Universidad de Alcalá

Programa de Doctorado en Electrónica: Sistemas Electrónicos Avanzados. Sistemas Inteligentes

Design of Efficient Heterogeneous Architectures for Broadband Power-Line Communications

Tesis Doctoral presentada por

Rubén Nieto Capuchino

2020

Universidad de Alcalá

Programa de Doctorado en Electrónica: Sistemas Electrónicos Avanzados. Sistemas Inteligentes

Diseño de Arquitecturas Eficientes Heterogéneas para Comunicaciones de Banda Ancha sobre Redes Eléctricas

Tesis Doctoral presentada por

Rubén Nieto Capuchino

Directores

Raúl Mateos Gil

Álvaro Hernández Alonso

Alcalá de Henares, enero del 2020

A mis padres y hermana.
Otra vez, y nunca serán bastantes.

"Dejen que el futuro diga la verdad y evalúe a cada uno de acuerdo a sus trabajos y a sus logros. El presente es de ellos, pero el futuro, por el cual trabajé tanto, es mío"
Nikola Tesla (1856-1943)

Agradecimientos

Esta tesis ha sido el resultado de muchas horas de trabajo y esfuerzo. Sin embargo, si echo la vista atrás, es también algo que no hubiera imaginado que fuera a conseguir hace unos años.

En primer lugar, quiero agradecer este trabajo a mis tutores Raúl Mateos y Álvaro Hernández, por el apoyo, conocimiento, metodologías y valores que me han aportado durante estos años que he estado trabajando con ellos. Ha sido una fortuna haber podido dedicar este tiempo con ellos y conocerlos tanto profesionalmente, como personalmente. Creo que no podré agradecer nunca todo lo que me han aportado, y nunca serán suficientes las veces que lo haga.

No me puedo olvidar a mis padres y hermana, por el apoyo que me han dado durante estos años. Aunque no han podido aportar aspectos técnicos, siempre se han preocupado por saber qué estaba haciendo y, sobre todo, para qué iba a servir. También agradecerles la paciencia que han tenido en ciertos momentos críticos en la vida de un doctorando. Agradecer a Ruth, mi pareja, por todo lo que me ha aguantado y me han comprendido durante esta etapa, así como todo lo que me ha ayudado y apoyado cuando le contaba cosas de mi tesis y mis cosas. Agradecer también todo el apoyo recibido por el resto de mi familia: abuelas, tíos/as y primos/as.

También agradecer al grupo de investigación GEINTRA. Por dejarme formar parte, de forma directa e indirecta, en los proyectos que se desarrollaban. En ocasiones ofreciendo ayuda, y en otras solicitándola, pero siempre con el objetivo de aportar y contribuir. Por la parte de profesores: Jesús U., Juan Jesús, Ana, Villa y M. Carmen; les tengo que agradecer que en cierto modo contaran conmigo y mi humilde opinión o aportación para sus cosas, que a veces han sido académicas y otras extra académicas.

Por otro lado, no me puedo olvidar del motor de conocimiento que mueve GEINTRA, siendo los doctorandos, becarios, contratados, etc. los que pasamos horas y horas en los fondos de pasillo. Gracias a la gente del Fondo O22 (Sara, Miguel, Fran, Ismael, Carlos, Alessandro, MMR (O21)), O31 (Gualda, Elena, Hans, Paco, Martín, Santiago, Edel y Sergios porque hay muchos), O33 (Álvaro, David y Rodrigo) y E34 (Diego, Boby y el otro) por compartir problemas del laboratorio, papeleos, dudas, miedos, preocupaciones, en los momentos de café o en las comidas. También agradecer a la gente de iSpace, que son GEINTRA, pero habitan en un ecosistema propio fuera de los fondos de pasillo (Casillas,

Fuentes, Leti, Juanma, Fran y Frank), por los ratos preparando el curso de MATLAB® y otros momentos más lúdicos. A todos mis compañeros, de fuera y dentro del laboratorio, agradecerles el apoyo, la ayuda, los consejos y los momentos de desahogo que se tienen a lo largo de la tesis.

Cierro este capítulo de agradecimientos con la sensación de que puedo estar olvidando a alguien. Pero estoy convencido que, si es así, me dirán algo como: "Anda qué me pusiste en los agradecimientos...". Por esta razón, quiero cerrar el capítulo agradeciendo a aquellos que me haya podido olvidar y justificar el olvido diciendo que ha sido por culpa de los malditos *deadlines*.

A todos, gracias.

Resumen

Para establecer la comunicación a través de la red eléctrica (PLC), el estándar IEEE 1901/2010 propone como técnica de acceso al medio las modulaciones multiportadora basadas en bancos de filtros (FBMC). Sin embargo, una desventaja que presentan las comunicaciones a través de la red eléctrica es el canal de comunicación, el cual contiene considerable ruido y notables interferencias. No obstante, el uso de técnicas de estimación de canal permite obtener un modelo de canal PLC para compensar, mediante un igualador de canal, los efectos no deseados introducidos por el canal. Esto hace que el sistema sea más complejo y requiera de una mayor carga computacional, sobre todo en la etapa de recepción. Por esta razón, se va a llevar a cabo un estudio de diferentes arquitecturas para su implementación.

Esta tesis presenta el diseño de arquitecturas heterogéneas para comunicaciones PLC de banda ancha. Para ello, se lleva a cabo un análisis de la técnica de acceso al medio que se va a utilizar, así como de las técnicas de estimación e igualación de canal. Este análisis describe matemáticamente cada una de las etapas que componen el sistema FBMC, tanto la etapa de transmisión, como la de recepción. Entre los métodos de estimación de canal se han seleccionado los que presentan menor carga computacional, como son los estimadores basados en mínimos cuadrados (LS). Por otro lado, entre las técnicas de igualación para sistemas multiportadora basadas en banco de filtros, se pueden encontrar los ecualizadores ASCET, los cuales presentan una arquitectura de filtros FIR para llevar a cabo la ecualización.

Los dispositivos FPGA permiten soportar la carga computacional que presenta el sistema de forma considerable. Además, algunos SoC integran un procesador ARM junto con la FPGA, haciendo que la propuesta de la arquitectura heterogénea se plantee para los dispositivos de la familia Zynq® de Xilinx. Dentro de la arquitectura, aquellas partes con mayores exigencias temporales se implementarán en la lógica programable del dispositivo; por otro lado, las partes que presenten restricciones temporales más laxas se codificarán para que sean ejecutadas en el procesador, definiendo así la arquitectura mixta HW/SW.

Cada uno de los bloques que compone la parte hardware de la arquitectura serán modelados mediante herramientas de síntesis de alto nivel (HLS). Por otro lado, con el fin de obtener un consumo de recursos hardware eficiente se realiza un estudio del *datapath* para ajustar el ancho de palabra a los bloques DSP del dispositivo.

Para las partes software de la arquitectura se describirán los métodos de aceleración disponibles en el dispositivo. Cabe destacar que se hace uso de la unidad vectorial NEON para acelerar ciertos cálculos complejos como la FFT. Además, la implementación software se extiende para usar todos los núcleos disponibles del procesador con el fin de alcanzar una mayor aceleración. Esto se realiza para los dispositivos Zynq® 7000 con uno y dos núcleos, y para Zynq® UltraScale+ con uno, dos y cuatro núcleos, estableciendo los mecanismos de comunicación entre núcleos para ambos casos.

Palabras clave: Power-Line Communications, Filter Bank Multi-Carrier, Arquitectura Heterogénea, Multi-Processor System-on-Chip, Ecualizador ASCET.

Abstract

In order to establish communication over a power line, the IEEE 1901/2010 standard for Power-Line Communications (PLC) proposes Filter-Bank Multi-Carrier Modulations (FBMC) as medium access technique. However, a disadvantage of communications through the mains is the channel, which contains considerable noise and significant interference. Nevertheless, the use of channel estimation techniques, together with the use of a channel equalizer, allows to obtain a PLC channel model and compensate the unwanted effects introduced by the channel. Nevertheless, this makes the system more complex and it requires a higher computational load, especially in the reception stage where the channel estimation and equalization are made.

This thesis presents the design of heterogeneous architectures for broadband PLC communications. Therefore an analysis is carried out about the medium access techniques to be used, as well as the channel estimation and equalization techniques. This analysis describes mathematically each one of the stages that compose the FBMC system, both the transmission and reception stages. Among the channel estimation methods, we have selected those with the lowest computational load, such as estimators based on least squares (LS). On the other hand, among the equalization techniques for multi-carrier systems based on filter banks, we can find the ASCET equalizers, which present an FIR filter architecture to carry out equalization. This equalizer requires that in the reception stage the filter bank system is duplicated, having in parallel both, one for the modulated cosine (CMFB) and the other for the modulated sine (SMFB).

FPGA devices can support the computational load of the system. In addition, some SoC integrate an ARM processor together with the FPGA, as the heterogeneous architecture proposed for the devices of the Xilinx Zynq® family. Within the architecture, those parts with higher timing requirements will be implemented in the programmable logic of the device; on the other hand, the parts that present more relaxed timing restrictions will be coded so that they are run in the processor, thus defining the mixed architecture HW/SW.

Each one of the blocks, that compose the hardware part of the architecture, will be modelled using high-level synthesis tools (HLS). On the other hand, in order to obtain an efficient consumption of hardware resources, a datapath study is carried out to adjust the word width to the DSP blocks of the device.

In relation to the software parts of the architecture, the acceleration methods available in the device will be described. It should be noted that the unit vector NEON is used to accelerate certain complex calculations, such as FFT. In addition, the software implementation is extended to use all available processor cores for further acceleration. This is done for Zynq® 7000 devices with one and two cores, and for Zynq® UltraScale+ devices with one, two and four cores, establishing inter-core communication mechanisms for both.

Keywords: Power-Line Communications, Filter Bank Multi-Carrier, Heterogeneous Architecture, Multi-Processor System-on-Chip, ASCET Equalizer.

Índice general

Resumen **xv**

Abstract **xvii**

Índice general **xix**

Índice de figuras **xxiii**

Índice de tablas **xxvii**

Lista de acrónimos **xxxii**

Lista de símbolos **xxxii**

1 Introducción **1**

1.1 Contexto de la tesis 3

1.2 Organización de la memoria 3

2 Antecedentes y estado del arte **5**

2.1 Introducción 5

2.2 Comunicaciones PLC de banda ancha 6

2.2.1 Técnicas de acceso al medio para PLC 7

2.3 Modelado del canal PLC 10

2.3.1 Técnicas de estimación de canal 12

2.3.2 Técnicas de igualación de canal 13

2.4 Arquitecturas *hardware* propuestas para PLC 15

2.5 Objetivos de la tesis 16

3 Aspectos de la modulación FBMC, estimación e igualación de canal 19

3.1 Introducción . . . 19

3.2 Transmisor FBMC . . . 21

3.2.1 Módulo *Multiplicador θ* . . . 21

3.2.2 Módulo *Transformada Discreta del Coseno* . . . 22

3.2.3 Módulo *Matrices IJ* . . . 24

3.2.4 Módulo *Banco de Filtros Polifásico* . . . 24

3.2.5 Módulo *Multiplicador constante* . . . 25

3.2.6 Módulo *Secuencias para la estimación* . . . 25

3.3 Receptor FBMC . . . 26

3.3.1 Módulo *Banco de Filtros Polifásico* . . . 27

3.3.2 Módulo *Matrices IJ* . . . 28

3.3.3 Módulo *Transformada Discreta del Seno* . . . 28

3.3.4 Módulo de *Estimación de canal* . . . 30

3.3.5 Módulo para la igualación de canal . . . 31

3.3.5.1 Coeficientes para el ecualizador 0-ASCET . . . 32

3.3.5.2 Coeficientes para el ecualizador 1-ASCET . . . 33

3.3.5.3 Coeficientes para el ecualizador 2-ASCET . . . 34

3.4 Subdivisión modular . . . 36

3.4.1 Restricciones del sistema FBMC . . . 36

3.4.2 Propuesta de particionado HW/SW . . . 37

3.4.3 Arquitectura propuesta . . . 38

3.5 Conclusiones parciales . . . 40

4 Definición de aceleradores *hardware* para comunicaciones PLC utilizando HLS 43

4.1 Introducción . . . 43

4.2 Estudio del dimensionado del *datapath* . . . 44

4.2.1 Estudio del error debido al dimensionado . . . 46

4.3 Modelado de aceleradores *hardware* mediante Vivado HLS . . . 50

4.4 Modelado del *Multiplicador θ* y *Constante* . . . 53

4.5 Modelado de la DCT y DST . . . 55

4.5.1 Primera y última etapa de la DCT . . . 56

4.5.2 Etapa FFT . . . 56
4.5.3 Etapa de *Cambio de Orden* . . . 58
4.5.4 Cálculo de la DST . . . 59
4.6 Modelado de *Matrices IJ* . . . 61
4.6.1 Módulo *Matrices IJ* en el transmisor . . . 61
4.6.2 Módulo *Matrices IJ* en el Receptor . . . 62
4.7 Modelado del *Banco de Filtros* . . . 63
4.7.1 Módulo *Banco de Filtros* en el Transmisor . . . 63
4.7.1.1 Módulo *Banco de Filtros* en el Receptor . . . 64
4.8 Modelado del igualador de canal ASCET . . . 65
4.9 Modelado del estimador de canal PLC . . . 66
4.10 Resultados del Transmultiplexor FBMC . . . 67
4.11 Conclusiones parciales . . . 68

5 Definición de los bloques *software* en la arquitectura heterogénea propuesta 71
5.1 Introducción . . . 71
5.2 Impacto de la estimación de canal y ecualización en el sistema . . . 72
5.2.1 Evaluación del efecto de la representación en punto fijo . . . 73
5.2.2 Evaluación del efecto del uso del estimador de canal . . . 77
5.2.3 Evaluación del sistema propuesto . . . 83
5.3 Arquitectura mixta HW/SW propuesta . . . 84
5.3.1 Estrategias de aceleración *software* . . . 89
5.3.1.1 Uso de operadores en punto fijo . . . 89
5.3.1.2 Uso de la Unidad Vectorial NEON . . . 92
5.4 Propuesta *software* mononúcleo . . . 96
5.5 Propuesta *software* multinúcleo . . . 98
5.5.1 Sincronización entre núcleos del procesador . . . 100
5.5.1.1 Sincronización en Zynq® 7000 . . . 100
5.5.1.2 Sincronización en Zynq® UltraScale+ . . . 102
5.5.2 Propuesta de doble núcleo (Zynq® 7000) . . . 106
5.5.3 Propuesta para cuatro núcleos (Zynq® UltraScale+) . . . 109
5.6 Conclusiones parciales . . . 114

6 Conclusiones y líneas futuras **115**

6.1 Conclusiones 115

6.2 Líneas futuras 118

6.3 Publicaciones relacionadas con la tesis 120

6.3.1 Publicaciones en revistas internacionales 120

6.3.2 Publicaciones en congresos internacionales 120

6.3.3 Publicaciones en congresos nacionales 121

Bibliografía **123**

Índice de figuras

2.1 Representación de la forma de onda de las subportadoras de OFDM que conforman el ancho de banda. 7

2.2 Diagrama de bloques de OFDM y FBMC para mostrar las diferencias entre las modulaciones [ElA14]. 8

2.3 Comparación de la respuesta en frecuencia entre OFDM y FBMC [Bul16]. 9

2.4 Reparto de los símbolos piloto entre las subportadoras: (a) tipo bloque, (b) tipo peine o (c) tipo inclinado [Tun13]. 12

2.5 Estructura de la ecualización ASCET para un sistema FBMC [Izu07]. . . . 14

2.6 Diagrama de bloques generalizado para el ecualizador L-ASCET [Izu07]. . 15

3.1 Diagrama de bloques del transmultiplexor FBMC, incluyendo el estimador y ecualizador de canal ASCET. 20

3.2 Diagrama de bloques del transmisor FBMC con el módulo *Secuencias estimación*, para la estimación de canal. 21

3.3 Diagrama de bloques del módulo DCT4e dividido en cuatro etapas. 22

3.4 Diagrama de bloques del módulo *Matrices IJ*. 24

3.5 Esquema de la trama de paquetes de sincronismo, estimación de canal y datos por el canal. 26

3.6 Diagrama de bloques del receptor FBMC, incluyendo la etapa de estimación y ecualización de canal, así como el cálculo de coeficientes para el ecualizador ASCET. 27

3.7 Diagrama de bloques del módulo *Matrices IJ* del receptor FBMC, el cual genera las salidas $\hat{p}_{C_m}[n]$ y $\hat{p}_{D_m}[n]$, correspondientes al CMFB y SMFB. . . 28

3.8 Diagrama de bloques del cálculo de la DST4e a partir de la DCT4e. 29

3.9 Esquema de la arquitectura heterogénea propuesta para el sistema FBMC, junto con la etapa de estimación y ecualización de canal. 39

4.1 Diagrama de bloques simplificado de los bloques DSP48e1 de Zynq® 7000. [Xil18a] 45

4.2 Valores de amplitud de una señal de entrada aleatoria en cada subportadora para tres instantes de tiempo diferentes, representados en punto flotante y punto fijo. 48

4.3 Error debido a la cuantificación en punto fijo de la señal de entrada en cada subportadora para tres instantes de tiempo diferentes. 48

4.4 Diagrama de flujo para el modelado mediante la herramienta Vivado HLS [Viv16, Figura 4]. 50

4.5 Arquitectura del módulo *Multiplicador* θ para un $R_P = 1$ en (a) y un $R_P = 2$ en (b). 53

4.6 Arquitectura eficiente para la implementación de multiplicaciones complejas [Jac05]. 56

4.7 Diagrama de mariposa para el cálculo de la FFT de ocho puntos. 57

4.8 Diagrama para el cálculo de la FFT de ocho puntos con cruces regulares. . 57

4.9 Desenredado de las etapas de la FFT para un número de puntos $M = 512$. 58

4.10 Arquitectura implementada para la etapa de reordenamiento de la DCT4e. 58

4.11 Arquitectura de la etapa previa de acondicionamiento para el cálculo de la DST4e, a partir de la DCT4e. 60

4.12 Arquitectura de la etapa posterior de acondicionamiento para el cálculo de la DST4e, a partir de la DCT4e. 60

4.13 Arquitectura implementada para el módulo *Matrices IJ* del transmisor. . . 61

4.14 Arquitectura implementada para el módulo *Matrices IJ* del receptor. . . . 62

4.15 Arquitectura para la implementación del módulo *Banco de Filtros* en el transmisor. 64

4.16 Arquitectura para la implementación del módulo *Banco de Filtros* en el receptor. 65

4.17 Estructura del igualador de canal L-ASCET basada en filtros FIR. 66

4.18 Arquitectura del módulo *Secuencias Estimación*. 67

4.19 Esquema de la arquitectura heterogénea propuesta para el sistema FBMC, junto con el módulo de estimación y ecualización de canal. 69

5.1 Diagrama de bloques del sistema simplificado con el esquema de modulación 4-PAM, donde se marcan las señales para el cálculo de la SNR y BER. 73

5.2 Respuesta en frecuencia a lo largo del tiempo de los modelos de canal PLC utilizados. A la izquierda el modelo A del canal PLC y a la derecha el modelo B del canal PLC. 74

5.3 Estructura de la trama intercambiada por el transmisor FBMC y receptor FBMC, donde se muestra el preámbulo (S_x y C_x) y las diferentes tramas de datos (D_x). 75

5.4 Comparación entre la señal de entrada $V_m[k]$, la señal de salida $\hat{V}_m[k]$ en punto flotante de doble y simple precisión, y en punto fijo. 76

5.5 Comparación entre la señal de entrada $V_m[k]$, la señal de salida $\hat{V}_m[k]$ del banco de filtros sin ecualizar y la señal ecualizada. 78

5.6 Diagrama de bloques de la arquitectura HW/SW propuesta para el sistema FBMC. 80

5.7 Diagrama de el flujo de datos en la etapa de recepción. 80

5.8 Diagrama de los escenarios a evaluar para comprobar el efecto de la estimación de canal y cálculo de coeficientes en el rendimiento del sistema. . . 81

5.9 BER para cada ecualizador L-ASCET en presencia del modelo de canal A, asumiendo el sistema ideal en punto flotante de doble precisión (Ideal) y la propuesta realizada en punto fijo (Propuesta). 85

5.10 BER para cada ecualizador L-ASCET en presencia del modelo de canal B, asumiendo el sistema ideal en punto flotante de doble precisión (Ideal) y la propuesta realizada en punto fijo (Propuesta). 86

5.11 Diagrama de bloques de la arquitectura HW/SW propuesta para el sistema FBMC. 88

5.12 Diagrama de bloques de la arquitectura de Zynq® 7000, donde se muestran los elementos que componen el SoC [Xil18d]. 90

5.13 Diagrama de bloques de la arquitectura de Zynq® UltraScale+, donde se muestran los elementos que componen el SoC [Xil19c]. 91

5.14 Tiempo medio de ejecución para el cálculo de la FFT utilizando Zynq® 7000 y Zynq® UltraScale+. 95

5.15 Gráfica con la aceleración obtenida mediante el uso de la unidad vectorial NEON en el cálculo de la FFT utilizando Zynq® 7000 y Zynq® UltraScale+. 95

5.16 Tiempo de procesamiento para la codificación de la estimación de canal y cálculo de coeficientes desglosado por las etapas del algoritmo utilizando Zynq® 7000. 97

5.17 Diagrama de bloques del algoritmo de estimación de canal y cálculo de coeficientes, dependiendo del orden L del ecualizador ASCET. 99

5.18 Diagrama del cálculo de la FFT de N puntos a partir de dos FFTs de $N/2$ puntos. 100

5.19 Diagrama que ilustra la gestión de las interrupciones en un sistema multiprocesador para dos núcleos. 101

5.20 Diagrama que ilustra la gestión de las variables de exclusión mutua (*mutex*) en un sistema multiprocesador para dos núcleos. 102

5.21 Diagrama de la arquitectura de la OCM de Zynq® UltraScale+ [Xil19c, Figura 18-1] 103

5.22 Arquitectura del intercambio de mensajes con IPI en Zynq® UltraScale+ [Xil19c, Figura 13-6]. 105

5.23 Diagrama del método de interrupción entre emisor y receptor utilizando IPI [Xil19c, Figura 13-4]. 106

5.24 Diagrama de bloques del algoritmo de estimación de canal y cálculo de coeficientes para la versión de doble núcleo con los puntos de sincronización en verde. 106

5.25 Línea temporal y pseudocódigo de la sincronización entre los núcleos mediante el uso de la gestión de las interrupciones. 107

5.26 Línea temporal y pseudocódigo de la sincronización entre los núcleos mediante el uso de instrucciones atómicas y variables de exclusión mutua. . . 108

5.27 Tiempos de ejecución para los diferentes órdenes del ecualizador, desglosados en función de las tareas del algoritmo, así como los dos métodos de sincronización: interrupciones y variable de exclusión mutua. 108

5.28 Tiempos de ejecución para los diferentes órdenes del ecualizador en las propuestas de mononúcleo (SC) y doble núcleo (DC) con sincronización mediante interrupciones. 109

5.29 Diagrama de bloques del algoritmo de estimación de canal y cálculo de coeficientes para la versión de cuatro núcleos con los puntos de sincronización indicados. 111

5.30 Diagrama y pseudocódigo de la gestión de la interrupción mediante IPI. . . 112

5.31 Tiempos de ejecución para los diferentes órdenes del ecualizador en las propuestas de mononúcleo, doble núcleo y cuádruple núcleo con sincronización mediante IPI en Zynq® UltraScale+. 112

Índice de tablas

2.1 Resumen con las características principales de las arquitecturas recogidas en el apartado 2.4. 16

4.1 Tabla de los valores mínimos y máximos de los coeficientes para cada uno de los módulos del sistema FBMC. 47

4.2 Tabla de los valores mínimos y máximos de amplitud para los diferentes niveles de modulación PAM del estándar IEEE 1901/2010 para PLC. . . . 47

4.3 Tabla con el dimensionado del *datapath* utilizado tras el estudio realizado. . 49

4.4 Tabla de latencia, `II` y consumo de recursos de los módulos *Multiplicador* θ y *Multiplicador Constante*. 55

4.5 Tabla de latencia, `II` y consumo de recursos de los módulos involucrados en el cálculo de la DCT4e. 59

4.6 Tabla de latencia, `II` y consumo de recursos de los módulos involucrados en la adaptación de la DST4e. 60

4.7 Tabla de latencia, `II` y consumo de recursos de los módulos *Matrices IJ* de la etapa de transmisión y recepción. 63

4.8 Tabla de latencia, `II` y consumo de recursos del módulo *Banco de Filtros* del transmisor. 64

4.9 Tabla de latencia, `II` y consumo de recursos del módulo *Banco de Filtros* del receptor. 65

4.10 Tabla de latencia, `II` y consumo de recursos del módulo de ecualizador PSE para los diferentes órdenes de *L*-ASCET. 66

4.11 Tabla del consumo de recursos del módulo *Secuencias estimación*. 67

4.12 Tabla de latencia, `II` y consumo de recursos de los módulos de transmisión y recepción implementado en Vivado HLS. 68

4.13 Tabla de latencia, `II` y consumo de recursos para la etapa de transmisión y recepción. 68

5.1 Resultados de SNR y RMSE obtenidos de la simulación del sistema para los diferentes órdenes de ecualizador L-ASCET en presencia de canal ideal y modelos A y B de canal PLC, para los diferentes métodos de representación. 75

5.2 Resultados de SNR y RMSE obtenidos de la simulación del sistema para los diferentes órdenes de ecualizador L-ASCET y casos presentados, realizando la estimación de canal y cálculo de coeficientes cumpliendo $t_{Coef} \leq t_{RX}$. . . 82

5.3 Resultados de SNR y RMSE obtenidos de la simulación del sistema para los diferentes órdenes de ecualizador L-ASCET y casos presentados, realizando la estimación de canal y cálculo de coeficientes en el caso que no se cumple $t_{Coef} \leq t_{RX}$. 83

5.4 Resultados de SNR y RMSE obtenidos de la simulación del sistema completo para los diferentes órdenes de ecualizador L-ASCET. 84

5.5 Resultados de los tiempos de ejecución para el cálculo de la FFT mediante el uso de la UV NEON y la codificación en C (C-Code). 94

5.6 Tiempos de procesamiento determinados para los tres tipos de ecualizadores L-ASCET implementados en el sistema PS. 96

5.7 Análisis de la complejidad de la propuesta realizada, incluyendo los módulos transmisor y receptor. 98

5.8 Desglose de los tiempos de ejecución del algoritmo de estimación de canal y cálculo de coeficientes obtenidos para los diferentes órdenes de ecualizador con ambos métodos de sincronización. 108

5.9 Resultados de los tiempos de ejecución para el cálculo del algoritmo de estimación de canal y cálculo de coeficientes en Zynq® UltraScale+ con uno, dos y cuatro núcleos. 113

Lista de acrónimos

ACP Accelerator Coherency Port.

ADSL Asymmetric Digital Subscriber Line.

AFE Analog Front End.

ALU Arithmetic Logic Unit.

AM Amplitude Modulation.

ASCET Adaptive Sine-Modulated/Cosine-Modulated Filter Bank Equalizer for Trans-multiplexers.

AWGN Additive White Gaussian Noise.

BB-PLC Broadband-PLC.

BER Bit Error Rate.

BRAM Block Random Access Memory.

CDMA Code Division Multiple Access.

CMFB Cosine Modulated Filter Bank.

CP Cylic Prefix.

CSS Complementary Sets of Sequences.

DCT Discrete Cosine Transform.

DCT4e Type-IV Even Discrete Cosine Transform.

DDR Double Data Rate.

DMA Direct Memory Access.

DSL Digital Subscriber Line.

DSP Digital Signal Proccesors.

DST Discrete Sine Transform.

DST4e Type-IV Even Discrete Sine Transform.

DTT Discrete Trigonometric Transform.

EMC Electromagnetic Compatibility.

FBMC Filter Bank Multiple Carrier.

FF Flip Flop.

FFT Fast Fourier Transform.

FIR Finite Impulse Response.

FM Frequency Modulation.

FPGA Field-Programmable Gate Array.

GIC Generic Interrupt Controller.

GPU Graphic Process Unit.

HLS High Level Shynthesis.

HW Hardware.

I-UWB Impulsive-UltraWideBand.

ICI Inter-Carrier Interference.

IEC International Electrotechnical Commission.

IEEE Institute of Electrical and Electronics Engineers.

IFFT Inverse Fast Fourier Transform.

IIR Infinite Impulse Response.

IoT Internet of Things.

IPI Inter-Processor Interrupts.

ISI Inter-Symbol Interference.

ITU-T ITU-Telecommunication Standardization Sector.

LMMSE Linear Minimum Mean Square Error.

LS Least Squares.

LTE Long Term Evolution.

LTI Linear Time-Invariant.

LTV Linear Time Variant.

LUT LookUp Table.

MIMO Multiple Input Multiple Output.

MLSE Maximum-Likelihood Sequence Estimation.

MMSE Minimum Mean Square Error.

MPSoC Multi-Processor System-on-Chip.

NB-PLC Narrowband-PLC.

OCM On Chip Memory.

OFDM Orthogonal Frequency Division Multiplexing.

PAM Pulse Amplitude Modulation.

PL Progammable Logic.

PLC Power-Line Communications.

PMU Power Management Unit.

PS Processing System.

PSE Per-Subcarrier Equalizer.

QAM Quadrature Amplitude Modulation.

QPSK Quadrature Phase-Shift Keying.

RAM Randon Access Memory.

RMSE Root-Mean-Square Error.

RPU Real-Time Processing Unit.

RTL Register-Transfer Level.

SIMD Single Instruction Multiple Data.

SMFB Sine Modulated Filter Bank.

SNR Signal-Noise Ratio.

SoC System-on-Chip.

SW Software.

UMTS Universal Mobile Telecommunications System.

UV Unidad Vectorial.

VHDL VHDL.

WiMAX Worldwide Interoperability for Microwave Access.

WLAN Wireless Local Area Network.

Capítulo 1

Introducción

Cincuenta años han pasado desde el primer envío de paquetes dentro de la red de computadoras de ARPANET en 1969. Este hito marcó el inicio de Internet, aunque no fue hasta la década de los noventa, cuando se establecieron las bases de la World Wide Web (WWW), es decir, una red informática global. El crecimiento de la red fue asombroso, en el transcurso de cinco años había diez millones de computadoras conectadas en el mundo. La aparición del correo electrónico, los motores de búsqueda o el comercio electrónico evolucionaron rápidamente desde ese momento. La transformación tecnológica permitió que ciertos servicios se crearan o se adaptaran, impulsados por la aparición de los teléfonos inteligentes y la evolución en las comunicaciones inalámbricas.

Hoy en día el mundo vive conectado y la gran mayoría tiene acceso a Internet desde su ordenador, teléfono inteligente, tablet o televisor. La forma de conectarse a Internet ha ido cambiando y se ha ido adaptando con el tiempo; desde la aparición de los primeros módems telefónicos, pasando por las tecnologías xDSL junto con la variedad de tecnologías inalámbricas (WiFi, WiMAX, GSM o 4G), hasta llegar a los accesos de alta velocidad de fibra óptica o cable.

Esta evolución, en la mayoría de los casos, ha tratado de utilizar la infraestructura existente con el fin de ofrecer la mayor cobertura posible. En el caso de los módems telefónicos o la tecnología xDSL, aprovecharon el despliegue del par de telefonía utilizado para voz para proporcionar servicio de Internet. Con esta misma filosofía, la comunicación a través de la red eléctrica (PLC, *Power-Line Communications*) pretende hacer uso de la infraestructura desplegada para la transmisión de datos a la vez que se suministra energía.

Las compañías eléctricas emplean las comunicaciones PLC para controlar estaciones de abastecimiento de forma remota allí donde la red de telefonía no llega. Estos enlaces tienen velocidades pequeñas (alrededor de 2 Mbit/s) pero suficientes para el intercambio de mensajes. Las compañías han tratado de explotar el potencial de PLC realizando pruebas en numerosos países, con el fin de proporcionar acceso a Internet en zonas rurales, aprovechando la infraestructura ya existente.

Sin embargo, el medio de comunicación PLC es bastante hostil ya que presenta ruido, notables interferencias y cambios repentinos. Una forma de combatir estos efectos molestos en la comunicación es la utilización de modulaciones multiportadora, las cuales dividen el ancho de banda en subportadoras, potenciando de esta forma las frecuencias con menor nivel de ruido y anulando aquellas con mayor, con la consiguiente mejora en la comunicación punto a punto.

En el mercado se pueden encontrar dispositivos PLC para establecer una red en el ámbito doméstico, donde los niveles de ruido y atenuación son menores, proporcionando acceso a Internet en cualquier punto del hogar. No obstante, como le ocurrió a la tecnología xDSL, las comunicaciones PLC deben afrontar el reto de establecer los mecanismos necesarios para transmitir datos por la red eléctrica, diseñada para el transporte de energía. Además, con el incremento de los servicios de Internet deberá ser capaz de proporcionar enlaces de banda ancha y definir los mecanismos necesarios para otorgar robustez frente al ruido y la interferencia.

En la última década se ha presenciado el auge de las redes inteligentes de energía (*Smart Grids*), las cuales pretenden realizar una transmisión eficiente de la electricidad. Esto requiere que los sistemas de energía se comuniquen para soportar la demanda en los momentos que sea necesario, y han considerado las comunicaciones PLC como un medio de comunicación útil para este aspecto. Por otro lado, el Internet de las cosas (IoT, *Internet of Things*) ha visto en las comunicaciones PLC la posibilidad de suministrar energía a los dispositivos al mismo tiempo que permite el envío de información por la red eléctrica. La comunicación entre sensores y actuadores desplegados en diferentes puntos de un vehículo, avión o barco, es posible realizarla mediante PLC. De este modo se reduce la infraestructura necesaria aportando mejoras como la reducción en el peso del conjunto.

El ruido y la interferencia se puede combatir con mecanismos que mejoren la señal recibida. Para ello, es necesario tener un modelo del canal PLC para contrarrestar el ruido y las interferencias. Sin embargo, la variedad de entornos y topologías que presenta la red eléctrica hace que sea difícil establecer un modelo de canal que se adapte a los diferentes casos. Además, dada la variabilidad que presenta el medio de comunicación no es posible establecer un modelo estático del canal. Por esta razón, se suelen utilizar técnicas de estimación de canal, que proporcionan un modelo del canal en ese instante. A partir de esta estimación, se puede realizar la ecualización de las tramas en la etapa de recepción.

Esta técnicas requieren utilizar algoritmos de procesamiento con una elevada carga computacional, por ese motivo las soluciones *System-on-Chip* (SoC) son idóneas para este tipo de aplicaciones. Los SoC integran procesadores, periféricos, sistemas de memoria o lógica programable (FPGA, Field-Programmable Gate Array) en una misma pastilla. Son dispositivos más complejos, sin embargo la combinación con la FPGA otorga una gran versatilidad debido a la reconfiguración de este tipo de dispositivos, permitiendo a su vez

aplicar paralelismo a la arquitectura. Por otro lado, el uso de procesadores multinúcleo proporciona también cierto grado de paralelismo en el SoC.

El objetivo de esta tesis es realizar el diseño de una arquitectura SoC heterogénea que integre las técnicas de acceso al medio para las comunicaciones a través de la red eléctrica. En primer lugar, se pretende realizar la implementación de la arquitectura de un transmultiplexor para comunicaciones PLC mediante el uso de herramientas de síntesis de alto nivel. En segundo lugar, se trata de dotar a la arquitectura de las técnicas de estimación y ecualización de canal que permitan compensar aquellos efectos no deseados introducidos por el canal PLC, mejorando la comunicación punto a punto. Con todo esto, se evalúa la posibilidad de establecer una arquitectura mixta HW/SW, haciendo un uso eficiente de los elementos disponibles en el SoC. En todo momento se tratan de alcanzar las restricciones temporales impuestas por los estándares de comunicación PLC.

1.1 Contexto de la tesis

Este trabajo se ha desarrollado dentro del Grupo de Investigación GEINTRA (Grupo de Ingeniería Electrónica Aplicada a Espacios Inteligentes y Transporte). La temática de la tesis se enmarca dentro de la investigación del proyecto "Arquitecturas eficientes en SoCs para tecnologías de capa física fiables en comunicaciones sobre redes eléctricas (SOC-PLC)", 2016-2018. Dicho proyecto ha sido financiado por el Ministerio de Economía y Competitividad (ref. TEC2015-64835-C3-2-R).

Este proyecto, del que también forman parte el grupo de Teoría de la Señal de la Universidad de Alcalá y de la Universidad Politécnica de Madrid, tiene como objetivo establecer técnicas de acceso al medio para comunicaciones de banda ancha sobre PLC. Uno de los paquetes de trabajo del proyecto trata la definición de arquitecturas eficientes y su implementación en tiempo real sobre sistemas SoC basados en FPGAs. Estas arquitecturas implementarán las técnicas de acceso al medio, estimación e igualación de canal que serán propuestas dentro del grupo de trabajo del proyecto. La solución alcanzada debe mantener una relación de compromiso entre el consumo de recursos y el rendimiento global del sistema.

1.2 Organización de la memoria

Esta memoria se organiza en cinco grandes capítulos, cada uno de ellos se va a describir brevemente a continuación.

- Capítulo 2: "Antecedentes y estado del arte". Se realiza un análisis de las técnicas de acceso al medio para comunicaciones PLC de banda ancha, evaluando las diferentes alternativas propuestas en los estándares de PLC. Este estudio evalúa los modelos de

canal PLC que se proponen en estudios previos, con el objetivo de establecer las bases para utilizar las técnicas de estimación y ecualización de canal adecuadas. También se evalúan las propuestas desarrolladas para PLC en dispositivos FPGA o SoC. Por último, se detallan en mayor profundidad los objetivos que se pretenden alcanzar en la tesis.

- Capítulo 3: "Aspectos de la modulación FBMC, estimación e igualación de canal". Este capítulo realiza la descripción teórica del sistema FBMC, desglosando cada una de las partes que engloba al módulo transmisor y al receptor. Además, se describen los métodos de estimación de canal y ecualización que se van a utilizar. Por último, se realiza un repaso de las restricciones que presenta el sistema FBMC con el fin de plantear la propuesta de una arquitectura heterogénea.

- Capítulo 4: "Definición de aceleradores *hardware* para comunicaciones PLC utilizando HLS". En este capítulo se detalla el modelado *hardware* de la arquitectura propuesta. Para llevar a cabo dicha propuesta se han utilizado las herramientas de síntesis de alto nivel (HLS). Por otro lado, se expone el estudio del *datapath* que se ha llevado a cabo. Los resultados se muestran para cada una de las etapas del sistema FBMC en términos de consumo de recursos, latencia o intervalo de inicialización.

- Capítulo 5: "Definición de los bloques *software* en la arquitectura heterogénea propuesta". Este capítulo muestra el impacto que tiene la estimación de canal y ecualización en las prestaciones del sistema. Se establecen los métodos de aceleración *software* disponibles en la implementación de una arquitectura heterogénea. Por último, este capítulo presenta los resultados obtenidos en la implementación software realizada con uno, dos y cuatro núcleos. Cabe destacar que se lleva a cabo sobre los dispositivos Zynq-7000 y Zynq-UltraScale+, detallando los mecanismos de sincronización entre núcleos.

- Capítulo 6: "Conclusiones y líneas futuras". En este último capítulo, se recogen las conclusiones obtenidas en el desarrollo de la tesis, además se enumeran las publicaciones derivadas de la misma. Para finalizar, se exponen las posibles líneas de investigación para abordar en trabajos futuros.

Capítulo 2

Antecedentes y estado del arte

2.1 Introducción

Como se ha comentado en el capítulo anterior, las comunicaciones PLC han tenido un gran impulso en los últimos años debido al auge de las redes inteligentes de distribución de energía (*Smart Grids*) [Gun11, Gal11, Kha16], el Internet de las cosas (*Internet of Things*, IoT) [Xu14, Bas18, Zhi17], las comunicaciones entre diferentes sistemas dentro del sector automovilístico [Gue16, Pin15, Zha17, Liu13], del sector aeronáutico [Bri17, Lar14] o naval [Ant11, Pan18]. Además, el auge de nuevas tecnologías, tales como el vehículo eléctrico [Pit16, Bas09, Obe14], formas de auto-abastecimiento en el hogar [Pen14, Tur12] o sistemas de medición inteligentes [Oks06, Yin19, Cho08, Puk17], han hecho que en la última década se hayan aprobado una serie de estándares desde organismos como la ITU-T (ITU-Telecommunication Standardization Sector) [ITU12a, ITU12b, ITU17a, ITU17b, ITU18a, ITU18b], el IEC (*International Electrotechnical Commission*) [Com01] o el IEEE (*Institute of Electrical and Electronics Engineers*) [IEE10, IEE13b, IEE18] para las Comunicaciones a través de la Red Eléctrica (PLC).

Entre los estándares disponibles hay que distinguir los sistemas de banda estrecha (NB-PLC, *Narrowband-PLC*) [Gal15] y los sistemas de banda ancha (BB-PLC, *Broadband-PLC*) [Yin19]; la principal diferencia radica en la banda de transmisión que se sitúa entre 3 kHz-500 kHz para NB-PLC y entre 2 MHz-86 MHz para BB-PLC, además la tasa de transmisión para BB-PLC es hasta 20 mayor que la de NB-PLC, siendo de 84 MSymb/s y 0.497 MSymb/s respectivamente [Zei17]. A la vista de estas diferencias, BB-PLC es utilizado en sistemas con una tasa de transmisión elevada, aunque tiene el inconveniente que, al utilizar frecuencias superiores, la complejidad de implementación aumente, así como las restricciones de compatibilidad electromagnética (EMC, *Electromagnetic Compatibility*) [IEE95, Ber14, IEE11].

Otro aspecto que dificulta su despliegue e interés son las características del medio o canal de comunicación [Lag08]. A grandes rasgos, el canal presenta selectividad en frecuencia que enfatiza la propagación multicamino y variaciones temporales dificultando

su modelado [Lak18]. A esto hay que añadir el ruido, el cual se presenta con fuertes componentes estacionarias, cicloestacionarias y no estacionarias, siendo la principal fuente de interferencia debida a los sistemas de radiofrecuencia, a otros nodos PLC y a líneas DSL [Gal16, Ker15]. Estas características están presentes tanto en NB-PLC, como en BB-PLC, aunque con ciertas diferencias.

Actualmente lo que predomina en las comunicaciones PLC es el uso de modulaciones multiportadora, como OFDM (Orthogonal Frequency Division Multiplexing), para hacer frente a las exigencias del canal PLC y alcanzar las tasas de transmisión. Sin embargo, existe la posibilidad de mejorar el enlace de la capa física (*PHY layer*) mediante el uso de modulaciones basadas en bancos de filtros [Pou16, Bal18] y modulación impulsiva de banda ultra-ancha (I-UWB, *Impulsive-UltraWideBand*) [Ton13b]. A continuación se realizará un repaso del estado del arte y antecedentes a este trabajo, proporcionando una visión global de la evolución que ha presentado esta tecnología. En primer lugar, se hará un repaso de las comunicaciones de banda ancha y diferentes técnicas de acceso al medio que se encuentran disponibles; se continuará con el modelado de los canales PLC, detallando en este caso las diferentes técnicas de estimación y ecualización de canal que se pueden aplicar. Un último punto que se va a explorar en este capítulo son las implementaciones de arquitecturas eficientes en tiempo real para sistemas PLC, así como la tendencia de los dispositivos de *hardware* programable. Al final del capítulo se muestran los objetivos generales y parciales que se pretenden abordar en la tesis.

2.2 Comunicaciones PLC de banda ancha

Para comunicaciones con una baja tasa de datos se han utilizado las comunicaciones PLC en banda estrecha, como por ejemplo para el envío de mediciones remotas [Puk17, Raf12, Lee11]. Sin embargo, el reciente crecimiento de sistemas multimedia, sistemas conectados en red (*Smart Grids*, IoT, ...) y aplicaciones en vehículos (aviones, coches, barcos, ...), ha provocado que tenga especial interés para el campo de la investigación los sistemas de comunicación de banda ancha. En este sentido, las comunicaciones PLC han despertado interés en la industria o en la investigación, debido principalmente a que su infraestructura está desplegada, y además es más extensa que otras infraestructuras de telecomunicación [Gun11].

No obstante, los mayores esfuerzos en investigación se han centrado en la capa física, debido a la complejidad que presentan los sistemas de la red eléctrica y los retos que propone el medio de comunicación. Además, hay que tener en cuenta las restricciones regulatorias que tienen estos sistemas relacionados con la compatibilidad electromagnética [IEE11]. Sin embargo, la aparición de los estándares y propuestas, por parte de instituciones y comunidades [Men17, IEE10, ISO15], ha hecho posible la adaptación o aparición de técnicas de acceso al medio que mejoran los sistemas PLC a nivel de capa física, como el uso de modulaciones multiportadora [Com01, IEE13b].

En esta línea, en [Oli18, Alo17] se presenta un sistema de comunicación híbrido, el cual utiliza las comunicaciones PLC y las comunicaciones WiFi para mejorar el rendimiento de la capa física. A continuación se detallan diferentes técnicas de acceso al medio utilizadas en comunicaciones PLC.

2.2.1 Técnicas de acceso al medio para PLC

Si se observan otros sistemas de comunicación de banda ancha, cableado o inalámbrico, como DSL, WiFi, WiMAX o la telefonía móvil 4G LTE, se puede comprobar que en todos ellos hacen uso de técnicas de acceso al medio como Multiplexación por División de Frecuencias Ortogonales (OFDM, *Orthogonal Frequency Division Multiplexing*).

OFDM realiza una división del ancho de banda disponible en múltiples subportadoras que son las encargadas de transportar la información. En la figura 2.1 se muestra un ejemplo de señal OFDM donde se identifican las diferentes suportadoras que componen el ancho de banda. Para evitar la interferencia entre portadoras (ICI, *Inter-Carrier Interference*), OFDM utiliza la ortogonalidad entre las subportadoras, permitiendo además que puedan solaparse, evitando así tener que introducir bandas de guarda entre subportadoras.

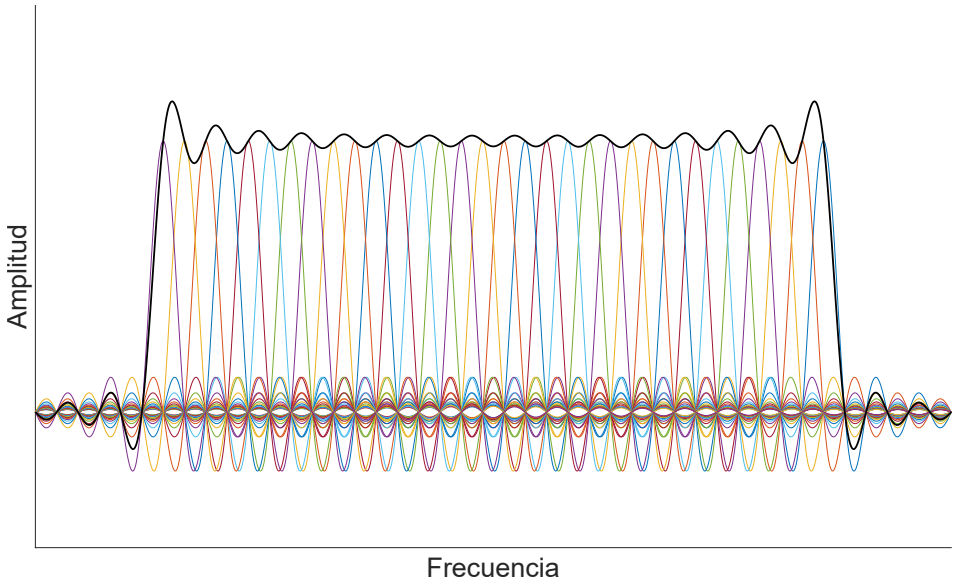


Figura 2.1: Representación de la forma de onda de las subportadoras de OFDM que conforman el ancho de banda.

Por otro lado, OFDM soporta muy bien los desvanecimientos que pueda sufrir el canal de comunicación, simplificando en cierto modo la complejidad del receptor, que hace uso de métodos de ecualización más sencillos. Además, para evitar la interferencia entre símbolos (ISI, *Inter-Symbol Interference*), OFDM añade un prefijo cíclico (CP, *Cylic Prefix*) a cada símbolo transmitido, que previamente ha sido modulado mediante un esquema QPSK, 16QAM o 64QAM. El uso del CP permite soportar los desvanecimientos que sufre el canal; sin embargo, introduce redundancia en la transmisión, empeorando la eficiencia espectral.

El método de transmisión basado en multiportadora con bancos de filtros (FBMC, *Filter Bank Multiple Carrier*) permite mejorar la comunicación mediante el uso de filtros digitales en lugar de utilizar el prefijo cíclico que usa OFDM [Bal18, Bel10]. En la figura 2.2 se muestra el diagrama de bloques de cada una de las modulaciones para comprobar las diferencias entre OFDM y FBMC.

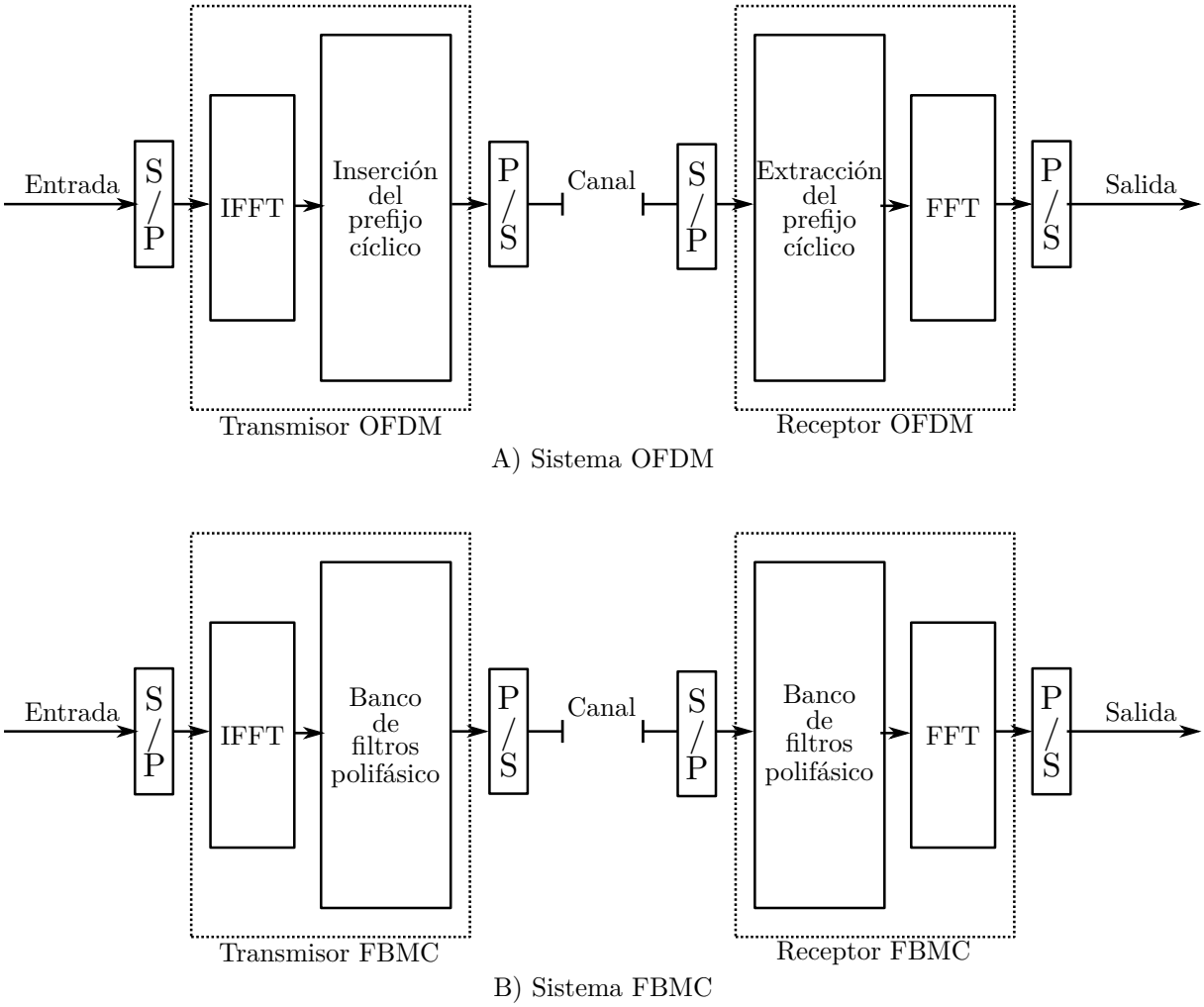


Figura 2.2: Diagrama de bloques de OFDM y FBMC para mostrar las diferencias entre las modulaciones [ElA14].

Como se muestra en la figura 2.3, el uso de FBMC permite mejorar la eficiencia espectral que se perdía con el uso del CP. Además, la inserción de filtros digitales por cada subportadora permite reducir la ICI mediante el uso de filtros prototipos con lóbulos laterales más atenuados.

El uso de sistemas FBMC se está extendiendo en los últimos años por las mejoras que presenta frente a OFDM. Por ejemplo, en la nueva generación de telefonía móvil 5G se propone como modulación candidata frente a OFDM [Far14]. En lo que respecta a las comunicaciones PLC, el estándar IEEE 1901/2010 [IEE10] recomienda el uso de FBMC como técnica de acceso al medio para la comunicación de banda ancha a través de la red eléctrica, proponiendo un filtro prototipo para su desarrollo. Existen propuestas [Cru16]

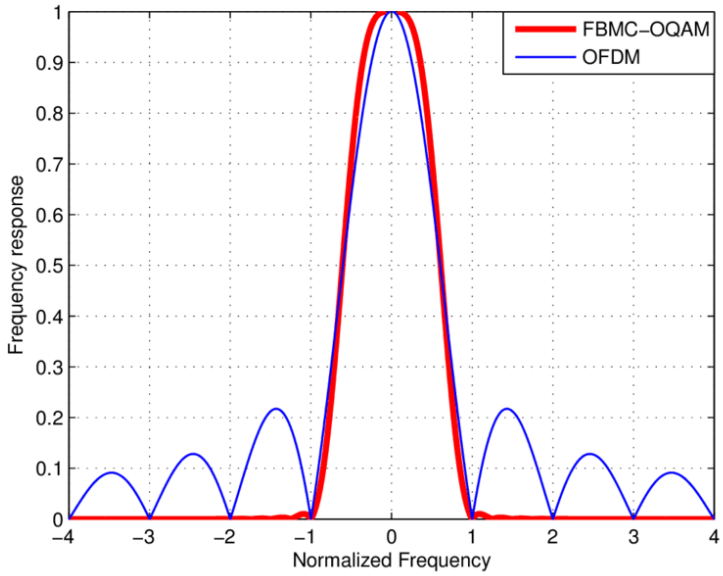


Figura 2.3: Comparación de la respuesta en frecuencia entre OFDM y FBMC [Bul16].

que evalúan este filtro prototipo y aportan alternativas con otros filtros diferentes. En este sentido, varios autores apuntan al uso de la transformada trigonométrica discreta (DTT, *Discrete Trigonometric Transform*) para modulaciones multiportadora, en lugar de utilizar la FFT.

En [Bel10] se estudia y compara el uso de FBMC frente a OFDM en comunicaciones PLC, obteniendo resultados de prestaciones superiores para el caso de FBMC, con mayores velocidades binarias y robustez frente al ruido. Sin embargo, el incremento computacional que aporta FBMC es notable y superior a OFDM. Esto se hace evidente en la etapa de recepción, ya que al no utilizar CP, FBMC requiere de algún método de sincronización entre transmisor y receptor, así como otro método de ecualización alternativo.

Con el fin de mejorar la robustez y mejorar la fiabilidad de las comunicaciones PLC, en [Dib18] se propone el uso de *Hybrid*-PLC, donde se combina la comunicación PLC con la comunicación inalámbrica. Este concepto no se aplica de forma extendida y el número de contribuciones es limitado; sin embargo, en aplicaciones para *Smart Grids* o IoT es una alternativa que permite proporcionar conectividad a dispositivos inalámbricos dentro del área de cobertura del módulo *Hybrid*-PLC. En [Maf15] se puede encontrar una solución para una infraestructura de contadores automatizada donde se considera la utilización de las comunicaciones PLC con otros métodos de comunicación inalámbrica. En el ámbito de las *Smart Grids*, pueden encontrarse diferentes aplicaciones en [Say14]. En el estándar IEEE 1905.1-2013 [IEE13a] se adoptan otros ampliamente acogidos, como WiFi (IEEE 802.11) [IEE12], Ethernet (IEEE 802.3) [IEE07] y HomePlug AV (IEEE 1901) [IEE10], para combinar el rendimiento de cada uno de ellos y mejorar la fiabilidad de las redes cableadas, añadiendo flexibilidad mediante la tecnología inalámbrica.

Otra forma de mejorar la velocidad de transmisión y fiabilidad en la comunicación es

optar por los sistemas con múltiples entradas y múltiples salidas (MIMO, *Multiple Input Multiple Output*), que se utilizan también en sistemas de comunicaciones móviles como UMTS, LTE, WiMAX o WiFi [Ber15]. Existen recomendaciones y estándares que regulan los sistemas MIMO en PLC, como la ITU-T [ITU18c].

Estas últimas tendencias mencionadas (*Hybrid*-PLC y MIMO-PLC) están menos desarrolladas y las contribuciones son menores. No obstante, se trata de las tendencias más actuales en las comunicaciones PLC. Algo que influye, y es común a todas las propuestas mostradas, es que requieren de un modelado y estudio del canal PLC para una mejor transmisión.

2.3 Modelado del canal PLC

Como se ha visto, las comunicaciones PLC se presentan como un medio económico para la transmisión de datos. El estándar IEEE 1901/2010 regula la forma de acceder al medio de comunicación. Sin embargo, para las comunicaciones PLC, el medio de comunicación puede variar enormemente y presentar gran variabilidad entre entornos. Pensando en el tipo de variaciones que pueda presentar el canal en un entorno industrial, frente a las que se pueden encontrar en un hogar, se puede decir que las diferencias son notables entre ellos, debido a que las máquinas eléctricas usadas en la industria introducen un mayor nivel de ruido y atenuación al canal. Del mismo modo ocurre si se compara el canal de comunicaciones de un coche frente al que presenta un avión o barco. De este modo, las diferentes contribuciones previas evaluadas realizan el modelado de canales PLC en un determinado entorno, ya sea en vehículos, aviones, en el entorno del hogar o la industria. Además, influyen otros aspectos como la topología de la red, la longitud de los enlaces, el tipo de cableado o el número de elementos conectados, entre otros.

A la hora de determinar un modelo, se puede obtener de forma determinista, donde se tiene conocimiento de la topología de la red, cableado, etc., o bien realizarlo de forma empírica efectuando una serie de medidas en el entorno. Es posible obtener una combinación de ambas formas, determinista y empírica. Más allá del entorno y de las características del mismo, entre los diferentes modelos de canal se puede extraer una serie de aspectos comunes a todos ellos. En primer lugar, se debe establecer la función de transferencia que defina el canal de comunicación, y en segundo lugar, es necesario realizar una caracterización del ruido presente en el entorno.

En cuanto a la función de transferencia que define el canal PLC se pueden encontrar una serie de propuestas en [Phi99, Zim99, Zim02b, Zim02a, Mel06, Ana08, And09, Ton12, Sti14, Bri17]. En ellas se describe la función de transferencia que define el canal PLC. Es importante obtener la representación más realista del canal, para ello la función de transferencia debe incluir los desvanecimientos, propagación multicamino, reflexiones, etc. Entre ellas, cabe destacar la propuesta de Zimmermann [Zim02a], la cual considera la propagación multicamino además de los desvanecimientos en frecuencia que

aparecen en la red eléctrica. Otra referencia a considerar son los modelos de canal de Tonello [Ton12, Ton13a, Ton14, Ton15], el cual considera un modelo de propagación determinista compuesto por propagación multicamino, reflexiones debidas a la variedad de impedancias que presentan las terminaciones de red, y longitud de los enlaces. Además, su propuesta permite generar una serie de canales aleatorios con unas características determinadas que modelan la variabilidad que se tiene en estos canales en el ámbito del hogar o la industria.

Por otro lado, es necesario identificar el tipo de ruido presente en el medio de comunicación PLC. En el canal PLC no es posible representar el ruido con el modelo de ruido blanco Gaussiano aditivo (AWGN, *Additive White Gaussian Noise*). En este caso es más complejo, ya que se puede considerar que está compuesto por ruido de fondo coloreado, ruido de banda estrecha y ruido impulsivo. Este último se puede considerar de tres formas: ruido impulsivo aperiódico asíncrono, periódico síncrono, y periódico asíncrono.

El ruido de fondo coloreado se caracteriza por tener una densidad espectral de potencia baja, que aumenta en frecuencias bajas; en el entorno del hogar se pueden asociar a electrodomésticos [Ma10]. El ruido de banda estrecha [Mly16] viene provocado por estaciones de radio y es debido a las modulaciones utilizadas en Amplitude Modulation (AM) y Frequency Modulation (FM) por debajo de los 22 MHz. En cuanto al ruido impulsivo periódico síncrono y asíncrono [Lak18], viene provocado principalmente por fuentes de alimentación de corriente continua o conmutadas, que por lo general provocan repeticiones en múltiplos de la frecuencia de la red en diferentes bandas de frecuencias. Por último, en cuanto al ruido impulsivo aperiódico asíncrono, es el más variable de todos y se modela mediante métodos estadísticos definiendo el intervalo, la amplitud y durabilidad.

Además de las fuentes de ruido comentadas anteriormente, los canales PLC presentan una serie de variaciones temporales debidas al funcionamiento de ciertos dispositivos o conexiones en la red, lo que provoca la intensificación de los niveles de ruido y la variación en las impedancias de la red, originando atenuación por reflexiones o multicaminos. Estos cambios hacen que el canal PLC se considere como un sistema lineal variante en el tiempo (LTV, *Linear Time Variant*); sin embargo, las medidas tomadas en el canal [Can11] permiten considerar el canal como un sistema lineal e invariante en el tiempo (LTI, *Linear Time-Invariant*) durante un periodo de tiempo denominado tiempo de coherencia del canal $t_{C_{PLC}}$. En [Can11, Pic14] se fija el tiempo de coherencia del canal en 600 µs, permitiendo así que el canal se considere estable durante ese intervalo.

Por tanto, la estimación se debe calcular antes de superar el tiempo de coherencia del canal $t_{C_{PLC}}$, para considerar el canal como un sistema LTI y realizar la ecualización de la señal recibida. Puede ocurrir que la estimación de canal obtenida haya superado el tiempo de coherencia del canal $t_{C_{PLC}}$, en tal caso, se puede realizar la ecualización teniendo en cuenta que afectará a las prestaciones del sistema. A continuación se detallan las técnicas de estimación y ecualización de canal utilizadas en comunicaciones PLC.

2.3.1 Técnicas de estimación de canal

A la hora de realizar la estimación del canal hay que tener en cuenta el tipo de respuesta que se desea obtener, ya sea en el dominio del tiempo o en el dominio de la frecuencia. Es posible realizar la transformación de una a otra, mediante el uso de técnicas de transformación de dominio como la FFT o IFFT. Para poder llevar a cabo esta estimación es necesario el envío de símbolos piloto en la trama de transmisión, pudiendo así realizar la estimación de canal en la etapa de recepción. En los sistemas multiportadora, estos símbolos piloto pueden formar parte de un preámbulo inicial a la comunicación, o bien pueden estar repartidos a lo largo del tiempo en diferentes subportadoras. En la figura 2.4 se muestra un ejemplo del reparto de las diferentes formas que se tiene de enviar los símbolos piloto [Tun13, Moy16b].

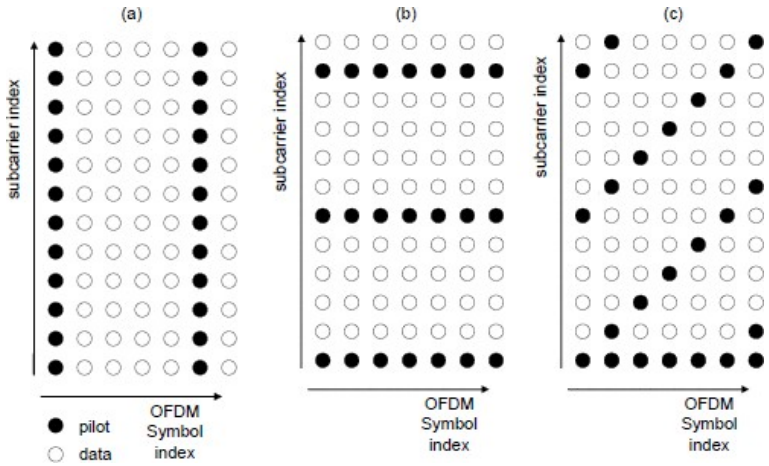


Figura 2.4: Reparto de los símbolos piloto entre las subportadoras: (a) tipo bloque, (b) tipo peine o (c) tipo inclinado [Tun13].

Independientemente de la forma de envío de los símbolos piloto, los métodos de estimación más utilizados en OFDM son el estimador de mínimos cuadrados (LS, *Least Squares*) [Bog14] y el estimador lineal del mínimo error cuadrático medio (LMMSE, *Linear Minimum Mean Square Error*) [Lai19]. Ambos poseen baja complejidad computacional y la principal diferencia entre ellos es que LMMSE utiliza información estadística del canal para llevar a cabo la estimación, mientras que LS no lo hace. Esto requiere de un ligero aumento de la carga computacional para el estimador LMMSE, sin embargo el rendimiento que se obtiene es superior comparado con el caso de estimador LS [Moy16a].

El estimador LS [Bog14] trata de minimizar el error entre la señal transmitida y la señal recibida, de forma que se obtiene la estimación del canal. En (1) se muestra la expresión matemática que tendría este tipo de ecualizador, donde $\mathbf{X}$ representa la señal transmitida e $\mathbf{Y}$ la señal recibida.

$$\hat{\mathbf{H}}_{LS} = \mathbf{X}^{-1}\mathbf{Y} \tag{1}$$

En cuanto al estimador LMMSE [Lai19], de forma simplificada, se puede considerar la salida del estimador Least Squares (LS) a la que se le multiplica por la matriz de auto-correlación de la respuesta en frecuencia del canal [Alo17]. En (2) se muestra la expresión que tendría el estimador LMMSE, donde $\mathbf{A}$ representa la matriz de auto-correlación de la respuesta en frecuencia del canal y $\hat{\mathbf{H}}_{LS}$ la estimación LS. Hay que tener en cuenta que la matriz de auto-correlación de la respuesta en frecuencia del canal depende del canal, del tipo de modulación utilizada o de la relación señal a ruido (SNR), entre otras.

$$\hat{\mathbf{H}}_{MMSE} = \mathbf{A}\hat{\mathbf{H}}_{LS} \tag{2}$$

Independientemente del estimador utilizado, la calidad de la estimación vendrá dada por los símbolos piloto utilizados en la transmisión. Para mejorar la calidad de la estimación se suelen utilizar secuencias con buenas propiedades de auto-correlación. La forma de obtener este tipo de secuencias se puede consultar en [Cai01, Ng98].

En comunicaciones inalámbricas se utiliza una gran variedad de secuencias con determinadas características; en [Vel17] se puede encontrar un resumen de todas ellas. Las secuencias Golay [Gol61], Gold [Gol68] o Kasami [Fen12, Zen06, Gos88], son ejemplos de secuencias que tienen buenas propiedades de auto-correlación. Cabe destacar también los conjuntos complementarios de secuencias [Zol09, Tur63, Che06], los cuales son utilizados ampliamente en sistema CDMA (*Code Division Multiple Access*). Sin embargo, la utilización de secuencias no sólo se lleva a cabo para la estimación de canal, sino también se usan para realizar la sincronización ente emisor y receptor [Nom15].

Para poder utilizar la estimación del canal es necesario considerar que presenta un comportamiento lineal e invariante en el tiempo LTI. El canal PLC presenta bruscas variaciones a lo largo del tiempo y para poder considerar el canal como un sistema LTI no se debe superar el tiempo de coherencia del canal $t_{C_{PLC}}$ [Can11, Ant11]. De esta forma, será necesario repetir la estimación del canal con una periodicidad que vendrá dada por el tiempo $t_{C_{PLC}}$.

2.3.2 Técnicas de igualación de canal

Como se ha mencionado anteriormente, una de las características que presenta FBMC frente a OFDM es la ausencia de CP. El uso del CP le permite a OFDM realizar la ecualización del canal en el dominio de la frecuencia. En el caso de FBMC, ofrece mejor ICI e ISI en medios de comunicación más hostiles que OFDM, sin embargo presenta la desventaja que se requiere realizar la ecualización, debido a la ausencia de CP, aumentando la complejidad del sistema receptor.

Los métodos de ecualización utilizados en sistemas FBMC se pueden clasificar en los tres grupos descritos a continuación. El primero de ellos trata de seleccionar los filtros prototipo que permitan ofrecer mejor ICI e ISI, sin embargo esta opción es limitada

debido a las características del medio y de la modulación utilizada. Con la mejora de los filtros prototipo utilizados es posible utilizar un método de ecualización sencillo por cada subportadora y mejorar la recuperación de los símbolos recibidos [Lou08, Lel07, Far03]. En este caso, este tipo de ecualización utiliza la inversa de la respuesta en frecuencia del canal, de modo que el rendimiento obtenido en la estimación de canal afectará directamente al rendimiento de la ecualización.

Otra manera de realizar la ecualización en FBMC es utilizar ecualizadores que traten de minimizar el error cuadrático medio (MMSE, *Minimum Mean Square Error*) [Wal08, Hol09, Ikh09, Hir80], o ecualizadores basados en estimaciones de máxima verosimilitud (MLSE, *Maximum-Likelihood Sequence Estimation*) [Bal10]. También es posible encontrar propuestas con ecualizadores de baja complejidad [Iha06], o que realizan el seguimiento del canal [Ber02, Wal08, Wal09].

Otro enfoque que se está utilizando en sistemas FBMC es el uso de filtros FIR de más de una etapa por cada subportadora para mejorar la recuperación de símbolos. Esta ecualización es costosa desde el punto de vista computacional, ya que requiere de un filtro FIR por cada subportadora, pero mejora la comunicación en entornos más hostiles como es el canal PLC. Algunos autores han considerado como ecualizador el uso de *Adaptive Sine-Modulated/Cosine-Modulated Filter Bank Equalizer for Transmultiplexers* (ASCET) [Alh01, Iha05, Iha06, Izu07]. Esta técnica de ecualización se basa en disponer de dos bloques receptores en paralelo, un banco de filtros del coseno modulado (CMFB, *Cosine Modulated Filter Bank*) y un banco de filtros del seno modulado (SMFB, *Sine Modulated Filter Bank*). En cada banco de filtros, se realiza la ecualización por cada subportadora, basado en un filtro FIR, que permite compensar los efectos del canal tras combinar los resultados de ambas ecualizaciones CMFB y SMFB. En la figura 2.5 se muestra el diagrama de bloques de este tipo de ecualización. En [Pin14, Cru16] se propone este tipo de ecualizador con diferentes órdenes y se evalúa el rendimiento que tiene el solapamiento de los filtros en cada subportadora. En la figura 2.6 se muestra el diagrama de bloques del propio ecualizador L-ASCET para una de las subportadoras procedentes de los bancos CMFB y SMFB.

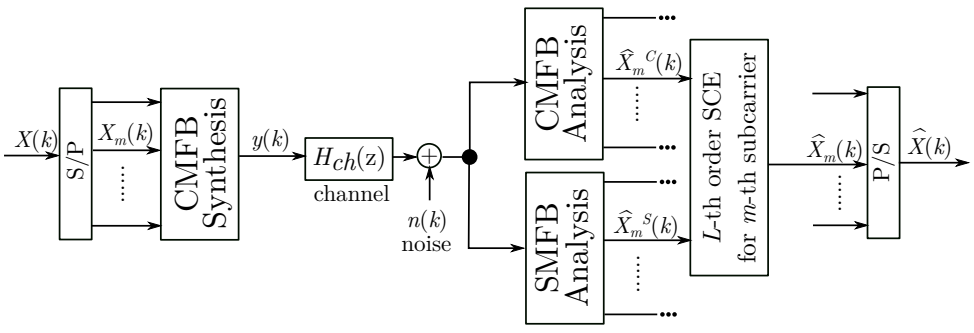


Figura 2.5: Estructura de la ecualización ASCET para un sistema FBMC [Izu07].

En cualquier caso, la complejidad computacional del sistema aumenta, sobre todo en

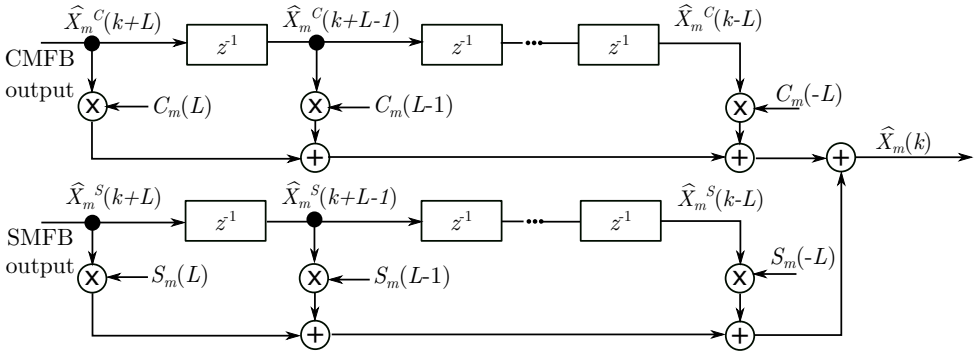


Figura 2.6: Diagrama de bloques generalizado para el ecualizador L-ASCET [Izu07].

la etapa de recepción, donde se llega a duplicar el banco de filtros de la etapa de recepción en algunos casos presentados. Además, cabe destacar que el rendimiento de la ecualización va a ir ligado al rendimiento con el que se obtenga la estimación de canal, ya que esta ecualización se realiza a partir de la estimación obtenida.

2.4 Arquitecturas *hardware* propuestas para PLC

La implementación de las técnicas de acceso al medio de banda ancha para PLC viene desarrollándose en dispositivos FPGA (*Field-Programmable Gate Array*) desde la aparición del estándar IEEE 1901/2010. Estos dispositivos permiten soportar, de forma significativa, la carga computacional que presentan estos sistemas.

Las soluciones de PLC basadas en FPGA se utilizan para la implementación de transmultiplexores, simuladores o emuladores para PLC. En [Mez14] se presenta un transmultiplexor PLC en un dispositivo Xilinx Spartan-6. Continuando con la evolución de los dispositivos, en [Pou18] proponen una arquitectura para la implementación de FBMC para PLC sobre el dispositivo Xilinx Virtex-6, donde se integra también un procesador *soft-core* encargado de monitorizar el sistema y gestionar las transferencias de datos. Aplicado a comunicaciones móviles 5G, en [Nad14] evalúan el rendimiento que presenta FBMC frente a OFDM mediante la implementación del algoritmo en un dispositivo Zynq® 7000 de Xilinx, el cual incluye un ARM junto con la FPGA. También con la familia de *Multi-Processor System-on-Chip* (MPSoC) de Xilinx, en [Mak15] se propone una arquitectura heterogénea para comunicaciones 4G; en esta propuesta se implementa un sistema MIMO-OFDM, en los dos núcleos que tiene el ARM de Zynq® 7000 junto con varios procesadores *soft-core*, evaluando el rendimiento de la arquitectura heterogénea para un diferente número de procesadores ARM y procesadores *soft-core*.

Por último, se van a destacar una serie de implementaciones llevadas a cabo en dispositivos FPGA sobre técnicas de estimación de canal e igualación de canal. En [ElW09] se propone un estimador de canal para ser usado con WiMAX implementado sobre el dispositivo Stratix III de Altera. En [Can12] realiza la implementación sobre la FPGA

Virtex II de un receptor para WLAN basado en OFDM, además en esta propuesta se propone realizar la sincronización entre emisor y receptor. Otra implementación de OFDM para el estándar WiFi se encuentra en [Man04], en este caso se realiza también sobre la Virtex II evaluando el consumo de recursos que tiene el sistema. En [Par11] se muestra un sistema MIMO para OFDM con una FPGA Virtex II. Para el caso de PLC, en [Nom17] se realiza la propuesta de estimación e igualación de canal para sistemas FBMC. Además, se presenta en [Nom15] la forma de llevar a cabo la sincronización entre transmisor y receptor, ambas propuestas se llevan a cabo para la FPGA Xilinx Kintex-7. En la tabla 2.1 se resumen las características principales de las arquitecturas mencionadas.

Ref.	**Descripción**	**Técnica de acceso al medio**	**Dispositivo utilizado**	**Uso de procesador**
[Mez14]	Implementación de un transmultiplexor para PLC	OFDM	Spartan-6 de Xilinx	-
[Pou18]	Implementación de un transmultiplexor para PLC	FBMC	Virtex-6 de Xilinx	Usa el procesador *soft-core* para gestión
[Nad14]	Comparativa de prestaciones en 5G	OFDM FBMC	Zynq-7000 de Xilinx	Usa el procesador ARM para la gestión
[Mak15]	Arquitectura heterogénea para 4G	MIMO OFDM	Zynq-7000 de Xilinx	Combina los núcleos de ARM con varios procesadores *soft-core*
[ElW09]	Estimador de canal para WiMAX	OFDM	Startix-III de Altera	-
[Can12]	Algoritmo de sincronización para WLAN	OFDM	Virtex-II de Xilinx	-
[Man04]	Estimador de canal para WiFi	OFDM	Virtex-II de Xilinx	-
[Par11]	Algoritmo de sincronización y estimación de canal	MIMO OFDM	Vitex-II de Xilinx	-
[Nom17]	Estimador e igualador de canal	FBMC	Kintex-7 de Xilinx	-
[Nom15]	Implementación de algoritmo de sincronización	FBMC	Kintex-7 de Xilinx	-

Tabla 2.1: Resumen con las características principales de las arquitecturas recogidas en el apartado 2.4.

2.5 Objetivos de la tesis

A partir de los antecedentes descritos previamente, el objetivo principal de la tesis doctoral es la definición de una arquitectura SoC eficiente y configurable para un sistema transmisor y receptor para PLC de banda ancha. Dicho sistema utilizará un banco de filtros multiportadora FBMC [Far11, Ten14] como técnica de acceso al medio e incluirá técnicas de estimación e igualación del canal en la parte del receptor, mejorando así el rendimiento del sistema global. Todo el diseño de la arquitectura eficiente se hará mediante el uso de herramientas de síntesis de alto nivel (HLS) para el modelado de la componente hardware y se obtendrá una aproximación HW/SW, utilizando una plataforma SoC.

Para conseguir este objetivo global se ha realizado una lista de objetivos parciales que se van a ir abordando con más detalle a lo largo de los siguientes capítulos:

- Estudio de la técnica de acceso al medio de modulación multiportadora basada en bancos de filtros (FBMC) teniendo en cuenta las particularidades del canal PLC de banda ancha recogidas en el estándar IEEE 1901/2010 [IEE10]. El objetivo es conocer cada una de las partes que compone esta técnica de modulación en los sistemas de transmisión PLC.

- Estudio de la técnica de estimación de canal e igualación de canal para sistemas multiportadora FBMC, con el fin de mejorar la transmisión punto a punto. Para ello se determinará la técnica de estimación de canal a utilizar, que permita obtener una representación del canal lo más fidedigna posible. Posteriormente se realizará la igualación de canal o ecualización, para compensar los efectos no deseados que se han introducido en la señal recibida tras su paso por el canal PLC.

- Definición de arquitecturas eficientes para la implementación en tiempo real de las propuestas realizadas. Una vez realizado los estudios previos, se llevará a cabo el diseño de la arquitectura, lo cual requerirá de un estudio en coma fija y coma flotante de la arquitectura. A partir de los resultados y las características del sistema, se evaluará la posibilidad de realizar un diseño mixto HW/SW que contenga partes en coma fija y otras en coma flotante, mejorando así las prestaciones de precisión del sistema.

- Con el modelo anterior definido, se pretende realizar una implementación de la propuesta realizada haciendo uso de herramientas de síntesis de alto nivel. Este proceso evaluará el sistema en términos de recursos *hardware* necesarios para su implementación. Se evaluará la posibilidad de realizar la implementación de la arquitectura de forma paralela o secuencial. Esto implica que se estudie la posibilidad de reutilizar los recursos disponibles alcanzando las restricciones de tiempo real impuestas por el sistema. El objetivo es alcanzar el compromiso entre consumo de recursos y el rendimiento de la arquitectura.

- Para aquellas partes de la arquitectura que se han definido en coma flotante o que presentan una menor exigencia computacional, se estudiará la viabilidad de utilizar sistemas basados en multiprocesador frente al diseño *hardware*. Esto conduce al planteamiento de una solución heterogénea lo más eficiente posible, para ello se debe establecer el medio de comunicación entre los diferentes núcleos, explorando las posibilidades que tiene cada dispositivo.

- Con la arquitectura definida, se procederá a evaluar y validar la propuesta mediante pruebas experimentales. Este punto abordará la implementación en un SoC de las etapas de emisor y receptor con las diferentes técnicas de estimación e igualación de canal evaluadas anteriormente.

Capítulo 3

Aspectos de la modulación FBMC, estimación e igualación de canal

3.1 Introducción

Entre todos los estándares que regulan las comunicaciones PLC se va a tomar como referencia el estándar IEEE 1901/2010 [IEE10]. En este documento se establecen las especificaciones de acceso a la capa física para PLC de banda ancha. Entre las especificaciones se pueden encontrar las técnicas de acceso al medio propuestas por el estándar, las cuales proponen usar las técnicas multiporadora basadas en banco de filtros (FBMC, Filter Bank Multiple Carrier).

Como se ha indicado en el capítulo anterior, a diferencia de OFDM, las técnicas FBMC no hacen uso del prefijo cíclico (CP, Cylic Prefix) mejorando la eficiencia espectral. Además, el uso de filtros prototipos adecuados permite reducir el ICI debido a que los lóbulos laterales que presentan están más atenuados.

Por otro lado, para mejorar la comunicación se va a hacer uso de un estimador e igualador de canal. El estándar no establece ninguna técnica de estimación de canal a utilizar, por lo tanto, cualquiera de las mencionadas en el capítulo anterior puede ser utilizada. En [Nom17] se hace uso de un estimador LS para canales PLC. Este estimador trata de minimizar el error entre la señal transmitida y recibida, obteniendo así la estimación del canal. Al tratarse de un estimador sencillo, se va a considerar éste para llevar a cabo la implementación. Hay que tener en cuenta las secuencias que se van a usar, ya que deben tener buenas propiedades de auto-correlación y correlación cruzada para obtener de forma eficiente la estimación del canal.

En cuanto a la igualación de canal, los ecualizadores ASCET tienen la ventaja de estar preparados para usarse con bancos de filtros basados en el coseno modulado (CMFB, *Cosine Modulated Filter Bank*) y seno modulado (SMFB, *Sine Modulated Filter Bank*), que serán los utilizados. Estos ecualizadores también poseen la desventaja de tener que duplicar en la etapa de recepción ambos bancos filtros, uno para CMFB y otro para

SMFB. No obstante, este tipo de ecualizador presenta un buen rendimiento en medios de comunicación ruidosos y con notables interferencias, siempre que se utilice un orden superior al 0-ASCET.

Con todo esto, en la figura 3.1 se muestra el diagrama de bloques de la arquitectura propuesta. En ella se muestra el transmisor y receptor FBMC basados en CMFB. Además, la etapa de recepción incluye los bloques de estimación de canal y cálculo de coeficientes para el ecualizador ASCET. Al utilizar el igualador ASCET, se puede comprobar que el receptor contiene dos bancos de filtros en paralelo (CMFB y SMFB). Por último, cabe destacar que el transmisor FBMC incluye un bloque donde se almacenan las secuencias que se enviarán por el canal para obtener la estimación.

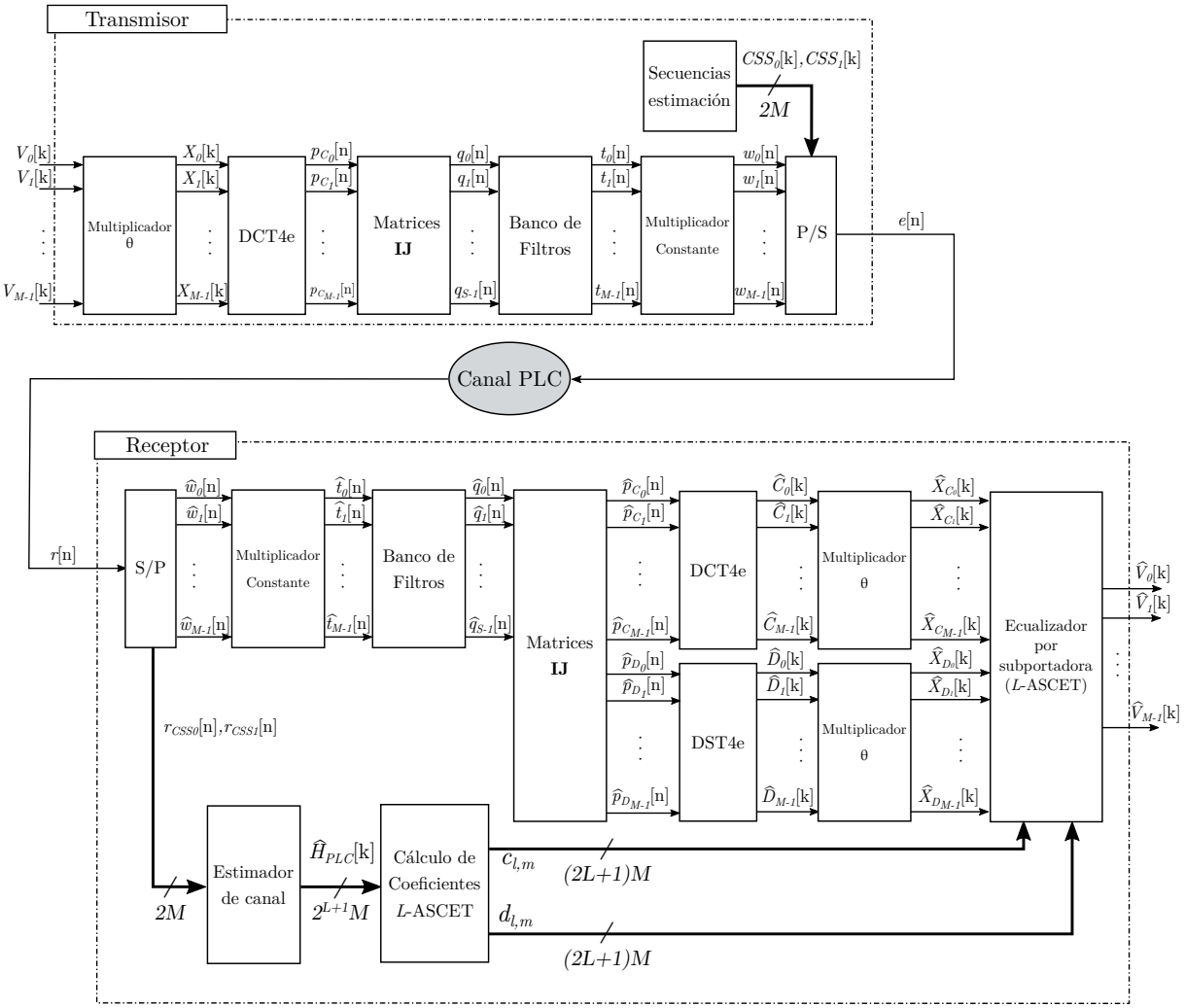


Figura 3.1: Diagrama de bloques del transmultiplexor FBMC, incluyendo el estimador y ecualizador de canal ASCET.

A continuación se va a realizar una descripción de cada uno de los bloques que componen el transmisor FBMC. Para el receptor FBMC sólo se describirán aquellos módulos que difieren con respecto al transmisor, como son el *banco de filtros*, el módulo *matrices IJ* y el módulo *DST4e*, así como los módulos de estimación y ecualización de canal que se incluyen en el receptor FBMC.

Una vez descrito cada módulo del sistema, se procederá a realizar un estudio sobre la subdivisión modular, con el fin de determinar el reparto de los módulos entre la PL (*hardware*) y el PS (*software*). Este reparto tendrá en cuenta las restricciones temporales del conjunto impuestas por el estándar, así como las características de cada módulo, definiendo una arquitectura mixta HW/SW.

3.2 Transmisor FBMC

En la figura 3.2 se muestra un diagrama de bloques del transmisor FBMC. En esta figura se puede observar un conjunto de bloques conectados en cascada que conforman el CMFB del transmisor FBMC. Cada uno de los bloques que lo componen serán descritos funcionalmente a lo largo de esta sección. Por otro lado, el bloque *Secuencias estimación* almacena las tramas que permitirán realizar la estimación de canal en la etapa de recepción.

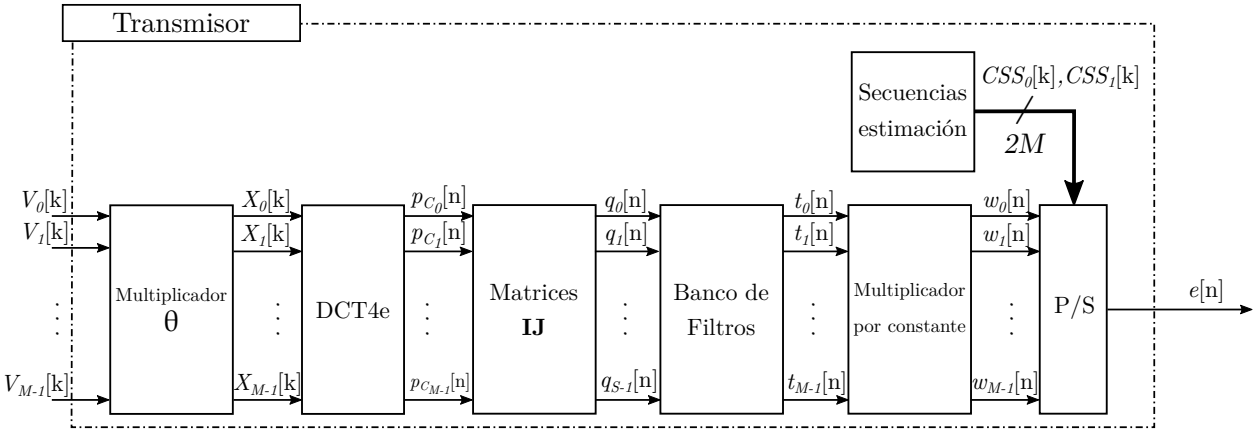


Figura 3.2: Diagrama de bloques del transmisor FBMC con el módulo *Secuencias estimación*, para la estimación de canal.

3.2.1 Módulo *Multiplicador θ*

El módulo *Multiplicador θ* realiza el producto de cada una de las señales de entrada $V_m[k]$ por la constante θ_m. En (3.1) se muestra la descripción matemática de θ_m; siendo m el índice de cada una de las M subportadoras de entrada y Θ_k un ángulo de fase que permite reducir la potencia de pico de la señal de salida. El ángulo de fase Θ_k viene definido en el estándar IEEE 1901/2010 y se pueden consultar en [IEE10, Tabla 14-10]. Como se indica en la referencia, los valores que puede tomar Θ_k son 0 ó π, siendo k el índice de la subportadora.

$$\theta_m = \cos(\Theta_k) \cdot \cos\left(\left(\frac{\pi}{2} \cdot \left(m + \frac{1}{2}\right)\right)\right) \cdot \sqrt{2} \qquad \text{siendo } m = 0, \ldots, M-1 \tag{3.1}$$

Los valores resultantes de (3.1) son 1 ó -1. Por lo tanto, el resultado $X_m[k]$ generado por este módulo será la misma señal de entrada $V_m[k]$ cuando se multiplique por $\theta_m = 1$; o ella misma negada cuando se multiplique por $\theta_m = -1$. La expresión de la señal de salida $X_m[k]$ del módulo *Multiplicador* θ se expresa en (3.2), donde m es el índice de cada una de las M subportadoras.

$$X_m[k] = V_m[k] \cdot \theta_m \qquad \text{siendo } m = 0, \ldots, M-1 \tag{3.2}$$

3.2.2 Módulo *Transformada Discreta del Coseno*

Siguiendo el orden, el siguiente bloque del transmisor es el módulo *Transformada Discreta del Coseno*; este módulo realiza la transformada DCT tipo IV par (DCT4e). Esta se basa en la transformada rápida de Fourier (FFT, *Fast Fourier Transform*); pero, a diferencia de la FFT, la DCT4e se caracteriza por su capacidad de compactar la energía, y además tiene la propiedad de que la DCT4e directa e inversa son análogas, de forma que no es necesario implementar un bloque diferente en la etapa de recepción, como sí ocurre con el caso de la FFT, implementando la inversa de la transformada rápida de Fourier (IFFT, *Inverse Fast Fourier Transform*).

En (3.3) se describe matemáticamente la expresión de la DCT4e, donde $X_m[k]$ es la entrada y $p_m[n]$ es la correspondiente salida transformada. Como anteriormente, el subíndice m se corresponde con cada una de las M subportadoras del sistema.

$$p_m[n] = \sum_{l=0}^{M-1} X_l[k] \cdot cos\left[\frac{\pi}{M}\left(m+\frac{1}{2}\right)\left(l+\frac{1}{2}\right)\right] \qquad \text{siendo } m = 0, \ldots, M-1 \tag{3.3}$$

Como se ha mencionado con anterioridad, la descripción matemática de la DCT4e se puede expresar en función de la FFT. Esto se ha realizado definiendo cuatro etapas como se muestra en la figura 3.3. A continuación se describe matemáticamente la forma de obtener la DCT4e a partir de la FFT.

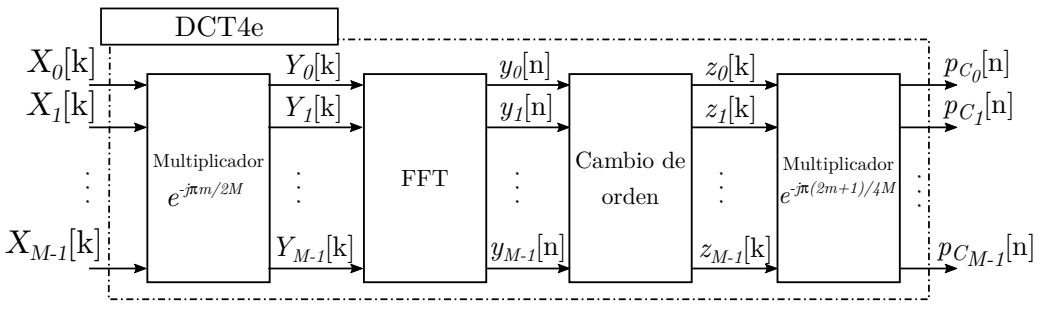


Figura 3.3: Diagrama de bloques del módulo DCT4e dividido en cuatro etapas.

La primera de las etapas de la DCT4e realiza la multiplicación de la señal de entrada $X_m[k]$ por un factor $\left(e^{\frac{-jm\pi}{2M}}\right)$ como se muestra en (3.4), donde m representa a cada una de las M subportadoras.

$$Y_m[k] = X_m[k] \cdot \left(e^{\frac{-jm\pi}{2M}}\right) \qquad \text{siendo } m = 0, \ldots, M-1 \tag{3.4}$$

La segunda etapa consiste en realizar la FFT de M puntos. En este caso, se usa un algoritmo eficiente para la FFT, como el algoritmo Cooley-Tukey [Coo65]. Esta etapa se describe en (3.5), donde $Y_l[k]$ es la señal de entrada a la FFT, $y_m[n]$ es el resultado de la FFT y m es el índice de cada una de las M subportadoras.

$$y_m[n] = \sum_{l=0}^{M-1} Y_l[k] \cdot \left(e^{\frac{-2\pi j}{M} \cdot l \cdot m}\right) \qquad \text{siendo } m = 0, \ldots, M-1 \tag{3.5}$$

En la tercera etapa, se ordena la salida de la FFT $y_m[n]$, aplicando el conjugado a las muestras impares del resultado de la FFT. En (3.6) se describe matemáticamente esta etapa, donde $i = 0, \ldots, {}^{M}/_{2} - 1$ y $z_m[n]$ es la salida de esta etapa.

$$\begin{aligned} z_{2i}[n] &= y_i[n] \\ z_{2i+1}[n] &= conj\left(y_{M-i-1}[n]\right) \end{aligned} \qquad \text{siendo } i = 0, \ldots, {}^{M}/_{2} - 1 \tag{3.6}$$

La última etapa de la FFT realiza una multiplicación por un factor similar a la primera etapa, pero en este caso el factor es $\left(e^{\frac{-j(2m+1)\pi}{4M}}\right)$. El resultado de la DCT4e $p_{C_m}[n]$ es el resultado de la parte real $\mathbb{R}\mathrm{e}$ del producto de la señal de entrada $z_m[n]$ por el factor mencionado $\left(e^{\frac{-j(2m+1)\pi}{4M}}\right)$ y multiplicado por dos. Esta etapa se puede expresar como se muestra en (3.7), donde m es el índice de cada una de las M subportadoras.

$$p_{C_m}[n] = 2 \cdot \mathbb{R}\mathrm{e}\left[z_m[n] \cdot \left(e^{\frac{-j(2m+1)\pi}{4M}}\right)\right] \qquad \text{siendo } m = 0, \ldots, M-1 \tag{3.7}$$

Cabe destacar que en el caso de operar con entradas complejas, esta etapa se debería realizar dos veces, una para la parte real de los datos de entrada, lo que genera la parte real de los datos de salida; y otra para la parte imaginaria de los datos de entrada, lo que genera la parte imaginaria de los datos de salida.

Como se ha mencionado anteriormente, la DCT4e directa e inversa presentan la misma definición. Sin embargo, ha de tenerse en cuenta que en el caso del receptor, la señal de entrada será $\hat{p}_{C_m}[n]$ y el resultado se denota como $\hat{C}_m[k]$ en la figura 3.1, en lugar de $X_m[k]$ y $p_{C_m}[n]$ para el caso del transmisor.

3.2.3 Módulo *Matrices IJ*

El siguiente módulo, denominado *Matrices IJ*, realiza la multiplicación de la señal de entrada $p_{C_m}[n]$ por una combinación de las matrices identidad $\mathbf{I}_M$ y la matriz de intercambio (*exchange matrix*) $\mathbf{J}_M$. A diferencia de la matriz identidad, la matriz de intercambio presenta en la diagonal secundaria todos los elementos a 1 y el resto de elementos de la matriz son 0. La combinación de ambas matrices en este módulo genera S señales de salida $q_s[n]$, siendo $S = 2 \cdot M$. En (3.8) se muestra la descripción matemática de este módulo, donde $\mathbf{p}_C[n]$ representa el vector de M subportadoras de entrada y m el índice de cada una de las M subportadoras. Cabe destacar que las $S/2$ últimas señales de salida tienen un retardo de una unidad.

$$\begin{aligned} q_m[n] &= \mathbf{p}_C[n]\ (\ \mathbf{I}_M - \mathbf{J}_M) \ = \ p_{C_m}[n] - p_{C_{M-1-m}}[n] \\ q_{M+m}[n-1] &= \mathbf{p}_C[n]\,(-\mathbf{I}_M - \mathbf{J}_M) = -p_{C_m}[n] - p_{C_{M-1-m}}[n] \\ &\qquad\qquad \text{siendo } m = 0, \ldots, M-1 \end{aligned} \tag{3.8}$$

En la figura 3.4 se muestra el diagrama de bloques de esta etapa, donde se identifican las M señales de entrada $p_{C_m}[n]$ y las S señales de salida $q_s[n]$ con el retardo correspondiente.

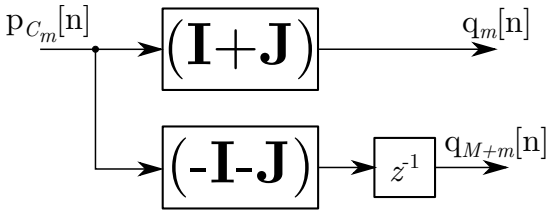


Figura 3.4: Diagrama de bloques del módulo *Matrices IJ*.

Este módulo permite adaptar las M subportadoras de entrada al sistema con el banco de filtros polifásico que, como se verá a continuación, se define como un conjunto de $S = 2 \cdot M$ filtros de Respuesta Finita al Impulso (FIR, *Finite Impulse Response*) de orden siete.

3.2.4 Módulo *Banco de Filtros Polifásico*

Después del módulo *Matrices IJ*, se encuentra el banco de filtros polifásico. Este módulo consiste en un banco de filtros $G_s(z)$, el cual se define como un conjunto de $S = 2 \cdot M$ filtros FIR de orden siete, cuyos coeficientes no nulos son únicamente el primero $g_{s,0}$ y el tercero $g_{s,2}$. En (3.9) se muestra la definición de filtro $G_s(z)$.

$$G_s(z) = g_{s,0} + g_{s,2} \cdot z^{-2} \tag{3.9}$$

Tanto en el transmisor como en el receptor se utiliza el mismo banco de filtros. Es necesario mencionar que a la salida del módulo del banco del filtros se obtienen de nuevo M señales $t_m[n]$, a partir de las S señales de entrada $q_s[n]$; para ello, las $S/2$ primeras señales de la salida del banco de filtros se suman con las siguientes $S/2$ señales. Esto se describe en (3.10), donde m representa cada una de las M subportadoras del sistema.

$$\begin{aligned} t_m[n] = q_m[n] \cdot g_{m,0} + q_m[n-2] \cdot g_{m,2} + ... \\ ... + q_{M+m}[n] \cdot g_{M+m,0} + q_{M+m}[n-2] \cdot g_{M+m,2} \\ \text{siendo } m = 0, \ldots, M-1 \end{aligned} \tag{3.10}$$

3.2.5 Módulo *Multiplicador constante*

El último de los módulos del FBMC transmisor es el *Multiplicador constante* $1/\sqrt{2}$. En este caso se multiplica cada una de las M señales $t_m[n]$ por el factor constante $1/\sqrt{2}$, obteniendo así la salida del banco de filtros $w_m[n]$. La descripción matemática de este módulo se muestra en (3.11).

$$w_m[n] = t_m[n] \cdot (1/\sqrt{2}) \text{ siendo } m = 0, ..., M-1 \tag{3.11}$$

3.2.6 Módulo *Secuencias para la estimación*

Como se ha comentado en el apartado 2.3.1, uno de los estimadores de canal más sencillos es el estimador de mínimos cuadrados (LS, *Least Squares*). Este tipo de estimador minimiza el error cuadrático entre los símbolos piloto transmitidos y los recibidos. Esta propiedad va a permitir que se obtenga una estimación del canal $\hat{H}_{PLC}$. La forma detallada de obtener esta estimación se describirá en el apartado 3.3.4 de la etapa de recepción, ya que este bloque únicamente se encarga de enviar dichos símbolos piloto. A continuación, se detallan las características de estos símbolos piloto.

Los símbolos piloto son secuencias formadas por los conjuntos complementarios de secuencias (CSS, *Complementary Sets of Sequences*). En este caso se intercalan dos secuencias CSS de longitud M antes del envío de información útil por el canal. La principal característica de las secuencias CSS es que tienen buenas propiedades de auto-correlación y correlación cruzada.

Por tanto, en la etapa de transmisión se almacenan las dos secuencias CSS que se emiten antes del envío de datos. Sin embargo, para realizar correctamente la estimación de canal, es necesario que el transmisor y el receptor se encuentren sincronizados; para ello antes de realizar el envío de las secuencias utilizadas para llevar a cabo la estimación, se deberán enviar otro tipo de secuencias para la sincronización del sistema. En este caso,

se considera que el sistema tiene una sincronización perfecta y no se hace uso de las secuencias para la sincronización.

En la figura 3.5 se muestra un ejemplo de envío de tramas. En ésta se indica el preámbulo formado por los paquetes S_0, S_1, C_0 y C_1, así como la trama de datos formado por los sucesivos paquetes $D_0, D_1, \ldots$. Cada uno de los paquetes está compuesto por M muestras, coincidiendo con el número de M subportadoras. Como se puede observar, antes de las tramas de estimación de canal (C_0 y C_1), se enviarían los paquetes S_0 y S_1 para llevar a cabo la sincronización del sistema.

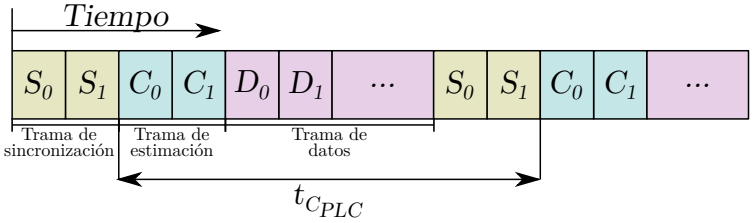


Figura 3.5: Esquema de la trama de paquetes de sincronismo, estimación de canal y datos por el canal.

En el apartado 2.3 se ha explicado el concepto de tiempo de coherencia del canal $t_{C_{PLC}}$, el cual se establece en 600 µs en [Cor06, Zim02b]. Si no se supera este tiempo, es posible considerar el canal PLC como un sistema lineal e invariante en el tiempo (LTI, *Linear Time-Invariant*), compensando así los efectos introducidos por el canal mediante el uso de un ecualizador. Cabe destacar que el tiempo máximo que debe trascurrir entre el envío de las tramas de estimación de canal no debe superar el $t_{C_{PLC}}$.

En resumen, este módulo realiza el control para la multiplexación entre las tramas de estimación de canal y los paquetes de datos, según corresponda en cada caso, en base al tiempo de coherencia del canal PLC ($t_{C_{PLC}}$).

3.3 Receptor FBMC

El receptor FBMC es análogo al transmisor FBMC, siendo complementarios los bloques descritos anteriormente. Sin embargo, para realizar la igualación de canal en esta etapa, es necesario disponer del CMFB y el SMFB en paralelo. Como se ve en la figura 3.6, se han fusionado ambos bancos de filtros para simplificar y disponer de una arquitectura final más eficiente.

Las partes comunes a ambos bancos de filtros son el módulo *Multiplicador por constante* y el *Banco de Filtros*. A partir del módulo *Matrices IJ* difieren, siendo este módulo el que genera las salidas $\hat{p}_{C_m}[n]$ y $\hat{p}_{D_m}[n]$ para ambos bancos de filtros CMFB y SMFB, respectivamente.

A continuación se detallan los módulos *Banco de Filtros*, *Matrices IJ*, *DST4e*, *Estimación de canal* y *Cálculo de Coeficientes* para el Ecualizador por subportadora *L*-ASCET.

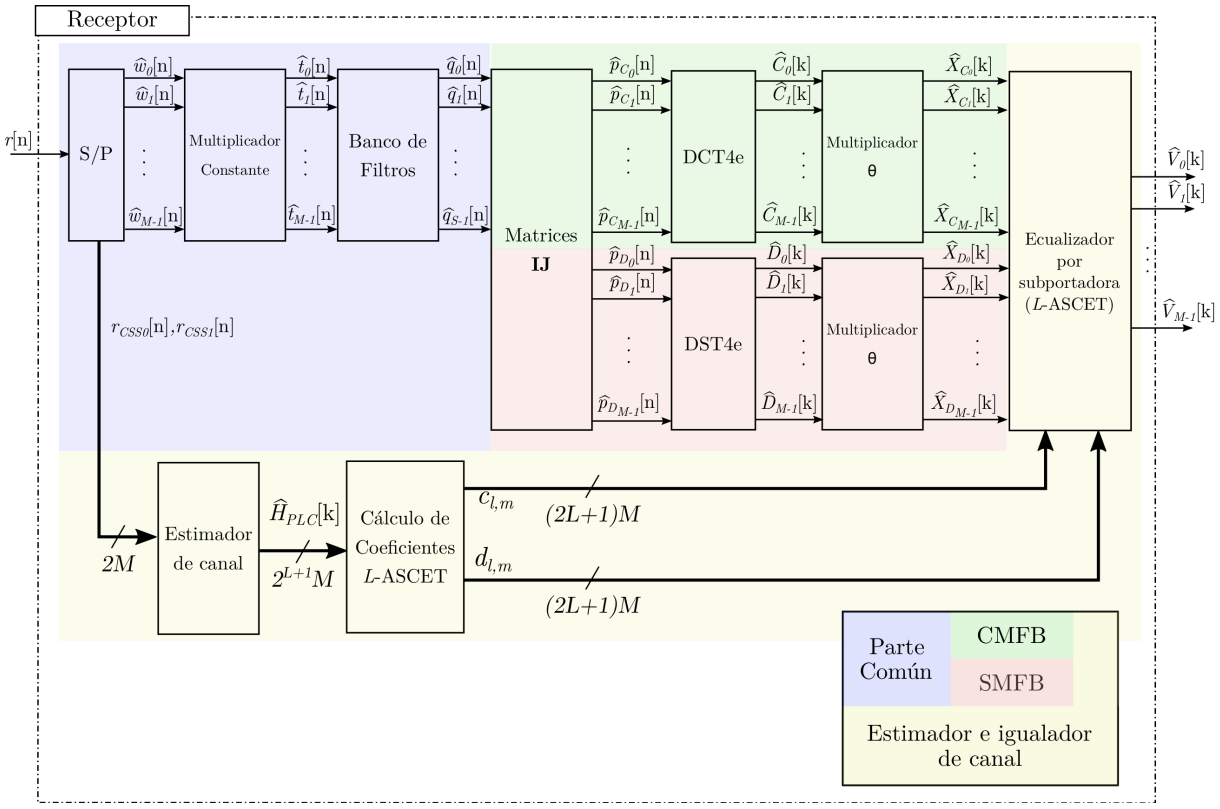


Figura 3.6: Diagrama de bloques del receptor FBMC, incluyendo la etapa de estimación y ecualización de canal, así como el cálculo de coeficientes para el ecualizador ASCET.

En este último módulo se detallará el cálculo de los coeficientes necesarios para la igualación de canal.

3.3.1 Módulo *Banco de Filtros Polifásico*

El módulo *Banco de Filtros* del receptor es ligeramente diferente al del transmisor. El filtro $G_s(z)$ no varía, por lo que la definición anterior del filtro (3.9) sirve para este módulo. Sin embargo, en este caso, el número de entradas de este módulo $\hat{t}_m[n]$ es de M señales y produce $S = 2 \cdot M$ señales de salida $\hat{q}_s[n]$. En (3.12) se muestra la expresión matemática de este módulo.

$$\begin{aligned} \hat{q}_m[n] &= \hat{t}_m[n] \cdot g_{m,0} + \hat{t}_m[n-2] \cdot g_{m,2} \\ \hat{q}_{M+m}[n-1] &= \hat{t}_m[n] \cdot g_{m,0} + \hat{t}_m[n-2] \cdot g_{m,2} \\ &\text{siendo } m = 0, \ldots, M-1 \end{aligned} \tag{3.12}$$

Cabe destacar que en la etapa de recepción las M últimas señales de salida $\hat{q}_s[n]$ tienen un retardo de una unidad. Este retardo es análogo al que existía en el transmisor FBMC en el módulo *Matrices IJ*.

3.3.2 Módulo *Matrices IJ*

Como se ha comentado anteriormente, este módulo es el que difiere entre el CMFB y el SMFB, y por tanto genera las M señales de salida para ambos filtros. Cada una de las salidas de este módulo $\hat{p}_{C_m}[n]$ y $\hat{p}_{D_m}[n]$, se corresponden con la salida al CMFB y el SMFB, respectivamente. Ambas salidas se obtienen a partir de la señal de entrada $\hat{q}_s[n]$, mediante la combinación de la matriz identidad $\mathbf{I}_M$ y la matriz de intercambio (*exchange matrix*) $\mathbf{J}_M$, siendo $S = 2 \cdot M$. Esta combinación es complementaria a la que se realizaba en el transmisor; y además, para el caso del SMFB, es complementaria a la del CMFB. En (3.13) y (3.14) se muestra la definición de las salidas del CMFB y el SMFB respectivamente, donde $\hat{\mathbf{q}}_1[n]$ es el vector de las M primeras señales de entrada $\hat{q}_s[n]$ y $\hat{\mathbf{q}}_2[n]$ el vector de las M últimas señales de entrada $\hat{q}_s[n]$, para el número de entradas $S = 2 \cdot M$.

$$\begin{aligned}\hat{p}_{C_m}[n] &= \hat{\mathbf{q}}_1[n](\mathbf{I}_M + \mathbf{J})_M + \hat{\mathbf{q}}_2[n](\mathbf{I}_M - \mathbf{J}_M) \\ &= \hat{q}_m[n] + \hat{q}_{M-1-m}[n] + \hat{q}_{M+m}[n] - \hat{q}_{2\cdot M-1-m}[n] \\ &\qquad \text{siendo } m = 0, \ldots, M-1\end{aligned} \tag{3.13}$$

$$\begin{aligned}\hat{p}_{D_m}[n] &= \hat{\mathbf{q}}_1[n](\mathbf{I}_M - \mathbf{J}_M) + \hat{\mathbf{q}}_2[n](\mathbf{I}_M + \mathbf{J}_M) \\ &= \hat{q}_m[n] - \hat{q}_{M-1-m}[n] + \hat{q}_{M+m}[n] + \hat{q}_{2\cdot M-1-m}[n] \\ &\qquad \text{siendo } m = 0, \ldots, M-1\end{aligned} \tag{3.14}$$

En la figura 3.7 se muestra el diagrama de bloques de este módulo. Cabe destacar que las S señales de entrada generan un conjunto de M señales a la salida para cada uno de los bancos de filtros que corresponda (CMFB y SMFB).

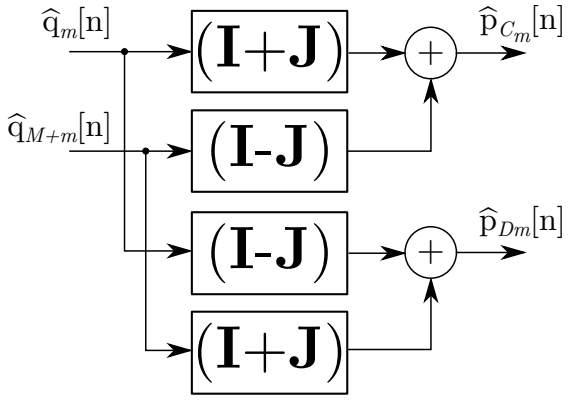


Figura 3.7: Diagrama de bloques del módulo *Matrices IJ* del receptor FBMC, el cual genera las salidas $\hat{p}_{C_m}[n]$ y $\hat{p}_{D_m}[n]$, correspondientes al CMFB y SMFB.

3.3.3 Módulo *Transformada Discreta del Seno*

Siguiendo el diagrama de bloques de la figura 3.6, el siguiente módulo en el receptor FBMC son los módulos DCT4e y DST4e, para el CMFB y el SMFB respectivamente.

Como se ha mencionado en el apartado 3.2.2, una de las propiedades de las DCT4e es que la forma directa e inversa son iguales, por tanto, en el receptor FBMC el módulo DCT4e es igual que el descrito en la subsección 3.2.2 para el transmisor FBMC.

En cuanto a la transformada DST4e, se puede definir matemáticamente como la expresión (3.15), donde $\hat{p}_{Dm}[n]$ y $\hat{D}_m[k]$ se corresponden con la señal de entrada y salida respectivamente.

$$\hat{D}_m[k] = \sum_{n=0}^{M-1} \hat{p}_{Dm}[n] \sin\left(\frac{\pi}{M}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right) \quad \text{siendo } k = 0, \ldots, M-1 \tag{3.15}$$

Al igual que anteriormente, se podría partir de (3.15), descomponiendo el algoritmo en etapas y utilizar la FFT para obtener la DST4e. Sin embargo, de acuerdo con la propiedad descrita en [Sha08], se puede obtener la DST4e a partir de la DCT4e, multiplicando las entradas impares de la DCT4e por -1 e invirtiendo el orden de las señales de salida de la DCT4e. Considerando esta propiedad, la DST4e se puede definir como se muestra en (3.16) a partir de la DCT4e.

$$\hat{D}_m[M-1-k] = \sum_{n=0}^{M-1} (-1)^n \hat{p}_{Dm}[n] \cos\left(\frac{\pi}{M}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right) \quad \text{siendo } k = 0, \ldots, M-1 \tag{3.16}$$

En la figura 3.8 se muestra el diagrama de bloques del módulo DST4e, a partir de la DCT4e; cabe destacar que estos módulos adicionales no implican una sobrecarga operacional, ya que sólo implica realizar un cambio de signo y reordenar la salida de la DCT4e.

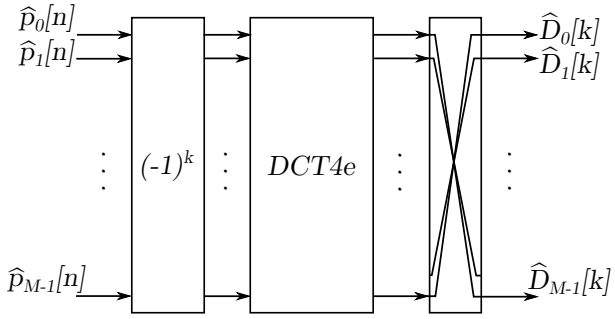


Figura 3.8: Diagrama de bloques del cálculo de la DST4e a partir de la DCT4e.

La propiedad descrita permite simplificar la posterior forma de implementar el algoritmo del receptor, desarrollando la arquitectura para la DCT4e y aplicando la propiedad anterior para obtener la DST4e.

3.3.4 Módulo de *Estimación de canal*

Considerando el canal PLC como un canal discreto, con un número finito de etapas, este canal se puede modelar con un sistema lineal variante en el tiempo. Sin embargo, teniendo en cuenta el tiempo de coherencia del canal $t_{C_{PLC}}$, si dicho tiempo no se supera, el canal se puede considerar un sistema lineal e invariante en el tiempo (LTI). Es decir, el canal puede modelarse como un conjunto de ganancias o constantes complejas como se muestra en (3.17), siendo $H_{PLC}[k]$ la respuesta en frecuencia del canal y l cada una de las componentes del canal con longitud L.

$$H_{PLC}[k] = \sum_{l=0}^{L} h_l[k] \cdot z^{-l} \tag{3.17}$$

Como se ha indicado en el apartado 3.2.6, el módulo transmisor envía dos secuencias CSS a través del canal ($CSS_0[n]$ y $CSS_1[n]$), recibiendo esas señales en el sistema receptor ($r_{CSS_0}[n]$ y $r_{CSS_1}[n]$). Estas señales recibidas se pueden expresar como se muestra en (3.18), siendo $R_{CSS_x}[k]$ la FFT de la señal recibida y el subíndice x el identificador de cada una de las secuencias CSS. $H_{PLC}[k]$ es la respuesta en frecuencia del canal PLC y $\mathcal{N}[k]$ la FFT del ruido presente en el canal.

$$R_{CSS_x}[k] = FFT(CSS_x[n]) \cdot H_{PLC}[k] + \mathcal{N}[k] \tag{3.18}$$

Para obtener la estimación del canal $\hat{H}_{LS_x}[k]$ mediante un estimador LS, se realiza la división de los símbolos piloto recibidos $R_{CSS_x}[k]$ entre la FFT de los símbolos piloto transmitidos $CSS_x[k]$. En (3.19) se muestra la descripción matemática de esta etapa, donde el subíndice x es el identificador de cada símbolo piloto CSS. Cabe destacar que $R_{CSS_x}[k]$ y $CSS_x[k]$ se representan en el dominio de la frecuencia.

$$\begin{aligned} \hat{H}_{LS_x}[k] &= \frac{R_{CSS_x}[k]}{CSS_x[k]} = \\ &= \frac{CSS_x[k] \cdot H_{PLC}[k] + \mathcal{N}[k]}{CSS_x[k]} = \\ &= H_{PLC}[k] + \frac{\mathcal{N}[k]}{CSS_x[k]} \\ &= H_{PLC}[k] + \frac{1}{SNR} \end{aligned} \tag{3.19}$$

Para obtener la respuesta en frecuencia de la estimación de canal, este módulo debe realizar la FFT de cada uno de los símbolos piloto recibidos $r_{CSS_x}[n]$ y dividirlo por la FFT de su correspondiente símbolos transmitidos $CSS_x[k]$, los cuales son conocidos a priori. Por tanto, esto es equivalente a realizar la multiplicación de la FFT de los símbolos

piloto recibidos $r_{CSS_x}[n]$ por una constante conocida $\mathcal{K}_{CSS_x}$, siendo $\mathcal{K}_{CSS_x}$ la inversa de la FFT de los símbolos transmitidos $CSS_x[k]$, como se muestra en (3.20).

$$\begin{aligned}\hat{H}_{LS_x}[k] &= \frac{FFT\{r_{CSS_x}[n]\}}{FFT\{CSS_x[n]\}} = \\ &= FFT\{r_{CSS_x}[n]\} \cdot \frac{1}{FFT\{CSS_x[n]\}} = \\ &= FFT\{r_{CSS_x}[n]\} \cdot \mathcal{K}_{CSS_x}\end{aligned} \tag{3.20}$$

Una vez que se ha obtenido la estimación LS de cada uno de los símbolos con cada una de las estimaciones $\hat{H}_{LS_0}[k]$ y $\hat{H}_{LS_1}[k]$, se realiza una media aritmética de ambas. En (3.21) se expresa este proceso de forma matemática, siendo $\hat{H}_E[k]$ la respuesta en frecuencia del canal estimado.

$$\hat{H}_{E_m}[k] = \frac{\hat{H}_{LS_0}[k] + \hat{H}_{LS_1}[k]}{2} \tag{3.21}$$

Para evitar el ruido impulsivo asíncrono presente en los canales PLC [And09], se limita la salida de la estimación obtenida $\hat{H}_E[k]$. Si se supera el umbral Thr, a la salida se le asignará el valor del $^{Thr}/_{2}$. En (3.22) se muestra la descripción de este proceso, donde m es la longitud de la estimación de canal obtenida que coincide con el número de muestras de las secuencias CSS.

$$\hat{H}_{PLC_m}[k] = \begin{cases} \hat{H}_{E_m}[k] & \text{si } \hat{H}_{E_m}[k] < Thr \\ Thr/2 & \text{si } \hat{H}_{E_m}[k] > Thr \end{cases} \tag{3.22}$$

Esta estimación $\hat{H}_{PLC}[k]$ tiene el mismo número de puntos que el número M de subportadoras del sistema. Sin embargo, para obtener los coeficientes del ecualizador se precisa de un mayor número de puntos y depende del orden del ecualizador ASCET utilizado. Por tanto, para aumentar el número de puntos de la estimación $\hat{H}_{PLC}[k]$ y adaptarlo al orden de ecualizador ASCET utilizado, se ha optado por realizar la interpolación de la estimación de canal obtenida $\hat{H}_{PLC}[k]$. En la siguiente subsección se indica el número de puntos requeridos por cada orden de ecualizador ASCET que definirá el factor de interpolación R que se aplica a la estimación $\hat{H}_{PLC}[k]$.

3.3.5 Módulo para la igualación de canal

La etapa de ecualización de canal permite compensar los efectos no deseados introducidos por el canal. Para ello es necesario tener un modelo del canal utilizado, o una estimación del mismo, y así realizar la igualación de la señal recibida.

En los sistemas multiportadora, la ecualización se debe realizar en cada una de las subportadoras de forma independiente; a este tipo de ecualizador se le conoce como ecualizador por subportadora (PSE, *Per-Subcarrier Equalizer*). Un caso particular de PSE para los sistemas FBMC es el ASCET, el cual se basa en dos bancos de filtros FIR paralelos, uno para el banco de filtros del coseno CMFB y otro para el seno modulado SMFB. Por último, para cada subportadora se suma el resultado de ambos filtros FIR. El número de etapas del filtro FIR es $(2 \cdot L + 1)$, siendo L el orden del ecualizador ASCET. Cabe destacar que el uso de ecualizadores ASCET con órdenes superiores a dos no implica una mejora significativa en la calidad de la señal recibida para canales PLC [Pin14], pero sí que aumenta la complejidad del cálculo de los coeficientes para el ecualizador.

En términos generales, la etapa de ecualización se puede expresar como se muestra en (3.23), siendo $\hat{X}_{C_m}[k]$ y $\hat{X}_{D_m}[k]$ las salidas del CMFB y SMFB, respectivamente. Por otro lado, los símbolos $c_{l,m}$ y $d_{l,m}$ representan los coeficientes del filtro FIR, los cuales son dependientes del canal, para $m = 0, \ldots, M-1$, siendo L el orden del ecualizador ASCET.

$$\hat{V}_m[k] = \sum_{l=0}^{2L} c_{l,m} \cdot \hat{X}_{C_m}[k-l] + \sum_{l=0}^{2L} d_{l,m} \cdot \hat{X}_{D_m}[k-l] \tag{3.23}$$

De forma general, el número de puntos que debe tener la estimación del canal $\hat{H}_{PLC}[k]$, se puede generalizar en función del orden del ecualizador ASCET como $2^{L+1} \cdot M$, obteniendo así que el número de puntos del 0-ASCET, 1-ASCET y 2-ASCET es de $2 \cdot M$, $4 \cdot M$ y $8 \cdot M$, respectivamente.

El método para obtener los coeficientes del ecualizador $c_{l,m}$ y $d_{l,m}$ para el 0-ASCET y 1-ASCET se ha descrito en [Cru16]. Para el caso 2-ASCET se describe en [Pin15]. Con el fin de analizar la complejidad que tiene esta etapa, a continuación se recoge la descripción matemática para realizar el cálculo de los coeficientes necesarios para el ecualizador ASCET.

3.3.5.1 Coeficientes para el ecualizador 0-ASCET

Como se muestra en (3.24), el ecualizador $E_m(\Omega)$ puede expresarse en función del canal, siendo $H_{Ch}(\Omega)$ la respuesta en frecuencia del canal y el símbolo * es el operador conjugado.

$$E_m(\Omega) = \frac{H_{Ch}^*(\Omega)}{\|H_{Ch}(\Omega)\|^2} \tag{3.24}$$

Particularizando (3.24) para $\Omega = (2k+1)\frac{2\pi}{4 \cdot M}$, se obtiene que $E_m(\Omega)$ puede expresarse de forma compleja como se muestra en (3.25), siendo $c_{0,m}$ la parte real y $d_{0,m}$ la parte compleja de $E_m(\Omega)$. Cabe destacar que $c_{0,m}$ y $d_{0,m}$ son los coeficientes del filtro FIR para el caso $L = 0$.

$$E_m(\Omega)|_{\Omega=(2k+1)\frac{2\pi}{4 \cdot M}} = c_{0,m} - j \cdot d_{0,m} \tag{3.25}$$

Para el orden cero del ecualizador ASCET (0-ASCET), los efectos introducidos por el canal se compensan multiplicando cada salida del CMFB y del SMFB por las constantes $c_{0,m}$ y $d_{0,m}$ respectivamente, ya que el orden del filtro FIR es la unidad.

Como se indica en [Alh01], el ecualizador 0-ASCET es utilizado en canales que no sufren variaciones rápidas en el ancho de banda de cada subportadora. Sin embargo, para el caso de PLC esto no ocurre y, por tanto, es recomendable utilizar ordenes superiores de este tipo de ecualizadores.

3.3.5.2 Coeficientes para el ecualizador 1-ASCET

Para el caso del ecualizador 1-ASCET, en [Yan07] se define como un filtro FIR de tres etapas, como se muestra en (3.26).

$$E_m(e^{j\omega}) = \epsilon_{0,m}e^{j\omega} + \epsilon_{1,m} + \epsilon_{2,m}e^{-j\omega} \tag{3.26}$$

Con el objetivo de obtener los valores para cada filtro FIR de (3.26), se seleccionan algunos puntos frecuenciales para cada subportadora m. Siguiendo los pasos mostrados en [Izu07, Yan07, Iha06], los puntos seleccionados son $\omega = 0, \frac{\pi}{2}, \pi$ para las subportadoras pares; y $\omega = -\pi, -\frac{\pi}{2}, 0$ para las subportadoras impares. El resultado del ecualizador 1-ASCET se muestra en (3.27) y (3.28) para las subportadoras pares e impares, respectivamente, siendo m cada una de las M subportadoras del sistema.

Subportadoras pares:

$$E_m(e^{j\omega}) = \left\{ \begin{array}{rl} \epsilon_{0,m} + \epsilon_{1,m} + \epsilon_{2,m} = \eta_{0,m} & \text{para } \omega = 0 \\ j\epsilon_{0,m} + \epsilon_{1,m} - j\epsilon_{2,m} = \eta_{1,m} & \text{para } \omega = \dfrac{\pi}{2} \\ -\epsilon_{0,m} + \epsilon_{1,m} - \epsilon_{2,m} = \eta_{2,m} & \text{para } \omega = \pi \end{array} \right\} \tag{3.27}$$

Subportadoras impares:

$$E_m(e^{j\omega}) = \left\{ \begin{array}{rl} -\epsilon_{0,m} + \epsilon_{1,m} - \epsilon_{2,m} = \eta_{0,m} & \text{para } \omega = -\pi \\ -j\epsilon_{0,m} + \epsilon_{1,m} + j\epsilon_{2,m} = \eta_{1,m} & \text{para } \omega = -\dfrac{\pi}{2} \\ \epsilon_{0,m} + \epsilon_{1,m} + \epsilon_{2,m} = \eta_{2,m} & \text{para } \omega = 0 \end{array} \right\} \tag{3.28}$$

En (3.27) y (3.28), cada uno de los términos $\eta_{x,m}$ mencionados se definen como se muestra en (3.29), siendo el símbolo * es el operador conjugado.

$$\eta_{l,m} = \frac{H^*_{Ch}(e^{j(2k+l)\frac{\pi}{4M}})}{\left\| H_{Ch}(e^{j(2k+l)\frac{\pi}{4M}}) \right\|^2} \qquad \text{siendo } l = 0, 1, 2 \tag{3.29}$$

Por tanto, cada uno de los filtros FIR del 1-ASCET se definen como se muestra en (3.30), (3.31) y (3.32), siendo de signo positivo las subportadoras pares y de signo negativo las impares.

$$\epsilon_{0,m} = \pm\frac{1}{2}\left(\frac{\eta_{0,m} - \eta_{2,m}}{2} - j\left(\eta_{1,m} - \frac{\eta_{0,m} + \eta_{2,m}}{2}\right)\right) \tag{3.30}$$

$$\epsilon_{1,m} = \frac{\eta_{0,m} + \eta_{2,m}}{2} \tag{3.31}$$

$$\epsilon_{2,m} = \pm\frac{1}{2}\left(\frac{\eta_{0,m} - \eta_{2,m}}{2} + j\left(\eta_{1,m} - \frac{\eta_{0,m} + \eta_{2,m}}{2}\right)\right) \tag{3.32}$$

Se puede determinar que cada uno de los filtros (3.30), (3.31) y (3.32) se pueden descomponer como coeficientes complejos según (3.33), siendo $c_{l,m}$ la parte real y $d_{l,m}$ la parte imaginaria, que se corresponden con los coeficientes del filtro FIR del CMFB y del SMFB, respectivamente.

$$\epsilon_{l,m} = c_{l,m} - jd_{l,m} \qquad \text{siendo } l = 0, 1, 2 \tag{3.33}$$

3.3.5.3 Coeficientes para el ecualizador 2-ASCET

Por último, el ecualizador 2-ASCET puede tratarse como un filtro FIR de cinco etapas y se puede definir como se muestra en (3.34).

$$E_m(e^{j\omega}) = \epsilon_{0,m}e^{j2\omega} + \epsilon_{1,m}e^{j\omega} + \epsilon_{2,m} + \epsilon_{3,m}e^{-j\omega} + \epsilon_{4,m}e^{-j2\omega} \tag{3.34}$$

Como en el caso anterior, se seleccionan cinco frecuencias que permiten resolver el sistema de ecuaciones para obtener los coeficientes del filtro. Las frecuencias seleccionadas en este caso son $\omega = 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi$ para las subportadoras pares y $\omega = -\pi, -\frac{3\pi}{4}, -\frac{\pi}{2}, -\frac{\pi}{4}, 0$ para las subportadoras impares. En (3.35) y (3.36) se recoge el sistema de ecuaciones para las subportadoras pares e impares, respectivamente, siendo $\alpha = \sqrt{2}/2(1+j)$ y $\beta = \sqrt{2}/2(1-j)$.

Subportadoras pares:

$$E_m(e^{j\omega}) = \left\{\begin{array}{rl} \epsilon_{0,m} + \epsilon_{1,m} + \epsilon_{2,m} + \epsilon_{3,m} + \epsilon_{4,m} = \eta_{0,m} & \text{para } \omega = 0 \\ j\epsilon_{0,m} + \alpha\epsilon_{1,m} + \epsilon_{2,m} + \beta\epsilon_{3,m} - j\epsilon_{4,m} = \eta_{1,m} & \text{para } \omega = \dfrac{\pi}{4} \\ -\epsilon_{0,m} - j\epsilon_{1,m} + \epsilon_{2,m} + j\epsilon_{3,m} - \epsilon_{4,m} = \eta_{2,m} & \text{para } \omega = \dfrac{\pi}{2} \\ -j\epsilon_{0,m} - \beta\epsilon_{1,m} + \epsilon_{2,m} - \alpha\epsilon_{3,m} + j\epsilon_{4,m} = \eta_{3,m} & \text{para } \omega = \dfrac{3\pi}{4} \\ \epsilon_{0,m} - \epsilon_{1,m} + \epsilon_{2,m} - \epsilon_{3,m} + \epsilon_{4,m} = \eta_{4,m} & \text{para } \omega = \pi \end{array}\right\} \tag{3.35}$$

Subportadoras impares:

$$E_m(e^{j\omega}) = \left\{ \begin{array}{rl} \epsilon_{0,m} - \epsilon_{1,m} + \epsilon_{2,m} - \epsilon_{3,m} + \epsilon_{4,m} = \eta_{0,m} & \text{para } \omega = -\pi \\ j\epsilon_{0,m} - \alpha\epsilon_{1,m} + \epsilon_{2,m} - \beta\epsilon_{3,m} - j\epsilon_{4,m} = \eta_{1,m} & \text{para } \omega = -\frac{3\pi}{4} \\ -\epsilon_{0,m} - j\epsilon_{1,m} + \epsilon_{2,m} + j\epsilon_{3,m} - \epsilon_{4,m} = \eta_{2,m} & \text{para } \omega = -\frac{\pi}{2} \\ -j\epsilon_{0,m} + \beta\epsilon_{1,m} + \epsilon_{2,m} + \alpha\epsilon_{3,m} + j\epsilon_{4,m} = \eta_{3,m} & \text{para } \omega = -\frac{\pi}{4} \\ \epsilon_{0,m} + \epsilon_{1,m} + \epsilon_{2,m} + \epsilon_{3,m} + \epsilon_{4,m} = \eta_{4,m} & \text{para } \omega = 0 \end{array} \right\} \quad (3.36)$$

Cada uno de los términos $\eta_{x,m}$ mencionados se definen como se muestra en (3.37).

$$\eta_{l,m} = \frac{H^*_{Ch}(e^{j(2k+l)\frac{\pi}{4M}})}{\left\| H_{Ch}(e^{j(2k+l)\frac{\pi}{4M}}) \right\|^2} \qquad \text{para } l = 0, 1, 2, 3, 4 \quad (3.37)$$

Resolviendo los sistemas de ecuaciones (3.35) y (3.36), se definen los coeficientes del filtro FIR para el 2-ASCET como se muestra en (3.38), (3.39), (3.40), (3.41) y (3.42), siendo de signo positivo las subportadoras pares y de signo negativo las subportadoras impares.

$$\begin{aligned} \epsilon_{0,m} = & \frac{1}{4}\Big[(\eta_{1,m} - 2\eta_{2,m} + \eta_{3,m}) + \frac{\sqrt{2}}{2}(-\eta_{0,m} + 2\eta_{1,m} - 2\eta_{2,m} + 2\eta_{3,m} - \eta_{4,m}) \\ & + j\left(-\eta_{1,m} + \eta_{3,m} + \frac{\sqrt{2}}{2}(\eta_{0,m} - \eta_{4,m}) \right) \Big] \end{aligned} \quad (3.38)$$

$$\begin{aligned} \epsilon_{1,m} = & \pm \frac{1}{4}\Big[(\eta_{0,m} - \eta_{4,m}) \\ & - j\left((\eta_{0,m} - 2\eta_{1,m} + 2\eta_{2,m} - 2\eta_{3,m} + \eta_{4,m}) \cdot \left(1 + \sqrt{2}\right) \right) \Big] \end{aligned} \quad (3.39)$$

$$\begin{aligned} \epsilon_{2,m} = & \frac{1}{2}\Big[(\eta_{0,m} - \eta_{1,m} + 2\eta_{2,m} - \eta_{3,m} + \eta_{4,m}) \\ & + \frac{\sqrt{2}}{2}(\eta_{0,m} - 2\eta_{1,m} + 2\eta_{2,m} - 2\eta_{3,m} + \eta_{4,m}) \Big] \end{aligned} \quad (3.40)$$

$$\begin{aligned} \epsilon_{3,m} = & \pm \frac{1}{4}\Big[(\eta_{0,m} - \eta_{4,m}) \\ & + j\left((\eta_{0,m} - 2\eta_{1,m} + 2\eta_{2,m} - 2\eta_{3,m} + \eta_{4,m}) \cdot \left(1 + \sqrt{2}\right) \right) \Big] \end{aligned} \quad (3.41)$$

$$\begin{aligned} \epsilon_{4,m} = & \frac{1}{4}\Big[(\eta_{1,m} - 2\eta_{2,m} + \eta_{3,m}) + \frac{\sqrt{2}}{2}(-\eta_{0,m} + 2\eta_{1,m} - 2\eta_{2,m} + 2\eta_{3,m} - \eta_{4,m}) \\ & - j\left(-\eta_{1,m} + \eta_{3,m} + \frac{\sqrt{2}}{2}(\eta_{0,m} - \eta_{4,m}) \right) \Big] \end{aligned} \quad (3.42)$$

Por último, cabe destacar que cada uno de los $\eta_{x,m}$ mostrados anteriormente son complejos, y por tanto, cada uno de los filtros (3.38), (3.39), (3.40), (3.41) y (3.42) se pueden descomponer como coeficientes complejos como se muestra en (3.43); siendo $c_{l,m}$ la parte real y $d_{l,m}$ la parte imaginaria, que se corresponden con los coeficientes del filtro FIR del CMFB y del SMFB, respectivamente.

$$\epsilon_{l,m} = c_{l,m} - jd_{l,m} \qquad \text{para } l = 0, 1, 2, 3, 4 \tag{3.43}$$

3.4 Subdivisión modular

En los apartados 3.2 y 3.3 se ha detallado matemáticamente cada uno de los módulos que componen el transmisor y receptor FBMC, así como la técnica de estimación y ecualización de canal utilizada. En este apartado se van a tratar las restricciones temporales y número de subportadoras que tiene el sistema FBMC, así como el planteamiento de la arquitectura mixta HW/SW, definiendo qué módulos se implementarán en *hardware* y *software*.

3.4.1 Restricciones del sistema FBMC

En relación a las restricciones temporales del sistema, el estándar IEEE 1901/2010 [IEE10] fija la frecuencia de muestreo en $f_s = 62.5\,\text{MHz}$; esta frecuencia f_s será un parámetro de diseño. Por otro lado, el número de subportadoras total seleccionado para el sistema FBMC es de $M = 512$ subportadoras. El estándar FFT 1901/2010 [IEE10] propone usar un número mayor de subportadoras (1024 y 2048) de forma opcional, sin embargo esto conlleva escalar los módulos para un número de puntos superior, manteniendo la misma frecuencia de muestreo de cada subportadora f_s.

Dada la frecuencia de muestreo f_s para las $M = 512$ subportadoras, la frecuencia para cada subportadora será de $f_C = 122.07\,\text{kHz}$. Es decir, cada uno de los módulos relacionados con el sistema FBMC mostrados en los apartados 3.2 y 3.3 deberá procesar los datos de entrada con esta frecuencia f_C. Dicho de otro modo, cada módulo deberá procesar las $M = 512$ subportadoras con una periodo máximo de $8.192\,\mu\text{s}$, que es la inversa de f_C. En [IEE10, Tabla 14-12] se indica que se utiliza este parámetro como longitud del símbolo.

Sin embargo, los módulos encargados de llevar a cabo la estimación de canal y cálculo de coeficientes para la ecualización no son tan exigentes desde el punto de vista temporal, ya que la estimación debe realizarse antes de superar el tiempo de coherencia del canal $t_{C_{PLC}}$ que se fija a $600\,\mu\text{s}$. Por tanto, en la etapa de recepción se llevará a cabo la estimación de canal y cálculo de coeficientes con una frecuencia menor que la frecuencia necesaria para realizar la demodulación de las subportadoras.

3.4.2 Propuesta de particionado HW/SW

Estas restricciones temporales, junto con la complejidad matemática inherente del sistema FBMC, hacen que este algoritmo presente una elevada carga computacional. Con el objeto de soportar dicha carga computacional, los sistemas basados en FPGA ofrecen una alternativa mediante la implementación de aceleradores *hardware*. Además, proporcionan a la arquitectura del sistema flexibilidad a costa de consumir los recursos específicos del dispositivo FPGA.

La tendencia actual de fabricación de los dispositivos integrados es incluir la mayor parte de los módulos que componen un sistema electrónico en un mismo integrado o chip. Estos dispositivos, conocidos como SoC, a diferencia de los microcontroladores, presentan una arquitectura más compleja. En este sentido, su complejidad se debe a las arquitecturas multinúcleo, donde más de un procesador se encuentra en un mismo chip, y en ocasiones, pueden integrar también unidades de cálculo específicas o GPU (*Graphic Process Unit*).

La familia de MPSoC (*Multi-Processor System-on-Chip*) Zynq® de Xilinx [Bop15] integra en el SoC una FPGA, lo que permite beneficiarse de las características de ambos. La combinación de FPGA junto con un potente microprocesador permite desarrollar arquitecturas heterogéneas, donde las tareas *software* las puede llevar a cabo el procesador o procesadores (dependiendo del dispositivo), y las tareas más exigentes desde el punto de vista computacional se llevan a cabo en la parte *hardware* (FPGA).

El particionado HW/SW permite realizar un reparto de las diferentes tareas entre la parte *hardware* (FPGA) y la parte *software* (procesador). Los módulos que tengan unas exigentes restricciones temporales son idóneos para llevar su implementación en la parte *hardware*, así como aquellos que presenten cierta regularidad en su algoritmo. En contraposición, aquellos módulos que no tengan restricciones temporales tan exigentes, pueden llevarse a cabo en la parte *software*.

Otro aspecto a considerar es el intercambio de información entre la parte *hardware* y la parte *software*, así como la sincronización entre ambos. En ambos casos se introducen penalizaciones temporales adicionales. En cuanto al intercambio de información, los MPSoC Zynq® de Xilinx disponen de diferentes mecanismos que soportan el intercambio de información, considerando el correspondiente consumo de recursos *hardware* en la FPGA. Por tanto, para definir la arquitectura heterogénea, se van a considerar los siguientes criterios para determinar el particionado HW/SW.

- Uno de los objetivos es consumir el menor número de recursos lógicos utilizados en la FPGA. Sin embargo, la complejidad que presenta el sistema FBMC, sobre todo en la etapa de recepción, hace que el sistema global presente un alto consumo de recursos.

- Aquellos módulos que tengan exigentes restricciones temporales se implementarán en la parte *hardware* del dispositivo, con el objetivo de alcanzar dichas restricciones mediante la paralelización del algoritmo en la implementación sobre la FPGA.

- Los módulos con una menor exigencia temporal se implementarán en la parte *software*, siempre y cuando se alcancen dichas restricciones temporales.

3.4.3 Arquitectura propuesta

Con todo lo expuesto hasta ahora, se va a definir la arquitectura heterogénea para el sistema descrito. En la figura 3.9 se muestra un esquema de la arquitectura propuesta con la subdivisión realizada entre la parte *hardware* (PL) y *software* (PS) según los criterios enumerados previamente. Cabe destacar que, para acceder a la red eléctrica, es necesario hacer uso de los correspondientes *Analog Front End* (AFE). Por tanto, debe haber una etapa de adaptación entre los módulos mostrados hasta ahora y los AFE. Sin embargo, aunque estarán presentes en la parte *hardware*, no se van a tratar en este estudio.

Dadas las exigentes restricciones temporales del sistema FBMC, lo más apropiado es que los módulos relacionados con el CMFB y SMFB del transmisor y receptor FBMC se implementen en la parte *hardware*; esto permite paralelizar el algoritmo para alcanzar las restricciones temporales impuestas por el estándar IEEE 1901/2010 [IEE10].

Para no introducir una latencia adicional al sistema global, la etapa de ecualización se va a implementar también en la parte *hardware*. Esta etapa consiste en un filtro FIR de $2 \cdot L + 1$ etapas para las M subportadoras, lo que conlleva $M \cdot (2 \cdot L + 1)$ multiplicaciones. Sin embargo, se aplica paralelismo, segmentación o multiplexación para consumir el mínimo número de recursos y alcanzar las restricciones temporales.

Por otro lado, para obtener la estimación de canal se necesita realizar la FFT de ambas secuencias CSS_k recibidas, lo que conlleva un consumo de recursos *hardware* elevado para obtener el cálculo de la FFT. Sin embargo, la etapa de estimación de canal y cálculo de coeficientes no tienen restricciones temporales exigentes, dado que las secuencias CSS se reciben aproximadamente con una periodicidad igual al tiempo de coherencia del canal ($t_{C_{PLC}} \approx 600\,\mu s$). Esto hace posible que la implementación de la estimación de canal y cálculo de coeficientes se realice en paralelo al resto de módulos que compone el receptor FBMC en la parte *hardware*.

Para llevar a cabo la implementación del sistema en la parte *hardware* a partir de la descripción matemática de cada uno de los módulos del transmisor y receptor FBMC, se va a hacer uso del sintetizador de comportamiento Vivado HLS. Esta herramienta permite, a partir de la descripción matemática en lenguaje C++, generar los módulos en lenguaje de descripción *hardware*.

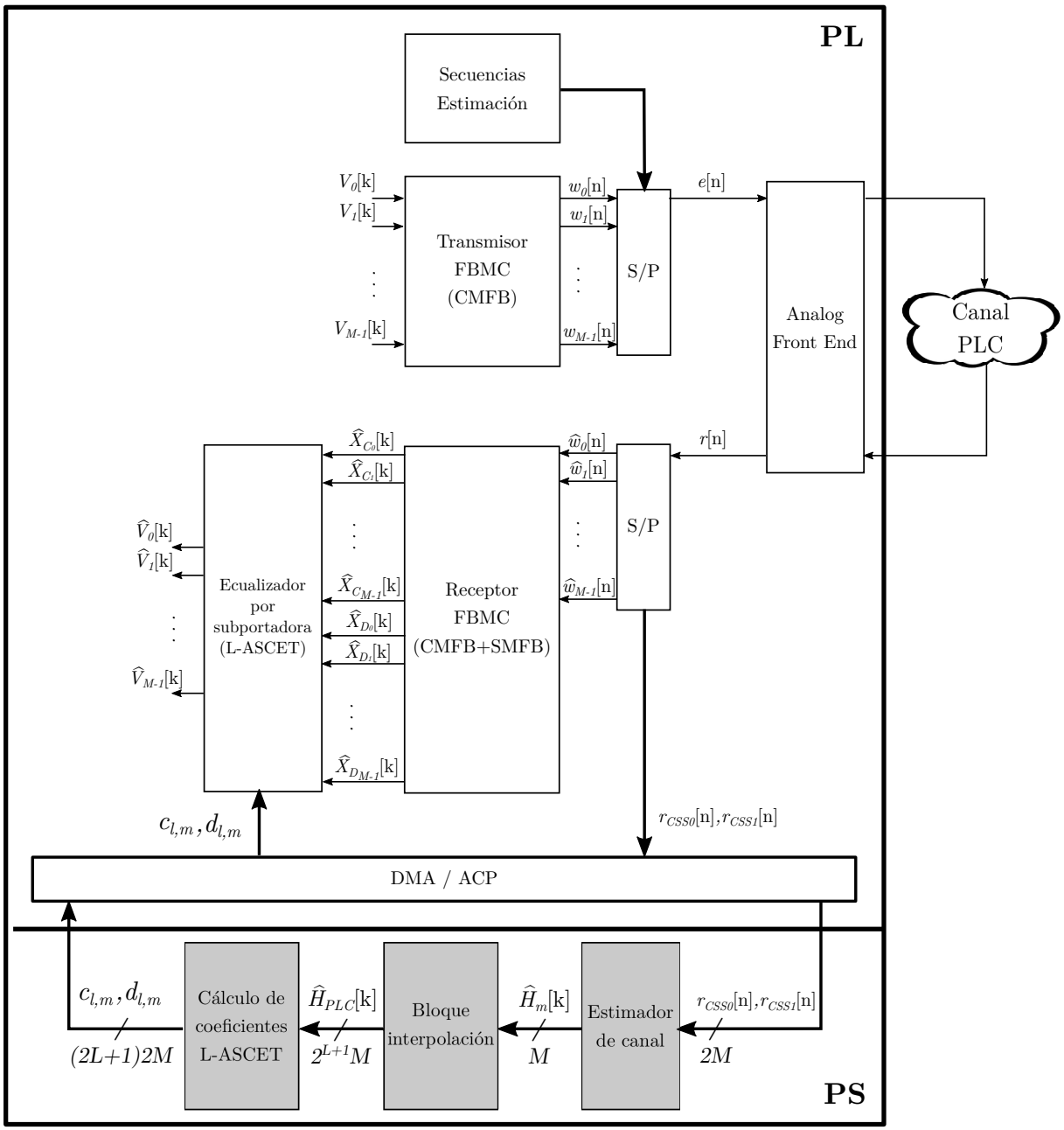


Figura 3.9: Esquema de la arquitectura heterogénea propuesta para el sistema FBMC, junto con la etapa de estimación y ecualización de canal.

En cuanto a la implementación *software* de la arquitectura, con el fin de reducir el tiempo de procesamiento de esta parte, se puede hacer uso de los módulos disponibles en el dispositivo Zynq®. En este sentido, tanto Zynq® 7000, como Zynq® UltraScale+, incorporan un procesador Cortex™ de la serie A, concretamente el Cortex™-A9 y Cortex™-A53, de doble y cuádruple núcleo respectivamente.

Ambos procesadores incorporan una unidad vectorial, denominada NEON, que permite acelerar ciertos cálculos como la FFT, necesaria para la estimación de canal. Además, dichos procesadores son multinúcleo, lo que va a permitir paralelizar en los núcleos disponibles el algoritmo mostrado, teniendo en cuenta que se deberá hacer uso de algún tipo de algoritmo de sincronización entre núcleos.

Por último, para llevar a cabo el intercambio de información entre la parte *hardware* y la parte *software*, lo más conveniente es usar transferencias de datos a través de DMA (*Direct Memory Access*). Además, para mantener en todo momento la coherencia de caché en el dispositivo, se va a hacer uso del puerto ACP (Accelerator Coherency Port).

3.5 Conclusiones parciales

En este capítulo se ha llevado a cabo la descripción matemática de la arquitectura del sistema FBMC. Los bloques que más carga computacional requieren son aquellos que precisan calcular la FFT, entre ellos se encuentra el módulo *DCT4e*, *DST4e* y *Estimador de canal*. La carga computacional de la FFT es considerable frente a otros módulos del sistema FBMC.

Por otro lado, el receptor FBMC incluye el ecualizador L-ASCET, para el cual se debe duplicar el banco de filtros teniendo el CMFB y SMFB en paralelo. En este caso, se han fusionado ambos bancos de filtros compartiendo las partes comunes con el fin de disponer de una arquitectura más eficiente. Además, en la etapa de recepción se incluye también el ecualizador L-ASCET junto con la etapa de estimación de canal y cálculo de coeficientes.

Para llevar a cabo la estimación de canal, en el transmisor se deben almacenar las secuencias que se envían por el canal y permiten obtener la estimación en la etapa de recepción. El receptor en el módulo *Estimador de canal* realiza la FFT de los símbolos recibidos y la posterior estimación LS. A partir de la estimación de canal se calculan los coeficientes del igualador L-ASCET. Como se ha visto en el capítulo, la complejidad requerida en el cálculo de coeficientes aumenta en función del orden L del ecualizador ASCET.

Dada la complejidad matemática que presenta, así como las restricciones temporales que se imponen por parte del estándar PLC, se ha optado por proponer una arquitectura heterogénea, que permita llevar a cabo el algoritmo con dichas restricciones de la forma más eficiente posible. Para ello se va a tratar que el consumo de recursos de la parte *hardware* sea eficiente, y por otro lado, se va a explotar todos los recursos disponibles de la parte *software* del SoC.

En el capítulo 4 se describirá la implementación de los módulos para la parte *hardware* haciendo uso del sintetizador de comportamiento Vivado HLS. Para la parte *software*, se detallará en el capítulo 5 el desarrollo de los módulos destinados a esta parte. Para ello, se hará uso de los recursos disponibles en el procesador con el fin de obtener el máximo rendimiento a la hora de obtener la estimación de canal y cálculo de coeficientes.

Capítulo 4

Definición de aceleradores *hardware* para comunicaciones PLC utilizando HLS

4.1 Introducción

A partir de la descripción matemática que se ha hecho en el capítulo 3, se debe realizar la implementación *hardware* de los distintos módulos de la arquitectura mixta HW/SW propuesta. A la hora de llevar a cabo la implementación sobre un dispositivo FPGA, se utilizan lenguajes de descripción hardware que proporcionan al diseñador cierto nivel de abstracción. De este modo, a partir de la descripción de comportamiento, en VHDL o Verilog, el sintetizador genera el código RTL (*Register-Transfer Level*). Con este diseño en RTL es posible obtener el sistema combinacional que permita programar la FPGA a partir del diseño en VHDL o Verilog.

Las nuevas herramientas de síntesis permiten obtener el diseño RTL de una descripción de comportamiento realizada en lenguajes de más alto nivel, como C, C++ o SystemC, proporcionando un nivel de abstracción más al diseñador. Por tanto, a partir de la descripción de comportamiento en C, C++ o SystemC, el sintetizador genera el código VHDL o Verilog y posteriormente la síntesis a RTL.

Para llevar a cabo el modelado HLS, se debe tener en cuenta el dispositivo que se va a utilizar para realizar un correcto dimensionado del *datapath*. Dado que la familia de MPSoC Zynq® de Xilinx [Bop15] integra en el SoC una FPGA, hace que estos dispositivos sean los seleccionados para el modelado de la arquitectura heterogénea propuesta en el capítulo 3. Sin embargo, existen pequeñas diferencias dentro de los dispositivos de la familia Zynq® 7000 y Zynq® UltraScale+, que se detallarán en los próximos apartados, concretamente en el 4.2.

El capítulo está organizado como se especifica a continuación; en el apartado 4.2 se mostrará el estudio para el dimensionado del *datapath* utilizado en el modelado HLS, así como el error debido al mismo. En el apartado 4.3 se detallará el modelado HLS llevado a cabo, mostrando la arquitectura para cada módulo descrito en el capítulo 3, así como los resultados obtenidos en términos de consumo de recursos *hardware*, latencia y *throughput*. Por último, se presentan una serie de conclusiones parciales en el apartado 4.11.

4.2 Estudio del dimensionado del *datapath*

A la hora de realizar el dimensionado del *datapath* hay que tener en cuenta las características del dispositivo; en este caso, se van a utilizar dos dispositivos de la familia Zynq® de Xilinx: Zynq® 7000 [Xil18c] y Zynq® UltraScale+ [Xil18e]. Entre ellas existen grandes diferencias con respecto a su arquitectura interna. Principalmente Zynq® UltraScale+ tiene mejores prestaciones con respecto a Zynq® 7000 ya que incluye un procesador ARM Cortex™-A53, en lugar del ARM Cortex™-A9 que incorpora Zynq® 7000. Además, el procesador que incluye Zynq® UltraScale+ tiene una unidad de procesamiento de tiempo real (RPU, *Real-Time Processing Unit*) para uso de sistemas en tiempo real, además de la unidad de procesamiento gráfico (GPU, *Graphic Process Unit*) cuya funcionalidad es accesible a través de librerías.

Por otro lado, los recursos *hardware* de la FPGA también presentan ligeras diferencias, como los bloques DSP (*Digital Signal Proccesors)*, que para llevar a cabo el dimensionado del *datapath* es necesario tener en cuenta las características de los componentes internos a utilizar. En este caso, para realizar el estudio del dimensionado del *datapath*, se van a tener en cuenta las características del modelo Zynq® 7000, haciendo referencia a los recursos *hardware* del mismo a lo largo del capítulo. Sin embargo este estudio se podría realizar y/o extender del mismo modo al modelo Zynq® UltraScale+ con las características de sus recursos *hardware*.

El objetivo de realizar el dimensionado del *datapath* es conseguir un uso eficiente de los elementos *hardware*, así como aprovechar al máximo aquellos recursos más escasos, como los bloques de memoria o bloques DSPs. El resto de elementos disponibles en la FPGA no son tan escasos, y por tanto, el *datapath* de los bloques de memoria y DSP fijará el *datapath* del sistema FBMC, con el fin de consumir el menor número de recursos de este tipo. A continuación, se procede a revisar cada uno de estos recursos *hardware*.

En cuanto a los bloques de memoria disponibles en la parte *hardware* de Zynq® 7000 [Xil19a], son bloques de memoria RAM *True Dual-Port* de 36-Kbit (RAM36B), y cuyo tamaño de palabra es de 32 bits. Por tanto, para hacer un uso eficiente de los bloques RAM36B, el *datapath* no deberá ser mayor al tamaño de palabra de los bloques de memoria. Cabe destacar que, si se precisara de un ancho de palabra mayor, los elementos de memoria RAM36B se podrían distribuir en cascada para aumentar el ancho de palabra o usar *Simple Dual Port*.

Por otro lado, los bloques DSP incluidos en la Zynq® 7000 son el modelo DSP48e1 [Xil18a], cuyo diagrama de bloques simplificado se muestra en la figura 4.1. En este diagrama de bloques, se puede observar que incluye diferentes elementos, entre ellos se encuentra un multiplicador de 25 bits por 18 bits con signo, el cual produce una salida de 43 bits con signo. A continuación del bloque multiplicador se encuentra una pequeña unidad aritmético-lógica (ALU, *Arithmetic Logic Unit*) con un comparador y un sumador, entre otros, de hasta 48 bits de salida. En una parte previa al bloque multiplicador, se encuentra un *pre-adder* con un ancho de palabra de 25 bits a la salida, que forma parte de una de las entradas del multiplicador. En este caso, el *datapath* que presentan los bloques multiplicadores DSP48e1 es inferior al ancho de palabra de los bloques RAM36B, y por tanto, al ser más restrictivo, hace que el dimensionado del *datapath* se ajuste según este factor.

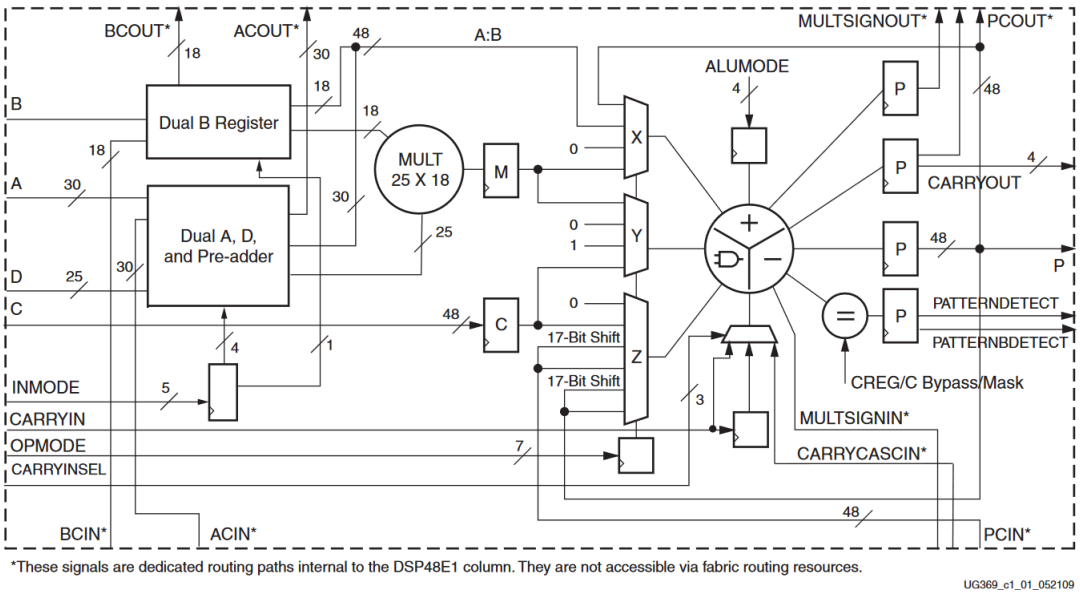


Figura 4.1: Diagrama de bloques simplificado de los bloques DSP48e1 de Zynq® 7000. [Xil18a]

Para hacer un uso eficiente de los bloques DSP disponibles, se va a tratar que cada multiplicación real emplee un único bloque DSP48e1. No obstante, esto dependerá del tipo de datos usados en el sistema FBMC, en cuyo caso, se va a considerar que las señales de entrada y salida al sistema FBMC serán números reales.

Sin embargo, las constantes usadas en alguno de los módulos, como el bloque de la FFT y DCT, no son reales, sino que tienen parte imaginaria. En este caso, para realizar una multiplicación entre un número real y otro complejo se precisan al menos dos bloques DSP. Del mismo modo, para llevar a cabo la multiplicación entre números complejos, el consumo de bloques DSP eficiente sería de tres, dada la simplificación realizada en la multiplicación compleja, mostrada en [Jac05] y detallada en (4.1), siendo $x = (a + bj)$ e $y = (c + dj)$.

$$x \cdot y = \begin{cases} \texttt{Parte Real} = d \cdot (a-b) + a \cdot (c-d) \\ \texttt{Parte Imaginaria} = c \cdot (a+b) - a \cdot (d-d) \end{cases} \tag{4.1}$$

Por tanto, el dimensionado del *datapath* se fijará en función del tamaño del bloque DSP. Es decir, de acuerdo con la descripción matemática que se ha mostrado en el capítulo 3, donde la mayor parte de las operaciones son multiplicaciones entre señales y constantes, se fijará el tamaño de *datapath* de acuerdo con el ancho de palabra de los bloques multiplicadores, de 25 bits y 18 bits. Hay que tener en cuenta que, de no ser así, a la hora de realizar una multiplicación real, el consumo de bloques DSP se verá incrementado, debido al aumento del ancho de la palabra.

Una vez fijado el tamaño del *datapath*, hay que determinar la representación que tendrán los datos. En este caso, se ha optado por la representación de punto fijo, la cual dedica un número finito de bits para representar la parte entera y otro para la parte fraccionaria. Dependiendo del ancho de palabra del *datapath*, y del número de bits dedicados a la parte fraccionaria, será posible realizar la representación de cualquier número real, asumiendo cierto error debido a la cuantificación aplicada.

Con el fin de comprender el efecto que conlleva el dimensionado del *datapath*, y la representación de punto fijo sobre el sistema, a continuación se realiza un estudio del error, donde se define el *datapath* para cada uno de los módulos.

4.2.1 Estudio del error debido al dimensionado

Dado que la principal limitación es el ancho de palabra de los bloques multiplicadores, para definir el *datapath* de cada uno de los módulos del sistema FBMC, es conveniente determinar el número de bits dedicados a la parte fraccionaria de forma adecuada según esta limitación, con el fin de obtener el mínimo error posible en la representación de los datos.

Por tanto, el ancho de palabra que se va a considerar es de 25 y 18 bits a la hora de realizar una multiplicación de acuerdo con el bloques DSP48e1. Dado que las multiplicaciones que se realizan son entre la señal de entrada y un coeficiente, es necesario determinar el rango dinámico de los coeficientes para determinar el número de bits más conveniente en cada caso. Para ello, en la tabla 4.1 se recogen los valores mínimos y máximos de cada uno de los coeficientes del sistema FBMC.

Como se muestra en la tabla 4.1, el rango dinámico que presentan los coeficientes no supera la unidad en ningún caso. Según se muestran en el estándar IEEE 1901/2010 [IEE10, Tabla 14-8], se va a considerar que los datos de entrada serán datos modulados mediante una modulación por amplitud de pulsos (PAM), donde se indica que cada portadora tendrá una longitud de símbolo de 1 bit para 2PAM, 2 bits para 4PAM, 3 bits para 8PAM, 4 bits para 16PAM y 5 bits para 32PAM. Según [IEE10, Tabla 14-8] cada símbolo de la

Módulo	Valor mínimo (Real)	Valor máximo (Real)	Valor mínimo (Imaginario)	Valor máximo (Imaginario)
Multiplicación θ	-1	1	-	-
DCT4e factor $\left(e^{\frac{-jm\pi}{2M}}\right)$	0.0031	1	-1	0
Coeficientes de la FFT	$-0.999\,99$	1	-1	0
DCT4e factor $\left(e^{\frac{-j(2m+1)\pi}{4M}}\right)$	9.59×10^{-5}	0.0625	-0.0625	-9.59×10^{-5}
Coeficiente $g_{s,0}$ del banco de filtros	-0.9362	0.0638	-	-
Coeficiente $g_{s,2}$ del banco de filtros	-0.0638	0.9362	-	-
Multiplicación constante $1/\sqrt{2}$	0.7071	0.7071	-	-

Tabla 4.1: Tabla de los valores mínimos y máximos de los coeficientes para cada uno de los módulos del sistema FBMC.

modulación presenta un determinado nivel de amplitud. En la tabla 4.2 se recogen los valores mínimos y máximos de amplitud de cada una de las modulaciones definidas en el estándar. En este caso, se van a considerar estos datos como parámetros de diseño, sin embargo es posible adaptar el dimensionado del *datapath* si las especificaciones lo requieren o cambiar el tipo de modulación del estándar PLC.

Tipo de modulación	Amplitud mínima	Amplitud máxima
2 PAM	-1	1
4 PAM	-1.3416	1.3416
8 PAM	-1.5275	1.5275
16 PAM	-1.627	1.627
32 PAM	-1.6787	1.6787

Tabla 4.2: Tabla de los valores mínimos y máximos de amplitud para los diferentes niveles de modulación PAM del estándar IEEE 1901/2010 para PLC.

La notación que se va a emplear para definir la cuantificación utilizada es Q[IWL,FWL], donde IWL representa el tamaño de palabra para la parte entera incluyendo el signo y FWL indica el tamaño de palabra dedicado a la parte fraccionaria.

Con todo lo anterior, lo más conveniente es utilizar el ancho de palabra de 18 bits del bloque DSP para los coeficientes, dado que los valores no superan la unidad en valor absoluto, y así, aplicando una cuantificación Q[1,17], donde el 1 representa la parte entera con signo y el 17 la parte fraccionaria, se alcanza una resolución de $2^{-17} = 7.63 \times 10^{-6}$, suficiente para representar los diferentes coeficientes con un rango de $\pm 0.\widehat{9}$. Por otro lado, el ancho de palabra de mayor tamaño del bloque DSP (25 bits) se destinará para el flujo de datos de entrada que recorre el sistema, proporcionando así una mayor resolución y flexibilidad al diseño final, ya que, pese a dedicar un mayor número de bits a la parte entera (por ejemplo, Q[5,20]), la parte fraccionaria tendría una resolución superior que la mostrada en el caso anterior.

Una vez se dispone del rango dinámico de las partes más generales del sistema, se debe realizar un dimensionado del *datapath* de modo que permita representar los valores de entrada, valores intermedios y valores de salida, sin que se produzca ningún tipo de desbordamiento (*overflow*). Para ello, se ha llevado a cabo un análisis matemático de la

diferentes módulos del sistema FBMC en punto fijo mediante la herramienta de cálculo matemático MATLAB®.

En este caso, para proporcionar mayor versatilidad al sistema y posteriormente poder evaluar otros tipos de modulaciones, los datos de entrada del transmisor FBMC son valores aleatorios con una amplitud de ±3 sin aplicar ninguna modulación. Estos datos presentan una amplitud ligeramente superior al rango de entrada definido por el estándar para las modulaciones PAM mostradas en la tabla 4.2. En la figura 4.2 se muestran los valores de amplitud de entrada de cada una de las subportadoras, en punto flotante (Patrón de Oro) y su representación en punto fijo (Punto Fijo) con una cuantificación Q[5,20]. Además, la figura 4.3 recoge el error que se obtiene debido a la cuantificación para las señales de entrada, que como se puede comprobar está por debajo de 1×10^{-6}.

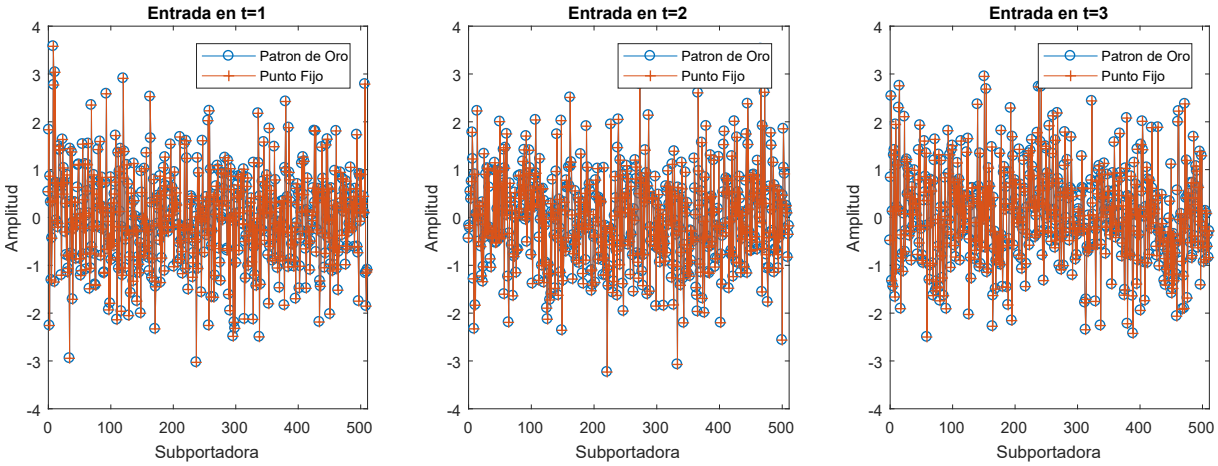


Figura 4.2: Valores de amplitud de una señal de entrada aleatoria en cada subportadora para tres instantes de tiempo diferentes, representados en punto flotante y punto fijo.

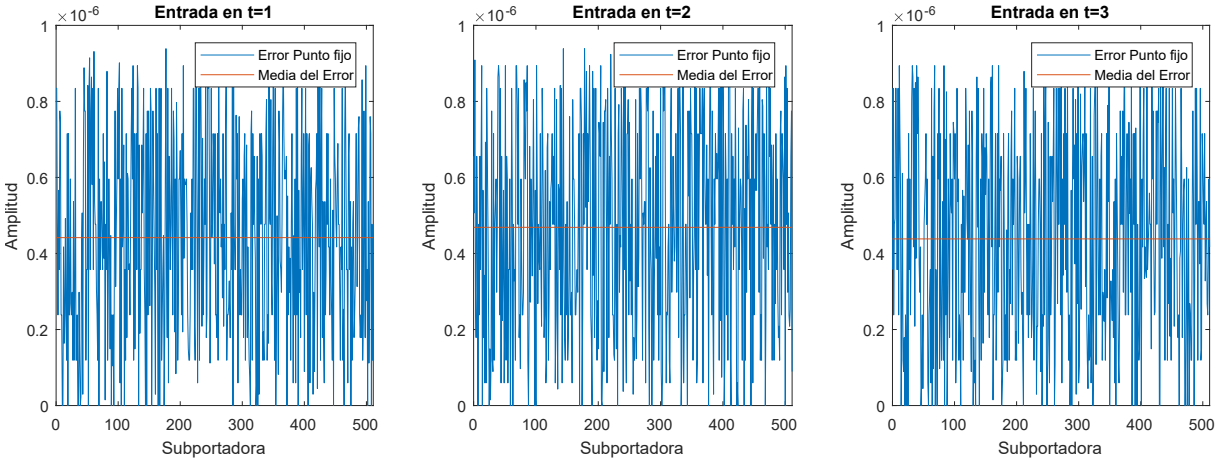


Figura 4.3: Error debido a la cuantificación en punto fijo de la señal de entrada en cada subportadora para tres instantes de tiempo diferentes.

Para el resto del dimensionado del *datapath* en cada uno de los módulos, se ha tenido en cuenta que los datos producidos tras una operación de multiplicación de dos elementos con un ancho de palabra de 25 y 18 bits generan un ancho de palabra resultante que

es la suma de los dos anchos anteriores, es decir, 43 bits en este caso. Por otro lado, la operación de suma generará un resultado con un ancho de palabra que será un bit más del mayor de los anchos de palabra para evitar el desbordamiento. Como se ha mencionado anteriormente, el DSP48e1 dispone de un *pre-adder* antes del multiplicador; si se hace uso de este *pre-adder*, hay que considerar que el resultado de la suma aumentará un bit más el *datapath* con respecto al ancho de palabra de entrada.

Debido a que las opciones de optimización de la herramienta de síntesis HLS y al ser un elemento interno del bloque DSP48e1, el uso del *pre-adder* se realiza siempre que sea posible, incrementando en una unidad el ancho de palabra de entrada. Esto provoca que, para realizar una multiplicación real de 26 bits por 18 bits, se requiera más de un bloque DSP48e1, lo que conlleva un uso ineficiente de los mismos. Por tanto, el dimensionado del *datapath* aplicado a las señales de entrada no será de 25 bits, sino de 24 bits para cada uno de los módulos, con el fin de evitar que se incremente el uso de bloques DSP.

En la tabla 4.3 se muestra el dimensionado utilizado para cada uno de los módulos del sistema FBMC y la implementación de la FFT. Este dimensionado se ha realizado por igual para cada uno de los módulos que comprende el transmisor y receptor FBMC, excepto para el módulo de la FFT donde se ha aplicado una cuantificación ligeramente diferente. Por tanto, la tabla 4.3 indica el tamaño para la parte entera y parte fraccionaria que se aplica a los datos de entrada, salida y coeficientes. Además, se muestra el dimensionado resultante tras la operación de suma y multiplicación.

	Tamaño de la parte entera	Tamaño de la parte decimal	Notación utilizada
Entrada - Módulos	4	2	Q[4,20]
Coeficientes - Módulos	2	16	Q[2,16]
Multiplicación - Módulos	7	36	Q[7,36]
Sumas- Módulos	5	20	Q[5,20]
Salida - Módulos	4	20	Q[4,20]
Entrada - FFT	8	16	Q[8,16]
Coeficientes - FFT	2	15	Q[2,15]
Multiplicación - FFT	12	31	Q[12,31]
Sumas - FFT	9	16	Q[9,16]
Salida - FFT	8	16	Q[8,16]

Tabla 4.3: Tabla con el dimensionado del *datapath* utilizado tras el estudio realizado.

Una vez mostrado el análisis realizado para seleccionar el dimensionado del *datapath* correctamente, se debe realizar el modelado *hardware* de cada uno de los módulos. Para ello, se ha optado por usar la herramienta de síntesis de alto nivel de Vivado (Vivado HLS) para realizar una descripción de cada uno de los módulos mediante lenguajes de alto nivel. En los siguientes apartados, se detallan los puntos clave del modelado HLS y la arquitectura de cada módulo.

4.3 Modelado de aceleradores *hardware* mediante Vivado HLS

La herramienta de síntesis de alto nivel (HLS, *High Level Shynthesis*) permite acelerar el tiempo de diseño *hardware*, transformando una descripción realizada en lenguajes de alto nivel como C, C++ o SystemC, en lenguajes de descripción *hardware* como VHDL o Verilog, y posteriormente llevando a cabo una implementación RTL (*Register-Transfer Level*) capaz de ser sintetizable en una FPGA de Xilinx.

Esto proporciona gran flexibilidad al diseñador en varios aspectos. Desde el punto de vista de programación, es posible incluir librerías propias, además de las librerías que incorpora la herramienta. De forma adicional, a la hora de validar los diseños realizados, es posible llevar a cabo una simulación C, teniendo la ventaja de la rapidez y flexibilidad que proporciona. A partir de la simulación descrita en C, la herramienta realiza una transformación para llevar a cabo una co-simulación RTL. En resumen, la principal ventaja de Vivado HLS es que se trata de un sintetizador de comportamiento, es decir, a partir de la descripción funcional permite obtener la descripción RTL tras la síntesis.

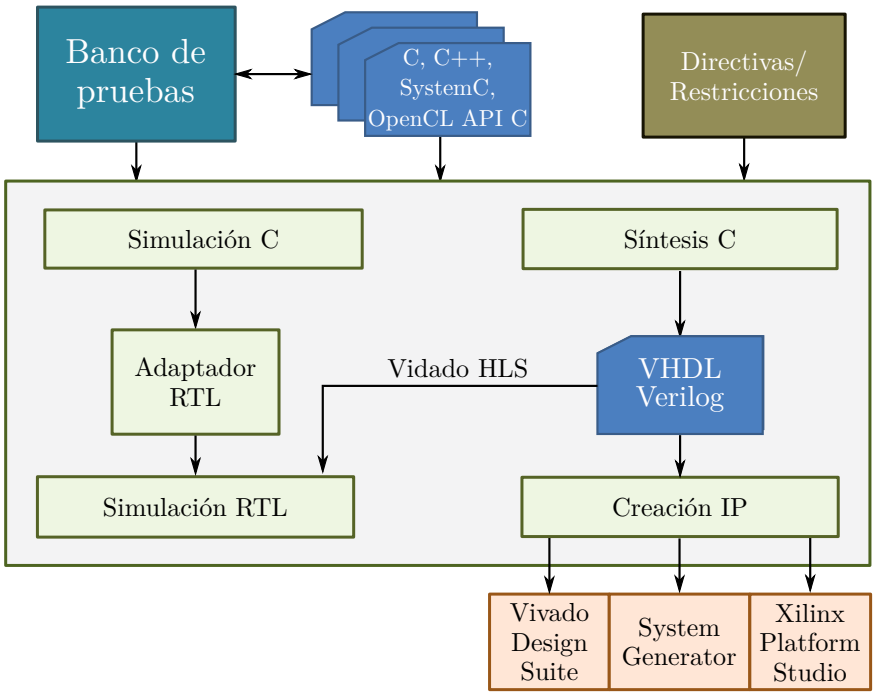


Figura 4.4: Diagrama de flujo para el modelado mediante la herramienta Vivado HLS [Viv16, Figura 4].

En la figura 4.4 se muestra un diagrama de flujo de la metodología de diseño con Vivado HLS. Durante el proceso de modelado HLS es posible explorar las diferentes soluciones del espacio de diseño mediante el uso de directivas. Estas directivas permiten definir que la herramienta use un determinado tipo de recursos *hardware* cuando realice la síntesis, así como aplicar segmentación, etc. Esto se consigue mediante el uso de pragmas en el código C, haciendo posible tener diferentes implementaciones *hardware* a partir de

una descripción en C, variando cada una de las directivas. Para evaluar el rendimiento del sistema, la herramienta proporciona unas métricas en términos de área, latencia o intervalo de inicialización. Estas métricas de rendimiento se definen a continuación:

- El área hace referencia al conjunto de recursos *hardware* disponibles en la FPGA requeridos para la implementación RTL del diseño. Esto incluye *look-up tables* (LUT), *Flip-Flop* (FF) o registros, bloques de memoria RAM y bloques DSP.
- Se define la latencia como el número de ciclos de reloj que precisa una función para procesar todos los datos a la salida. Cuando la salida produce vectores o matrices, la latencia se proporciona para el último valor de salida.
- El intervalo de inicialización (*Initiation Interval*, `II`) se refiere al número de ciclos de reloj necesarios para que la función pueda procesar una nueva muestra. Nótese que para un conjunto de datos, el `II` se referirá al número de ciclos necesarios para procesar un nuevo conjunto de datos.

La herramienta de Vivado HLS proporciona parámetros adicionales, como la latencia o el `II` de los bucles, que se definen igual que antes, pero se aplican únicamente a los bucles. Esto proporciona información adicional al diseñador para corroborar la arquitectura del sistema.

La aplicación de directivas tiene un impacto en el rendimiento del sistema que se refleja en la latencia, `II` y área utilizada. Aquellos sistemas con una reducida latencia y el mayor *throughput* (o menor `II`), utilizarán una mayor área que el que tenga una latencia mayor y menor *throughput*. Por tanto, mediante el uso de directivas se llegará al compromiso entre el área utilizada, latencia e `II`.

La descripción matemática del sistema FBMC se ha realizado mediante el lenguaje C++. La descripción está orientada al uso de bucles que permitan recorrer las M subportadoras. Para reducir la latencia y el `II` de estos bucles, se suelen utilizar las directivas `UNROLL` y `PIPELINE`. La primera de ellas permite desenredar un bucle para reducir la latencia, a costa de aumentar el área del diseño. La directiva `PIPELINE` permite reducir el `II` aplicando segmentación al bucle, ejecutando así la operación de lectura, ejecución y escritura en paralelo. Mediante la directiva `DATAFLOW` es posible aplicar segmentación a las diferentes funciones que componen el diseño.

Sin embargo, cada una de las directivas comentadas hasta ahora presentan un cuello de botella importante que va a impedir mejorar las prestaciones del sistema. Se trata del acceso a los datos, ya sea un acceso de lectura o escritura, al segmentar o desenredar el bucle, se produce más de un acceso de lectura o escritura de forma simultánea. Dado que los recursos *hardware* sólo permiten un único acceso de lectura o escritura simultáneo, se debe solventar este inconveniente dividiendo los *arrays* o memorias en alguna de sus dimensiones.

La directiva `ARRAY_PARTITION` permite dividir el *array* de tres formas. La primera es en bloques (*block*), donde los elementos del *array* original se encuentran de forma consecutiva dentro de cada uno de los bloques en los que se dividen. La segunda manera es de forma cíclica (*cyclic*), donde los elementos consecutivos del *array* original se reparten en cada uno de los bloques en los que se han dividido. Y por último, de forma completa (*complete*), donde cada elemento del *array* original se dispone de forma independiente.

Para poder realizar el modelado HLS del sistema FBMC y aplicar las directivas correctamente, es necesario tener en cuenta las restricciones del sistema mostradas en la sección 3.4.1. Principalmente el número de subportadoras del sistema, que en este caso se ha seleccionado $M = 512$, y la frecuencia de muestreo f_s definida en el estándar PLC (IEEE 1901/2010 Std.). Estos dos parámetros definen la cadencia de los módulos del sistema FBMC, que procesa las $M = 512$ subportadoras con un periodo máximo de 8.192 µs.

Con el fin de no sobrepasar el periodo máximo de 8.192 µs, se hará uso de los pragmas citados anteriormente. Además, con el objetivo de proporcionar cierta flexibilidad a la arquitectura, un parámetro de diseño será el número de subportadoras M y el Ratio de Paralelismo R_P, que será el número de *datapaths* que se procesan en paralelo en cada módulo del sistema FBMC de forma simultánea. Cabe destacar que alguno de los módulos tienen un R_P superior a la unidad por definición, como por ejemplo el caso del módulo *Matrices IJ* en el transmisor FBMC, el cual procesa dos *datapaths* de entrada y genera sólo un *datapath* de salida. En este caso, se va a considerar un valor de $R_P = 1$ como parámetro de diseño para toda la arquitectura, sin que afecte a los módulos con un R_P superior por definición.

Por otro lado, es necesario definir la frecuencia de operación de los módulos, que en este caso estará definida en la parte *hardware* (*Progammable Logic*, PL) del SoC, fijada en $f_{clk} = 100\,\text{MHz}$, lo que proporciona un periodo de reloj de 10 ns. Dicho periodo de reloj determina que cada módulo se debe procesar en al menos 819 ciclos de reloj, que es el resultado de dividir el periodo máximo (8.192 µs) entre el periodo de reloj definido para el sistema, redondeado al entero inferior. Por tanto, se puede determinar que el `II` máximo que se puede alcanzar es de 819 ciclos, siendo una métrica de diseño de cada módulo; en caso de superarlo se deberá segmentar para estar por debajo.

Por último, las restricciones de latencia no tendrán un impacto en el diseño, ya que la métrica de latencia sólo aporta una mejor definición de la arquitectura y el periodo de tiempo que pasa desde que se introduce un conjunto de muestras, hasta que se procesan y alcanzan la salida. Sin embargo, en la arquitectura mixta HW/SW propuesta es necesario tener en cuenta la latencia del receptor FBMC, que determinará el tiempo máximo de procesamiento de la estimación de canal y cálculo de coeficientes implementado en la parte *software* (PS, *Processing System*).

Con todos estos parámetros y restricciones de diseño, a continuación se muestra el modelado llevado a cabo para cada uno de los módulos del sistema FBMC. Para cada uno de ellos se mostrará la arquitectura utilizada, los resultados obtenidos en términos de latencia e `II`, así como los recursos *hardware* consumidos. También se describirán brevemente las directivas utilizadas para alcanzar las restricciones de diseño.

4.4 Modelado del *Multiplicador* θ y *Constante*

Estos módulos se caracterizan por sólo realizar la multiplicación de un valor por cada una de las subportadoras, o bien, un único valor constante por cada subportadora; tal y como se ha descrito en las secciones 3.2.1 y 3.2.5.

En la figura 4.5 se muestra la arquitectura del módulo *Multiplicador* θ. En este caso, la parte (a) y (b) de la figura 4.5 muestra la arquitectura del módulo multiplicador con un el valor de $R_P = 1$ y $R_P = 2$ respectivamente. Cabe destacar que para $R_P = 2$ admite más de una solución válida; la mostrada realiza un paginado de los datos y la otra posibilidad sería entrelazando los datos. En cualquiera de las dos soluciones, se puede comprobar como en la figura (b) el número de *datapaths* que se procesan es el doble, y como se verá a continuación, el consumo de recursos resultante es el doble. Para el caso del *Multiplicador por Constante*, bastaría con sustituir el bloque de memoria de los valores de θ por el valor de la constante.

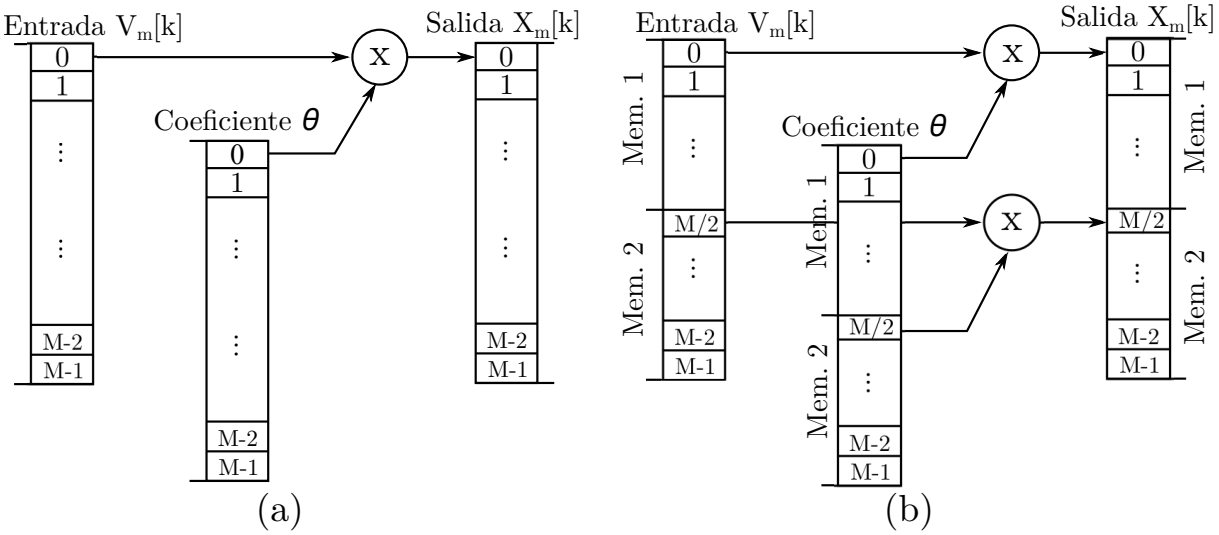


Figura 4.5: Arquitectura del módulo *Multiplicador* θ para un $R_P = 1$ en (a) y un $R_P = 2$ en (b).

Los pragmas o directivas utilizadas van a tener relación con el R_P seleccionado, que en este caso es $R_P = 1$. Sin embargo, aprovechando la poca complejidad que tiene el módulo, es posible mostrar el efecto que tiene aumentar el R_P, en este caso a $R_P = 2$. Para ello, se han aplicado las directivas referentes a memoria (`ARRAY_PARTITION`) y las referentes a la segmentación, como `PIPELINE` y `UNROLL`, para alcanzar el `II` deseado.

Como se ha mencionado anteriormente, es posible tener los datos paginados o entrelazados. En el caso de tener los datos entrelazados, el desenredado de los bucles se debe hacer mediante la directiva UNROLL, ya que la herramienta de síntesis HLS aplica el factor de desenredado del bucle a muestras consecutivas. En el caso del ejemplo mostrado en el código 4.1, se paraleliza el bucle con un factor determinado (R_P) para las muestras consecutivas del bucle.

```
void MultTheta(TxDataIn_t dataIn[M], TxDataOut_t dataOut[M])   {
        loop_MultTheta: for(int k = 0; k < M; k++)     {
#pragma HLS PIPELINE
#pragma HLS UNROLL factor = 2
                dataOut[k] = dataIn[k] * thetaCoef[k];
        }
}
```

Algoritmo 4.1: Código de la función *Multiplicador θ* aplicando la directiva UNROLL.

Aunque esta opción proporciona resultados válidos, no se corresponde a la arquitectura mostrada anteriormente en la figura 4.5, en la que los datos se encuentran paginados. Para que el diseño se corresponda con la arquitectura mostrada, se ha optado por anidar dos bucles; uno para el R_P, que se encontraría anidado dentro de otro global correspondiente a las subportadoras, y el otro, se le aplica la directiva de UNROLL al bucle que hace referencia al R_P. Este último paralelizaría el bucle interno, accediendo a los elementos de memoria con saltos proporcionales a R_P, que en este caso son de $M/2$ muestras. Desde el punto de vista de programación, sólo implica anidar dos bucles y aplicar la directiva UNROLL al bucle interno. A continuación en el código 4.2, se muestra el código de este desenrollado manual.

```
void MultTheta(TxDataIn_t dataIn[M], TxDataOut_t dataOut[M]){
        loop_MultTheta: for(int k = 0; k < M/RP; k++)    {
#pragma HLS PIPELINE
                loop_MultTheta_unroll: for(int i = 0; i < RP; i++)    {
#pragma HLS UNROLL
                        dataOut[k+(i*M/RP)] = dataIn[k+(i*M/RP)] * thetaCoef[k+(i*M/RP)];
                        }
        }
}
```

Algoritmo 4.2: Código de la función *Multiplicador θ* realizando el desenredado manual mediante un bucle anidado y la directiva UNROLL.

Esto permite que el código se sintetice de acuerdo a la arquitectura mostrada anteriormente en la figura 4.5, ya que a lo largo del proceso de modelado, y del sistema FBMC, no es posible acceder a elementos consecutivos de memoria. La otra directiva utilizada para alcanzar el II deseado es PIPELINE, la cual segmenta el bucle más externo, proporcionando una mejora en el II.

Habría que solventar el cuello de botella que se tiene en los accesos de lectura y escritura de cada uno de los datos al desenredar el bucle. Para ello, la directiva `ARRAY_PARTITION` realiza la división de las memorias de entrada y salida, como se ha descrito anteriormente. En este caso, lo más optimo sería realizar la división en bloques (*block*) de $^{M}/_{2}$ para hacer posible los accesos al elemento 0 y elemento $^{M}/_{2}$ de forma simultánea en un ciclo, es decir, el factor de particionado sería dos.

En la tabla 4.4 se muestran los recursos consumidos por el módulo *Multiplicador* θ y el *Multiplicador por constante*. Además, se muestran los valores de latencia e `II` que se obtienen en cada uno. Como se puede comprobar, el `II` obtenido es inferior al límite de 819 ciclos en cualquier caso.

	Latencia (ciclos)	`II` (ciclos)	BRAM	DSP	FF	LUT
Multiplicador θ	515	515	1	1	46	29
Multiplicador Constante	516	516	0	1	77	72

Tabla 4.4: Tabla de latencia, `II` y consumo de recursos de los módulos *Multiplicador* θ y *Multiplicador Constante*.

Por último, los valores de θ son coeficientes que toman el valor 1 o -1, es decir, la misma señal de entrada, o la señal de entrada cambiada de signo. Esto se podría describir mediante una comprobación previa (sentencia `if`) y realizar el cambio de signo correspondiente, sin ser necesario el uso de un bloque multiplicador. Sin embargo, al ser una multiplicación entre dos números reales, donde sólo se consumiría un bloque DSP, se considera poco relevante con respecto al consumo de recursos que tiene el sistema global. Por otro lado, el ratio R_P seleccionado en este módulo es $R_P = 1$; sin embargo, a la hora de incrementar el R_P del sistema, el número de bloques DSP necesarios se vería incrementado y, en tal caso, sería conveniente prescindir de usar los bloques DSP en este módulo e implementarlo mediante una comprobación.

4.5 Modelado de la DCT y DST

El módulo de la DCT se divide en cuatro etapas según se ha mostrado en la sección 3.2.2. Además, como se ha mencionado anteriormente en la sección 3.3.3, la forma de obtener la DST se basa en aplicar un pre-procesado y post-procesado a la DCT descrita. En este sentido, se va a proceder a explicar las características de la arquitectura implementada con la DCT, y una vez descrita, se detallarán las características de los módulos de pre-procesado y post-procesado para obtener la DST.

A continuación se van a describir cada una de estas etapas de la DCT. En primer lugar se detallará la arquitectura de la primera y última etapa, continuando con el módulo de la FFT y la etapa de reordenamiento. Por último, para la descripción del módulo DST, se centrará únicamente en los módulos de pre-procesado y post-procesado, ya que el resto de módulos son los mismos que los descritos para DCT.

4.5.1 Primera y última etapa de la DCT

La primera etapa de la DCT consiste en la multiplicación de la señal de entrada por un coeficiente complejo. En este caso, la arquitectura es idéntica a la explicada en la sección anterior 4.4, salvo por el coeficiente, que en este caso es complejo.

Para llevar a cabo la implementación de la multiplicación compleja, se ha optado por utilizar el algoritmo mostrado en [Jac05], el cual hace uso del mínimo número de bloques DSP, o lo que es lo mismo, el mínimo número de multiplicaciones posible. La descripción matemática de esta simplificación se ha detallado en la ecuación (4.1).

En la figura 4.6 se muestra la arquitectura que tendría la implementación de la multiplicación compleja. Esta arquitectura estaría incluida en cada una de las etapas de la DCT donde se lleve a cabo una multiplicación compleja. Tanto la primera como la última etapa incluyen esa multiplicación, pero también se realizan multiplicaciones complejas a lo largo de las diferentes etapas de la FFT, como se mostrará a continuación.

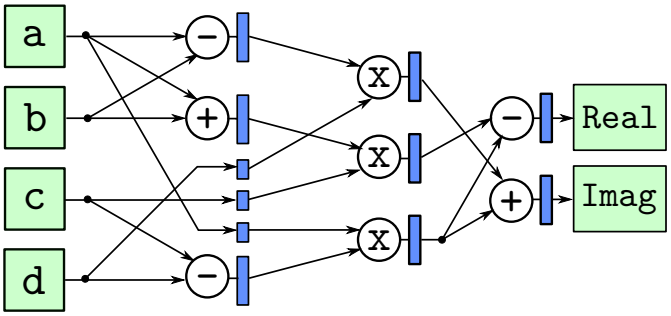


Figura 4.6: Arquitectura eficiente para la implementación de multiplicaciones complejas [Jac05].

4.5.2 Etapa FFT

La implementación directa y paralela de la FFT se considera costosa desde el punto de vista computacional. Además, son muchos los trabajos previos [Ber69] que proponen formas de obtener la FFT de forma eficiente y rápida. En la figura 4.7 se muestra el diagrama de mariposa de la FFT de ocho puntos ($M = 8$), donde cada mariposa presenta una estructura fija o regular y se entrelaza de forma diferente en cada una de las etapas de la FFT. El hecho de que los cruces de las mariposas no se realicen de forma regular entre cada etapa hace que sea difícil reutilizar los recursos empleados en una etapa y la siguiente.

En este caso, dadas las características de la herramienta de síntesis Vivado HLS, se ha utilizado una estructura de FFT que realiza los cruces de las mariposas de forma regular entre cada una de las etapas; esta estructura se encuentra descrita en trabajos previos [Mar15, Mem17]. Esto permite que cada una de las etapas de la FFT se replique con la misma estructura, siendo más fácil y cómodo de implementar dentro de un bucle. En la figura 4.8 se muestra un esquema de la FFT de ocho puntos con la estructura con cruces regulares.

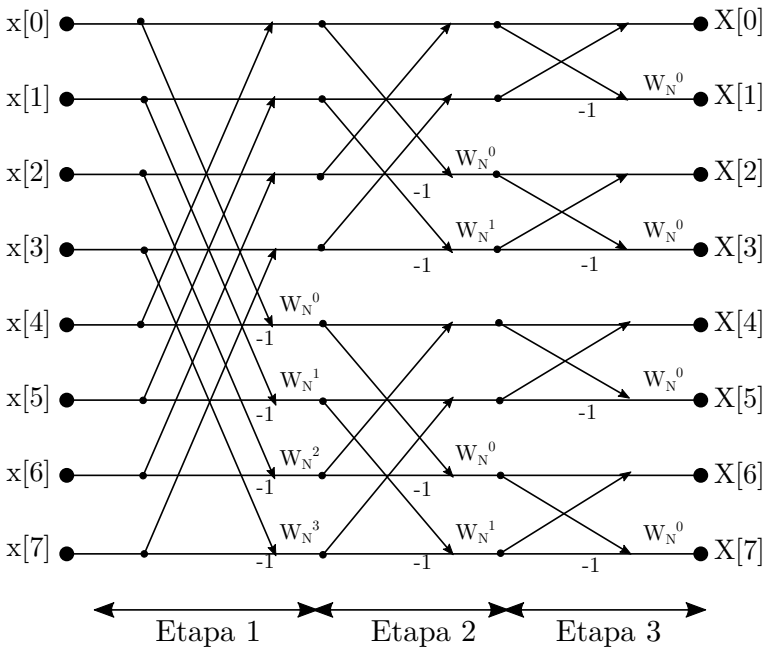


Figura 4.7: Diagrama de mariposa para el cálculo de la FFT de ocho puntos.

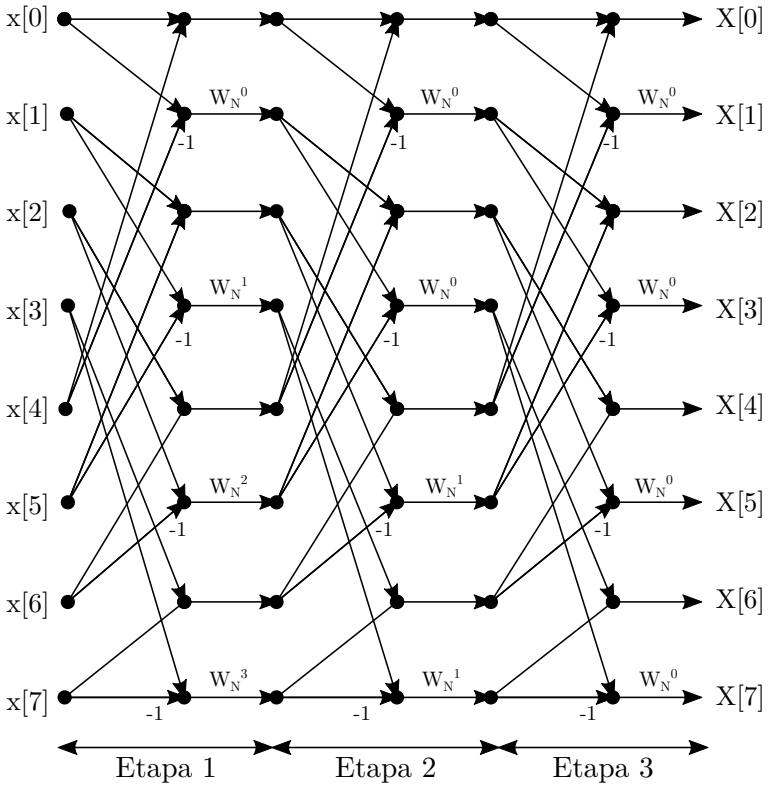


Figura 4.8: Diagrama para el cálculo de la FFT de ocho puntos con cruces regulares.

Esta estructura regular permite realizar la implementación de la FFT con un número fijo de iteraciones en cada etapa del bucle de la FFT, manteniendo la estructura regular en cada una de las etapas. Para llevar a cabo la multiplicación compleja en cada una de las etapas de la FFT, se realiza del mismo modo que se ha descrito para las etapas que precede a la etapa de la FFT.

Por un lado, se ha implementado un bucle para cada una de las etapas de la FFT, que incluye otro bucle anidado para los cruces que se producen en las etapas de la FFT. La directiva `UNROLL` permite desenredar completamente el bucle que conforma cada una de las etapas de la FFT, para alcanzar las restricciones de tiempo real impuestas. Cabe destacar que el número de etapas de la FFT está definido como el $\log_2(M)$, siendo M el número de puntos de la FFT, que para la aplicación es $M = 512$, lo que supone nueve etapas. En la figura 4.9 se muestra un diagrama de bloques para las nueve etapas que componen la FFT de 512 puntos. Dentro de cada uno de los bucles de los cruces de cada una de las etapas, se hace uso de la directiva `PIPELINE` con el fin de segmentar el bucle y alcanzar las restricciones del sistema.

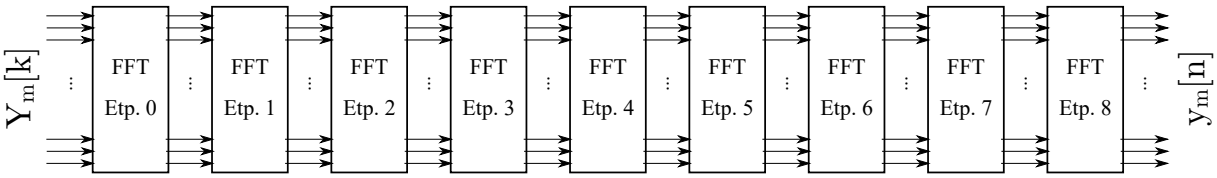


Figura 4.9: Desenredado de las etapas de la FFT para un número de puntos $M = 512$.

4.5.3 Etapa de *Cambio de Orden*

Por último, la etapa de *Cambio de Orden* que forma parte de la DCT posee una arquitectura más sencilla que las vistas hasta ahora. En la figura 4.10 se muestra la arquitectura de esta etapa, donde se aplica el conjugado a las $^{M}/_{2}$ últimas muestras de $y_m[n]$ y se entrelazan a la salida $z_m[n]$, produciéndose el cambio de orden. Para obtener el conjugado, se realiza únicamente el cambio de signo de la parte imaginaria, representado por $-(imag)$.

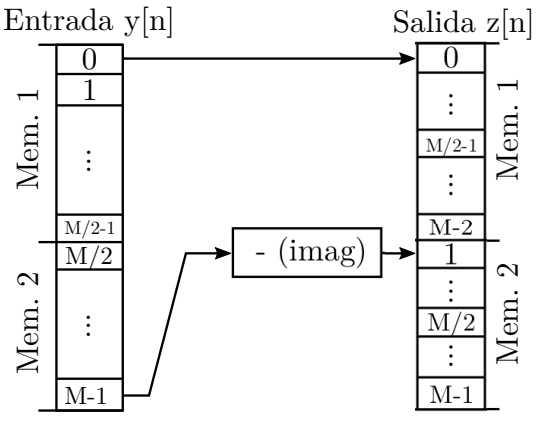


Figura 4.10: Arquitectura implementada para la etapa de reordenamiento de la DCT4e.

Para alcanzar las restricciones de tiempo real, se hace uso de las directivas referentes a memoria (`ARRAY_PARTITION`), así como las referentes a `PIPELINE` y `UNROLL`, para alcanzar la latencia e `II` deseados. En este caso, para evitar el cuello de botella en los datos de salida, donde se realizan dos accesos de escritura a las muestras consecutivas, se realiza un particionado de la memoria de tipo cíclico (*cyclic*), repartiendo las muestras consecutivas entre las diferentes memorias, que en este caso son dos.

Una vez analizadas las arquitecturas de cada una de las etapas de la DCT, se procederá a mostrar el consumo de recursos del conjunto global de todas las etapas en la tabla 4.5, así como la latencia y el `II`.

	Latencia (ciclos)	`II` (ciclos)	BRAM	DSP	FF	LUT
Etapa Multiplicación $\left(e^{\frac{-jm\pi}{2M}}\right)$	516	516	2	2	111	38
Etapa 0 de la FFT	260	260	2	3	168	185
Etapa 1 de la FFT	260	260	2	3	181	358
Etapa 2 de la FFT	260	260	2	3	181	358
Etapa 3 de la FFT	260	260	2	3	181	358
Etapa 4 de la FFT	260	260	2	3	181	358
Etapa 5 de la FFT	260	260	2	3	181	358
Etapa 6 de la FFT	260	260	2	3	181	358
Etapa 7 de la FFT	260	260	2	3	180	357
Etapa 8 de la FFT	260	260	3	2	147	280
Etapa *Cambio de Orden*	258	258	0	0	24	72
Etapa Multiplicación $\left(e^{\frac{-j(2m+1)\pi}{4M}}\right)$	516	516	2	2	127	112

Tabla 4.5: Tabla de latencia, `II` y consumo de recursos de los módulos involucrados en el cálculo de la DCT4e.

4.5.4 Cálculo de la DST

Para el caso de la DST, se tiene la arquitectura de la DCT descrita anteriormente, a la cual se le añade un bloque previo y otro posterior. En cuanto al bloque que le precede, presenta una arquitectura como se muestra en la figura 4.11. En ella se puede comprobar que la arquitectura no denota dificultad, simplemente se le aplica un cambio de signo a las subportadoras impares. En este caso, los datos de entrada están entrelazados y para ello se utiliza la directiva `ARRAY_PARTITION` de forma cíclica con el objetivo de separar las muestras pares e impares.

Para la última de las etapas de la DST, se realiza un reordenamiento de las señales. Es decir, la salida se obtiene invirtiendo el orden de las subportadoras. La arquitectura se muestra en la figura 4.12. Cabe destacar que en la figura los datos de entrada como los de salida están entrelazados, haciendo uso de la directiva `ARRAY_PARTITION` de forma cíclica. Sin embargo, es posible utilizar la misma directiva de tipo bloque (*block*) realizando accesos al primer y último elemento, en lugar de elementos consecutivos. Por otro lado, para alcanzar la latencia e `II` se utiliza también la directiva `PIPELINE`. Téngase en

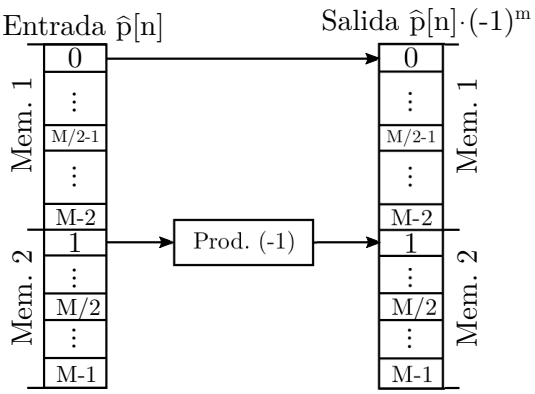


Figura 4.11: Arquitectura de la etapa previa de acondicionamiento para el cálculo de la DST4e, a partir de la DCT4e.

cuenta que existen diferentes maneras de plantear esta arquitectura, y todas pueden ser válidas y cumplir las restricciones temporales.

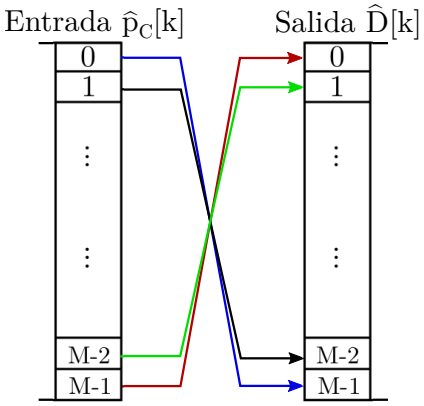


Figura 4.12: Arquitectura de la etapa posterior de acondicionamiento para el cálculo de la DST4e, a partir de la DCT4e.

Por último, el consumo de recursos que tienen estas dos etapas de la DST se muestra en la tabla 4.6, junto con los valores de latencia e `II` obtenidos. Para conocer el consumo total que tiene la DST, habría que añadir los recursos mostrados en la tabla 4.5 obtenidos para la DCT4e.

	Latencia (ciclos)	`II` (ciclos)	BRAM	DSP	FF	LUT
Etapa de acondicionamiento previa	256	256	0	0	30	126
Etapa de acondicionamiento posterior	514	514	0	0	20	38

Tabla 4.6: Tabla de latencia, `II` y consumo de recursos de los módulos involucrados en la adaptación de la DST4e.

Las etapas mostradas para el cálculo de la DST4e a partir de la DCT4e sólo introducen una latencia adicional. Esta latencia se podría evitar si se fusionan las etapas de acondicionamiento de la DST con la primera y última etapa de la DCT.

4.6 Modelado de *Matrices IJ*

El módulo *Matrices IJ* presenta ligeras diferencias entre el transmisor y el receptor, aunque son complementarias entre sí. Las diferencias más notables se deben a que en la etapa de recepción se realiza la fusión de ambos bancos de filtros, CMFB y SMFB. Además, en la etapa de transmisión la descripción matemática incluye un retardo que en la etapa de recepción se encuentra en el banco de filtros. A continuación se detalla el modelado de *Matrices IJ* para el transmisor y receptor.

4.6.1 Módulo *Matrices IJ* en el transmisor

La arquitectura que tiene el módulo *matrices IJ* en la etapa de transmisión se muestra en la figura 4.13. En ella se puede comprobar que el conjunto de M señales de entrada produce un conjunto de $S = 2 \cdot M$ señales de salida, con el retardo de una unidad para las últimas M señales de salida.

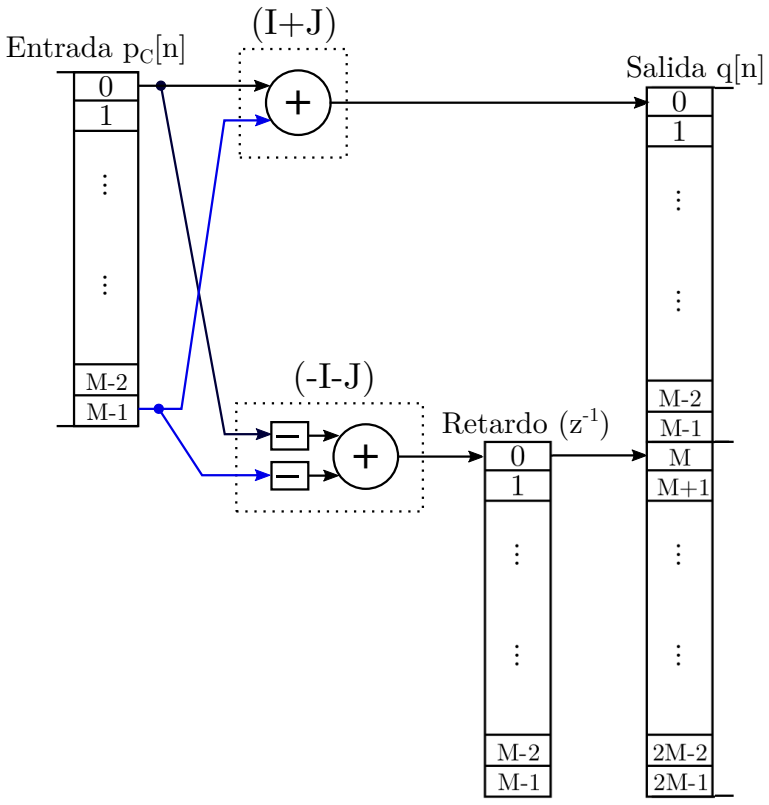


Figura 4.13: Arquitectura implementada para el módulo *Matrices IJ* del transmisor.

Cabe destacar que, aplicando la descripción matemática para un $R_P = 1$, se procesan dos señales de entrada, produciendo a su vez dos señales de salida. Como se muestra en la figura 4.13, las dos señales de entrada se paralelizan y se procesan en dos bloques, $(\mathbf{I} + \mathbf{J})$ y $(-\mathbf{I} - \mathbf{J})$, independientes para generar las S señales de salida.

Para alcanzar las restricciones temporales, se utilizan las directivas `ARRAY PARTITION` para evitar el cuello de botella en la memoria de entrada y salida, en este caso de tipo bloque. Además, la directiva de `PIPELINE` permite segmentar el bucle y reducir el `II`.

4.6.2 Módulo *Matrices IJ* en el Receptor

En cuanto a la arquitectura del módulo *Matrices IJ* del receptor, se muestra en la figura 4.14. Nótese que en este caso el número de señales de entrada al módulo es $S = 2 \cdot M$ y se producen dos conjuntos de M señales a la salida, que se corresponden con el banco de filtros del coseno $\hat{p}_{C_m}$ y seno $\hat{p}_{D_m}$, respectivamente.

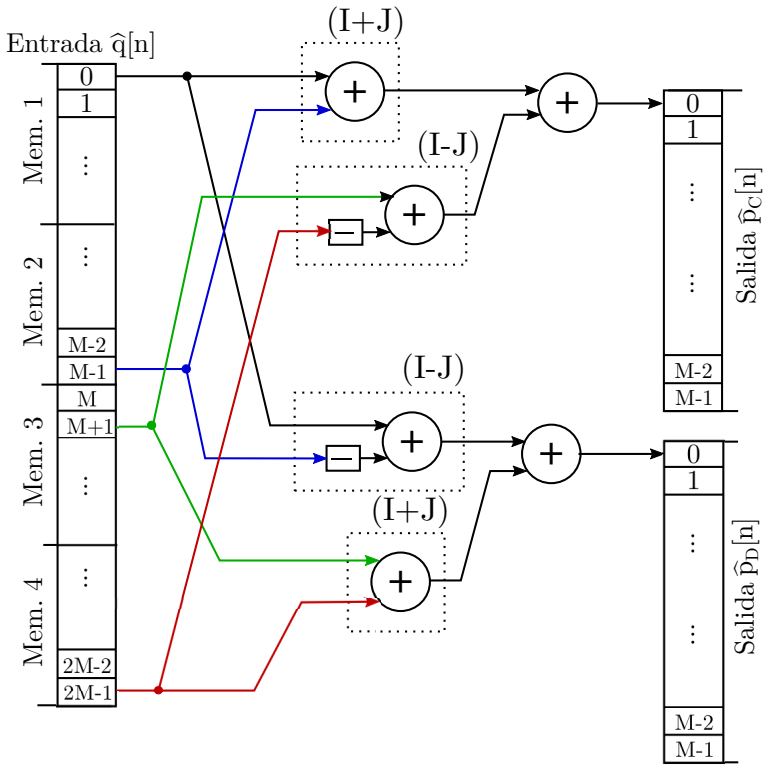


Figura 4.14: Arquitectura implementada para el módulo *Matrices IJ* del receptor.

En este caso, para un $R_P = 1$ el número de señales de entrada que se procesan en paralelo es dos; sin embargo éstas se desagregan obteniendo hasta cuatro señales, dos para el banco de filtros de coseno y otras dos para el seno. Según la descripción matemática, las M señales de salida de cada banco de filtros se obtienen a partir de la combinación de las señales de entrada por las matrices identidad e intercambio, generando cuatro señales que se suman dos a dos para obtener la salida para cada uno de los filtros.

Dado que las descripciones del banco de filtros del coseno y el seno son complementarias, se puede observar en la figura 4.14 que la combinación de las señales de entrada para obtener $\hat{p}_{C_m}$ es complementaria a la combinación para obtener $\hat{p}_{D_m}$.

Al igual que en el transmisor, se utilizan la directiva `ARRAY_PARTITION` de tipo bloque para la memoria, y evitar así el cuello de botella que pueden producir los accesos de lectura y escritura. Además, la directiva `PIPELINE` permite alcanzar el `II` deseado.

En la tabla 4.7, se muestra el consumo de recursos obtenido en este módulo *Matrices IJ* para el transmisor y el receptor, así como la latencia e `II`. El mayor consumo de recursos se da en bloques de memoria y, al no producirse ningún tipo de multiplicación en este módulo, el consumo de bloques DSP es nulo.

	Latencia (ciclos)	`II` (ciclos)	BRAM	DSP	FF	LUT
Módulo *Matrices IJ* del transmisor	259	259	2	0	93	167
Módulo *Matrices IJ* del receptor	515	515	0	0	150	289

Tabla 4.7: Tabla de latencia, `II` y consumo de recursos de los módulos *Matrices IJ* de la etapa de transmisión y recepción.

4.7 Modelado del *Banco de Filtros*

Como ocurre con el módulo *Matrices IJ*, la etapa de transmisión y la etapa de recepción del *banco de filtros* presentan una arquitectura ligeramente diferente, sin embargo entre ellas son complementarias. La principal diferencia radica en que el banco de filtros de la etapa de transmisión está precedido del módulo *Matrices IJ*, el cual genera $S = 2 \cdot M$ señales de salida; mientras que en el receptor, el módulo *Matrices IJ* se encuentra después, siendo el banco de filtros el que genera las S señales. Esto hace que el número de señales de entrada que procesa el banco de filtros sea de S para la etapa de transmisión y en la etapa de recepción sea de M señales desagregadas.

4.7.1 Módulo *Banco de Filtros* en el Transmisor

En la figura 4.15 se muestra la arquitectura que presenta el banco de filtros en la etapa de transmisión. De acuerdo con la descripción matemática, las M primeras señales del conjunto de $S = 2 \cdot M$ señales de entrada se procesan en el banco de filtros en paralelo a las M siguientes señales de entrada. El resultado obtenido de ambos se suma para producir el resultado de las M señales del banco de filtros.

La arquitectura que tiene este módulo se corresponde con una estructura FIR. Esta estructura (G_m y G_{m+M} en la figura) no se replica para las S señales de entrada, sino que recorre la memoria de entrada.

Como ocurría anteriormente, para evitar el cuello de botella que producen los accesos de lectura o escritura en memoria se recurre al uso de la directiva `ARRAY_PARTITION`. Para alcanzar el `II` deseado en este módulo se utiliza la directiva de `PIPELINE`.

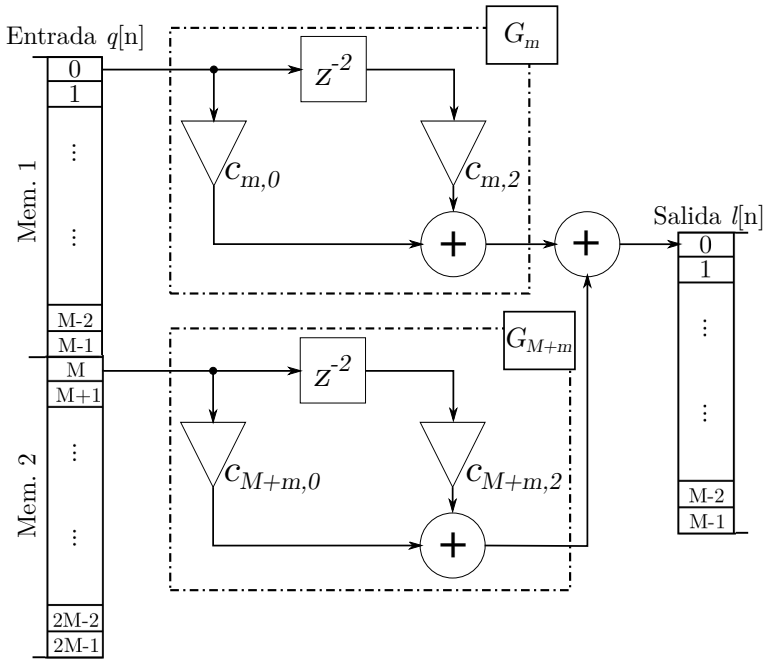


Figura 4.15: Arquitectura para la implementación del módulo *Banco de Filtros* en el transmisor.

Por último, en la tabla 4.8 se muestra el consumo de recursos que tiene este módulo, así como la latencia e `II`. Las multiplicaciones llevadas a cabo son productos entre números reales, lo que hace que el consumo de DSP sea de un único DSP por cada multiplicación. Teniendo en cuenta que el número de multiplicaciones llevadas a cabo es de cuatro, coincide con el consumo de recursos obtenidos.

	Latencia (ciclos)	`II` (ciclos)	BRAM	DSP	FF	LUT
Banco de Filtros del transmisor	516	516	8	4	289	110

Tabla 4.8: Tabla de latencia, `II` y consumo de recursos del módulo *Banco de Filtros* del transmisor.

4.7.1.1 Módulo *Banco de Filtros* en el Receptor

En la etapa de recepción se tienen M señales de entrada y tras pasar el banco de filtros se producen $S = 2 \cdot M$ señales a la salida. En este caso, la arquitectura del banco de filtros no presenta una gran diferencia con respecto a la mostrada para el transmisor. Sin embargo, como se muestra en la figura 4.16, dado que el conjunto de M señales de entrada se aplica a ambos subconjuntos de M coeficientes del total $S = 2 \cdot M$ que compone el banco de filtros, se utiliza sólo un bloque de retardo de dos unidades para ambos, manteniendo así la estructura FIR.

En este caso, la directiva `ARRAY_PARTITION` se aplica a la memoria de salida del módulo, donde se producen las escrituras para el elemento m y $(M + m)$ de forma simultánea. En la entrada sólo se produce un único acceso de lectura al elemento m, por lo que no es necesario. La directiva `PIPELINE` se aplica de la misma forma que se ha venido haciendo hasta ahora.

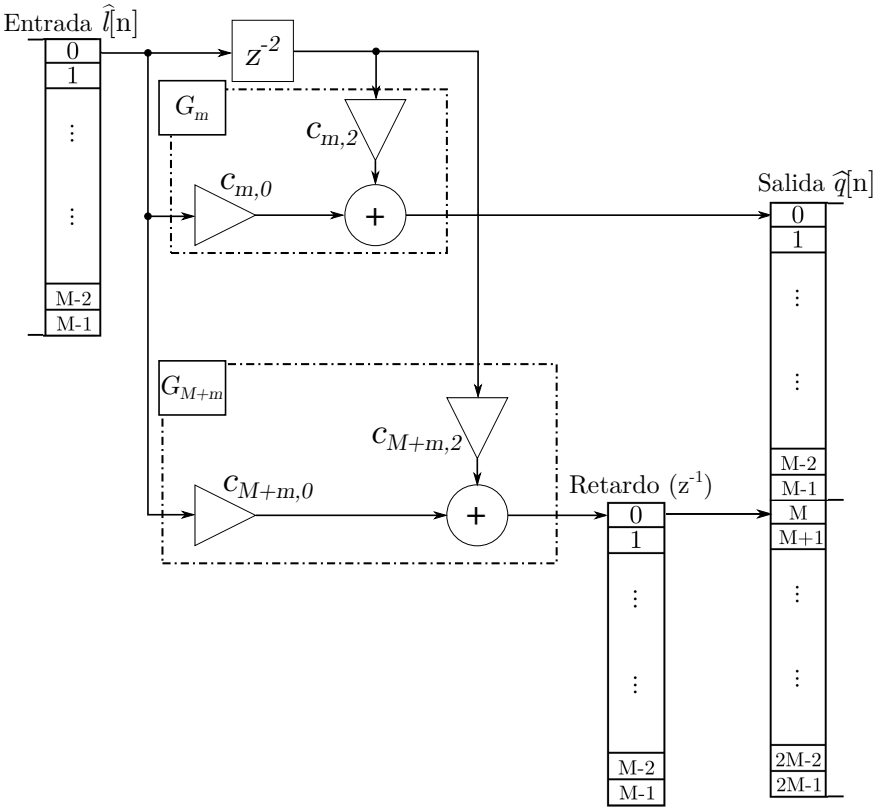


Figura 4.16: Arquitectura para la implementación del módulo *Banco de Filtros* en el receptor.

La tabla 4.9 muestra el consumo de recursos, así como la latencia e `II`. Igual que antes, los productos son entre números reales y se producen el mismo número de multiplicaciones reales que en el transmisor, coincidiendo con el consumo de DSP obtenido.

	Latencia (ciclos)	`II` (ciclos)	BRAM	DSP	FF	LUT
Banco de Filtros del receptor	514	512	5	4	219	38

Tabla 4.9: Tabla de latencia, `II` y consumo de recursos del módulo *Banco de Filtros* del receptor.

Cabe destacar que para un valor de R_P superior, el efecto que tendría en la arquitectura sería únicamente duplicar los módulos mostrados anteriormente, es decir, se procesarían en paralelo los filtros G_m, $G_{m+M/2}$, G_{m+M} y $G_{m+3\cdot M/2}$, con el consumo de recursos que esto conlleva.

4.8 Modelado del igualador de canal ASCET

La arquitectura del igualador ASCET es muy similar a la del banco de filtros, ya que se basa en una arquitectura FIR. Sin embargo, el número de coeficientes que tiene el igualador de canal L-ASCET depende del orden L del mismo, siendo el número total de coeficientes de $2 \cdot L + 1$. En la figura 4.17 se presenta la arquitectura que tiene el modelado del ecualizador de forma general en función de L.

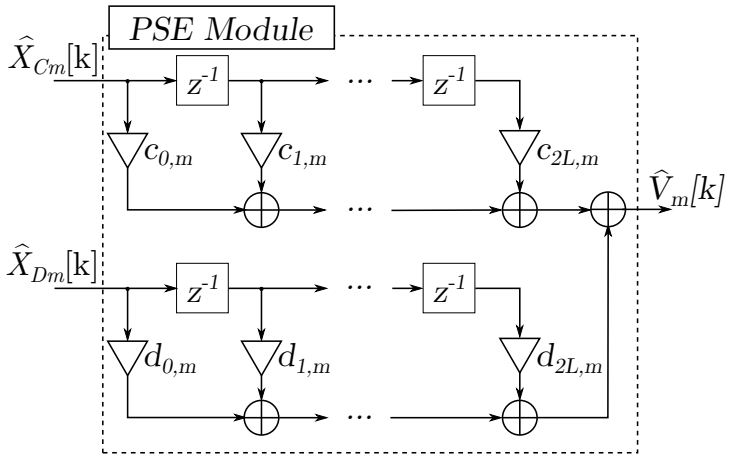


Figura 4.17: Estructura del igualador de canal L-ASCET basada en filtros FIR.

Este igualador realiza únicamente multiplicaciones reales, por tanto el número de bloques DSP que utiliza es uno por cada multiplicación real. Sin embargo, al tener dos arquitecturas FIR en paralelo, una para la salida generada por el CMFB y otra por el SMFB, el consumo de multiplicadores en cada etapa del filtro es el doble.

En la tabla 4.10 se muestra el consumo total de recursos, latencia e `II` de este módulo en función del orden L del ASCET utilizado. Tal y como se indica en [Pin14], para órdenes del igualador superiores a dos, el impacto que tiene, en términos de SNR en el sistema, no es significativo, y sí es la carga computacional necesaria para el cálculo de los coeficientes para órdenes superiores, ya que el número de coeficientes del igualador es superior.

	Latencia (ciclos)	`II` (ciclos)	BRAM	DSP	FF	LUT	Slice
Ecualizador PSE 0-ASCET	515	512	0	2	87	74	33
Ecualizador PSE 1-ASCET	515	512	4	6	155	210	94
Ecualizador PSE 2-ASCET	516	512	8	10	258	300	137

Tabla 4.10: Tabla de latencia, `II` y consumo de recursos del módulo de ecualizador PSE para los diferentes órdenes de L-ASCET.

4.9 Modelado del estimador de canal PLC

Para realizar la igualación de canal es necesario tener un modelo de canal, o una estimación del mismo. En el capítulo 3 se describe la manera de calcular la estimación de canal. Dado que la estimación de canal se basa en el cálculo de dos FFT, una para cada uno de los símbolos recibidos, se pretende liberar recursos *hardware*, de modo que el cálculo de la estimación de canal y de los coeficientes del igualador se va a realizar en la parte *software*, reduciendo así el alto consumo de recursos *hardware* que tiene la implementación de la FFT. Sin embargo, el envío de los símbolos piloto por el canal PLC está implementado en *hardware* y se muestra a continuación.

Para ello, estos símbolos piloto se almacenan en una memoria, de modo que antes de superar el tiempo de coherencia de canal $t_{C_{PLC}}$, el transmisor realizará el envío de estos símbolos piloto por el canal leyendo los dos símbolos piloto de M muestras cada uno, que se encuentran almacenados en memoria. En la figura 4.18 se muestra el esquema de la arquitectura de esta parte. Para proporcionar más versatilidad al diseño, se ha establecido una comunicación *AXI-Lite*, que permite cambiar los símbolos piloto si las necesidades lo requieren, o bien, si los símbolos a transmitir cambian. De esta forma no es necesario generar un nuevo bloque, sino que se podría aprovechar el bus *AXI-Lite* para actualizar esa información.

Por tanto, el módulo *Secuencias estimación* tiene los símbolos piloto almacenados y una pequeña lógica de control que permite no superar el tiempo de coherencia del canal, además de controlar el banco CMFB del transmisor para no perder las muestras que se puedan generar mientras se realiza el envío de los símbolos piloto.

En la tabla 4.11 se muestra el consumo de recursos que tiene este módulo. En este caso no se ha utilizado la herramienta de síntesis de Vivado HLS, sino que se ha optado por realizar el diseño mediante VHDL y la herramienta de síntesis de Vivado.

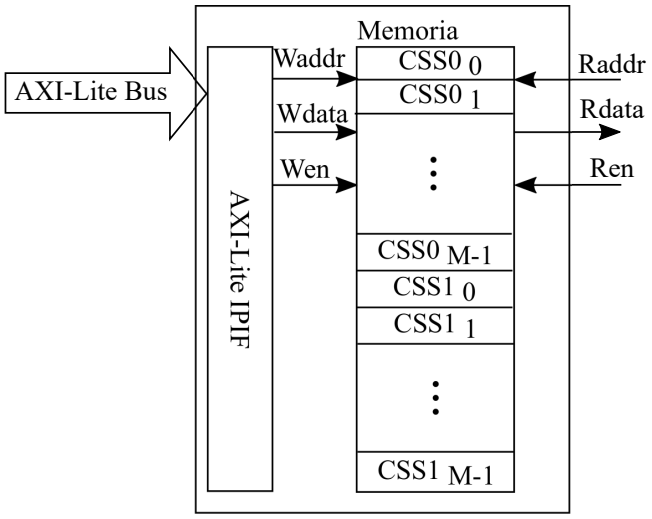


Figura 4.18: Arquitectura del módulo *Secuencias Estimación.*

	BRAM	DSP	FF	LUT
Secuencias Estimación	8	0	510	300

Tabla 4.11: Tabla del consumo de recursos del módulo *Secuencias estimación.*

4.10 Resultados del Transmultiplexor FBMC

Por último, en este apartado se recoge el resultado de consumo de recursos y latencias para todo el conjunto del transmultiplexor FBMC, incluyendo todos los módulos descritos anteriormente. Para integrar cada uno de los módulos anteriores se define una función global que incluye cada uno de los módulos definidos. Se hace uso de la directiva `DATAFLOW`,

la cual realiza un pipeline a nivel de función de los módulos definidos anteriormente, reduciendo el `II` del sistema.

En la tabla 4.12 se muestra el consumo de recursos obtenido por la herramienta de síntesis de Vivado HLS, entre paréntesis se muestra el porcentaje de utilización respecto al total de recursos disponibles en la tarjeta de desarrollo ZC706. La etapa de transmisión no incluye el módulo *Secuencias estimación*; sólo se refiere al CMFB transmisor y en la etapa de recepción sólo están los bancos de filtros CMFB y SMFB, sin tener en cuenta el igualador.

	BRAM (18K)	DSP48E1	FF	LUT	Latencia	`II`
Transmisor FBMC	185 (16.97 %)	36 (4.00 %)	1950 (0.45 %)	8001 (3.66 %)	4957	515
Receptor FBMC (CMFB + SMFB)	362 (33.21 %)	67 (7.40 %)	3102 (0.71 %)	17877 (8.18 %)	5498	512

Tabla 4.12: Tabla de latencia, `II` y consumo de recursos de los módulos de transmisión y recepción implementado en Vivado HLS.

En la tabla 4.13 se incluye el consumo de recursos tras la síntesis y la implementación en la herramienta de Vivado, donde se tienen en cuenta todos los periféricos adicionales a la arquitectura, como controlador DMA, interconexión de los bloques y lógica adicional. En este caso sí que se encuentra el módulo *Secuencias Estimación* y el ecualizador del receptor, como se muestra en la figura 4.19. Al igual que antes, entre paréntesis se muestra el porcentaje de recursos utilizado respecto al total total disponibles en la tarjeta de desarrollo ZC706.

Modulo	Slice LUTs	Slice Registers	Block RAM Tile	DSPs
Transmisor	8066 (3.69 %)	2082 (0.48 %)	93,5 (17.16 %)	36 (4.00 %)
Receptor (0-ASCET)	19770 (9.04 %)	5725 (1.31 %)	194 (35.60 %)	71 (7.89 %)
Receptor (1-ASCET)	19924 (9.11 %)	5795 (1.33 %)	198 (36.33 %)	79 (8.78 %)
Receptor (2-ASCET)	20019 (9.16 %)	5899 (1.35 %)	202 (37.06 %)	87 (9.67 %)

Tabla 4.13: Tabla de latencia, `II` y consumo de recursos para la etapa de transmisión y recepción.

4.11 Conclusiones parciales

A lo largo del capítulo se ha presentado la arquitectura para los diferentes módulos del sistema FBMC mediante el modelado con Vivado HLS. Para ello, se ha definido el dimensionado del *datapath* en función de las características del dispositivo que se está utilizando. Se ha mostrado el estudio del error debido a la cuantificación utilizada, determinando el *datapath* para cada uno de los módulos. El *datapath* utilizado introduce el mínimo error, pero además se ajusta a las restricciones del dispositivo utilizado.

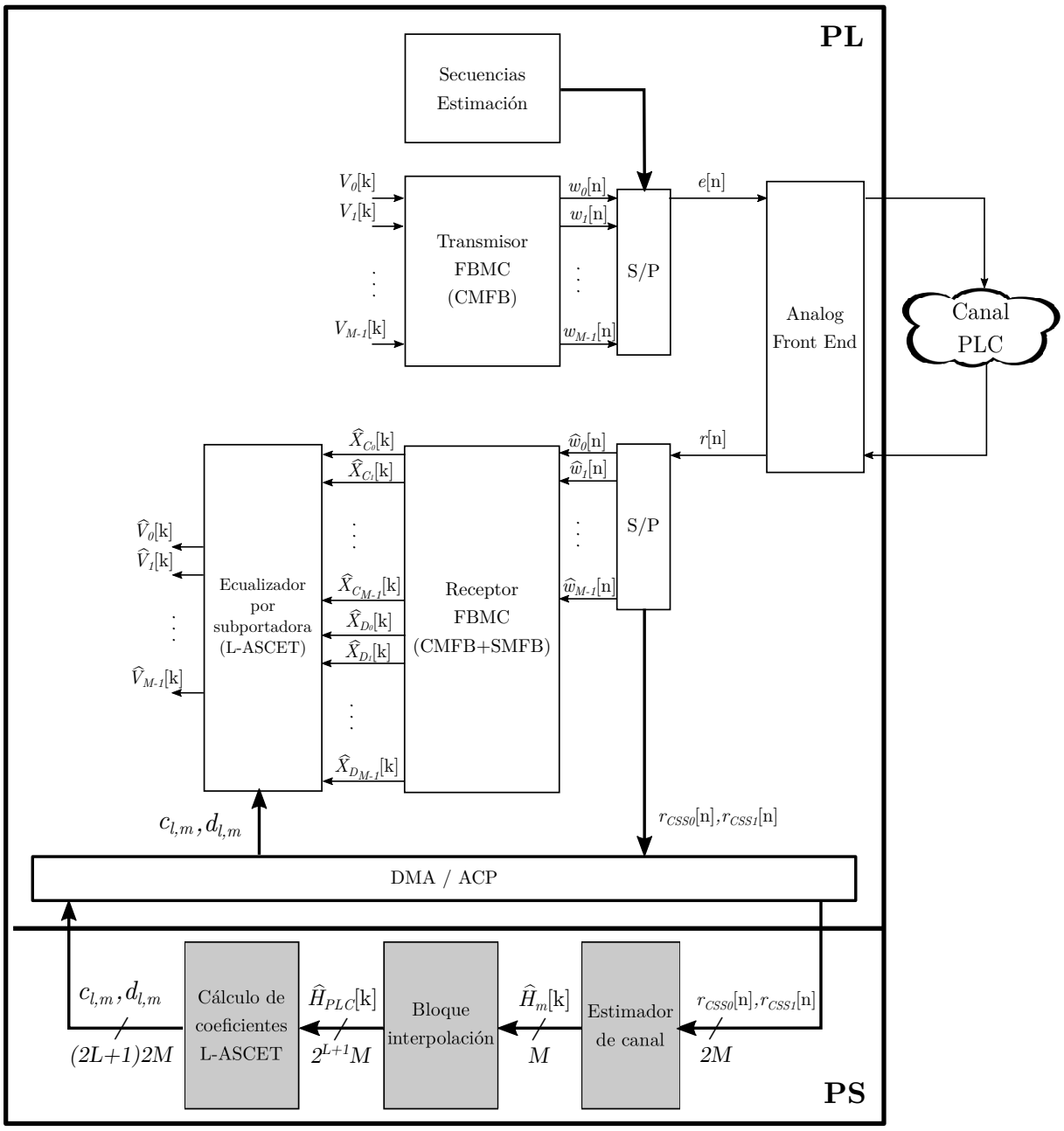


Figura 4.19: Esquema de la arquitectura heterogénea propuesta para el sistema FBMC, junto con el módulo de estimación y ecualización de canal.

Una vez establecido el dimensionado del *datapath*, se han presentado las características que tiene la herramienta de síntesis Vivado HLS, mostrando las principales directivas utilizadas para alcanzar las restricciones establecidas por el estándar PLC. Estas directivas permiten reducir la latencia e intervalo de inicialización de los diseños, mediante el uso de la segmentación o la división de memoria. La principal ventaja de utilizar la herramienta de síntesis Vivado HLS es la capacidad de explorar las diferentes soluciones disponibles dentro del espacio de diseño mediante el uso de directivas.

A partir de la descripción de los módulos del sistema FBMC, se ha realizado el modelado mediante Vivado HLS, aplicando las directivas necesarias en cada caso. El modelado de cada uno de los módulos se ha realizado de manera eficiente, reduciendo el consumo de bloques de memoria y bloques DSP, escasos en la mayoría de los dispositivos FPGA.

Se ha realizado la descripción de la arquitectura y posterior implementación para todos los módulos, excepto para los módulos de estimación de canal y el cálculo de coeficientes del igualador. Teniendo en cuenta la cantidad de recursos que se invierten en el desarrollo de la FFT, además del elevado consumo de recursos que tiene el sistema global, se ha optado por plantear una implementación *hardware*/*software*, donde la estimación de canal y el cálculo de coeficientes se lleva a cabo en la parte *software*, mientras que el resto de la arquitectura se implementa en *hardware*. En el siguiente capítulo, se detallarán las características relevantes del modelado *software* de la arquitectura heterogénea.

Capítulo 5

Definición de los bloques *software* en la arquitectura heterogénea propuesta

5.1 Introducción

En el capítulo 4 se ha mostrado el consumo de recursos que presenta la arquitectura. Sin embargo, debido a la complejidad que presenta el sistema y las restricciones temporales que se deben alcanzar, el consumo de recursos obtenido se puede considerar elevado. Para evitar aumentar aún más este consumo de recursos, se ha propuesto la arquitectura mixta HW/SW, descrita en 3.4.3. Los módulos que no presentan exigentes restricciones temporales se van a codificar en la parte *software*, tratando de alcanzar las restricciones temporales que estos módulos tengan impuestas.

En este capítulo se detalla el modelado *software* que se ha llevado a cabo con dos objetivos claros; el primero de ellos, se centra en obtener el mayor rendimiento posible de la arquitectura, utilizando los recursos disponibles en la tecnología destino. En este sentido, se muestra en el capítulo la forma de utilizar las unidades vectoriales para acelerar determinados cálculos o el uso de los operadores enteros. El otro objetivo en la parte del modelado *software* es alcanzar las restricciones temporales que se imponen en el sistema global. Para ello se ha realizado una serie de propuestas multinúcleo de la arquitectura mixta HW/SW, con el objetivo de reducir el tiempo de procesamiento que requiere el algoritmo presentado. En relación con esto, se ha propuesto una arquitectura multiprocesador para dos y cuatro núcleos, estableciendo un canal de comunicación entre los diferentes procesadores que permitan sincronizar el sistema, tanto en la parte *software* como en la *hardware*.

Este capítulo se ha organizado como sigue. En la sección 5.2 se muestra el estudio teórico llevado a cabo para evaluar el rendimiento del sistema, cuyo principal objetivo es

determinar el impacto que tiene la estimación de canal en el sistema considerando diferentes opciones que se pueden dar en la implementación. En la sección 5.3 se presenta la propuesta *hardware/software* que se ha planteado a lo largo de los capítulos anteriores. En este caso, la propuesta se engloba dentro de la arquitectura del dispositivo aprovechando las características que tiene para obtener el máximo rendimiento. Dentro de la sección se detallarán los aspectos más relevantes para obtener el máximo rendimiento. En la sección 5.5 se muestra la propuesta mixta HW/SW para una arquitectura multiprocesador, mostrando los resultados para doble núcleo y cuatro núcleos. Cabe destacar que en esta parte se realiza la implementación sobre dos arquitecturas diferentes de la familia Zynq® Xilinx: Zynq® 7000 y Zynq® UltraScale+. En ambos casos se detallan los modos de sincronización utilizados entre los diferentes núcleos del procesador. Por último, en la sección 5.6 se muestran una serie de conclusiones parciales recogidas a lo largo del capítulo.

5.2 Impacto de la estimación de canal y ecualización en el sistema

Hasta ahora se ha presentado la definición de la arquitectura *hardware* llevada a cabo. Sin embargo, a la hora de definir la arquitectura, no se ha mostrado la mejora sobre las prestaciones del sistema que introduce el ecualizador ASCET, así como tampoco se ha profundizado en un estudio de las prestaciones del sistema en términos de SNR (*Signal-Noise Ratio*), RMSE (*Root-Mean-Square Error*) o BER (*Root-Mean-Square Error*).

En esta sección se presenta el impacto que tiene el uso del ecualizador de canal, así como el efecto de la representación en punto fijo en el sistema global. Para ello se ha dividido el estudio en tres partes; la primera de ellas (5.2.1) evalúa el uso de la representación en punto fijo sobre el transmisor FBMC, receptor FBMC y el ecualizador ASCET, realizando una estimación de canal perfecta, es decir, considerando el canal conocido en todo momento. La segunda parte (5.2.2) evalúa el efecto de realizar la estimación de canal propuesta y el cálculo de coeficientes para el ecualizador ASCET, considerando ideales el transmisor y receptor FBMC, así como el ecualizador ASCET. En este caso se tendrán en cuenta ciertos casos particulares que se pueden dar a la hora de realizar la estimación de canal y cálculo de coeficientes cuando se implemente en la parte *software*. Por último, se llevará a cabo la evaluación del conjunto global de la propuesta (5.2.3), considerando las cuantificaciones utilizadas para la representación en punto fijo y realizando la estimación de canal, así como el cálculo de coeficientes para el ecualizador ASCET. A continuación, en las siguientes subsecciones, se profundizará en cada una de las tres partes que comprende este estudio.

5.2.1 Evaluación del efecto de la representación en punto fijo

En primer lugar se pretende evaluar el efecto que tiene la representación en punto fijo en el sistema. Para ello se va a llevar a cabo una comparación en punto fijo y en punto flotante de doble y simple precisión. El hecho de evaluar el sistema con la representación en punto flotante de simple precisión es debido a que la parte *hardware* y parte *software* de la arquitectura heterogénea pueden utilizar aritmética de punto flotante de simple precisión. Por otro lado, la máxima precisión se consigue con la representación en punto flotante de doble precisión.

En el esquema de la figura 5.1, se muestra el diagrama de bloques del sistema simplificado. Para obtener el rendimiento del sistema en términos de SNR y RMSE, se va a realizar una comparación de la señal de entrada al transmisor FBMC $V_m[k]$ con la señal obtenida tras el módulo del ecualizador L-ASCET $\hat{V}_m[k]$. Por otro lado, el análisis del BER se realizará añadiendo ruido blanco gaussiano aditivo (AWGN) al modelo del canal y se compararán los valores demodulados $\hat{V}_m^{PAM}[k]$ con los correspondientes valores previos a la modulación PAM aplicada $V_m^{PAM}[k]$. Téngase en cuenta que el bloque denominado *Transmisor FBMC* y *Receptor FBMC* conforma el conjunto de bloques del sistema mostrados en el capítulo 4. Además, el módulo *Ecualizador L-ASCET* se refiere únicamente al igualador de canal mostrado en el punto 4.8. Por último, el modulador y demodulador trabajan con un esquema de modulación 4-PAM, tal y como define el estándar PLC IEEE 1901/2010 [IEE10].

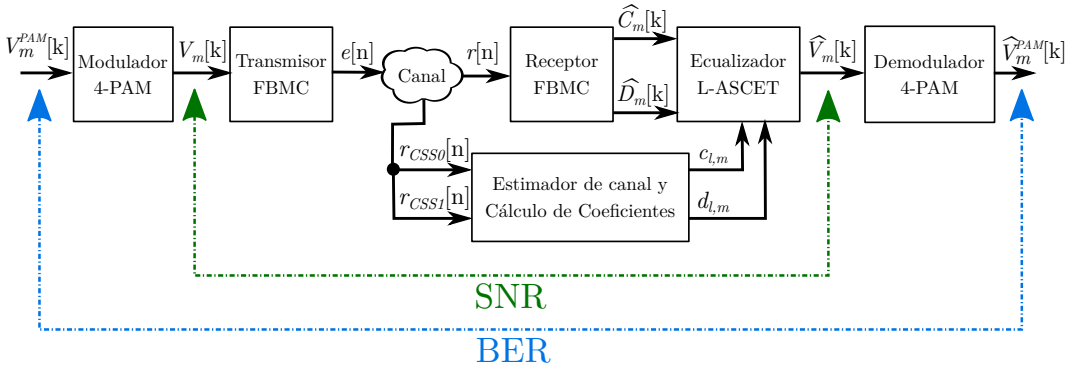


Figura 5.1: Diagrama de bloques del sistema simplificado con el esquema de modulación 4-PAM, donde se marcan las señales para el cálculo de la SNR y BER.

Para realizar la simulación del sistema se ha optado por utilizar los modelos de canal PLC definido en [Ton12]. En este caso se han usado dos modelos de canal: el modelo A y modelo B. La diferencia entre ellos radica en que el modelo A es más exigente que el B y por tanto, el modelo A presenta un mayor nivel de ruido, interferencia, multicamino, etc. En la figura 5.2 se muestra el comportamiento de los modelos de canal PLC utilizados. En cuanto al modelo A, se puede comprobar cómo la respuesta en frecuencia tiene variaciones más rápidas y los desvanecimientos son más numerosos, comparados con el modelo B. Entre modelos de canal sucesivos en el tiempo se ha respetado el tiempo de coherencia de

canal $t_{C_{PLC}}$.

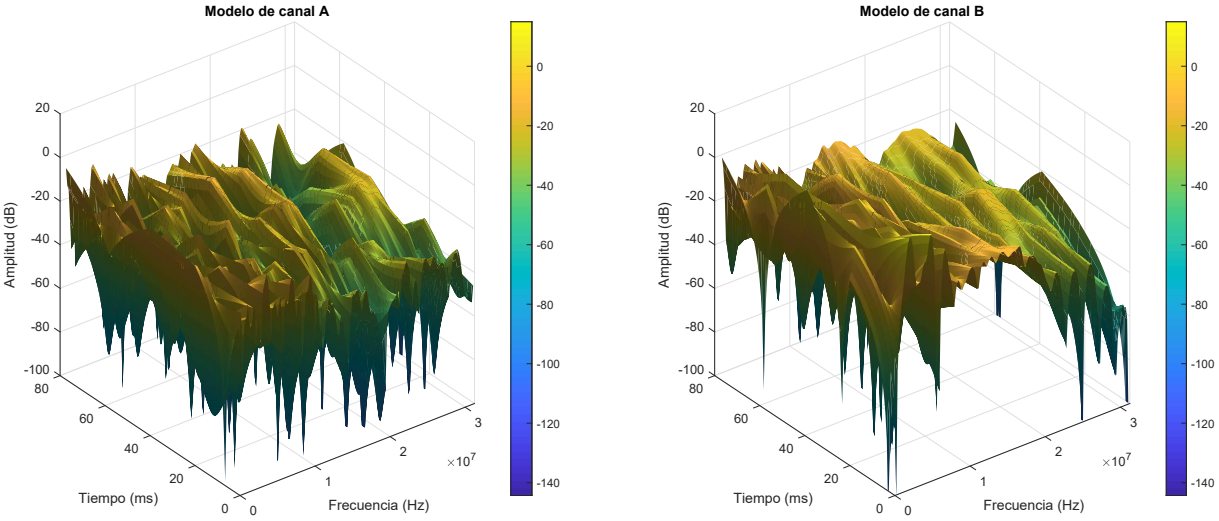


Figura 5.2: Respuesta en frecuencia a lo largo del tiempo de los modelos de canal PLC utilizados. A la izquierda el modelo A del canal PLC y a la derecha el modelo B del canal PLC.

Por otro lado se van a definir los parámetros de SNR y RMSE de acuerdo con (5.1) y (5.2), respectivamente, donde $V_m[k]$ y $\hat{V}_m[k]$ representan las señales de entrada al transmisor FBMC y la salida del ecualizador, siendo $m = 0, ..., M-1$.

$$SNR = 20 \cdot log\left(\frac{|V_m[k]|}{|V_m[k] - \hat{V}_m[k]|}\right) \tag{5.1}$$

$$RMSE = \sqrt{\frac{\sum_{m=0}^{M-1}(V_m[k] - \hat{V}_m[k])^2}{M}} \tag{5.2}$$

Para evaluar el efecto que tiene la representación en punto fijo en el sistema, se realiza la simulación del mismo. Esta simulación realiza el cálculo de los valores de SNR y RMSE en cien instantes para los modelos de canal mostrados anteriormente en la figura 5.2. Por cada instante del modelo de canal, se ha realizado el envío de una trama compuesta por 64 paquetes $D_0, \dots D_{63}$ de información, con $M = 512$ muestras cada uno, junto con las dos tramas C_0 y C_1 de $M = 512$ muestras necesarias para realizar la estimación de canal. En resumen, si cada paquete de información con $M = 512$ muestras se envía cada 8.192 µs, la trama compuesta por los 66 paquetes se envía cada 540.672 µs, que está por debajo del tiempo de coherencia del canal $t_{C_{PLC}}$, fijado en 600 µs [Cor06]. Si se desea ampliar el número de paquetes de información que se envían, hay que tener en cuenta que en ningún caso se debe superar el $t_{C_{PLC}}$, para poder considerar el canal como un sistema LTI, y garantizar así que la estimación del canal se pueda aplicar al conjunto de datos enviados. En la figura 5.3 se muestra la estructura de la trama transmitida entre el emisor y el receptor. Cabe señalar que el preámbulo está formado por dos tramas S_0 y S_1, además de C_0 y C_1, encargadas de sincronizar el sistema transmisor con el receptor. En este caso, no se incorporan ya que el sistema se considera completamente sincronizado, aunque habría que tener en cuenta estas tramas para no superar el tiempo de coherencia del canal $t_{C_{PLC}}$.

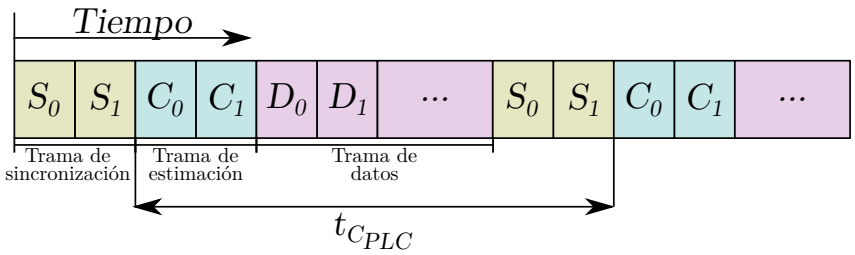


Figura 5.3: Estructura de la trama intercambiada por el transmisor FBMC y receptor FBMC, donde se muestra el preámbulo (S_x y C_x) y las diferentes tramas de datos (D_x).

Con todo esto, se va a evaluar el sistema en punto fijo y en punto flotante, considerando en todo momento una estimación de canal perfecta para comprobar dicho efecto correctamente. Además, para evitar los efectos que pueda tener el canal PLC, el sistema se ha evaluado en presencia de un canal ideal, lo que permite obtener el deterioro debido a la cuantificación utilizada. En la tabla 5.1 se recogen los valores de SNR y RMSE en presencia del canal ideal, así como los modelos de canal A y B comentados anteriormente para los tres órdenes de ecualizador L-ASCET. Además, se realiza el estudio para los formatos de representación de punto flotante de doble precisión (DP), simple precisión (SP) y punto fijo.

		0-ASCET		1-ASCET		2-ASCET	
		SNR (dB)	RMSE	SNR (dB)	RMSE	SNR (dB)	RMSE
Punto flotante (DP)	Canal Ideal	225.69	9.99×10^{-6}	230.65	7.06×10^{-6}	230.65	7.06×10^{-6}
	Canal PLC (A)	40.60	2.23×10^{-1}	60.94	8.47×10^{-2}	63.57	1.66×10^{-1}
	Canal PLC (B)	46.22	1.93×10^{-1}	58.95	1.28×10^{-1}	59.83	1.72×10^{-1}
Punto flotante (SP)	Canal Ideal	220.74	7.99×10^{-6}	223.74	6.40×10^{-6}	223.74	6.40×10^{-6}
	Canal PLC (A)	40.60	2.23×10^{-1}	60.93	8.48×10^{-2}	62.35	1.72×10^{-1}
	Canal PLC (B)	46.22	1.90×10^{-1}	54.36	1.62×10^{-1}	54.72	2.41×10^{-1}
Punto fijo	Canal Ideal	165.06	2.10×10^{-4}	165.06	2.10×10^{-4}	165.06	2.10×10^{-4}
	Canal PLC (A)	40.60	2.23×10^{-1}	60.93	8.48×10^{-2}	62.35	1.72×10^{-1}
	Canal PLC (B)	46.23	1.93×10^{-1}	54.35	1.62×10^{-1}	51.80	2.14×10^{-1}

Tabla 5.1: Resultados de SNR y RMSE obtenidos de la simulación del sistema para los diferentes órdenes de ecualizador L-ASCET en presencia de canal ideal y modelos A y B de canal PLC, para los diferentes métodos de representación.

Las pérdidas debidas a la cuantificación empleada se pueden comprobar observando el valor de SNR en presencia de un canal ideal para los tres formatos de representación mostrados en la tabla 5.1. Para la representación en punto fijo se obtiene una diferencia de aproximadamente 55 dB y 60 dB con respecto al caso de punto flotante de simple y doble precisión, respectivamente. Estas pérdidas son considerables, pero si se atiende a la señal obtenida en la etapa de recepción mostrada en la figura 5.4, se puede comprobar cómo la representación en punto fijo se ajusta correctamente a la señal obtenida por la representación en punto flotante de doble precisión.

Ahora bien, en presencia de un modelo de canal PLC, la diferencia entre el sistema en punto fijo y punto flotante de simple o doble precisión no es tan notable. En la tabla 5.1, se puede comprobar como la diferencia entre punto fijo es casi nula para punto flotante de simple precisión y doble precisión en el caso del modelo A y modelo B de canal

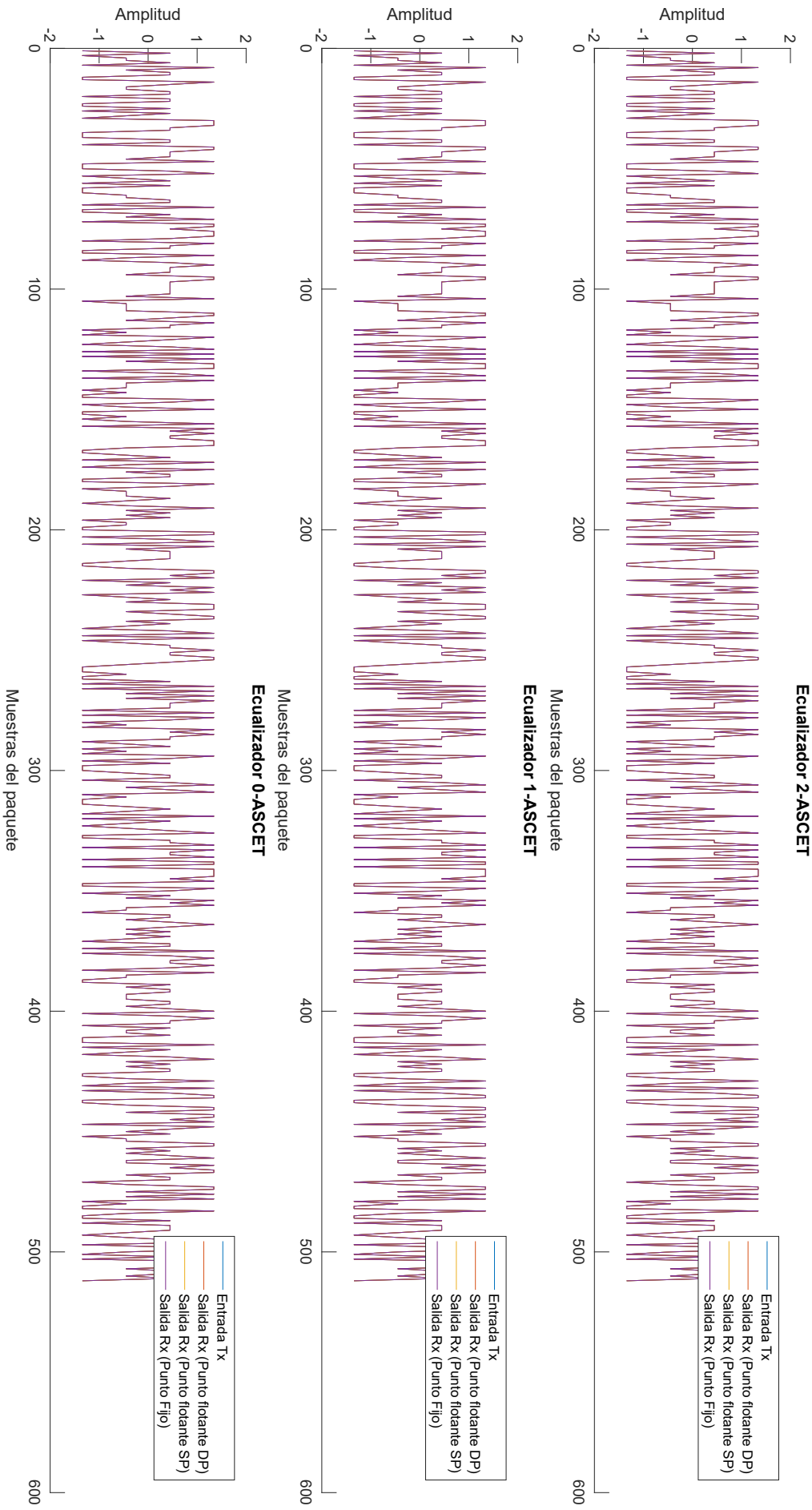


Figura 5.4: Comparación entre la señal de entrada $V_m[k]$, la señal de salida $\hat{V}_m[k]$ en punto flotante de doble y simple precisión, y en punto fijo.

PLC. Como se puede observar, las características que tiene el canal PLC, en términos de ruido, interferencia o multicamino, entre otros, hacen que la SNR del sistema descienda notablemente, si se comparan los resultados obtenidos para los canales PLC con el caso del canal ideal.

En cuanto al error cuadrático medio, para los modelos de canal PLC se obtienen errores ligeramente elevados comparados con el caso de canal ideal. Sin embargo, teniendo en cuenta que se trata de una señal modulada con unos niveles muy bien definidos, se puede considerar este nivel de RMSE adecuado para el sistema.

En los resultados mostrados hasta ahora se ha utilizado en la simulación el ecualizador L-ASCET. En la tabla 5.1, se puede comprobar que el uso de un ecualizador L-ASCET de orden 2 mejora hasta 20 dB la relación señal a ruido del sistema frente al ecualizador 0-ASCET. La mejora que supone el uso del ecualizador se muestra en la figura 5.5 donde se presenta la señal transmitida, previa al banco de filtros del transmisor, con la señal del banco de filtros del receptor sin ecualizar y la señal obtenida tras el módulo del ecualizador. Como se puede comprobar, al realizar la ecualización se mejora la señal recibida, permitiendo obtener una representación más fidedigna de la señal transmitida. Además, el uso de un ecualizador 2-ASCET mejora notablemente la SNR del sistema frente al 0-ASCET.

De esta forma se pretende reflejar la mejora que supone aplicar la ecualización de la señal recibida, haciendo que sea conveniente realizar dicha ecualización para contrarrestar los efectos no deseados introducidos por el canal y mejorando así las prestaciones del sistema.

Sin embargo, para realizar correctamente la ecualización es necesario tener un modelo del canal, o bien realizar la estimación del mismo. En este caso, dadas las variaciones tan rápidas que presenta el canal PLC, así como las dificultades que se tiene a la hora de modelarlo debido a la gran variedad de entornos que existen, lo más conveniente es realizar la estimación del canal y compensar los efectos perniciosos en tiempo de ejecución y no trabajar con modelos de canal predefinidos.

5.2.2 Evaluación del efecto del uso del estimador de canal

Como se ha visto, la necesidad de realizar la ecualización de canal se ve justificada por la mejora que supone al sistema. No obstante, hay que tener en cuenta que el hecho de realizar el cálculo de la estimación de canal aumenta la carga computacional en la etapa de recepción. Eso se debe a que habrá dos flujos de datos que se deben procesar en paralelo. Por un lado es necesario procesar las tramas de datos en el banco de filtros del sistema receptor, y por otro, se necesita calcular la estimación del canal y los coeficientes del ecualizador L-ASCET para compensar así los efectos no deseados introducidos por el canal.

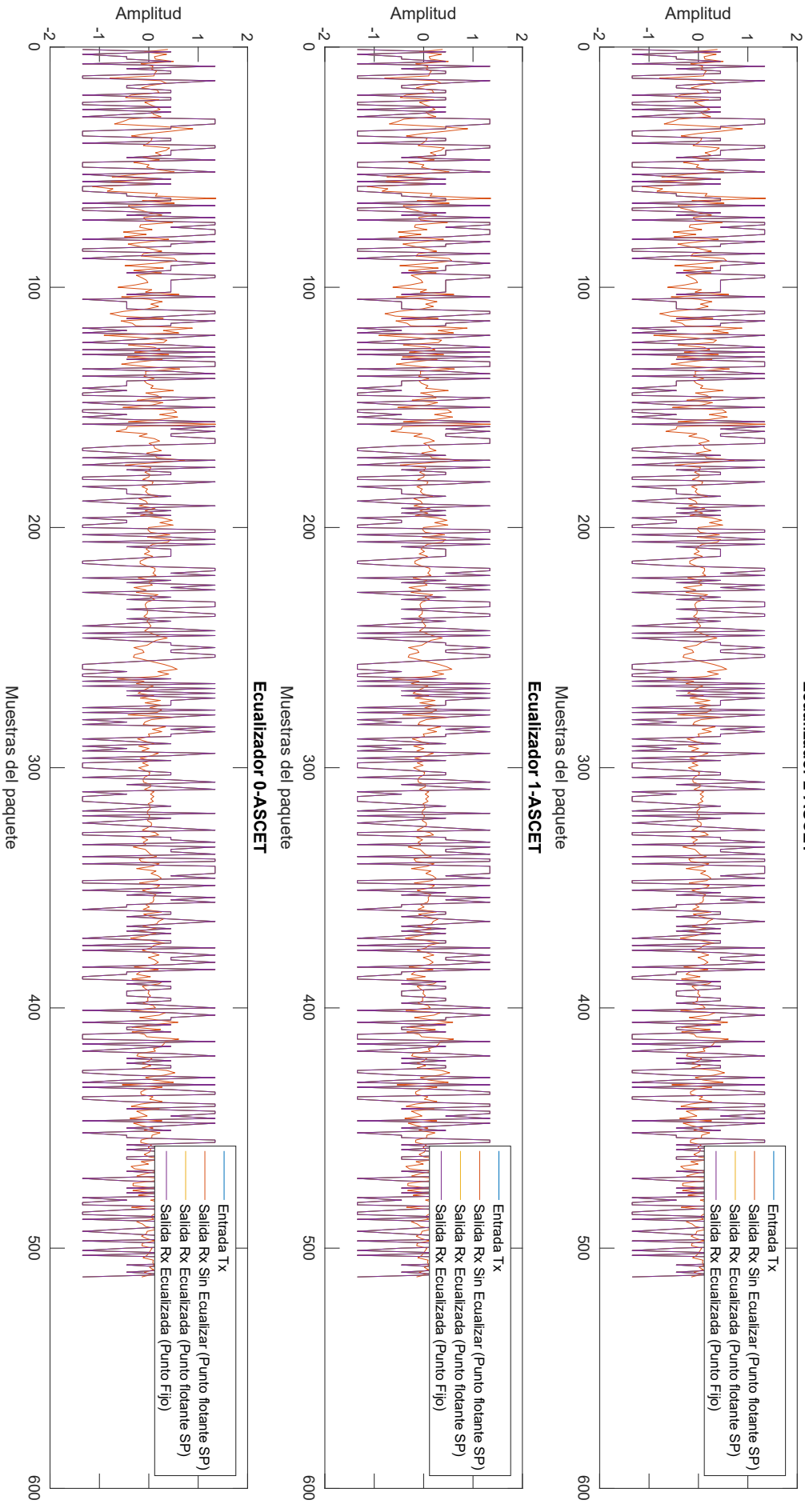


Figura 5.5: Comparación entre la señal de entrada $V_m[k]$, la señal de salida $\hat{V}_m[k]$ del banco de filtros sin ecualizar y la señal ecualizada.

La propuesta de la arquitectura mixta HW/SW permite reducir el consumo de recursos *hardware* de la arquitectura global, ya que el aumento de la carga computacional que supone la estimación de canal y cálculo de coeficientes es significativo. Dado que las etapas de estimación de canal y cálculo de coeficientes se realiza con un periodo mayor, que viene dado por el tiempo de coherencia del canal $t_{C_{PLC}}$, se codificarán en la parte *software*, mientras que el banco de filtros receptor y el igualador L-ASCET se implementa en *hardware*.

En la figura 5.6 se muestra la arquitectura descrita en anteriores capítulos, incluyendo ahora los dos tiempos: t_{RX} y t_{Coef}. El primero de ellos, t_{RX}, representa la latencia que tiene el subsistema implementado en PL, desde que los datos están disponibles en el receptor hasta que se generan las correspondientes salidas de los bancos de filtros del coseno (CMFB) y del seno (SMFB) y están disponibles en la entrada del ecualizador L-ASCET. Esta latencia se corresponde con la mostrada en la tabla 4.12 para el receptor FBMC. El segundo t_{Coef} representa la latencia que tiene el subsistema codificado en PS, el cual engloba la transferencia DMA de las tramas de estimación de canal (PL a PS), el cálculo de la estimación de canal, el cálculo de coeficientes y la transferencia DMA de los coeficientes al ecualizador L-ASCET (PS a PL).

En esta arquitectura propuesta se pueden dar dos posibles situaciones que afectan al rendimiento del sistema. El primer caso es el más favorable, donde la estimación de canal y el cálculo de coeficientes se realiza antes de que los datos procedentes del módulo receptor estén disponibles en el ecualizador ASCET, cumpliendo así que $t_{Coef} \leq t_{RX}$. En este caso, si no se supera el tiempo de coherencia del canal, la SNR obtenida no se vería degradada.

La segunda situación que se puede dar es que no se cumpla la premisa $t_{Coef} \leq t_{RX}$ y que esto se deba a que el tiempo requerido para realizar el cálculo en PS sea superior al tiempo de procesamiento de módulo de recepción. En la figura 5.7 se muestra gráficamente este caso y los posibles escenarios a considerar.

En la parte superior de la figura 5.7, se muestra el efecto que se produce al inicializar el sistema con las primeras muestras ecualizadas, ya que éstas serán erróneas hasta que se actualicen los coeficientes del ecualizador y comience a producir resultados válidos. Este error se verá reflejado en el rendimiento del sistema en términos de SNR o BER, pero al tratarse de un periodo inicial, es posible despreciarlo, ya que se puede considerar la inicialización del sistema. Ahora bien, una vez inicializado el sistema, siempre que se actualicen los coeficientes del ecualizador sin superar $t_{C_{PLC}}$, los datos producidos serán válidos y el rendimiento del sistema se verá afectado ligeramente en términos de SNR o BER debido a que la ecualización no se está realizando con los coeficientes correctos.

Sin embargo, si se supera el tiempo de coherencia del canal $t_{C_{PLC}}$ y no se han actualizado los coeficientes del ecualizador, se producirán errores en la señal ecualizada, provocando que el rendimiento del sistema se vea deteriorado en términos de SNR y BER. Cabe destacar que cuanto más complejo sea el cálculo de los coeficientes, como ocurre con

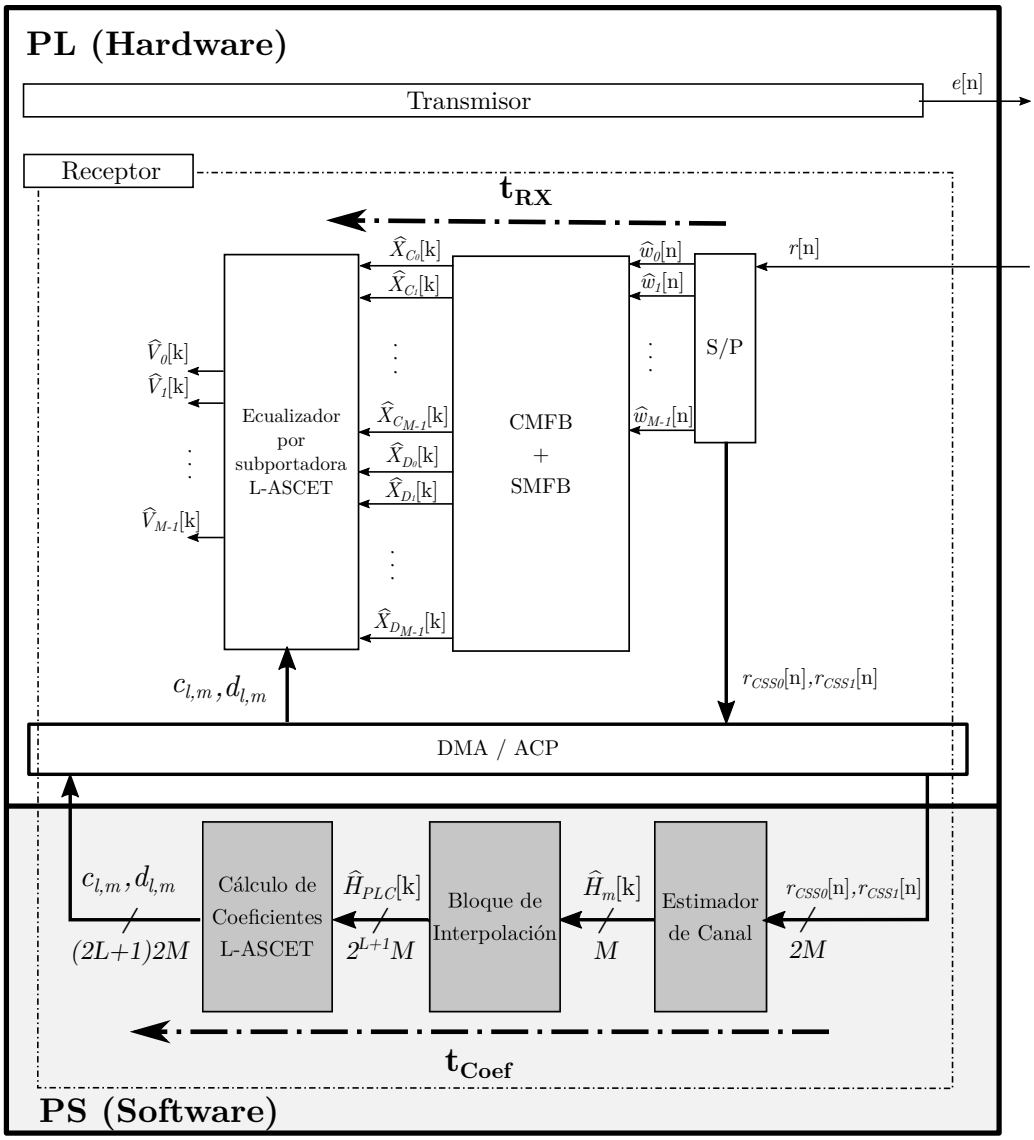


Figura 5.6: Diagrama de bloques de la arquitectura HW/SW propuesta para el sistema FBMC.

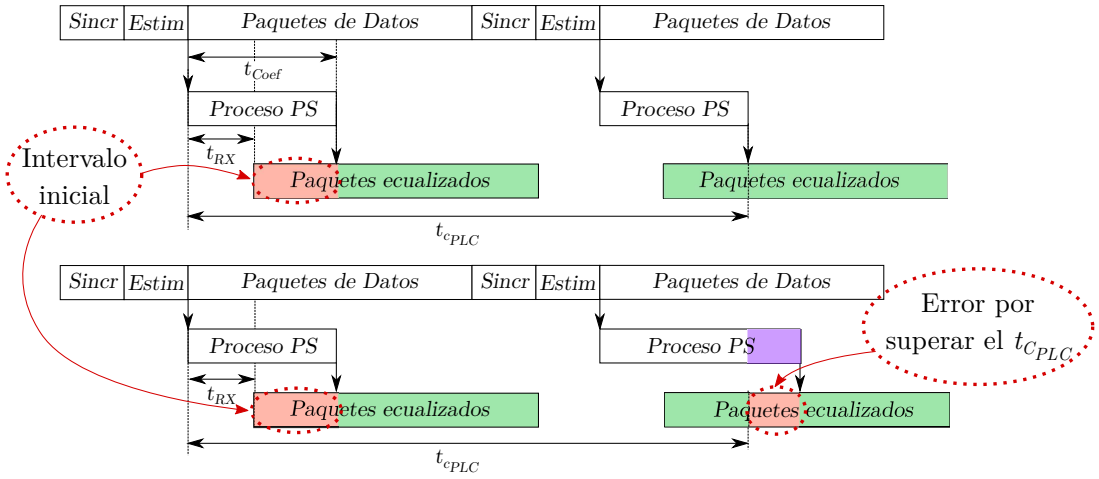


Figura 5.7: Diagrama de el flujo de datos en la etapa de recepción.

el ecualizador 2-ASCET, el retardo puede ser mayor, y esto afecta al tiempo necesario para el cálculo de los coeficientes del ecualizador.

Con estas consideraciones se va a comprobar el efecto que tiene en el sistema la estimación de canal y cómo afecta a las prestaciones el retardo que pueda tener la actualización de los coeficientes del ecualizador, evitando superar en todo momento el tiempo de coherencia del canal PLC $t_{C_{PLC}}$. De acuerdo con el esquema mostrado en la figura 5.8 se muestran los diferentes ensayos llevados a cabo. El primero de ellos (I) es una situación ideal que servirá de referencia a considerar, dado que se realiza el cálculo de los coeficientes a partir del modelo del canal H_{PLC}. La segunda evaluación (II) proporciona la degradación debida a la interpolación que se realiza en la estimación de canal, sin embargo en este caso se realiza a partir del modelo del canal H_{PLC}. El tercer ensayo (III) incorpora la estimación de canal, en lugar del modelo, y a partir de esta estimación de canal se realiza la interpolación y el cálculo de coeficientes, proporcionando la degradación debida a la estimación de canal. Por último, la rama (IV) evalúa la posibilidad de incluir un suavizado en la estimación de canal con el objetivo de mejorar el rendimiento final en términos de SNR.

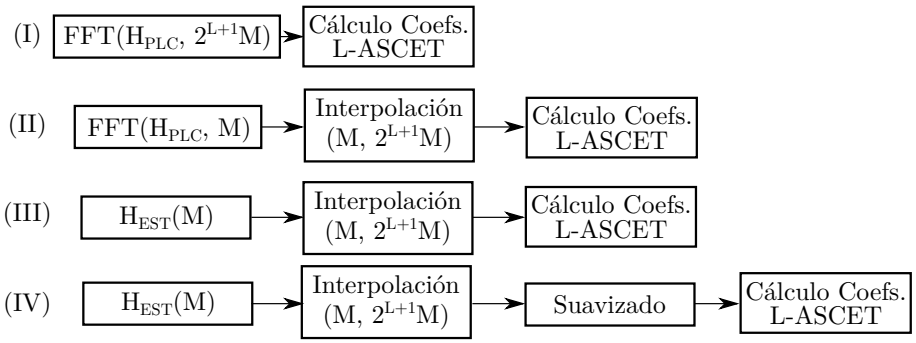


Figura 5.8: Diagrama de los escenarios a evaluar para comprobar el efecto de la estimación de canal y cálculo de coeficientes en el rendimiento del sistema.

Cada uno de los ensayos descrito anteriormente ha sido llevado a cabo considerando la representación en punto flotante de doble precisión y en punto fijo para la parte *software*, que se detalla en la sección 5.3.1.1. Los resultados se muestran en la tabla 5.2 en términos de SNR y RMSE con un ecualizador L-ASCET para $L = 0, 1, 2$. En este caso, la estimación de canal y los coeficientes del ecualizador se calculan cumpliendo $t_{Coef} \leq t_{RX}$. Estos resultados se han obtenido para el modelo de canal A y B mostrados en la sección anterior.

Con respecto a la tabla 5.2, en términos de SNR existe una diferencia que se sitúa en torno a los 14 dB y 19 dB comparando el caso ideal (I) y el caso en el que se realiza con la estimación de canal (III) para ambos modelos de canal PLC (A y B). La propuesta (IV), en la que realiza un suavizado de la estimación de canal obtenida, no obtiene mejores resultados para el modelo de canal A, ya que es muy exigente y presenta variaciones muy rápidas en la respuesta en frecuencia del canal, que se pierden al realizar el suavizado en la estimación de canal obtenida y empeora la SNR. Para el caso del modelo B, que es menos

		0-ASCET		1-ASCET		2-ASCET	
		SNR (dB)	RMSE	SNR (dB)	RMSE	SNR (dB)	RMSE
Canal PLC (A)	(I)	42.22	0.189	62.75	0.175	65.41	0.175
	(II)	41.85	0.190	58.06	0.179	59.70	0.178
	(III)	28.03	0.164	29.18	0.161	29.47	0.164
	(IV)	24.88	0.204	25.61	0.210	25.70	0.210
	(II-FP)	41.85	0.190	58.06	0.179	59.70	0.178
	(III-FP)	28.04	0.164	29.18	0.161	29.48	0.163
	(IV-FP)	24.88	0.204	25.61	0.210	25.70	0.210
Canal PLC (B)	(I)	58.33	0.324	76.03	0.247	78.05	0.245
	(II)	58.17	0.272	74.28	0.241	75.90	0.237
	(III)	39.24	0.153	39.68	0.127	40.00	0.130
	(IV)	40.12	0.187	41.16	0.199	41.25	0.200
	(II-FP)	58.17	0.272	74.28	0.241	75.90	0.237
	(III-FP)	39.24	0.153	39.68	0.127	40.00	0.130
	(IV-FP)	40.12	0.186	41.16	0.199	41.25	0.199

Tabla 5.2: Resultados de SNR y RMSE obtenidos de la simulación del sistema para los diferentes órdenes de ecualizador L-ASCET y casos presentados, realizando la estimación de canal y cálculo de coeficientes cumpliendo $t_{Coef} \leq t_{RX}$.

exigente y la variaciones en la respuesta en frecuencia son menores, la propuesta (IV) con el suavizado supone una ligera mejora con respecto al caso (III). Por otro lado, el uso de operadores de punto fijo en la parte *software* no presenta una influencia significativa en el rendimiento del sistema con respecto a los operadores de punto flotante de doble precisión.

Hasta ahora se ha presentado el caso más favorable, pero es interesante estudiar el efecto que tendría en el sistema global el caso en el que no se cumpla $t_{Coef} \leq t_{RX}$. En la tabla 5.3 se muestran los resultados obtenidos en términos de SNR y RMSE para los casos descritos anteriormente. En este caso, no se cumple $t_{Coef} \leq t_{RX}$ pero no se supera el tiempo de coherencia del canal PLC t_{CPLC}; para ello se ha definido un retardo que equivale al número de paquetes ecualizados con los coeficientes no actualizados. En este sentido, dado que la carga computacional para el cálculo de coeficientes del ecualizador 2-ASCET es superior, se ha asignado un retardo de 32 paquetes, de 16 paquetes para el caso 1-ASCET y de 8 paquetes para el 0-ASCET. Estos valores se han considerado de forma arbitraria, ya que depende de los tiempos de procesamiento requeridos para el cálculo de la estimación de canal y cálculo de coeficientes en la parte *software*, y este tiempo dependerá de cada una de las estrategias mostradas a lo largo del capítulo en los apartados 5.4 y 5.5.

Con respecto a los resultados obtenidos en la tabla 5.3, se puede comprobar que los valores obtenidos son ligeramente inferiores con respecto al caso anterior mostrado en la tabla 5.2. Esto se debe a que los datos se están ecualizando a partir de una estimación de canal que no corresponde y, por tanto, se produce un deterioro en las señales recibidas del sistema reduciendo el SNR. Para el caso 2-ASCET, el cual tiene un mayor retardo asignado, los resultados de SNR se ven degradados aún más que para el caso 1-ASCET,

		0-ASCET		1-ASCET		2-ASCET	
		SNR (dB)	RMSE	SNR (dB)	RMSE	SNR (dB)	RMSE
Canal PLC (A)	(I)	40.95	0.369	54.54	0.574	52.55	0.580
	(II)	40.63	0.367	51.60	0.542	49.74	0.550
	(III)	27.60	0.247	28.31	0.294	28.12	0.319
	(IV)	24.50	0.261	24.86	0.302	24.58	0.322
	(II-FP)	40.63	0.367	51.61	0.542	49.75	0.550
	(III-FP)	27.60	0.247	28.32	0.294	28.13	0.319
	(IV-FP)	24.50	0.261	24.86	0.302	24.58	0.322
Canal PLC (B)	(I)	65.09	0.515	67.49	0.698	65.09	0.696
	(II)	64.22	0.514	66.50	0.685	64.22	0.686
	(III)	36.18	0.432	36.23	0.481	36.18	0.508
	(IV)	39.85	0.326	40.20	0.389	39.85	0.416
	(II-FP)	64.22	0.514	66.51	0.685	64.22	0.686
	(III-FP)	38.78	0.299	38.87	0.349	38.78	0.382
	(IV-FP)	39.85	0.326	40.20	0.389	39.85	0.416

Tabla 5.3: Resultados de SNR y RMSE obtenidos de la simulación del sistema para los diferentes órdenes de ecualizador L-ASCET y casos presentados, realizando la estimación de canal y cálculo de coeficientes en el caso que no se cumple $t_{Coef} \leq t_{RX}$.

debido a que la ecualización llevada a cabo se produce con un retardo mayor. Sin embargo, los resultados obtenidos son muy similares a los mostrados en la tabla 5.2, donde las diferencias entre una estimación ideal (I) y la propuesta (III) se mantienen en torno a los 14 dB y 19 dB para el modelo de canal A y B respectivamente.

Por tanto, esto permite determinar que la ecualización llevada a cabo con coeficientes que no han sido actualizados, es decir, cuando no se cumple la premisa $t_{Coef} \leq t_{RX}$, no implica una significativa degradación en el rendimiento del sistema global en términos de SNR. Sin embargo, es conveniente que el retardo sea siempre el mínimo posible, optimizando al máximo el rendimiento de la parte *software*, para cumplir $t_{Coef} \leq t_{RX}$ y obtener la menor degradación en SNR para el sistema global.

5.2.3 Evaluación del sistema propuesto

En los apartados anteriores, 5.2.1 y 5.2.2, se han mostrado los resultados de SNR y RMSE para la parte *hardware* y la parte *software* de forma independiente. En este apartado se va a realizar el estudio de ambas partes en conjunto. En cuanto a la parte *hardware*, se utilizarán los módulos mostrados en punto fijo que han sido detallados en el capítulo anterior. En cuanto a la parte *software*, se va a considerar el caso que cumple la premisa $t_{Coef} \leq t_{RX}$, realizando el cálculo de la estimación de canal y el cálculo de coeficientes mediante aritmética de punto fijo. Para ello se va a considerar en este estudio el caso III-FP donde se realiza el cálculo de la estimación de canal y de los coeficientes del ecualizador mediante los operadores de enteros con una representación en punto fijo. Además, este caso III-FP no realiza el suavizado de la estimación de canal, permitiendo así reducir el tiempo de procesamiento.

Los resultados de la tabla 5.4 muestran los valores obtenidos en términos de SNR y RMSE para la propuesta realizada. En este caso se compara con el caso ideal, donde la representación del sistema se realiza mediante aritmética en punto flotante de doble precisión. En este caso se puede comprobar como, para el caso del canal ideal, la SNR obtenida en la propuesta es 58 dB inferior con respecto a éste. En cuanto a la diferencia que existe entre la propuesta en presencia de un canal PLC, se puede comprobar que para el modelo de canal A y para el modelo B la diferencia es prácticamente nula con respecto al sistema en punto flotante de doble precisión.

		0-ASCET		1-ASCET		2-ASCET	
		SNR (dB)	RMSE	SNR (dB)	RMSE	SNR (dB)	RMSE
Canal Ideal	Caso Ideal	221.09	1.448×10^{-5}	223.83	1.26×10^{-5}	223.92	1.25×10^{-5}
	Propuesta	165.05	2.10×10^{-4}	165.04	2.10×10^{-4}	165.05	2.10×10^{-4}
Canal PLC (A)	Caso Ideal	26.39	5.99×10^{-1}	27.76	6.01×10^{-1}	28.09	8.65×10^{-1}
	Propuesta	26.39	5.99×10^{-1}	27.76	6.01×10^{-1}	28.86	8.65×10^{-1}
Canal PLC (B)	Caso Ideal	37.31	7.73×10^{-1}	37.51	8.35×10^{-1}	38.55	2.03×10^{-1}
	Propuesta	37.31	7.73×10^{-1}	37.51	8.35×10^{-1}	37.69	1.80×10^{-1}

Tabla 5.4: Resultados de SNR y RMSE obtenidos de la simulación del sistema completo para los diferentes órdenes de ecualizador L-ASCET.

Por último, se ha realizado un análisis de este sistema propuesto en términos de BER; para ello se ha realizado la demodulación de las tramas recibidas y se ha comparado la representación en binario de dicha trama con la representación binaria de la trama transmitida. Esto se ha realizado con los modelos de canal PLC mostrados anteriormente, añadiendo diferentes niveles de ruido blanco gaussiano aditivo (AWGN) en el canal PLC. Los resultados se muestran en la figura 5.9 para el modelo de canal A y en la figura 5.10 para el modelo de canal B.

En las figuras se muestran también los resultados para los diferentes órdenes de ecualizador en el caso ideal (Ideal) y el sistema propuesto en coma fija (Propuesta). Se puede comprobar como el hecho de realizar la estimación de canal reduce el BER del sistema. Cabe destacar que, para valores superiores a 40dB de SNR, la BER obtenida es prácticamente constante. Por último, la diferencia entre los ecualizadores 1- y 2-ASCET es poco significativa, aunque el BER obtenido para el caso 2-ASCET es ligeramente inferior.

5.3 Arquitectura mixta HW/SW propuesta

Los módulos detallados en el capítulo 4, así como la estimación de canal y cálculo de coeficientes, serán implementados en la FPGA de la familia Zynq®. Por otro lado, los módulos descritos en la sección anterior para llevar a cabo el cálculo de coeficientes y la estimación de canal se implementarán en el PS del dispostivo.

Sin embargo, a la hora de realizar el desarrollo de una arquitectura mixta HW/SW hay que tener en cuenta cómo realizar la sincronización y el intercambio de información entre ambas partes de la arquitectura. Para ello, el uso de comunicaciones basadas en

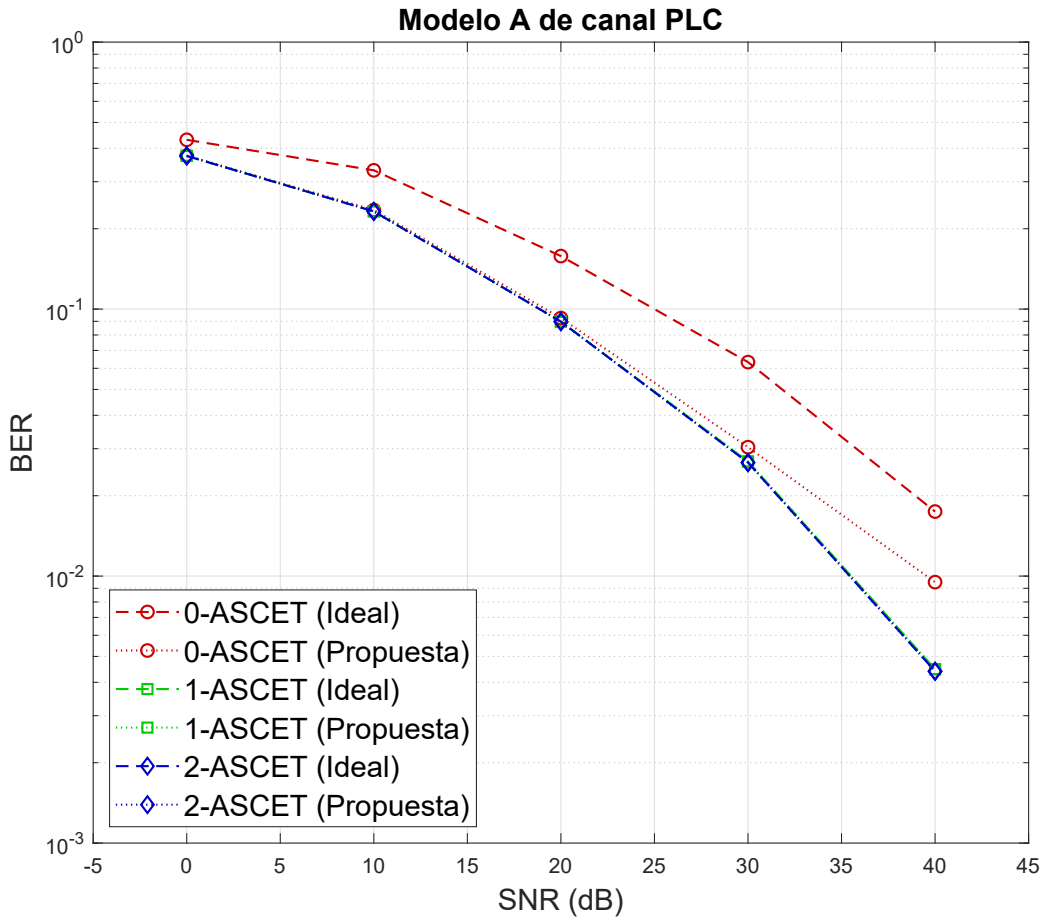


Figura 5.9: BER para cada ecualizador L-ASCET en presencia del modelo de canal A, asumiendo el sistema ideal en punto flotante de doble precisión (Ideal) y la propuesta realizada en punto fijo (Propuesta).

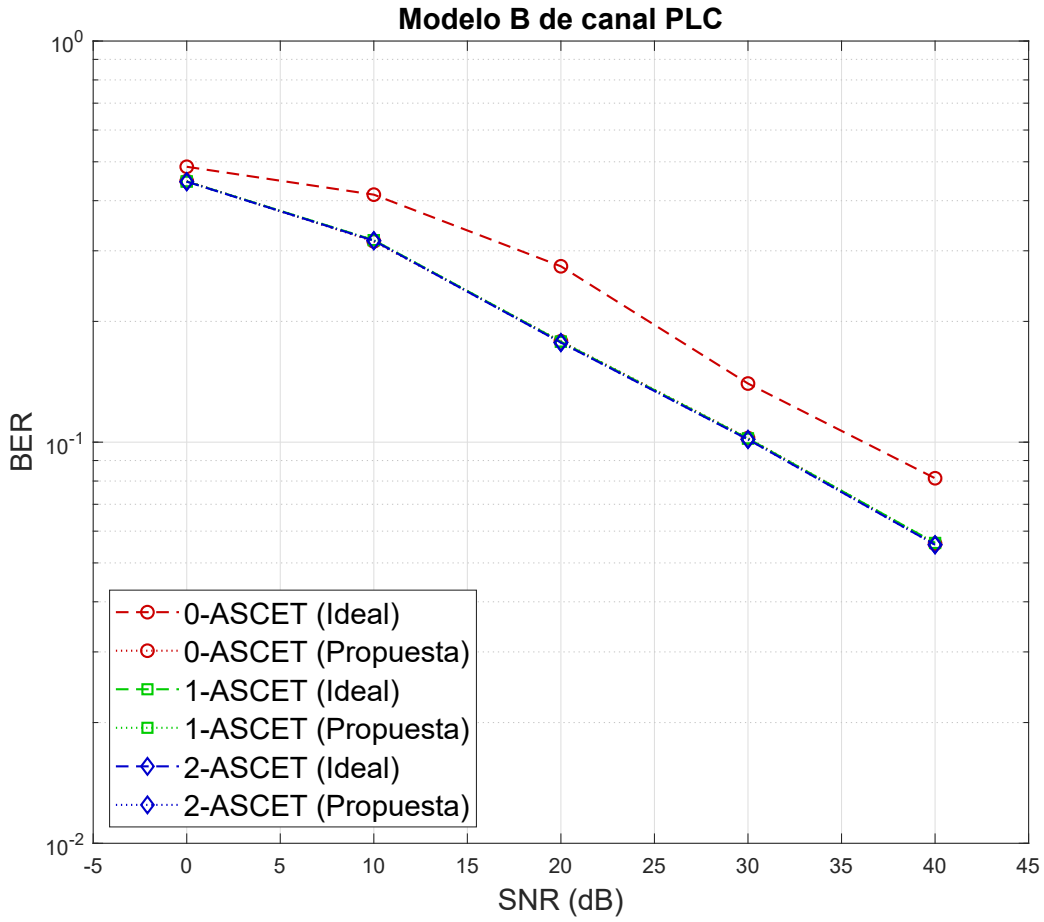


Figura 5.10: BER para cada ecualizador L-ASCET en presencia del modelo de canal B, asumiendo el sistema ideal en punto flotante de doble precisión (Ideal) y la propuesta realizada en punto fijo (Propuesta).

mailboxes [Xil18b] entre PL y PS permite intercambiar información y manejar las interrupciones. No obstante, su implementación en la PL hace que se incremente la latencia en el intercambio de información, comparado con otras soluciones en las que los accesos a periféricos internos se realiza de forma más rápida.

Con el objeto de reducir lo máximo posible la latencia producida por el intercambio de datos entre PL y PS, se ha optado por utilizar el controlador AXI DMA [Xil19b] para llevar a cabo el intercambio de información. En este sentido, el controlador DMA, a través del bus AXI4-Stream, permite reducir la latencia de la comunicación generando una corriente de datos continua. Además, este controlador permite realizar la escritura directa en la memoria DDR del dispositivo u otra región que sea accesible para el PS, como puede ser la memoria OCM.

Sin embargo, la memoria DDR, ya que la OCM no es cacheable, puede originar fallos de caché, provocados por una modificación en la memoria DDR que no haya sido modificada en la copia que se tiene en la caché. Para evitar esta situación el dispositivo Zynq® incluye un puerto ACP (*Accelerator Coherency Port*), que proporciona conectividad entre el PS y un acelerador *hardware* en la PL. El puerto ACP conecta la PL con la unidad SCU (*Snoop Control Unit*) del procesador ARM, la cual permite realizar una transacción DMA desde la PL manteniendo la coherencia de la caché de nivel uno (L1) y nivel dos (L2) de los procesadores. El uso del puerto ACP mejora el rendimiento de la caché, ya que se reduce la tasa de fallos de caché, decrementando los tiempos de acceso.

Como se ha mencionado en el capítulo 4 las tramas de estimación de canal se envían con menos frecuencia sin superar el tiempo de coherencia del canal $t_{C_{PLC}}$. Esto permite que la arquitectura propuesta pueda realizar la estimación de canal y cálculo de coeficientes en el PS. En este sentido, las tramas recibidas para llevar a cabo la estimación de canal se reciben en la PL, por tanto se debe notificar a PS que los datos están disponibles tras la transferencia DMA. Para ello, los elementos de sincronización entre PL y PS que están disponibles en Zynq® son principalmente las interrupciones. Éstas se pueden generar desde PL a PS; de esta forma, a través de las interrupciones es posible realizar la notificación a PS que los datos están disponibles en la región de memoria designada. Para ello, el controlador DMA incorpora unas señales de interrupción de forma que cuando finaliza la transacción DMA se habilita la señal de interrupción correspondiente. El PS recibe la interrupción y entonces procesa los datos para calcular la estimación de canal. Una vez que el PS termina de calcular los coeficientes del ecualizador, notifica a PL que se han modificado, actualizando así los coeficientes del ecualizador.

En la propuesta HW/SW de la figura 5.11 se muestran los módulos de la arquitectura que serán implementados en la parte *hardware* (PL) y qué módulos de la arquitectura se desarrollarán en *software* (PS). De esta forma, en el capítulo 4 se han detallado los módulos de la PL, sin embargo el controlador DMA no se ha detallado, ya que forma parte de los módulos disponibles en la herramienta de Xilinx Vivado.

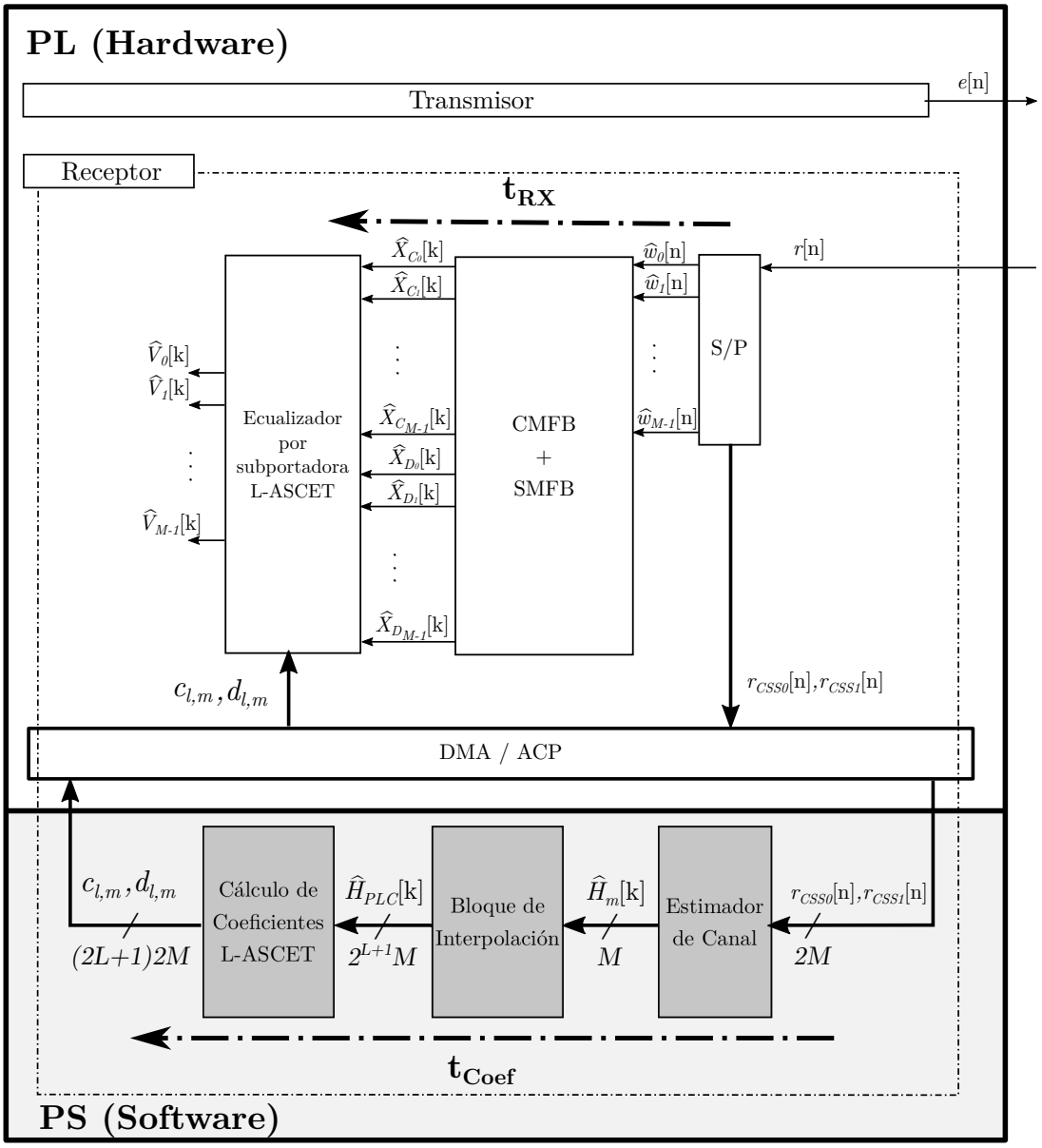


Figura 5.11: Diagrama de bloques de la arquitectura HW/SW propuesta para el sistema FBMC.

En la figura 5.11, se muestra el uso de controlador DMA a través del puerto ACP para mantener la coherencia de caché y además reducir la latencia en el intercambio de información entre el PS y la PL. Sin embargo, se requiere también una mejora en el rendimiento del sistema PS con el fin de que el tiempo de procesamiento t_{Coef} sea el mínimo posible. Para ello, se han planteado una serie de estrategias de aceleración *software* con el fin de mejorar el rendimiento, haciendo uso de todos los elementos disponibles de la arquitectura Zynq®. Estas estrategias de aceleración se detallan a continuación.

5.3.1 Estrategias de aceleración *software*

Entre las estrategias de aceleración *software* que se pueden llevar a cabo se pueden distinguir dos notablemente; por un lado las relacionadas con la aceleración de las operaciones que se realizan y por otro lado se tienen las propias de la arquitectura del dispositivo que se esté utilizando. En cuanto a las operaciones, esta estrategia lo que va a tratar es de realizar operaciones menos costosas dentro del algoritmo siempre que sea posible, para ello se plantea el uso de aritmética en punto fijo. La arquitectura RISC del procesador ARM realiza las operaciones con enteros en un solo ciclo por operación, mientras que las operaciones en punto flotante requieren de más ciclos. En cuanto a las aceleraciones debidas a la arquitectura del dispositivo, se pretende hacer uso de los elementos disponibles en el procesador, como la unidad de punto flotante o la unidad vectorial NEON. En la figura 5.12 y 5.13 se muestra el diagrama de bloques de la arquitectura Zynq® 7000 [Xil18d] y Zynq® UltraScale+ [Xil19c], en ella se muestran los elementos que compone el SoC y se puede comprobar cómo en ambas arquitecturas existe una unidad vectorial NEON. Además, la arquitectura de Zynq® UltraScale+ incorpora una unidad de procesamiento gráfico GPU junto al Cortex™ A53.

5.3.1.1 Uso de operadores en punto fijo

Como se ha mencionado, el procesador Cortex™ que integran los SoCs Zynq® 7000 y Zynq® UltraScale+ dispone de una unidad de punto flotante. Esta unidad de punto flotante permite realizar operaciones con formatos de punto flotante de simple y doble precisión; sin embargo la sobrecarga que introducen este tipo de formatos es mayor que la introducida por el formato de enteros con representación en punto fijo. Además, se debe realizar un estudio del *datapath* con el fin de conocer el error debido a la cuantificación del sistema.

En este caso los datos van a definirse de tipo entero, con un ancho de palabra de 32 bits. Dado que los datos de entrada son los símbolos transmitidos por el canal codificados en punto fijo con un formato Q[4,20], se va a aplicar la misma cuantificación con los 32 bits extendiendo la parte entera, de forma que la cuantificación utilizada será Q[12,20] con signo. El hecho de extender la parte entera, y no aumentar la precisión del sistema,

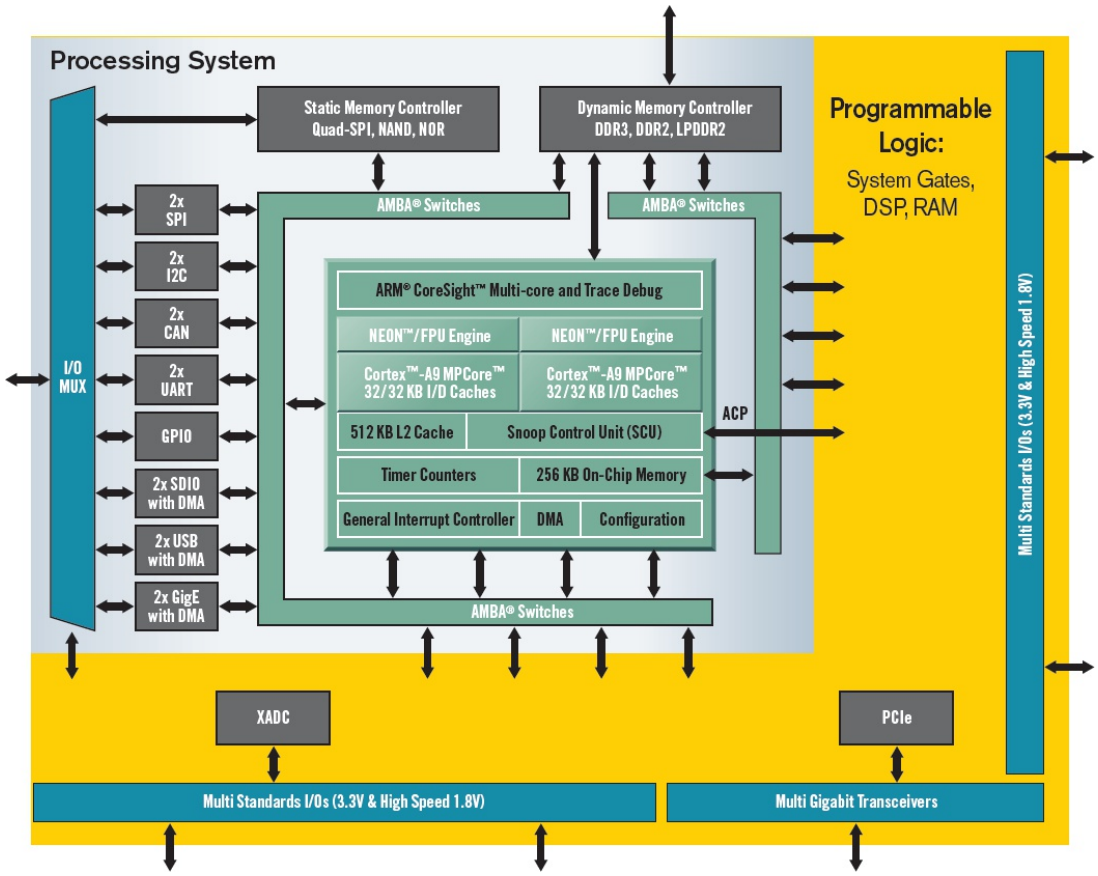


Figura 5.12: Diagrama de bloques de la arquitectura de Zynq® 7000, donde se muestran los elementos que componen el SoC [Xil18d].

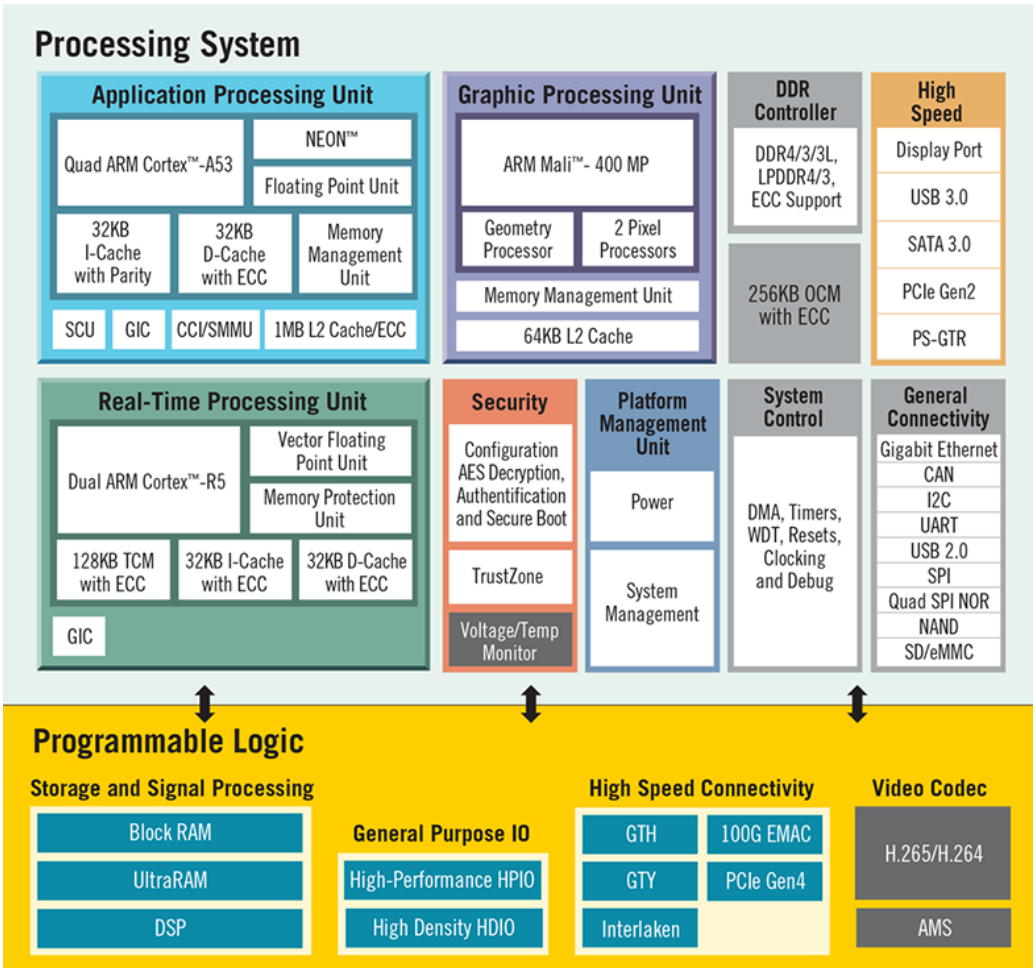


Figura 5.13: Diagrama de bloques de la arquitectura de Zynq® UltraScale+, donde se muestran los elementos que componen el SoC [Xil19c].

es debido a que se requiere un rango dinámico mayor para la parte entera a la hora de calcular la estimación del canal, por el alto nivel de ruido que presenta el canal PLC.

Todas las operaciones que se llevan a cabo para la estimación de canal y cálculo de coeficientes se realizan con enteros con representación en punto fijo, considerando que las operaciones de suma generan un bit más y en las operaciones de multiplicación se duplica el ancho de palabra. El cálculo de la FFT se realiza también en punto fijo, que, como se mostrará en la siguiente sección 5.3.1.2, se realiza mediante el uso de la unidad vectorial NEON. Sin embargo, la única operación que no es posible realizar en punto fijo es la división necesaria para el cálculo de coeficientes $c_{l,m}$ y $d_{l,m}$ del ecualizador L-ASCET. Esta operación, mostrada en (5.3), realiza la división del conjugado del canal estimado entre el módulo al cuadrado del mismo. Esta división se realiza en punto flotante, lo que conlleva la conversión de los datos enteros con representación en punto fijo a datos en punto flotante de simple precisión, para realizar la operación de división en punto flotante y posteriormente se pasa a punto fijo el resultado de la operación.

$$E_m(\Omega) = \frac{H^*_{Ch}(\Omega)}{\|H_{Ch}(\Omega)\|^2} \tag{5.3}$$

El hecho de operar en punto flotante esta parte es debido a que en la división se pierde precisión si se realiza con aritmética de punto fijo. Además, la operación de división tiene una latencia elevada, ya sea implementada en aritmética de punto fijo o punto flotante, por lo que se va a tratar de aprovechar la máxima precisión.

5.3.1.2 Uso de la Unidad Vectorial NEON

En la propuesta mixta HW/SW, mostrada en 3.4.3, se considera realizar la estimación de canal y cálculo de coeficientes del ecualizador ASCET en la parte *software*. Para llevar a cabo la estimación de canal, es necesario calcular la FFT de los símbolos recibidos, ya que el estimador LS realiza la división de la FFT de los símbolos recibidos $R_{CSS_x}[k]$, entre la FFT de los símbolos transmitidos originalmente CSS_x como se muestra en (3.20). Sin embargo, el cálculo de la FFT de los símbolos transmitidos originalmente CSS_x se puede realizar previamente y tenerlo almacenado en memoria, ya que es conocido con anterioridad. Por tanto, el número de FFTs que se requieren para realizar el cálculo de la estimación de canal se corresponde con el número de símbolos que se transmiten, que en este caso son dos símbolos piloto que forman parte del preámbulo de la trama y permiten realizar el cálculo de la estimación de canal.

El cálculo de la FFT es costoso desde el punto de vista computacional. En este caso se realizará en la parte *software* del SoC (PS). Entre la familia de MPSoC de Xilinx, Zynq® 7000 y Zynq® UltraScale+, se integran procesadores ARM diferentes en el PS. Para el caso Zynq® 7000, incluye un procesador Cortex™ A9, sin embargo Zynq® UltraScale+ integra el Cortex™ A53. La principal diferencia entre uno y otro es la capacidad de

operación, ya que el Cortex™ A9 solo tiene dos núcleos, mientras que el Cortex™ A53 tiene cuatro núcleos; además este último, permite instrucciones de 64-bit. En este caso, las estrategias de optimización *software* que se van a llevar a cabo se pueden aplicar a ambas familias de MPSoC (Zynq® 7000 y Zynq® UltraScale+), teniendo en cuenta que el conjunto de instrucciones y librerías que se usen deben ser compatibles para cada caso.

La principal ventaja que tiene la familia de procesadores ARM Cortex™ A es que cada núcleo incorpora una UV NEON con una arquitectura SIMD (*Single Instruction Multiple Data*), que mediante una única instrucción puede realizar la misma operación en paralelo a múltiples elementos de datos del mismo tipo y tamaño. El tipo de datos puede ser entero, punto flotante de simple precisión (*single-precision floating-point*) y punto flotante de precisión doble (*double-precision floating-point*). Esta unidad permite obtener los resultados de una región consecutiva de memoria más rápido que sin hacer uso de ella. De esta forma se va a utilizar para acelerar el cálculo de la FFT y del resto de partes de algoritmo siempre que se pueda aprovechar la arquitectura SIMD.

Existen dos métodos para hacer uso de la UV NEON que incluyen los procesadores Cortex™ A de ARM. El primero de ellos se basa en el soporte vectorial que incluye el propio compilador de GCC, es decir, en delegar en la capacidad que tenga el compilador para identificar las operaciones que puedan usar la unidad vectorial NEON y generar el código correspondiente. Sin embargo, el inconveniente de este método radica en que no siempre produce resultados satisfactorios. El otro de los métodos se basa en la codificación directa en ensamblador para usar la unidad vectorial NEON, lo que permite hacer un uso efectivo de la UV y obtener un mejor rendimiento. En este caso es posible encontrar librerías especializadas que contienen funciones en lenguajes de alto nivel. Esta librería contiene rutinas programadas en ensamblador, de forma que permite abstraer al programador de las instrucciones de bajo nivel y hacer un uso directo de la UV NEON.

La librería *open-source* Ne10 [Pro19] es compatible con las plataformas Linux, Android o iOS. Esta librería proporciona al programador una abstracción del uso de la UV NEON, ya que mediante el conjunto de funciones C que conforma la interfaz de programación de aplicaciones (API), oculta el manejo y la programación de la UV NEON en ensamblador, haciendo más sencillo su uso.

Para aprovechar al máximo el rendimiento de los procesadores ARM que incluye el SoC, y reducir las penalizaciones que incluiría el sistema operativo, la librería Ne10 se ha compilado para que pueda ser utilizada sin necesidad de disponer un sistema operativo en el ARM, conocido como modo *stand-alone* o *baremetal*. La librería Ne10 incluye una serie de funciones dedicadas al procesado digital de señales, entre las que se encuentra la función FFT o filtros FIR, IIR (*Infinite Impulse Response*). En la librería también se incluye una serie de funciones que permiten realizar operaciones vectoriales o matriciales, como el producto vectorial, escalar o el cálculo del determinante.

Para evaluar la aceleración que proporciona la UV NEON, se pretende evaluar el cálculo de la FFT haciendo uso de esta UV frente a la codificación en C. Aunque la FFT necesaria para el estimador LS es de 512 puntos, se va a extender el estudio a un número de puntos de la FFT más amplio, que va de los 128 a los 16384 puntos. Por otro lado, se ha realizado también para las dos familias de MPSoC de Xilinx, Zynq® 7000 y Zynq® UltraScale+.

La plataforma utilizada para la evaluación de Zynq® 7000 ha sido la ZC706, mientras que para la evaluación de Zynq® UltraScale+ se ha utilizado la ZCU102. En ambas se ha configurado la máxima frecuencia de operación de los microprocesadores, 800 MHz y 1.2 GHz respectivamente.

En la tabla 5.5 se muestran los tiempos medios obtenidos para los diferentes números de puntos de la FFT en 1000 ejecuciones del cálculo. En este caso, la FFT evaluada se ha considerado con entradas reales y salidas complejas con enteros con representación en punto fijo Q[12,20]. En términos generales, para los diferentes tamaños de la FFT evaluados, el tiempo obtenido utilizando la UV NEON, frente al tiempo sin ésta, es aproximadamente la mitad. Para el caso que concierne a este trabajo, la FFT de 512 puntos requiere de 16.62 µs haciendo uso de NEON y de 31.60 µs sin utilizar la UV para Zynq® 7000, y de 6.13 µs haciendo uso de NEON y de 10.72 µs para el caso de Zynq® UltraScale+.

Nº Puntos FFT		128	256	**512**	1024	2048	4096	8192	16384
Zynq®	C-Code	6.53 µs	13.14 µs	**31.60 µs**	64.18 µs	157.03 µs	354.68 µs	927.54 µs	1996.20 µs
7000	NEON	3.61 µs	7.19 µs	**16.62 µs**	34.15 µs	88.68 µs	209.09 µs	534.04 µs	1120.82 µs
Zynq®	C-Code	2.35 µs	5.82 µs	**10.78 µs**	27.26 µs	53.77 µs	171.64 µs	415.43 µs	1216.54 µs
UltraScale+	NEON	1.58 µs	3.24 µs	**6.13 µs**	13.81 µs	28.17 µs	72.88 µs	215.37 µs	536.79 µs

Tabla 5.5: Resultados de los tiempos de ejecución para el cálculo de la FFT mediante el uso de la UV NEON y la codificación en C (C-Code).

En la figura 5.14 se muestran los tiempos obtenidos anteriormente de forma gráfica. Se puede comprobar cómo a medida que aumenta el número de puntos de la FFT, los tiempos también crecen. Por último, la figura 5.15 muestra la aceleración que supone el uso de la UV NEON a la hora de realizar el cálculo de la FFT para los diferentes números de puntos mencionados anteriormente. En valor medio, la aceleración que supone es aproximadamente x1.75. Sin embargo, a medida que el número de puntos de la FFT aumenta, la aceleración que se obtiene es ligeramente inferior.

A lo largo de esta sección se ha mostrado la forma de acelerar la codificación *software* de la arquitectura mixta *hardware*/*software* propuesta. En este caso, los elementos de comunicación y sincronización entre ambas partes son cruciales para la implementación. Además, las estrategias de aceleración *software* mostradas, como el uso de operadores con enteros y el aprovechamiento de recursos del procesador como la UV NEON para el cálculo de la FFT, permiten optimizar la codificación *software*. En la siguiente sección se presenta la propuesta para un solo núcleo, haciendo uso de todas las estrategias de aceleración expuestas. Sin embargo, existe la posibilidad de utilizar más de un núcleo del procesador, escalando la arquitectura propuesta como se verá en las posteriores secciones.

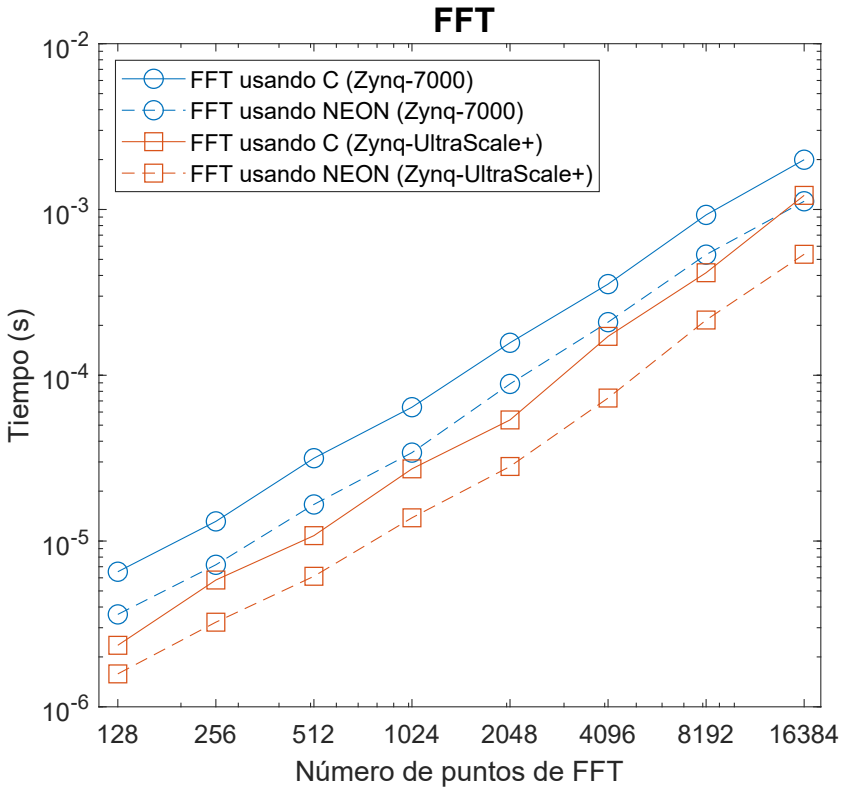


Figura 5.14: Tiempo medio de ejecución para el cálculo de la FFT utilizando Zynq® 7000 y Zynq® UltraScale+.

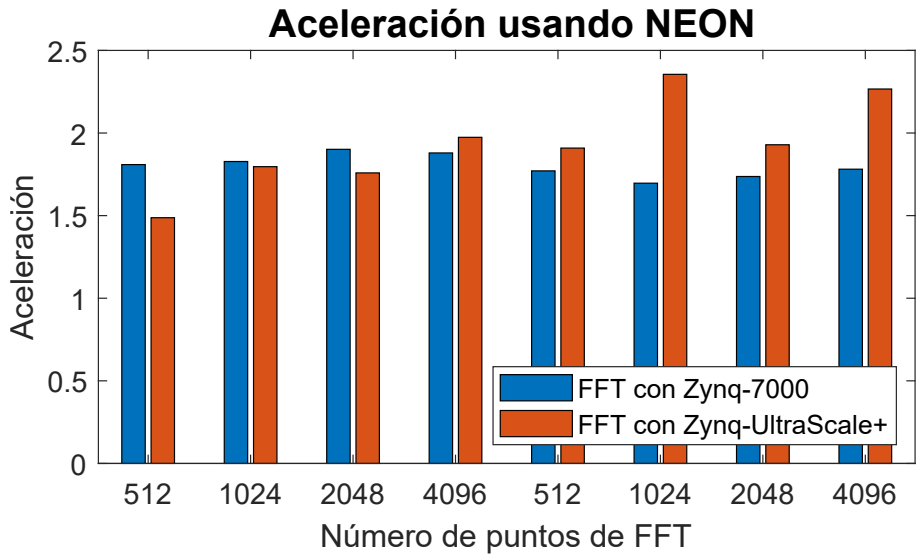


Figura 5.15: Gráfica con la aceleración obtenida mediante el uso de la unidad vectorial NEON en el cálculo de la FFT utilizando Zynq® 7000 y Zynq® UltraScale+.

5.4 Propuesta *software* mononúcleo

A partir de la descripción realizada en la sección anterior sobre la propuesta HW/SW se va a detallar la parte codificada en *software*. En esta primera propuesta sólo se hace uso de un único núcleo de los disponibles en Zynq® 7000, para llevar a cabo el cálculo de la estimación de canal y cálculo de coeficientes del ecualizador L-ASCET. Para ello, a partir de la descripción matemática recogida en 3.3.4 y 3.3.5, se ha realizado la programación en C de las etapas del algoritmo para obtener el tiempo de procesamiento y comprobar si se cumple $t_{Coef} \leq t_{RX}$.

Como se ha descrito anteriormente, el cálculo de la FFT se realiza mediante el uso de la unidad vectorial NEON y las librerías Ne10. Esto permite acelerar el cálculo de la FFT hasta aproximadamente x2. Además, el uso de operadores enteros con representación en punto fijo permite aligerar el cálculo de las operaciones requiriendo menos tiempo de procesamiento.

Por tanto, a partir de la descripción matemática del algoritmo y las estrategias de aceleración mostradas, se ha obtenido el tiempo de procesamiento que tiene el algoritmo en la codificación mononúcleo. Esta implementación se ha llevado a cabo en la plataforma de desarrollo ZC706, la cual incorpora un Zynq® 7000 con su respectivo procesador ARM Cortex™ A9. La frecuencia de reloj del PS se ha fijado al máximo permitido, 800 MHz. En la tabla 5.6 se muestra el tiempo de procesamiento para el cálculo de los coeficientes de los tres órdenes del ecualizador. Además, en la misma tabla se desglosan los tiempos de cada una de las etapas del algoritmo, donde se puede apreciar que el tiempo para el cálculo de coeficientes es mucho mayor para el caso 2-ASCET que en el caso 0-ASCET.

	0-ASCET		**1-ASCET**		**2-ASCET**	
Parte del Algoritmo	Media	Desviación Estándar	Media	Desviación Estándar	Media	Desviación Estándar
Estimador LS	52.3 µs	0.056 µs	52.3 µs	0.056 µs	52.3 µs	0.056 µs
Media del Canal	5.48 µs	0.066 µs	5.48 µs	0.066 µs	5.48 µs	0.066 µs
Interpolación del Canal	22.13 µs	0.049 µs	37.86 µs	0.068 µs	58.02 µs	0.048 µs
Cálculo de coeficientes	59.1 µs	0.061 µs	207.87 µs	0.055 µs	369.52 µs	0.062 µs
Tiempo de procesamiento en PS	**139.01 µs**	**0.073 µs**	**245.73 µs**	**0.096 µs**	**485.32 µs**	**0.088 µs**

Tabla 5.6: Tiempos de procesamiento determinados para los tres tipos de ecualizadores L-ASCET implementados en el sistema PS.

En este caso, ninguno de los tres órdenes de ecualizador permite alcanzar la premisa $t_{Coef} \leq t_{RX}$, ya que la latencia del receptor FBMC implementado en la PL es de aproximadamente 5500 ciclos, como se muestra en la tabla 4.12. La PL se configura con un periodo de reloj de 10 ns lo que equivale a que la latencia del receptor es de aproximadamente 55 µs.

En este caso las prestaciones del sistema en términos de SNR serán similares al mostrado en 5.2.2, donde la situación no cumple $t_{Coef} \leq t_{RX}$ y por tanto, los coeficientes se actualizan con cierto retraso, lo que provoca un deterioro de la señal recibida y se produce

una reducción del rendimiento del sistema en términos de SNR. Para conocer el número de paquetes de datos que no se ecualizan con los coeficientes correspondientes, se debe hacer la diferencia del tiempo de procesamiento de la tabla 5.6 con la latencia de 55 µs del sistema receptor, y dividirlo por el periodo de recepción de datos 8.192 µs. En este caso, para el 0-ASCET no se ecualizan correctamente 11 paquetes de 512 muestras, para el 1-ASCET 24 paquetes y para el 2-ASCET 53 paquetes. Si recordamos el valor arbitrario que se había asignado en la simulación, es muy similar a éste y por tanto, los resultados de SNR que se obtendrán del sistema se situarán en el mismo rango.

Con el fin de conocer el tiempo que cada parte del algoritmo requiere para llevar a cabo la estimación de canal y el cálculo de coeficientes, en la figura 5.16 se muestra de forma gráfica los tiempos de procesamiento obtenidos desglosados en cada una de las etapas del algoritmo. Cabe destacar que un orden de ecualizador mayor requiere un coste computacional mayor, dada la complejidad de las operaciones matemáticas vista en la sección 3.3.5. Se puede comprobar que el orden 2-ASCET invierte la mayor parte del tiempo en realizar el cálculo de coeficientes, mientras que en el 0-ASCET la mayor parte del tiempo la requiere la estimación LS que se realiza, donde se incluye el cálculo de las dos FFT involucradas en el algoritmo.

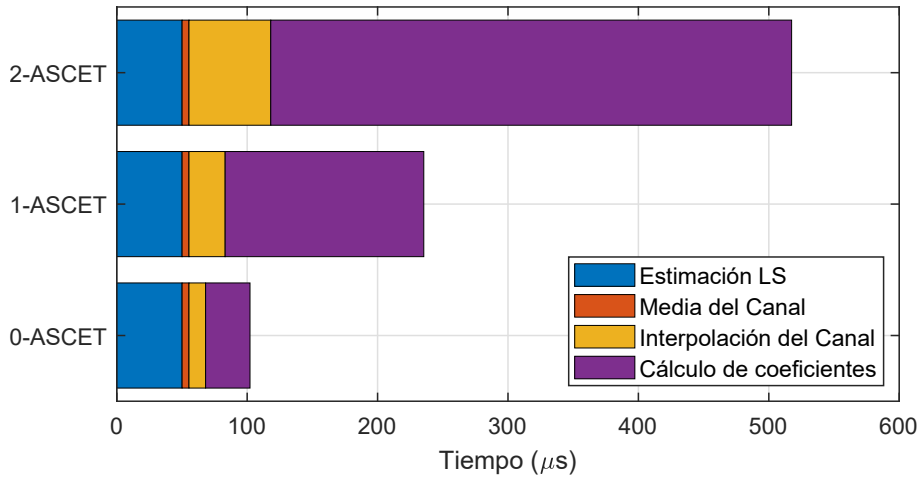


Figura 5.16: Tiempo de procesamiento para la codificación de la estimación de canal y cálculo de coeficientes desglosado por las etapas del algoritmo utilizando Zynq® 7000.

Para tener una idea sobre la carga computacional que presenta el algoritmo, en la tabla 5.7 se muestra el análisis de la complejidad del sistema global. Téngase en cuenta que el número de multiplicaciones y sumas que requiere el sistema global es significativo, teniendo en cuenta que la mayor parte de esas operaciones se realiza en el cálculo de la FFT. Además, el calculo de coeficientes realiza una operación de división que no está reflejada en la tabla 5.7. Hay que tener en cuenta que este tipo de operaciones son muy costosas desde el punto de vista computacional.

Módulo	Productos	Sumas	Memoria (KBytes)
Transmisor FBMC	$36 \cdot M$	$104 \cdot M$	$54 \cdot \log_2(M)$
Receptor FBMC	$66 \cdot M$	$197 \cdot M$	$108 \cdot \log_2(M)$
Ecualizador L-ASCET	$(2^{L+1}) \cdot M$	$(2^L) \cdot M$	$(2^L) \cdot \log_2(M)$
Estimador LS de canal	$52 \cdot M$	$174 \cdot M$	$(\log_2(M))^2$
Interpolación	–	$2 \cdot M \cdot (L+1)$	$5 \cdot (L+1) \cdot \log_2(M)$
Cálculo de coeficientes	$4 \cdot M$	$53 \cdot M$	$(2^{L+1}) \cdot \log_2(M)$

Tabla 5.7: Análisis de la complejidad de la propuesta realizada, incluyendo los módulos transmisor y receptor.

5.5 Propuesta *software* multinúcleo

Como se ha mostrado en la sección anterior, la solución mononúcleo proporciona resultados válidos. Sin embargo, el tiempo que precisa para llevar a cabo el cálculo de la estimación de canal y cálculo de coeficientes del ecualizador es superior al tiempo que el sistema receptor necesita para procesar las muestras de entrada del sistema FBMC receptor. En este caso, la ecualización se realiza con coeficientes no actualizados, produciendo una serie de datos que reducen ligeramente el SNR.

Para evitar la ecualización de los datos con coeficientes no actualizados, se pueden utilizar una serie de memorias intermedias, que almacenan la salida producida por el sistema FBMC receptor. Teniendo en cuenta el número de paquetes sin ecualizar correctamente, la memoria necesaria estaría en torno a 17 KB, 36 KB y 80 KB para el orden de ecualizador 0-, 1- y 2-ASCET, respectivamente. Sin embargo, otra opción pasa por acelerar aún más el procesamiento *software*; para ello, se propone una solución multinúcleo que comparte la carga computacional del algoritmo en dos o más núcleos.

Para ello hay que determinar la escalabilidad que tiene el algoritmo de estimación de canal y cálculo de coeficientes llevados a cabo en la parte *software* del SoC. Además, hay que tener en cuenta el intercambio de datos para llevar a cabo la codificación con más de un procesador. Por tanto se deberá determinar la región de memoria compartida por los núcleos utilizados y establecer un mecanismo de notificación entre los núcleos que permita sincronizar y gestionar todos ellos en las diferentes partes del algoritmo.

En la figura 5.17 se muestra el diagrama de bloques del algoritmo de estimación de canal y el cálculo de los coeficientes, en función del orden L del ecualizador. Como se puede observar en la figura 5.17, la primera parte del algoritmo tiene dos líneas claramente diferenciadas para ser procesadas en paralelo, siendo el estimador LS donde se procesa cada símbolo recibido. Siguiendo con el diagrama de bloques, tras las estimaciones LS se realiza la media de ambas estimaciones y se verifica el umbral para evitar el ruido impulsivo del canal PLC; en este caso, se puede realizar la media del canal y la verificación del umbral en paralelo para la mitad de las muestras procesadas, es decir, dividir el algoritmo en dos flujos de datos de $M/2$ señales, dado que cada una de las M señales es independiente entre sí. Del mismo modo, se procesarían los siguiente bloques en paralelo dividiendo el flujo de

datos, como sucede en el bloque de interpolación y la obtención de los coeficientes. Cabe mencionar que el número de señales que genera el bloque de interpolación es el número de señales M multiplicado por el factor de interpolación R, por tanto la división del *datapath* en este punto estará relacionada con el factor R.

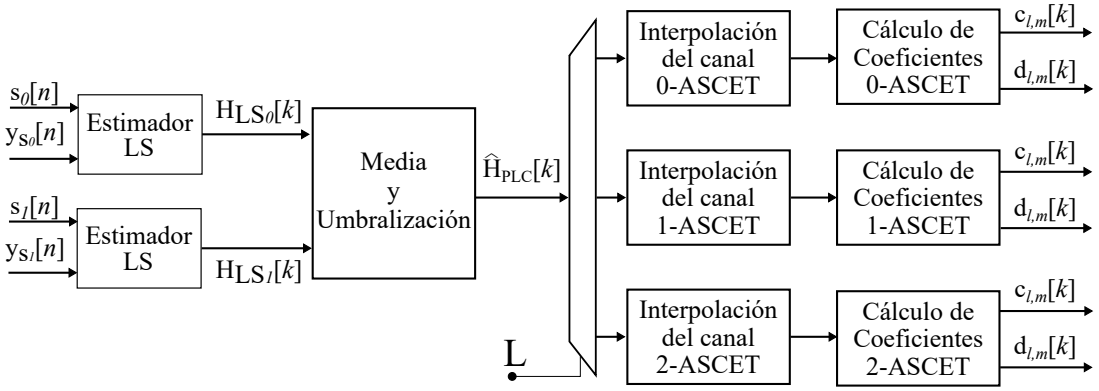


Figura 5.17: Diagrama de bloques del algoritmo de estimación de canal y cálculo de coeficientes, dependiendo del orden L del ecualizador ASCET.

Mediante la división del *datapath* en dos es posible escalar el algoritmo a dos núcleos, en lugar de uno. Sin embargo, a la hora de paralelizar este algoritmo aún más, se podría considerar seguir dividiendo cada una de las etapas en función del número de núcleos disponibles que, por lo general, suelen ser potencias de dos. Por tanto, para el caso de cuatro núcleos, se dividiría el *datapath* de M señales de entrada en cuatro *datapaths* de $M/4$ señales, y así para un número de núcleos mayor.

Sin embargo, la etapa de estimador LS lleva a cabo una FFT de M puntos para cada uno de los símbolos recibidos; en este caso, se podría adaptar el algoritmo para dividir la FFT de M puntos en dos FFTs de $M/2$ puntos, como se muestra en la figura 5.18. No obstante, aunque se divide la FFT, la última etapa de la misma se debe procesar en ambos núcleos. Esto se ha evaluado mediante el uso de la FFT de la librería Ne10, y se ha comprobado que el tiempo requerido en calcular la FFT de M puntos es similar al tiempo que se requiere para realizar cálculo de la FFT de $M/2$ puntos con el procesamiento de la última etapa. Ante esta situación, se puede optar por calcular la FFT de M puntos en dos núcleos y el resto de algoritmo extenderlo para los cuatro núcleos; o bien, se puede extender completamente el algoritmo en los cuatro núcleos dividiendo la FFT. Esto ocurre para un número de puntos de la FFT de 512 y es posible que para un número mayor sea posible acelerar el algoritmo dividiendo la FFT.

Una vez que se ha determinado la escalabilidad del algoritmo, hay que tener en cuenta la región de memoria compartida por ambos núcleos. La arquitectura de Zynq® 7000 y Zynq® UltraScale+ proporciona una región de memoria de tipo estático compartida integrada en el PS conocida como *On Chip Memory* (OCM), con un tamaño limitado (256 KB) pero tiene una latencia reducida. Además, se tiene una región de memoria externa, la memoria dinámica externa DDR, que presenta mayor latencia, pero es

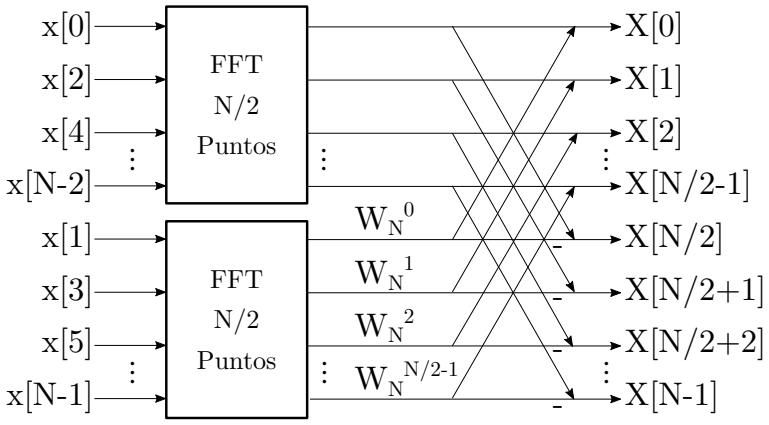


Figura 5.18: Diagrama del cálculo de la FFT de N puntos a partir de dos FFTs de $N/2$ puntos.

notablemente más grande. En este caso, dado que el tamaño de las tramas de datos contienen $M = 512$ muestras, siendo cada una de ellas de 32 bit, la capacidad de memoria requerida para almacenar una de las tramas es de 2 KB aproximadamente. Por tanto, la OCM es suficientemente grande como para utilizarla en esta aplicación como región de memoria compartida entre los núcleos del procesador.

5.5.1 Sincronización entre núcleos del procesador

Una vez que se ha definido la escalabilidad del sistema y la región de memoria compartida para los diferentes núcleos, queda por determinar la manera de sincronizar los diferentes núcleos. En este sentido, existen pequeñas diferencias entre la familia de Zynq® 7000 y Zynq® UltraScale+, que radican en la evolución de la arquitectura para el caso de Zynq® UltraScale+. A continuación, se detallan los métodos de sincronización para cada arquitectura, ya que la sincronización también varía en función del número de núcleos disponibles.

A grandes rasgos, la manera de sincronizar los procesadores se basa en el uso de instrucciones atómicas, que modifican el estado de una variable y lo hacen de forma exclusiva, o bien se puede hacer uso de las interrupciones, permitiendo así que ambos procesadores puedan interrumpirse entre ellos. De esta forma, es posible que los procesadores que componen la arquitectura puedan comunicar el punto de ejecución en el que se encuentran e intercambiar datos en la región de memoria compartida definida previamente. A continuación, se detalla en cada caso las características que se tienen a la hora de implementar las dos formas de sincronización que se han descrito en las plataformas Zynq® 7000 y Zynq® UltraScale+.

5.5.1.1 Sincronización en Zynq® 7000

Para el caso de Zynq® 7000, el cual incluye un ARM Cortex™ A9 de doble núcleo, es posible paralelizar el sistema hasta dos núcleos. La sincronización de ambos núcleos se puede realizar mediante interrupciones o instrucciones atómicas (*mutex*).

En cuanto al mecanismo de sincronización basado en interrupciones, se pueden diferenciar tres tipos de interrupciones que son soportadas por el controlador de interrupciones. Una de ellas es la interrupción privada de los periféricos (PPI), otra es la interrupción compartida de los periféricos (SPI) y por último se encuentra la interrupción generadas por *software* (SGI) que permiten provocar una interrupción desde el programa en ejecución. Además, las interrupciones SGI pueden ser provocadas hacia el mismo núcleo o hacia otro de los núcleos disponibles.

En la figura 5.19 se muestra un ejemplo de la gestión de una interrupción entre dos procesadores. En la figura se producen una serie de eventos que se detallan a continuación. En (1) el núcleo 1 alcanza un punto de su programa que genera una interrupción dirigida al núcleo 0; esta interrupción es procesada por el controlador general de interrupciones (GIC) (2), que es el encargado de interrumpir al núcleo 0 (3); dicho núcleo guarda su contador de programa (4) y atiende la interrupción mediante el manejador (*handler*) de la interrupción (5). Una vez que finaliza la atención de la interrupción (6), el contador de programa del núcleo 0 vuelve al punto del programa que se había quedado. Esto se puede realizar del mismo modo con la interrupción del núcleo 0 al núcleo 1. Sin embargo, esto presenta el inconveniente de que se requieren tantos identificadores de interrupción como puntos de sincronización se requieran en el programa; además la latencia que presenta es mayor debido al número de instrucciones que se ejecutan en el cambio de contexto de la interrupción. En cambio, permite al otro núcleo realizar cualquier otro tipo de tarea hasta que se reciba la interrupción.

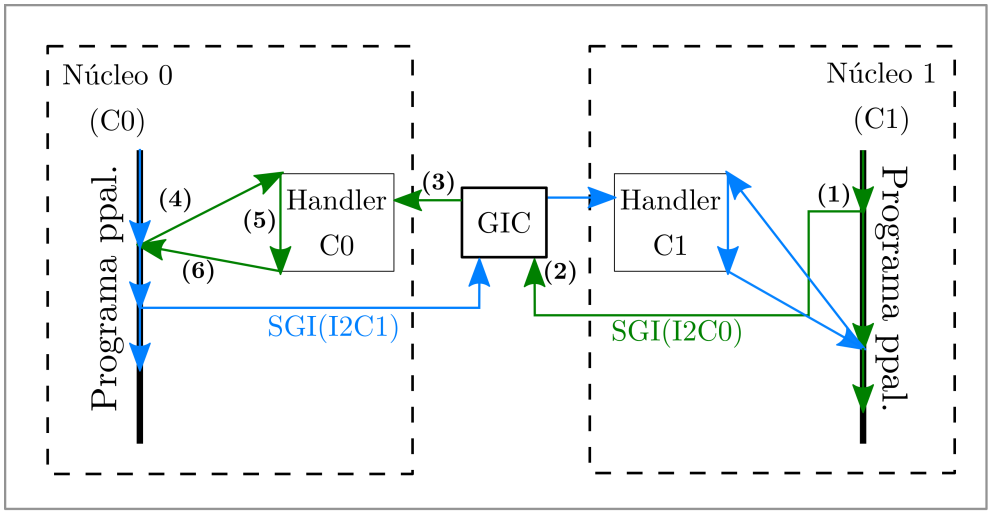


Figura 5.19: Diagrama que ilustra la gestión de las interrupciones en un sistema multiprocesador para dos núcleos.

Por otro lado, con el objetivo de reducir la latencia que pueda tener lugar con la sincronización, se puede hacer uso de las instrucciones atómicas `LDREX` y `STREX`, incluido en el conjunto de instrucciones del procesador ARM. Estas instrucciones requieren del uso de monitores exclusivos, los cuales rastrean y garantizan los accesos exclusivos de memoria; además estas instrucciones no pueden ser interrumpidas por otra instrucción.

Sin embargo, entre las regiones de memoria compartida de la arquitectura de Zynq® 7000, la caché L2 no soporta monitores exclusivos. Por otro lado, la memoria OCM tampoco tiene monitores exclusivos. Por último, la memoria compartida DDR podría soportar este tipo de instrucciones, pero para ello se debe tener la caché L2 deshabilitada en el rango de direcciones de la DDR donde se almacenan la variables de exclusión mutua.

En la figura 5.20 se muestra el ejemplo de gestión de una variable de exclusión mutua para Zynq® 7000. En este caso, los accesos a la variable se hacen a DDR y todas las variables de exclusión mutua (*mutex*) deberán inicializarse como desbloqueadas (*unlocked*). Cada uno de los procesadores realizará el bloqueo del *mutex* correspondiente. Mientras el *mutex* esté bloqueado, el otro procesador, si trata de acceder a la variable y ésta se mantiene bloqueada, se quedará intentándolo hasta que obtenga la dirección de la variable y la bloquee. Hasta entonces no avanzará con otra tarea que tenga programada, ya que consultará la variable de forma continua. Por tanto, se puede decir que esta solución presenta el inconveniente que bloquea al programa principal.

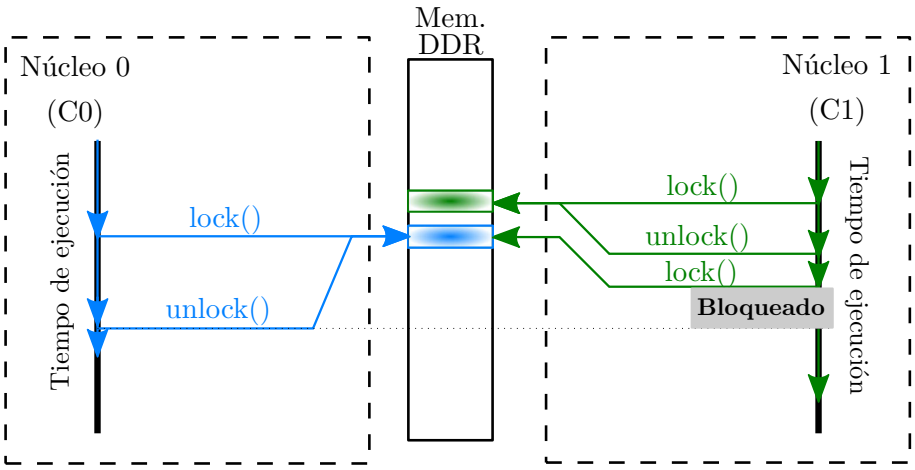


Figura 5.20: Diagrama que ilustra la gestión de las variables de exclusión mutua (*mutex*) en un sistema multiprocesador para dos núcleos.

Para evaluar qué opción de sincronización propuesta es la más adecuada, es necesario conocer el reparto que se realiza en cada uno de los núcleos del algoritmo de estimación de canal y cálculo de coeficientes, así como la dependencia que existe del número de núcleos disponibles.

5.5.1.2 Sincronización en Zynq® UltraScale+

Como se ha mencionado anteriormente, el ARM Cortex™ A53 que lleva Zynq® UltraScale+ es más potente que el ARM Cortex™ A9 que incorpora Zynq® 7000, pero además duplica el número de núcleos, que llega hasta cuatro. En la mejora de la arquitectura de Zynq® UltraScale+, se tiene que la OCM incluye monitores exclusivos, lo que facilita la programación del uso de instrucciones atómicas. Sin embargo, como se detallará a continuación, la caché de segundo nivel no da soporte a este tipo de instrucciones. Por tanto,

la sincronización entre núcleos con Zynq® UltraScale+ se puede realizar del mismo modo que con Zynq® 7000, mediante el uso de instrucciones atómicas, o bien utilizando las interrupciones pero adaptándolo a la evolución de la arquitectura de Zynq® UltraScale+.

En la figura 5.21 se muestra la arquitectura de la memoria OCM. Esta memoria tiene un tamaño de 256 KB, además, tiene ocho monitores exclusivos que permiten monitorizar hasta ocho transacciones de acceso exclusivo.

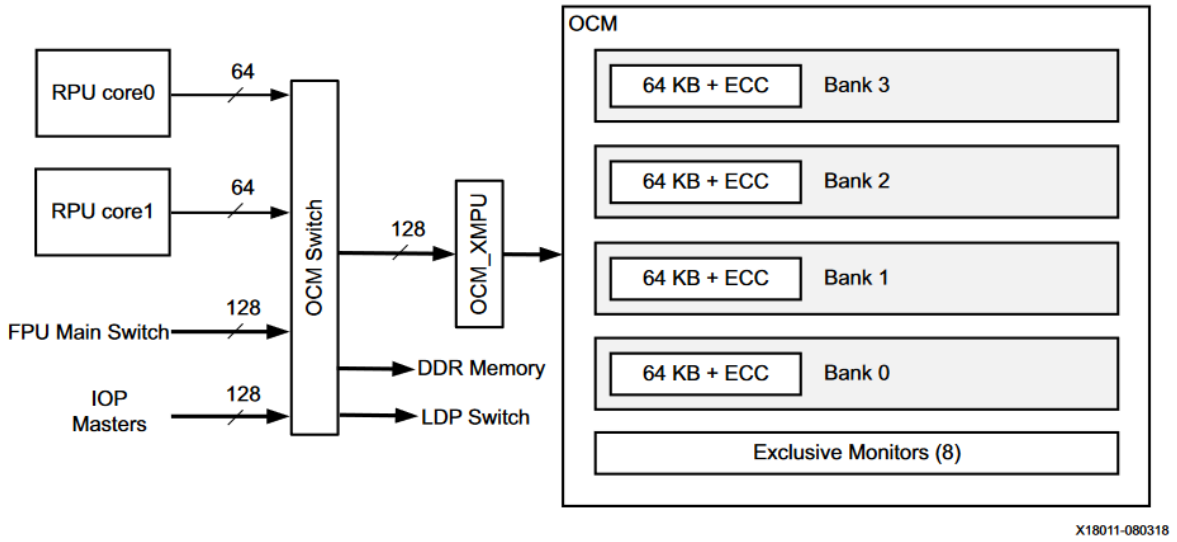


Figura 5.21: Diagrama de la arquitectura de la OCM de Zynq® UltraScale+ [Xil19c, Figura 18-1] .

A la hora de realizar la sincronización mediante el uso de instrucciones atómicas, las variables de exclusión mutua (*mutex*) se almacenarán en la OCM, ya que tiene los mecanismos necesarios para soportar este tipo de instrucciones. Además, son los únicos monitores exclusivos disponibles en el dispositivo, ya que la caché de segundo nivel no soporta este tipo de accesos.

Por otro lado, el controlador de interrupciones que incorpora Zynq® UltraScale+ incluye un canal de interrupción entre procesadores (IPI, *Inter-Processor Interrupts*), asociado a cada procesador, permitiendo la interrupción entre los procesadores y, además, el intercambio de mensajes y respuestas. A grandes rasgos, el IPI funciona del siguiente modo: el procesador de destino recibe una interrupción de otro procesador a través del IPI, éste atiende la interrupción, y además, puede acceder al *buffer* de mensajes donde se encuentra el mensaje origen de la interrupción. El protocolo de mensajes debe ser preestablecido antes del inicio de la comunicación y debe ser definido por el usuario. En este caso, los canales IPI pueden dirigirse a los procesadores del sistema (APU MPCore); a los procesadores de tiempo real (RPU); a la PMU (*Power Management Unit*); y por último, pueden destinarse a la aplicación que se desee, ya sea para el procesador o procesadores *soft* que se encuentren en la PL.

Con el objetivo de detallar la arquitectura de este elemento, a continuación se describe brevemente el mismo. En primer lugar los canales del IPI pueden ser configurables y suman 11 canales en total, repartidos como se indica a continuación.

- Canal 0: para APU MPCore.
- Canal 1-2: RPU0 y RPU1.
- Canal 3-6: del canal PMU0 al PMU3.
- Canal 7-10: no asignadas y disponibles para ser configurados por el usuario.

Cada canal está compuesto por un conjunto de seis registros. Éstos se pueden asociar con la actividad del canal, que será para el envío y la recepción de una interrupción. A continuación se detallan cada uno de estos registros, agrupados por el rol que plantean dentro de la interrupción.

- Envío de interrupción.
  - Registro de Trigger (TRIG), se trata de un registro de solo escritura, realiza el envío de una interrupción.
  - Registro de Observación (OBS), se trata de un registro de solo lectura, que permite observar el estado de una interrupción enviada.
- Recepción de interrupción.
  - Registro de estado de la interrupción (ISR), registro de lectura y escritura que sirve para comprobar el estado de la interrupción y limpiar el mismo.
  - Registro de máscara de la interrupción (IMR), registro de solo lectura.
    - Registro para habilitar la máscara del IMR (IER), registro de solo escritura y sirve para establecer el registro IMR a 1.
    - Registro para deshabilitar la máscara del IMR (IDR), registro de solo escritura y sirve para establecer el registro IMR a 0.

En el intercambio de mensajes, cada uno de esos canales es tratado como un agente, por lo que cuando se haga referencia al intercambio de mensajes se indicará que el agente 0 se corresponde con el canal 0, agente 1 con el 1, y así sucesivamente; excepto para los canales de la PMU (canales 3 a 6), cuyo conjunto se considera como un único agente. Para el intercambio de mensajes, los *buffers* permiten comunicación exclusiva entre cada emisor y cada receptor. Para ello, cada agente tiene 8 *buffers* de petición (*Request*) y 8 *buffers* de respuesta (*Response*). Esto quiere decir que, por cada agente se manejan 16 *buffers*, lo que hace un total de 128 *buffers* (el producto de los 8 agentes por los 16 *buffers*). En la figura 5.22 se muestra un diagrama de bloques de esta parte de la arquitectura del IPI.

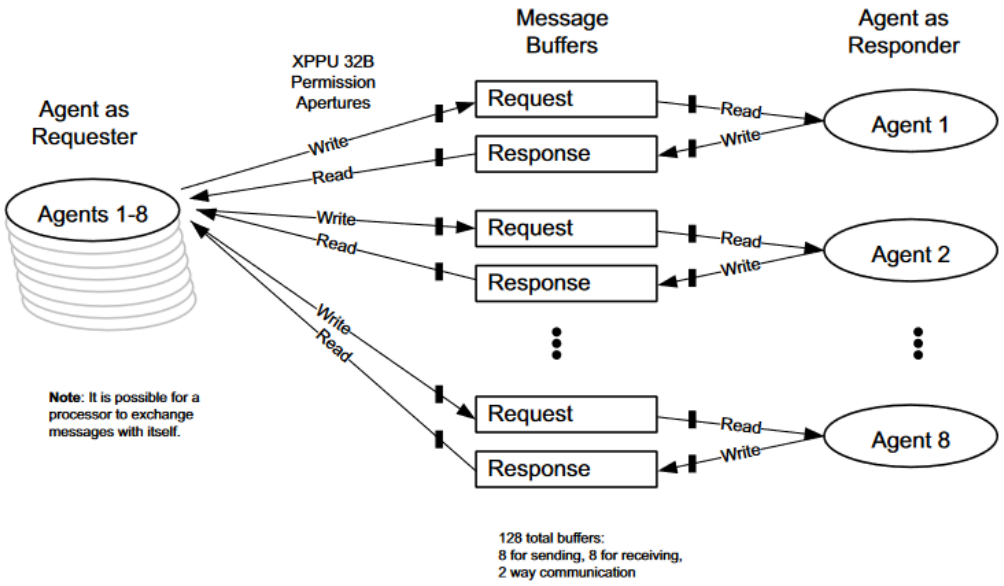


Figura 5.22: Arquitectura del intercambio de mensajes con IPI en Zynq® UltraScale+ [Xil19c, Figura 13-6].

En la figura 5.23 se muestra un ejemplo de envío y recepción de una interrupción a través del IPI. Para generar una interrupción, el emisor de la interrupción deberá escribir en el registro TRIG un 1 en el bit del correspondiente receptor, es decir, el canal 8 genera una interrupción al canal 2 y para ello escribe en su registro TRIG el 1 en el bit correspondiente al canal 2. El receptor recibe la interrupción entrante a través del registro de estado (ISR), el bit del registro de estado se corresponde con el canal que ha emitido la interrupción, en el caso de la figura 5.23 sería el 8. El emisor a su vez puede observar mediante el registro OBS el estado de las interrupciones que generó en los receptores. En el receptor se genera una interrupción en el GIC que es procesada como cualquier otro tipo de interrupción. El manejador (*handler*) de esta interrupción deberá verificar el registro ISR, y además, si el emisor ha escrito un mensaje en el *buffer* de salida, podrá leer dicho mensaje, así como escribir una respuesta.

En este sentido el IPI permite dos modos de comunicación. Ante la situación de que el emisor escriba el mensaje en el *buffer* y genere la interrupción escribiendo en el registro TRIG, el transmisor puede esperar la respuesta de su mensaje mediante *polling*, o bien, el receptor puede generar de nuevo una nueva interrupción a través del IPI y el transmisor gestionarla igual que se ha hecho hasta ahora. En cualquiera de los dos casos, es opcional el hecho de escribir una respuesta ante el mensaje recibido.

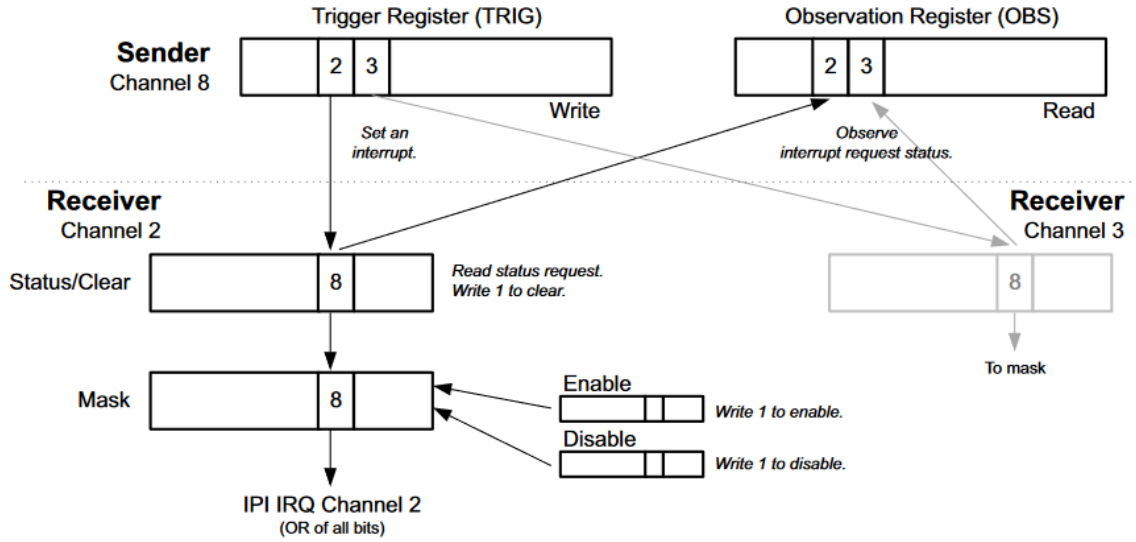


Figura 5.23: Diagrama del método de interrupción entre emisor y receptor utilizando IPI [Xil19c, Figura 13-4].

5.5.2 Propuesta de doble núcleo (Zynq® 7000)

Partiendo del algoritmo mostrado en la figura 5.17, se debe realizar una distribución entre los diferentes núcleos, añadiendo los puntos de sincronización donde se requiera un intercambio de información o se precisen los datos que está procesando el otro de los núcleos. En la figura 5.24 se muestra el diagrama de bloques que tendría el algoritmo, con los puntos de sincronización indicados en color verde. En este caso, se ha omitido la transacción DMA que tendría lugar entre la PL y el PS cuando se reciben los símbolos piloto para el cálculo de la estimación de canal, así como la transacción DMA entre PS y PL para traspasar al ecualizador L-ASCET los coeficientes calculados. Esto requiere utilizar un mecanismo de notificación que permita llevar a cabo la sincronización entre la PL y PS, independientemente del tipo de sincronización entre núcleos del procesador que se esté utilizando.

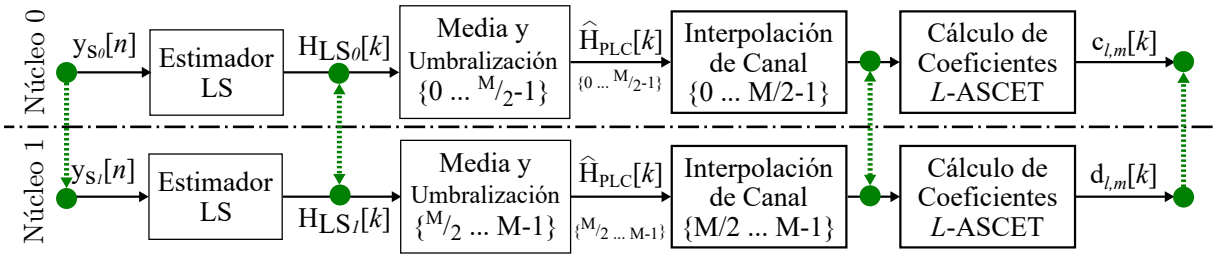


Figura 5.24: Diagrama de bloques del algoritmo de estimación de canal y cálculo de coeficientes para la versión de doble núcleo con los puntos de sincronización en verde.

Como se ha mencionado en el apartado 5.5.1.1, las dos formas que se tienen de sincronizar son haciendo uso de las interrupciones o hacer uso de las instrucciones atómicas. A continuación se va a realizar un breve estudio para cada una de las propuestas de

sincronización.

En la figura 5.25 se muestra el pseudocódigo utilizado para llevar a cabo la sincronización entre los dos núcleos del ARM Cortex™ A9 que integra Zynq® 7000. El procesador (núcleo 0) actúa como maestro frente al otro procesador (núcleo 1), que actúa como esclavo. El núcleo 0 se encarga de recibir la interrupción por parte de la PL que notifica que la transferencia DMA de las tramas de sincronismo ha concluido, y genera la correspondiente interrupción al núcleo 1 para que inicie el proceso de estimación LS.

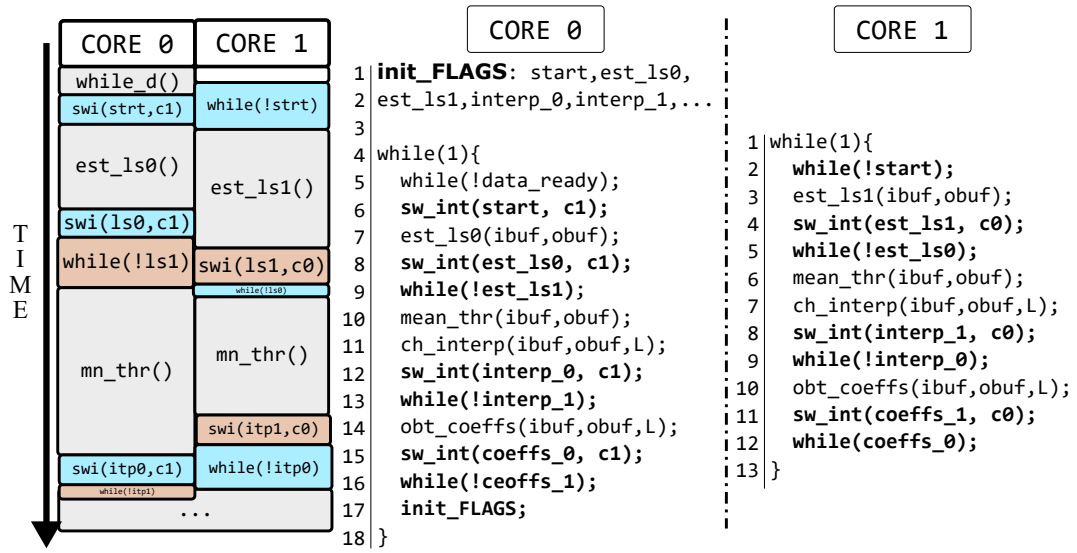


Figura 5.25: Línea temporal y pseudocódigo de la sincronización entre los núcleos mediante el uso de la gestión de las interrupciones.

Para la sincronización mediante el uso de instrucciones atómicas, se muestra el correspondiente pseudocódigo de parte del algoritmo en la figura 5.26. Como se ha mencionado anteriormente, el uso de variables de exclusión mutua hace que, cuando uno de lo núcleos del procesador trate de bloquear una variable, si ésta ya lo está porque otro núcleo lo haya hecho previamente, el programa principal se quedará en este punto tratando de hacerse con el control de la variable, esperando el momento en que sea desbloqueada.

Con todo esto, se ha evaluado la propuesta sobre la plataforma de desarrollo ZC706, la cual incorpora un MPSoC Zynq® 7000. La frecuencia de reloj del ARM Cortex™ A9 se ha seleccionado para que sea la máxima, es decir 800 MHz. El rendimiento se muestra en la figura 5.27 donde se recoge la media del tiempo de ejecución para los diferentes órdenes de ecualizador L-ASCET, sobre un total de un millón de ejecuciones. En la figura 5.27 se comparan ambos métodos de sincronización (interrupciones e instrucciones atómicas). El tiempo obtenido para el método de sincronización mediante instrucciones atómicas es ligeramente inferior (1.64 µs), frente a la sincronización mediante interrupciones (2.01 µs), siendo este tiempo el conjunto de todas las sincronizaciones llevadas a cabo en el algoritmo. En la tabla 5.8 se detallan los tiempos mostrados en la figura 5.27.

Con el objetivo de conocer la mejora que supone la propuesta para dos núcleos, frente a la propuesta de un solo núcleo, en la figura 5.28 se muestra el tiempo de ejecución obteni-

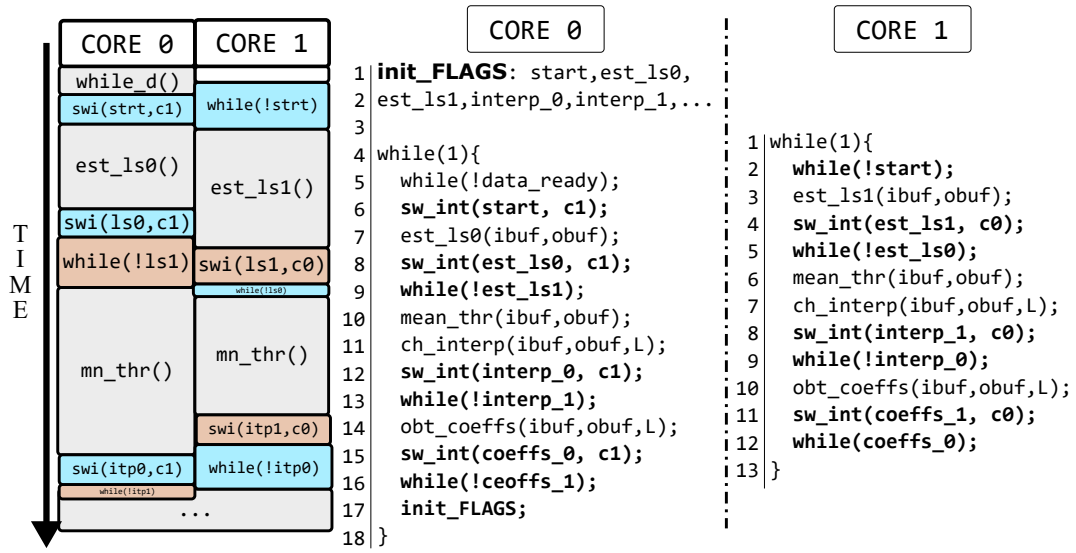


Figura 5.26: Línea temporal y pseudocódigo de la sincronización entre los núcleos mediante el uso de instrucciones atómicas y variables de exclusión mutua.

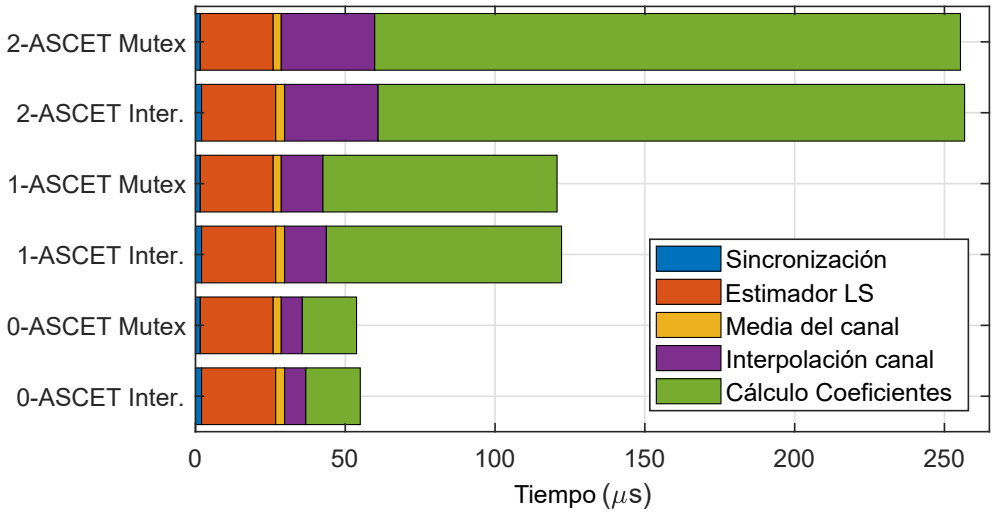


Figura 5.27: Tiempos de ejecución para los diferentes órdenes del ecualizador, desglosados en función de las tareas del algoritmo, así como los dos métodos de sincronización: interrupciones y variable de exclusión mutua.

	0-ASCET		1-ASCET		2-ASCET	
	Interrupciones	*mutex*	Interrupciones	*mutex*	Interrupciones	*mutex*
Sincronización	2.01	1.61	2.04	1.61	2.01	1.61
Estimación LS	24.83	24.83	24.85	24.83	24.83	24.83
Media del canal	2.91	2.66	2.88	2.66	2.91	2.66
Interpolación del canal	7.11	7.04	13.93	13.89	31.22	31.2
Cálculo de coeficientes	18.16	18.11	76.11	76.16	195.68	195.52
Tiempo total	**55.02**	**54.25**	**119.81**	**119.15**	**256.65**	**255.82**

Tabla 5.8: Desglose de los tiempos de ejecución del algoritmo de estimación de canal y cálculo de coeficientes obtenidos para los diferentes órdenes de ecualizador con ambos métodos de sincronización.

do para ambas. La propuesta de un solo núcleo está implementada sobre la plataforma de desarrollo ZC706 con la misma configuración mencionada anteriormente. Por otro lado, la sincronización utilizada para la propuesta de doble núcleo es la basada en interrupciones, pese a que el tiempo obtenido en llevar a cabo la sincronización es ligeramente superior al de las instrucciones atómicas. Sin embargo, la sincronización mediante interrupciones permite utilizar el procesador para otros propósitos mientras se espera recibir la interrupción.

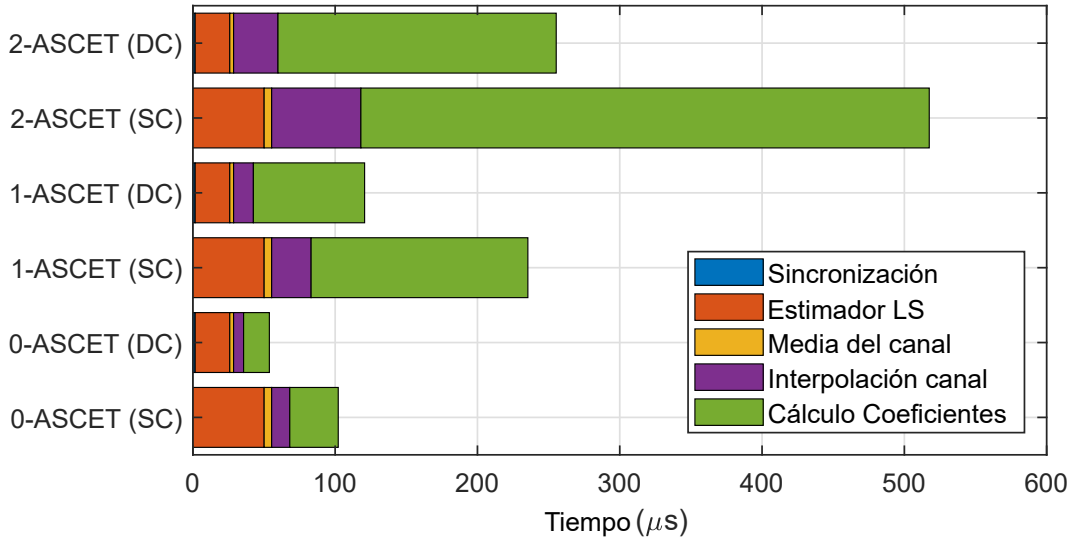


Figura 5.28: Tiempos de ejecución para los diferentes órdenes del ecualizador en las propuestas de mononúcleo (SC) y doble núcleo (DC) con sincronización mediante interrupciones.

La aceleración que proporciona la propuesta de doble núcleo, comparada con la solución mononúcleo, es de x2 aproximadamente. En la parte que se hace más notable esta aceleración es en el cálculo de coeficientes, para la cual, la propuesta de doble núcleo tarda la mitad de tiempo que la propuesta de un sólo núcleo.

Cabe destacar que en esta propuesta de doble núcleo el caso 0-ASCET permite actualizar los coeficientes antes que los datos alcancen el bloque receptor. Sin embargo, esta mejora no permite lo mismo para los otros dos órdenes de ecualizador L-ASCET. En el caso del 1-ASCET serían 8 los paquetes que se ecualizarían con los coeficientes no actualizados y 25 paquetes para el caso 2-ASCET. Si se utilizaran memorias intermedias se requerirían de 12 KB y 38 KB para el caso 1- y 2-ASCET respectivamente.

5.5.3 Propuesta para cuatro núcleos (Zynq® UltraScale+)

En la propuesta de cuatro núcleos es necesario aplicar la escalabilidad del algoritmo mostrada anteriormente, para utilizar los cuatro núcleos de manera óptima. De este modo, a partir de la figura 5.17 mostrada en la propuesta de doble núcleo, se ha extendido para la propuesta de cuatro núcleos. En este sentido, en la figura 5.29 se muestra el diagrama de bloques que presenta el algoritmo para la propuesta de cuatro núcleos. La zona sombreada

en gris es un transitorio inicial en el cual todos los núcleos notifican al núcleo 0 que se han iniciado correctamente. Además, la estrella roja representa la recepción de las tramas de estimación de canal que se notifica a través de una interrupción PL-PS.

Como se muestra en la figura 5.29, el número de puntos de sincronización que presenta el sistema es significativamente mayor, dada la subdivisión que se ha realizado en las diferentes funciones en los algoritmos. Sin embargo, la única parte del algoritmo que no se ha considerado dividir es el cálculo de la FFT, ya que el tiempo requerido para el cálculo de la FFT realizando la división es similar al cálculo sin realizar dicha división; además, a este tiempo habría que añadir la sincronización necesaria para realizar el cálculo de la última etapa, lo que conlleva un incremento en el tiempo total en esta etapa. Por tanto, se ha optado por no dividir la etapa de la FFT para evitar la sobrecarga temporal que supone.

Para realizar la sincronización entre los cuatro núcleos se utiliza el canal de interrupciones entre procesadores IPI explicado anteriormente. Cada uno de los núcleos del MPSoC utiliza un canal del IPI de los que no están asignados, siendo los canales 7 a 10, asignados a los núcleos 0 a 3 respectivamente. Cada generador de interrupción escribe un mensaje en el *buffer* y genera la interrupción al receptor. Éste recibe la interrupción y lee el mensaje, que actualiza la variable de estado y realiza la ejecución del algoritmo que le corresponda en función de dicha variable de estado. En la figura 5.30 se muestra de forma gráfica este proceso.

Con todo esto, se ha evaluado la propuesta en la plataforma de desarrollo Xilinx ZCU102, la cual incorpora el MPSoC Zynq® UltraScale+. La frecuencia de reloj del ARM Cortex™ A53 se configura al máximo, 1.2 GHz. El rendimiento obtenido se recoge en la tabla 5.9 donde se muestra la media del tiempo de ejecución para los diferentes órdenes de ecualizador *L*-ASCET, sobre una media de un millón ejecuciones. Dado que la arquitectura es diferente y la frecuencia de operación es superior, en la tabla 5.9 se han incluido los tiempos obtenidos para la propuesta mononúcleo y doble núcleo en esta arquitectura.

En la figura 5.31 se muestra de forma gráfica los tiempos de procesamiento obtenidos. La aceleración obtenida para el caso de doble núcleo con esta arquitectura se sitúa en torno a x1.7 para los diferentes órdenes del ecualizador L-ASCET. No obstante, para la propuesta de cuatro núcleos la aceleración es mayor excepto para el caso 0-ASCET, que se sitúa en x1.2, debido a la sobrecarga que supone la sincronización y la simplicidad del algoritmo. Esto hace que el rendimiento se vea reducido, incluso siendo inferior que para el caso de doble núcleo (x1.45). Para el caso 1-ASCET con cuatro núcleos la aceleración es de x1.83 y la mayor aceleración obtenida es para el caso 2-ASCET, siendo de x2.3.

Con esta propuesta de cuatro núcleos se consigue alcanzar la ecualización de los datos con los coeficientes adecuados para los órdenes de ecualizador 0- y 1-ASCET. Sin embargo, el número de paquetes que se procesan con los coeficientes inadecuados para el 2-ASCET

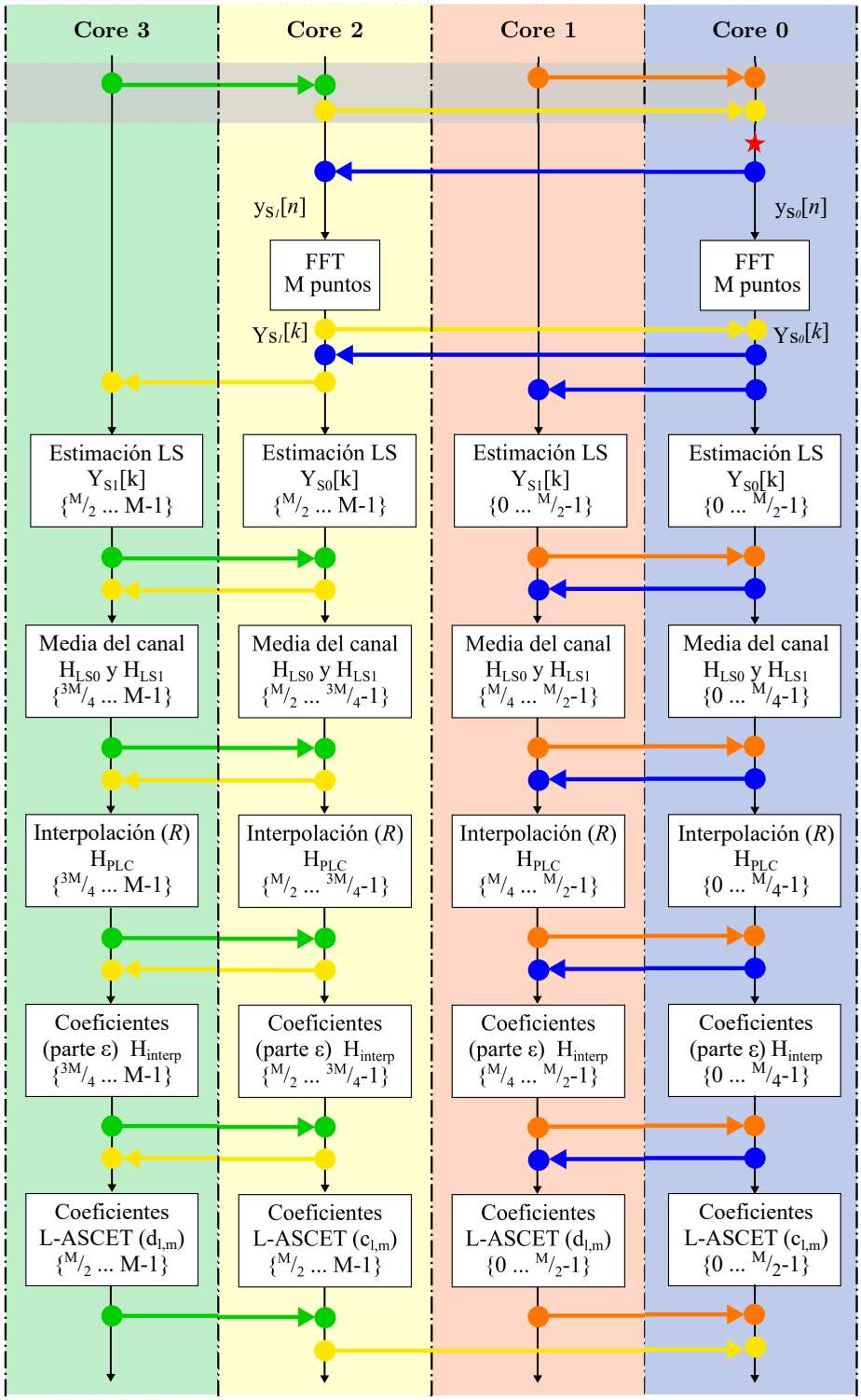
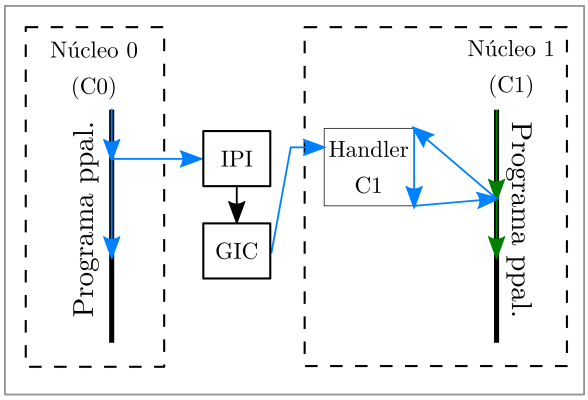


Figura 5.29: Diagrama de bloques del algoritmo de estimación de canal y cálculo de coeficientes para la versión de cuatro núcleos con los puntos de sincronización indicados.




```
void handlerIRQ(void *callBack){
    isrValue = getIsr(&ipiPtr);
    status = readMsg(ipiMaskSrc, buffRx, len);
    // Opcional
    status = writeMsg(ipiMaskDst, buffTx, len);
    // Clear IRQ Status
    clearIrq(&ipiPtr, ipiMaskSrc);
}
```

Figura 5.30: Diagrama y pseudocódigo de la gestión de la interrupción mediante IPI.

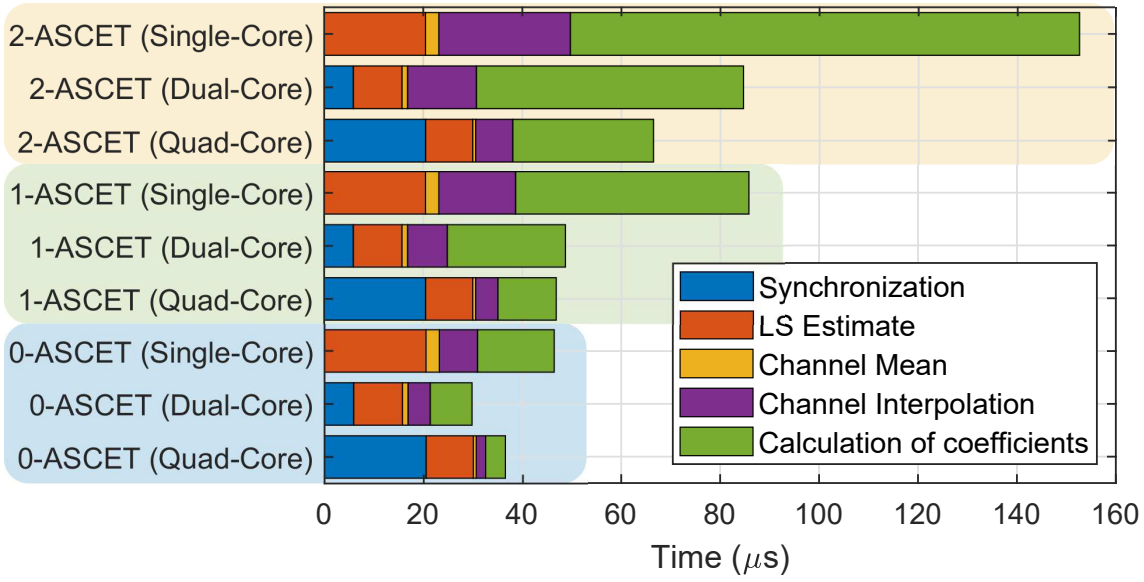


Figura 5.31: Tiempos de ejecución para los diferentes órdenes del ecualizador en las propuestas de mononúcleo, doble núcleo y cuádruple núcleo con sincronización mediante IPI en Zynq® UltraScale+.

			0-ASCET	1-ASCET	2-ASCET
Mono núcleo	**Tiempo Total**	**Media (µs)**	**43.300**	**85.744**	**152.476**
		Desv. Std.	**0.215**	**0.239**	**0.391**
	FFT de 512 Points	Media (µs)	13.677		
		Desv. Std.	0.020		
	Estimación LS	Media (µs)	6.781		
		Desv. Std.	0.008		
	Media del Canal	Media (µs)	2.691		
		Desv. Std.	0.010		
	Interpolación	Media (µs)	6.300	10.153	18.092
		Desv. Std.	0.001	0.008	0.053
	Calculo de Coeficientes	Media (µs)	14.044	41.736	94.312
		Desv. Std.	0.007	0.082	0.327
Doble núcleo	**Tiempo Total**	**Media (µs)**	**29.731**	**48.699**	**84.639**
		Desv. Std.	**0.180**	**0.171**	**0.331**
	Aceleración		**1.456**	**1.761**	**1.801**
	FFT de 512 Points	Media (µs)	6.626		
		Desv. Std.	0.010		
	Estimación LS	Media (µs)	3.198		
		Desv. Std.	0.013		
	Media del Canal	Media (µs)	1.142		
		Desv. Std.	0.013		
	Interpolación	Media (µs)	2.931	4.876	8.694
		Desv. Std.	0.006	0.013	0.020
	Calculo de Coeficientes	Media (µs)	6.886	20.694	48.713
		Desv. Std.	0.034	0.079	0.264
	Sincronización	Media (µs)	5.889		
		Desv. Std.	0.010		
Cuádruple núcleo	**Tiempo Total**	**Media (µs)**	**36.461**	**46.849**	**66.478**
		Desv. Std.	**0.161**	**0.119**	**0.343**
	Aceleración		**1.188**	**1.830**	**2.294**
	FFT de 512 Points	Media (µs)	6.504		
		Desv. Std.	0.025		
	Estimación LS	Media (µs)	3.002		
		Desv. Std.	0.028		
	Media del Canal	Media (µs)	0.592		
		Desv. Std.	0.029		
	Interpolación	Media (µs)	0.247	1.270	3.034
		Desv. Std.	0.024	0.026	0.026
	Cálculo de Coeficientes	Media (µs)	2.334	8.549	23.929
		Desv. Std.	0.048	0.041	0.160
	Sincronización	Media (µs)	20.488		
		Desv. Std.	0.025		

Tabla 5.9: Resultados de los tiempos de ejecución para el cálculo del algoritmo de estimación de canal y cálculo de coeficientes en Zynq® UltraScale+ con uno, dos y cuatro núcleos.

son únicamente 3. Si se comparan estos resultados frente a los 25 paquetes de la propuesta mononúcleo es una reducción considerable. Esto se traduce a una mejor SNR del sistema y/o un ahorro de memoria a la hora de evitar introducir memorias intermedias en la etapa de recepción.

5.6 Conclusiones parciales

Al principio del capítulo se ha justificado la idoneidad de la realización de la estimación de canal y la ecualización, cuantificando la mejora que supone en términos de SNR, RMSE y BER. Dada la complejidad del sistema y las exigencias del canal PLC, se ha optado por realizar dicha estimación a partir de una arquitectura mixta HW/SW. Sin embargo, la complejidad del algoritmo y las características temporales que se requieren hacen que los tiempos de procesamiento obtenidos en la parte *software* sean ligeramente superiores. Para reducir dichos tiempos, se han presentado una serie de propuestas de optimización, así como la posibilidad de escalar el sistema a más de un procesador.

En las propuestas que se han presentado a lo largo del capítulo, todas ellas producen datos válidos, sin embargo las soluciones mono y doble núcleo presentan la desventaja que no realizan una ecualización con los coeficientes actualizados. La forma de evitar esto, y realizar la ecualización correctamente, es definir una serie de memorias intermedias que almacenan los datos que produce el receptor que, de este modo, actúan de *buffer* para el ecualizador.

De las tres propuestas planteadas, mononúcleo, doble núcleo y cuatro núcleos, esta última es la que mayor reducción en términos de latencia consigue para el algoritmo de estimación de canal y cálculo de coeficientes presentado. Sin embargo, hay que tener en cuenta que aumentar el número de núcleos del sistema mejora el rendimiento a costa de aumentar el número de puntos de sincronización en el sistema. Esto hace que los órdenes más bajos, como el 0- ó 1-ASCET, presenten una sobrecarga temporal y la mayor parte del tiempo se utilice en llevar a cabo esta sincronización.

Por último, se ha realizado una evaluación de los diferentes métodos de sincronización entre núcleos. Dadas las características de cada uno de los modelos de FPGA (Zynq® 7000 y Zynq® UltraScale+), se han explorado las diferentes posibilidades que presentan cada uno de estos dispositivos. En definitiva, se ha optado por una sincronización basada en interrupciones, la cual permite utilizar el procesador para otros propósitos mientras se espera la señal de atención a la interrupción. El uso del IPI en Zynq® UltraScale+ simplifica la codificación para llevar a cabo la sincronización entre los núcleos; sin embargo, dadas las características de esta aplicación, se ha podido comprobar que realizar la sincronización entre los cuatro núcleos requiere de mayor tiempo debido al número de mensajes que se intercambian.

Capítulo 6

Conclusiones y líneas futuras

En este capítulo se recogen las conclusiones obtenidas a lo largo de la tesis. Además, se proponen las posibles futuras líneas de investigación que se derivan del trabajo. Por último, se detallarán las publicaciones relacionadas con la tesis, desglosándolas entre publicaciones en revistas internacionales, en congresos internacionales y en congresos nacionales.

6.1 Conclusiones

En esta tesis se ha presentado la definición de una arquitectura eficiente y configurable para un sistema transmisor y receptor PLC de banda ancha en un SoC. La definición de la arquitectura se ha realizado mediante el uso de herramientas de síntesis de alto nivel. Esta arquitectura se basa en técnicas de acceso al medio multiportadora para PLC; entre ellas, se centra en las basadas en bancos de filtros (FBMC).

Además, en la etapa de recepción se incorporan técnicas de estimación de canal y ecualización para mejorar la comunicación punto a punto. Las técnicas de estimación de canal y cálculo de coeficientes del ecualizador se implementan en la parte *software* del SoC, planteando una arquitectura mixta (HW/SW) que permite alcanzar las restricciones de tiempo real impuestas por el estándar IEEE 1901/2010.

A lo largo de la tesis, se han recogido una serie de conclusiones parciales que permiten corroborar la conclusión global expuesta. A continuación, se detalla cada una de estas conclusiones parciales plasmadas en los capítulos anteriores.

A lo largo del capítulo 2 ("Antecedentes y estado del arte"), se expone el papel fundamental que tiene la técnica de acceso al medio OFDM dentro de las comunicaciones. Sin embargo, las técnicas basadas en bancos de filtros (FBMC) han permitido mejorar la comunicación con la eliminación del prefijo cíclico de OFDM y, mediante el uso de filtros digitales, se reduce la interferencia entre portadoras (ICI). Por esta razón, los estándares IEEE 1901/2010 adoptan las técnicas multiporadora basadas en bancos de filtros en la definición de la capa física de PLC de banda ancha.

Sin embargo, las particularidades que presenta el medio de comunicación PLC hace que se pueda considerar un medio hostil para el envío de información. El canal PLC cuenta con diferentes tipos de ruido (coloreado, de banda estrecha, impulsivo), presenta notables interferencias y atenuaciones debidas a las reflexiones o multicaminos. Estos cambios no permiten considerar el canal como un sistema LTI, y establecer un modelo de canal que permita contrarrestar los efectos adversos que presenta el medio. Para poder considerar el canal PLC un sistema LTI, es necesario no superar el tiempo de coherencia del canal. Ésta es la razón principal por la que se debe incorporar algún mecanismo de estimación de canal en la etapa de recepción, sin superar el tiempo de coherencia del mismo.

De esta forma, es posible aplicar técnicas de ecualización de la señal recibida en el sistema receptor. Entre las técnicas mostradas, las basadas en filtros FIR, como ASCET, permiten mejorar la calidad de la señal recibida a costa de añadir complejidad al sistema. No obstante, estas técnicas son apropiadas para los sistemas basados en bancos de filtros como FBMC. Las técnicas de ecualización ASCET producen buenos resultados, sin embargo, plantean el inconveniente de que requieren duplicar los bancos de filtros en el receptor, incluyendo en paralelo el banco de filtros del seno modulado y del coseno modulado.

En el capítulo 3 ("Aspectos de la modulación FBMC, estimación e igualación de canal"), se ha descrito de forma matemática el transmisor y receptor FBMC para PLC. La descripción del sistema receptor FBMC incluye ambos bancos de filtros (seno modulado y coseno modulado) para aplicar el método de ecualizador ASCET. Este método de ecualización presenta cierta complejidad, sobre todo a la hora de determinar los coeficientes del ecualizador. Cabe destacar que la complejidad del cálculo de los coeficientes aumenta a la vez que se aumenta el orden de ecualizador (0-, 1- ó 2-ASCET), como se refleja en dicho capítulo.

En este capítulo también se detallan las restricciones temporales que fija el estándar IEEE 1901/2010 para los sistemas multiportadora PLC de banda ancha. A partir de estas restricciones temporales, se establecen los criterios para llevar a cabo un particionado HW/SW. El objetivo del particionado HW/SW es reducir el consumo de recursos lógicos utilizados por la FPGA, sobre todo a la hora de implementar el receptor FBMC. Por otro lado, a la hora de definir la arquitectura mixta HW/SW se ha establecido el criterio de particionado que se describe a continuación. Aquellas partes con restricciones temporales menos exigentes se implementarán en la parte *software* del dispositivo, siempre y cuando se alcancen dichas restricciones temporales. Para las otras partes, con restricciones temporales más exigentes, se llevará a cabo su implementación en la parte *hardware* del dispositivo. Este criterio trata de no penalizar al sistema global y alcanzar dichas restricciones temporales, buscando la implementación que minimice el consumo de recursos lógicos garantizando el funcionamiento en tiempo real. Al final del capítulo se realiza la propuesta de la arquitectura heterogénea.

En relación con la parte *hardware* del particionado, en el capítulo 4 ("Definición de aceleradores *hardware* para comunicaciones PLC utilizando HLS") se describe la implementación de transmisor y receptor FBMC mediante herramientas de síntesis de alto nivel. En este capítulo, se analiza el dimensionado del *datapath* en función de las características del dispositivo a utilizar, así como el rango y la precisión requerida de las variables utilizadas. Este estudio permite conocer el error debido a la cuantificación ajustándose a las restricciones físicas del dispositivo a utilizar.

Con el dimensionado establecido, en este mismo capítulo, se presentan las características que presenta la herramienta de síntesis Vivado HLS, indicando las principales directivas que permiten alcanzar las restricciones temporales establecidas por el estándar PLC. Este planteamiento trata de obtener las mejores prestaciones en términos de latencia y *throughput*, aplicando segmentación en el sistema. La principal ventaja que tiene el uso de herramientas HLS es la posibilidad de explorar diferentes soluciones dentro del espacio de diseño mediante el uso de directivas.

Los resultados mostrados tratan de buscar la solución más eficiente posible, minimizando el consumo de recursos con el fin de proporcionar las mejores prestaciones. Atendiendo al consumo de recursos más escasos dentro de la FPGA, como son las celdas DSP y bloques de memoria, el número de celdas DSP necesarias supera el centenar (13.67 % del dispositivo ZC706) para implementar la etapa de transmisión y recepción del sistema FBMC junto con el ecualizador ASCET. En cuanto al número de bloques de memoria, se acerca a un total de trescientos bloques de memoria de 18 KB (54.13 % del dispositivo ZC706) para la implementación del módulo transmisor y receptor con ecualizador ASCET. El significativo consumo de recursos obtenido viene dado principalmente por la implementación de la FFT, siendo esta operación la más costosa de realizar.

En el capítulo 5 ("Definición de los bloques *software* en la arquitectura heterogénea propuesta") se justifica el impacto que tiene sobre la arquitectura el realizar la estimación y ecualización del canal. Para ello se muestran los resultados de un estudio realizado bajo simulación, el cual proporciona una visión global de la calidad del sistema. En esta simulación, se tienen en cuenta aspectos del procesamiento *software*; de esta forma se puede estimar el impacto que tendría un retraso en el cálculo de la estimación de canal y cálculo de coeficientes en la calidad del sistema.

Por otro lado, con el fin de reducir los tiempos de procesamiento de la parte *software* se presentan un par de estrategias que permiten reducir los tiempos de procesamiento. Estas técnicas se basan en el uso de aritmética con enteros y el uso de la unidad vectorial NEON para acelerar el cálculo. El objetivo es que el tiempo de procesamiento de la parte *software* sea inferior a la latencia de la etapa de recepción de la parte *hardware*. En la solución mono núcleo propuesta no se alcanza dicho objetivo, por esta razón se proponen las soluciones multinúcleo.

En la solución multinúcleo se muestra la manera de realizar la sincronización entre núcleos. Entre las opciones disponibles el dispositivo Zynq® UltraScale+ proporciona en su arquitectura los mecanismos necesarios para establecer la comunicación/sincronización entre núcleos. El uso del IPI proporciona cierta sobrecarga en las comunicaciones pero facilita esta gestión. Los resultados obtenidos para este dispositivo son los que mejores prestaciones ofrecen, ya que la solución para dos o cuatro núcleos permite reducir los tiempos de procesamiento de forma considerable.

Por tanto, se puede decir que la arquitectura heterogénea propuesta permite establecer una comunicación punto a punto a través de la red eléctrica, haciendo uso de las técnicas de acceso al medio FBMC. Para mejorar dicha comunicación se propone la implementación de un estimador de canal y ecualizador ASCET dentro de la arquitectura.

Se puede ver como las comunicaciones PLC se utilizan para la comunicación remota de los medidores inteligentes de energía. Pero también hay que tener en cuenta que el espectro radioeléctrico tiene cada vez mayor demanda y es finito. Esto provoca que cada vez tenga mayor ruido e interferencia y se vea justificado el uso de técnicas de acceso al medio que presenten mayor robustez frente al ruido. Los resultados alcanzados en esta tesis se han aplicado a las comunicaciones PLC, pero en ella se muestran las técnicas de acceso al medio que pueden ser utilizadas en comunicaciones inalámbricas, como en la tecnología 5G.

6.2 Líneas futuras

En esta sección se proponen las futuras líneas de investigación derivadas de la elaboración de esta tesis.

1. Una primera vía es utilizar dispositivos que incluyan un mayor número de núcleos con el fin de mejorar las prestaciones del sistema implementado en PS. Los dispositivos más actuales tienen un número de núcleos mayor, como el ARM Cortex™ A55 o ARM Cortex™ A75, los cuales integran hasta ocho núcleos cada uno. Esta tendencia puede incorporase en la nueva generación de FPGAs.

2. Por otro lado, el cálculo de la FFT es muy costoso desde el punto de vista computacional. Aprovechando que Zynq® UltraScale+ tiene una GPU (ARM Mali™ 400MP), se propone como trabajo futuro el uso de esta unidad para acelerar el cálculo de la FFT y el resto de cálculos de la arquitectura que puedan ser procesados en la GPU. En relación a esto, la nueva generación de FPGAs (Intel® Stratix® 10 o Xilinx VERSAL) integra celdas DSP capaces de operar con aritmética de punto flotante de simple precisión, lo que mejoraría la prestación del sistema desarrollado en la PL para el cálculo de la FFT.

3. Un cuello de botella presente en la arquitectura presentada es la gestión de las interrupciones. Entre Zynq® 7000 y Zynq® UltraScale+ no existe una diferencia significativa en el tiempo obtenido para la atención de una interrupción. Para el caso del IPI, se proporciona una infraestructura para el intercambio de información entre procesadores que se suma al tiempo de procesar la interrupción. Por esta razón se propone como trabajo futuro el explorar y/o mejorar la gestión de las interrupciones en estos dispositivos.

4. Para completar la propuesta llevada a cabo en esta tesis se propone el uso de un *Analog Front End* (AFE) que proporcione acceso a la red eléctrica. De esta manera es posible realizar las pruebas experimentales en entornos realistas que permitan validar los resultados recogidos en capítulos anteriores. Es necesario que el AFE utilizado proporcione la tasa de transmisión definida en el estándar PLC de banda ancha.

5. Otro aspecto interesante para incluir en trabajos futuros es considerar alguna técnica de sincronismo entre transmisor y receptor que permita realizar la correcta demodulación de las tramas recogidas en la etapa de recepción. En este caso, es posible utilizar el preámbulo para enviar algún tipo de trama que permita la sincronización, al igual que se hace con las tramas de estimación de canal. Para no reducir la tasa de transmisión, se propone realizar la sincronización con las mismas tramas que se utilizan para obtener la estimación de canal.

6. Para proporcionar mayor robustez frente al ruido y las interferencias, y de este modo mejorar el rendimiento de las comunicaciones PLC, se propone la utilización de técnicas de modulación más complejas que las modulaciones PAM. Entre ellas, la modulación de amplitud en cuadratura (QAM) viene usándose en las comunicaciones ADSL o las transmisiones de televisión y es posible encontrar trabajos donde se utiliza con OFDM. En el nuevo estándar 5G, donde se utilizan también técnicas de acceso al medio basadas en FBMC, se utiliza FBMC/QAM, mejorando la calidad del sistema. Sin embargo, es necesario duplicar el banco de filtros en el transmisor, componiendo el banco del coseno (CMFB) para la parte real de la modulación QAM y el banco de filtros del seno (SMFB) y para la parte imaginaria de la modulación QAM. Del mismo modo, en el receptor se requerirá duplicar los bancos de filtros, para la parte real e imaginaria de la modulación.

7. Otra posibilidad que se propone es la utilización de sistemas con múltiples entradas y múltiples salidas (MIMO) para mejorar las prestaciones de las comunicaciones. Esta tesis permite la comunicación a través de la línea de fase y neutro de la red eléctrica, pero se está estudiando la posibilidad de utilizar la línea de protección de

tierra para expandir el sistema a un sistema MIMO PLC. La combinación de las tres líneas (fase, neutro y tierra) puede dar lugar a sistemas MIMO de tres entradas y tres salidas. Esto requiere de la caracterización del medio de comunicación, ya que las transmisiones entre fase-tierra o neutro-tierra presentarán características diferentes entre ellas, y diferentes al medio fase-neutro ya caracterizado.

6.3 Publicaciones relacionadas con la tesis

A continuación, se enumeran las publicaciones en revistas indexadas y las contribuciones realizadas en congresos internacionales y nacionales generadas a partir de la elaboración de esta tesis.

6.3.1 Publicaciones en revistas internacionales

- R. Nieto; R. Mateos; A. Hernández. "HW/SW Architecture for a Broadband Power-Line Communications System with LS Channel Estimator and ASCET Equalizer". *IEEE Transactions Industrial Informatics*. pp. 1-12, 2019. (provisionally accepted for publication)

- R. Nieto; R. Mateos; A. Hernández. "Performance Evolution of PLC Channel Estimator and ASCET equalizer in a FBMC Transmultiplexer using MPSoC Zynq UltraScale+". *IEEE Transactions on Circuits and Systems I*. pp. 1-8, 2020 (in first review round)

6.3.2 Publicaciones en congresos internacionales

- R. Nieto; R. Mateos ; A. Hernández; E. Díaz. "Dual-Core Architecture for PLC Channel Estimator and ASCET Equalizer in a FBMC Transmultiplexer". In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (2019) pp. 1342-1345.
doi:10.1109/ETFA.2019.8869476

- F. Membibre; R. Nieto; A. Hernández. "Design of FPGA-based Architecture for an Analog Front-End in Broadband PLC". In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (2019), pp. 1-4.
doi:10.1109/etfa.2019.8869166

- R. Nieto; R. Mateos; A. Hernández. "Finite Precision Analysis of FPGA-based Architecture for FBMC Transmultiplexer in Broadband PLC". In *2018 33rd Conference on Design of Circuits and Integrated Systems (DCIS)* (2018), pp. 1–6.
doi:10.1109/DCIS.2018.8681466

6.3.3 Publicaciones en congresos nacionales

- R. Nieto; R. Mateos; A. Hernández. "Aplicación de Unidad Vectorial NEON para la Implementación de un Transmultiplexor FBMC con Estimador e Igualador de Canal". In *2019 XXVI Seminario Anual de Automática, Electrónica Industrial e Intrumentación (SAAEI)* (2019), pp. 1-6.
- F. Membibre; R. Nieto; A. Hernández. "Arquitectura para comunicaciones PLC de Banda Ancha basada en un Analog Front-End". In *2019 XXVI Seminario Anual de Automática, Electrónica Industrial e Intrumentación (SAAEI)* (2019), pp. 1-6.
- R. Nieto; A. Hernández; R. Mateos. "Modelado HLS de un Transmultiplexor Multiportadora para PLC: Comparativa de Prestaciones". In *2018 XXV Seminario Anual de Automática, Electrónica Industrial e Intrumentación (SAAEI)* (2018), pp. 1-6.

Bibliografía

[Alh01] J. Alhava; M. Renfors. "Adaptive Sine-modulated/Sosine-modulated Filter Bank Equalizer for Transmultiplexers". In *European Conference on Circuit Theory and Design, Espoo, Finland* (2001) pp. 337–340.
http://lib.tkk.fi/Books/2001/isbn9512263378/papers/1049.pdf

[Alo17] A. Aloui; O. Ben Rhouma; C. Rebai. "Impact of Equalization on Hybrid Communication System for Smart Grid". In *2017 International Symposium on Networks, Computers and Communications (ISNCC)* (May 2017) pp. 1–6.
doi:10.1109/ISNCC.2017.8072017

[Ana08] J. Anatory; N. Theethayi; R. Thottappillil; M.M. Kissaka; N.H. Mvungi. "Broadband power-line communication channel model: Comparison between theory and experiments". In *2008 IEEE International Symposium on Power Line Communications and Its Applications* (2008) pp. 322–324.
doi:10.1109/ISPLC.2008.4510447

[And09] N. Andreadou; F. Pavlidou. "PLC Channel: Impulsive Noise Modelling and Its Performance Evaluation Under Different Array Coding Schemes". *IEEE Transactions on Power Delivery.* 24 (2), pp. 585–595, April 2009.
doi:10.1109/TPWRD.2008.2002958

[Ant11] M. Antoniali; A.M. Tonello; M. Lenardon; A. Qualizza. "Measurements and analysis of PLC channels in a cruise ship". In *2011 IEEE International Symposium on Power Line Communications and Its Applications* (April 2011) pp. 102–107.
doi:10.1109/ISPLC.2011.5764372

[Bal10] L.G. Baltar; A. Mezghani; J.A. Nossek. "MLSE and MMSE subchannel equalization for filter bank based multicarrier systems: Coded and uncoded results". In *2010 18th European Signal Processing Conference* (Aug 2010) pp. 2186–2190.
https://ieeexplore.ieee.org/document/7096630

[Bal18] R. Bale; T.S. Seetharamaiah; Y.S. Reddy; A. Dubey; T. Panigrahi. "OFDM vs FBMC: A comparative analysis for broadband-PLC". In *2018 10th International Conference on Communication Systems & Networks (COMSNETS)* (2018) pp. 480–483.
doi:10.1109/COMSNETS.2018.8328249

[Bas09] E. Bassi; F. Benzi; L. Almeida; T. Nolte. "Powerline communication in electric vehicles". In *2009 IEEE International Electric Machines and Drives Conference* (may 2009) pp. 1–5.
doi:10.1109/iemdc.2009.5075439

[Bas18] D. Bastos; M. Shackleton; F. El-Moussa. "Internet of Things: A Survey of Technologies and Security Risks in Smart Home and City Environments". In *Living in the Internet of Things: Cybersecurity of the IoT - 2018* (Institution of Engineering and Technology, 2018) .
doi:10.1049/cp.2018.0030

[Bel10] M. Bellanger; M. Renfors; T. Ihalainen; C.A.F. da Rocha. "OFDM and FBMC transmission techniques: a compatible high performance proposal for broadband power line communications". In *ISPLC2010* (March 2010) pp. 154–159.
doi:10.1109/ISPLC.2010.5479918

[Ber69] G. Bergland. "Fast Fourier transform hardware implementations–A survey". *IEEE Transactions on Audio and Electroacoustics.* 17 (2), pp. 109–119, June 1969.
doi:10.1109/TAU.1969.1162048

[Ber02] I. Berenguer; I.J. Wassell. "Efficient FMT equalization in outdoor broadband wireless systems". In *Proc. IEEE International Symposium on Advances in Wireless Communications, Victoria, Canada* (2002) pp. 1–2.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.1446&rep=rep1&type=pdf

[Ber14] L.T. Berger; A. Schwager; P. Pagani; D. Schneider. *MIMO power line communications: narrow and broadband standards, EMC, and advanced processing* (CRC Press, 2014).

[Ber15] L.T. Berger; A. Schwager; P. Pagani; D.M. Schneider. "MIMO Power Line Communications". *IEEE Communications Surveys & Tutorials.* 17 (1), pp. 106–124, 2015.
doi:10.1109/comst.2014.2339276

[Bog14] M. Bogdanović. "Frequency domain based LS channel estimation in OFDM based Power line communications". *Automatika.* 55 (4), pp. 487–494, jan 2014.
doi:10.7305/automatika.2014.12.639

[Bop15] V. Boppana; S. Ahmad; I. Ganusov; V. Kathail; V. Rajagopalan; R. Wittig. "UltraScale+ MPSoC and FPGA families". In *2015 IEEE Hot Chips 27 Symposium (HCS)* (2015) pp. 1–37.
doi:10.1109/HOTCHIPS.2015.7477457

[Bri17] M. Brison; M. Bensetti; R. De Lacerda; F.V.D. Santos; C. Taurand. "Optimised PLC power transfer on avionic DC Power Lines: Coupling circuit and lightning protection". In *2017 IEEE International Symposium on Power Line Communications and its Applications (ISPLC)* (April 2017) pp. 1–6.
doi:10.1109/ISPLC.2017.7897110

[Bul16] S.S.K.C. Bulusu. *Performance Analysis and PAPR Reduction Techniques for Filter-Bank based Multi-Carrier Systems with NonLinear Power Amplifiers.* Ph.D. thesis, Mahindra École Centrale, Hyderabad, India, 04 2016.
https://tel.archives-ouvertes.fr/tel-01401456/document

[Cai01] G. Caire; U. Mitra. "Structured multiuser channel estimation for block-synchronous DS/CDMA". *IEEE Transactions on Communications.* 49 (9), pp. 1605–1617, Sep. 2001.
doi:10.1109/26.950347

[Can11] F.J. Canete; J.A. Cortes; L. Diez; J.T. Entrambasaguas. "A channel model proposal for indoor power line communications". *IEEE Communications Magazine.* 49 (12), pp. 166–174, 2011.
doi:10.1109/MCOM.2011.6094022

[Can12] M.J. Canet; J. Valls; V. Almenar; J. MarÃn-Roig. "FPGA implementation of an OFDM-based WLAN receiver". *Microprocessors and Microsystems.* 36 (3), pp. 232 – 244, 2012.
doi:https://doi.org/10.1016/j.micpro.2011.11.004

[Che06] H.H. Chen; Y.C. Yeh; X. Zhang; A. Huang; Y. Yang; J. Li; Y. Xiao; H. Sharif; A. Vinck. "Generalized pairwise complementary codes with set-wise uniform interference-free windows". *IEEE Journal on Selected Areas in Communications.* 24 (1), pp. 65–74, jan 2006.
doi:10.1109/jsac.2005.858878

[Cho08] M. Choi; S. Ju; Y. Lim. "Design of integrated meter reading system based on power-line communication". In *2008 IEEE International Symposium on Power Line Communications and Its Applications* (apr 2008) pp. 1–5.
doi:10.1109/isplc.2008.4510439

[Com01] I.E. Commission. "IEC TS 61334-5-4:2001 Distribution automation using distribution line carrier systems - Part 5-4: Lower layer profiles - Multi-carrier

modulation (MCM) profile". Technical Specification, International Electrotechnical Commission, June 2001.
https://webstore.iec.ch/publication/5311

[Coo65] J.W. Cooley; J.W. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series". *Mathematics of Computation*. 19 (90), pp. 297–301, 1965.
doi:10.2307/2003354

[Cor06] F.J.C. Corripio; J.A.C. Arrabal; L.D. del Rio; J.T.E. Munoz. "Analysis of the Cyclic Short-Term Variation of Indoor Power Line Channels". *IEEE Journal on Selected Areas in Communications*. 24 (7), pp. 1327–1338, 2006.
doi:10.1109/JSAC.2006.874402

[Cro14] L.H. Crockett; R.A. Elliot; M.A. Enderwitz; R.W. Stewart. *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc* (Strathclyde Academic Media, 2014).

[Cro19] L. Crockett; D. Northcote; C. Ramsay; F. Robinson; R. Stewart. *Exploring Zynq MPSoC: With PYNQ and Machine Learning Applications* (Strathclyde Academic Media, 2019).

[Cru16] F. Cruz-Roldán; F.A. Pinto-Benel; J.D.O. del Campo; M. Blanco-Velasco. "A Wavelet OFDM Receiver for Baseband Power Line Communications". *Journal of the Franklin Institute*. 353 (7), pp. 1654 – 1671, 2016.
doi:https://doi.org/10.1016/j.jfranklin.2016.02.015

[Dib18] L. Dib; V. Fernandes; M. de L. Filomeno; M.V. Ribeiro. "Hybrid PLC/Wireless Communication for Smart Grids and Internet of Things Applications". *IEEE Internet of Things Journal*. 5 (2), pp. 655–667, apr 2018.
doi:10.1109/jiot.2017.2764747

[ElA14] M. ElAbsi; M. Shaat; F. Bader; T. Kaiser. "Power Loading and Spectral Efficiency Comparison of MIMO OFDM/FBMC for Interference Alignment Based Cognitive Radio Systems". In *2014 11th International Symposium on Wireless Communications Systems (ISWCS)* (08 2014) pp. 1–6.
doi:10.1109/ISWCS.2014.6933401

[ElW09] K. ElWazeer; M.M. Khairy; H.A.H. Fahmy; S.E.D. Habib. "FPGA implementation of an improved channel estimation algorithm for mobile WiMAX". In *2009 International Conference on Microelectronics - ICM* (dec 2009) pp. 1–4.
doi:10.1109/icm.2009.5418629

[Far03] B. Farhang-Boroujeny. "Multicarrier modulation with blind detection capability using cosine modulated filter banks". *IEEE Transactions on Communica-*

tions. 51 (12), pp. 2057–2070, dec 2003.
doi:10.1109/tcomm.2003.820753

[Far11] B. Farhang-Boroujeny. "OFDM Versus Filter Bank Multicarrier". *IEEE Signal Processing Magazine*. 28 (3), pp. 92–112, may 2011.
doi:10.1109/msp.2011.940267

[Far14] B. Farhang-Boroujeny. "Filter bank multicarrier modulation: A waveform candidate for 5G and beyond". *Advances in Electrical Engineering*. 2014, pp. 1–26, 2014.
doi:10.1155/2014/482805

[Fen12] X. Feng; H.C. Wu; Y. Wu; X. Wang. "Kasami sequence studies for DTV transmitter identification". *IEEE Transactions on Consumer Electronics*. 58 (4), pp. 1138–1146, nov 2012.
doi:10.1109/tce.2012.6414978

[Gal11] S. Galli; A. Scaglione; Z. Wang. "For the Grid and Through the Grid: The Role of Power Line Communications in the Smart Grid". *Proceedings of the IEEE*. 99 (6), pp. 998–1027, June 2011.
doi:10.1109/JPROC.2011.2109670

[Gal15] S. Galli; T. Lys. "Next generation Narrowband (under 500 kHz) Power Line Communications (PLC) standards". *China Communications*. 12 (3), pp. 1–8, 2015.
doi:10.1109/CC.2015.7084358

[Gal16] S. Galli; K.J. Kerpez; H. Mariotte; F. Moulin. "PLC-to-DSL Interference: Statistical Model and Impact on VDSL2, Vectoring, and G.Fast". *IEEE Journal on Selected Areas in Communications*. 34 (7), pp. 1992–2005, jul 2016.
doi:10.1109/jsac.2016.2566118

[Gol61] M. Golay. "Complementary series". *IEEE Transactions on Information Theory*. 7 (2), pp. 82–87, apr 1961.
doi:10.1109/tit.1961.1057620

[Gol68] R. Gold. "Maximal recursive sequences with 3-valued recursive cross-correlation functions (Corresp.)". *IEEE Transactions on Information Theory*. 14 (1), pp. 154–156, jan 1968.
doi:10.1109/tit.1968.1054106

[Gos88] C. Goswami; M. Beale. "Correlation properties of dual-BCH, Kasami and other sequences for spread-spectrum, multiple-access systems". *IEE Proceedings F Communications, Radar and Signal Processing*. 135 (1), p. 114, 1988.
doi:10.1049/ip-f-1.1988.0016

[Gue16] L. Guerrieri; G. Masera; I.S. Stievano; P. Bisaglia; W.R. Garcia Valverde; M. Concolato. "Automotive Power-Line Communication Channels: Mathematical Characterization and Hardware Emulator". *IEEE Transactions on Industrial Electronics*. 63 (5), pp. 3081–3090, 2016.
doi:10.1109/TIE.2016.2516508

[Gun11] V.C. Gungor; D. Sahin; T. Kocak; S. Ergut; C. Buccella; C. Cecati; G.P. Hancke. "Smart Grid Technologies: Communication Technologies and Standards". *IEEE Transactions on Industrial Informatics*. 7 (4), pp. 529–539, 2011.
doi:10.1109/TII.2011.2166794

[Hir80] B. Hirosaki. "An Analysis of Automatic Equalizers for Orthogonally Multiplexed QAM Systems". *IEEE Transactions on Communications*. 28 (1), pp. 73–83, jan 1980.
doi:10.1109/tcom.1980.1094576

[Hol09] N. Holte. "MMSE equalization of OFDM/OQAM systems for channels with time and frequency dispersion". In *2009 International Conference on Wireless Communications & Signal Processing* (nov 2009) pp. 1–5.
doi:10.1109/wcsp.2009.5371391

[IEE95] IEEE. "IEEE Recommended Practice for Monitoring Electric Power Quality". *IEEE Std 1159-1995*. pp. 1–80, Nov 1995.
doi:10.1109/IEEESTD.1995.79050

[IEE07] IEEE. "IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Corrigendum 2: IEEE Std 802.An-2006 10GBASE-T Correction", Aug. 2007.
doi:10.1109/ieeestd.2007.4293165

[IEE10] IEEE. "1901-2010-IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications". *IEEE: New York, NY, USA*. pp. 1–1586, 2010.
doi:10.1109/IEEESTD.2010.5678772

[IEE11] IEEE. "IEEE Standard for Power Line Communication Equipment–Electromagnetic Compatibility (EMC) Requirements–Testing and Measurement Methods", Jan. 2011.
doi:10.1109/ieeestd.2011.5682873

[IEE12] IEEE. "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC)

and Physical Layer (PHY) Specifications", Mar. 2012.
doi:10.1109/ieeestd.2012.6178212

[IEE13a] IEEE. "IEEE Standard for a Convergent Digital Home Network for Heterogeneous Technologies", Apr. 2013.
doi:10.1109/ieeestd.2013.6502164

[IEE13b] IEEE. "IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications". *IEEE Std 1901.2-2013*. pp. 1–269, Dec 2013.
doi:10.1109/IEEESTD.2013.6679210

[IEE18] IEEE. "IEEE Standard for Medium Frequency (less than 12 MHz) Power Line Communications for Smart Grid Applications". *IEEE Std 1901.1-2018*. pp. 1–192, May 2018.
doi:10.1109/IEEESTD.2018.8360785

[Iha05] T. Ihalainen; T.H. Stitz; M. Renfors. "Efficient Per-Carrier Channel Equalizer for Filter Bank based Multicarrier Systems". In *2005 IEEE International Symposium on Circuits and Systems* (May 2005) pp. 3175–3178, vol. 4.
doi:10.1109/ISCAS.2005.1465302

[Iha06] T. Ihalainen; T. Hidalgo Stitz; M. Rinne; M. Renfors. "Channel Equalization in Filter Bank Based Multicarrier Modulation for Wireless Communications". *EURASIP Journal on Advances in Signal Processing*. 2007 (1), p. 049389, Dec 2006.
doi:10.1155/2007/49389

[Ikh09] A. Ikhlef; J. Louveaux. "An enhanced MMSE per subchannel equalizer for highly frequency selective channels for FBMC/OQAM systems". In *2009 IEEE 10th Workshop on Signal Processing Advances in Wireless Communications* (jun 2009) pp. 1–5.
doi:10.1109/spawc.2009.5161772

[ISO15] ISO. "ISO 151183:2015 Road vehicles Vehicle to grid communication interface Part 3: Physical and data link layer requirements". Technical Specification, International Organization for Standardization, May 2015.
https://www.iso.org/standard/59675.html

[ITU12a] ITU-T. "G.9902 : Narrowband orthogonal frequency division multiplexing power line communication transceivers for ITU-T G.hnem networks". https://www.itu.int/rec/T-REC-G.9902/en, Oct. 2012.

[ITU12b] ITU-T. "G.9904 : Narrowband orthogonal frequency division multiplexing power line communication transceivers for PRIME networks". https://www.itu.int/rec/T-REC-G.9904/en, Oct. 2012.

[ITU17a] ITU-T. "G.9901 : Narrowband orthogonal frequency division multiplexing power line communication transceivers - Power spectral density specification". https://www.itu.int/rec/T-REC-G.9901/en, Jun. 2017.

[ITU17b] ITU-T. "G.9903 : Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks". https://www.itu.int/rec/T-REC-G.9903/en, Aug. 2017.

[ITU18a] ITU-T. "G.9960 : Unified high-speed wireline-based home networking transceivers - System architecture and physical layer specification". https://www.itu.int/rec/T-REC-G.9960/en, Nov. 2018.

[ITU18b] ITU-T. "G.9961 : Unified high-speed wireline-based home networking transceivers - Data link layer specification". https://www.itu.int/rec/T-REC-G.9961/en, Nov. 2018.

[ITU18c] ITU-T. "G.9963 : Unified high-speed wireline-based home networking transceivers - Multiple input/multiple output specification". https://www.itu.int/rec/T-REC-G.9963-201811-I/en, Nov. 2018.

[Izu07] K. Izumi; D. Umehara; S. Denno. "Performance Evaluation of Wavelet OFDM Using ASCET". In *2007 IEEE International Symposium on Power Line Communications and Its Applications* (March 2007) pp. 246–251.
doi:10.1109/ISPLC.2007.371131

[Jac05] P.A. Jackson; C.P. Chan; J.E. Scalera; C.M. Rader; M.M. Vai. "A systolic FFT architecture for real time FPGA systems". Tech. rep., MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 2005.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.6651&rep=rep1&type=pdf

[Ker15] K.J. Kerpez; S. Galli; H. Mariotte; F. Moulin. "The impact of PLC-to-DSL interference on VDSL2, Vectored VDSL2, and G.fast". In *2015 IEEE International Symposium on Power Line Communications and Its Applications (ISPLC)* (mar 2015) pp. 1–6.
doi:10.1109/isplc.2015.7147607

[Kha16] F. Khan; A. u. Rehman; M. Arif; M. Aftab; B.K. Jadoon. "A survey of communication technologies for smart grid connectivity". In *2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)* (April 2016) pp. 256–261.
doi:10.1109/ICECUBE.2016.7495234

[Lag08] G. Laguna; R. Barrón. "Survey on Indoor Power Line Communication Channel Modeling". In *2008 Electronics, Robotics and Automotive Mechanics Conferen-*

ce (CERMA '08) (Sep. 2008) pp. 163–168.
doi:10.1109/CERMA.2008.23

[Lai19] K. Lai; Y. Huang; C. Chen; C. Lin. "A Family of MMSE-Based Decision Feedback Equalizers and Their Properties for FBMC/OQAM Systems". *IEEE Transactions on Vehicular Technology.* 68 (3), pp. 2346–2360, March 2019.
doi:10.1109/TVT.2019.2891744

[Lak18] S. Laksir; A. Chaoub; A. Tamtaoui. "Impulsive Noise Reduction Techniques in Power Line Communication: A Survey and Recent Trends". In *2018 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)* (June 2018) pp. 13–20.
doi:10.1109/MoWNet.2018.8428879

[Lar14] T. Larhzaoui; F. Nouvel; J. Baudais; P. Degauque; V. Dégardin. "OFDM PLC transmission for aircraft flight control system". In *18th IEEE International Symposium on Power Line Communications and Its Applications* (March 2014) pp. 220–225.
doi:10.1109/ISPLC.2014.6812333

[Lee11] J.J. Lee; Y.H. Kim; J.M. Bae; J.K. Seo; D.H. Nam; J.Y. Kim; D.S. In. "AMR field trial on underground power distribution line using BPLC". In *2011 IEEE International Symposium on Power Line Communications and Its Applications* (apr 2011) pp. 1–5.
doi:10.1109/isplc.2011.5764393

[Lel07] C. Lele; P. Siohan; R. Legouable; J.P. Javaudin. "Preamble-based channel estimation techniques for OFDM/OQAM over the powerline". In *2007 IEEE International Symposium on Power Line Communications and Its Applications* (2007) pp. 1–6.
doi:10.1109/isplc.2007.371098

[Liu13] C. Liu; K.T. Chau; D. Wu; S. Gao. "Opportunities and Challenges of Vehicle-to-Home, Vehicle-to-Vehicle, and Vehicle-to-Grid Technologies". *Proceedings of the IEEE.* 101 (11), pp. 2409–2427, Nov 2013.
doi:10.1109/JPROC.2013.2271951

[Lou08] J. Louveaux; L. Baltar; D. Waldhauser; M. Renfors; M. Tanda; C. Bader; E. Kofidis. "Equalization and demodulation in the receiver (single antenna)". *ICT-211887 PHYDYAS deliverable D.* 3, 2008.
http://www.ict-phydyas.org/delivrables/PHYDYAS-D3.1.pdf/at_download/file

[Ma10] Y. tao Ma; K. hua Liu; Z. jun Zhang; J. xiao Yu; X. lin Gong. "Modeling the colored background noise of power line communication channel based on

artificial neural network". In *The 19th Annual Wireless and Optical Communications Conference (WOCC 2010)* (may 2010) pp. 1–4.
doi:10.1109/wocc.2010.5510658

[Maf15] J. Mafra; M. Hosami; L. Freitas; M. Martinelli; A. Almeida. "Hybrid communication module - motivations, requirements, challenges and implementations". In *2015 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT LATAM)* (oct 2015) pp. 1–5.
doi:10.1109/isgt-la.2015.7381124

[Mak15] M. Makni; M. Baklouti; S. Niar; M. Biglari-Abhari; M. Abid. "Heterogeneous Multi-Core Architecture for a 4G Communication in High-Speed Railway". In *2015 10th International Design & Test Symposium (IDT)* (2015) pp. 26–31.
doi:10.1109/IDT.2015.7396731

[Man04] F. Manavi; Y. Shayan. "Implementation of OFDM modem for the physical layer of IEEE 802.11a standard based on Xilinx Virtex-II FPGA". In *2004 IEEE 59th Vehicular Technology Conference. VTC 2004-Spring (IEEE Cat. No.04CH37514)* (2004) pp. 1768–1772.
doi:10.1109/vetecs.2004.1390560

[Mar15] L. Martín Lagares. *Algoritmos para el Filtrado Eficiente de Señales Reales Basados en la DFT.* Master's thesis, Universidad de Sevilla, 2015.
http://bibing.us.es/proyectos/abreproy/12300/

[Mel06] R.S. Mello; M.A. Grivet. "Modeling, simulation and estimation of PLC channels". In *2006 International Telecommunications Symposium* (Sep. 2006) pp. 201–206.
doi:10.1109/ITS.2006.4433270

[Mem17] F. Membibre Fernández. *Diseño de una arquitectura flexible para la implementación de la FFT en dispositivos FPGAs.* Master's thesis, Universidad de Alcalá, 2017.
http://hdl.handle.net/10017/38568

[Mem19a] F. Membibre; R. Nieto; A. Hernández. "Arquitectura para comunicaciones PLC de Banda Ancha basada en un Analog Front-End". In *2019 XXVI Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI)* (2019) pp. 1–6.

[Mem19b] F. Membibre; R. Nieto; A. Hernandez. "Design of FPGA-based Architecture for an Analog Front-End in Broadband PLC". In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (sep 2019) pp. 1–4.
doi:10.1109/etfa.2019.8869166

[Men17] A. Mengi; S. Ponzelar; M. Koch. "The ITU-T G.9960 broadband PLC communication concept for smartgrid applications". In *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)* (2017) pp. 492–496.
doi:10.1109/SmartGridComm.2017.8340715

[Mez14] E.M. Meza; J.B. Aspiazu. "Design of a Power Line Communications Transceiver Based on OFDM". *Procedia Technology*. 17, pp. 107 – 113, 2014. Conference on Electronics, Telecommunications and Computers CETC 2013.
doi:https://doi.org/10.1016/j.protcy.2014.10.200

[Mly16] P. Mlynek; R. Fujdiak; J. Misurec; J. Slacik. "Experimental measurements of noise influence on narrowband Power Line Communication". In *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)* (oct 2016) pp. 1–7.
doi:10.1109/icumt.2016.7765339

[Moy16a] S. Moya; M. Hadad; P. Donato; M. Funes; D. Carrica. "Channel estimation and equalization of broadband PLC systems - Part 2: Comparison between Single Carrier and OFDM approaches". In *2016 IEEE Biennial Congress of Argentina (ARGENCON)* (June 2016) pp. 1–6.
doi:10.1109/ARGENCON.2016.7585284

[Moy16b] S. Moya; M. Hadad; M. Funes; P. Donato; D. Carrica. "Broadband PLC-channel equalisation in the frequency domain based on complementary sequences". *IET Communications*. 10 (13), pp. 1605–1613, 2016.
doi:10.1049/iet-com.2015.0924

[Nad14] J. Nadal; C.A. Nour; A. Baghdadi; H. Lin. "Hardware prototyping of FBMC/OQAM baseband for 5G mobile communication". In *Proceedings of the 2014 25th IEEE International Symposium on Rapid System Prototyping, New Delhi, India*, vol. 1617 (2014) p. 7277.
https://metis2020.com/wp-content/uploads/publications/IEEE_2014_Nadal_etal_Hardware-Prototyping-of-FBMC_OQAN-Baseband.pdf

[Ng98] J. Ng; K. Letaief; R. Murch. "Complex optimal sequences with constant magnitude for fast channel estimation initialization". *IEEE Transactions on Communications*. 46 (3), pp. 305–308, mar 1998.
doi:10.1109/26.662635

[Nie18a] R. Nieto; A. Hernández; R. Mateo. "Modelado HLS de un Transmultiplexor Multiportadora para PLC: Comparativa de Prestaciones". In *2018 XXV Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI)* (2018) pp. 1–6.

[Nie18b] R. Nieto; R. Mateos; A. Hernández. "Finite Precision Analysis of FPGA-based Architecture for FBMC Transmultiplexers in Broadband PLC". In *2018 33rd Conference on Design of Circuits and Integrated Systems (DCIS)* (2018) pp. 1–6.
doi:10.1109/DCIS.2018.8681466

[Nie19a] R. Nieto; A. Hernández; R. Mateo. "Aplicación de Unidad Vectorial NEON para la Implementación de un Transmultiplexor FBMC con Estimador e Igualador de Canal". In *2019 XXVI Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI)* (2019) pp. 1–6.

[Nie19b] R. Nieto; R. Mateos; A. Hernández. "HW/SW Architecture for a Broadband Power-Line Communications System with LS Channel Estimator and ASCET Equalizer". *IEEE Transactions Industrial Informatics.* pp. 1–12, 2019. (in third review round).

[Nie19c] R. Nieto; R. Mateos; A. Hernández; E. Díaz. "Dual-Core Architecture for PLC Channel Estimator and ASCET Equalizer in a FBMC Transmultiplexer". In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (Sep. 2019) pp. 1342–1345.
doi:10.1109/ETFA.2019.8869476

[Nie20] R. Nieto; R. Mateos; A. Hernández. "Performance Evolution of PLC Channel Estimator and ASCET Equalizer in a FBMC Transmultiplexer Using MPSoC Zynq-UltraScale+". *IEEE Transactions Transactions on Circuits and Systems I.* pp. 1–8, 2020.

[Nom15] F. Nombela; E. García; J.U. na; A. Hernández; P. Poudereux. "Robust synchronization algorithm for broadband PLC based on Wavelet-OFDM". In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)* (2015) pp. 1–7.
doi:10.1109/ETFA.2015.7301480

[Nom17] F. Nombela; E. García; A. Hernández. "Preliminary study between frequency-domain equalizer and 0-ASCET for PLC". In *2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband (ICUWB)* (2017) pp. 1–5.
doi:10.1109/ICUWB.2017.8250958

[Obe14] E. Oberleithner; N. Hanik. "Impact of System Components on an Automotive PLC Channel". In *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)* (sep 2014) pp. 1–5.
doi:10.1109/vtcfall.2014.6966005

[Oks06] P. Oksa; M. Soini; L. Sydanheimo; M. Kivikoski. "Considerations of Using Power Line Communication in the AMR System". In *2006 IEEE International*

Symposium on Power Line Communications and Its Applications (2006) pp. 1–4.
doi:10.1109/isplc.2006.247462

[Oli18] T.R. Oliveira; A.A. Picorone; C.B. Zeller; S.L. Netto; M.V. Ribeiro. "On the statistical characterization of hybrid PLC-wireless channels". *Electric Power Systems Research*. 163, pp. 329 – 337, 2018.
doi:https://doi.org/10.1016/j.epsr.2018.07.004

[Pan18] L. Pan; J. Wang; J. Wang. "Research on control system of intelligent shore-to-ship power connecting pile based on PLC technology". In *2018 Chinese Control And Decision Conference (CCDC)* (June 2018) pp. 1120–1125.
doi:10.1109/CCDC.2018.8407297

[Par11] J.S. Park; T. Ogunfunmi. "FPGA implementation of channel estimation for MIMO-OFDM". In *2011 IEEE International Symposium of Circuits and Systems (ISCAS)* (may 2011) pp. 1–4.
doi:10.1109/iscas.2011.5937663

[Pen14] S.C. Peng; L.S. Dai; T.H. Su. "The design of smart electrical outlet for Smart Home base on power line communication". In *2014 International Conference on Intelligent Green Building and Smart Grid (IGBSG)* (apr 2014) pp. 1–4.
doi:10.1109/igbsg.2014.6835230

[Phi99] H. Phillipps. "Modelling of powerline communication channels". In *Proc. Int. Symp. Power Line Commun. Appl* (1999) pp. 14–21.
https://pdfs.semanticscholar.org/297d/9a14da2813e7cf252dead5a6ab5eceb0ee55.pdf

[Pic14] A.A.M. Picorone; R.S. Neto; M.V. Ribeiro. "Coherence time and sparsity of Brazilian outdoor PLC channels: A preliminary analysis". In *18th IEEE International Symposium on Power Line Communications and Its Applications* (March 2014) pp. 1–5.
doi:10.1109/ISPLC.2014.6812337

[Pin14] F.A. Pinto-Benel; F.J. Reynoso; F.C. Roldán. "Aplicación de un Sistema con Banco de Filtros Multiportadora de Banda Ancha con Igualador ASCET en Presencia de un Canal PLC". *COmtEL 2014 VI Congreso Internacional de Computación y telecomunicación*. pp. 1–5, 2014.
http://repositorio.uigv.edu.pe/handle/20.500.11818/734

[Pin15] F.A. Pinto-Benel; F. Cruz-Roldán. "2-ASCET for Broadband Multicarrier Transmission over In-home and In-vehicle Power Line Networks". In *2015 IEEE 18th International Conference on Intelligent Transportation Systems* (2015) pp.

1351–1356.
doi:10.1109/ITSC.2015.222

[Pit16] A. Pittolo; M.D. Piante; F. Versolatto; A.M. Tonello. "In-Vehicle Power Line Communication: Differences and Similarities Among the In-Car and the In-Ship Scenarios". *IEEE Vehicular Technology Magazine*. 11 (2), pp. 43–51, jun 2016.
doi:10.1109/mvt.2015.2480098

[Pou16] P. Poudereux; A. HernÃ¡ndez; R. Mateos; F.A. Pinto-Benel; F. Cruz-Roldán. "Design of a filter bank multi-carrier system for broadband power line communications". *Signal Processing*. 128, pp. 57 – 67, 2016.
doi:https://doi.org/10.1016/j.sigpro.2016.03.014

[Pou18] P. Poudereux; A. Hernández; F. Cruz-Roldán; R. Mateos. "FPGA-Based Architecture for Medium Access Techniques in Broadband PLC". *IEEE Access*. 6, pp. 9534–9542, 2018.
doi:10.1109/ACCESS.2018.2808371

[Pro19] O.S. Project. "Project Ne10 An Open Optimized Software Library Project for the ARM Architecture", 2019.
https://projectne10.github.io/Ne10/

[Puk17] R.S. Pukale; S.V. Bavache; S.N. Kashid; A.P. Sabale; P.P. Patil; P.H. Shinde. "Automatic meter reading of electricity by using power line communication". In *2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT)* (mar 2017) pp. 1–5.
doi:10.1109/igeht.2017.8094100

[Raf12] M. Rafiei; S.M. Eftekhari. *A practical smart metering using combination of power line communication (PLC) and WiFi protocols* (2012).
https://ieeexplore.ieee.org/document/6254548

[Say14] M. Sayed; N. Al-Dhahir. "Narrowband-PLC/wireless diversity for smart grid communications". In *2014 IEEE Global Communications Conference* (dec 2014) pp. 1–6.
doi:10.1109/glocom.2014.7037259

[Sha08] X. Shao; S.G. Johnson. "Type-IV DCT, DST, and MDCT Algorithms with Reduced Numbers of Arithmetic Operations". *Signal Processing*. 88 (6), pp. 1313 – 1326, 2008.
doi:https://doi.org/10.1016/j.sigpro.2007.11.024

[Sti14] I.S. Stievano; F.G. Canavero; W.R. Garcia Valverde; L. Guerrieri; P. Bisaglia. "Multipath Modeling Of Automotive Power Line Communication Channels".

IEEE Transactions on Industrial Informatics. 10 (2), pp. 1381–1391, 2014. doi:10.1109/TII.2014.2307199

[Ten14] B.D. Tensubam; N.L. Chanu; S. Singh. "Comparative analysis of FBMC and OFDM multicarrier techniques for wireless communication networks". *International Journal of Computer Applications*. 100 (19), pp. 1–5, 2014. https://pdfs.semanticscholar.org/dc13/bd17f3223357c1378c9f6ab343fcccdc768c.pdf

[Ton12] A.M. Tonello; F. Versolatto; B. Bejar; S. Zazo. "A Fitting Algorithm for Random Modeling the PLC Channel". *IEEE Transactions on Power Delivery*. 27 (3), pp. 1477–1484, 2012. doi:10.1109/TPWRD.2012.2196714

[Ton13a] A.M. Tonello; M. Girotto. "Cyclic block FMT modulation for broadband power line communications". In *2013 IEEE 17th International Symposium on Power Line Communications and Its Applications* (March 2013) pp. 247–251. doi:10.1109/ISPLC.2013.6525858

[Ton13b] A.M. Tonello; F. Versolatto; M. Girotto. "Multitechnology (I-UWB and OFDM) Coexistent Communications on the Power Delivery Network". *IEEE Transactions on Power Delivery*. 28 (4), pp. 2039–2047, Oct 2013. doi:10.1109/TPWRD.2013.2264881

[Ton14] A.M. Tonello; A. Pittolo; M. Girotto. "Power line communications: Understanding the channel for physical layer evolution based on filter bank modulation". *IEICE Transactions on Communications*. 97 (8), pp. 1494–1503, 2014. http://web.diegm.uniud.it/tonello/PAPERS/JOURNALS/IEICE2014.pdf

[Ton15] A.M. Tonello; A. Pittolo. "Considerations on narrowband and broadband power line communication for smart grids". In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)* (Nov 2015) pp. 13–18. doi:10.1109/SmartGridComm.2015.7436269

[Tun13] M.A. Tunç; E. Perrins. "Pilot based channel estimation and transform domain analysis in broadband PLC for Smart Grid". In *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)* (Oct 2013) pp. 283–288. doi:10.1109/SmartGridComm.2013.6687971

[Tur63] R. Turyn. "Ambiguity functions of complementary sequences (Corresp.)". *IEEE Transactions on Information Theory*. 9 (1), pp. 46–47, jan 1963. doi:10.1109/tit.1963.1057807

[Tur12] H. Turker; S. Bacha; D. Chatroux; A. Hably. "Modelling of system components for Vehicle-to-Grid (V2G) and Vehicle-to-Home (V2H) applications with Plug-in Hybrid Electric Vehicles (PHEVs)". In *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)* (jan 2012) pp. 1–8.
doi:10.1109/isgt.2012.6175652

[Vel17] J.M. Velazquez-Gutierrez; C. Vargas-Rosales. "Sequence Sets in Wireless Communication Systems: A Survey". *IEEE Communications Surveys & Tutorials*. 19 (2), pp. 1225–1248, 2017.
doi:10.1109/comst.2016.2639739

[Viv16] Vivado. *Vivado Design Suite User Guide (UG902) - High-Level Synthesis*. Xilinx, 2016.
https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_3/ug902-vivado-high-level-synthesis.pdf

[Wal08] D.S. Waldhauser; L.G. Baltar; J.A. Nossek. "MMSE subcarrier equalization for filter bank based multicarrier systems". In *2008 IEEE 9th Workshop on Signal Processing Advances in Wireless Communications* (jul 2008) pp. 1–5.
doi:10.1109/spawc.2008.4641663

[Wal09] D.S. Waldhauser; L.G. Baltar; J.A. Nossek. "Adaptive decision feedback equalization for filter bank based multicarrier systems". In *2009 IEEE International Symposium on Circuits and Systems* (may 2009) pp. 1–4.
doi:10.1109/iscas.2009.5118382

[Xil18a] Xilinx. "7 Series DSP48E1 Slice (UG479)". Tech. rep., Xilinx, 2018.
https://www.xilinx.com/support/documentation/user_guides/ug479_7Series_DSP48E1.pdf

[Xil18b] Xilinx. "Mailbox v2.1 LogiCORE IP Product Guide (PG114)". Tech. rep., Xilinx, 2018.
https://www.xilinx.com/support/documentation/ip_documentation/mailbox/v2_1/pg114-mailbox.pdf

[Xil18c] Xilinx. "Zynq-7000 SoC Data Sheet: Overview (DS190)". Tech. rep., Xilinx, 2018.
https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf

[Xil18d] Xilinx. "Zynq-7000 Technical Reference Manual (ug585)". Tech. rep., Xilinx, 2018.
https://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf

[Xil18e] Xilinx. "Zynq UltraScale+ MPSoC Data Sheet: Overview (DS891)". Tech. rep., Xilinx, 2018.
https://www.xilinx.com/support/documentation/data_sheets/ds891-zynq-ultrascale-plus-overview.pdf

[Xil19a] Xilinx. "7 Series FPGAs Memory Resources (UG473)". Tech. rep., Xilinx, 2019.
https://www.xilinx.com/support/documentation/user_guides/ug473_7Series_Memory_Resources.pdf

[Xil19b] Xilinx. "AXI DMA v7.1 LogiCORE IP Product Guide (PG021)". Tech. rep., Xilinx, 2019.
https://www.xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf

[Xil19c] Xilinx. "Zynq UltraScale+ Device Technical Reference Manual (UG1085)". Tech. rep., Xilinx, 2019.
https://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf

[Xu14] L.D. Xu; W. He; S. Li. "Internet of Things in Industries: A Survey". *IEEE Transactions on Industrial Informatics*. 10 (4), pp. 2233–2243, Nov 2014.
doi:10.1109/TII.2014.2300753

[Yan07] Y. Yang; T. Ihalainen; M. Rinne; M. Renfors. "Frequency-Domain Equalization in Single-Carrier Transmission: Filter Bank Approach". *EURASIP Journal on Advances in Signal Processing*. 2007 (1), p. 010438, Dec 2007.
doi:10.1155/2007/10438

[Yin19] Z. Yin; S. Dou; H. Bai; Y. Hou. "Light-Weighted Security Access Scheme of Broadband Power Line Communications for Multi-Source Information Collection". In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (mar 2019) pp. 1–4.
doi:10.1109/itnec.2019.8729357

[Zei17] A. Zeichner; Z. Chen; S. Frei. "Theoretical maximum data rate estimations for PLC in automotive power distribution systems". In *2017 International Symposium on Electromagnetic Compatibility - EMCEUROPE* (sep 2017) .
doi:10.1109/emceurope.2017.8094760

[Zen06] X. Zeng; Q. Liu; L. Hu. "A New Family of Codes and Generalized Kasami Sequences". In *2006 IEEE International Symposium on Information Theory* (IEEE, 2006) .
doi:10.1109/isit.2006.261791

[Zha17] Y. Zhang; S. Lin; L. Chen; W. Chang; W. Hu; J. Tang; C. Yu. "An implementation of an in-vehicle power line communication system". In *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)* (2017) pp. 1–2. doi:10.1109/GCCE.2017.8229422

[Zhi17] A.A. Zhilenkov; D.D. Gilyazov; I.I. Matveev; Y.V. Krishtal. "Power line communication in IoT-systems". In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)* (IEEE, 2017) . doi:10.1109/eiconrus.2017.7910538

[Zim99] M. Zimmermann; K. Dostert. "A multi-path signal propagation model for the power line channel in the high frequency range". In *Proc. IEEE Int. Symp. Power Line Commun. Appl* (1999) pp. 45–51. https://pdfs.semanticscholar.org/c599/717e063435af03c6465785c011a8ca061d9b.pdf

[Zim02a] M. Zimmermann; K. Dostert. "A multipath model for the powerline channel". *IEEE Transactions on Communications.* 50 (4), pp. 553–559, apr 2002. doi:10.1109/26.996069

[Zim02b] M. Zimmermann; K. Dostert. "Analysis and modeling of impulsive noise in broad-band powerline communications". *IEEE Transactions on Electromagnetic Compatibility.* 44 (1), pp. 249–258, 2002. doi:10.1109/15.990732

[Zol09] M.D. Zoltowski; T.R. Qureshi; R. Calderbank. "Channel estimation for MIMO-OFDM using complementary codes". In *2009 IEEE Radio and Wireless Symposium* (IEEE, 2009) . doi:10.1109/rws.2009.4957309