencias, se encuentra el seno representado como un pico ubicado en 5 Khz extendiéndose hasta 0 dBm (carga de referencia de 50Ω). Con el bloque *Bessel 4to Orden* se corroboró que el filtro pasabajos, con frecuencia de corte (3 dB) de 20KHz y atenuación igual a 80 dB por década, mitiga al ruido de cuantización en las altas frecuencias. La sinusoidal analógica finalmente resulta con un SNR ≈ 70 dB (Figura 6.15b).



(a) Señal digital de entrada.

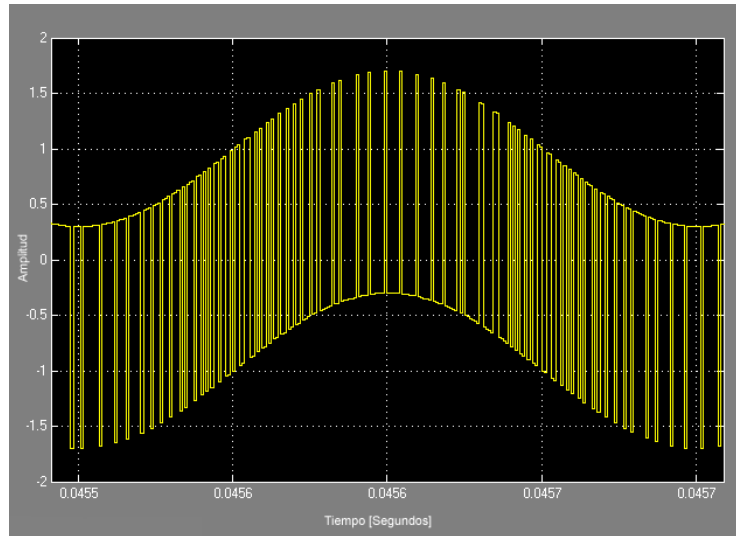


(b) Señal PDM de salida.

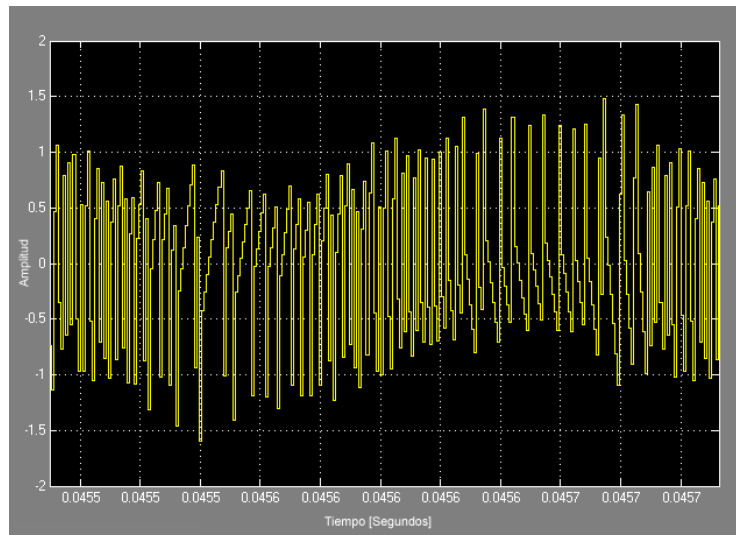


(c) Señal analógica a la salida del filtro.

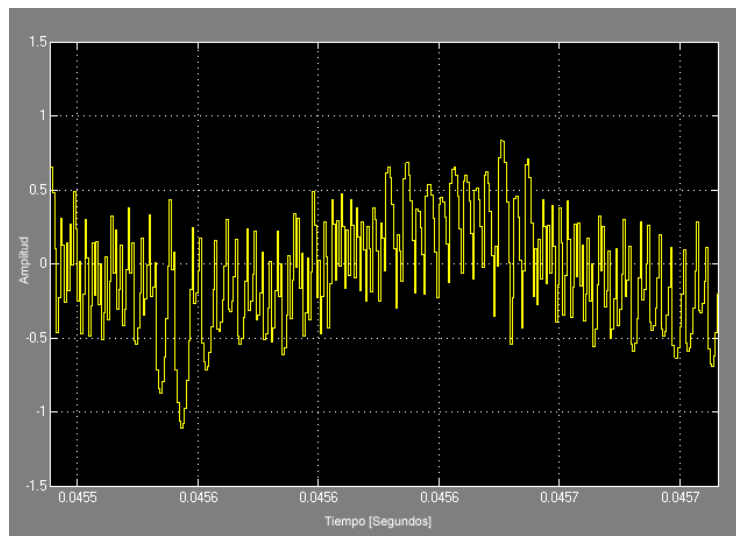
Figura 6.13: Proceso de transformación de la señal digital a analógica.



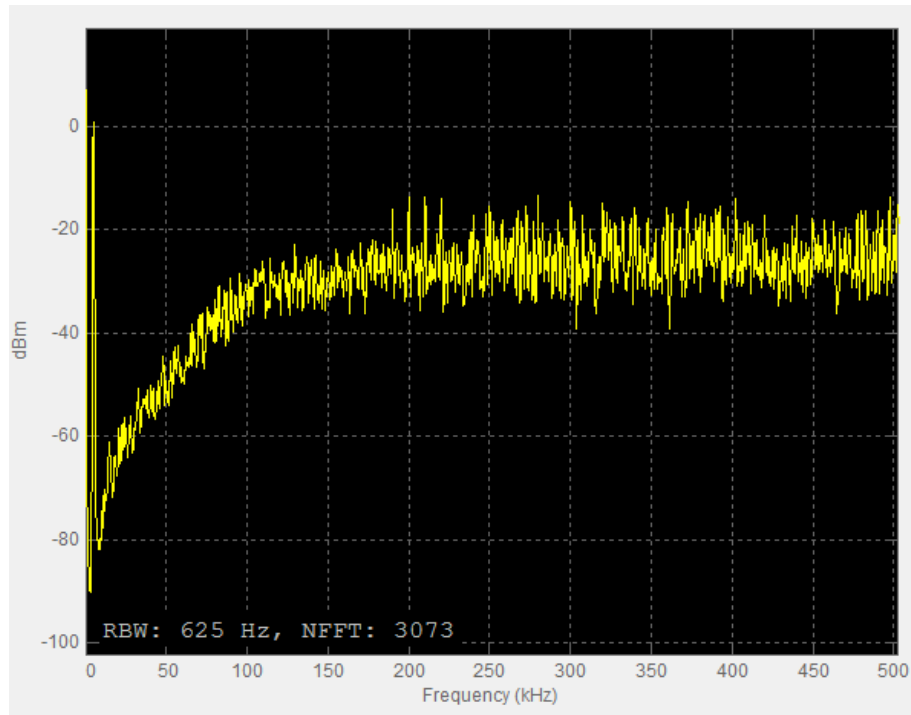
(a) Diferencia entre señal de entrada y salida PDM (señal de error 1).



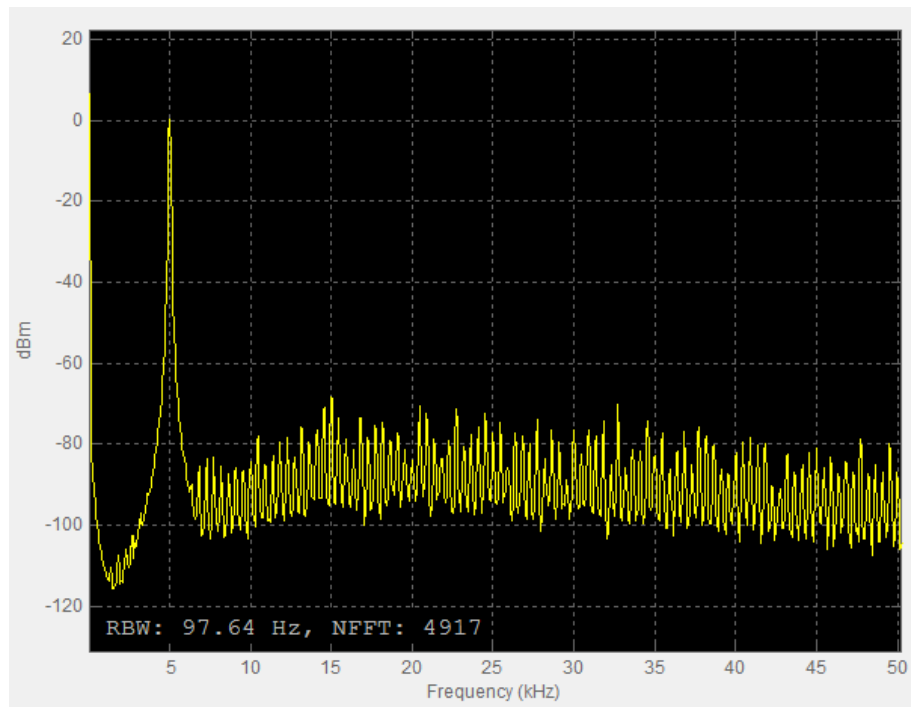
(b) Diferencia entre (señal de error 2).



(c) Entrada al DAC de 1 bit cuando el estímulo es sinusoidal (señal de control).



(a) Contenido espectral de la señal PDM (pre-filtrado)



(b) Contenido espectral de la señal PDM (post-filtrado)

Figura 6.15: Visualización con la herramienta *Spectrum Analyzer* en el dominio frecuencial de las señales analógicas simuladas.

6.4. Filtro Analógico

6.4.1. Diseño y Características

La transferencia normalizada de un filtro pasabajos, con dos polos reales conjugados, es la siguiente:

$$H(s) = \frac{K}{\left(\frac{s}{\omega_o}\right)^2 + \frac{s}{\omega_o Q} + 1} \quad (6.18)$$

Siendo s la variable compleja en el dominio transformado de Laplace, ω_o la frecuencia de resonancia en radianes por segundos, Q el factor de calidad y K la ganancia arbitraria en veces. Los filtros Bessel están optimizados para retardo de grupo constante. Es decir, que el atrazo de fase es lineal con la frecuencia [34]. Como se mencionó anteriormente, esta característica permite que las señales que atraviesan el filtro conserven el aspecto de sus envolventes. Por lo que, dentro del ancho de banda, se espera que el DAC preserve la forma de onda de la señal reconstruida con CS.

Utilizando la tabla que se encuentra en la referencia [34] de un polinomio de Bessel de segundo orden se obtiene los ceros: $z_1 = -1,103 + j0,6368$, $z_1^* = -1,103 - j0,6368$. Esto genera los coeficientes: $a_0 = 1,622$ y $a_1 = 2,206$. Entonces, la realización del filtro pasabajos Bessel es generada por un circuito cuya transferencia sea:

$$H(\omega) = \frac{K}{-\left(\frac{\omega}{2\pi f_c}\right)^2 + 2,206 \frac{j\omega}{2\pi f_c} + 1,622} \quad (6.19)$$

Con f_c la frecuencia de corte (3 dB de atenuación respecto a la ganancia DC) y j la unidad imaginaria. Sin embargo, es necesario normalizar (6.19) para que se corresponda con (6.18). Dividiendo por 1.622 se encuentra:

$$H(\omega) = \frac{K}{-\left(\frac{\omega}{2\pi \times 1,274 f_c}\right)^2 + 1,360 \frac{j\omega}{2\pi f_c} + 1} \quad (6.20)$$

Por lo que, para $f_c = 20$ KHz, $\omega_o = 2\pi \times 1,274 f_c = 160095,56$ [rad/seg] (25480 KHz) y $Q = 0,577$. El siguiente paso es adoptar una arquitectura circuital para implementar (6.20). La Figura 6.16 muestra la topología Sallen Key pasabajos con ganancia unitaria. Básicamente, consiste en un seguidor de tensión que sensa el voltaje de salida y con realimentación positiva mediante el condensador C_2 logra re-ubicar los polos de su transferencia. De esta manera, se puede conseguir prácticamente cualquier valor de Q [35]. A bajas frecuencias, donde C_1 y C_2 presentan altas impedancias, la señal de salida es simplemente igual a la de entrada. Si el amplificador operacional es ideal, a altas frecuencias los condensadores se comportan como corto circuitos. En este último caso, la entrada es enviada a masa.

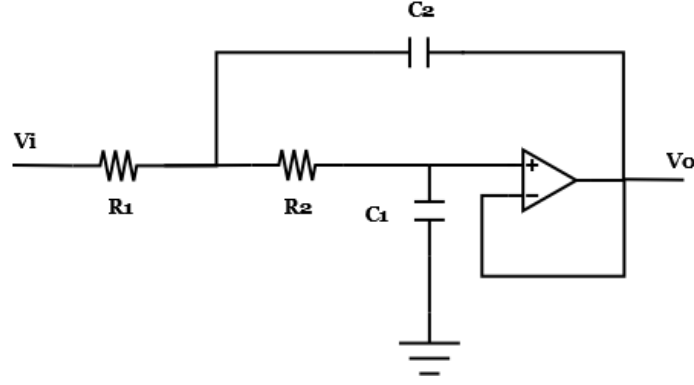


Figura 6.16: Arquitectura Sallen-Key

Desarrollando un modelo de esta arquitectura para encontrar su transferencias y realizar un análisis de su funcionamiento, en la Figura 6.17 se puede discernir los efectos de realimentación positiva y negativa de la salida; junto con el acoplamiento directo no deseado de la entrada. A partir de lo anterior, se desprende el diagrama en bloques de la Figura 6.18. Resolviendo el circuito, se hayan las siguientes transferencias:

$$G_1(s) = \frac{1}{sC_1R_1 + (sC_2R_1 + 1)(sC_1R_2 + 1)} \quad (6.21)$$

$$G_2(s) = \frac{sC_1R_2}{sC_1R_1 + (sC_2R_1 + 1)(sC_1R_2 + 1)} \quad (6.22)$$

$$G_3(s) = \frac{sC_1R_2R_1 + R_1}{z_o[sC_1(R_1 + R_2) + 1] + sC_1R_2R_1 + R_1} \quad (6.23)$$

$$G_4(s) = \frac{sC_2z_o(sC_1R_2 + 1)}{sC_2R_1(sC_1R_2 + 1) + (sC_2z_o + 1)} \quad (6.24)$$

En la simplificación del AO en la Figura 6.17, z_i y z_o representa su impedancia de entrada y salida respectivamente. Además, la transferencia del operacional es $A(s)$. Considerando que z_o es pequeña en comparación con las demás impedancias del circuito, entonces se pueden hacer las aproximaciones $G_3(s) \approx 1$ y $G_4(s) \approx 0$. Partiendo de esas premisas, se realiza el esquema de la Figura 6.19. Para la fabricación del filtro activo, se dispuso del integrado LM358 de la empresa *Texas Instruments*. Este contiene dos operacionales internamente compensados para realimentación unitaria, por lo que, el primer lazo de realimentación negativa es estable (Consultar hoja de datos [36] para más detalles). En esa situación, el ancho de banda con ganancia

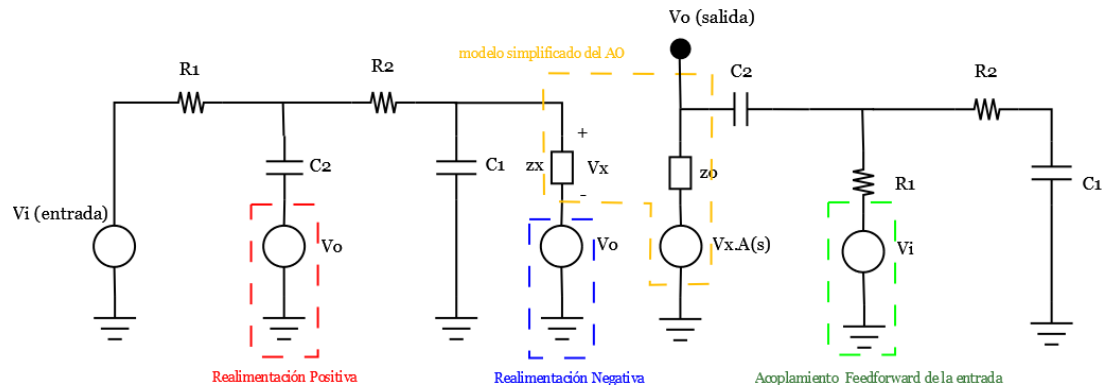


Figura 6.17: Desarrollo del circuito Sallen Key para identificar transferencias.

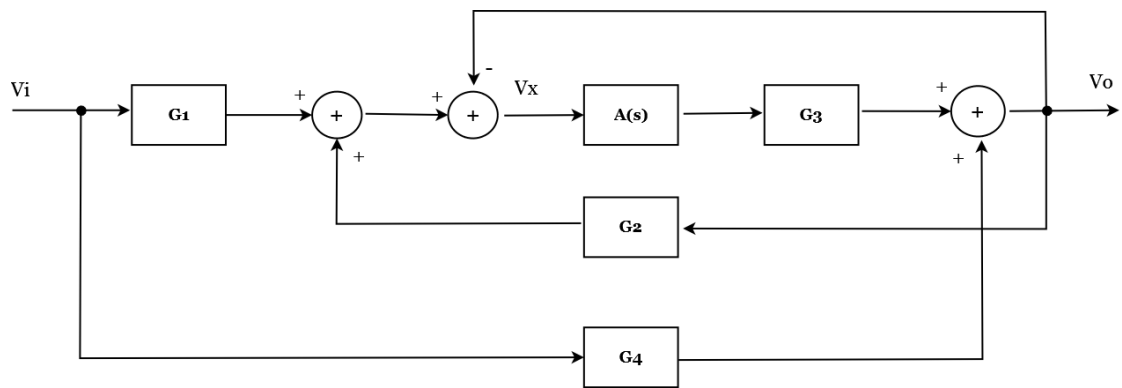


Figura 6.18: Diagrama en bloques de una etapa Sallen-Key

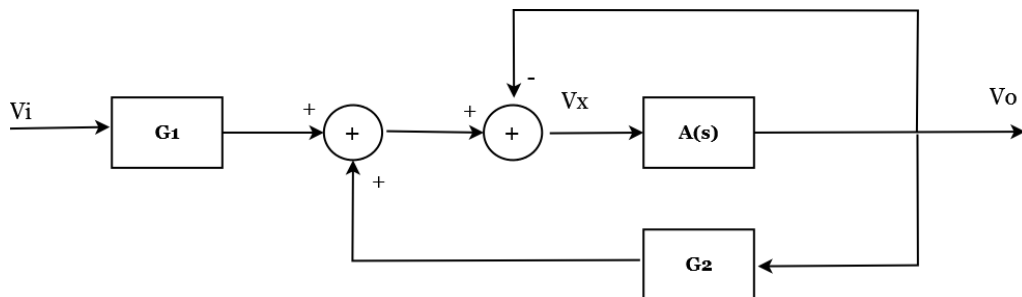


Figura 6.19: Diagrama en bloques simplificado de la etapa Sallen-Key

0dB de los amplificadores se extiende hasta los 0,7 MHz. Es decir, dentro del rango de frecuencias que el operacional se comporta como un seguidor de tensión, la transferencia del filtro se encuentra como:

$$H_f(s) = \frac{G_1(s)}{1 - G_2(s)} = \frac{1}{s^2(C_1C_2R_1R_2) + sC_1(R_1 + R_2) + 1} \quad (6.25)$$

La estabilidad se analiza con la transferencia $G_2(s)$ teniendo en cuenta realimentación positiva. Generando la correspondencia entre (6.25) y (6.20) se encuentra las ecuaciones de diseño:

$$C_1C_2R_1R_2 = \left(\frac{0,7849}{2\pi f_c} \right)^2 \quad (6.26)$$

$$C_1(R_1 + R_2) = \frac{1,360}{2\pi f_c}$$

Se debe tener en cuenta que f_c es la frecuencia de corte para una etapa de segundo orden (40 db/década). Como se necesita que el filtro sea de cuarto orden (80 db/década), hay que utilizar dos etapas Sallen-Key en cascada. En este caso, la atenuación total a 20 KHz tiene que ser de 3 dB. Para ello se utilizó la herramienta online [37] que obtiene los valores de los componentes⁶ ingresando el tipo (Bessel), la frecuencia de corte y el orden del filtro:

Componente	Etapla 1	Etapla 2
R_1	2k5 Ω	3k7 Ω
R_2	8k1 Ω	9k4 Ω
C_1	1 nf	470 pf
C_2	1.5 nf	1 nf

Cuadro 6.2: Componentes necesarios para armar el Filtro Bessel (Figura 6.20) con $f_c = 20$ KHz y 80 dB/dec de atenuación.

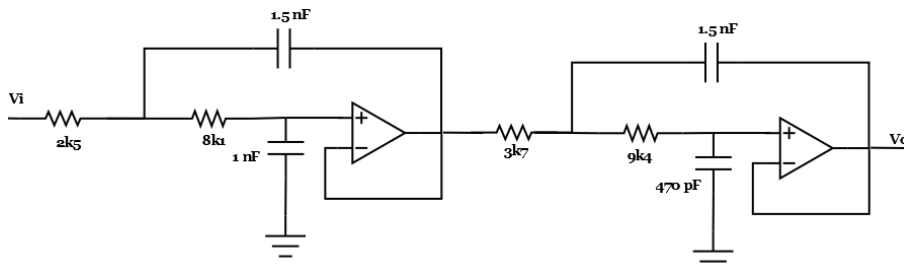


Figura 6.20: Filtro activo con dos etapas Sallen Key en cascada. [37].

⁶Se calcularon las resistencias a partir de valores comerciales de capacitores.

6.4.2. Efectos del Amplificador Operacional No Ideal

Analizando ahora el efecto de la impedancia de entrada y transferencia del AO, se encuentra que a altas frecuencias el filtro en vez de atenuar a 40 db/década comienza a tener un comportamiento pasa-altos. A frecuencias mucho mayores que la de corte $G_1(s)$ y $G_2(s) \rightarrow 0$; pero G_4 , que inicialmente era despreciable, comienza a tener preponderancia. Estas características se resumen en el diagrama en bloques de la Figura 6.21. La transferencia de acoplamiento de la entrada resulta:

$$\left. \frac{V_o}{V_1} \right|_{forward} \approx \frac{G_4(s)}{1 + A(s)G_3(s)} \approx \frac{z_o}{R_1(1 + A(s))} \quad (6.27)$$

Donde, a partir de las especificaciones provistas por la hoja de datos [36], se asume la respuesta $A(s)$ del operacional:

$$A(s) = \frac{100000}{\left(\frac{s}{2\pi 20} + 1\right)} \quad (6.28)$$

Combinando (6.27) con (6.25) se encuentra la transferencia real de una etapa de segundo orden (Figura 6.22). Como puede verse, utilizar sólo un bloque Sallen-Key no es suficiente para el filtrado de la señal PDM ya que la atenuación máxima en altas frecuencias es de 40 dB. (se requiere por lo menos 60 dB). Por lo tanto, agregando otra instancia del filtro en cascada no solo se consigue mayor *Roll-Off*⁷, sino que ahora la atenuación máxima es de 80 dB. Como aclaración, a partir de 100 MHz la inductancia parásita comienza a jugar un papel en la respuesta en frecuencia. También, la capacitancia entre pista aportará a la atenuación [35]. Ambos efectos no son contemplados en el análisis.

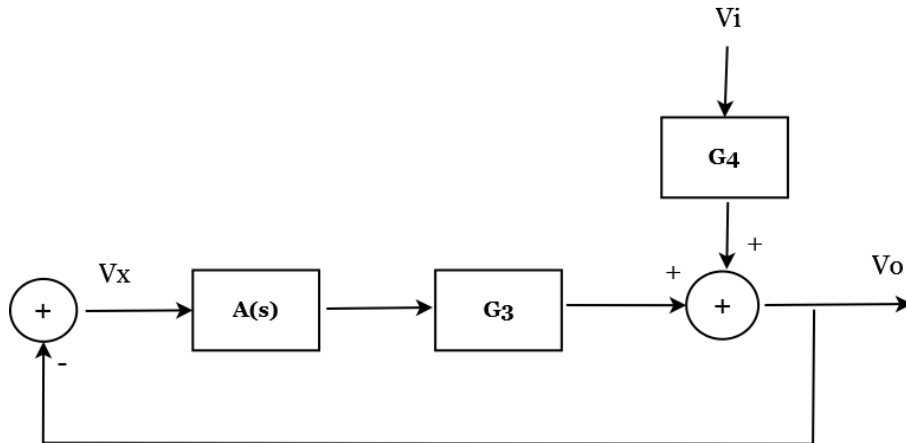


Figura 6.21: Diagrama en bloques para altas frecuencias de la etapa Sallen-Key

⁷Pendiente de atenuación.

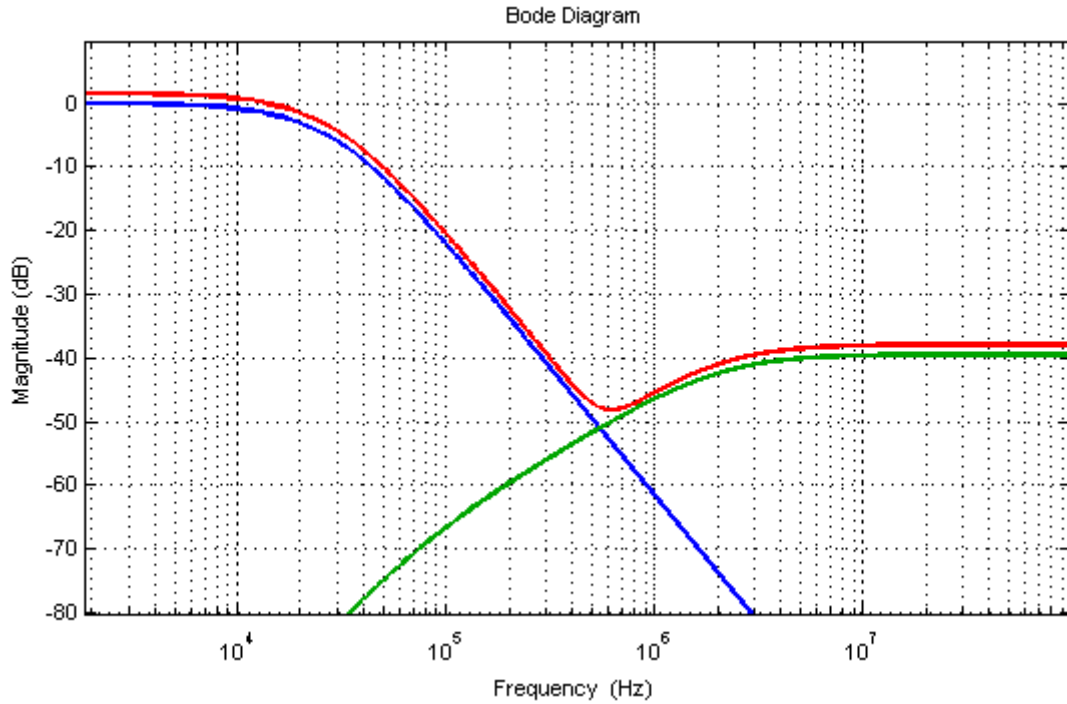
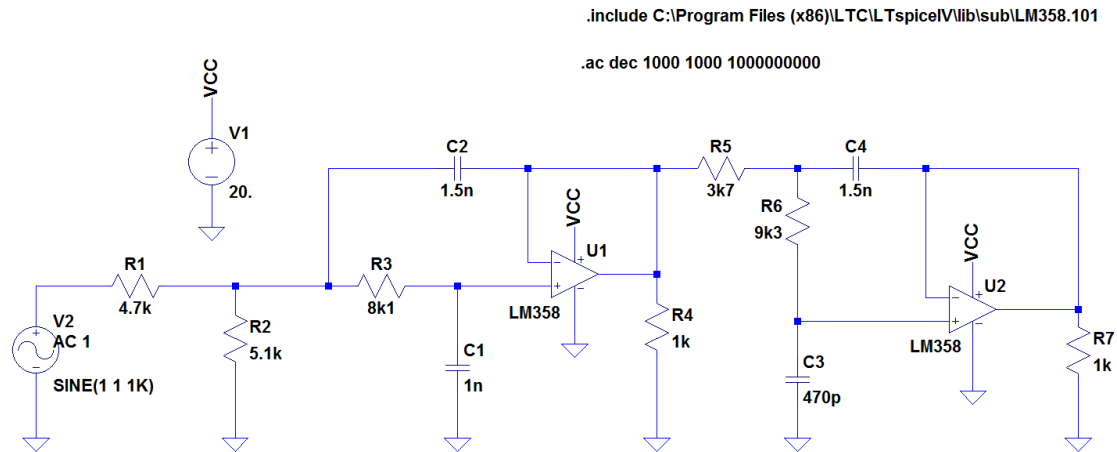


Figura 6.22: Transferencia Real del Filtro Sallen Key considerando z_o y $A(s)$. En azul se aprecia la respuesta en frecuencia ideal, en verde el efecto feed-forward no deseado y en rojo la transferencia total.

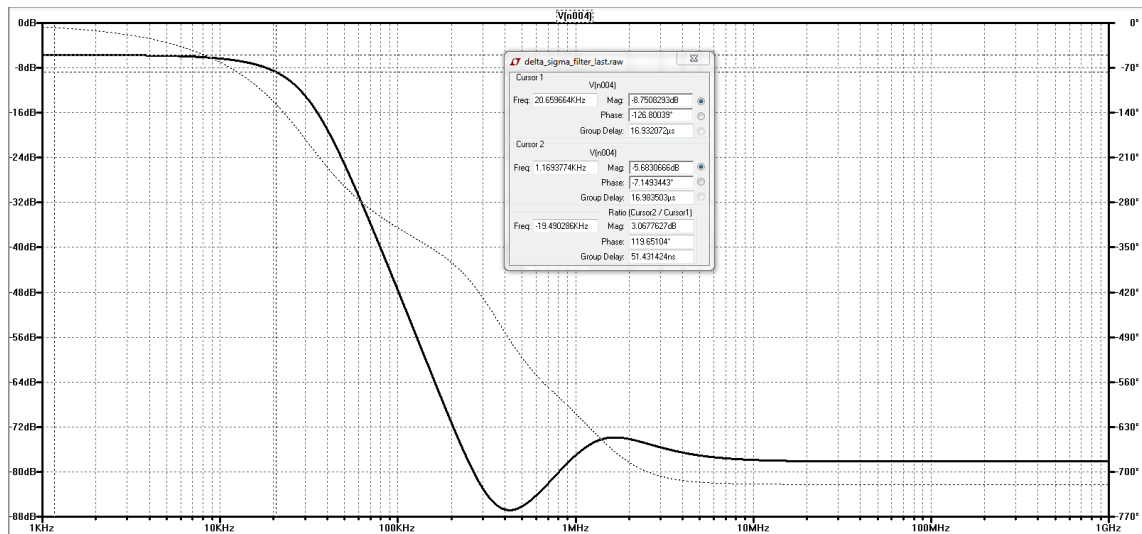
6.4.3. Simulación en LTspice

Previo a la construcción del filtro activo, se realizó una simulación circuital en el programa LTspice (Ver 4.23k). El esquemático utilizado es el de la Figura (6.23a), donde el voltage de alimentación (VCC) para los AO es 3.3 Volts con fuente simple. No obstante, se puede reconocer algunas diferencias con la Figura (6.20). En primera instancia, se reemplazó la resistencia de 2k5 a la entrada por dos resistencias de 4k7 y 5k1. Esto se debe a que el PMOD de la placa Zedboard sólo puede entregar tensiones de 3.3 o 0 Volts, por lo que se necesita un divisor resistivo a la entrada para evitar saturación. Aplicando el teorema de Thévenin a la tensión de entrada junto con R_1 y R_2 (Figura 6.23a) se obtiene la topología original con la salvedad que V_i se atenúa en 0.52 veces ó 5.7 dB. Por lo tanto, la amplitud máxima de la señal de entrada equivalente es de 1.71 Volts y se evita la saturación del operacional.⁸

⁸En la configuración empleada, el AO puede excursionar entre 0 a 2.3 Volts [36].



(a) Esquemático final del filtro activo.



(b) Diagrama de bode del filtro generado con simulación.

Figura 6.23: Simulación en el programa LTSpice (Versión 4.23k)

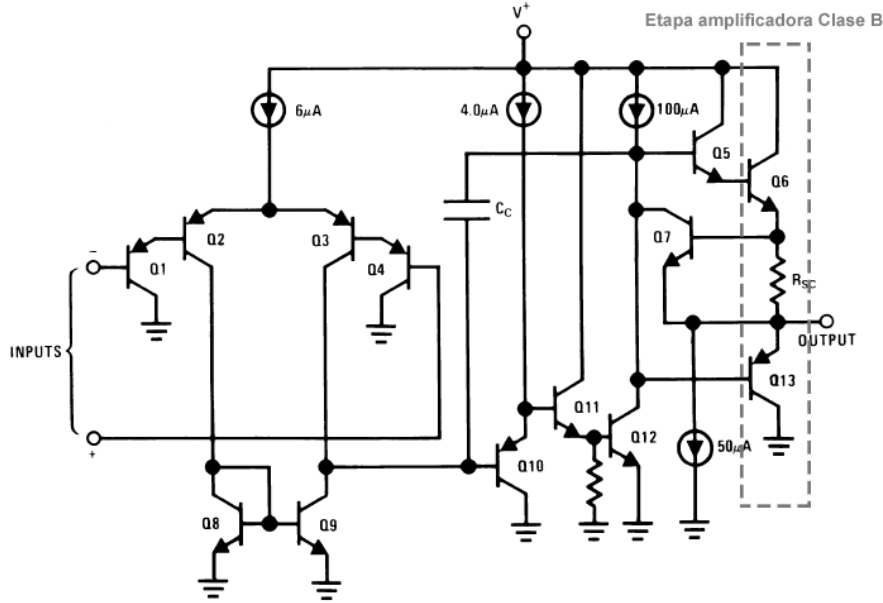


Figura 6.24: Esquemático simplificado de un AO del integrado LM358 (Fuente [36]).

Otra modificación al circuito original es el agregado de resistores a masa de $1k\Omega$ a la salida de cada amplificador. Esto es necesario para generar una corriente de polarización en la etapa final de transistores que funcionan en clase B (Figura 6.24) y prevenir distorsión por cruce. De esta manera, los únicos transistores que conducen en la última etapa son Q_5 y Q_6 . Si no se resolvía este inconveniente, la SNR hubiese empeorado bastante debido a la generación de picos de tensión en la conmutación de alta velocidad del PDM.

Para la simulación del diagrama de bode, se usó el modelo SPICE del LM358 descargado de la página web de Texas Instruments [38]. La fuente que realiza el barrido de frecuencia es $V2 (AC 1)$ y la salida de tensión se mide en R_7 . La transferencia del circuito se presenta en la Figura 6.23b, donde la respuesta de amplitud se muestra en línea gruesa y la respuesta en fase en línea puntuada. Se puede observar en el cuadro de dialogo agregado en la misma Figura que el retardo de grupo es aproximadamente $17 \mu Seg$ dentro de la banda de paso (20 KHz de frecuencia de corte). Asimismo, la atenuación a bajas frecuencias es de 5.68 dB. Finalmente, el efecto de acoplamiento de la entrada descrito anteriormente se comienza a visualizar a partir de los 400 KHz; terminando con una atenuación constante de 78 dB. Como conclusión, la simulación permitió verificar que el diseño cumple con los requerimientos de filtrado.

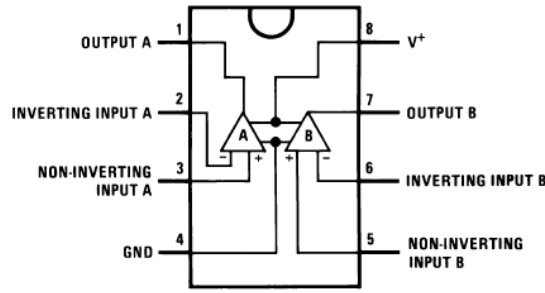


Figura 6.25: Vista superior del encapsulado LMC. Presenta dos operacionales LM358 (A y B) (Fuente [36]).

6.4.4. Fabricación del Periférico

El filtro activo se armó en una placa perforada utilizando resistores de carbon (5 % de tolerancia) y capacitores cerámicos multicapa (10 % de tolerancia). Como algunos componentes no están disponibles comercialmente, se utilizaron combinaciones en serie para lograr los valores de resistencia requeridos (Figura 6.26). El integrado que provee los dos operacionales es el de la Figura 6.25. Cada AO consume 0,5 mA de corriente independientemente de la tensión de alimentación [36]. Conjuntamente, según las especificaciones de los conectores PMOD [25], la placa puede suministrar hasta 100 mA. En consecuencia, no existen problemas en el consumo de corriente. Por otra parte, el rango de tensión de fuente que admite los AO esta comprendido desde 3 hasta 32 Volts. [36].

En la Figura 6.27 se muestra el filtro fabricado. Se puede contemplar sus dimensiones junto con el conector de 12 pines incorporado para enchufar a la placa de desarrollo. Los pines 1, 2, 11 y 12 son utilizados para alimentación (+3.3 V y GND). El pin 6 envía la señal PDM del SoC al filtro. Además, el periférico cuenta con dos cables de salida, uno para la señal analógica (SGN) y otro de referencia (GND).

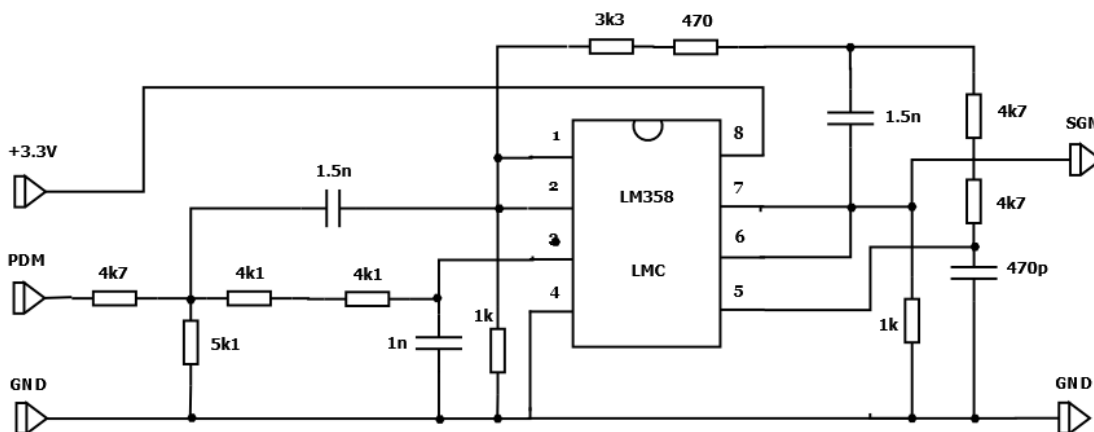


Figura 6.26: Esquemático del periférico desarrollado.

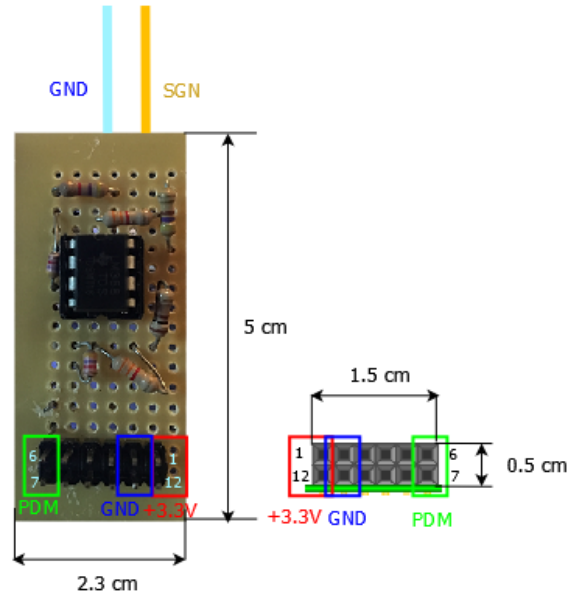


Figura 6.27: Filtro activo en placa perforada (izquierda) y ficha PMOD de la placa de desarrollo (derecha). El periférico presenta conector macho de 12 pines compatibles con la Zedboard.

6.4.5. Mediciones Experimentales

El banco de medición usado para determinar la respuesta del filtro es el de la Figura 6.28. Se tuvo a disposición un generador de funciones *Hewlett Packard* modelo 33120a para ingresar al circuito con señales de prueba sinusoidales de valor medio 0,6 V y amplitud $1 V_{pp}$. Además, con un osciloscopio digital marca *Tektronix* 1052B se visualizaron simultáneamente la tensión de entrada y de salida. La fuente de alimentación fue provista por la placa Zedboard a través de un PMOD.

Variando la frecuencia de la señal de entrada desde 100 Hz hasta 100 KHz, se realizó la Tabla 6.3 a partir de valores de voltaje y tiempo leídos en la pantalla del osciloscopio.

Frecuencia [KHz]	Atenuación [dB]	Retardo [uSeg]
.1	-5.72	
.5	-5.7	
1	-5.85	
5	-6.03	
10	-6.72	16
15	-7.55	15.8
20	-8.94	16
40	-19.01	N/A
100	-44.38	N/A

Cuadro 6.3: Valores medidos de atenuación y retardo en el filtro a distintas frecuencias.

Para ejemplificar cómo se efectuaron las mediciones, la Figura 6.29 muestra en el

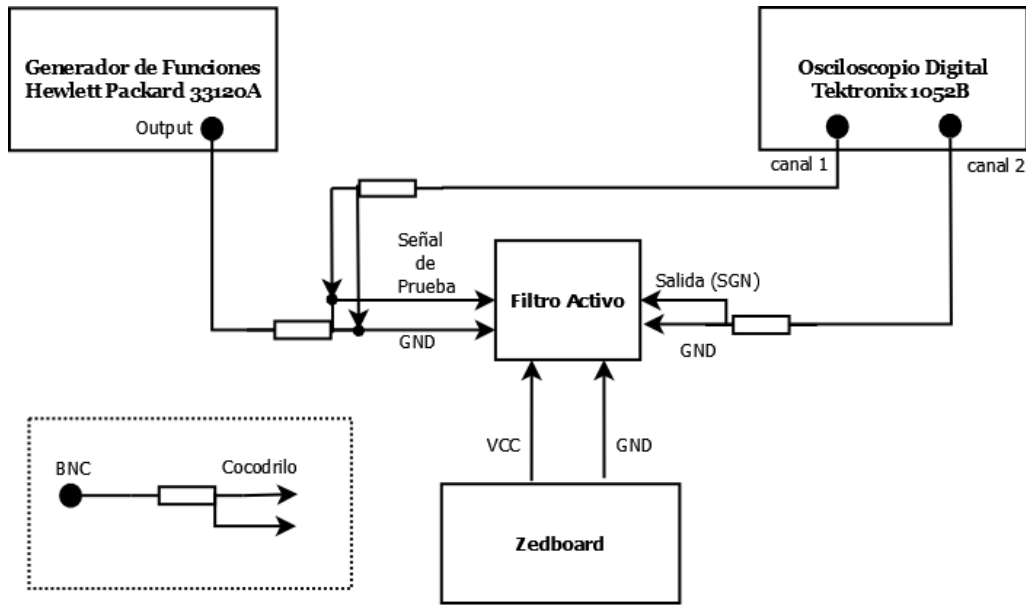
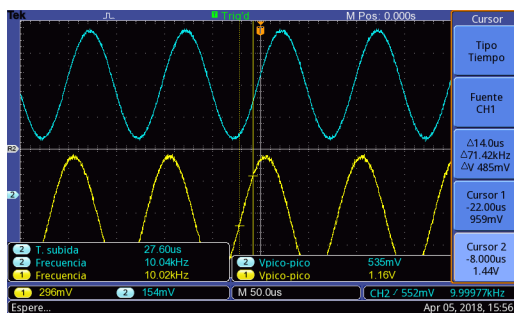


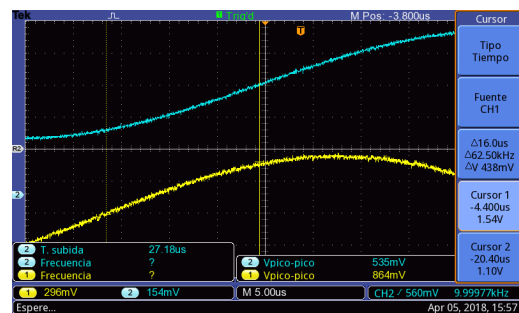
Figura 6.28: Arreglo experimental para evaluar el comportamiento del periférico construido. En el cuadro punteado se encuentra el símbolo que representa al conector BNC/doble cocodrilo que utilizan los instrumentos.

caso de una sinusoidal de 10 KHz la medición automática de V_{pp} en la entrada (amarillo) es igual a 1.6 Volts mientras que a la salida (azul) vale 0.535 Volts (Subfigura 6.29a). Después, efectuando la operación $20 \cdot \log(0,535/1,16)$ se encuentra que la atenuación a esa frecuencia es -6.72 dB. Adicionalmente, para determinar el retardo se expande la base de tiempo (en este caso a 5 μ Seg por división) y con dos cursores de tiempo se mide la diferencia temporal entre los cruces por cero de ambas señales (Subfigura 6.29b).

Con los datos experimentales, se procedió a realizar el diagrama de Bode de la Figura 6.30. El error adoptado para la amplitud es de $\pm 0,15$ dB ya que se corresponde aproximadamente con una incerteza del 1% en las lecturas de voltaje. Se utilizó un gráfico doble-log para visualizar mejor los rangos. Según puede apreciar-



(a) Medición de amplitud.



(b) Medición de retardo.

Figura 6.29: Captura de pantalla en el osciloscopio digital mientras se mensuraba la respuesta de amplitud y fase del filtro. La entrada se muestra en amarillo y la salida en azul.

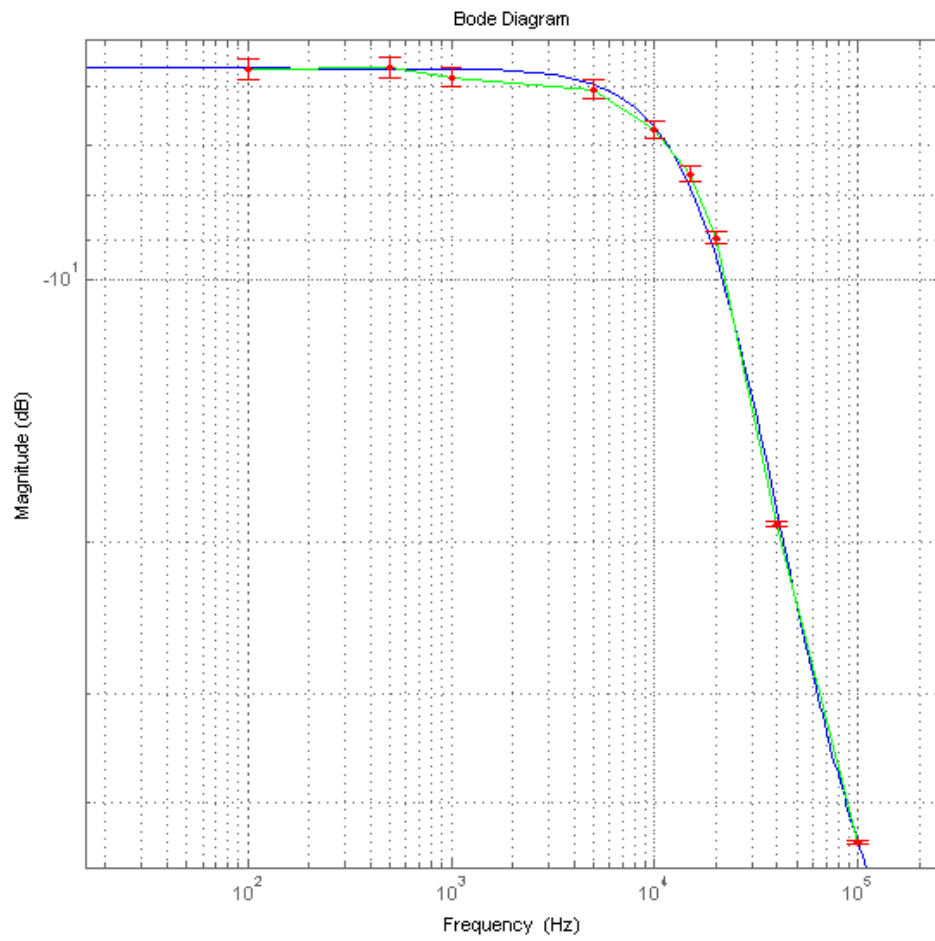


Figura 6.30: Respuesta en amplitud del filtro esperada (línea azul), junto con las mediciones experimentales (puntos rojos) y su interpolación de primer orden (línea verde).

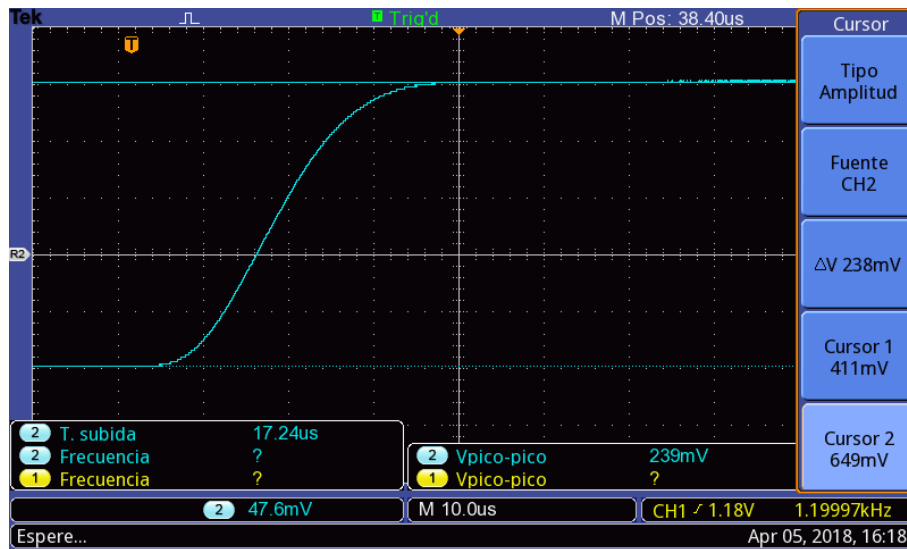


Figura 6.31: Respuesta al escalón experimental del filtro Bessel.

se, la atenuación real del filtro sigue la tendencia establecida por la respuesta teórica. La disminución en amplitud a 20 KHz es de -3.24 dB, por lo que, el error en dB en esa frecuencia es del 6 %; ya que se esperaba que fuese de -3 dB (frecuencia de corte). Como conclusión, se puede enunciar que el filtro fabricado cumple con sus especificaciones de diseño con un error menor al 10 %.

Ahora, sustituyendo la señal de prueba por una onda cuadrada de amplitud 0.5 Volts, valor medio 0.25 Volts y frecuencia 1 KHz se observó la respuesta al escalón del periférico (Figura 6.31). Sus parámetros característicos son presentados a continuación:

- **Tiempo de crecimiento:** 17.24 μ Seg.
- **Tiempo de Demora:** 14 μ Seg.
- **Tiempo de Establecimiento:** 32 μ Seg.
- **Tipo de respuesta:** Sobreamortiguada.

El tiempo de crecimiento está medido desde el 10 % hasta el 90 % del valor final, mientras que el tiempo de demora se establece del 0 % al 50 % y el tiempo de establecimiento del 0 % al 99 %. Se puede reconocer que el filtro activo presenta una respuesta suave y libre de oscilaciones. Recordando que la estabilidad de una etapa Sallen Key se analiza con la transferencia G_2 (ecuación 6.22) se procede a realizar los gráficos de la Figura 6.32. Usando el criterio de estabilidad de Nyquist se obtiene $N = Z - P = 0$ (nuevamente se verifica la estabilidad). Además, el margen de fase resulta ser infinito ya que la red de alimentación es pasiva; por lo tanto la magnitud de su diagrama de Nyquist nunca alcanza la unidad. Esto se corresponde con la ausencia de *Overshoot*⁹ en la respuesta de la Figura 6.31. Es interesante advertir

⁹Desviación máxima del pico en amplitud de la respuesta al escalón con respecto al valor final esperado [39].

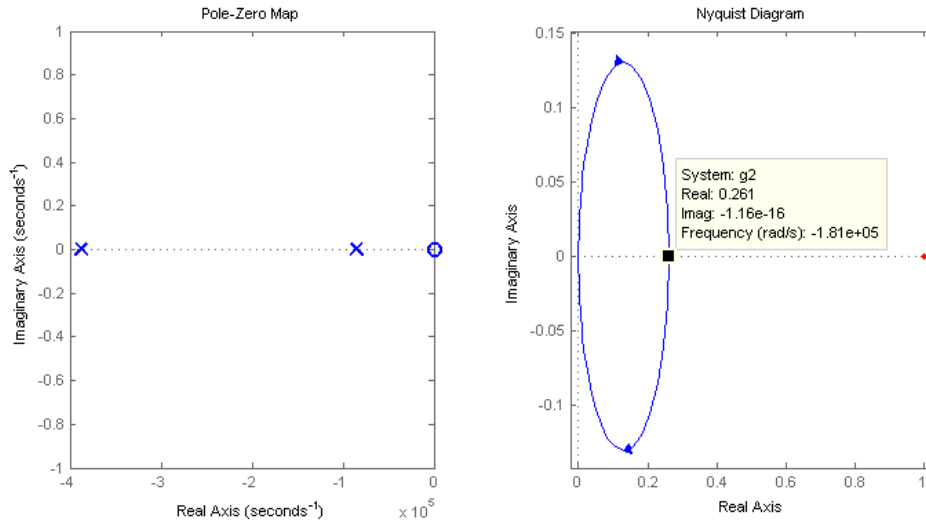


Figura 6.32: Polos y ceros de la transferencia de realimentación $G_2(s)$ (izquierda) junto con su diagrama de Nyquist (derecha) para el caso del filtro Bessel. Como se trata de realimentación positiva, el valor de referencia es $S = 1$ (punto rojo).

que el tiempo de crecimiento es similar al retardo constante teórico dentro de la banda de paso ($17 \mu\text{Seg}$). También, si consideramos al tiempo de establecimiento como medio período de una onda sinusoidal se obtiene una frecuencia máxima de 15625 Hz (levemente inferior al ancho de banda de 20000 Hz).

6.5. Interpolador

6.5.1. Principios básicos

La interpolación consiste en dos partes: el aumento de la tasa de muestreo agregando ceros entre muestras (*zero-stuffing* en Inglés) y un filtrado digital pasabajos. El proceso es ilustrado en la Figura 6.33 en el dominio temporal y frecuencial con un ejemplo de interpolación por dos. La primer secuencia $x[n]$ de longitud $N \in \mathbb{N}$ contiene la información de su espectro acotado en el intervalo $[-\pi, \pi]$ de frecuencia, luego se repite periódicamente cada 2π ya que la señal es discreta. Al agregar un cero entre muestras, se obtiene la secuencia $x[m]$. El efecto de *zero-stuffing* en el dominio transformado es el escalamiento del eje de frecuencia por el factor de sobremuestreo M . Es decir, para el caso del ejemplo ($M = 2$), ahora la información de la señal está contenida en $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Después, se aplica un filtro digital con ganancia¹⁰ M en $\omega \in [0, \frac{\pi}{2}]$ y 0 para $\omega \in [\frac{\pi}{2}, \pi]$. Como resultado, se consigue la secuencia $x^*[m]$ de longitud $2N$, siendo esta la versión sobremuestreada de $x[n]$.

¹⁰El espectro se ve disminuido M veces cuando se realiza la interpolación y en consecuencia el filtro debe compensar esta atenuación. Consultar [40] para la demostración matemática.

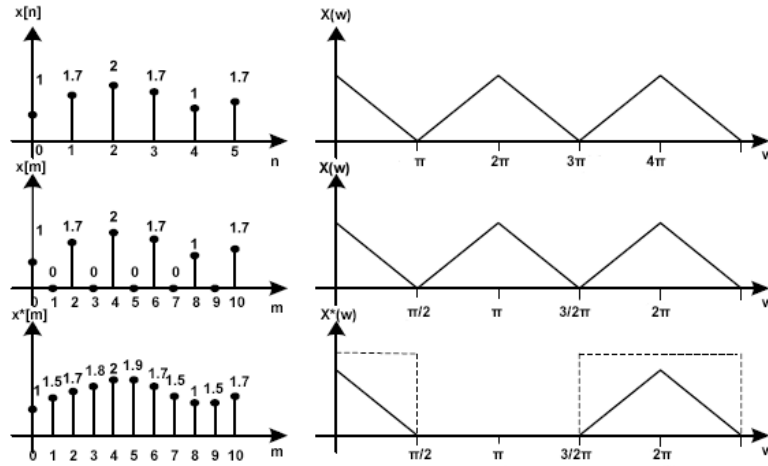


Figura 6.33: Proceso de interpolación digital en el dominio del tiempo discreto (izquierda) y en frecuencia (derecha). Secuencia original (arriba), con *zero-stuffing* (medio) y luego del filtrado (abajo).

La respuesta al impulso del filtro pasabajos ideal es:

$$H_M[m] = M \times \text{Sinc} \left[m \frac{1}{M} \right], \quad m \in [-\infty, \infty] \quad (6.29)$$

Por supuesto que $H_M[m]$ con longitud infinita no es realizable, siendo así que los filtros reales son diseñados en base a tolerancias respecto a la desviación esta característica deseada. Los parámetros a considerar son el ripple en la banda de paso, la atenuación en la banda de rechazo y el ancho de la banda de transición. También, como en el caso analógico, se desea retardo de grupo constante; debido a lo cual se utilizan filtros FIR de respuesta simétrica. Una primera aproximación para la interpolación con $M = 32$ requerida previo al modulador $\Delta\Sigma$ es utilizando un filtro con las características que se muestran en la Figura 6.34. Las especificaciones (atenuación 0.001 dB en la banda de paso y 80 dB en la banda de rechazo) son realistas para la aplicación del DAC. Como puede verse, debido a la elevada tasa de sobremuestreo, la banda de paso junto con la de transición son bastantes angostas en comparación con la banda de Nyquist. Esto produce que el orden necesario sea de 5482. Cuando se tienen demasiados coeficientes como en este caso, la memoria requerida junto a las multiplicaciones por muestras de entrada vuelven ineficiente e impráctica la implementación del interpolador. Afortunadamente, existen estrategias para simplificar la complejidad considerablemente [41].

Una forma de reducir el procesamiento es utilizando varios interpoladores en cascada (Figura 6.35). Para comprender la ventaja de esta implementación, la Figura 6.36 presenta la forma de los filtros en las primeras tres etapas. A medida que avanza la cadena de sobremuestreo, se precisa una banda de paso menos estrecha. Esto se debe a que el espectro periódico de la señal discreta se aleja luego de atravesar cada bloque. La Figura 6.37 muestra más detalladamente cómo es el proceso para aumentar la frecuencia de muestreo.

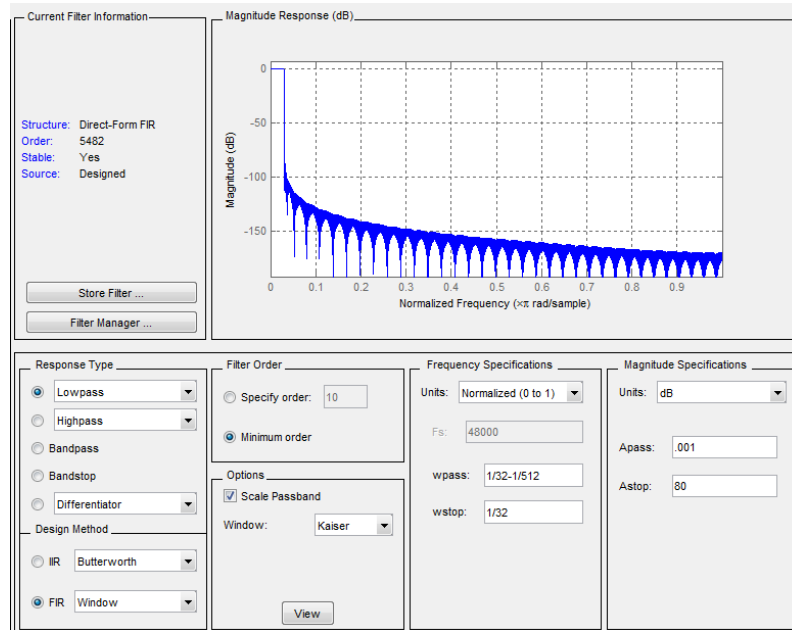


Figura 6.34: Filtro para interpolación con $M = 32$ obtenido con la herramienta *Filter Design & Analysis* del entorno Matlab R2013a.

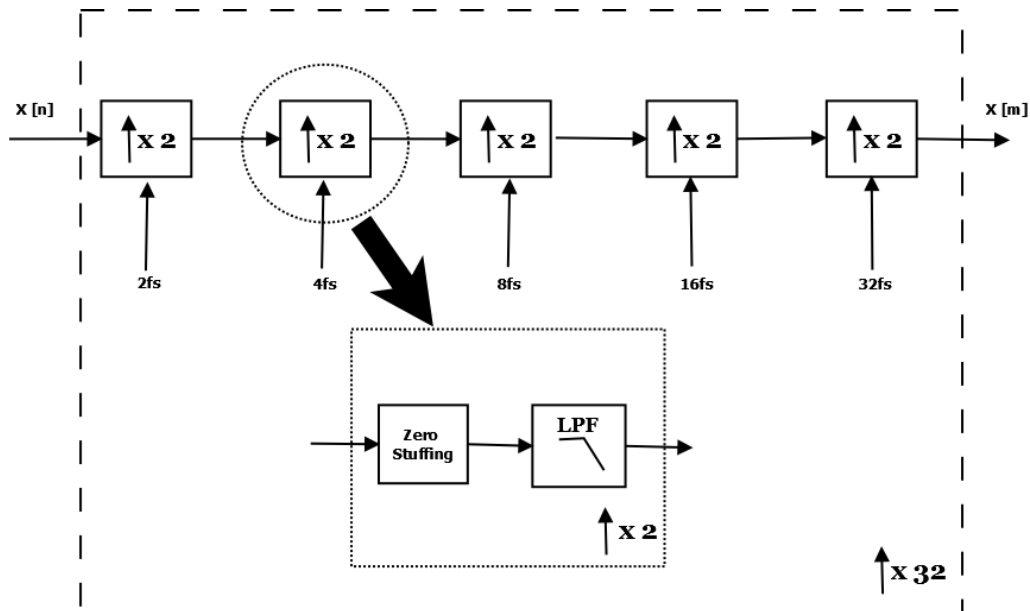


Figura 6.35: Interpolación x32 usando cinco bloques x2 en cascada.

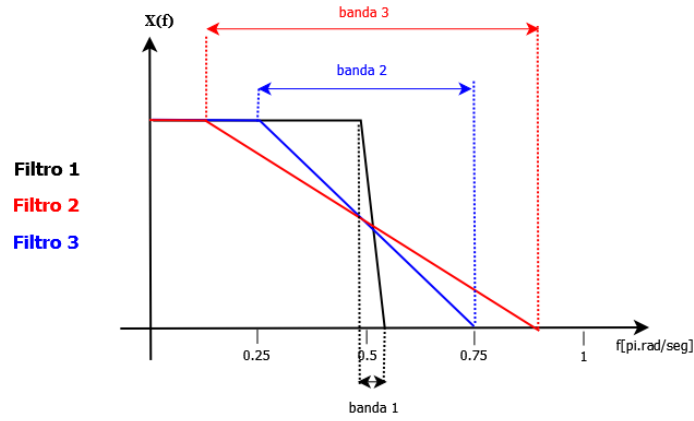


Figura 6.36: Envolventes de los filtros digitales en la interpolación sucesiva.

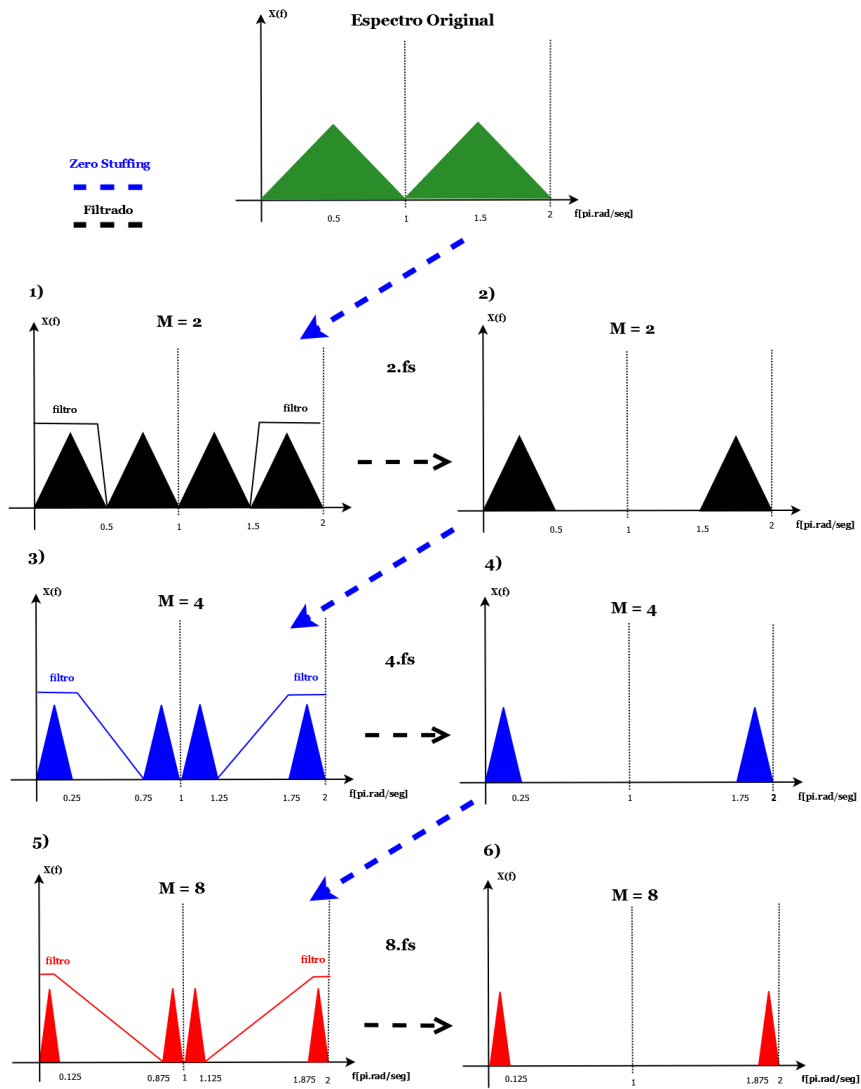


Figura 6.37: Transformación de la señal de entrada en la cadena de sobremuestreo. Cada vez que se realiza *zero-stuffing*, el eje de frecuencia sufre un escalamiento por dos; generando un espectro imagen en las altas frecuencias. Como la distancia entre picos aumenta, los requerimientos de filtrado pasabajos se vuelven menos estrictos.

6.5.2. Filtros FIR de Media Banda

La respuesta al impulso de los filtros FIR de longitud impar y simétricos logran que, mediante convolución discreta con una secuencia $x[n]$, presenten la propiedad de fase lineal [40]. Debido a lo cual, la cantidad de coeficientes a almacenar pueden ser reducidos a la mitad más uno aprovechando la propiedad de simetría. En este caso, si bien se disminuyen los requerimientos de memoria, la cantidad de bloques de retardos y sumadores permanece intacta. La estructura a implementar en VHDL para realizar la convolución se muestra en la Figura 6.38. En ella se puede observar cómo se comparten sumadores y multiplicadores con coeficientes idénticos.

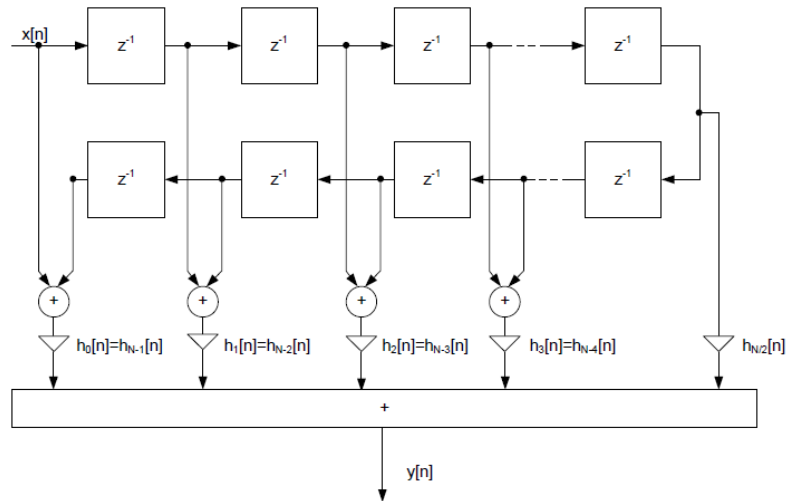


Figura 6.38: Arquitectura para la realización de filtros FIR simétricos

Por otra parte, cuando la respuesta *en frecuencia* es simétrica, o sea, la banda de transición se encuentra exactamente a la mitad de la frecuencia de Nyquist, las muestras impares resultan ser cero (Figura 6.39). Estos filtros son considerados una sub-clases de FIR y se denominan de *Media Banda*. En consecuencia, la cantidad de valores de la respuesta al impulso a utilizar se reducen a aproximadamente un cuarto del total. A su vez, es condición necesaria que el ripple de la banda de atenuación y de paso sean iguales para lograr esta propiedad.

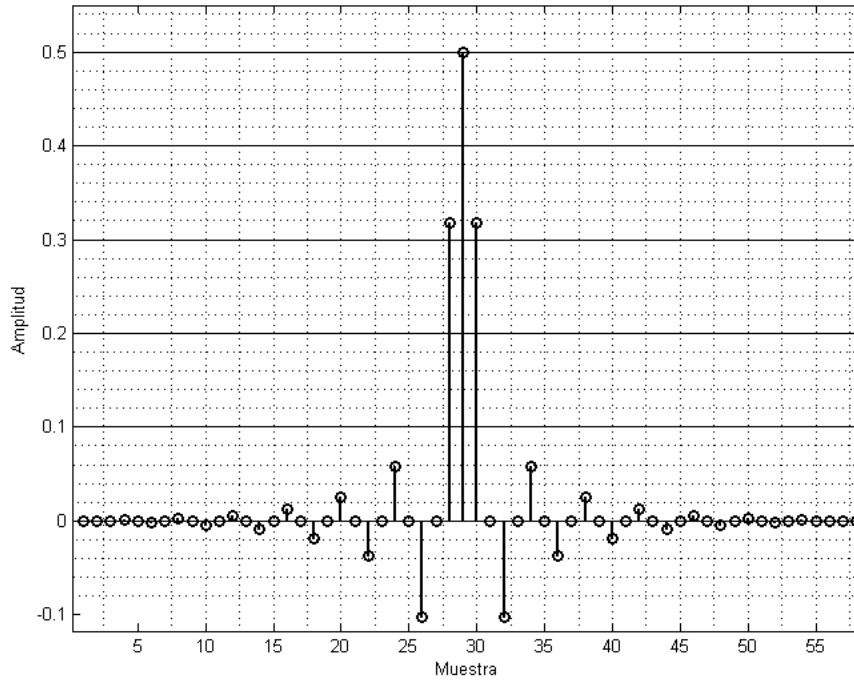


Figura 6.39: Respuesta al impulso de un filtro FIR simétrico de media banda con longitud $L=59$. Todas las muestras impares (excepto $\frac{L-1}{2} = 29$) son cero. Se requiere almacenar sólo 15 coeficientes en este caso.

6.5.3. Realización Polifásica

El hecho que la señal requiera el agregado de ceros al interpolar mediante el *zero-stuffing* puede ser aprovechado usando la *descomposición polifásica* [41]. Para ilustrar este principio, se muestra a continuación una respuesta al impulso $H(z)$ de longitud $L = 7$:

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} + h[6]z^{-6} \quad (6.30)$$

Reagrupando (6.30) convenientemente en potencias de z^{-2} se logra¹¹:

$$H(z) = (h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6}) + z^{-1}(h[1] + h[3]z^{-2} + h[5]z^{-4}) \quad (6.31)$$

Considerando un filtro de media banda se tiene que $h[1] = h[3] = 0$, entonces:

¹¹En el caso general, conviene reagrupar en potencias de z^{-m} ; donde m es el factor de interpolación.

$$H(z) = (h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6}) + z^{-1}(h[5]z^{-4}) \quad (6.32)$$

Llamamos al primer paréntesis de (6.32) E_0 y al segundo E_1 . Ergo, la descomposición polifásica de la respuesta al impulso es $H(z) = E_0 + z^{-1}E_1$. Volviendo al caso de la implementación en cascada con bloques de sobremuestreo x2, la arquitectura que realiza la interpolación con el filtrado polifásico es el de la Figura 6.40. Se sabe que, en este caso, una de cada dos muestras que ingresan al filtro es cero. Por lo tanto, podemos reemplazar las potencias z^{-2} en cada término E_i por z^{-1} . Esto permite utilizar cada rama de la Figura 6.40 de manera concurrente y con la misma tasa de muestreo que la señal $x[n]$ de entrada. Más aún, el retardo y el sumador puede ser sustituidos por una llave que conmuta entre E_0 y E_1 al doble de frecuencia y obtener así dos muestras por cada una de entrada; logrando el efecto de sobremuestreo.

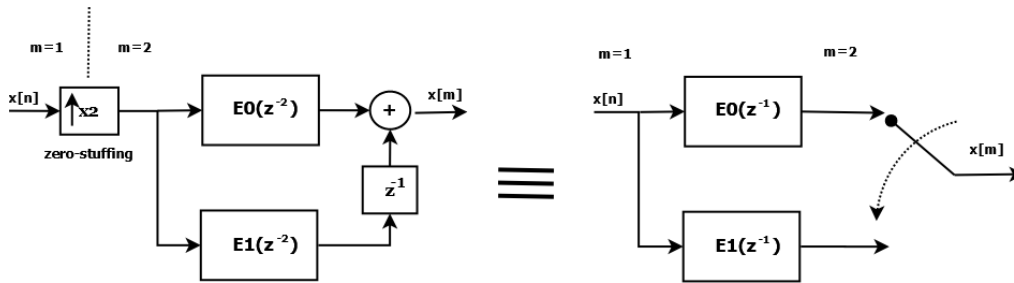


Figura 6.40: Interpolador x2 con filtro de estructura polifásica (izquierda), junto con su realización equivalente más eficiente (derecha).

6.5.4. Estructura de conmutación

Habiendo presentado cómo disminuir la cantidad necesaria de coeficientes en los filtros, junto con la forma polifásica que evita el agregado de ceros, se muestra ahora el interpolador x32 realizado para el DAC. Utilizando cinco estructuras de conmutación en cascada, como exhibe la Figura 6.41, se logra obtener una réplica 32 veces sobremuestreada de la señal $x[n]$. En cada etapa, la frecuencia de muestreo f_s aumenta el doble, ya que cada llave conmuta dos veces más rápido que la anterior. Luego de un transitorio inicial, se genera la señal de salida $x[m]$ manifestando un retardo temporal en relación a $x[n]$.

Las especificaciones utilizadas en los filtros fueron 80 dB de atenuación en la banda de rechazo y ripple de 0.001 dB en la banda de paso. La función *firhalfband()* disponible en Matlab R2013a permitió generar los coeficientes FIR simétricos y de media banda. Las frecuencias digital de corte resultaron $0, 5\pi$ (filtro 1), $0, 25\pi$ (filtro 2), $0, 125\pi$ (filtro 3) y $0, 0625\pi$ (filtro 4 y 5). En la tabla 6.4 se encuentran el orden N resultante de cada filtro y los valores de su respuesta al impulso $h[n]$ relevantes. Los bloques $E_0(z^{-1})$ se realizaron con la estructura de la Figura 6.38 empleando sólo las

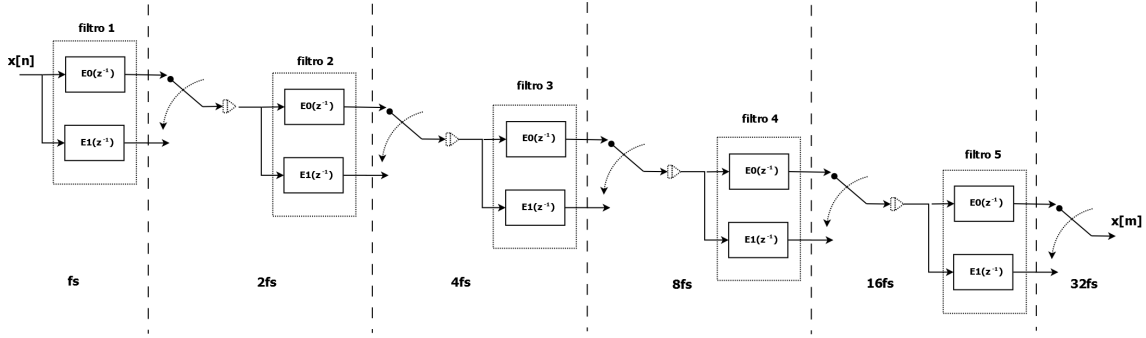


Figura 6.41: Diagrama en bloques del interpolador realizado en FPGA.

muestras¹² de $h[n]$ pares ($n = 0, 2, 4, \dots, \frac{N}{2}-1$):

$$E_0(z) = \sum_{i=0}^{\frac{N}{2}-1} z^{-i} h[2i] \quad (6.33)$$

Observando los $h[N/2]$, se encuentra que en todos los filtros valen 0,5. Entonces, $E_1(0)$ se construye de la siguiente manera:

$$E_1(z) = \frac{1}{2} z^{-\frac{N/2-1}{2}} \quad (6.34)$$

Por lo tanto, cada bloque interpolador x2 consiste en una estructura para FIR simétricos y una cadena de retardo ambas trabajando en paralelo. A su vez, es necesario aplicar una ganancia por dos a la salida de las etapas para compensar la disminución que provoca el sobremuestreo en este caso [40]. Con esto, se omite realizar la multiplicación por $\frac{1}{2}$ en (6.34). Más adelante se verá que la ganancia en (6.33) se efectúa con un registro desplazamiento.

A modo de realizar una comparación, al principio de esta sección se enunció que un sólo filtro requería 5483 coeficientes ($N+1$). Sin ningún tipo de optimización, se efectúan $5483 \times 32 = 175456$ multiplicaciones por muestra de entrada; ya que el sobremuestreo es x32. En el sistema de la Figura 6.41 se utilizan sólo $15 + 4 + 3 + 2 + 2 = 26$ multiplicadores. Puede apreciarse que la reducción en costos computacionales son bastante considerables.

En resumen, se logró realizar una implementación eficiente del interpolador utilizando menos recursos gracias a:

- La realización en cascada con bloques x2.
- La respuesta al impulso simétrica de los filtros.
- Utilizando FIRs de media banda.
- Estructura polifásica de conmutación.

¹²Tener en cuenta que $h[n] = h[N-n]$

	Filtro 1	Filtro 2	Filtro 3	Filtro 4	Filtro 5
Orden	58	14	10	6	6
$h[0]$	1.45×10^{-4}	2.881×10^{-3}	6.8×10^{-3}	-0.032	-0.032
$h[2]$	-3.44×10^{-4}	0.0182	-0.0515	0.282	0.282
$h[3]$	-	-	-	0.5	0.5
$h[4]$	7.25×10^{-4}	-0.0693	0.2947	-	-
$h[5]$	-	-	0.5	-	-
$h[6]$	1.35×10^{-3}	0.304	-	-	-
$h[7]$	-	0.5	-	-	-
$h[8]$	2.33×10^{-3}	-	-	-	-
$h[10]$	3.77×10^{-3}	-	-	-	-
$h[12]$	5.83×10^{-3}	-	-	-	-
$h[14]$	-8.7×10^{-3}	-	-	-	-
$h[16]$	0.0127	-	-	-	-
$h[18]$	-0.0182	-	-	-	-
$h[20]$	0.026	-	-	-	-
$h[22]$	-0.037	-	-	-	-
$h[24]$	0.058	-	-	-	-
$h[26]$	-0.1026	-	-	-	-
$h[28]$	-0.3171	-	-	-	-
$h[29]$	0.5	-	-	-	-

Cuadro 6.4: Tabla de coeficientes para los filtros en el interpolador. Se omitieron las muestras que valen cero y la segunda mitad de $h[n]$, ya que la respuesta es simétrica.

6.6. Proyecto en Vivado

El diseño en bloques creado en Vivado para poner bajo prueba al DAC es el de la Figura 6.42. Se puede divisar el ZYNQ PS, donde se ejecuta el software, un controlador AXI DMA para transferencia directa desde la memoria del PS hacia el PL, el *AXI Interconnect* que funciona como traductor entre protocolos AXI y un bloque llamado *DAC* que efectúa el sistema conversión D/A desarrollado. Este último presenta un bus de entrada de datos *AXI-STREAM* llamado *data in* (longitud de 32 bits), señal de reloj *clk*, señal de reset *ap_rst* y la salida *dac out* (longitud 1 bit) que genera el PDM enviado al filtro a través del pin V12 (ver [24] para más detalles del *Pin-Out*).

El programa en C creado en Xilinx SDK se encuentra en el Anexo A.3.4. Básicamente, la función *DMA_INIT()* configura e inicializa el controlador AXI DMA; luego se resetea al DAC con la función *XGpio_DiscreteWrite()* a través un bloque AXI_GPIO en el PL. La señal de prueba en punto flotante se genera en el vector llamado *datosFloat[]* con longitud 1024, seguidamente se formatea a punto fijo con la función *convert_to_dac_format()* y se envía al PL con *DMA_TRANSFER()*.

La Figura 6.44 muestra el interior del bloque *DAC*. El *DAC_CONTROLLER* es un IP desarrollado en *High Level Synthesis* que recibe la señal $x[n]$ enviada desde el PS a través de AXI stream, la almacena en una memoria RAM que funciona como buffer circular y la transmite periódicamente hacia al interpolador; con una frecuencia de muestreo $f_s = 48828,125$ Hz (Ver Anexo A.3.2). Después, sucede la secuencia de interpoladores constituida por los IP *INTER_X*. Cada uno de estos núcleos cuenta con una entrada y salida de datos de 24 bits (*data_in* y *data_out* respectivamente), *RESET* activo alto y dos entradas de reloj: una llamada *clk* de frecuencia igual a la de muestreo en la *i-ésima* etapa ($i \times f_s$) y otra *OSRclk* (reloj de sobremuestro) funcionando a $2 \times i \times f_s$.

Todos los componentes en el conversor están sincronizados gracias al distribuidor de reloj llamado *CLOCK_DIST*. Este elemento recibe la señal *clk* global del PL (100 MHz) e implementa un contador módulo 32, produciendo así una señal interna de 3,125MHz llamada *TEMPORAL*. En cada flanco positivo de *TEMPORAL* se incrementa un registro de 6 bits en uno. La salida **x1** está conectada al MSB del registro, **x2** al MSB-1 y así sucesivamente hasta llegar a **x32** unido al LSB. La Tabla 6.5 muestra la frecuencia de señal en cada pin del distribuidor. A su vez, se presenta sus formas de onda obtenidas en simulación (Figura 6.43). El *DAC_CONTROLLER* envía una muestra de $x[n]$ al interpolador en cada flanco positivo de **a**. Seguidamente **x1**, que está en contrafase de **a**, comanda la lectura de datos a la entrada de *INTER_X2* que genera a su vez una muestra a la salida cada vez que **x2** esta en alto. De esta manera, los bloques *INTER* subsiguientes operan al doble de frecuencia respecto a la anterior gracias a las restantes señales del distribuidor. En la última fase, el modulador $\Delta\Sigma$ usa a **X32** como señal de reloj.

Para detallar la operación de los bloques interpoladores, en la Figura 6.45 se aprecia



Pin	Frecuencia [KHz]
a	48.828
x1	48.828
x2	97.656
x4	195.312
x8	390.625
x16	781.125
x32	1562.5

Cuadro 6.5: Valores medidos de atenuación y retardo en el filtro a distintas frecuencias.

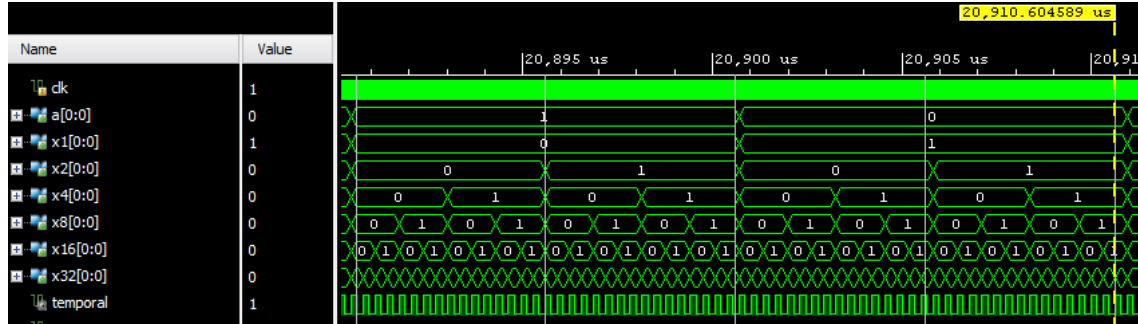
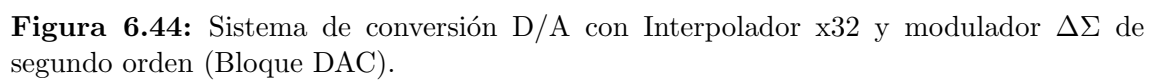


Figura 6.43: Simulación lógica realizada en Vivado del distribuidor de reloj. La frecuencia de *clk* es 100 MHz.

los elementos constituyentes de *INTER_X2*. En primer lugar, se encuentran los bloques *fir_X2* y *DELAY_14* que se corresponden respectivamente con E_0 y E_1 de la representación polifásica. Luego, el multiplexor *MUX* junto con el contador módulo dos *COUNTER* hacen de llave de conmutación. Cuando la entrada *SELECTOR* vale '0', se selecciona la salida el filtro FIR y en caso contrario la del retardo. Por último, se genera un latch de datos (*LATCH_0*) por flanco positivo de reloj para dejar un resultado estable a la parte siguiente. Todo los códigos VHDL del sistema están en la sección A.3.3 del apéndice.

En relación a los filtros FIR, los coeficientes fueron representado en formato $Q_{1,17}$ mientras que las muestras de la señal en $Q_{7,17}$. Cuando se produce la multiplicación, el resultado de 42 bits se almacena en acumuladores/sumadores. Utilizar estas longitudes de datos resulta ventajoso, ya que los bloques *DSP* en el FPGA contienen internamente multiplicadores de 18×25 bits y acumuladores de 48 bits [42]. Por lo que, se necesita sólo 26 de estos elementos para realizar el sobremuestreo. Además, se obtiene el beneficio del paralelismo gracias a la estructura de la Figura 6.38. Para volver a convertir el resultado del producto a formato $Q_{7,17}$ es necesario realizar un desplazamiento binario hacia la derecha de 17 lugares. Sin embargo, como también se debe aplicar una ganancia por dos según lo dicho en la sección del interpolador, el desplazamiento finalmente se realiza con 16 lugares.



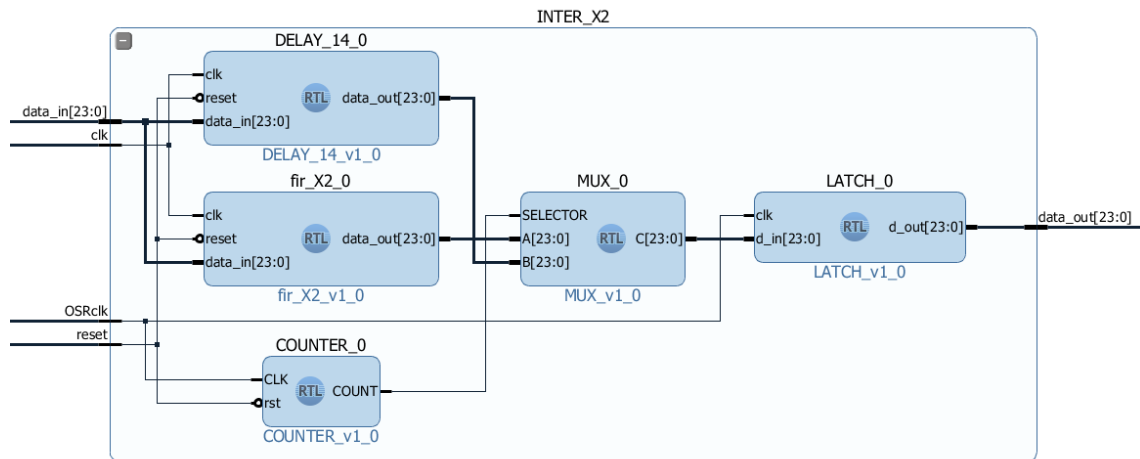


Figura 6.45: Interior de un bloque interpolador x2.

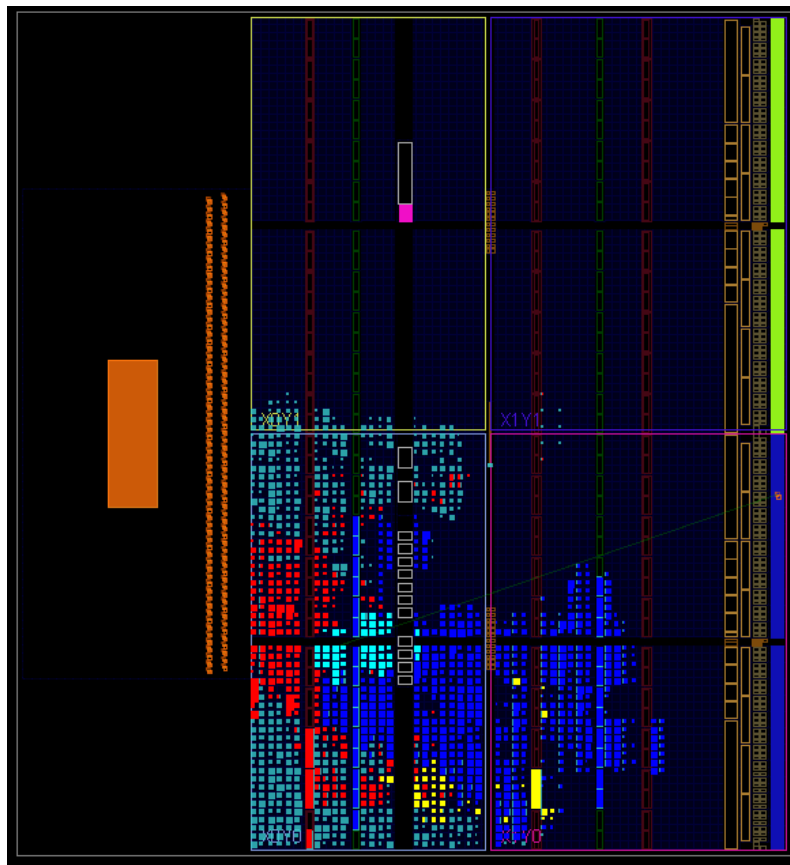


Figura 6.46: Ubicación de los bloques del DAC en el SoC. En rojo: AXI DMA, en azul: INTERPOLADOR, en Amarillo: DAC_CONTROLLER y en Celeste: MODULADOR $\Delta\Sigma$.

Luego de la síntesis e implementación del diseño en *Vivado Design Suite*, se obtuvo la utilización del área en el ZYNQ 7000 (Figura 6.46). Para la generación del bistream no fue utilizada ninguna directiva de *síntesis* ni de *placement*. Finalmente, los recursos necesarios para el convertor D/A diseñado en este capítulo se ven la Figura 6.47.

Hierarchy	Name	Slice LUTs (17600)	Slice Registers (35200)	Slice (4400)	LUT as Logic (17600)	LUT as Memory (6000)	LUT Flop Pairs (17600)	Block RAM Tile (60)	DSPs (80)	Bonded IOB (100)	Bonded IOPADs (130)	BUFGCTRL (32)
design_1_wrapper		3406	4312	1306	3247	159	1300	3.5	26	1	130	1
design_1_1 (design_1)		3406	4312	1306	3247	159	1300	3.5	26	0	0	1
axi_dma_0 (design_1_a...		525	761	246	490	35	290	2.5	0	0	0	0
axi_gpio_0 (design_1_a...		27	26	10	27	0	17	0	0	0	0	0
axi_mem_intercon (desi...		604	832	238	566	38	282	0	0	0	0	0
DAC (DAC_imp_1YBUH12)		1687	1972	611	1663	24	359	1	26	0	0	0
CLOCK_DIST_0 (des...		10	13	4	10	0	9	0	0	0	0	0
DAC_CONTROLLER...		67	157	52	67	0	16	0	0	0	0	0
DELTA_SIGMA_0 (de...		180	75	46	180	0	49	0	0	0	0	0
INTER_X2 (INTER_X...		508	831	231	464	24	77	0	15	0	0	0
INTER_X4 (INTER_X...		220	314	98	220	0	39	0	4	0	0	0
INTER_X8 (INTER_X...		184	242	78	184	0	42	0	3	0	0	0
INTER_X16 (INTER_...		259	170	78	259	0	59	0	2	0	0	0
INTER_X32 (INTER_...		259	170	87	259	0	50	0	2	0	0	0
processing_system7_0 (...)		0	0	0	0	0	0	0	0	0	0	1
ps7_0_axi_periph (desig...		547	692	243	486	61	331	0	0	0	0	0

Figura 6.47: Utilización de recursos totales (LUTs, BRAM, DSPs, etc.) del DAC desarrollado.

6.7. Resultados

Una vez construido el convertor D/A y verificado su diseño mediante simulaciones, se armó el banco de medición de la Figura 6.48; con el fin de evaluar su funcionamiento real. El canal uno del osciloscopio digital se usa para visualizar la señal PDM producida por el modulador $\Delta\Sigma$ mientras que en el canal dos se mide la señal analógica resultante después del filtrado. Primero, se enviaron las secuencias constantes $x[n] = 0$, $x[n] = 0,9$ y $x[n] = -0,9$ obteniéndose las formas de onda de la Figura 6.49. Posteriormente, se generó una señal sinusoidal de amplitud 0.9 manera tal que su período abarque exactamente la longitud de 1024 muestras de $x[n]$ (Figura 6.50). De igual forma, se volvió a repetir este último procedimiento pero ahora con una señal diente de sierra (Figura 6.51).



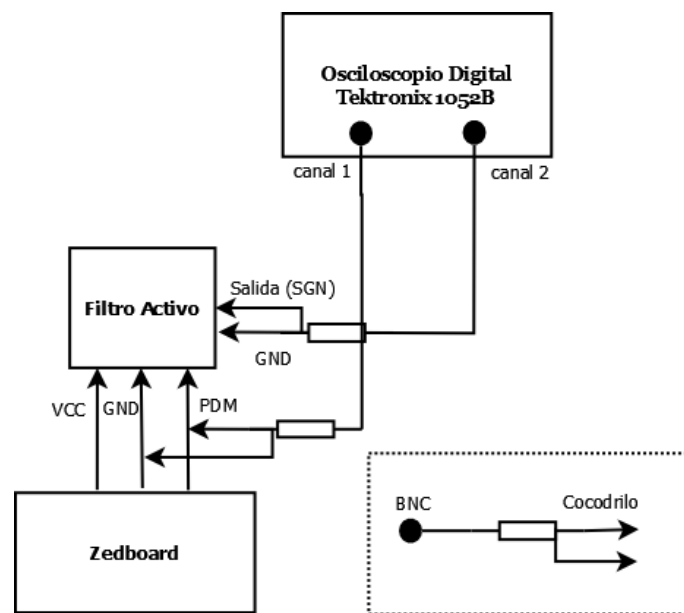
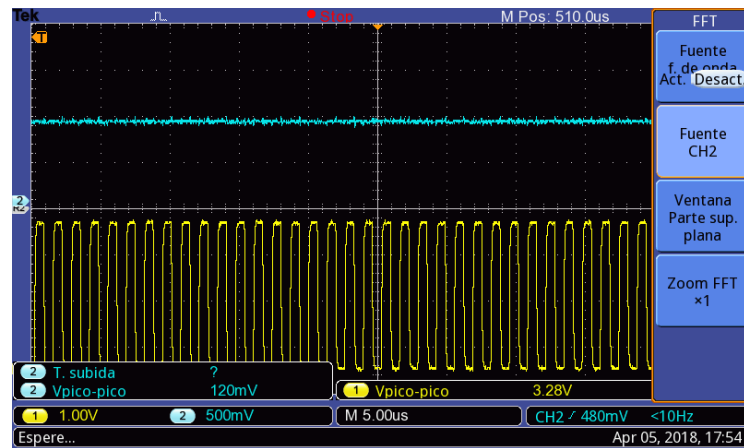
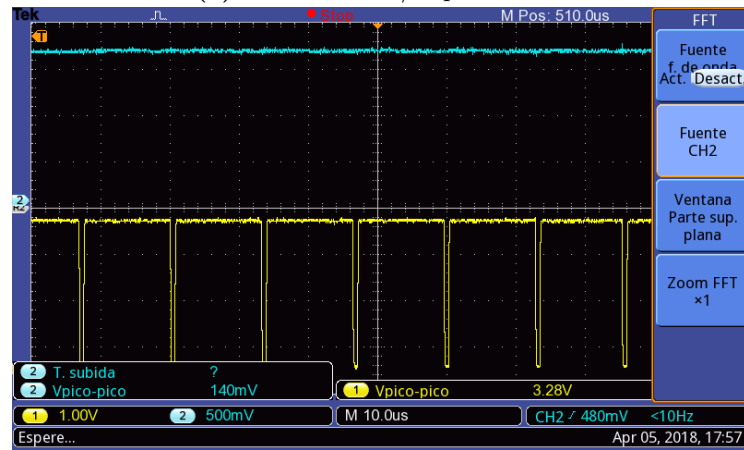


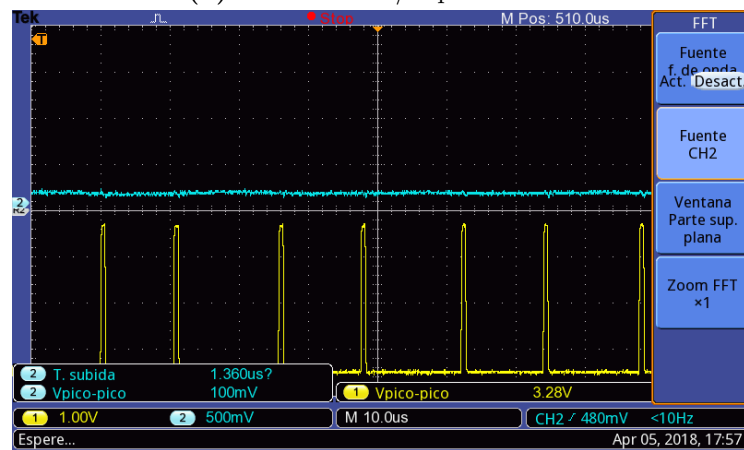
Figura 6.48: Banco de medición para el DAC.



(a) Conversión D/A para $x = 0$.

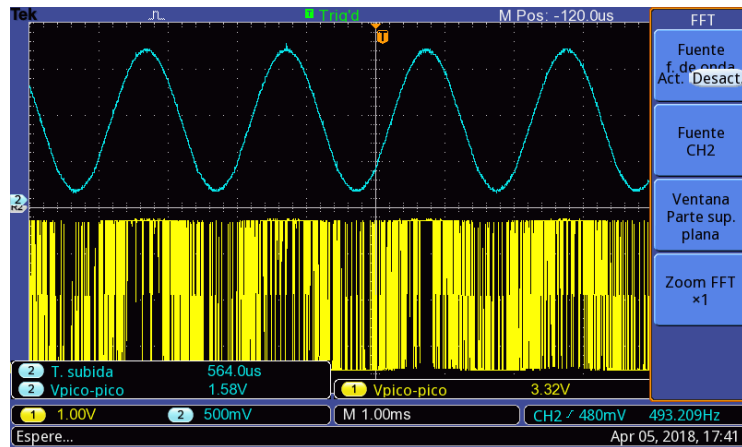


(b) Conversión D/A para $x = 0.9$

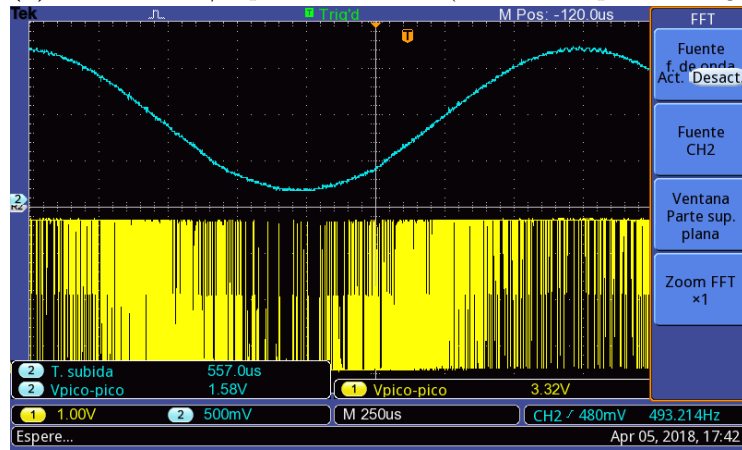


(c) Conversión D/A para $x = -0.9$

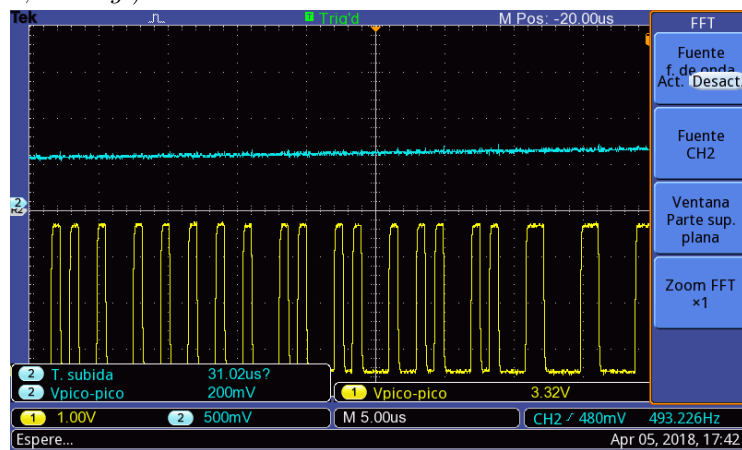
Figura 6.49: Medición de las respuesta del DAC ante entradas constantes. En amarillo: señal PDM. En celeste: señal analógica.



(a) Conversión D/A para x sinusoidal (base de tiempo = 1mSeg.)

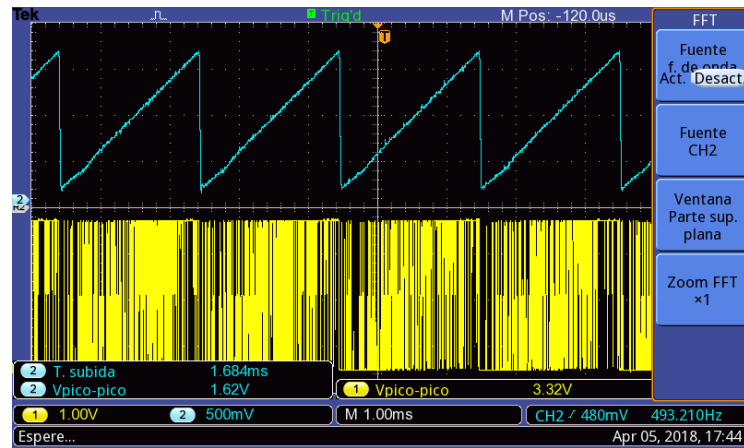


(b) Conversión D/A para x sinusoidal (base de tiempo = 0,25mSeg.)

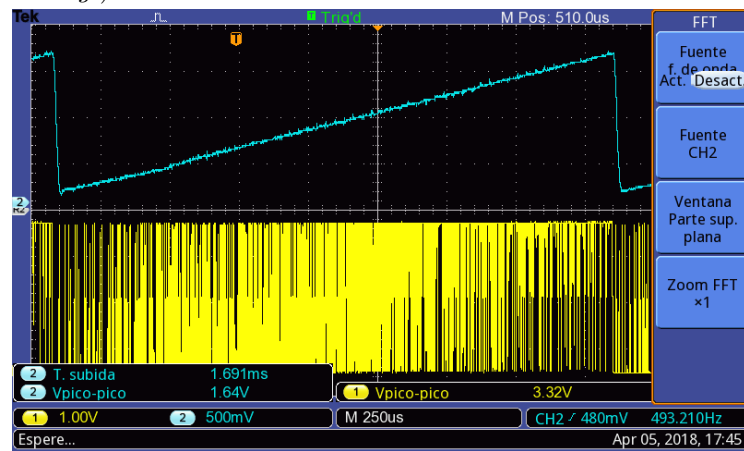


(c) Conversión D/A para x sinusoidal (base de tiempo = 5μSeg.)

Figura 6.50: Medición de las respuesta del DAC ante una entrada sinusoidal (493.22 Hz). En amarillo: señal PDM. En celeste: señal analógica.



(a) Conversión D/A para x diente de sierra (base de tiempo = 1mSeg.)



(b) Conversión D/A para x sinusoidal (base de tiempo = 5mSeg.)

Figura 6.51: Medición de las respuesta del DAC ante una entrada diente de sierra (493.22 Hz). En amarillo: señal PDM. En celeste: señal analógica.

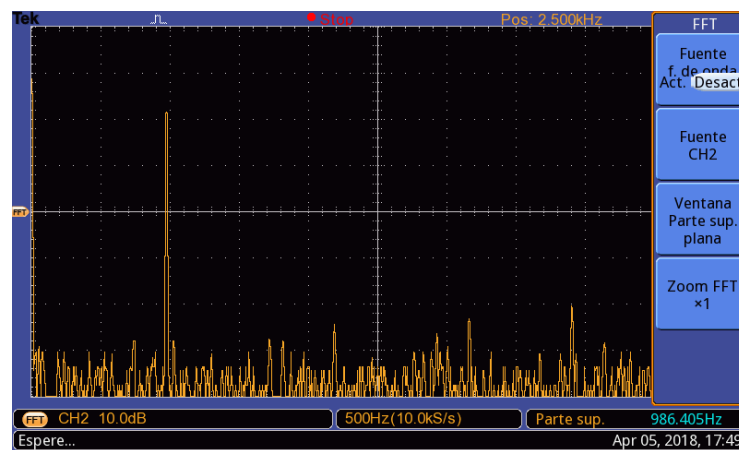


Figura 6.52: FFT a la salida del filtro para una señal sinusoidal de 986.405 Hz y máxima amplitud posible.