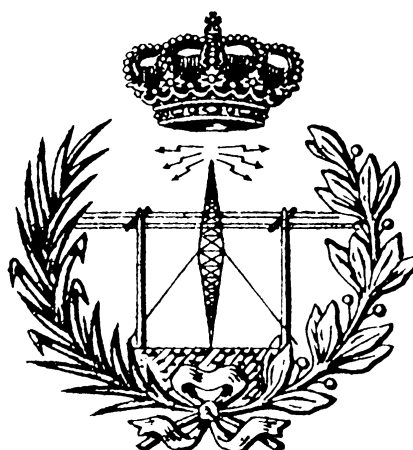


ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

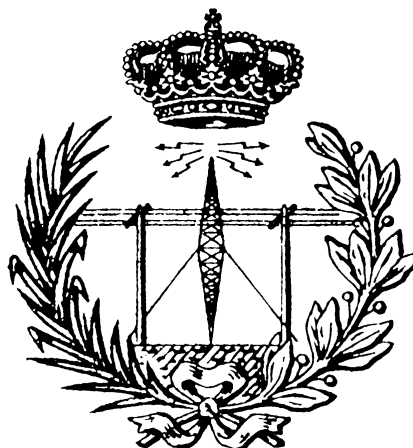


PROYECTO FIN DE CARRERA

DISEÑO DE UN FILTRO ASINCRONO DIGITAL DE ALTA FRECUENCIA PARA RECEPTORES DIGITALES

Autor: D. Carlos Manuel Fernández Reyes
Tutor: Dr. Francisco Javier del Pino Suárez
Fecha: Junio 2018

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



PROYECTO FIN DE CARRERA

DISEÑO DE UN FILTRO ASINCRONO DIGITAL DE ALTA FRECUENCIA PARA RECEPTORES DIGITALES

HOJA DE FIRMAS

Alumno

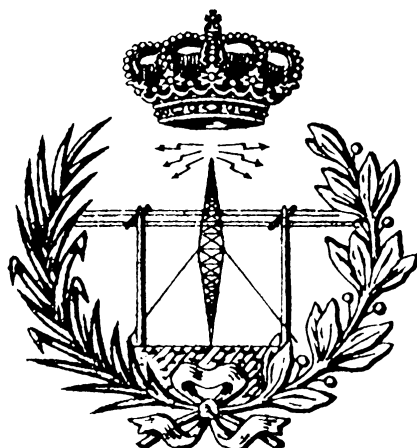
Fdo.: D. Carlos Manuel Fernández Reyes

Tutor

Fdo.: Dr. Francisco Javier del Pino Suárez

Fecha: Junio 2018

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



PROYECTO FIN DE CARRERA

DISEÑO DE UN FILTRO ASINCRONO DIGITAL DE ALTA FRECUENCIA PARA RECEPTORES DIGITALES

HOJA DE EVALUACIÓN

Calificación: _____

Presidente

Fdo.:

Vocal

Fdo.:

Secretario/a

Fdo.:

Fecha: Junio 2018

Agradecimientos

En principio quería dar un agradecimiento en general, compañero, profesores y todos los miembros que componen la comunidad educativa a la que estuve ligado tanto tiempo.

Aunque me ha costado entregar parece que ha llegado a su fin, seguramente no hubiese sido posible sin mi compañera y amiga Enara, siempre está para ayudarme y darme ánimos

Tampoco puede faltar el agradecimiento a mi Bea que no se ha aburrido de este PFC ni de mí aún, y especialmente por preocuparse en que pudiera entregar.

Un especial agradecimiento a mi tutor Javier que me ha aguantado mis inesperadas visitas y siempre con una actitud positiva.

Y finalmente a la persona más especial y que más me ha dado, mi Madre.

ÍNDICE

1. Introducción	1
Motivación	1
Objetivos y organización de la memoria.....	3
2. Introducción a los convertidores A/D Sigma-Delta.....	5
Introducción	5
Convertidor A/D Sigma-Delta	6
Conclusiones	11
3. Filtros digitales.....	13
Introducción	13
Filtros digitales.....	14
Respuesta finita al impulso (FIR)	14
Respuesta infinita al impulso (IIR)	18
Características de los filtros digitales.....	20
Filtro FIR asíncrono	22
Diezmado	24
Implementación del filtro digital en Matlab	29
Conclusiones	34
4. Efectos de Cuantificación y no linealidades	37
Introducción	37
Cuantificación del filtro digital	38
Formato de datos y representación numérica.....	39
Cuantificación de los coeficientes	40
Cuantificación de salida y operaciones aritméticas	44
No-linealidad en los elementos de retardo	49
Desviación estándar	50
Implementación en Matlab.....	51
Resultados de las simulaciones	53
Conclusiones	56
5. Estudio de la implementación a nivel de transistor.....	59
Introducción	59
Variaciones en circuitos digitales	60

Impacto de las variaciones en el retardo de los sistemas digitales	62
Optimización del retardo por las variaciones	62
Análisis estadístico de tiempos	63
Elementos de retardo asíncrono.....	63
Implementación	64
Simulaciones.....	71
Diezmado y bloque multiplicador	74
Implementación	75
Simulaciones.....	78
Conclusiones.....	80
6. Conclusiones.....	83
Orden y coeficientes del filtro	84
Cuantificación del filtro.....	85
No linealidades en lo elementos de retardo	86
Estudio de la implementación a nivel de transistor	87
Conclusiones.....	89
Presupuesto.....	93
Costes debidos a los Recursos Humanos.....	93
Costes de Ingeniería.....	93
Costes de amortización	94
Otros costes.....	95
Coste Total.....	95
BIBLIOGRAFIA	97
ANEXO: MATLAB	101
ANEXO: HSPICE	105

1.Introducción

Motivación

En los últimos años, la continua evolución de la tecnología *deep-submicron* en los circuitos VLSI (*Very Large Scale of Integration*) ha sido evidente, logrando un impacto significativo en el rendimiento de los mismos [1]. De esta forma, se han alcanzado grandes avances en la familia CMOS (*Complementary Metal Oxide Semiconductor*), especialmente en los transistores MOS de canal corto que han logrado mejorar su transconductancia permitiendo reducir consumos e incrementar velocidades de conmutación, siendo así muy atractivas para aplicaciones de alta velocidad y bajo consumo [2].

Hoy en día, las tecnologías inalámbricas están en constante evolución. Como consecuencia, cada vez aparecen nuevos estándares inalámbricos con mayores exigencias y, por lo tanto, es necesario contar con sistemas electrónicos avanzados de altas prestaciones, tales como ancho de banda, resolución o tasas de transmisión de datos. Entre estos sistemas se encuentran los modernos receptores de radiofrecuencia (RF).

En este sentido, los actuales receptores de RF deben tener unos convertidores analógicos digitales (A/D) con altas resoluciones e inmunidad al ruido. A diferencia de los convertidores A/D convencionales, los convertidores sigma-delta (convertidores de sobremuestreo) ofrecen la posibilidad de utilizar circuitos analógicos más simples a expensas de unos circuitos digitales más sofisticados y de mayor velocidad. Esta reducción

de la complejidad en los circuitos analógicos los hace más insensibles a las no idealidades, las cuales se acentúan en las tecnologías nanométricas.

Por lo tanto, los convertidores Sigma-Delta son capaces de conseguir altas resoluciones gracias al empleo de sobremuestreo y técnicas de *noise shaping*. En algunos casos, para poder conseguir una resolución dada es necesario emplear un sobremuestreo muy elevado, esto hace necesario el diseño de un filtro digital a la salida del modulador con una tasa de muestreo muy alta.

El filtro digital a la salida del modulador sigma-delta es empleado para eliminar todas las componentes de alta frecuencia que están fuera de la banda de la señal y reducir la frecuencia de salida, es decir, diezmar. En la Figura 1.1 se muestra el diagrama de bloques modulador Sigma-Delta básico.

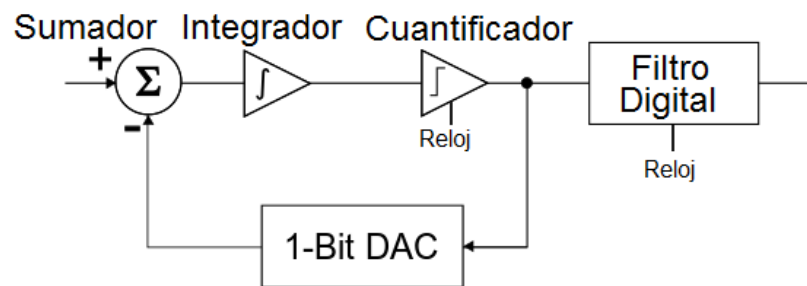


Figura 1.1. Diagrama de bloques de un modulador Sigma-Delta

Por otro lado, la creciente complejidad en los sistemas electrónicos integrados está amplificando un conjunto de fenómenos que degradan su funcionamiento. Estos fenómenos en los sistemas de lógica síncrona pueden ocasionar diferentes inconvenientes en aspectos de energía, distribución de reloj, emisiones electromagnéticas, etc. Con todo esto la lógica síncrona parece alcanzar ciertas limitaciones tecnológicas para resolver estos problemas. En muchos casos, la lógica asíncrona ha demostrado ser una mejor alternativa, proporcionando características interesantes como la inmunidad a los estados metaestables [3], la baja emisión de ruido electromagnético [4], bajo consumo de energía [5] [6], elevadas velocidades de operación y robustez a variaciones de tensión tanto por temperatura como por parámetros en los procesos de fabricación [7] [8] [9].

Objetivos y organización de la memoria

Dentro del contexto considerado en el apartado anterior, este proyecto fin de carrera propone la implementación un filtro FIR con alta frecuencia de muestreo combinado con un diezmador, cuyos elementos de retardo se basan en una línea de retardo asíncrona. La línea de retardo asíncrona se realiza a través de una cadena de inversores implementada con tecnología CMOS de 45nm [10].

El principal objetivo del proyecto fin de carrera es estudiar y analizar los efectos de emplear los filtros FIR, teniendo en cuenta las no idealidades asociadas a este tipo de arquitectura, y la viabilidad de emplearlos en conversores Sigma-Delta por sus altas frecuencias de muestreo.

Inicialmente para el desarrollo del proyecto fin de carrera se han establecido unas condiciones de diseño. Estas condiciones de diseño son las especificaciones que debe cumplir el filtro, las cuales están basadas en estudios y diseños previos de filtros para convertidores A/D Sigma-Delta. Por otro lado, también se ha tenido en consideración diseños de subsistemas que han empleado líneas de retardo síncronas. De este modo, durante el transcurso del proyecto fin de carrera se hemos adoptado diferentes decisiones de diseño que permitan cumplir con las especificaciones indicadas para el filtro. En la tabla 1.1 se pueden observar las especificaciones iniciales para el diseño del filtro.

Tabla 1.1 Especificaciones del filtro.

Tipo de respuesta	Paso baja
Método de diseño	Ventana
Tipo de Ventana	Kaiser
Beta Kaiser	4
Frecuencia de Corte	250 MHz
Frecuencia de muestreo	100 GHz
Reloj local (síncrono)	1 GHz
Atenuación en la banda de rechazo	> 40dB
Banda de transición	< Frecuencia de corte
Rizado	< 0.5dB

A continuación, se expone la organización de la memoria, indicando los diferentes capítulos realizados.

En el capítulo 2 realizamos una introducción y exponemos los conceptos básicos que consideramos adecuados para obtener una visión general de los convertidores Sigma-Delta y permitan comprender decisiones de diseño relacionadas.

Durante el capítulo 3 se proporciona una descripción general de los filtros digitales y una explicación más detallada del filtro FIR para alta tasa de muestreo con diezmador propuesto.

En el capítulo 4 se analizan los efectos de cuantificación y diezmado en el filtro. También se realiza la implementación del filtro en Matlab para poder estudiarlo más en detalle con las especificaciones seleccionadas, comparando así, el empleo de diferentes números de bits en el filtro y sus respuestas en frecuencia. Además, se estudia los efectos de las no-idealidades en la línea de retardo asíncrona del diseño, empleando simulaciones en Matlab.

El capítulo 5 estudiamos a nivel de transistor las desviaciones en los tiempos de propagación de la cadena de inversores usada como elemento de retardo del filtro. Además, hacemos diferentes simulaciones que nos van a permitir ajustar la desviación estándar de los elementos de retardo.

Por otro lado, en el capítulo 5 también diseñamos el bloque diezmador a nivel de transistor para estudiar sus restricciones temporales, muy importantes en la viabilidad del diseño. Para los diseño y simulaciones del capítulo 5 empleamos el programa Hspice.

Finalmente, en el capítulo 6 se exponen las conclusiones y posibles trabajos futuros relacionados con este proyecto fin de carrera.

Para concluir la memoria incluye dos anexos donde se detalla el código empleado en el proyecto. El primer anexo se puede encontrar los scripts de Matlab desarrollados y en el segundo anexo los *netlists* empleados en Hspice.

2. Introducción a los convertidores A/D Sigma-Delta

Introducción

Durante el capítulo anterior indicamos que el diseño del filtro FIR de alta tasa de muestreo emplea líneas de retardo asíncronas, en lugar de las clásicas líneas de retardo síncronas que suelen usar los filtros digitales.

Mediante las líneas de retardos asíncronas se logra alcanzar altas tasas de muestreo y se evita el uso de elevadas frecuencias de reloj. En capítulos siguiente exponemos otras ventajas del empleo de las líneas de retardo asíncronas y los beneficios en el rendimiento del filtro en el caso del diseño propuesto.

Para entender de manera más clara la necesidad de altas tasas de muestreo en los filtros digitales con el enfoque propuesto en el proyecto fin de carrera, así como la idoneidad de emplear las líneas de retardo asíncronas en los Sigma-Delta con alta relación de sobremuestreo (OSR). Debemos situarnos dentro del ámbito de los convertidores A/D, siendo fundamental explicar el funcionamiento de los convertidores A/D Sigma-Delta.

Con la información expuesta en este capítulo, se puede interpretar de manera más clara las especificaciones de diseño propuestas en el primer capítulo. Debido

principalmente a que ciertas decisiones de diseño están muy relacionadas con los convertidores Sigma-Delta.

Convertidor A/D Sigma-Delta

El procesamiento de señales en el dominio digital frente al analógico ofrece importantes ventajas, entre las que se encuentran reducción de área, reducción de consumo, mejor inmunidad al ruido y a la distorsión, o facilidad y flexibilidad a la hora de implementar los diseños [11]. Estas ventajas junto al empleo de DSPs (*Digital Signal Processor*) para su procesamiento, han logrado que los sistemas digitales sean ampliamente usados en muchos y diferentes ámbitos. Para poder realizar el procesamiento en los sistemas digitales se requiere convertir las señales analógicas a digitales mediante convertidores analógicos digitales (ADC).

Aunque existen diferentes tipos de convertidores A/D, entre las arquitecturas más comunes para alta velocidad se pueden destacar el convertidor A/D de pipeline, convertidor A/D *flash*, convertidor A/D de aproximaciones sucesivas y el convertidor A/D de sobremuestreo [12]. En la actualidad la opción más popular empleada en sistemas de audio digital y medición industrial de precisión, es el convertidor A/D de sobremuestreo, también conocido como convertidor A/D Sigma-Delta [13].

Las técnicas de modulación empleadas en los convertidores Sigma-Delta existen desde hace varias décadas atrás [14], pero ha sido en los últimos años cuando han ganado gran popularidad, siendo usadas en sistemas de comunicación como parte de la topología de los convertidores A/D para receptores de RF [15], atribuyéndole algunos de los principales avances tecnológicos recientes en materia de bajo consumo de energía, bajo costo de fabricación y menor exigencia en sus filtros *anti-aliasing* [16].

En la Figura 2.1 se muestra un diagrama de bloques de un convertidor A/D Sigma-Delta. En el diagrama se puede observar dos bloques, el modulador Sigma-Delta y un filtro paso bajo con diezmado. Las técnicas empleadas en el modulador sigma-delta permite alcanzar altas resoluciones de conversión usando baja resolución de cuantificación operando a altas velocidades [17], proporcionando así un flujo de bits que es filtrado y diezmado por un filtro paso bajo con diezmador.

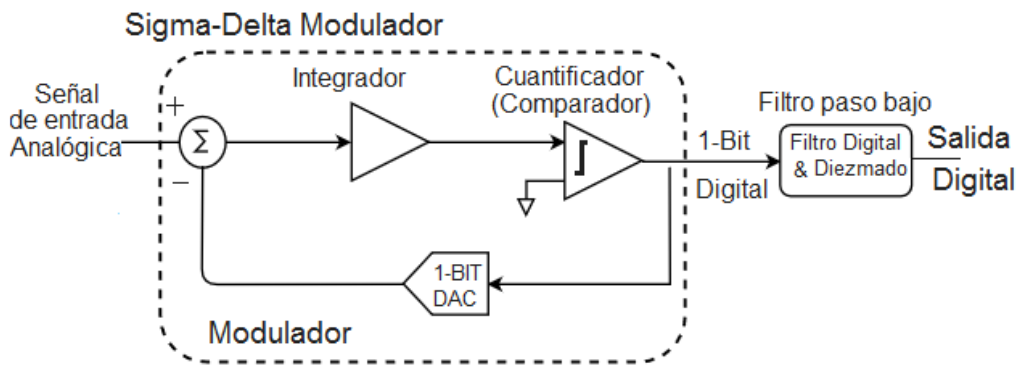


Figura 2.1. Diagrama de bloques de un convertidor A/D Sigma-Delta [15]

El modulador Sigma-Delta es el núcleo del convertidor. Es el encargado de digitalizar las señales analógicas de entrada y reducir el ruido en las frecuencias más bajas [18]. Como se puede observar en la Figura 2.1, el modulador Sigma-Delta consiste en un integrador seguido de un comparador y un convertidor digital analógico (DAC) en bucle que realimenta el bloque. En el dominio del tiempo, el integrador se encarga de integrar la diferencia entre la señal de entrada y la señal de salida que llega mediante la realimentación del modulador, distribuyendo así el error de cuantificación o ruido en espectro, reduciéndolo en la banda de interés y elevándolo en las frecuencias más elevada [19]. La salida del integrador pasa al comparador actuando como un cuantificador de un bit. Finalmente, a través de la retroalimentación negativa se intenta minimizar la diferencia entre la señal de entrada y la señal de salida del modulador Delta-Sigma, obteniendo así en la salida del modulador, el promedio del seguimiento de la señal de entrada [20] como se puede observar en la Figura 2.2.

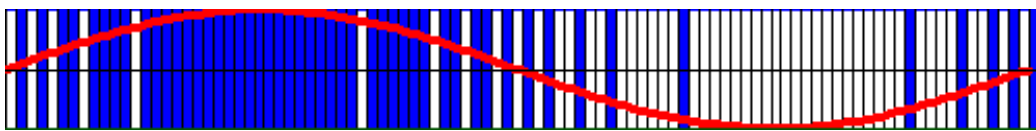


Figura 2.2 Señal de entrada y salida en el dominio del tiempo en un modulador Sigma-Delta. [21]

Debido a la técnica que emplea los convertidores Sigma-Delta de sobremuestreo, normalmente deben usar filtros con altas frecuencias de muestreo, aunque la banda de las señales analógicas de entrada se encuentre en frecuencias mucho más bajas. Esta técnica realiza el muestreo a frecuencias mucho más elevadas que la *frecuencia de Nyquist*, la cual

es la mínima frecuencia de muestreo necesaria para evitar que se produzca *aliasing*. Mediante el sobremuestreo se reduce el ruido de cuantificación en la banda de la señal de entrada [22].

La relación entre la frecuencia de muestreo empleada en la modulación y la frecuencia de Nyquist se le denomina relación de sobremuestreo (*Over Sampling Ratio, OSR*), definido como la relación entre la frecuencia de muestreo y la frecuencia de Nyquist, como se muestra en la ecuación 2.1. Donde f_s es la frecuencia de muestreo, f_N es la frecuencia de Nyquist y f_B es la banda de frecuencia máxima de la señal de entrada.

$$OSR = \frac{f_s}{f_N} = \frac{f_s}{2 \cdot f_B} \quad (2.1)$$

Si analizamos los efectos en potencia que produce la técnica de sobremuestreo empleada en los Sigma-Delta se puede observar la reducción de la potencia de cuantificación en el ancho de banda de la señal de entrada. Esta reducción de potencia es debida a que la potencia del error de cuantificación es distribuida de manera uniforme, extendiéndose sobre el ancho de banda de la frecuencia de muestreo ($-f_s/2$ a $+f_s/2$). De esta manera la potencia de cuantificación que se encuentra acotada en el rango del ancho de banda de la señal ($-f_B$ a f_B) es menor que el error de cuantificación producido por el ancho de banda total.

Como se muestra en la Figura 2.2 y definido en la ecuación 2.2, podemos calcular la densidad espectral de potencia con el ruido reducido que proporciona el convertidor, denominada como P_e , donde OSR es la relación de sobremuestreo, Δ es el intervalo de cuantificación que a su vez es definido como $\Delta = V_{ref} / 2N$, $S_e(f)$ es la densidad espectral de potencia del ruido total y se define como $S_e = \Delta^2 / 12$. Finalmente, $H(f)$ es el filtro con diezmado ya realizado y que elimina el ruido de cuantificación en la frecuencia fuera de la banda de la señal f_B [21] [20].

$$P_e = \int_{\frac{f_s}{2}}^{-\frac{f_s}{2}} S_e(f) |H(f)|^2 df = \left(\frac{\Delta^2}{12} \right) \cdot \frac{1}{OSR} \quad (2.2)$$

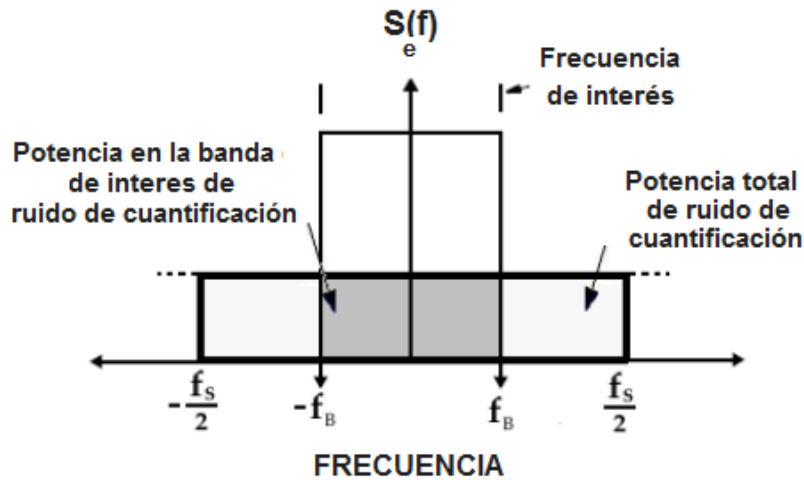


Figura 2.2. PSD del error de cuantificación en un convertidor sobremuestreado [15].

En la figura 2.3 se muestra la diferencia entre el espectro de ruido de cuantificación en un convertidor A/D Sigma-Delta con la frecuencia de muestreo de Nyquist y el espectro de ruido de cuantificación con un convertidor que emplea la técnica de sobremuestreo expuesta.

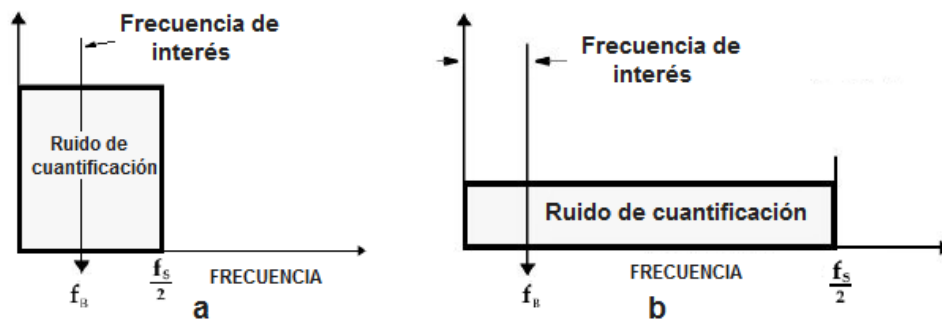


Figura 2.3. Densidad espectral del ruido de cuantificación del convertidor con Nyquist (a) y con sobremuestreo (b).

Además del sobremuestreo, los convertidores A/D Sigma-Delta hacen uso de *noise shaping*, técnica encargada de distribuir el error de cuantificación (o también denominado ruido del convertidor). Mediante esta técnica se realiza una reducción de la potencia de error de cuantificación en las frecuencias de la banda de interés, incrementándose en frecuencias más elevadas.

En la figura 2.4 se puede observar el efecto que produce la técnica *noise shaping* en el espectro, donde se puede apreciar que el ruido de cuantificación total se mantiene, aunque concentrado en las frecuencias más elevadas. De este modo es posible obtener altas resoluciones en un ancho de banda relativamente reducido [23], entendiendo como la resolución de un modulador Sigma-Delta a la relación máxima de señal a ruido (SNR) que puede obtenerse en su señal de salida, estimada de forma aproximada en los convertidores Sigma-Delta para una SNR máxima con un cuantificador uniforme con la ecuación 2.3, donde N es el número de bits de cuantificador [24].

$$\text{SNR}_{\text{uniforme}} = 6,02N + 1,76\text{dB} \quad (2.3)$$

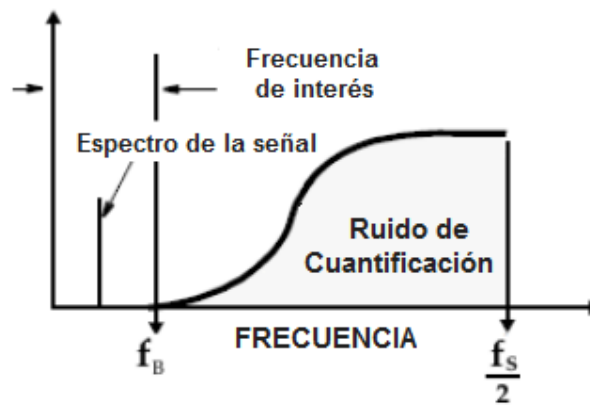


Figura 2.4. Espectro de ruido de cuantificación del convertidor Sigma-Delta con *noise shaping* [14].

Como se puede observar en la figura 2.4, el ruido de cuantificación está concentrado en las frecuencias altas debido a la técnica *noise shaping* que se ha indicado en el párrafo anterior.

Para eliminar o atenuar el ruido de cuantificación es necesario emplear un filtro capaz de atenuar las frecuencias altas de la señal de salida del modulador. Como se ha mencionado al inicio del capítulo, la señal de entrada después de ser procesada por el modulador Sigma Delta, (donde es sobremuestreada y se le aplica la técnica *noise shaping*), pasa a través del filtro digital con diezmado como se muestra en la Figura 2.1. La función del filtro digital es proporcionar una corte lo más brusco posible en las frecuencias superiores, con el objetivo principal de eliminar el ruido de cuantificación que se encuentra fuera de la banda de interés [25]. En la Figura 2.5 se muestra el espectro de la señal y el

filtro digital, antes y después eliminar el ruido de cuantificación que el modulador desplazó hacia las frecuencias más altas.

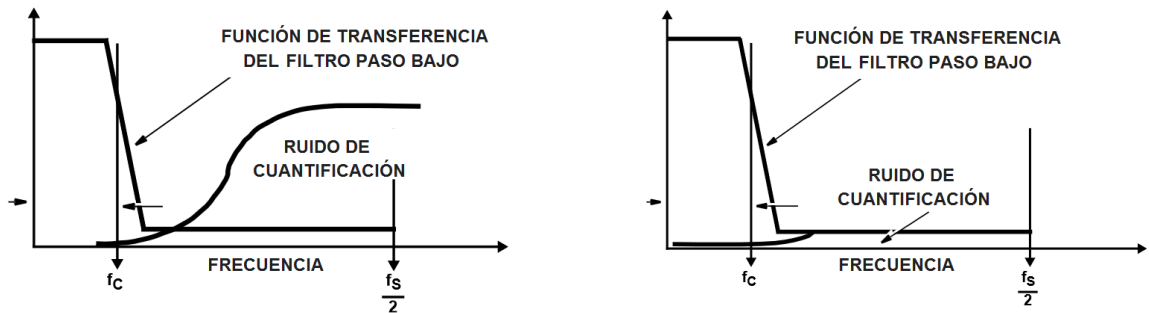


Figura 2.5. Espectro del ruido de cuantificación y filtro, antes (derecha) y después de ser filtrado (izquierda) [15].

Conclusiones

En este capítulo se han estudiado los aspectos más relevantes de los convertidores analógico/digital Sigma-Delta, centrándose en las técnicas que emplea y la relación de las altas tasas de muestreo y la caracterización del ruido, lo que nos permite entender mejor las especificaciones de diseño, como pueden ser el uso de un filtro paso bajo, la alta tasa de muestreo del filtro o la frecuencia de corte muy distancia de la frecuencia máxima de muestreo. Por otro lado, también ayudará a comprender algunas decisiones de diseño adoptadas durante los siguientes capítulos del proyecto fin de carrera para la implementación del filtro digital.

3. Filtros digitales

Introducción

En los últimos años los procesadores digitales de señales (DSPs) han incrementado notablemente su presencia en productos de telecomunicaciones y electrónica, debido en gran parte a la mejora de la tecnología VLSI. Estos dispositivos se emplean en numerosos ámbitos, como pueden ser las comunicaciones inalámbricas, el procesamiento de imagen o audio, etc.

Los filtros digitales son una parte muy significativa de los DSPs [26]. Estos permiten la separación de alguna señal que ha sido mezclada o la recuperación de una señal que ha sufrido distorsión [27] [28].

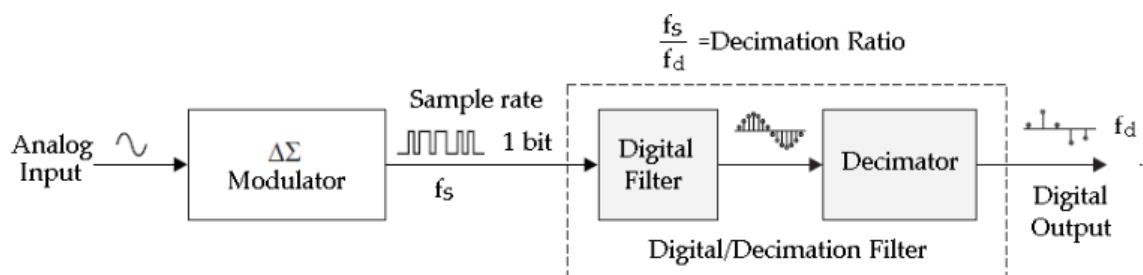


Figura 3.1. Diagrama de bloques de un convertidor A/D Sigma-Delta con filtro diezmador digital (bloques sombreados) [29].

Aunque existen diferentes formas de implementar un filtro digital, en este proyecto se ha propuesto la implementación de un filtro FIR con líneas de retardo asíncronas. Este será

empleado como filtro diezmador en un convertidor A/D sigma-delta con altos requerimientos en la tasa de muestreo, como se puede observar en la Figura 3.1

Como introducción, en este capítulo se realiza una descripción general de los filtros digitales, indicando sus propiedades y características principales. Además, se expone el proceso de diseño de los filtros con respuesta al impulso de longitud finita (FIR).

Posteriormente, se desarrolla la arquitectura del filtro FIR propuesto indicando todos los requisitos que debe presentar. Para ello, se calcula el orden necesario del mismo y sus coeficientes. Esto se hará considerando un filtro ideal, sin valorar otras consideraciones como la cuantificación o no linealidades las cuales se estudiarán en el siguiente capítulo.

Filtros digitales

El filtro digital es un tipo de filtro que opera con señales discretas y cuantificadas, puede implementarse mediante software en ordenadores con datos almacenados en memoria o con circuitos digitales específicos, estos reciben una secuencia discreta de números (señal de entrada) que es transformada mediante un algoritmo en otra secuencia discreta de números (señal de salida), que tendrá un espectro en el dominio de frecuencia modificado [30]. Normalmente se encuentran en los procesadores digitales de señales (DSPs) de propósito general para aplicaciones de audio o circuitos integrados de aplicación específica (ASIC) en aplicaciones con altas tasas de muestreo [31].

El filtrado de una señal se puede realizar mediante dos tipos de filtros digitales, denominados filtro FIR (*Finite Impulse Response* o Respuesta Finita al Impulso) o filtro IIR (*Infinite Impulse Response* o Respuesta Infinita al Impulso). Aunque en el presente proyecto fin de carrera el tipo escogido en las especificaciones iniciales es de tipo FIR, dedicaremos un apartado a los filtros IIR a modo de introducción y con el fin de conocer las diferencias con los filtros FIR.

Respuesta finita al impulso (FIR)

En los filtros digitales se pueden distinguir dos tipos de funcionamiento, filtro no-recursivo y filtro recursivo. En el caso de los filtros con respuesta finita al impulso, el tipo

de funcionamiento es no-recursivo y este viene definido cuando la salida está exclusivamente calculada a partir de los valores de entrada ($Y[n]=X[n]+X[n-1]+ X[n-2]...$), como se puede apreciar en la Figura 3.2a.

Por otro lado, los filtros digitales recursivos, como veremos en el apartado de los filtros digitales IIR, además de los valores de entrada para calcular la salida, se necesitan emplear valores previos de salida ($Y[n-1]$, $Y[n-2]...$), como se puede observar en la Figura 3.2 b.

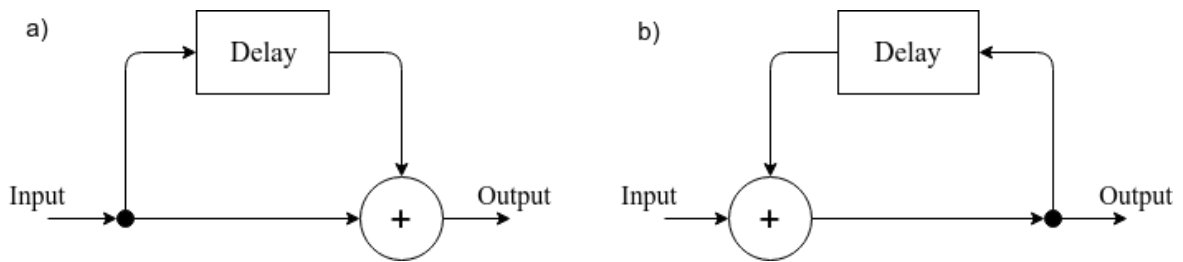


Figura 3.2. Diagrama de bloques de los dos tipos de filtros digitales: (a) no-recursivo (FIR) y (b) recursivo (IIR).

Los filtros con respuesta finita al impulso son filtros digitales caracterizados por la respuesta obtenida al aplicar una señal impulso a la entrada, la cual será de duración finita. Además, como se comentó en el párrafo anterior la salida depende exclusivamente de las entradas actuales y anteriores. Así, un filtro de longitud N puede ser definido como muestra la ecuación 3.1:

$$y(n) = b_0x(n) + b_1x(n-1) + \dots + b_{N-1}x(n-N+1) = \sum_{k=0}^{N-1} b_kx(n-k) \quad (3.1)$$

donde b_k son los coeficientes del filtro y el orden del filtro viene dado el número de coeficientes N .

Además, se puede obtener la expresión en el dominio temporal mediante la ecuación 3.3. En la misma, la salida del filtro $y(n)$ es obtenida mediante la convolución de la entrada $x(n)$ con la respuesta al impulso $h(n)$ (ecuación 3.2).

$$y(n) = \sum_{k=-\infty}^{+\infty} h(k) \cdot x(n-k) \quad \text{con} \quad \begin{cases} 0 & k < 0 \\ h(k) & 0 \leq k < N \\ 0 & k \geq N \end{cases} \quad (3.2)$$

$$y(n) = \sum_{k=0}^{N-1} h(k) \cdot x(n-k) \quad (3.3)$$

Seguindo la ecuación 3.3 se puede representar la estructura básica de un filtro FIR como se aprecia en la Figura 3.2.

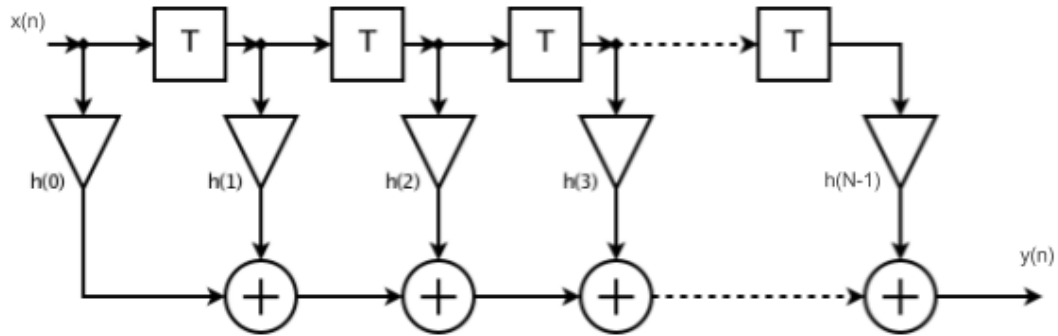


Figura 3.2. Estructura básica de un filtro FIR, forma directa I.

Teniendo en cuenta la ecuación 3.1 y empleando como entrada la señal impulso se puede deducir la respuesta al impulso con la ecuación 3.4.

$$h(n) = \sum_{i=0}^N b_i \cdot \delta(n-i) = b_n \quad (3.4)$$

Por otro lado, la transformada de Fourier de una respuesta impulsional de un filtro corresponde a su función de transferencia y permite caracterizar a los filtros en el dominio frecuencial. En el caso de un filtro FIR se puede observar su función de transferencia en la ecuación 3.5, la cual también puede ser expresada en términos de los coeficientes b_n si consideramos la ecuación 3.4.

$$h(n) = \sum_{n=-\infty}^{\infty} h[n] \cdot z^{-n} = \sum_{n=0}^N b_n \cdot z^{-n}$$

(3.5)

Dadas las ecuaciones expuestas para los filtros FIR, se puede deducir que todos los polos se encuentran localizados en el origen, indicando así que son estables. Otra ventaja a considerar es que nos permite diseñar el filtro con fase lineal [32].

Entre las desventajas de los filtros FIR se encuentra la necesidad de orden elevado para cumplir ciertas especificaciones de diseño y lograr que la respuesta del filtro sea la deseada, ocasionando un mayor tiempo de cómputo, así como un aumento de componentes en el caso de implementaciones electrónicas específicas.

Para el diseño de un filtro FIR se debe seleccionar los coeficientes (b_n) adecuados que cumplan con las especificaciones de diseño dadas. Existen diferentes métodos para el cálculo de los coeficientes de entre los cuales los más empleados son el método de diseño de ventanas, el método *equiripple*, el método de muestreo de frecuencia y el método de mínimos cuadrados [26]. A continuación, se realiza una breve descripción de cada uno de los métodos comentados:

- El método de diseño de ventanas implementa la respuesta en frecuencia ideal deseada, definida con la respuesta al impulso correspondiente, la cual se calcula con la multiplicación de la respuesta al impulso deseada en el tiempo por la función ventana. Las funciones ventanas se basan simplemente en truncar la respuesta del impulso para obtener una aproximación al filtro FIR causal. Dependiendo del tipo de ventana usado para el truncamiento se pueden modificar ciertos parámetros (ancho de transición, rizado pasabanda, atenuación máxima en la banda de rechazo, relación lóbulo principal y lóbulos laterales) en la respuesta en frecuencia del filtro [26].
- El método *equiripple* (rizado uniforme) emplea una técnica indirecta para encontrar los coeficientes del filtro óptimos, donde los valores son ajustados para conseguir un rizado que se mantenga uniforme en toda la banda. Este método utiliza un procedimiento de optimización más eficiente para minimizar el ancho de transición en relación con el rizado de la banda de paso y la de rechazo [33].

- El método de muestreo en frecuencia consiste simplemente en muestrear uniformemente la respuesta de frecuencia deseada y realizar la inversa de la transformada de fourier discreta (DFT) para obtener la correspondiente respuesta al impulso del filtro [34]. Generalmente, el método de muestreo proporciona buen rendimiento con un número suficiente de frecuencias muestreadas, pero el rendimiento es limitado cuando la banda de transición deseada es muy estrecha [35].
- El método de aproximación por mínimos cuadrados (también denominado como minimización del error por mínimos cuadrados) es similar al de muestreo en frecuencia, pero en este caso se controla la respuesta mediante la integral del error al cuadrado respecto a la respuesta deseada entre los puntos muestreados, considerando un número de puntos muestreados mayor que el orden del filtro. Como la energía de la señal está relacionada con el cuadrado de la misma, este método es apropiado para optimizar el diseño del filtro [36].

Respuesta infinita al impulso (IIR)

A diferencia de los filtros FIR, los filtros IIR son filtros digitales de tipo recursivo los cuales presentan una respuesta al impulso infinita o, dicho de otra manera, si la entrada del filtro IIR es una señal impulso, en la salida se tendrá un número infinito de términos no nulos, impidiendo así su retorno al estado de reposo.

La expresión matemática de los filtros IIR está indicada en la ecuación 3.6, y se puede observar que la salida del filtro (y_n) depende de las entradas actuales (x_n), pasadas (x_{n-k}) y las salidas de instantes anteriores (y_{n-k}).

$$y_n = b_0 \cdot x_n + b_1 \cdot x_{n-1} + \dots + b_N \cdot x_{n-N} - a_1 \cdot y_{n-1} - a_2 \cdot y_{n-2} - \dots - a_M \cdot y_{n-M} \quad (3.6)$$

donde a_k y b_k son los coeficientes del filtro y el orden viene determinado por el máximo entre los valores de M y N, que a su vez indica el número de polos y ceros en la función de transferencia.

Aunque existen numerosas implementaciones posibles para los filtros IIR, en la Figura 3.3 se muestra una estructura básica, la cual cumple la ecuación 3.6 y permite comprender de manera más adecuada su funcionamiento.

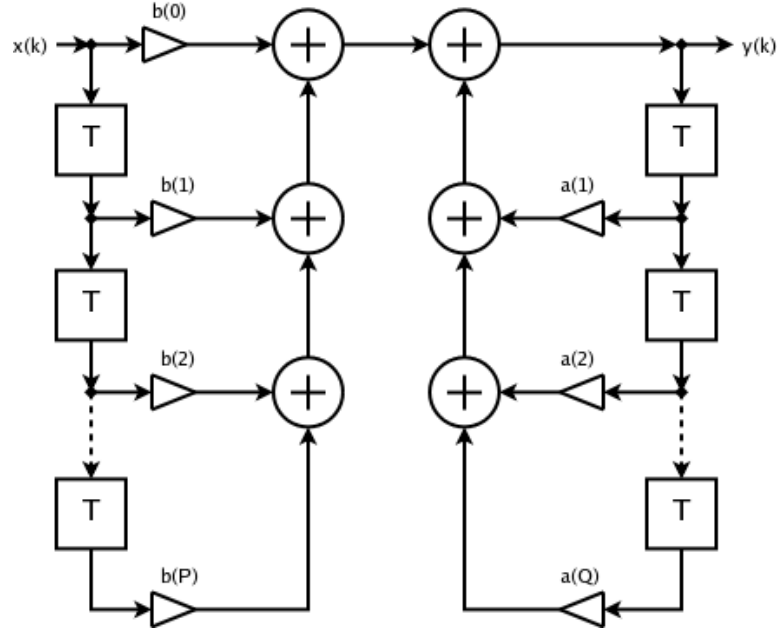


Figura 3.3. Estructura básica para filtros IIR.

Se puede observar que la estructura de la Figura 3.3 cumple con la ecuación 3.6. Además, si se aplica la transformada Z a la ecuación es posible caracterizar la función de transferencia como se puede observar en la ecuación 3.7.

$$H(z) = \frac{\sum_{k=0}^K b_k \cdot z^{-k}}{1 + \sum_{k=1}^M a_k \cdot z^{-k}} \quad (3.7)$$

Al contrario que los filtros digitales FIR, en la función de transferencia de los filtros IIR existe denominador, lo que introduce ceros que pueden causar inestabilidades, comprometiendo así la estabilidad y causalidad del sistema. Otro de los inconvenientes de los filtros IIR es la dificultad de tener fase lineal, solo posible con técnicas más complejas que acaban añadiendo más coeficientes. Por otro lado, los filtros IIR pueden lograr las mismas exigencias que los filtros FIR, pero con un número de coeficientes menor, reduciendo la carga computacional.

Características de los filtros digitales

Los filtros digitales pueden ser clasificados en base a la respuesta en el dominio de la frecuencia. Esta clasificación comprende cuatro tipos básicos según sea la posición relativa de las bandas de paso y las bandas atenuadas. Los cuatro tipos existentes son filtro paso bajo, filtro paso alto, filtro paso banda y filtro elimina banda [37].

- El filtro paso bajo permite el paso de señal a baja frecuencia (banda de paso) mientras atenúa las frecuencias superiores a la frecuencia de corte del filtro.
- El filtro paso alto presenta un comportamiento complementario al paso bajo, siendo las frecuencias bajas las atenuadas y las frecuencias altas las paso banda.
- Los filtros paso banda tienen un rango de frecuencias definido por las frecuencias de corte superior e inferior, permitiendo pasar las frecuencias que se encuentran dentro del rango indicado.
- El filtro elimina banda presentan un comportamiento complementario a los paso banda, siendo así las frecuencias del rango definido las atenuadas.

En la Figura 3.4 podemos ver los cuatro tipos de filtro comentados, según el módulo de su respuesta frecuencial.

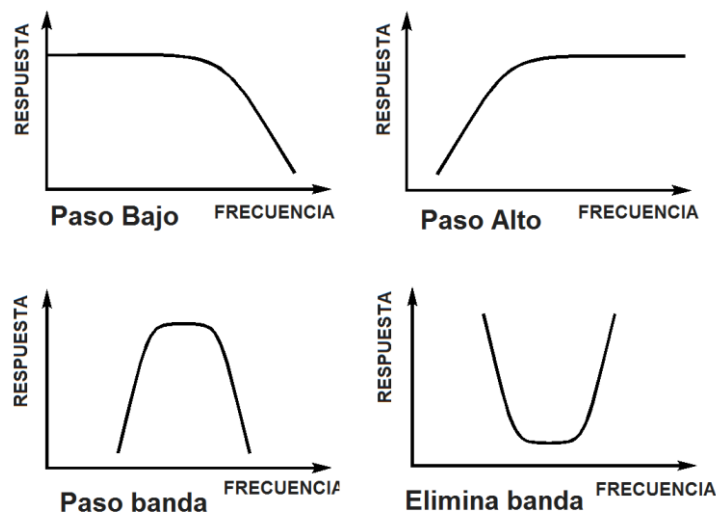


Figura 3.4. Tipos de filtros.

Idealmente la respuesta en frecuencia de los filtros viene caracterizada por una ganancia máxima para las frecuencias de interés (denominada también banda de paso), la cual en los filtro FIR es plana y sin atenuación (0 dB), y una mínima para las frecuencias

no deseadas (denominadas también banda de rechazo), las cuales presentan una atenuación. Además de esta característica, existen varios parámetros que nos permiten caracterizar la respuesta frecuencial del filtro de una manera más específica, entre los que se pueden destacar:

- Frecuencia de corte. Frecuencia a la cual la amplitud de la señal sufre una atenuación de 3 dB (la señal se reduce a 0.707). En el caso de filtros paso banda o filtros elimina banda, existen dos frecuencias de corte, frecuencia de corte inferior y frecuencia de corte superior. Como su denominación indica la inferior será la frecuencia de corte con frecuencia más baja que la superior.
- Ancho de la banda de transición. Conjunto de frecuencias comprendidas entre la banda de paso y la banda de rechazo.
- Rizado. Indica la presencia de una ondulación no deseada en la banda de paso/rechazo o en ambas.
- Atenuación mínima en banda de rechazo. Valor de la atenuación mínima que el filtro es capaz de proporcionar en la banda de rechazo.

En la Figura 3.5 se puede observar los parámetros comentados de los filtros sobre una respuesta frecuencial común de un filtro paso bajo.

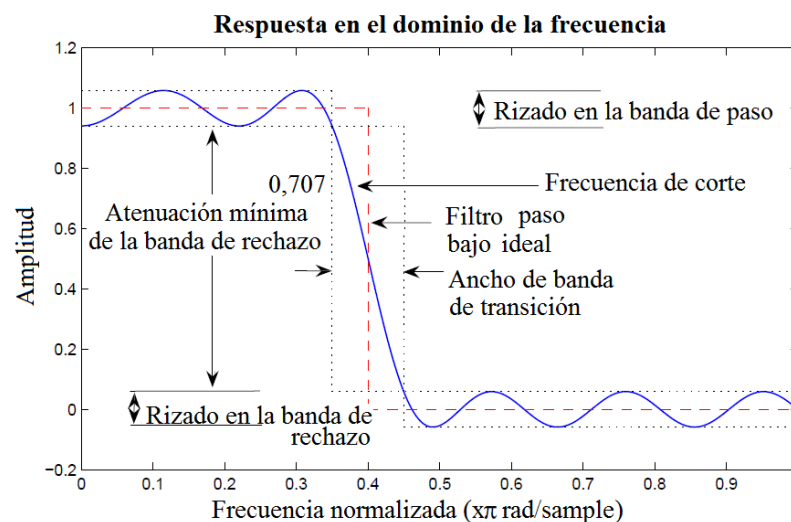


Figura 3.5. Respuesta en frecuencia de un filtro paso bajo (indicando los parámetros)

Filtro FIR asíncrono

En las estructuras básicas de filtros FIR como la mostrada en la Figura 3.2, se pueden apreciar los elementos de retardos clásicos que actúan como registros síncronos, encargados de introducir un retardo a la señal de entrada. La mayoría de los diseños lógicos actuales emplean este tipo de registros síncronos coordinados por una señal de reloj y para su funcionamiento se basan principalmente en dos suposiciones: todas las señales son binarias y el tiempo es discreto. Esto facilita el diseño lógico y permite simplificarlo [38].

Aunque actualmente el desarrollo de circuitos síncronos domina los diseños en la industria de semiconductores, existen importantes limitaciones en la sincronización cuando son necesarias altas frecuencia de muestreo, lo cual dificulta de manera notable el rendimiento y el diseño [39]. Por otro lado, la lógica asíncrona tiene propiedades inherentes que pueden ser utilizadas para mejorar aspectos del diseño, como reducción de consumo, alta velocidad de operación, robustez ante variaciones de alimentación, temperatura y parámetros de proceso de fabricación [40].

De esta forma, para cumplir con las especificaciones descritas en el capítulo 1 de este proyecto fin de carrera, principalmente con la alta tasa de muestreo indicada para el filtro digital, se propone una implementación asíncrona de bajo consumo para los elementos de retardos del mismo. Los bloques restantes se mantienen síncronos, como los empleados en los filtros digitales convencionales, de esta manera se tendrá una combinación de bloques síncronos y asíncronos en la implementación, intentando conseguir las ventajas del diseño síncrono con la velocidad de muestreo que se puede lograr con el bloque asíncrono.

Para el bloque compuesto por los elementos de retardo del filtro FIR, hemos empleado líneas de retardo asíncronas mediante una cadena de inversores. Según las especificaciones iniciales de diseño, la tecnología CMOS empleada, PTM de 45nm [10], permite retardos de propagación alrededor de los 10 ps. como se estudiará en más profundidad en el capítulo 5, permitiendo operar alrededor de los 100 GHz. Los bloques síncronos del filtro FIR propuesto emplean flip-flops y multiplexores, siendo su frecuencia de trabajo menor, 1

GHz. El cambio de frecuencia será realizado por el diezmador como se explica en el apartado 3.3.4 del presente capítulo.

Como se ha comentado, el filtro con alta tasa de muestreo y diezmado propuesto está implementado mediante un filtro tipo FIR, siendo esta una condición inicial del diseño. Esto se debe principalmente a que el diseño está basado en el uso de líneas de retardo asíncronas y las arquitecturas de los filtros IIR dificultan notablemente el empleo de dichas líneas de retardo.

El diseño del filtro FIR está basado en la arquitectura de forma directa, la cual comprende tres tipos de elementos básicos: sumador, multiplicador y unidad/bloque de retardo.

En la Figura 3.6 se puede observar la estructura del filtro FIR de forma directa y los elementos básicos que la componen, donde los valores b_k son los coeficientes del filtro usados por el multiplicador. De esta manera la salida $y[n]$ será la suma de las muestras retardadas multiplicadas por el coeficiente correspondiente. Además, se puede observar en la Figura 3.6 que presenta un bloque diezmador justamente antes de la salida del filtro FIR, el cual realiza el diezmado, reduciendo así la frecuencia de muestreo original a una menor para evitar sobremuestreo en la salida.

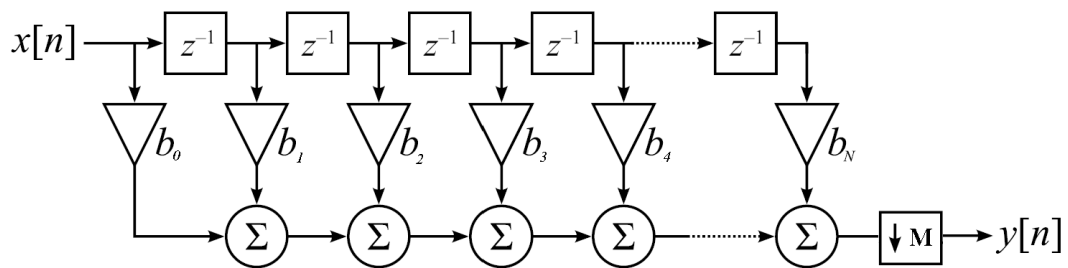


Figura 3.6. Estructura básica de forma directa en un filtro FIR.

El diseño propuesto en el presente proyecto ha modificado la clásica estructura de la forma directa de la figura anterior reemplazando los elementos de retardo por líneas de retardo asíncrono y, modificando la posición del bloque diezmador que ahora se ubica en las entradas de los bloques multiplicadores en lugar de la salida del filtro, como se puede apreciar en la Figura 3.7.

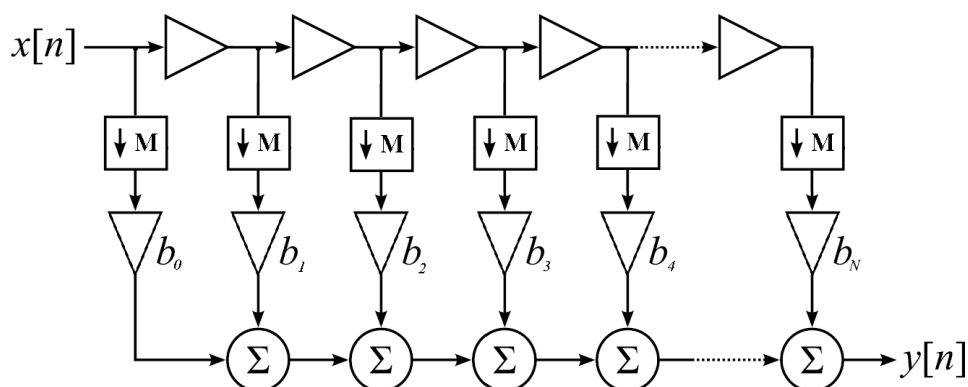


Figura 3.7. Estructura de filtro FIR propuesta, con diezmado más eficiente y los elementos de retardos con una línea de retardo asíncrona (empleando inversores).

Por otro lado, como se indicó en las especificaciones del capítulo 1, el filtro debe usar el método de ventana para su diseño. Sin llegar a ser el método más óptimo, es un método rápido, fácil de usar y robusto [41], facilitando de esta manera el estudio de los efectos de emplear líneas de retardo asíncronos en el diseño del filtro FIR. El tipo de ventana empleado es el Kaiser, el cual proporciona el parámetro, que permite mejorar el rizado o la banda de transición, de forma que la mejora de una de ellas significa la degradación de la otra. Por ello, es necesario llegar a un compromiso entre las dos propiedades, de tal forma que se cumplan los requisitos propuestos al inicio, sobre todo en materia de banda de transición, evitando el posible *aliasing* durante el diezmado de la señal, como se verá más en profundidad en las siguientes secciones del capítulo.

El cálculo de los coeficientes y el ajuste de las especificaciones del filtro se han realizado mediante la herramienta *toolbox Signal Processing* de Matlab, el cual proporciona un conjunto de funciones y *apps* para generar, medir, transformar, filtrar y visualizar señales. En el apartado 3.3.2 se explica más en profundidad el uso del *toolbox* para el diseño del filtro. Recalcar que durante este capítulo los cálculos y simulaciones de diseño son basadas en la implementación ideal del filtro, sin considerar factores no-lineares o la cuantificación del mismo, que será estudiada en el siguiente capítulo.

Diezmado

En el capítulo 2 se pudo apreciar la importancia del proceso de sobremuestreo en los actuales convertidores analógicos/digitales Sigma-Delta [41], lo cual implica que la señal

de salida de los mismos tenga una frecuencia de muestreo muy superior a la necesaria. Por lo tanto, para reducir este sobremuestreo es necesario un diezmador, encargado reducir el número de muestras de la señal en tiempo discreto, equivalente a una reducción en la frecuencia de muestreo.

Sin embargo, antes de diezmar la señal de salida del convertidor será necesario filtrar la misma para evitar el fenómeno de *aliasing* (filtro *anti-aliasing*), debido a que dos señales separadas por un múltiplo de la frecuencia de muestreo son indistinguibles.

Por tanto, al usar el diezmador y modificar la frecuencia de muestreo existe el riesgo de alterar las frecuencias de la banda de interés, distorsionando así la señal original e impidiendo una futura reconstrucción exacta de la misma [42]. Así, en aplicaciones con convertidores A/D sigma-delta, el filtro diseñado cumplirá con la función de filtro *anti-aliasing*, requerido para este tipo de convertidores A/D.

Esta disminución de la frecuencia de muestreo se lleva a cabo tomando una de cada M muestras y desechando las $M-1$ muestras intermedias. En la Figura 3.8 se puede observar la relación entre la entrada y la salida en tiempo discreto del bloque diezmador el efecto del diezmado en las muestras de una señal discreta de entrada con frecuencia f_s y su señal de salida discreta con la nueva frecuencia de muestreo $f'_s = f_s / M$.

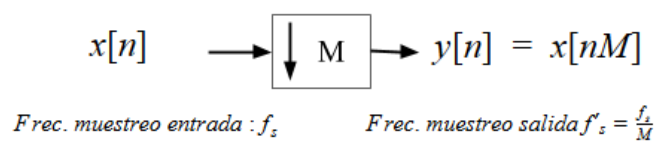
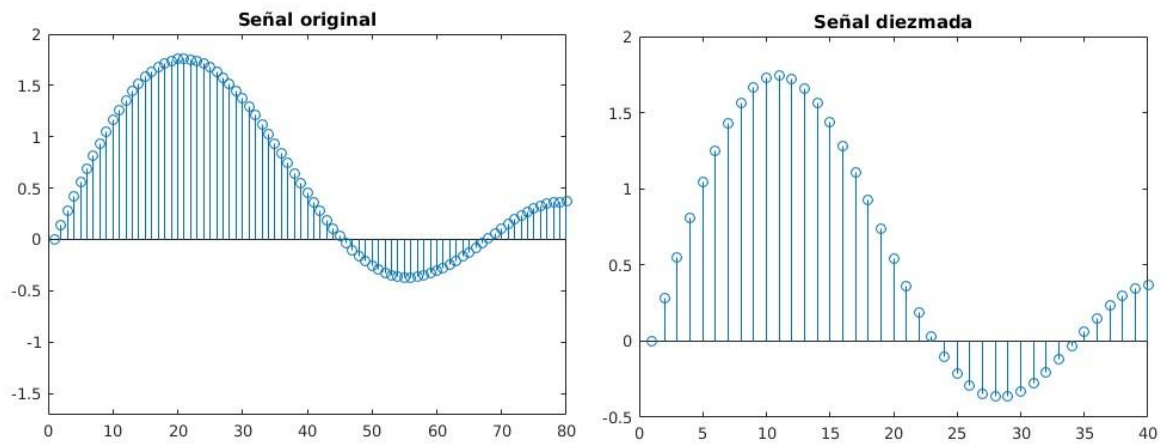


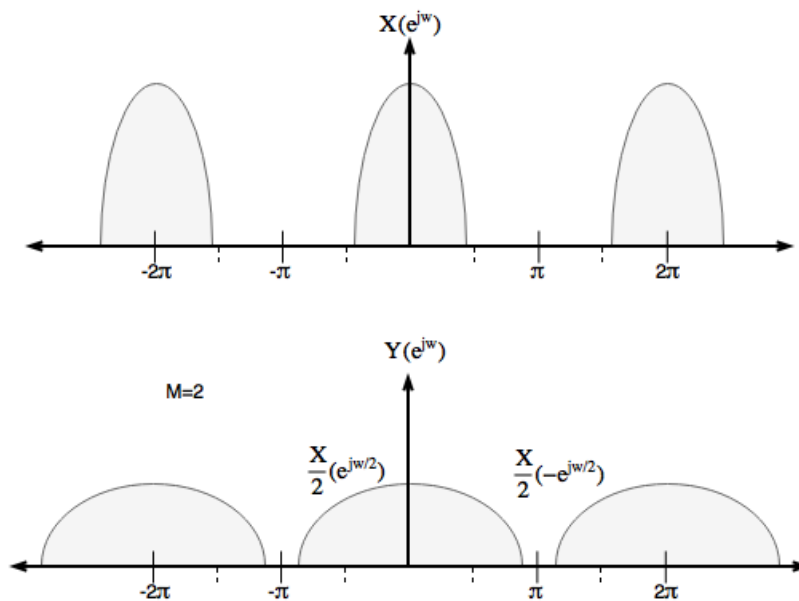
Figura 3.8. Entrada y salida del bloque diezmador y conversión de frecuencias de muestreo.

En la Figura 3.9 se puede observar de manera más clara el efecto que produce en las señales discretas el diezmado, en este caso un diezmado con factor de $M=2$.


 Figura 3.9. Ejemplo de diezmado en una señal discreta con un factor de $M=2$.

La caracterización en el dominio de la frecuencia es algo más compleja. Aplicando la transformada Z obtenemos la ecuación 3.8 y, evaluando la misma sobre la frecuencia unidad, obtenemos la ecuación 3.9 [35], siendo esta una sumatoria de espectros de la señal de entrada desplazados uniformemente y escalados un factor $1/M$ como se puede observar en la Figura 3.10, en este caso con un factor $M=2$.

$$Y(e^{jw}) = \frac{1}{M} \sum_{k=0}^{M-1} X e^{j(w-2\pi k)/M} \quad (3.9)$$


 Figura 3.10. Espectros de la señal original (superior) y la señal diezmada (inferior) con un factor $M=2$.

Con el fin de evitar el efecto del *aliasing* comentado, el diseño debe cumplir el Teorema de Nyquist, el cual determina que la frecuencia de muestreo debe ser superior al doble de la máxima frecuencia a muestrear [43]. De este modo y en un filtro paso bajo con las especificaciones descritas en el capítulo 1 y con bandas de transición tan elevadas, el filtro debe mantener en la banda de atenuación las frecuencias que sobrepasen el doble del valor de la frecuencia de muestreo que resulta una vez la señal diezmada, o dicho de otra manera, la frecuencia de muestreo mínima, una vez diezmada, debe ser superior al doble de la suma de la frecuencia de corte más la banda de transición, pudiendo ser comprobado con la ecuación 3.10.

$$\frac{F_s}{M} \geq 2 \cdot (\text{Frecuencia de corte} + \text{banda de transición}) \quad (3.10)$$

donde F_s es la frecuencia de muestreo del filtro y M es el factor de sobremuestreo del filtro, el cual está definido por la división entre la frecuencia de muestreo del filtro asíncrono (F_s) y la frecuencia de muestreo síncrona encargada del diezmado (F'_s), calculado con la ecuación 3.11.

$$M = \frac{F_s}{F'_s} = \frac{100 \text{ GHz}}{1 \text{ GHz}} = 100 \quad (3.11)$$

Así, con el factor de sobremuestreo calculado en la ecuación 3.11 y obteniendo los valores de las especificaciones de diseño indicadas en el capítulo 1 (frecuencia de corte, banda de transición y frecuencia de muestreo) es posible sustituir los valores de la ecuación 3.10 y comprobar que se cumple con las exigencias del teorema del muestreo de Nyquist, como se puede observar en la ecuación 3.12. Finalmente, en el apartado 3.3.2 se tendrá en cuenta que se cumple la especificación de banda de transición y frecuencia de corte, para el cálculo de los coeficientes y evitar que se produzca el *aliasing* en el diezmado.

$$\frac{100 \text{ GHz}}{100} \geq 2 \cdot (250 \text{ MHz} + 250 \text{ MHz})$$

(3.12)

Como se adelantó en el apartado 3.3 del presente capítulo, la implementación del bloque diezmador ha sido modificada respecto a la estructura de filtro FIR de forma directa tal como se aprecia en la Figura 3.11a. La decisión de emplear una estructura diferente a la clásica es debido a que los bloques sumadores y multiplicadores del diseño propuesto operan de forma síncrona, al contrario de los bloques retardo del filtro que operan de forma asíncrona, siendo necesario una señal de reloj sólo para los bloques síncronos.

Con la estructura clásica, la señal de reloj tendría que operar a una frecuencia cercana a los 100 GHz (la frecuencia de muestreo del filtro) puesto que también es empleada en los bloques de retardo. Esta frecuencia es demasiado elevada y, por lo tanto, emplear bloques de retardo asíncronos nos permitirá lograr un filtro digital con una frecuencia de muestreo alta sin necesidad de usar una señal de reloj a esas frecuencias.

Para evitar esta limitación, la estructura propuesta implementa el bloque de retardo del filtro, previo a los bloques multiplicador y sumador, reduciendo la frecuencia de muestreo en un factor de 100 como se calculó en la ecuación 3.11 y, consecuentemente, la frecuencia de operación de estos bloques. Aunque esta nueva estructura requiere más bloques (lo cual implica más lógica/transistores), el consumo es reducido al operar a una frecuencia menor, logrando así ser realizable y mejorando la eficiencia del filtro digital. En la Figura 3.11b se puede observar la estructura del filtro FIR propuesta, donde se puede apreciar la nueva ubicación de los bloques diezmadores.

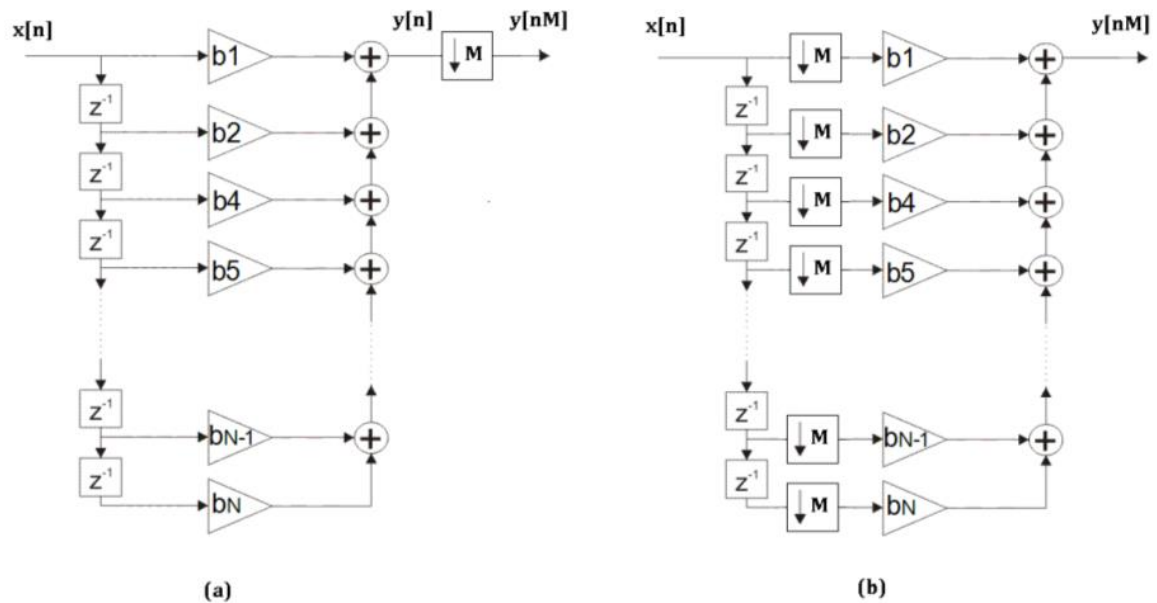
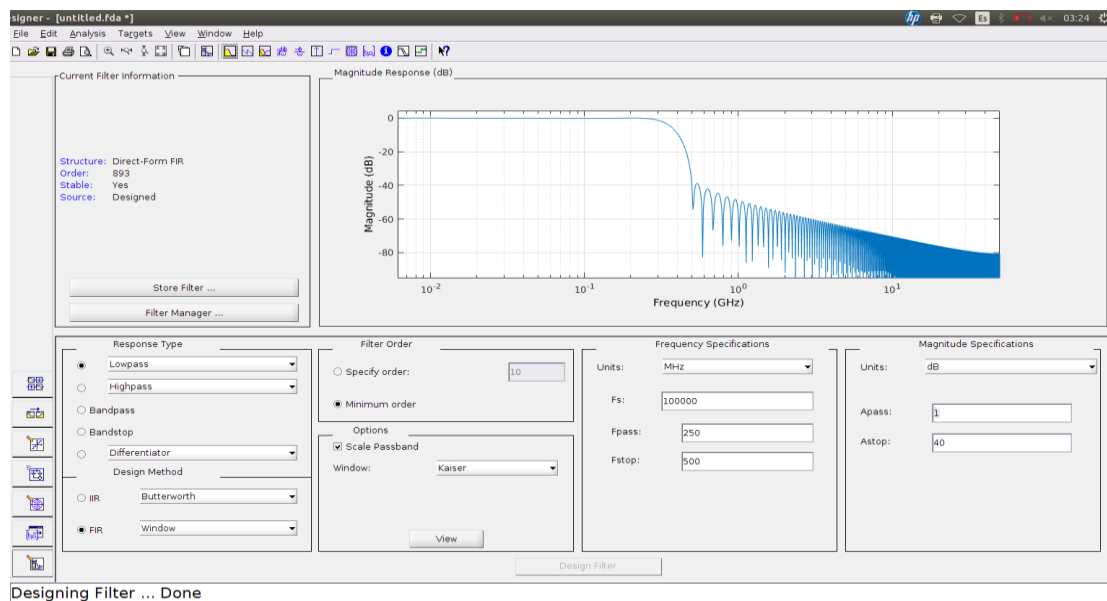


Figura 3.11. a) Filtro FIR con estructura forma directa I con diezmador. b) Filtro FIR con estructura forma directa I con diezmador ubicado de forma eficiente.

Implementación del filtro digital en Matlab

Para realizar el filtro digital con las especificaciones que se indican en el capítulo 1, el filtro ha sido implementado mediante la herramienta software Matlab. Matlab puede ampliar sus capacidades mediante cajas de herramientas, también conocidos como *toolbox*. En el caso de nuestro diseño se ha empleado el *toolbox Signal Processing*, que proporciona técnicas para diseñar, simular y analizar filtros digitales tanto en punto fijo como en punto flotante, lo que permite generar, medir, transformar, filtrar, visualizar señales o comparar señales en los dominios de tiempo, frecuencia y tiempo-frecuencia.

En este capítulo del proyecto, tanto los coeficientes como el orden del filtro se ha calculado mediante varias funciones (`fir1`, `freqz`) y herramientas (`fvtool`), así como la aplicación *Filter Designer*, todos componentes de la *toolbox Signal Processing*. La aplicación *Filter Designer* permite diseñar y analizar filtros digitales mediante un interfaz gráfico muy amigable, donde se deben insertar los parámetros que se desea que cumpla el filtro digital a diseñar. En la Figura 3.12 se puede observar el interfaz gráfico que proporciona la aplicación *Filter Designer* de Matlab.

3.12. Aplicación *Filter Designer* de Matlab.

Además de obtener las diferentes respuestas del filtro y sus coeficientes con los valores introducidos, la aplicación permite emplear varios tipos de análisis o ajustar la cuantificación para estudiar su respuesta al impulso, como se explicará con profundidad en el siguiente capítulo. En nuestro caso para tener una primera aproximación al diseño del filtro se realizaron varios tests con la aplicación *Filter Designer*, rellenando los campos necesarios con los valores de las especificaciones inicialmente dadas, obteniendo una respuesta en frecuencia al impulso como se muestra en la figura 3.12.

Por otro lado, con el fin de facilitar el manejo del filtro y representarlo gráficamente de manera más precisa, se ha empleado la función `fir1`, la cual nos permite obtener coeficientes del filtro, la función `freqz` que devuelve la respuesta en frecuencia del filtro y con la ayuda de la herramienta `fvtool` (*Filter Visualization Tool*) se podrá visualizar gráficamente las respuestas del filtro obtenidas. En la Tabla 3.1 se indican las funciones con los parámetros necesarios para su uso.

Tabla 3.1. Conjunto de funciones y herramientas empleadas en Matlab.

Función	Sintaxis	Argumentos
Obtener coeficientes del filtro	<code>fir1(Orden,Wn,Ventana(Orden+1,Parámetro))</code>	Orden: Orden del filtro. bWn: Restricciones en frecuencia. Ventana: Tipo de

ventana. Parámetro: Forma enventanado.		
Obtener respuesta en frecuencia del filtro	freqz(B,A,N)	B, A: Coeficientes del filtro. N: Número de puntos evaluados.
Representar gráficamente la respuesta de filtro	fvtool(Hd)	Hd:Respuesta del filtro (obj)

Inicialmente, mediante la aplicación *Filter Designer* y empleando la opción ‘*Minimum order*’ se ha realizado una estimación del posible orden del filtro. Para ello se ha introducido los parámetros necesarios, como son el tipo de respuesta del filtro (*Response Type*), método de diseño del filtro FIR (*Design Method*), tipo de ventana (*Options*), frecuencia de muestreo (F_s), frecuencia final de la banda de paso (F_{pass}), frecuencia inicial de la banda de rechazo (F_{stop}), atenuación mínima en la banda de rechazo (A_{stop}) y rizado máximo en la banda de paso (A_{pass}).

En la figura 3.13 se puede observar todos los valores introducidos, que coinciden con los indicados en las especificaciones de diseño del capítulo 1, excepto la frecuencia final de la banda de paso (F_{pass}) y la frecuencia inicial de la banda de rechazo (F_{stop}), las cuales no están indicadas en las especificaciones del proyecto, pero mediante la frecuencia de corte y la banda de transición permitida, se han estimado empleando las siguientes expresiones mostradas en la ecuación 3.13.

$$\begin{aligned}
 \text{Frecuencia final de la banda de paso} &= \text{Frecuencia de corte} = 250 \text{ Mhz} \\
 \text{Frecuencia inicial de la banda de rechazo} &= \text{Frec. de corte} + \text{Banda de transición} = 500 \text{ Mhz}
 \end{aligned}
 \tag{3.13}$$

Response Type <input checked="" type="radio"/> Lowpass <input type="radio"/> Highpass <input type="radio"/> Bandpass <input type="radio"/> Bandstop <input type="radio"/> Differentiator Design Method <input type="radio"/> IIR Butterworth <input checked="" type="radio"/> FIR Window	Filter Order <input type="radio"/> Specify order: 512 <input checked="" type="radio"/> Minimum order Options <input type="checkbox"/> Scale Passband Window: Kaiser View	Frequency Specifications Units: MHz Fs: 100000 Fpass: 250 Fstop: 500	Magnitude Specifications Units: dB Apass: 0.5 Astop: 40
--	--	---	---

Figura 3.13. Valores rellenos en la aplicación Filter Designer de Matlab.

El orden del filtro proporcionado mediante la aplicación *Filter Designer* de Matlab es 893 y la respuesta en el dominio de la frecuencia se puede observar en la Figura 3.14.

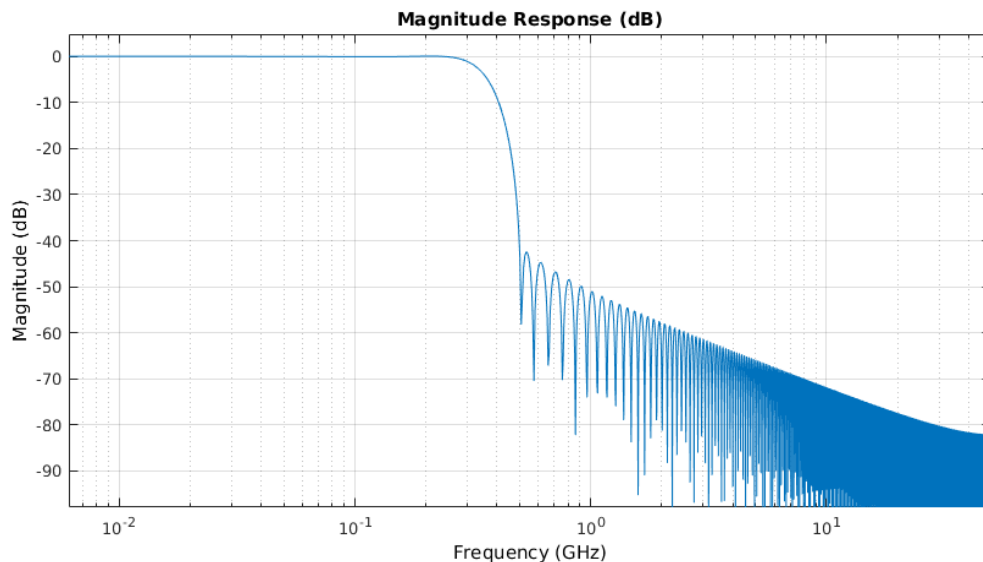


Figura 3.14. Respuesta en frecuencia del filtro obtenida de la aplicación Filter Designer.

En este primer diseño se ha logrado cumplir las especificaciones impuestas inicialmente, pero el orden del filtro obtenido, 893, se considera bastante elevado si se desea una implementación del filtro a nivel de transistor, necesario para implementar las líneas de retardo asíncrono del filtro. Además, mediante la aplicación *Filter Designer* con la opción '*minimum order*' no se ha podido ajustar la *beta Kaiser* a 4, como veremos a continuación este ajuste logra reducir el orden manteniendo las especificaciones.

Para intentar reducir este orden tan elevado se ha decidido emplear la función '*fir1*' para calcular los coeficientes. Mediante esta función se puede indicar la frecuencia de corte en lugar de la frecuencia final de la banda de paso y la frecuencia inicial de la banda de

rechazo. Por otro lado, permite introducir la *beta Kaiser*, que en el caso de nuestro diseño es 4, tal y como indican las especificaciones iniciales del filtro.

Una vez obtenido los coeficientes se ha empleado la función '*freqz*' para generar la respuesta en frecuencia y finalmente se ha usado la herramienta '*fvtool*' para representarla gráficamente. Este procedimiento se ha repetido modificando el orden del filtro, con el fin de encontrar un orden que permita cumplir con las especificaciones iniciales.

En la figura 3.15 se puede observar las diferentes respuestas en el dominio de la frecuencia de los filtros diseñados con diferente orden.

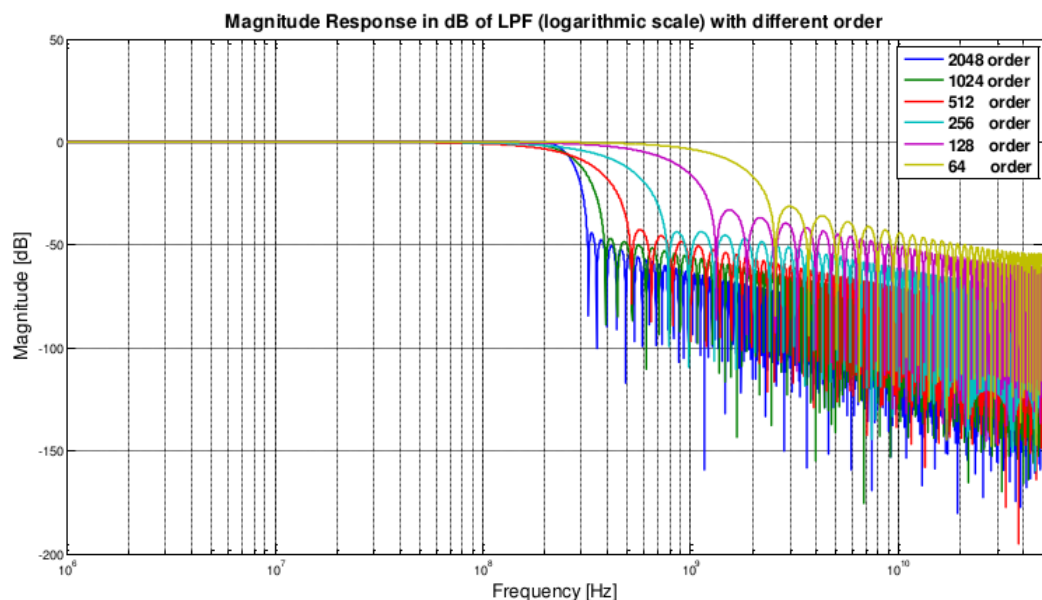


Figura 3.15. Respuesta en frecuencia de los filtros con diferente orden diseñados.

Además de representar gráficamente la respuesta en el dominio de la frecuencia del filtro, se deben determinar los parámetros de los filtros diseñados con los diferentes órdenes y conocer si cumplen con las especificaciones de diseño iniciales. Para ello, se ha creado una rutina en el lenguaje de Matlab (almacenado en un fichero .m) que permite obtener a partir de la respuesta del filtro en frecuencia los valores de frecuencia de corte, banda de transición, atenuación mínima en la banda de rechazo, rizado en la banda de paso e integral del error al cuadrado (ISE - Integral Square Error), este último parámetro será explicado en el siguiente capítulo. En el Anexo se pueden consultar las rutinas en Matlab

empleadas para el diseño de los filtros de este capítulo, así como el *script* generado para obtener los parámetros de los filtros digitales FIR.

Después de procesar las respuestas en el dominio de la frecuencia de los filtros digitales que se muestran en la Figura 3.15 con la rutina creada, se comprueba que cumplen en todos los casos con las especificaciones de frecuencia de corte y máximo rizado en la banda de paso, no así para la atenuación mínima en la banda de rechazo y la banda de transición tal como se aprecia en la Tabla 3.3.

Tabla 3.3. Relación entre el orden con la banda de transición y su banda de rechazo.

Orden	Banda de Transición	Atenuación en la Banda de Paso
2048	117 MHz	-44
1024	146 MHz	-46
512	249 MHz	-42
256	520 MHz	-43
128	1271 MHz	-33
64	2698 MHz	-31

La Tabla 3.3 permite seleccionar el orden mínimo de los filtros evaluados el cual permite cumplir con las especificaciones inicialmente establecidas para el filtro digital propuesto en el proyecto. El orden mínimo resultante funcional es 512.

Conclusiones

En este capítulo hemos visto una introducción al funcionamiento de los filtros digitales, dando a conocer los tipos que existen y las principales características de éstos.

Además, hemos hecho una introducción a los filtros digitales asíncronos y se ha explicado las propiedades de este tipo de filtros, indicando la arquitectura elegida y los bloques funcionales necesarios, centrándonos en el diezmador.

Una vez conocida la arquitectura y el tipo de filtro, se ha diseñado el filtro con las especificaciones de diseño indicadas en el capítulo 1. Para ello se ha empleado Matlab, software que nos ha permitido calcular los coeficientes y el orden del filtro.

Los cálculos y el diseño del filtro realizados en este capítulo corresponden a una primera aproximación ideal al diseño final ya que suponen que la cadena de retardos asíncronos es lineal y que los cálculos de los sumadores, multiplicadores y los coeficientes del filtro tienen una resolución infinita. En el siguiente capítulo se va a estudiar los efectos de la cuantificación del filtro, así como la no linealidad de los elementos de retardo.

4. Efectos de Cuantificación y no linealidades

Introducción

En el diseño del filtro FIR propuesto nos encontramos con dos importantes fuentes de no-linealidad, una primera fuente debida a la cuantificación del filtro, presente en todos los filtros digitales, y una segunda debida al uso de elementos de retardo asíncrono como son la cadena de inversores que se ha decidido emplear en nuestro diseño. Este último requiere especial estudio debido a lo característico que es y la reducida literatura que existe al respecto.

Las no-linealidades debido a la cuantificación comprende la cuantificación a la entrada del filtro, la de los coeficientes y la de la salida del filtro. En algunas ocasiones la cuantificación del filtro es trivial debido a que el filtro digital está implementado en ordenadores o equipos con alta prestaciones de cómputo, permitiendo el uso de una gran cantidad de bits y evitando así posibles errores de cuantificación. Sin embargo, en nuestro caso es un diseño *full-custom* implementado con elementos de retardo mediante líneas de retardo asíncronas.

De esta forma es indispensable minimizar el número de bits para reducir la complejidad en el diseño a nivel de transistor al mínimo posible, sin olvidar que el rendimiento del filtro debe continuar cumpliendo con las especificaciones de diseño iniciales.

Por otro lado, la no-linealidad causada por los elementos de retardo, requiere de un análisis previo que nos ayude a determinar los posibles efectos en el filtro digital diseñado, más concretamente en las especificaciones de diseño deseadas. Para ello se va a estudiar las variaciones de tiempo de retardo que pueden sufrir los elementos de retardo por el empleo de una cadena de inversores. En el capítulo siguiente se explicarán de manera más concreta las variaciones que sufren los elementos de retardo por emplear puertas lógicas.

Cuantificación del filtro digital

Los diseños de filtros digitales normalmente se realizan suponiendo una anchura de palabra infinita o muy grande. En la práctica, los sistemas están restringidos a un conjunto finito de valores discretos limitado por la longitud de los registros (precisión finita). De esta manera es necesario seleccionar un número finito de bits en la entrada del filtro, en los coeficientes y en la salida del filtro. Los efectos de la cuantificación son muy significativos debido a que dicho proceso transforma el filtro en un sistema no lineal, pudiendo tener consecuencias nefastas en el rendimiento del filtro digital, llegando incluso a que un filtro estable se convierta en inestable [44] [45].

Generalmente los efectos de la cuantificación son complejos de analizar y cuantificar, pero existen técnicas sin una elevada complejidad y mediante simulaciones con software específico que permiten minimizar las perturbaciones que ocasionan, siempre y cuando la contribución de los efectos de cuantificación no sea significativa respecto a la señal. Los errores de cuantificación en un filtro pueden ser divididos según su origen en cuantificación en la entrada, cuantificación de los coeficientes y la precisión finita en las operaciones aritméticas del filtro [46]. El caso del primero origen indicado, cuantificación en la entrada, no será estudiada en el presente proyecto fin de carrera, teniendo en cuenta que lo consideramos dentro del ámbito del convertidor Sigma-Delta previo al filtro, ya descrito en el capítulo 2, y suponemos una señal ya cuantificada correctamente a la entrada del filtro, considerando las especificaciones de diseño indicadas en el capítulo 1.

Con el fin de limitar los efectos de cuantificación y favorecer un diseño adecuado, inicialmente se va a analizar el formato de los datos y su representación numérica. A continuación, se estudiará la cuantificación de los coeficientes y finalmente se realiza la

elección del número de bits de salida para evaluar el error de cuantificación por precisión finita en las operaciones aritméticas del filtro.

Formato de datos y representación numérica

En el diseño de sistemas de procesamiento digital de señales existen dos tipos de representación numérica:

- Aritmética en coma flotante: En general la representación en coma flotante está basada en la notación exponencial. En la notación exponencial cualquier número real diferente de cero puede ser expresado en sistema decimal como muestra la siguiente ecuación, 4.1.

$$r = c \cdot b^e \quad (4.1)$$

donde ‘b’ es la base, en el sistema lineal 10, ‘e’ es un número entero y se denomina exponente y ‘c’ es un número real con un solo dígito entero seguido de una coma y varios dígitos fraccionarios (noción científica estandarizada), denominado coeficiente o mantisa.

Las características más destacables en la implementación de filtros digitales son:

- Después de operaciones aritméticas requiere normalización de los números.
 - Se emplean donde la precisión y el rango dinámico son determinantes.
 - Es necesario más algoritmos para su implementación que en la aritmética de punto fijo.
 - Facilidad en la codificación.
 - Mayor necesidad de área, frecuencia y potencia.
- Aritmética de punto fijo: La aritmética de punto fijo proporciona una alternativa a la aritmética en coma flotante. Los números en coma flotante almacenan dos valores, el exponente y la mantisa, la multiplicación de la mantisa por el exponente define su valor numérico. Los números de punto fijo solo emplean una parte de sus dígitos disponibles exclusivamente para representar la parte entera y el resto de

dígitos para la parte fraccionaria. En la ecuación 4.2 se puede observar la forma de representar un número real.

$$R = e_{n-1}e_{n-2}\dots e_2e_0, f_{-1}f_{-2}f_{-p} \quad (4.2)$$

donde ‘e’ son los valores binarios de la parte entera, ‘f’ los valores binarios de la parte fraccionaria y ‘n’ y ‘p’ son el número de bits de la parte entera y fraccionaria respectivamente.

En el caso de la aritmética de punto fijo las características a tener en consideración para la implementación de filtros digitales son:

- La parte decimal es fija.
- Mayor simplicidad en hardware, baja potencia y menor área necesaria.
- Las Multiplicaciones doblan en número de bits empleado (NxN bits produce 2N bits).
- Código menos entendible, necesario control de desbordamiento y escalado.

Considerando las características de cada tipo de representación numérica, para el presente proyecto se ha decidido que la aritmética de punto fijo es la más adecuada, principalmente por su menos compleja implementación a nivel de transistores, menor consumo y menor área, en comparación con la aritmética en coma flotante.

Cuantificación de los coeficientes

La decisión de emplear aritmética de punto fijo en el diseño propuesto obliga a realizar una cuantificación de los coeficientes con un número finito de bits, debido a que los coeficientes calculados en el capítulo anterior emplean números en punto flotante de precisión doble. A continuación, analizaremos la cuantificación de los coeficientes para minimizar los errores por cuantificación en el rendimiento de la respuesta en frecuencia del filtro que permita mantener las especificaciones de diseño indicadas en el capítulo 1.

Como se pudo observar en la Figura 3.15 del capítulo anterior, la atenuación mínima en la banda de rechazo alcanzada con los coeficientes calculados es de 42dB. Teniendo en cuenta esta atenuación mínima en la banda de rechazo y considerando la ecuación 4.3 [47],

podemos realizar una primera estimación inicial de cuantos son los bits mínimos necesarios en la cuantificación de los coeficientes para proporcionar al filtro un adecuado rango dinámico. Según esta regla se asume que es necesario un bit por cada 5dB de atenuación. Así el valor calculado es:

$$Bits_{min.} = \frac{AtenuacinMin.BandaRechazo}{5dB/bit} = \frac{42dB}{5dB/bit} = 8,4 \approx 9 \quad (4.3)$$

Como se puede observar en la ecuación 4.3 el mínimo exigible según esta regla es 9 bits, para comprobarlo y mostrarlo gráficamente de forma más clara, en la Figura 4.1 se muestra una serie de simulaciones con diferente cuantificación de coeficientes, 8, 10, 12, 14 y 16 bits realizadas con la función '*qfilt*' que proporciona el *toolbox Filter Design* de Matlab. En la tabla 4.1 se indica la sintaxis y parámetros que emplea la función '*qfilt*'. Aunque la norma establece 9 bits para lograr un adecuado margen dinámico, la cuantificación con ese número de bits puede deteriorar ciertas especificaciones del filtro, pudiendo incumplir algunas de las especificaciones de diseño del filtro o no logrando una respuesta en frecuencia del filtro lo suficientemente fina para algunas aplicaciones.

Tabla 4.1. Funciones *qfilt* de Matlab.

Función	Sintaxis	Argumentos
Generar filtro cuantificado	<code>qfilt('Struce',{Coef},'prop1',value1,...)</code>	Struc: Estructura del filtro. Coef: Coeficientes del filtro. prop1: Propiedades. value1: Valor propiedad1
Configura el tamaño de palabra y tamaño parte fraccionaria	<code>qfilt('quantizer',[f_length length])</code>	length: Tamaño de la palabra (n+ bits) f_length: Tamaño parte fraccionaria.

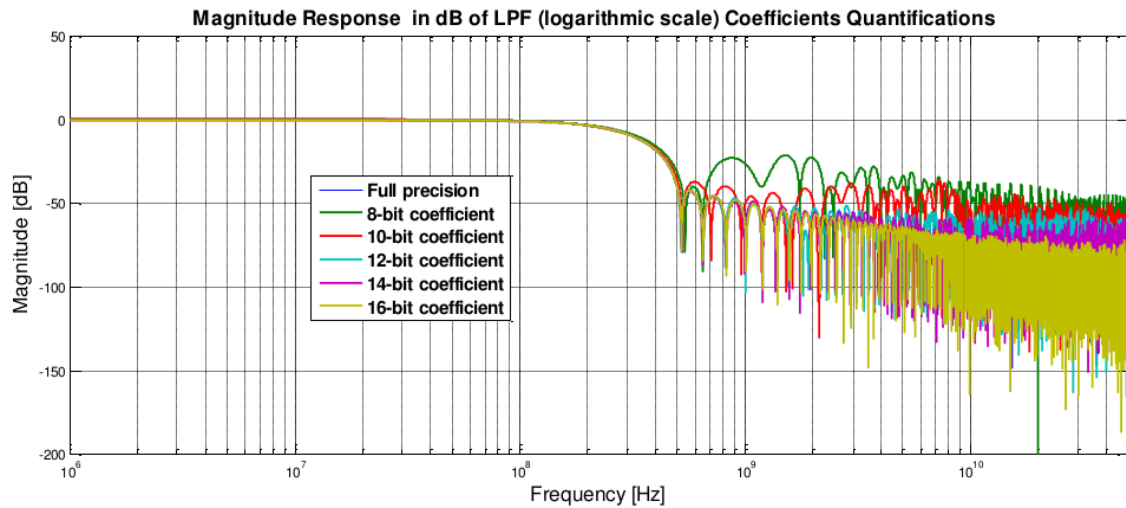


Figura 4.1. Respuesta en frecuencia del filtro para diferente cuantificación de coeficientes.

Analizando más en detalle la Figura 4.1 se puede apreciar, que la respuesta en frecuencia del filtro sufre una distorsión evidente con una cuantificación en los coeficientes de 8 bits.

Esta distorsión se produce principalmente en la atenuación en la banda de rechazo. Por otro lado, la cuantificación de los coeficientes con 10 bits mejora la atenuación en la banda de rechazo, pero no es posible identificar visualmente si cumple con las especificaciones indicadas en el capítulo 1.

Para ser capaces de evaluar mejor si cumple las especificaciones deseadas, la respuesta en frecuencia de los filtros con los coeficientes cuantificados ha sido procesada por la rutina generada en el proyecto, la cual ya fue empleada en el capítulo anterior para caracterizar mejor el filtro y, como se indicó, permite obtener a partir de la respuesta en frecuencia del filtro los valores de frecuencia de corte, banda de transición, atenuación mínima en la banda de rechazo y rizado. Además, para determinar mejor la distorsión que se produce en el filtro, se ha añadido la integral del error al cuadrado (ISE, *integrated squared error*) en módulo y en fase, que vendrá dada por la ecuación 4.4, donde H es la respuesta del filtro después de ser cuantificado y H_i es la respuesta del filtro ideal, sin cuantificar.

$$ISE = \frac{2}{f_s} \sum |H_i - H|^2 \quad (4.4)$$

Los valores obtenidos para las diferentes cuantificaciones de los coeficientes del filtro propuesto se muestran en la Tabla 4.2.

Tabla 4.2. Parámetros del filtro para diferente cuantificación de coeficientes.

Filtro	Coefi. Bits	Frec. Corte [MHz]	Banda Trans. [MHz]	Atenu. [dB]	Rizado [dB]	ISE Magnit.	ISE Fase
Filtro Precisión doble	-	256	249	-42,46	0	0	0
Filtro cuantificado coeficientes - 8 bits	8	266	228	-21,33	0,05	3,40E-10	2,00E-38
Filtro cuantificado coeficientes - 10 bits	10	254	256	-36,3	0,088	6,10E-11	9,80E-39
Filtro cuantificado coeficientes - 12 bits	12	256	257	-42,11	0,012	9,50E-13	1,40E-38
Filtro cuantificado coeficientes - 14 bits	14	256	255	-42,26	0,006	4,60E-13	1,60E-38
Filtro cuantificado coeficientes - 16 bit	16	256	255	-42,46	0,001	0	2,00E-38

Con los valores de los parámetros de la tabla 4.2 ya somos capaces de seleccionar un número de bits adecuado para la cuantificación de los coeficientes del filtro.

Como muestra la columna de la banda de atenuación para 8 y 10 bits, los resultados no son óptimos. Aunque la ecuación 4.3 indica que con 10 bits se cumple con la especificación del margen dinámico, la tabla muestra que la banda de atenuación se ve reducida a -36.30 dB, incumpliendo así una de las especificaciones de diseño proporcionada en el capítulo 1. Sin embargo, si elegimos usar 12 bits para cuantificar los coeficientes del filtro, además de cumplir con la atenuación mínima en la banda de rechazo, tendremos un margen para los efectos de las no-linealidades que aún no se han cuantificado.

Por otro lado, las integrales del error al cuadrado tienen un valor muy reducido, y el rizado y la banda de transición, aunque no se mantiene estrictamente dentro de las

especificaciones de diseño, son tan pequeños los errores que se ha dado por satisfactorio para continuar.

Cuantificación de salida y operaciones aritméticas

Para finalizar la cuantificación en el filtro digital se debe realizar la cuantificación de las operaciones aritméticas, las cuales nos van a permitir seleccionar el número de bits de salida y el número de bits necesarios en las operaciones aritméticas internas (en nuestro diseño nos referimos a los sumadores). Aunque la estructura del diseño propuesto es singular, por la modificación puntual de la ubicación del diezmador, está basado en la estructura de forma directa. Dada esta estructura y continuando con la decisión de emplear aritmética de punto fijo, en la Figura 4.2 se puede observar la estructura de cuantificación con el número de bits que deben tenerse en consideración durante la cuantificación de las operaciones aritméticas.

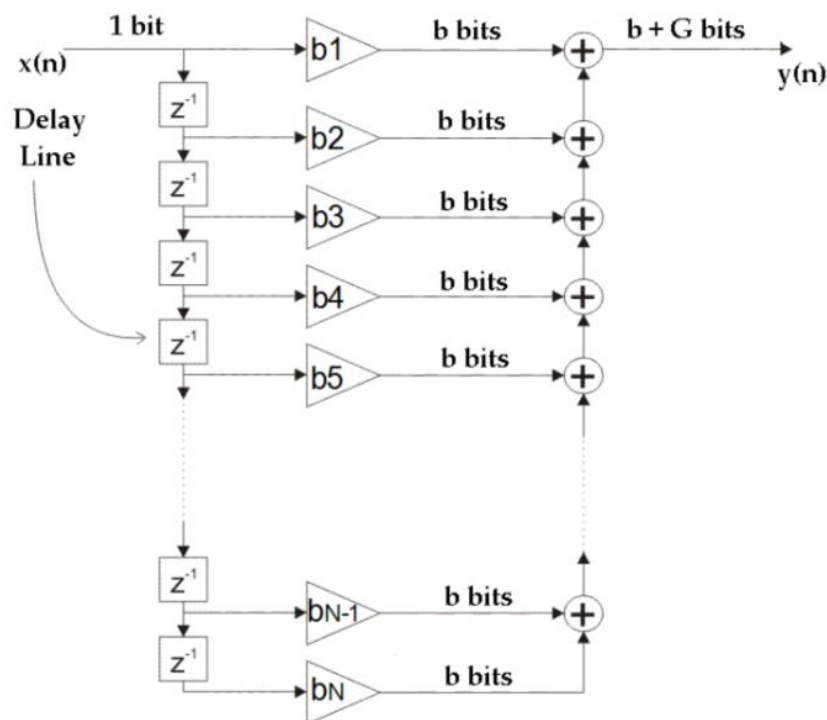


Figura 4.2. Estructura de forma directa cuantificada para el filtro FIR diseñado [48].

Debido a que la entrada del filtro digital es de un bit, no es necesario realizar cuantificación por la multiplicación ya que el número de bits de salida de la multiplicación siempre va a ser el mismo número de bits de los coeficientes cuantificados, evitando en estos bloques posibles errores de redondeo. Descartados los errores de cuantificación en

los bloques multiplicadores, los efectos negativos de la cuantificación se van a concentrar en los bloques sumadores.

En general, para evitar los errores por redondeo cuando se realiza una suma entre dos números en punto fijo con el mismo número de bits se añade un bit extra, lo que permite contemplar el bit de acarreo que puede producir la suma. Así para un filtro de orden N , se deberían de emplear el número de bits de los valores de los sumandos más $N-1$ bits adicionales. Sin embargo, en la práctica no es necesario incrementar un bit por cada suma para evitar errores de redondeo. El incremento se puede calcular a partir de la ecuación 4.5:

$$G = \log_2(N) + 1 = \log_2(512) + 1 = 10 \quad (4.5)$$

Esta ecuación con crecimiento logarítmico en base 2 se puede entender mejor si se aplica para las sumas una arquitectura de árbol de sumadores. En la Figura 4.3 se puede observar este efecto con la suma de 4 números representados con B bits. Para lograr el resultado final de la suma de esos 4 números será necesario tres sumas e inicialmente se podría pensar que es necesario 3 bits adicionales. Sin embargo, en la Figura 4.3 se puede apreciar que con 2 bits adicionales es suficiente.

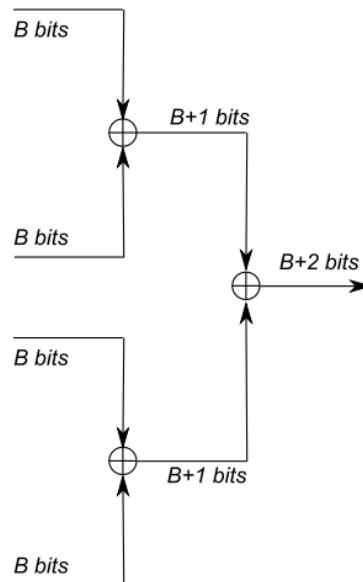


Figura 4.3. Árbol de sumadores, para 4 sumandos indicando el número de bits necesarios [48].

Teniendo en cuenta esto y aplicando la ecuación 4.5, el valor obtenido de G es de 10. Si a continuación realizamos la suma de $G + b$ como indica la salida del filtro en la Figura, se obtiene una salida de 22 bits. Una cuantificación con sumadores y a la salida del filtro a 22 bits no parece adecuada, debido al notable aumento de tamaño del circuito y su consumo asociado.

Existen métodos que permiten reducir la G de la Figura 4.2 y, por consiguiente, la salida del filtro. En nuestro caso para obtener precisión infinita obtuvimos una cuantificación en los bloques sumadores de 22 bits, pero antes de seleccionar esta cuantificación se va a intentar reducir basándonos en la ecuación 4.5, la cual indica una convolución para un filtro FIR con un orden de $N + 1$:

$$y[n] = \sum_{m=0}^N h[m]x[n - m] \quad (4.5)$$

Mediante la ecuación 4.5, si somos capaces de determinar los valores de entrada, es posible calcular la máxima salida que se va a obtener y, de esta manera, conocer el número de bits necesarios para una cuantificación correcta. Para ello, inicialmente se debe pasar a valor absoluto cada uno de los miembros de la ecuación y, a continuación, se aplica la desigualdad del triángulo para la norma 1, quedando como se muestra en la ecuación 4.6.

$$|y[n]| \leq \left| \sum_{m=0}^N h[m] \cdot x[n - m] \right| = \sum_{m=0}^N |h[m] \cdot x[n - m]| = \sum_{m=0}^N |h[m]| \cdot |x[n - m]| \quad (4.6)$$

Finalmente, y partir de la ecuación 4.6, se debe aplicar en para x el rango máximo que se tiene para las entradas normalmente viene expresado por $[-R/2, R/2]$, donde R es el rango que puede cubrir la entrada (en nuestro caso hace referencia a los coeficientes). Al aplicar obtenemos la ecuación 4.7, donde solo queda calcular la salida máxima del filtro. Esto se realiza mediante la suma todos los valores absolutos de los coeficientes (norma 1) y, finalmente se multiplica por el rango de entrada.

$$|y[n]| = \frac{R}{2} \sum_{m=0}^N |h[m]| = \frac{R}{2} \cdot \|h[n]\|_1 = 1 \cdot 4,096 = 4,096 \quad (4.7)$$

Con el intervalo de salida calculado y conociendo el intervalo de entrada, simplemente es necesario obtener el factor de ganancia de la salida respecto a la entrada y sumárselo al número de bits ya cuantificado de los coeficientes. En el caso del filtro digital propuesto $R/2$ es igual a 1, debido a que la entrada es de 1 bit y la norma 1 de los coeficientes del filtro es igual a 4,096. Teniendo en cuenta estos valores, con un incremento de 2 bits es posible cubrir el rango de $[-4,4)$, valor muy cercano al 4,096 calculado. De esta manera el número de bits sería 14: 12 bits de los coeficientes más los 2 calculados.

Notar que, con los dos bits añadidos, aunque está muy próximo, no consigue alcanzar el valor máximo de la salida calculada por la norma 1, pudiendo ocasionar una respuesta en frecuencia del filtro no adecuada, haciendo necesario añadir algún bit más.

Para verificar si son correctos los cálculos de cuantificación realizados, se han realizado simulaciones de la densidad espectral de potencia de ruido aleatorio gaussiano a la salida del filtro, con diferente número de bits. Para ello se ha empleado la función del *toolbox DSP System* denominada *noisepsd*. También se ha calculado la varianza del ruido de cuantificación, introduciendo datos aleatorios en la entrada del filtro cuantificado y, por otro lado, los mismos datos aleatorios en el filtro sin cuantificar, permitiendo así calcular la varianza de las diferencias entre las señales de salida. Finalmente se implementa la respuesta en frecuencia del filtro con los coeficientes cuantificados según el número de bits calculado en el apartado anterior, además de comprobar con diferentes cuantificaciones para la salida del filtro y en los bloques sumadores, entre ellas la obtenida en este apartado. Con cada implementación se han obtenido sus parámetros característicos. En la Tabla 4.3 se muestra el ruido de los filtros con diferentes números de bits de salida y sus parámetros característicos.

Tabla 4.3. Parámetros del filtro para las diferentes cuantificaciones de salida.

Filtro	Preci. Doble	LPF 8 bits salida	LPF 10 bits salida	LPF 12 bits salida	LPF 14 bits salida	LPF 16 bits salida	LPF 32 bits salida
Bits de Salida	-	8	10	12	14	16	32
Bits Coeficientes	-	12	12	12	12	12	12
Densidad espectral de potencia de ruido	-210	-49	-42	-52	-62	-71	-158
Varianza del ruido de cuantificación	0	4,69	0,279	0,0224	9,98E-08	9,98E-08	9,98E-08
Frecuencia de corte [MHz]	256	-	-	245	247	255	254
Banda de transición [MHz]	249	-	-	255	253	255	249
Mínima atenuación B de rechazo [MHz]	-42,46	-	-	-21	-34,50	-40,20	-42,11
Rizado [dB]	0	-	-	0,64	0,14	0,015	0,012
ISE Magnitud	0	6,62E-7	5,84E-08	1,70E-09	8,53E-11	8,48E-12	1,07E-12
ISE Fase	0	$3.31e-05$	1,44E-07	2,41E-09	9,10E-11	1,08E-11	2,97E-21

Como se puede observar en la Tabla 4.3, en términos de varianza de ruido con 14 bits a la salida del filtro se mantiene sin cambio respecto a 16 bits y 32 bits, cumpliendo así con los cálculos que se han realizado en este apartado para la aritmética en punto fijo de los sumadores. Por otro lado, sí se observa diferencia en la densidad espectral de potencia de ruido aleatorio simulada, observando con 32 bits notables mejoras en los resultados respecto a 16 bits o 14 bits, siendo la diferencia entre estos últimos prácticamente despreciable. Aunque parece que una cuantificación de 14 bits a la salida es adecuada según la Tabla 4.3, hay un parámetro importante que no cumple con las especificaciones de diseño propuestas en el capítulo 1: la atenuación mínima en la banda de rechazo. Con 14 bits se obtiene -34,5 dB, que es mayor que los -40 dB indicados en las especificaciones. Debido a esto se ha seleccionado a la salida y en los bloques sumadores una cuantificación en punto fijo de 16 bits.

En la Figura 4.4 se muestran las respuestas en frecuencia del filtro con las diferentes cuantificaciones indicadas en la Tabla 4.3.

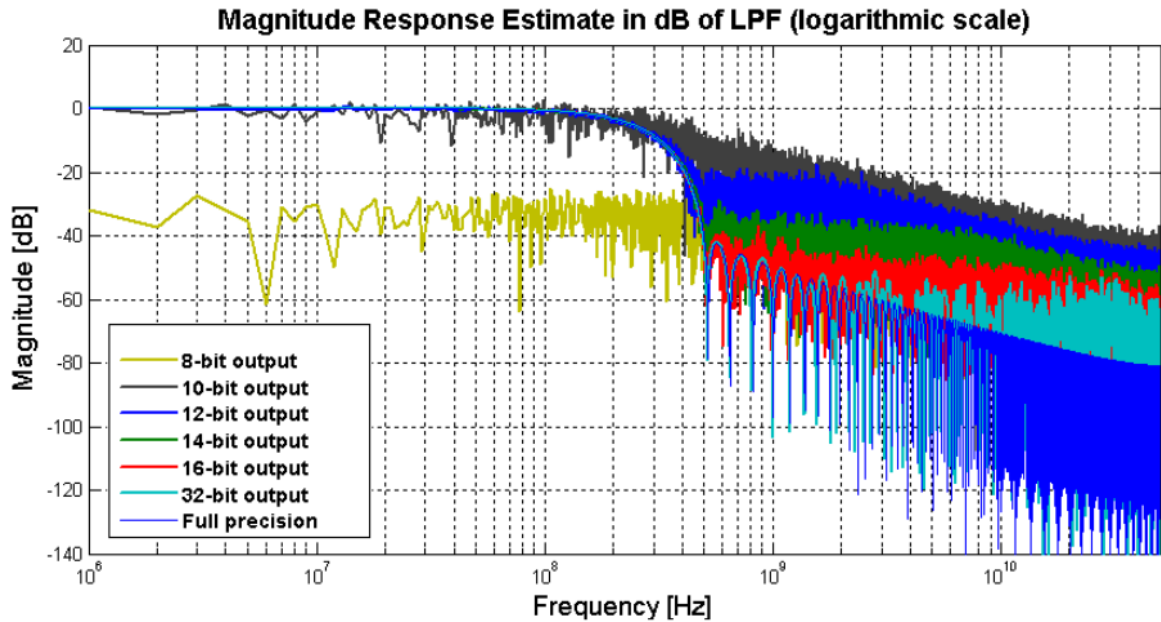


Figura 4.4. Respuesta en frecuencia del filtro para diferentes cuantificaciones de salida.

No-linealidad en los elementos de retardo

La característica más significativa del filtro digital con alta tasa de muestreo y diezmado propuesto es la utilización de elementos de retardo mediante puertas inversoras sin sincronismo (reloj) externo en lugar de los clásicos sistemas síncronos de retardo, usualmente implementados a bajo nivel mediante flip-flops. Dado que cada elemento de retardo es un inversor, al concatenarlos como indica la estructura de la forma directa en la que se basa nuestro diseño del filtro FIR, lo que tenemos es una cadena de inversores.

En una cadena de inversores, cada elemento de retardo está formado por un inversor el cual tiene asociado un tiempo de retardo nominal. Este retardo nominal no es el retardo real, o retardo pin a pin. Definimos retardo real como el retardo nominal más las variaciones por las no-linealidades de los circuitos integrados, debido a múltiples causas como pueden ser por variaciones ambientales, variaciones en el proceso de fabricación, etc. De esta manera, y teniendo en cuenta la variación existente en el retardo nominal se puede realizar una caracterización temporal de los elementos de retardo según su

desviación estándar asociada. La inclusión en los elementos de retardo de una desviación estándar en su tiempo de retardo asociado produce efectos no deseados en los filtros digitales pudiendo ser más notorios en un filtro FIR con 512 elementos de retardo como se propone en este proyecto fin de carrera.

En la siguiente sección se estudiará la distorsión en la respuesta en frecuencia y su influencia en los parámetros más significativos del filtro FIR diseñado, según la desviación estándar en los elementos de retardos. Para su estudio haremos uso del programa Matlab, asignando diferentes valores de desviación estándar a los elementos de retardo y analizando la respuesta del filtro. En el capítulo 5 se explicará con más detalle la implementación de los elementos de retardo a bajo nivel y la influencia de las variaciones por el proceso de fabricación en el retardo de los inversores.

Desviación estándar

Con el objetivo de estudiar el efecto de la variación del retardo nominal de los elementos de retardo en el filtro FIR diseñado, se ha empleado la medida de desviación estándar. La desviación estándar es una medida de dispersión empleada en estadística que nos indica cuánto se alejan los tiempos del retardo reales del retardo nominal en el conjunto de los elementos de retardo.

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\tau_i - \mu)^2} \quad (4.8)$$

En la expresión 4.8 σ es la desviación estándar, N es el número de elementos de retardo, τ es el retardo real de cada elemento y μ es el retardo nominal. La desviación estándar es una medida más estable que otras medidas empleadas en estadísticas como por ejemplo el recorrido. En nuestro caso nos proporciona una forma útil de caracterizar los elementos de retardo y poder estudiar sus efectos en el filtro según la variación en sus elementos de retardo.

Implementación en Matlab

En este apartado veremos la implementación que se debe realizar del filtro digital diseñado para poder estudiar su respuesta y parámetros al producirse una variación en el retardo nominal en los elementos de retardo del filtro.

Para lograr realizar las simulaciones se ha tenido que modificar la estructura del filtro mediante un incremento de los elementos de retardo. El incremento de retardos proporciona un aumento en la frecuencia de muestreo, reduciendo así el tiempo de retardo y permitiendo a los coeficientes del filtro desplazarse entre retardos menores a los 10ps, valor indicado en las especificaciones de diseño según su frecuencia de muestreo. Este método permite simular variaciones de los retardos menores al de la frecuencia de muestreo del filtro. En la Figura 4.5 se muestra el método indicado en la estructura de un filtro FIR.

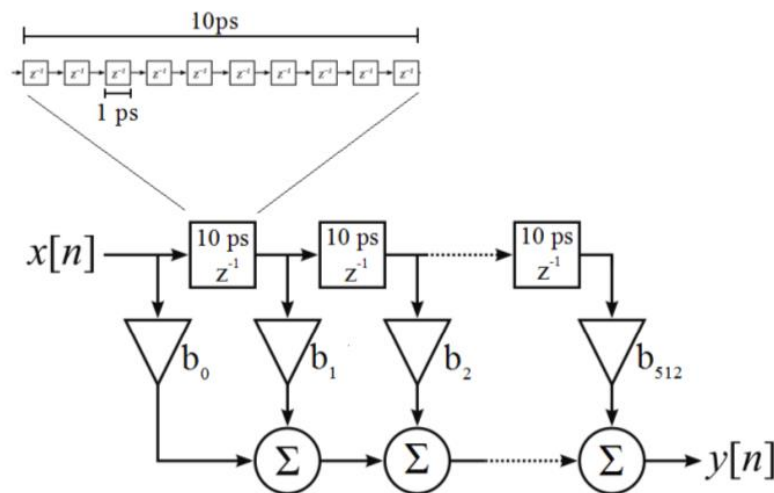


Figura 4.5. Incremento de elementos de retardo en la estructura de un filtro FIR.

Las funciones empleadas en Matlab para el diseño de filtros no contemplan un método estándar para añadir elementos de retardo aislados como se propone. Para ello se ha tenido que agregar nuevos coeficientes aumentando así la resolución. De esta manera si se desea una resolución de 0,1ps se debe añadir 100 coeficientes (resultado de dividir la resolución actual por la deseada). Los nuevos coeficientes deben ser anulados para que mantenga el mismo número de ceros y la respuesta en frecuencia sea la previamente diseñada. Como ya hemos visto en el capítulo 3, la ecuación de salida de un filtro FIR viene dada por la

ecuación 4.9 y, teniendo en cuentas las modificaciones indicadas, la expresión matemática para la salida del filtro resultante se puede expresar como muestra la ecuación 4.10.

$$y[n] = b_0 \cdot x[n] + b_1 \cdot x[n] + b_2 \cdot x[n] + \dots + b_N \cdot x[n] \quad n = 0, 1, 2, 3, \dots \quad (4.9)$$

$$\begin{aligned} y_u[n] = & b_{00} \cdot x_u[n] + b_{01} \cdot x_u[n-1] + b_{02} \cdot x_u[n-2] + \dots \\ & + b_{0(L-1)} \cdot x_u[n-(L-1)] + b_{10} \cdot x_u[n-(L-1)-1] + b_{11} \cdot x_u[n-(L-1)-2] + \dots + b_{1(L-1)} \cdot x_u[n-(L-1)-L] + \dots \\ & + b_{(N)(L-1)} \cdot x_u[n-(L \cdot N) - 1] \end{aligned} \quad (4.10)$$

donde N es el número de coeficientes del filtro original y L es el factor de sobremuestreo empleado.

Por otro lado, si sobre la entrada del filtro $x[n]$ se realiza un sobremuestreo, se obtiene $x_u[n]$ definido por la ecuación:

$$x_u[n] = \begin{cases} x\left(\frac{n}{L}\right) & n = 0, \pm L, \pm 2L, \pm 3L, \dots \\ 0 & \text{en otro caso} \end{cases} \quad (4.11)$$

donde L es el factor de sobremuestreo.

Si consideramos en la ecuación 4.10 que b_{Nx} es igual a 0 (donde $x=1,2,3,4 \dots L-1$), debido a la anulación de coeficientes comentada y que la entrada al filtro modificado cumple con la ecuación $x_u[n]$ de la ecuación 4.11, al sustituir se comprueba que la ecuación 4.10 y 4.9 son equivalentes y se puede verificar que $y_u[n]=y[n]$ y $h_u[n]=h[n]$.

Verificado que el método empleado no afecta a la respuesta del filtro original, a continuación, para realizar la variación del retardo nominal en el filtro se debe desplazan los coeficientes del filtro original entre los nuevos coeficientes nulos añadidos. Si realizamos este desplazamiento en la ecuación 4.10 se obtiene la ecuación 4.12.

$$\begin{aligned} y_u[n] = & b_{02} \cdot x_u[n] + b_{01} \cdot x_u[n-1] + b_{00} \cdot x_u[n-2] + \dots \\ & + b_{0(L-1)} \cdot x_u[n-(L-1)] + b_{11} \cdot x_u[n-(L-1)-1] + b_{10} \cdot x_u[n-(L-1)-2] + \dots \\ & + b_{1(L-1)} \cdot x_u[n-(L-1)-L] + \dots + b_{(N)(L-1)} \cdot x_u[n-(L \cdot N) - 1] \end{aligned} \quad (4.12)$$

Estos desplazamientos de los coeficientes nos permiten variar el retardo nominal y, dependiendo del número de desplazamientos que se realice, la desviación estándar tomará el valor deseado por nosotros. En la Figura 4.6 se muestra gráficamente el desplazamiento de los coeficientes comentado e indicados por las ecuaciones 4.10 (sin desplazar) y 4.12 (desplazados).

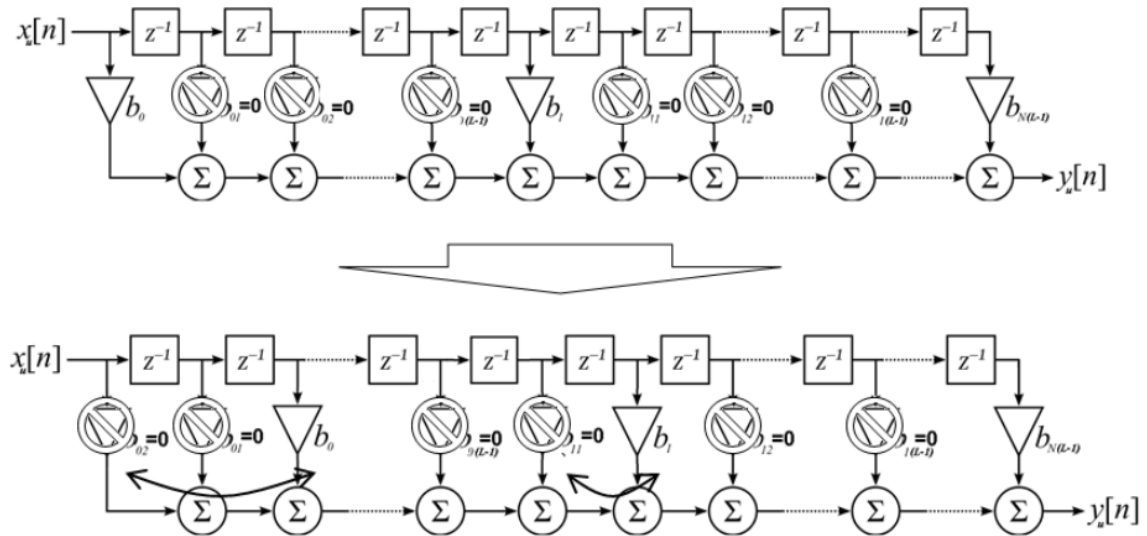


Figura 4.6. Desplazamiento de los coeficientes para modificar el tiempo de retardo.

Resultados de las simulaciones

El método explicado en el apartado anterior ha permitido analizar la respuesta en frecuencia del filtro diseñado, así como la de sus parámetros más importantes cuando se produce una variación en el tiempo de retardo de los elementos de retardo por las no idealidades. Para determinar estos efectos en el filtro se han tenido que simular en Matlab alrededor de 100 filtros con diferentes valores de la desviación estándar obtenidos alrededor del retardo nominal de 10ps. Una vez generadas estas simulaciones se han obtenido la respuesta en frecuencia y los parámetros característicos como son frecuencias de corte, banda de transición, atenuación mínima en la banda de rechazo, rizado e integral del error al cuadrado (ISE, *integrated squared error*) en módulo y en fase. En el Anexo podremos ver la rutina de Matlab empleada.

En la Figura 4.7 se muestra la respuesta en frecuencia del filtro con algunas de las simulaciones generadas para diferentes valores de la desviación estándar. En la Tabla 4.4 se pueden observar los parámetros obtenidos del filtro para los valores de desviación estándar más relevantes.

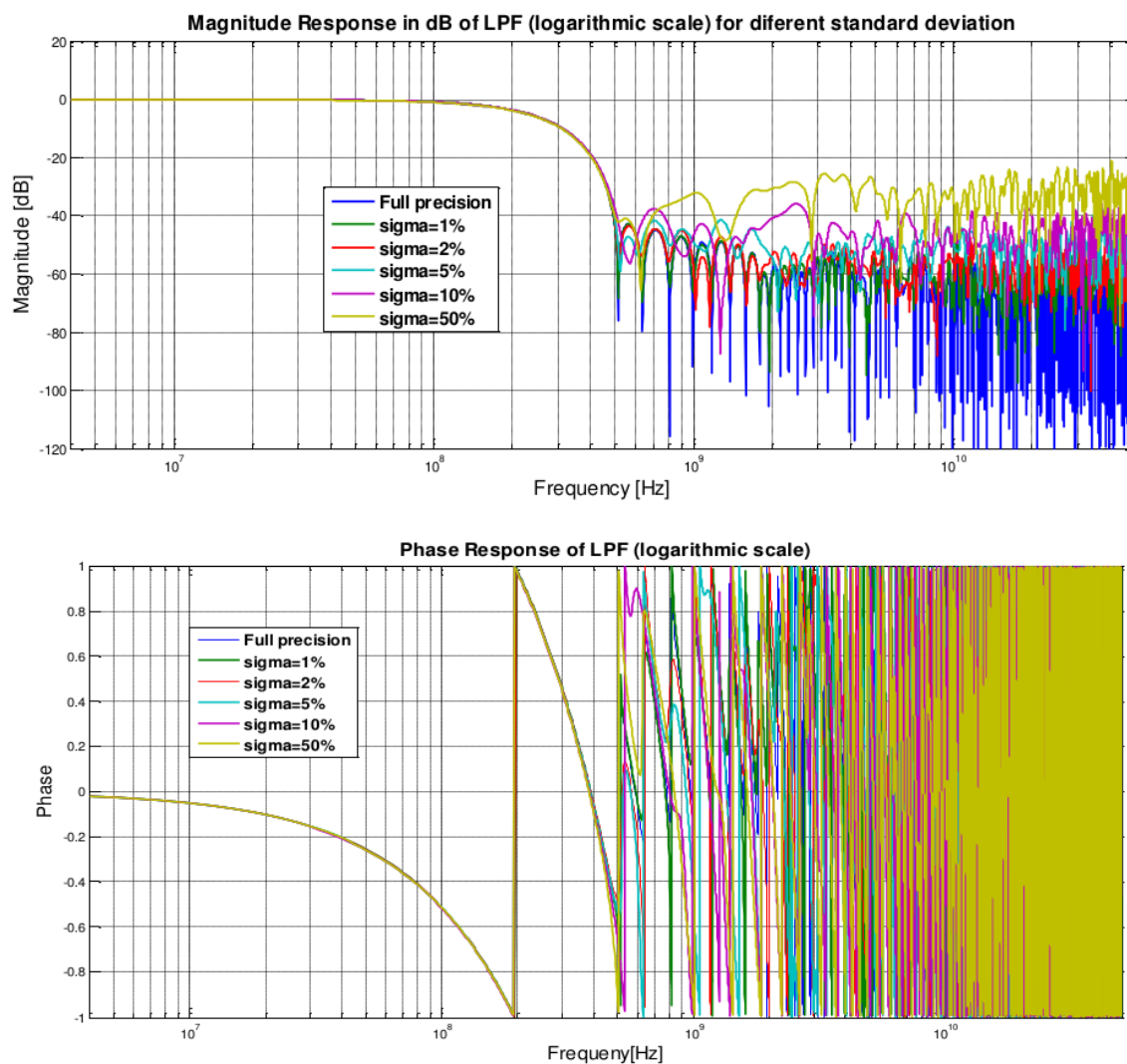


Figura 4.7. Respuesta del filtro en frecuencia y en fase con diferentes valores de desviación estándar.

Tabla 4.4. Parámetros del filtro para los diferentes valores de la desviación estándar.

Filtro	Coefi. Bits	Frec. Corte [MHz]	Banda Trans. [MHz]	Rizado [dB]	Atenu. [dB]	ISE Magnit.	ISE Fase
Filtro Precisión doble (ideal)	0,00%	250	249	0,012	42,84	0	0
Filtro asíncrono con desviación estándar	0,50%	250	275	0,012	42,85	3,50E-14	7,00E-15
Filtro asíncrono con desviación estándar	1,00%	250	275	0,012	42,94	1,10E-13	1,60E-13
Filtro asíncrono con desviación estándar	2,00%	250	253	0,012	43,05	5,50E-13	1,90E-11
Filtro asíncrono con desviación estándar	5,00%	250	253	0,012	40,53	5,40E-12	2,70E-11
Filtro asíncrono con desviación estándar	10,00%	252,5	251	0,012	36,37	1,76E-11	6,30E-14
Filtro asíncrono con desviación estándar	20,00%	250	234	0,012	29,41	8,00E-11	3,60E-11
Filtro asíncrono con desviación estándar	50,00%	245	188	0,039	22,04	4,40E-10	4,50E-10

Para obtener los resultados mostrados se realizaron alrededor de 100 simulaciones con una desviación estándar comprendida entre 50fs a 5ps en intervalos de 50fs, es decir, si consideramos en porcentaje respecto al retardo nominal de 10 ps, resulta del 0,5% al 50%. Analizando los datos de los parámetros proporcionados en las simulaciones se puede apreciar que no existen importantes modificaciones hasta una desviación estándar de aproximadamente el 5%, donde se distingue una mayor variación de la respuesta del filtro según su ISE de magnitud. Por otro lado, y siendo más significativo, se puede observar un descenso de la atenuación mínima de la banda de rechazo, constatando que a partir del 5% de desviación estándar se deja de cumplir con las especificaciones de diseño indicadas en el capítulo 1. Con el fin de observar de manera más clara esta tendencia de la atenuación mínima de la banda de rechazo, en la Figura 4.8 se muestra la atenuación mínima de la banda de rechazo respecto a la desviación estándar de los elementos de retardo.

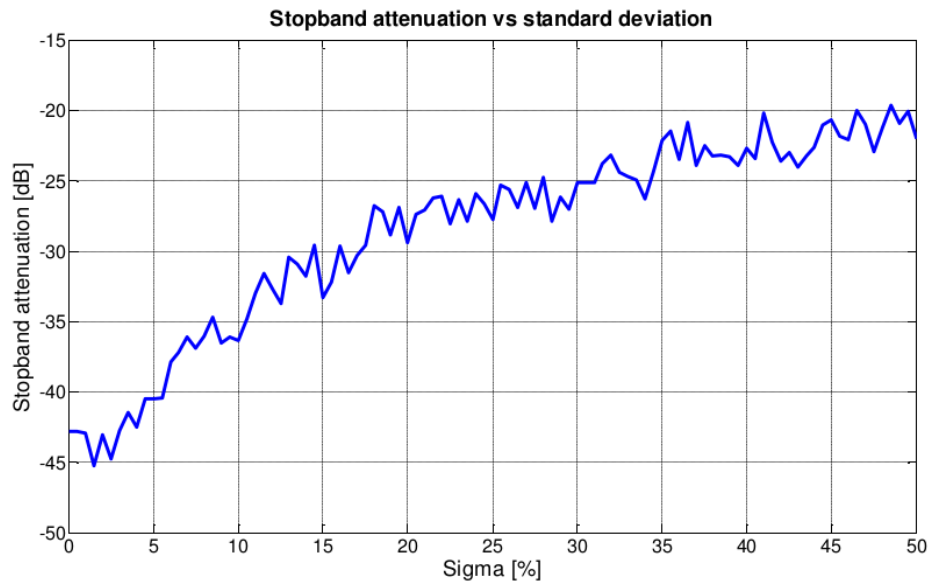


Figura 4.8. Atenuación mínima en la banda de rechazo en función de la desviación estándar de los elementos de retardo.

La Figura 4.8 confirma que, si la desviación estándar es superior al 5%, la atenuación de la banda de rechazo disminuye por debajo de -40 dB, incumpliendo así las especificaciones de diseño.

Conclusiones

En este capítulo se ha cuantificado el filtro digital diseñado idealmente en el capítulo anterior, obteniéndose el número de bits de los coeficientes, así como el de la salida del filtro para que cumpla con las especificaciones de diseño.

Además, se ha caracterizado los parámetros del filtro digital diseñado según la desviación estándar del tiempo de retardo de los elementos de retardo del filtro. Para ello se han realizado simulaciones en Matlab con diferentes desviaciones estándar en los elementos de retardo y se ha verificado cuales son admisibles para cumplir con las especificaciones de diseño.

En el siguiente capítulo se estudiará más en profundidad la variación en los tiempos de retardo de las puertas lógicas, principalmente en los inversores y se realizarán simulaciones

en Hspice que nos permitan verificar la viabilidad de los elementos de retardos asíncronos propuestos en el proyecto.

5. Estudio de la implementación a nivel de transistor

Introducción

Los avances tecnológicos de los últimos años han propiciado circuitos con mayor densidad de integración logrando diseños con mejores prestaciones a menor coste. Sin embargo, esta disminución de tamaño en los transistores aumenta el impacto de los procesos de fabricación en los diseños, provocando una desviación en las características esperadas del sistema. Debido a estas desviaciones en las características de los diseños, existen casos donde es obligatorio o muy recomendable evaluar los efectos que puede provocar en el diseño, por si fuera necesario implementar optimizaciones o rediseñar para mejorar su robustez, pudiéndose encontrar casos en los que finalmente se tenga que descartar ciertos diseños.

Aunque suele afectar diferentes parámetros, la desviación en el retardo en las puertas lógicas es de los más significativos, debido a las importantes implicaciones que tiene en el circuito, ya que delimita la frecuencia de operación y la velocidad de conmutación. Además de lo comentado, si añadimos que en nuestro diseño el retardo de las puertas lógicas lo empleamos como elemento del retardo del filtro, la influencia de la desviación del retardo podría ser mucho más perjudicial.

A la repercusión que tiene emplear puertas lógicas como elementos de retardos hay que añadir que el filtro se diseña empleando la tecnología CMOS de 45nm, comprendida dentro de las tecnologías CMOS sub-micrométricas. Estas dos circunstancias exigen una consideración especial en el proyecto, lo que implica que en este capítulo se estudie los efectos de las posibles desviaciones en los elementos de retardo a nivel de transistor y, debido a la propagación de los posibles efectos, se ha integrado el bloque diezmador en el estudio.

Una vez finalizada las simulaciones a más alto nivel mediante Matlab en los capítulos anteriores y teniendo en cuenta la implicación que pueden tener las variaciones de proceso de tecnologías semiconductoras nanométricas en los elementos de retardo asíncronos empleados en el filtro propuesto, este capítulo se centra en estudiar la viabilidad del diseño cumpliendo las especificaciones impuestas al inicio del proyecto, pero en este caso, considerando los efectos de las desviaciones de los parámetros de diseño a nivel de transistor.

Antes de comenzar a estudiar las posibles desviaciones en los parámetros de diseño, se explicará brevemente las variaciones de proceso en las tecnologías de semiconductores nanométricas, indicando cuales son las más significativas para nuestro diseño. A continuación, se estudiará la desviación del retardo en las puertas lógicas asociado a la tecnología empleada. Y, finalmente, se estudia la implementación del diezmador.

Variaciones en circuitos digitales

La complejidad del proceso de fabricación de los circuitos integrados debido a su mejora tecnológica ha provocado desviaciones relevantes en valores nominales de sus parámetros. Al aumentar la densidad de integración en los semiconductores, las desviaciones se han hecho más significativas hasta el punto de afectar al comportamiento y al rendimiento de los circuitos, provocando que en algunos casos deba ser analizado el impacto antes de su fabricación [49].

Las variaciones en los circuitos digitales se pueden clasificar según su fuente, siendo tres los tipos existentes:

Ambientales: variaciones temporales con duración determinada que están presentes durante el funcionamiento del dispositivo. Las tres más significativas son variaciones por voltaje de polarización, por temperatura y velocidad de conmutación [50].

Proceso: variaciones que ocurren durante el proceso de fabricación y afectan al rendimiento de los circuitos produciendo fluctuaciones en valores nominales de los parámetros de fabricación. Son producto de numerosos factores como puede ser el espesor del óxido, el número de átomos dopantes, la densidad del dopado, la posición de los dopantes, etc. Existen de dos tipos, por variaciones físicas de su *layout*, denominadas sistemáticas, y las que no tienen relación con el *layout* y afectan directamente a los parámetros propios del integrado [51].

Envejecimiento: variaciones producidas por la degradación, el desgaste o el envejecimiento que ocasionan defectos en su comportamiento. No se pueden predecir previamente, puesto que su aparición depende de la tecnología y a las condiciones de operación que el dispositivo fuera sometido. Entre los fenómenos de envejecimiento más destacados se encuentran NBTI (*Negative Bias Temperature Instability*), HCI (*Hot Carriers Injection*), TDDB (*Time Dependent Dielectric Break-down*) y electromigración [52].

En la Figura 5.1 se puede observar un gráfico con la clasificación las variaciones según las fuentes que han sido comentadas.

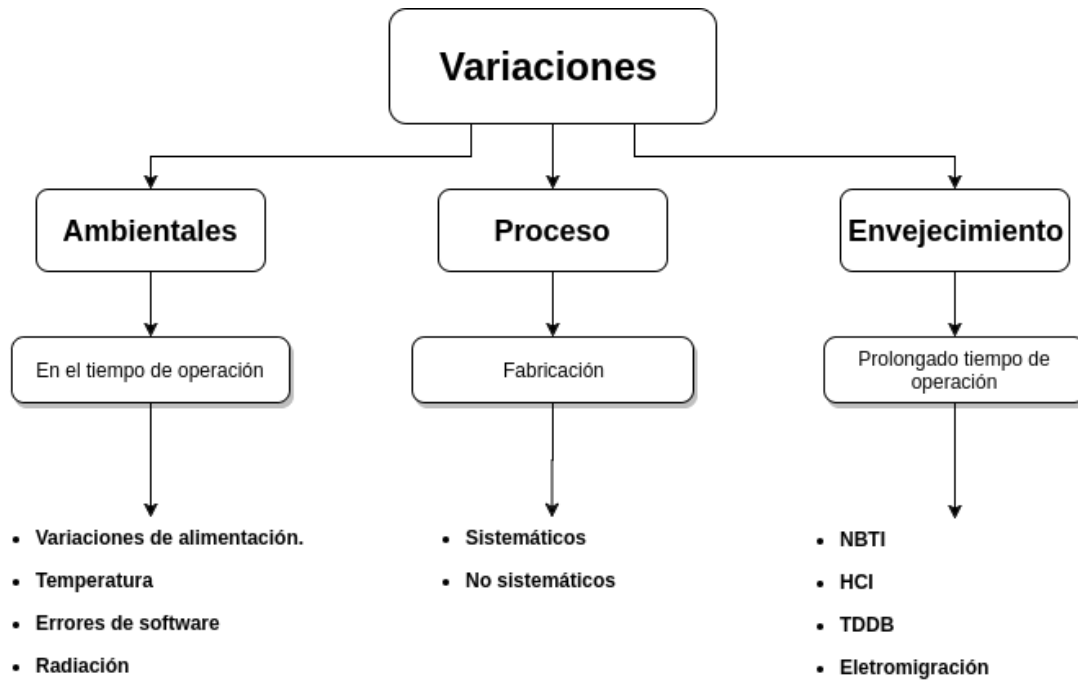


Figura 5.1. Clasificación de las variaciones según su fuente.

Impacto de las variaciones en el retardo de los sistemas digitales

Entre las variaciones comentadas, las de proceso están entre las que mayor impacto tienen en los retardos de los circuitos, debido a que el retardo mayor delimita la frecuencia de operación y, por consiguiente, su velocidad. Estas variaciones causan un aumento o disminución del retardo para los cuales fueron diseñados.

Por otro lado, las variaciones ambientales afectan principalmente al retardo, en especial la variación de voltaje de alimentación y la variación de temperatura. Normalmente se suelen realizar análisis con variaciones del 10% en la alimentación y para la temperatura rangos limitados dentro de 25 a 125 grados Celsius [53].

Finalmente, quedarían las variaciones por envejecimiento. En nuestro proyecto no serán tenidas en consideración ya que presentan una alta dificultad para identificarlas debido a que no se puede conocer el momento de su aparición y son dependientes de otras variaciones como las ambientales y de proceso.

Optimización del retardo por las variaciones

Actualmente existen técnicas para reducir el impacto de las variaciones por proceso en el retardo de las puertas lógicas. Estas técnicas se pueden dividir en dos categorías:

técnicas a nivel de *layout* y técnicas a nivel de circuito. Entre las técnicas de *layout* encontramos posicionamiento de transistores en paralelo, uso de múltiples *fingers* e interdigitado mientras que en el caso de las técnicas a nivel de circuito las principales y más empleadas son las del dimensionado del tamaño de la puerta y el reemplazo de estructuras.

Análisis estadístico de tiempos

Existen diferentes métodos para implementar el análisis estadístico de los tiempos en las puertas lógicas. Un método comúnmente empleando es el uso de modelos que son capaces aproximar las variaciones usando los parámetros tecnológicos y de funcionamiento.

Entre esos modelos se encuentra el modelo analítico, muy usado debido a su simplicidad en la implementación. Y, por otro lado, se encuentra el modelo lineal, más exacto que el anterior, pero con más complejidad ya que realiza una combinación del modelado analítico y el de Montecarlo.

En este proyecto se decidió emplear simulaciones con Hspice con análisis de Montecarlo, ya que ofrece muy buenos resultados, aunque para ello es necesario una gran capacidad de cómputo. El análisis de Montecarlo consiste en ejecutar varias simulaciones asignando en cada una un valor aleatorio a ciertos parámetros del circuito dentro de sus márgenes de tolerancia. De esta forma al analizar los valores de las señales de salida se tiene una medida de la dispersión en función de la tolerancia de los componentes. Debido a la alta frecuencia de operación de nuestro diseño, la tasa de cómputo resulta demasiado elevada para altos valores de iteraciones del análisis de Montecarlo, obligando a especificar una cantidad reducida de iteraciones que permitan un tiempo de simulación moderado.

Elementos de retardo asíncrono

La implementación a nivel de transistor de los elementos de retardo asíncronos propuestos se ha realizado mediante una cadena de inversores. Como se puede observar en la Figura 5.2, la cadena de inversores consiste en una red de inversores conectados en serie.

Cada inversor constituye un elemento de retardo debido a su retardo de propagación intrínseco. Tal y como se indicó en el capítulo 4, el retardo de propagación no es perfecto y su retardo real viene caracterizado por un retardo nominal más las desviaciones por las no-linealidades que introducen las variaciones comentadas en el apartado anterior. Por ello, los retardos entre cada uno de los elementos de la cadena de inversores no son iguales.

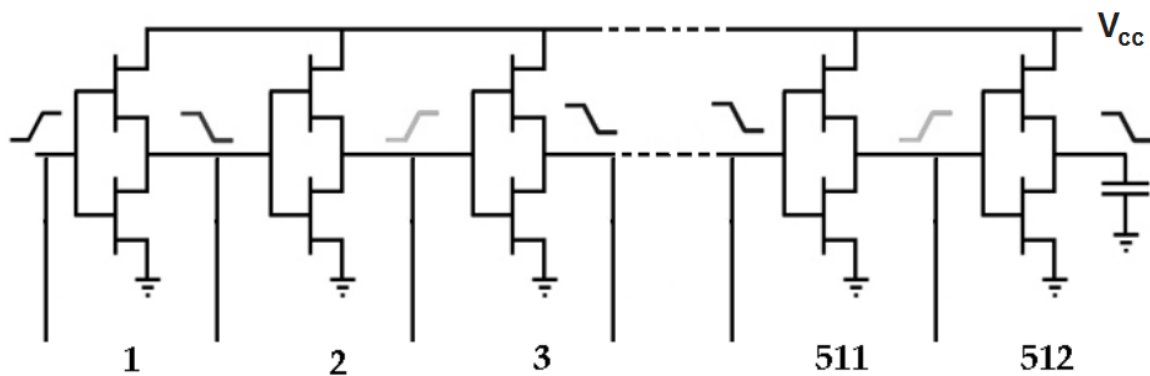


Figura 5.2. Cadena de inversores CMOS en serie.

En el capítulo anterior se caracterizó los efectos de estas variaciones en los elementos de retardo del filtro mediante su desviación estándar, en este apartado se va a obtener la desviación que la tecnología propuesta proporciona, así como los efectos que producen estas desviaciones en varios parámetros y las técnicas de optimización posibles.

Implementación

El diseño de los elementos de retardo se ha realizado en Hspice empleando la tecnología CMOS 45nm de PTM. Los elementos de retardo son implementados por inversores y están formados por un transistor NMOS y un PMOS siguiendo la estructura que se puede observar en la Figura 5.3.

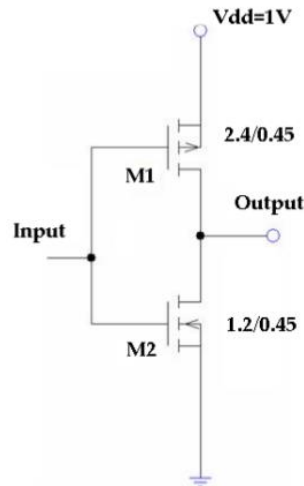


Figura 5.3. Estructura del inversor CMOS de 45nm.

La cadena de inversores que se necesita en el filtro está formada por 512 inversores, los cuales están conectados en serie con la salida de cada inversor conectada a la entrada del siguiente como se indica en la Figura 5.2.

Inicialmente se ha calculado el tiempo de retardo de una puerta inversora en Hspice con el fin de realizar una aproximación a los tiempos de retardo que se podrían alcanzar en el filtro. El ancho de puerta y la longitud de esta se han establecido en función del mínimo factible por la tecnología empleada para lograr el mínimo retardo de propagación posible. Por otro lado, debido a la variación de la movilidad entre los transistores ‘p’ y los ‘n’ se ha establecido una ratio entre ellos de 2:1. Los tamaños calculados de los transistores del inversor se muestran en la Tabla 5.1.

Tabla 5.1. Relación tamaños del inversor.

Transistor	W/L	W	L
M1	5,3	240 nm.	45 nm.
M2	2,6	120 nm.	45 nm.

Con el tamaño de transistor seleccionado se realiza una simulación del inversor y se estima el tiempo de propagación (t_p) utilizando una señal de entrada (V_{in}) cuadrada y una

alimentación de 1V, (tensión recomendada por la tecnología). Con las señales de entrada y salida (Vout) obtenidas en la simulación se calcula el tiempo de propagación de la señal de entrada en subida a la señal de salida en bajada y el tiempo de propagación, al contrario, señal de entrada en bajada y señal de salida en subida. Posteriormente, se calcula la media de estos valores, como indica la ecuación 5.1.

$$t_P = \frac{1}{2}(t_{PHL} + t_{PLH}) \quad (5.1)$$

donde t_{PHL} es el tiempo de propagación de nivel alto a bajo y t_{PLH} es el tiempo de propagación de nivel bajo a alto. En la Figura 5.4 se puede observar la salida respecto a la entrada y los tiempos de propagación al cambiar los diferentes niveles.

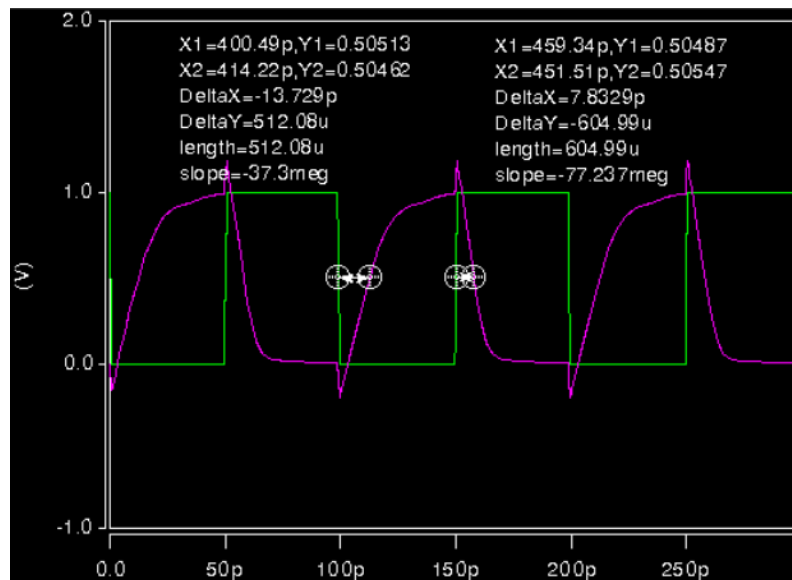


Figura 5.4. Tiempo de propagación del inversor simulado.

El tiempo medio de propagación proporcionado por la simulación se encuentra entorno 10.7 ps, un valor cercano a los 10 ps necesarios para lograr los 100 GHz indicados en las especificaciones y con los que se calcularon los coeficientes.

Antes de realizar ningún ajuste en los parámetros del diseño se deben tener en consideración algunos aspectos de los transistores MOS. Los transistores MOS tienen tres zonas de trabajo: zona lineal, zona de saturación y zona de corte. En la Tabla 5.2 se indican qué condiciones se deben dar para cada zona de trabajo y cuánto vale la corriente de drenador (I_d) asociada:

Tabla 5.2. Zonas de trabajo transistores MOS

Zona	Condición	I_D
Lineal	$V_{GS} > V_T$; $V_{DS} < V_{GS} - V_T$	$I_D = \beta_{\{n,p\}} \left((V_{GS} - V_T) \cdot V_{DS} - \frac{V_{DS}^2}{2} \right)$
Saturación	$V_{GS} > V_T$; $V_{DS} > V_{GS} - V_T$	$I_D = \beta_{\{n,p\}} \cdot \frac{(V_{GS} - V_T)^2}{2}$
Corte	$V_{GS} \leq V_T$	0

En la Tabla 5.2 V_{DS} es la tensión drenador surtidor, la V_{GS} es la tensión puerta surtidor, V_T es la tensión umbral y β_n se define en la ecuación 5.2.

$$\beta_{\{n,p\}} = \mu_{\{n,p\}} \cdot \frac{\varepsilon_{ox}}{t_{ox}} \cdot \frac{W_{\{n,p\}}}{L_{\{n,p\}}} \quad (5.2)$$

siendo μ_n la movilidad de los electrones, ε_{ox} la constante dieléctrica del aislante, t_{ox} el grosor del aislante, V_{Tn} la tensión umbral, W anchura de la puerta y L la longitud de la misma.

En la lógica digital los transistores se encuentran principalmente en la zona de saturación y en la zona de corte. De este modo, para nuestro inversor de la Figura 5.2 y alimentado con V_{cc} , dependiendo de la entrada V_{in} cambiará los estados de cada transistor. En las Figuras 5.5 y 5.6 se puede observar el estado de cada transistor con cada entrada, así como la corriente que circula a través de ellos.

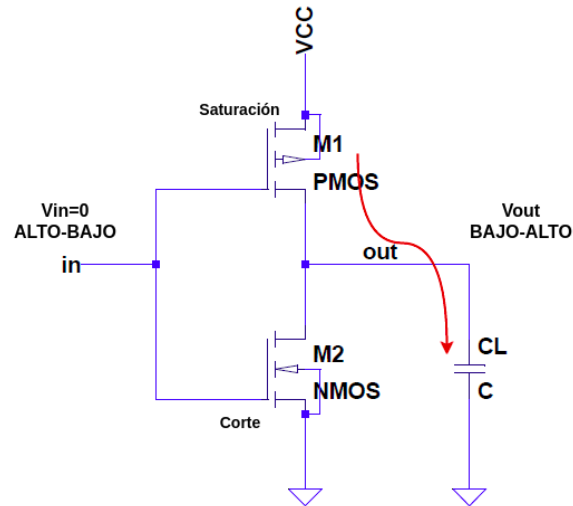


Figura 5.5. Flujo de corriente por el inversor con entrada a nivel bajo.

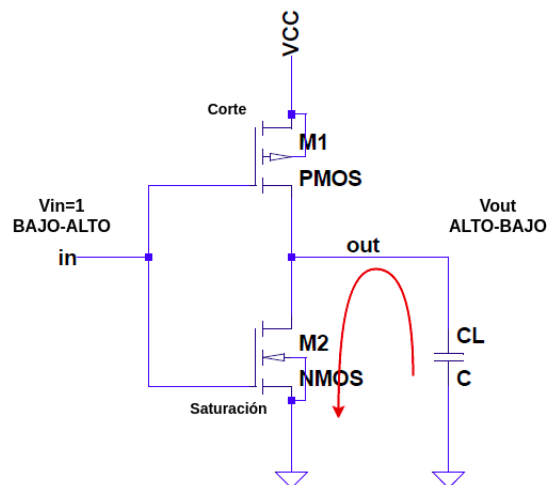


Figura 5.6. Flujo de corriente por el inversor con entrada a nivel alto.

Aunque en la Figura 5.5 se indica que existe una carga de salida señalada por el condensador C_L y situado entre V_{out} y GND, realmente esta carga son cargas parásitas que contienen los inversores que para facilitar el análisis son agrupadas en un solo condensador denominado C_L .

La carga de salida del inversor es otro aspecto muy significativo en el tiempo de propagación que debe ser considerado. Aunque se denomine carga de salida, ésta comprende dos tipos de cargas intrínsecas, que vienen definidas por las cargas asociadas a

la misma puerta tales como carga de difusión o cargas de '*overlap*', y cargas extrínsecas, asociadas a la entrada de la puerta siguiente conectada a la salida o la carga debida a las interconexiones. En la ecuación 5.3 se pueden observar las capacidades existentes.

$$C_L = C_{DBn} + C_{DBp} + C_{wire} + C_G \quad (5.3)$$

Donde C_{DBn} y C_{DBp} son las capacidades *Drain-Body* a la salida del propio inversor, C_{wire} es la capacidad de interconexión entre la salida y la entrada del siguiente inversor, y C_G es la capacidad de entrada del siguiente inversor y está formada por varias capacidades, entre ellas las capacidades de '*overlap*', las capacidades de la unión o la capacidad de puerta.

Algunas de las capacidades son dependientes de parámetros de diseño como son la longitud de canal y el ancho del canal y otras por procesos de diseño como sus interconexiones. El aumento en la densidad de integración en los circuitos actuales y por consiguiente su reducción de tamaños de los transistores y su longitud de puerta ha provocado una disminución de las capacidades, disminuyendo así su relevancia y aumentando la influencia de las menos dependientes de los parámetros de diseño. En la Figura 5.7 se pueden observar las capacidades más significativas en una cadena de inversores indicadas en la ecuación 5.3.

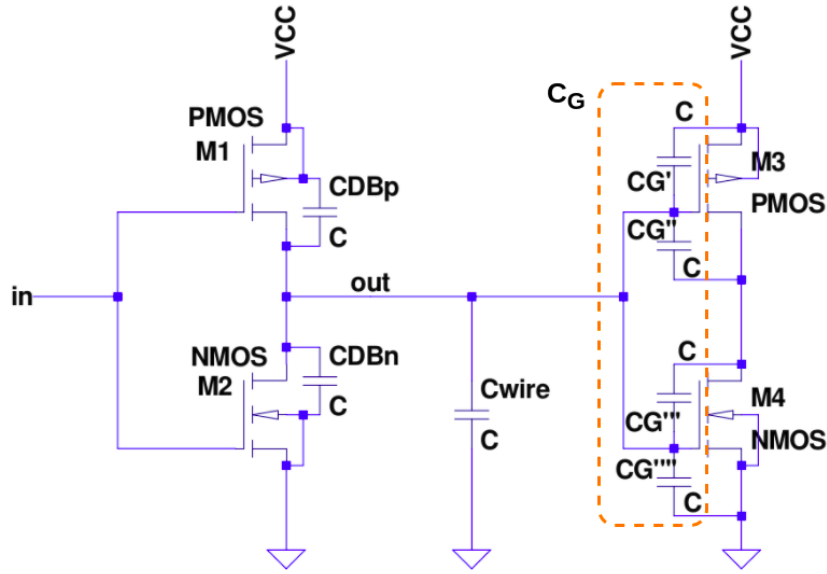


Figura 5.7. Capacidades más significativas en la cadena de inversores.

Con el fin de centrarnos en los parámetros que afectan al tiempo de propagación, a continuación, se indica mediante la ecuación 5.4 y 5.5 el tiempo empleado en descargar y en cargar la mitad la capacidad de salida del inversor CMOS, respectivamente.

$$T_{PLH} = \frac{\frac{1}{2} \cdot \text{Carga } C_L}{\text{Corriente carga}} = \frac{\frac{1}{2} \cdot C_L \cdot V_{CC}}{\beta_p (-V_{CC} - V_{Tp})^2}$$

$$T_{PHL} = \frac{\frac{1}{2} \cdot \text{Carga } C_L}{\text{Corriente descarga}} = \frac{\frac{1}{2} \cdot C_L \cdot V_{CC}}{\beta_n (V_{CC} - V_{Tn})^2}$$

Considerando $\mu_p = 2\mu_n$, $V_{Tp} \approx -V_{Tn}$, β definida por la ecuación 5.2 y suponiendo el caso simétrico $\beta_n = \beta_p$ ($L_n = L_p$ y $W_p = 2W_n$).

Finalmente, haciendo una sustitución en los tiempos de subida y de bajada en la ecuación 5.1, obtenemos el tiempo de propagación que se muestra en la ecuación 5.6.

$$t_p = \frac{2 \cdot C_L \cdot V_{CC}}{\frac{3W_n}{2L_n} \cdot \mu_n \cdot C_{ox} \cdot (V_{CC} - V_{Tn})^2}$$

Según la ecuación 5.6, se puede deducir que al aumentar la tensión conseguimos una reducción de tiempo de propagación, considerando que la VCC en el denominador está elevada al cuadrado, al contrario que la del numerador. Además, al reducir la longitud de puerta de la tecnología CMOS, también se logra reducir el tiempo de propagación, si atendemos a la capacidad de carga indicada en la ecuación 5.3, donde habitualmente predominan las cargas dependientes de los parámetros de diseño. En menor medida también se puede reducir el tiempo de propagación mediante la relación W/L que existe en el denominador de la ecuación 5.6, aunque el efecto que produce esta relación usualmente queda neutralizado por el efecto del tamaño de puertas en la capacidad de carga, siempre y cuando sea predominante las cargas relacionadas con la W y L del inversor.

La opción a priori más sencilla para reducir el tiempo de propagación es aumentar la alimentación de los transistores, pero siempre la tecnología tiene sus propias limitaciones y en el caso de la tecnología empleada de 45nm es un 10% del valor nominal recomendado que es 1V, permitiendo aumentar como máximo a 1.1V. Mediante el aumento de ese 0.1V se puede ajustar el tiempo de propagación a los 10ps deseados, valor muy cercano al mostrado en la Figura 5.4 con 1V de VCC.

Simulaciones

Se han realizado una serie de simulaciones con el fin de estudiar adecuadamente las variaciones de tiempos de retardo que nos proporciona la cadena de inversores que implementa los elementos de retardo según las variaciones de los circuitos digitales y sus desviaciones de los valores nominales indicados en el apartado 2 del presente capítulo. Además, estas simulaciones nos permitirán suponer que ancho de puerta será necesario para cumplir con las especificaciones del capítulo 1.

Para realizar las simulaciones se empleó el software Hspice con modelos de la tecnología de 45nm proporcionados por el departamento ESAT (*Department of Electrical Engineering*) de la universidad K.U. Leuven obtenido de los modelos predictivos PTM. Las simulaciones emplearon el método de Montecarlo en los transistores del diseño. En el anexo XX se indican los *netlist* usados en las simulaciones.

La primera simulación realizada en la cadena de inversores fue la relacionada con el tiempo medio de propagación de todos los inversores respecto a la variación del factor de multiplicación de las de las puertas del inversor. En la Figura 5.8 se puede ver la relación del tiempo medio de propagación con el ancho del canal de los transistores del inversor.

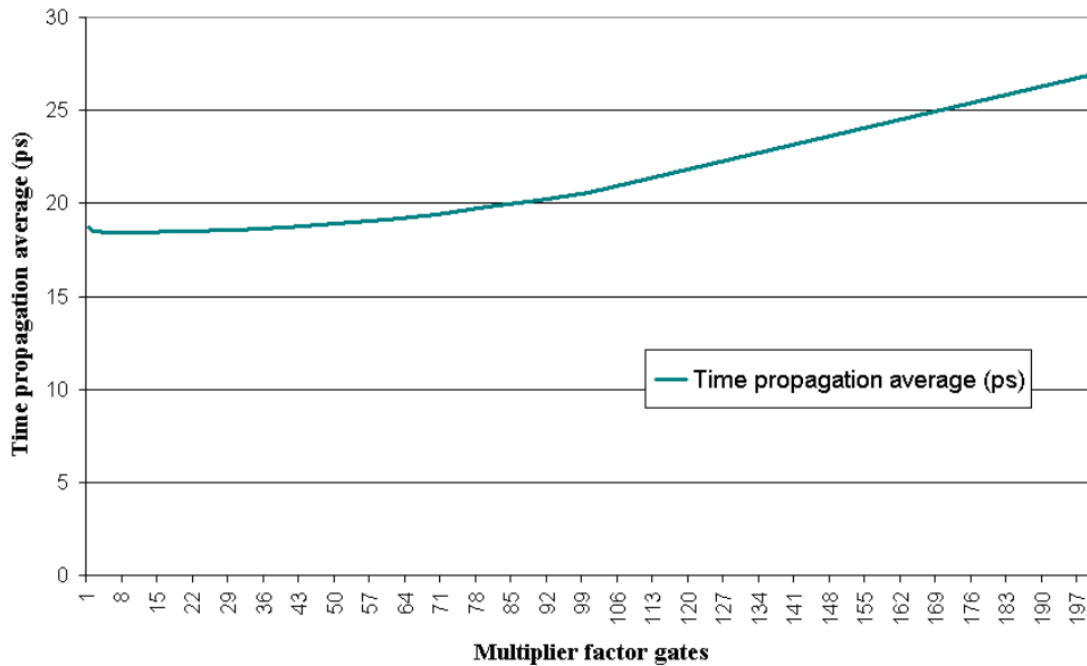


Figura 5.8. Relación del tiempo de propagación en los inversores con el tamaño de la puerta.

Como se puede observar en la Figura 5.8, a medida que el factor de multiplicación se incrementa, el ancho del canal aumenta y su tiempo medio de propagación también aumenta, como ya se podía intuir si consideramos la ecuación 5.6, donde el valor de la carga de salida es determinante en el valor del tiempo de propagación. Cabe destacar que inicialmente tiene un tiempo de programación mayor y disminuye ligeramente hasta que los efectos de las capacidades parásitas son predominantes. Este comportamiento no coincide con el modelo de tiempo de propagación indicado. Sin embargo, para transistores de canal corto se puede obtener una resistencia equivalente que permite calcular el modelo en pequeña señal mediante un recalculado de las transconductancias g_m y g_{mb} . Este caso no es el habitual en las tecnologías micrométricas, pero sí en las submicrométricas, donde los valores de la capacitancia dependiendo del ancho y longitud del canal son muy reducidas.

Otro aspecto destacado en la simulación es el aumento del tiempo de propagación cuando consideramos una cadena con la carga de las puertas que le siguen. Como se puede observar, hemos pasado de los 10 ps aproximados que se calcularon originalmente a valores del orden de 18-19 ps, muy superior a lo estimado, incumpliendo así la velocidad de muestreo que se indicaba inicialmente, debido a las capacidades parásitas que tienen de *fanout* la cadena de inversores.

Por otro lado, también se ha simulado la cadena de retardos para obtener la desviación estándar del tiempo de propagación de la red de inversores con respecto a un factor de multiplicación del ancho de la puerta, modificando así la relación de tamaños de los transistores de los inversores. En la Figura 5.9 se puede observar la gráfica obtenida.

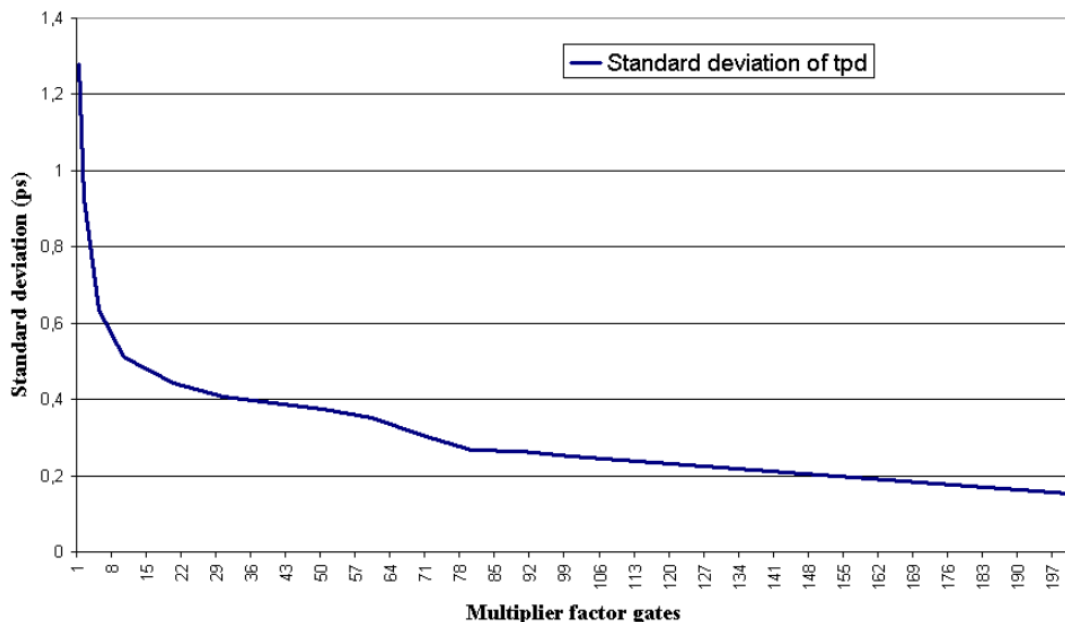


Figura 5.9. Relación de la desviación estándar en la cadena de inversores en su tiempo de propagación respecto al tamaño de la puerta.

Como ya se indicó en el apartado 2 del presente capítulo, uno de los métodos de optimización en las desviaciones de los tiempos de retardo es el dimensionado de los tamaños de puerta. De este modo al realizar los incrementos en el ancho de puerta según un factor de multiplicación, se produce una mejora en la desviación estándar del tiempo de propagación, debida a la atenuación de los efectos de no-linealidad indicados en el primer apartado del capítulo.

En la tabla 5.3 se muestra un resumen con los tiempos de propagación y la desviación estándar producida en el tiempo de propagación obtenidos al modificar el tamaño de puerta.

Tabla 5.3. Desviación estándar respecto al tamaño de puerta

Factor Mult.	Longitud de puerta(nm)	Ancho de puerta (um)	Desviación estándar tpd (%)	Desviación estándar tpd (ps)	Tiempo medio de propagación
1	45	0,12	6,84%	1,27	18,71
2	45	0,24	5,01%	0,92	18,53
5	45	0,6	3,45%	0,63	18,42
10	45	1,2	2,79%	0,51	18,42
20	45	2,4	2,40%	0,44	18,46
40	45	4,8	2,18%	0,40	18,56
50	45	6	2,09%	0,39	18,70
60	45	7,2	1,98%	0,37	18,89
70	45	8,4	1,83%	0,35	19,14
80	45	9,6	1,58%	0,30	19,42
90	45	10,8	1,35%	0,26	19,78
100	45	12	1,31%	0,26	20,15
200	45	24	0,57%	0,24	20,57

Diezmado y bloque multiplicador

Además de los elementos de retardo, el bloque formado por el diezmado y la parte de multiplicación del filtro FIR contiene consideraciones especiales, debido principalmente a que la entrada del filtro tiene una alta frecuencia de muestreo y su implementación asíncrona en los elementos de retardo está conectado a ellos. Por todo ello se va a estudiar su implementación a nivel de puerta lógica.

Como se indica en la Figura 3.7 del capítulo 3 y según las especificaciones iniciales, la señal a la entrada del diezmador es de un bit y está muestreada alrededor de los 100 GHz. A la salida del diezmador se debe tener una señal de 1 bit pero un muestreo inferior. En este caso según el reloj empleado será de 1 GHz, que está conectada al multiplicador. La implementación de estos dos módulos se puede simplificar e integrarlos juntos mediante *flip-flops*. En el siguiente apartado se analiza su implementación.

Implementación

La arquitectura propuesta para el bloque del diezmador y el multiplicador permite una implementación sencilla mediante el uso de *flip-flops* y multiplexadores, debido principalmente a dos razones. La primera razón es que la multiplicación de la entrada y el coeficiente de cada elemento solo puede tener dos valores (el coeficiente positivo o negativo), considerando que la entrada es de un bit. La segunda razón es porque al seleccionar en el capítulo 3 la representación digital de la señal en punto fijo, esto permite emplear complemento a dos, lo que permite usar un multiplexor con el valor positivo o negativo del coeficiente usando la salida del *flip-flop* como entrada de selección en el multiplexor. En la Figura 5.10 se puede observar el esquema de la implementación propuesta para el bloque de diezmado y multiplicador.

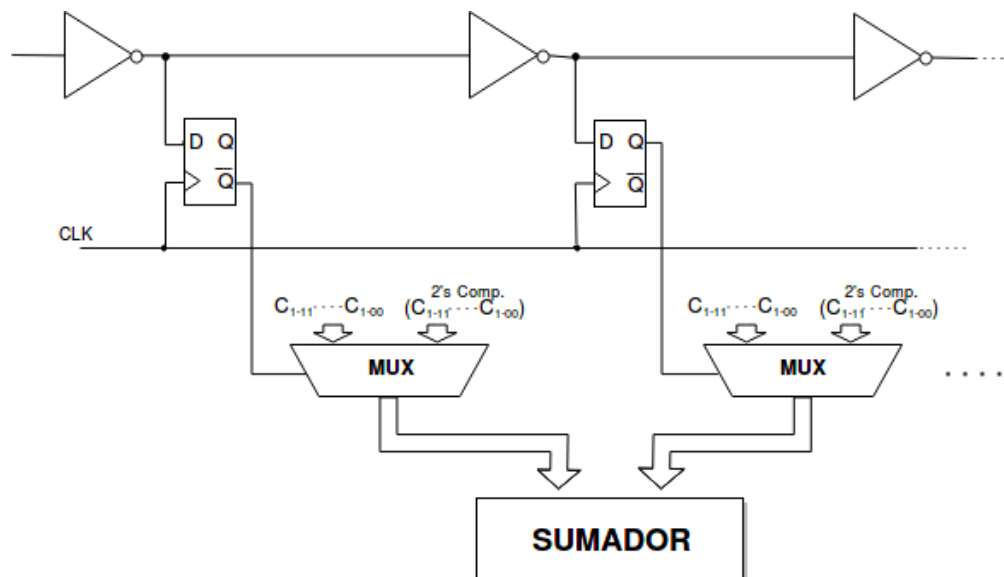


Figura 6Figura 5.10. Esquema del bloque diezmador.

En la Figura 5.10 se puede observar que la salida de los *flip-flops* que están conectada a la entrada de selección de los multiplexores son negadas en los elementos de retardo impares y no negada para los pares (debido al carácter inversor de los elementos de retardo). Por otro lado, la entrada de los coeficientes en los multiplexores serán un conjunto de bits fijos, según los coeficientes cuantificados obtenidos en el capítulo 4 y su respectivo complemento a dos.

Los *flip-flops* (FFs) son los elementos básicos de almacenamiento ampliamente utilizados en electrónica digital. Su capacidad de almacenamiento nos va a permitir mantener el último estado de la entrada a la salida hasta que señal de reloj vuelva a activarse, permitiendo así diezmar la señal según la frecuencia de la señal de reloj. Dependiendo si incluyen o no una entrada de sincronismo, puede ser síncronos o asíncronos. En el caso del presente proyecto serán síncronos debido a que se ha indicado al inicio que existe una señal de reloj de un 1GHz.

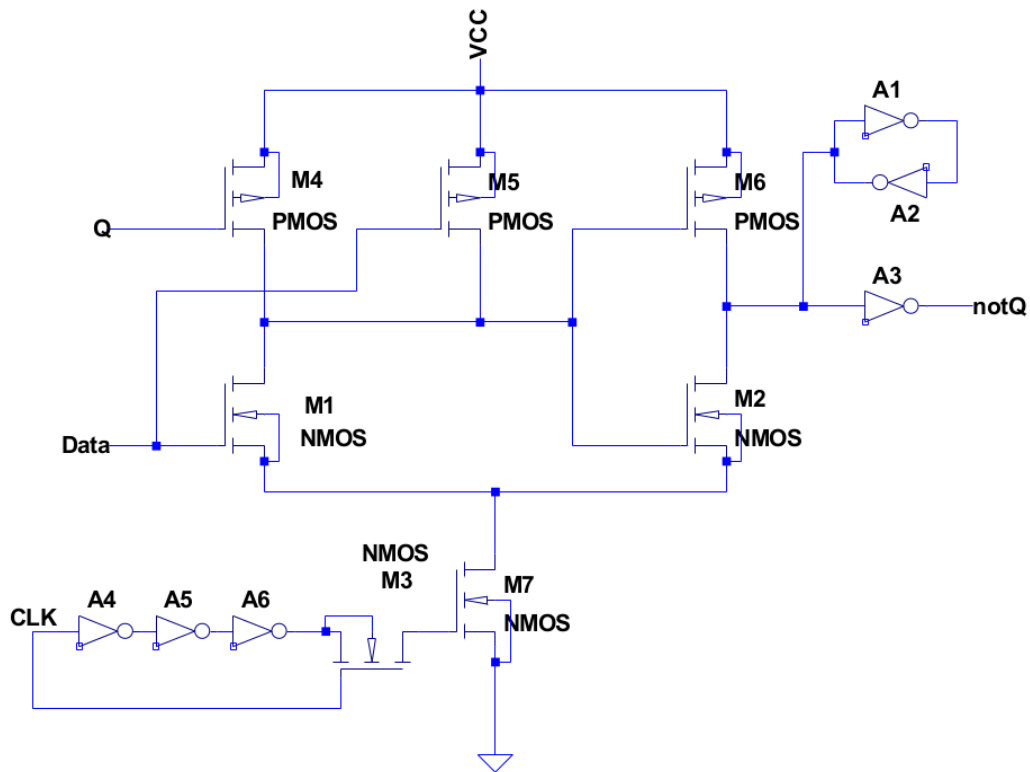
Para la implementación de los *flip-flops* en Hspice, en lugar emplear el esquema básico de *flip-flop* maestro/esclavo se ha empleado el esquema P-FF (*Pulse-triggered Flip-Flop*) por su mejor rendimiento en aplicaciones con alta frecuencia de operación [54] y su simplicidad de diseño.

El P-FF consiste en un generador de pulsos encargado de producir señales de '*strobe*' y un biestable para el almacenamiento de señales. Dado que los pulsos generados por el flanco del reloj son muy estrechos el biestable actúa como disparador por flanco. Otra ventaja es que posee una menor sensibilidad a las fluctuaciones del reloj mediante la anticipación temporal más allá del ciclo de reloj permitiendo un tiempo de *setup* 0 o negativo. La versión del P-FF empleada en este proyecto fin de carrera está basada en la versión P-FF con *Conditional Pulse-Enhancement Scheme* [55] con algunas modificaciones, denominado MHLFF (*Modified Hybrid Latch Flip flop*). Gracias al empleo de esta estructura, se consiguen unas propiedades de retardo muy favorables, además de un menor número de transistores que la versión sin modificar.

En la Figura 5.11 se puede observar el esquemático del P-FF implementado en Hspice y en la Tabla 5.4 se indican el tamaño de los transistores según otros diseños ya implementados, aunque con diferente tecnología [54].

Tabla 5.4. Relación tamaños del *Flip-Flop*.

Transistor	W/L	W	L
M1	4	180 nm.	45 nm.
M2	8	360 nm.	45 nm.
M3	30	1350 nm.	45 nm.
M4	2	90 nm.	45 nm.
M5	2	90 nm.	45 nm.
M6	13	585 nm.	45 nm.
M7	18	810 nm.	45 nm.
A1'	4	180 nm.	45 nm.
A1''	2	90 nm.	45 nm.
A2'	2	90 nm.	45 nm.
A2''	1	45 nm.	45 nm.
A3'	10	450 nm.	45 nm.
A3''	5	225 nm.	45 nm.
A4'	6	270 nm.	45 nm.
A4''	2	90 nm.	45 nm.
A5'	2	90 nm.	45 nm.
A5''	6	270 nm.	45 nm.
A6'	6	270 nm.	45 nm.
A6''	2	90 nm.	45 nm.

Figura 5.11. Esquemático del *Flip-Flop* implementado.

Simulaciones

Los requerimientos de los *flip-flops* al trabajar con una entrada tan elevada en frecuencia de muestreo obliga a caracterizar mediante simulaciones las propiedades temporales del *flip-flop*. Las propiedades más sensibles a la entrada para altas tasas de muestreo son los tiempos de *setup* y de *hold*, y en menor medida el retardo mínimo de *data* a Q.

Para obtener los tiempos de *setup* y *hold*, se realiza una simulación con análisis paramétrico. El tiempo de *setup* se consigue al realizar modificaciones de la entrada *data* antes del flanco de subida de la señal de reloj, en cada modificación se comprueba si existe una transición a la salida Q. Por otro lado, el tiempo de *hold* se logra verificando cuánto tiempo es necesario mantener el valor de *data* respecto a la señal de reloj para que exista una transición a la salida Q. Finalmente, el tiempo de retardo mínimo de *data* a Q se mide modificando *data*, cumpliendo siempre con el tiempo de *hold* y de *setup*, y calculando el tiempo hasta la transición de Q. En la Tabla 5.4 se muestra los tiempos calculados para el *flip-flop* implementado.

Tabla 5.5. Temporización del *Flip-Flop* implementado.

P-FF	(ps)
Tiempo de setup	9.5
Tiempo de hold	81.3
Tiempo de retardo min. de Data a Q	177.1

Con estos valores podemos comprobar varios datos muy importantes. En primer lugar, la comprobación del tiempo de retardo, para verificar si puede trabajar a la frecuencia de reloj que se desea hacer funcionar para el diezmado. En la siguiente ecuación 5.6 calculamos la frecuencia máxima que puede funcionar con el retardo calculado.

$$FrecuenciaMax.dereloj = \frac{1}{2 \cdot Retardo \text{ Data a } Q} = \frac{1}{2 \cdot 177.1ps} = 2.82GHz \quad (5.6)$$

Se puede apreciar en el resultado de la ecuación 5.6 que puede funcionar sin ninguna limitación para el diseño, teniendo en cuenta que la frecuencia resultante es mayor que 1GHz, que es el valor mínimo indicado en las especificaciones de diseño.

El dato obtenido más preocupante y que mayor limitación va a provocar es el tiempo de *hold*, debido a que es el tiempo que la señal debe permanecer en un valor (alto o bajo), sin sufrir cambio. En la Ecuación 5.7 se calcula la frecuencia máxima que puede soportar el *flip-flop* para poder capturar correctamente el nivel.

$$FrecuenciaMax.deentrada = \frac{1}{2 \cdot Tiempo \text{ de } Setup} = \frac{1}{2 \cdot 78.7ps} = 6.37GHz \quad (5.7)$$

Como observamos anteriormente, el tiempo de *hold* es considerablemente mayor que los elementos de retardo de los inversores, lo cual imposibilita de manera significativa alcanzar los valores estimados inicialmente.

Conclusiones

En este capítulo hemos visto la problemática de las desviaciones de los valores nominales en los circuitos integrados, explicando dos de los bloques con mayor complejidad por la elevada frecuencia de trabajo a la que debe funcionar en el filtro digital. Además, hemos realizado la implementación en Hspice con la tecnología indicada al inicio del proyecto (CMOS de PTM a 45nm).

Una vez implementados estos bloques, hemos realizado las simulaciones para obtener los parámetros que hemos considerados más importante para obtener las especificaciones de diseño indicadas al inicio del proyecto.

Las simulaciones han sido especialmente interesantes. Nos han servido para poder delimitar de manera precisa el tamaño de puerta necesario para conseguir una desviación estándar que cumpla con los valores obtenidos en el capítulo 4 evitando que las no linealidades resten rendimiento.

Por otro lado, más importante y reveladores son los valores temporales de retardo de las puertas y su desviación, se han obtenido valores lejanos en la frecuencia de funcionamiento a los estimados inicialmente si se desea mantener una desviación estándar baja.

Por último, la caracterización temporal del *flip-flop* ha demostrado la inviabilidad de funcionar a la frecuencia estimada inicialmente. Esto es debido principalmente al tiempo de *hold*, el cual está muy por encima del exigido para que el bloque diezmador/multiplicador funcione correctamente. En un principio se deseaba una frecuencia de entrada de unos 100 GHz, pero se ha visto que con el tiempo de *hold* del *flip-flop* diseñado no es posible alcanzar frecuencias tan altas. En el mejor de los casos, sería posible alcanzar valores cercanos a los 10 GHz, invalidando así todos los cálculos previos realizados para el cálculo del filtro.

En el siguiente capítulo daremos las conclusiones obtenidas después de realizar el estudio del diseño propuesto e indicaremos ciertas pautas que se pueden seguir para lograr hacerlo viable.

6. Conclusiones

Como hemos indicado en la introducción, el objetivo final de este proyecto es el diseño de un filtro digital asíncrono con alta frecuencia de muestreo que cumpla con unas especificaciones de diseño comentadas en el capítulo 1, siendo apropiado para ser utilizado en convertidores Sigma-Delta.

En el capítulo 2 hemos visto las propiedades y los requerimientos de filtrado de los convertidores Sigma-Delta. Además, en el capítulo 3 se ha explicado cómo trabajan los filtros digitales, sus diferentes tipos, así como sus principales características.

Una vez aclarados estos conceptos, hemos pasado al diseño del filtro digital a nivel de sistema mediante Matlab. Para ello, hemos implementado el filtro con las especificaciones iniciales establecidas para calcular los coeficientes y sus valores de cuantificación. El filtro ha sido modificado para forzar una variación en el valor nominal de los elementos de retardo y poder obtener diferentes desviaciones estándar del retardo. Esto nos va a permitir estudiar los efectos de esta variación y determinar los valores para que el rendimiento del filtro cumpla con las especificaciones de diseño.

Finalmente, los bloques del diseño más implicados en la estructura asíncrona del filtro lo hemos llevado a cabo a nivel de transistores con Hspice. Se han realizado simulaciones para estudiar de manera más concreta las desviaciones del valor nominal. Además, hemos estudiado la modificación de parámetros de diseño mediante simulaciones y poder ajustar

la desviación estándar de los elementos de retardo para que cumpla con el calculado obtenido en Matlab anteriormente.

En los apartados siguientes indicaremos los resultados y conclusiones más relevantes durante el proceso de diseño del filtro digital.

Orden y coeficientes del filtro

Para calcular el orden del filtro y sus coeficientes empleamos Matlab. Mediante este software y los scripts desarrollados realizamos simulaciones con diferentes órdenes. A continuación, mediante un script desarrollado obtuvimos las características de su respuesta en frecuencia.

Las especificaciones de diseño más relevantes con la modificación del orden y sus coeficientes fueron la banda de transición y la mínima atenuación en la banda de paso. Para cumplir con las especificaciones de diseño dadas obtuvimos un orden mínimo algo superior a 500. En la Tabla 6.1 se puede observar la banda de transición y la atenuación de la banda de paso obtenidas para los diferentes órdenes.

Tabla 6.1 Orden seleccionado con la banda de transición y su banda de rechazo asociado.

Orden	Banda de Transición	Atenuación en la Banda de Paso
2048	117 MHz	-44
1024	146 MHz	-46
512	249 MHz	-42
256	520 MHz	-43
128	1271 MHz	-33
64	2698 MHz	-31

Cuantificación del filtro

La cuantificación del filtro digital es otra de las partes críticas del diseño, en nuestro caso hemos obtenido los bits de cuantificación para los coeficientes, como para las operaciones aritméticas y para la salida empleando ecuaciones matemáticas y simulaciones en Matlab. Además, se verificó a través de simulaciones que el filtro mantiene las especificaciones de diseño indicadas al inicio del proyecto.

En la Tabla 6.2 se puede observar los bits seleccionados para cumplir con las especificaciones del filtro. En la Figuras 6.1 se muestra las respuestas en frecuencia del filtro para diferentes números de bits de cuantificación de sus coeficientes. Igualmente, en la Figura 6.2 se muestra la respuesta en frecuencia del filtro para diferentes bits de cuantificación, pero de su salida.

Tabla 6.2 Número de Bits de cuantificación seleccionado en el filtro digital diseñado.

Coeficientes	Operaciones aritméticas	Salida del filtro
12 bits	16 bits	16 bits

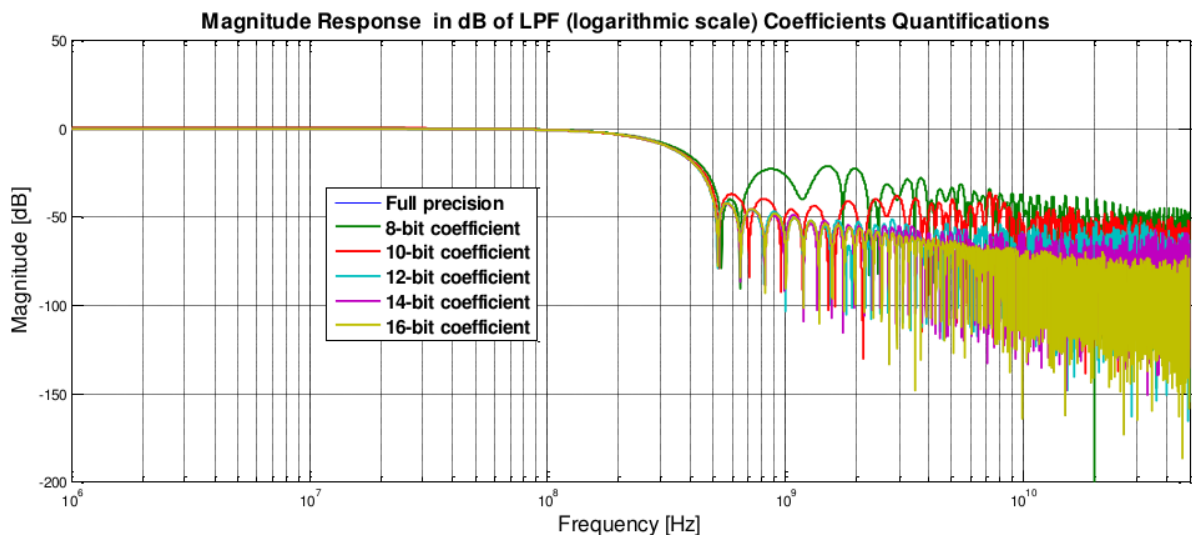


Figura 6.1. Respuesta en frecuencia del filtro para diferente número de bits de cuantificación de los coeficientes.

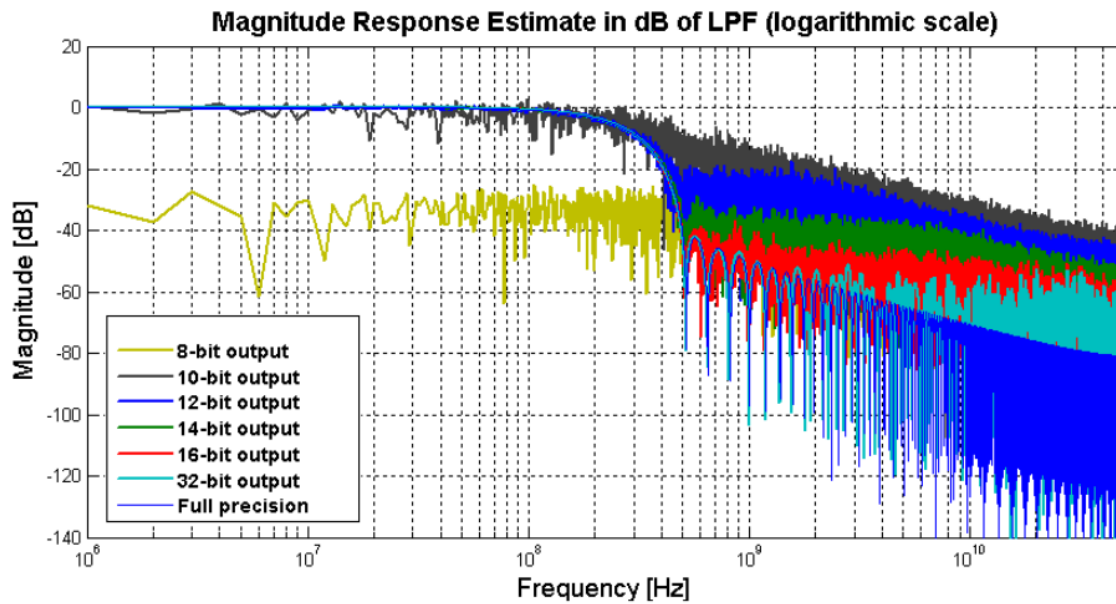


Figura 6.2. Respuesta en frecuencia del filtro para diferente número de bits de cuantificación de la salida.

No linealidades en los elementos de retardo

Al emplear inversores como elementos de retardo sin ningún sistema síncrono que fije la frecuencia, se van a producir variaciones en el tiempo nominal debido a múltiples factores, explicados en el capítulo 5. Entre ellos destacan las variaciones por proceso de fabricación y las variaciones ambientales. Estas variaciones producen unos efectos negativos que degradan las características del filtro y, por consiguiente, su rendimiento. Mediante Matlab realizamos simulaciones variando la desviación estándar de los elementos de retardo del filtro y caracterizando su salida en frecuencia para valorar la degradación del filtro.

Entre las características más perjudicadas por esta variación en los elementos de retardo está la banda de atenuación mínima. Para mantenerla dentro del mínimo exigido en las especificaciones de diseño de 40dB, la desviación estándar del tiempo de retardo permitida debe estar en torno al 5%. En la Figura 6.3 se puede observar las diferentes respuestas en frecuencia del filtro para varios valores de la desviación estándar del retardo.

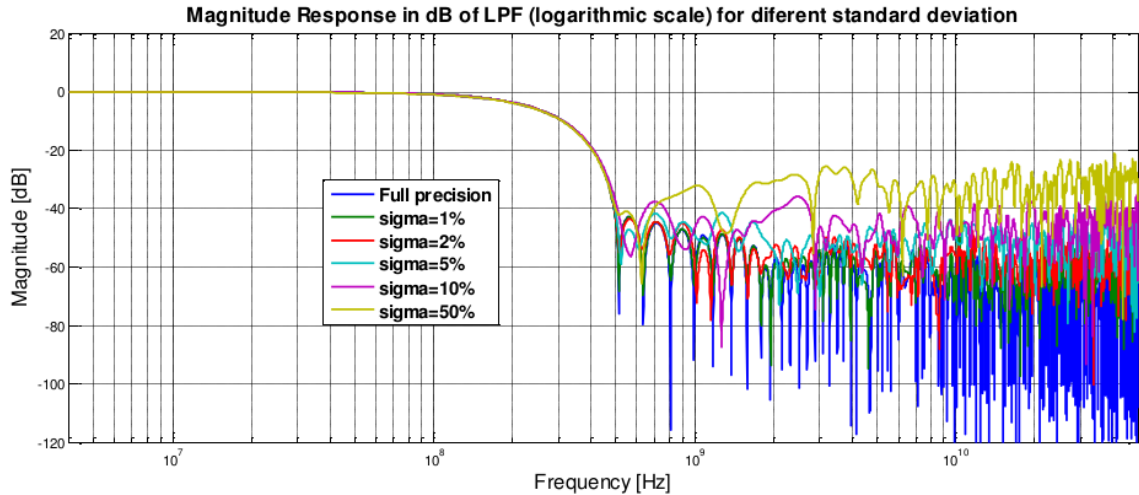


Figura 6.3. Respuesta en frecuencia del filtro con diferentes valores de desviación estándar.

Estudio de la implementación a nivel de transistor

Como se explica en el capítulo 5, para caracterizar las desviaciones en los elementos de retardo se ha implementado una cadena de inversores y posteriormente se realizaron simulaciones midiendo el tiempo de propagación de cada puerta. Por otro lado, estas simulaciones nos permitieron verificar la tecnología de 45nm PTM proporcionada para el desarrollo del proyecto.

En las primeras simulaciones verificamos que la frecuencia de muestreo del filtro se encuentra muy al límite de la indicada por las especificaciones de diseño debido al tiempo de propagación de los inversores. Aunque, el problema se vuelve más manifiesto al realizar la optimización en las desviaciones del tiempo de propagación nominal. Estas optimizaciones se realizan aumentando el tamaño de puerta y tiene una repercusión directa en el tiempo de propagación como se puede observar en la Figura 6.4.

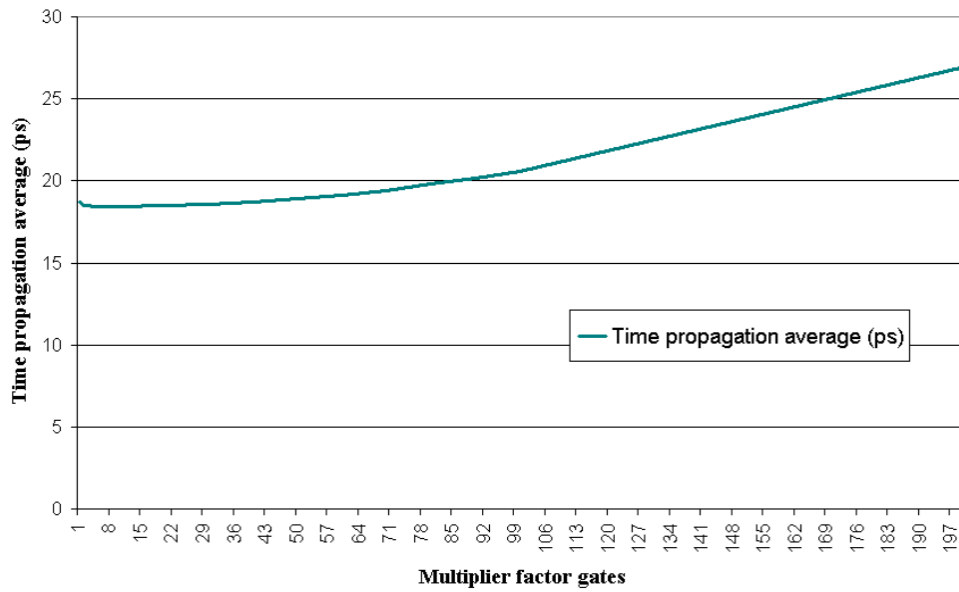


Figura 6.4. Tiempo de propagación del inversor respecto al tamaño de la puerta.

Si observamos la siguiente Figura 6.5 se puede apreciar la mejora producida en la desviación estándar al aumentar el tamaño de puerta.

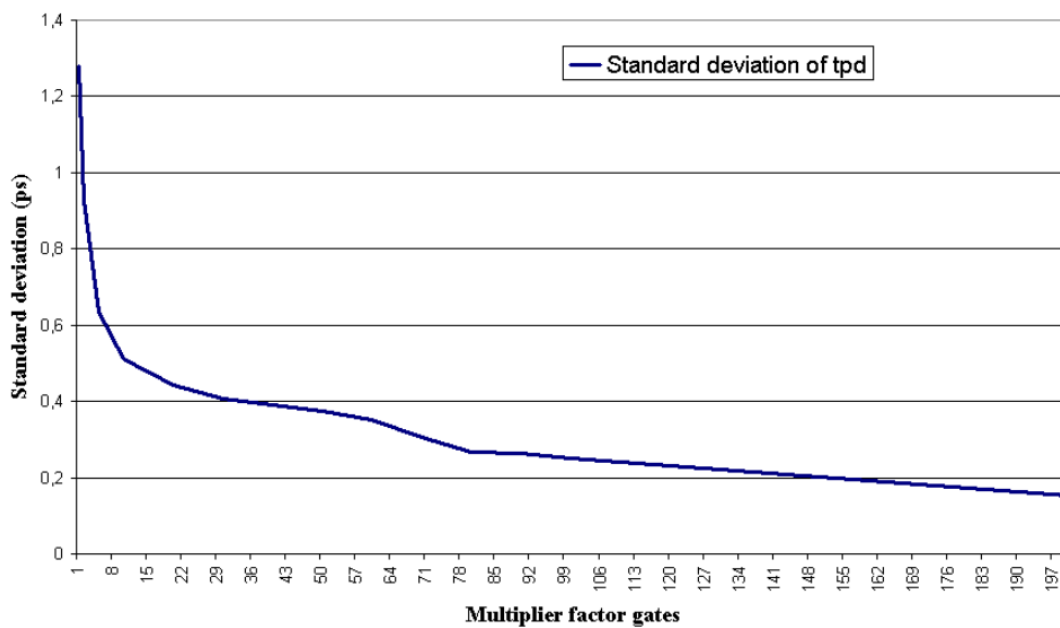


Figura 6.5. Desviación estándar del tiempo de propagación del inversor respecto al tamaño de la puerta.

Sabiendo del apartado anterior que la desviación estándar máxima de los elementos de retardo del filtro debe ser del 5%, En la Tabla 6.3 se pueden observar la desviación estándar y los tiempos de propagación calculados.

Tabla 6.3. Resultado de desviación estándar y tiempo de propagación de la cadena de inversores respecto a diferentes tamaños de puerta

Factor Mult.	Ancho de puerta (nm)	Longitud de puerta (um)	Desviación estándar tpd (%)	Desviación estándar tpd (ps)	Tiempo medio de propagación
1	45	0,12	6,84%	1,27	18,71
2	45	0,24	5,01%	0,92	18,53
5	45	0,6	3,45%	0,63	18,42
10	45	1,2	2,79%	0,51	18,42
50	45	6	2,09%	0,39	18,70
100	45	12	1,31%	0,26	20,15
200	45	24	0,57%	0,24	20,57

Según la Tabla 6.3 se logra satisfacer la desviación estándar, pero el tiempo de propagación ha aumentado, reduciendo la velocidad de muestreo que inicialmente lograba. Con el tiempo de propagación obtenido la frecuencia máxima que puede trabajar sea encuentra alrededor de los 50 GHz.

Por otro lado, también realizamos un estudio nivel de transistor del bloque diezmadador, en el cual se halló el mayor obstáculo no previsto para el diseño propuesto. Concretamente el problema lo encontramos al implementar los *flip-flops* que componen el bloque de diezmadado, debido a que su tiempo de *hold*, no permite una frecuencia tan elevada. El tiempo de *hold* calculado mediante simulación 177,2 ps. lo que limita la frecuencia de muestreo a un valor cercano a los 3GHz, muy por debajo de lo indicado en las especificaciones de diseño.

Conclusiones

En este proyecto hemos dado una visión de cómo trabaja un filtro digital con líneas de retardos asíncronas con alta frecuencia de muestreo y los principales problemas que se encuentran asociados. Además, hemos diseñado el filtro a nivel de sistema en Matlab, calculando los coeficientes, los bits de cuantificación necesarios y estudiando las

consecuencias en las prestaciones de este debidas a las variaciones de los valores nominales de los elementos de retardo.

Una vez realizado el diseño en Matlab, hemos pasado al diseño en Hspice de los bloques que se han considerado más singulares en el filtro y con mayores requerimientos en frecuencia de trabajo por su emplazamiento, como es el bloque de elementos de retardos asíncronos implementado mediante una cadena de inversores. Además de diseñarlos se han realizado diferentes simulaciones para comprobar las desviaciones obtenidas en los elementos de retardo, caracterizándolos mediante su desviación estándar.

Las simulaciones de los elementos de retardo y del bloque diezmador a nivel de transistor puso de manifiesto la dificultad para alcanzar las especificaciones de diseño inicial con la tecnología de 45nm propuesta.

El tiempo de *hold* de los *flip-flops* del bloque diezmador y, en menor medida, el aumento del tiempo de propagación debido al ajuste de tamaño de puerta para cumplir con la desviación estándar calculada. Acabo siendo fatal para el diseño y demostró la inviabilidad del filtro con las especificaciones deseadas.

A pesar de que no se han logrado cumplir las especificaciones iniciales de diseño, el presente proyecto ha puesto de manifiesto los principales problemas a la hora de diseñar un filtro digital con elementos de retardo asíncrono, presentando las partes donde focalizar los esfuerzos para lograr altas prestaciones. Estos esfuerzos se deben concentrar en una adecuada caracterización temporal de las puertas lógicas y de los bloques funcionales a nivel de transistor.

Por otro lado, también se puede sacar importantes conclusiones de las simulaciones realizadas, principalmente sobre el efecto de la modificación del ancho de puerta en la desviación estándar o de su tiempo de retardo.

Con todo esto, el presente trabajo tiene continuidad en aspectos como en el ajuste de especificaciones más acorde a la tecnología empleada, mejoras del bloque diezmador para reducir su tiempo de *hold* o emplear otras estructuras a nivel de puertas lógicas. Este

último aspecto, aunque ha sido valorado durante el diseño, la limitación temporal ha impedido profundizar en su estudio.

En cualquier caso, este proyecto se presenta como una primera toma de contacto en la implementación de filtros digitales empleando cadenas de inversores como elementos de retardo, proporcionando una visión más realista de futuras líneas de investigación de más envergadura.

Presupuesto

Una vez finalizado el presente proyecto fin de carrera con las conclusiones expuestas, para concluir con el proyecto, en este capítulo se realizará un estudio económico con los costes tanto parciales como totales de este.

Costes debidos a los Recursos Humanos

Este coste es el producido por el personal empleado para el mantenimiento de las herramientas y las estructuras necesarias. Este personal está formado por dos técnicos a tiempo completo para un total de cien usuarios.

<i>Descripción</i>	<i>Gastos</i>
<i>2 técnicos a tiempo completo</i>	36.060,73 €/año
<i>Proporción suponiendo 100 usuarios</i>	360,61 €/año-usuario
<i>TOTAL (9 meses)</i>	270,46 €

Costes de Ingeniería

Se ha invertido un total de 9 meses a tiempo completo en el desarrollo de este proyecto. Durante este tiempo se han realizados diferentes tareas como son la de formación, estudio y análisis, especificación, desarrollo, análisis de resultados y documentación; como muchas de estas tareas se solapan en el tiempo se han incluido los períodos de tiempo en función de la tarea predominante.

En este presupuesto se ha incluido los costes de formación al ser específicos para el desarrollo de este proyecto. Tomando la tarifa por costes de ingeniería aplicada a un ingeniero junior en España, asciende a un total de 2.232,16€/mes englobando en la misma el salario, la Seguridad Social, y el IRPF (Impuesto sobre la Renta de las personas físicas). En la siguiente tabla se muestra los costes de ingeniería desglosados.

<i>Descripción</i>	Tiempo (Meses)	Gastos (€/mes)	Gastos (€)
<i>Formación y estudios previos</i>	2	2.232,16	4.464,32
<i>Estudio de las herramientas de trabajo</i>	1	2.232,16	2.232,16
<i>Desarrollo del diseño</i>	5	2.232,16	1.1160,80
<i>Realización de la memoria</i>	1	2.232,16	2.232,16
TOTAL			20.089,44

Costes de amortización

En este apartado se realiza el análisis de los costes relacionados con el uso de paquetes software, material hardware y el mantenimiento de estos. Estos equipos hardware y paquetes software presentan un coste de amortización, en función del periodo de tiempo usado y el número de usuarios que accedan a estos, los cuales se ha estimado en 10.

<i>Descripción</i>	Tiempo de uso (€/mes)	Total (€/años) - Anual	Usuario (€/año-usuario) - Anual	Total
<i>Estaciones de trabajo SUN</i>	9	6.000	600	450,00
<i>Impresora Hewlett Packard Laserjet 4L</i>	9	2.000	200	150,00
<i>Ordenador portatil i5</i>	9	1.200	120	90,00
<i>Sistema operativo SunOs</i>	9	900	90	67,50

<i>Release 4.1.3, Open Windows y aplicaciones X11</i>				
<i>Microsoft Office 2010</i>	9	500	50	37,50
<i>Hspice</i>	9	5.000	500	375,00
<i>Matlab</i>	9	900	90	67,50
TOTAL				1.237,5

Otros costes

Para finalizar con el desglose de costes parciales a continuación se muestran los costes necesarios al material fungible y la elaboración del documento final del proyecto.

<i>Descripción</i>	<i>Unidades</i>	<i>Coste Unidad (€)</i>	<i>Gastos (€)</i>
<i>Paquetes de DIN_A4 80 gr/m</i>	3	4	12,00
<i>Fotocopias Blanco y Negro</i>	600	0,04	24,00
<i>Fotocopias Color</i>	20	0,65	13,00
<i>CDs</i>	3	2	6,00
<i>Otros gastos</i>			40,00
TOTAL			95,00

Coste Total

Para finalizar en la siguiente tabla se recoge el coste total del proyecto en función de los costes parciales comentados en las secciones anteriores.

<i>Descripción</i>	<i>Gastos (€)</i>
<i>Costes de recursos humanos</i>	270,46
<i>Costes de ingeniería</i>	20.089,44
<i>Costes amortización</i>	1.237,50
<i>Otros costes</i>	95,00
SUBTOTAL	21.692,40
TOTAL (7% IGIC)	23.210,87

D. Carlos Manuel Fernández Reyes declara que el proyecto “Diseño de un filtro asíncrono digital de alta frecuencia para receptores digitales” asciende a un total de veinte tres mil doscientos diez euros con ochenta y siete céntimos.

Fdo.: Carlos Manuel Fernández Reyes

DNI: 42201374P.

Las Palmas de Gran Canaria, a 13 de Junio de 2018

BIBLIOGRAFÍA

- [1] P. Sutar Ga, D. D. Tang, J. ALTieri, L.E. Thon, G. Coleman, S. Setbbanna, J. Y. C. Sun, "55-mW 300-MHz analog-digital converters using digital VLSI technology," IEEE Low Power Electronic Symposiums, pp.68-69, October 1995.
- [2] [2] Z. Shi, "Sigma -delta ADC and DAC for digital wireless communication," IEEE Radio Frequency Integrated Circuits (RFIC) Symposium , pp.57-62, June 1999.
- [3] D. Kinniment, A. Yakovlev, B. Gao. "Synchronous and Asynchronous A-D Conversion" IEEE Transactions on VLSI Systems, Vol. 8, n° 2, pp. 217-220, April 2000.
- [4] N.C. Paver, P. Day, C. Farnsworth, D.L. Jackson, W.A. Lien, and J. Liu. A lowpower, low-noise configurable self-timed DSP. In Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, pages 32–42, 1998.
- [5] S.B. Furber, J.D. Garside, S. Temple, J. Liu, P. Day, and N.C. Paver. AMULET2e: An asynchronous embedded controller. In Proc. International Symposium on advanced Research in Asynchronous Circuits and Systems, pages 290–299. IEEE Computer Society Press, 1997.
- [6] L.S. Nielsen. Low-power Asynchronous VLSI Design. PhD thesis, Department of Information Technology, Technical University of Denmark, 1997. IT-TR:1997-12.
- [7] SPeedster, a very high speed FPGA by Achronix: <http://www.achronix.com/>
- [8] A.J. Martin, A. Lines, R. Manohar, M. Nyström, P. Penzes, R. Southworth, U.V. Cummings, and T.-K. Lee. The design of an asynchronous MIPS R3000. In Proceedings of the 17th Conference on Advanced Research in VLSI, pages 164–181. MIT Press, September 1997.
- [9] L.S. Nielsen, C. Niessen, J. Sparsø, and C.H. van Berkel. Low-power operation using self-timed circuits and adaptive scaling of the supply voltage. IEEE Transactions on VLSI Systems, 2(4):391–397, 1994.
- [10] Predictive technology model. <http://ptm.asu.edu/>
- [11] Richard C. Dorf, "The Electrical Engineering Handbook, Second Edition", CRC Press, 1997.
- [12] Kester, W.A. and Analog Devices, inc, "Data Conversion Handbook", Newnes, 2005.
- [13] Tadeusz Sidor, "Effective Elimination of Analog Impairments Error in Parallel Interleaving Sigma Delta A/D Converters", Electrical and Electronic Engineering, 6(2): 25-29, 2016.
- [14] Anup G. Dakre, Prof. A.M. Chopde, "Design and Simulation of 1-Bit Sigma-Delta ADC Using Ngspice Tool" ISSN: 2277 – 9043 International Journal of Advanced Research in Computer Science and Electronics Engineering Volume 1, Issue 2, April 2012.
- [15] David Jarman, "A Brief Introduction to Sigma Delta Conversion" Intersil, Application Note AN9504, 1995.
- [16] Lucien Breems, Johan Huijsing, "Continuous-Time Sigma-Delta Modulation for A/D Conversion in Radio Receivers", Kluwer Academic Publishers, 2001.

- [17] Renaldi Winoto Borivoje Nikolic, "Downconverting Sigma-Delta A/D Converter for a Reconfigurable RF Receiver", Electrical Engineering and Computer Sciences University of California at Berkeley, Technical Report No. UCB/EECS-2009-81, 2009.
- [18] Bonnie Baker "How delta-sigma ADCs work, Part 1" Signal Integrity Engineer, Texas Instruments Incorporated, 2011.
- [19] Janssen, Erwin, van Roermund, Arthur "Look-Ahead Based Sigma-Delta Modulation", Illus, Springer, 2011.
- [20] Yao, Libin, Steyaert, Michiel, Sansen, Willy M. C, "Low-Power Low-Voltage Sigma-Delta Modulators in Nanometer CMOS", Springer Netherlands, 2006.
- [21] Kaldosh at en.wikipedia - Transferred from en.wikipedia to Commons by User:Wdwd using CommonsHelper on Labs., Public Domain, <https://commons.wikimedia.org/w/index.php?curid=18238755>
- [22] James C. Candy, Gabor C. Temes. "Oversampling Methods for A/D D/A Conversion, Oversampling Delta-Sigma Converters," New Jersey, IEEE Press, 1992.
- [23] A. Rusu, Babita R. J., M. Ismail, and H. Tenhunen, "A dualband sigma-delta modulator for gsmwcdma receivers". In Proceedings of The International XIX Conference on Design of Circuits and Integrated, Systems(DCIS), pages 673–676, 2004.
- [24] D. Johns, K. Martin, "Analog Integrated Circuit Design", John Wiley & Sons, 1997.
- [25] Xignal Technologies AG, "Putting continuous time delta-sigma modulators to work," ADC_whitepaper, 2005.
- [26] Dietrich Schlichthärle, "Digital Filters: Basics and Design", 2nd Edition, Springer, 2011.
- [27] A. Sheno, "Introduction to digital signal processing and filter design", Wiley-Interscience, 1 edition, 2005.
- [28] Steven W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing", California Technical Pub, 1997.
- [29] Bonnie Baker, "How delta-sigma ADCs work, Part 1 (Rev. A)", Data Acquisition, Texas Instruments Incorporated, 2016.
- [30] Wai-Kai Chen, "The Circuits and Filters Handbook, Second Edition", Taylor & Francis, 2002.
- [31] C Chou, S. Mohanakrishnan and J. B. Evans "FPGA implementation of digital filters", In proceeding of the Fourth International Conference on Signal processing Application and Technology, pp. 80-88, September 1993.
- [32] Saeed V. Vaseghi, "Multimedia Signal Processing: Theory and Applications in Speech, Music and Communications", Wiley, 1 edition, 2007.
- [33] N. V. Murali , "Design of FIR filters with arbitrary responses", M.Tech. credit seminar report, Electronic Systems Group, EE Dept, IIT Bombay, submitted November 2004.
- [34] Julius O. Smith, "Spectral Audio Signal Processing", Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, 2011.
- [35] Proakis J. G. and Manolakis D. G. "Digital Signal Processing: Principles, Algorithms and Applications". Macmillan, New York, NY, third edition., 1996.

-
- [36] L.R. Rabiner and B. Gold, "Theory and Applications of Digital Signal Processing", New Jersey: Prentice-Hall, 1975.
 - [37] Hank Zumbahlen, "Linear Circuit Design Handbook", Analog Devices, Newnes, 2008.
 - [38] S. Hauck, "Asynchronous Design Methodologies: An Overview", Proceedings of the IEEE, Vol. 83, No. 1, pp. 69-93, January, 1995
 - [39] S. C. Smith, W. K. Al-Assadi, and J. Di, "Integrating Asynchronous Digital Design into the Computer Engineering Curriculum", IEEE Transactions on Education, Vol. 53/3, pp. 349-357, August 2010.
 - [40] J. Sparsø and S. Furber, "Principles of asynchronous circuit design - A systems perspective", Kluwer Academic Publishers, 2001.
 - [41] Saptarshi Mukherjee, Rajib Kar, Durbadal Mandal, "Linear Phase Low Pass FIR Filter Design Using Improved Particle Swarm Optimization", IEEE Student Conference On Research and Development, pp-358-363, 2011.
 - [42] J.A. Cherry y W.M. Snelgrove. "Continuous-time delta-sigma modulators for high-speed A/D conversion: theory, practice, and fundamental performance limits", Kluwer Academic Publishers, Massachusetts, 2000.
 - [43] Ronald W. Schafer (Author), Mark A. Yoder (Author) "Signal Processing First", 1st Edition by James H. McClellan, 2003.
 - [44] B. S. Chen and C. T. Kuo, "Stability analysis of digital filters under finite wordlength effects", Circuits, Devices and Systems, IEE Proceedings, vol.136, no.4, pp. 167- 172, Aug 1989.
 - [45] R. Yates, "Technical Reference Practical Considerations in Fixed-Point FIR Filter Implementations", <http://www.digitalsignallabs.com/fir.pdf>, 2010.
 - [46] T. Salim, J. Devlin, J. Whittington, "Quantization Effects in Digital Upconversion and Digital Beam Forming for TIGER Radar System", Department of Electronic Engineering, La Trobe University Bundoora, VIC 3086 Australia.
 - [47] F. Harris, "Multirate Signal Processing in Transmitter & Receiver Designs," UCLA Extension course, November, 2000.
 - [48] Ricardo A. Losada. "Digital filters with Matlab. Technical report", The MathWorks, Inc., May 2008.
 - [49] A. Srivastava, D. Sylvester & D. Blaauw Statistical Analysis and Optimization for VLSI: Timing and Power, Springer (2005)
 - [50] S. S. Sapatnekar, Overcoming Variations in Nanometer-Scale Technologies, IEE Journal on Emerging and Selected Topic in Circuits and Systems, vol.1, no.1 (2011)
 - [51] D. Blaauw, S. Member, K. Chopra, S. Member, Statistical Timing Analysis: From Basic Principles to State of the Art, vol.27, no.4, (2008).
 - [52] M. A. Alam, A broad overview of reliability of semiconductor MOSFET, Purdue University, W. Lafayette, IN.
 - [53] M. Orshansky, Sani R. Nassafi, Duane Boning Design for Manufacturability and Statistical Design, Springer (2008)
 - [54] S. H. Rasouli, A. Khademzadeh, A. Afzali-Kusha, and M. Nourani, "Low power single-

- and double-edge-triggered flip-flops for high speed applications,” IEE Proc. Circuits Devices Syst., vol. 152, no. 2, pp. 118–122, Apr. 2005.
- [55] Y.-T. Hwang J.-F. Lin M. hwa Sheu "Low-power pulse-triggered flip-flop design with conditional pulse-enhancement scheme" IEEE Trans. Very Large Scale Integr. (VLSI) Syst. vol. 20 no. 2 pp. 361-366 Feb. 2012.

ANEXO: MATLAB

En el desarrollo del proyecto, tanto para calcular las gráficas empleadas como los valores de caracterización y figuras de mérito de los filtros, ha sido ampliamente utilizado Matlab. En muchos de los casos debido a las particularidades de la salida del filtro han tenido que implementarse exclusivamente para el proyecto. Las rutinas empleadas más importantes se muestran a continuación.

```
function [F,p_f] = cutofffilter6db(in_H,f,n)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   USAGE: [F,p_f] = cutofffilter6db(in_H,f,n)
%
%   AUTHOR: Carlos M. Fernández Reyes
%
%   DESCRIPTION: Obtiene los valores de las figura de merito para un filtro
%               dado respecto a uno ideal con arimetica full-precision.
%
%   INPUTS: in_H: Respuesta en frecuencia del filtro de entrada.
%           f: Vector de frecuencia del filtro.
%           n: Numero de muestras.
%
%   OUTPUTS: fom_data: Informacion con las figuras de merito.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

i=0;
atte_db=-6;
if (mean(20*log10(abs(in_H(1:1:10))))<(-3))
    atte_db=mean(20*log10(abs(in_H(1:1:10)))) - 3;
end;
for (i=1:1:n)
    if((20*log10(abs(in_H(i))))<atte_db)
        break;
    end;
end;
if (i<2)
    i=2;
end;
F=f(i-1);
p_f=i-1;
```

```
function Q = computemagnitudISE(in_H,ideal_H,m,F,f_ini,f_end,p_ini,p_end)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   USAGE: Q = computemagnitudISE(in_H,ideal_H,m,F,f_ini,f_end,p_ini,p_end)
%
%   AUTHOR: Carlos M. Fernández Reyes
%
% DESCRIPTION: Calcula el valor de el ISE (Integrated Squared Error)
%              en magnitud de la respuesta en frecuencia del filtro.
%
%   INPUTS: in_H: Respuesta en frecuencia del filtro de entrada.
%           ideal_H: Respuesta en frecuencia del filtro ideal
%           F: Frecuencias del filtro.
%           f_ini: Frecuencia inicial.
%           f_end: Frecuencia final.
%           p_ini: Punto inicial del calculo.
%           p_end: Punto final del calculo.
%
%   OUTPUTS: Q: Error ISE con respecto a la respuesta ideal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ISE Integrated Squared Error.
% Get interval
if((p_ini==0)&(p_end==0))
    for i=1:1:length(F)
        if (F(i)>=f_ini)
            p_ini=i;
            break
        end;
    end;
    for i=1:1:length(F)
        if (F(i)>=f_end)
            p_end=i;
            break
        end;
    end;
end;
in_H=in_H(p_ini:1:p_end);
ideal_H=ideal_H(p_ini:1:p_end);
% starting frequency point
a = p_ini/(length(F)*2);
% ending frequency point
b = p_end/(length(F)*2);
% absolute squared error
Q = ((sum((abs(abs(ideal_H)-abs(in_H))).^2))/(F(p_end)-F(p_ini)));
```

```

Function [fp,fs,twid,twid_i,ripple_pass,drpm1,drpm1_dB,ripple_stop,stop_att] =
computeFOM(signal,f,db6i)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% USAGE:          [fp,fs,twid,twid_i,ripple_pass,drpm1,drpm1_dB,ripple_stop,stop_att]=
%                  computeFOM(signal,f,db6i)
%
%   AUTHOR: Carlos M. Fernández Reyes
%
% DESCRIPTION: Calcula las diferentes figuras de meritos de un filtro
%              paso bajo con alta tolerancia al ruido según su respuesta
%              en frecuencia.
%
%   INPUTS: signal: Respuesta en frecuencia del filtro a caracterizar.
%           f: Vector de frecuencia de la señal.
%           db6i: Punto donde la respuesta en frecuencia esta a 6dB.
%
%   OUTPUTS: fp: Frecuencia de corte en la banda de paso.
%           fs: Frecuencia de corte en la banda de rechazo.
%           twid: Banda de transicion Frecuencias del filtro.
%           twid_i: Ancho de muestras de la banda de transicion.
%           ripple_pass: Rizado en la banda de paso.
%           drpm1: Valor máximo de rizado en la banda de paso
%                 (método con respuestas muy planas).
%           drpm1_dB: Valor máximo de rizado en la banda de paso en dB.
%           ripple_stop: Rizado en la banda de atenuación.
%           stop_att: Atenuación producida entre la banda de paso y la
%                    de atenuacion.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate delta ripple passband for signal without high noise-distortion.
[drpm1,drpm1_i] = max(abs(signal));
drpm1 = drpm1 - 1;
drpm1_dB = 20*log10(drpm1);

if (db6i==1)
    [delP,delS,fid] = err(signal,f);
    twid=fid;twid_i=0;
    fp=0;fs=0;
    ripple_pass=delP;
    ripple_stop=delS;
    stop_att=delS;
else
    signaldb=20*log10(abs(signal));
    i=0;
    signal_stop=signaldb(db6i:1:end);
    signal_pass=signaldb(1:1:db6i-1);
    [elem_p_p,value_p_p]=findpeaks(signal_pass);    % higher than 0
    [elem_v_p,value_v_p]=findpeaks(signal_pass,'v');
    [min_ph_pi,m] = findpeaks(angle(signal(1:1:db6i-1)),'v');
    if((isempty(elem_v_p)) && (isempty(elem_p_p)))
        % Without positive and negative peaks
        [min_pass,p]=max(signal_pass);    % Ideal must be 0, max ripple
        ripple_pass=min_pass;
    else
        if isempty(elem_p_p)

```

```
[min_pass,p]=max(signal_pass);
ripple_pass=min_pass;
else
    [min_pass,p]=min(signal_pass(elem_p_p(end):-1:1));
    ripple_pass=max(signal_pass)-min_pass;
end
if((isempty(elem_v_p))|(min_pass<0))
    ripple_pass=max(signal_pass);
end;
end;
for (i=length(signal_pass):-1:1)
    if((signal_passband(i))>=min_pass)
        break;
    end;
end;
if (i==1)
    if(isempty(min_ph_pi))
        [min_ph_pi,m] = findpeaks(angle(signal(1:1:end)),'v');
    end;
    elem_fp=min_ph_pi(end);
    fp=f(min_ph_pi(end));
else
    elem_fp=i;
    fp=f(i);
end;
[min_ph_si,m] = findpeaks(angle(signal(db6i:1:end)),'v');
[value,max_v_i] = max(abs(mean(m(1:1:3))-m(1:1:3)));
min_ph_si(1)=min_ph_si(max_v_i);
% Maximum and minimum values of phase from last peak
[max_stop,Mi]=max(signal_stop(min_ph_si(1):1:length(signal_stop)));
[min_stop,mi]=min(signal_stop(min_ph_si(1):1:length(signal_stop)));
ripple_stop=max_stop-min_stop;
stop_att=max_stop;
for (i=1:1:length(signal_stop))
    if((signal_stop(i))<max_stop)
        break;
    end;
end;
elem_fs=i+length(signal_pass);
fs=f(i+length(signal_pass));
twid_i=elem_fs-elem_fp;
twid=fs-fp;
end;
```

ANEXO: HSPICE

Durante la realización del proyecto también se ha empleado HSPICE, aunque se han realizado varias descripciones de circuitos en este anexo se va a mostrar un extracto simplificado del fichero '.cir' (en los espacios con puntos suspensivos sigue una continuación con el número de elementos de retardo empleado) y la librería de transistores PTM usada.

```
*****
***** INVERTER CHAIN *****
*****

.parhier=local
.lib 'tech_wrapper.lib' mc

.param freq = 1e7
.param supply=1

VIN vin1 vss PULSE ('supply' 0 0 1e-12 1e-12 '1/(2*freq)' '1/freq')
VDD vdd vss 'supply'
VSS vss 0 0
VSS2 vss2 0 0
VDD2 vdd2 vss2 'supply'

.SUBCKT MYNOT input output vdd vss multfac='1'
xM1 output input vss vss MOSN w='multfac*120e-9' l='45e-9'
xM2 output input vdd vdd MOSP w='multfac*2*120e-9' l='45e-9'
.ENDS MYNOT

.param mfac=20
XM700 vin1 1 vdd2 vss2 MYNOT multfac='mfac'
XM1 1 2 vdd vss MYNOT multfac='mfac'
XM2 2 3 vdd vss MYNOT multfac='mfac'
XM3 3 4 vdd vss MYNOT multfac='mfac'
XM4 4 5 vdd vss MYNOT multfac='mfac'
```

```

XM5 5 6 vdd vss MYNOT multifac='mfac'
XM6 6 7 vdd vss MYNOT multifac='mfac'
XM7 7 8 vdd vss MYNOT multifac='mfac'
XM8 8 9 vdd vss MYNOT multifac='mfac'
XM9 9 10 vdd vss MYNOT multifac='mfac'
XM10 10 11 vdd vss MYNOT multifac='mfac'
...
...
XM510 510 511 vdd vss MYNOT multifac='mfac'
XM511 511 512 vdd vss MYNOT multifac='mfac'
XM512 512 513 vdd vss MYNOT multifac='mfac'
XM600 513 514 vdd2 vss2 MYNOT multifac='mfac'

*uncomment for nominal simulation
*.TRAN '1/(freq*1000)' '10*1/freq' uic
*uncomment for monte carlo simulation
.TRAN '1/(freq*100000)' '1.3/freq' start= '1/(3*freq)' sweep monte='5' uic
.option NUMDGT = 6 MEASDGT = 6 POST=1
*.option post=1

****DELAY INVERTER1
.measure tpdf1 TRIG v(1) VAL=0.5 RISE=1 TARG v(2) VAL=0.5 FALL=1
* falling prop delay total (prop delay falling total 1) - tpHL1
.measure tpdf1 TRIG v(1) VAL=0.5 RISE=1 TARG v(2) VAL=0.5 FALL=1
* falling prop delay (prop delay falling 1)
.measure tpdrt1 TRIG v(1) VAL=0.5 FALL=1 TARG v(2) VAL=0.5 RISE=1
* rising prop delay total (prop delay rising total 1) - tpLH1
.measure tpdr1 TRIG v(1) VAL=0.5 FALL=1 TARG v(2) VAL=0.5 RISE=1
* rise time
.measure tfall1 TRIG v(2) VAL='0.8*SUPPLY' FALL=1 TARG v(2) VAL='0.2*SUPPLY'
FALL=1 * fall time
.measure tpd1 param=(tpdr1+tpdf1)/2'
* average prop delay -tp-
***DELAY INVERTER2
.measure tpdf2 TRIG v(1) VAL=0.5 FALL=1 TARG v(3) VAL=0.5 FALL=1
.measure tpdf2 param='tpdf2-tpdrt1'
.measure tpdrt2 TRIG v(1) VAL=0.5 RISE=1 TARG v(3) VAL=0.5 RISE=1

```

```

.measure tpdr2 param='tpdrt2-tpdft1'
.measure tpd2 param='(tpdr2+tpdf2)/2'
....
....
****DELAY INVERTER512
.measure tpdft512 TRIG v(1) VAL=0.5 FALL=1 TARG v(513) VAL=0.5 FALL=1
.measure tpdf512 param='tpdft512-tpdrt511'
.measure tpdrt512 TRIG v(1) VAL=0.5 RISE=1 TARG v(513) VAL=0.5 RISE=1
.measure tpdr512 param='tpdrt512-tpdft511'
.measure tpd512 param='(tpdr512+tpdf512)/2'

.measure width param='mfac*120e-9'
.measure m_fac param='mfac'
.measure length param='45e-9'
.measure abs_variation_vt0 param='(2.5n)/sqrt(1*width*length)'
.measure abs_variation_betan param='(1.1n)/sqrt(1*width*length)'
.measure VDD param='supply'
.measure tpdm param='((tpdft512+tpdrt512)/2)/512'
***Desviación estándar.
.measure sd param='sqrt((((tpd1^2+tpd2^2 +..... +tpd511^2+tpd512^2)-(512*(tpdm^2)))/(512-1)))'
.END

```

```

*****

***** tech_wrapper.lib *****
*****

.lib tt
.include '45nm_LP.pm'

.subckt mosn d g s b mult='1' w='50e-9' l='45e-9'
m1 d g s b nmos w=w l=l m='mult'
.ends

.subckt mosp d g s b mult='1' w='50e-9' l='45e-9'
m1 d g s b pmos w=w l=l m='mult'
.ends

.endl

.lib mc
.include '45nm_LP.pm'

.subckt mosn d g s b mult='1' w='50e-9' l='45e-9'
.param abetan='1.1n'
.param avtn='2.5n'
.param dvtn=agauss(0,'avtn/sqrt(mult*w*1)',1)
.param dbetan_op_beta=agauss(0,'abetan/sqrt(mult*w*1)',1)
m1 d g s b nmos w=w l=l delvto=dvtn m='mult*(1+dbetan_op_beta)'
.ends

.subckt mosp d g s b mult='1' w='50e-9' l='45e-9'
.param abetan='1.1n'
.param avtn='2.5n'
.param dvtn=agauss(0,'avtn/sqrt(mult*w*1)',1)
.param dbetan_op_beta=agauss(0,'abetan/sqrt(mult*w*1)',1)
m1 d g s b pmos w=w l=l delvto=dvtn m='mult*(1+dbetan_op_beta)'
.ends

```

