



Universidad
de Alcalá

COMISIÓN DE ESTUDIOS OFICIALES
DE POSGRADO Y DOCTORADO

ACTA DE EVALUACIÓN DE LA TESIS DOCTORAL
(FOR EVALUATION OF THE ACT DOCTORAL THESIS)

Año académico (academic year): 2016/17

DOCTORANDO (candidate PHD): **POUDEREUX CLEMENTE, PABLO**

D.N.I./PASAPORTE (Id.Passport): ******7840L**

PROGRAMA DE DOCTORADO (Academic Committee of the Programme): **D332 DOCTORADO EN ELECTRÓNICA:SISTEMAS ELECTRÓNICOS AVANZADOS.SISTEMAS INTELIGENTES**

DEPARTAMENTO DE (Department): **ELECTRÓNICA**

TITULACIÓN DE DOCTOR EN (Phd title): **DOCTOR/A POR LA UNIVERSIDAD DE ALCALÁ**


En el día de hoy 15/02/17, reunido el tribunal de evaluación, constituido por los miembros que suscriben el presente Acta, el aspirante defendió su Tesis Doctoral **con Mención Internacional** (In today assessment met the court, consisting of the members who signed this Act, the candidate defended his doctoral thesis with mention as International Doctorate), elaborada bajo la dirección de (prepared under the direction of) ALVARO HERNÁNDEZ ALONSO // RAUL MATEOS GIL.

Sobre el siguiente tema (Title of the doctoral thesis): **DISEÑO DE ARQUITECTURAS EFICIENTES BASADAS EN DISPOSITIVOS LÓGICOS PROGRAMABLES PARA TÉCNICAS DE ACCESO AL MEDIO EN COMUNICACIONES PLC**

Finalizada la defensa y discusión de la tesis, el tribunal acordó otorgar la CALIFICACIÓN GLOBAL² de (**no apto, aprobado, notable y sobresaliente**) (After the defense and defense of the thesis, the court agreed to grant the GLOBAL RATING (fail, pass, good and excellent): **SOBRESALIENTE**

Alcalá de Henares, a 15 de FEBRERO de 2017

Fdo. (Signed):  **JAVIER UTRERA**

Fdo. (Signed):  **GUSTAVO SUTTER**

Fdo. (Signed):  **PATRICIO DONATO**

FIRMA DEL ALUMNO (candidate's signature),



Con fecha 22 de febrero de 2017 la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado, a la vista de los votos emitidos de manera anónima por el tribunal que ha juzgado la tesis, resuelve:

- Conceder la Mención de "Cum Laude"
 No conceder la Mención de "Cum Laude"

Fdo. (Signed): PABLO POUDEREUX CLEMENTE La Secretaria de la Comisión Delegada



² La calificación podrá ser "no apto" "aprobado" "notable" y "sobresaliente". El tribunal podrá otorgar la mención de "cum laude" si la calificación global es de sobresaliente y se emite en tal sentido el voto secreto positivo por unanimidad. (The grade may be "fail" "pass" "good" or "excellent". The panel may confer the distinction of "cum laude" if the overall grade is "Excellent" and has been awarded unanimously as such after secret voting.).

INCIDENCIAS / OBSERVACIONES:
(Incidents / Comments)

1. The first observation is that the data is not consistent with the expected results. This is due to the fact that the system is not properly configured. The second observation is that the system is not properly configured. The third observation is that the system is not properly configured. The fourth observation is that the system is not properly configured. The fifth observation is that the system is not properly configured. The sixth observation is that the system is not properly configured. The seventh observation is that the system is not properly configured. The eighth observation is that the system is not properly configured. The ninth observation is that the system is not properly configured. The tenth observation is that the system is not properly configured.

En aplicación del art. 14.7 del RD. 99/2011 y el art. 14 del Reglamento de Elaboración, Autorización y Defensa de la Tesis Doctoral, la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado y Doctorado, en sesión pública de fecha 22 de febrero, procedió al escrutinio de los votos emitidos por los miembros del tribunal de la tesis defendida por *POUDEREUX CLEMENTE, PABLO*, el día 15 de febrero de 2017, titulada *DISEÑO DE ARQUITECTURAS EFICIENTES BASADAS EN DISPOSITIVOS LÓGICOS PROGRAMABLES PARA TÉCNICAS DE ACCESO AL MEDIO EN COMUNICACIONES PLC*, para determinar si a la misma se le concede la mención “cum laude”, arrojando como resultado, 2 votos a favor y 1 en contra.

Por lo tanto, la Comisión de Estudios Oficiales de Posgrado **resuelve no otorgar la Mención de “cum laude”** a dicha Tesis.

Alcalá de Henares, 22 de febrero de 2017

EL PRESIDENTE DE LA COMISIÓN DE ESTUDIOS
OFICIALES DE POSGRADO Y DOCTORADO



Juan Ramón Velasco Pérez

Copia por e-mail a:

Doctorando: *POUDEREUX CLEMENTE, PABLO*

Secretario del Tribunal: *PATRICIO DONATO*.

Directores de Tesis: *ALVARO HERNÁNDEZ ALONSO//RAUL MATEOS GIL*

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

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



Tesis Doctoral

Diseño de arquitecturas eficientes basadas en dispositivos lógicos programables para técnicas de acceso al medio en comunicaciones PLC

Pablo Poudereux Clemente

2016

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Programa de Doctorado en Electrónica Sistemas Electrónicos Avanzados. Sistemas Inteligentes



Diseño de arquitecturas eficientes basadas en dispositivos lógicos programables para técnicas de acceso al medio en comunicaciones PLC

Autor:

Pablo Poudereux Clemente

Directores:

Álvaro Hernández Alonso

Raúl Mateos Gil

2016

Tesis Doctoral

Resumen

En el trabajo propuesto en esta tesis se han estudiado, analizado y desarrollado nuevas arquitecturas para la implementación de técnicas de acceso al medio en comunicaciones PLC de banda ancha. Estas arquitecturas se han incluido como periféricos avanzados en un sistema SoC general basado en FPGAs que integra un microprocesador soft encargado de realizar la supervisión del sistema y la gestión de las transferencias de datos. Además, la arquitectura SoC general dispone de un módulo DMA que asegura el flujo de datos necesario para cada técnica de acceso al medio, permitiendo que todo el sistema opere en tiempo real, atendiendo a los requisitos del estándar de PLC de banda ancha.

Para la definición de las distintas arquitecturas, se ha realizado un exhaustivo estudio de las técnicas de acceso al medio consideradas. Dentro de este estudio se ha efectuado un análisis de distintos algoritmos para su implementación, evaluándose las opciones más idóneas en cada caso. El análisis de distintas alternativas ha permitido obtener una arquitectura con un bajo consumo de recursos y que a su vez disponga de un tiempo de cómputo que permita la implementación en tiempo real. Esto se ha conseguido con el ratio de paralelismo, el cual ha permitido una reutilización a lo largo del tiempo de los recursos implicados, obteniéndose un compromiso entre tiempo de procesamiento y recursos empleados. Asimismo, dado que las arquitecturas se van a implantar sobre un dispositivo FPGA, es necesario considerar en todo este estudio el efecto de la precisión finita, ya que es determinante a la hora de obtener unas prestaciones adecuadas. Para ello, se ha tenido en especial consideración el dispositivo en el que se iba a realizar la implementación, utilizando el ancho de palabra máximo de las celdas aritméticas y de las memorias disponibles.

Para la comprobación de la calidad de las arquitecturas diseñadas se han desarrollado modelos de simulación en coma flotante y en coma fija. La utilización de modelos de simulación permite realizar análisis cuantitativo del efecto de la precisión finita y a su vez comprobar que la implementación desarrollada es correcta. Tanto los modelos de simulación, como las posteriores pruebas experimentales, han sido evaluados en distintos escenarios prácticos, permitiendo una verificación precisa de los datos obtenidos y su corroboración con los resultados simulados. Los escenarios considerados han tenido en cuenta distintos tipos de canales de transmisión, incrementando el nivel de ruido y atenuación gradualmente.

En el primer caso se ha empleado un canal ideal, que ha permitido realizar un estudio del efecto de la precisión finita sobre las arquitecturas propuestas. En el segundo caso se ha empleado un cable SMA como canal de transmisión. El cable SMA presenta una buena respuesta en frecuencia y por tanto se puede comparar la calidad de las arquitecturas en un sistema de transmisión completo, cuyo canal se aproxima al ideal. Por último, el tercer caso introduce un canal más agresivo formado por un cable eléctrico de doce metros. Con este canal se ha realizado una estimación del comportamiento de la técnica de acceso al medio en un canal similar al real sin el empleo de ningún módulo de estimación e igualación de canal.

Abstract

This thesis deals with the study, design and development of new architectures for medium access techniques in broadband PLC (PowerLine Communications). These architectures have been included as advanced peripheral in a general SoC (System-on-Chip) system based on FPGAs (Field-Programmable Gate Arrays). It integrates a soft microprocessor which supervises the system and manages the data transfer. In addition, the general SoC architecture has a DMA (Direct Access Memory) module that assures the necessary data flow rate for every medium access technique, allowing the global system to operate in real time while meeting all these requirements from the broadband PLC standard.

For the definition of the different architectures, an exhaustive study of the considered medium access techniques has been done. An analysis of different algorithms for the implementation has been carried out in this study, evaluating the most suitable option. The analysis of different approaches has allowed an architecture with a low resource consumption to be obtained, as well as a processing time that makes real-time implementation feasible. This has been achieved by means of the parallelism ratio, which has been enabled the reutilization of the implied resources over time, obtaining a balance between processing time and resource consumptions. Likewise, as the architectures are intended to be implemented on a FPGA device, it is necessary to consider in all the study the effect of the finite precision representation. The finite precision effect is significant when obtaining suitable performance. For it, the device on which the architecture is about to be implemented, has been carefully considered using the maximum word width for both, the arithmetical cells and the available memories.

A floating point model and a fixed point model have been developed to verify the correctness of the designed architectures. These models allow to quantify the effect of the finite precision and in turn to verify that the developed implementation is correct. The simulation models and the experimental tests have been evaluated in different practical environments. This allows not only an accurate verification of proposals, but also the corroboration with the simulated results. The considered scenarios have taken into account different types of transmission channels, gradually increasing the level of noise and attenuation.

The first case has used an ideal channel in order to study the effect of the finite precision representation on the proposed architectures. The second case has employed a SMA cable as a transmission channel. The SMA cable presents a good frequency response and therefore it is possible to compare the quality of the architectures in a complete transmission system, whose transmission channel is close to the ideal one. Finally, the third case includes a more aggressive channel formed by a twelve-meter electrical cable. With this channel, an estimation of the behavior of the medium access technique has been achieved in a channel similar to the real one. This estimation has been done without using any module of channel estimation/equalization module.

Agradecimientos

En primer lugar quiero agradecer a mis directores de tesis, Álvaro Hernández Alonso y Raúl Mateos Gil, su ayuda durante el transcurso de ésta. Puedo decir que he tenido la inmensa fortuna de trabajar con unos magníficos investigadores y aún mejores personas. Desde que empecé mi etapa como investigador hace seis años el apoyo de Álvaro ha sido clave. Durante todo este tiempo, su exigencia, dedicación y crítica ha sido fundamental en el día a día. No puedo tener nada más que palabras de agradecimiento a una persona que ha sido paciente conmigo aconsejándome en los momentos de duda y siempre con su mejor disposición. Mostrar mi gratitud a Raúl por su incondicional estímulo y apoyo en el ámbito de las FPGAS y su entorno, y así como sus constructivos comentarios que me han ayudado a formarme académicamente y profesionalmente. Seguro que en un futuro, lo agradeceré.

Quisiera agradecer a mis padres, que me han apoyado a lo largo de todos estos años de estudio. Nunca dudaron de que yo pudiera alcanzar metas como esta que hoy intento agradecerles. También he contado con el entusiasmo de mi hermano que me ha alegrado las tardes de estudio.

Echando la vista atrás me doy cuenta que no pude empezar mi trayectoria en mejor laboratorio que éste, puedo considerar que más que compañeros de laboratorio me llevo grandes amigos. Ha sido muy importante para mí su apoyo día a día desde el principio. Gracias a Jorge por esas cenas y esos paseos en bici, a José Manuel por esos momentos de diversión en el laboratorio y a David por enseñarme a encontrar siempre la excepción a la regla.

Agradecer también al resto de compañeros del grupo GEINTRA con los que he tenido el gusto de compartir esta etapa de mi vida, Enrique, Raquel, David, José María, Osmell, Cristina, Daniel, Edel, Francisco, Joaquín, Alfredo, César, y demás integrantes del grupo. Gracias a su *Excelencia* me han hecho disfrutar de la Universidad como nunca hubiera imaginado. De igual forma tampoco me puedo olvidar de mi breve estancia en Cork, donde conocí a gente que me hicieron sentir como en casa e hicieron de esta estancia una bonita experiencia personal y profesional de mi vida. Y por supuesto, mis amigos de siempre de Marchamalo y Guadalajara, con los que he tenido grandes momentos y estos últimos años me han ayudado a aliviar esos momentos en los que creía que esta tesis no llegaría a su fin.

Tampoco me puedo olvidar de los profesores Jesús Ureña Ureña, Mari Carmen Pérez, Juan Jesús García, Juan Carlos García, José M. Villadangos y el resto de profesores pertenecientes al grupo de investigación. Es un placer haber podido trabajar con ellos ya que forman un excelente equipo y fomentan un gran ambiente de trabajo.

Finalmente, agradecer a Marta su apoyo y ayuda incondicional, aun cuando encontraba muy difícil comprender lo que estaba desarrollando. Sin ella, no hubiera podido llegar tan lejos, y además consiguiendo transformarme en una mejor persona.

Índice general

Resumen.....	I
Abstract.....	III
Agradecimientos	V
Índice general.....	VII
Índice de figuras.....	XIII
Índice de tablas	XVII
Abreviaturas.....	XIX
Introducción	1
1.1. Contexto de la tesis	3
1.2. Estructura de la tesis.....	4
Antecedentes / Estado del arte	7
2.1. Transmisiones de banda ancha.....	7
2.2. Modulación Multi-portadora (MCM)	8
2.3. Modulación multi-portadora ortogonal OFDM	11
2.4. Alternativas a OFDM.....	16
2.5. Alternativas tecnologías	19
2.6. Definición de objetivos	23
Definición del SoC.....	27
3.1. Arquitectura SoC propuesta	28
3.2. Controlador DMA	30

3.3. Transmultiplexor como Periférico Avanzado	31
3.3.1. Módulo interfaz de transmisión	32
3.3.2. Módulo interfaz de recepción	36
3.4. Comunicaciones en la arquitectura propuesta.....	38
3.5. Conclusiones	40
Técnica de acceso al medio basada en la Transformada Trigonométrica Discreta	41
4.1. Extensión Simétrica	43
4.1.1. Efecto de la precisión finita	44
4.1.2. Consumo de recursos	44
4.1.3. Latencia	44
4.2. Algoritmo 1 para la realización de la Transformada Discreta del Coseno	45
4.2.1. Efecto de la precisión finita del transmisor	46
4.2.2. Efecto de la precisión finita del receptor	49
4.3. Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 1	50
4.3.1. Consumo de recursos.....	53
4.3.2. Cálculo de las restricciones de la arquitectura.....	54
4.3.3. Obtención de los parámetros para la FFT y la DCT.....	55
4.3.4. Arquitectura del transmisor	56
4.3.5. Arquitectura del receptor	60
4.4. Algoritmo 2 para la realización de la Transformada Discreta del Coseno	63
4.4.1. Efecto de la precisión finita del transmisor	64
4.4.2. Efecto de la precisión finita del receptor	66
4.5. Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 2.....	67
4.5.1. Consumo de recursos.....	68
4.5.2. Cálculo de las restricciones de la arquitectura.....	68
4.5.3. Obtención de los parámetros para la FFT y la DCT.....	69
4.5.4. Arquitectura del transmisor	70
4.5.5. Arquitectura del receptor	74

4.6. Algoritmo 3 para la realización de la Transformada Discreta del Coseno	77
4.6.1. Efecto de la precisión finita del transmisor	77
4.6.2. Efecto de la precisión finita del receptor	79
4.7. Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 3	81
4.7.1. Consumo de recursos	81
4.7.2. Cálculo de las restricciones de la arquitectura.....	82
4.7.3. Obtención de los parámetros para la FFT y la DCT.....	83
4.7.4. Arquitectura del transmisor	84
4.7.5. Arquitectura del receptor	88
4.8. Comparativa de los tres algoritmos desarrollados en la técnica basada en Transformadas Trigonómicas Discretas	91
4.8.1. Comparativa de consumo de recursos	91
4.8.2. Comparativa salida del transmisor.....	91
4.8.3. Comparativa salida del receptor	92
4.8.4. Comparativa salida del sistema implementado.....	92
4.8.5. Comparativa de prestaciones	94
4.9. Arquitectura de la Transformada Trigonómica Discreta para banda estrecha... 95	
4.9.1. Estudio de la precisión finita	95
4.9.2. Consumo de recursos	96
4.9.3. Cálculo de la latencia.....	97
4.9.4. Resultado SNR	97
4.10. Conclusiones	98
Técnica de acceso al medio basada en bancos de filtros polifásicos	99
5.1. Módulo multiplicación Θ	102
5.2. Transformada Discreta del Coseno	102
5.2.1. Efecto de la precisión finita	103
5.2.2. Consumo de recursos	105
5.2.3. Cálculo de las restricciones de la arquitectura.....	106
5.3. Matrices I – J	107

5.4. Banco de filtros polifásicos	108
5.4.1. Efecto de la precisión finita	109
5.4.2. Consumo de recursos	110
5.4.3. Latencia	110
5.5. Arquitectura del sistema.....	110
5.5.1. Obtención de parámetros de la FFT y de la DCT.....	110
5.5.2. Obtención parámetros del Banco de filtros y el módulo Matriz.....	111
5.5.3. Arquitectura del transmisor	112
5.5.3.1. Bloque DCT y bloque Theta.....	112
5.5.3.2. Módulo Matriz	116
5.5.3.3. Banco de filtros.....	117
5.5.4. Arquitectura del receptor	120
5.6. Sistema completo basado en FBMC	124
5.6.1. Estudio de la precisión finita	124
5.6.2. Consumo de recursos	125
5.6.3. Latencia	125
5.6.4. Comparativa entre los distintos modelos y la implementación real	126
5.6.5. Resultados obtenidos para el modelo en coma fija y el modelo en coma flotante.....	126
5.7. Estudio para la mejora de los filtros polifásicos	127
5.7.1. Efecto de la cuantificación del coeficiente	127
5.7.2. Efecto de la precisión finita sobre la estructura del filtro.....	128
5.7.2.1. Ampliación del datapath	132
5.7.3. Estudio del consumo de recursos.....	133
5.7.4. Análisis de la relación señal-ruido en el diseño final	134
5.8. Conclusiones de la transformada basada en banco de filtros.....	135
Técnica de acceso al medio basada en portadora única.....	137
6.1. Arquitectura del sistema.....	140
6.1.1. Arquitectura del transmisor	140

6.1.2. Arquitectura del receptor	142
6.2. Sistema Completo de la técnica de portadora Única.....	148
6.2.1. Estudio de la precisión finita, consumo de recursos y latencia	148
6.2.2. Resultados obtenidos para el modelo en coma fija y el modelo en coma flotante.....	149
6.3. Conclusiones	149
Resultados	151
7.1. Etapas de calibración y sincronización	154
7.1.1. Módulo de control del DAC	155
7.1.2. Módulo de control del ADC	156
7.1.3. Módulo de sincronismo	157
7.2. Recursos empleados por las arquitecturas en la FPGA.....	158
7.3. Resultados obtenidos con un canal ideal.....	160
7.4. Resultados obtenidos con un canal consistente en un cable SMA.....	164
7.5. Resultados obtenidos con un canal similar al real	168
Conclusiones	173
8.1. Trabajos futuros	176
8.2. Publicaciones relacionadas con la Tesis	177
8.2.1. Publicaciones en Revistas Internacionales	177
8.2.2. Publicaciones en Congresos Internacionales	178
8.2.3. Publicaciones en Congresos Nacionales.....	178
Bibliografía	179

Índice de figuras

Figura 1. Comparativa de las tecnologías disponibles para la transmisión en banda ancha.	8
Figura 2. Comparativa del espectro de frecuencia entre mono-portadora y multi-portadora.	9
Figura 3. Ejemplo de canal con desvanecimiento selectivo en frecuencia para modulación multi-portadora. 9	
Figura 4. Interferencias ICI e ISI en sistemas multi-portadora.....	11
Figura 5. Diagrama de bloques típico de una transmisión OFDM.	12
Figura 6. Señal compuesta de tres tonos representada en el tiempo y visualización de éstos en la representación en frecuencia.	13
Figura 7. Espectro de emisión de OFDM.	13
Figura 8. Espectro OFDM y canal con desvanecimiento selectivo de frecuencias.....	14
Figura 9. Pérdida de información por interferencias de banca estrecha.	15
Figura 10. Representación del efecto de offset en la frecuencia de portadora de muestreo.	15
Figura 11. Diferencia espectral entre una portadora de OFDM (azul) y de FBMC (rojo).....	17
Figura 12. Diagrama de la técnica basada en Banco de filtros (FBMC).....	17
Figura 13. Diagrama de bloques de la modulación de portadora única con igualador en el dominio de la frecuencia SC-FDE.....	18
Figura 14. Diagrama de Bloques del SoC propuesto.....	29
Figura 15. Ejemplo de comunicación del controlador DMA en el bus AXI-Stream.	31
Figura 16. Diagrama de bloques del periférico avanzado.....	31
Figura 17. Diagrama de bloques de la interfaz de transmisión.....	32
Figura 18. Diagrama de la máquina de estados de la interfaz del transmisor.....	34
Figura 19. Cronograma de la interfaz de transmisión.....	35
Figura 20. Diagrama de bloques de la interfaz de recepción.	36
Figura 21. Diagrama de la máquina de estados de la interfaz de recepción.	37

Figura 22. Cronograma de la interfaz de recepción.....	38
Figura 23. Diagrama de funcionamiento del modo Scatter-Gather.....	39
Figura 24. Diagrama de bloques del transmisor DTT.....	42
Figura 25. Diagrama de bloques del receptor DTT.....	43
Figura 26. Diagrama de bloques con cuantificación empleada en el módulo DCT con el algoritmo 1.....	47
Figura 27. Error obtenido para una trama de datos en la DCT del algoritmo 1 del transmisor para una entrada aleatoria entre -1 y 1.....	48
Figura 28. Error en la DCT del algoritmo 1 del receptor para una trama de datos con una entrada aleatoria de -1 y 1.....	49
Figura 29. Diagrama de bloques del receptor para $R_p=2$, con la tasa de datos que se emplea en cada punto.....	52
Figura 30. Diagrama de bloques de la arquitectura del transmisor DTT con la DCT del algoritmo 1.....	57
Figura 31. Cronograma de la arquitectura del transmisor DTT con la DCT del algoritmo 1.....	59
Figura 32. Diagrama de bloques de la arquitectura del receptor DTT con la DCT del algoritmo 1.....	60
Figura 33. Cronograma de la arquitectura del receptor DTT con la DCT del algoritmo 1.....	62
Figura 34. Diagrama de bloques con cuantificación empleada en el módulo DCT del algoritmo 2.....	64
Figura 35. Error obtenido para una trama de datos en la DCT del algoritmo 2 del transmisor con una entrada aleatoria entre -1 y 1.....	65
Figura 36. Error en la DCT del algoritmo 2 del receptor para una trama de datos con una entrada aleatoria de -1 y 1.....	66
Figura 37. Diagrama de bloques de la arquitectura del transmisor DTT con DCT del algoritmo 2.....	70
Figura 38. Cronograma de la arquitectura del transmisor DTT con la DCT del algoritmo 2.....	73
Figura 39. Diagrama de bloques de la arquitectura del receptor DTT con DCT del algoritmo 2.....	74
Figura 40. Cronograma de tiempos de la arquitectura del receptor DTT con la DCT del algoritmo 2.....	76
Figura 41. Diagrama de bloques con cuantificación empleada en el módulo DCT del algoritmo 3.....	78
Figura 42. Error obtenido para una trama de datos en la DCT del algoritmo 3 del transmisor con una entrada aleatoria entre -1 y 1.....	78
Figura 43. Error en la DCT del algoritmo 3 del receptor para una trama de datos con una entrada aleatoria de -1 y 1.....	80
Figura 44. Diagrama de bloques de la arquitectura del transmisor DTT con la DCT del algoritmo 3.....	84
Figura 45. Cronograma de la arquitectura del transmisor DTT con la DCT del algoritmo 3.....	87
Figura 46. Diagrama de bloques de la arquitectura del receptor DTT con la DCT del algoritmo 3.....	88
Figura 47. Cronograma de la arquitectura del receptor DTT con la DCT del algoritmo 3.....	90
Figura 48. Error medio absoluto del sistema con la DCT del algoritmo 1, para una entrada aleatoria entre [-1,1].....	93

Figura 49. Error medio absoluto del sistema con la DCT del algoritmo 2, para una entrada aleatoria entre [-1,1].	93
Figura 50. Error medio absoluto del sistema con la DCT del algoritmo 3, para una entrada aleatoria entre [-1,1].	94
Figura 51. Diagrama de bloques del transmisor FBMC.	101
Figura 52. Diagrama de bloques del receptor FBMC.	101
Figura 53. Diagrama de bloques con cuantificación empleada en el módulo DCT.	104
Figura 54. Error absoluto de la salida de DCT4e de sistema con coma fija [18 12] y con coma flotante para una entrada $X[k]$ constante de 0.5.	105
Figura 55. Coeficientes empleados en el banco de filtros FIR polifásicos.	108
Figura 56. Diagrama de bloques de Matriz y banco de filtros indicando zonas de cuantificación.	109
Figura 57. Error absoluto a la salida del banco de filtros de sistema con coma fija [18 12] y con coma flotante para una entrada $x[k]$ constante de 0.5.	109
Figura 58. Diagrama de bloques de la arquitectura del módulo DCT.	112
Figura 59. Cronograma de la DCT implementada.	115
Figura 60. Diagrama de bloques de la arquitectura realizada para el módulo de matrices I y J	116
Figura 61. Diagrama de bloques del banco de filtros implementado.	117
Figura 62. Cronograma del conjunto módulo Matriz y Filtro para la transmisión.	119
Figura 63. Diagrama de bloques de la arquitectura realizada en el receptor.	120
Figura 64. Cronograma de la arquitectura del módulo Matriz y banco de filtros en el receptor.	121
Figura 65. Diagrama de bloques de la arquitectura del módulo DCT y multiplicador Θ en el receptor.	122
Figura 66. Cronograma de la DCT del receptor FBMC.	123
Figura 67. Diagrama de bloques de filtro en celosía genérico para transmisión.	129
Figura 68. Diagrama de bloques de filtro en celosía genérico para recepción.	129
Figura 69. Diagrama de bloques de filtro en celosía tipo I para transmisión.	129
Figura 70. Diagrama de bloques de filtro en celosía tipo I para recepción.	129
Figura 71. Diagrama de bloques de filtro en celosía tipo II para transmisión.	130
Figura 72. Diagrama de bloques de filtro en celosía tipo II para recepción.	130
Figura 73. Diagrama de bloques de filtro directo para transmisión.	130
Figura 74. Diagrama de bloques de filtro directo para recepción.	131
Figura 75. Diagrama de bloques del transmisor de la técnica de portadora única.	138
Figura 76. Diagrama de bloques del receptor de la técnica de portadora única.	138
Figura 77. Diagrama de bloques de la arquitectura del transmisor para portadora única.	140
Figura 78. Cronograma del transmisor de portadora única.	142

Figura 79. Diagrama de bloques de la arquitectura del receptor de la técnica de portadora única (SE y DCT).	143
Figura 80. Diagrama de bloques de la arquitectura del receptor de la técnica de portadora única (iDCT)..	143
Figura 81. Cronograma del receptor de la técnica de portadora única (SE y DCT).	146
Figura 82. Cronograma del receptor de la técnica de portadora única (iDCT).	147
Figura 83. Diagrama de bloques del sistema completo para realizar pruebas reales.	154
Figura 84. Fases para una correcta recepción de los datos para la realización de las pruebas reales.....	155
Figura 85. Diagrama de bloques del módulo de control del DAC.	156
Figura 86. Diagrama de bloques del periférico encargada del control del ADC.	157
Figura 87. Diagrama de bloques del periférico de sincronismo.....	158
Figura 88. Representación de SNR en el canal ideal para las distintas subportadoras en la técnica DTT...	162
Figura 89. Representación de SNR en el canal ideal para las distintas subportadoras de la técnica FBMC.....	162
Figura 90. Representación de SNR en el canal ideal para las distintas subportadoras de la técnica de portadora única.	163
Figura 91. Pérdida a la salida del DAC por el transformador TC4-1W (imagen del datasheet de TC4-1w).....	164
Figura 92. Representación de SNR en el canal SMA para las distintas subportadoras de la técnica DTT. .	166
Figura 93. Representación de SNR en el canal SMA para las distintas subportadoras de la técnica FBMC.....	166
Figura 94. Representación de SNR en el canal SMA para las distintas subportadoras de la técnica mono-portadora.	167
Figura 95. Representación de SNR para el canal basado en el cable eléctrico de 12 m para las distintas subportadoras de la técnica DTT.....	170
Figura 96. Representación de SNR para el canal basado en el cable eléctrico de 12 m para las distintas subportadoras de la técnica FBMC.	170
Figura 97. Representación de SNR para el canal basado en el cable eléctrico de 12 m para las distintas subportadoras de la técnica de mono-portadora.	171

Índice de tablas

Tabla 1. Error medio, máximo y desviación típica en los puntos intermedios de la DCT del algoritmo 1. ...	48
Tabla 2. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 1.	49
Tabla 3. Error medio, máximo y desviación típica en cada punto intermedio en la DCT del algoritmo 1 del receptor.....	50
Tabla 4. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 1 del receptor.	50
Tabla 5. Latencia, tiempo de procesamiento (M_c) y multiplicadores empleados para cada tipo de FFT.....	55
Tabla 6. Error medio, máximo y desviación típica en cada punto de cuantificación en la DCT del algoritmo 2.....	65
Tabla 7. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 2.	66
Tabla 8. Error medio, máximo y desviación típica en cada punto intermedio de la DCT del algoritmo 2 del receptor.....	67
Tabla 9. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 2 del receptor.	67
Tabla 10. Error medio, máximo y desviación típica en cada punto intermedio en la DCT del algoritmo 3..	79
Tabla 11. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 3.	79
Tabla 12. Error medio, máximo y desviación típica en cada punto intermedio en la DCT del algoritmo 3 del receptor.....	80
Tabla 13. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 3 del receptor. ...	81
Tabla 14. Comparativa de consumo de multiplicadores por los tres algoritmos implementados.	91
Tabla 15. Comparativa de errores obtenidos por los tres algoritmos implementados en el transmisor.	91
Tabla 16. Comparativa de errores obtenidos por los tres algoritmos implementados en el receptor.....	92
Tabla 17. Parámetros obtenidos de simulación del DTT cuantificado con distintas DCT, para el caso con canal ideal, sin ruido y con conversor de 16 bits.	94
Tabla 18. Representación en coma fija del ancho de palabra para la arquitectura propuesta DTT.	95
Tabla 19. Error medio y desviación estándar para las señales intermedias en la arquitectura propuesta DTT, asumiendo un rango de entrada de $[-1,+1]$ en $X_m[k]$	96

Tabla 20. Comparación entre los modelos en coma fija y coma flotante de la arquitectura propuesta para DTT de banda estrecha.....	98
Tabla 21. Error obtenido en el módulo DCT4e para cada caso de cuantificación.	104
Tabla 22. Representación en coma fija del ancho de palabra para la arquitectura basada en FBMC.	124
Tabla 23. Error medio absoluto y desviación estándar para las señales intermedias en la arquitectura propuesta FBMC, asumiendo una rango de entrada de [-1,+1] en $v_m[k]$	125
Tabla 24. Comparativa del error absoluto y la desviación típica del modelo en coma flotante, fija y arquitectura vhdl.....	126
Tabla 25. Comparación entre los modelos en coma fija y coma flotante de la arquitectura para FBMC.	127
Tabla 26. Comparación de errores de cuantificación en los coeficientes.	128
Tabla 27. Comparación del error de cuantificación evaluado a la salida del transmisor.	131
Tabla 28. Comparación del error de cuantificación en la salida del receptor.	132
Tabla 29. Error de cuantificación evaluado a la salida del transmisor para cada una de las arquitecturas. .	132
Tabla 30. Error de cuantificación evaluado a la salida del receptor para cada una de las arquitecturas.	133
Tabla 31. Evaluación de las operaciones empleadas para el filtro de transmisión.	133
Tabla 32. Evaluación de las operaciones empleadas para el filtro de recepción.....	133
Tabla 33. Estimación del consumo de recursos en el dispositivo Virtex6 para transmisión.....	134
Tabla 34. Estimación del consumo de recursos en el dispositivo Virtex6 para recepción.....	134
Tabla 35. Parámetros obtenidos de simulación del FBMC cuantificado con distintos bancos de filtros.....	135
Tabla 36. Representación en coma fija del ancho de palabra para la arquitectura de portadora única.	148
Tabla 37. Error medio y desviación típica para las señales intermedias en la arquitectura de portadora única, asumiendo una rango de entrada de [-1,+1] en $x_m[n]$	149
Tabla 38. Comparación entre los modelos en coma fija y coma flotante para la técnica mono-portadora. .	149
Tabla 39. Consumo de recursos del sistema global en una FPGA Virtex 6 XC6VLX240T.	158
Tabla 40. Consumo de recursos de la técnica DTT en una FPGA Virtex 6 XC6VLX240T.....	159
Tabla 41. Consumo de recursos de la técnica FBMC en una FPGA Virtex 6 XC6VLX240T.	159
Tabla 42. Consumo de recursos de la técnica de portadora única en una FPGA Virtex 6 XC6VLX240T..	159
Tabla 43. Resultados obtenidos en las pruebas experimentales para el canal ideal.....	160
Tabla 44. Resultados obtenidos en las pruebas reales realizadas para el canal SMA.	165
Tabla 45. SER obtenido para distintas modulaciones PAM en el canal SMA.....	168
Tabla 46. Resultados obtenidos en las pruebas para una canal real.....	169
Tabla 47. SER obtenido para las modulaciones PAM en el canal basado en el cable eléctrico de 12 m.....	171

Abreviaturas

ADC	Analog Digital Converter
ADSL	Asymmetric Digital Subscriber Line
AM	Amplitude Modulation
AMS	Analog Mixed Signal
ASIC	Application-Specific Integrated Circuit
ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
AXI	Advanced eXtensible Interface
BRAM	Block Random Access Memory
CFO	Carrier Frequency Offset
CP	Cyclic Prefix
CPU	Central Processing Unit
CSoC	Configurable Systems-on-Chip
DAB	Digital Audio Broadcasting
DAC	Digital-to-Analog Conversion
DCT	Discrete Cosine Transform
DDR	Double Data Rate
DFT	Discrete Fourier Transform
DMA	Direct Memory Access
DSL	Digital Subscriber Line

DSP	Digital Signal Processors
DTT	Discrete Trigonometric Transform
DVB	Digital Video Broadcasting
FBMC	Filter Bank Multiple Carrier
FCC	Federation Communications Commission
FDE	Frequency-Domain Equalizer
FFT	Fast Fourier Transform
FIFO	First In First Out
FIR	Finite Impulse Response
FM	Frequency Modulation
FMC	FPGA Mezzanine Card
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine
HDSL	High bit rate DSL
HC-OFDM	Hermite-symmetric subcarrier Coded OFDM
ICI	Inter-Carrier Interference
IDCT	Inverse Discrete Cosine Transform
ISDL	ISDN DSL
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
ISDN	Integrated Services Digital Network
ISI	Inter-Symbol Interference
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
LUT	LookUp Table
MAC	Multiplier-ACumulator
MCM	Multi-Carrier Modulation
ME	Maximum Error
MIMO	Multiple Input Multiple Output

MMCM	Mixed-Mode Clock Manager
MSE	Mean Square Error
OFDM	Orthogonal Frequency Division Multiplexing
OQAM	Offset Quadrature Amplitude Modulation
OOK	On-Off-Keying
PAM	Pulse Amplitude Modulation
PAPR	Peak-to-Average Power Ratio
PLC	Power-Line Communications
PSNR	Peak Signal Noise Ratio
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
RAM	Randon Access Memory
rSoC	reconfigurable Systems-on-Chip
ROM	Read-Only Memory
SC-FDE	Single Carrier Frequency Domain Equalizer
SE	Symmetric Extension
SER	Symbol Error Rate
SMA	SubMiniature version A
SNR	Signal-Noise Ratio
SoC	Systems-on-Chip
TDM	Time Division Multiplexing
UWB	Ultra-WideBand
VDSL	Very High data rate DSL
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Capítulo 1

Introducción

Desde el principio de los tiempos, el hombre se ha caracterizado por su capacidad para la comunicación. Los egipcios representaban sus ideas mediante jeroglíficos, escribiendo los símbolos en piedra, madera o papiros, que después podían ser transportados con el mensaje. Los griegos empleaban superficies que reflejaban la luz del sol para la comunicación, mientras que los romanos utilizaban antorchas distribuidas en las cimas de las montañas para comunicarse. En África, se empleaban comunicaciones acústicas como el tambor o cantos, y en Norteamérica, los indios hacían señales de humo.

Con el descubrimiento de la electricidad en el siglo XVIII, se abrió una nueva ventana de posibilidades para la realización de comunicaciones. Para el desarrollo de estas comunicaciones se emplean distintas técnicas de acceso al medio. Una de las primeras técnicas es el telégrafo, empleando el código Morse en las transmisiones en 1830. Le siguió el teléfono en 1860, el cual permitía la comunicación de dos personas mediante la voz. Después se dió paso a la era de las comunicaciones inalámbricas. Con ellas se comenzaron a aplicar distintas modulaciones en las señales. La primera de ellas fue la transmisión AM (Amplitud Modulada) a mitad de la década de 1870. Más tarde en 1926, apareció la modulación en frecuencia (FM) que permitía mejor calidad en la transmisión. En esta época nace también la televisión, las comunicaciones vía microondas (1930) y la multiplexión por división de tiempo (TDM).

Con la transmisión del primer satélite en 1957, se produjo una ampliación de las posibilidades de comunicación. Junto a esto, se introduce la tecnología de banda ancha para

la transmisión lo que permitía la distribución de numerosos canales de televisión, radio y telefonía. En esta época, también se realizan las primeras pruebas de transmisión de datos mediante impulsos de luz.

Hasta este momento se han trabajado con tecnologías de transmisión analógica, para a continuación centrarse en transmisiones digitales. Con esto, nacen tecnologías inalámbricas, como el Bluetooth o el Wi-Fi en 2000. Pero el nacimiento de Internet en 1969 y de la telefonía móvil en 1973 son, sin duda, los dos mayores portales para la expansión de las comunicaciones. Con ellos, los tipos de transmisiones y las modulaciones han evolucionado de una manera exponencial, lo que ha permitido que en un periodo corto de tiempo se desarrollen sistemas inimaginables años atrás. Dado que en los últimos años ha aumentado la necesidad de un ancho de banda mayor y la necesidad de transmitir más datos en las comunicaciones inalámbricas, los sistemas actuales han intentado evolucionar con el fin de adaptarse a nuevas mejoras en las prestaciones. Para ello se han introducido nuevas modulaciones que emplean múltiples portadoras en la transmisión de datos, lo cual permite mejores prestaciones que las modulaciones con una única portadora.

La modulación multi-portadora (MCM, multi-carrier modulation) basada en la Transformada Discreta de Fourier (DFT, Discrete Fourier Transform) ha sido la técnica de acceso al medio dominante en las comunicaciones de banda ancha (xDSL, Wi-Fi, WiMax, DAB, DVB, PLC, etc.) [1] [2] [3] [4] y recientemente también se ha propuesto su utilización en comunicaciones móviles (LTE y LTE-A, enlace descendente) [5]. La modulación multi-portadora presenta grandes ventajas, como su efectividad para combatir los efectos del multitrayecto o los desvanecimientos selectivos en frecuencia, o la posibilidad de utilizar algoritmos de asignación de bits (bit loading) [6] para mejorar el rendimiento del sistema con un incremento significativo de la tasa de transmisión por subportadora. Sin embargo, la MCM basada en la DFT no está exenta de inconvenientes: sensibilidad a la sincronización en tiempo y especialmente en frecuencia, elevada relación entre la potencia de pico y la potencia media (PAPR), y escasa discriminación frecuencial. Esto conlleva un mal comportamiento del sistema en entornos ruidosos fijos y móviles, especialmente con interferencias de banda estrecha. Como consecuencia de las anteriores limitaciones, diversos estándares contemplan ya la utilización de otro tipo de técnicas de acceso al medio. Éste es el caso del IEEE 1901 para PLC (*power-line communications*), que establece las normas para las comunicaciones de alta velocidad a través de la red eléctrica, incluyendo bancos de filtros coseno modulado. Por otro lado, el estándar de 3GPP para LTE (long term evolution) propone el uso de modulación de portadora única con acceso múltiple por división en frecuencia (single-carrier frequency division multiple access, SC-FDMA) para el caso del enlace ascendente.

Atendiendo a criterios de implementación, resulta cada vez más relevante la necesidad de, una vez desarrolladas nuevas técnicas de acceso al medio, proceder a la implementación de las mismas, adaptando y optimizando los algoritmos en función de los estándares y aplicaciones. Por tanto, si se tiene en mente como objetivo final el funcionamiento en tiempo real de cualquier técnica de acceso al medio, resulta de vital importancia una

adecuada elección de la tecnología para su implementación [7]. Concretamente, en aquellas aplicaciones en las que la elevada e intensiva carga computacional de datos así como los requisitos de banda ancha suelen ser un denominador común, no resultan viables aquellas soluciones basadas únicamente en desarrollos software.

El desarrollo actual de la tecnología permite llegar a un grado de integración tal que es posible incluir en un único circuito integrado un sistema electrónico basado en un microprocesador completo. Estos dispositivos, llamados normalmente *Systems-on-Chip* (SoC) [8] [9], pueden presentar distinta complejidad y abarcar procesadores, periféricos, memoria, o aceleradores hardware.

Para la fabricación de los SoCs, aparte de la tecnología ASIC (Application-Specific Integrated Circuit), los dispositivos FPGA (Field-Programmable Gate Array) [10] se han postulado desde el principio como tecnología base debido a sus reducidos costes de desarrollo, dando lugar a veces a derivaciones del término, tales como CSoC (Configurable SoC), o rSoC (reconfigurable SoC) [11]. En este sentido, los SoCs basados en FPGAs presentan un área de Silicio dedicado a lógica configurable, cuyo uso puede ser destinado a procesamiento específico de cada aplicación particular [12]. De esta manera, se puede aprovechar toda la versatilidad de la reconfiguración de los dispositivos FPGA, y posteriormente, cuando ya se hayan realizado las comprobaciones de funcionamiento sobre la FPGA, fabricar el prototipo verificado en un SoC propiamente dicho.

La tendencia actual de las comunicaciones hace que tanto el volumen de datos a transmitir, como la velocidad de la transmisión aumente exponencialmente. Este hecho se produce sobre todo por el empleo cada vez más grande de nuevos dispositivos y aplicaciones, siendo uno de ellos los consumibles wearables, que son sistemas que pueden monitorizar nuestra actividad. También es muy común la comunicación con el coche, o incluso, realizar la compra con el teléfono móvil. Otra de estas nuevas aplicaciones es el almacenamiento y procesamiento de la información en la nube (cloud computing), siendo una de las aplicaciones tecnológicas más empleadas en los últimos meses. Estos nuevos dispositivos generan el empleo cada vez más demandado de acceso a Internet de gran velocidad. Estos hechos provocan que cada vez sea más necesario nuevas técnicas de comunicación que permitan la transmisión de altas tasas de datos (en torno a los Gigabytes/s).

1.1. Contexto de la tesis

El trabajo realizado en esta tesis ha sido desarrollado bajo el marco de la investigación del proyecto DISSECT-SOC (ref. TEC2012-38058-C03-03), subvencionado por el Ministerio de Economía y Competitividad de España, y a través de la Universidad de Alcalá con el proyecto iPULSE (ref. CCG2014/EXP-084). Además, esta tesis ha sido apoyada económicamente por la Universidad de Alcalá con el programa FPI/UAH (ref. FPI/UAH2013) y su programa de movilidad. Con este programa, ha sido posible la realización de una estancia en la School of Engineering en la University College Cork. La

motivación del proyecto DISSECT es buscar técnicas de acceso al medio para comunicaciones multi-portadora y de portadora única con herramientas diferentes a la DFT, así como la definición de arquitecturas eficientes basadas en SoC para acometer su implementación en tiempo real. Para la realización de estas arquitecturas eficientes se debe determinar las estructuras óptimas para el funcionamiento de los algoritmos propuestos en hardware dedicado, contemplando aspectos relativos al uso de los recursos a lo largo del tiempo (secuencial, semi-paralelo, o paralelo), a la representación en coma fija, el ancho de palabra, etc. También se debe tener en cuenta una solución de compromiso entre rendimiento (throughput y latencia) y recursos consumidos. En cuanto a las técnicas de acceso al medio que se van a desarrollar, se ha de buscar propuestas que incrementen la selectividad y la discriminación frecuencial, así como que estas propuestas sean capaces de trabajar con una tasa de error de bit menor cuando existen condiciones desfavorables típicas en las comunicaciones.

Para la realización de las distintas propuestas, se ha de elegir una plataforma en la que poder implementar las arquitecturas, si bien ésta se ha de caracterizar por su versatilidad y adaptabilidad a distintos tipos de arquitecturas eficientes basadas en SoC. Esta plataforma debe permitir un diseño e integración de aceleradores hardware dedicados bajo la supervisión de un procesador de propósito general.

1.2. Estructura de la tesis

Aparte de este capítulo de introducción, la tesis ha sido dividida en los siguientes capítulos:

- Capítulo 2: Estado del arte.

Este capítulo contiene una revisión de las transmisiones de banda ancha, así como de las modulaciones multi-portadora. Primero se realiza una introducción de distintos tipos de canales para la transmisión en banda ancha, como puede ser la red eléctrica, inalámbrica, fibra óptica, etc, mostrando algunos de los estándares empleados para la comunicación en estos medios (WiMAX, PLC, DSL, etc). Después, se introduce la modulación multi-portadora y se muestran sus ventajas respecto a mono-portadora. A continuación, se expone la técnica OFDM (Orthogonal Frequency Division Multiplexing) y diversas técnicas multi-portadora que tratan de mejorar las prestaciones de ésta. Más tarde, se muestran distintas alternativas tecnológicas para realizar las arquitecturas. Después de esto, se muestran distintas implementaciones predecesoras de otras técnicas realizadas empleando distintas alternativas tecnológicas. Por último, los objetivos de esta tesis son descritos.

- Capítulo 3: Definición del SoC.

Este capítulo propone el diseño de una arquitectura SoC global para la integración de las distintos transmultiplexores de las técnicas de acceso al medio que se integren como periféricos avanzados. Esta arquitectura global será adaptable para poder dar soporte a las distintas técnicas que se desarrollen. Su versatilidad permitirá entregar un gran rango de

flujo de datos, así como disponer de interfaces para posibilitar la conexión de distintos tipos de arquitecturas.

- Capítulo 4: Técnica basada en Transformada Trigonométrica Discreta.

En este capítulo se estudia y analiza la técnica multi-portadora basada en transformadas trigonométricas discretas. En el estudio se propondrán distintas alternativas para la implementación de la Transformada Discreta del Coseno (DCT, Discrete Cosine Transform), y se evaluarán en función del consumo de recursos, latencia, tiempo de procesamiento y throughput. Para ello se utilizará como parámetro de diseño el ratio de paralelismo, el cual determinará el número de datapaths que se utilizarán para que los datos se procesen al mismo tiempo. Con este ratio de paralelismo se podrá analizar tanto la latencia del sistema como el número de recursos que se consumen.

- Capítulo 5: Técnica basada en bancos de filtros polifásicos.

En este capítulo se estudia y analiza la técnica multi-portadora basada en bancos de filtros. En este caso, se propondrán distintas alternativas para la arquitectura del banco de filtros polifásicos. El estudio y análisis del consumo de recursos, latencia, tiempo de procesamiento y throughput será realizado teniendo en cuenta el valor seleccionado en el ratio de paralelismo.

- Capítulo 6: Técnica basada en portadora única.

Se estudia y analiza la técnica basada en portadora única. Al contrario de las técnicas anteriores, en este caso no existen varias portadoras para la transmisión de la información. A la hora de desarrollar la arquitectura, se utilizarán los módulos desarrollados anteriormente. Como en anteriores ocasiones se analizará tanto la latencia del sistema, como el consumo de recursos del mismo, en función del ratio de paralelismo.

- Capítulo 7: Resultados experimentales.

Primero, se describirá la metodología empleada para el desarrollo de las pruebas reales. Después, se presentarán los recursos empleados en las implementaciones en la FPGA considerada. Más tarde, se evaluarán los resultados obtenidos probando las técnicas en distintos canales de transmisión de distinta complejidad. Para realizar esta evaluación, se compararán los resultados del modelo en coma flotante, el modelo en coma fija y la prueba experimental realizada.

- Capítulo 8: Conclusiones.

Se expondrán las conclusiones obtenidas del desarrollo de esta tesis, incluyendo las referencias a las publicaciones derivadas de la misma. Por último, se discutirán las distintas trayectorias posibles que se podrán continuar como trabajos futuros.

Capítulo 2

Antecedentes / Estado del arte

2.1. Transmisiones de banda ancha

A diferencia de las transmisiones en banda base, las transmisiones en banda ancha [13] permiten la transmisión de una mayor información, con una mayor velocidad de transmisión, así como una multiplexación por división en frecuencia, lo que posibilita transmitir en paralelo varias señales independientes o la misma por varios canales.

Según la Federation Communications Commission (FCC) [14], existen distintos tipos de canales para las transmisiones de banda ancha. Digital Subscriber Line (DSL) [15] es un tipo de tecnología por cable que permite el acceso a Internet transmitiendo datos digitales a través de los cables de la red telefónica. DSL de banda ancha proporciona transmisiones desde varios cientos de kbps hasta varios Mbps. Este tipo de transmisión se divide en distintos tipos de tecnologías: Asymmetric DSL (ADSL) [16], capaz de alcanzar hasta 24 Mbits/s; High bit rate *DSL* (HDSL) [17], hasta 2048 kbps; ISDN DSL (IDSL) [18], con 144 kbps, Very High data rate DSL (VDSL), hasta 52 Mbps. También se encuentra el cable modem [19], que proporciona transmisión de banda ancha empleando un cable coaxial para la transmisión de video, datos y voz. Esta tecnología es capaz de proporcionar tasas de transmisión de más de 1.5 Mbps, lo que lo hace comparable a DSL. En fibra óptica [20] la transmisión de datos supera al actual DSL en velocidad, con tasas de Gbps. La transmisión de fibra óptica tiene varias ventajas: el peso y tamaño del cable se reduce significativamente respecto al cable convencional; la fibra posee una notable mejora del ancho de banda y es capaz de alcanzar una mayor tasa de transmisión.

2.2 Modulación Multi-portadora (MCM)

Por otro lado, se encuentra la transmisión inalámbrica, con distintos tipos de estándares, WiMAX [21] [22], WiFi [23],... Este último tipo de conexión es la preferida para comunicaciones inalámbricas gracias a que ofrece un gran grado de movilidad con transmisión de banda ancha en la zona de cobertura de hasta 54 Mbps. En otra faceta, se encuentra la transmisión por satélite [24] [25], permitiendo la transmisión de datos a zona geográficas diversas, sin la necesidad de cableado, a velocidades de hasta 8 Mbps.

Relacionada con las anteriores propuestas, se encuentra la comunicación por red eléctrica (Power-Line Communication, PLC) [26] [27], consistente en transmitir empleando las líneas de distribución de energía eléctrica. PLC proporciona comunicaciones de banda ancha añadiendo una señal modulada al cableado de la red eléctrica, evitando así tener que emplear cables adicionales para la transmisión. Esta tecnología es capaz de superar los 500 Mbps en la transmisión.

La principal característica de esta tecnología es que es capaz de emplear la instalación desplegada para la red eléctrica, lo que reduce los costes por infraestructura. Esta ventaja supone que se pueda establecer un canal de transmisión en cualquier hogar o entorno que disponga de red eléctrica, proporcionando tasas de transmisión muy superiores a otros sistemas inalámbricos. Además, al emplear la red eléctrica se dispone de distintas tomas de corriente en distintos espacios, lo que permite que existan distintos puntos de conexión.

Como se observa en la Figura 1, existen diversos tipos de canales y sus correspondientes estándares de transmisión, cada uno con sus características particulares. A modo de resumen, en la Figura 1 se puede observar una comparativa entre ellos, en relación con el ancho de banda disponible y las tasas de transmisión alcanzadas.

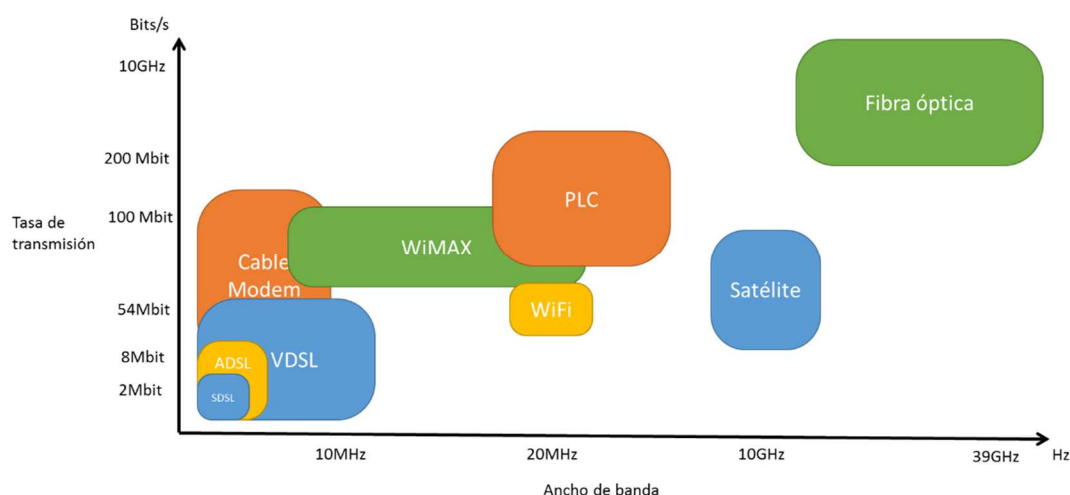


Figura 1. Comparativa de las tecnologías disponibles para la transmisión en banda ancha.

2.2. Modulación Multi-portadora (MCM)

La modulación multi-portadora MCM (Multi-Carrier Modulation) es un método de transmisión de datos basado en la división del ancho de banda del canal, y el envío de cada uno de los datos por portadoras independientes. Las portadoras tienen un ancho de banda

estrecho, pero la señal compuesta trabaja en banda ancha. Este método permite una inmunidad relativa al desvanecimiento causado por multicamino, menos susceptible que los sistemas mono-portadora al desvanecimiento selectivo en frecuencia, y mejor inmunidad a las interferencias.

En la Figura 2 se puede observar la diferencia en el espectro en un sistema mono-portadora y en un sistema multi-portadora, y cómo los sistemas multi-portadora incrementan la eficiencia espectral, empleando varias portadoras a lo largo del ancho de banda de transmisión.

La idea de usar múltiples portadoras es incrementar la eficiencia espectral del canal de transmisión, simplemente dividiendo la energía de una portadora en varias subportadoras. La modulación multi-portadora se selecciona a menudo para situaciones en las que la respuesta en frecuencia del canal y la interferencia de ruido varían con la frecuencia. En la Figura 3 se muestra un ejemplo de cómo puede ayudar una modulación multi-portadora frente a un canal selectivo en frecuencia permanente en el tiempo, evitando la transmisión por las subportadoras que se encuentran en las frecuencias que introducen una mayor atenuación o facilitando la posterior igualación del canal.

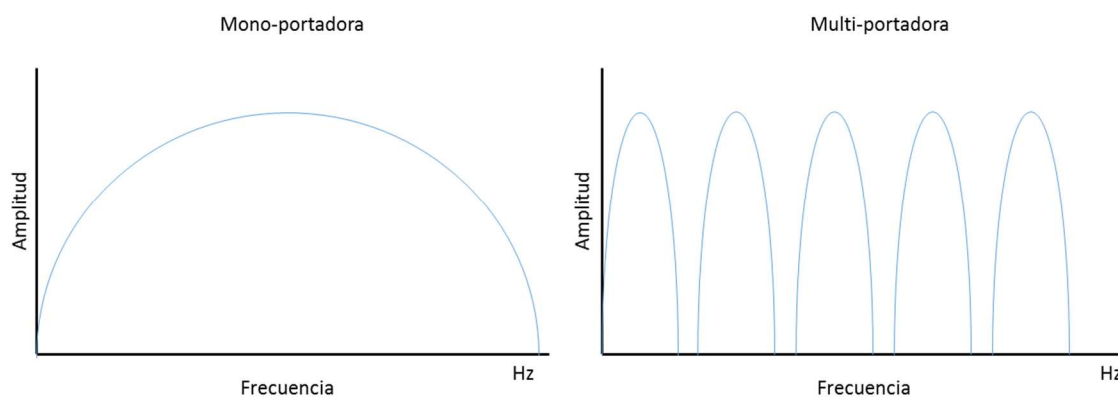


Figura 2. Comparativa del espectro de frecuencia entre mono-portadora y multi-portadora.

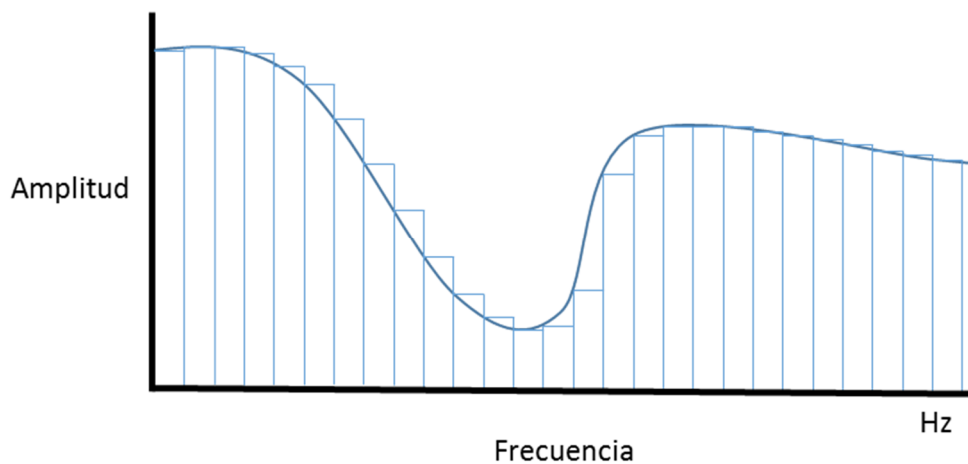


Figura 3. Ejemplo de canal con desvanecimiento selectivo en frecuencia para modulación multi-portadora.

Como se observa en la figura, al dividir un canal selectivo en frecuencia en un conjunto de subcanales se consigue un desvanecimiento aproximadamente plano en banda estrecha para cada subcanal. Esto habilita que determinando los subcanales en los que se produce un desvanecimiento en frecuencia permanente en el tiempo se puedan obviar, permitiendo que se transmita toda la información por el resto de los subcanales. Además, posibilita que la etapa de igualación necesaria se simplifique, ya que, al realizar la estimación de cada subcanal por separado, estos subcanales se pueden considerar planos. Esto hace que no se produzcan interferencias, al contrario de como ocurre en portadora única, en la que se ha de realizar una igualación adaptativa [28] de todo el canal de transmisión. Esta igualación se realiza para combatir tanto la dispersión del canal, como las interferencias adecuadamente, y ocasiona un aumento considerable de la complejidad de la arquitectura del igualador.

Esto se produce debido a que para realizar una correcta transmisión en los sistemas mono-portadora, el canal debe estar perfectamente compensado por el igualador. Sin embargo, a medida que aumenta la velocidad de símbolo, el ancho de banda de la señal es mayor. Cuando el ancho de banda es mayor que el ancho de banda de coherencia, la transmisión sufre desvanecimiento por multicamino, lo que provoca interferencias. El ancho de banda de coherencia consiste en la medida estadística del rango de frecuencias, en la cual el canal puede considerarse plano.

En general, los igualadores adaptativos basados en el dominio del tiempo se emplean para mitigar estas interferencias, con lo que la complejidad de los igualadores se incrementa con la tasa de transmisión. Esto provoca que, cuando la tasa de transmisión aumenta lo suficiente, sea poco viable la implementación del igualador. Sin embargo, con el empleo de la modulación multi-portadora se consigue dividir el ancho de banda del canal en un conjunto de subportadoras. Al dividir un canal selectivo en frecuencia en un conjunto de subcanales, se consigue un desvanecimiento plano en los subcanales de banda estrecha. Esto provoca que no existan dichas interferencias. Todo ello se consigue con un igualador en el dominio de la frecuencia (FDE) [29], el cual permite evitar la larga respuesta impulsiva que se produce en el canal.

Debido al uso de modulación multi-portadora aparecen dos tipos de interferencias que deben ser descritas: interferencia entre símbolos ISI (Inter-Symbol Interference), la cual ya se produce en modulaciones mono-portadora; e interferencia entre portadoras ICI (Inter-Carrier Interference). La interferencia ISI se produce cuando en el receptor un símbolo se ve afectado por símbolos anteriores, debido principalmente a la existencia de un canal multicamino. Un símbolo se conoce como el dato que se encuentra en un ciclo de portadora. La interferencia ICI se produce cuando una subportadora interfiere con las subportadoras contiguas. En la Figura 4 se puede observar cómo se produce las interferencias ISI e ICI.

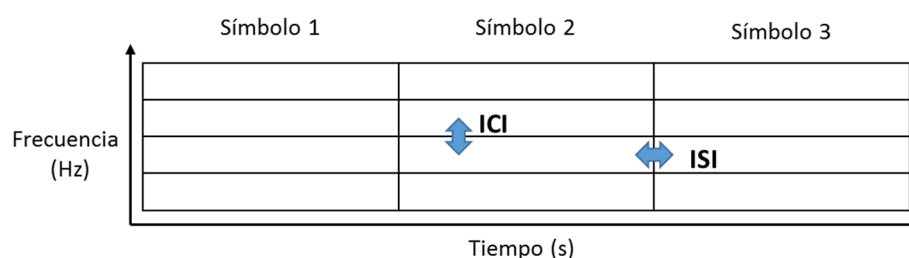


Figura 4. Interferencias ICI e ISI en sistemas multi-portadora.

Resumiendo, se puede considerar que las técnicas multi-portadora ofrecen una serie de ventajas respecto a las técnicas de portadora única. Las técnicas multi-portadora simplifican la igualación del canal, permitiendo emplear un igualador en el dominio de la frecuencia FDE (Frequency-Domain Equalizer). También se facilita el uso de técnicas adaptativas, que modifican la amplitud de la modulación en cada subportadora, ajustándolo de acuerdo a las condiciones del canal.

Por el contrario, las técnicas multi-portadoras tienen algunos inconvenientes. Poseen un mal comportamiento en ambientes ruidosos, especialmente con interferencias de banda estrecha de carácter aleatorio. Las técnicas multi-portadora también tienen una alta sensibilidad al desplazamiento en frecuencia de las subportadoras. En comparación, la modulación con portadora única es menos sensible al desplazamiento en frecuencia de la portadora, pero aumenta su sensibilidad a desplazamiento en el dominio del tiempo.

2.3. Modulación multi-portadora ortogonal OFDM

La modulación OFDM (Orthogonal Frequency División Multiplexing) [30] [31] [32] [33] es la técnica más empleada en las comunicaciones multi-portadora de banda ancha en las últimas décadas. OFDM se basa en el empleo de subportadoras ortogonales para realizar su transmisión. Esta propiedad le otorga la capacidad para emplear un número mayor de subportadoras en el ancho de banda, ya que al ser ortogonales pueden estar muy próximas entre sí, con lo que la banda de guarda que existía entre subportadoras se elimina. Con el empleo de subportadoras ortogonales se consigue aumentar la eficiencia espectral, debido a que el espectro de las subportadoras se puede superponer. Dos señales son ortogonales entre sí cuando el resultado de su producto escalar es nulo.

En la Figura 5 se muestra el diagrama de bloques básico de una transmisión OFDM. En él se puede observar cómo los datos son introducidos en un módulo, en el que se pretende realizar algún tipo de modulación en amplitud. Después, se modulan con la técnica OFDM propiamente dicha, para después añadir el intervalo de guarda, que en este caso es un prefijo cíclico. El intervalo de guarda consiste en duplicar parte de la trama de datos a transmitir. Esto permite que la interferencia ISI recaiga en este intervalo de guarda. Por último, los datos preparados se transmiten con la ayuda de un conversor digital-analógico. En el caso del receptor, primero se obtiene los datos digitales. Después se separa el prefijo cíclico añadido y con él se elimina la interferencia ISI y se demodulan los datos con la técnica

2.3 Modulación multi-portadora ortogonal OFDM

OFDM. Por último, se encuentra el módulo que elimina la modulación en amplitud introducida en el transmisor para obtener los datos.

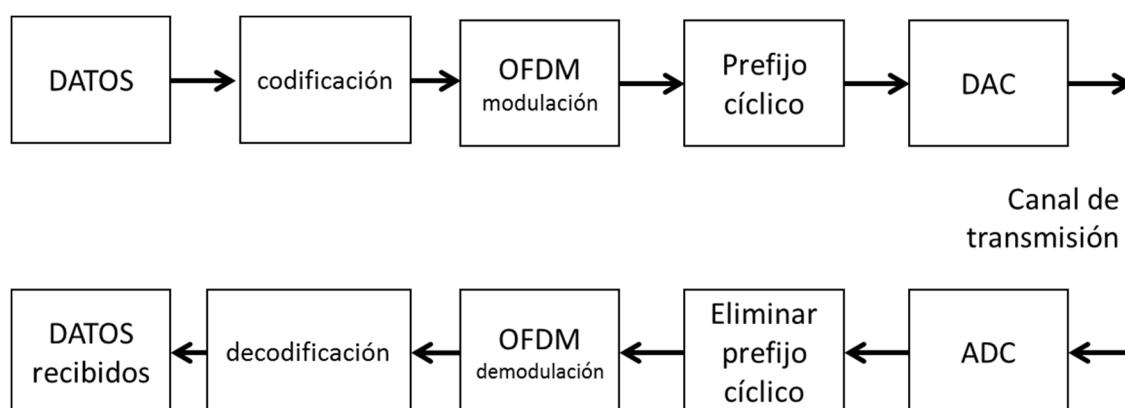


Figura 5. Diagrama de bloques típico de una transmisión OFDM.

La forma general que permite la generación y separación de las subportadoras transmitidas y recibidas es, a través, de una Transformada Discreta de Fourier DFT (Discrete Fourier Transform) para la recepción, o una Transformada Inversa Discreta de Fourier IDFT (Inverse Discrete Fourier Transform) para la transmisión, o sus versiones rápidas FFT e iFFT (Fast Fourier Transform e Inverse Fast Fourier Transform). Las funciones sinusoidales de la FFT forman un conjunto de base de señales ortogonales, y una señal en el espacio vectorial de la FFT se puede representar como una combinación lineal de las funciones seno ortogonales. Dicho de otra manera, la FFT modula cada una de las señales de entrada con cada una de las funciones base seno. Por ejemplo, si la señal de entrada tiene energía a una frecuencia determinada, habrá un pico en la correlación de la señal de entrada con la función seno correspondiente a esa frecuencia. En la Figura 6 se muestra un ejemplo; la señal está formada por tres tonos distintos de frecuencia y éstos se pueden observar en el dominio de la frecuencia. Esta transformada se utiliza en el transmisor OFDM para mapear una señal de entrada en un conjunto de subportadoras ortogonales, es decir, las funciones base ortogonales de la FFT. Del mismo modo, la transformación se utiliza de nuevo en el receptor OFDM para procesar las subportadoras recibidas. Las señales de las subportadoras se combinan entonces para formar una estimación de la señal original del transmisor.

Dado que las funciones base de la FFT son ortogonales, sólo se verá afectada la energía para una subportadora dada. La energía de otras subportadoras no contribuye en ésta, ya que es ortogonal. Esa separación de energía en la señal es la razón por la que los espectros de las subportadoras de OFDM se pueden solapar sin causar ninguna interferencia. Gracias a esta ortogonalidad, se consigue minimizar la interferencia entre subportadoras ICI, un problema que sí existe en técnicas multi-portadora sin subportadoras ortogonales. En la Figura 7 se puede observar el espectro de emisión característico de la técnica OFDM. En ella, se observa cómo se aprovecha mejor el ancho de banda, al crear un espectro de emisión más plano que con la técnica mono-portadora.

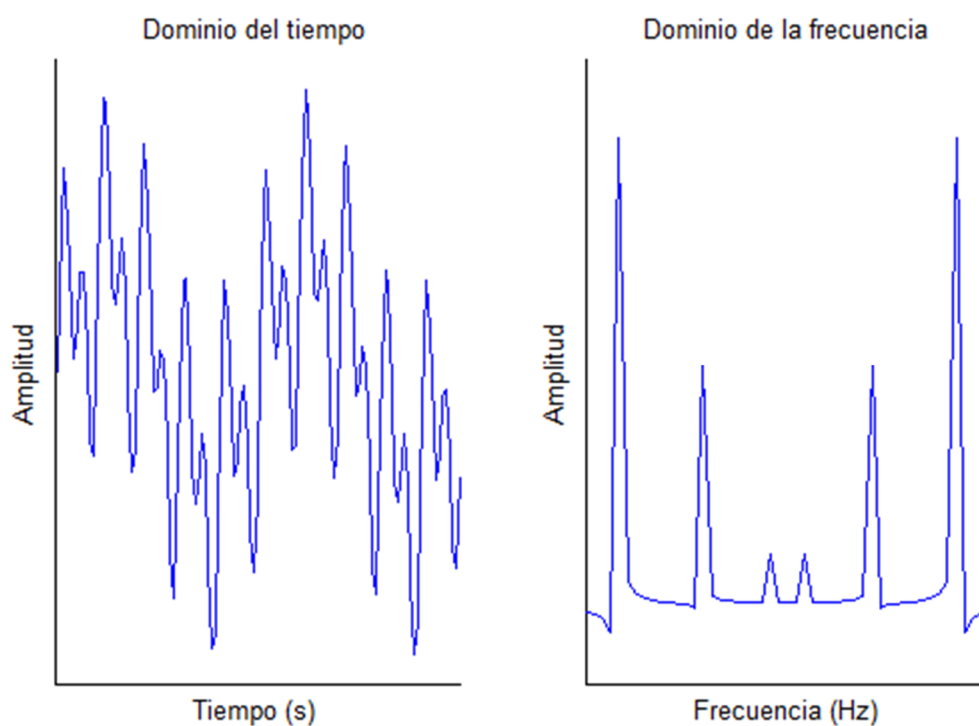


Figura 6. Señal compuesta de tres tonos representada en el tiempo y visualización de éstos en la representación en frecuencia.

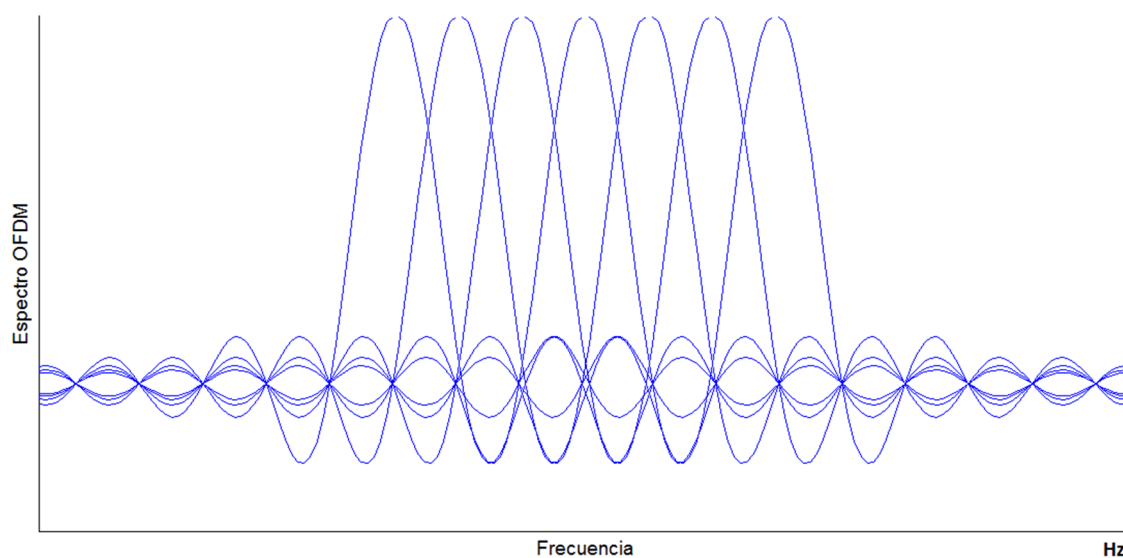


Figura 7. Espectro de emisión de OFDM.

Otra propiedad de la técnica OFDM es su robustez frente al desvanecimiento selectivo de frecuencias constante en el tiempo, ya que, como se observa en el ejemplo de la Figura 8, si se coloca el espectro de emisión de OFDM junto con la respuesta de un canal de transmisión con desvanecimiento selectivo en frecuencia, sólo ciertas subportadoras se ven afectadas, mientras que el resto se muestran intactas.

2.3 Modulación multi-portadora ortogonal OFDM

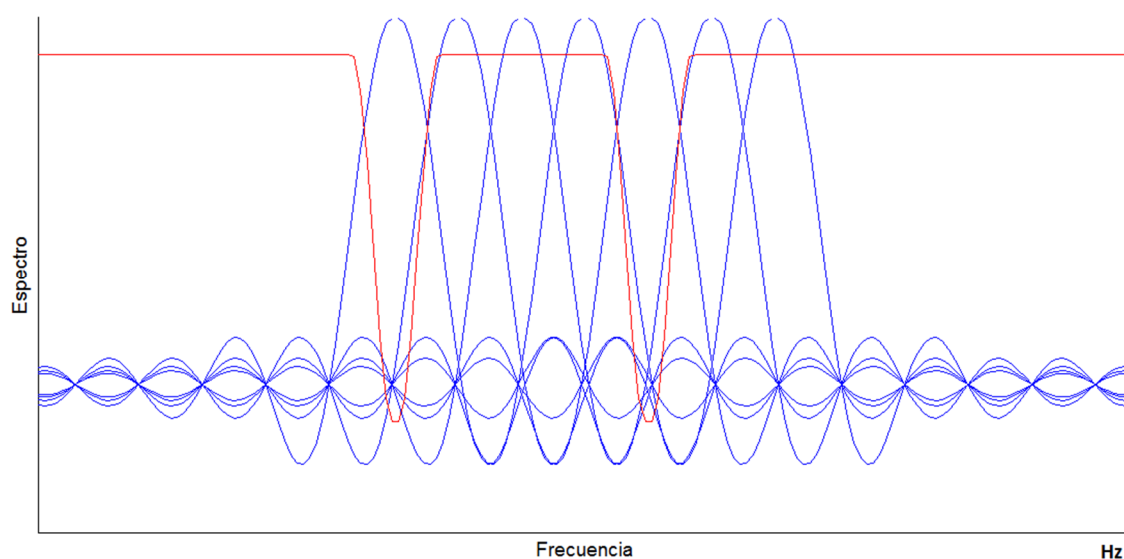


Figura 8. Espectro OFDM y canal con desvanecimiento selectivo de frecuencias.

Con estas características, OFDM se ha convertido en la técnica más utilizada, siendo empleada tanto en comunicaciones inalámbricas [31], como en comunicaciones de fibra óptica [34] [35], pasando por comunicaciones por cable [36]. Todos estos usos tienen un marco común: la mejora de la eficiencia espectral del ancho de banda del canal empleado, debido a la ortogonalidad de la técnica de acceso al medio.

Por el contrario, la técnica OFDM presenta algunas restricciones. Una de ellas es que requiere un adecuado intervalo de guarda para eliminar la interferencia entre símbolos ISI. Para tratar de evitar esta situación, es muy común que se empleen prefijos cíclicos (CP). El prefijo cíclico se genera replicando la última parte de la secuencia a transmitir y colocándola al principio. Esto produce que sea el CP el que absorbe el efecto del ISI y, eliminando en el receptor el prefijo cíclico, se pueden obtener las subportadoras sin interferencia. El empleo de un CP genera un coste de energía, y por tanto, una menor eficiencia energética.

Otra de las restricciones que presenta OFDM es que tiene un mal comportamiento en ambientes ruidosos, principalmente con interferencias de banda estrecha aleatorias. Como se observa en la Figura 9, debido al hecho de transmitir múltiples subportadoras de banda estrecha, algunas de ellas podrían verse afectadas de manera aleatoria por este tipo de ruido, impidiendo obtener la señal transmitida por esa subportadora.

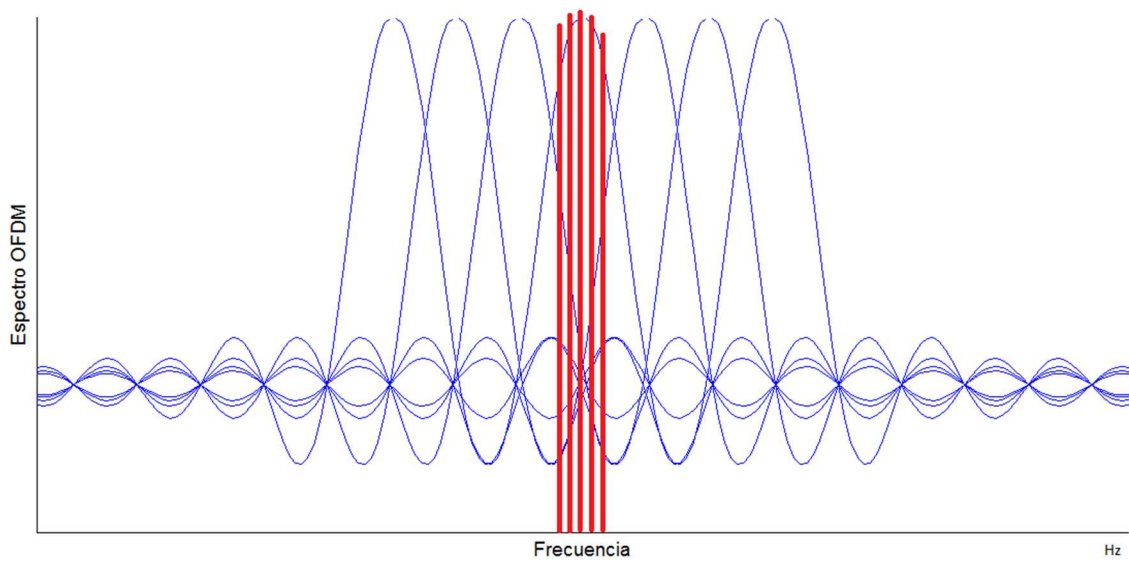


Figura 9. Pérdida de información por interferencias de banda estrecha.

Un último inconveniente de OFDM es su alta sensibilidad a errores de sincronización, en particular al desplazamiento en frecuencia de las subportadoras (Carrier Frequency Offset, CFO) [37]. El desplazamiento en frecuencia de las subportadoras se produce como consecuencia de diferencias en los osciladores de transmisor y receptor, o por desplazamiento Doppler [38]. Con el desplazamiento en frecuencia de las subportadoras se producen dos efectos: la reducción en la magnitud de la señal, por no muestrear las subportadoras en el instante óptimo, y la aparición de ICI en el sistema, provocado por la pérdida de la ortogonalidad entre subportadoras. En la Figura 10 se muestra un ejemplo del error por offset en frecuencia de portadora. En azul se muestra el punto en el cual se debería muestrear la señal y en rojo el punto en el que realmente se muestrea debido al CFO.

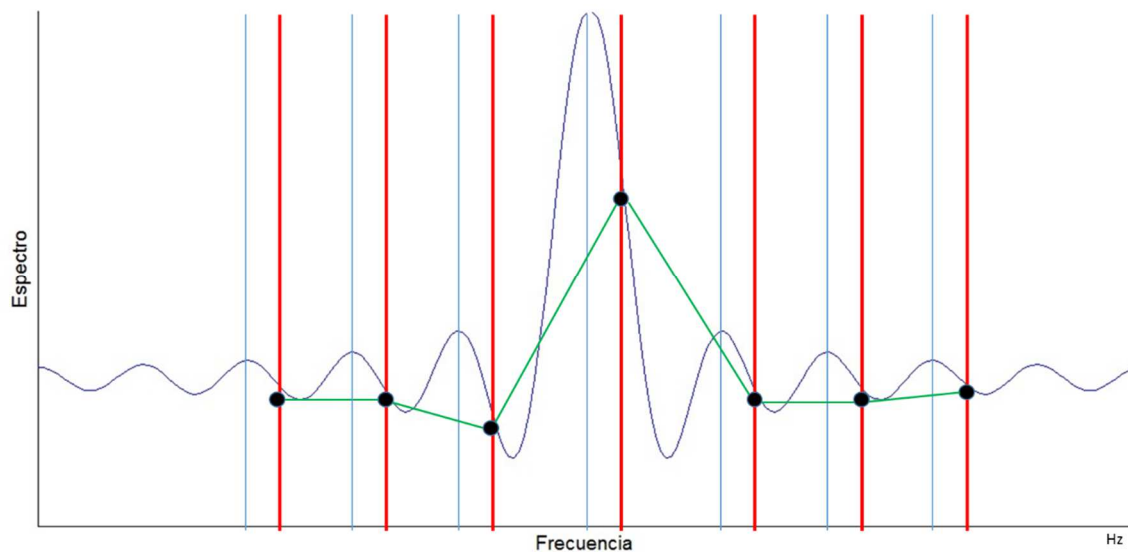


Figura 10. Representación del efecto de offset en la frecuencia de portadora de muestreo.

2.4. Alternativas a OFDM

Dadas las desventajas que presenta la técnica OFDM, en los últimos años se ha comenzado a buscar alternativas viables a esta técnica. Esto ha permitido que estas nuevas técnicas de acceso al medio sean posibles candidatas para próximas transmisiones multi-portadora [39].

Una de las alternativas que se han estudiado es la Transformada Trigonométrica Discreta (Discrete Trigonometric Transform, DTT) [40] [41]. Ésta ha sido comparada con OFDM y muestra que existe una mejora en la calidad de las transmisiones respecto a OFDM. La DTT ofrece el uso de una Transformada Discreta del Coseno (Discrete Cosine Transform, DCT) frente al uso de DFT en OFDM, permitiendo la reducción del PAPR (Peak-to-Average Power Ratio). Como se ha mencionado anteriormente, un alto PAPR obliga al aumento del tamaño del rango dinámico empleado en los convertidores analógicos-digitales, lo que favorece el incremento de la tasa de error de bits de transmisiones. Con el uso de DCT se consigue una mayor concentración de la energía y compactación del espectro. Esto permite que, bajo condiciones de desplazamiento en frecuencia de las subportadoras (CFO), la DCT genera menos interferencias ICI a subportadoras adyacentes que empleando DFT [42], resultando en una mejor robustez frente al desplazamiento en frecuencia de las subportadoras (CFO).

Otra de las características del uso de la DCT es que este tipo de transformada sólo emplea la parte real en la aritmética. Esto quiere decir que a la hora de realizar las operaciones sobre las señales de transmisión, se reduce la complejidad del procesamiento y el consumo de recursos. En cuanto al resto de características que definen a OFDM se mantienen en DTT, siendo igual de ventajosas que en OFDM. Estas características son la ortogonalidad de las subportadoras, y su consiguiente eficiencia del espectro de transmisión. También se consigue robustez frente al desvanecimiento selectivo en frecuencias, al emplearse una gran cantidad de subportadoras. Al igual que sucedía en OFDM, se ha de emplear un intervalo de guarda entre subportadoras para eliminar la interferencia ISI.

Por otro lado, se ha estudiado otra alternativa a OFDM, llamada multi-portadora basada en bancos de filtros FBMC (Filter Bank Multiple-Carrier) [43] [44]. FBMC emplea junto con la Transformada Discreta del Coseno (DCT) un banco de filtros polifásicos. Este banco de filtros polifásicos permite una separación espectral entre los subcanales más efectiva [45]. Por lo tanto, el empleo de este banco de filtros polifásicos otorga a FBMC mayor eficiencia espectral que OFDM [46]. La estimación de canal y el rendimiento del sistema pueden ser mejorados en entornos ruidosos en comparación con OFDM. Además, el uso de este banco de filtros reduce los lóbulos laterales de cada subportadora, lo que permite prácticamente eliminar las interferencias por ICI [47]. En la Figura 11 se muestra esta reducción de los lóbulos laterales (rojo), comparándola con la generada por OFDM (azul).

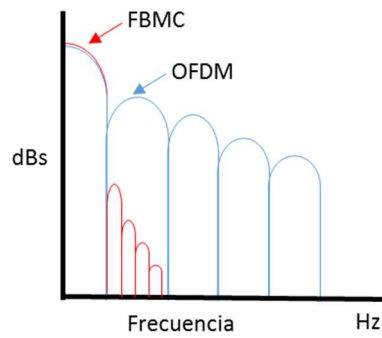


Figura 11. Diferencia espectral entre una portadora de OFDM (azul) y de FBMC (rojo).

Otra de las ventajas que produce el empleo de un banco de filtros polifásico es que no es necesario utilizar ningún intervalo de guarda, como si ocurría en OFDM o DTT. Esto permite que la tasa de bits de la transmisión sea incluso mayor [48], consiguiendo una reducción mayor de la interferencia entre subportadoras ICI y entre símbolos ISI, que con OFDM con prefijo cíclico. Una desventaja de la transformada basada en banco de filtros es que aumenta la complejidad de la arquitectura del sistema. Aunque esta complejidad se compensa debido a que los sistemas para realizar la sincronización de OFDM resultan aún más costosos computacionalmente que los empleados por FBMC [49]. La complejidad en la arquitectura de FBMC se puede observar en la Figura 12. En ella se observa el módulo encargado de la modulación, que en este caso es una DCT. Después se encuentran las matrices **I** y **J**, encargadas de realizar operaciones aritméticas con los datos que entrega la DCT. Estas operaciones aritméticas proporcionan la separación de los distintos subcanales. Además éstas son operaciones reales, lo que simplifica la arquitectura de estos módulos. A continuación se encuentra el banco de filtros polifásicos; cada uno de los filtros (representados por G_m) posee unos coeficientes específicos para cada subportadora. Se puede observar que el receptor posee los mismos módulos utilizados en el transmisor, pero colocados en orden inverso.

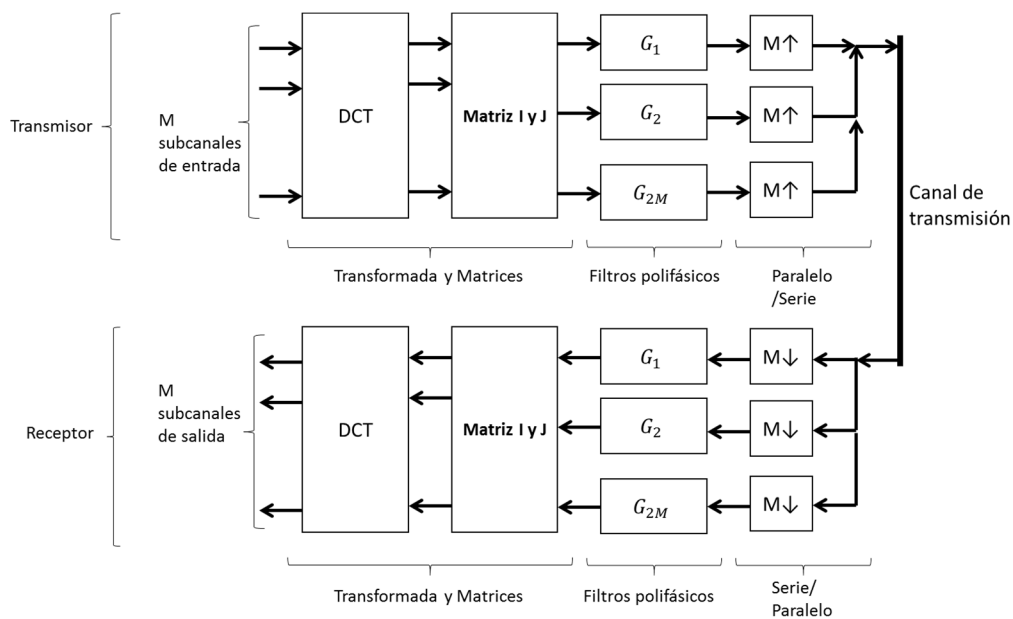


Figura 12. Diagrama de la técnica basada en Banco de filtros (FBMC).

2.4 Alternativas a OFDM

Por último se va a analizar una técnica de portadora única. Se ha comentado que uno de los inconvenientes de la utilización de modulación de portadora única era la imposibilidad de realizar un igualador adaptativo en canales de banda ancha. Sin embargo, gracias al igualador en el dominio de la frecuencia FDE (Frequency Domain Equalization) [29], ésta se puede aprovechar de las ventajas que ofrece la igualación empleada en el dominio de la frecuencia [50]. La ecualización compensa la distorsión lineal introducida por el multicamino en el canal de propagación [51]. La modulación de portadora única con igualación en el dominio de la frecuencia (Single Carrier Frequency Domain Equalizer, SC-FDE) [52] [53] es una técnica práctica para mitigar los efectos de desvanecimiento selectivo en frecuencia. Y también ofrece una sensibilidad menor al desplazamiento en frecuencia de portadora (CFO) [54]. Ésta permite un rendimiento similar a OFDM con la misma complejidad global, aunque el tiempo de respuesta impulsivo del canal sea largo.

En las transmisiones de banda ancha, los igualadores en el dominio del tiempo convencionales son poco prácticos, debido a la larga respuesta al ruido impulsivo del canal en el dominio del tiempo. En cambio, una igualación en el dominio de la frecuencia (FDE) es más efectiva en este tipo de canales. Para cambiar al dominio de la frecuencia puede emplearse una Transformada Discreta de Coseno (DCT). Con esto, la señal puede ser ecualizada con una estimación de la respuesta en frecuencia del canal. La complejidad del igualador en el dominio de la frecuencia es mucho menor que la de un igualador en el dominio del tiempo equivalente para un canal de banda ancha.

Por lo tanto, como se observa en la Figura 13, la modulación SC-FDE desplaza la DCT del transmisor, al receptor. Esto permite simplificar la arquitectura del transmisor.

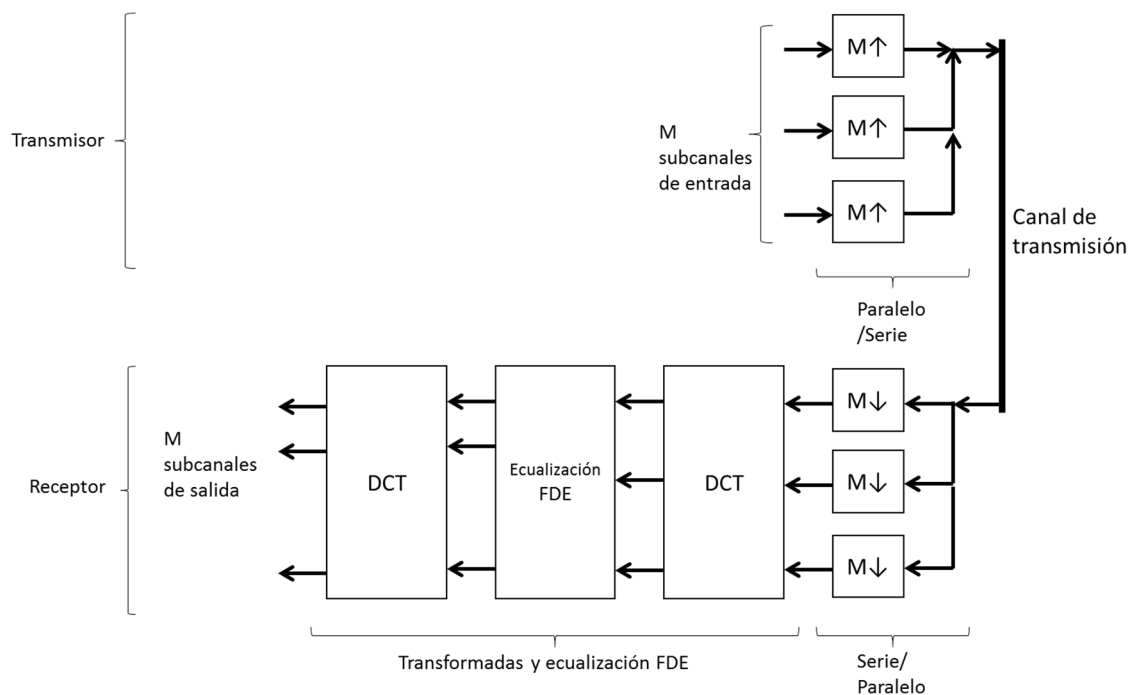


Figura 13. Diagrama de bloques de la modulación de portadora única con igualador en el dominio de la frecuencia SC-FDE.

La técnica SC-FDE emplea dos DCTs en el receptor, mientras que OFDM emplea una DCT en el transmisor y otra en el receptor. En el receptor, OFDM realiza la detección de datos por subportadora en el dominio de la frecuencia, mientras que SC-FDE realiza la detección de datos en el dominio del tiempo después de la operación adicional DCT. Esta diferencia significa que OFDM es más sensible a un espectro nulo del canal y requiere la codificación del canal o control de la potencia para superar esa vulnerabilidad, como se menciona en [55]. Además, la técnica de portadora única no se ve afectada por las no-linealidades del amplificador del transmisor, lo que permite que puedan emplearse amplificadores de potencia más eficientes y baratos [56].

Por estas propiedades SC-FDE posee las siguientes ventajas frente al uso de OFDM: En [57] se menciona que SC-FDE genera un menor PAPR debido a emplear una modulación de portadora única en el transmisor. En [58] se comenta que tiene más resistencia ante un nulo espectral. SC-FDE posee baja sensibilidad al desplazamiento en frecuencia de la portadora (CFO) comparándolo con OFDM [59]. Y debido a que la DCT del transmisor se traspasa al receptor, SC-FDE posee una baja complejidad en la arquitectura del transmisor.

2.5. Alternativas tecnológicas

El empleo de comunicaciones multi-portadora de banda ancha provoca que los flujos de datos, frecuencias de funcionamiento y número de señales a procesar aumenten. Esto implica la necesidad de buscar nuevas alternativas tecnológicas para la implementación de las técnicas multi-portadora de acceso al medio, ya que, tanto el procesamiento, como la transmisión de esas señales van a necesitar un rendimiento mayor. En este apartado, se van a estudiar distintas alternativas, y exponer cuáles son sus ventajas e inconvenientes.

La primera alternativa que se presenta es el diseño usando procesadores de propósito general [60]. El procesador de propósito general es un circuito integrado capaz de ejecutar instrucciones, realizando operaciones aritméticas y lógicas, y accesos a memoria. El procesador de propósito general se compone normalmente de una unidad de control, una unidad aritmético-lógica, varios registros y una unidad de coma flotante. La principal característica de los procesadores de propósito general es su velocidad de procesamiento, lo cual ofrece una elevada potencia de cálculo y la posibilidad de manejo de gran cantidad de memoria. En [61] se puede observar como un procesador de propósito general ha sido empleado para la realización de una comunicación PLC. Para ello, se emplea una comunicación por impulso en Ultra-WideBand (UWB), ya que este tipo de comunicación necesita una arquitectura más sencilla que permita su realización en el procesador de propósito general.

Una segunda alternativa posible es el diseño basado en microcontroladores [62] [63]. Los microcontroladores se pueden definir como un circuito integrado programable, capaz de ejecutar las sentencias almacenadas en su memoria. Dentro de un microcontrolador se puede encontrar una unidad central de procesamiento (CPU), bancos de memoria, líneas de

entrada y salida (I/O), puertos serie y paralelo, temporizadores, e incluso, conversores analógico-digitales o digitales-analógicos. La gran ventaja del uso de microcontroladores es el alto nivel de integración, ya que en un chip se dispone de toda la electrónica para realizar un sistema. Además, la configuración del diseño es otra de las ventajas que aporta, a diferencia de otros tipos de tecnologías, en los que una modificación implicaba construir un nuevo circuito, o agregar o quitar integrados. En [64] se muestra una implementación de un sistema HC-OFDM (Hermite-symmetric subcarrier coded OFDM). Éste emplea una modulación QPSK (Quadrature Phase-Shift Keying) y utiliza una FFT de 64 puntos. La cuantificación empleada es de 8 bits con lo que la sensibilidad del sistema es reducida. La tasa de transmisión empleada es de tan sólo 25ksps, debido a que, por una limitación de recursos en el microcontrolador, el tiempo de latencia causado por el procesamiento de los datos reduce el ratio de transmisión.

Otro ejemplo del empleo de microcontroladores se encuentra en [65], donde se puede observar una implementación en un microcontrolador de un sistema de comunicación en PLC. El diseño emplea una modulación on-off-keying OOK (es un caso especial de ASK, Amplitude Shift Keying) para reducir la complejidad del sistema. La frecuencia máxima de funcionamiento es de 140kHz. Su uso, debido al empleo de un microcontrolador, está indicado para bajas tasas de transmisión de datos, como por ejemplo, lecturas métricas y aplicaciones de control remoto.

A continuación, aparecen los procesadores digitales de señal (Digital Signal Processors, DSP) [66] [67] [68]; es un tipo de procesador específicamente diseñado para el procesamiento digital de señal. Los DSP utilizan arquitecturas especiales para acelerar los cálculos matemáticos que se realizan en los procesados de señal en tiempo real. Incluyen unidades MAC (Multiplier-Acumulator) para realizar de forma rápida operaciones de multiplicación y acumulación, además de arquitecturas de memoria que permiten un acceso múltiple. La mayoría de los DSP incluyen en el propio chip periféricos especiales e interfaces de entrada salida que permiten comunicaciones más eficientes, tales como conversores analógico-digital.

En [69] se presenta una modulación OFDM para comunicaciones acústicas implementada en un DSP TMS320C6713. Éste funciona a una frecuencia de 225MHz. El ancho de banda es de 5kHz y la tasa de transmisión se sitúa en 44.1ksps. La realización de la implementación en un DSP permite una arquitectura más avanzada que en el caso de un microcontrolador, como puede ser OFDM; si bien la tasa de transmisión es reducida. En [70] se realiza la implementación de un sistema OFDM. El sistema se ha implementado en dos DSPs TMS320C6201, uno para el transmisor y otro para el receptor. La tasa de transmisión es 1Mbps, empleando una frecuencia de reloj de 10MHz. Aunque la resolución interna de los DSP es de 16 bits, los conversores empleados tienen un ancho de palabra de 10 bits, lo que limita aún más la resolución. El sistema emplea una modulación QPSK en un ancho de banda de 1MHz. Para la arquitectura emplea una FFT semi-paralela de 512 subportadoras.

Como tercera alternativa tecnológica se encuentran los circuitos integrados de aplicación específica (ASIC) [71] [72]. En [73] se muestra un sistema 4x4 MIMO-OFDM para procesamiento de WLAN en banda base, implementado en un ASIC. La frecuencia de reloj del sistema es de 80MHz. La FFT que emplea es de 64 puntos y posee un ancho de banda de 20MHz. La tasa de transmisión alcanzada es 20Msps. Como se observa con esta alternativa tecnológica es posible la realización de sistemas más complejos y alcanzar tasas de transmisión más elevadas que con otras alternativas. Gracias a esta tecnología es posible alcanzar tasas de transmisión que superan ampliamente los MHz, mientras que con otras alternativas tecnológicas como los microprocesadores de propósito general esto no es posible. Además, se puede observar que el nivel de complejidad de las arquitecturas implementadas en ASIC es mucho mayor que las realizadas en otras alternativas. Esto permite la implementación de técnicas de acceso al medio mucho más elaboradas computacionalmente, ya que esta alternativa tecnológica lo permite.

Por último, queda la alternativa de los dispositivos FPGAs. Comparando las FPGAs con los procesadores de propósito general, se puede observar que, mientras éstos están basados en CPU y ejecutan instrucciones de una manera secuencial, la FPGA son dispositivos de lógica programable, lo cual le permite implementar el algoritmo de una manera paralela. Esta característica hace idóneas a las FPGAs para la implementación de las arquitecturas paralelas para sistemas en tiempo real, a diferencia de los sistemas basados en micro-controladores o micro-procesadores de propósito general, que no son viables. En la comparación de las FPGAs con los ASICs se puede observar que, pese a tener una funcionalidad similar, las FPGAs poseen un coste inicial menor; la capacidad de reconfigurarse, permitiendo una enorme flexibilidad al flujo de diseño; y sus costes de desarrollo y adquisición son mucho menores para pequeñas cantidades. Por el contrario, la FPGA ofrece un consumo de potencia mayor y una velocidad ligeramente inferior. Además, las FPGAs admiten menos complejidad y no permiten diseño mixto AMS (Analog Mixed Signal).

Con la utilización de una FPGA es posible conseguir en ciertos casos un mejor rendimiento, reduciendo el consumo de recursos, mejorando la eficiencia computacional y empleando para ello un menor consumo de energía. Como se ha observado, la FPGA también permite una adaptabilidad a distintos tipos de técnicas y en distintos medios de transmisión. Es por esto que se ha decidido realizar una búsqueda más exhaustiva de implementaciones de diseños de técnicas de acceso al medio en FPGA. Si bien, pese a la búsqueda, no se han encontrado apenas referencias en los que se muestre el desarrollo de un demostrador experimental y/o de arquitecturas eficientes.

Entre todas las técnicas a implementar, destaca OFDM, al ser la técnica multi-portadora más empleada, existiendo multitud de implementaciones de esta técnica en diversos tipos de transmisión. A continuación se van a mostrar en detalle dos diseños de implementaciones realizadas en FPGA [74] [75]. Para la implementación de la arquitectura se ha tenido en cuenta el consumo de recursos y la latencia y throughput del sistema.

Además se ha considerado el ancho de palabra máximo que dispone cada DAC, ya que influye en la resolución final.

Analizando más en detalle [74], se observa que la implementación ha sido realizada en una Zynq-7000 XC7Z020-2, conectando un front-end AD-FMCOMMS1-EBZ que posee un doble DAC de 16-bit. Este DAC posee una respuesta en frecuencia plana entre los 450MHz y los 3.8GHz. Se ha implementado un transmisor OFDM y uno FBMC/OQAM (Offset Quadrature Amplitude Modulation) para comunicación LTE (Long-term Evolution). La técnica emplea 512 subcanales con una modulación 16-QAM. Para el desarrollo de la arquitectura se han segmentado los procesos; de esta manera, consiguen una frecuencia máxima de funcionamiento de 210MHz.

En [75] se muestra una implementación en una Xilinx ML605 Virtex 6 de un transmisor OFDM para comunicaciones ópticas. Esta FPGA está conectada a un DAC AD9739A, que proporciona 2Gsps con 14 bits. La frecuencia de funcionamiento de la FPGA es de 125MHz. En el caso del transmisor OFDM posee una FFT de 64 puntos y realiza una modulación 64-QAM. La tasa de transmisión es 3.5Gbps. En los resultados se puede observar que se consigue una buena precisión al generar la cuadrícula de los 64 puntos de la modulación QAM.

Además de esas dos implementaciones, en [76] se diseña un transmisor OFDM para comunicación inalámbrica en una Virtex 2-Pro, con el que alcanzan una frecuencia máxima de funcionamiento de más de 200MHz gracias al diseño de la FFT de manera secuencial. La FFT utilizada es de 512 subcanales y emplean una modulación QAM (Quadrature Amplitude Modulation). Para el desarrollo del sistema se han empleado 32 celdas multiplicadoras, el 18% de los slices de la FPGA y un 10% de las memorias internas. El empleo de una FFT secuencial permite una mejora en la eficiencia del uso de los recursos empleados. En [77] se diseña un transmisor y receptor también para OFDM en una comunicación WLAN empleando una FPGA Quartux II. En este caso la FFT empleada es de 64 puntos. Para este trabajo se ha empleado una modulación QPSK, si bien, no se muestra la implementación realizada. En [78] se realiza la simulación de un transmisor OFDM sobre una Spartan-3A. Éste emplea modulaciones de 4-QAM, 16-QAM y 64-QAM, para una FFT de hasta 1048 puntos. Para la realización de las pruebas se emplea ruido AWGN (Additive White Gaussian Noise) con una SNR (Signal-Noise Ratio) comprendido entre 10 y 60dB.

En [79] se muestra la implementación en una FPGA Virtex 5 de un sistema FBMC que se compara con OFDM. El sistema se emplea en una comunicación WiMAX. Éste se caracteriza por tener 1024 subportadoras, con un ancho de banda de 10MHz. La tasa de transmisión es 11.2Msps.

Además de OFDM, existen implementaciones en FPGAs con otras técnicas multiportadora. En [80] se encuentra una breve descripción con enfoque LTE de un receptor FBMC. Un demostrador similar es propuesto en [81] para comunicaciones inalámbricas WiMAX, con una arquitectura multiplexada en el tiempo para un banco de filtros. Por otra

parte, se presenta una solución general de baja complejidad para un transmisor FBMC en [82]. En [83] se muestra la implementación de un sistema con modulación de portadora única con igualador en el dominio de la frecuencia. Para la implementación se ha utilizado una Virtex 4. La FFT empleada era de 128 puntos, utilizando modulación QPSK y con una tasa de transmisión de 3.125Msps.

Por último, en [84] se realiza una comparativa de un sistema OFDM desarrollado en distintas tecnologías para una comunicación LTE. Las tecnologías elegidas son un procesador RISC de propósito general, un DSP TMS320C6416 y una FPGA XC2VP30-7FF896. En la comparativa se prueban FFTs de distintos puntos (128-2048), mostrando la latencia generada. Los resultados muestran que la implementación en la FPGA es la más eficiente computacionalmente, generando una latencia menor que el resto, y siendo la que menos potencia consume.

2.6. Definición de objetivos

El objetivo principal de la tesis es el diseño de arquitecturas SoC eficientes para técnicas de acceso al medio en comunicaciones PLC de banda ancha. Para ello la arquitectura global debe ser lo suficientemente adaptable y configurable, como para soportar distintas tasas de transmisión, distintas interfaces y distintos números de subportadoras. Además, la arquitectura global tiene que dar cabida a las arquitecturas eficientes que se corresponden al hardware específico de las distintas técnicas de acceso al medio y que se implementan como periféricos avanzados de la arquitectura global. A su vez, la arquitectura global deberá ser capaz de controlar y utilizar las técnicas empleando un elevado flujo de datos, para lo cual necesitará tener la capacidad de generar y almacenar gran cantidad de información.

A partir de este objetivo principal, se pueden desglosar varios puntos:

- El primer punto es la elección de un medio de transmisión. Existen diversos medios para la transmisión de banda ancha, cada uno de los cuales ofrece distintas ventajas e inconvenientes. Dentro de esta tesis, se ha centrado en el empleo de los sistemas de comunicación de PLC de banda ancha. Este se caracteriza por emplear una infraestructura ya desplegada, como es la de la red eléctrica. Gracias a ello este sistema posee un gran potencial para convertirse en uno de los medios de transmisión más empleados en el futuro. Los nuevos sistemas PLC de banda ancha son capaces de entregar mayores tasas de datos. Para ello utilizan modulaciones multi-portadoras, con el fin de difundir los datos a lo largo de un ancho de banda más amplio.
- El segundo punto es el diseño de las arquitecturas de las técnicas de acceso al medio multi-portadora, creando un reto a la hora de su implementación en un sistema de tiempo real, ya que se ha de trabajar con un gran flujo de datos. Esto provoca que se tenga que trabajar con arquitecturas semi-paralelas, lo que hace que los sistemas basados en micro-controladores o micro-procesadores de propósito general no sean

viales. Por ello, se han propuesto arquitecturas System-on-Chip (SoC) [85], en concreto basadas en FPGAs. Con el uso de una FPGA, se consigue una flexibilidad y adaptabilidad en el dispositivo que permite el desarrollo de novedosas arquitecturas en tiempo real para técnicas de acceso al medio.

- Una vez fijada como se va a diseñar la arquitectura, se va a desarrollar una arquitectura eficiente para la implementación de la técnica de acceso al medio basada en Transformadas Trigonométricas Discretas (DTT), a partir de la Transformada Discreta del Coseno (DCT). Para la implementación de esta técnica se analizarán distintos algoritmos de la DCT de cara a la obtención de la arquitectura más eficiente, evaluando distintos parámetros como puede ser el consumo de recursos, el tiempo de procesamiento, la latencia y el throughput.
- Por otro lado, se va a desarrollar una arquitectura eficiente para la implementación de la técnica basada en banco de filtros (FBMC). Ésta se aprovecha del uso de filtros polifásicos para mejorar la eficiencia espectral de la transmisión. Dentro de esta técnica, también se evaluarán distintas aproximaciones para la arquitectura del banco de filtros polifásico, determinando cuál es la que mejor se ajusta a las restricciones impuestas al diseño.
- Además de estas dos técnicas multi-portadora, se va a diseñar la arquitectura de una técnica de acceso al medio mono-portadora. Esta técnica reduce el PAPR y posee mayor inmunidad al desplazamiento en frecuencia de las portadoras.
- Las tres arquitecturas serán evaluadas en la FPGA seleccionada, implementándose como periféricos avanzados en la arquitectura SoC global, y para ello, se debe tener especial cuidado con el efecto de la representación en coma fija. Se ha realizado un análisis de este efecto, y de cómo afecta a los resultados proporcionados por la arquitectura final. En este análisis, se ha tenido en cuenta los recursos internos disponibles en el dispositivo FPGA, como el uso de módulos específicos para la realización de multiplicaciones y acumulaciones, así como de memorias para realizar almacenamientos. Estos módulos suelen disponer de un ancho máximo de palabra limitado, que afecta a la cuantificación en la arquitectura global. Por tanto, en el análisis del efecto de la coma fija se evaluará de forma flexible al ancho de palabra a lo largo de los distintos datapaths existentes en las arquitecturas propuestas, de forma que puedan ser ajustados y minimizarse los errores procedentes de la representación en coma fija sobre los resultados finales. En este punto se hace imprescindible hablar de un nuevo parámetro, el ratio de paralelismo. La FPGA habilita la posibilidad de realizar un diseño totalmente paralelo, el cual permite aumentar el procesamiento de datos y el throughput, reduciendo la latencia del sistema. Una arquitectura completamente paralela dispara el consumo de recursos necesario para su desarrollo y reduce la eficiencia en el uso de los mismos. A partir del ratio de paralelismo se determina tanto el número de recursos empleados, como el tiempo de procesamiento y latencia del sistema. Por lo tanto, el ratio de paralelismo se convierte en un parámetro determinante a la hora de realizar la implementación maximizando la eficiencia del uso de recursos y,

permitiendo a su vez, que el sistema cumpla los requisitos para operar en tiempo real.

Para comprobar todo el análisis y arquitecturas desarrolladas, se van a realizar pruebas experimentales siguiendo la normativa del estándar para PLC, obteniendo resultados en distintos tipos de canales. Para realizar una correcta evaluación las técnicas se desarrollarán de tres maneras distintas. Primero se realizará un estudio sobre un modelo de simulación en coma flotante, el cual servirá de referencia al no incluir errores y limitaciones propias de las arquitecturas propuestas, como los procedentes de la cuantificación. Después se evaluará el modelo de simulación en coma fija, con el cual se podrá determinar la calidad del diseño y el ancho de palabra fijado a lo largo de los distintos elementos de la arquitectura. Y por último, se implementará la arquitectura de la técnica en un dispositivo FPGA. Los resultados obtenidos de estas tres aproximaciones serán comparados y analizados. Finalmente, para la realización de un demostrador experimental será necesario la configuración de sendos conversores analógico-digital y digital-analógico de altas prestaciones. Éstos deberán poseer un ancho de banda necesario para este tipo de transmisiones, así como ser capaces de entregar las tasas de transmisión definidas por el estándar para PLC.

Capítulo 3

Definición del SoC

En este apartado se presenta el diseño de una arquitectura SoC basada en FPGA [86] para la implementación en tiempo real de técnicas de acceso al medio (transmultiplexores) en PLC [26]. La arquitectura diseñada presenta la flexibilidad para acoplar como periféricos avanzados (coprocesadores hardware) el emisor y el receptor de las posibles técnicas de acceso al medio para PLC, soportando el flujo de datos de entrada y salida deseado mediante una solución basada en DMA (Direct Memory Access) [87]. Por tanto, la arquitectura supone una propuesta eficiente y flexible de implementación de la capa de acceso al medio en el diseño de transmultiplexores para PLC.

En la actualidad existen muchas técnicas multi-portadoras de acceso al medio para distintos estándares de comunicaciones [46] [88], como LTE (Long Term Evolution) [89] o PLC (Power-Line Communication) [26]. Concretamente, el estándar PLC es un tipo de comunicación que permite el uso de la red eléctrica para proporcionar información, reduciendo posibles cableados adicionales. Esto conlleva a menudo que, al emplear cables de red eléctrica, existan problemas de transmisión a elevadas frecuencias y de propagación, que son considerados en el diseño de la técnica de acceso. El nuevo sistema PLC de banda ancha es capaz de proporcionar mayores tasas de datos, mediante un conjunto de portadoras en paralelo, con el fin de difundir los datos a lo largo de un ancho de banda más amplio.

A la hora de realizar la implementación en un sistema real, se han de considerar que la arquitectura sea capaz de generar un gran flujo de datos y tratar con sistemas paralelos, y sea capaz de trabajar a altas frecuencias, en torno a los MHz. Por ello, se han propuesto

arquitecturas System-on-Chip (SoC) [85] en las que se pueden implementar en hardware las técnicas de acceso al medio deseadas, gracias a la flexibilidad y paralelismo que ofrecen.

En consecuencia, es necesario proponer y diseñar nuevas arquitecturas hardware donde las técnicas de acceso al medio se puedan implementar en tiempo real, proporcionando el ancho de banda y régimen binario requerido. De esta manera, los dispositivos FPGA se ha utilizado en trabajos anteriores [90] [91], tratando de explotar el paralelismo, la flexibilidad y las altas frecuencias de operación ofrecidas por este tipo de dispositivos.

Además, soportan frecuencias significativamente altas de reloj, necesarias para estos estándares de comunicaciones con altas velocidades de datos. Sin embargo, el diseño y la aplicación de estas técnicas multi-portadora complejas todavía tiene que considerar algunas cuestiones, como el efecto de la precisión de la coma fija en el rendimiento, la latencia, el throughput y el consumo de recursos. Teniendo en cuenta la implementación basada en FPGA, algunos trabajos previos ya se han ocupado de enfoques OFDM [92] [93] [94]. En cuanto a FBMC, se propone en [95] un transmultiplexor implementado en una FPGA, aunque no se proporcionan detalles sobre la arquitectura hardware. Otro transmultiplexor FBMC está implementado en una FPGA para un estándar LTE 64-QAM en [74], proporcionando una descripción detallada de la arquitectura paralela propuesta. También en [80] se puede encontrar un enfoque LTE para un receptor FBMC, con una breve descripción de la arquitectura propuesta. Un demostrador similar se expuso en [81], pero para el estándar WiMAX y con la arquitectura multiplexada en el tiempo para el banco de filtros, mientras que una solución general de baja complejidad se presenta en [82] para un transceiver FBMC, aunque no es descrita la arquitectura en la FPGA.

3.1. Arquitectura SoC propuesta

Para la implementación de las distintas técnicas de acceso al medio, se ha propuesto un sistema SoC global en el que se integre la técnica de acceso al medio que se quiere probar como periférico avanzado. Para ello, el sistema global posee un módulo DMA que se encarga de proporcionar el flujo de datos demandado por el periférico avanzado, así como, un microprocesador soft, llamado Microblaze [96], que se encarga de la gestión y control de todos los periféricos. Para la implementación de este sistema global, se ha decidido emplear una tarjeta ML605 de Xilinx que posee una FPGA Virtex 6 xc6vlx240t. Esta tarjeta se caracteriza por tener, entre otras cosas, dos puertos FMC necesarios para la conexión de conversores analógicos-digitales, memoria externa con capacidad para almacenar un número suficiente de datos, y recursos para la implementación de las distintas técnicas. En la Figura 14 se puede observar el diagrama de bloques del sistema implementado.

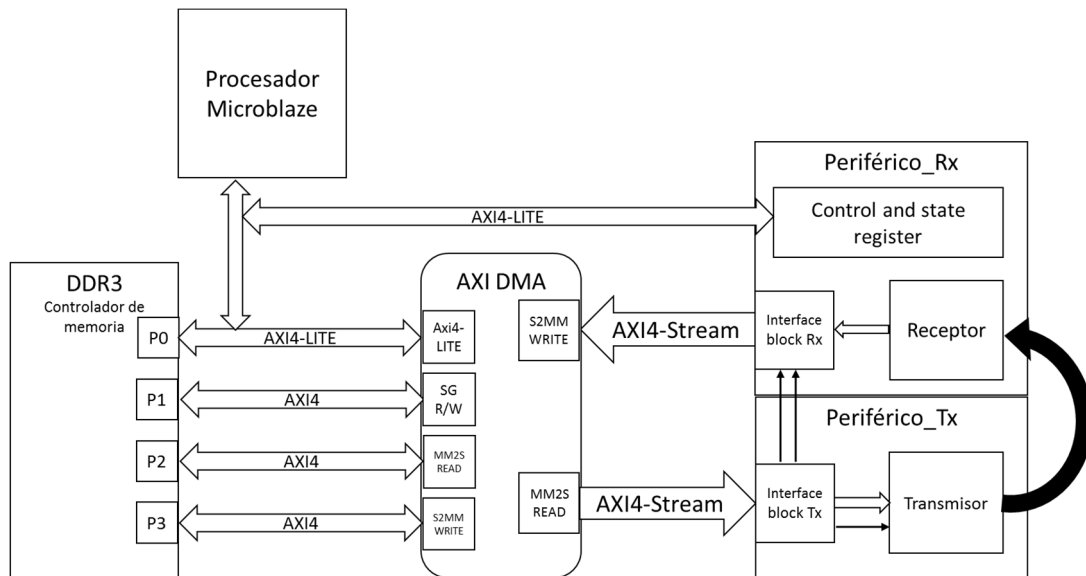


Figura 14. Diagrama de Bloques del SoC propuesto.

Como se puede observar, el microprocesador Microblaze se encarga de la gestión de los distintos módulos del sistema: un controlador DMA (Acceso Directo a Memoria), el cual se encarga de proporcionar el flujo de datos que demande el periférico avanzado; un banco de memoria externo; y el periférico avanzado que integra la arquitectura hardware de la técnica de acceso al medio bajo análisis en cada momento.

Dado que las técnicas de acceso al medio que se van a implementar poseen una solución multi-portadora, que suele presentar múltiples datapath de entrada y salida que se manejan simultáneamente y en tiempo real, se ha decidido emplear un módulo DMA. De esta manera, éste se encarga de enviar y recibir los datos del periférico avanzado, conectándose a la memoria DDR3 a través de un controlador de memoria [97]. Esto permite que el procesador Microblaze sólo realice la tarea de control en la transferencia de datos. Este módulo DMA posee dos buses, una para la escritura y otro para la lectura, lo que permite que se puedan realizar ambos procesos simultáneamente.

La memoria empleada para el envío y recepción de datos desde nuestro periférico, así como para almacenar el código del microprocesador, es una memoria externa DDR3 de 4GB, la cual asegura que se puedan transmitir y recibir más de $P=3000000$ paquetes de datos para comprobar el sistema. Este tamaño se considera un número suficiente de paquetes para realizar el testeo del transmultiplexor de la técnica de acceso al medio.

El sistema emplea, para la comunicación con los periféricos, el estándar de buses AXI [98]. Este estándar posee tres tipos diferentes de especificaciones, el bus AXI-Lite, un bus reducido para comunicaciones mediante mapa de registros; un bus AXI4, que es el bus completo de comunicaciones; y un bus AXI-Stream, que se trata de un bus punto a punto. De esta forma, se emplea un bus AXI-Lite para la comunicación con los registros de control de los periféricos, un bus AXI4 para el acceso a memoria a alta velocidad (DMA [87], controlador de memoria DDR3 [97]), y dos buses AXI-Stream, para el envío de datos entre el periférico avanzado y el DMA.

En este sentido, dentro del periférico avanzado, además de la técnica de acceso al medio propuesta, es necesario llevar cabo la arquitectura de las correspondientes interfaces para el envío y recepción de los datos a través de los buses dedicados a ese fin, el bus AXI-Stream en este caso. El empleo de interfaces entre la técnica de acceso al medio y el DMA es debido a que se hace necesario proporcionar los datos según las necesidades de la técnica implementada. Con este módulo se consigue que, independientemente de la técnica de acceso al medio que se desee implementar, el DMA sea capaz de entregar correctamente el flujo de datos demandado por el transmultiplexor. Además, se ha implementado un módulo de registros para la interacción y control, por parte del Microblaze, del periférico avanzado.

A continuación, una vez se ha mostrado el funcionamiento general de la arquitectura global del SoC, se procede a explicar los distintos módulos que la componen. Estos módulos son: microprocesador Microblaze [96], encargado de la gestión de todos los periféricos; controlador de memoria externa DDR3 [97]; módulo DMA [87], encargado de proporcionar el flujo de datos al transmultiplexor; y el periférico avanzado, que contiene el transmultiplexor de la técnica de acceso al medio que se desea implementar.

3.2. Controlador DMA

El controlador DMA [87] es el módulo encargado de escribir en memoria los datos recibidos por el periférico avanzado y de leer de la misma los datos que necesita enviar el módulo de transmisión de éste. De esta manera, se libera a Microblaze de la realización de esta tarea y permite que pueda desempeñar otras funciones de control. El DMA se encarga de los accesos a memoria que generan el flujo de datos demandado por el periférico avanzado, ya sean de escritura o de lectura.

Para realizar estos accesos de escritura y lectura a memoria, el controlador DMA emplea el bus AXI4 mencionado anteriormente [98]. Así mismo, emplea dos buses AXI-Stream para la comunicación directa con el periférico avanzado, uno para la transmisión y otro para la recepción. Este bus tipo stream (ausencia de direcciones) es un tipo de bus punto a punto, unidireccional, capaz de alcanzar grandes flujos de datos, lo cual le convierte en ideal para la realización de este diseño. Para la utilización de este bus AXI-Stream será necesario, por parte del periférico avanzado, el uso de una interfaz que permita adaptar el flujo de datos que entrega el módulo DMA, para que se ajuste a la demanda que imponga el transmultiplexor de la técnica de acceso al medio implementada. Como se emplean dos realizaciones del bus AXI-Stream diferentes para transmisión y para recepción, se han desarrollado dos interfaces diferentes según la necesidad de cada una de ellas.

A continuación, se va a describir el funcionamiento del controlador DMA con el bus AXI-Stream, pero primero, se van a describir las líneas que forman el bus. Éste está formado por la señal de datos, la señal de validación (*Tvalid*), la señal que indica que el módulo está listo (*Tready*) y la señal de último dato (*Tlast*). Como se puede observar en la Figura 15, el controlador DMA espera hasta que la señal *Tready* está activa, lo cual indica que el módulo que va a recibir los datos está preparado. Una vez activa, el DMA activa la

señal de validación (*Tvalid*) y comienza a transmitir los datos. Cuando llega al último dato transmitido, se activa la señal *Tlast*. En la figura se puede observar un ejemplo del empleo del bus AXI-Stream.

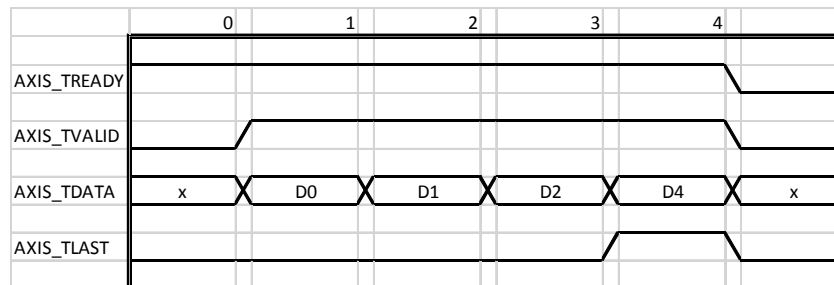


Figura 15. Ejemplo de comunicación del controlador DMA en el bus AXI-Stream.

3.3. Transmultiplexor como Periférico Avanzado

Para poder introducir la técnica de acceso al medio en el SoC [85], se ha realizado un periférico avanzado, mediante el cual poder integrar la arquitectura de la primera. Para realizar el control y la configuración del transmultiplexor, se emplea una interfaz AXI-Lite mediante una serie de registros de control (*Axi_Lite_IPIF*). También, como ya se ha comentado, se emplean dos interfaces para sendos buses AXI-Stream para proporcionar (Interfaz Tx) y transferir (Interfaz Rx) los datos obtenidos por la técnica de acceso al medio. Con todo esto se consigue que, independientemente de la técnica de acceso al medio que se emplee, la interfaz externa en la arquitectura SoC, el control y manejo del periférico avanzado sea el mismo. A su vez, permite independizar el reloj de funcionamiento del periférico avanzado del sistema global, lo cual permite que se puedan verificar distintas implementaciones de diversas técnicas de acceso al medio empleando distintas tasas de transmisión de datos. A continuación, se puede observar en la Figura 16 el diagrama de bloques del periférico avanzado propuesto, el cuál aunque se considere un único periférico a la hora de implementarlo, transmisor y receptor son dos periféricos independientes.

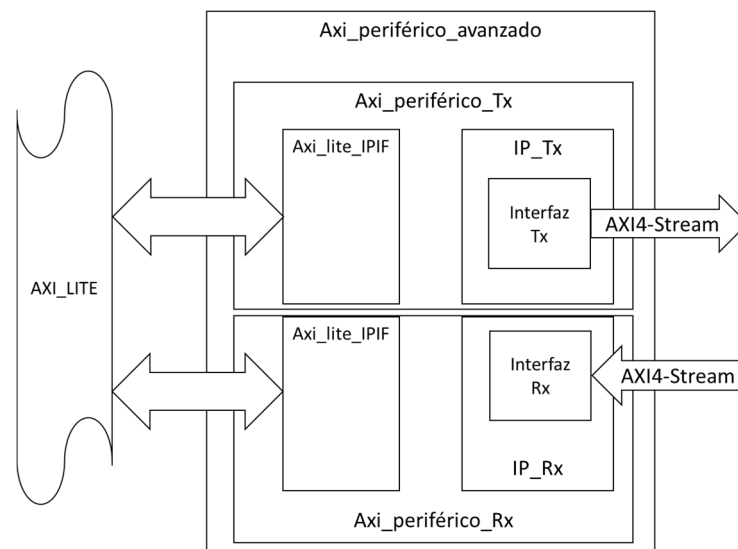


Figura 16. Diagrama de bloques del periférico avanzado.

En el diagrama se puede observar los dos buses AXI-Stream que se emplean, así como la interfaz que se ha desarrollado para cada uno de ellos y que se explica a continuación.

3.3.1. Módulo interfaz de transmisión

La interfaz de transmisión es la encargada de transferir al transmisor del transmultiplexor de la técnica de acceso al medio implementada los datos proporcionados por el módulo DMA. Además, debido a que el DMA es capaz de entregar una tasa de datos mayor de la que es capaz de procesar el transmisor del transmultiplexor de la técnica de acceso al medio, la interfaz debe ser capaz de indicar al DMA cuándo puede proporcionar los datos y a su vez, transferirlos adecuadamente al transmultiplexor. De este modo, el transmisor del transmultiplexor es el que impone la limitación del flujo de datos generado. A su vez, la interfaz de transmisión sirve para independizar el reloj de trabajo del sistema global f_{clk} del reloj de funcionamiento del transmisor del transmultiplexor f_{per} , ya que éste puede ser diferente debido a las distintas necesidades de tasas de transmisión que pueden presentar las diversas técnicas de acceso al medio.

Para el desarrollo de las funciones de la interfaz, se ha de diseñar un sistema que sea capaz de entregar todos los datos proporcionados por el módulo DMA al transmisor de una manera correcta. Para ello, se deben diseñar unos módulos para generar distintas señales de control, tanto para el transmisor como para el bus AXI-Stream [98]. Estos módulos son el banco de memorias de doble puerto, cuyo espacio se divide en dos memorias gestionadas en modo ping-pong; y un módulo generador de la señal de habilitación CE_M . La señal CE_M es la señal de habilitación de los datos de entrada al transmisor, y depende del número M de subcanales de entrada. En la Figura 17 se muestra el diagrama de bloques de la interfaz de transmisión.

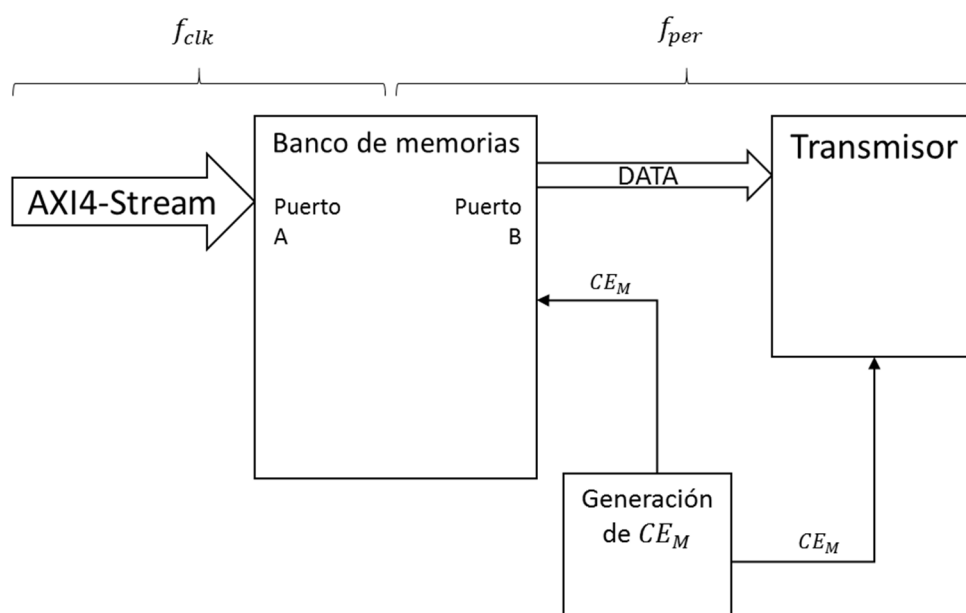


Figura 17. Diagrama de bloques de la interfaz de transmisión.

Como se observa, los datos pasan a través del banco de memorias de doble puerto para su transmisión, mientras que, se genera la señal CE_M para que se introduzcan correctamente en el transmisor. Se asume que una determinada técnica de acceso al medio dispone de una tasa de datos de entrada en cada subcanal de $T_{ch}=T_{Tx}/M$, donde T_{Tx} es la tasa de datos en el medio de transmisión (o tasa de datos de transmisión), para un transmultiplexor que divide el ancho de banda disponible en M subcanales independientes. Por lo tanto, para poder realizar una transmisión completa, es necesario disponer de un dato en cada uno de los M subcanales; de este modo, se considera al conjunto de M datos necesarios un paquete de datos P , contemplando que cada uno de ellos es para un subcanal distinto.

El puerto A del banco de memorias es el encargado de la comunicación con el bus AXI-Stream, así como de proporcionar los datos obtenidos a dicho banco. Para ello, genera tanto las direcciones y señales de control del puerto A del banco de memorias de doble puerto, como las señales del bus AXI-Stream.

Como se ha comentado antes, el bus AXI-Stream es un bus punto a punto que permite la entrega de datos sin el empleo de direcciones de memoria. En la Figura 18 se muestra el diagrama de la máquina de estados que se emplea. Como se observa, el bus AXI-Stream está compuesto por los datos que se van a proporcionar ($Tdata$); una señal de validación ($Tvalid$); una señal de último dato ($Tlast$), que se activa cada M datos; y la señal $Tready$, que indica que la interfaz está lista para recibir los datos. El puerto A del banco de memorias se encarga de activar la señal $Tready$ en función de si la memoria está llena o no ($full$), para indicar al módulo DMA que está disponible para recibir datos. A su vez, cuando la señal de habilitación $Tvalid$ está activa, el puerto A del banco de memorias genera la señal $fifo_we$ y comienza a generar direcciones de memoria, a la vez que, comienza a transmitir los datos que le proporciona $Tdata$. Cuando la señal $Tlast$ se activa, el puerto A activa a su vez la señal wr_pkt , que a su vez activa la señal $full$, y espera hasta que se vuelva a activar la señal de habilitación $Tvalid$ de nuevo, para volver a enviar datos al banco de memorias de doble puerto.

Dentro del banco de memorias existen dos memorias que funcionan en modo ping-pong, que permite que mientras una está siendo leída, se pueda estar realizando una acción de escritura en la otra, y viceversa. Cada una de las memorias tiene el tamaño para guardar un paquete de transmisión completo. Con estas características, el banco de memorias consigue independizar el reloj de funcionamiento del sistema global f_{clk} del reloj de funcionamiento del periférico transmisor f_{per} ; además de asegurar que los datos proporcionados por el DMA se entreguen de la manera correcta al transmisor.

Por último, el puerto B del banco de memorias se encarga de generar la dirección y la señal de habilitación de lectura ($fifo_rd$), sincronizada con la señal de habilitación CE_M . La señal CE_M es la señal de habilitación de los datos de entrada al transmisor, y depende del número M de subcanales de entrada. Cuando un buffer ha sido leído activa la señal Rd_pkt , que a su vez activa la señal $empty$, que indica que los bancos de memoria están vacíos, y por tanto se puede recibir nuevos paquetes del DMA.

3.3 Transmultiplexor como Periférico Avanzado

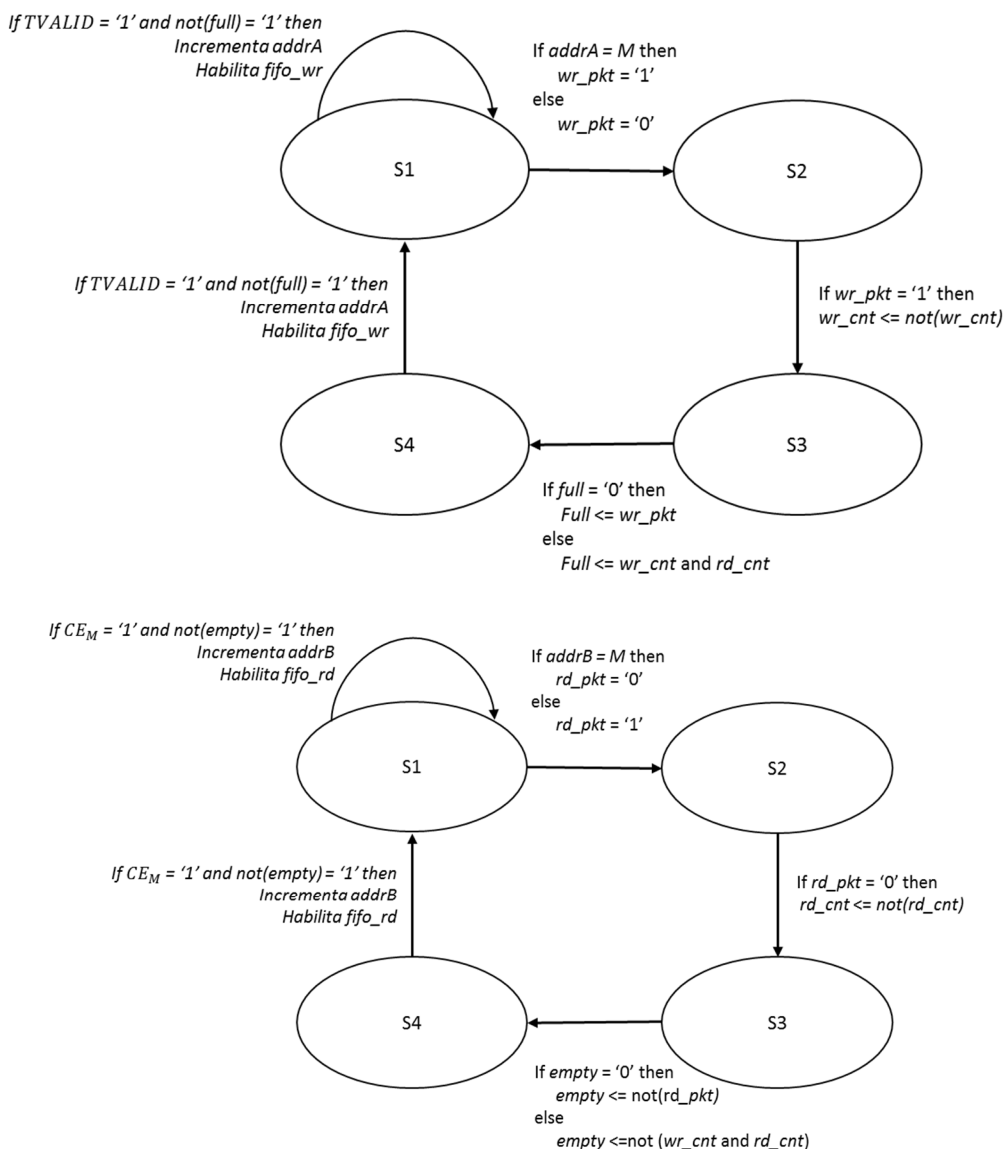


Figura 18. Diagrama de la máquina de estados de la interfaz del transmisor.

Como se observa, en función de la señal de habilitación CE_M , se produce la entrega de los datos a la entrada del transmisor. Con esto se consigue que el DMA proporcione los datos al ritmo que demanda el transmultiplexor, asegurando, además, que no se produce la pérdida de los datos que se desean transmitir, gracias al empleo de dos memorias en modo ping-pong, ya que este modo, permite escribir en una memoria, mientras se están realizando accesos de lectura en la otra.

En el Figura 19 se pueden observar todas las señales que aparecen en la Figura 18. Con este cronograma se pretende mostrar la evolución de las señales internas de la interfaz de transmisión. Se puede observar cómo se asegura la transferencia de todos los datos al transmisor, ya que está sincronizado con la señal de habilitación de datos CE_M .

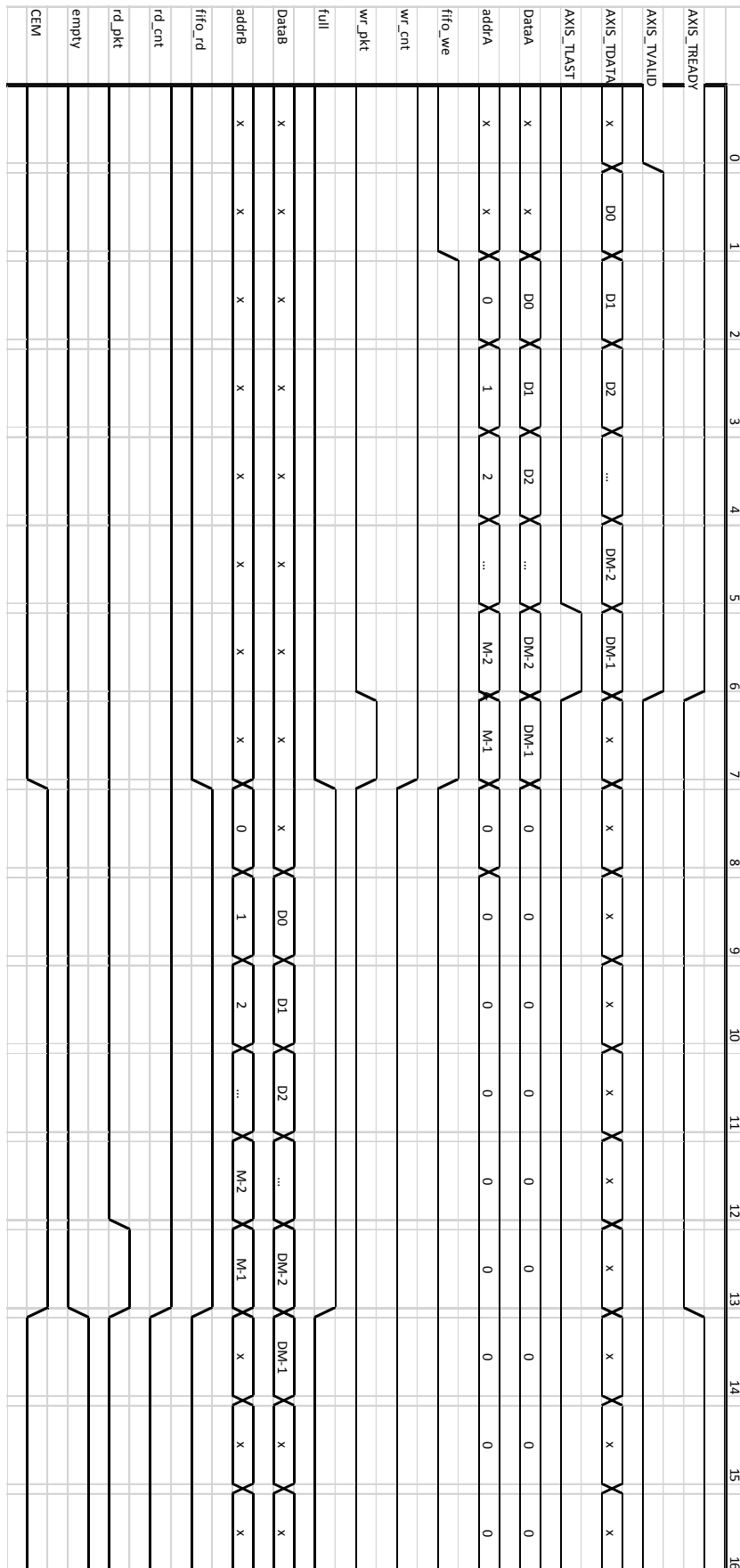


Figura 19. Cronograma de la interfaz de transmisión.

3.3.2. Módulo interfaz de recepción

La interfaz de recepción es la encargada de entregar al bus AXI-Stream los datos que recibe el receptor del transmultiplexor. Al igual que ocurría en el transmisor, el módulo DMA es capaz de recibir una tasa de datos mayor de la que puede proporcionar el receptor, por lo que la interfaz debe indicar al módulo DMA cuándo va a recibir los datos y, a su vez, proporcionárselos adecuadamente. De este modo, la realización concreta de la técnica de acceso al medio considerada vuelve a ser la que impone las limitaciones en las tasas de datos en las transferencias.

Otra de las características que tiene la interfaz de recepción es la de independizar nuevamente el reloj del transmultiplexor de recepción de la del sistema global. Para ello, se emplea una memoria FIFO asíncrona que permite esta independencia. Y por último, la interfaz de recepción también ha de ser capaz de indicar cuándo se ha enviado el último dato, activando la señal *Tlast* del bus AXI-Stream. Por tanto, los módulos que componen la interfaz de recepción son una FIFO asíncrona y una máquina de estados. En la Figura 20 se puede observar el diagrama de bloques de la interfaz de recepción.

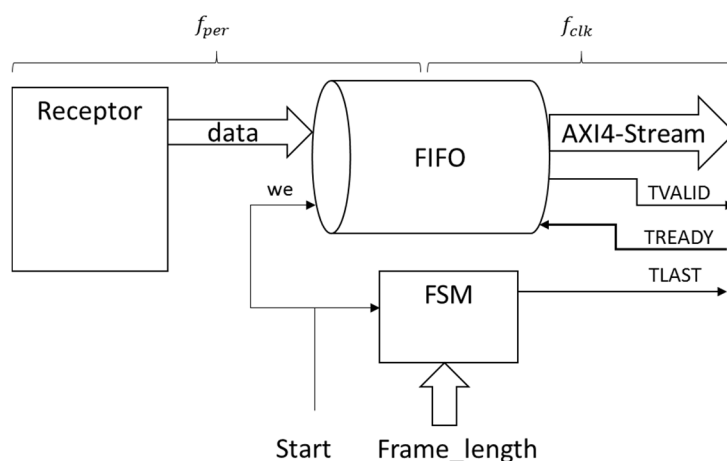


Figura 20. Diagrama de bloques de la interfaz de recepción.

Como se puede observar, la interfaz de recepción es más sencilla que la de transmisión. Si bien, en esta interfaz se introduce una serie de parámetros a considerar para transferir correctamente los datos, que son: la señal de *start* y el número de paquetes que se van a transferir (*frame_length*). Debido a que no se contempla por el momento un módulo de sincronismo, es necesario emplear estas dos señales para calcular cuándo se ha terminado de recibir los paquetes de datos transmitidos.

Al igual que el transmisor, el receptor posee una tasa de datos de salida en cada subcanal de $T_{ch}=T_{Tx}/M$, donde T_{Tx} es la tasa de datos en el medio de transmisión (o tasa de datos de transmisión), asumiendo que el transmultiplexor divide el ancho de banda disponible en M subcanales independientes. Como sucedía en el transmisor, el módulo DMA es capaz de procesar una tasa de datos mayor que la que proporciona el receptor.

La memoria FIFO es la encargada de proporcionar los datos al bus AXI-Stream. Además, al tratarse de una memoria FIFO asíncrona permite independizar el reloj de funcionamiento del receptor f_{per} del reloj del sistema global f_{clk} . Para almacenar los datos, la FIFO sólo necesita la señal we y los datos de salida del receptor. Para entregarlos en el bus AXI-Stream, la memoria FIFO necesita la señal $Tready$ que le proporciona el bus AXI-Stream. Con esta señal, la FIFO genera los datos de salida, junto con la señal de habilitación $Tvalid$.

En la interfaz de recepción, la máquina de estados es la encargada de generar la señal $Tlast$. Para ello, debido a que no existe sincronización entre transmisor y receptor, la máquina de estado necesita la señal de comienzo de transmisión ($start$) y el número de paquetes de datos que se van a recibir. Con estos datos, la máquina de estados es capaz de calcular el momento en el que se debe activar la señal $Tlast$. En la Figura 21 se puede observar el diagrama de la máquina de estados de la interfaz de recepción.

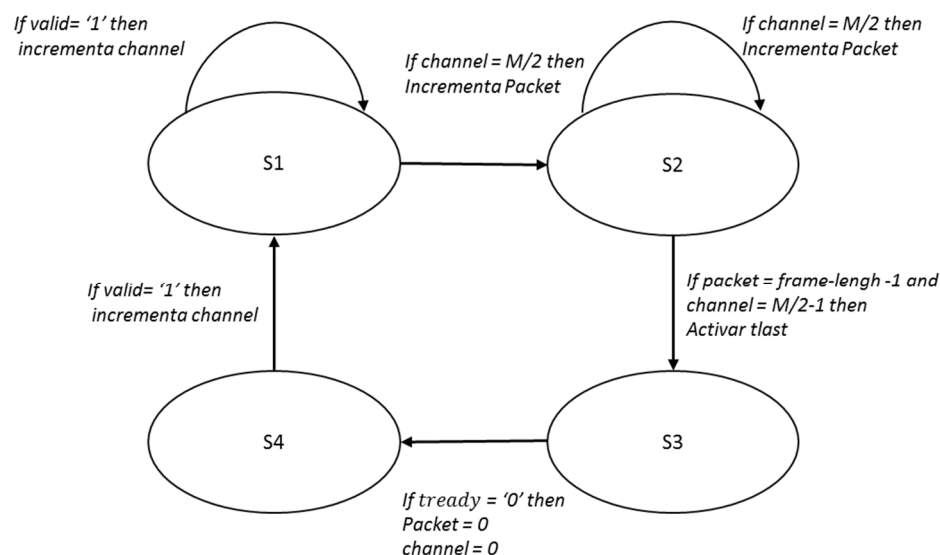


Figura 21. Diagrama de la máquina de estados de la interfaz de recepción.

Como se observa, en función del valor de $frame-length$ se genera la señal $Tlast$, mientras que la señal $start$ coincide con el primer flanco de la señal CE_M del transmisor. Con estas dos señales, se asegura que el receptor y el transmisor estén sincronizados. Esta solución es provisional, y podría eliminarse en aquellas realizaciones del receptor del transmultiplexor que presenten un bloque de sincronismo. Con esta interfaz se consigue que el DMA reciba los datos al ritmo que le marca el transmultiplexor.

3.4 Comunicaciones en la arquitectura propuesta

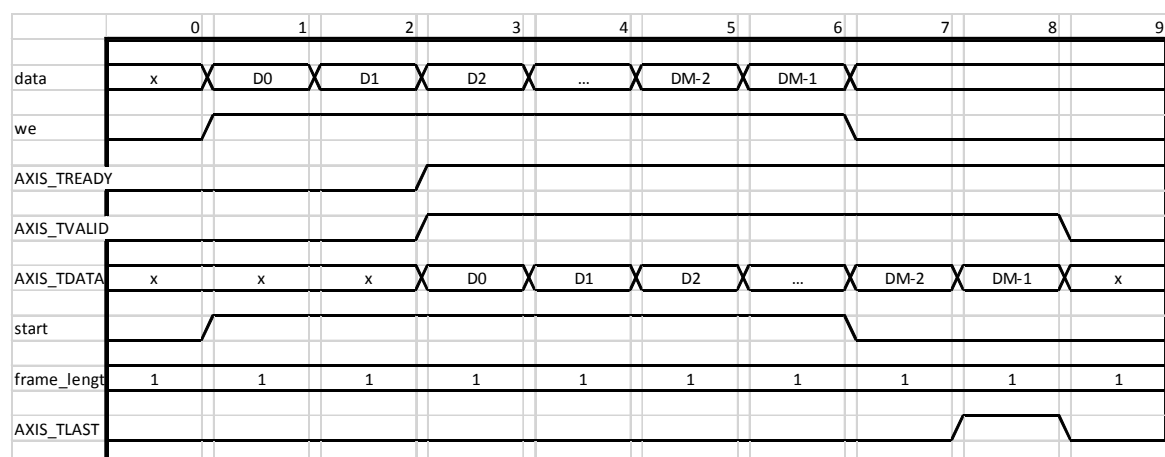


Figura 22. Cronograma de la interfaz de recepción.

En la Figura 22 se pueden observar todas las señales que aparecen en la Figura 20. Con este cronograma se pretende mostrar la evolución de las señales internas de la interfaz de recepción. Se puede observar cómo se asegura la recepción de todos los datos del receptor, ya que éste está sincronizado con la señal de habilitación de datos *start*.

3.4. Comunicaciones en la arquitectura propuesta

Como se ha mostrado, la arquitectura SoC propuesta está basada en un microprocesador soft Microblaze [96], que controla varios periféricos para establecer la transmisión con el transmutiplexor bajo estudio en cada caso. Los periféricos que controla son: un periférico avanzado, en el que se implementa la técnica de acceso al medio que se quiere desarrollar; un módulo DMA, encargado de generar la tasa de flujo de datos que demanda el periférico avanzado; y un controlador de memoria DDR3, para realizar los accesos a memoria externa.

Una transacción DMA consiste en transferir una serie de datos contiguos a través de una única transacción. En cambio, el DMA opera en modo Scatter-Gather [99]; este modo permite que se realicen varios accesos a distintas zonas de memoria no contiguas en una única transacción. Esto permite que no se tenga que reservar grandes espacios físicos de memoria contigua. Aunque esta característica no se aprovecha plenamente en este caso, se ha decidido abordar para futuras mejoras de sistema, en las que incluso se pueda introducir un sistema operativo, para el control de sistemas de comunicación más complejos.

En la Figura 23 se puede observar el diagrama de funcionamiento del modo Scatter-Gather. Como se observa, en la configuración del modo, se crea un anillo de descriptores, que serán los encargados de realizar la transferencia de los datos. Con este anillo, se consigue ir realizando la transacción de los datos ininterrumpidamente, y colocándolos en los distintos bloques para su almacenamiento en memoria.

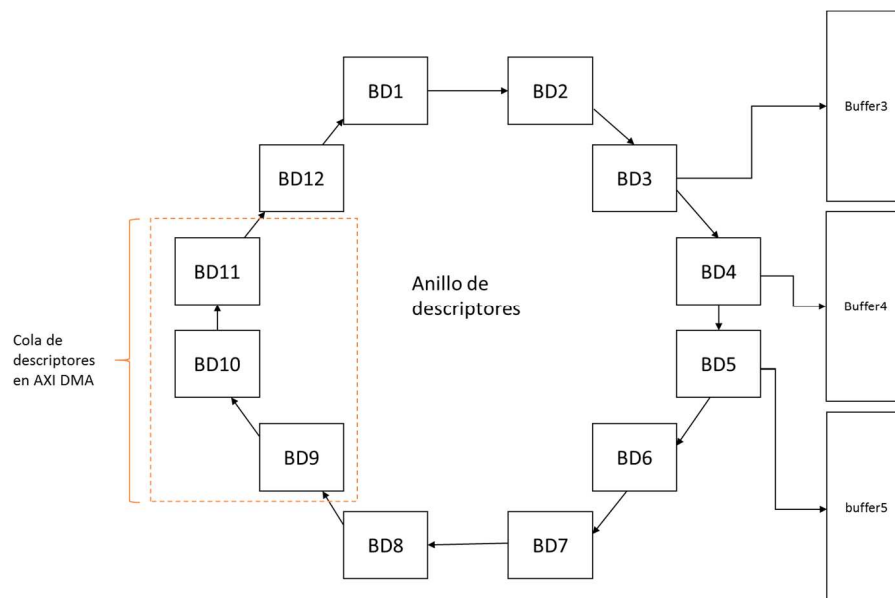


Figura 23. Diagrama de funcionamiento del modo Scatter-Gather.

Como ya se ha mencionado antes, es importante recordar que no existe ningún tipo de alineación o módulo de sincronismo entre el transmisor y el receptor de la técnica de acceso al medio. Actualmente éstos se encuentran sincronizados al emplear el mismo reloj y la misma señal de comienzo, además de un pequeño módulo que estima el número de ciclos de reloj de latencia del canal, si bien, se plantea en trabajos futuros el desarrollo de un módulo de sincronismo que consiga independizar completamente el transmisor del receptor.

El procesador Microblaze realiza la tarea de control de los periféricos. Además, también es capaz de generar la señal a transmitir que se desee, permitiendo de esta manera, poder enviar cualquier tipo de señal y conseguir realizar un estudio exhaustivo del transmultiplexor. Como complemento, para dotar a la arquitectura SoC global de una mayor capacidad de análisis del transmultiplexor implementado, el procesador Microblaze también realiza el cálculo del error medio y la parámetros estadísticos entre la señal transmitida y recibida.

Por último, una vez que se ha mostrado la arquitectura SoC global, se quiere exponer ciertos parámetros que definen el sistema SoC desarrollado. Estos parámetros permitirán conocer el límite para implementar cualquier transmultiplexor, ya que, con estos parámetros, se limita las posibilidades de desarrollo de la arquitectura del transmultiplexor.

Estos parámetros son la frecuencia máxima de funcionamiento, el máximo ancho de palabra y la cantidad de datos que se pueden transmitir, que dependen de la técnica acceso al medio que se haya implementado. Si bien, el único parámetro limitador de la arquitectura global es el ancho de palabra del bus de comunicaciones AXI, el cual limita la tasa de datos T_{max} , como se observa en (1).

$$T_{max} = f_{clk} \cdot DW_{bus} \text{ (Mbits/s)} \quad (1)$$

Donde f_{clk} es la frecuencia del sistema global y DW_{Bus} es el ancho máximo del bus.

También, hay que mencionar el ratio de frecuencias total R_T existente entre la frecuencia del sistema global f_{clk} (procesador, buses y DMA) y la frecuencia de funcionamiento del periférico f_{per} (2).

$$R_T = \frac{f_{clk}}{f_{per}} \quad (2)$$

Para el caso más desfavorable, ambas frecuencias son iguales y el parámetro $R_T = 1$. Por lo tanto, la frecuencia máxima del periférico es $f_{per} = f_{clk}$. Para cualquier caso, dentro de la arquitectura global realizada, el módulo DMA es capaz de proporcionar la tasa de flujo de datos demanda por la arquitectura implementada.

Por otro lado, se puede calcular el número de subcanales N_{ch} del transmultiplexor que pueden ser procesados simultáneamente y en paralelo por la arquitectura en (3).

$$N_{ch} = \frac{DW_{DMA}}{DW_{ch}} \cdot R_T \quad (3)$$

Donde DW_{DMA} es el ancho de palabra que emplea el DMA; y DW_{ch} es el ancho de palabra de cada subcanal del transmultiplexor.

3.5. Conclusiones

En este apartado se ha mostrado el diseño de la arquitectura del SoC propuesta. Se ha expuesto la necesidad de crear un sistema SoC que permita la implementación en tiempo real de transmultiplexores de técnicas de acceso al medio. Además la arquitectura debe ser lo suficientemente flexible para acoplar las diferentes propuestas de técnicas de acceso al medio como periféricos avanzados del sistema. Para ello, se emplea un módulo DMA para asegurar la entrega de flujo de datos demandada por la técnica de acceso al medio.

Para poder emplear el módulo DMA, es necesario el empleo de dos interfaces para asegurar que la transferencia de datos entre DMA y periférico avanzado se realiza correctamente.

Por último se han mostrado los parámetros que muestran la limitación de la arquitectura SoC desarrollada, ya que la implementación de la arquitectura de los transmultiplexores debe respetar estas limitaciones para poder ser integrada como periférico avanzado en la arquitectura realizada.

Capítulo 4

Técnica de acceso al medio basada en la Transformada Trigonométrica Discreta

Dentro de la búsqueda de nuevas técnicas de acceso al medio multi-portadoras que eviten los inconvenientes que posee OFDM, se ha realizado el estudio de la técnica multi-portadora basada en la Transformada Trigonométrica Discreta (DTT). Esta técnica permite una mejor compactación de la energía, lo cual provoca una reducción del PAPR (Peak-to-Average Power Ratio), uno de los inconvenientes más acusados de la técnica OFDM [40]. Un alto PAPR se produce al haber una gran diferencia entre la potencia máxima y la potencia media de la señal, lo cual hace que se incremente el grado de distorsión de intermodulación, provocando un aumento de la tasa de error. Para realizar la transmisión, la DTT divide el ancho de banda en varios subcanales independientes en paralelo y además añade una secuencia al inicio y al final, conocida como extensión simétrica (SE).

La extensión simétrica es una extensión de la propia secuencia que se transmite, haciendo que la convolución lineal realizada se parezca a una convolución circular. Esto hace que, tanto la interferencia inter-símbolo (ISI) entre los sucesivos símbolos transmitidos, como la interferencia inter-portadora (ICI) entre los subcanales de frecuencia en cada bloque, se minimicen.

Para la arquitectura de esta técnica de acceso al medio se ha realizado un estudio sobre distintas aproximaciones posibles, optando desde un diseño totalmente paralelo, a otro totalmente secuencial, o a un diseño intermedio semi-paralelo. Dependiendo de la

arquitectura realizada se conseguirá modificar el número de recursos, o el número de ciclos a emplear. Dado que el número de subcanales será elevado, se ha pensado en trabajar con éstos de manera semi-paralela, esto permite que se procesen sin restricciones todos los canales y a su vez el número de recursos dedicados no se eleve demasiado. Para definir cuántos subcanales son procesados en paralelo, se define un ratio de paralelismo R_p . Este ratio se empleará en los diferentes cálculos para obtener tanto el número de recursos, como la latencia, el tiempo de procesamiento y el throughput de las distintas partes que formaran la arquitectura de la técnica de acceso al medio.

La introducción de un ratio de paralelismo R_p otorga a la arquitectura una flexibilidad que permite poder adaptar el diseño a distintos tipos de restricciones. Este parámetro junto a la elección de distintos números de subcanales de entrada y la frecuencia de funcionamiento del sistema, permite que la arquitectura sea adaptable a distintas configuraciones que se desee considerar.

A su vez, la Transformada Discreta del Coseno elegida es la de tipo DCT-IV [100] [101]. El empleo de la Transformada Discreta del Coseno de tipo IV posee la ventaja, frente a otros tipos, de que tanto la transformada directa como la inversa poseen la misma expresión, lo que simplifica el diseño de la arquitectura. En la Figura 24 y Figura 25 se muestra el diagrama de bloques del transmisor y del receptor respectivamente. Siguiendo el diagrama de bloques del transmisor (Figura 24) se observa como la entrada $X_m[k]$ se conduce a la Transformada Discreta del Coseno de tipo IV (módulo DCT). Más tarde, a la señal de salida $p_m[n]$ del bloque DCT se le añade una extensión simétrica SE, tanto delante como detrás de la trama de datos a enviar en la salida. Después de esto, las $S=M+\alpha+\beta$ salidas resultantes $q_s[n]$ son serializadas para obtener la señal $e[n]$, que será transmitida; donde M es el número de subcanales de la técnica de acceso al medio; α el número de datos de la extensión simétrica anterior; y β el número de datos de la extensión simétrica posterior.

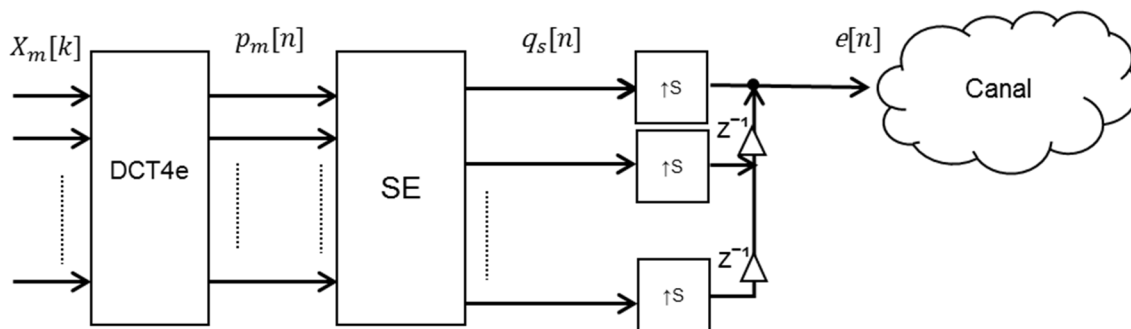


Figura 24. Diagrama de bloques del transmisor DTT.

Por otro lado, en el diagrama de bloques de la Figura 25, la señal de entrada $r[n]$ se paraleliza obteniendo $S=M+\alpha+\beta$ señales $q'_s[n]$; siendo de nuevo M el número de subcanales del receptor; α el número de datos de la extensión simétrica anterior; y β el número de datos de la extensión simétrica posterior. Después, el módulo SE procesa las señales $q'_s[n]$, para obtener $p'_m[n]$. A continuación, la señal resultante $p'_m[n]$ se dirige al módulo DCT. Finalmente, se procesa para obtener la señal final de recepción $X'_m[k]$.

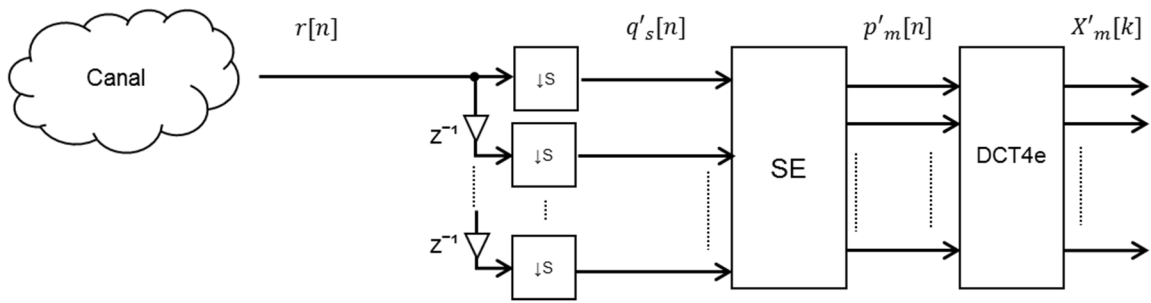


Figura 25. Diagrama de bloques del receptor DTT.

A continuación, se va a realizar un estudio de las distintas partes que componen la arquitectura, abarcando los aspectos teóricos, efectos de la representación en coma fija, consumo de recursos y frecuencias de funcionamiento. Primero se analizará el módulo de la extensión simétrica; después se evaluará teóricamente tres propuestas distintas para la DCT-IV; una vez realizado este estudio teórico, se procederá a mostrar la arquitectura de la técnica basada en Transformadas Trigonométricas Discretas con cada una de estas propuestas; más tarde, se realizará una comparación de las tres propuestas de arquitectura; y por último, se elegirá la mejor de las propuestas para diseñar la arquitectura de la técnica para comunicaciones de banda estrecha.

4.1. Extensión Simétrica

Una vez se han procesado las señales en la Transformada Discreta del Coseno y justo antes de su transmisión, se incorpora la extensión simétrica. Para ello, se añade α datos delante de la trama de datos a transmitir y β datos detrás de esta trama. Estos datos añadidos se corresponden con los primeros α datos o los últimos β datos de la trama de datos que se transmiten. El duplicado de estos datos permite una mejor sincronización de la transmisión basada en DTT. Con esto, en el módulo SE tiene una entrada $p_m[n]$ de M elementos y una salida $q_s[n]$ de $S=M+\alpha+\beta$ elementos, en el caso del transmisor. En el caso del receptor, este tiene una entrada $q'_s[n]$ de $S=M+\alpha+\beta$ elementos y una salida $p'_m[n]$ de M elementos.

La parte duplicada de la trama de datos a transmitir se añade en orden inverso. Además, en el caso de la parte posterior de la extensión simétrica β , los datos son invertidos. A continuación se explica cómo se produce la extensión simétrica anterior a partir de $p_m[n]$, según (4):

$$q_i[n] = p_{\alpha-1-i}[n] \text{ donde } i=0, \dots, \alpha-1. \quad (4)$$

Y en (5) se explica cómo se produce la extensión simétrica posterior a partir de $p_m[n]$:

$$q_{i+\alpha}[n] = -p_{M-1-i}[n] \text{ donde } i=0, \dots, \beta-1. \quad (5)$$

Una vez se ha explicado cómo se produce cada una de las extensiones simétricas, se procede a mostrar cómo se genera la salida $q_s[n]$ a partir de la entrada $p_m[n]$ en el transmisor. La salida del módulo de extensión simétrica en el transmisor se describe como se muestra en (6):

$$\begin{aligned}
 q_i[n] &= p_{\alpha-1-i}[n], & \text{si } 0 < i < \alpha \\
 q_i[n] &= p_{i-\alpha}[n], & \text{si } \alpha \leq i < M + \alpha \\
 q_i[n] &= -p_{M-1-i+M+\alpha}[n], & \text{si } i \geq M + \alpha
 \end{aligned} \tag{6}$$

Donde $i = 0, \dots, M+\alpha+\beta$.

En el caso del módulo de la extensión simétrica en el receptor, lo único que se ha de realizar es quitar esta extensión simétrica añadida, por lo que la salida $p'_i[n]$ del módulo SE en el receptor se genera a partir de la entrada $q'_s[n]$, como se muestra en (7):

$$p'_i[n] = q'_{i-\alpha}[n] \quad \text{donde } i = \alpha, \dots, S-\beta-1. \tag{7}$$

4.1.1. Efecto de la precisión finita

Como se ha observado anteriormente, no se produce ningún tipo de operación matemática en los datos, salvo en (5). Aquí se ha de realizar una inversión de los datos, sin embargo, se ha comprobado que esta inversión no añade ningún tipo de error debido a la cuantificación.

4.1.2. Consumo de recursos

Como se ha mencionado, se ha realizado una inversión de los datos en el transmisor. Para ello, se ha empleado un multiplicador, realizando la operación producto de la señal por la constante -1. Esta operación se podría realizar de distintas maneras, pero el hecho de emplear una multiplicación es la opción más rápida y que menos latencia genera. Por lo tanto, el número $C_m^{SE,Tx}$ de multiplicadores empleados en este módulo es (8):

$$C_m^{SE,Tx} = 1 \text{ multiplicador} \tag{8}$$

En el caso del receptor, el número $C_m^{SE,Rx}$ de multiplicadores empleados es cero, debido a que no es necesario realizar ningún tipo de operación aritmética.

4.1.3. Latencia

Aunque la latencia no tiene impacto en el diseño, se han realizado los cálculos para obtener una mejor definición del sistema. La latencia se ve incrementada respecto a la que añade la DCT, puesto que para conseguir añadir la extensión simétrica es necesario tener previamente almacenado toda la trama de transmisión. Con esto, la latencia $L^{SE,Tx}$ de este módulo en el transmisor es (9):

$$L^{SE,Tx} = M + L^m \tag{9}$$

Siendo $L^m=2$ ciclos de reloj, debido a la multiplicación.

En el caso del receptor, al igual que sucede en el transmisor, se ha de tener almacenado previamente toda la trama de datos, por lo que la latencia $L^{SE,Rx}$ que se obtiene es (10):

$$L^{SE,Rx} = M \quad (10)$$

4.2. Algoritmo 1 para la realización de la Transformada Discreta del Coseno

Para la realización del módulo DCT se va a realizar un estudio en el que se van a desarrollar tres alternativas: El algoritmo 1, que fue propuesto en el artículo [102], trata de realizar una DCT a partir de una FFT de los mismos puntos. El algoritmo 2, el cual trata de reducir el número de puntos de la Transformada Rápida de Fourier (Fast Fourier Transform FFT). El algoritmo 2 ha sido tratado en artículos como [103], donde se ha realizado DCTs de $2N$ -puntos, a partir de FFTs de N -puntos, empleando para ello, propiedades de las señales complejas. Y por último, el algoritmo 3, el cual emplea una etapa de pre-filtrado para reducir el error que genera la cuantificación de la FFT. El algoritmo 3 ha sido empleado para realizar técnicas de compresión de imágenes como se puede observar en [104] [105].

En cada una de las propuestas se realizará un estudio de su algoritmo, el efecto de la precisión finita sobre éste, el consumo de recursos que supone su uso, y el tiempo de procesamiento y la latencia que genera. En este caso se va a analizar la DCT realizada con el algoritmo 1. Como se ha mencionado anteriormente, el módulo a implementar es la DCT (Discrete Cosine Transform) de tipo IV que incluye una FFT [106] [107] [108]. Esta transformada se define por (11):

$$p[n] = \sum_{k=0}^{M-1} X[k] \cos \left[\frac{\pi}{M} \left(n + \frac{1}{2} \right) \left(k + \frac{1}{2} \right) \right] \quad k = 0, \dots, M - 1 \quad (11)$$

Donde $p[n]$ es la salida de la señal de la DCT; M es el número de puntos de entrada; y $X[k]$ es la señal de entrada.

Tomando como referencia (11), el módulo DCT puede dividirse en cuatro fases:

- a) Generación de la secuencia $Y_m[k]$ a partir de la señal de entrada dada $X_m[k]$ multiplicando por la constante $e^{-\frac{jm\pi}{2M}}$, según (12):

$$Y_m[k] = X_m[k] \cdot e^{-\frac{jm\pi}{2M}} \quad (12)$$

- b) Obtención de la señal $y_m[n]$ mediante la FFT de $Y_m[k]$ según (13):

$$y_m[n] = F\{Y_m[k]\} \quad (13)$$

Donde el operador $F\{\}$ realiza una FFT de M puntos.

- c) Reordenación de la señal $y_m[n]$, obteniendo la señal $z_m[n]$ según (14):

$$\begin{aligned} z_{2i+1}[n] &= y_i[n] \\ z_{2i}[n] &= \text{conj} \{y_{m-i}[n]\} \end{aligned} \quad (14)$$

Donde $i = 0, \dots, \frac{M}{2} - 1$; $y_i[n]$ es la salida de la FFT; y el operador $conj\{\}$ es el conjugado de la muestra.

- d) Generación de la secuencia $p_m[n]$ a partir de la señal $z_m[n]$ multiplicando por la constante $e^{-\frac{j\pi(2m+1)}{4M}}$ según (15):

$$p_m[n] = 2 \cdot \frac{1}{\sqrt{2M}} \cdot \text{Re} \left\{ z_m[n] \cdot e^{-\frac{j\pi(2m+1)}{4M}} \right\} \quad (15)$$

Donde $z_m[n]$ es la salida del reordenamiento de la trama; $p_m[n]$ es la salida del módulo DCT; y $\text{Re}\{\}$ es la parte real.

La arquitectura de la DCT del receptor (Figura 25), ya que las transformadas directa e inversa tienen la misma definición en la DCT-IV, es idéntica a la del transmisor. Sin embargo, nótese que la señal de entrada de la DCT del receptor es $p'_m[n]$ y la de salida $X'_m[k]$, en lugar de $X_m[k]$ y $p_m[n]$ respectivamente.

4.2.1. Efecto de la precisión finita del transmisor

A continuación, se presenta el estudio del efecto de la representación en precisión finita (cuantificación) realizado en la DCT de tipo IV con el algoritmo 1. Dado que se va a emplear la tarjeta ML605 de Xilinx para la implementación del diseño en las pruebas experimentales y ésta posee una FPGA Virtex 6 xc6vlx240t, se van a utilizar las celdas multiplicadoras DSP48E1 [109]. Éstas son proporcionadas por la arquitectura de esta FPGA y disponen de una entrada de 18 bits y otra de 25 bits. Como estos tamaños van a ser empleados para realizar las operaciones de multiplicación, se ha decidido emplear una cuantificación de 18 bits para las señales que se desean transmitir, variando el número de bits de la parte fraccionaria dependiendo de los valores máximos de cada zona. Se emplea la entrada de 25 bits para los coeficientes, intentando de esta manera el empleo de un único multiplicador por operación (sin necesidad de extensiones). En este documento, la cuantificación elegida se representa mediante dos enteros $[t f]$, siendo t el número total de bits empleados en la representación numérica, y f el número de bits de la parte fraccionaria. La decisión de emplear la cuantificación de 18 bits para los datos es debida a que, en el core de la FFT empleado, éste es el máximo ancho de palabra para el empleo de un solo multiplicador por operación.

En la Figura 26 se pueden observar los puntos de cuantificación utilizados, para el análisis detallado de los efectos de la precisión finita en el interior de la DCT y su valor, así como las distintas fases descritas anteriormente.

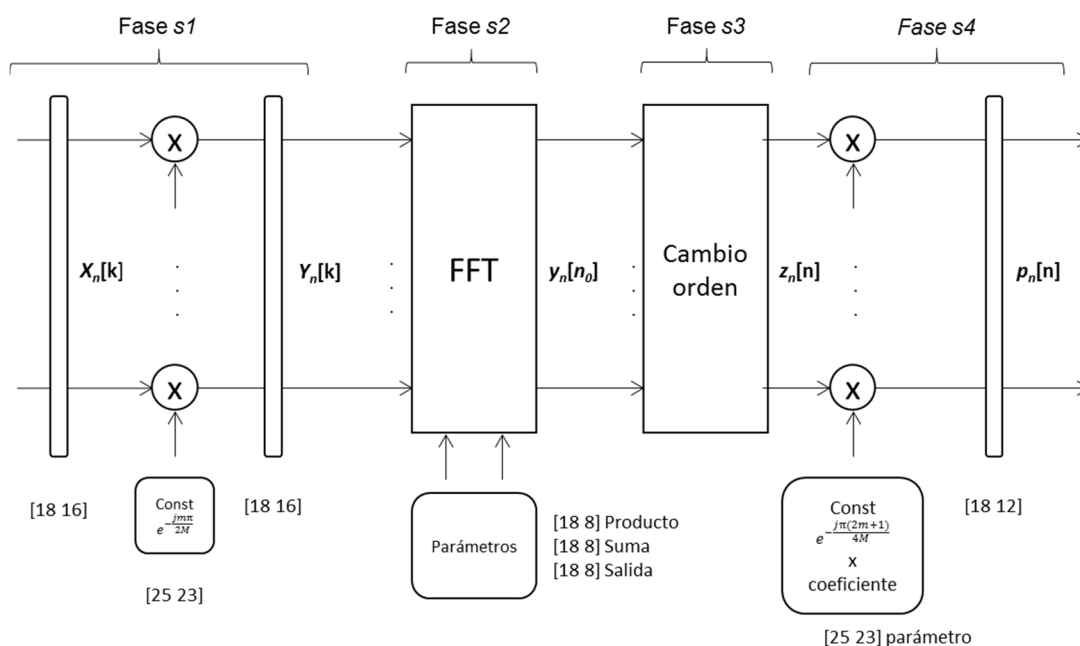


Figura 26. Diagrama de bloques con cuantificación empleada en el módulo DCT con el algoritmo 1.

Como se ha mencionado anteriormente, el módulo DCT está formado por 4 fases, que coinciden con las descritas teóricamente: La primera consiste en una multiplicación por una constante compleja $e^{-\frac{j m \pi}{2M}}$, en la cual, como las celdas multiplicadoras son de 18x25 bits, la cuantificación en las constantes es de 25 bits de ancho de palabra. Para realizar la multiplicación compleja la parte real e imaginaria, tanto de los coeficientes como de los datos, se consideran independientes, disponiendo cada una de ellas del ancho de palabra fijado, lo cual favorece obtener una mayor precisión en los resultados. Después se encuentra la FFT, en este caso se emplea una cuantificación de 18 bits en todo su diseño. En la fase s3, se encuentra el cambio de orden, y por tanto, en esta fase no se produce ninguna operación, simplemente un reordenamiento de la trama de datos. Por último, se encuentra la fase s4 en la cual se realiza una multiplicación por una constante compleja $e^{-\frac{j \pi (2m+1)}{4M}}$ y un coeficiente $\frac{1}{\sqrt{2M}}$. Al igual que como sucedía en la fase s1, los coeficientes tienen 25 bits de ancho de palabra.

En la Figura 27 se muestra el error obtenido en una trama de salida de la DCT-IV del transmisor, para una entrada aleatoria entre -1 y 1 que es el máximo rango de entrada. En la figura se puede observar el error obtenido en cada subportadora, correspondiente a cada subcanal de los $M=512$ disponibles.

4.2 Algoritmo 1 para la realización de la Transformada Discreta del Coseno

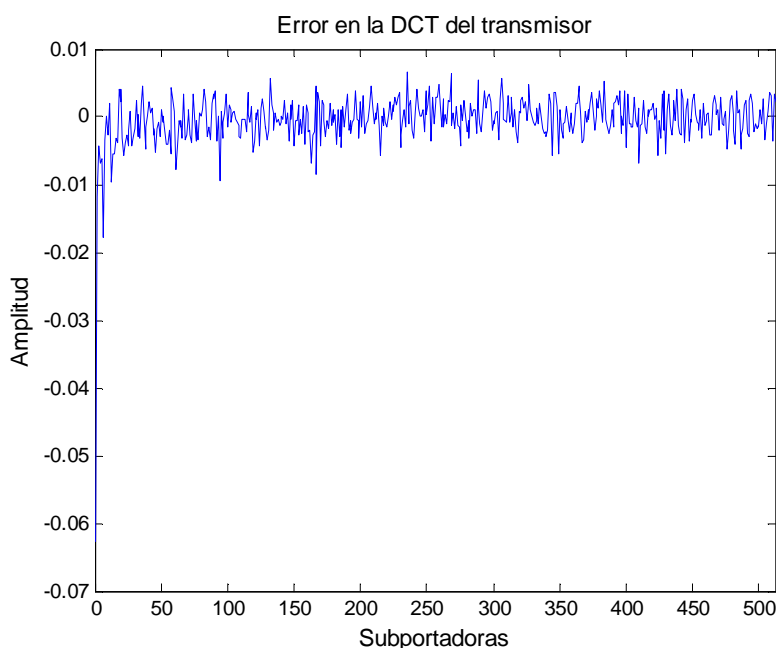


Figura 27. Error obtenido para una trama de datos en la DCT del algoritmo 1 del transmisor para una entrada aleatoria entre -1 y 1.

Como se observa, en las primeras subportadoras el error es mayor. Este aumento es debido a que el efecto de la precisión finita en la FFT es mayor en las primeras subportadoras.

A continuación en la Tabla 1 se van a mostrar los errores en puntos intermedios de la arquitectura cometidos en la DCT del algoritmo 1, debidos a la cuantificación. Los resultados obtenidos son el error parcial que introduce cada parte. Para su obtención se ha utilizado la cuantificación en cada punto, mientras que en el resto se encontraban sin cuantificación. Para obtener los valores se han realizado 1000 simulaciones, obteniéndose el error medio, error máximo y desviación típica.

Tabla 1. Error medio, máximo y desviación típica en los puntos intermedios de la DCT del algoritmo 1.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Q1	[25 23]	$9.0292 \cdot 10^{-8}$	$1.6257 \cdot 10^{-7}$	$3.1723 \cdot 10^{-8}$
Q2	[18 16]	$1.1553 \cdot 10^{-5}$	$2.1268 \cdot 10^{-5}$	$4.1571 \cdot 10^{-6}$
Qfft	[18 8]	$6.3600 \cdot 10^{-2}$	$1.4037 \cdot 10^0$	$7.7100 \cdot 10^{-2}$
Q4	[18 16]	0	0	0
Q5	[25 23]	$9.0496 \cdot 10^{-8}$	$1.6555 \cdot 10^{-7}$	$3.4069 \cdot 10^{-8}$
Q6	[18 12]	$1.2195 \cdot 10^{-4}$	$2.4340 \cdot 10^{-4}$	$7.1295 \cdot 10^{-5}$

Donde Q1 es la cuantificación del primer coeficiente $e^{-\frac{jm\pi}{2M}}$; Q2 es la salida de la multiplicación por el primer coeficiente $Y_m[k]$; Qfft es la salida de la FFT cuantificada; Q4 es la salida del reordenamiento de la trama de datos $z_m[n]$; Q5 es la cuantificación del segundo coeficiente $e^{-\frac{j\pi(2m+1)}{4M}}$; y Q6 es la salida de la multiplicación por el segundo coeficiente $p_m[n]$.

Como se puede observar, la FFT es el punto en el que mayor error introduce la cuantificación, debido a que hay que reservar bits suficientes para la parte entera de su

salida. Este inconveniente determina en gran medida la calidad de la transmisión, al reducir la precisión del transmisor.

Después de explicar los errores intermedios debidos a la cuantificación, se procede a mostrar el error medio, máximo y desviación típica a la salida global de la DCT del algoritmo 1. El cálculo de estos errores se ha realizado introduciendo una entrada aleatoria entre $[-1,+1]$, realizando 1000 simulaciones. Los resultados se observan a continuación en la Tabla 2. En ella se puede observar cómo se produce una acumulación de los errores de los puntos intermedios mostrados anteriormente.

Tabla 2. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 1.

Salida DCT		Error absoluto medio	Error absoluto máximo	Desviación típica
[18 12]		0.0023	0.0627	0.0033

Aun con el inconveniente de que la FFT limita la cuantificación empleada en la DCT del transmisor, se observa que es posible conseguir un error absoluto medio bajo. Este valor se comparará más adelante con los valores obtenidos en los demás algoritmos.

4.2.2. Efecto de la precisión finita del receptor

En la Figura 28 se muestra el error obtenido para una trama de datos en la DCT del algoritmo 1 del receptor para una entrada aleatoria entre -1 y 1. Esta DCT difiere de la del transmisor en la cuantificación elegida para cada punto intermedio, por lo que, los resultados obtenidos serán distintos.

Se puede observar, como el error introducido por la DCT del receptor es mucho menor que la del transmisor. Esto es debido al empleo de una cuantificación distinta, que permite mayor precisión en el receptor.

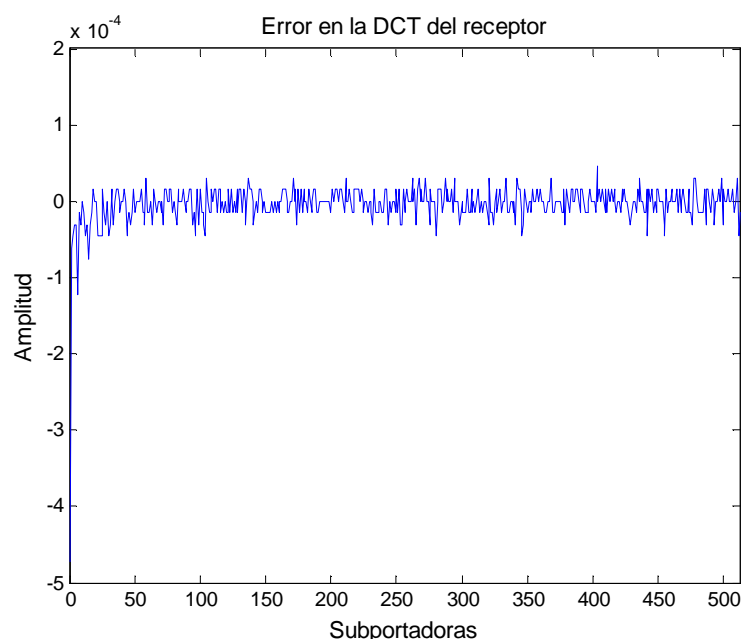


Figura 28. Error en la DCT del algoritmo 1 del receptor para una trama de datos con una entrada aleatoria de -1 y 1.

4.3 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 1

A continuación en la Tabla 3 se van a mostrar los errores en los puntos intermedios de la arquitectura cometidos en la DCT del algoritmo 1 en el receptor, debidos a la cuantificación. Para obtener los valores se han realizado 1000 simulaciones, calculando el error medio, error máximo y desviación típica.

Tabla 3. Error medio, máximo y desviación típica en cada punto intermedio en la DCT del algoritmo 1 del receptor.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Q1	[25 23]	$9.0292 \cdot 10^{-8}$	$1.6257 \cdot 10^{-7}$	$3.1723 \cdot 10^{-8}$
Q2	[18 16]	$1.2092 \cdot 10^{-5}$	$2.1246 \cdot 10^{-5}$	$4.3146 \cdot 10^{-6}$
Qfft	[18 16]	$3.7435 \cdot 10^{-4}$	$5.7 \cdot 10^{-3}$	$4.6270 \cdot 10^{-4}$
Q4	[18 16]	0	0	0
Q5	[25 23]	$9.0496 \cdot 10^{-8}$	$1.6555 \cdot 10^{-7}$	$3.4069 \cdot 10^{-8}$
Q6	[18 16]	$7.7131 \cdot 10^{-6}$	$1.5234 \cdot 10^{-5}$	$4.3278 \cdot 10^{-6}$

Donde Q1 es la cuantificación del primer coeficiente $e^{-\frac{j\pi n}{2M}}$; Q2 es la salida de la multiplicación por el primer coeficiente $y'_m[n]$; Qfft es la salida de la FFT cuantificada; Q4 es la salida del reordenamiento de la trama de datos $Z'_m[k]$; Q5 es la cuantificación del segundo coeficiente $e^{-\frac{j\pi(2m+1)}{4M}}$; y Q6 es la salida $X'_m[k]$ de la multiplicación por el segundo coeficiente.

Como se observa, la FFT sigue siendo el elemento que mayor error introduce, si bien, en este caso el error se ha reducido. Esta reducción se debe a que la FFT del receptor permite una cuantificación con un menor número de bits reservados para la parte entera. Gracias a estos, se consigue una mayor precisión en el receptor.

En la Tabla 4 se muestra el error medio, máximo y desviación típica a la salida global de la DCT del algoritmo 1, para una entrada aleatoria entre [-1,+1]. Para su obtención se han realizado 1000 simulaciones. Este resultado acumula el error parcial de cada punto intermedio mostrado antes.

Tabla 4. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 1 del receptor.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Salida	[18 16]	$1.4216 \cdot 10^{-5}$	$4.7302 \cdot 10^{-4}$	$2.4604 \cdot 10^{-5}$

El error que se obtiene a la salida de la DCT del receptor es menor que el obtenido en el transmisor. Esto se produce por el empleo de una cuantificación distinta en los puntos intermedios, lo que favorece una mejor precisión del receptor.

4.3. Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 1

Una vez se han estudiado todos los módulos que componen la Transformada Trigonométrica Discreta, se van a fijar los parámetros de diseño con los que se va a realizar la arquitectura, así como a establecer el ratio de paralelismo R_p y el ratio de frecuencias R_f que se va a emplear. La FPGA en la que se va a realizar la implementación es una Virtex 6 xc6vlx240t [110]. Esta FPGA dispone de $M_c^{FPGA}=768$ celdas multiplicadoras DSP48E1.

Teniendo en cuenta los diagramas de bloques generales del transmisor y receptor basado en DTT que se muestran en la Figura 24 y Figura 25 para el algoritmo 1, respectivamente, el correspondiente diseño se ha implementado considerando tres parámetros generales. Primero, se analizó el consumo de recursos, especialmente las celdas DSP48E1 (multiplicador-acumulador) [109], debido a la gran presencia de esta operación en la propuesta DTT. Segundo, se calculó la latencia, el throughput y el tiempo de procesamiento que hay que considerar para operaciones en tiempo real. Finalmente, se estudió la representación en coma fija y el error de cuantificación asociado.

Se asume una tasa de transmisión de $T_{Tx}=62.5\text{Mpsps}$ en el canal, propuesta en el estándar de PLC [27]. Teniendo en cuenta $M=512$ subcanales de entrada correspondientes cada uno a una subportadora, se considera que la tasa de datos a la entrada y la salida para cada subcanal es de $T_{ch}=T_{Tx}/M=122.07\text{kpsps}$. Estas tasas de datos se convierten en la limitación de frecuencia a tener en cuenta para el diseño de la arquitectura. En este punto, no se considera ningún tipo de latencia, aunque cada muestra de entrada $X_m[k]$ debe ser procesada y transmitida, antes de que la siguiente $X_m[k+1]$ esté disponible a una velocidad de $T_{ch}=122.07\text{kpsps}$.

Con respecto al diseño, se han analizado tres tipos diferentes de arquitectura. La primera, un enfoque paralelo, donde $M=512$ muestras de entrada $X_m[k]$ son procesadas simultáneamente. Esta estrategia logra un procesamiento rápido de los datos, pero el consumo de recursos es demasiado alto y la eficiencia del uso de cada recurso a lo largo del tiempo no está garantizada. El segundo enfoque está basado en un sistema secuencial, donde las $M=512$ muestras de entrada $X_m[k]$ son multiplexadas en el tiempo a través del mismo recurso, con lo que se consigue más eficiencia en el uso del mismo. Esta opción tiene el riesgo de que el throughput y el tiempo de procesamiento final conseguido no permita una implementación en tiempo real, debido a que no se alcance la tasa de transmisión T_{Tx} . Finalmente, un enfoque semiparalelo permite obtener un compromiso entre ambos enfoques: el procesamiento paralelo de datos y la reutilización de los recursos de computación. Para ello, en lo sucesivo, es necesario remarcar la importancia del ratio de paralelismo R_p en el desarrollo final de la arquitectura. El ratio de paralelismo R_p está definido como el número de muestras que son simultáneamente procesadas por la arquitectura y está relacionado con el número de datapaths que existen en la propuesta. El parámetro R_p puede ser modificado, dependiendo de las necesidades que requiera la arquitectura. Valores elevados de R_p implican mayor consumo de recursos, mientras que bajos valores de R_p significan menor uso de recursos, reutilizándolos en el tiempo, lo que implica menor throughput y mayor tiempo de procesamiento. El parámetro R_p debe fijarse de acuerdo con algunos parámetros clave del diseño: la tasa de transmisión T_{Tx} , el número de subcanales M , el throughput de la arquitectura T_c y la frecuencia de reloj de la arquitectura f_{per} . Esta dependencia se describe en (16). Nótese que el throughput T_c viene determinado por el número R_p de datapaths disponibles en la arquitectura.

$$T_c \left[\frac{\text{samples}}{\text{clock cycle}} \right] \cdot f_{per} [\text{Hz}] \geq T_{Tx} \left[\frac{\text{samples}}{\text{second}} \right] \quad (16)$$

4.3 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 1

En este caso, la solución más adecuada es la que cumple con la tasa de transmisión requerida de $T_{Tx}=62.5\text{Msps}$, $T_{ch}=122.07\text{kps}$ por subcanal, mientras que el consumo de recursos se minimice lo máximo posible. Del anterior análisis de la propuesta llevado a cabo en el apartado 1, una solución totalmente paralela con $R_p=M=512$ puede descartarse, ya que implica que se agoten los multiplicadores en los dispositivos FPGA actuales. Por otra parte, el grado de reutilización de estos multiplicadores se reduciría, ya que una tasa de datos $T_{ch}=122.07\text{kps}$ es mucho menor que la frecuencia normal del reloj de estas FPGA (incluso más de 100MHz).

Por otro lado, una aproximación totalmente multiplexada en el tiempo, considerando sólo los recursos para la ejecución del procesamiento asociado a un solo subcanal ($R_p=1$), no consigue un tiempo de procesamiento adecuado, ya que las $M=512$ muestras de entrada $X_m[k]$ no se procesan y transmiten antes de la llegada del próximo bloque de $M=512$ muestras $X_m[k+1]$ a $T_{ch}=122.07\text{kps}$. En este sentido, el ratio de paralelismo se ha fijado en $R_p=2$, que es el valor mínimo para reducir el número de recursos y para lograr el tiempo de procesamiento suficiente para alcanzar la tasa de datos de entrada de $T_{ch}=122.07\text{kps}$ en los $M=512$ subcanales.

El ratio de paralelismo R_p indica cuántos datapath (o líneas de proceso) son realmente implementados en el diseño, para ser luego reutilizadas para procesar los M subcanales. Gracias a esta arquitectura semi-paralela, se consigue una optimización de los recursos, ya que son empleados para procesar varios subcanales a lo largo del tiempo. En la Figura 29 se puede observar el diagrama de bloques general del receptor para un $R_p=2$, siendo el inverso para el transmisor del sistema.

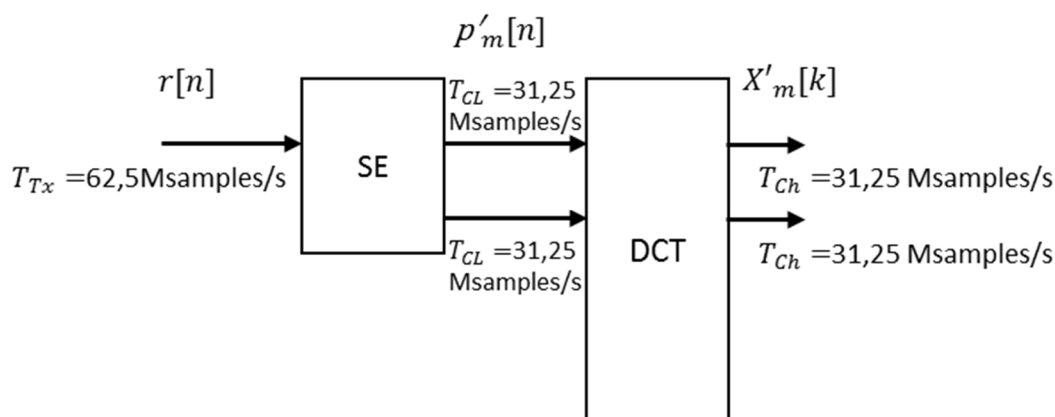


Figura 29. Diagrama de bloques del receptor para $R_p=2$, con la tasa de datos que se emplea en cada punto.

Si la arquitectura del sistema se realizara de una manera completamente paralela, se tendría a la entrada una tasa de datos para cada uno de los $M=512$ datapath desplegados de $T_{ch}=T_{Tx}/M=122.07\text{kps}$, donde T_{ch} es la tasa de datos requerida por subcanal en el transmultiplexor. Sin embargo, debido a que la arquitectura se implementa de una manera semi-paralela con un ratio de paralelismo $R_p=2$, la tasa de datos en cada uno de los dos datapath realmente implementados es $T_{ch}=T_{Tx}/R_p=32.5\text{Msps}$, siendo este valor el mismo en los dos datapaths de salida implementados en el receptor. El número de subcanales sigue

siendo en todo caso, tanto para el transmisor como para el receptor, de $M=512$, independientemente de la arquitectura considerada en el diseño y los correspondientes datapaths desplegados en función del ratio de paralelismo R_p .

Una vez se han fijado los parámetros propios de la arquitectura como pueden ser el ratio de paralelismo, el número de subcanales y la tasa de transmisión, se ha de realizar un estudio de la frecuencia de funcionamiento de la arquitectura. Esta frecuencia de reloj f_{per} es la que se proporcionará a la arquitectura de la técnica de acceso al medio implementada en el periférico avanzado. Dependiendo de la frecuencia que se fije, se modificará el ratio de frecuencia R_f entre la frecuencia del periférico y la frecuencia de muestreo del conversor f_s . Este ratio $R_f = \frac{f_s}{f_{per}}$ influye directamente en el tiempo de procesamiento máximo M_c^{max} de la arquitectura, según la función $M_c^{max}=R_f \cdot M$. Esto significa que dependiendo de la frecuencia del periférico elegida, se modificará el número de ciclos de reloj disponibles para el procesamiento de las señales.

Para la realización de la arquitectura, gracias a que se ha elegido un ratio de paralelismo R_p adecuado, se puede emplear una frecuencia del periférico $f_{per}=62.5\text{MHz}$. Como la frecuencia de muestreo del conversor es de $f_s=62.5\text{MHz}$, se fija el ratio de frecuencia $R_f=1$ y, por tanto, el tiempo de procesamiento máximo en $M_c^{max}=512$ ciclos de reloj.

Por otro lado, conociendo que la plataforma sobre la que se va a trabajar es la ML605 de Xilinx, se conoce el número máximo de multiplicadores disponibles en la FPGA $C_m^{FPGA}=768$. Conocidos estos datos se pueden determinar parámetros como el consumo de recursos, la latencia y el tiempo de procesamiento de la arquitectura de la técnica de acceso al medio.

4.3.1. Consumo de recursos

Como se ha explicado anteriormente, el módulo DCT del algoritmo 1 está formado por 4 fases (véase la Figura 26). A continuación se van a estimar los recursos empleados en cada fase de la DCT. La primera y cuarta fase consisten en una multiplicación por una constante compleja. En la segunda fase se realiza una FFT, para lo cual se propone el uso de un core generado por Spiral [111]. Se ha seleccionado este core, en lugar del que proporciona Xilinx, debido a que este core se ajusta mejor a las necesidades de implementación que se requieren para el diseño de arquitecturas semi-paralelas. El core de Spiral permite modificar el ratio de paralelismo de la arquitectura de la FFT, mientras que el proporcionado por Xilinx sólo permite una arquitectura secuencial. Teniendo en cuenta estos comentarios, se puede establecer el número de multiplicadores C_m y número de sumadores C_s necesarios para el algoritmo 1:

Fase 1 y 4, multiplicación por una constante compleja:

$$C_m^{s1} = C_m^{s4} = 4 \cdot R_p \text{ multiplicadores por cada fase} \quad (17)$$

$$C_s^{s1} = C_s^{s4} = 2 \cdot R_p \text{ sumadores por cada fase} \quad (18)$$

4.3 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 1

Fase 2, realización de un FFT de M puntos:

$$C_m^{s2} = C_m^{FFT} \text{ multiplicadores} \quad (19)$$

$$C_s^{s2} = C_s^{FFT} \text{ sumadores} \quad (20)$$

Donde C_m^{FFT} y C_s^{FFT} son los consumos de recursos de la FFT generada por Spiral.

La fase 3 no necesita ninguna operación matemática, al tratarse de un reordenamiento de los datos de la trama de datos. Sin embargo, si que empleará memorias BRAM para realizar este reordenamiento.

Por lo tanto, el número total de multiplicadores C_m^{DCT} a emplear por la DCT se obtiene al considerar el consumo de recursos parcial de cada fase que la compone, según (21):

$$C_m^{DCT} = C_m^{s1} + C_m^{s2} + C_m^{s4} = 4 \cdot R_p + C_m^{FFT} + 4 \cdot R_p = C_m^{FFT} + 8 \cdot R_p \quad (21)$$

4.3.2. Cálculo de las restricciones de la arquitectura

Se va a realizar un estudio conjunto de la latencia y el tiempo de procesamiento de la DCT del algoritmo 1 y del consumo de multiplicadores que puede emplear la FFT, para obtener las restricciones que impone la implementación de la arquitectura en la FPGA elegida. Para ello se tiene en cuenta la frecuencia de reloj del periférico f_{per} , el número de multiplicadores C_m^{DCT} usados, el ratio de frecuencia R_f entre la frecuencia del periférico y la frecuencia de muestreo del conversor $R_f = f_{per}/f_s$, y el ratio de paralelismo R_p .

Para realizar el cálculo del tiempo de procesamiento y el consumo de multiplicadores se van a plantear dos restricciones: la primera teniendo en cuenta el número máximo de multiplicadores disponibles; y en la segunda el tiempo de procesamiento disponible para realizar las operaciones, teniendo como parámetros el tiempo de procesamiento máximo M_c^{max} en ciclos de reloj, el número C_m^{FFT} de multiplicadores usado por la FFT y el tiempo de procesamiento M_c^{FFT} necesario.

La primera restricción se obtiene del número de multiplicadores C_m^{DCT} (21), con el número máximo de multiplicadores disponibles C_m^{FPGA} , siendo en el caso de las multiplicaciones dependiente del ratio R_p . Para la segunda restricción se ha calculado el tiempo de procesamiento M_c^{DCT} disponible para la DCT, a partir del que se emplea en cada fase, y el tiempo de procesamiento M_c^{max} límite fijado anteriormente.

$$C_m^{DCT} = C_m^{FFT} + 8 \cdot R_p \leq C_m^{FPGA} \quad (22)$$

$$M_c^{DCT} = M_c^{s1} + M_c^{s2} + M_c^{s3} + M_c^{s4} \leq M_c^{max} \quad (23)$$

Despejando en cada ecuación, $M_c^{s1} = M_c^{s4} = 4$ (ciclos de reloj empleados en multiplicación compleja), y $M_c^{s3} = 2$ (ciclos para leer una memoria BRAM), se puede determinar el número máximo de multiplicadores $C_m^{s2} = C_m^{FFT}$ y ciclos de reloj $M_c^{s2} = M_c^{FFT}$ que se pueden utilizar para la FFT empleada en función del valor de R_p .

$$C_m^{FFT} \leq C_m^{FPGA} - 8 \cdot R_p \quad (24)$$

$$M_c^{FFT} = M_c^{s2} \leq M_c^{max} - 4 \cdot 2 - 2 \quad (25)$$

Por otro lado, la latencia L^{DCT} de la DCT es la suma de las latencias de cada fase según, (26):

$$L^{DCT} = L^{s1} + L^{s2} + L^{s3} + L^{s4} \quad (26)$$

Donde $L^{s1}=L^{s4}=8$, son las latencias de la multiplicación compleja; L^{s2} es la latencia de la FFT; y $L^{s3}=M$ es la latencia del cambio de orden. En este punto, se recuerda que la latencia no tiene impacto en el diseño, solamente se emplea para tener una mejor definición de la arquitectura.

4.3.3. Obtención de los parámetros para la FFT y la DCT

Una vez se han fijado los valores que se emplearán en la arquitectura de la técnica de acceso al medio y la plataforma en la cual se desarrollará, y se han obtenido las restricciones de la arquitectura, se procede a realizar el estudio de la FFT. Pese a que se dispone de un rango bastante amplio de multiplicadores, siempre interesará utilizar el menor número de ellos. La FFT se ha estudiado en función del número de datos que se proporcionan en paralelo (R_p), observando el número de multiplicadores, tiempo de procesamiento y throughput. Estos valores han de tenerse en cuenta para el diseño de la arquitectura. Indicar que Spiral [111] proporciona un core cuyo mínimo número de entradas en paralelo es $R_{pmin}^{FFT}=2$, no pudiéndose elegir una FFT totalmente secuencial.

En la Tabla 5 se muestra los datos para cada FFT, y si no cumpliera las restricciones, por qué se descarta.

Tabla 5. Latencia, tiempo de procesamiento (M_c) y multiplicadores empleados para cada tipo de FFT.

GRADO PARALELISMO FFT (R_p)	LATENCIA	M_c	MULTIPLICADORES	DESCARTADA
2	728	256	32	
4	414	128	44	
8	240	64	68	
16	150	32	136	
32	100	16	256	
64	73	8	504	
128	52	4	1048	Sobrepasa máximo de multiplicadores

Atendiendo al ratio de paralelismo R_p , existen varias configuraciones de FFT que se pueden utilizar, primando ante todo el empleo del menor número de multiplicadores posible. En este caso, como se ha mencionado anteriormente, se ha fijado el tiempo de procesamiento $M_c^{max}=512$ ciclos de reloj, siendo la FFT elegida la de $R_p=2$. Con esto se obtiene el número de multiplicadores C_m^{FFT} (27), y el tiempo de procesamiento M_c^{FFT} que necesita para procesar la trama de datos, junto con la latencia L^{FIT} (28):

$$C_m^{FFT} = 32 \text{ multiplicadores} \quad (27)$$

4.3 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 1

$$M_c^{FFT} = 256 \text{ ciclos}, \quad L^{FFT} = 728 \text{ ciclos} \quad (28)$$

Una vez conocido los parámetros de la FFT que se va a utilizar, se emplean sobre (24) y (25), para obtener (29) (30):

$$C_m^{FFT} = 32 \leq C_m^{FPGA} - 8 \cdot R_p^{DCT} = 768 - 16 = 752 \quad (29)$$

$$M_c^{FFT} = M_c^{s2} = 256 \leq M_c^{max} - 4 \cdot 2 + 2 = 502 \quad (30)$$

Como se observa, con la FFT con $R_p^{DCT}=2$, se cumplen los límites especificados, por lo que ésta será la FFT implementada en el diseño.

Por lo tanto, se puede establecer, a su vez, el número de multiplicadores C_m^{DCT} y la latencia L^{DCT} empleados en la DCT. De esta manera, a partir de la formula descrita en (21), se calcula el número de multiplicadores C_m^{DCT} obteniendo el resultado mostrado en (31) y despejando en (26) se calcula la latencia del sistema L^{DCT} , obteniendo (32):

$$C_m^{DCT} = C_m^{s1} + C_m^{s2} + C_m^{s4} = 8 + 32 + 8 = 48 \text{ multiplicadores} \quad (31)$$

$$L^{DCT} = L^{s1} + L^{s2} + L^{s3} + L^{s4} = 8 + 728 + 512 + 8 = 1256 \text{ ciclos} \quad (32)$$

Conociendo $L^{s1}=L^{s4}=8$, la latencia de la multiplicación compleja; y $L^{s3}=M=512$, el periodo completo de lectura para la memoria ping-pong.

4.3.4. Arquitectura del transmisor

A continuación se va a mostrar la arquitectura para el desarrollo de la propuesta DTT, en este caso empleando la DCT del algoritmo 1, junto con el módulo de la extensión simétrica (SE). Para la arquitectura, se han realizado dos modificaciones respecto a lo mostrado en el apartado 4.2, ya que se ha buscado reducir lo máximo posible tanto el consumo de recursos como el número de ciclos. Para ello, se ha decidido que la extensión simétrica y la tercera fase de la DCT, se realicen conjuntamente, con lo que de este modo se puede ahorrar la latencia empleada en la extensión simétrica (9). Para ello, se ha modificado el orden de la fases s3 y s4 de la DCT para, de esta manera, poder unir las memorias empleadas en la fase s3, con las memorias empleadas para realizar el módulo SE, ya que de este modo, se ahorra un número significativo de ciclos de reloj. En la Figura 30 se muestra la arquitectura del módulo DCT, en la que se pueden distinguir las 4 fases que posee.

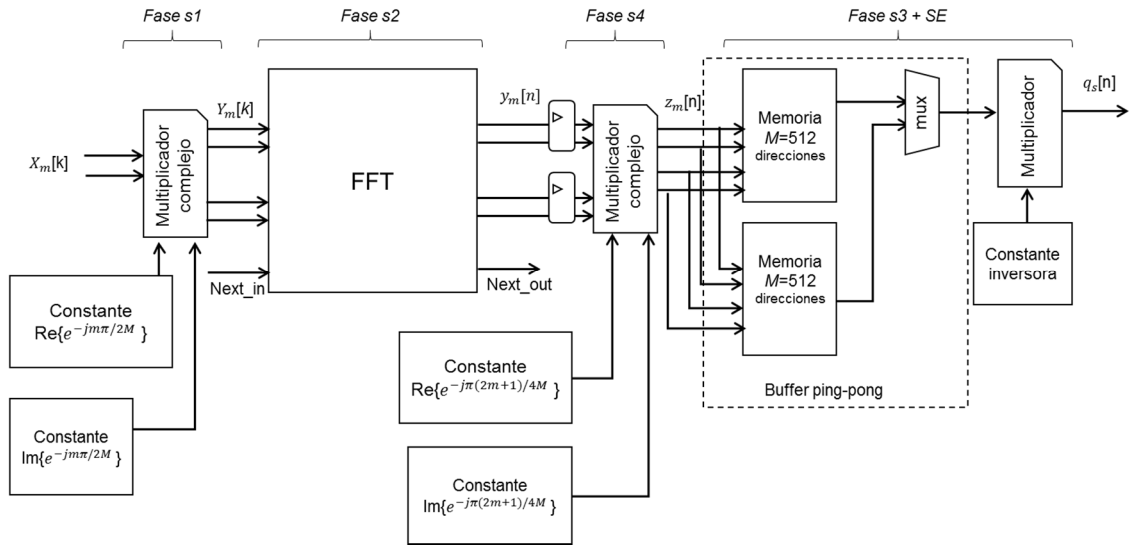


Figura 30. Diagrama de bloques de la arquitectura del transmisor DTT con la DCT del algoritmo 1.

Dado que el ratio de paralelismo se ha fijado en $R_p=2$, $C_m^{s1}=8$ multiplicadores son necesarios en la fase s1 (12). Además, la latencia es de $L^{s1}=8$ ciclos de reloj. Después, la segunda fase s2 es una FFT que requiere, no sólo multiplicadores y sumadores, sino también memoria donde almacenar los valores intermedios (13). Este módulo es parametrizado con un tamaño de $M=512$ muestras y una representación en coma fija de 18 bits, valor fijado debido a que el ancho de palabra de una de las entradas de los multiplicadores de la FPGA es de 18 bits. La FFT presenta una arquitectura radix-2 semi-paralela, con $\log_2 M$ bloques en cascada. De nuevo, la selección del ratio de paralelismo es $R_p=2$, que implica que sólo dos muestras de entrada $Y_m[k]$ son procesadas cada ciclo de reloj. Una FFT se completa cada $M_c^{FFT}=256$ ciclos de reloj, con una latencia $L^{FFT}=728$ ciclos, compatible con el tiempo de procesamiento disponible indicado en las restricciones.

La cuarta fase s4 del módulo DCT es similar a la primera fase s1 (15), donde se usan $C_m^{s4}=8$ multiplicadores y la latencia es de $L^{s4}=8$ ciclos de reloj. Por último, la tercera fase s3 de la DCT es un reordenamiento de la trama de datos (14), donde todas las muestras de la trama $y_m[n]$ deben estar disponibles al mismo tiempo. Por lo tanto, los datos deben ser almacenados en memoria, para que sean leídos en el orden correcto. Para evitar pérdida de datos, se han empleado dos memorias en modo ping-pong, por lo que en una se puede acceder para organizar los datos en el orden correcto, mientras que los nuevos datos se escriben en la otra. Esta fase implica una latencia de $L^{s3}=512$ ciclos.

Como se puede observar, las cuatro fases que forman el módulo DCT están segmentadas. Gracias a que, los datapaths emplean un throughput de $T_{ch}=1$ dato por ciclo de reloj y como el ratio de paralelismo es $R_p=2$, el throughput global de la DCT es $T_c^{DCT}=T_{ch}\cdot R_p=2$. Con todo esto, todas las muestras de entrada $X_m[k]$ son procesadas cada $M_c^{DCT}=256$ ciclos. Como una nueva trama de muestras de entrada $X_m[k]$ está disponible cada $M_c^{max}=512$ ciclos, la arquitectura propuesta cumple la limitación de tiempo debida a este parámetro.

4.3 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 1

Por lo tanto, el número C_m^{Tx} de multiplicaciones empleadas para la realización de la arquitectura del transmisor de la DTT con la DCT del algoritmo 1 es (33):

$$C_m^{Tx} = C_m^{s1} + C_m^{FFT} + C_m^{s4} + C_m^{SE,Tx} = 8 + 32 + 8 + 1 = 49 \text{ multiplicadores} \quad (33)$$

Donde $C_m^{s1}=8$ es el número de multiplicadores en la fase s1; $C_m^{FFT}=32$ es el número de multiplicadores empleados en la FFT; $C_m^{s4}=8$ es el número de multiplicadores empleados en la fase s4; y $C_m^{SE,Tx}=1$ es el número de multiplicadores empleados en el módulo SE.

La latencia L^{Tx} de la arquitectura propuesta para la DCT del algoritmo 1 y el módulo de la extensión simétrica se obtiene en (34):

$$L^{Tx} = L^{s1} + L^{s2} + L^{SE,Tx} + L^{s4} = 8 + 728 + 514 + 8 = 1258 \text{ ciclos} \quad (34)$$

Donde $L^{s1}=8$ es la latencia de la fase s1 (12); $L^{s2}=728$ es la latencia de la FFT (13); $L^{SE,Tx}=514$ es la latencia del módulo SE (9), que contiene también a la fase s3 (14); y $L^{s4}=8$ es la latencia de la fase s4 (15).

Una vez se ha explicado la arquitectura realizada, se procede a mostrar un cronograma del funcionamiento de la DCT del algoritmo 1 y el módulo de la extensión simétrica, que se puede observar en la Figura 31. En este cronograma se observan las principales señales de la propuesta y gráficamente los retardos internos.

Primero, se recibe la señal $X_m[k]$, sincronizada con la señal CE_M , junto con esta señal CE_M se activa el enable y se comienza a generar direcciones para la memoria que almacena los coeficientes $e^{-\frac{jm\pi}{2M}}$ de la primera multiplicación (12) (muestra de reloj 2). A continuación se añade un retardo en $X_m[k]$, para que esté sincronizado con el coeficiente $e^{-\frac{jm\pi}{2M}}$ en la entrada del multiplicador (muestra de reloj 3), produciéndose 4 ciclos de reloj después la salida $Y_m[k]$ del multiplicador (muestra de reloj 7). Esta señal $Y_m[k]$ se retrasa para que esté sincronizada con la señal de comienzo de la FFT $next_in$ (muestra de reloj 9).

Cuando la FFT ha sido realizada (13), se activa la señal $next_out$ y a continuación, comienza a proporcionarse la salida $y_m[n]$ de la FFT (muestra de reloj 13). Esta señal se sincroniza con los coeficientes $e^{\frac{j\pi(2m+1)}{4M}}$ de la segunda multiplicación (15) (muestra de reloj 15). La salida $z_m[n]$ se produce a la vez que las direcciones para el puerto A de la memoria ping-pong que realiza el reordenamiento de la trama de datos (14) (muestra de reloj 20). La salida $p_m[n]$ se produce un ciclo después de la dirección del puerto B de la memoria ping-pong (muestra de reloj 22). Por último esta salida, se multiplica por el coeficiente -1 para generar la extensión simétrica SE y, por tanto, la salida $q_s[n]$ (muestra de reloj 25).



Figura 31. Cronograma de la arquitectura del transmisor DTT con la DCT del algoritmo 1.

4.3.5. Arquitectura del receptor

En el caso de la arquitectura del receptor, no se pueden realizar las mismas adaptaciones de sistema para ahorrar recursos y latencia en la arquitectura, debido al orden inverso de aparición de los módulos. Por lo tanto, la unión que se producía entre la fase s3 de la DCT y el módulo de la extensión simétrica, no se puede realizar, aumentando la latencia final. En la Figura 32 se muestra la arquitectura realizada para el receptor DTT.

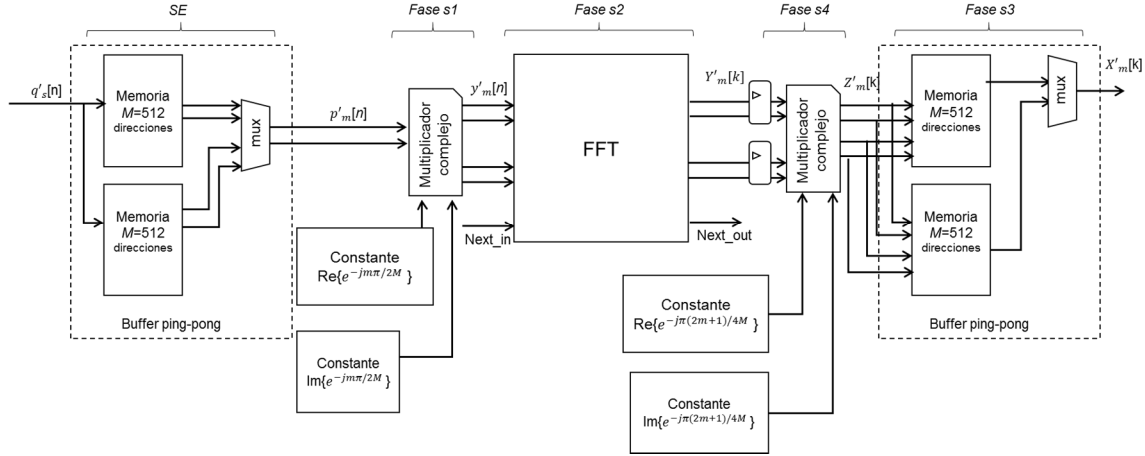


Figura 32. Diagrama de bloques de la arquitectura del receptor DTT con la DCT del algoritmo 1.

En el caso del receptor, al igual que en el del transmisor, el ratio de paralelismo se ha fijado en $R_p=2$. Debido a que la DCT implementada es ortogonal, sigue el mismo proceso de arquitectura que la DCT del transmisor.

Por lo tanto, el número de multiplicaciones C_m^{Rx} empleadas para la realización de la arquitectura del receptor de la DTT con la DCT del algoritmo 1 es (35):

$$C_m^{Rx} = C_m^{s1} + C_m^{FFT} + C_m^{s4} + C_m^{SE,Rx} = 8 + 32 + 8 + 0 = 48 \text{ multiplicadores} \quad (35)$$

Donde $C_m^{s1}=8$ es el número de multiplicadores en la fase s1 (12); $C_m^{FFT}=32$ es el número de multiplicadores empleados en la FFT (13); $C_m^{s4}=8$ es el número de multiplicadores empleados en la fase s4 (15); y $C_m^{SE,Rx}=0$ es el número de multiplicadores empleados en el módulo SE.

La latencia L^{Rx} de la arquitectura propuesta para la DCT del algoritmo 1 y el módulo de la extensión simétrica se obtiene en (36):

$$L^{Rx} = L^{SE,Rx} + L^{s1} + L^{s2} + L^{s3} + L^{s4} = 512 + 8 + 728 + 512 + 8 = 1768 \text{ ciclos} \quad (36)$$

Donde $L^{SE,Rx}=512$ es la latencia del módulo SE (10); $L^{s1}=8$ es la latencia de la fase s1 (12); $L^{s2}=728$ es la latencia de la FFT (13); $L^{s3}=512$ es la latencia de la fase 3 (14); y $L^{s4}=8$ es la latencia de la fase s4 (15).

Una vez se ha mostrado la arquitectura realizada, se procede a mostrar en la Figura 33 un cronograma del funcionamiento de la DCT y el módulo de la extensión simétrica. En

este cronograma se observan las principales señales de la propuesta y gráficamente los retardos internos.

Primero, se recibe la señal $q'_s[n]$, sincronizada con la señal CE_M (muestra de reloj 2). A partir de esta señal se quita la extensión simétrica, para obtener $p'_m[n]$ (muestra de reloj 3). También, se activa el enable y se comienzan a generar direcciones para la memoria que almacena los coeficientes $e^{-\frac{j m \pi}{2M}}$ de la primera multiplicación (12). A continuación se añade un retardo en $p'_m[n]$, para que esté sincronizado con el coeficiente $e^{-\frac{j m \pi}{2M}}$ en la entrada del multiplicador (muestra de reloj 4), produciéndose 4 ciclos de reloj después la salida del multiplicador $y'_m[n]$ (muestra de reloj 8). Esta señal $y'_m[n]$ se retrasa para que esté sincronizada con la señal de comienzo de la FFT $next_in$ (muestra de reloj 9).

Cuando la FFT (13) ha sido realizada, se activa la señal $next_out$ y a continuación, comienza a proporcionarse la salida $Y'_m[k]$ de la FFT (muestra de reloj 13). Esta señal se sincroniza con los coeficientes $e^{-\frac{j \pi (2m+1)}{4M}}$ de la segunda multiplicación (15) (muestra de reloj 15). La salida $Z'_m[k]$ se sincroniza con las direcciones para el puerto A de la memoria ping-pong que realiza el reordenamiento de la trama de datos (14) (muestra de reloj 20). La salida $X'_m[k]$ se produce un ciclo después de la dirección del puerto B de la memoria ping-pong (muestra de reloj 22).

4.3 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 1

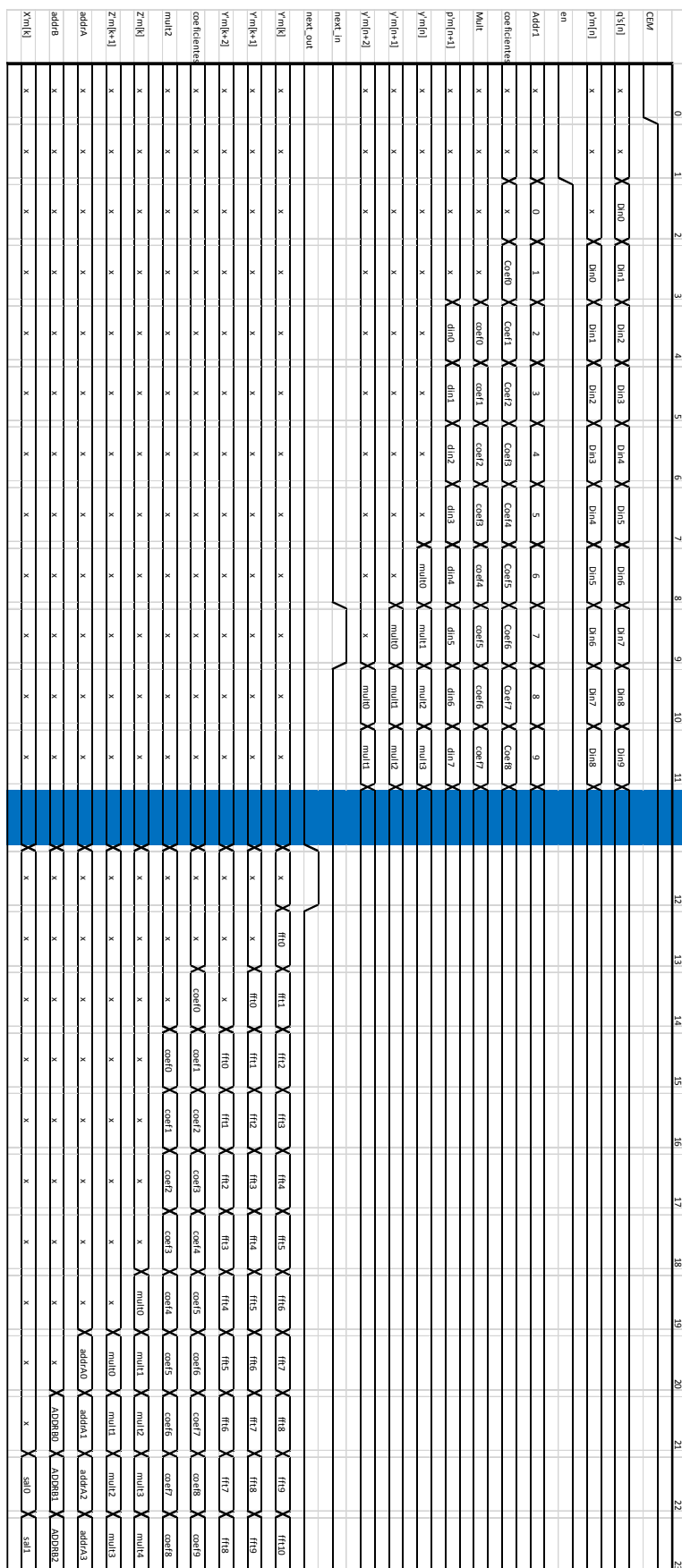


Figura 33. Cronograma de la arquitectura del receptor DIT con la DCT del algoritmo 1.

4.4. Algoritmo 2 para la realización de la Transformada Discreta del Coseno

Uno de los problemas de la DCT del algoritmo 1 es la cuantificación en la FFT, la cual reduce la precisión del transmisor. Por lo tanto, se ha desarrollado un segundo algoritmo tratando de mitigar este efecto, el cual se menciona en [103]. Para la realización de este segundo algoritmo, se ha buscado el empleo de una iFFT en la arquitectura de la DCT, la cual, además de emplear una cuantificación distinta, reduce el tamaño de la iFFT a la mitad de puntos. Esto se consigue empleando señales complejas en la entrada en lugar de señales puramente reales. Con esto se ha intentado reducir, además del efecto de la cuantificación, el consumo de recursos y la latencia de la misma.

En esta propuesta el cálculo de la DCT se realiza en cinco pasos:

- a) Se reordena la señal de entrada $X_m[k]$, obteniendo la señal $W_i[k]$ según el siguiente método (37):

$$Re\{W_i[k]\} = X_{2i}[k], i=0, \dots, M/2-1 \quad (37)$$

$$Im(W_i[k]) = -X_{M-2i}[k], i=0, \dots, M/2-1$$

Donde el operador $Re\{\}$ indica la parte real y el operador $Im\{\}$ indica la parte imaginaria.

- b) Generación de la secuencia $Y_l[k]$ a partir de la señal de entrada dada $W_i[k]$ según (38), donde l es el índice de cada uno de los $M/2$ canales de entrada.

$$Y_l[k] = W_l[k]e^{jl\pi/M} \quad l=0, \dots, M/2-1 \quad (38)$$

- c) Dado un instante de tiempo $n=n_0$, se obtiene la señal $y_l[n_0]$ mediante la iFFT de $M/2$ puntos de $Y_l[k_0]$ según (39).

$$y_l[n_0] = iFFT\{Y_l[k_0]\} \quad (39)$$

Donde el operador $iFFT\{\}$ realiza la FFT inversa de $M/2$ puntos.

- d) Generación de la secuencia $z_l[n]$ a partir de la señal $y_l[n]$ según (40).

$$z_l[n] = y_l[n]e^{\frac{j\pi(l+1/4)n}{M}} \cdot \sqrt{\frac{M}{2}} \quad l=0, \dots, M/2-1 \quad (40)$$

- e) Se reordena la señal de salida $z_l[n]$, obteniendo la señal $p_m[n]$ según el siguiente método (41):

$$p_{2i}[n] = Re\{z_i[n]\}, i=0, \dots, M/2-1 \quad (41)$$

$$p_{M-2i}[n] = Im\{z_i[n]\}, i=0, \dots, M/2-1$$

Donde el operador $Re\{\}$ indica la parte real y el operador $Im\{\}$ indica la parte imaginaria.

4.4.1. Efecto de la precisión finita del transmisor

Al igual que sucedía para la anterior DCT, el objetivo de esta propuesta es su arquitectura en sistemas basados en FPGA. Por lo tanto, es necesario realizar un estudio sobre el efecto de la cuantificación (representación en precisión finita) en la arquitectura, tratando de determinar el número de bits y su configuración más adecuada. Para ello se tendrá en cuenta que la plataforma en la que se desea implementar es la ML605 de Xilinx, que dispone de una FPGA Virtex 6 xc6vlx240t. Esta FPGA dispone de celdas multiplicadoras DSP48E1, las cuales poseen una entrada de 18x25 bits.

En la Figura 34 se puede observar los puntos de cuantificación utilizados en el interior de la DCT y su valor. Se ha decidido emplear una cuantificación de 18 bits para el datapath de los datos, variando el número de bits de la parte fraccionaria dependiendo de los valores máximos de cada zona; y emplear la entrada de 25 bits para los coeficientes.

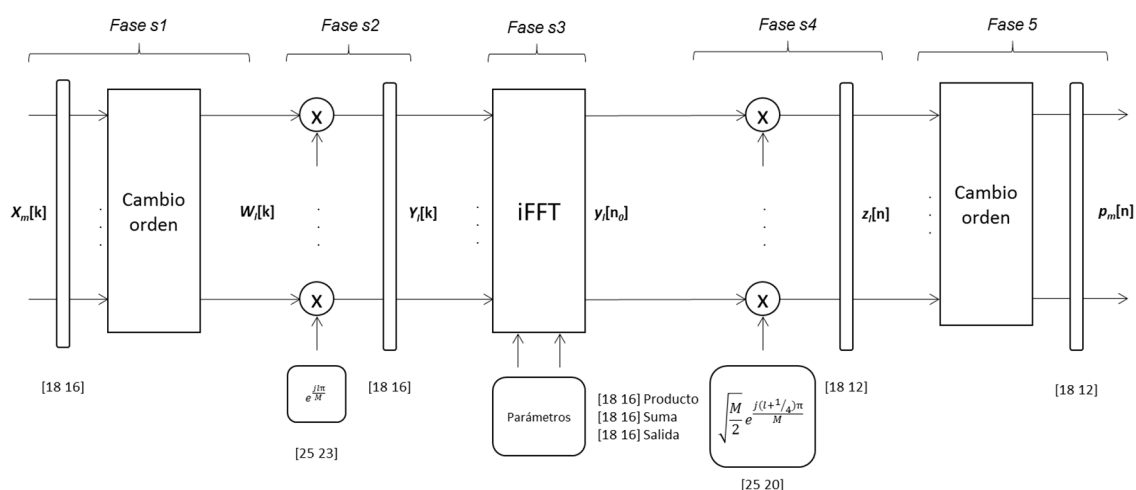


Figura 34. Diagrama de bloques con cuantificación empleada en el módulo DCT del algoritmo 2.

Como se observa en la figura, el módulo DCT está formado por 5 fases, las cuales se corresponde con las fases descritas teóricamente. La primera consiste en un reordenamiento de la trama de datos, en el cual no se producen operaciones matemáticas. La segunda fase s2 realiza una multiplicación compleja, empleando un ancho de palabra de 25 bits para los coeficientes. La tercera fase s3 consisten en la iFFT, en este caso de $M/2$ puntos. En la cuarta fase s4, se realiza otra multiplicación compleja, y al igual que en la fase s2, se emplean 25 bits para los coeficientes. Por último, en la fase s5 se realiza otro cambio de orden para el cual no es necesario realizar ninguna operación matemática.

A modo de ejemplo, en la Figura 35 se muestra el error puntual obtenido para una trama de datos a la salida de la DCT, para una entrada aleatoria entre -1 y 1. La figura representa el error obtenido en cada subcanal de entrada de la DCT.

Como se observa, se ha conseguido reducir el error en las subportadoras iniciales, gracias al empleo del nuevo algoritmo. Esto permite que la precisión del transmisor aumente considerablemente.

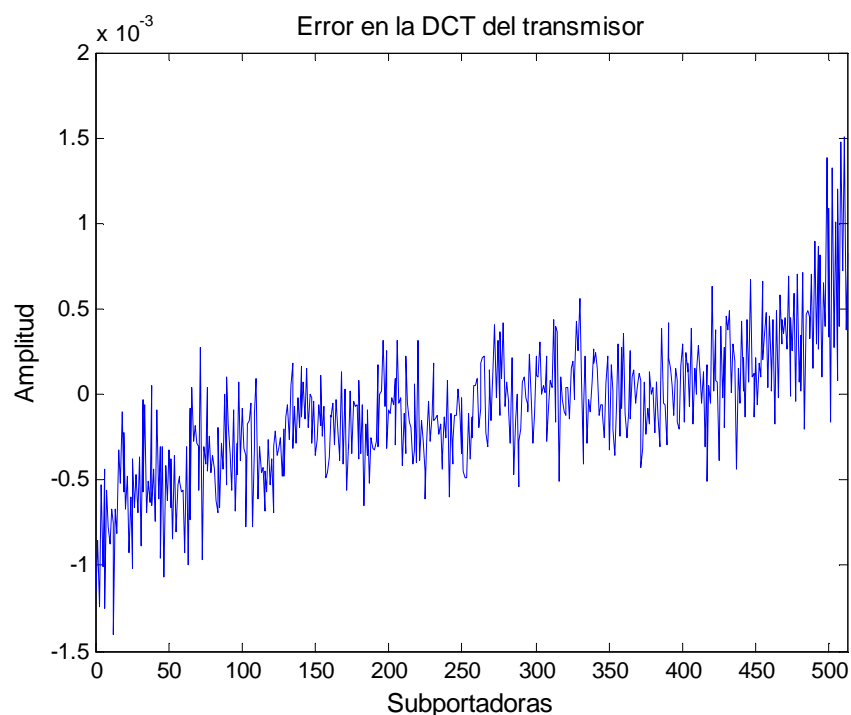


Figura 35. Error obtenido para una trama de datos en la DCT del algoritmo 2 del transmisor con una entrada aleatoria entre -1 y 1.

A continuación en la Tabla 6 se van a mostrar los errores en puntos intermedios de la arquitectura cometidos en la DCT del algoritmo 2, debidos a la cuantificación. En la tabla se muestra el error que introduce cada punto intermedio independientemente.

Tabla 6. Error medio, máximo y desviación típica en cada punto de cuantificación en la DCT del algoritmo 2.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Q1	[25 23]	$8.9611 \cdot 10^{-8}$	$1.6022 \cdot 10^{-7}$	$3.5229 \cdot 10^{-8}$
Q1.2	[25 20]	0	0	0
Q2	[25 23]	$9.3981 \cdot 10^{-8}$	$1.5828 \cdot 10^{-7}$	$3.2882 \cdot 10^{-8}$
Q3	[18 16]	$1.1337 \cdot 10^{-5}$	$2.0865 \cdot 10^{-5}$	$4.5055 \cdot 10^{-6}$
Qifft	[18 16]	$2.8106 \cdot 10^{-5}$	$1.2284 \cdot 10^{-4}$	$2.1728 \cdot 10^{-5}$
Q4	[18 12]	$1.9369 \cdot 10^{-4}$	$3.2186 \cdot 10^{-4}$	$6.6521 \cdot 10^{-5}$
Q5	[18 12]	0	0	0

Donde Q1 es la cuantificación intermedia del segundo coeficiente $e^{\frac{j\pi(l+1/4)}{M}}$ (40); Q1.2 es la cuantificación del segundo coeficiente, después de multiplicar por $\sqrt{\frac{M}{2}}$; Q2 es la cuantificación del primer coeficiente $e^{\frac{j\pi}{M}}$ (38); Q3 es la salida de la multiplicación por el primer coeficiente $Y_l[k]$; Qifft es la salida de la iFFT cuantificada (39); Q4 es la salida de la multiplicación por el segundo coeficiente $z_l[n]$ (40); y Q5 el último cambio de orden de la trama de datos $p_m[n]$ (41).

El objetivo de este algoritmo era la reducción del error de cuantificación de la FFT, el cual ha sido conseguido. En este algoritmo el mayor error lo introduce la multiplicación posterior a la FFT. En esta multiplicación existe un coeficiente bastante grande que hace que se reduzca la precisión final del transmisor.

4.4 Algoritmo 2 para la realización de la Transformada Discreta del Coseno

Después de mostrar los errores en los puntos intermedios de la DCT, se va a mostrar el error medio, máximo y desviación típica a la salida global de la DCT del algoritmo 2, con una entrada aleatoria entre $[-1,+1]$. Para obtener los resultados se han realizado 1000 simulaciones. Estos resultados acumulan los errores de los puntos intermedios observándose el error medio, error máximo y desviación típica en la Tabla 7.

Tabla 7. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 2.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Salida DCT	[18 12]	$3.0470 \cdot 10^{-4}$	$1.5000 \cdot 10^{-3}$	$2.6470 \cdot 10^{-4}$

El error introducido por el algoritmo 2 es menor que el introducido por el algoritmo 1 en el transmisor. Esto se produce gracias a la reducción de la iFFT empleada que reduce a la mitad los puntos empleados.

4.4.2. Efecto de la precisión finita del receptor

En la Figura 36 se muestra el error obtenido para una trama de datos en la DCT del algoritmo 2 que se emplea en el receptor (y por tanto sus cuantificaciones son distintas que la DCT del transmisor) para una entrada aleatoria entre -1 y 1.

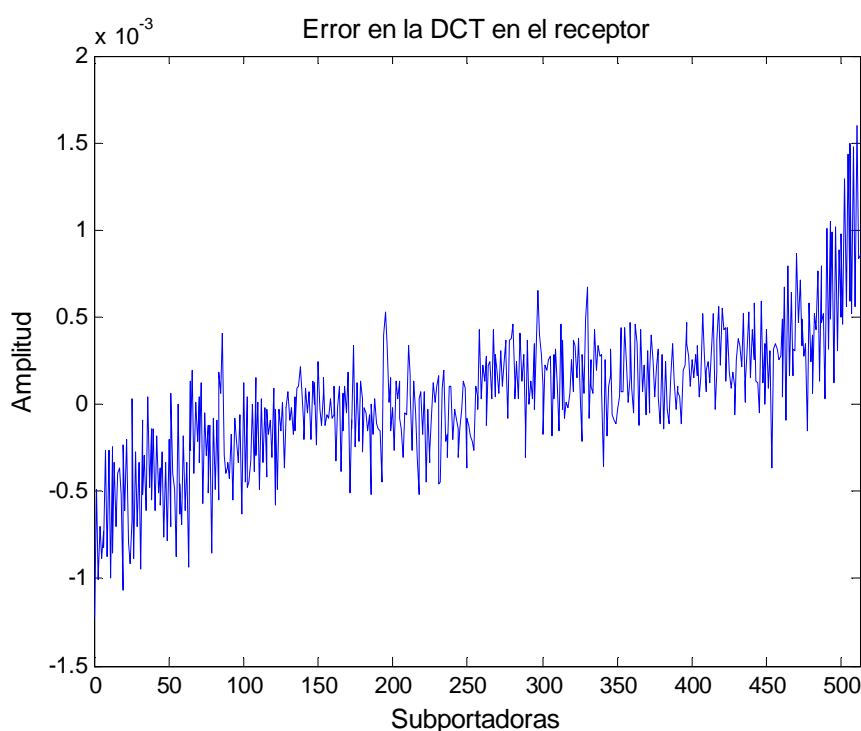


Figura 36. Error en la DCT del algoritmo 2 del receptor para una trama de datos con una entrada aleatoria de -1 y 1.

Se puede observar que en este algoritmo, tanto el transmisor como el receptor, aportan errores similares. A continuación en la Tabla 8 se van a mostrar los errores en los puntos intermedios de la arquitectura cometidos en la DCT del receptor, debidos a la cuantificación. Se han realizado 1000 simulaciones para la obtención del error medio, error máximo y desviación típica.

Tabla 8. Error medio, máximo y desviación típica en cada punto intermedio de la DCT del algoritmo 2 del receptor.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Q1	[25 23]	$8.9611 \cdot 10^{-8}$	$1.6022 \cdot 10^{-7}$	$3.5229 \cdot 10^{-8}$
Q1.2	[25 20]	0	0	0
Q2	[25 23]	$9.3981 \cdot 10^{-8}$	$1.5828 \cdot 10^{-7}$	$3.2882 \cdot 10^{-8}$
Q3	[18 16]	$1.1171 \cdot 10^{-5}$	$2.0711 \cdot 10^{-5}$	$4.5197 \cdot 10^{-6}$
Qifft	[18 16]	$2.7473 \cdot 10^{-5}$	$1.3423 \cdot 10^{-4}$	$2.1233 \cdot 10^{-5}$
Q4	[18 16]	$1.1643 \cdot 10^{-4}$	$2.0345 \cdot 10^{-4}$	$4.1667 \cdot 10^{-5}$
Q5	[18 16]	0	0	0

Donde Q1 es la cuantificación intermedia del segundo coeficiente $e^{\frac{j\pi(l+1/4)}{M}}$ (40); Q1.2 es la cuantificación del segundo coeficiente, después de multiplicar por $\sqrt{\frac{M}{2}}$; Q2 es la cuantificación del primer coeficiente $e^{\frac{j\pi}{M}}$ (38); Q3 es la salida de la multiplicación por el primer coeficiente $y'_l[n]$; Qifft es la salida de la iFFT cuantificada (39); Q4 es la salida de la multiplicación por el segundo coeficiente $Z'_m[k]$ (40); y Q5 el último cambio de orden de la trama de datos $X'_m[k]$ (41).

Como se observa, de nuevo la multiplicación posterior a la iFFT es la que más error introduce debido a la cuantificación. A continuación, en la Tabla 9 se obtiene el error medio, máximo y desviación típica a la salida global de la DCT del algoritmo 2 del receptor. Estos valores contienen una acumulación de los errores de los puntos intermedios. Se han realizado 1000 simulaciones, para una entrada aleatoria entre [-1,+1].

Tabla 9. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 2 del receptor.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Salida	[18 16]	$2.8744 \cdot 10^{-4}$	$1.6000 \cdot 10^{-3}$	$2.59470 \cdot 10^{-4}$

El error que se observa a la salida de la DCT del receptor es similar al obtenido en el transmisor. Sin embargo, este valor es mayor que el obtenido con la DCT del algoritmo 1 en el receptor.

4.5. Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 2

Para la implementación de la arquitectura de la DCT del algoritmo 2 se van a emplear los mismos parámetros que los utilizados en el algoritmo 1. Estos parámetros son la tasa de transmisión $T_{Tx}=62.5\text{Msps}$, el número de subcanales de entrada $M=512$ y el ratio de paralelismo $R_p=2$.

Además, como la implementación se va a realizar en la misma FPGA Virtex 6 xc6vlx240t [110], las restricciones a la arquitectura van a ser las mismas. Estas restricciones son el número máximo de multiplicadores disponibles $C_m^{FPGA}=768$ y la frecuencia de muestreo del convertor $f_s=62.5\text{MHz}$. Con estas restricciones se ha decidido fijar la frecuencia de la técnica de acceso a $f_{per}=62.5\text{MHz}$, lo que fija el ratio de frecuencia en $R_f =$

$\frac{f_s}{f_{per}} = 1$. Con este ratio se obtiene el tiempo de procesamiento máximo $M_c^{max}=512$ ciclos de reloj.

4.5.1. Consumo de recursos

Para el diseño de la DCT, se evalúan las operaciones necesarias para el algoritmo descrito anteriormente. De las cinco fases en las que se puede dividir, la segunda y la cuarta constituyen una multiplicación por una constante compleja. En la tercera fase se realiza una iFFT, para lo cual se utilizará otro core generado por Spiral [111]. Para definir este formato semi-paralelo, se establece un ratio de paralelismo (R_p), que indica el número de datos que trabajan en paralelo. Finalmente, en la primera fase se produce un cambio en el orden de los elementos y en la quinta fase el reordenamiento contrario. Teniendo en cuenta estos comentarios, se puede establecer la siguiente carga computacional para el algoritmo:

Fases 2 y 4, multiplicación por una constante:

$$C_m^{s2} = C_m^{s4} = 4 \cdot R_p \quad \text{multiplicadores por cada etapa} \quad (42)$$

$$C_s^{s2} = C_s^{s4} = 2 \cdot R_p \quad \text{sumadores por cada etapa} \quad (43)$$

Siendo el parámetro R_p el ratio de paralelismo utilizado para proporcionar datos tanto a la iFFT como a las multiplicaciones. El valor de R_p puede variar desde 1 hasta M para cambiar el ratio de paralelismo. Esta modificación afecta tanto al número de recursos utilizados, como a los tiempos de procesamiento y latencia de la arquitectura.

Fase 3, realización de una iFFT, para $L= M/2$ muestras de entrada:

$$C_m^{s3} = C_m^{FFT} \quad \text{multiplicadores} \quad (44)$$

$$C_s^{s3} = C_s^{FFT} \quad \text{sumadores} \quad (45)$$

Fase 1 y 5: no es necesario ningún cálculo matemático, al tratarse de un reordenamiento de la trama de datos.

Por lo tanto, si se suman todas las fases que componen la DCT, se obtiene el número de multiplicadores C_m^{DCT} empleados para la DCT en (46):

$$C_m^{DCT} = C_m^{s2} + C_m^{s3} + C_m^{s4} = 4 \cdot R_p^{DCT} + C_m^{FFT} + 4 \cdot R_p^{DCT} = C_m^{FFT} + 8 \cdot R_p^{DCT} \quad (46)$$

Siendo C_m^{s1} y C_m^{s5} igual a 0.

4.5.2. Cálculo de las restricciones de la arquitectura

Se va a realizar un estudio conjunto de la latencia y el tiempo de procesamiento de la DCT del algoritmo 2 y del consumo de multiplicadores que puede emplear la arquitectura, teniendo en cuenta la frecuencia de reloj del periférico f_{per} , el número de multiplicadores C_m^{DCT} usados, el ratio de frecuencia R_f entre la frecuencia del periférico y la frecuencia del

conversor $R_f=f_{per}/f_s$, y el ratio de paralelismo R_p . Con este estudio se pretende determinar cuáles son las restricciones de la arquitectura para realizar su implementación.

Para realizar el cálculo del tiempo de procesamiento y el consumo de multiplicadores se van a plantear dos restricciones: la primera teniendo en cuenta el número máximo de multiplicadores disponibles; y la segunda el tiempo de procesamiento disponible para realizar las operaciones, teniendo como parámetros el tiempo de procesamiento máximo M_c^{max} en ciclos de reloj, el número C_m^{FFT} de multiplicadores usados por la FFT y el tiempo de procesamiento M_c^{FFT} necesario.

La primera restricción se obtiene del número de multiplicadores C_m^{DCT} (46), con el número máximo de multiplicadores disponibles C_m^{FPGA} , siendo en el caso de las multiplicaciones dependiente del ratio R_p . Para la segunda restricción se ha calculado el tiempo de procesamiento M_c^{DCT} disponible para la DCT, a partir del que se emplea en cada fase, utilizando para ello, el tiempo de procesamiento M_c^{max} límite fijado anteriormente.

$$C_m^{DCT} = C_m^{FFT} + 8 \cdot R_p \leq C_m^{FPGA} \quad (47)$$

$$M_c^{DCT} = M_c^{s1} + M_c^{s2} + M_c^{s3} + M_c^{s4} + M_c^{s5} \leq M_c^{max} \quad (48)$$

Despejando en cada ecuación, $M_c^{s2}=M_c^{s4}=4$ (ciclos de reloj empleados en la multiplicación compleja), y $M_c^{s1}=M_c^{s5}=2$ (ciclos para leer una memoria BRAM), se puede determinar el número máximo de multiplicadores $C_m^{s3}=C_m^{FFT}$ y ciclos de reloj $M_c^{s3}=M_c^{FFT}$ que se pueden utilizar para la iFFT utilizada en función del valor de R_p .

$$C_m^{FFT} \leq C_m^{FPGA} - 8 \cdot R_p \quad (49)$$

$$M_c^{FFT} = M_c^{s3} \leq M_c^{max} - 4 \cdot 2 - 2 \cdot 2 \quad (50)$$

Por otro lado, la latencia L^{DCT} de la DCT es la suma de las latencias de cada fase según, (51):

$$L^{DCT} = L^{s1} + L^{s2} + L^{s3} + L^{s4} + L^{s5} \quad (51)$$

Donde $L^{s2}=L^{s4}=8$, son las latencias de la multiplicación compleja, L^{s3} es la latencia de la iFFT y $L^{s1}=L^{s5}=M$ es la latencia del cambio de orden. Se recuerda que la latencia no tiene impacto en el diseño, solamente se emplea para tener una mejor definición de la arquitectura.

4.5.3. Obtención de los parámetros para la FFT y la DCT

Existen varias configuraciones de iFFT que se pueden utilizar, primando ante todo el empleo del menor número de multiplicadores posible. Para generar la iFFT se ha empleado un core que proporciona Spiral [111], el cual permite generar distintas arquitecturas con distintos ratios de paralelismo R_p . En este caso se ha elegido también una iFFT con un ratio de paralelismo $R_p=2$, de tamaño $M/2=256$ puntos cuyos parámetros más característicos son el número de multiplicadores C_m^{FFT} (52), y el tiempo de procesamiento M_c^{FFT} que necesita para procesar la trama de datos, junto con la latencia L^{FIT} (53):

4.5 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 2

$$C_m^{FFT} = 28 \text{ multiplicadores} \quad (52)$$

$$M_c^{FFT} = 128 \text{ ciclos}, \quad L^{FFT} = 403 \text{ ciclos} \quad (53)$$

Estos datos permiten que la DCT cumpla con los límites calculados en (49) y (50). Y por tanto se puede obtener el número final de multiplicadores C_m^{DCT} empleados descrito en (46), según (54):

$$C_m^{DCT} = C_m^{s2} + C_m^{s3} + C_m^{s4} = 8 + 28 + 8 = 44 \text{ multiplicadores} \quad (54)$$

Donde $R_p=2$.

Por otro lado, la latencia L^{DCT} queda definida por (55):

$$L^{DCT} = L^{s1} + L^{s3} + L^{s3} + L^{s4} + L^{s5} = 1443 \text{ ciclos} \quad (55)$$

Donde $L^{s1}=M$, latencia de la fase s1 (37), debida a las memorias ping-pong; $L^{s2}=8$ es la latencia de la fase s2, debida a la multiplicación compleja (38); $L^{s3}=L^{FFT}=403$ es la latencia de la iFFT (39); $L^{s4}=8$ es la latencia de fase s4, debida a la multiplicación compleja (40); $L^{s5}=M$ es la latencia de la fase s5 (41), debida a las memorias ping-pong.

4.5.4. Arquitectura del transmisor

Para el caso del emisor se puede unir el módulo de la extensión simétrica con la última fase s5 de la DCT, consiguiendo de esta manera, reducir la latencia y el consumo de recursos final del diseño. En la Figura 37 se puede observar la arquitectura propuesta para el transmisor de la DTT.

La primera fase s1 de la DCT del algoritmo 2 es un reordenamiento de la trama de datos (37), donde todas las muestras de la trama $X_m[k]$ debe estar disponible al mismo tiempo. Por lo tanto, los datos deben ser almacenados en memoria, para que sean leídos en el orden correcto. Para evitar pérdida de datos, se han empleado dos memorias en modo ping-pong, por lo que en una se puede acceder para organizar los datos en el orden correcto, mientras que los nuevos datos se escriben en la otra. Esta fase implica una latencia de $L^{s1}=M=512$ ciclos.

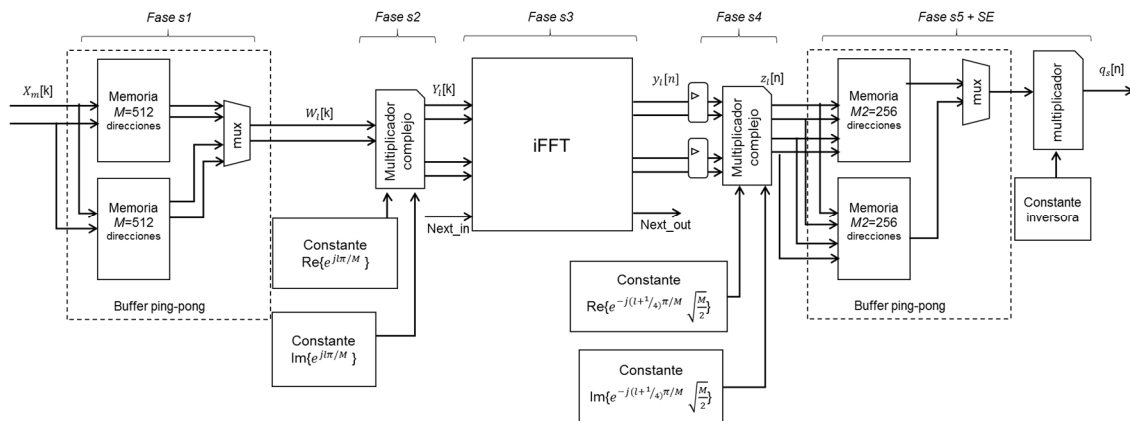


Figura 37. Diagrama de bloques de la arquitectura del transmisor DTT con DCT del algoritmo 2.

Dado que el ratio de paralelismo se ha fijado en $R_p=2$, $C_m^{s2}=8$ multiplicadores son necesarios en la fase s2 (38), donde la latencia es de $L^{s2}=8$ ciclos de reloj. Después, la tercera fase s3 es una iFFT que requiere no sólo multiplicadores y sumadores, si no también memoria donde almacenar los valores intermedios (39). Este módulo es parametrizado con un tamaño de $M/2 = 256$ muestras y una representación en coma fija de 18 bits, valor fijado por el ancho de palabra que proporcionan una de las entradas de los multiplicadores de la FPGA. La iFFT presenta una arquitectura radix-2 semi-paralela, con $\log_2 M/2$ bloques en cascada. De nuevo, la selección del ratio de paralelismo $R_p=2$ implica que sólo dos muestras de entrada $Y_l[k]$ son procesadas cada ciclo de reloj. Una iFFT se completa cada $M_c^{FFT}=128$ ciclos de reloj, con una latencia $L^{FFT}=403$ ciclos, compatible con el tiempo de procesamiento disponible indicado en las restricciones.

La cuarta fase s4 del módulo DCT es similar a la segunda fase s2 (40), donde $C_m^{s4}=8$ multiplicadores son usados y la latencia es de $L^{s4}=8$ ciclos de reloj. Por último, la quinta fase s5 de la DCT es un reordenamiento de la trama de datos (41), donde todas las muestras de la trama de datos $z_l[n]$ deben estar disponibles al mismo tiempo. Por lo tanto, los datos deben ser almacenados en memoria, para que sean leídos en el orden correcto. Para evitar pérdida de datos, se han empleado dos memorias en modo ping-pong. Esta fase implica una latencia de $L^{s5}=M=512$ ciclos.

Como se puede observar, las cinco fases que forman el módulo DCT están segmentadas. En consecuencia, el throughput de cada datapath es de $T_{ch}=1$ dato por cada ciclo de reloj y como el ratio de paralelismo es $R_p=2$, el throughput global de la DCT es $T_c^{DCT}=T_{ch} \cdot R_p=2$. Con todo esto, todas las muestras de entrada $X_m[k]$ son procesadas cada $M_c^{DCT}=128$ ciclos. Como una nueva trama de muestras de entrada $X_m[k]$ está disponible cada $M_c^{max}=512$ ciclos, la arquitectura propuesta cumple con esta restricción.

Por lo tanto, el número de multiplicaciones C_m^{Tx} empleadas para la realización de la arquitectura del transmisor de la DTT con la DCT del algoritmo 2 es (56):

$$C_m^{Tx} = C_m^{s2} + C_m^{s3} + C_m^{s4} + C_m^{SE,Tx} = 8 + 28 + 8 + 1 = 45 \text{ multiplicadores} \quad (56)$$

Donde $C_m^{s2}=8$ es el número de multiplicadores en la fase s2 (38); $C_m^{s3}=28$ es el número de multiplicadores empleados en la iFFT (39); $C_m^{s4}=8$ es el número de multiplicadores empleados en la fase s4 (40); y $C_m^{SE,Tx}=1$ es el número de multiplicadores empleados en el módulo de la extensión simétrica (8).

La latencia L^{Tx} de la arquitectura propuesta para el transmisor basado en Transformadas Trigonométricas Discretas con la DCT del algoritmo 2 y el módulo de extensión simétrica se obtiene en (57):

$$L^{Tx} = L^{s1} + L^{s2} + L^{s3} + L^{s4} + L^{SE,Tx} = 1445 \text{ ciclos} \quad (57)$$

Donde $L^{s1}=512$ es la latencia de la fase s1 (37), debida a las memorias ping-pong; $L^{s2}=8$ es la latencia de fase s2, debida a la multiplicación compleja (38); $L^{s3}=L^{FFT}=403$ es la

4.5 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 2

latencia de la iFFT (39); $L^{s4}=8$ es la latencia de fase s4, debida a la multiplicación compleja (40); $L^{SE,Tx}=514$ es la latencia del módulo SE (9), debida a las memorias ping-pong.

Una vez se ha mostrado la arquitectura realizada, se procede a mostrar un cronograma del funcionamiento del transmisor basado en la DTT con la DCT del algoritmo 2 y el módulo de extensión simétrica, que se puede observar en la Figura 38. En este cronograma se observan las principales señales de la propuesta y gráficamente los retardos internos para producir la salida.

Primero, se recibe la señal $X_m[k]$, sincronizada con la señal CE_M ; junto con esta señal CE_M se activa el enable y se comienzan a generar direcciones para la memoria que almacena los coeficientes $e^{jl\pi/M}$ de la primera multiplicación (38) (muestra de reloj 1). A continuación se genera el cambio de la trama de datos obteniendo la señal $W_l[k]$ a partir de $X_m[k]$ (37) (muestra de reloj 2). Esta señal está sincronizada con el coeficiente $e^{jl\pi/M}$ en la entrada del multiplicador, produciéndose 4 ciclos de reloj después la salida del multiplicador $Y_l[k]$ (muestra de reloj 7). Esta señal $Y_l[k]$ se retrasa para que esté sincronizada con la señal de comienzo de la iFFT $next_in$ (muestra de reloj 8).

Cuando la iFFT (39) ha sido realizada, se activa la señal $next_out$ y a continuación, comienza a proporcionarse la salida $y_l[n]$ de la FFT (muestra de reloj 13). Esta señal se sincroniza con los coeficientes $e^{\frac{j\pi(l+1/4)}{M}} \cdot \sqrt{\frac{M}{2}}$ de la segunda multiplicación (40) (muestra de reloj 15). La salida $z_l[n]$ se sincroniza con las direcciones para el puerto A de la memoria ping-pong que realiza el reordenamiento del array (41) (muestra de reloj 20). La salida $q_s[n]$ se produce un ciclo después de la dirección del puerto B de la memoria ping-pong (muestra de reloj 22). Por último, esta salida se multiplica por el coeficiente -1 para generar la extensión simétrica y por tanto la salida $q_s[n]$ (muestra de reloj 24).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
CEM																										
Xn[k]		Dn0	Dn1	Dn2	Dn3	Dn4	Dn5	Dn6	Dn7	Dn8	Dn9															
Vn[k]			Dn0	Dn1	Dn2	Dn3	Dn4	Dn5	Dn6	Dn7	Dn8															
en																										
Addr1		0	1	2	3	4	5	6	7	8	9															
coeficientes			Coef0	Coef1	Coef2	Coef3	Coef4	Coef5	Coef6	Coef7	Coef8															
Mult				Coef0	Coef1	Coef2	Coef3	Coef4	Coef5	Coef6	Coef7															
Vn[k+1]				din0	din1	din2	din3	din4	din5	din6	din7															
Vn[k]								mult0	mult1	mult2	mult3															
Vn[k+1]																										
Vn[k+2]																										
next_in																										
next_out																										
Yn[0]																										
Yn[1]																										
Yn[2]																										
Yn[3]																										
coeficientes																										
mult2																										
Zn[0]																										
Zn[1]																										
Zn[2]																										
addrA																										
addrB																										
qs[0]																										
qs[1]																										
qs[2]																										
qs[3]																										
qs[4]																										
qs[5]																										
qs[6]																										
qs[7]																										
qs[8]																										
qs[9]																										
qs[10]																										
qs[11]																										
qs[12]																										
qs[13]																										
qs[14]																										
qs[15]																										
qs[16]																										
qs[17]																										
qs[18]																										
qs[19]																										
qs[20]																										
qs[21]																										
qs[22]																										
qs[23]																										
qs[24]																										
qs[25]																										
qs[26]																										
qs[27]																										
qs[28]																										
qs[29]																										
qs[30]																										

Figura 38. Cronograma de la arquitectura del transmisor DTT con la DCT del algoritmo 2.

4.5.5. Arquitectura del receptor

En el caso del receptor, el módulo de la extensión simétrica se une con la fase s1 de la DCT del algoritmo 2. Con lo que en este caso también se consigue reducir la latencia del receptor, al igual que sucedía en el transmisor. En la Figura 39 se muestra la arquitectura realizada para el receptor DTT.

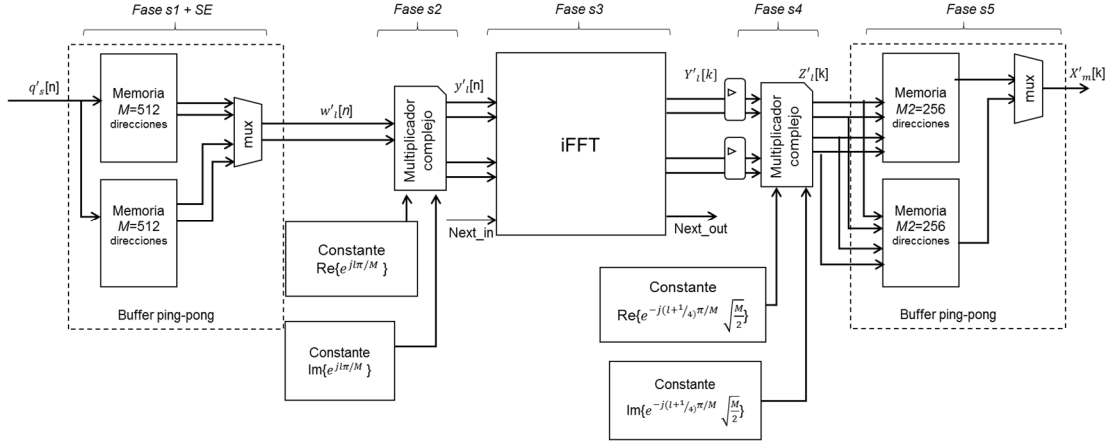


Figura 39. Diagrama de bloques de la arquitectura del receptor DTT con DCT del algoritmo 2.

En el caso del receptor, al igual que en el del transmisor, el ratio de paralelismo se ha fijado en $R_p=2$. Debido a que la DCT implementada es ortogonal, sigue el mismo proceso de arquitectura que la DCT del transmisor. Por lo tanto, el número de multiplicaciones C_m^{Rx} empleadas para la realización de la arquitectura del receptor de la DTT con DCT del algoritmo 2 es (58):

$$C_m^{Rx} = C_m^{s2} + C_m^{s3} + C_m^{s4} = 8 + 28 + 8 = 44 \text{ multiplicadores} \quad (58)$$

Donde $C_m^{s2}=8$ es el número de multiplicadores en la fase 2 (38); $C_m^{s3}=28$ es el número de multiplicadores empleados en la FFT (39); $C_m^{s4}=8$ es el número de multiplicadores empleados en la fase 4 (40); y $C_m^{SE,Rx}=0$ es el número de multiplicadores empleados en el módulo SE.

La latencia L^{Rx} de la arquitectura propuesta para la DCT del algoritmo 2 y el módulo de la extensión simétrica es obtenida en (59):

$$L^{Rx} = L^{s1} + L^{s2} + L^{s3} + L^{s4} + L^{s5} = 1443 \text{ ciclos} \quad (59)$$

Donde $L^{s1}=512$ es la latencia de la fase s1(37), debida a memorias ping-pong, y también contiene al módulo de la extensión simétrica (10); $L^{s2}=8$ es la latencia de fase s2, debida a la multiplicación compleja (38); $L^{s3}=L^{FFT}=403$ es la latencia de la iFFT (39); $L^{s4}=8$ es la latencia de fase s4, debida a la multiplicación compleja (40); $L^{s5}=512$ es la latencia de la fase s5 (41), debida a las memorias ping-pong.

Una vez se ha mostrado la arquitectura realizada, se procede a mostrar un cronograma del funcionamiento de la DCT y el módulo SE, que se puede observar en la Figura 40. En

este cronograma se observan las principales señales de la propuesta y gráficamente los retardos internos para producir la salida.

Primero, se recibe la señal $q'_s[n]$, sincronizada con la señal CE_M , junto con esta señal CE_M se activa el enable y se comienzan a generar direcciones para la memoria que almacena los coeficientes $e^{jl\pi/M}$ de la primera multiplicación (38) (muestra de reloj 1). A continuación se genera el cambio de la trama de datos obteniendo la señal $w'_l[n]$ a partir de $q'_s[n]$ (37) (muestra de reloj 2). Esta señal está sincronizada con el coeficiente $e^{jl\pi/M}$ en la entrada del multiplicador, produciéndose 4 ciclos de reloj después la salida del multiplicador $y'_l[n]$ (muestra de reloj 7). Esta señal $y'_l[n]$ se retrasa para que esté sincronizada con la señal de comienzo de la iFFT $next_in$ (muestra de reloj 9).

Cuando la iFFT (39) ha sido realizada, se activa la señal $next_out$ y a continuación, comienza a proporcionarse la salida $Y'_l[k]$ de la FFT (muestra de reloj 13). Esta señal se sincroniza con los coeficientes $e^{\frac{j\pi(l+1/4)}{M}} \cdot \sqrt{\frac{M}{2}}$ de la segunda multiplicación (40) (muestra de reloj 15). La salida $Z'_l[k]$ se sincroniza con las direcciones para el puerto A de la memoria ping-pong que realiza el reordenamiento de la trama de datos (41) (muestra de reloj 20). La salida $X'_m[k]$ se produce un ciclo después de la dirección del puerto B de la memoria ping-pong (muestra de reloj 22).

4.5 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 2

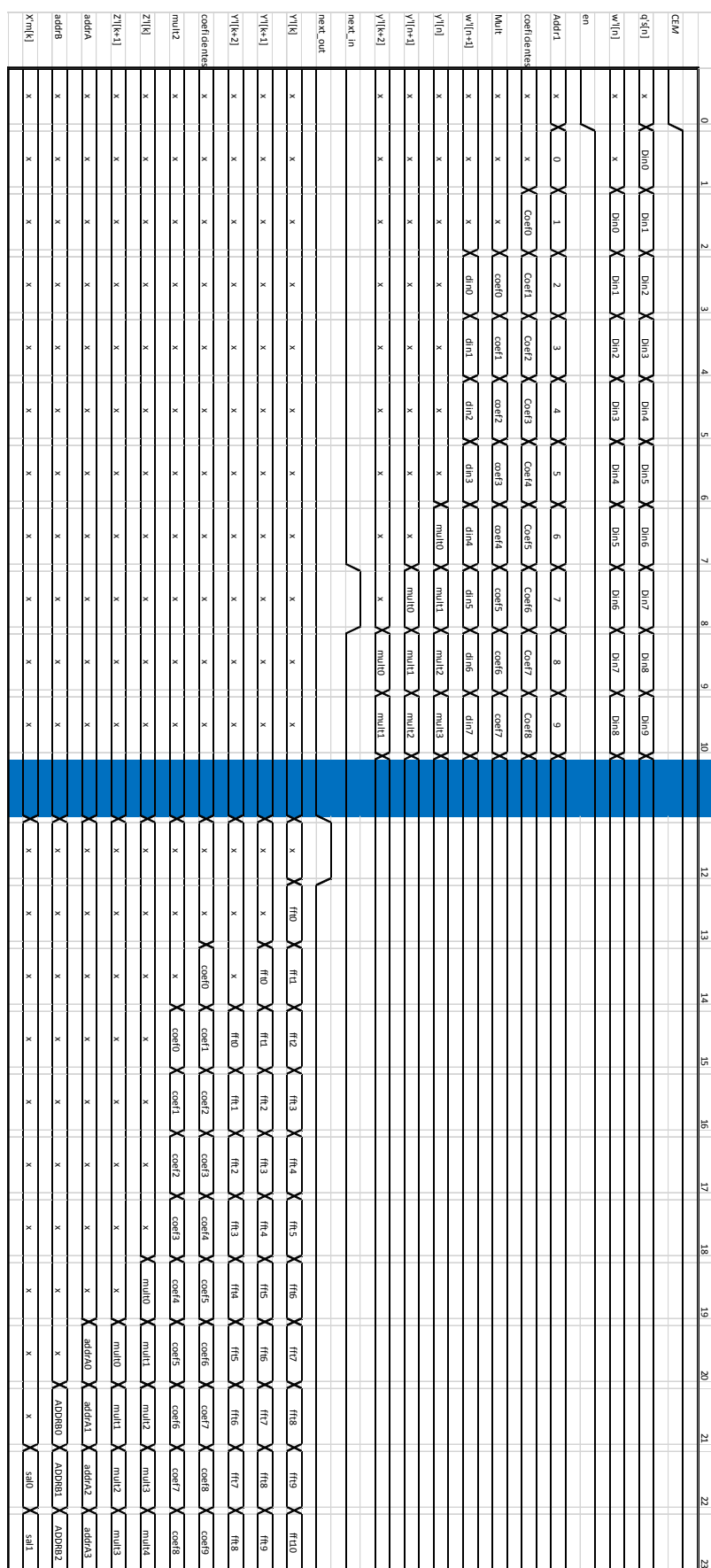


Figura 40. Cronograma de tiempos de la arquitectura del receptor DTT con la DCT del algoritmo 2.

4.6. Algoritmo 3 para la realización de la Transformada Discreta del Coseno

Se va a estudiar un nuevo algoritmo que, como característica principal, posee un pre-filtro a la salida. Con este pre-filtrado se pretende eliminar parte del ruido de cuantificación que se introduce en la FFT. La característica de pre-filtrado en la DCT se comenta en artículos como [104] [105]. En esta propuesta el cálculo de la DCT se realiza en seis pasos:

- a) Generación de la secuencia $W_m[k]$ a partir de la señal de entrada dada $X_m[k]$ según (60):

$$W_m[k] = X_m[k]e^{-j(m+1/2)\pi/2M} \quad (60)$$

- b) Se reordena la señal $W_m[k]$, obteniendo la señal $Y_m[k]$ según el siguiente método (61):

$$Y_{2i+1}[k] = W_i[k], \quad i=0, \dots, M/2-1 \quad (61)$$

$$Y_{2i}[k] = W_{M-i}[k], \quad i=0, \dots, M/2-1$$

- c) Dado un instante de tiempo $n=n_0$, se obtiene la señal $y_m[n_0]$ mediante la FFT de M puntos de $Y_m[k_0]$ según (62).

$$y_m[n_0] = F\{Y[k_0]\} \quad (62)$$

Donde el operador $F\{\}$ realiza la FFT de M puntos.

- d) Generación de la secuencia $z_m[n]$ a partir de la señal $y_m[n]$ según (63).

$$z_m[n] = \text{Re}\{y_m[n] \cdot 2 \cdot \frac{e^{-j\pi m}}{\sqrt{2M}}\} \quad m=0, \dots, M-1 \quad (63)$$

Donde $\text{Re}\{\}$ toma la parte real de los datos calculados.

- e) Se divide el primer coeficiente de la señal de $z_m[n]$, según el siguiente método (64):

$$z_0[n] = \frac{z_0[n]}{2} \quad (64)$$

- f) Se realiza una etapa de pre-filtrado en $z_m[n]$, obteniendo la señal $p_m[n]$ según (65):

$$p_m[n] = \text{filter}\{z_m[n], H_z\} \quad (65)$$

Donde el operador $\text{filter}\{\}$ realiza una etapa de pre-filtrado sobre la señal $y_m[n]$, empleando $H_z = \frac{2}{[z^{-1}+1]}$.

4.6.1. Efecto de la precisión finita del transmisor

Como se hizo para los otros modelos de DCT, se ha realizado un estudio del efecto de la cuantificación (representación en precisión finita) en la arquitectura, tratando de determinar el número de bits y su configuración más adecuada. También se ha considerado que la plataforma en la que se desea implementar es la ML605 de Xilinx con una FPGA

4.6 Algoritmo 3 para la realización de la Transformada Discreta del Coseno

Virtex 6 xc6vlx240t y que ésta posee celdas multiplicadoras DSP48E1 con entrada de 18x25 bits.

En la Figura 41 se puede observar los puntos de cuantificación utilizados dentro de la DCT y su valor. Se ha decidido emplear una cuantificación de 18 bits, variando el número de bits de la parte fraccionaria dependiendo de los valores máximos de cada zona, y una cuantificación de 25 bits para los coeficientes.

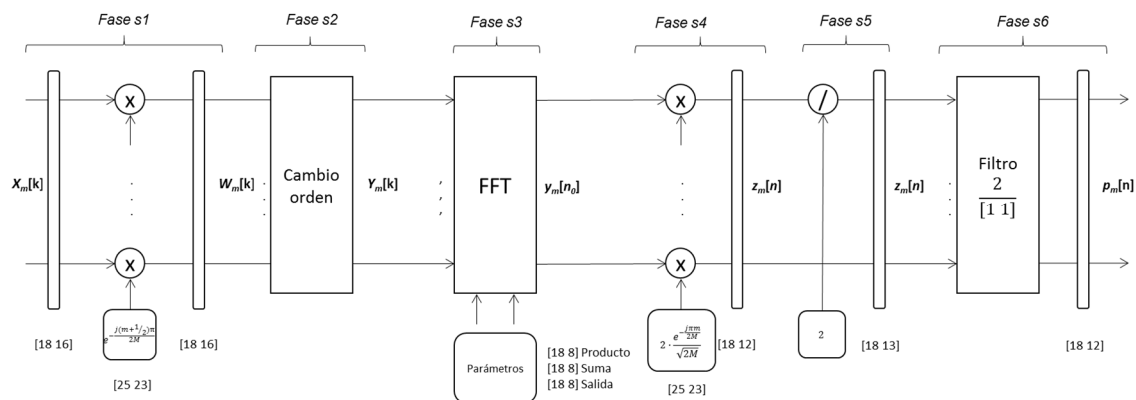


Figura 41. Diagrama de bloques con cuantificación empleada en el módulo DCT del algoritmo 3.

Este diseño posee 6 fases, siendo la primera una multiplicación por una constante compleja y emplea 25 bits para los coeficientes. Después, en la fase s2, se realiza un reordenamiento de la trama de datos. En la fase s3 se realiza la FFT de M puntos. En la fase s4 se realiza otra multiplicación por una constante compleja. En la fase s5 se divide entre dos el primer elemento de la trama. Y por último, en la fase s6 se realiza el pre-filtrado, utilizando para ello las celdas multiplicadoras DSP48E1. En la Figura 42 se muestran el error obtenido en la DCT para una trama de datos, para una entrada aleatoria entre -1 y 1. En la figura se muestra el error en cada subportadora de la DCT del transmisor.

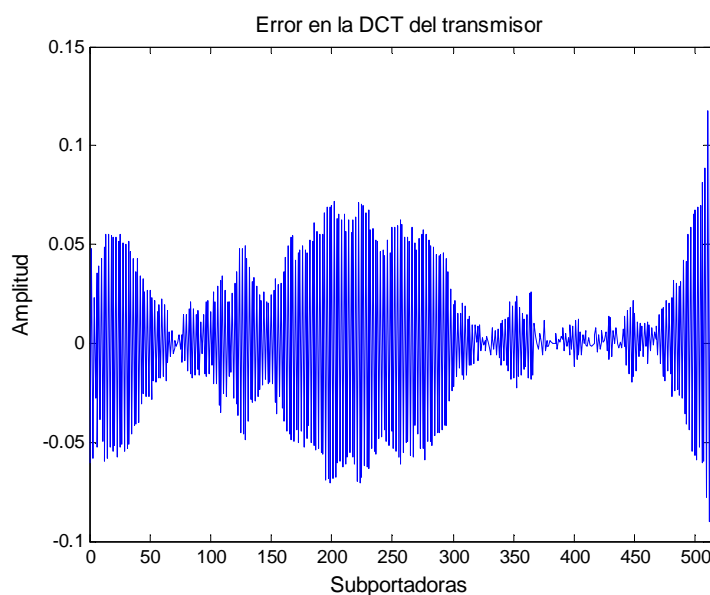


Figura 42. Error obtenido para una trama de datos en la DCT del algoritmo 3 del transmisor con una entrada aleatoria entre -1 y 1.

Como se observa, el pre-filtro consigue eliminar el error que produce la FFT en las primeras subportadoras, pero en cambio, introduce mayor error en otras subportadoras del transmisor.

A continuación en la Tabla 10 se van a mostrar los errores en los puntos intermedios de la arquitectura cometidos en la DCT, debidos a la cuantificación. Para obtener los resultados se han realizado 1000 simulaciones con una señal aleatoria entre [-1,+1].

Tabla 10. Error medio, máximo y desviación típica en cada punto intermedio en la DCT del algoritmo 3.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Q1	[25 23]	$5.9287 \cdot 10^{-8}$	$1.1845 \cdot 10^{-7}$	$3.3473 \cdot 10^{-8}$
Q2	[18 16]	$7.8496 \cdot 10^{-6}$	$1.5253 \cdot 10^{-5}$	$4.2421 \cdot 10^{-6}$
Q3	[18 16]	0	0	0
Q4	[25 23]	$9.1838 \cdot 10^{-8}$	$1.6251 \cdot 10^{-7}$	$3.3019 \cdot 10^{-8}$
Qfft	[18 8]	$5.2800 \cdot 10^{-2}$	$9.7020 \cdot 10^{-1}$	$6.1900 \cdot 10^{-2}$
Q5	[18 12]	$6.2728 \cdot 10^{-5}$	$1.2182 \cdot 10^{-4}$	$3.5489 \cdot 10^{-5}$
Q6	[18 13]	0	0	0
Q7	[18 12]	$1.37 \cdot 10^{-2}$	$2.6400 \cdot 10^{-2}$	$5.9000 \cdot 10^{-3}$

Donde Q1 es la cuantificación del primer coeficiente $e^{-j(m+1/2)\pi/2M}$; Q2 es la salida de la multiplicación por el primer coeficiente (60); Q3 es el reordenamiento de la trama de datos (61); Q4 es la cuantificación del segundo coeficiente $2 \cdot \frac{e^{-j\pi m}}{\sqrt{2M}}$; Qfft es la salida de la FFT cuantificada (62); Q5 es la salida de la multiplicación por el segundo coeficiente (63); Q6 es la división del primer parámetro entre dos (64); y Q7 la etapa de pre-filtrado (65).

En este caso, la FFT vuelve a ser el elemento que mayor error introduce debido a la cuantificación. Además en este algoritmo, existe un segundo elemento que introduce un error considerable, siendo este elemento el pre-filtrado.

Una vez se han mostrado los errores en los puntos intermedios que se cometen en la DCT del algoritmo 3, se procede a mostrar el error medio, máximo y desviación típica que se obtiene a la salida global de la DCT, para una entrada aleatoria entre [-1,+1], realizando 1000 simulaciones. En la Tabla 11 se muestran los valores obtenidos a la salida de la DCT, produciéndose en éstos una acumulación de los resultados parciales mostrados anteriormente

Tabla 11. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 3.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Salida DCT	[18 12]	$3.8200 \cdot 10^{-2}$	$1.5410 \cdot 10^{-1}$	$2.6700 \cdot 10^{-2}$

4.6.2. Efecto de la precisión finita del receptor

En la Figura 43 se muestra el error obtenido en la DCT del algoritmo 3 del receptor para una trama de datos, empleando una entrada aleatoria entre -1 y 1.

4.6 Algoritmo 3 para la realización de la Transformada Discreta del Coseno

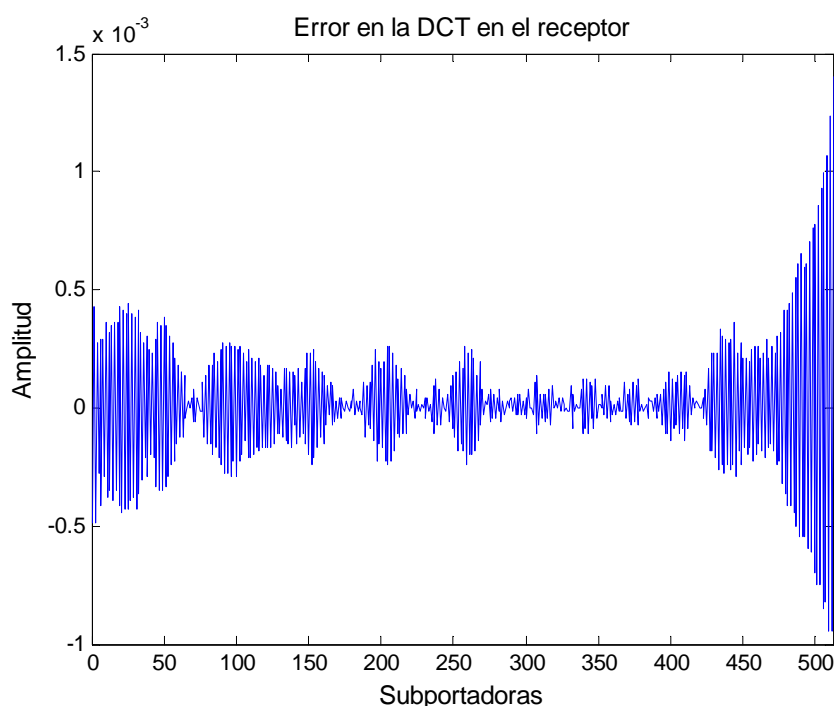


Figura 43. Error en la DCT del algoritmo 3 del receptor para una trama de datos con una entrada aleatoria de -1 y 1.

Como se observa en la figura, el receptor introduce menos error que el transmisor. Esto se produce por la cuantificación empleada en el receptor. A continuación en la Tabla 12 se van a mostrar los errores en los puntos intermedios de la arquitectura cometidos en la DCT del receptor, debidos a la cuantificación. Para obtener estos errores, se han realizado 1000 simulaciones, empleando una señal aleatoria entre [-1,+1].

Tabla 12. Error medio, máximo y desviación típica en cada punto intermedio en la DCT del algoritmo 3 del receptor.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
Q1	[25 23]	$5.9287 \cdot 10^{-8}$	$1.1845 \cdot 10^{-7}$	$3.3473 \cdot 10^{-8}$
Q2	[18 16]	$7.6149 \cdot 10^{-6}$	$1.5254 \cdot 10^{-5}$	$4.3862 \cdot 10^{-6}$
Q3	[18 16]	0	0	0
Q4	[25 23]	$9.1838 \cdot 10^{-8}$	$1.6251 \cdot 10^{-7}$	$3.3019 \cdot 10^{-8}$
Qfft	[18 15]	$3.1271 \cdot 10^{-4}$	$4.0000 \cdot 10^{-3}$	$4.0551 \cdot 10^{-4}$
Q5	[18 15]	$7.7933 \cdot 10^{-6}$	$1.5231 \cdot 10^{-5}$	$4.2374 \cdot 10^{-6}$
Q6	[18 16]	0	0	0
Q7	[18 16]	$1.1340 \cdot 10^{-4}$	$3.6621 \cdot 10^{-4}$	$9.3847 \cdot 10^{-5}$

Donde Q1 es la cuantificación del primer coeficiente $e^{-j(m+1/2)\pi/2M}$; Q2 es la salida de la multiplicación por el primer coeficiente (60); Q3 es el reordenamiento de la trama de datos (61); Q4 es la cuantificación del segundo coeficiente $2 \cdot \frac{e^{-j\pi m}}{\sqrt{2M}}$; Qfft es la salida de la FFT cuantificada (62); Q5 es la salida de la multiplicación por el segundo coeficiente (63); Q6 es la división del primer parámetro entre dos (64); y Q7 la etapa de pre-filtrado (65).

Al igual que sucedía en el transmisor, la DCT del receptor posee dos elementos que introducen mayor error. Si bien estos errores son menores que los que se introduce en el transmisor. Estos dos elementos son la FFT y la etapa de pre-filtrado.

En la Tabla 13 se puede observar el error medio, máximo y desviación típica obtenidos a la salida global de la DCT con una entrada aleatoria entre $[-1,+1]$. Para la obtención de los datos se han realizado 1000 simulaciones. Se recuerda que este error acumula el error parcial de cada punto intermedio mostrado antes.

Tabla 13. Error medio, máximo y desviación típica en la salida de la DCT del algoritmo 3 del receptor.

Salida	[18 16]	Error absoluto medio	Error absoluto máximo	Desviación típica
		$1.8200 \cdot 10^{-4}$	$1.4000 \cdot 10^{-3}$	$1.8672 \cdot 10^{-4}$

El error generado por el receptor, es significativamente menor que el generado por el transmisor. Aunque este valor no mejora el obtenido por el algoritmo 1 en la recepción.

4.7. Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 3

Para la implementación de la arquitectura de la DCT del algoritmo 3 se van a emplear los mismos parámetros que los utilizados anteriormente. Estos parámetros son la tasa de transmisión $T_{Tx}=62.5\text{Msps}$, el número de subcanales de entrada $M=512$ y el ratio de paralelismo $R_p=2$.

Además, como la implementación se va a realizar en la misma FPGA Virtex 6 xc6vlx240t [110], las restricciones a la arquitectura van a ser las mismas. Estas restricciones son el número máximo de multiplicadores disponibles $C_m^{FPGA}=768$ y frecuencia de muestreo del convertor $f_s=62.5\text{MHz}$. Con estas restricciones se ha decidido fijar la frecuencia de la técnica de acceso a $f_{per}=62.5\text{MHz}$, lo que fija el ratio de frecuencia en $R_f = \frac{f_s}{f_{per}} = 1$. Con este ratio se obtiene el tiempo de procesamiento máximo $M_c^{max}=512$ ciclos de reloj.

4.7.1. Consumo de recursos

Para el diseño de la DCT, se ha evaluado las operaciones necesarias para el algoritmo descrito anteriormente. De las seis fases en los que se puede dividir, la primera y la cuarta constituyen una multiplicación por una constante compleja. En la tercera fase s3 se realiza una FFT, para lo cual se utilizará un core generado por Spiral [111]. En la segunda fase s2 se produce un cambio en el orden de los elementos y en la quinta fase s5 la división entre dos del primer elemento de la trama de datos. Finalmente, en la fase s6, se produce un pre-filtrado de la señal de salida. Teniendo en cuenta estos comentarios, se puede establecer la siguiente carga computacional para el algoritmo:

Fases s1 y s4, multiplicación por una constante:

$$C_m^{s1} = C_m^{s4} = 4 \cdot R_p \quad \text{multiplicadores por cada etapa} \quad (66)$$

$$C_s^{s1} = C_s^{s4} = 2 \cdot R_p \quad \text{sumadores por cada etapa} \quad (67)$$

4.7 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 3

La fase s2, realiza un reordenamiento de la trama de datos, por tanto no emplea ningún multiplicador.

Fase s3, realización de una FFT para M muestras de entrada:

$$C_m^{s3} = C_m^{FFT} \text{ multiplicadores} \quad (68)$$

$$C_s^{s3} = C_s^{FFT} \text{ sumadores} \quad (69)$$

Fase s5: división del primer elemento de la trama de datos entre 2. O lo que es lo mismo, se produce un desplazamiento de los bits hacia la derecha, y por lo tanto, no se emplea ningún operador.

Fase s6: pre-filtrado de la salida. Se realiza un pre-filtrado de la señal de salida con la función $H_z = \frac{2}{[z^{-1} + 1]}$. Por lo tanto el número de multiplicadores empleados es el calculado según (70) y (71):

$$C_m^{s6} = C_m^{preFiltro} \text{ multiplicadores} \quad (70)$$

$$C_s^{s6} = C_s^{preFiltro} \text{ sumadores} \quad (71)$$

Donde $C_m^{preFiltro} = 1 \cdot R_p$ multiplicadores y $C_s^{preFiltro} = 1 \cdot R_p$ sumadores.

Por lo tanto, si se suman todas las fases que componen la DCT, se obtiene el número de multiplicadores C_m^{DCT} empleados para la DCT del algoritmo 3 en (72):

$$C_m^{DCT} = C_m^{s1} + C_m^{s3} + C_m^{s4} + C_m^{s6} = 4 \cdot R_p + C_m^{FFT} + 4 \cdot R_p + 1 \cdot R_p = C_m^{FFT} + 9 \cdot R_p \quad (72)$$

Recuerde que C_m^{s2} y C_m^{s5} no emplean ningún multiplicador.

4.7.2. Cálculo de las restricciones de la arquitectura

Se va a realizar un estudio conjunto de la latencia y el tiempo de procesamiento de la DCT del algoritmo 3 y del consumo de multiplicadores que puede emplear la arquitectura, teniendo en cuenta la frecuencia de reloj del periférico f_{per} , el número de multiplicadores C_m^{DCT} usados, el ratio de frecuencia entre la frecuencia del periférico y la frecuencia del conversor $R_f = f_{per}/f_s$, y el ratio de paralelismo R_p . Con este estudio se pretende obtener las restricciones de la arquitectura.

Se plantean dos restricciones: la primera teniendo en cuenta el número máximo de multiplicadores disponibles; y la segunda el tiempo de procesamiento disponible para realizar las operaciones, teniendo como parámetros el tiempo de procesamiento máximo M_c^{max} disponible en ciclos de reloj, el número de multiplicadores C_m^{FFT} usado por la FFT y el tiempo de procesamiento M_c^{FFT} necesario.

La primera restricción se obtiene del número de multiplicadores C_m^{DCT} (72) empleando el número máximo de multiplicadores disponibles C_m^{FPGA} , siendo en el caso de las multiplicaciones dependiente del ratio de paralelismo R_p . Para la segunda restricción se ha

calculado el tiempo de procesamiento M_c^{DCT} disponible para la DCT, a partir del que se emplea en cada fase, y el tiempo de procesamiento M_c^{max} límite fijado anteriormente.

$$C_m^{DCT} = C_m^{FFT} + 9 \cdot R_p \leq C_m^{FPGA} \quad (73)$$

$$M_c^{DCT} = M_c^{s1} + M_c^{s2} + M_c^{s3} + M_c^{s4} + M_c^{s6} \leq M_c^{max} \quad (74)$$

Despejando en cada ecuación, $M_c^{s1}=M_c^{s4}=4$ (ciclos de reloj empleados en la multiplicación compleja), $M_c^{s2}=2$ (ciclos para leer una memoria BRAM), y $M_c^{s6}=6$ (ciclos para realizar multiplicación y acumulación), se puede saber el número máximo de multiplicadores C_m^{s3} y ciclos de reloj M_c^{s3} que se pueden utilizar para la FFT empleada en función del valor de R_p .

$$C_m^{FFT} \leq C_m^{FPGA} - 9 \cdot R_p \quad (75)$$

$$M_c^{FFT} = M_c^{s3} \leq M_c^{max} - 4 \cdot 2 - 8 \quad (76)$$

Por otro lado, la latencia L^{DCT} de la DCT es la suma de las latencias de cada fase según (77):

$$L^{DCT} = L^{s1} + L^{s2} + L^{s3} + L^{s4} + L^{s5} + L^{s6} \quad (77)$$

Donde $L^{s2}=L^{s4}=8$ son las latencias de la multiplicación compleja; L^{s3} es la latencia de la FFT; $L^{s2}=M$ es la latencia del cambio de orden; $L^{s5}=1$ es la latencia del desplazamiento de primer elemento; y $L^{s6}=6$ es la latencia del pre-filtro.

Asimismo, se recuerda que la latencia no tiene impacto en el diseño, solamente se emplea para tener una mejor definición de la arquitectura.

4.7.3. Obtención de los parámetros para la FFT y la DCT

Existen varias configuraciones de FFT que se pueden utilizar, primando ante todo la utilización del menor número de multiplicadores posible. Para generar la FFT se ha empleado un core que proporciona Spiral [111], el cual permite generar distintas arquitecturas con distintos ratios de paralelismo R_p . Como se ha realizado con los otros algoritmos, se ha elegido la FFT con un ratio de paralelismo $R_p=2$, de un tamaño de $M=512$ puntos cuyos parámetros más característicos son (78) y (79):

$$C_m^{FFT} = 32 \text{ multiplicadores} \quad (78)$$

$$M_c^{FFT} = 256 \text{ ciclos}, \quad L^{FFT} = 728 \text{ ciclos} \quad (79)$$

Estos datos permiten que la DCT cumpla con los límites calculados en (75) y (76).

Por otro lado, la fase s5 se puede unir con la fase s4 para ahorrar ciclos de reloj en la latencia y reducir el consumo de recursos. Con estos datos se puede obtener el número final de multiplicadores C_m^{DCT} empleados descrito en (72), según (80):

$$C_m^{DCT} = C_m^{s1} + C_m^{s3} + C_m^{s4} + C_m^{s6} = 8 + 32 + 8 + 2 = 50 \text{ multiplicadores} \quad (80)$$

Por otro lado, la latencia L^{DCT} queda definida por (81):

$$L^{DCT} = L^{s1} + L^{s2} + L^{s3} + L^{s4} + L^{s6} = 1262 \text{ ciclos} \quad (81)$$

Donde $L^{s1}=8$ es la latencia de la fase s1, debida a la multiplicación compleja (60); $L^{s2}=M$ es la latencia de la fase s2, debida a las memorias ping-pong (61); $L^{s3}=L^{FFT}=728$ es la latencia de la FFT (62); $L^{s4}=8$ es la latencia de la fase s4, debida a la multiplicación compleja (63), aunque también se realiza el desplazamiento de bits (64); y $L^{s6}=6$ es la latencia en la fase s6, debida al pre-filtro (multiplicación más acumulación, 65).

4.7.4. Arquitectura del transmisor

En el caso de esta arquitectura, no se puede unir el módulo de la extensión simétrica con ninguna de las fases de la DCT del algoritmo 3 desarrollada, por lo que el número de recursos y la latencia empleada aumentarán respecto a las anteriores DTT. A continuación en la Figura 44, se muestra el diagrama de bloques de la arquitectura de la DTT con la DCT del algoritmo 3 y el módulo de la extensión simétrica para el transmisor.

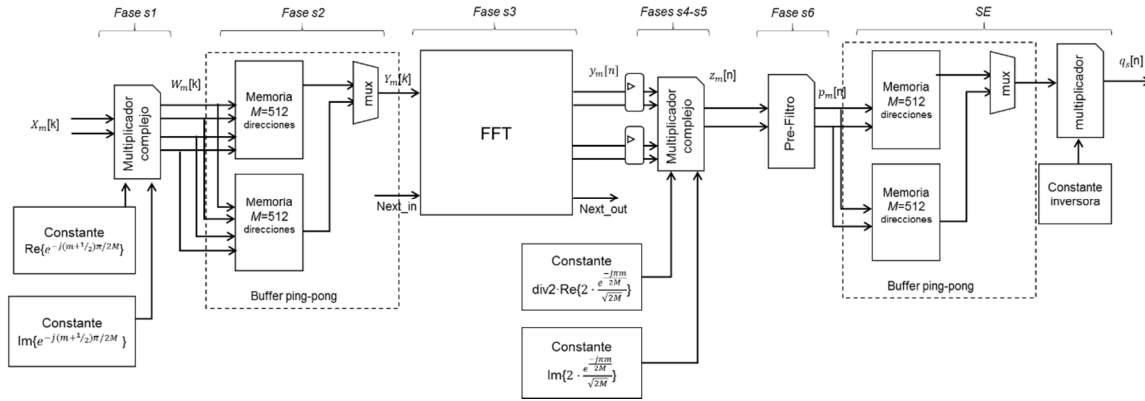


Figura 44. Diagrama de bloques de la arquitectura del transmisor DTT con la DCT del algoritmo 3.

Dado que el ratio de paralelismo se ha fijado en $R_p=2$, $C_m^{s1}=8$ multiplicadores son necesarios en la fase s1 (60). Además, la latencia es de $L^{s1}=8$ ciclos de reloj. Después, la segunda fase s2 de la DCT del algoritmo 3 es un reordenamiento de la trama de datos (61), donde todas las muestras de la trama de datos $W_m[k]$ deben estar disponibles al mismo tiempo. Por lo tanto, los datos deben ser almacenados en memoria, para que sean leídos en el orden correcto. Para evitar pérdida de datos, se han empleado dos memorias en modo ping-pong, por lo que en una se puede acceder para organizar los datos en el orden correcto, mientras que los nuevos datos se escriben en la otra. Esta fase implica una latencia de $L^{s2}=M=512$ ciclos.

La tercera fase s3 es una FFT que requiere no sólo multiplicadores y sumadores, sino también memoria donde almacenar los valores intermedios (62). Este módulo se ha parametrizado con un tamaño de $M=512$ muestras y una representación en coma fija de 18 bits, valor fijado por el ancho de palabra que proporcionan una de las entradas de los multiplicadores de la FPGA. La FFT presenta una arquitectura radix-2 semi-paralela, con $\log_2 M$ bloques en cascada. De nuevo, la selección del ratio de paralelismo es $R_p=2$, que

implica que sólo dos muestras de entrada $Y_m[k]$ son procesadas cada ciclo de reloj. Una FFT se completa cada $M_c^{FFT}=256$ ciclos de reloj, con una latencia $L^{FFT}=728$ ciclos.

La cuarta fase $s4$ del módulo DCT es similar a la primera fase $s1$ (63), donde $C_m^{s4}=8$ multiplicadores son usados y la latencia es $L^{s4}=8$ ciclos de reloj. Además esta fase se fusiona con la quinta fase $s5$ (64), que realiza una división entre dos del primer elemento de la trama de datos. En el caso de la sexta fase $s6$ de la DCT es un pre-filtrado de la señal $z_m[n]$ (65), con la función $H_z = \frac{2}{[z^{-1}+1]}$. Esta fase implica una latencia de $L^{s6}=6$ ciclos.

Como se puede observar, las seis fases que forman el módulo DCT están segmentadas. En consecuencia, como el throughput de cada datapath es de $T_{ch}=1$ dato por cada ciclo de reloj y el ratio de paralelismo es $R_p=2$, el throughput global de la DCT es $T_c^{DCT} = T_{ch} \cdot R_p=2$. Con todo esto, todas las muestras de entrada $X_m[k]$ son procesadas cada $M_c^{DCT}=256$ ciclos. Como en las anteriores propuestas, la arquitectura cumple con la restricción de procesamiento máximo $M_c^{max}=512$ ciclos de reloj.

Por lo tanto, el número de multiplicaciones C_m^{Tx} empleadas para la realización de la arquitectura del transmisor de la DTT con la DCT del algoritmo 3 y el módulo de la extensión simétrica es (82):

$$C_m^{Tx} = C_m^{s1} + C_m^{s3} + C_m^{s4} + C_m^{s6} + C_m^{SE,Tx} = 8 + 32 + 8 + 2 + 1 = 51 \text{ multiplicadores} \quad (82)$$

Donde $C_m^{s1}=8$ es el número de multiplicadores en la fase $s1$ (60); $C_m^{s3}=28$ es el número de multiplicadores empleados en la FFT (62); $C_m^{s4}=8$ es el número de multiplicadores empleados en la fase $s4$ (63); $C_m^{s6}=2$ es el número de multiplicadores empleados en la fase $s6$ (65); y $C_m^{SE,Tx}=1$ es el número de multiplicadores empleados en el módulo de la extensión simétrica (8).

La latencia L^{Tx} de la arquitectura propuesta para la DCT del algoritmo 3 y el módulo de extensión simétrica es obtenida en (83):

$$L^{Tx} = L^{s1} + L^{s2} + L^{s3} + L^{s4} + L^{s6} + L^{SE,Tx} = 1776 \text{ ciclos} \quad (83)$$

Donde $L^{s1}=8$ es la latencia de fase $s1$, debida a la multiplicación compleja (60); $L^{s2}=512$ es la latencia de la fase $s2$ (61), debida a las memorias ping-pong; $L^{s3}=L^{FFT}=728$ es la latencia de la FFT (62); $L^{s4}=8$ es la latencia de la fase $s4$, debida a la multiplicación compleja (63), aunque también contiene a la fase $s5$ (64); $L^{s6}=6$ es la latencia de fase $s6$, debida al pre-filtro (multiplicación más acumulación, 65); y $L^{SE,Tx}=514$ es la latencia del módulo de la extensión simétrica (9), debida a las memorias ping-pong.

En la Figura 45 se puede observar el cronograma de la arquitectura realizada en el transmisor, que incluye la DCT del algoritmo 3 y el módulo de extensión simétrica. Primero, se recibe la señal $X_m[k]$, sincronizada con la señal CE_M (muestra de reloj 1). Esta señal CE_M activa la señal de enable y hace que se comiencen a generar direcciones para la memoria que almacena los coeficientes $e^{-j(m+1/2)\pi/2M}$ de la multiplicación (60) (muestra

4.7 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 3

de reloj 1). De esta multiplicación se obtiene $W_m[k]$ (muestra de reloj 7), y a esta señal se le realiza un cambio en el orden de la trama de datos para obtener $Y_m[k]$ (muestra de reloj 9). La señal *next_in* se activa para indicar a la FFT el comienzo de una nueva trama de datos (muestra de reloj 9).

Una vez la FFT se ha producido, se activa la señal *next_out* y a continuación, se comienzan a entregar la señal $y_m[n]$ (muestra de reloj 13). También se comienzan a proporcionar los coeficientes $2 \cdot \frac{e^{-\frac{j\pi n}{2M}}}{\sqrt{2M}}$ para la multiplicación 2 (63) (muestra de reloj 15). En esta operación también se realiza la división de la fase s5 (64). Con todo esto, se obtiene la señal $z_m[n]$ (muestra de reloj 19), la cual se filtra, para obtener $p_m[n]$ (muestra de reloj 25). Por último, se añade la extensión simétrica, obteniendo $q_s[n]$ (muestra de reloj 26).

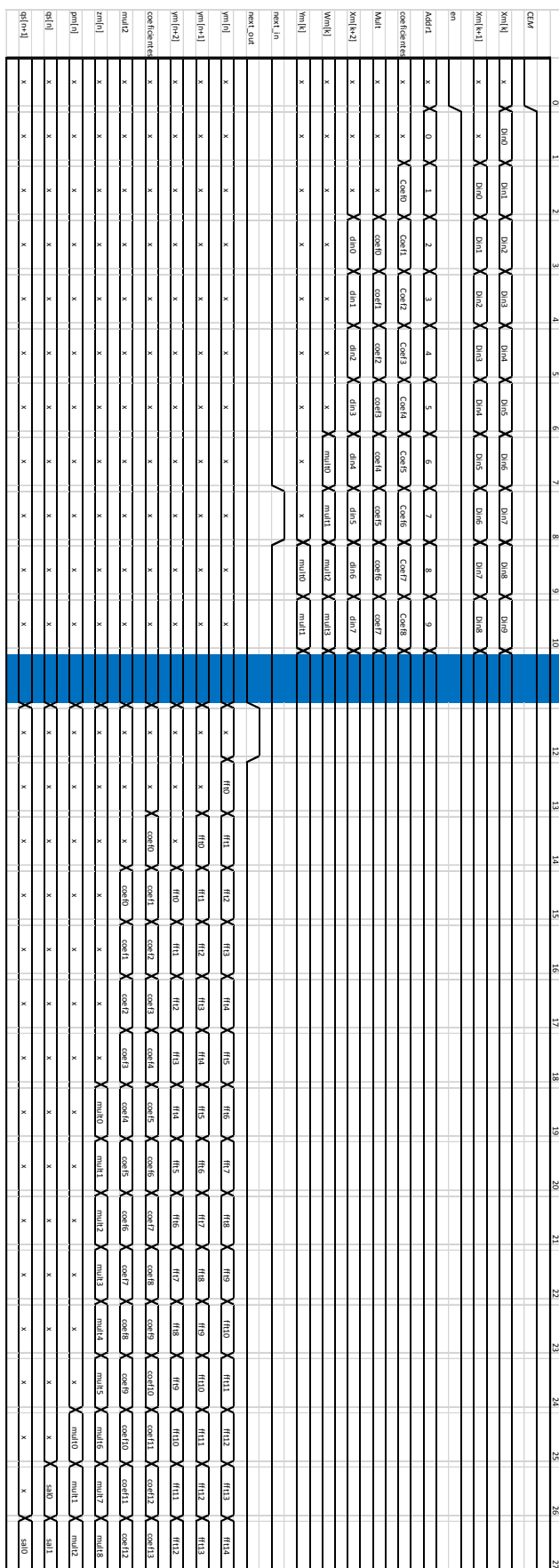


Figura 45. Cronograma de la arquitectura del transmisor DTT con la DCT del algoritmo 3.

4.7.5. Arquitectura del receptor

En el receptor no se puede unir el módulo de la extensión simétrica con ninguna de las fases de la DCT del algoritmo 3. Por lo tanto, el diagrama de bloques queda como se muestra en la Figura 46.

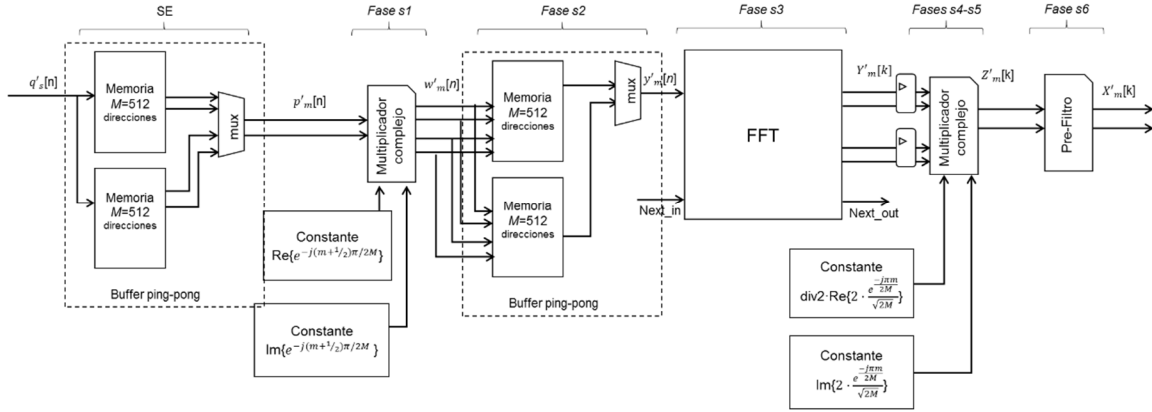


Figura 46. Diagrama de bloques de la arquitectura del receptor DTT con la DCT del algoritmo 3.

El ratio de paralelismo se ha fijado en $R_p=2$. Y debido a que la DCT implementada es ortogonal, sigue el mismo proceso de arquitectura que la DCT del transmisor. Por lo tanto, el número de multiplicaciones C_m^{Rx} empleadas para la realización de la arquitectura del receptor de la DTT con la DCT del algoritmo 3 y el módulo de la extensión simétrica es (84):

$$C_m^{Rx} = C_m^{s1} + C_m^{s3} + C_m^{s4} + C_m^{s6} = 8 + 32 + 8 + 2 = 50 \text{ multiplicadores} \quad (84)$$

Donde $C_m^{s1}=8$ es el número de multiplicadores en la fase s1 (60); $C_m^{s3}=32$ es el número de multiplicadores empleados en la FFT (62); $C_m^{s4}=8$ es el número de multiplicadores empleados en la fase s4 (63); $C_m^{s6}=2$ es el número de multiplicadores empleados en la fase s6 (65); y $C_m^{SE,Rx}=0$ es el número de multiplicadores empleados en el módulo de la extensión simétrica.

La latencia L^{Rx} de la arquitectura propuesta para la DCT del algoritmo 3 y el módulo de la extensión simétrica es obtenida en (85):

$$L^{Rx} = L^{s1} + L^{s2} + L^{s3} + L^{s4} + L^{s6} + L^{SE,Rx} = 1774 \text{ ciclos} \quad (85)$$

Donde $L^{s1}=8$ es la latencia de fase s1, debida a la multiplicación compleja (60); $L^{s2}=512$ es la latencia de la fase s2 (61), debida a las memorias ping-pong; $L^{s3}=L^{FFT}=728$ es la latencia de la FFT (62); $L^{s4}=8$ es la latencia de fase s4, debida a la multiplicación compleja (63), que también contiene a la fase s5 (64); $L^{s6}=6$ es la latencia de la fase 6, debida al pre-filtro (multiplicación más acumulación, 75); y $L^{SE,Rx}=512$ es la latencia del módulo de la extensión simétrica (10), debida a las memorias ping-pong.

Como se observa en el cronograma de la Figura 47, la señal $q'_s[n]$ llega sincronizada con CE_M (muestra de reloj 1). A partir de esta señal, se obtiene $p'_m[n]$ (muestra de reloj 2). Junto con esta señal, se comienzan a proporcionar los coeficientes $e^{-j(m+1/2)\pi/2M}$ de la

primera multiplicación (60) (muestra de reloj 3). De esta multiplicación se obtiene la señal $w'_m[n]$ (muestra de reloj 7), la cual recibe un reordenamiento de la trama de datos, generando $y'_m[n]$ (muestra de reloj 9). Esta señal se introduce en la FFT, sincronizada con la señal $next_in$.

Una vez que se ha producido la FFT, se activa la señal $next_out$ para indicar que se va a comenzar a proporcionar la señal $Y'_m[k]$ (muestra de reloj 13). Esta señal se introduce en el multiplicador, junto con los coeficientes $2 \cdot \frac{e^{-j\pi m}}{\sqrt{2M}}$ para la multiplicación 2 (63) (muestra de reloj 15). En esta operación también se realiza la división de la fase s_5 (64). Por último, la señal generada $Z'_m[k]$ (muestra de reloj 19) es filtrada, obteniendo la señal de salida $X'_m[k]$ (muestra de reloj 25).

4.7 Diseño de la arquitectura de la técnica de acceso basada en la DCT del algoritmo 3

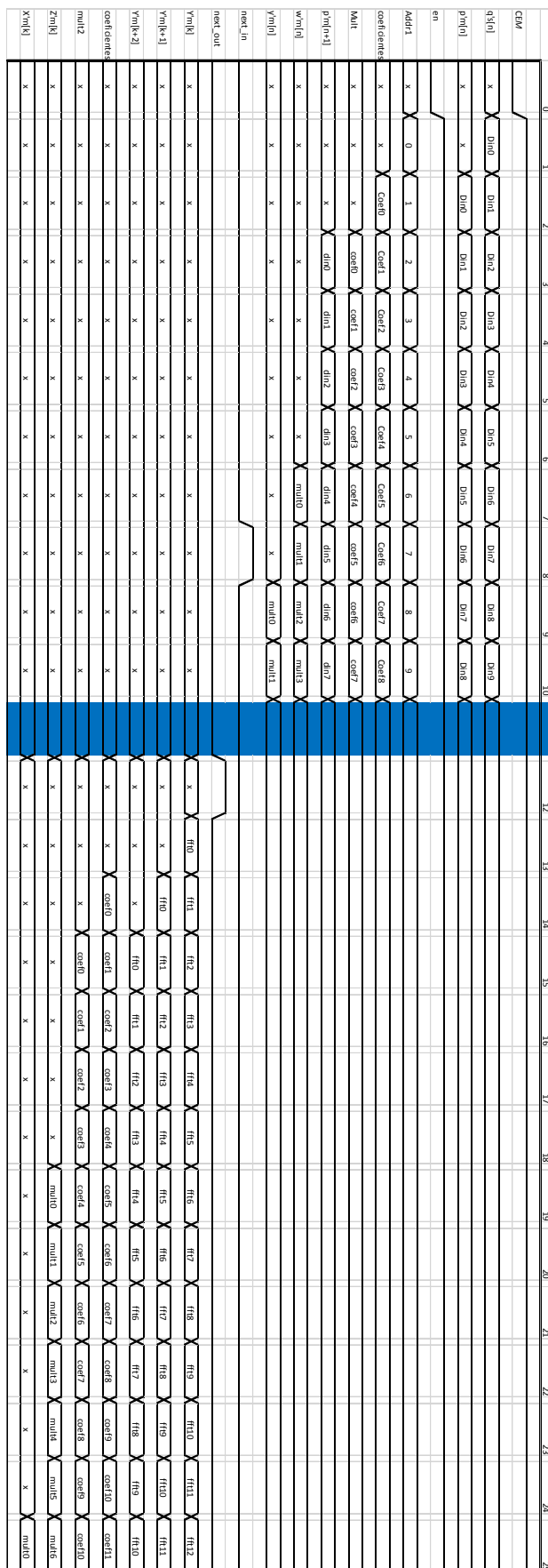


Figura 47. Cronograma de la arquitectura del receptor DIT con la DCT del algoritmo 3.

4.8. Comparativa de los tres algoritmos desarrollados en la técnica basada en Transformadas Trigonométricas Discretas

Después de realizar una explicación detallada de las tres posibles DCT que se pueden implementar, y de completar un exhaustivo estudio de cada una de ellas en términos de consumo de recursos, latencia, efecto de la precisión finita y arquitectura a desarrollar, se procede a mostrar una comparativa de los tres algoritmos empleados para la arquitectura de la DCT dentro del diseño de la arquitectura de la DTT. Primero, se realiza una comparativa del consumo de recursos de cada una de las aproximaciones evaluadas; después, se realiza una comparativa del error en la salida del transmisor; más tarde, en la salida del receptor; luego, se muestra el error comparando la salida del sistema completo con la entrada, y por último, se muestra los valores de SNR obtenidos para cada una de las implementaciones.

4.8.1. Comparativa de consumo de recursos

Se procede a reunir el consumo de multiplicadores empleados por cada una de las aproximaciones empleadas, ya que estos valores han sido calculados particularmente en cada apartado. Por tanto, se puede observar en la Tabla 14 el consumo de multiplicadores de cada aproximación, tanto en el transmisor, como en el receptor.

Tabla 14. Comparativa de consumo de multiplicadores por los tres algoritmos implementados.

	<i>transmisor</i>	<i>receptor</i>
DCT del algoritmo 1	49	48
DCT del algoritmo 2	45	44
DCT del algoritmo 3	51	50

Se observa que en los tres algoritmos se emplea un multiplicador más en la transmisión que en la recepción. Esto es debido a que ese multiplicador adicional se corresponde con el empleado para realizar la extensión simétrica. A su vez, se observa que la técnica que menos multiplicadores utiliza es la arquitectura con la DCT del algoritmo 2.

4.8.2. Comparativa salida del transmisor

En esta comparativa se estudia el error absoluto de cada arquitectura del transmisor propuesto con su arquitectura ideal. En la Tabla 15 se observa los valores obtenidos por cada uno de los algoritmos. Estos valores provienen de distintas tablas que se han presentado a lo largo de este apartado. En ellas se puede observar una tabla con el error medio, absoluto y desviación típica, siendo en la Tabla 2 los valores obtenidos para la DCT del algoritmo 1, en la Tabla 7 los valores obtenidos para la DCT del algoritmo 2, y en la Tabla 11 los valores obtenidos para la DCT del algoritmo 3.

Tabla 15. Comparativa de errores obtenidos por los tres algoritmos implementados en el transmisor.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
algoritmo 1	[18 12]	$2.3000 \cdot 10^{-3}$	$6.2700 \cdot 10^{-2}$	$3.3000 \cdot 10^{-3}$
algoritmo 2	[18 12]	$3.0470 \cdot 10^{-4}$	$1.5000 \cdot 10^{-3}$	$2.6470 \cdot 10^{-4}$
algoritmo 3	[18 12]	$3.8200 \cdot 10^{-2}$	$1.5410 \cdot 10^{-1}$	$2.6700 \cdot 10^{-2}$

4.8 Comparativa de los tres algoritmos desarrollados en la técnica basada en la DTT

Como se observa en estos resultados, la comparativa en la salida del transmisor dictamina que la arquitectura con la DCT del algoritmo 2 es la que mejores errores obtiene. La principal diferencia en los errores obtenidos se debe a que la DCT del algoritmo 2 realiza una iFFT de la mitad de elementos, lo cual implica que el peor caso tenga una amplitud menor, y por tanto, el número de bits empleados en la parte entera sea menor. Esto permite que se utilicen en las operaciones intermedias de la DCT más bits en la parte fraccionaria.

4.8.3. Comparativa salida del receptor

En este caso, se realiza la comparativa de los errores absolutos de los receptores implementados con su arquitectura ideal. Estos errores se han mostrado anteriormente en los estudios de efectos de precisión finita de los receptores. A continuación, se muestra una tabla en la que se pueden observar el error absoluto, máximo y desviación típica de cada uno de los algoritmos de DCT empleados en el receptor. Estos valores provienen de las tablas aportadas en cada apartado, siendo en la Tabla 4 los valores obtenidos para la algoritmo 1, en la Tabla 9 los valores obtenidos para la DCT del algoritmo 2, y en la Tabla 13 los valores obtenidos para la DCT del algoritmo 3.

Tabla 16. Comparativa de errores obtenidos por los tres algoritmos implementados en el receptor.

		<i>Error absoluto medio</i>	<i>Error absoluto máximo</i>	<i>Desviación típica</i>
algoritmo 1	[18 16]	$1.4216 \cdot 10^{-5}$	$4.7302 \cdot 10^{-4}$	$2.4604 \cdot 10^{-5}$
algoritmo 2	[18 16]	$2.8744 \cdot 10^{-4}$	$1.6000 \cdot 10^{-3}$	$2.5947 \cdot 10^{-4}$
algoritmo 3	[18 16]	$1.8200 \cdot 10^{-4}$	$1.4000 \cdot 10^{-3}$	$1.8672 \cdot 10^{-4}$

Como se observa en la Tabla 16, la comparativa en la salida del receptor dictamina que la arquitectura con la DCT del algoritmo 1 es la que menores errores obtiene. Esto es debido a que, la DCT del algoritmo 1 es capaz de aprovechar mejor la cuantificación de la FFT, para los datos de entrada que se proporcionan en el receptor.

4.8.4. Comparativa salida del sistema implementado

Una vez se han mostrado el error absoluto que se genera en el transmisor y en el receptor de cada arquitectura, se procede a mostrar el error obtenido a la salida del sistema. Para ello se ha desarrollado una simulación del sistema completo trabajando en un canal ideal, sin ruido externo y empleando un canal de conversión de 16 bits. Se han realizado 1000 pruebas en la simulación con una entrada aleatoria entre $[-1,+1]$ y el resultado es la obtención del error medio absoluto del sistema para cada subportadora. En la Figura 48 se muestra el error absoluto para cada una de las subportadoras del sistema empleando la DCT del algoritmo 1.

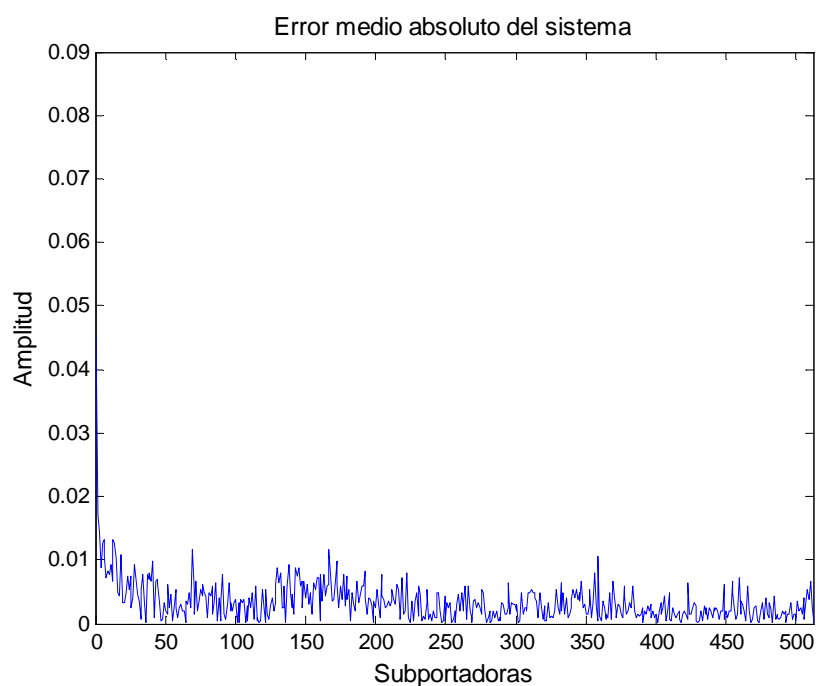


Figura 48. Error medio absoluto del sistema con la DCT del algoritmo 1, para una entrada aleatoria entre $[-1,1]$.

Con estos valores absolutos se obtiene un error medio de 0.0033, para el caso de la arquitectura de la DCT del algoritmo 1. En la Figura 49 se muestra el error medio absoluto para cada subportadora del sistema empleando la DCT del algoritmo 2.

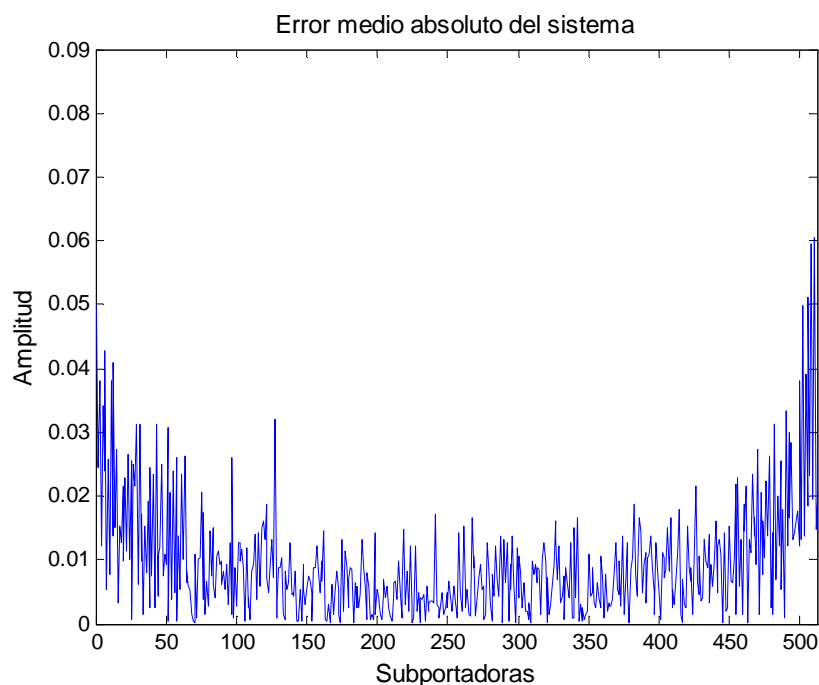


Figura 49. Error medio absoluto del sistema con la DCT del algoritmo 2, para una entrada aleatoria entre $[-1,1]$.

Con estos valores absolutos se obtiene un error medio de 0.0095, para el caso de la arquitectura de la DCT del algoritmo 2. En la Figura 50 se muestra el error absoluto para cada subportadora del sistema empleando la DCT del algoritmo 3.

4.8 Comparativa de los tres algoritmos desarrollados en la técnica basada en la DTT

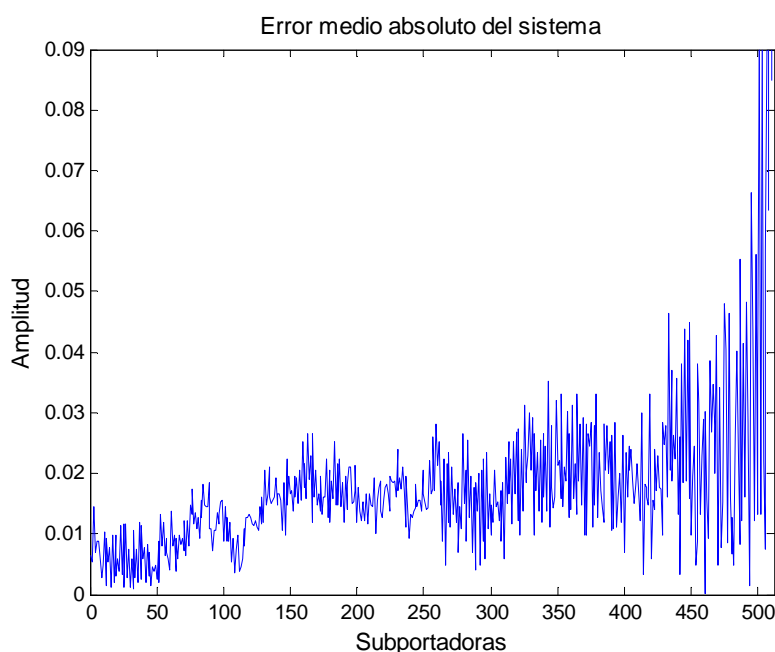


Figura 50. Error medio absoluto del sistema con la DCT del algoritmo 3, para una entrada aleatoria entre $[-1,1]$.

Con estos valores absolutos se obtiene un error medio de 0.0213 para el caso de la arquitectura de la DCT del algoritmo 3.

Por lo tanto, en el sistema completo la implementación con la DCT del algoritmo 1 obtiene unos errores más bajos que las otras dos implementaciones. Este hecho se produce debido a la cuantificación empleada en el receptor, que permite que el error en el receptor sea 10 veces más pequeño que en las otras dos implementaciones.

4.8.5. Comparativa de prestaciones

Por último, se va a mostrar en la Tabla 17 los valores obtenidos para el sistema con las tres implementaciones realizadas. Los sistemas están formados por los tres tipos de algoritmos de la DCT y el módulo de la extensión simétrica. Aunque, el módulo SE no añade ningún tipo de error por el efecto de la precisión finita. Los valores calculados son el SNR medio, el error cuadrático medio, el SNR máximo y el máximo error.

Tabla 17. Parámetros obtenidos de simulación del DTT cuantificado con distintas DCT, para el caso con canal ideal, sin ruido y con conversor de 16 bits.

	<i>SNR medio</i>	<i>Error cuadrático medio</i>	<i>SNR máximo</i>	<i>Error máximo</i>
algoritmo 1	43.4817 dB	$4.4940 \cdot 10^{-5}$	43.4953 dB	0.0632
algoritmo 2	34.9841 dB	$3.1748 \cdot 10^{-4}$	34.9846 dB	0.0823
algoritmo 3	19.8667 dB	$1.3700 \cdot 10^{-2}$	19.9075 dB	2.2737

Como se puede observar de la Tabla 17, la arquitectura que obtiene un mejor SNR es la que contiene a la DCT del algoritmo 1. Fundamentalmente, obtiene un mejor SNR que los otros dos algoritmos gracias a aprovechar de una manera más eficiente la cuantificación en el receptor. En cambio la arquitectura de la DCT del algoritmo 3, se ve perjudicada por el empleo de un pre-filtro a la salida de la DCT que introduce más ruido.

4.9. Arquitectura de la Transformada Trigonométrica Discreta para banda estrecha

Una vez se ha realizado el estudio comparativo entre las distintas implementaciones de la DCT para banda ancha, obteniendo en la comparativa que la DCT del algoritmo 1 es la que obtiene mejores resultados, se procede a mostrar los valores obtenidos para la arquitectura de la DTT, empleando la DCT del algoritmo 1 para banda estrecha. El empleo de banda estrecha implica que el número de subcanales queda reducido a $M=256$ y la frecuencia de transmisión reducida a $f_{Tx}=500\text{kpsps}$. Este cambio en los parámetros implica que el efecto de la precisión finita mejora, debido a que hay que modificar la cuantificación para adaptarse a un menor número de subcanales; y además, se dispone de más ciclos de reloj para realizar las operaciones, debido a la menor tasa de transmisión.

La arquitectura del sistema para banda estrecha es similar a la arquitectura empleada para banda ancha. Por lo tanto, como ya se ha realizado todo el estudio de la arquitectura para la DCT del algoritmo 1 para banda ancha, en este apartado sólo se realizará un resumen de las características de la DTT, empleando la DCT del algoritmo 1.

4.9.1. Estudio de la precisión finita

La Tabla 18 muestra la representación en coma fija definida para cada módulo y señal en el diseño propuesto, no sólo para el transmisor sino también para el receptor. El ancho de palabra más común en el sistema es de 18 bits, fijado por el ancho de palabra de las celdas multiplicadoras (DSP48E1) disponibles en la FPGA.

Tabla 18. Representación en coma fija del ancho de palabra para la arquitectura propuesta DTT.

Módulo	Parámetro	No. bits global	Bits parte fraccional
Entrada Tx	$X_m[k]$	18	16
	Entrada $X_m[k]$	18	16
Tx DCT	Constante $e^{-jm\pi/2M}$	25	23
	Multiplicador eq. (59)	36	32
	$Y_m[k]$	18	16
	$Y_m[k_0]$	18	9
	$y_m[n_0]$	18	9
	$z_m[n]$	18	9
	Constante $e^{-j\pi(2m+1)/4M}$	25	23
	Multiplicador eq. (62)	36	32
Módulo SE	Entrada $p_m[n]$	18	13
	Salida $q_s[n]$	18	13
Salida Tx	Constante $1/\sqrt{2}$	25	23
	Salida $e[n]$	18	13
Entrada Rx	Entrada $r[n]$	18	16
	Constante $1/\sqrt{2}$	25	23
	Salida $q'_s[n]$	18	16
Módulo SE	Entrada $q'_s[n]$	18	16
	Salida $p'_m[n]$	18	16
Rx DCT	Entrada $p'_m[n]$	18	16
	Constante $e^{-jm\pi/2M}$	25	23
	Multiplicador eq. (59)	36	32

	$y_m[n]$	18	16
	$y_m[n_0]$	18	16
	$Y_m[k_0]$	18	15
	$Z_m[k]$	18	15
	Constante $e^{-j\pi(2m+1)/4M}$	25	23
	Multiplicador eq. (62)	36	31
	Salida $X'_m[k]$	18	16
Salida Rx	$X'_m[k]$	18	16

Con estos valores de tamaño de ancho de palabra, los diferentes módulos del sistema tienen que ser evaluados y comparados con un modelo en coma flotante. Las simulaciones se han llevado a cabo, proporcionando una señal de entrada aleatoria $X_m[k]$, de 1000 muestras de longitud y normalizada al máximo rango de entrada $[-1,+1]$, para cada uno de los $M=256$ canales de entrada. La Tabla 19 describe el error medio y la desviación estándar para algunas señales intermedias de la arquitectura. Como puede observarse, desde un punto de vista global, el error medio absoluto es de $1.3096 \cdot 10^{-4}$ para el transmisor, mientras que en el receptor es de 0.0031. Es importante destacar que la FFT es la que tiene una influencia más remarcable en las figuras finales.

Tabla 19. Error medio y desviación estándar para las señales intermedias en la arquitectura propuesta DTT, asumiendo una rango de entrada de $[-1,+1]$ en $X_m[k]$.

Módulo	Parámetro	error medio	desviación estándar
Tx DCT	$p_m[n]$	$1.3096 \cdot 10^{-4}$	$2.0792 \cdot 10^{-4}$
Módulo SE	$q_s[n]$	0	0
Salida Tx	$e[n]$	$1.3096 \cdot 10^{-4}$	$2.0792 \cdot 10^{-4}$
Entrada Rx	$r[n]$	0	0
Módulo SE	$q'_s[n]$	0	0
Rx DCT	$X'_m[k]$	$3.1000 \cdot 10^{-3}$	$2.8000 \cdot 10^{-3}$

4.9.2. Consumo de recursos

El número de multiplicaciones C_m^{Tx} empleadas para la realización de la arquitectura del transmisor de la DTT con la DCT del algoritmo 1 queda modificado respecto al calculado en (33), según (86):

$$C_m^{Tx} = C_m^{s1} + C_m^{FFT} + C_m^{s4} + C_m^{SE,Tx} = 8 + 4 + 8 + 1 = 21 \text{ multiplicadores} \quad (86)$$

Donde $C_m^{s1}=8$ es el número de multiplicadores en la fase s1 (12); $C_m^{FFT}=4$ es el número de multiplicadores empleados en la FFT (13); $C_m^{s4}=8$ es el número de multiplicadores empleados en la fase s4 (15); y $C_m^{SE,Tx}=1$ es el número de multiplicadores empleados en el módulo de la extensión simétrica (8).

El empleo de un número menor de multiplicadores en la FFT se debe a que, al disponer de un mayor número de ciclos de reloj en el tiempo de procesamiento para la arquitectura, gracias a trabajar en banda estrecha, se ha decidido emplear un diseño basado en una arquitectura iterativa. Esta arquitectura posee una mayor latencia y tiempo de procesamiento, pero a cambio necesita emplear un número menor de multiplicadores.

El número de multiplicaciones C_m^{Rx} empleados para la realización de la arquitectura del receptor de la DTT con la DCT del algoritmo 1 también se modifica respecto al calculado en (35), según (87):

$$C_m^{Rx} = C_m^{s1} + C_m^{FFT} + C_m^{s4} + C_m^{SE,Rx} = 8 + 4 + 8 + 0 = 20 \text{ multiplicadores} \quad (87)$$

Donde $C_m^{s1}=8$ es el número de multiplicadores en la fase s1 (12); $C_m^{FFT}=4$ es el número de multiplicadores empleados en la FFT (13); $C_m^{s4}=8$ es el número de multiplicadores empleados en la fase s4 (15); y $C_m^{SE,Rx}=0$ es el número de multiplicadores empleados en el módulo de extensión simétrica.

Como ocurre en el transmisor, para el diseño de la FFT en el receptor se ha decidido emplear una arquitectura iterativa que reduce el número de multiplicadores utilizados.

4.9.3. Cálculo de la latencia

Aunque la latencia no tiene impacto en el diseño, se emplea para tener una mejor definición de la arquitectura desarrollada. En este caso, la latencia L^{Tx} del transmisor de la arquitectura propuesta para la DCT del algoritmo 1 y el módulo SE se modifica respecto a la calculada en (34), según (88):

$$L^{Tx} = L^{s1} + L^{s2} + L^{SE,Tx} + L^{s4} = 8 + 1157 + 514 + 8 = 1687 \text{ ciclos} \quad (88)$$

Donde $L^{s1}=8$ es la latencia de la fase s1 (12); $L^{s2}=L^{FFT}=1157$ es la latencia de la FFT (13); $L^{SE,Tx}=514$ es la latencia del módulo SE (9), que contiene también a la fase s3 (14); y $L^{s4}=8$ es la latencia de la fase s4 (15).

El incremento del número de ciclos se debe al empleo de la arquitectura iterativa en la FFT, que aumenta la latencia a $L^{FFT}=1157$ ciclos.

La latencia L^{Rx} del receptor de la arquitectura propuesta para la DCT del algoritmo 1 y el módulo SE también se modifica respecto a (36), según (89):

$$L^{Rx} = L^{SE,Rx} + L^{s1} + L^{s2} + L^{s3} + L^{s4} = 512 + 8 + 1157 + 512 + 8 = 2197 \text{ ciclos} \quad (89)$$

Donde $L^{SE,Rx}=512$ es la latencia del módulo SE (10); $L^{s1}=8$ es la latencia de la fase s1 (12); $L^{s2}=L^{FFT}=1157$ es la latencia de la FFT (13); $L^{s3}=512$ es la latencia de la fase s3 (14); y $L^{s4}=8$ es la latencia de la fase s4 (15).

Como sucede en el transmisor, el aumento del número de ciclos se debe al empleo de la arquitectura iterativa en la FFT, que aumenta la latencia a $L^{FFT}=1157$ ciclos.

4.9.4. Resultado SNR

En el apartado de resultados, se va a realizar una comparación entre el modelo en coma flotante y el modelo en coma fija, teniendo en cuenta la relación señal-ruido (SNR) del transmultiplexor, el error cuadrático medio, la relación señal-ruido de pico y el error

4.10 Conclusiones

máximo absoluto como se define en [112]. La Tabla 20 muestra estas figuras de mérito, y también para una simulación con una entrada aleatoria $X_m[k]$ entre $[-1,+1]$ de 1000 muestras de longitud y una amplitud normalizada al máximo valor de entrada.

Tabla 20. Comparación entre los modelos en coma fija y coma flotante de la arquitectura propuesta para DTT de banda estrecha.

	<i>Coma flotante</i>	<i>Coma fija</i>
<i>SNR</i>	301.28 dB	49.7527 dB
<i>error cuadrático medio</i>	$1.1638 \cdot 10^{-31}$	$1.0599 \cdot 10^{-5}$
<i>Pico SNR</i>	309.3471 dB	49.7504 dB
<i>Error absoluto máximo</i>	$8.9689 \cdot 10^{-16}$	$2.4800 \cdot 10^{-2}$

Como se observa, al emplear la DTT para banda estrecha, se ha mejorado los resultados obtenidos respecto a banda ancha. Esto se debe a que para banda estrecha se ha empleado un número menor de subcanales, que influye directamente en la cuantificación de la FFT, aumentando los bits en la parte fraccionaria. El hecho de disponer de un número mayor de bits en la parte fraccionaria reduce el error debido a la precisión finita.

4.10. Conclusiones

Para el estudio de la técnica multi-portadora basada en transformadas trigonométricas se han establecido tres posibilidades de arquitectura de la Transformada del Coseno Discreta (DCT). Estas transformadas han sido analizadas en cuanto a consumo de recursos, efecto de la precisión finita y latencia necesaria, obteniendo que la DCT del algoritmo 1 es la arquitectura más eficiente a realizar en la FPGA seleccionada.

Como se ha visto, gracias a la flexibilidad del diseño, éste se ha podido implementar tanto para banda estrecha, como para banda ancha. Esto se consigue modificando los parámetros de entrada y la frecuencia de funcionamiento del diseño para adaptarlo a la demanda solicitada.

Estos parámetros de entrada permiten modificar, por ejemplo, el ratio de paralelismo del sistema (R_p), lo cual indica el número de datapaths que se procesan simultáneamente. Este parámetro afecta también al consumo de recursos, ya que al seleccionar más datapaths, aumenta el número de recursos necesarios para realizarlo.

Por otro lado, también se puede seleccionar el número M de subcanales de entrada, así como, el número de elementos que se repetirán con la extensión simétrica (SE). Gracias a lo cual, se puede ajustar el número según el medio en el que se quiera transmitir.

Por último, como se ha observado al diseñar tres tipos diferentes de arquitecturas de la DCT, se ha realizado un estudio exhaustivo del efecto de la precisión finita para minimizar su impacto sobre el error final que aparece en la arquitectura de la técnica basada en transformadas trigonométricas discretas.

Capítulo 5

Técnica de acceso al medio basada en bancos de filtros polifásicos

La técnica multi-portadora basada en banco de filtros (FBMC) se presenta como una alternativa real a las basadas en DFT [43]. FBMC presenta una serie de ventajas a tener en cuenta: mayor separación espectral de la información transmitida en cada subportadora y mayor robustez frente al ruido. Además al igual que la técnica de acceso anterior, emplea una DCT en lugar de un DFT. Esto hace que se produzca una mayor compactación de la energía, y junto al empleo de un banco de filtros, se genere mayor eficiencia espectral. A su vez, FBMC no necesita extensión simétrica para eliminar las interferencias ICI, ya que gracias al banco de filtros polifásico se reducen los lóbulos laterales de las subportadoras.

Gracias a estas características, FBMC es un serio candidato para la tecnología 5G [45] [113]. El sistema FBMC propuesto tiene su base en la capa física de banda base de wavelet OFDM definido en [26]. Los coeficientes de la respuesta impulsiva de los M canales de transmisión de los filtros son dados por (90), para el transmisor.

$$f_m[n] = \sqrt{\frac{2}{M}} \cdot h[n] \cdot \cos \left[\left(k + \frac{1}{2} \right) \frac{\pi}{M} \left(n + \frac{M+1}{2} \right) \right] \quad (90)$$

Donde $m=0 \dots M-1$.

Usando matrices, estos filtros en el dominio Z son (91):

$$\mathbf{f}^T(z) = [\mathbf{F}_0(z) \mathbf{F}_1(z) \dots \mathbf{F}_{M-1}(z)] \quad (91)$$

Que pueden expresarse como (92):

$$\mathbf{f}^T(z) = \mathbf{e}^T(z) \cdot [\mathbf{g}_0(z^{2M})z^{-M} \mathbf{g}_1(z^{2M})] \cdot \begin{bmatrix} (\mathbf{I} - \mathbf{J}) \\ (-\mathbf{I} - \mathbf{J}) \end{bmatrix} \cdot \mathbf{C}_{4e} \cdot \Theta \quad (92)$$

Donde Θ es una matriz diagonal de $(M \times M)$; \mathbf{C}_{4e} es una DCT-IV de M elementos; y \mathbf{g}_0 y \mathbf{g}_1 son matrices diagonales de $(M \times M)$ con los elementos de (93):

$$\begin{aligned} [\mathbf{g}_0(z)]_{l,l} &= \mathbf{G}_l(-z) \\ [\mathbf{g}_1(z)]_{l,l} &= \mathbf{G}_{l+M}(-z) \end{aligned} \quad (93)$$

Donde $\mathbf{G}_l(z^{2M})$, con $0 \leq l \leq (2M-1)$, son las $2M$ descomposiciones de los filtros tipo I polifásicos de la función del sistema prototipo $\mathbf{H}(z)$ [114].

Con el receptor, se ha seguido el siguiente esquema de modulación coseno (94):

$$h_m[n] = \sqrt{\frac{2}{M}} \cdot h[n] \cdot \cos \left[\left(k + \frac{1}{2} \right) \frac{\pi}{M} \cdot \left(N - 1 - n + \frac{M+1}{2} \right) \right] \quad (94)$$

Donde $N = 2048$ es la longitud del filtro prototipo.

La expresión de la matriz de los filtros del receptor $\mathbf{h}(z)$ es (95):

$$\mathbf{h}(z) = \Theta \cdot \mathbf{C}_{4e} \cdot [(\mathbf{I} + \mathbf{J}) (\mathbf{I} - \mathbf{J})] \cdot \begin{bmatrix} \mathbf{g}_0(z^{2M}) \\ z^{-M} \mathbf{g}_1(z^{2M}) \end{bmatrix} \cdot \mathbf{e}(z) \quad (95)$$

Una vez se ha introducido teóricamente la técnica multi-portadora basada en banco de filtros, se procede a definir la arquitectura que la compondrá. Dado que el número de subcanales será elevado, se ha pensado en procesar éstos de manera semi-paralela. Esto permite que todos los canales se procesen cumpliendo los tiempos de procesamiento para su ejecución en tiempo real y a su vez el número de recursos dedicados no se eleve demasiado. Para definir cuántos subcanales son procesados en paralelo, se define un ratio de paralelismo R_p . Este ratio se empleara en los diferentes cálculos para obtener tanto el número de recursos, como la latencia, tiempo de procesamiento y throughput de las distintas partes que formarán la técnica de acceso al medio. La introducción de un ratio de paralelismo R_p otorga a la arquitectura una flexibilidad que permite poder adaptar el diseño a cualquier tipo de dispositivo. Junto a la elección de distintos números de subcanales de entrada y la frecuencia de funcionamiento del sistema, permite que sea adaptable a cualquier entorno que se desee considerar.

Con estos parámetros configurables, se ha definido el transmisor y el receptor FBMC. El diseño consta de varios módulos en los que se podrán configurar estos parámetros, para definir el número de subcanales, el ratio de paralelismo y la frecuencia de funcionamiento. En la Figura 51 y Figura 52 se muestra el diagrama de bloques del transmisor y del receptor respectivamente. Siguiendo el diagrama de bloques del transmisor (Figura 51) se observa como la entrada $V_m[k]$ se introduce en el módulo multiplicador Θ . Después, la señal procesada $X_m[k]$ se conduce a la Transformada Discreta del Coseno de tipo IV (módulo DCT). Más tarde, cada señal de salida $p_m[n]$ del bloque DCT es duplicada e introducida en

un módulo de matrices. Después de esto, las $S=2 \cdot M$ salidas resultantes $q_s[n]$ son procesadas por el banco de filtros polifásicos y, finalmente, las señales $t_s[n]$ son sumadas y serializadas para obtener la señal $e[n]$, que será transmitida.

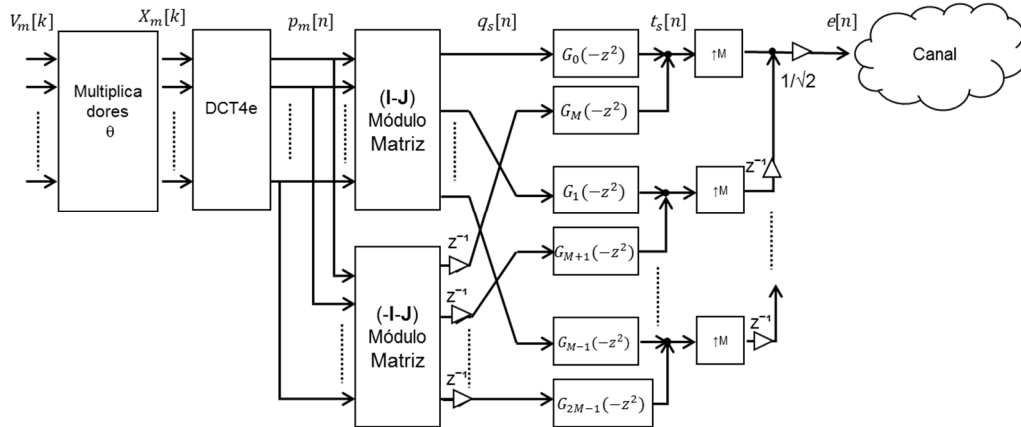


Figura 51. Diagrama de bloques del transmisor FBMC.

Por otro lado, en el diagrama de bloques de la Figura 52, la señal de entrada $r[n]$ se paraleliza obteniendo $S=2 \cdot M$ señales $t'_s[n]$. Después del banco de filtros polifásicos, el módulo Matriz procesa las señales $q'_s[n]$, y más tarde la suma en pares. A continuación, la señal resultante $p'_m[n]$ se dirige al módulo DCT. Finalmente, las señales $X'_m[k]$ se procesan en el módulo multiplicador Θ , para obtener la señal final de recepción $V'_m[k]$.

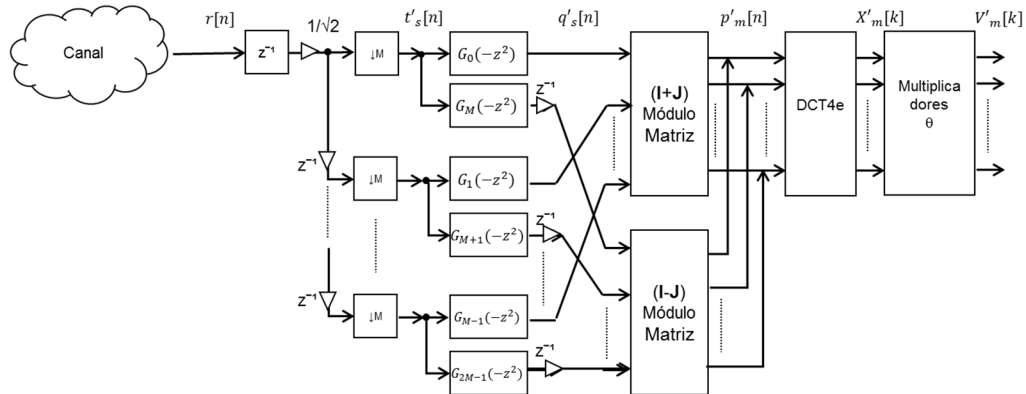


Figura 52. Diagrama de bloques del receptor FBMC.

A continuación, se va a realizar un estudio de las distintas partes que componen la arquitectura, abarcando los aspectos teóricos, efectos de la representación en coma fija, consumo de recursos y frecuencia de funcionamiento. Primeramente, se va a describir la DCT (Discrete Cosine Transform), básicamente constituida por una FFT (Fast Fourier Transform) y las multiplicaciones por los cosenos; después se analizarán las matrices de transformación **I-J**; y por último el banco de filtros polifásicos FIR. Una vez descrita la arquitectura de cada módulo, se procede a describir la arquitectura completa del sistema. Después se realizará un estudio del consumo de recursos, representación en coma fija y latencia del sistema completo.

Por último, se realizará un estudio para el diseño del banco de filtros polifásico, en el que se evaluarán distintas aproximaciones, realizando tanto un análisis del efecto de la precisión finita, como del consumo de recursos de cada una de las aproximaciones que se propongan.

5.1. Módulo multiplicación Θ

La multiplicación Θ consiste en un módulo que multiplica cada señal de entrada $V_m[k]$ por la constante Θ_m . Este módulo se describe matemáticamente según (96), donde m es el índice de cada uno de los M subcanales de entrada:

$$X_m[k] = \Theta_m \cdot V_m[k] \quad m = 0, \dots, M - 1 \quad (96)$$

La constante Θ_m está formada por 1 y -1, por lo que la acción que se realiza sobre la señal de entrada $V_m[k]$ es invertir la señal o dejarla igual.

El efecto de precisión finita que se comete al trabajar en coma fija, en este caso es nulo, puesto que la acción de invertir la señal de entrada no supone introducir ningún error.

Después se va a realizar una estimación de consumo de recursos para la realización de esta acción. Para ello, se va a considerar el empleo de celdas multiplicadoras DSP48E1 [109] y de sumadores. Como se observa en (96), sólo se realizan operaciones de multiplicación, por lo que el empleo de sumadores es nulo.

Para el cálculo de multiplicadores se va a emplear el ratio de paralelismo del diseño R_p . Este ratio indica el número de subcanales de entrada que son procesados en paralelo, por lo que el gasto de multiplicadores C_m^{theta} en el módulo Θ queda definido por (97):

$$C_m^{theta} = R_p \quad (97)$$

Pese a que la latencia no tiene impacto en el diseño, se ha decidido calcular para tener una mejor definición de la arquitectura desarrollada. La latencia L^{theta} que presenta en el módulo Θ se corresponde con (98):

$$L^{theta} = 2 \quad (98)$$

Que se corresponde con 2 ciclos de reloj, para realizar la multiplicación.

5.2. Transformada Discreta del Coseno

El siguiente módulo a implementar es la DCT (Discrete Cosine Transform); este módulo consiste en una DCT de tipo IV que incluye una FFT [106] [107] [108]. La DCT que se va a desarrollar es la misma que la empleada en el apartado 4.2. Se recuerda que esta transformada se define por (99):

$$p[n] = \sum_{k=0}^{M-1} X[k] \cos \left[\frac{\pi}{M} \left(n + \frac{1}{2} \right) \left(k + \frac{1}{2} \right) \right] \quad k = 0, \dots, M - 1 \quad (99)$$

Donde $p[n]$ es la salida de la señal de la DCT; M es el número de puntos de entrada; y $X[k]$ es la señal de entrada.

Tomando como referencia (99), el módulo DCT puede dividirse en cuatro fases:

- a) Generación de la secuencia $Y_m[k]$ a partir de la señal de entrada dada $X_m[k]$ multiplicando por la constante $e^{-\frac{j\pi k}{2M}}$, según (100):

$$Y_m[k] = X_m[k] \cdot e^{-\frac{j\pi k}{2M}} \quad (100)$$

- b) Obtención de la señal $y_m[n]$ mediante la FFT de $Y_m[k]$ según (101):

$$y_m[n] = F\{Y_m[k]\} \quad (101)$$

Donde el operador $F\{\}$ realiza una FFT de M puntos.

- c) Reordenación de la señal $y_m[n]$, obteniendo la señal $z_m[n]$ según (102):

$$\begin{aligned} z_{2i+1}[n] &= y_i[n] \\ z_{2i}[n] &= \text{conj}\{y_{m-i}[n]\} \end{aligned} \quad (102)$$

Donde $i = 0, \dots, \frac{M}{2} - 1$; $y_i[n]$ es la salida de la FFT; y el operador $\text{conj}\{\}$ es el conjugado de la señal.

- d) Generación de la secuencia $p_m[n]$ a partir de la señal $z_m[n]$ multiplicando por una constante $e^{-\frac{j\pi n}{4M}}$ según (103):

$$p_m[n] = 2 \cdot \text{Re}\left\{z_m[n] \cdot e^{-\frac{j\pi(2m+1)n}{4M}}\right\} \quad (103)$$

Donde $z_m[n]$ es la salida del reordenamiento de la trama de datos; $p_m[n]$ es la salida del módulo DCT; y $\text{Re}\{\}$ es el operador parte real.

La arquitectura de la DCT del receptor (véase la Figura 52) es idéntica a la del transmisor, ya que las transformadas directa e inversa tienen la misma definición en la DCT-IV. Sin embargo, nótese que la señal de entrada de la DCT del receptor es $p'_m[n]$ y la de salida $X'_m[k]$, en lugar de $X_m[k]$ y $p_m[n]$ respectivamente.

5.2.1. Efecto de la precisión finita

A continuación, se presenta el estudio del efecto de la representación en precisión finita (cuantificación) realizado en la DCT. Para la realización de este diseño, al igual que sucedía en la arquitectura de la DTT, se ha buscado el empleo de un único multiplicador por multiplicación, sin necesidad de extensiones de palabra. Como se va a emplear la tarjeta ML605 de Xilinx para la implementación del diseño y ésta posee una FPGA Virtex6 xc6vlx240t, las celdas multiplicadoras DSP48E1 [109] proporcionados por la arquitectura de esta FPGA disponen de una entrada de 18 bits y otra de 25 bits. Por esto, se ha decidido emplear una cuantificación de 18 bits para los datos a transmitir, variando el número de bits

de la parte fraccionaria dependiendo de los valores máximos de cada zona. La cuantificación de 25 bits se deja para los coeficientes. La decisión de emplear la cuantificación de 18 bits para los datos es debida a que, en el core de la FFT, éste es el máximo ancho de palabra para el empleo de un solo multiplicador por operación. En la Figura 53 se puede observar los puntos de cuantificación utilizados y su valor, así como las distintas fases descritas anteriormente.

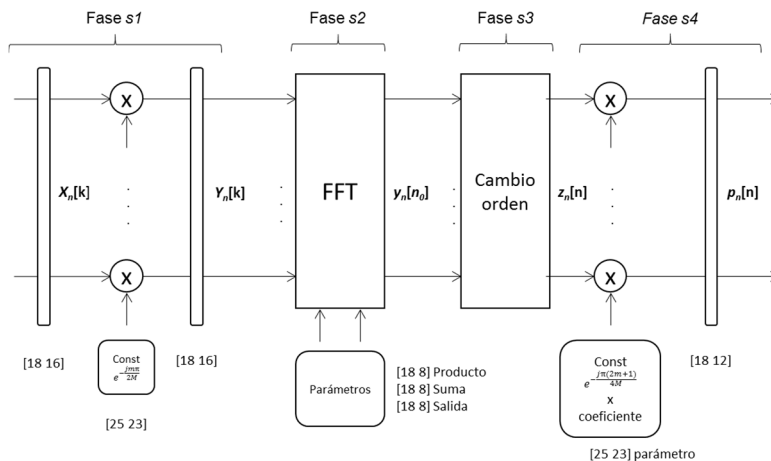


Figura 53. Diagrama de bloques con cuantificación empleada en el módulo DCT.

Con el objetivo de analizar la influencia de la cuantificación, además de la cuantificación de 18 bits, se ha probado una cuantificación menor y otra superior para comprobar la idoneidad de la propuesta de 18 bits, obteniendo los resultados mostrados en la Tabla 21. El parámetro *error absoluto* indica la máxima diferencia obtenida entre la señal en coma fija y en doble precisión en coma flotante, mientras que el *error relativo* divide el *error absoluto* entre el mayor valor de la salida en coma flotante. Para la realización de esta comprobación se han transmitidos 100 tramas de datos con valores aleatorios entre [-1,1].

Tabla 21. Error obtenido en el módulo DCT4e para cada caso de cuantificación.

	[16 10]	[18 12]	[20 14]
error absoluto	0.2364	0.0692	0.0150
error relativo	1.264	0.3372	0.0791

Se observa que a medida que se aumenta el número de bits de la cuantificación se reduce el error. En el caso de la cuantificación [18 12] se obtienen unos valores adecuados para realizar la DCT-IV. En la Figura 54 se puede observar la diferencia entre el módulo DCT4e con y sin efecto de la precisión finita. Para ello, se ha transmitido una trama de datos con un valor constante de 0.5. La media del error introducido por la cuantificación [18 12] es del 0.64% respecto al máximo valor de salida en coma flotante.

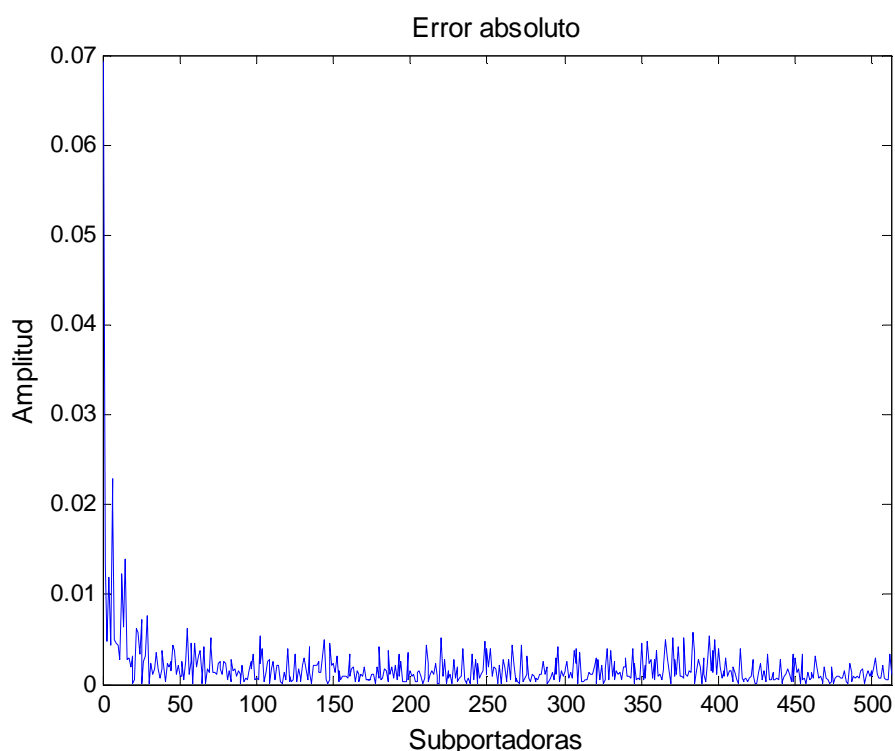


Figura 54. Error absoluto de la salida de DCT4e de sistema con coma fija [18 12] y con coma flotante para una entrada $X[k]$ constante de 0.5.

5.2.2. Consumo de recursos

Como se ha explicado anteriormente, el módulo DCT está formado por 4 fases (véase la Figura 53). Al igual que ocurría en el módulo Θ , se va a emplear un ratio de paralelismo en el bloque R_p^{DCT} . Con este ratio se consigue una reducción del número de recursos a emplear, indicando el número de subcanales que se entregarán en paralelo, tanto a la FFT, como a las multiplicaciones de la DCT. Modificando el valor de R_p^{DCT} , que puede variar desde 1 hasta M , se puede modificar este ratio de paralelismo, aunque también afecta al número de recursos que se emplea y al tiempo de procesamiento final.

A continuación se van a desarrollar los recursos empleados en cada fase de la DCT. La primera y cuarta fase consisten en una multiplicación por una constante compleja. En la segunda fase se realiza una FFT, para lo cual se propone el uso de un core IP generado por Spiral [111]. Esto es debido a que el core proporcionado por Spiral se ajusta mejor a los requisitos de un modelo semi-paralelo que se va a emplear en esta arquitectura. Teniendo en cuenta estos comentarios, se puede establecer la siguiente carga computacional para el algoritmo:

Fase s1 y s4, multiplicación por una constante compleja:

$$C_m^{s1/s4} = 4 \cdot R_p^{DCT} \text{ multiplicadores por cada fase} \quad (104)$$

$$C_s^{s1/s4} = 2 \cdot R_p^{DCT} \text{ sumadores por cada fase} \quad (105)$$

Fase s2, realización de un FFT de M elementos:

$$C_m^{s2} = C_m^{FFT} \quad \text{multiplicadores} \quad (106)$$

$$C_s^{s2} = C_s^{FFT} \quad \text{sumadores} \quad (107)$$

La fase s3 no necesita ninguna operación matemática, al tratarse de un reordenamiento de los datos de la trama de datos. Sin embargo, si que empleará memorias BRAM para realizar este reordenamiento.

El número total de multiplicadores C_m^{DCT} a emplear por la DCT se obtiene de la suma de las distintas fases que la componen, según (108):

$$C_m^{DCT} = C_m^{s1} + C_m^{s2} + C_m^{s4} = 4 \cdot R_p^{DCT} + C_m^{FFT} + 4 \cdot R_p^{DCT} = C_m^{FFT} + 8 \cdot R_p^{DCT} \quad (108)$$

5.2.3. Cálculo de las restricciones de la arquitectura

Se va a realizar un estudio conjunto de la latencia de la DCT y del consumo de multiplicadores que puede emplear la FFT, teniendo en cuenta la frecuencia de reloj del periférico f_{clk} , el número de multiplicadores C_m^{DCT} usados, el ratio de frecuencia entre la frecuencia del periférico y la frecuencia del conversor $R_f=f_{clk}/f_s$, y el ratio de paralelismo R_p . Con este estudio se pretende obtener las restricciones de la arquitectura que se imponen al emplear la FPGA Virtex6 xc6vlx240t.

Para realizar el cálculo de la latencia y el consumo de multiplicadores se van a plantear dos restricciones: la primera teniendo en cuenta el número máximo de multiplicadores disponibles; y en la segunda el tiempo de procesamiento disponible para realizar las operaciones; teniendo como parámetros tiempo de procesamiento máximo M_c^{max} , el número C_m^{FFT} de multiplicadores usado por la FFT y el tiempo de procesamiento M_c^{FFT} necesario.

La primera restricción se obtiene de la suma del número de multiplicadores C_m^{DCT} (108) siendo este valor dependiente del ratio R_p^{DCT} . Esta suma esta restringida por el número máximo de multiplicadores disponibles C_m^{FPGA} . Para la segunda ecuación se ha calculado el tiempo de procesamiento M_c^{DCT} disponible, a partir del que se emplea en cada fase, empleando para ello, la limitación del tiempo de procesamiento M_c^{max} límite.

$$C_m^{DCT} = C_m^{FFT} + 8 \cdot R_p^{DCT} \leq C_m^{FPGA} \quad (109)$$

$$M_c^{DCT} = M_c^{s1} + M_c^{s2} + M_c^{s3} + M_c^{s4} \leq M_c^{max} \quad (110)$$

Despejando en cada ecuación, de $M_c^{s1}=M_c^{s4}=4$ (ciclos de reloj empleados en la multiplicación compleja), y de $M_c^{s3}=2$ (ciclos para leer una memoria BRAM), se puede saber el número máximo de multiplicadores C_m^{s2} y tiempo de procesamiento M_c^{s2} que se pueden utilizar para la FFT en función del valor de R_p^{DCT} .

$$C_m^{FFT} \leq C_m^{FPGA} - 8 \cdot R_p^{DCT} \quad (111)$$

$$M_c^{FFT} = M_c^{s2} \leq M_c^{max} - 4 \cdot 2 + 2 \quad (112)$$

Por otro lado, se recuerda que la latencia no tiene impacto en el diseño, pero sí se emplea para tener una mejor definición de la arquitectura. Dicho esto, la latencia L^{DCT} de la DCT queda según (113):

$$L^{DCT} = L^{S^1} + L^{S^2} + L^{S^3} + L^{S^4} \quad (113)$$

Donde $L^{S^1}=L^{S^4}=8$ son las latencias típicas de la multiplicación compleja; L^{S^2} es la latencia de la FFT; y $L^{S^3}=M$ es la latencia del cambio de orden.

5.3. Matrices **I** – **J**

Este módulo realiza la multiplicación de la señal de entrada $p_m[n]$ con dos matrices diagonales **I** y **J**. Estas dos matrices **I** y **J** se combinan de manera que generan S señales de salida $q_s[n]$ en el transmisor. Nótese que hay M señales de entrada $p_m[n]$ en el módulo transmisor, mientras que se generan $S = 2 \cdot M$ salidas $q_s[n]$, según (114) y (115):

$$q_m[n] = \overbrace{p_m[n] - p_{M-m-1}[n]}^{Matriz(I-J)} \quad (114)$$

$$q_{m+M}[n] = \overbrace{-p_m[n] - p_{M-m-1}[n]}^{Matriz(-I-J)} \quad \text{donde } m = 0, \dots, M-1. \quad (115)$$

En el caso del receptor, como se ve en la Figura 52, la señal de entrada del módulo es $q'_s[n]$, mientras que $p'_m[n]$ es la señal de salida, que se combina con las matrices **I** y **J** según (116):

$$p'_m[n] = \overbrace{q'_s[n] + q'_{\frac{s}{2}-s-1}[n]}^{Matriz(I+J)} + \overbrace{q'_{\frac{s}{2}+s}[n] - q'_{s-s-1}[n]}^{Matriz(I-J)} \quad (116)$$

Donde $s = 0, \dots, S-1$.

Las matrices empleadas utilizan la señal de entrada (matriz **I**) o la señal de entrada en orden inverso (matriz **J**). Con esto, la combinación de ambas matrices es la suma o resta de la entrada consigo misma, por lo que no se han de emplear multiplicaciones para realizar estas operaciones.

En este caso, no se producen errores por cuantificación, ya que las sumas y restas que se realizan por las combinaciones de las matrices no producen acarreo. Por lo tanto, no es necesario modificar la cuantificación asignada anteriormente [18 12]. También se ha observado que no se produce desbordamiento, ya que las multiplicaciones de la DCT-IV así lo favorecen.

Para esta fase, no habrá multiplicaciones, ya que sólo se realiza una suma por cada elemento, por lo que el número de sumadores C_s^{matriz} depende del ratio de paralelismo R_p de la matriz. El número de sumadores queda según (117).

$$C_s^{matriz} = 2 \cdot R_p \quad (117)$$

La latencia L^{Matriz} que presenta en la Matriz **I** y **J** se corresponde con (118):

$$L^{Matriz} = 4 + M \quad (118)$$

Que se corresponde con 2 ciclos del acceso a memoria, más otro ciclo por las operaciones de suma y un último ciclo de reloj correspondiente al retardo de $M=512$ ciclos de reloj que se implementa en $q_{m+M}[n]$. Además hay que considerar una latencia extra de M ciclos producida por el empleo de memorias ping-pong para producir los datos en el orden correcto.

5.4. Banco de filtros polifásicos

El banco de filtros polifásicos consiste en un banco $G_s(z)$ de $S=2 \cdot M$ filtros FIR (*Finite Impulse Response*) de orden siete. Sólo el primer y tercer coeficiente, $c_{s,0}$ y $c_{s,2}$, no son nulos, por lo que el banco de filtros queda definido según (119):

$$G_s(z) = c_{s,0} + c_{s,2} \cdot z^{-2} \quad (119)$$

Esto reduce significativamente la carga computacional. Tanto el transmisor como el receptor incluyen el mismo banco de filtros polifásicos. En la Figura 55 se muestran los coeficientes proporcionados para el filtro diseñado.

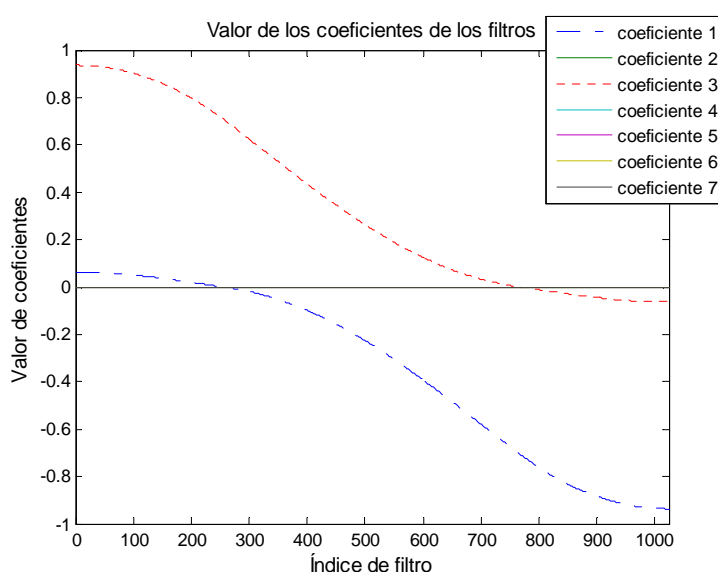


Figura 55. Coeficientes empleados en el banco de filtros FIR polifásicos.

Como se observa, los coeficientes de los filtros no superan la unidad en ningún caso. Este dato es importante en términos de efecto de la precisión finita, puesto que se ha de determinar el número de bits que se destinan a la parte entera y a la parte fraccionaria.

Para la realización del filtro se ha buscado un compromiso entre utilización de recursos y número de ciclos de reloj necesarios para operar. Por ello, se optó por realizar bloques de filtros que trabajan en paralelo. Con esto se consigue disminuir el número de ciclos para realizar todos los filtros, pero se utiliza un número reducido de recursos al realizar cada bloque M/R_p^F filtros, siendo R_p^F el ratio de paralelismo en los filtros.

5.4.1. Efecto de la precisión finita

La representación en coma fija utilizada en los coeficientes que determinan el filtro es de 25 bits con 23 bits de parte fraccionaria (véase el apartado 5.2.1), mientras que la cuantificación de los datos proporcionados desde las matrices **I-J** es de 18 bits con 12 bits de parte fraccionaria. En la Figura 56 se pueden observar las distintas cuantificaciones utilizadas en el sistema a partir de la Matriz **I** y **J**.

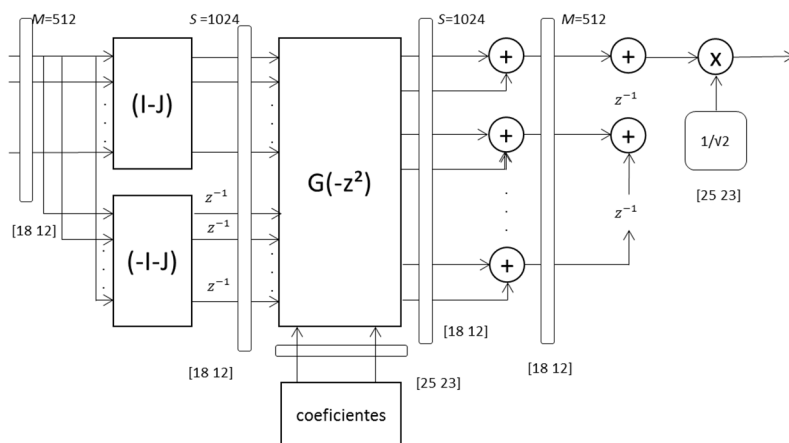


Figura 56. Diagrama de bloques de Matriz y banco de filtros indicando zonas de cuantificación.

Por tanto, una vez definida la precisión finita en los coeficientes del filtro, se realiza el filtrado de las señales proporcionadas por las matrices **I-J**. En la Figura 57 se muestra la comparativa entre la salida del filtro en coma fija y la salida del filtro en coma flotante, mostrando la diferencia de ambas. Para esta comparativa se ha empleado una sola trama de datos con un valor constante de 0,5.

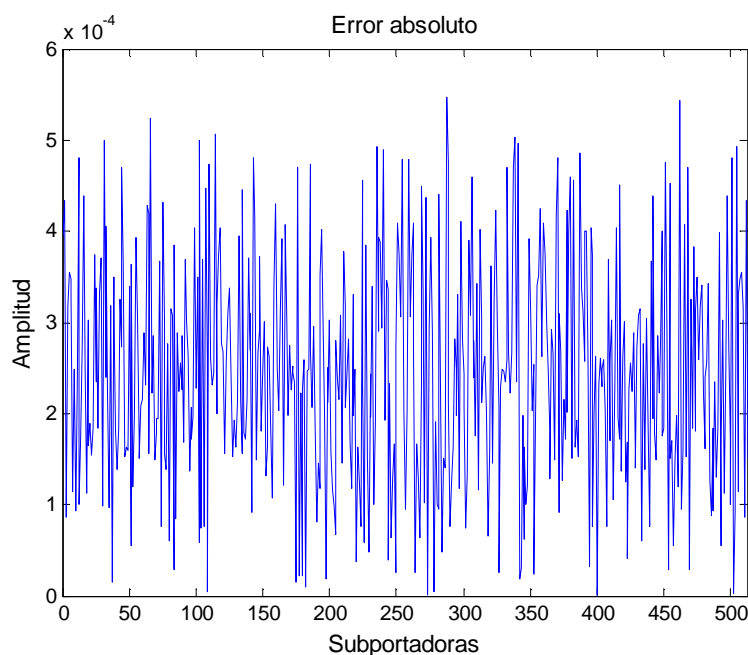


Figura 57. Error absoluto a la salida del banco de filtros de sistema con coma fija [18 12] y con coma flotante para una entrada $x[k]$ constante de 0,5.

5.4.2. Consumo de recursos

Por cada bloque del filtro se utilizan dos multiplicadores para la realización de la etapa de filtrado, ya que, como se ha descrito antes, el filtro posee tres coeficientes, aunque el segundo es nulo. Así, en función del parámetro R_p^F (ratio de paralelismo en el banco de filtros), el número de multiplicadores C_m^F utilizados será (120):

$$C_m^F = M_{coef}^F \cdot R_p^F + C_m^{CTE} \quad (120)$$

Donde C_m^{CTE} se corresponde con la multiplicación final por la constante $1/\sqrt{2}$; y M_{coef}^F es el número de coeficientes no nulos.

5.4.3. Latencia

Se recuerda que la latencia no tiene impacto en el diseño, solamente se emplea para tener una mejor definición de la arquitectura. Para la realización de la etapa de filtrado, se va a emplear una celda básica de multiplicación más acumulador. Esta celda, llamada DSP48E1 [109], necesita $L^{DSP}=3$ ciclos de reloj para realizar las operaciones. Por tanto, para la obtención del primer dato de la etapa de filtrado y suma, se necesitan L^F ciclos de reloj (121):

$$L^F = 2 \cdot L^{DSP} + L^{suma} + L^{CTE} \quad (121)$$

Siendo $L^{suma}=3$ el número de ciclos necesario para la suma de los pares de señales del filtro $t_m[k]$ y $t_{m+M}[k]$; y $L^{CTE}=2$ el número de ciclos de reloj necesario para realizar la multiplicación por la contante $1/\sqrt{2}$.

5.5. Arquitectura del sistema

Para la implementación de la arquitectura de FBMC se van a emplear los mismos parámetros que los utilizados en la DTT. Estos parámetros son la tasa de transmisión $T_{Tx}=62.5\text{Mps}$, el número de subcanales de entrada $M=512$ y el ratio de paralelismo $R_p=2$.

Además, como la implementación se va a realizar en la misma FPGA Virtex 6 xc6vlx240t [110], las restricciones a la arquitectura van a ser las mismas. Estas restricciones son el número máximo de multiplicadores disponibles $C_m^{FPGA}=768$ y la frecuencia de muestreo del conversor $f_s=62.5\text{MHz}$. Con estas restricciones se ha decidido fijar la frecuencia de la técnica de acceso a $f_{per}=62.5\text{MHz}$, lo que fija el ratio de frecuencia en $R_f = \frac{f_s}{f_{per}} = 1$. Con este ratio se obtiene el tiempo de procesamiento máximo $M_c^{max}=512$ ciclos de reloj.

5.5.1. Obtención de parámetros de la FFT y de la DCT

Como se va a emplear el algoritmo 1 de la DCT desarrollada para la Transformada Trigonométrica Discreta, los parámetros característicos de la FFT van a ser los mismos. Para generar la FFT se ha empleado un core que proporciona Spiral [111], el cual permite

generar distintas arquitecturas con distintos ratios de paralelismo R_p . Como se ha realizado anteriormente, se ha elegido la FFT con un ratio de paralelismo $R_p=2$, de un tamaño de $M=512$ puntos cuyos parámetros más característicos son (122) (123):

$$C_m^{FFT} = 32 \text{ multiplicadores} \quad (122)$$

$$M_c^{FFT} = 256 \text{ ciclos, } L^{FFT} = 728 \text{ ciclos} \quad (123)$$

Con estos nuevos datos y trabajando sobre las restricciones calculadas sobre la arquitectura de la FFT en (111) y (112), se obtiene (124) (125):

$$C_m^{FFT} = 32 \leq C_m^{FPGA} - 8 \cdot R_p^{DCT} = 768 - 16 = 752 \quad (124)$$

$$M_c^{FFT} = M_c^{S2} = 256 \leq R_f \cdot M - 4 \cdot 2 + 2 = 502 \quad (125)$$

Como se observa, con la FFT con $R_p^{DCT}=2$, se cumplen las restricciones especificadas, por lo que ésta será la FFT implementada en el diseño.

Se puede establecer, a su vez, los valores de número de multiplicadores y el tiempo de procesamiento empleado en la DCT. Empleando los valores obtenidos en la FFT se puede calcular el número de multiplicadores C_m^{DCT} empleados en la DCT (109) y también, calcular el valor de la latencia L^{DCT} de la DCT (113), obteniendo los resultados mostrados para C_m^{DCT} (126) y para la latencia L^{DCT} (127):

$$C_m^{DCT} = 4 \cdot R_p^{FFT} + C_m^{FFT} + 4 \cdot R_p^{FFT} = 8 + 32 + 8 = 48 \text{ multiplicadores} \quad (126)$$

$$L^{DCT} = L^{S1} + L^{S2} + L^{S3} + L^{S4} = 8 + 728 + 512 + 8 = 1256 \text{ ciclos} \quad (127)$$

Conociendo que $L^{S1}=L^{S4}=8$ (latencia de la multiplicación compleja); y que $L^{S3}=512$ (periodo completo por memoria ping-pong).

5.5.2. Obtención parámetros del Banco de filtros y el módulo Matriz

Una vez se han fijado los parámetros para la arquitectura del FBMC se procede a calcular los multiplicadores empleados y latencia en los bloques de las matrices **I** y **J** y el banco de filtros.

Con los valores fijados en este apartado, se puede resolver la latencia introducida por el módulo Matriz calculada anteriormente en (118), obteniendo la latencia L^{Matriz} (128):

$$L^{Matriz} = 4 + M = 4 + 512 \text{ ciclos} \quad (128)$$

También, a su vez se puede resolver el número de multiplicadores C_m^F empleados por el banco de filtros, calculado en (120) y la latencia L^F que introduce, calculada en (121), obteniendo (129) y (130):

$$C_m^F = M_{coef}^F \cdot R_p^F + C_m^{CTE} = 2 \cdot 2 + 1 = 5 \text{ multiplicadores} \quad (129)$$

$$L^F = 2 \cdot L^{DSP} + L^{suma} + L^{CTE} = 6 + 3 + 2 = 11 \text{ ciclos} \quad (130)$$

Eso supone que se utilizarán $C_m^F=5$ multiplicadores y se tendrá una latencia en procesar datos de entrada $L^F=11$ ciclos de reloj.

5.5.3. Arquitectura del transmisor

5.5.3.1. Bloque DCT y bloque Theta

A continuación se va a mostrar la arquitectura realizada en cada uno de los bloques implementados para el desarrollo de la propuesta FBMC, en este caso la DCT y el módulo Theta. Para la arquitectura, se ha realizado dos modificaciones respecto a lo mostrado en el apartado 5.2, ya que se ha buscado reducir lo máximo posible tanto el consumo de recursos como el número de ciclos del tiempo de procesamiento. Para ello, se ha decidido que el bloque multiplicador Theta y la primera fase de la DCT, se realicen conjuntamente, ya que de este modo se puede ahorrar los multiplicadores empleados en el módulo de multiplicación Theta (97). A su vez, se ha modificado el orden de la fase s3 y s4 de la DCT para de esta manera, poder unir las memorias empleadas en la fase s3, con las memorias empleadas para realizar el bloque de Matrices **I** y **J**, ya que, se ahorra un número significativo de ciclos de reloj y recursos. Con estos cambios, las ecuaciones (96) y (100) pasan a formar la siguiente expresión (131):

$$Y_m[k] = \theta_m \cdot V_m[k] \cdot e^{-\frac{jm\pi}{2M}} \quad (131)$$

Con todos estos cambios se procede a mostrar la arquitectura realizada. En la Figura 58 se muestra el módulo DCT implementado.

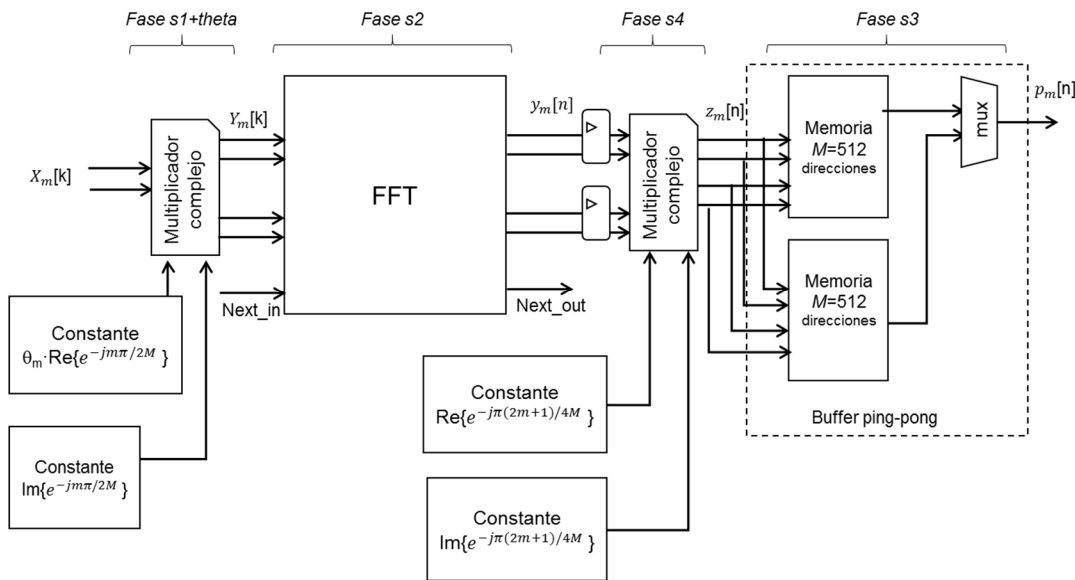


Figura 58. Diagrama de bloques de la arquitectura del módulo DCT.

Dado que el ratio de paralelismo se ha fijado en $R_p^{DCT}=2$, $C_m^{s1}=8$ multiplicadores son necesarios en la fase s1 y, la latencia es de $L^{s1}=8$ ciclos de reloj. Después, la segunda fase s2 es una FFT que requiere no sólo multiplicadores y sumadores, sino también memoria donde almacenar los valores intermedios (101). Este módulo es parametrizado con un

tamaño de $M=512$ puntos y una representación en coma fija de 18 bits, valor fijado por el ancho de palabra que proporciona una de las entradas de los multiplicadores de la FPGA. La FFT presenta una arquitectura radix-2 semi-paralela, con $\log_2 M$ bloques en cascada. De nuevo, la selección del ratio de paralelismo es $R_p^{DCT}=2$, que implica que sólo dos muestras de entrada $Y_m[k]$ son procesadas cada ciclo de reloj. Una FFT se completa cada $M_c^{FFT}=256$ ciclos de reloj, con una latencia $L^{FFT}=728$ ciclos.

La tercera fase s3 de la DCT es un reordenamiento de la trama de datos (102), donde todas las muestras de la trama $y_m[n]$ deben estar disponible al mismo tiempo. Por lo tanto, los datos deben ser almacenados en memoria, para que sean leídos en el orden correcto. Para evitar pérdida de datos, se han empleado dos memorias en modo ping-pong, por lo que en una se puede acceder para organizar los datos en el orden correcto, mientras que los nuevos datos se escriben en la otra. Esta fase implica una latencia de $L^{s3}=512$ ciclos. Por último, la cuarta fase s4 del módulo DCT es similar a la primera fase s1 (103), donde $C_m^{s4}=8$ multiplicadores son usados y la latencia es de $L^{s4}=8$ ciclos de reloj.

Como se puede observar, las cuatro fases que forman el módulo DCT están segmentadas. Como los datapaths empleados tienen un throughput de $T_{ch}=1$ dato por ciclo de reloj, y como el ratio de paralelismo es $R_p^{DCT}=2$, el throughput global de la DCT es $T_c^{DCT}=T_{ch} \cdot R_p=2$. Con todo esto, todas las muestras de entrada $V_m[k]$ son procesadas cada $M_c=256$ ciclos. Como un nuevo conjunto de muestras de entrada $V_m[k]$ está disponible cada $M_c^{max}=512$ ciclos, la arquitectura propuesta cumple esta limitación.

Por lo tanto, el número de multiplicaciones C_m^{DCT} empleadas para la realización de la arquitectura de la DCT es (132):

$$C_m^{DCT} = C_m^{s1} + C_m^{FFT} + C_m^{s4} = 8 + 32 + 8 = 48 \text{ multiplicadores} \quad (132)$$

Donde $C_m^{s1}=8$ es el número de multiplicadores en la fase s1, que se realiza junto a la multiplicación por theta (Θ); $C_m^{FFT}=32$ es el número de multiplicadores empleados en la FFT; y $C_m^{s4}=8$ es el número de multiplicadores empleados en la fase s4.

La latencia L^{DCT} de la arquitectura propuesta para la DCT y el módulo Theta es obtenida en (133):

$$L^{DCT} = L^{s1} + L^{s2} + L^{s3} + L^{s4} = 8 + 728 + 512 + 8 = 1256 \text{ ciclos} \quad (133)$$

Donde $L^{s1}=8$ es la latencia de la fase s1 junto con el módulo Theta (131); $L^{s2}=728$ es la latencia de la FFT (101); $L^{s3}=512$ es la latencia de la fase s3 (102); y $L^{s4}=8$ es la latencia de la fase s4 (103).

Una vez se ha mostrado la arquitectura realizada, se procede a mostrar un cronograma del funcionamiento de la DCT y el módulo Theta, que se puede observar en la Figura 59. Primero, la señal de entrada $V_m[k]$ entra sincronizada con la señal CE_M (muestra de reloj 1). Con esta señal se comienza a generar las direcciones y se activa el enable de la memoria que almacena los coeficientes $\Theta_m \cdot e^{-\frac{j m \pi}{2M}}$ para la primera multiplicación (131) (muestra de reloj 3). Los coeficientes y la señal de entrada entran en el multiplicador para generar $Y_m[k]$

cuatro ciclos de reloj después (muestra de reloj 7). La señal generada se sincroniza con la señal *next_in*, para indicarle a la FFT cuando comienza a recibir los datos (muestra de reloj 9).

Una vez ha sido realizada la FFT, se activa la señal *next_out* y se comienzan a proporcionar la señal $y_m[n]$ (muestra de reloj 13). A su vez, se comienza a entregar los coeficientes $e^{-\frac{j\pi(2m+1)}{4M}}$ para realizar la segunda multiplicación (103) (muestra de reloj 15). La señal que se obtiene es $z_m[n]$, que a su vez está sincronizada con las direcciones de la puerta A de las memorias ping-pong, para realizar la fase de reordenación de la trama de datos (102) (muestra de reloj 20). Una vez se ha realizado el reordenamiento, se obtiene la señal $p_m[n]$ (muestra de reloj 22).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
CEM																									
Yn[k]	x	Dn0	Dn1	Dn2	Dn3	Dn4	Dn5	Dn6	Dn7	Dn8															
en																									
Addr1	x	0	1	2	3	4	5	6	7	8	9														
coefDenes	x	Coef0	Coef1	Coef2	Coef3	Coef4	Coef5	Coef6	Coef7	Coef8															
Mult	x	x	Coef0	Coef1	Coef2	Coef3	Coef4	Coef5	Coef6	Coef7															
Yn[k+1]	x	x	din0	din1	din2	din3	din4	din5	din6	din7															
Yn[k]	x	x	x	x	x	x	mult0	mult1	mult2	mult3															
Yn[k+1]	x	x	x	x	x	x	mult0	mult1	mult2																
Yn[k+2]	x	x	x	x	x	x	x	mult0	mult1	mult2															
next_in																									
next_out																									
Yn[0]	x	x	x	x	x	x	x	x	x	x															
Yn[n+1]	x	x	x	x	x	x	x	x	x	x															
Yn[n+2]	x	x	x	x	x	x	x	x	x	x															
coefDenes	x	x	x	x	x	x	x	x	x	x															
mult2	x	x	x	x	x	x	x	x	x	x															
zn[n]	x	x	x	x	x	x	x	x	x	x															
zn[n+1]	x	x	x	x	x	x	x	x	x	x															
addrA	x	x	x	x	x	x	x	x	x	x															
addrB	x	x	x	x	x	x	x	x	x	x															
pn[0]	x	x	x	x	x	x	x	x	x	x															
pn[n+1]	x	x	x	x	x	x	x	x	x	x															
fn[0]																									
fn[1]																									
fn[2]																									
fn[3]																									
fn[4]																									
fn[5]																									
fn[6]																									
fn[7]																									
fn[8]																									
fn[9]																									
fn[10]																									
fn[11]																									
fn[12]																									
fn[13]																									
fn[14]																									
fn[15]																									
fn[16]																									
fn[17]																									
fn[18]																									
fn[19]																									
fn[20]																									
fn[21]																									
fn[22]																									
fn[23]																									

Figura 59. Cronograma de la DCT implementada.

5.5.3.2. Módulo Matriz

En este apartado se va a mostrar la arquitectura realizada para las matrices **I** y **J**. Como se ha mencionado anteriormente, las memorias empleadas en la última fase de la DCT y las memorias empleadas en el módulo Matriz se han unido para reducir significativamente el número de ciclos de latencia y consumo de recursos. En la Figura 60 se muestra el diagrama de bloques de la arquitectura realizada, sin tener en cuenta esta fusión de memorias, para las matrices **I** y **J**.

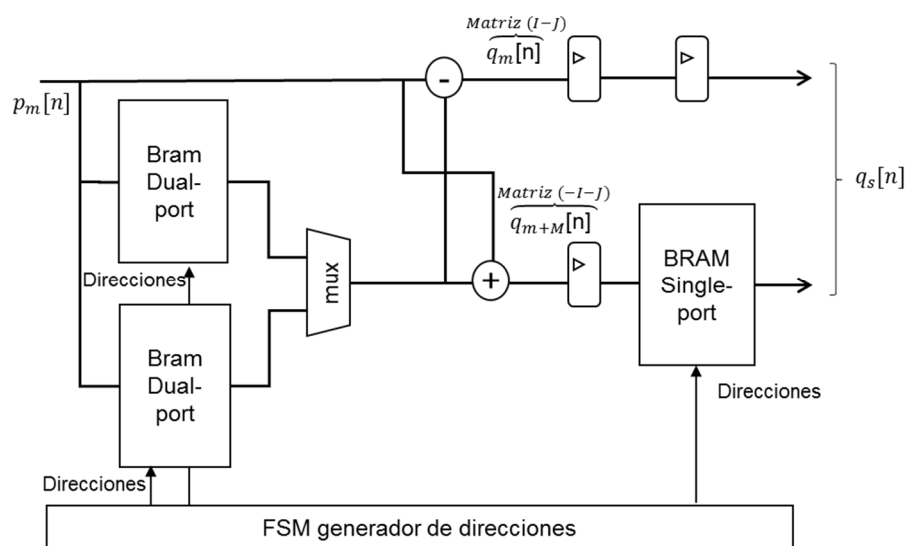


Figura 60. Diagrama de bloques de la arquitectura realizada para el módulo de matrices **I** y **J**.

Como se ha mencionado, la Matriz **I** es la matriz diagonal identidad, con lo que la entrada $p_m[n]$ va directamente a la salida $q_s[n]$; por otro lado, la Matriz **J** es la matriz diagonal invertida identidad, que realiza una inversión en el orden de la señal $p_m[n]$ en las salidas $q_s[n]$.

Al igual que sucedía en la tercera fase s_3 de la DCT, es necesario tener dos memorias ping-pong para invertir la trama de datos. Después, las señales de salida de ambas matrices **I** y **J** son sumadas o restadas para obtener la combinación aritmética para $q_s[n]$. Como se va a realizar una arquitectura semi-paralela, con un ratio de paralelismo $R_p=2$, ambas matrices se pueden reducir en la arquitectura como dos sumadores. Las correspondientes señales $p_m[n]$ se proporcionan en los sumadores a través del correspondiente direccionamiento de las memorias ping-pong, de acuerdo con la máquina de estados finitos (FSM).

Existe un retardo de $M=512$ ciclos para cada salida $q_s[n]$ que proviene de la combinación de **(-I-J)**, como se puede observar en la Figura 60. Este retardo de $M=512$ ciclos está implementado con una memoria BRAM. En la Figura 51 este retardo está representado solamente con un bloque z^{-1} en la salida de la Matriz **(I-J)**, ya que este diagrama representaba un enfoque totalmente paralelo. Sin embargo, la propuesta que se describe aquí es semi-paralela, con un ratio de paralelismo $R_p=2$, con lo que el nuevo retardo pasa a ser de $M=512$ ciclos.

Es importante remarcar que el módulo matriz presenta una arquitectura similar en el receptor, lo que implica los mismos recursos y latencias y solamente se diferencia en las operaciones aritméticas, de acuerdo con (116). La propuesta tiene una latencia de $L^{Matriz}=4$ ciclos de reloj y no requiere del empleo de multiplicadores. Ambos sumadores han sido calculados para no tener desbordamiento, y así evitar errores adicionales.

5.5.3.3. Banco de filtros

Por último se muestra la arquitectura realizada en el banco de filtros polifásicos. Este módulo consiste en $S=2 \cdot M=1024$ filtros FIR, de tres coeficientes cada uno, como se muestra en (119). En la Figura 61 se muestra la arquitectura propuesta para este módulo, donde dos filtros trabajan en paralelo para las señales $q_m[n]$ y $q_{m+M}[n]$, debido al ratio de paralelismo $R_p^F=2$. El proceso de filtrado puede ser reducido a dos estados M_{coef}^F (dos coeficientes no nulos), implementando dos memorias para almacenar las últimas tres muestras de la entrada $q_m[n]$ y $q_{m+M}[n]$, mientras que los coeficientes para los S filtros son almacenados en $M_{coef}^F \cdot R_p^F=4$ memorias ROM (*Read-Only Memory*) [115].

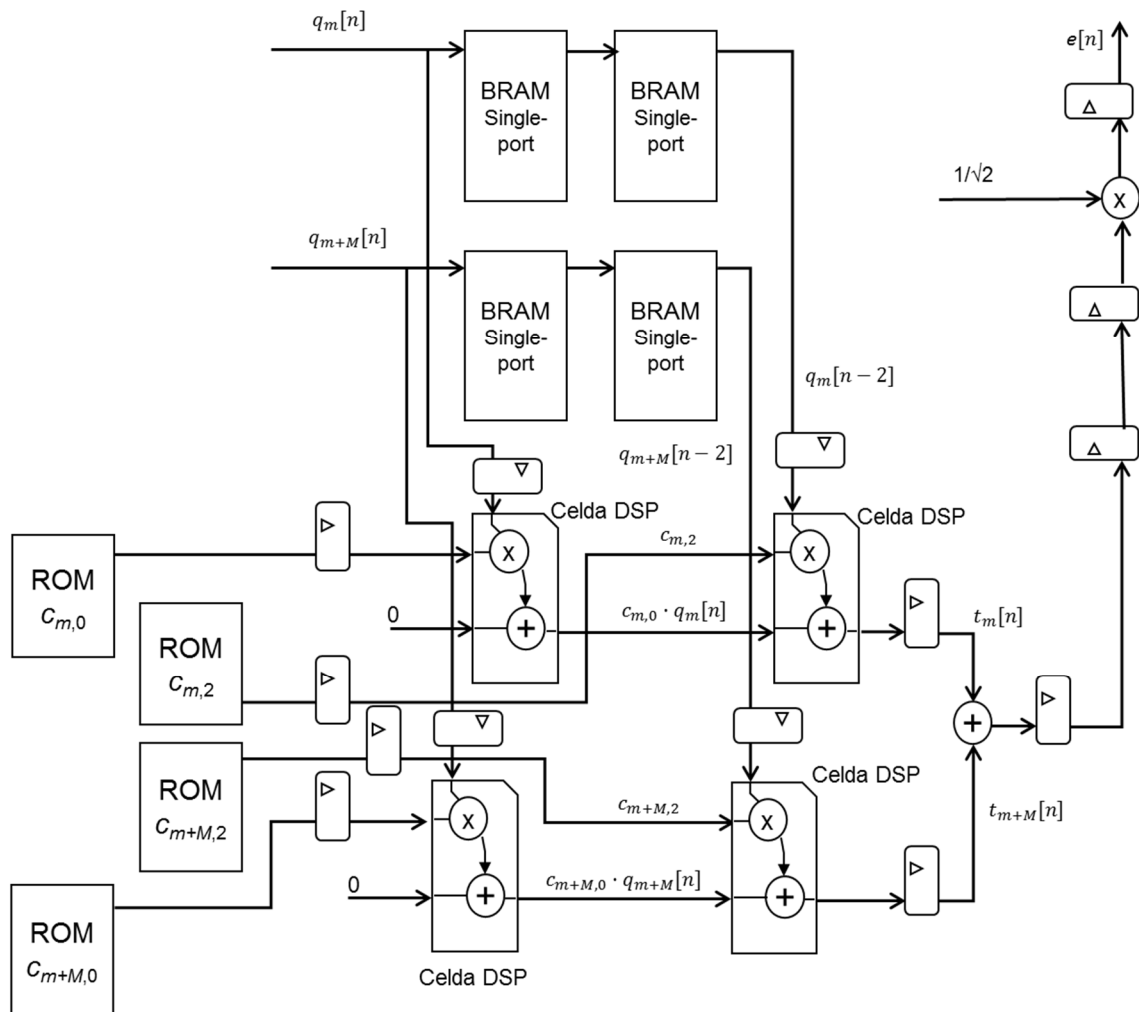


Figura 61. Diagrama de bloques del banco de filtros implementado.

En la arquitectura del filtro, dos celdas multiplicadoras-acumuladores (DSP48E1) son empleadas para las correspondientes multiplicaciones y acumulaciones de las señales $q_s[n]$ y $q_s[n-2]$ con los coeficientes $c_{s,0}$ y $c_{s,2}$, respectivamente. Después de los filtros polifásicos, la señales $q_m[n]$ y $q_{m+M}[n]$ son sumadas, y el resultado es multiplicado por una constante $1/\sqrt{2}$, para obtener la señal $e[n]$ que será transmitida.

Al igual que en el bloque DCT, el banco de filtros ha sido implementado en modo semiparalelo, con un ratio de paralelismo de $R_p^F=2$, con lo que según (120) el número de multiplicadores es $C_m^F=5$, y según (121), la latencia es $L^F=11$ ciclos de reloj, mientras el throughput es de $T_c^F=1$ ciclo de reloj.

Una vez se ha mostrado la arquitectura realizada tanto para el módulo Matriz como para el banco de filtros, se procede a mostrar un cronograma del funcionamiento de ambos, que se refleja en la Figura 62. Como se puede observar, primero, a partir de la señal CE_M se comienzan a generar las direcciones de memoria para realizar tanto la Matriz **I**, como la Matriz **J** (muestra de reloj 3). Para ello, se genera las direcciones en orden directo e inverso (muestra de reloj 2). Con esto se obtiene el dato **I** y el dato **J**, para un ciclo de reloj después obtener las operaciones de **I-J** y **-I-J** (muestra de reloj 4). La señal de **-I-J**, se retarda hasta el siguiente grupo de señales, pasando a llamarse $q_{m+M}[n]$, mientras que la otra operación se sitúa en $q_m[n]$ (muestra de reloj 5). Estas señales se retardan hasta dos grupos de señales más tarde para la realización del filtro (muestra de reloj 6-8), y se realizan las operaciones intermedias de multiplicación y acumulación (muestra de reloj 10). A la salida de los filtros se obtiene $t_m[n]$ y $t_{m+M}[n]$ (muestra de reloj 11), las cuales se suman en pares, obteniendo la señal $s_m[n]$ (muestra de reloj 13). Esta señal se multiplica por la constante $1/\sqrt{2}$ para obtener la señal $e[n]$ (muestra de reloj 15).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CEM																	
en																	
addr	x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
addr2	x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dato1	x	x	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14
addrN	x	511	510	509	508	507	506	505	504	503	502	501	500	499	498	497	496
Dato1	x	D511	D510	D509	D508	D507	D506	D505	D504	D503	D502	D501	D500	D499	D498	D497	D496
dato1-1	x	x	x	x	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
dato1-1	x	x	x	x	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
qn[n]	x	x	x	x	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
qn+M[n]	x	x	x	x	R0 ant	R1 ant	R2 ant	R3 ant	R4 ant	R5 ant	R6 ant	R7 ant	R8 ant	R9 ant	R10 ant	R11 ant	R12 ant
qn[n]	x	x	x	x	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
qn[n-1]	x	x	x	x	S0 2ant	S1 2ant	S2 2ant	S3 2ant	S4 2ant	S5 2ant	S6 2ant	S7 2ant	S8 2ant	S9 2ant	S10 2ant	S11 2ant	S12 2ant
qn[n-2]	x	x	x	x	S0 2ant	S1 2ant	S2 2ant	S3 2ant	S4 2ant	S5 2ant	S6 2ant	S7 2ant	S8 2ant	S9 2ant	S10 2ant	S11 2ant	S12 2ant
qn+M[n]	x	x	x	x	R0 ant	R1 ant	R2 ant	R3 ant	R4 ant	R5 ant	R6 ant	R7 ant	R8 ant	R9 ant	R10 ant	R11 ant	R12 ant
qn+M[n-1]	x	x	x	x	R0 3ant	R1 3ant	R2 3ant	R3 3ant	R4 3ant	R5 3ant	R6 3ant	R7 3ant	R8 3ant	R9 3ant	R10 3ant	R11 3ant	R12 3ant
qn+M[n-2]	x	x	x	x	R0 3ant	R1 3ant	R2 3ant	R3 3ant	R4 3ant	R5 3ant	R6 3ant	R7 3ant	R8 3ant	R9 3ant	R10 3ant	R11 3ant	R12 3ant
qn[n]-cs,0	x	x	x	x	M50	M51	M52	M53	M54	M55	M56	M57	M58	M59	M60	M61	M62
qn+M[n]-cs	x	x	x	x	M50 ant	M51 ant	M52 ant	M53 ant	M54 ant	M55 ant	M56 ant	M57 ant	M58 ant	M59 ant	M60 ant	M61 ant	M62 ant
tn[n]	x	x	x	x	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
tn+M[n]	x	x	x	x	M0 ant	M1 ant	M2 ant	M3 ant	M4 ant	M5 ant	M6 ant	M7 ant	M8 ant	M9 ant	M10 ant	M11 ant	M12 ant
tn[n+1]	x	x	x	x	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
tn+M[n+1]	x	x	x	x	M0 ant	M1 ant	M2 ant	M3 ant	M4 ant	M5 ant	M6 ant	M7 ant	M8 ant	M9 ant	M10 ant	M11 ant	M12 ant
sn[n]	x	x	x	x	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
sn[n+1]	x	x	x	x	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
sn[n+2]	x	x	x	x	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
em[n]	x	x	x	x	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12

Figura 62. Cronograma del conjunto módulo Matriz y Filtro para la transmisión.

5.5.4. Arquitectura del receptor

En el caso de la arquitectura del receptor, no se pueden realizar las mismas adaptaciones de sistema para ahorrar recursos y latencia en la arquitectura, debido al orden inverso de aparición de los módulos. Por lo tanto, la unión que se producía entre la fase s3 de la DCT y el módulo Matriz no se puede realizar, aumentando la latencia final. A su vez, la unión que se producía entre la fase s1 de la DCT y el módulo Theta se modifica, realizándose para el receptor estas operaciones por separado. En la Figura 63 se muestra la arquitectura realizada para el banco de filtros y las matrices **I** y **J** en el receptor.

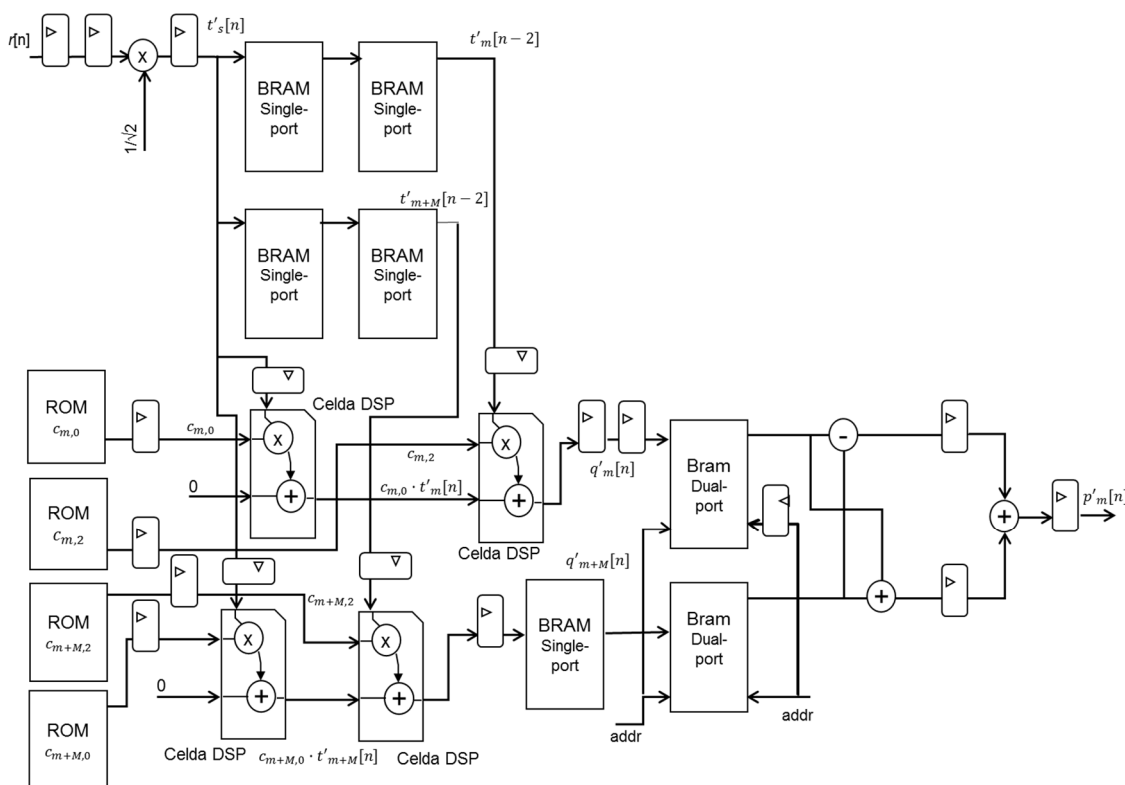


Figura 63. Diagrama de bloques de la arquitectura realizada en el receptor.

Una vez se ha mostrado la arquitectura realizada tanto para el módulo Matriz como para el banco de filtros en el receptor, se procede a mostrar un cronograma del funcionamiento de ambos, que se puede observar en la Figura 64. La señal de entrada $r[n]$ entra sincronizada con la señal CE_M (muestra de reloj 1). Con esta señal se generan las direcciones de memoria para las memorias que almacenan los coeficientes de los filtros (muestra de reloj 2). Después de las operaciones intermedias del filtro, se obtiene a la salida la señal $q'_s[n]$ (muestra de reloj 9). Esta señal pasa a las memorias para poder realizar las operaciones de las matrices **I** y **J**, obteniendo la señal $q'_m[n]$ (muestra de reloj 13). Por último, se sincroniza la señal $p'_m[n]$ con la señal $next_in$ (muestra de reloj 14).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CEM																	
r[n]	x	din0	din1	din2	din3	din4	din5	din6	din7	din8	din9	din10	din11	din12	din13	din14	din15
t[s[n]]	x	x	x	din0	din1	din2	din3	din4	din5	din6	din7	din8	din9	din10	din11	din12	din13
addr	x	x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
cs0	x	x	x	cte0	cte1	cte2	cte3	cte4	cte5	cte6	cte7	cte8	cte9	cte10	cte11	cte12	cte13
cs0	x	x	x	cte0	cte1	cte2	cte3	cte4	cte5	cte6	cte7	cte8	cte9	cte10	cte11	cte12	cte13
t[s[n]]	x	x	x	din0	din1	din2	din3	din4	din5	din6	din7	din8	din9	din10	din11	din12	din13
cs2	x	x	x	cte0	cte1	cte2	cte3	cte4	cte5	cte6	cte7	cte8	cte9	cte10	cte11	cte12	cte13
cs2	x	x	x	cte0	cte1	cte2	cte3	cte4	cte5	cte6	cte7	cte8	cte9	cte10	cte11	cte12	cte13
t[s[n-2]]	x	x	x	x	x	x	din0	din1	din2	din3	din4	din5	din6	din7	din8	din9	din10
cm,0,t[s[n]]	x	x	x	x	x	x	sum0	sum1	sum2	sum3	sum4	sum5	sum6	sum7	sum8	sum9	sum10
q[s[n]]	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
q[s[n-1]]	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
q[s[n-2]]	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
datol	x	x	x	x	x	x	x	x	x	x	x	x	datol0	datol1	datol2	datol3	datol4
multiplexor																	
q[n[n]]	x	x	x	x	x	x	x	x	x	x	x	x	datol0	datol1	datol2	datol3	datol4
next_in(DCT)																	
p[n[n]]	x	x	x	x	x	x	x	x	x	x	x	x	datol0	datol1	datol2	datol3	datol4

Figura 64. Cronograma de la arquitectura del módulo Matriz y banco de filtros en el receptor.

5.5 Arquitectura del sistema

Como se ha mencionado, la arquitectura de la DCT no sufre cambios, a excepción de los coeficientes, ya que la multiplicación por la constante Θ pasa a implementarse por separado de la DCT. En la Figura 65 se muestra la arquitectura desarrollada para el receptor.

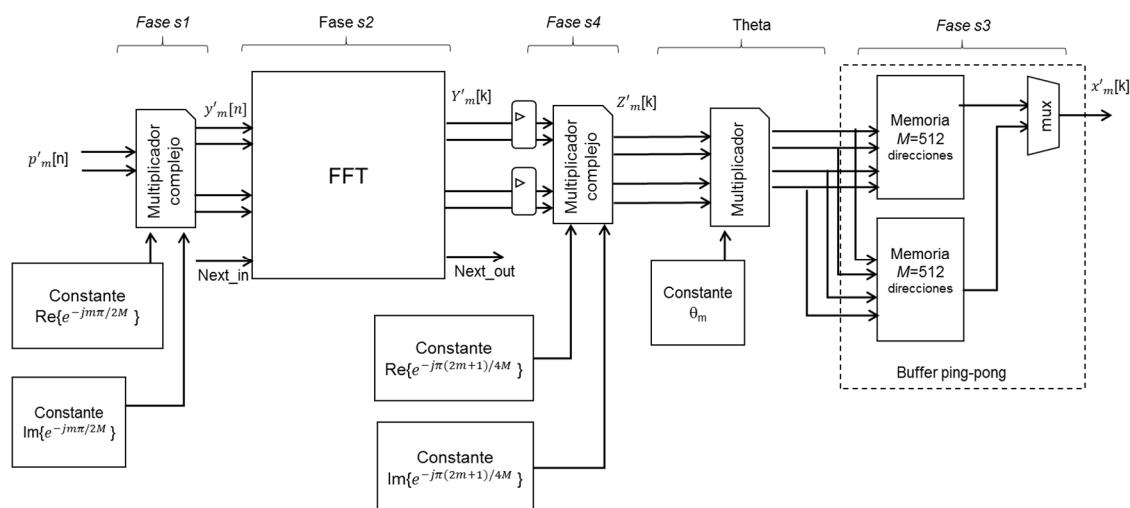


Figura 65. Diagrama de bloques de la arquitectura del módulo DCT y multiplicador Θ en el receptor.

Al igual que para el conjunto matriz-banco de filtros, se muestra en la Figura 66 el diagrama de tiempos de la arquitectura en el receptor. Este cronograma es muy parecido al que se explicaba en la DCT del transmisor, y es debido a que ambas DCT son iguales. Primero se recibe la señal $p'_m[n]$, sincronizada con la señal CE_M (muestra de reloj 1). Con esta señal se comienza a generar las direcciones y se activa el enable de las memorias que almacenan los coeficientes $e^{-\frac{j\pi m}{2M}}$ de la primera multiplicación (100) (muestra de reloj 1). En la salida de esta multiplicación se obtiene $y'_m[n]$ (muestra de reloj 7), la cual se sincroniza con la señal $next_in$, para ser procesada por la FFT (muestra de reloj 9).

A la salida de la FFT, una vez se activa $next_out$, se obtiene la señal $Y'_m[k]$ (muestra de reloj 13). Esta señal se multiplica por el coeficiente $e^{-\frac{j\pi(2m+1)}{4M}}$ de la segunda multiplicación (103) (muestra de reloj 15). La salida que se produce $Z'_m[k]$ se multiplica por la constante Θ_m (96) (muestra de reloj 20). La salida de la multiplicación por Θ_m se introduce en las memorias ping-pong para realizar el reordenamiento de la trama de datos (102) (muestra de reloj 22). Por último, a la salida del reordenamiento se tiene la señal $X'_m[k]$ (muestra de reloj 24).

CEM	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25		
p/m/0	x		Dm0	Dm1	Dm2	Dm3	Dm4	Dm5	Dm6	Dm7	Dm8																	
en																												
Addr1	x	0	1	2	3	4	5	6	7	8	9																	
coeficientes	x		Coef0	Coef1	Coef2	Coef3	Coef4	Coef5	Coef6	Coef7	Coef8																	
Mult	x			coef0	coef1	coef2	coef3	coef4	coef5	coef6	coef7																	
p/m/n-1	x			din0	din1	din2	din3	din4	din5	din6	din7																	
Y/m/0	x																											
Y/m/n-1	x																											
Y/m/n-2	x																											
next in																												
next out																												
Y/m/8	x																											
Y/m/k-1	x																											
Y/m/k-2	x																											
coeficientes	x																											
mult2	x																											
Zm/8	x																											
Zm/k-1	x																											
Thera	x																											
sal Thera	x																											
addA	x																											
addB	x																											
Zm/8	x																											
Zm/k-1	x																											

Figura 66. Cronograma de la DCT del receptor FBMC.

5.6. Sistema completo basado en FBMC

5.6.1. Estudio de la precisión finita

La Tabla 22 muestra la representación en coma fija definida para cada módulo y señal en el diseño propuesto, no sólo para el transmisor sino también para el receptor. El ancho de palabra más común en el sistema es de 18 bits, fijado por el ancho de palabra de una de las entradas de las celdas multiplicadoras (DSP48E1) disponibles en la FPGA.

Tabla 22. Representación en coma fija del ancho de palabra para la arquitectura basada en FBMC.

Módulo	Parámetro	No. bits global	Bits parte fraccional
Entrada Tx	$V_m[k]$	18	16
DCT Tx	Entrada $V_m[k]$	18	16
	Constante $e^{-jm\pi/2M}$	25	23
	Multiplicador eq. (3)	36	32
	$Y_m[k]$	18	16
	$Y_m[k_0]$	18	8
	$y_m[n_0]$	18	8
	$z_m[n]$	18	8
	Constante $e^{-j\pi(2m+1)/4M}$	25	23
	Multiplicador eq. (6)	36	32
	Salida $p_m[n]$	18	12
Matriz I y J Tx	Datos internos	18	12
	Salida $q_m[n]$	18	12
Filtro polifásico Tx	Entrada $q_m[n]$	18	12
	Coefficientes c	25	23
	Producto	36	28
	Acumulador	36	28
	Salida $t_s[n]$	18	12
	Sumador final de salida	18	12
Salida Tx	Constante $1/\sqrt{2}$	25	23
	Salida $e[n]$	18	12
Entrada Rx	Entrada $r[n]$	16	15
	Constante $1/\sqrt{2}$	25	23
	Salida $t'_s[n]$	18	16
Filtro polifásico Rx	Entrada $t'_s[n]$	18	16
	Coefficientes c	25	23
	Producto	36	32
	Acumulador	36	32
	Salida $q'_s[n]$	18	16
Matriz I y J Rx	Datos internos	18	16
	Salida $p'_s[n]$	18	16
DCT Rx	Entrada $p'_m[n]$	18	16
	Constante $e^{-jm\pi/2M}$	25	23
	Multiplicador eq. (3)	36	32
	$y_m[n]$	18	16
	$y_m[n_0]$	18	15
	$Y_m[k_0]$	18	15
	$Z_m[k]$	18	15
	Constante $e^{-j\pi(2m+1)/4M}$	25	23
	Multiplicador eq. (6)	36	31
	Salida $V'_m[k]$	18	16
Salida Rx	$V'_m[k]$	18	16

Con estos valores de tamaño de ancho de palabra, los diferentes módulos del sistema tienen que ser evaluados y comparados con un modelo en coma flotante. Las simulaciones se han llevado a cabo proporcionando una señal de entrada aleatoria $v_m[n]$, de 1000 muestras de longitud y normalizada al máximo rango de entrada $[-1,+1]$, para cada uno de los $M=512$ canales de entrada. La Tabla 23 describe el error medio absoluto y la desviación estándar para algunas señales intermedias de la arquitectura. Como puede observarse, desde un punto de vista global, el error medio absoluto es de 0.0023 para el transmisor, mientras que en el receptor es 0.0034. Es importante destacar que la FFT es la que tiene una influencia más remarcable en las cifras finales.

Tabla 23. Error medio absoluto y desviación estándar para las señales intermedias en la arquitectura propuesta FBMC, asumiendo un rango de entrada de $[-1,+1]$ en $v_m[k]$.

Módulo	Parámetro	Error medio	Desviación estándar
Entrada Tx	$V_m[k]$	0	0
DCT Tx	$p_m[n]$	0.0022	0.0020
Matriz I y J Tx	$q_s[n]$	0.0032	0.0028
Filtro polifásico Tx	$t_s[n]$	0.0033	0.0028
Salida Tx	$e[n]$	0.0023	0.0020
Entrada Rx	$r[n]$	0.0019	0.0014
Filtro polifásico Rx	$q'_s[n]$	0.0012	$0.9665 \cdot 10^{-3}$
Matriz I y J Rx	$p'_s[n]$	0.0027	0.0020
Salida Rx	$V'_m[k]$	0.0034	0.0021

5.6.2. Consumo de recursos

Para obtener el consumo de multiplicadores C_m^{Tx} en el transmisor, se realizará una suma de los recursos utilizados por cada módulo calculado previamente. Los parámetros se han fijado para $R_p^{DCT}=2$ y $R_p^F=2$. De esta manera se obtiene (134):

$$C_m^{Tx} = C_m^{DCT} + C_m^F = 48 + 5 = 53 \text{ multiplicadores} \quad (134)$$

A su vez, se realiza la misma operación para obtener los multiplicadores C_m^{Rx} empleados en el receptor, obteniéndose (135):

$$C_m^{Rx} = C_m^{DCT} + C_m^F + C_m^{Theta} = 48 + 5 + 2 = 55 \text{ multiplicadores} \quad (135)$$

5.6.3. Latencia

Como se ha comentado, la latencia se muestra para una mejor definición del diseño y no tiene impacto en el mismo. En este caso, la latencia L^{Tx} , al igual que sucedía con el consumo de recursos, se obtiene sumando las distintas latencias de los módulos que componen el transmisor, obteniéndose (136):

$$L^{Tx} = L^{DCT} + L^{Matriz} + L^F = 1256 + 4 + 11 = 1271 \text{ ciclos} \quad (136)$$

Recuérdese que la fase s3 de la DCT y el módulo Matriz se implementan de manera conjunta para reducir el número de ciclos de la latencia, al igual que el módulo Θ y la fase s1 de la DCT.

En el caso del receptor, no se puede realizar la unión de la fase s3 de la DCT con el módulo de las matrices **I** y **J**, ni el módulo Θ con la fase s1 de la misma; por lo que, la latencia L_{Rx} obtenida es (137):

$$L^{Rx} = L^F + L^{Matriz} + L^{DCT} + L^{Theta} = 11 + 260 + 1256 + 2 = 1785 \text{ ciclos} \quad (137)$$

Como se observa, debido a la imposibilidad de unir módulos, el número de ciclos de latencia es mayor que en el transmisor.

5.6.4. Comparativa entre los distintos modelos y la implementación real

Una vez se ha obtenido los datos del estudio del sistema completo, se procede a realizar una comparación del sistema modelado en VHDL con el sistema modelado en coma flotante y el sistema modelado en coma fija. Para la realización de la comparación entre los tres modelos se ha empleado en la entrada el peor caso, el valor Θ . Este valor es el peor caso debido a que provoca que la FFT genere una delta de valor M (valor máximo en la FFT). Se han realizado tres comparaciones entre los distintos modelos. La primera es la comparativa entre el modelo en coma flotante y coma fija, después una comparativa entre el modelo en coma fija y la implementación en la FPGA y por último una comparativa entre el modelo en coma flotante y la implementación en la FPGA. En la Tabla 24 se puede observar el error y la desviación que existe en las comparativas.

Tabla 24. Comparativa del error absoluto y la desviación típica del modelo en coma flotante, fija y arquitectura vhdl.

ERROR	FLOTANTE-FIJA		FIJA-VHDL		FLOTANTE-VHDL	
	Error	Desviación	Error	Desviación	Error	Desviación
THETA	0	0	0	0	0	0
DCT	0.0020	$2.7262 \cdot 10^{-17}$	$8.5783 \cdot 10^{-4}$	0	0.0019	$2.4949 \cdot 10^{-17}$
MATRIZ I-J	0.0031	$9.8073 \cdot 10^{-5}$	0.0013	$4.0391 \cdot 10^{-5}$	0.0029	$9.0623 \cdot 10^{-5}$
FILTRO	0.0031	$1.1739 \cdot 10^{-4}$	0.0013	$5.0286 \cdot 10^{-5}$	0.0029	$1.0748 \cdot 10^{-4}$
EMISOR	0.0022	$8.5113 \cdot 10^{-5}$	$9.0862 \cdot 10^{-4}$	$3.7122 \cdot 10^{-5}$	0.0021	$7.6425 \cdot 10^{-5}$
ENTRADA RECEPTOR	0.0018	$8.8827 \cdot 10^{-5}$	$6.3939 \cdot 10^{-4}$	$4.1155 \cdot 10^{-5}$	0.0017	$8.2720 \cdot 10^{-5}$
FILTRO	0.0011	$8.5521 \cdot 10^{-5}$	$3.8883 \cdot 10^{-4}$	$3.6816 \cdot 10^{-5}$	0.0011	$8.1072 \cdot 10^{-5}$
MATRIZ I-J	0.0027	$1.8486 \cdot 10^{-4}$	$9.4368 \cdot 10^{-4}$	$8.1792 \cdot 10^{-5}$	0.0026	$1.7599 \cdot 10^{-4}$
DCT	0.0038	$2.4123 \cdot 10^{-4}$	0.0013	$1.1191 \cdot 10^{-4}$	0.0038	$2.3645 \cdot 10^{-4}$
SALIDA RECEPTOR	0.0038	$2.4108 \cdot 10^{-4}$	0.0013	$1.1233 \cdot 10^{-4}$	0.0038	$2.3632 \cdot 10^{-4}$

El error entre el sistema en coma flotante y el sistema en coma fija se debe al efecto de cuantificación, mientras que el error entre el sistema en coma fija y el sistema implementado en VHDL se debe a que el modelo no caracteriza el comportamiento interno exacto del core de la FFT empleada, creada por Spiral [111].

5.6.5. Resultados obtenidos para el modelo en coma fija y el modelo en coma flotante

En el apartado de resultados, se va a realizar una comparación entre el modelo en coma flotante y el modelo en coma fija, teniendo en cuenta la relación señal-ruido (SNR) del transmultiplexor, el error cuadrático medio, la relación señal-ruido de pico y el error

máximo absoluto como se define en [112]. La Tabla 25 muestra estas figuras de mérito, para una simulación con una entrada aleatoria $v_m[n]$ de 1000 muestras de longitud y una amplitud normalizada al máximo valor de entrada $[-1,+1]$.

Tabla 25. Comparación entre los modelos en coma fija y coma flotante de la arquitectura para FBMC.

	Coma flotante	Coma fija
SNR	137.3148 dB	44.5217 dB
error cuadrático medio	$6.1865 \cdot 10^{-15}$	$2.7016 \cdot 10^{-5}$
Pico SNR	142.0791 dB	49.2860 dB
Error absoluto máximo	$1.4214 \cdot 10^{-7}$	$9.4000 \cdot 10^{-3}$

Se observa una pérdida de 93dB en la relación señal-ruido, aunque esta pérdida parece excesiva hay que mencionar que gran parte de este error proviene de la limitación en el ancho de palabra de los conversores. El máximo ancho de palabra es de 16 bits aplicado sólo en el caso del modelo en coma fija. El convertir en este punto los valores de entrada de coma flotante a coma fija hace que se produzca una pérdida de más de 50dB.

5.7. Estudio para la mejora de los filtros polifásicos

Una vez se ha desarrollado el diseño completo, se ha analizado una mejora en la arquitectura de los filtros polifásicos. Para ello, se propuso la utilización de arquitectura en celosía, que a priori, mejoran en el SNR al filtro directo que se implementaba. Para evaluar cuál de las propuestas era mejor, se ha procedido a realizar un estudio comparativo entre distintas arquitecturas de filtros, para su integración dentro de la arquitectura de la técnica de acceso basada en banco de filtros (FBMC). Dentro de esta comparativa, se van a estudiar los filtros directos y los filtros en celosía genérico, tipo I y tipo II. Para implementar esta técnica de acceso al medio se va a emplear una FPGA Virtex 6 xc6vlx240t [110], por lo que se ha de tener en cuenta las limitaciones de implementación que esta FPGA presenta, tanto en el máximo datapath disponible, como en las celdas lógicas que posee en su interior para desarrollar los algoritmos.

Para la realización de este estudio, primero se ha realizado una comparativa del error de representación en coma fija (cuantificación), tanto de coeficientes, como de los filtros (para transmisión y recepción). Más tarde, se ha realizado un estudio de consumo de recursos para cada versión de filtro y, por último, un estudio de las variaciones en la relación señal-ruido obtenidas en cada caso.

5.7.1. Efecto de la cuantificación del coeficiente

Para el estudio del efecto de cuantificación en los coeficientes, se ha considerado los coeficientes para los $S=2 \cdot M=2 \cdot 512=1024$ filtros, y a partir de ellos, se ha calculado la media, máximo y desviación típica del error producido en los coeficientes por el efecto de la coma fija respecto a los coeficientes en coma flotante. Para esto, se ha tenido únicamente en cuenta el efecto de la cuantificación en los coeficientes del filtro. El ancho de palabra considerado es el máximo disponible que se corresponde con 25 bits. En la Tabla 26 se

muestra la comparativa de los errores de cuantificación en los coeficientes de los distintos tipos de filtros.

Tabla 26. Comparación de errores de cuantificación en los coeficientes.

---- error medio ----	
Celosía general:	$7.6240 \cdot 10^{-6}$
Celosía tipo 1:	0.0313
Celosía tipo 2:	$7.6616 \cdot 10^{-6}$
Filtro directo:	$2.3522 \cdot 10^{-4}$
---- error máximo ----	
Celosía general:	$1.5255 \cdot 10^{-5}$
Celosía tipo 1:	0.0625
Celosía tipo 2:	$1.5255 \cdot 10^{-5}$
Filtro directo:	$6.3136 \cdot 10^{-4}$
---- desviación típica ----	
Celosía general:	$4.4795 \cdot 10^{-6}$
Celosía tipo 1:	0.0194
Celosía tipo 2:	$4.3926 \cdot 10^{-6}$
Filtro directo:	$1.1554 \cdot 10^{-4}$

Como se puede observar, el error producido por el efecto de cuantificación en los coeficientes cuantificados del filtro directo es mayor que en los coeficientes del filtro de celosía general y tipo II, siendo el tipo I peor debido a que sus coeficientes tienen mayor rango dinámico, y esto empeora el efecto de la cuantificación para este filtro.

5.7.2. Efecto de la precisión finita sobre la estructura del filtro

Para el estudio del efecto de la cuantificación en la estructura global de los filtros, analizando sus salidas, se ha diferenciado el filtro de la transmisión, del de la recepción, puesto que, en la transmisión se produce una suma en pares de las salidas de los filtros. También se ha considerado para este estudio el valor máximo posible de anchura de datapath para cada punto intermedio del filtro, ya que este valor afecta directamente al efecto de cuantificación. A continuación, se muestran las arquitecturas para cada una de las opciones propuestas. Dentro de las figuras, entre corchetes, se muestra la dimensión para cada una de las señales intermedias. Esta dimensión es la que se empleará en las simulaciones para la cuantificación en esas señales intermedias. Las distintas dimensiones en las señales intermedias se deben a que se está considerando la FPGA en la que se van a implementar (Virtex 6) y que ésta dispone de unas celdas multiplicadoras específicas, que incluye un multiplicador y un sumador en cascada. Esta celda multiplicadora, DSP48E1 [109] dispone de un datapath interno a la salida de la multiplicación de 36 bits, lo que permite que en ciertas señales intermedias se pueda disponer de hasta 36 bits para la cuantificación. A continuación se muestra el diagrama de bloques para el filtro en celosía genérico para transmisión (Figura 67) y para recepción (Figura 68).

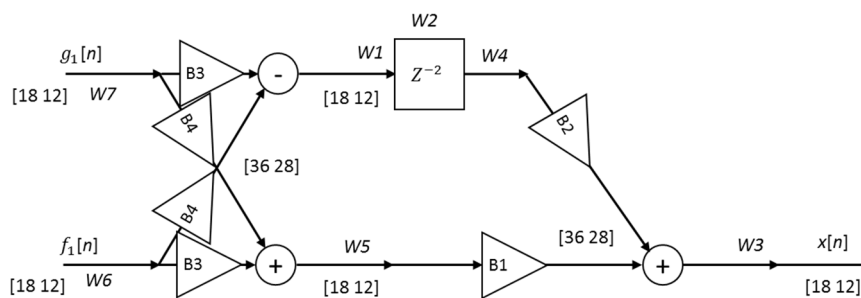


Figura 67. Diagrama de bloques de filtro en celosía genérico para transmisión.

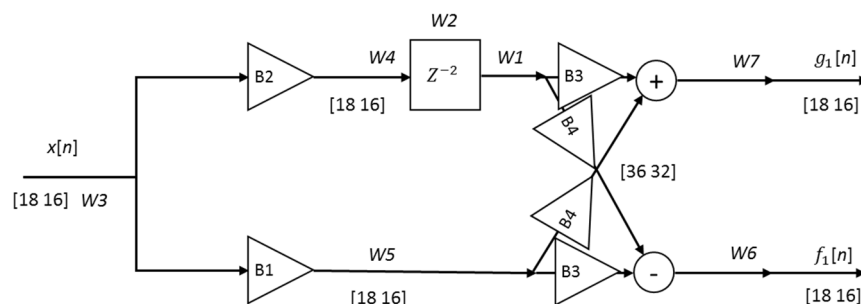


Figura 68. Diagrama de bloques de filtro en celosía genérico para recepción.

El caso del filtro de celosía tipo I es distinto a los demás, ya que los coeficientes presentan mayor rango dinámico, con lo que la cuantificación se debe ajustar a este rango y de esta manera se empeora su efecto. En la Figura 69 se muestra el filtro diseñado para transmisión y en la Figura 70 para recepción.

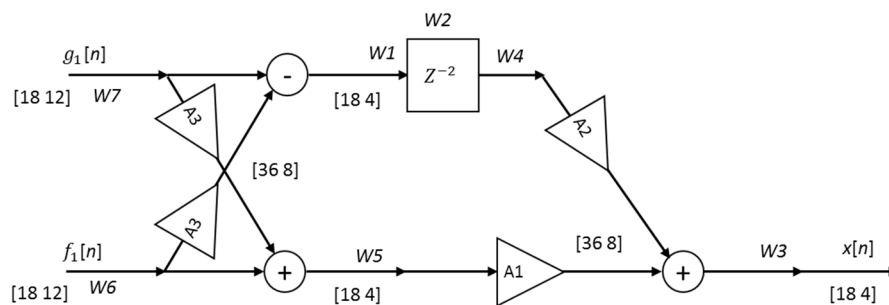


Figura 69. Diagrama de bloques de filtro en celosía tipo I para transmisión.

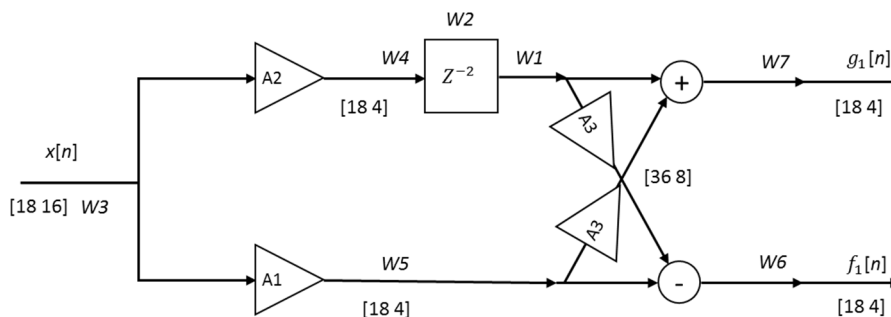


Figura 70. Diagrama de bloques de filtro en celosía tipo I para recepción.

5.7 Estudio para la mejora de los filtros polifásicos

En la Figura 71 se muestra el diagrama de bloques del filtro en celosía tipo II para la transmisión, mientras que en la Figura 72 se muestra el diagrama de bloques del filtro en celosía tipo II para la recepción.

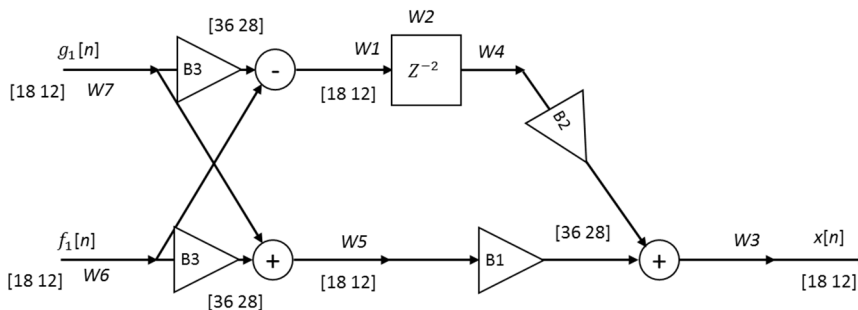


Figura 71. Diagrama de bloques de filtro en celosía tipo II para transmisión.

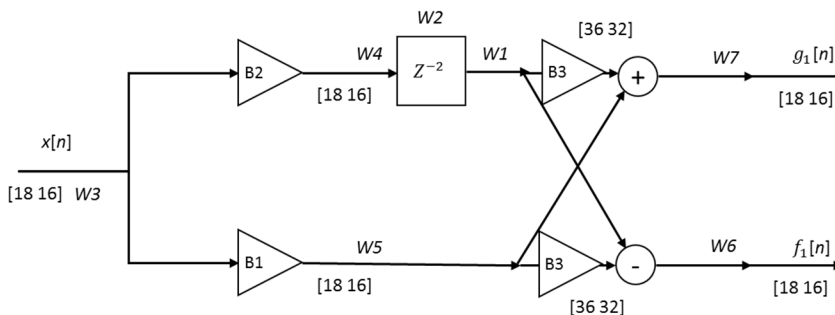


Figura 72. Diagrama de bloques de filtro en celosía tipo II para recepción.

El filtro directo es idéntico para transmisión y para recepción, ya que su arquitectura se basa en una multiplicación más acumulación. Esto permite un diseño modular, fácilmente replicable. En la Figura 73 se muestra el filtro directo para transmisión y en la Figura 74 se muestra el filtro directo para la recepción.

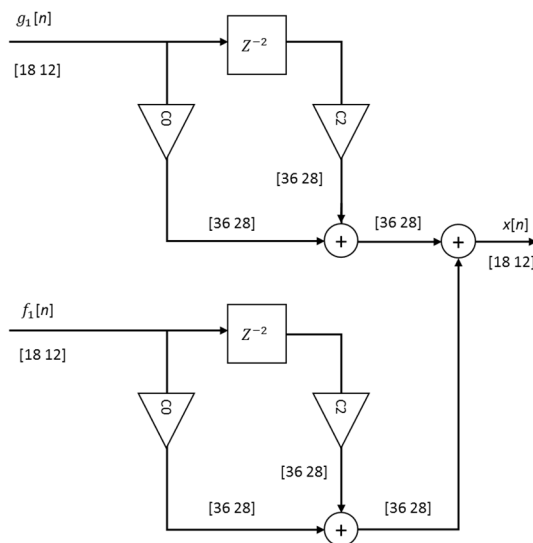


Figura 73. Diagrama de bloques de filtro directo para transmisión.

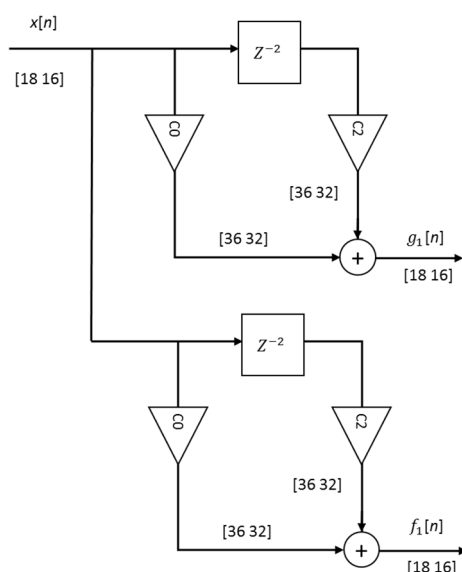


Figura 74. Diagrama de bloques de filtro directo para recepción.

En los distintos diagramas de bloques se ha observado distintos parámetros (A1, A2, A3, B1, B2, B3, B4, C0, C2). Estos se corresponden con los distintos coeficientes aplicados en las distintas arquitecturas. Para estos coeficientes se ha empleado un ancho de palabra de 25 bits.

Una vez se ha mostrado la arquitectura de las propuestas así como los tamaños de datapath considerados en cada caso para las señales intermedias, se procede a mostrar los resultados estadísticos del error de cuantificación obtenidos. Estos errores han sido estimados generando señales a la entrada aleatorias, con una amplitud entre ± 1 , o lo que es lo mismo, el máximo rango de entrada posible. Se han generado 200 simulaciones, formadas por 1024 señales de entrada y se ha obtenido la media, máximo y desviación típica de los $S=1024$ filtros. En la Tabla 27 se muestra el error de cuantificación con cada una de las arquitecturas para la salida del transmisor y en la Tabla 28 el error de cuantificación con cada una de las arquitecturas para la salida del receptor.

Tabla 27. Comparación del error de cuantificación evaluado a la salida del transmisor.

---- error medio ----	
Celosía general:	$1.2076 \cdot 10^{-4}$
Celosía tipo 1:	0.6170
Celosía tipo 2:	$1.2092 \cdot 10^{-4}$
Filtro directo:	$5.9500 \cdot 10^{-5}$
---- error máximo ----	
Celosía general:	$4.9740 \cdot 10^{-4}$
Celosía tipo 1:	$1.9748 \cdot 10^2$
Celosía tipo 2:	$4.8318 \cdot 10^{-4}$
Filtro directo:	$1.8565 \cdot 10^{-4}$
---- desviación típica ----	
Celosía general:	$8.7964 \cdot 10^{-5}$
Celosía tipo 1:	4.6381
Celosía tipo 2:	$8.8328 \cdot 10^{-5}$
Filtro directo:	$3.9439 \cdot 10^{-5}$

Tabla 28. Comparación del error de cuantificación en la salida del receptor.

---- error medio ----	
<i>Celosía general:</i>	$7.0047 \cdot 10^{-6}$
<i>Celosía tipo 1:</i>	1.7950
<i>Celosía tipo 2:</i>	$8.9552 \cdot 10^{-6}$
<i>Filtro directo:</i>	$5.0663 \cdot 10^{-6}$
---- error máximo ----	
<i>Celosía general:</i>	$2.7127 \cdot 10^{-5}$
<i>Celosía tipo 1:</i>	$2.7035 \cdot 10^2$
<i>Celosía tipo 2:</i>	$2.7464 \cdot 10^{-5}$
<i>Filtro directo:</i>	$1.4346 \cdot 10^{-5}$
---- desviación típica ----	
<i>Celosía general:</i>	$5.2871 \cdot 10^{-6}$
<i>Celosía tipo 1:</i>	$1.3048 \cdot 10^1$
<i>Celosía tipo 2:</i>	$6.1601 \cdot 10^{-6}$
<i>Filtro directo:</i>	$3.7021 \cdot 10^{-6}$

Como se puede observar, tanto para el caso de transmisión como para el caso de recepción, el filtro directo obtiene mejores cifras. Esto es debido a que se dispone de un tamaño mayor de datapath en las señales intermedias del filtro directo. Esto principalmente permite que se produzca un error menor por el efecto de la cuantificación en estas multiplicaciones característica que no se puede conseguir en los filtros de celosía.

5.7.2.1. Ampliación del datapath

Una vez se han realizado las simulaciones con el tamaño de datapath considerado, se ha decidido aumentar éste para poder mejorar los errores de cuantificación obtenidos. El tamaño de datapath empleado es de 18 bits, que se corresponde con el máximo tamaño de datapath de entrada de los recursos. Ahora, se ha considerado duplicar el tamaño del datapath, evaluándose los nuevos resultados. La generación de la entrada es idéntica a la anterior, calculándose los errores medios, máximos y desviaciones típicas para los $S=1024$ filtros con cada tipo de arquitectura. A continuación se exponen los mismos.

Tabla 29. Error de cuantificación evaluado a la salida del transmisor para cada una de las arquitecturas.

---- error medio ----	
<i>Celosía general:</i>	$4.6151 \cdot 10^{-10}$
<i>Celosía tipo 1:</i>	$1.8342 \cdot 10^{-5}$
<i>Celosía tipo 2:</i>	$4.5217 \cdot 10^{-10}$
<i>Filtro directo:</i>	$2.2218 \cdot 10^{-10}$
---- error máximo ----	
<i>Celosía general:</i>	$1.7588 \cdot 10^{-9}$
<i>Celosía tipo 1:</i>	0.0027
<i>Celosía tipo 2:</i>	$1.5644 \cdot 10^{-9}$
<i>Filtro directo:</i>	$6.6419 \cdot 10^{-10}$
---- desviación típica ----	
<i>Celosía general:</i>	$3.4616 \cdot 10^{-10}$
<i>Celosía tipo 1:</i>	$1.4912 \cdot 10^{-4}$
<i>Celosía tipo 2:</i>	$3.4953 \cdot 10^{-10}$
<i>Filtro directo:</i>	$1.4835 \cdot 10^{-10}$

Tabla 30. Error de cuantificación evaluado a la salida del receptor para cada una de las arquitecturas.

---- error medio ----	
<i>Celosía general:</i>	$2.6903 \cdot 10^{-11}$
<i>Celosía tipo 1:</i>	$2.7261 \cdot 10^{-5}$
<i>Celosía tipo 2:</i>	$6.2451 \cdot 10^{-10}$
<i>Filtro directo:</i>	$2.0041 \cdot 10^{-11}$
---- error máximo ----	
<i>Celosía general:</i>	$1.1244 \cdot 10^{-10}$
<i>Celosía tipo 1:</i>	0.0033
<i>Celosía tipo 2:</i>	$1.8370 \cdot 10^{-9}$
<i>Filtro directo:</i>	$5.7274 \cdot 10^{-11}$
---- desviación típica ----	
<i>Celosía general:</i>	$1.9476 \cdot 10^{-11}$
<i>Celosía tipo 1:</i>	$1.6316 \cdot 10^{-4}$
<i>Celosía tipo 2:</i>	$4.5203 \cdot 10^{-10}$
<i>Filtro directo:</i>	$1.4051 \cdot 10^{-11}$

Como se observa, al duplicar el tamaño en el datapath, el error de cuantificación disminuye. Pero como todas las arquitecturas amplían su datapath la arquitectura del filtro directo sigue siendo la mejor, ya que las señales intermedias siguen teniendo un mejor datapath. Este hecho se debe al empleo de las celdas DSP48E1, en la arquitectura de la FPGA, que permiten que la salida del multiplicador posea un datapath de mayor anchura, que las arquitecturas en celosía.

5.7.3. Estudio del consumo de recursos

Para el estudio del consumo de recursos de cada una de las opciones, se ha evaluado de forma teórica cada una de las arquitecturas a partir de los modelos en coma fija en Matlab, según las estructuras mostradas en las figuras anteriores. A partir de esos diagramas de bloques se ha obtenido una estimación del número de multiplicaciones, sumas y bloques de retardos empleados por cada propuesta. Esta estimación está realizada para la arquitectura de un solo filtro. Para la arquitectura de todo el banco, entra en juego el ratio de paralelismo R_p [116] empleado en el diseño, con lo que la estimación del número total de recursos empleados no es directa. A continuación en la Tabla 31 se muestra el número de operaciones empleadas para el diseño de un filtro para el transmisor y en la Tabla 32 para la recepción.

Tabla 31. Evaluación de las operaciones empleadas para el filtro de transmisión.

Filtro	Multiplicaciones	Sumas	Bloque retardo
<i>Celosía general</i>	6	3	1
<i>Celosía tipo I</i>	4	3	1
<i>Celosía tipo II</i>	4	3	1
<i>Directo</i>	4	3	2

Tabla 32. Evaluación de las operaciones empleadas para el filtro de recepción.

Filtro	Multiplicaciones	Sumas	Bloque retardo
<i>Celosía general</i>	6	2	1
<i>Celosía tipo I</i>	4	2	1
<i>Celosía tipo II</i>	4	2	1
<i>Directo</i>	4	2	2

Debe aclararse que el bloque z^{-2} realiza un retardo de dos ciclos de reloj, para de esta manera poder obtener el valor $x[n-2]$, referido al coeficiente $c_{s,2}$ del filtro implementado. Hay que destacar que el filtro directo genera sólo una salida para transmisión y tiene sólo una entrada para recepción. Por tanto, para obtener los mismos resultados que en los filtros de celosía, se necesitan dos estructuras de filtrado directo y una suma adicional, según se muestra en la Figura 74.

Una vez se ha obtenido el número de operaciones necesarias para cada filtro, se procede a calcular el número de celdas de la FPGA que se van a emplear. Debe destacarse que la Virtex6 a utilizar dispone de la celda DSP48E1 [109], que incluye un multiplicador y una sumador en cascada. De este modo, empleando este bloque dedicado, se puede realizar las operaciones para los filtros de una manera más efectiva. Por su parte, para la arquitectura del bloque de retardo, se emplearán bloques de memoria interna BRAM. En la Tabla 33 se muestra el consumo de ambos tipos de recursos para el diseño de un filtro para la transmisión, mientras en la Tabla 34 se recoge el mismo consumo para la recepción. Ambos consumos se representan para la arquitectura de un solo filtro. La extrapolación a todo el banco con S filtros dependerá del ratio de paralelismo fijado en la arquitectura.

Tabla 33. Estimación del consumo de recursos en el dispositivo Virtex6 para transmisión.

Filtro	DSP48E1	BRAM
<i>Celosía general</i>	6	1
<i>Celosía tipo I</i>	4	1
<i>Celosía tipo II</i>	4	1
<i>Directo</i>	4	1

Tabla 34. Estimación del consumo de recursos en el dispositivo Virtex6 para recepción.

Filtro	DSP48E1	BRAM
<i>Celosía general</i>	6	1
<i>Celosía tipo I</i>	4	1
<i>Celosía tipo II</i>	4	1
<i>Directo</i>	2	1

Como se puede observar en las tablas, los resultados obtenidos indican que, salvo el caso de celosía general, el consumo de recursos dentro de la FPGA es muy parecido. Además, a los valores obtenidos en el consumo de recursos, se añadiría otro importante factor para el diseño de la arquitectura: la regularidad. En este sentido, el diseño del filtro directo permite un mejor aprovechamiento y modularidad de las celdas DSP48E1 que el que ofrecen los filtros en celosía, como se puede observar en la Figura 74, donde la salida para transmisión se obtiene de la suma de dos filtros directos

5.7.4. Análisis de la relación señal-ruido en el diseño final

Una vez se han obtenido los efectos de cuantificación del banco de filtros independientemente, se va a implementar con el resto del diseño FBMC en coma fija, calculando posteriormente el efecto de cuantificación que se obtiene para cada uno de las propuestas de filtro abordadas. A continuación, en la Tabla 35 se pueden observar los

valores obtenidos, para una entrada generada de manera aleatoria. La amplitud de ésta es el máximo rango posible: de -1 a 1. Los datos obtenidos son la media obtenida por 1000 simulaciones.

Tabla 35. Parámetros obtenidos de simulación del FBMC cuantificado con distintos bancos de filtros.

	Directo	General	Tipo I	Tipo II
<i>SNR en media</i>	44.5121 dB	44.4816 dB	0.0143 dB	44.4743 dB
<i>Error cuadrático medio en media</i>	$2.2770 \cdot 10^{-5}$	$2.3540 \cdot 10^{-5}$	0.3328	$2.3556 \cdot 10^{-5}$
<i>Pico SNR en media</i>	49.2760 dB	49.2453 dB	4.7704 dB	49.2379 dB
<i>Máximo error en media</i>	0.0093	0.0094	0.9979	0.0094

Como se puede observar en la tabla, la arquitectura FBMC con filtros directos es mejor que las opciones de filtros en celosía, siendo la del filtro en celosía tipo I la peor, debido al mayor rango dinámico que posee.

5.8. Conclusiones de la transformada basada en banco de filtros

Se ha presentado el diseño de la técnica de acceso al medio multi-portadora basada en banco de filtros. Se ha presentado la arquitectura que se ha desarrollado, realizando un exhaustivo estudio tanto del consumo de recursos, como de la latencia generada. También se ha calculado el error que se introduce en el diseño debido al efecto de la precisión finita. Por último, se ha mostrado un estudio para obtener la mejor arquitectura para el banco de filtros polifásicos, con el fin de mejorar el SNR y reducir el error generado por cuantificación.

El diseño presenta una serie de características a destacar, que son:

La arquitectura realizada permite que modificando una serie de parámetros, se pueda procesar distinto número de subcanales, trabajar con una tasa de transmisión distinta o aumentar el número de datapaths empleados. Todo ello permite que la arquitectura se adapte a distintos tipos de transmisiones. A su vez, modificando el ratio de paralelismo R_p se puede modificar el número de entradas que son procesadas en paralelo. Esto permite modificar la arquitectura interna del diseño, lo cual hace que se modifique el número de recursos que se emplean en el diseño. Además, la modificación de la arquitectura hace que se modifique el tiempo de procesamiento, la latencia y throughput del diseño.

La arquitectura que se ha llevado a cabo permite que el sistema trabaje en tiempo real. Esto se consigue con un tiempo de procesamiento bajo, ya que se procesan los M subcanales de forma semi-paralela y obteniendo en la salida del receptor estos mismo datos transmitidos. Este tiempo de procesamiento bajo permite que el diseño pueda transmitir toda la trama de datos de entrada antes de que comience con la siguiente trama.

Optimización del error por efecto de la precisión finita: El estudio que se ha realizado, permite conocer el rango dinámico de cuantificación en cada punto del diseño, permitiendo que se distribuyan el mayor número de bits en la parte fraccionaria, sin que se pierda información de la parte entera, lo que implica una mayor resolución.

Capítulo 6

Técnica de acceso al medio basada en portadora única

Después de mostrar e implementar dos técnicas multi-portadora, se ha decidido implementar una técnica mono-portadora con igualación en el dominio de la frecuencia (SC-FDE, Single-Carrier Frequency Domain Equalizer) para realizar una comparativa más completa. Si bien en el desarrollo de esta tesis no se ha realizado el módulo de igualación, realizándose en su lugar una multiplicación por uno para cada subportadora. Para el desarrollo de esta técnica mono-portadora, se ha partido de la técnica de la Transformada Trigonométrica Discreta, pero traspasando la Transformada del Coseno Discreta de tipo IV del transmisor al receptor, para de esta manera, transmitir los datos con una única portadora. En la técnica de portadora única que se va a implementar se mantiene el uso de la extensión simétrica para ofrecer en el receptor el efecto de la convolución circular, que ofrece mayor robustez frente a interferencias entre portadoras (ICI). Esta extensión simétrica consiste en añadir tanto en la parte anterior, como en la posterior, parte de la trama de datos a transmitir, aunque en sentido inverso, para generar la convolución circular.

Al traspasar la DCT-IV del transmisor al receptor, se implementan dos DCT-IV contiguas en el receptor, empleándose una para pasar los datos al dominio de la frecuencia y otra para pasar los datos ecualizados al dominio del tiempo. Aunque la arquitectura de la DCT-IV es idéntica para la transformada directa e inversa, en este caso se va a diferenciar cada una de las DCT-IV implementadas dependiendo de su función. Según este criterio, la primera DCT-IV encargada de realizar el paso al dominio de la frecuencia se denomina

DCT, y la segunda DCT-IV encargada del paso al dominio del tiempo se denomina iDCT. Entre ellas, se inserta una etapa de igualación, para compensar el error producido por el efecto del canal, si bien en este estudio no ha sido incluida. En la Figura 75 se puede observar cómo queda implementado el transmisor de la técnica mono-portadora y en la Figura 76 la arquitectura del receptor de la técnica mono-portadora.

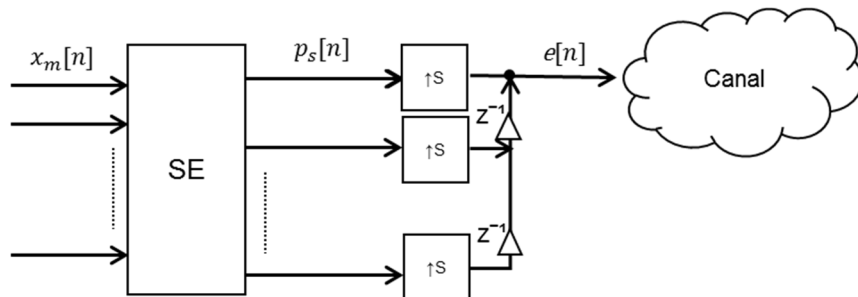


Figura 75. Diagrama de bloques del transmisor de la técnica de portadora única.

Como se observa, la señal $x_m[n]$ entra en el módulo que añade la extensión simétrica para generar, a partir de $x_m[n]$, la señal $p_s[n]$ a su salida, siendo $S=M+\alpha+\beta$ el nuevo tamaño de la trama de datos que se transmite en $e[n]$; donde M es el número de subcanales de la técnica de acceso al medio, α es el número de datos de la extensión simétrica anterior, y β es el número de datos de la extensión simétrica posterior.

La señal $r[n]$ que llega al receptor se paraleliza para obtener $q'_s[n]$. Ésta se envía al módulo que quita la extensión simétrica, con lo que se obtiene la señal $v'_m[n]$. La señal $v'_m[n]$ se envía a la DCT, de donde se genera $S'_m[k]$. A esta señal $S'_m[k]$ se le aplica el algoritmo de igualación deseado, aunque en este caso no se analizará, multiplicando cada señal $S'_m[k]$ por 1 en el módulo FDE. Después de la igualación la señal $P'_m[k]$ pasa a la iDCT, de la cual se obtiene $x'_m[n]$.

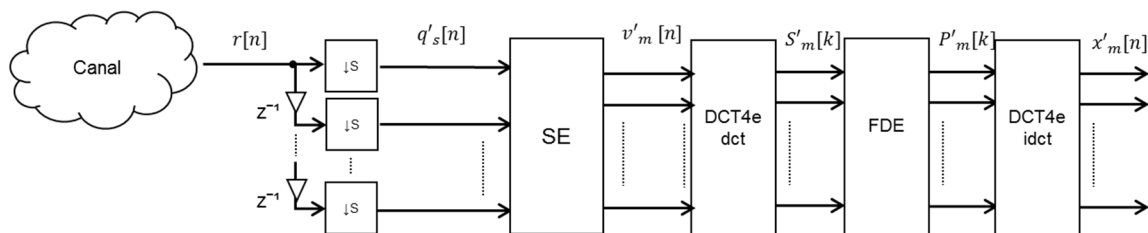


Figura 76. Diagrama de bloques del receptor de la técnica de portadora única.

Dado que, tanto la Transformada Discreta del Coseno (DCT-IV), como el módulo de extensión simétrica se han estudiado y evaluado previamente, en este apartado simplemente se van a recordar los parámetros más relevantes. En el caso de la extensión simétrica, tal y como sucedía en la DTT (apartado 4.1), se recuerda que tanto el valor de α como de β es de 32 muestras. La extensión simétrica duplica las α primeras muestras del array, colocándolas en orden inverso; y también duplica las β últimas muestras del array, colocándolas en orden inverso e invertidas. El valor de α y β indica que el tamaño de la trama de datos que se va a transmitir pasa de M muestras a la entrada al valor de $S=M+\alpha+\beta$ muestras a la salida.

En el caso de la DCT-IV, al igual que sucedía en la técnica basada en la Transformada Trigonométrica Discreta (DTT) y en la técnica basada en la multi-portadora con banco de filtros (FBMC), se ha decidido emplear la DCT con el algoritmo 1 desarrollado en el apartado 4.2. Esta DCT se caracteriza por estar compuesta por cuatro fases: la primera es una multiplicación por una constante $e^{-\frac{j m \pi}{2M}}$; la segunda es la FFT; la tercera propone un cambio de orden en la trama de datos; y la cuarta consiste en la multiplicación por otra constante $e^{-\frac{j \pi m}{4M}}$.

Los valores de latencia, throughput y consumo de recursos serán los mismos que los aplicados en los cálculos obtenidos previamente para cada una de las fases mencionadas, por lo que se pasará directamente a mostrar el resultado final de la latencia y el consumo de recursos de la DCT-IV completa. Recordando los valores para la DCT-IV que se obtuvieron en la arquitectura desarrollada para la Transformada Trigonométrica Discreta (DTT), se observa en (138) el número de multiplicadores C_m^{DCT} necesarios, y en (139) la latencia L^{DCT} :

$$C_m^{DCT} = 48 \text{ multiplicadores} \quad (138)$$

$$L^{DCT} = 1256 \text{ ciclos} \quad (139)$$

En el caso de la extensión simétrica, existe una diferenciación entre la parte del transmisor y la del receptor. En la parte del transmisor se añade la extensión simétrica, tanto en la parte posterior, como anterior. Si bien, en la parte posterior estos datos añadidos son invertidos, por lo que será necesario además de memorias, un multiplicador para realizar esta inversión. Por lo tanto, en el transmisor de la extensión simétrica se obtuvo el consumo de multiplicadores $C_m^{SE,Tx}$ en (140) y la latencia $L^{SE,Tx}$ (141) :

$$C_m^{SE,Tx} = 1 \text{ multiplicador} \quad (140)$$

$$L^{SE,Tx} = M + L^m \text{ ciclos} \quad (141)$$

Siendo $L^m=2$ ciclos de reloj, producidos por la multiplicación de la constante para la inversión y $M=512$ subcanales de transmisión.

En el caso del receptor de la extensión simétrica, sólo se realiza una extracción de esta extensión, por lo que no es necesario ninguna multiplicación $C_m^{SE,Rx}$ (142). Además debido a que no se emplean multiplicaciones la latencia es menor que en el caso del transmisor $L^{SE,Rx}$ (143).

$$C_m^{SE,Rx} = 0 \text{ multiplicadores} \quad (142)$$

$$L^{SE,Rx} = M \text{ ciclos} \quad (143)$$

Se recuerda que para todos los casos la latencia no tiene impacto en el diseño, solamente se emplea para tener una mejor definición de la arquitectura.

6.1. Arquitectura del sistema

La FPGA empleada para la implementación de la técnica de acceso mono-portadora es una FPGA Virtex 6 xc6vlx240t [110]. Además, se van a emplear los mismos parámetros para realizar la transmisión que los utilizados anteriormente. Estos parámetros son la tasa de transmisión $T_{Tx}=62.5\text{Msps}$, definida por el standard de PLC de banda ancha [27]; y el número de subcanales de entrada $M=512$. Esto supone que las restricciones a la arquitectura van a ser las mismas que anteriormente. Estas restricciones son el número máximo de multiplicadores disponibles $C_m^{FPGA}=768$ y la frecuencia de muestreo del conversor $f_s=62.5\text{MHz}$. Con estas restricciones se ha decidido fijar la frecuencia de funcionamiento del periférico donde se implementa la técnica de acceso a $f_{per}=62.5\text{MHz}$, lo que fija el ratio de frecuencia $R_f = \frac{f_s}{f_{per}} = 1$. Con este ratio se obtiene el tiempo de procesamiento máximo del array $M_c^{max}=512$ ciclos de reloj.

La forma de realizar la arquitectura va a ser de manera semi-paralela con un ratio de paralelismo de $R_p=2$, tal y como se ha estudiado en la arquitectura de las otras técnicas de acceso al medio. Este ratio indica cuántos datapath (o líneas de proceso) son realmente implementados en el diseño, para ser luego reutilizadas para procesar los M subcanales. Gracias a esta arquitectura semi-paralela, se consigue una optimización de los recursos, ya que son empleados para procesar varios subcanales a lo largo del tiempo.

6.1.1. Arquitectura del transmisor

Dado que en el transmisor solamente se realiza la extensión simétrica, la arquitectura de éste se simplifica considerablemente. Para el diseño del transmisor se necesitarán dos memorias en modo ping-pong de $M=512$ posiciones y una lógica para la transmisión de los $S=M+\alpha+\beta$ datos. En la Figura 77 se puede observar el diagrama de bloques de la arquitectura del transmisor realizada.

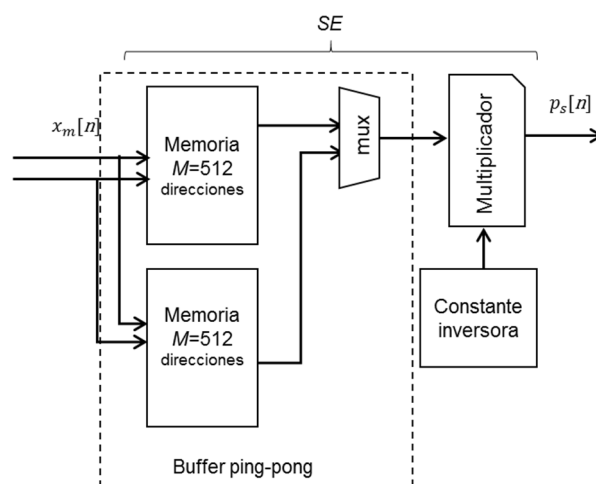


Figura 77. Diagrama de bloques de la arquitectura del transmisor para portadora única.

Como se observa en la Figura 77, los datos llegan a las memorias ping-pong a través de la señal $x_m[n]$. El empleo de una memoria dividida en dos bancos o partes y operando en modo ping-pong se usa para evitar la pérdida de datos, por lo que a una se puede acceder en modo lectura para enviar los datos con la extensión simétrica ya realizada, mientras que los nuevos datos entrantes $x_m[n]$ se escriben en la otra. Una vez se ha añadido la extensión simétrica, se multiplican los últimos datos para provocar su inversión, ya que, en la extensión simétrica empleada, los datos posteriores se invierten respecto a los datos transmitidos. Con todo esto, se obtiene la señal $p_s[n]$ lista para transmitirse.

Con lo explicado anteriormente, se observa que el throughput del transmisor para la técnica mono-portadora es un dato por cada ciclo de reloj. Además, el número de multiplicadores C_m^{Tx} empleados para la realización de esta arquitectura es (144):

$$C_m^{Tx} = C_m^{SE,Tx} = 1 \text{ multiplicador} \quad (144)$$

Donde $C_m^{SE,Tx}=1$ es el número de multiplicadores empleados en el módulo de la extensión simétrica para el transmisor (140).

Por otro lado, la latencia L^{Tx} de la arquitectura propuesta en el transmisor de portadora única es obtenida en (145):

$$L^{Tx} = L^{SE,Tx} = 514 \text{ ciclos} \quad (145)$$

Donde $L^{SE,Tx}=514$ ciclos es la latencia del módulo de la extensión simétrica para el transmisor(141).

Una vez se ha mostrado la arquitectura realizada, se procede a mostrar un cronograma del funcionamiento del transmisor en la Figura 78. En este cronograma se observan las principales señales de la propuesta y gráficamente los retardos internos para producir la salida.

Primero, como se observa en la muestra de reloj 1 se recibe la señal CE_M . La señal CE_M hace que se active un ciclo de reloj después el enable sincronizada con la señal $x_m[n]$ y se comienzan a generar direcciones para introducir los datos en el puerto A de la memoria ping-pong (muestra de reloj 2).

Después se generan las direcciones de memoria para la salida del puerto B de las memorias ping-pong, con el orden correcto para generar la extensión simétrica (muestra de reloj 4). Los coeficientes para la realización de la inversión de la extensión simétrica posterior están sincronizados con la entrada de datos en el multiplicador (muestra de reloj 6). De esta manera se obtiene después la salida $p_s[n]$ (muestra de reloj 8).

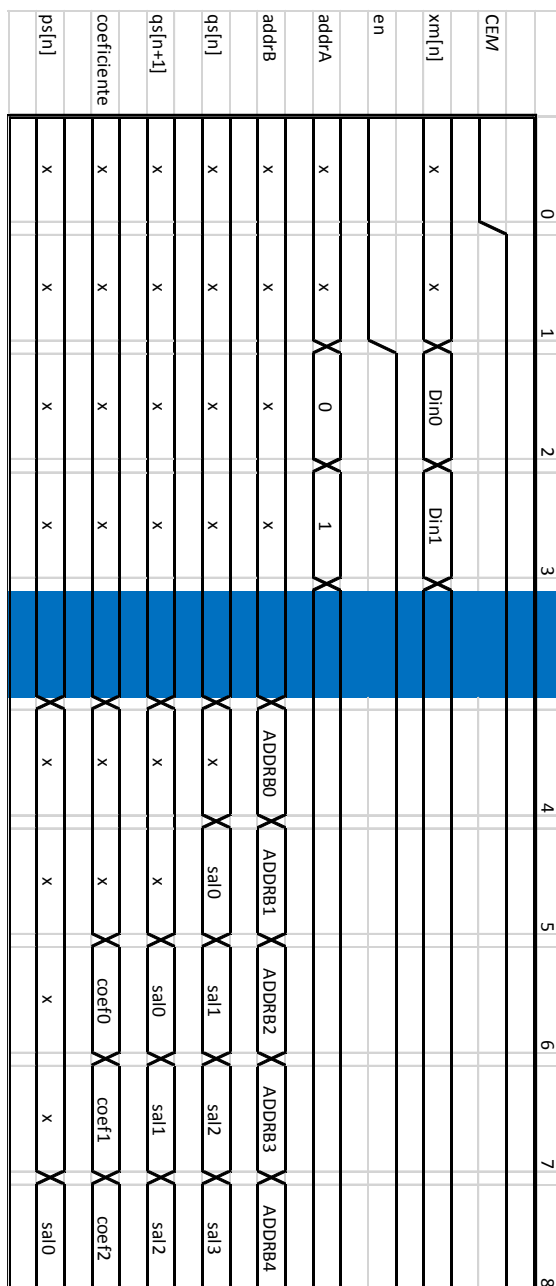


Figura 78. Cronograma del transmisor de portadora única.

6.1.2. Arquitectura del receptor

La arquitectura del receptor de la técnica de portadora única con igualación en el dominio de la frecuencia se encuentra determinada por el hecho de emplear dos DCT-IV, con lo que su diseño tendrá una complejidad añadida. Junto con las dos DCT-IV se implementará el módulo que quita la extensión simétrica que se añade en el transmisor. En este caso, puesto que el igualador en el dominio de la frecuencia no se implementa, la señal de salida $S'[k]$ de la DCT es igual a la señal de entrada $P'[k]$ de la iDCT. De este modo, en la Figura 79 y la Figura 80 se puede observar el diagrama de bloques de la arquitectura realizada en el receptor de la técnica mono-portadora. Aunque tanto para la DCT como para

la iDCT la arquitectura es la misma, la cuantificación de la representación en coma fija en cada punto es distinta, y por tanto se hará una diferenciación en las señales internas.

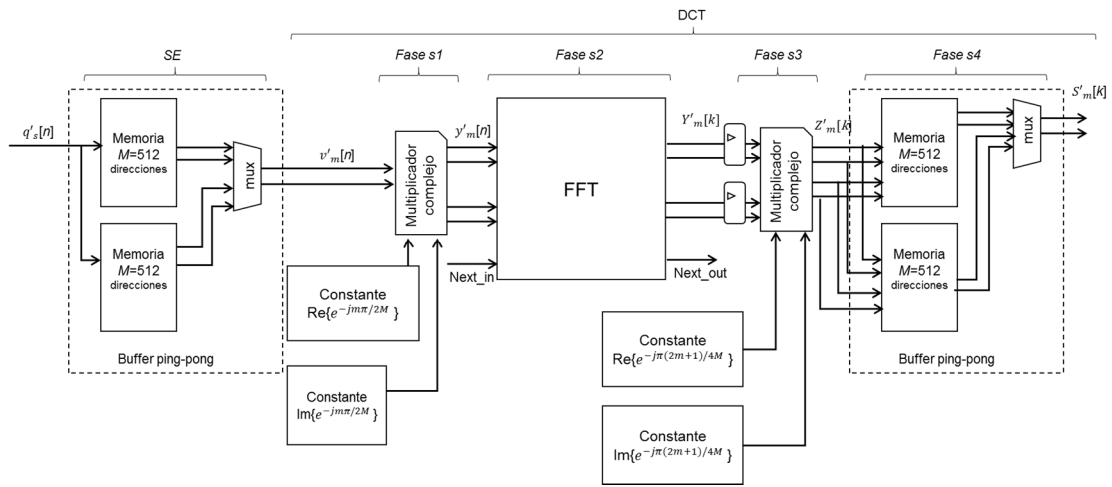


Figura 79. Diagrama de bloques de la arquitectura del receptor de la técnica de portadora única (SE y DCT).

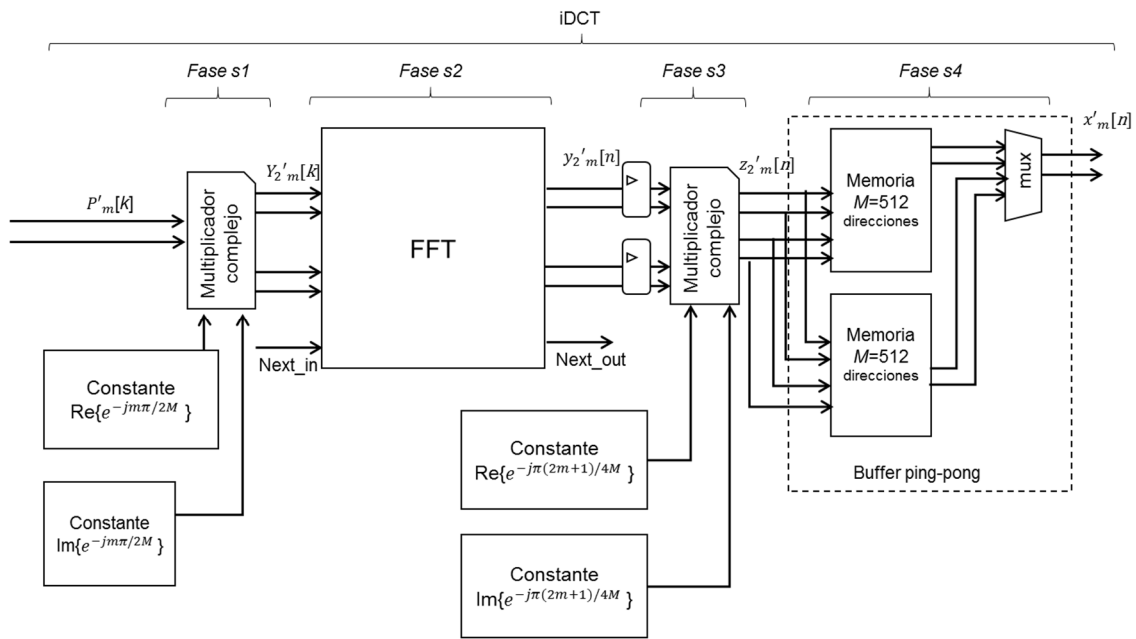


Figura 80. Diagrama de bloques de la arquitectura del receptor de la técnica de portadora única (iDCT).

Como se observa en las figuras, la señal paralelizada $q'_s[n]$ de entrada al receptor pasa a través del módulo que quita la extensión simétrica, para convertirse en $v'_m[n]$. Esta señal se multiplica por la primera constante $e^{-\frac{jm\pi}{2M}}$ para obtener $y'_m[n]$. La señal $y'_m[n]$ pasa a la FFT, de la cual se obtiene $Y'_m[k]$. Ésta se multiplica por la segunda constante $e^{-\frac{j\pi(2m+1)}{4M}}$ y se obtiene $Z'_m[k]$. La señal $Z'_m[k]$ se introduce en las memorias ping-pong, de las cuales sale la señal $S'_m[k]$, que se corresponde con la salida de la DCT.

En este punto es donde se introduce el módulo del igualador en el dominio de la frecuencia, ecualizando la señal $S'_m[k]$ de salida de la DCT, en la señal $P'_m[k]$ de entrada a la iDCT.

La señal $P'_m[k]$ se introduce en la iDCT y será multiplicada por la constante $e^{-\frac{j m \pi}{2M}}$, obteniendo la señal $Y_2'_m[k]$. Esta señal se introduce en la FFT para generar la señal $y_2'_m[n]$. La señal $y_2'_m[n]$ se multiplica por la segunda constante $e^{-\frac{j \pi (2m+1)}{4M}}$ y se obtiene la señal $z_2'_m[n]$. Esta señal pasa a las memorias ping-pong para obtener a la salida la señal $x'_m[n]$, que se corresponde con la salida de la iDCT y, a su vez, con la salida del receptor de la técnica de portadora única.

Al igual que en el transmisor, el throughput de un datapath del receptor de la técnica mono-portadora es $T_{ch}=1$ dato por ciclo de reloj y, al implementarse con un ratio de paralelismo $R_p=2$, el throughput del receptor es $T_c=T_{ch} \cdot R_p=2$ datos por ciclos de reloj. Con todo esto, todas las muestras de entrada $x_m[n]$ son procesadas cada $M_c=256$ ciclos. Como un nuevo conjunto de muestras de entrada $x_m[n]$ está disponible cada $M_c^{max}=512$ ciclos, la arquitectura propuesta cumple esta limitación ($M_c < M_c^{max}$).

Como la DCT y la iDCT se componen de la misma forma, el número de recursos se multiplica por dos, por lo que el número de multiplicadores C_m^{Rx} empleados para la realización de la arquitectura del receptor de la técnica mono-portadora es (146):

$$C_m^{Rx} = C_m^{SE,Rx} + 2 \cdot C_m^{DCT} = 0 + 2 \cdot 48 = 96 \text{ multiplicadores} \quad (146)$$

Donde $C_m^{DCT}=48$ es el número de multiplicadores empleado por una DCT (138); y $C_m^{SE,Rx}=0$ es el número de multiplicadores empleados en el módulo de la extensión simétrica (142).

Por otro lado, la latencia L^{Rx} de la arquitectura propuesta para el receptor de la técnica mono-portadora es obtenida en (147):

$$L^{Rx} = L^{SE,Rx} + 2 \cdot L^{DCT} = 512 + 2 \cdot 1256 = 3024 \text{ ciclos} \quad (147)$$

Donde $L^{SE,Rx}=512$ ciclos es la latencia del módulo SE (143); y $L^{DCT}=1256$ ciclos es la latencia de la DCT (139).

A continuación, se muestra en la Figura 81 y en la Figura 82 el cronograma de funcionamiento del receptor de la técnica de portadora única. En éste se observan las principales señales del diseño así como los retardos que existen entre ellas. En la primera parte del cronograma, se recibe la señal de entrada al receptor $q'_s[n]$, sincronizada con la señal *enable* que se produce un ciclo de reloj después de CE_M (muestra de reloj 2). A partir de esta señal se quita la extensión simétrica, para obtener $v'_m[n]$ (muestra de reloj 3). Con la activación del enable, se comienza a generar direcciones para la memoria que almacena los coeficientes $e^{-\frac{j m \pi}{2M}}$ de la primera multiplicación (12) (muestra de reloj 2). A continuación se añade un retardo en $v'_m[n]$, para que esté sincronizado con el coeficiente $e^{-\frac{j m \pi}{2M}}$ en la entrada del multiplicador (muestra de reloj 4), produciéndose 4 ciclos de reloj después la salida del multiplicador $y'_m[n]$ (muestra de reloj 8). Esta señal $y'_m[n]$ se retrasa para que esté sincronizada con la señal de comienzo de la FFT *next_in* (muestra de reloj 10).

Cuando la FFT ha sido realizada (13) (muestra de reloj 12), se activa la señal *next_out* y en el siguiente ciclo de reloj, comienza a proporcionarse la salida de la FFT $Y'_m[k]$ (muestra de reloj 13). Esta señal se sincroniza con los coeficientes $e^{-\frac{j\pi(2m+1)}{4M}}$ de la segunda multiplicación (15) (muestra de reloj 15). La salida $Z'_m[k]$ se sincroniza con las direcciones para el puerto A de la memoria ping-pong que realiza el reordenamiento de la trama de datos (14) (muestra de reloj 20). La salida $P'_m[k]$ se produce un ciclo después de la dirección del puerto B de la memoria ping-pong (muestra de reloj 22).

En la segunda parte del cronograma se describe la iDCT. Ésta se caracteriza por tener la misma arquitectura que la DCT anterior. Primero, se activa el enable y se producen las direcciones para la memoria que almacena los coeficientes $e^{-\frac{j\pi\pi}{2M}}$ de la primera multiplicación (12) (muestra de reloj 21). La señal $P'_m[k]$ llega un un ciclo de reloj después, para que esté sincronizado con el coeficiente $e^{-\frac{j\pi\pi}{2M}}$ en la entrada del multiplicador (muestra de reloj 23). Cuatro ciclos de reloj después se genera la salida del multiplicador $Y_2'_m[k]$ (muestra de reloj 27). Se añade un retardo a la señal $Y_2'_m[k]$ para que se sincronice con la señal de comienzo de la FFT *next_in* (muestra de reloj 29).

Cuando la FFT ha terminado (13), se activa la señal *next_out* (muestra de reloj 31) y se proporciona la señal de salida $y_2'_m[n]$ de la FFT un ciclo de reloj después (muestra de reloj 32). Esta señal se entrega junto a los coeficientes $e^{-\frac{j\pi(2m+1)}{4M}}$ en la segunda multiplicación (15) (muestra de reloj 34). La multiplicación produce la señal $z_2'_m[n]$ que se entrega en el puerto A de la memoria ping-pong que realiza el reordenamiento de la trama de datos (14) (muestra de reloj 39). La salida $x'_m[n]$ se produce en el puerto B de la memoria ping-pong, señal que se corresponde con la salida de la técnica de acceso al medio (muestra de reloj 41).

6.1 Arquitectura del sistema

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
CEM																							
q3[n]	x		Div0	Div1	Div2	Div3	Div4	Div5	Div6	Div7	Div8	Div9											
Ym[n]	x		Div0	Div1	Div2	Div3	Div4	Div5	Div6	Div7	Div8												
en																							
Add1	x	0	1	2	3	4	5	6	7	8	9												
coef1entes	x		Coef0	Coef1	Coef2	Coef3	Coef4	Coef5	Coef6	Coef7	Coef8												
Multi	x				coef0	coef1	coef2	coef3	coef4	coef5	coef6	coef7											
Ym[n+1]	x				div0	div1	div2	div3	div4	div5	div6	div7											
Ym[n]	x																						
Ym[n+1]	x																						
Ym[n+2]	x																						
next_in																							
next_out																							
Ym[k]	x																						
Ym[k+1]	x																						
Ym[k+2]	x																						
coef1entes	x																						
mul2	x																						
Zm[k]	x																						
Zm[k+1]	x																						
addA	x																						
addB	x																						
Prm[k]	x																						
				</																			

	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
$P^m[k]$	x	x																						
en																								
Addr1	x	x	0	1	2	3	4	5	6	7	8	9												
coeficientes	x	x		Coef0	Coef1	Coef2	Coef3	Coef4	Coef5	Coef6	Coef7	Coef8												
Mult	x	x		coef0	coef1	coef2	coef3	coef4	coef5	coef6	coef7	coef8												
$P^m[k+1]$	x	x		den0	den1	den2	den3	den4	den5	den6	den7	den8												
$P^m[k]$	x	x							mult0	mult1	mult2	mult3												
$P^m[k+1]$	x	x								mult0	mult1	mult2												
$P^m[k+2]$	x	x									mult0	mult1												
near_in																								
near_out																								
$Y^2[m][n]$	x	x																						
$Y^2[m][n-1]$	x	x																						
$Y^2[m][n-2]$	x	x																						
coeficientes	x	x																						
mult2	x	x																						
$Z^2[m][n]$	x	x																						
$Z^2[m][n-1]$	x	x																						
addrA	x	x																						
addrB	x	x																						
$X^m[n]$	x	x																						

Figura 82. Cronograma del receptor de la técnica de portadora única (iDCT).

6.2. Sistema Completo de la técnica de portadora Única

6.2.1. Estudio de la precisión finita, consumo de recursos y latencia

Como ya se realizó en las otras dos técnicas, se va a mostrar en la Tabla 36 la representación en coma fija de cada módulo y señal en el diseño propuesto, para el transmisor y el receptor. El ancho de palabra más común es de 18 bits, debido a las celdas multiplicadoras DSP48E1 empleadas.

Tabla 36. Representación en coma fija del ancho de palabra para la arquitectura de portadora única.

Módulo	Parámetro	No. bits global	No. bits fraccional
Entrada Tx	$x_m[n]$	18	16
Módulo SE	Entrada $x_m[n]$	18	16
	Salida $p_s[n]$	18	16
Salida Tx	Salida $e[n]$	18	16
Entrada Rx	Entrada $r[n]$	18	16
Módulo SE	Entrada $q'_s[n]$	18	16
	Entrada $v'_m[n]$	18	16
Rx DCT	Entrada $v'_m[n]$	18	16
	Constante $e^{-jm\pi/2M}$	25	23
	Multiplicador eq. (59)	36	32
	$y'_m[n]$	18	16
	$y'_m[n_0]$	18	8
	$Y'_m[k_0]$	18	8
	$Z'_m[k]$	18	8
	Constante $e^{-j\pi(2m+1)/4M}$	25	23
	Multiplicador eq. (62)	36	32
	Salida $P'_m[k]$	18	12
Rx iDCT	Entrada $P'_m[k]$	18	12
	Constante $e^{-jm\pi/2M}$	25	23
	Multiplicador eq. (59)	36	32
	$Y_{2'_m}[k]$	18	12
	$Y_{2'_m}[k_0]$	18	12
	$y_{2'_m}[n_0]$	18	12
	$z_{2'_m}[n]$	18	12
	Constante $e^{-j\pi(2m+1)/4M}$	25	23
Multiplicador eq. (62)	36	32	
Salida Rx	$x'_m[k]$	18	16

Una vez que se tienen los anchos de palabra de cada módulo, se realiza una comparación de ellos con el modelo en coma flotante, para comprobar tanto su error medio, como su desviación máxima. Para la realización de esta simulación se ha empleado una entrada aleatoria $x_m[n]$, de 1000 muestras de longitud y normalizada al máximo rango de entrada $[-1, +1]$, para cada uno de los $M=512$ canales de entrada. En la Tabla 37 se puede observar el error medio y la desviación estándar en cada una de las señales intermedias.

El consumo de recursos ha sido calculado anteriormente, obteniéndose para el transmisor el resultado de 1 multiplicador (144). En el caso del receptor el resultado obtenido es de 96 multiplicadores (146). La latencia de la técnica mono-portadora también ha sido calculada previamente con lo que aquí solo se expondrá el resultado final. En el

caso del transmisor la latencia L^{Tx} es la calculada en (145), que se corresponde con 514 ciclos. Para el caso del receptor, la latencia L^{Rx} es de 3024 ciclos (147).

Tabla 37. Error medio y desviación típica para las señales intermedias en la arquitectura de portadora única, asumiendo una rango de entrada de $[-1, +1]$ en $x_m[n]$.

Módulo	Parámetro	Error absoluto medio	Desviación típica
Módulo SE	$p_s[n]$	0	0
Salida Tx	$e[n]$	0	0
Entrada Rx	$r[n]$	0	0
Módulo SE	$v'_m[n]$	0	0
Rx DCT	$S'_m[k]$	0.0020	0.0038
Rx iDCT	$x'_m[n]$	0.0032	0.0035

6.2.2. Resultados obtenidos para el modelo en coma fija y el modelo en coma flotante

A continuación se va a realizar una comparación entre el modelo en coma flotante y el modelo en coma fija, obteniendo los parámetros de SNR (Signal Noise Ratio), MSE (Mean Square Error), PSNR (Peak Signal Noise Ratio), y ME (Maximum Error), tal y como se define en [112]. En la Tabla 38 se muestran los resultados obtenidos, empleando una simulación con una entrada aleatoria $x_m[n]$ de 1000 muestras de longitud y una amplitud normalizada al máximo valor de entrada.

Tabla 38. Comparación entre los modelos en coma fija y coma flotante para la técnica mono-portadora.

	Coma flotante	Coma fija
SNR	310.5491 dB	45.1182 dB
Error cuadrático medio MSE	$3.0154 \cdot 10^{-32}$	$1.8700 \cdot 10^{-5}$
Pico SNR	315.2686 dB	49.8376 dB
Error absoluto máximo ME	$4.6847 \cdot 10^{-16}$	$7.6000 \cdot 10^{-3}$

6.3. Conclusiones

La técnica de portadora única con igualación en el dominio de la frecuencia ha sido implementada realizando un estudio tanto del consumo de recursos y la latencia del sistema, como del efecto de la precisión finita.

Una de las características a destacar nuevamente es la flexibilidad del diseño, ya que modificando distintos parámetros se puede emplear la técnica mono-portadora en distintos ámbitos o medios de transmisión. Los parámetros que se pueden modificar son el número de subcanales empleados, el número de elementos que se duplicarán por la extensión simétrica, y el cambio de la frecuencia de funcionamiento del diseño.

A su vez, el empleo de un ratio de paralelismo R_p permite modificar la arquitectura interna del diseño, lo cual hace que se varíe el número de recursos que se emplean en el diseño. Además, la modificación de la arquitectura hace que se altere el tiempo de procesamiento, la latencia y throughput del diseño. Todo esto hace que la arquitectura del

6.3 Conclusiones

sistema sea capaz de trabajar en tiempo real, consiguiendo un tiempo de procesamiento bajo. Esto demuestra la versatilidad del diseño para adaptarse a distintos tipos de transmisiones.

Otra característica que ofrece la técnica mono-portadora es la simplificación obtenida en la arquitectura del transmisor, debido a que se desplaza al receptor la iDCT. Este hecho supone que se pueda emplear una plataforma de menores prestaciones y por tanto más barata para implementar el transmisor, que la que se use en el receptor.

Por último, se ha de destacar la optimización de la cuantificación empleada para reducir el efecto de la precisión en coma fija. Gracias al estudio realizado previamente sobre la arquitectura de la DCT-IV, y la elección de la cuantificación óptima para cada señal intermedia, se ha conseguido obtener unas prestaciones similares a las obtenidas en las técnicas multi-portadora estudiadas.

Capítulo 7

Resultados

En este punto el diseño de las arquitecturas de las distintas técnicas de acceso al medio se ha terminado. Para su realización se ha llevado a cabo un estudio del algoritmo, un análisis de la arquitectura a desarrollar y una optimización de los recursos empleados. Además de la implementación eficiente de las arquitecturas como periféricos específicos en el SoC, se ha desarrollado la propia arquitectura del SoC. A partir de aquí se procede a evaluar cada una de las arquitecturas y analizar las prestaciones obtenidas de estas arquitecturas con las pruebas reales. Para ello, se utiliza un convertor digital-analógico DAC FMC204 [117] y un convertor analógico-digital ADC AD9467 [118] para efectuar la correcta transmisión de los datos que envía el transmisor fuera de la FPGA, y también proceder con la correcta recepción de estos datos en el receptor. Ambos convertidores trabajarán con una frecuencia de muestreo $f_{Tx}=62.5\text{MHz}$ y ambos tienen un ancho de palabra de 16 bits. Debido a la imposibilidad de encontrar unos AFEs (Analog Front-Ends) comerciales para la conexión a la red eléctrica, se conectarán los convertidores directamente al canal de transmisión, pero siempre sin red eléctrica; de este modo, se verificará todo el diseño a $T_{Tx}=62.5\text{Msps}$ en un entorno simplificado. Se ha elegido esta tasa de transmisión de $T_{Tx}=62.5\text{Msps}$ en el canal por ser una de las propuestas en el estándar de PLC [27].

Para comprobar el diseño de las arquitecturas implementadas se van a realizar tres tipos de pruebas experimentales: en la primera se va a conectar el transmisor con el receptor dentro de la FPGA, con lo que se van a obtener los resultados de las técnicas sin emplear la conversión analógica. De este modo, la conexión del transmisor con el receptor se realiza en modo digital y esto hace que no se genere ningún tipo de ruido en el canal de transmisión.

Con esto se consigue comprobar la idoneidad de las arquitecturas, ya que el canal de transmisión es ideal. En la segunda prueba experimental, para la conexión del transmisor con el receptor se utiliza el DAC y el ADC que se encuentran conectados mediante un cable SMA de 30cm. El cable SMA presenta una buena respuesta en frecuencia y por tanto se aproxima mucho a un canal ideal. Esto permite comprobar el diseño completo introduciendo los conversores analógicos en un canal de transmisión ideal. Por último, en la tercera prueba experimental, el DAC y el ADC se conectan a un cable de red eléctrica de 12m de longitud. En esta prueba se podrá evaluar el diseño en un canal de transmisión con unas condiciones más parecidas a las reales.

Para la obtención de los resultados en las pruebas definidas se han utilizado tres aproximaciones diferentes para las técnicas desarrolladas. Primero se ha realizado la simulación del modelo en coma flotante, después otra simulación con el modelo en coma fija, y por último, una prueba experimental con la técnica implementada en la FPGA. En estas simulaciones se pretende evaluar la arquitectura desarrollada en las condiciones más parecidas a la prueba experimental, por ello tanto la entrega y recepción de datos, como los conversores siempre están desarrollados con modelos en precisión finita idéntica a la que poseen en la implementación de la FPGA. Con el uso de la simulación en coma flotante se obtienen los resultados que obtendría la arquitectura, sin el efecto de la precisión finita. La simulación en coma fija ha sido realizada considerando las limitaciones de ancho de palabra que se fijan en la FPGA, con lo que los resultados obtenidos deben ser iguales a los obtenidos con el diseño en la FPGA. Por último, se realiza la prueba empírica, con el objeto de constatar el diseño desarrollado en la FPGA con los diseños simulados.

En los tres casos se empleará la misma señal de entrada, definida por un tamaño de 40960 muestras, y generada de manera aleatoria en el rango de amplitud de $[-1,1]$. De esta manera, será posible establecer una comparativa directa entre los resultados obtenidos en la prueba empírica y los obtenidos en las simulaciones.

Para la evaluación de los resultados obtenidos, se ha decidido calcular los parámetros de SNR (Signal Noise Ratio), MSE (Mean Square Error), PSNR (Peak Signal Noise Ratio) y ME (Maximum Error). El SNR (relación señal-ruido) (148) calcula la relación media que existe entre la potencia de la señal que se transmite y la potencia del error, considerando al error la diferencia de la señal que se transmite con la que se recibe. Este parámetro permite determinar la media de la calidad de la señal recibida, respecto a la que se ha transmitido.

$$SNR = 10 \log \frac{(\text{sum}(S_{Tx})^2)}{(\text{sum}(S_{Tx}-S_{Rx})^2)} \quad (148)$$

Por otro lado, el MSE (error cuadrático medio) (149) calcula la media de la diferencia entre la señal que se transmite y la que se recibe, al cuadrado. El MSE determina el error introducido en la señal recibida respecto a la transmitida.

$$MSE = \frac{\text{sum}((S_{Tx}-S_{Rx})^2)}{L} \quad (149)$$

El PSNR (pico de relación señal-ruido) (150) calcula la relación entre la máxima potencia de entrada y el MSE. Con este parámetro se obtiene la máxima calidad de la señal recibida respecto a la señal transmitida.

$$PSNR = 10 \log \frac{(\max(S_{Tx}))^2}{MSE} \quad (150)$$

Por último, el ME (error máximo) (151) calcula la máxima diferencia que existe entre la señal que se transmite y la señal recibida.

$$ME = \max(|S_{Tx} - S_{Rx}|) \quad (151)$$

Estos parámetros se calcularán para cada subcanal de las arquitecturas implementadas, realizando después una media para obtener el valor que define a la arquitectura completa.

Además, será posible integrar una fase de modulación previa a la técnica de acceso al medio. En este caso, se han empleado distintas modulaciones PAM (Pulse Amplitude Modulation) [119], debido a que así lo indica el estándar de PLC de banda ancha. La modulación PAM consiste en asignar niveles de amplitud a la señal a transmitir. Además de aplicar esta modulación se ha decidido realizar también la transmisión de los datos sin ningún tipo de modulación. Con el empleo de la modulación PAM se calculará el parámetro SER (Symbol Error Rate) para cada una de las técnicas implementadas. El SER calcula la relación de símbolos erróneos recibidos respecto al número total de símbolos. Este parámetro permite determinar qué porcentaje de los datos modulados recibidos han llegado con error.

En estas pruebas experimentales no se ha considerado el uso de un igualador de canal, al estar el estudio de la igualación y sincronización del canal de transmisión fuera de los objetivos de esta tesis. En su lugar, se ha empleado en todos los casos un módulo de sincronismo ideal sin igualación de canal.

El empleo de los convertidores digital-analógico y analógico-digital implica la modificación de la arquitectura presentada en la sección 3, introduciendo un módulo de control para el DAC y otro para el ADC. Asimismo, dado que la latencia del bucle analógico es constante pero no determinista al transmitir las señales fuera de la FPGA, es necesario la introducción de otro periférico extra para la sincronización del transmisor y el receptor. De este modo, la sincronización realiza el alineamiento de la trama de datos de entrada en el receptor, situando el primer valor en el primer subcanal. En la Figura 83 se muestra el diagrama de bloques del diseño del SoC implementado con las modificaciones añadidas.

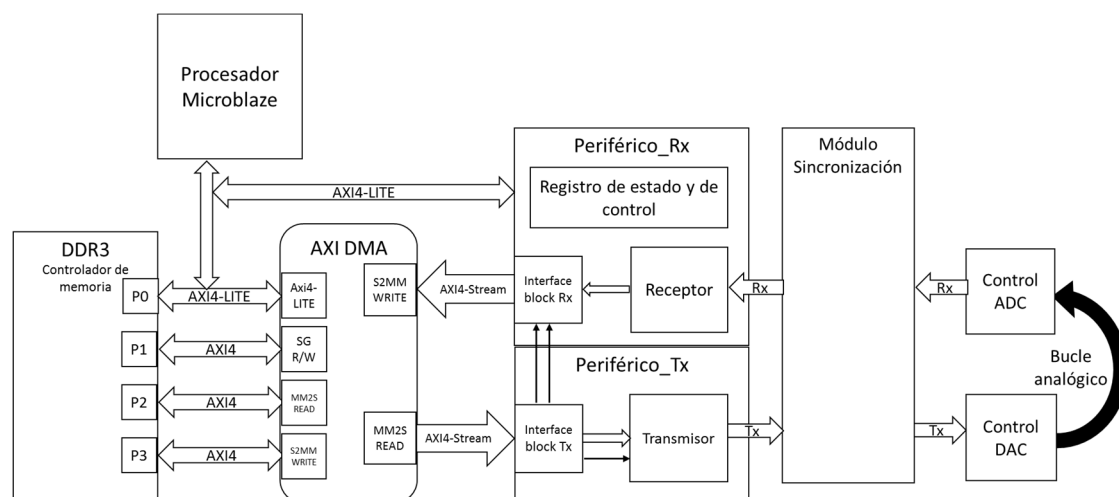


Figura 83. Diagrama de bloques del sistema completo para realizar pruebas reales.

Como se observa en el diagrama de bloques, tanto la señal transmitida, como la señal recibida, entran al módulo de sincronismo, de forma que sea posible llevar a cabo una sincronización ideal de la señal recibida.

7.1. Etapas de calibración y sincronización

Para la realización de las pruebas experimentales, es imprescindible asegurar que los datos recibidos en el receptor son correctos y que se encuentran sincronizados con su subportadora correspondiente. Para realizar esto, es necesario realizar una configuración de la frecuencia empleada, una calibración de los datos recibidos y un alineamiento de la trama de datos. La configuración de la frecuencia consiste en que el transmisor y el receptor dispongan de la misma frecuencia real. Esto es necesario para que no se produzcan pérdidas de datos originado por el desplazamiento en frecuencia de portadoras (CFO) [37], debido a que la frecuencia del receptor sea distinta que la del transmisor. Por otro lado, se debe garantizar una correcta captura en la FPGA de los datos proporcionados por el ADC. Para ello, se lleva a cabo una calibración consistente en centrar el flanco del reloj de captura en la zona estable del dato que proporciona el ADC, ya que si el flanco no se encuentra en esta zona, se puede entregar información mezclada de datos de dos ciclos de reloj distintos (el ADC transmite los datos en DDR). Por último, se realiza el alineamiento de la trama de datos, ya que como se trata de una transmisión multi-portadora es necesario que el primer dato se introduzca en el primer subcanal, pues de otro modo la decodificación de los datos no se producirá correctamente. En la Figura 84 se muestran las tres fases necesarias para asegurar una correcta recepción de los datos.

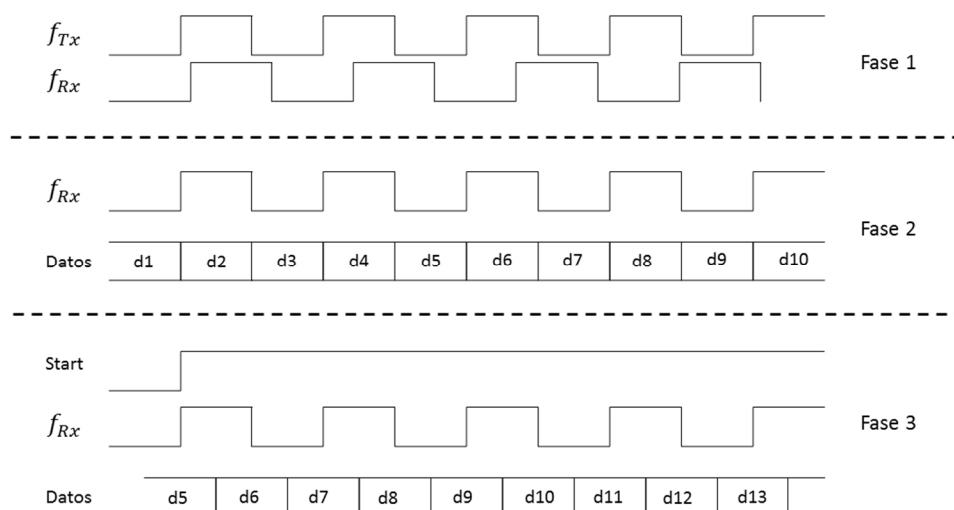


Figura 84. Fases para una correcta recepción de los datos para la realización de las pruebas reales.

Para asegurar que los datos transmitidos se reciben correctamente en la entrada del receptor, es necesario que las tres fases se cumplan. Con este objetivo se han desarrollado los módulos de control del DAC, ADC y el módulo de sincronismo.

7.1.1. Módulo de control del DAC

El módulo de control del DAC, además de realizar el control y la entrega de los datos al convertidor, proporciona el reloj que utilizan todos los bloques involucrados en el proceso (Transmisor, Receptor, ADC,...). El módulo de control obtiene el reloj que se genera en el DAC y lo distribuye por el resto de la FPGA, puesto que este reloj es el que se empleará en todo el diseño, para asegurar que todos los periféricos disponen de la misma frecuencia real. En la Figura 85 se puede observar el diagrama de bloques del módulo de control del DAC. Este módulo está formado por un enlace SPI, encargado de la comunicación con el DAC; un registro de estado y control, que permite que el microprocesador se comunique y configure el módulo y el DAC; y la parte encargada de entregar los datos a transmitir.

La transmisión de los datos al DAC se realiza en modo DDR (Double Data Rate), esto implica que se transmitan datos tanto en el flanco de subida de reloj, como en el de bajada. Para la correcta transmisión de los datos al DAC se toma el reloj que genera el propio DAC y se introduce en un MMCM (Mixed-Mode Clock Manager). Este módulo permite generar las señales de reloj necesarias. Por un lado, genera la señal de reloj global del sistema, la cual se empleará tanto en el transmisor, como en el receptor. Y por otro lado, se genera una señal de reloj desfasada 90° respecto a la anterior, que se transmitirá al DAC junto con los datos. Como los datos del transmisor se generan en DDR con la señal de reloj del sistema, y la señal de reloj que se envía al DAC está desfasada 90° , se consigue que los flancos del reloj siempre estén alineados en la mitad de la ventana de bit, asegurando la correcta entrega de éstos. Aparte, la señal To_adc_clk es la señal de reloj que se proporciona al ADC, ya que está configurado para que reciba la señal de reloj de manera externa. Por lo tanto, al entregar el reloj al ADC, se asegura que tanto transmisor y receptor, como el DAC y el ADC, dispongan de la misma frecuencia real.

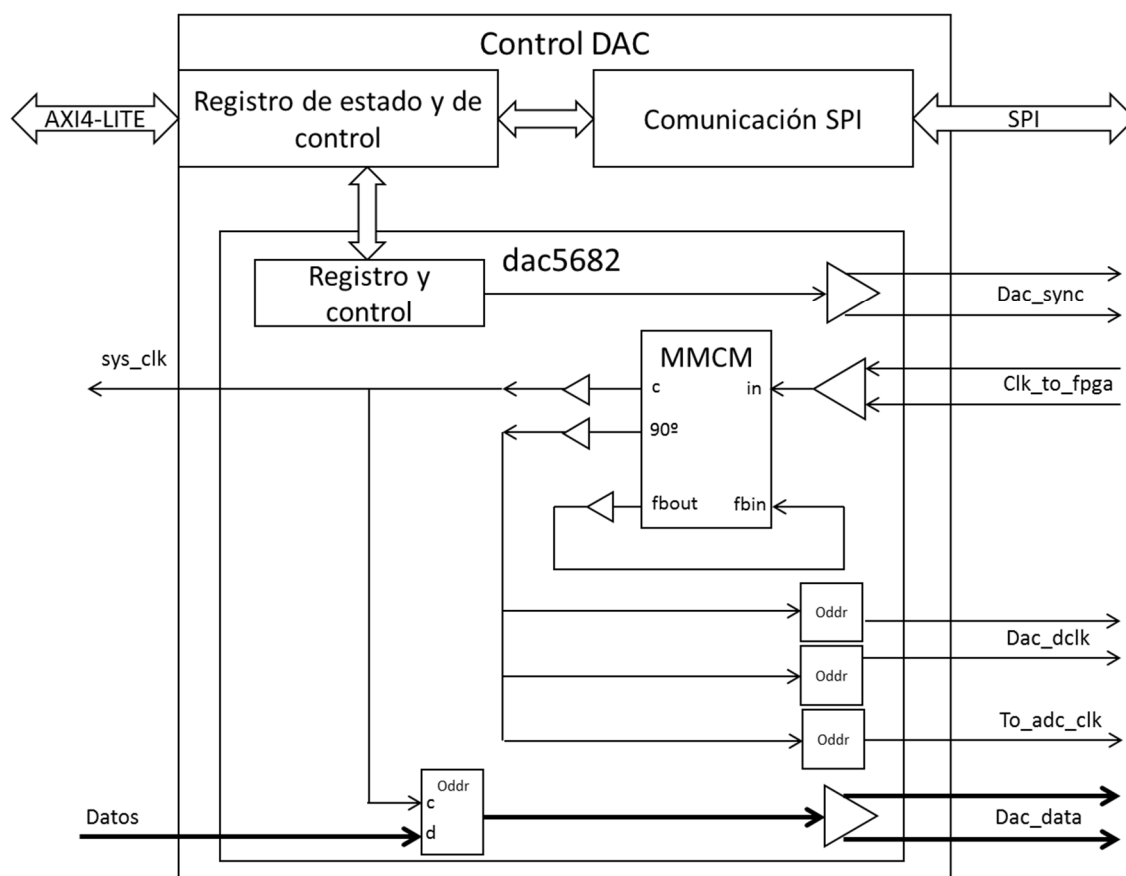


Figura 85. Diagrama de bloques del módulo de control del DAC.

7.1.2. Módulo de control del ADC

El módulo de control del ADC, además de realizar la configuración y el control del ADC, realiza la calibración de los datos del ADC. Como se va a emplear el reloj del sistema proporcionado por el DAC, es necesario ajustar el flanco del reloj de captura en la ventana de los datos que proporciona el ADC, para asegurar que el flanco del reloj se encuentra en la zona estable de los datos proporcionados.

En la Figura 86 se muestra el diagrama de bloques del control del ADC. Como se observa, para realizar esta calibración, se va a emplear la señal de reloj que proporciona el ADC como patrón de calibración, puesto que el fabricante del ADC asegura que la señal de reloj y los datos tienen un skew mínimo menor a los 100ps. Además, el ADC proporciona los datos en formato DDR; esto quiere decir que se entrega un dato con el flanco de subida y otro con el flanco de bajada. Se va a emplear un bloque *iodelay* que permite introducir retardos de hasta 5ns, y un bloque *IDDR* para capturar la señal de calibración. Combinando estos dos elementos, se puede determinar la posición de los flancos del reloj del ADC en función del flanco del reloj que proporciona el DAC a todo el sistema (*sys_clk*), controlando el retardo introducido hasta situar el flanco de reloj del sistema en el centro de los flancos de reloj del ADC. Como el reloj y los datos del ADC están en fase, si se introduce el mismo retardo en los datos que el calculado para el reloj, se dispondrá de los datos estables. Por último, el orden de entrega de los datos de los dos

flancos dependerá de si los datos están en fase con el flanco de subida o de bajada. Para ello el módulo de calibración se encarga de ajustar el bloque multiplexor dependiendo de los valores obtenidos. Toda esta operación es controlada por Microblaze a través de la interfaz AXI-LITE y el registro de estado y control.

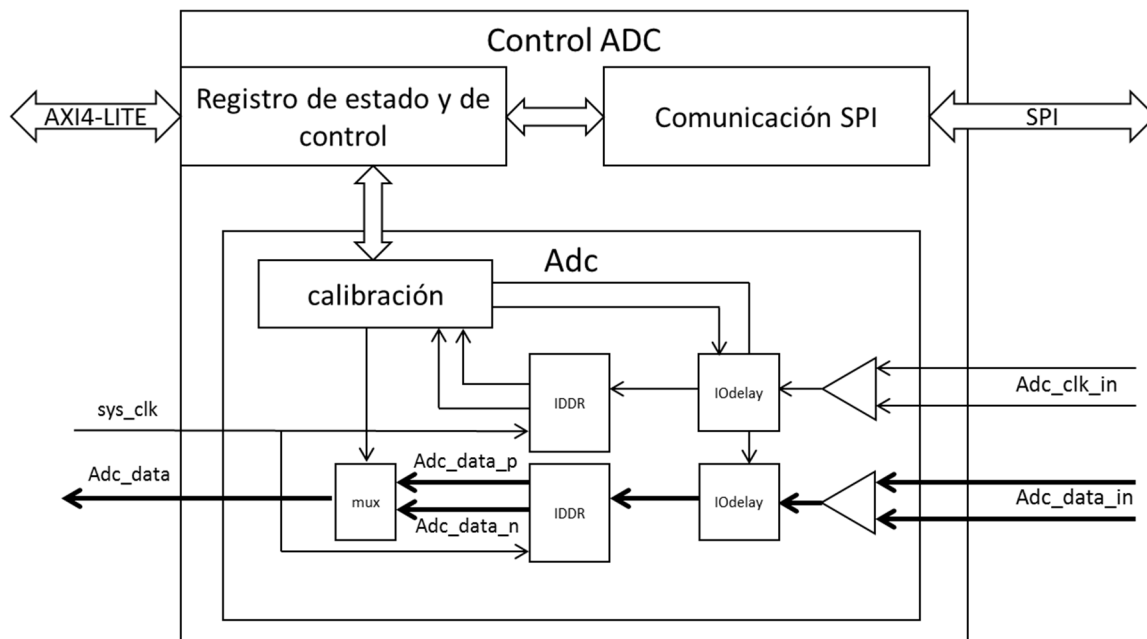


Figura 86. Diagrama de bloques del periférico encargada del control del ADC.

7.1.3. Módulo de sincronismo

El módulo de sincronismo se encarga de realizar el alineamiento a nivel de trama, que consiste en que cada dato recibido se introduzca en su subcanal correspondiente del receptor. Como se observa en el diagrama de bloques del módulo de sincronismo de la Figura 87, para realizar este alineamiento se emplea una señal de entrenamiento (*senal_sync*) que está multiplexada con la señal del transmisor. Esta señal se transmite en la fase de inicialización y cuando es recibida, un contador determinará el número de ciclos de reloj que está desfasada una señal respecto a la otra. Conociendo este valor, se introducirá el número de retardos obtenido en el bloque *retardos*, con lo que la señal que se introduce en el receptor, ya estará sincronizada. Además, el módulo de sincronismo posee una serie de registros de control accesibles a través de la interfaz AXI-LITE, que permite añadir los retardos manualmente desde Microblaze. Con estos registros también se puede monitorizar el número de retardos que se están aplicando a la señal y elegir que se transmita una señal patrón de tipo seno. Una vez el sistema está alineado a nivel de trama, se cambia el multiplexor disponible en el módulo de sincronismo para que se emita la señal del transmisor.

7.2 Recursos empleados por las arquitecturas en la FPGA

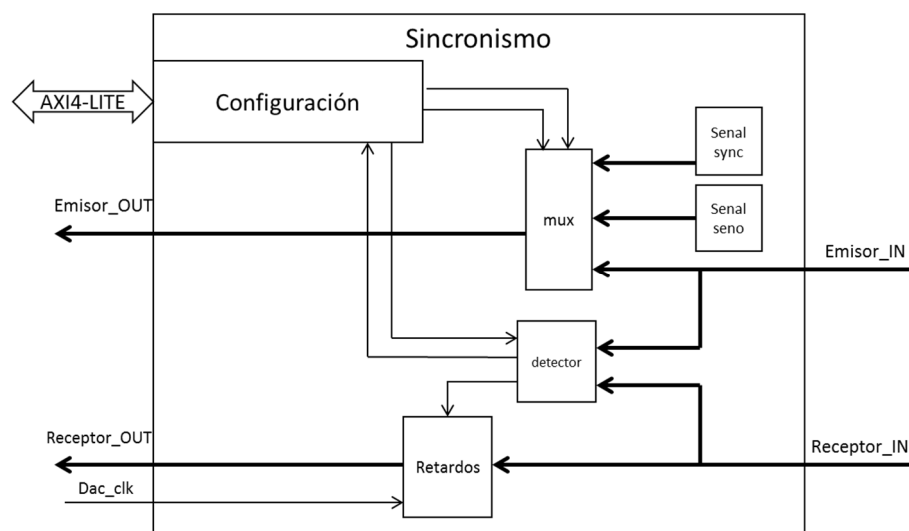


Figura 87. Diagrama de bloques del periférico de sincronismo.

Con este bloque, se termina la configuración, calibración y sincronización de los datos recibidos por el ADC, con lo que a partir de este momento se pueden obtener los resultados de las pruebas experimentales, independientemente de la longitud del cable empleado, ya que se asegurará que los datos recibidos siempre estén sincronizados.

7.2. Recursos empleados por las arquitecturas en la FPGA

Se van a mostrar los recursos consumidos por las distintas implementaciones realizadas. Para el cálculo de los recursos consumidos se ha considerado que el diseño de las arquitecturas ha sido realizado en la FPGA Virtex 6 xc6vlx240t [110], la cual dispone de 301440 flip-flops, 150720 LUTs, 416 BRAMs de 36 bits de ancho de palabra, 768 celdas DSP48E1 y 600 pines de entrada salida.

Primero, se van a mostrar los recursos empleados por el sistema global. El sistema global está formado por el microprocesador Microblaze, el controlador de memoria externa, el módulo DMA, el módulo de sincronismo y el control de los convertidores. En la Tabla 39 se muestra el valor obtenido para los flip-flops, LUTs, BRAMs y DSP48E1. El consumo de recursos global incluye además el consumo de cada módulo que forma parte del sistema global. Junto a éste, se muestra el porcentaje de utilización respecto a la arquitectura total, para mostrar cómo repercute cada módulo en la arquitectura.

Tabla 39. Consumo de recursos del sistema global en una FPGA Virtex 6 XC6VLX240T.

	<i>flip-flops</i>	<i>LUTs</i>	<i>BRAMs</i>	<i>DSP48E1</i>
Sistema global	12993	31659	107	93
Sistema global sin periférico avanzado	11314	25726	36	4
Microblaze	1445(11.12%)	3746 (11.83%)	19 (17.76%)	4 (4.30%)
AXI4	3658(28.15%)	9230(29.15%)	9 (8.41%)	0 (0.00%)
Controlador DDR3	3123(24.04%)	6506 (20.55%)	0 (0.00%)	0 (0.00%)
DMA	1170 (9.00%)	2608 (8.24%)	2 (1.87%)	0 (0.00%)
Módulo sincronismo	57 (0.44%)	131 (0.41%)	3 (2.80%)	0 (0.00%)
Control DAC	594 (4.57%)	1178 (3.72%)	3 (2.80%)	0 (0.00%)
Control ADC	144 (1.11%)	358 (1.13%)	0 (0.00%)	0 (0.00%)

Cabe destacar que el porcentaje de utilización de recursos de la FPGA es bajo, ya que el sistema completo ocupa un 4.3% de los flip-flops, un 21% de LUTs, 25.7% de BRAMs y un 12.1% de DSP48E1.

Una vez se ha mostrado el consumo de recursos del sistema global, se procede a mostrar el consumo de recursos de cada una de las técnicas empleadas. En primer lugar, se mostrará el consumo de recursos para la técnica DTT en la Tabla 40. Nótese que el porcentaje indicado es respecto al total de recursos consumidos.

Tabla 40. Consumo de recursos de la técnica DTT en una FPGA Virtex 6 XC6VLX240T.

	<i>flip-flops</i>	<i>LUTs</i>	<i>BRAMs</i>	<i>DSP48E1</i>
Sistema global	12993	31659	107	93
Interfaz Tx	90 (0.69%)	181 (0.57%)	2 (1.87%)	0 (0.00%)
Transmisor	1245 (9.58%)	3567 (11.27%)	33 (30.84%)	45 (48.39%)
Interfaz Rx	172 (1.32%)	393 (1.24%)	0 (0.00%)	0 (0.00%)
Receptor	1060 (8.16%)	3197 (10.10%)	36 (33.64%)	44 (47.31%)

Como se observa, la mayor parte de las celdas DSP48E1 empleadas se utilizan en la técnica implementada, debido al procesamiento realizado; mientras que el sistema global no necesita realizar ningún tipo de procesamiento, ya que sólo realiza la tarea de proporcionar los datos.

En la Tabla 41 se muestra el consumo de recursos empleado por la técnica FBMC. Dado que emplea un banco de filtros polifásicos, además de un módulo Matriz y una multiplicación por una constante theta (Θ), su consumo de recursos es más elevado que la anterior técnica.

Tabla 41. Consumo de recursos de la técnica FBMC en una FPGA Virtex 6 XC6VLX240T.

	<i>flip-flops</i>	<i>LUTs</i>	<i>BRAMs</i>	<i>DSP48E1</i>
Sistema global	12993	31659	141	104
Interfaz Tx	87 (0.67%)	185 (0.58%)	2 (1.42%)	0 (0.00%)
Transmisor	1397 (10.75%)	3724 (11.76%)	45 (31.91%)	49 (52.68%)
Interfaz Rx	200 (1.54%)	403 (1.27%)	0 (0.00%)	0 (0.00%)
Receptor	1096 (8.44%)	3232 (10.21%)	58 (41.13%)	51 (54.84%)

Por último, se va a mostrar en la Tabla 42 los resultados obtenidos para la técnica mono-portadora. En este caso, dado que en el transmisor no se realiza ningún tipo de procesamiento, salvo añadir la extensión simétrica (SE), el número de recursos se verá reducido; si bien, en el receptor se verá aumentado este consumo debido al empleo de dos DCT-IV.

Tabla 42. Consumo de recursos de la técnica de portadora única en una FPGA Virtex 6 XC6VLX240T.

	<i>flip-flops</i>	<i>LUTs</i>	<i>BRAMs</i>	<i>DSP48E1</i>
Sistema global	12993	31659	107	93
Interfaz Tx	88 (0.68%)	205 (0.65%)	2 (1.87%)	0 (0.00%)
Transmisor	42 (0.32%)	116 (0.37%)	5 (4.67%)	1 (1.08%)
Interfaz Rx	219 (1.69%)	401 (1.27%)	0 (0.00%)	0 (0.00%)
Receptor	1942 (14.95%)	5976 (18.88%)	68 (63.55%)	88 (94.62%)

7.3 Resultados obtenidos con un canal ideal

Comparando el consumo de recursos obtenido de las distintas técnicas implementadas, se puede observar que la técnica FBMC es la que más recursos consume, dado que es la que realiza un procesamiento más complejo de la señal (DCT-IV más filtrado). Por otro lado, se observa que la utilización de celdas DSP48E1 entre la técnica DTT y la monoportadora es el mismo. Esto es debido a que ambas técnicas emplean los mismos módulos (DCT-IV y SE), pero ubicados de distinta manera entre transmisor y receptor.

Como conclusión se confirma que pese al desarrollo de arquitecturas de técnicas de acceso al medio que realizan operaciones complejas, se ha conseguido emplear un número de recursos reducido que permite que en un futuro se puedan integrar nuevos periféricos al sistema. Para lograr este objetivo ha sido de vital importancia el ratio de paralelismo R_p , que como se ha explicado previamente, permite la reutilización de cada recurso a lo largo del tiempo, evitando así que se dispare el uso de éstos.

7.3. Resultados obtenidos con un canal ideal

Una vez analizado el consumo de recursos se procede a caracterizar la calidad de las técnicas empleadas. Para ello, en este primer caso se van a conectar el transmisor y el receptor internamente en la FPGA, y de esta manera conseguir un canal ideal. Por tanto, en este caso, no se emplearán los conversores analógico-digitales, aunque sí que se considerará la limitación del ancho de palabra de 16 bits. Asimismo, los datos de entrada y salida de la arquitectura de la técnica de acceso al medio también disponen de una limitación de ancho de palabra de 16 bits.

Con la utilización del canal ideal se pretende obtener una evaluación de las arquitecturas de las técnicas implementadas sin la aportación de ningún ruido o efecto adicional. De esta forma, se podrá conocer cómo se comportan las implementaciones realizadas, respecto a los modelos en coma flotante y coma fija obtenidos para la simulación.

En la Tabla 43 se muestran los resultados obtenidos para los distintos parámetros, tanto para la implementación en la FPGA de las técnicas de acceso al medio estudiadas, como de las simulaciones realizadas en Matlab con los modelos en coma flotante y coma fija.

Tabla 43. Resultados obtenidos en las pruebas experimentales para el canal ideal.

	Técnica	SNR	Error cuadrático medio	Pico de SNR	Error máximo
Prueba experimental	FBMC	45.3532 dB	$2.4296 \cdot 10^{-5}$	50.0426 dB	0.0073
	DTT	45.4423 dB	$1.8541 \cdot 10^{-5}$	50.1318 dB	0.0072
	MONO	45.4654 dB	$1.7301 \cdot 10^{-5}$	50.1544 dB	0.0071
Modelo coma fija	FBMC	43.6433 dB	$2.8862 \cdot 10^{-5}$	48.3327 dB	0.0115
	DTT	43.8381 dB	$2.6555 \cdot 10^{-5}$	48.5476 dB	0.0089
	MONO	44.4524 dB	$1.9415 \cdot 10^{-5}$	49.1419 dB	0.0083
Modelo coma flotante	FBMC	57.3448 dB	$1.5711 \cdot 10^{-6}$	62.0342 dB	0.0080
	DTT	65.8419 dB	$3.1352 \cdot 10^{-7}$	70.5314 dB	$7.9971 \cdot 10^{-4}$
	MONO	83.4227 dB	$1.5193 \cdot 10^{-9}$	88.1121 dB	$6.1035 \cdot 10^{-5}$

Como se observa, tanto los valores obtenidos en el modelo en coma fija, como los obtenidos en la prueba empírica no son iguales. Este hecho se debe a que el modelo de la FFT utilizado en el modelo de punto fijo no caracteriza el comportamiento interno exacto del core de la FFT empleado en la implementación. Esta afirmación se corroboró eliminando el modelo de la FFT y realizando una nueva comprobación parcial del sistema. En este caso sí se observó que la diferencia entre los valores obtenidos en la implementación y los de la simulación realizada en coma fija era nula.

Por otro lado, en el caso del modelo en coma flotante, los mayores valores de SNR se obtienen al no tener el error por el efecto de la cuantificación del modelo. Esta diferencia con el modelo en coma fija se genera en su gran mayoría en la FFT, debido a que a la hora de realizar la cuantificación del sistema es necesario tener en cuenta el peor caso posible, para evitar desbordamientos de señales que provocarían la degradación de los datos. En la FFT el peor caso se encuentra cuando todas las salidas se encuentran a cero, menos una que posee el valor máximo. Este valor máximo se corresponde con el número de puntos M de la FFT. Por lo tanto, la cuantificación debe ser capaz de poder soportar este valor generado. Para ello, es necesario reservar bits adicionales para la parte entera, reduciendo el número de bits para la parte decimal; como consecuencia se producirá una reducción de la resolución del sistema.

Prestando atención a la simulación en coma flotante, se puede observar que la técnica mono-portadora obtiene unos valores mayores de SNR y menores de error que las técnicas DTT y FBMC. Esta diferencia se debe a la limitación del ancho de palabra del canal de transmisión, limitación impuesta por los conversores y que aunque no se utilizan en esta primera prueba, sí se ha tenido en cuenta. La limitación del ancho de palabra del canal de transmisión en el caso de la técnica mono-portadora no genera ningún error, ya que no se produce ningún tipo de tratamiento de los datos en el transmisor. Mientras, en las técnicas DTT y FBMC se produce un efecto de atenuación a bajas frecuencias que se observará en la Figura 88, Figura 89 y Figura 90 y que se explicará más adelante.

Profundizando un poco más en los resultados que se han obtenido, se procede a mostrar en las figuras mencionadas una representación del valor medio de la relación señal-ruido (SNR) en cada uno de los subcanales definidos en la técnicas multi-portadora correspondiente ($M=512$), para las dos simulaciones y la prueba experimental.

7.3 Resultados obtenidos con un canal ideal

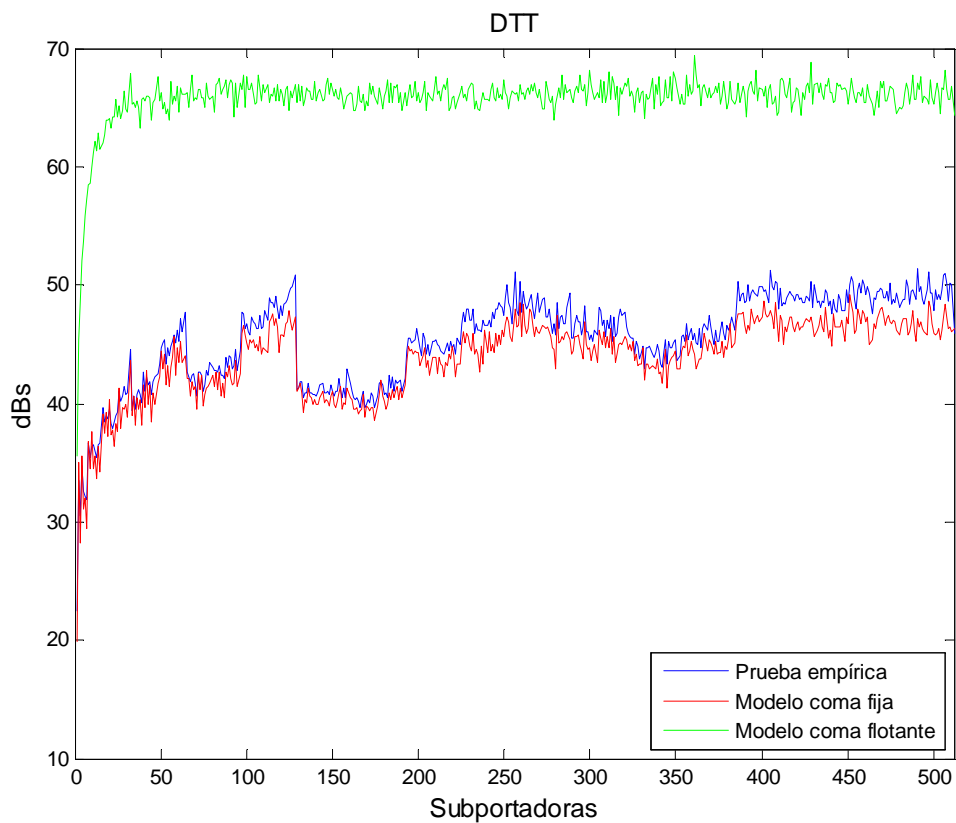


Figura 88. Representación de SNR en el canal ideal para las distintas subportadoras en la técnica DTT.

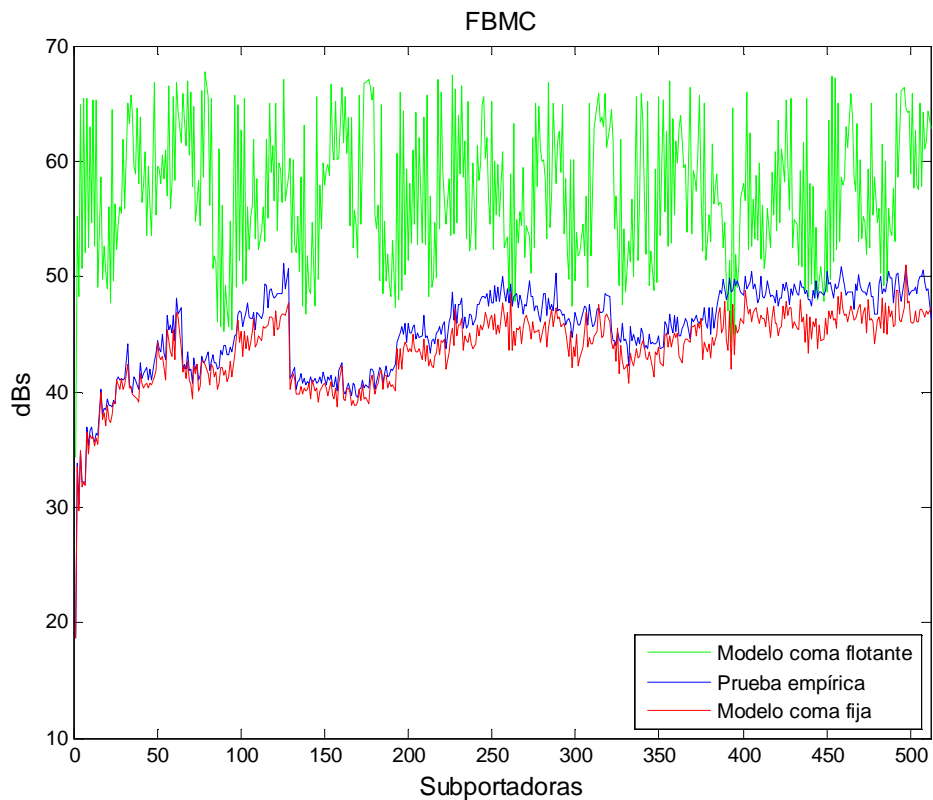


Figura 89. Representación de SNR en el canal ideal para las distintas subportadoras de la técnica FBMC.

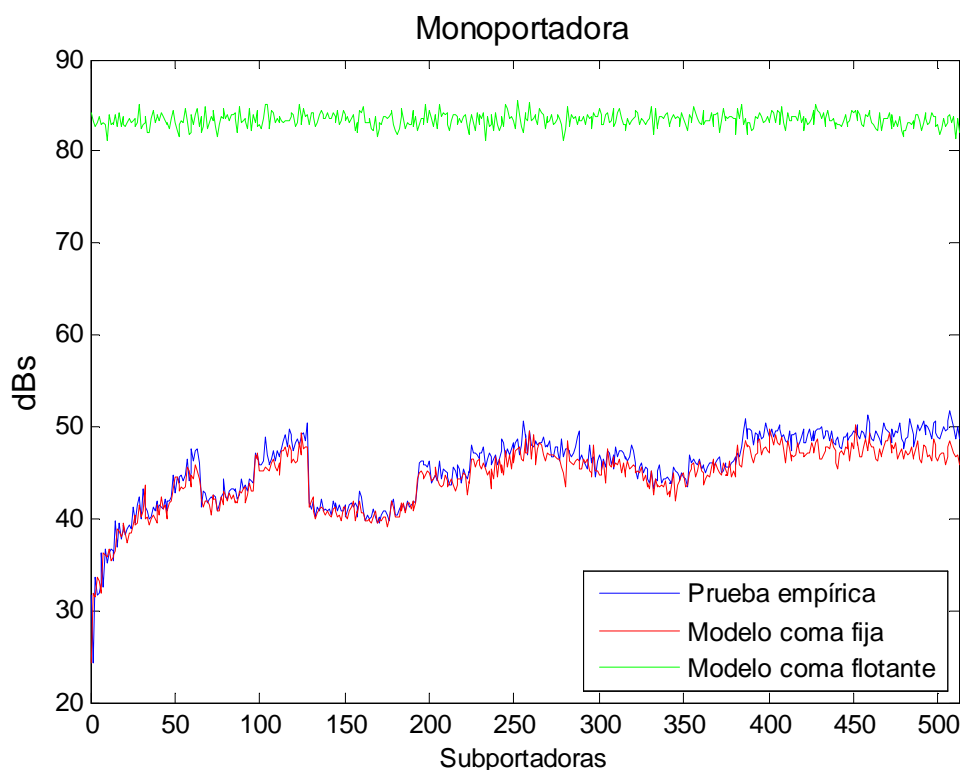


Figura 90. Representación de SNR en el canal ideal para las distintas subportadoras de la técnica de portadora única.

Como se observa en las figuras, en la simulación en coma flotante de la técnica DTT y FBMC se produce una atenuación en las primeras subportadoras. Como se ha comentado, esto es debido a la limitación de la cuantificación de los conversores combinado con el procesamiento de la FFT realizado en el transmisor. También se puede observar que en el caso de la técnica mono-portadora, al no producirse ningún tipo de procesamiento en los datos del transmisor, este hecho no se produce, observándose un mayor valor de SNR.

Por otro lado, tanto en la simulación en coma fija, como en la prueba empírica de las tres técnicas, se observa que existe una caída del valor de SNR en torno a la portadora número 120. Este efecto en los valores de los SNR de cada subportadora se debe al hecho de que existe una cuantificación que limita la precisión de las arquitecturas desarrolladas y en este caso es generado por la limitación de la precisión en la FFT del transmisor.

También se puede observar que aunque el modelo empleado para la FFT no caracteriza exactamente el comportamiento interno del core de la FFT utilizado en la implementación, los valores de SNR obtenidos para cada subportadora son muy parecidos, sirviendo para comprobar que el modelo en coma fija se ajusta a la arquitectura implementada en la FPGA.

Asimismo, se observa que pese a la reducción del valor del SNR generado por el efecto de la precisión finita en las arquitecturas, se obtiene un SNR suficiente para realizar transmisiones por un canal PLC de banda ancha.

Por último, se realizaron pruebas experimentales y simulaciones en los modelos, introduciendo las distintas modulaciones PAM (2, 4, 8, 16, 32) según el estándar de PLC,

para calcular el SER (Symbol Error Rate). Al tratarse de un canal ideal y disponer de SNR elevados, el SER obtenido en todas las pruebas era nulo, lo cual indica que no se produce ningún error de transmisión en los datos enviados.

7.4. Resultados obtenidos con un canal consistente en un cable SMA

Para la conexión del cable SMA es necesario el empleo de los dos convertidores analógico-digitales. Estos convertidores han sido evaluados, observándose que el ADC no introduce ninguna atenuación en la transmisión. Asimismo, en el caso del DAC se observa que la salida no alcanzaba la amplitud que se determinan en las hojas de características del mismo. Esto es debido a que este convertidor DAC dispone a su salida de un transformador que introduce pérdidas como se observa en la Figura 91. Esto provocaba que los datos transmitidos fuera de la FPGA sufran una pérdida de 14dB respecto a la salida del transmisor. Por otro lado, se puede observar en la gráfica que este transformador atenúa mucho las frecuencias comprendidas entre 0-3MHz, con lo que este rango de frecuencias no será considerado en las pruebas reales.

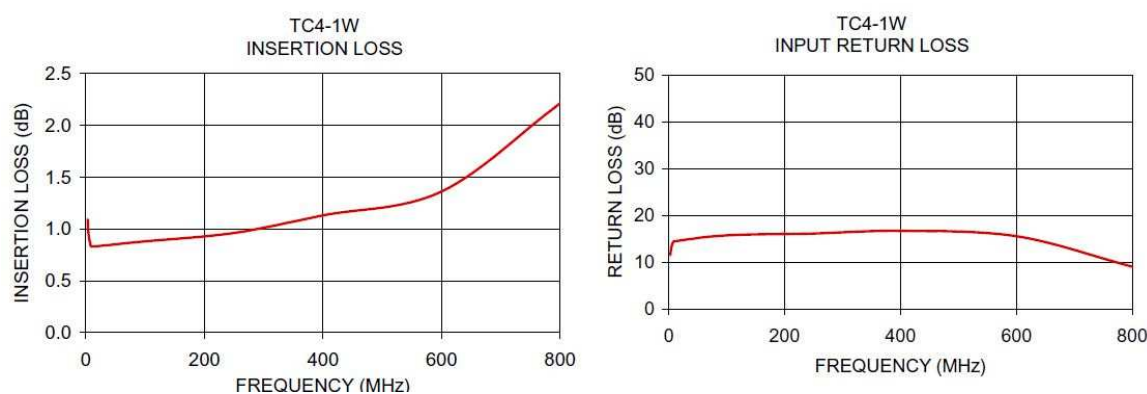


Figura 91. Pérdida a la salida del DAC por el transformador TC4-1W (imagen del datasheet de TC4-1w).

Pese a que se emplea un cable SMA que puede considerarse prácticamente ideal, gracias a su buena respuesta en frecuencia, el hecho de que exista esta atenuación debida al transformador hace que los resultados de SNR obtenidos se reduzcan.

Para la realización de las simulaciones se ha generado un modelo del canal de transmisión. Para obtener el modelo de canal se han realizado transmisiones en el canal a lo largo de las distintas subportadoras y almacenando los datos recibidos. Con los datos almacenados de distintas transmisiones se ha obtenido la estimación para cada subcanal, obteniendo así un modelo de atenuación para cada una de las subportadoras. Este modelo se ha empleado en todas las simulaciones realizadas, tanto en coma fija, como en coma flotante. De igual modo, se mantiene la limitación de ancho de palabra de 16 bits en la entrada y salida de la arquitectura de la técnica de acceso al medio.

Como se observa en la Tabla 44, en este caso existe una mayor diferencia entre los valores obtenidos con el modelo en coma fija y la prueba experimental, debido a que el modelo de canal empleado en simulación es una estimación del canal real basado en el

cable SMA, y por tanto, no se corresponde exactamente con la realidad. Si bien, pese a este inconveniente, se puede observar como la técnica basada en banco de filtros y en DTT son ligeramente mejores en prestaciones que la técnica mono-portadora.

Tabla 44. Resultados obtenidos en las pruebas reales realizadas para el canal SMA.

	Técnica	SNR	Error cuadrático medio	Pico de SNR	Error máximo
Prueba experimental	FBMC	29.9597 dB	$5.0000 \cdot 10^{-4}$	34.6636 dB	0.0581
	DTT	31.0939 dB	$5.0000 \cdot 10^{-4}$	35.7978 dB	0.0475
	MONO	27.1159 dB	$6.0000 \cdot 10^{-4}$	33.7458 dB	0.0809
Modelo coma fija	FBMC	35.8589 dB	$3.3256 \cdot 10^{-4}$	40.5629 dB	0.0351
	DTT	35.9611 dB	$3.2851 \cdot 10^{-4}$	40.6650 dB	0.0331
	MONO	30.1301 dB	$2.7838 \cdot 10^{-4}$	36.7607 dB	0.0432
Modelo coma flotante	FBMC	37.1366 dB	$3.1648 \cdot 10^{-4}$	41.8406 dB	0.0332
	DTT	37.5151 dB	$3.1322 \cdot 10^{-4}$	42.2190 dB	0.0304
	MONO	30.4274 dB	$2.5763 \cdot 10^{-4}$	37.0580 dB	0.0418

De la tabla se puede extraer que vuelve a existir una diferencia entre el modelo en coma fija y la prueba experimental. Esta diferencia se debe no sólo al hecho de disponer de un modelo que no caracteriza exactamente el comportamiento interno del core de la FFT, sino también al hecho de que el modelo del canal de transmisión es una estimación. El modelo del canal es menos agresivo que el cable SMA y por lo tanto se observa la diferencia de hasta 6dB del modelo en coma fija, respecto a la prueba experimental.

En el caso del modelo en coma flotante, también se utiliza este modelo del canal de transmisión. Atendiendo a los valores de las simulaciones en coma flotante y coma fija, se observa que se ha reducido significativamente la diferencia que existía entre ellos respecto a la prueba con el canal ideal. Esto se debe a que el error que introduce el canal de transmisión es predominante sobre el error producido por el efecto de la precisión finita.

Por último, los resultados obtenidos con la técnica mono-portadora son diferentes que los obtenidos con las técnicas multi-portadoras DTT y FBMC. Debido a que sólo se realiza procesamiento de datos en el receptor, el efecto del canal de transmisión afecta de una manera mayor que en los casos de las otras técnicas, las cuales realizan en el transmisor una modulación de las señales en distintas subportadoras en el dominio de la frecuencia.

Al igual que para el anterior canal, a continuación se procede a mostrar la representación de la media de la relación señal ruido de las subportadoras de las técnicas, para las simulaciones y la prueba experimental.

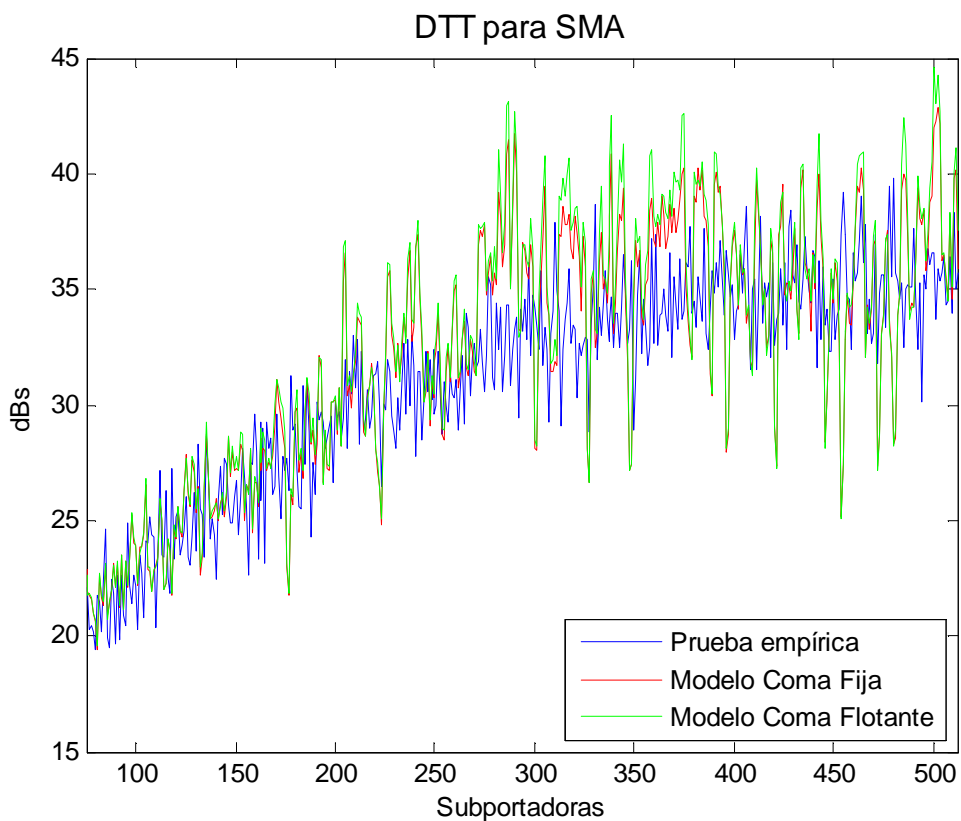


Figura 92. Representación de SNR en el canal SMA para las distintas subportadoras de la técnica DTT.

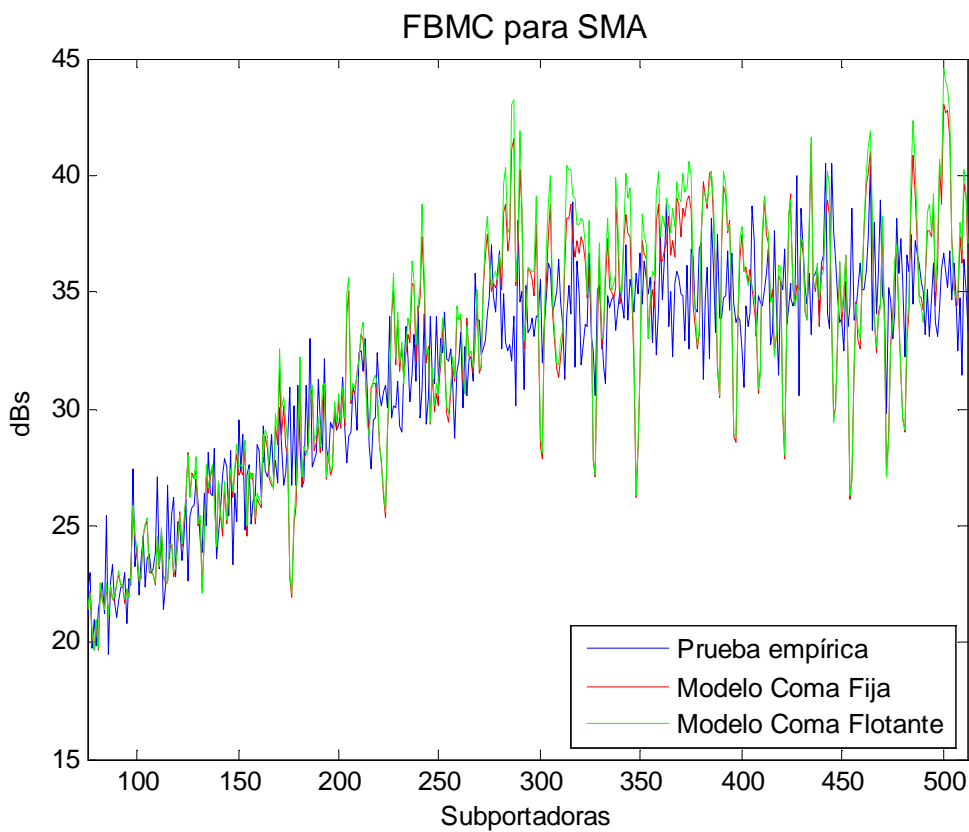


Figura 93. Representación de SNR en el canal SMA para las distintas subportadoras de la técnica FBMC.

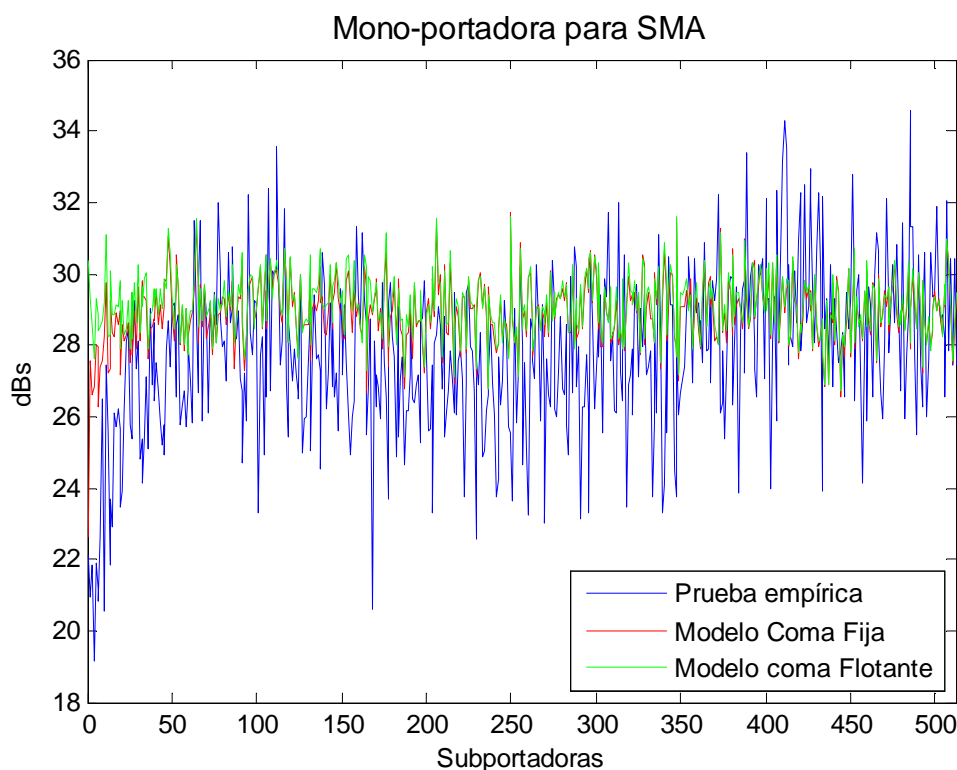


Figura 94. Representación de SNR en el canal SMA para las distintas subportadoras de la técnica mono-portadora.

De la observación de las figuras se pueden extraer consideraciones similares a las que ya se obtuvieron en la Tabla 44. La diferencia que existía entre la simulación en coma flotante y en coma fija y la prueba empírica se ha reducido. Esto es debido a que el efecto del canal de transmisión es predominante sobre el efecto de la precisión finita.

Por otro lado, existe una diferencia entre las simulaciones y la prueba empírica. Ésta se produce por la utilización de una estimación del canal empleado en la prueba empírica, que no se ajusta totalmente a la realidad. Este modelo del canal de transmisión ha sido estimado a partir de la media de varias transmisiones, lo cual da lugar a la obtención de un modelo menos agresivo que el empleado en la realidad.

Además, en las técnicas DTT y FBMC se produce una atenuación progresiva de mayor a menor frecuencia en las subportadoras. Esto se produce a causa del efecto del canal de transmisión combinado con el procesamiento de la FFT realizada en el transmisor. También se puede observar que en el caso de la técnica mono-portadora, al no producirse ningún tipo de procesamiento en los datos del transmisor, este hecho no se produce. Si bien, el resultado final es que la atenuación media de la técnica mono-portadora es mayor, debido a que los datos son transmitidos sin empleo de subportadoras y por tanto está más expuesta al efecto del canal.

Para este canal de transmisión, además de obtener los parámetros definidos para cada subportadora, se ha realizado el cálculo del SER (Symbol Error Rate). Para ello, se han transmitido los datos con distintas modulaciones PAM (2, 4, 8, 16, 32) según indica el

7.5 Resultados obtenidos con un canal similar al real

estándar de PLC. En la Tabla 45 se muestran los parámetros obtenidos para cada una de ellas en las distintas técnicas.

Tabla 45. SER obtenido para distintas modulaciones PAM en el canal SMA.

	PAM	2	4	8	16	32
Prueba experimental	FBMC	0.0000	0.0000	0.0002	0.0170	0.1142
	DTT	0.0000	0.0000	0.0003	0.0186	0.1094
	MONO	0.0000	0.0000	0.0011	0.0169	0.1491
Modelo coma fija	FBMC	0.0000	0.0000	$3.1465 \cdot 10^{-4}$	0.0156	0.0792
	DTT	0.0000	0.0000	$7.3242 \cdot 10^{-5}$	0.0132	0.0693
	MONO	0.0000	0.0000	0.0000	0.0140	0.0890
Modelo coma flotante	FBMC	0.0000	0.0000	$2.2883 \cdot 10^{-4}$	0.0149	0.0764
	DTT	0.0000	0.0000	$6.7656 \cdot 10^{-5}$	0.0127	0.0667
	MONO	0.0000	0.0000	0.0000	0.0100	0.0795

Como se observa en la tabla, el valor del SER aumenta con el número de niveles empleados en la modulación PAM debido a que, al usar un número mayor de niveles, la distancia entre ellos es menor y por tanto existe una mayor probabilidad de error entre ellos.

Al mismo tiempo, se observa que para las modulaciones PAM de 2 y 4 niveles la tasa de error es nula. Esto indica que pese a la pérdida de dB debido al efecto del canal de transmisión y del transformador, es posible recuperar todos los datos transmitidos con estas modulaciones.

7.5. Resultados obtenidos con un canal similar al real

Esta prueba emplea un canal similar al real, ya que se conecta la salida del DAC a un cable eléctrico de doce metros de longitud, pero sin alimentación eléctrica debido a que no se dispone en el momento de las pruebas de un AFE comercial para la conexión. Con este canal se van a observar tanto efectos de multi-camino, como efectos de atenuación de la señal transmitida.

Al igual que en el anterior apartado, para la realización de las simulaciones se empleará un modelo del canal de transmisión. Para la generación de este canal se enviarán señales a través del canal para cada subportadora, almacenando los valores recibidos. De este modo, se realizará una estimación de la atenuación de cada subcanal, generando el modelo del canal. Como se observa en la Tabla 46 existe una diferencia entre los valores obtenidos con el modelo en coma fija y la prueba experimental, debido a que el modelo de canal empleado en simulación no se corresponde exactamente con la realidad.

De acuerdo a los resultados de la tabla, continúa existiendo diferencias entre la prueba experimental y el modelo en coma fija, debido al empleo en la simulación de un modelo del canal de transmisión que no se ajusta exactamente al empleado en la prueba real. No obstante, es lo suficientemente parecido como para poder continuar con la comparativa.

Adicionalmente, se observa que los resultados obtenidos por la técnica mono-portadora difieren de las otras dos técnicas. Se observa que la prueba empírica obtiene un resultado mejor que en las simulaciones. Esto se produce porque el modelo del canal de transmisión

implementado afecta en mayor medida a la técnica que no emplea las subportadoras y por tanto está más expuesta al efecto del canal.

Tabla 46. Resultados obtenidos en las pruebas para una canal real.

	Técnica	SNR	Error cuadrático medio	Pico de SNR	Error máximo
Prueba experimental	FBMC	4.6250 dB	0.1095	10.4919 dB	0.7393
	DTT	4.3635 dB	0.1141	10.2529 dB	0.7744
	MONO	5.0397 dB	0.1045	9.7292 dB	0.6659
Modelo coma fija	FBMC	5.7819 dB	0.1127	10.4714 dB	0.5853
	DTT	5.7616 dB	0.1123	10.4510 dB	0.5799
	MONO	4.4173 dB	0.1208	9.1076 dB	0.8018
Modelo coma flotante	FBMC	5.7850 dB	0.1126	10.4745 dB	0.5846
	DTT	5.7632 dB	0.1122	10.4526 dB	0.5791
	MONO	4.4182 dB	0.1207	9.1077 dB	0.8012

También hay que destacar el hecho de que los resultados del modelo en coma flotante sean prácticamente iguales que el modelo en coma fija. Esto se produce porque el error introducido por el canal de transmisión enmascara las diferencias que existen entre la simulación en coma flotante y coma fija.

Por último, la excesiva pérdida de dB sufrida por las técnicas en esta prueba es debida a la gran atenuación y a la respuesta no plana en frecuencia sufrida por la transmisión de los datos a través de un cable eléctrico de semejante longitud, junto al hecho de que no se emplea ningún tipo de igualación o de amplificación de señal. Además, sigue presente la atenuación introducida por el transformador del DAC.

A continuación se muestra la representación de la media de la relación señal-ruido para cada subportadora de las técnicas analizadas, tanto para la prueba empírica, como para las simulaciones.

7.5 Resultados obtenidos con un canal similar al real

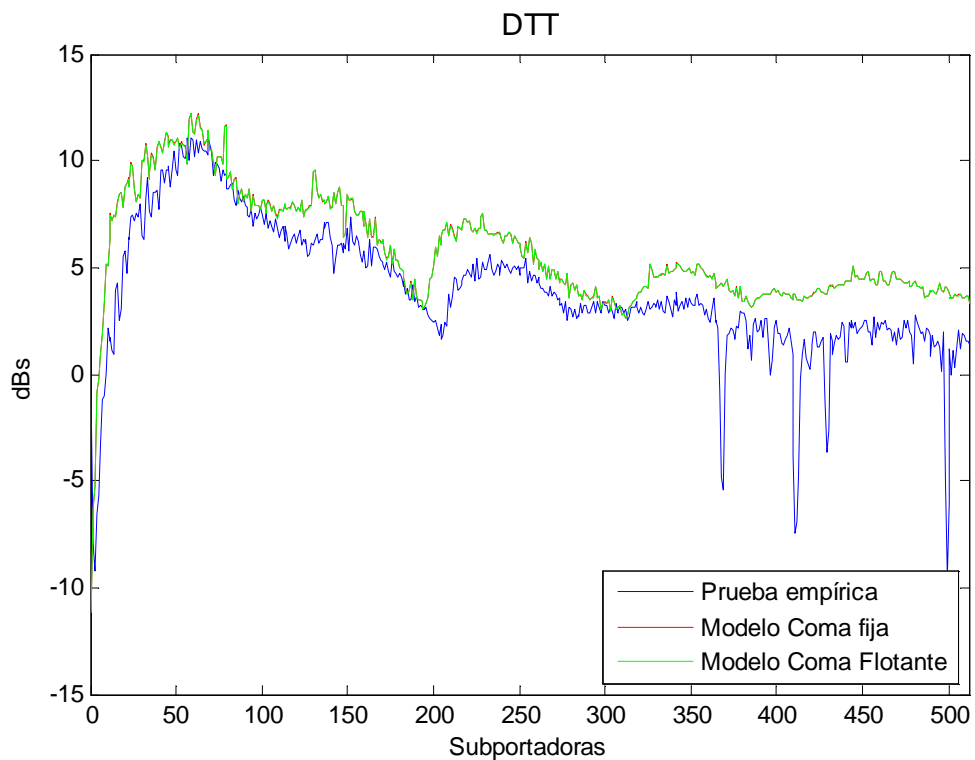


Figura 95. Representación de SNR para el canal basado en el cable eléctrico de 12 m para las distintas subportadoras de la técnica DTT.

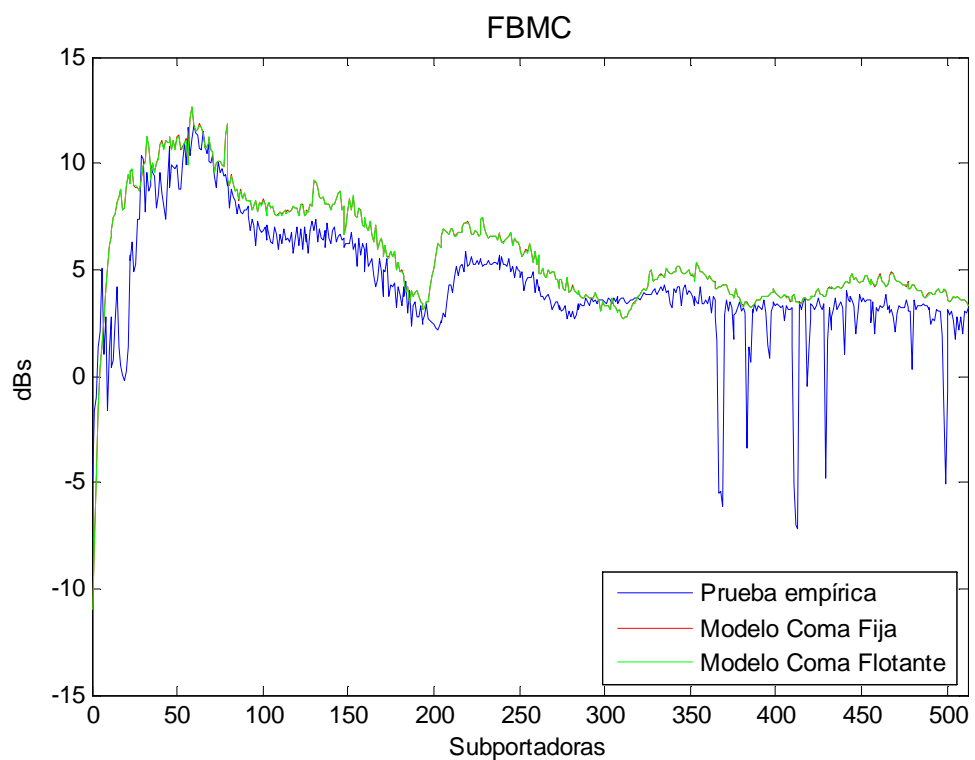


Figura 96. Representación de SNR para el canal basado en el cable eléctrico de 12 m para las distintas subportadoras de la técnica FBMC.

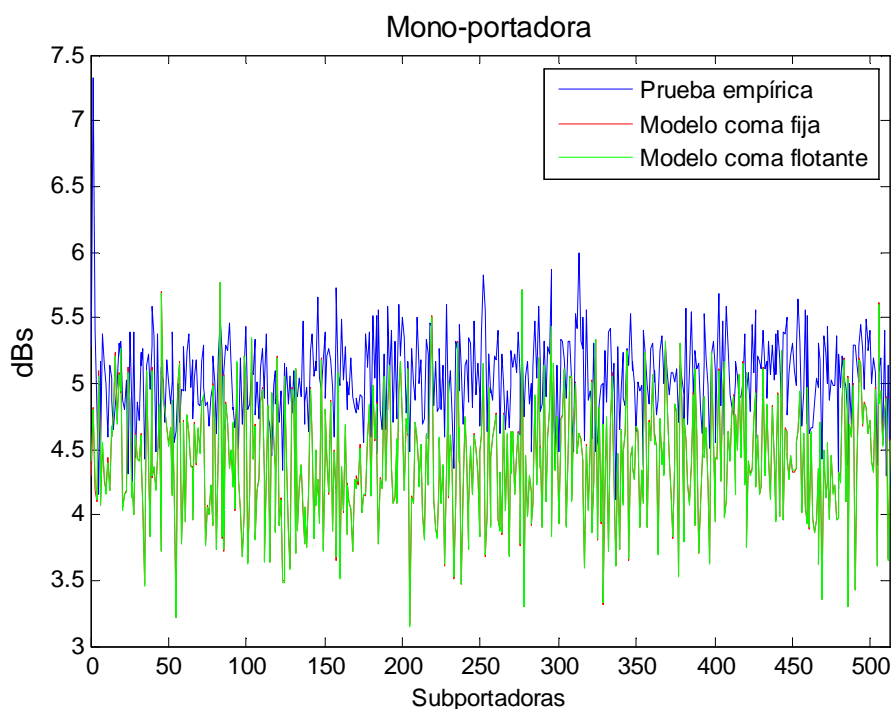


Figura 97. Representación de SNR para el canal basado en el cable eléctrico de 12 m para las distintas subportadoras de la técnica de mono-portadora.

Como se observa en las imágenes, no existe diferencia entre la simulación en coma flotante y en coma fija y la prueba empírica, debido a que el efecto del canal de transmisión es tan predominante que enmascara cualquier otro efecto de diferencia entre las pruebas.

Por su parte, se observa en las técnicas multi-portadora una atenuación progresiva del SNR según aumenta la frecuencia de transmisión en las subportadoras. Esto se debe principalmente a que el canal de transmisión empleado no está diseñado para transmisiones a altas frecuencias. Sin embargo, este efecto no se percibe en la técnica mono-portadora, debido a que los datos transmitidos no son enviados en múltiples subportadoras a distintas frecuencias.

Al igual que en la anterior prueba, para este canal de transmisión se ha realizado el cálculo del SER (Symbol Error Rate), transmitiendo los datos con distintas modulaciones PAM (2, 4, 8, 16, 32) según indica el estándar de PLC. Los resultados obtenidos se muestran en la Tabla 47.

Tabla 47. SER obtenido para las modulaciones PAM en el canal basado en el cable eléctrico de 12 m.

	PAM	2	4	8	16	32
Prueba real	FBMC	0.0719	0.4807	0.7424	0.8767	0.9396
	DTT	0.0838	0.5111	0.7540	0.8787	0.9365
	MONO	0.1064	0.5286	0.7692	0.8807	0.9409
Modelo coma fija	FBMC	0.0712	0.4976	0.7542	0.8773	0.9400
	DTT	0.0785	0.5089	0.7563	0.8783	0.9392
	MONO	0.1816	0.5889	0.7924	0.8953	0.9496
Modelo coma flotante	FBMC	0.0919	0.5047	0.7563	0.8766	0.9379
	DTT	0.0971	0.5062	0.7543	0.8763	0.9370
	MONO	0.2230	0.5875	0.7856	0.8900	0.9452

7.5 Resultados obtenidos con un canal similar al real

Se observa que los valores de SER obtenidos con este tipo de canal se incrementan respecto a la anterior prueba realizada, debido principalmente a la fuerte presencia de ruido y a una respuesta no plana en frecuencia en la señal provocado por el canal de transmisión, sumado a la existencia de la atenuación por transmitir los datos a cierta distancia. En cualquier caso, dado que los módulos de sincronismo y estimación e igualación de canal son básicos en cualquier etapa de transmisión PLC, así como el empleo de un AFE para la conexión de la técnica al cable eléctrico, cabe esperar la mejora de las cifras obtenidas con la presencia de los mismos en futuras pruebas experimentales.

Capítulo 8

Conclusiones

Como resultado de la investigación, estudio, análisis y desarrollo presentado, se concluye el diseño satisfactorio de arquitecturas eficientes para la implementación en tiempo real sobre dispositivos FPGA de diversas técnicas de acceso al medio para un canal PLC de banda ancha. Para ello, se ha propuesto una arquitectura SoC global en la cual se insertan las arquitecturas particulares para cada transmisor y receptor. Las pruebas experimentales han validado de forma preliminar las propuestas realizadas. A partir de esta conclusión general, se pueden deducir las siguientes conclusiones parciales:

Como se ha descrito en el estado del arte, pese a que la técnica OFDM ha sido la técnica multi-portadora de acceso al medio más empleada en las últimas décadas, posee varios inconvenientes que hacen necesario la búsqueda de alternativas a ésta. Por ello, en diversos artículos se ha propuesto el uso de la Transformada Discreta del Coseno como alternativa a la Transformada Discreta de Fourier, ya que se consigue una mejor compactación de la energía y una mejora del PAPR.

Para el desarrollo de las arquitecturas de estas nuevas técnicas de acceso al medio se hace necesario la búsqueda de alternativas tecnológicas adecuadas que se revisan en el estado del arte. Entre las posibles destaca la utilización de las FPGAs. Éstas se caracterizan por la posibilidad de implementar arquitecturas paralelas en tiempo real, lo cual es inviable en microcontroladores y microprocesadores. Además, el uso de FPGAs permite una reconfiguración de la arquitectura sin la necesidad de rehacer el diseño que no permiten los

ASIC. Por tanto, estas características hacen idóneas a las FPGAs para la definición y diseño de nuevas arquitecturas para las técnicas de acceso al medio analizadas en esta tesis.

Para la implementación de las distintas arquitecturas de las técnicas de acceso al medio se ha diseñado una arquitectura SoC general en la que poder implementarlas como periféricos avanzados. Esta arquitectura general se ha diseñado a lo largo de todo el trabajo desarrollado en el capítulo 3 dedicado a la descripción de la arquitectura SoC propuesta. La arquitectura ha integrado un microprocesador Microblaze para realizar, tanto la supervisión del sistema, como la gestión de la transferencia de los datos. Además Microblaze también realiza una comprobación de la calidad de la transmisión, al comparar los datos transmitidos con los datos recibidos. Por otro lado la arquitectura general incluye un módulo DMA encargado de proporcionar los datos a transmitir y almacenar los datos recibidos. Gracias a sus propiedades, el DMA es capaz de proporcionar el flujo de datos necesario para cada técnica de acceso al medio.

Esta arquitectura ha permitido que se puedan implementar distintas técnicas de acceso al medio, tanto multi-portadora como mono-portadora, siguiendo el estándar de PLC de banda ancha y permitiendo que todo el sistema opere en tiempo real. Este estándar indica una tasa de transmisión y un número de subportadoras tal, que hacen que la implementación sea un reto.

En el capítulo 4 referente a la implementación de la arquitectura para la técnica de acceso al medio basada en la Transformada Trigonométrica Discreta, se ha realizado el estudio de tres algoritmos para el desarrollo de la arquitectura de la Transformada Discreta del Coseno (DCT). Con estos algoritmos se ha buscado un bajo consumo de recursos, un tiempo de ejecución que permita su implementación en tiempo real, y obtener el menor efecto de la precisión finita. Como se ha observado a lo largo del capítulo, la DCT y más exactamente la FFT, ha sido el módulo que más consumo de recursos emplea debido a la realización de una operación tan compleja. Además, debido a esta operación tan compleja, el efecto de la precisión finita tiene un mayor impacto en el módulo.

El capítulo también ha servido para mostrar la versatilidad de la arquitectura diseñada, ya que es totalmente configurable, y gracias al ratio de paralelismo R_p es posible obtener una reutilización de los recursos empleados, consiguiendo un compromiso entre throughput y recursos empleados. Además, gracias al desarrollo y comparación de los tres algoritmos estudiados se ha podido apreciar cómo el algoritmo 1 propuesto en [102] y presentado para la arquitectura de la DCT genera un menor error debido al efecto de la precisión finita para la técnica de acceso al medio.

En el capítulo de la técnica de acceso al medio basada en filtros polifásicos se ha realizado el estudio y diseño de la correspondiente arquitectura. Como esta técnica también se basa en una DCT, se ha decidido emplear el mismo algoritmo expuesto en el capítulo anterior. Asimismo, debido a la existencia de un banco de filtros, se ha realizado un estudio de cuatro tipos de estructuras de filtros para reducir el efecto de la precisión finita en este banco y a la vez obtener un consumo de recursos reducido. Dentro de este estudio se ha

contemplado la utilización de filtros directos y los filtros en celosía genérico, tipo I y tipo II, realizándose una comparativa entre ellos tanto a nivel de consumo de recursos, como de efecto de la precisión finita. Como se ha observado en el estudio para el desarrollo de los filtros polifásicos en la FPGA Virtex 6 xc6vlx240t [110] la arquitectura de filtros directos es la mejor opción, gracias a que la FPGA dispone de unas celdas de multiplicación-acumulación idóneas para el mismo.

En el capítulo 6, se ha desarrollado la arquitectura para una técnica de acceso al medio mono-portadora. La característica de esta arquitectura es que la DCT empleada en el transmisor se sitúa en el receptor. De esta manera, el transmisor queda simplificado, mientras que la recepción incluye dos DCTs para incluir etapa de igualación de canal en el dominio de la frecuencia (FDE), si bien, en esta tesis no se ha abordado la misma. De este capítulo se deduce que debido al algoritmo propuesto en la técnica mono-portadora hace que se produzca una simplificación muy considerable de la arquitectura del transmisor. Mientras, el estudio realizado en la DCT de la precisión finita demuestra que es posible la implementación de dos DCTs contiguas para poder realizar la igualación en el dominio de la frecuencia, sin que el efecto de la precisión finita genere error adicional.

En el capítulo de resultados se ha explicado cómo se han evaluado las arquitecturas implementadas, realizando distintas pruebas reales con canales de transmisión de distintas longitudes y diferentes tipos de ruidos. A su vez, para realizar un estudio más completo, estas pruebas reales han sido comparadas con simulaciones de los modelos implementados en coma fija y coma flotante. Esto ha permitido obtener una comprobación directa entre los datos obtenidos y los simulados.

Para comprobar las arquitecturas implementadas se han realizado tres pruebas reales. La primera prueba realizada conecta internamente en la FPGA el transmisor con el receptor, lo que permite obtener una canal de transmisión ideal en modo digital. Con esta prueba se obtienen las prestaciones de las técnicas de transmisión de formas aislada, sin canal, pudiendo comparar claramente los efectos de la precisión finita en la arquitectura. Asimismo, se deduce que el hecho de tener una limitación en el ancho de palabra del canal de transmisión hace que exista una atenuación en las primeras subportadoras, aún en el caso del modelo en coma flotante. En la técnica mono-portadora esto no se produce, al no realizar ningún procesamiento de datos en el transmisor. Para el modelo en coma fija y la prueba empírica esta atenuación se acentúa, debido al efecto de la precisión finita.

La segunda prueba realizada emplea un conversor digital-analógico y un conversor analógico-digital para transmitir los datos fuera de la FPGA y de esta manera, realizar una prueba con el diseño completo implementado. Para la conexión entre los conversores se emplea un canal SMA que presenta una buena respuesta en frecuencia y por tanto se aproxima a un canal ideal. Al realizar esta prueba se observa una caída significativa del SNR respecto a la anterior prueba, generada por un transformador situado a la salida del DAC. Pese a este inconveniente se observa cómo se consigue realizar una transmisión correcta, e incluso sin errores al aplicar modulación en amplitud (PAM).

Por último, en la tercera prueba se conectan los convertidores a un cable eléctrico de 12 metros, con lo que se obtiene un entorno más próximo a la situación real para probar las arquitecturas desarrolladas. En este caso se alcanzan unos resultados bajos de SNR, debido a los distintos ruidos que se introducen, junto al hecho de que no se utiliza un AFE para atacar la parte analógica, ni se emplea una estimación e igualación de canal. Sin embargo, pese al bajo SNR, aplicando modulación PAM es posible una transmisión con un SER (Symbol Error Rate), variable según las condiciones de la prueba.

Se advierte que las mayores fuentes de error se deben principalmente a los siguientes factores: a la fuerte presencia de ruido en la señal y a la respuesta en frecuencia no plana, ya que no se realiza ningún tipo de igualación del canal de transmisión; y al hecho de no disponer de un AFE (Analog Front-End) comercial para la conexión a la red eléctrica. En cualquier caso, dado que los módulos de sincronismo y estimación e igualación de canal, así como el empleo de un AFE comercial son básicos en cualquier etapa de transmisión PLC, cabe esperar la mejora de las cifras obtenidas con la presencia de los mismos en futuras pruebas experimentales.

8.1. Trabajos futuros

En esta sección se perfilan las líneas de investigación futuras que se derivan del trabajo realizado en esta tesis. Aunque los resultados obtenidos establecen un sólido punto de partida en cuanto al diseño de una arquitectura para el desarrollo de nuevas técnicas de acceso al medio para comunicaciones PLC de banda ancha, existen sin embargo evidentes márgenes de mejora en diversos ámbitos que el autor considera que deben ser la continuación de los trabajos que se han desarrollado en la tesis. La ampliación y mejora de estos trabajos tendrá que ver, tanto con la realización de una estimación e igualación de canal y la realización de una sincronización apropiada, como con el desarrollo de un AFE para realizar pruebas en un entorno completamente real.

En primer lugar, después de observar la gran influencia que posee el módulo FFT en los resultados obtenidos, debido al efecto de la precisión finita, se hace necesario el estudio y desarrollo de un módulo propio de la FFT. Se tendrá que realizar un estudio de distintas arquitecturas, un análisis de cuál es la arquitectura con mejor optimización posible y el desarrollo final. Será de vital importancia conseguir reducir el efecto de la precisión finita en la arquitectura, ya que a lo largo de toda la tesis se ha observado que su impacto en el resultado final es considerable.

Desde el punto de vista de la estimación e igualación de canal y la sincronización, la idea en un futuro próximo es la implementación de un nuevo módulo en la arquitectura general que permita, no sólo la sincronización de transmisor y receptor en una misma FPGA, sino que sea capaz de realizar esta sincronización cuando transmisor y receptor están en distintas FPGA, con la dificultad añadida que implica. La realización de esta sincronización supone también la realización de una estimación e igualación de canal que permita mejorar las prestaciones obtenidas al realizar la transmisión en un canal agresivo.

En la parte que concierne al desarrollo de un AFE y la transmisión de los datos fuera de la FPGA, son varias las mejoras que se tiene intención de llevar a cabo. Estas mejoras esencialmente tienen que ver con la creación y diseño de un AFE capaz de trabajar a la tasa de transmisión que se determina en el estándar de PLC de banda ancha [27]. De este modo, se buscará hacer un sistema capaz de conectarse en cualquier entorno para la realización de transmisiones, y obteniendo por tanto, una notable mejora en la transmisión de los datos.

Otra de las líneas futuras de trabajo que se llevará a cabo tiene que ver con el desarrollo de la modulación previa a la transmisión con las técnicas de acceso al medio. Dado que en el trabajo desarrollado en la tesis sólo se ha contemplado la transmisión de los datos directos o aplicando un tipo de modulación sencilla, se considera que introduciendo una modulación más compleja y técnicas de corrección de errores se pueden mejorar las prestaciones obtenidas en las técnicas de acceso al medio. En este sentido, existen numerosos trabajos previos que determinan que las técnicas de bit loading reducen el número de bits que se reciben de manera errónea, mejorando la calidad de la transmisión en técnicas multi-portadoras.

Por último, gracias a que la idea de esta propuesta es albergar distintas arquitecturas de técnicas de acceso al medio y que se puedan adaptar a distintos estándares, la propuesta en sí se convierte en una perfecta herramienta de trabajo para implementar nuevas técnicas de acceso al medio que se desarrollen en el futuro. Si bien, se debe incidir en la actualización de la plataforma SoC utilizada hasta ahora, dado que han aparecido nuevas plataformas hardware y entornos de desarrollo, quedando el empleado hasta ahora obsoleto. Asimismo, gracias a los convertidores analógicos-digitales que permiten trabajar con distintas frecuencias de muestreo, sería posible conectar la plataforma a distintos canales y/o estándares.

8.2. Publicaciones relacionadas con la Tesis

En esta sección se relacionan las publicaciones en revistas indexadas y las contribuciones a congresos nacionales e internacionales derivadas del trabajo realizado en esta tesis.

8.2.1. Publicaciones en Revistas Internacionales

Pablo Poudereux, Álvaro Hernández, Raúl Mateos, Freddy A. Pinto-Benel, Fernando Cruz-Roldan, *Design of a filter bank multi-carrier system for broadband power line communications*, Signal Processing, 2016, vol. 128, pp. 57-67.

Pablo Poudereux, Álvaro Hernández, Raúl Mateos. *A comparison between real time implementations for Broadband PLC channel*, IEEE Transactions on Circuits and Systems I, 2016, en revisión.

8.2.2. Publicaciones en Congresos Internacionales

Pablo Poudereux, Raúl Mateos, Álvaro Hernández, Fernando Cruz-Roldán, David Osés, *FPGA-based Implementation of a Filter Bank-based Transmultiplexer for Multicarrier Communications*, IEEE Emerging Technology and Factory Automation (ETFA'14), 2014, Barcelona (España), pp. 1-6.

Francisco Pérez Fermoselle, Francisco Daniel Ruiz Pereda, Álvaro Hernández, Enrique García, Pablo Poudereux, Jesús Ureña, *Design of a System-on-Chip for controlling and Ultrasonic Beacon Array*, IEEE Emerging Technology and Factory Automation (ETFA'14), 2014, Barcelona (España), pp. 1-4.

Pablo Poudereux, Raúl Mateos, Álvaro Hernández, Francisco Nombela, Fernando Cruz-Roldán, *Study of Suitable Filter Architectures for FBMC Techniques Applied to PLC Communications*, IEEE Emerging Technology and Factory Automation (ETFA'15), 2015, Luxembourg (Luxembourg), pp. 1-6.

Francisco Nombela, Enrique García, Jesús Ureña, Álvaro Hernández, Pablo Poudereux, *Robust Synchronization Algorithm for Broadband PLC based on Wavelet-OFDM*, IEEE Emerging Technology and Factory Automation (ETFA'15), 2015, Luxembourg (Luxembourg), pp. 1-7.

8.2.3. Publicaciones en Congresos Nacionales

Pablo Poudereux, Álvaro Hernández, Raúl Mateos, Francisco Nombela, *Arquitectura SoC para el diseño de transmultiplexores en comunicaciones PLC*, Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI'15), 2015, Zaragoza (España), pp. 188-192.

Pablo Poudereux, Álvaro Hernández, Raúl Mateos, Francisco Nombela, *Implementación y Evaluación de Técnicas de Acceso al Medio en PLC de Banda Ancha*, Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI'16), 2016, Elche (España), pp. 1-6.

Francisco Nombela, Enrique García, Álvaro Hernández, Pablo Poudereux, *Diseño e Implementación de un Sistema Receptor para PLC de Banda Ancha*, Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI'16), 2016, Elche (España), pp. 1-5.

Bibliografía

- [1] I.-T. R. G.992.5, Asymmetric digital subscriber line (ADSL) transceivers - Extended bandwidth ASAL2 (ADSL2plus), 2005.
- [2] I.-T. R. G.993.2, Very high speed digital subscriber line 2 (VSLD2), 2006.
- [3] European Telecommunications Standards Institute ETSI EN 300 401, V1.4.1, Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers, Enero 2001.
- [4] European Telecommunications Standards Institute ETSI EN 302 769, V1.2.1, Digital Video Broadcasting (DVB); Frame structure, channel coding and modulation for a second generation digital transmission system for cable systems (DVB-C2), Abril 2011.
- [5] 3GPP Evolved Universal Terrestrial Radio Access (E-UTRA) v 8.4.0 Technical specification, TS 36.211, Physical channels and modulation, Septiembre 2008.

-
- [6] A. M. Wyglinski, F. Labeau y P. Kabal, «Bit Loading with BER-Constraint for Multicarrier Systems,» *IEEE Transactions on Wireless Communications*, vol. 4, n° 4, pp. 1383-1387, 2005.
- [7] M. Etelapera y J. Soininen, 4G Mobile Terminal Architectures, Technical Research Centre of Finland (VTT) Technical report, 2007.
- [8] J. Becker, N. Libeau, T. Pionteck y M. Glesner, «Efficient mapping of pre-synthesized IP-cores onto dynamically reconfigurable array architectures,» *11th International Conference on Field-Programmable Logic and Applications (FPL)*, pp. 584-589, 2001.
- [9] H. Zhang, V. Prabhu, V. George, M. Wan, M. Benes y A. Abnous, «A IV heterogeneous reconfigurable processor IC for baseband wireless applications,» *Proc. of the 2000 IEEE International Solid-State Circuits Conference*, pp. 68-69, 2000.
- [10] J. Rabaey, «Reconfigurable processing: the solution to low-power programmable DSP,» *Proc. of the 22nd IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, pp. 275-278, 1997.
- [11] Y. Ha, B. Mei, P. Schaumont, S. Vernalde, R. Lauwereins y H. De Man, «Development of a design framework for platform-independent networked reconfiguration for software and hardware,» *Proc. of the 11th International Conference on Field-Programmable Logic and Applications (FPL)*, 2001.
- [12] S. Knapp, «Field configurable system on chip devices architecture,» *Proc. of the 22nd IEEE Custom Integrated Circuits Conference (CICC)*, pp. 155-158, 2000.
- [13] M. Fijnvandraat y H. Bowman, «Flexibility and broadband evolution,» *Telecommunications Policy*, vol. 30, n° 8, pp. 424-444, 2006.
- [14] E. Davison y S. R. Cotten, «Connection discrepancies: Unmasking further layers of the digital divide,» *First Monday*, vol. 8, n° 3, Marzo 2003.
- [15] K. J. Kerpez, «DSL spectrum management standard,» *IEEE Communications Magazine*, vol. 40, n° 11, pp. 116-123, Nov 2002.

-
- [16] D. L. Waring, «The asymmetrical digital subscriber line (ADSL): a new transport technology for delivering wideband capabilities to the residence,» *GLOBECOM'91. Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services*, pp. 1979-1986, Dic 1991.
- [17] J. Lechleider, «High bit rate digital subscriber lines: a review of HDSL progress,» *IEEE Journal on Selected Areas in communications*, vol. 9, n° 6, pp. 769-784, Ago 1991.
- [18] J. Lechleider, «Line codes for digital subscriber lines (ISDN basi access),» *IEEE Communications Magazine*, vol. 27, n° 9, pp. 25-32, Ago 2002.
- [19] D. Fellows y D. Jones, «DOCSIS™ cable modem technology,» *IEEE Communications Magazine*, vol. 39, n° 3, pp. 202-209, Ago 2002.
- [20] E. Strange y B. E. Keiser, «Fiber-optic Transmission,» de *Digital Telephony and Network Integration*, Springer, 1995, pp. 333-372.
- [21] L. Nuaymi, *WiMAX technology for Broadband wireless Access*, Bretagne: John Wiley & Sons, 2007.
- [22] S. W. Peters y R. W. Heath, «The future of WiMAX: Multihop relaying with IEEE 802.16j,» *IEEE Communications Magazine*, vol. 47, n° 1, pp. 104-111, Ene 2009.
- [23] P. Henry y H. Luo, «WiFi: what's next?,» *IEEE Communications Magazine*, vol. 40, n° 12, pp. 66-72, Dic 2002.
- [24] J. Farserotu y R. Prasad, «A survey of future broadband multimedia satellite systems, issues and trends,» *IEEE Communications Magazine*, vol. 38, n° 6, pp. 128-133, Jun 2000.
- [25] D. J. Bem, «Broadband satellite systems,» *IEEE Communications surveys & Tutorials*, vol. 3, n° 1, pp. 2-15, Diciembre 2009.
- [26] K. Dostert, *Powerline Communications*, USA: NJ, 2001.
- [27] IEEE Std 1091-2010, «IEEE standard for broadband over power line networks: Medium access control and physical layer specifications,» pp. 1-1586, 2010.

-
- [28] S. Qureshi, «Adaptive equalization,» *Proceedings of the IEEE*, vol. 73, n° 9, pp. 1349-1387, 1985.
- [29] H. Sari, G. Karam y I. Jeanclaude, «Frequency-domain equalization of mobile radio and terrestrial broadcast channels,» *Global Telecommunications Conference GLOBECOM Communications: The Global Bridge*, pp. 1-5, 1994.
- [30] K. Pietikäinen, «Orthogonal frequency division multiplexing,» Internet presentation, www.comlab.hut.fi/opetus/333/2004_2005_slides/ofdm_text.pdf, 2005.
- [31] R. van Nee y R. Prasad, *OFDM for Wireless Multimedia Communications*, Norwood: Artech House, Inc, 2000.
- [32] O. Edfors, M. Sandell, J. van de Beek, D. Landström y F. Sjöberg, «An introduction to orthogonal frequency-division multiplexing,» de *Research Report div. of Signal Processing*, 1996.
- [33] L. Litwin y M. Pugel, «The principles of OFDM,» *RF signal processing*, vol. 2, pp. 30-48, 2001.
- [34] W. Shieh y I. Djordjevic, *OFDM for optical communications*, Academic Press, 2009.
- [35] J. Armstrong, «OFDM for optical communications,» *Lightwave Technology, Journal of*, vol. 27, n° 3, pp. 189-204, 2009.
- [36] J. S. Chow, J. Tu y J. Cioffi, «A discrete multitone transceiver system for HDSL applications,» *IEEE Journal on Selected Areas in Communications*, vol. 9, n° 6, pp. 895-908, 1991.
- [37] J. Lee, H.-L. Lou, D. Toumpakaris y J. M. Cioffi, «Effect of Carrier Frequency Offset on OFDM Systems for Multipath Fading Channels,» *IEEE Global Telecommunications Conference GLOBECOM*, vol. 6, pp. 3721-3725, 2004.
- [38] P. Robertson y S. Kaiser, «Analysis of the Loss of Orthogonality through Doppler Spread in OFDM systems,» *Global Telecommunications Conference GLOBECOM*, vol. 1b, pp. 701-706, 1999.

-
- [39] S. Ohmori, Y. Yamao y N. Nakajima, «The future generations of mobile communications based on broadband access technologies,» *IEEE Communications Magazine*, vol. 38, n° 12, pp. 134-142, 2000.
- [40] N. Al-Dhahir, H. Minn y S. Satish, «Optimum DCT-based multicarrier transceivers for frequency-selective channels,» *IEEE Trans. Commun.*, vol. 54, n° 5, pp. 911-921, May 2006.
- [41] F. Cruz-Roldán, M. Dominguez-Jimenez, G. Sansigre, M. Blanco-Velasco, P. Amo-López y Á. Bravo Santos, «On the use of discrete cosine transforms for multicarrier communications,» *IEEE Transactions on Signal Processing*, vol. 60, n° 11, pp. 6085-6091, 2012.
- [42] P. Tan y N. C. Beaulie, «A Comparison of DCT-Based OFDM and DFT-Based OFDM in Frequency Offset and Fading Channels,» *IEEE Transactions on Communications*, vol. 54, n° 11, pp. 2113-2125, 2006.
- [43] B. Farhang-Boroujeny, «OFDM versus filter bank multicarrier,» *IEEE Signal Processing Magazine*, vol. 28, n° 3, pp. pp. 92-112, 2011.
- [44] B. Devi Tensubam, N. Lalleima Chanu y S. Singh, «Comparative Analysis of FBMC and OFDM Multicarrier Techniques for Wireless communication Networks,» *International Journal of computer Applications*, vol. 100, n° 19, pp. 27-31, 2014.
- [45] B. Farhang-Boroujeny, «Filter bank multicarrier modulation: A waveform candidate for 5G and beyond,» *Advances in Electrical Engineering*, 2014.
- [46] F. Cruz-Roldán y M. Blanco-Velasco, «Joint Use of DFT Filter Banks and modulated Transmultiplexers for multicarrier Communications,» *Signal Processing*, vol. 91, n° 7, pp. 1622-1635, 2011.
- [47] E. Ospina-Martinez y L. Betancur, «FBMC, una opción para un uso óptimo del espectro electromagnético,» *Revista en Telecomunicaciones e Informatica*, vol. 3, n° 5, pp. 39-53, 2013.
- [48] Q. Bai, N. Passas y J. A. Nossek, «Scheduling and resource allocation in OFDM and FBMC systems: An interactive approach and performance comparison,» *European wireless Conference*, pp. 1042-1050, 2010.

-
- [49] M. Bellanger, M. Renfors, T. Ihalainen y C. A. F. Da Rocha, «OFDM and FBMC transmission techniques: a compatible high performance proposal for broadband power line communications,» de *2010 International Symposium on Power Line Communications and Its applications (ISPLC)*, Rio de Janeiro, 2010.
- [50] H. Witschnig, T. Mayer, A. Springer y A. Koppler, «A different look on cyclic prefix for SC/FDE,» *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 2, pp. 824-828, 2002.
- [51] D. Falconer, S. Ariyavisitakul, A. Benyamin-Seeyar y B. Eidson, «Frequency domain equalization for single-carrier broadband wireless systems,» *IEEE Communications Magazine*, vol. 40, n° 4, pp. 58-66, 2002.
- [52] A. Gusmao, R. Dinis, J. Conceicao y N. Esteves, «comparison of two modulation choices for broadband wireless communications,» *IEEE Vehicular Technology Conference Proceedings*, pp. 1300-1305, 2000.
- [53] H. G. Myung, «Introduction to single carrier FDMA,» *15th European Signal Processing Conference*, pp. 2144-2148, 2007.
- [54] Y. Wang y X. Dong, «Comparison of Frequency Offset and Timing Offset Effects on the Performance of SC-FDE and OFDM Over UWB Channels,» *IEEE Transactions on Vehicular Technology*, vol. 54, n° 1, pp. 242-250, 2009.
- [55] H. Witschnig, A. Koppler, A. Springer, R. Weigel y M. Huemer, «A comparison of an OFDM system and a single carrier system using frequency domain equalization,» *European Transactions on Telecommunications*, vol. 13, n° 5, pp. 519-530, 2002.
- [56] N. Benvenuto, R. Dinis, D. Falconer y S. Tomasin, «Single Carrier Modulation with Nonlinear Frequency Domain Equalization: An Idea Whose Time Has Come-Again,» *Proceedings of the IEEE*, vol. 98, n° 1, pp. 69-96, 2010.
- [57] H. G. Myung, J. Lim y D. J. Goodman, «Peak-To-Average Power Ratio of Single Carrier FDMA Signals with Pulse Shaping,» *IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1-5, 2006.
- [58] J. Li, Y. Du y Y. Liu, «Comparison of Spectral Efficiency for OFDM and SC-FDE under IEEE 802.16 Scenario,» *11th Symposium of Computers and Communications*, pp. 467-471, 2006.

-
- [59] Y. Han, H. huh y J. Krogmeier, «Comparison of error probability for OFDM and SC-FDE,» *Conference Record of the Thirty-seventh Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 497-501, 2004.
- [60] A. Nichols, «An overview of microprocessor applications,» *Proceedings of the IEEE*, vol. 64, n° 6, pp. 951-953, 1976.
- [61] V. Chawla y D. Ha, «Dual use of power lines for data communications in microprocessors,» *IEEE 14th international symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, pp. 23-28, 2011.
- [62] M. Predko, *Programming and customizing the PIC Microcontroler*, New York: McGraw-Hill, 1997.
- [63] J. Iovine, *PIC microcontroller project book*, McGraw-Hill, 2004.
- [64] K. Hayashi, F. Sasamori, O. Takyu y S. Handa, «Design and implementation of OFDM signal processing on PSOC microcontroller,» *International Conference on ICT Convergence (ICTC)*, pp. 391-392, 2013.
- [65] Q. Al-Zobi, I. Al-Tawil, K. Gharaibeh y I. Al-Kofahi, «Design of power-line communication system (PLC) using a PIC microcontroller,» *Journal of Active and Passive Electronic Devices*, vol. 1, n° 3, pp. 331-3340, 2008.
- [66] M. Frerking, *Digital signal processing in communications systems*, Springer Science & Business Media, 2013.
- [67] D. Stranneby, *Digital signal processing: DSP and applications*, Newnes, 2001.
- [68] J. Salazar, «Procesadores digitales de señal (DSP),» *Mundo electronico*, n° 314, pp. 46-57, 2000.
- [69] H. Yan, S. Zhou, Z. J. Shi y B. Li, «A DSP implementantion of OFDM acoustic modem,» *Proceedings of the second workshop on Underwater networks*, pp. 89-92, 2007.
- [70] M. Schöbinger y S. Meier, «DSP-based signal processing for OFDM transmission,» *IEEE International Conference on Acoustics, speech, and Signal Processing*, vol. 2, pp. 1249-1252, 2001.

-
- [71] M. Smith, *Application-specific integrated circuits*, Addison-Wesley Professional, 2008.
- [72] G. Lutz, «Application Specific Integrated Circuits,» de *XXIV International conference on High Energy Physics*, Berlin Heidelberg, Springer, 1989, pp. 1249-1255.
- [73] D. Perels, S. Haene, P. Luethi, A. Burg, N. Felber, W. Fichtner y H. Bölcskei, «ASIC implementation of a MIMO-OFDM Transceiver for 192 Mbps WLANs,» de *Proceedings of ESSCIRC*, Grenoble, France, 2005.
- [74] J. Nadal, C. Nour, A. Baghdadi y H. Lin, «Hardware prototyping of FBMC/OQAM baseband for 5G mobile communication,» *Proc. of 2014 25th IEEE International Symposium on Rapid System Prototyping (RSP)*, pp. 72-77, 2014.
- [75] L. Yu, M. Wang, J. Feng, X. Jiang y Y. Li, «The high-speed optical OFDM transmitter based on FPGA,» *IET International Conference on Smart and Sustainable City (ICSSC)*, pp. 415-418, 2013.
- [76] S. Kaur y R. Mehra, «FPGA implementation of OFDM transceiver using FFT algorithm,» *International Journal of Engineering Science and Technology (IJEST)*, vol. 4, n° 4, pp. 1532-1537, 2012.
- [77] N. Mev y B. Khaire, «Implementation of OFDM Transmitter and Receiver using FPGA,» *International Journal of Soft computing and Engineering (IJSCE)*, vol. 3, n° 3, pp. 199-202, 2013.
- [78] M. Mohamed, A. Samarah y M. Allah, «Implementation of the OFDM Physical Layer Using FPGA,» *IJCSI International Journal of computer Science ISSUES*, vol. 9, n° 2, pp. 612-618, 2012.
- [79] V. Ringset, H. Rustad, F. Schaich, J. Vandermot y M. Najar, «Performance of a FilterBank MultiCarrier (FBMC) Physical Layer in the WiMAX Context,» *Future Network & Mobile Summit*, pp. 1-8, 2010.
- [80] V. Berg, J. Dore y D. Noguét, «A Multiuser FBMC Receiver Implementation for Asynchronous Frequency Division Multiple Access,» *Proc. of 2014 17th Euromicro Conference on Digital System Design (DSD)*, pp. 16-21, 2014.

-
- [81] V. Ringset, H. Rustad, F. Schaich, J. Vandermot y M. Najar, «Performance of a FilterBank MultiCarrier (FBMC) physical layer in the WiMAX context,» *Future Network & MobileSummit 2010 Conference Proc.*, pp. 1-8, 2010.
- [82] L. Varga y Z. Kollar, «Low complexity FBMC transceiver for FPGA implementation,» *Proc. of 2013 23rd International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 219-223, 2013.
- [83] M. Huermer, L. Reindl, A. Springer y R. Weigel, «Implementation aspects on single carrier transmission with frequency domain equalization,» *Proceedings of the 4th International OFDM-workshop*, vol. 99, pp. 18-19, 1999.
- [84] H. K. Boyapati y R. Kumar, «A comparison of DSP, ASIC and RISC DSP based implementations of multiple access in LTE,» *4th International Symposium on communications Control and Signal Processing (ISCCSP)*, pp. 1-5, 2010.
- [85] G. Martin y H. Chang, «System-on-Chip design,» *Proc. of 4th International Conference on ASIC*, pp. 12-17, 2001.
- [86] S. Brown, R. Francis, J. Rose y Z. Vranesic, «Field Programmable Gate Arrays,» *Kluwer international series in engineering and computer science*, pp. 12-17, 1992.
- [87] Xilinx Inc., «LogiCORE IP AXI DMA,» Product Specification, 2011.
- [88] V. Gil Jiménez, M. Fernández-Getino García, F. González Serrano y A. García Armada, «Design and Implementation of Synchronization and AGC for OFDM-based Wlan Receivers,» *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 1016-1025, 2004.
- [89] A. Oudah, T. Rahman y N. Seman, «Resource Element-Level Computations for Long Term Evolution Networks,» *International Conference on Computer and Communication Engineering*, pp. 904-908, 2012.
- [90] M. Mefenza y C. Bobda, «FPGA implementation of subcarrier index modulation OFDM transceiver,» *Parallel and Distributed Processing Symposium Workshops and PhD Forum (IPDPSW)*, pp. 268-272, 2013.
- [91] V. Berg, J. Doré y D. Noguét, «A flexible radio transceiver for TVWS based on FBMC,» *Microprocessors and Microsystems*, vol. 38, pp. 743-753, 2014.

-
- [92] A. Jiménez-Pacheco, A. Fernández-Herrero y J. Casajús-Quirós, «Design and implementation of a hardware module for MIMO decoding in a 4G wireless receiver,» *VLSI*, vol. 2008, n° 2, p. 10, 2008.
- [93] J. Im, S. Lee y J. Kim, «Design and implementation of a high data rate MICS digital baseband transmitter,» *IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 13-18, 2011.
- [94] S. Saponara, M. Rovini, L. Fanucci, A. Karachalios, G. Lentaris y D. Reisis, «Design and comparison of FFT VLSI architectures for SoC telecom applications with different flexibility, speed and complexity trade-offs,» *Circuits, Systems, and Signal Processing*, vol. 31, n° 2, pp. 627-649, 2012.
- [95] C. Pavithra y R. Ajin, «Efficient frequency band reallocation using uniform transmultiplexers with FPGA implementation,» *Proc. of 2014 International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 1-5, 2014.
- [96] Xilinx Inc., «MicroBlaze Processor Reference Guide,» Product Specification, 2008.
- [97] Xilinx Inc., «7 series FPGAs Memory Interface Solutions User Guide,» Product specification, 2011.
- [98] Xilinx Inc., «AXI Reference Guide,» Product Specification, 2012.
- [99] H. Kavianipour y C. Bohm, «High performance FPGA-based Scatter/Gather DMA Interface for PCIe,» *IEEE Nuclear Science Symposium and Medical Imaging conference record (NSS/MIC)*, pp. 1517-1520, 2012.
- [100] P. Amo-Lopez, M. Domínguez-Jiménez, G. Sansigre, D. Sanz de la Fuente y F. Cruz-Roldán, «Discrete cosine transform Type-IV multicarrier modulators in frequency offset channels,» *Proc. of 19th IEEE International Conference on Electronics Circuits and Systems (ICECS-2012)*, pp. 1-6, 2012.
- [101] F. Cruz-Roldán, M. Dominguez-Jimenez, G. Vidal, J. Pineiro-Ave y M. Blanco-Velasco, «Single-Carrier and Multicarrier Transceivers Based on Discrete Cosine Transform Type-IV,» *IEEE Transactions on Wireless Communications*, vol. 12, n° 12, pp. 6454-6463, 2013.

-
- [102] A. Jain, «A sinusoidal family of unitary transform,» *IEEE Trans. Pattern Analogic Machine intell*, vol. 1, pp. 356-365, 1979.
- [103] M. Narasimha y A. Peterson, «On the Computation of the Discrete Cosine Transform,» *IEEE Transactions on communication*, vol. 26, n° 6, pp. 934-936, 1978.
- [104] T. Tran, J. Liang y C. Tu, «Lapped Transform via Time-Domain Pre- and Post-Filtering,» *IEEE Transactions on signal processing*, vol. 51, n° 6, pp. 1557-1570, 2003.
- [105] S. Chan, T. Ng y C. Kwok, «A class of M-channel linear-phase biorthogonal filter banks and their applications to subband coding,» *IEEE Trans. Signal Processing*, vol. 47, n° 2, pp. 564-571, 1999.
- [106] M. Püschel y J. Moura, «The Discrete Trigonometric transforms and their fast algorithms: an algebraic symmetry perspective,» *Proc. of 2002 IEEE 10th Digital Signal Processing Workshop*, pp. 268-273, 2002.
- [107] N. Ahmed, T. Natarajan y K. Rao, «Discrete cosine transform,» *IEEE Transactions on Computers*, vol. 100, n° 1, pp. 90-93, 1974.
- [108] X. Shao y S. Johnson, «Type-IV DCT, DST, and MDCT algorithms with reduced numbers of arithmetic operations,» *Signal Processing*, vol. 88, n° 6, pp. 1313-1326, 2008.
- [109] Xilinx Inc., «Virtex-6 FPGA DSP48E1 Slice, User Guide,» Febrero 2011.
- [110] Xilinx Inc, «Virtex-6 Family Overview Product Specification,» Enero 2012.
- [111] G. Nordin, P. Milder, J. Hoe y M. Puschel, «Automatic Generation of Customized Discrete Fourier Transform IPs,» *Proc. Design Automation Conference (DAC)*, pp. 471-474, 2005.
- [112] F. Cruz-Roldán, P. Martín, J. Sáez-Landete, M. Blanco y T. Saramäki, «A fast windowing-based technique exploiting spline functions for designing modulated filter banks,» *IEEE Transactions on Circuits and Systems -I: Regular Papers*, vol. 56, n° 1, pp. 168-178, 2009.

-
- [113] P. Banelli, S. Buzzi, G. Colavolpe, A. Modenini, F. Rusek y A. Ugolini, «Modulation Formats and Waveforms for 5G Networks: Who will be the Heir of OFDM?,» *IEEE Signal Processing Magazine*, vol. 31, n° 6, pp. 80-93, 2014.
- [114] R. E. Crochiere y L. R. Rabiner, «Multirate Digital Signal Processing,» *Prentice Hall*, 1983.
- [115] K. W. Martin, «Small side-lobe filter design for multitone data-communication applications,» *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 45, n° 8, pp. 1155-1161, 1998.
- [116] M. Bailey y L. Snyder, «An Empirical Study of On-chip Parallelism,» *25th ACM/IEEE DAC*, pp. 160-165, 1988.
- [117] 4DSP, «FMC204 USER MANUAL,» April 2014.
- [118] Analog Devices, «16-Bit, 200 MSPS/250 MSPS Analog-to-Digital Converter,» December 2013.
- [119] J. Barry, E. Lee y D. Messerschmitt, «Pulse-Amplitud-Modulation,» de *Digital Communication*, Springer, 2004, pp. 131-201.