Capítulo 3

Receptor para comunicaciones por línea de potencia

1. Elección de la arquitectura

2. Diseño de celdas

1 Receptor para comunicaciones por línea de potencia

1.1 Elección de la arquitectura

1.1.1 Descripción general del Transmisor-Receptor

El receptor que nos disponemos a diseñar en este proyecto se muestra descrito en la figura 13, acompañado de una primera etapa de transmisión. Todo nuestro diseño conservará la arquitectura clásica de un sistema de comunicaciones OFDM tradicional.



Fig. 1: Descripción funcional del Transmisor-Receptor

1.1.2 Elección de la cadena de filtrado y diezmado

En el presente documento se presentan una serie de cadenas de filtrado y diezmado, para las cuales se estudiará su función de transferencia |H(w)|(dB) para comprobar cual de ellas se va a comportar mejor en el rango de frecuencias que nos interesa, que será el *BW* de nuestra señal y la componente doble de la frecuencia de portadora (f_c). Cada modelo será analizado desde el punto de vista de la distorsión en amplitud y en fase.

También se analizará la *SNR* comparando los modelos de filtro en punto flotante que nos proporciona MATLAB y los modelos de filtros en punto fijo propios de nuestro diseño.



Fig. 2: Cadena de Filtrado y Diezmado CIC-HB-HB

Vamos a comenzar con la elección de los parámetros del filtro CIC.

Mediante un estudio previo se llegó a la conclusión de que una $R_{cic}=64$ sería una buena elección para nuestro diseño. A continuación comprobaremos como cambia |H(w)|(dB) en caso de que tengamos una R_{cic} diferente.



Fig. 3: Efecto de la variable R_{cic} en el diseño de la cadena CIC-HB-HB en el BW de nuestra señal



Fig. 4: Efecto de la variable R_{cic} en el diseño de la cadena CIC-HB-HB en el doble de la frecuencia f_c .

 R_{cic} define el cambio en la tasa de muestreo, y por ende, el número de lóbulos que se adicionarán a la banda de paso tras el diezmado. Aumentando conseguimos que la caída del filtro sea más abrupta, aunque atenuaremos más el rango de frecuencias de nuestra señal. La elección de una R_{cic} u otra dependerá del ruido de nuestra sistema, ya que sino hay suficiente ruido, con una R_{cic} de 64 sería suficiente.

Una vez conocida la tasa de muestreo R_{cic} , se variará el parámetro M_{cic} y valoraremos cual es el valor más adecuado.

- Parámetros del CIC: R=64 M∈ [1,4] N=1
- Parámetros de HB: N=20 Rizado=0.01



Fig. 5: Efecto de la variable M_{cic} en el diseño de la cadena CIC-HB-HB

 M_{cic} se define como el factor de retardo de los derivadores y nos define la posición de los lóbulos. La ganancia del filtro se hace cero en las frecuencias $f_k = \frac{2k}{M_{cic} \cdot R_{cic}}$, luego variando

M_{cic} se reduce el "ancho" de los lóbulos permitiendo hacer la caída del filtro más abrupta.

En nuestro caso nos interesa que el primer cero de transmisión (f_1) caiga por debajo del doble de la $\Omega_c \in [0.0097, 0.0488]$.

Observando la figura 17 vemos que nos interesa una M_{cic} lo más baja posible y así conseguir la menor atenuación posible en la banda de interés. Sin embargo, si escogiésemos $M_{cic} = 1$

comprobaríamos que el primer cero de transmisión cae dentro de la banda de $2\Omega_c \in [0.0195, 0.0976]$, y en caso de que nuestro primer cero de transmisión (f_1) caiga por encima de $2\Omega_c$ no estaríamos atenuando suficiente la componente parásita de la mezcla y estaríamos introduciendo más ruido ha nuestro sistema.

Por lo tanto nos quedaremos con \mathbf{M}_{cic} =2. Otro aspecto importante será cancelar el armónico situado en 2 Ω_c , para ello colocaremos un cero de transmisión f_k justo a esa frecuencia.

También podríamos pensar que poniendo un M_{cic} mayor podríamos atenuar más los armónicos que se encuentren fuera de la banda de interés, aunque de ese modo las componentes dentro de nuestra banda de interés se viesen mas atenuadas. Está elección dependerá del ruido que haya en nuestro sistema.

Debemos tener en cuenta que la respuesta la banda de paso no es plana y provocará distorsión en amplitud que será conveniente compensar mediante un ecualizador. Al aumentar los parámetros Rcic y Mcic estaremos aumentando la distorsión.

Una vez escogido el valor de M_{cic}, se variará el parámetro N_{cic}:

- Parámetros del CIC: R=64 M=2 N∈ [1,8]
- Parámetros de HB: N=20 Rizado=0.01



Fig. 6: Efecto de la variable N_{cic} en el diseño de la cadena CIC-HB-HB

 N_{cic} es el orden del filtro, que define el número de integradotes y derivadores.

 N_{cic} es el parámetro que más influye a la hora de calcular la ganancia máxima G_{max} del filtro CIC, la cual se encuentra en f \rightarrow 0. De la función de transferencia del filtro podemos ver que $G_{max} = \left(R_{cic} \cdot C_{cic}\right)^{N_{CIC}}$.

Debemos de tener en cuanta el dimensionamiento de los registros internos para evitar "overflow". Empleando representación en complemento a dos puede deducirse a partir de G_{max} que la salida del filtro debiese tener a lo menos B_{out} bits de representación:

$$B_{out} = \left[N_{cic} \cdot \log(R_{cic} \cdot M_{cic}) + B_{in} \right] bits$$

Donde B_{in} corresponde al número de bits de representación de la señal de entrada.

Este mismo número de bits es el necesario para los registros intermedios de los derivadotes e integradores.

Observando la gráfica vemos que la respuesta en frecuencia que menos atenúa mi rango deseado de frecuencias es con $N_{cic} = 1$, y además como no queremos disparar los bits de nuestros registros nos quedaremos con ese valor.

La estrategia que seguiremos para ahorrar recursos será la de considerar los ceros en los coeficientes de los filtros. Esta consideración consiste en omitir los multiplicadores cuando el coeficiente es igual a cero. Los filtros de media banda (HB) tendrán sistemáticamente valores nulos.

Una vez escogido los parámetros M_{cic} y N_{cic} de nuestro filtro CIC, comenzaremos con la elección de los parámetros del filtro HB. Empezaremos variando los parámetros N_{hb} y el rizado (δ):



Fig. 7: Efecto de la variable Rizado (δ) en el diseño de la cadena CIC-HB-HB



Fig. 8: Efecto de la variable Nhb en el diseño de la cadena CIC-HB-HB





1.1.2.2 <u>Modelo 2</u>



Fig. 11: Cadena de Filtrado y Diezmado CIC-HB-HB

Para las siguientes simulaciones utilizaremos los parámetros del CIC que obtuvimos en el 1º Modelo.



• Parámetros del CIC: R=64 M=2 N=1

Fig. 12: Efecto de las variables Fc en el diseño de la cadena CIC-FIR (R1=64 y R2=8)



Podemos observar que mediante el 2º Modelo conseguimos que la atenuación en la banda de frecuencias a estudiar es menor que con el 1º Modelo.

Debemos tener en cuenta que puede suceder que cuando tengamos en cuenta el coste hardware, puede que nos resulte más beneficioso utilizar el 1º Modelo.



Fig. 14: |H(w)|(dB) del filtro FIR



Fig. 15: Cadena de Filtrado y Diezmado CIC-CSD

Para trasladar los coeficientes de los filtros FIR, usaremos un programa en MATLAB que nos generará los coeficientes CSD de nuestro filtro FIR.

Deberemos tener especial cuidado a la hora de definir el tamaño de los registros, ya que será el aspecto que más me limitará mi diseño mediante los coeficientes CSD. A medida que el tamaño de estos registros aumente la precisión entre los filtros con coeficientes ideales y coeficientes CSD será mayor.

A la hora de implementar la parte hardware, debemos comprobar que dispondremos de suficientes recursos para realizar las sumas y restas, que en nuestro caso se habrán incrementado respecto al caso de utilizar coeficientes ideales.



Fig. 16: Efecto de la variable Ncsd en el diseño de la cadena CIC-CSD (R1=64 y R2=8)



Fig. 17: Efecto de la variable Fcsd en el diseño de la cadena CIC-CSD (R1=64 y R2=8)



Fig. 18: Cadena de Filtrado y Diezmado CIC-CSD-CSD



Fig. 19: Efecto de la variable Fcsd en el diseño de la cadena CIC-CSD-CSD (R1=64 ; R2=4; R3=2)

1.1.2.5 Conclusiones

Parece evidente que los modelos con los que conseguiremos una menor atenuación en la banda de interés serán los modelos 3 y 4. A continuación haremos una comparativa más detallada para decidir cuál será el más conveniente.



Fig. 20: Comparativa modelos 3 y4

Basándonos en la simplicidad y en que la respuesta en frecuencia de la cadena de filtrado es más lineal en la banda de frecuencias de interés hemos elegido el Modelo 3. A continuación se harán una serie de simulaciones que nos permitirán ajustar un poco más nuestra elección de la frecuencia de corte del filtro $FIR_{CSD}(f_{CSD})$ y el número de coeficientes del filtro (N_{CSD}) .

Debemos de hacer notar que N_{CSD} es mucho mayor que en el caso de un FIR ideal, ya que estamos utilizando coeficientes CSD, y esto nos da la posibilidad de incrementar N sin aumentar la complejidad hardware.





Fig. 21: Efecto de las variable Fcsd con Ncsd=40 en el diseño de la cadena CIC-CSD (R1=64 y R2=8)

Fig. 22: Efecto de la variable Fcsd con Ncsd=32 en el diseño de la cadena CIC-CSD (R1=64 y R2=8)

El siguiente paso en nuestro diseño será el de comprobar los posibles problemas de realizabilidad de los filtros que vamos a utilizar en nuestra cadena de filtrado y diezmado. Por ello empezaremos a trabajar con modelos HW, donde se presentarán diferentes configuraciones de los filtros FIR_{CSD} y CIC y proporcionaremos la SNR a la salida y así comprobar cual será nuestra configuración óptima.

Un problema que nos surgirá a la hora de realizar el filtro será la resolución de los coeficientes CSD. Hasta el momento, en las simulaciones anteriores hemos utilizados una resolución de 16 bits para representar los coeficientes CSD, lo cual supone un coste hardware muy elevado. Mediante la siguiente simulación se intenta llegar a una óptima elección del nº de bits para los coeficientes CSD que no nos suponga un deterioro de nuestra respuesta en frecuencia.



Fig. 23: Efecto de NB_COEF en el diseño de la cadena CIC-CSD (R1=64 y R2=8)

Filtro	Configuración	n	R	gd
CIC	M _{eic} =2 N _{cic} =1	-	64	63.5
(Cascade Integrator Comb)				
FIR _{CSD}	F _{CSD} =0.175	33	8	16
(FIR <u>Canonic</u> Signed Digit)				

Tabla 1. Especificaciones de la cadena de filtrado

Uno de los resultados más interesantes que podremos sacar de analizar nuestra cadena de filtrado será el retraso de grupo. La presencia de la etapa de diezmado después de cada filtro provocará que el retraso de grupo sea un número no entero¹. El retraso de grupo para la configuración especificada en la tabla 1 será de 2.124 muestras. La parte fraccional del retraso de grupo nos va a provocar mucha distorsión, por lo que intentaremos ajustar el orden del FIR_{CSD} para que el retraso de grupo se aproxime a un número entero.

1.2 Diseño de celdas

1.2.1 Celda de Conversión Directa Digital (DDC)

1.2.1.1 Fundamentos teóricos

Para comenzar, explicaremos de forma general en que consiste el principio de funcionamiento de nuestra celda DDC. A nuestro convertidor analógico-digital, le llegará una señal en banda de paso $s_{BP}(t)$, de la cual obtendremos una secuencia digital $s_{BP}(n)$, que se mezclará a la banda base mediante la multiplicación por la portadora digital que obtendremos del DDS.

Como resultado de esta mezcla digital, se obtendrá una secuencia digital compleja $s_{BB}(n)=s_{BB,I}(n) + is_{BB,Q}(n)$, la cual está caracterizada por tener una alta tasa de sobremuestreo u *OSR* (*Over-Sampling Ratio*).

¹ El retraso de grupo en el filtro CIC se ha calculado mediante $g_d = N \cdot \left(\frac{R \cdot M - 1}{2}\right)$

Esta tasa de sobremuestreo se define según:

$$OSR = \frac{fs}{B}$$

- f_S es la frecuencia de muestreo del convertidor.
- *B* es la frecuencia de Nyquist de la señal en banda base

Esta tasa de sobremuestreo será reducida mediante una etapa de filtrado y diezmado. La arquitectura de dicha etapa de filtrado y diezmado será estudiada con detenimiento en apartados posteriores. No existirá una única opción para esta decisión, pero se intentará tomar la arquitectura que provoque un comportamiento de nuestro diseño más óptimo.



Fig. 24: DDC (Direct Conversion Analog Front End)

1.2.1.2 Especificaciones funcionales del diseño

El siguiente apartado lo dedicaremos a especificar cuales serán las especificaciones funcionales de nuestro diseño, detallando y razonando el porqué de algunas de las decisiones que se tomarán de aquí en adelante.

- $BW_{BB} = 8Khz$
- $f_c = [40 200] Khz$
- $f_s = 16 \times 2(f_{c_{max}} + BW) = 6.656Msps \approx 8Msps$
- $OSR = \frac{f_s}{BW_{BP}} = \frac{8Mhz}{16Khz} = 500 \approx 512$

Para esta nueva OSR vamos a calcular de nuevo f_s y las respectivas frecuencias digitales (Ω).

• $f_s = BW_{RP} \times OSR = 16Khz \times 512 = 8.192Mhz$

•
$$BW_{BB} = 8Khz \rightarrow \Omega_{BB} = 0.00195$$

• $f_c = [40 - 200] Khz \rightarrow \Omega_c = [0.00976 - 0.04883]$

Un problema que debemos empezar a tener en cuenta desde ahora es el tema del reloj de la FPGA, y en nuestro caso dicha frecuencia de reloj será de 24 Mhz. La mayor parte de las FPGA's pueden bajar y en algunos casos subir, su frecuencia de reloj, pudiendo funcionar en el interior de nuestro diseño a una frecuencia inferior (p.e. 1Mhz) con objetivo de consumir menos potencia. Esto será un objetivo prioritario para nosotros, ya que como ya sabemos, para un circuito digital la potencia consumida es directamente proporcional a la tensión de alimentación y la frecuencia de conmutación.

Ya que la frecuencia de este reloj en nuestro caso será de 24 Mhz, no podremos generar la señal de 8.192 Mhz que necesitamos para la frecuencia de muestreo (f_s). La solución que adoptaremos ya que nuestras especificaciones no se agarran a ninguna norma será la de tomar como frecuencia de muestreo (f_s) una frecuencia de 8Mhz y recalcular el ancho de banda de nuestra señal.

$$BW_{BP} = \frac{f_s}{OSR} = \frac{8Mhz}{512} = 15.625Khz$$

Si existiera norma y nos pidieran explícitamente que el ancho de banda (*BW*) de nuestra señal fuera de 16 Khz, tendríamos que complicar la parte hardware de nuestro diseño, ya que la frecuencia de reloj y la frecuencia de muestreo no coinciden. Esta solución pasaría por la utilización de filtros Farrow.

Valor Parámetro Ancho de Banda 15.625 Khz 32 Número de portador as Separación entre portadoras 488 Hz Duración del símbolo OFDM 2.048 ms Frecuencia de muestreo 8 Mhz OSR 512 [40-200] Khz Εc

Las especificaciones de nuestro sistema se detallan en la siguiente tabla:

Tabla 2. Especificaciones funcionales del diseño

En la figura 37 podemos ver cuales serán los rangos de frecuencias que nos deben interesar en nuestro estudio:



Fig. 25: Bandas de frecuencias

1.2.1.3 Descripción de la arquitectura propuesta

El propósito de nuestra celda de conversión digital (DDC) será la de muestrear la señal analógica que le llegará, mediante un convertidor analógico-digital (ADC). El siguiente paso será la mezcla en banda base para obtener las correspondientes componentes I-Q. Esto se conseguirá a partir de la multiplicación de la señal muestreada por una portadora digital que obtendremos a partir de la celda DDS (*Direct Digital Synthesizer*). La configuración de esta celda será propósito de estudio en apartados posteriores.

Para llevar a cabo el filtrado y reducción de la tasa de sobremuestreo (*OSR*) se ha seleccionado, después de realizar un detallado estudio con multitud de posibilidades, la de una arquitectura compuesta por un filtro CIC (*Cascade Integrator Comb*) y un filtro FIR (*Finite Impulsive Response*).

El filtro FIR que utilizaremos será de la forma CSD (*Canonic Signed Digit*) para conseguir un consumo de área muy eficiente, ya que no requieren de la utilización de multiplicadores, a la vez que eliminan el ruido por cuantización de los coeficientes.



Fig. 26: Arquitectura hardware de la celda DDC

1.2.2 Celda DDS

1.2.2.1 Descripción y especificación funcional

Direct Digital Synthesis (DDS) es una técnica digital para la generación de una onda senoidal a partir de un reloj fuente con una frecuencia fija.

La arquitectura básica de un DDS puede observarse en la figura 39.





De esta figura se obtienen los tres bloques principales:

- Acumulador: Bloque digital que consiste en un sumador y en una realimentación hacia atrás.
- Convertidor de fase a amplitud: Bloque digital que convierte los valores de fase digital a valores de amplitud digital.
- Convertidor Digital Analógico: Convierte números en digital a una cantidad analógica escalada (voltaje o intensidad). Convierte el seno muestreado generado por el bloque digital a una señal continua (analógica).



Fig. 28: Síntesis de la onda senoidal

El acumulador de fase es actualmente un contador de modulo M que se incrementa cada vez que recibe un pulso de reloj. La magnitud del incremento está determinada por la palabra binaria de entrada M. Esta palabra forma el tamaño de paso de fase, su efectividad establecerá cuantos puntos contendrá el círculo de fase. El número de puntos de fase discretos contenidos en el anillo está determinado por la resolución de acumulador de fase (*n*) que determina la resolución del sintonizado del DDS. Para un acumulador de fase de n=28 bits, un valor 0000...001 de M resultaría en que el acumulador de fase desbordará después de 2²⁸ ciclos de reloj de referencia (incrementos). Si el valor de M se cambia a 0111...111, el acumulador de fase desbordará después de solo dos ciclos del reloj de referencia (el mínimo requerido por Nyquist). Esta relación se encuentra en la ecuación básica de sintonizado para la arquitectura DDS:

$$f_{out} = \frac{M \times f_c}{2^n}$$

Donde:

- f_{out} es la frecuencia de salida del DDS.
- *M* es la palabra binaria de sintonizado.
- f_c es la frecuencia interna del reloj de referencia.
- *n* es la longitud del acumulador de fase en bits.



Fig. 29: Círculo de fase digital

Cambios en el valor de *M* resultan en un cambio inmediato y continuo en fase en la frecuencia de salida. Mientras que la frecuencia de salida es incrementada, el número de muestras por ciclo se decrementa. Como la teoría de muestreo establece que al menos dos muestras por ciclos son requeridas para la reconstrucción de una forma de onda a la salida, la frecuencia fundamental máxima a la salida del DDS será $f_c/2$. Sin embargo, para aplicaciones prácticas, la frecuencia de salida está limitada por algo menos que eso, mejorando la calidad de la reconstrucción de la forma de onda y permitiendo filtrado a la salida. Cuando generamos una frecuencia constante, la salida del acumulador de fase se incrementa linealmente, así que la forma de onda analógica es inherentemente una rampa.

A continuación detallaremos cuales serán las especificaciones funcionales que intentaremos implementar en nuestra celda DDS. Si en cualquier momento se necesitará modificar cualquier valor de nuestros parámetros para conseguir un diseño más óptimo, se recalcularán todos los parámetros.

Parámetro	Valor
Frecuencia Normalizada	0.01
Portadora (Mhz)	0.08
Resolución en Frecuencia (Hz)	1
SFDR (dB)	60

Tabla 3. Especificaciones funcionales del DDS

1.2.2.2 Especificación hardware

La arquitectura que utilizaremos en nuestro diseño se basa en la suma incremental de una cantidad de fase que sirve como índice de una LUT donde se almacenan los valores del seno y coseno de un único cuadrante con precisión NB_DDS_DO. Para evitar un tamaño excesivo de la LUT se realiza una cuantización a la salida del acumulador.

La fidelidad de la señal generada está afectada por la cuantización de fase (debido a que la profundidad de la LUT es finita), y de amplitud (producida por el ancho NB_DDS_DO). En el dominio del tiempo dicha pérdida de resolución se traduce en jitter y cuantización de amplitud. En el dominio de la frecuencia aparecerán armónicos indeseados y un suelo de ruido de banda ancha.

La configuración de la frecuencia de portadora de la celda DDS está controlada por dos señales que nos disponemos a caracterizar:

- > *IDddsFreq*: el cual fija el valor inicial de la portadora.
- IDddsFreqOffset: que hará posible un ajuste fino a partir de la información facilitada por el sistema de sincronización.

Ambas señales nos proporcionan una frecuencia de portadora digital, donde f_{clk} es la frecuencia de reloj de la celda y NB_DDS_AC es la resolución en bits del acumulador de fase.

$$f_c = f_{clk} \frac{IDddsFreq}{2^{NB} DDS_AC} \rightarrow IDddsFreq = \frac{f_c \cdot 2^{NB} DDS_AC}{f_{clk}} = \frac{80Khz \cdot 2^{23}}{8Mhz} \approx 83886$$

$$\Delta f_c = f_{clk} \frac{IDddsFreqOffset}{2^{NB_-DDS_-AC}}$$

La resolución en frecuencia viene determinada por el ancho del acumulador según la siguiente expresión:

$$\Delta f = f_{clk} \cdot 2^{-NB_{-}DDS_{-}AC} \rightarrow NB_{-}DDS_{-}AC = -\log_2\left(\frac{\Delta f}{f_{clk}}\right) \approx 23bits$$

La cuantización del valor de la fase a la salida del acumulador producirá la aparición de componentes en el dominio de la frecuencia que son fruto del comportamiento periódico del error de cuantización de la fase. Para obtener una determinada relación SFDR la profundidad de la tabla ha de ser la mostrada por la siguiente ecuación:

$$NB _ DO = \left\lceil \frac{SFDR[dBc]}{6} \right\rceil = \left\lceil \frac{60dBc}{6} \right\rceil = 10bits$$



Fig. 30: Arquitectura hardware de la celda DDS

Parámetro	Valor
NB_DDS_DI	23
NB_DDS_DO	10
NB_DDS_AC	23
NB_DDS_LUT	10
CLK(Mhz)	8
IDddsFreq	83886

En la tabla 4 resumimos las especificaciones hardware que implementará nuestra celda DDS.

Tabla 4. Especificaciones hardware del DDS

1.2.2.3 Simulación Verilog

Para comprobar el correcto funcionamiento de nuestra celda DDS, se han realizado multitud de simulaciones Verilog de nuestra celda, comprobando que su funcionamiento corresponde con el esperado y que se cumplen las especificaciones funcionales que se plantearon en primer lugar.



Fig. 31: Simulación Verilog de la celda DDS para f_c=80Khz

1.2.2.4 Co-Simulación

A continuación, estudiaremos como responderá nuestro DDS ante las nuevas especificaciones que le impondremos, comprobando cual será su respuesta en el rango de frecuencias a estudiar. Para no tener que comprobar para cada una de las frecuencias el comportamiento del DDS, estudiaremos su respuesta para f_{min} , f_{max} y f_c , y comprobaremos que efectivamente el resultado obtenido en Verilog coincide con el que esperábamos, con un cierto margen de error provocado por la desviación en frecuencia. Si a medida que avanzara el proyecto fuera necesario un reajuste de la desviación en frecuencia, porque quisiéramos conseguir un mayor control de nuestra frecuencia, se recalcularían todos los parámetros.



Fig. 32: Respuesta en frecuencia de nuestra celda DDS en Verilog: (a) para una frecuencia de 40 Khz; (b) para una frecuencia de 200 Khz; y (c) para una frecuencia de 80 Khz.

Comprobamos que efectivamente la frecuencia que nos proporciona el DDS coincide con la que esperábamos, menos por ese error producido por la desviación en frecuencia.

1.2.3 Celda FFT

1.2.3.1 Descripción y especificación funcional

La transformada rápida de Fourier (*Fast Fourier Transform FFT*) es un algoritmo extremadamente rápido para calcular la transformada discreta de Fourier (*Discrete Fourier Transform DFT*). Existen distintos métodos para calcular la FFT, en general los podemos dividir en 2 tipos: decimación en el tiempo, y diezmado en la frecuencia. El algoritmo busca reducir el número de multiplicaciones efectuadas en la DFT, reduciendo el número de cálculos para N datos de $2N^2$ a $2N \cdot \log_2 N$, donde N debe ser una potencia de 2.

Partiendo de la DFT para una señal dada polinomialmente como:

$$X(k) = \sum_{n=0}^{N-1} x[n] w_N^{k-n}$$

Y desarrollando podemos llegar a generalizar el siguiente método:

Sea \dot{x} un vector de datos, de longitud N=2^m. Entonces sobre el intervalo [0,N-1] se pueden realizar m particiones hasta llegar a una DFT de longitud 2, esta es la unidad básica de FFT conocida como mariposa (Butterfly) en donde se necesitará de una multiplicación y 2 sumas complejas.



Fig. 33: Los elementos computacionales básicos de la transformada rápida de Fourier es la mariposa. Toma dos números complejos, representados por a y b, y forma las cantidades mostradas. Cada mariposa requiere una multiplicación compleja y dos sumas complejas.

1.2.3.2 Especificación hardware

Para nuestro diseño utilizaremos una FFT de 32 puntos que se realizará empleando una arquitectura *radix-2* basada en Butterfly, cuya arquitectura podemos apreciar en la figura 46. La celda diseñada empleará la técnica del escalado de bloque en cada iteración para mantener el rango dinámico y mejorar la relación SNR.

Nuestra celda FFT utilizará los siguientes parámetros de diseño:

- NB_MEM: Ancho de la memoria de doble puerto que se encargará de almacenar los datos de entrada y las operaciones intermedias.
- NB_TRMULT: Ancho de los multiplicadores reales empleados en la multiplicación compleja. La cuantización se lleva a cabo inmediatamente después del producto y antes de la suma de las componentes. Tras la suma se realiza una saturación controlada obteniéndose como resultado un escalado de bloque.
- NB_TWE: Ancho de los coeficientes complejos que recorren el circulo unitario del plano complejo (denominado *tweedle*).



Fig. 34: Arquitectura hardware de la celda FFT

A continuación detallaremos cuales serán las especificaciones hardware que intentaremos implementar en nuestra celda FFT. Si en cualquier momento se necesitara modificar cualquier valor de nuestros parámetros para conseguir un diseño más óptimo, se recalcularán todos los parámetros.

Parámetro	Valor
NB_MEM	10
NB_TWE	10
NB_TRMUL	11

Tabla 1. Especificaciones hardware de la celda FFT

1.2.3.3 Simulación Verilog

Para comprobar el correcto funcionamiento de nuestra celda FFT, se han realizado simulaciones Verilog de nuestra celda utilizando tramas de 32 símbolos OFDM modulados con todas las modulaciones disponibles, comprobando que su funcionamiento corresponde con el esperado.



Fig. 35: Simulación Verilog de la celda FFT para comprobar su correcto funcionamiento donde se puede apreciar la técnica de escalado de bloque

1.2.3.4 Co-Simulación

3.2.3.4.1 Obtención Hardware

Para hallar el error por representación y procesamiento en punto fijo compararemos un modelo de FFT en punto flotante (modelo ideal que nos proporciona Matlab) con otro modelo de FFT en punto fijo (modelo hardware que utilizaremos en nuestras simulaciones). Estas simulaciones

nos servirán de primer contacto, para comprobar como varía la SNR de nuestra celda a medida que varían los parámetros de diseño. Pero realmente, será en las simulaciones que realizaremos con nuestro modelo Verilog, donde tomemos la decisión final de cual será la configuración de nuestra celda.

Calcularemos la SNR tanto en transmisión como en recepción mediante los esquemas que se muestran en las figura 48 y figura 49.



Fig. 36: Esquema para el cálculo de la SNR en transmisión de la celda FFT



Fig. 37: Esquema para el cálculo de la SNR en recepción de la celda FFT

Mediante las siguientes gráficas se conseguirá el ajuste de los parámetros de configuración de la celda FFT. Para ello se ha simulado la transmisión y recepción de tramas de 32 símbolos OFDM modulados en DBPSK (128 bytes por trama).

Para el proceso de optimización se comenzará variando el principal recurso consumido por la celda que en nuestro caso es el tamaño de las memorias (NB_MEM) como se muestra en la figura 50. Podemos comprobar que su comportamiento será lineal en todo el rango analizado.

El segundo recurso más consumido por nuestra celda será el ancho de los coeficientes complejos (NB_TWE). Podemos observar en la figura 51 como la curva satura debido a que hemos impuesto con anterioridad el valor del tamaño de las memorias.

Para el cálculo del ancho de los multiplicadores (NB_TRMUL) actuaremos de igual forma, fijando los valores que se han obtenido anteriormente y obtendremos nuevamente una curva que satura debido a estas imposiciones. El resultado lo podemos observar en la figura 52.

Parámetro	Modulación	SNR _{Ts} /SNR _{Rx} (dB)
NB_MEM = 10 bits	DBPSK_k	57/35
NB_TWE = 10 bits	DBPSK_k	60/39
NB_TRMUL = 11 bits	DBPSK_k	58/40

Tabla 6. Parámetros de diseño de la celda FFT



Fig. 38: SNR de cuantización a la salida de la i(FFT) frente al tamaño de las memorias (NB_MEM)



Fig. 39: SNR de cuantización a la salida de la i(FFT) frente al tamaño del tweedle (NB_TWE)



Fig. 40: SNR de cuantización a la salida de la i(FFT) frente al tamaño de los multiplicadores (NB_TRMUL)

A pesar de haber alcanzado una configuración para nuestros parámetros de diseño, será ahora, con las simulaciones en Verilog cuando tomaremos nuestra decisión sobre los parámetros de la celda. Tenemos que tener en cuenta que los resultados que hemos obtenido han sido utilizando una celda FFT modelada en Matlab y por ello obtenemos SNR's que rondan los 60 dB, lo cual no es del todo real. Podemos considerar como razonables, si las SNR rondan los 30 dB para nuestra celda FFT, que serán los resultados que obtendremos a continuación.

Para llevar a cabo estas simulaciones compararemos el modelo ideal de la FFT en Matlab con nuestro modelo hardware de la FFT en Verilog.

Los esquemas que utilizaremos para el cálculo de la SNR serán los mismos a los utilizados en las simulaciones anteriores.

Parámetro	Modulación	SNR _{Ty} /SNR _{Ry} (dB)		
		Min.	Max.	
NB_MEM = 10 bits	DBPSK_k	33/31	35/32	43/40
NB_TWE = 10 bits				
NB_TRMUL = 12 bits				

Tabla 7. Ajuste de los parámetros de diseño de la celda FFT

3.2.3.4.2 Rango dinámico

Uno de los aspectos fundamentales que tenemos que tener en cuenta es el rango dinámico de nuestra señal. A modo de comprobación haremos unas simulaciones para comprobar que con la configuración de los parámetros de diseño que hemos escogido, nuestra celda tiene unos valores de rango dinámico correctos. Para ello se ha simulado la transmisión y recepción de tramas de 32 símbolos OFDM modulados en DBPSK (128 bytes por trama).



Fig. 41: Rango dinámico a partir de una simulación Verilog de la celda FFT para una trama de 32 símbolos OFDM modulada en DBPSK.

1.2.4 Celda Mixer

1.2.4.1 Descripción y especificación funcional

La función principal de nuestra celda Mixer es corregir el offset en frecuencia que sufre la señal en el receptor. Como se observa en la figura 54, en el receptor tenemos un par de mezcladores que multiplican la señal para obtener sus componentes en fase y cuadratura. Lo que ocurre es que estos mezcladores poseen un reloj distinto al del receptor por lo que puede existir cierta diferencia entre ambos. Esta es la diferencia que deberemos corregir.



Offset de frecuencia: $\Delta I = F_{c,TX}$. $F_{c,RX}$

Fig. 42: DDC (Direct Conversion Analog Front End)

El offset en frecuencia provoca la aparición de dos errores: perder la referencia de portadoras e introducir ICI. El ICI viene provocado por la pérdida de ortogonalidad entre símbolos OFDM que se produce debido al offset. Es este segundo error el que tratamos de corregir en este bloque, gráficamente:



Fig. 43: Offset de frecuencia

De la figura 55:

- \blacktriangleright 1/T_s es la separación entre portadoras.
- Senéricamente: $\Delta f = n x (1/T_s) + \delta f$ donde:
 - \circ *n* es entero.
 - o δf es un número real entre $[-1/(2T_s):1/(2T_s)]$.
- \succ *n* no provoca ICI, aunque se pierde la referencia de las portadoras.
- > el verdadero responsable de la perdida de la ortogonalidad es δf .
 - o tenemos que corregirlo antes de pasar al dominio de la frecuencia.

1.2.4.2 Especificación hardware

Primeramente empezaremos por especificar cual será la arquitectura hardware que utilizará nuestra celda Mixer en la parte de transmisión, especificando cuales serán los parámetros hardware que implementará dicha celda.



Fig. 44: Arquitectura hardware de la celda Mixer-Tx

Celda	Parámetro	Valor
	NB_DI (bits)	10
	NB_DDS (bits)	10
MIXER	NB_MULTR (bits)	12
	NB_ADDER (bits)	13
	NB_ADDERSAT (bits)	12
	NB_MIXER_TX (bits)	10

Tabla 8. Parámetros hardware de la celda Mixer-Tx

Análogamente a la parte de transmisión, detallamos a continuación cual será la arquitectura hardware y los parámetros hardware en nuestra parte de recepción.



Fig. 45: Arquitectura hardware de la celda Mixer-Rx

Celda	Parámetro	Valor
	NB_DI (bits)	10
	NB_DDS (bits)	10
MIXER	NB_MULTR (bits)	12
	NB_DO (bits)	10



1.2.4.3 Co-Simulación

3.2.4.3.1 Rango dinámico

Para comprobar la correcta funcionalidad de la celda Mixer, introduciremos una señal escalón en la celda, para así comprobar cual será su comportamiento ante una señal en fondo de escala.



Fig. 46: Comparativa de las Salidas del Adder y AdderSat en la celda Mixer-Tx

1.2.5 Celda CIC

1.2.5.1 Descripción y especificación funcional

Una vez que hemos realizado la mezcla a la banda base y hemos obtenido las componentes I-Q a partir de la multiplicación por una portadora digital proporcionada por la celda DDS necesitamos filtrar las señales I-Q para reducir la tasa de muestreo y quedarnos con la información que deseamos.

Las muestras llegarán a la cadena de filtrado y diezmado a unas tasas de varios Mbps y saldrán con una tasa de muestreo del orden de Khz. Esta reducción no será posible mediante filtros FIR

tradicionales, por lo que tendremos que recurrir a los filtros de Hogenauer más conocidos como filtros CIC.

Estos filtros pasobajos tienen la ventaja de implementarse exclusivamente con sumas, restas y retardos. Una de las principales características de estos filtros es que se pueden diseñar para lograr caídas suficientemente pronunciadas.

El filtro CIC se construye colocando en serie integradores, un decimador de muestras y una serie de derivadores, conociendo dicha estructura con el nombre de *Cascaded integrador-comb CIC filter*.



Fig. 47: Cascaded integrador-comb CIC filter

Observamos que los integradores son implementados como sumadores realimentados. Seguidamente se coloca una etapa de decimación, cuya salida corresponde con la entrada submuestreada por un factor entero R (se toma una muestra de salida cada R muestras de entrada). Finalmente nos encontramos con una serie de derivadores, implementados como restadores de la muestra actual menos la muestra anterior (o bien la muestra de M tiempos anteriores si $M \neq 1$). Es muy importante darnos cuenta que la segunda sección del sistema funciona con una frecuencia de reloj distinta que la de la entrada, la nueva frecuencia de reloj es $f_{clk2}=f_{clk}/R$.

El número de integradores debe ser idéntico al número de derivadores y será denominado como *N*.

La función de transferencia del filtro completo es:

$$H_{CIC}(z) = \frac{(1 - z^{-RM})^{N}}{(1 - z^{-1})^{N}}$$

Estos filtros tienen tan solo tres variables: N el orden del filtro, que define el número de integradores y derivadores; R el factor de decimación y M el factor de retardo de los derivadores.

Cuanto mayor es el orden del filtro más abrupta es la caída del filtro y mayor la atenuación de los lóbulos reflejados. Por su parte R define el cambio en la tasa de muestreo, y por ende, el número de lóbulos que se adicionarán a la banda de paso tras el submuestreo. Finalmente M define la posición de los lóbulos. La ganancia del filtro se hace cero en las frecuencias

 $f = i \cdot \frac{f_{clk}}{R} \cdot \frac{1}{M}$ i = 1, 2, ..., luego variando *M* se reduce el ancho de los lóbulos permitiendo

hacer la caída del filtro aún más abrupta.

La característica pasobajo del filtro nos proporciona una ganancia máxima G_{max} que se encuentra centrada a frecuencia cero.

$$G_{\rm max} = (RM)^N$$

Luego debemos tener presente que debemos dimensionar los registros internos para evitar overflow. Empleando representación en complemento a dos puede deducirse a partir de G_{max} que la salida del filtro debe tener al menos B_{out} bits de representación

$$B_{out} = \left\lceil N \log_2(RM) + B_{in} \right\rceil [bits]$$

donde B_{in} corresponde al número de bits de representación de la señal de entrada.

A continuación detallaremos cuales serán las especificaciones funcionales que intentaremos implementar en nuestra celda CIC.

Parámetro	Valor
Tasa de <u>diezmado</u>	64
Máxima distorsión de amplitud (dB)	42
Ancho de banda normalizado a la entrada	0.0019
Ancho del lóbulo principal	128
Retraso diferencial	2
Número de etapas	1

Tabla 10. Especificaciones funcionales del filtro CIC

1.2.5.2 Especificación hardware

En este tipo de filtros es muy importante prestar atención a la sección recursiva del filtro, y al posible desbordamiento del acumulador. Cuando esta sección recursiva procesa una señal de entrada con una cierta componente en DC se produce un crecimiento indefinido del acumulador.

La propiedad cíclica presente en las operaciones realizadas en aritmética en complemento a dos nos permite emplear un tamaño de acumulador que resulta de sumar al ancho de bus de entrada una cantidad:

$$b = \left\lceil N \cdot \log_2 \left(R \cdot M_{cic} \right) \right\rceil$$

Este ajuste nos permitirá garantizar que ante una entrada en escalón sólo se producirá una rotación cíclica entre dos muestras restadas en el derivador.



Fig. 48: Arquitectura hardware del filtro CIC

Filtro	Parámetro	Valor
	NB_DI (bits)	12
CIC	NB_INT (bits)	20
(Cascade Integrator Comb)	NB_CIC_SAT (bits)	19
R _{cia} =64 M _{cic} =2 N _{cia} =1	NB_DO (bits)	12
	SNR (dB)	68



1.2.5.3 Simulación funcional

Para mostrar como se comportará nuestro filtro CIC, realizaremos una serie de simulaciones variando los parámetros de diseño del filtro (R_{cic} , M_{cic} y N_{cic}) evaluando cuales serán los más convenientes para nuestro diseño y escogiendo aquellos que finalmente utilizaremos en nuestro diseño final.





Fig. 49: Efecto de la variable R_{cic} en el filtro CIC



Fig. 51: Efecto de la variable N_{cic} en el filtro CIC

Fig. 50: Efecto de la variable M_{cic} en el filtro CIC



Fig. 52: Comportamiento del filtro CIC con las variables de diseño seleccionadas

Mediante las simulaciones anteriores no podemos apreciar la zona que más no interesa, que será aquella que coincida con el ancho de banda de entrada de nuestra señal. No obstante, estas simulaciones se realizarán en la elección de la cadena de filtrado, donde se analizarán todos los parámetros de diseño y se escogerán aquellos que nos proporcionen un diseño más óptimo.

En la siguiente gráfica se mostrará la ganancia introducida por nuestro filtro CIC para nuestro ancho de banda.



Fig. 53: Respuesta en frecuencia del filtro CIC

De las figura 65 se obtiene que la ganancia introducida por nuestra celda CIC son de unos 42 dB para un ancho de banda de 15.625MHz. En la tabla 12 se especifican los valores obtenidos para nuestro ancho de banda.

FRECUENCIA NORMALIZADA	GANANCIA
0.0019	42 dB

Tabla 2. Respuesta en frecuencia del filtro CIC

La atenuación introducida en torno a la frecuencia fs/2 es lo suficiente como para eliminar cualquier componente que se encuentre en esa situación, que era uno de los objetivos que queríamos conseguir con el filtro CIC.

1.2.5.4 Simulación Verilog

Para comprobar el correcto funcionamiento de nuestra celda CIC en Verilog, hemos comparado la celda en Verilog con la de Matlab, obteniendo que la SNR obtenida que compara ambas celdas tiende a infinito.

También se ha realizado un test del rango dinámico y el máximo de señal para comprobar que no existe ningún error en la configuración de los parámetros de la celda.





1.2.5.5 <u>Co-Simulación</u>

Para hallar el error por representación y procesamiento en punto fijo compararemos un modelo de filtros ideales obtenidos de Matlab en punto flotante con otro modelo de filtros con un comportamiento hardware idéntico al que utilizaremos.



Fig. 55: Esquema para el cálculo de la SNR en la celda CIC

Filtro	Parámetro	1	2	3	4	5	6	7	8	9	10
	NB_DI (bits)	10	10	10	12	12	12	14	14	14	14
CIC	NB_INT (bits)	19	19	19	20	20	20	23	23	23	23
(Cascade Integrator Comb)	NB_CIC_SAT (bits)	17	17	17	19	19	19	21	21	21	21
R _{cic} =64 M _{cic} =2 N _{cic} =1	NB_DO (bits)	12	13	14	10	12	14	11	12	14	16
	SNR (dB)	61	68	70	63	68	73	78	81	85	90

Tabla 13. Parámetros de diseño del filtro CIC

1.2.6 Celda FIR

1.2.6.1 Descripción y especificación funcional

Los filtros implementados usando la representación numérica Canonical Signed Digit (CSD) para los coeficientes han sido estudiados como un modo de mejorar la velocidad y reducir la complejidad hardware.

La representación en SD (Signed Digit) de un número en base 2 se expresará de la siguiente forma:

$$x = \sum_{i=1}^{K} c_i 2^{-n_i}$$

Donde $c_i \in \{-1,0,1\}$ y $n_i \in \{0,1,...,M\}$. De todas las representaciones posibles habrá una con el menor número de valores distintos de 0 (*forma canónica*). La representación en SD tiene un total de M+1 dígitos y K dígitos distintos de 0. Una representación en CSD se define como la

mínima representación para la cual dos dígitos distintos de 0 no son adyacentes. Así el mínimo número de sumas y restas requeridas para realizar el coeficiente es uno menos que el número de dígitos distintos de 0 en la representación.



Fig. 56: Esquema general de un filtro FIR

Esta celda se trata tan solo de un filtro FIR de 33° orden, cuyos coeficientes han sido calculados por simulación y son los siguientes:

H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	Η
/17	/18	/19	/20	/21	/22	/23	/24	/25	/26	/27	/28	/29	/30	/31	/32	16
0	0	0	0	0	-1	-2	-2	-3	-3	-1	2	7	12	17	21	22

Tabla 3. Coeficientes del filtro $\ensuremath{\text{FIR}_{\text{CSD}}}$

A continuación detallaremos cuales serán las especificaciones funcionales que intentaremos implementar en nuestra celda FIR.

Parámetro	Valor					
Frecuencia de corte normalizada	0.175					
Distorsión maxima de amplitud (dB)	1.24					
Retraso de grupo en banda de paso	63.5					
Tasa de <u>diezmado</u>	8					
Orden del filtro	33					
<u>Tipo</u> de filtro	FIR <u>Simétrico</u> par					
Arquitectura	CSD					

Tabla 15. Especificaciones funcionales del filtro FIR

1.2.6.2 Especificación hardware

La arquitectura hardware que utilizará nuestra celda CIC se muestra en la figura 69, especificando cuales serán los parámetros hardware que implementará dicha celda.



Fig. 57: Arquitectura hardware del filtro FIR

Filtro	Parámetro	Valor
	NB_DI (bits)	12
	NB_COEF (bits)	8
FIR _{CSD}	NB_MULT (bits)	14
(FIR <u>Canonic</u> Signed Digit)	NB_INT (bits)	14
N _{CSD} =33 F _{CSD} =0.175	NB_SAT (bits)	14
R=8	NB_DO (bits)	10
	SNR (dB)	50

Tabla 16. Parámetros hardware del filtro FIR_{CSD}

1.2.6.3 Simulación funcional

En la siguiente gráfica se mostrará la atenuación introducida por nuestro filtro para nuestro ancho de banda.



Fig. 58: Respuesta en frecuencia del filtro FIR

De las figura 70 se obtiene que las atenuaciones introducidas por nuestra celda FIR son de unos 1.24 dB para un ancho de banda de 15.625MHz. En la tabla 17 se especifican los valores obtenidos para nuestro ancho de banda.

FRECUENCIA NORMALIZADA	ATENUACIÓN
0.125	1.24 dB

Tabla 4. Respuesta en frecuencia del filtro FIR

Podemos observar en la figura 70 como la atenuación introducida en torno a la frecuencia fs/2 es lo suficiente como para eliminar cualquier componente que se encuentre en esa situación, que es lo que también queríamos conseguir con este filtro.

1.2.6.4 <u>Simulación Verilog</u>

Para comprobar el correcto funcionamiento de la celda FIR en Verilog, hemos comparado la celda en Verilog con la de Matlab, obteniendo que la SNR obtenida que compara ambas celdas tiende a infinito.

Al igual que para la celda CIC se ha realizado un test del rango dinámico y el máximo de señal para comprobar que no existe ningún error en la configuración de los parámetros de la celda.



Fig. 59: Simulación Verilog de la celda FIR para comprobar su correcto funcionamiento ante una entrada escalón

1.2.6.5 <u>Co-Simulación</u>

Para hallar el error por representación y procesamiento en punto fijo compararemos un modelo de filtros ideales obtenidos de Matlab en punto flotante con otro modelo de filtros con un comportamiento hardware idéntico al que utilizaremos.



Fig. 60: Esquema para el cálculo de la SNR en la celda FIR

Filtro	Parámetro	1	2	3	4	5	6	7	8	9	10
	NB_DI (bits)	10	10	10	10	12	12	12	14	12	14
	NB_COEF (bits)	8	8	8	8	8	8	8	8	8	8
FIR _{CSD}	NB_MULT (bits)	13	14	17	13	14	18	17	17	19	21
(FIR Canonic Signed Digit)	NB_INT (bits)	14	15	18	14	14	19	16	18	20	22
N _{CSD} =40 F _{CSD} =0.175	NB_SAT (bits)	13	14	17	13	14	18	17	17	19	21
R=8	NB_DO (bits)	10	10	10	10	10	12	12	12	14	14
	SNR (dB)	42	48	51	44	50	58	62	62	70	75

Tabla 18. Parámetros de diseño del filtro FIR_{CSD}