



UNIVERSIDAD NACIONAL DEL SUR

TESIS PRESENTADA PARA OPTAR AL TÍTULO DE  
DOCTOR EN INGENIERÍA

ARQUITECTURAS DE COMPLEJIDAD REDUCIDA  
PARA LA COMPENSACIÓN ELECTRÓNICA DE LA  
DISPERSIÓN EN SISTEMAS DE COMUNICACIONES  
DE ALTA VELOCIDAD

ARIEL LUIS POLA

BAHÍA BLANCA

ARGENTINA

2016



## Prefacio

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctor en Ingeniería , de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el ámbito del Departamento de Ingeniería Eléctrica y de Computadoras durante el período comprendido entre el 22 de diciembre de 2009 y el 15 de marzo de 2016 , bajo la dirección del Dr. Juan E. Cousseau, y la co-dirección del Dr. Oscar E. Agazzi.



UNIVERSIDAD NACIONAL DEL SUR  
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el ...../...../.....,  
mereciendo la calificación de ..... (.....)



*A Luis, Gladis, Leonardo,  
Maria Laura,  
y Abuelos.*



## Agradecimientos

*Es importante destacar y agradecer el apoyo incondicional de un gran número de personas e instituciones que cedieron su espacio y tiempo para concretar el desarrollo de la presente Tesis.*

*A mi Familia y mi Novia.*

*A mis Amigos.*

*A Mario Hueda, Oscar Agazzi y Juan Cousseau quienes me ayudaron y guiaron desde el primer momento en que inicié las actividades en el doctorado.*

*A la Fundación Tarpuy y la Fundación Fulgor por el apoyo permanente durante todo el trabajo doctoral.*

*A mis compañeros de trabajo y amigos que comparten cada momento en la Fundación Fulgor.*

*A compañeros del Laboratorio de Comunicaciones Digitales de la FCEFyN, UNC.*

*A colegas de ClariPhy Argentina S.A..*

*A todo el grupo GISEE-IIIE-UNS por el acompañamiento y soporte.*

*A aquellas personas que siempre estuvieron desde lo emocional y que nunca me dejaron rendirme.*





# Resumen

Como resultado del constante aumento del tráfico de información, en los últimos años la industria de las telecomunicaciones ha evolucionado de manera vertiginosa. Este hecho exige el diseño de nuevos transceptores de comunicaciones digitales que permitan aumentar la velocidad de procesamiento. Este incremento de velocidad en combinación con las limitaciones del ancho de banda del canal de comunicaciones, exacerbaban los efectos de la interferencia inter-símbolo (*Intersymbol Interference - ISI*). Para compensar este efecto se requiere implementar en el receptor potentes esquemas de *ecualización*. El ecualizador realimentado por decisiones (*Decision Feedback Equalizer - DFE*) representa una de técnicas de ecualización más utilizadas en la industria. El DFE se caracteriza por tener una buena relación entre desempeño y complejidad. Desafortunadamente, su aplicación en sistemas de alta velocidad ha sido limitada debido a la elevada complejidad que aparece cuando se utilizan técnicas de procesamiento en paralelo como resultado de la existencia de lazos realimentados. En particular, la complejidad de las técnicas existentes incrementa *exponencialmente* con la memoria del canal. Esto lleva a restringir el uso de este tipo de ecualizadores para una ISI moderada.

La presente Tesis propone un nuevo esquema de ecualización iterativo de complejidad reducida para receptores de alta velocidad. El nuevo *ecualizador directo asistido por decisiones* (*Decision FeedForward Equalizer - DFFE*) permite obtener un rendimiento similar al DFE pero con una arquitectura paralelizable cuya complejidad aumenta *cuadráticamente* con la memoria del canal. Para canales con gran ISI, esto se traduce en una drástica reducción de la complejidad en comparación con el DFE. La idea central detrás del DFFE, es la iteración de decisiones tentativas para mejorar la precisión de la estimación de la ISI. Para investigar el desempeño del nuevo receptor se desarrolla un estudio teórico y se lo verifica por exhaustivas simulaciones en computadora.

Como una segunda contribución de la Tesis se presenta un detallado análisis de complejidad del procesamiento y además se realiza la implementación en FPGA del DFFE en paralelo. Este estudio permite demostrar los importantes beneficios que tiene utilizar una arquitectura de implementación directa (*forward*) y además verificar experimentalmente el desempeño del DFFE. Todas estas ventajas convierten al DFFE en una excelente opción para receptores de sistemas de comunicaciones digitales de alta velocidad.



# Abstract

As a result of the steady increase in data traffic, the telecommunications industry has evolved dramatically in recent years. In this context, new digital communications transceivers that outperform processing speed are required. This speed increase combined with the limitations of the bandwidth communications channel exacerbate the impacts of the intersymbol interference (*ISI*). In order to compensate for this effect, it is necessary to implement efficient receiver *equalization* schemes. The decision feedback equalizer (*DFE*) is one of the most popular equalization techniques in industry, featuring a good relationship between performance and complexity. Unfortunately, its use in high speed systems has been limited due to the high complexity reached when processing techniques are used in parallel as a result of the existence of feedback loops. In particular, the complexity of the existing techniques increases exponentially with the channel memory, leading to a restriction in the use of such equalizers for moderate ISI.

This Thesis proposes a new scheme of reduced complexity iterative equalization for high-speed receivers. The new *Decision FeedForward Equalizer (DFFE)* allows for a similar performance to the DFE but with a parallelizable architecture whose complexity increases *quadratically* with the channel memory. For channels with large ISI, this results in a drastic reduction in complexity compared to the DFE. The main feature of the DFFE is the iteration of tentative decisions to improve the accuracy of the ISI estimation. With the purpose of investigating the performance of the new receiver, a theoretical study is developed and the DFFE is verified by extensive computer simulations.

The second contribution of the present Thesis is a detailed analysis of processing complexity and the implementation in FPGA of parallel DFFE. This study allows to demonstrate the important benefits of using a forward implementation architecture and verify experimentally the performance of DFFE. All these advantages make the DFFE an excellent choice for system receivers of digital high-speed communications.



# ÍNDICE GENERAL

Índice de Figuras	V
Índice de Tablas	XIII
Publicaciones	XV
Siglas	XIX
<b>1. Introducción</b>	<b>1</b>
1.1. Introducción . . . . .	2
1.1.1. La Necesidad de Incrementar la Velocidad de los Receptores . . . . .	7
1.1.2. La Complejidad de Implementación . . . . .	10
1.2. Estado del Arte . . . . .	12
1.2.1. Segmentación y Paralelismo . . . . .	13
1.2.2. Cancelación Iterativa . . . . .	13
1.2.3. Decisiones Tentativas . . . . .	14
1.2.4. Propuesta . . . . .	14
1.3. Objetivos y Resultados . . . . .	15
1.3.1. Nueva Arquitectura Ecualizadora Iterativa de Baja Complejidad . . . . .	15

---

1.3.2.	Complejidad e Implementación Paralela del DFFE . . . . .	15
1.3.3.	Implementación en FPGA del DFFE . . . . .	16
1.4.	Organización . . . . .	16
<b>2.</b>	<b>Ecualización de Canal</b>	<b>17</b>
2.1.	Introducción . . . . .	18
2.2.	Modelo de Canal . . . . .	20
2.3.	Ecualizador Lineal . . . . .	23
2.3.1.	Criterio de Forzado a Cero . . . . .	25
2.3.2.	Criterio de Error Cuadrático Medio . . . . .	27
2.3.3.	Ecualizador de Espaciamiento Fraccionado . . . . .	28
2.4.	Ecualizador Realimentado por Decisiones . . . . .	28
2.4.1.	Caracterización del DFE . . . . .	29
2.4.2.	Propagación del Error en el DFE . . . . .	31
2.5.	Análisis Teórico de Desempeño . . . . .	35
2.5.1.	Ejemplo . . . . .	38
2.6.	Resultados Numéricos en Canales Exponenciales . . . . .	44
2.7.	Conclusión Parcial . . . . .	51
<b>3.</b>	<b>Arquitecturas de Alta Velocidad de Ecualizadores No Lineales</b>	<b>53</b>
3.1.	Introducción . . . . .	54
3.2.	Diseño Digital de Sistemas Realimentados . . . . .	55
3.2.1.	Análisis de Tiempos . . . . .	55
3.2.2.	Segmentación y Paralelismo . . . . .	58
3.2.3.	Retiming . . . . .	59
3.3.	Arquitectura Paralela Directa del DFE . . . . .	61
3.4.	Técnicas para Implementación en Alta Velocidad de Cancelación No Lineal	65
3.4.1.	Mirar Adelante . . . . .	65
3.4.2.	Arquitectura Concurrente CSA . . . . .	69
3.4.3.	Segmentación de Lazo de Multiplexores Anidados . . . . .	69
3.4.4.	Esquema de Pre-Cálculo Parcial . . . . .	73
3.5.	Comparación de Complejidad . . . . .	76
3.6.	Conclusión Parcial . . . . .	78

---

<b>4. Nueva Arquitectura Ecualizadora Iterativa de Baja Complejidad</b>	<b>79</b>
4.1. Introducción . . . . .	80
4.2. Ecualizador Directo Asistido por Decisiones . . . . .	81
4.2.1. Arquitectura del DFFE . . . . .	82
4.3. Confiabilidad de las Decisiones Tentativas . . . . .	84
4.3.1. Información Mutua . . . . .	84
4.3.2. Estimación de Confiabilidad . . . . .	86
4.3.3. Resultados Numéricos . . . . .	89
4.4. Evaluación de Desempeño . . . . .	91
4.4.1. Ejemplo . . . . .	92
4.4.2. Impacto de la Estimación Incorrecta del Canal . . . . .	97
4.4.3. Resultados Numéricos . . . . .	98
4.5. Desempeño del DFFE en un Canal de Cobre . . . . .	99
4.5.1. Escenario . . . . .	101
4.5.2. Simulador . . . . .	102
4.5.3. Ejemplo . . . . .	103
4.6. DFFE más Algoritmo de Viterbi . . . . .	106
4.6.1. Arquitectura del DFFE-VA . . . . .	108
4.6.2. Decisiones Tentativas Proporcionadas por VA . . . . .	110
4.7. Desempeño del DFFE-VA en un Canal de Fibra Multimodo . . . . .	113
4.8. Conclusión Parcial . . . . .	118
<b>5. Complejidad e Implementación Paralela del DFFE</b>	<b>119</b>
5.1. Introducción . . . . .	120
5.2. Análisis de la Arquitectura DFFE . . . . .	120
5.2.1. Procesamiento en Paralelo . . . . .	120
5.2.2. Complejidad del DFFE . . . . .	122
5.2.3. Comparación con Resultados de Síntesis . . . . .	124
5.2.4. Análisis de Camino Crítico . . . . .	127
5.3. Reducción de Complejidad en el DFFE . . . . .	128
5.3.1. Complejidad del DFFE-VA . . . . .	128
5.4. Conclusión Parcial . . . . .	132

---

<b>6. Implementación en FPGA del DFFE</b>	<b>133</b>
6.1. Introducción . . . . .	134
6.2. Flujo de Diseño . . . . .	135
6.3. Arquitectura Implementada . . . . .	136
6.3.1. Procesador MicroBlaze . . . . .	137
6.3.2. Controlador de Reloj . . . . .	141
6.3.3. Registro de Archivo . . . . .	141
6.3.4. Módulo de Logueo . . . . .	144
6.3.5. Módulo PRBS9 Paralelo . . . . .	145
6.3.6. Módulo de Canal . . . . .	145
6.3.7. Módulo de BER . . . . .	147
6.3.8. Módulo de Ecualizadores . . . . .	148
6.4. Reporte de Síntesis e Implementación . . . . .	149
6.5. Mediciones . . . . .	158
6.5.1. Calibración del Nivel de Ruido . . . . .	158
6.5.2. Desempeño del FFE . . . . .	159
6.5.3. Desempeño del FFE-DFFE . . . . .	160
6.6. Conclusión Parcial . . . . .	165
<b>7. Conclusiones</b>	<b>167</b>
7.1. Discusión Final . . . . .	168
7.2. Trabajo Futuro . . . . .	171
<b>Bibliografía</b>	<b>173</b>



# ÍNDICE DE FIGURAS

1.1. <i>Algunas aplicaciones donde la tecnología VLSI juega un papel crítico en la realización de sistemas de procesamiento de señales [2]. . . . .</i>	3
1.2. <i>Flujo de diseño de sistemas digitales [2]. . . . .</i>	4
1.3. <i>Relación entre el área, retardo, potencia y capacidad de prueba en el diseño digital de un sistema [3]. . . . .</i>	5
1.4. <i>Proyecciones al 2019 expuestas por CISCO [4] en donde se muestra (a) el incremento del flujo de datos en los centros de datos tradicionales y nube (b) el número de usuarios que utilizan las aplicaciones de almacenamiento en la nube. . . . .</i>	8
2.1. <i>Diagrama en bloque de un sistema de comunicaciones digitales típico. . . .</i>	19
2.2. <i>Se muestra una clasificación de las diferentes técnicas de ecualización según el tipo, estructura y algoritmo de adaptación. Los tipos se clasifican en Lineal y No lineal (DFE). Entre los algoritmos de adaptación se destacan el LMS, recursivo de mínimos cuadrados (Recursive Least Squares - RLS), rápido RLS (Fast RLS) y raíz cuadrada RLS [9]. . . . .</i>	20

2.3. Sistema básico de transmisión digital, el cual consiste en un filtro transmisor, canal, ruido aditivo Gaussiano, filtro receptor o filtro apareado (Matched Filter - MF) y muestreador. . . . .	21
2.4. Modelo de canal equivalente en tiempo discreto en banda base. . . . .	23
2.5. Modelo de sistema del ecualizador lineal, el cual incluye el modelo de canal equivalente en tiempo discreto. . . . .	23
2.6. Ecualizador Lineal implementado con un filtro transversal. . . . .	24
2.7. Modelo de sistema del ecualizador realimentado por decisiones (DFE), el cual incluye el modelo de canal equivalente en tiempo discreto. . . . .	29
2.8. Ecualizador realimentado por decisiones (DFE). . . . .	30
2.9. (a) Representación del canal con pre y post cursores antes de ser filtrado por el LE. (b) Canal equivalente después de ser filtrado por el FFF. . . . .	31
2.10. (a) Canal Equivalente en Tiempo Discreto ( $p_m = 0,4\delta_{m+1} + 0,8\delta_m + 0,4\delta_{m-1}$ ). (b) Espectro plegado del canal equivalente . . . . .	39
2.11. Comportamiento del ZF-LE. (a) y (b) Respuesta del LE. (c) y (d) Convolución del canal equivalente con los coeficientes del ecualizador. . . . .	40
2.12. Comportamiento del MMSE-LE. (a) y (b) Respuesta del LE. (c) y (d) Convolución del canal equivalente con los coeficientes del ecualizador. . . . .	41
2.13. Comportamiento del ZF-DFE. (a) y (b) Respuesta del FFF. (c) y (d) Respuesta del FBF. (e) y (f) Respuesta del canal equivalente a la entrada del detector de umbral. . . . .	42
2.14. Comportamiento del MMSE-DFE. (a) y (b) Respuesta del FFF. (c) y (d) Respuesta del FBF. (e) y (f) Respuesta del canal equivalente a la entrada del detector de umbral. . . . .	43
2.15. Curvas de BER para el canal de la Fig. 2.10. Comparación del desempeño del ecualizador LE y DFE para los criterios ZF y MSE. . . . .	44
2.16. (a) Canal Equivalente en Tiempo Discreto ( $p_k = \alpha^k \mu_k$ con $\alpha = 0,6/L = 30$ ). (b) Espectro plegado del canal equivalente. . . . .	45
2.17. Comportamiento del ZF-LE. (a) y (b) Respuesta del LE. (c) y (d) Convolución del canal equivalente con los coeficientes del ecualizador. . . . .	46
2.18. Comportamiento del MMSE-LE. (a) y (b) Respuesta del LE. (c) y (d) Convolución del canal equivalente con los coeficientes del ecualizador. . . . .	47

2.19. Comportamiento del ZF-DFE. (a) y (b) Respuesta del FFF. (c) y (d) Respuesta del FBF. (e) y (f) Respuesta del canal equivalente a la entrada del detector de umbral. . . . .	48
2.20. Comportamiento del MMSE-DFE. (a) y (b) Respuesta del FFF. (c) y (d) Respuesta del FBF. (e) y (f) Respuesta del canal equivalente a la entrada del detector de umbral. . . . .	49
2.21. Curvas de BER para el canal de la Fig. 2.16. Comparación del desempeño del LE y DFE para los criterios ZF y MSE. . . . .	50
2.22. Comparación del desempeño del LE y DFE utilizando el criterio MSE. Se considera un canal exponencial $p_k = \alpha^k \mu_k$ con (a) $\alpha = 0,6/L = 22$ , (b) $\alpha = 0,82/L = 62$ , (c) $\alpha = 0,92/L = 122$ y (d) $\alpha = 0,95/L = 202$ . . . . .	50
3.1. Ejemplo de implementación serial del DFE para tres (3) coeficientes del filtro realimentado. (a) Detalle de los lazos o bucles en el DFE. (b) En línea de trazo se resalta el camino crítico en el diseño. . . . .	56
3.2. Esquemas de transformación de segmentación y paralelismo. (a) Camino de dato. (b) Estructura de segmentación de dos (2) niveles. (c) Estructura en procesamiento en paralelo de dos (2) niveles. . . . .	58
3.3. Ejemplo de reducción del camino crítico utilizando “retiming”. (a) Camino de dato original. (b) “Retiming” de los registros. . . . .	59
3.4. Ejemplo de reducción del camino crítico y reducción de complejidad utilizando “retiming”. (a) Camino de dato original. (b) Se modifica la posición de los registros reduciendo la complejidad. . . . .	60
3.5. Diagrama detallado de la arquitectura del DFE en paralelo para $P = 4$ y $N_b = 3$ . En línea de trazo de color roja se marca uno de los caminos críticos. . . . .	64
3.6. Curvas de frecuencia máxima del reloj y paralelismo en función del número de coeficientes del filtro realimentado del DFE para obtener una velocidad máxima de 40GS/s. La restricción de la frecuencia máxima del reloj es limitada por el camino crítico. . . . .	65
3.7. (a) Estructura tradicional del DFE considerando dos (2) coeficientes en el filtro realimentado. (b) Transformación del DFE aplicando la técnica mirar adelante. . . . .	66

3.8.	<i>Arquitectura paralela del DFE implementando la técnica mirar adelante (look-ahead) para un paralelismo <math>P = 4</math> y <math>N_b = 2</math>. La línea de trazo de color rojo marca el camino crítico de la arquitectura. . . . .</i>	67
3.9.	<i>Esquema de implementación del DFE serial proponiendo la arquitectura CSA [28] . . . . .</i>	69
3.10.	<i>Esquema de implementación del DFE utilizando el criterio de segmentación de lazo de multiplexores anidados. Considera dos coeficientes del filtro realimentado (<math>N_b = 2</math>) y dos niveles de multiplexores. . . . .</i>	71
3.11.	<i>Arquitectura paralela del DFE aplicando la técnica de segmentación de lazo de multiplexores anidados [31] para un paralelismo de <math>P = 4</math>, dos coeficientes (<math>N_b = 2</math>) y cuatro niveles (<math>M = 4</math>). En línea de trazo de color rojo se resalta uno de los caminos críticos de la arquitectura. . . . .</i>	72
3.12.	<i>Esquema de implementación del DFE serial utilizando la técnica de pre-cálculo parcial [27]. . . . .</i>	74
3.13.	<i>Arquitectura paralela del DFE implementando la técnica de pre-cálculo parcial para un paralelismo de <math>P = 4</math> y <math>N_b = 4</math>. En línea de trazo de color rojo se resalta un posible camino crítico del diseño . . . . .</i>	75
3.14.	<i>Número de sumadores, registros y multiplexores 2-a-1 en función del número de coeficientes del filtro realimentado (<math>L</math>) para diferentes arquitecturas del DFE paralelo propuestos en [22, 27, 31, 28]. Factor de paralelismo: <math>P = 16</math>. Formato de modulación: 2-PAM. . . . .</i>	77
4.1.	<i>Ejemplo de un DFFE con tres (3) coeficientes (<math>L = 3</math>) y <math>R = 5</math> iteraciones. Notar que la latencia entre la señal de entrada y la decisión es <math>R - 1</math>. . . .</i>	83
4.2.	<i>Confiabilidad de las decisiones tentativas del DFFE. (a) Información mutua en función de la SNR para <math>\alpha = 0,6</math>, <math>L = 10</math> y <math>R = 11</math>. (b) Información mutua en función del número de iteraciones para diferentes canales con <math>SNR = 15dB</math>. . . . .</i>	90
4.3.	<i>Las gráficas representan la distribución de probabilidad a la entrada del detector de umbral en las cuatro (4) primeras iteraciones del DFFE considerando el canal duo binario para una <math>SNR = 15dB</math>. . . . .</i>	94

4.4. <i>BER en función de la SNR y el número de iteraciones. Canal con post-cursos <math>\alpha = 0,5/L = 6</math>.</i>	99
4.5. <i>Desempeño del DFFE con <math>R = 7</math> y el DFE. Canal con post-cursos <math>\alpha = 0,5/L = 6</math>.</i>	100
4.6. <i>Desempeño del DFE y el DFFE adaptivo con <math>R = L + 1</math> para diferentes canales con ISI post-cursor.</i>	100
4.7. <i>Esquemático general de conexiones entre distintos dispositivos que componen un rack de un centro de datos.</i>	101
4.8. <i>Diagrama en bloques de la arquitectura del simulador utilizado para determinar el desempeño de los ecualizadores. El lenguaje de programación en el cual se desarrolló el simulador fue C++.</i>	103
4.9. <i>Respuesta al impulso de un canal de cobre muestreado a una frecuencia de 32GHz.</i>	104
4.10. <i>Respuesta al impulso del canal equivalente después de ser compensado parcialmente por el ecualizador lineal.</i>	104
4.11. <i>Confiabilidad de las decisiones tentativas del DFFE para un canal de cobre. (a) Información mutua en función de la SNR para tres (3) iteraciones diferentes. (b) Información mutua en función del número de iteración para una SNR = 15dB.</i>	105
4.12. <i>Curvas de desempeño (BER) del DFFE comparado con el DFE considerando un canal de cobre detallado en la Fig. 4.9. (a) Comparativa de curvas de desempeño considerando diferente número de iteraciones (<math>R = [20, 30, 40, 50, 60]</math>). (b) Curvas de desempeño del DFE y DFFE considerando <math>R = 60</math> iteraciones.</i>	107
4.13. <i>Ejemplo de tres (3) coeficientes DFFE-VA con <math>L = 3</math>, <math>L_v = 1</math>, <math>R_v = 1</math>, <math>R = 5</math> y <math>N = 2</math>.</i>	108
4.14. <i>BER en función del número de iteraciones que utilizan decisiones tentativas proporcionadas por el VA (<math>R_v</math>).</i>	111
4.15. <i>Curvas de BER en función de la SNR para el DFFE (<math>R = 30</math> y <math>60</math>), VA (<math>N = 4</math>), y DFFE-VA (<math>R = 30</math>, <math>R_v = 3</math>, <math>N = 4</math>).</i>	112
4.16. <i>Respuesta al impulso de un canal de fibra multimodo muestreado a 10GHz</i>	114

4.17. Respuesta la impulso del canal equivalente después de ser compensado parcialmente por el ecualizador lineal . . . . .	114
4.18. Confiabilidad de las decisiones tentativas del DFFE para un canal de fibra multi-modo. (a) Información mutua en función de la SNR para tres (3) iteraciones diferentes. (b) Información mutua en función del número de iteración para una SNR = 14dB. . . . .	115
4.19. Curvas de desempeño (BER) del DFFE comparado con el DFE considerando un canal MMF detallado en la Fig. 4.16. (a) Comparativa de curvas de desempeño considerando diferente número de iteraciones ( $R = [20, 30, 40, 50, 60]$ ). (b) Curvas de desempeño del DFE y DFFE considerando $R = 60$ iteraciones. . . . .	116
4.20. Desempeño del DFFE en función del número de iteraciones que utilizan decisiones tentativas provenientes del algoritmo de Viterbi ( $R_v$ ) . . . . .	116
4.21. Curvas de desempeño del DFE, VA con $N = 4$ , DFFE con $R = [20, 60]$ y DFFE-VA con $N = 4$ , $R = [20, 60]$ y $R_v = 4$ . . . . .	117
5.1. Arquitectura paralela del DFFE para $P = 4$ , $L = 3$ y $R = 5$ . Los bloques DFFEn son detallados en la Fig. 5.2. . . . .	121
5.2. Ejemplo de un DFFE con tres (3) coeficientes ( $L = 3$ ) y $R = 5$ iteraciones. a) Notar que la latencia entre la señal de entrada y la decisión es $R - 1$ . b) La línea de trazo de color roja denota el camino crítico . . . . .	121
5.3. Número de sumadores, registros y multiplexores 2-a-1 en función del número de coeficientes del filtro realimentado $L$ , para el DFFE paralelo con $R = L + 1$ y las arquitecturas DFE propuestas en [22, 27, 31] y [28]. Factor de paralelización: $P = 16$ . Formato de modulación: 2-PAM . . . . .	124
5.4. Curva BER en función de la SNR para le DFFE con $L = 30$ , $R = 31$ , $N_i = 7/8$ bits, y $N_c = 7$ bits. . . . .	126
5.5. Arquitectura paralela del DFFE y SBVD para $N = 2$ , $L = 3$ y $P = 4$ . Los bloques DFFEn y SBVD son mostrados en la Fig. 4.13 y [48], respectivamente. 129	129

5.6. <i>Número de componentes básicos para diferentes tipos de ecualizadores, donde el número de coeficientes del filtro realimentado es <math>L = 10</math> y paralelismo <math>P = 16</math>. El número de iteraciones del DFFE es <math>R = 60</math> y DFFE-VA utiliza <math>R = 30</math>, <math>R_v = 3</math> y <math>N = 4</math> estados del VA. . . . .</i>	131
6.1. <i>Diagrama en bloques de la arquitectura implementada en la FPGA. . . . .</i>	136
6.2. <i>Diagrama en bloques del procesador MicroBlaze. Se detallan los módulos que son empleados en el presente diseño. . . . .</i>	138
6.3. <i>Estructura de la trama utilizada para la comunicación entre la computadora y la FPGA transferida por la interfaz UART. . . . .</i>	139
6.4. <i>Diagrama de flujo del logueo de variables donde se observa como opera cada una de las interfaces de cada módulo. . . . .</i>	141
6.5. <i>Descripción de las señales que componen el RF. . . . .</i>	142
6.6. <i>Detalle del GPO utilizado para la carga de comandos en el RF. . . . .</i>	142
6.7. <i>Configuraciones de las memorias de logueo de variables. (a) Configuración utilizada en el logueo de la señal de salida del canal con ruido, salida del FFE y el error de detección. (b) Configuración utilizada en el logueo de los coeficientes de los ecualizadores. . . . .</i>	144
6.8. <i>Esquemático del módulo PRBS9 paralelo, detallando las operaciones lógicas XORs utilizadas en la actualización de los registros. . . . .</i>	145
6.9. <i>Diagrama en bloques del banco de filtros y generadores de ruido Gaussianos utilizados para generar los efectos de dispersión y ruido en el canal. . . . .</i>	146
6.10. <i>Diagrama en bloques del módulo de BER. . . . .</i>	147
6.11. <i>Diagrama en bloques de los ecualizadores. . . . .</i>	148
6.12. <i>Comparación de complejidad de las diferentes jerarquías que integran el diseño implementado en base al número de celdas detalladas en la Tabla 6.2. (a) Nivel superior (top-level). (b) Nivel de integración de módulos (top-phy). (c) Nivel de ecualización (top-rx). . . . .</i>	151
6.13. (a) <i>Histograma de tiempo de holgura de configuración (setup). (b). Histograma de tiempo de holgura de retención (hold). . . . .</i>	154
6.14. <i>Resultado de la implementación de la arquitectura propuesta detallando el conexionado de las señales. . . . .</i>	156

6.15. Resultado de la implementación de la arquitectura propuesta solamente considerando las instancias de los bloques lógicos de la FPGA. El color rojo representa el DFFE paralelo incluyendo la etapa de adaptación de coeficientes, el color verde remarca el FFE y el algoritmo de adaptación, el color amarillo identifica el procesador MicroBlaze y el color celeste el resto de los módulos. . . . .	157
6.16. Comportamiento de la señal a la entrada de la etapa de ecualización. (a) Histograma de la entrada del detector de umbral de la última iteración del DFFE. (b) Curva de BER del canal Gaussiano. . . . .	158
6.17. (a) Respuesta al impulso del canal ( $p_k = \alpha^k \mu_k$ con $\alpha = 0,6/L = 15$ ). (b) Espectro plegado del canal equivalente. . . . .	159
6.18. Comportamiento del FFE. (a) y (b) Respuesta del FFE. (c) y (d) Respuesta del canal equivalente en la entrada del detector de umbral. . . . .	160
6.19. Comportamiento del FFE. (a) Histograma de la señal de entrada al FFE. (b) Señal de error estimada entre el símbolo detectado y la señal de entrada al detector de umbral. (c) Comportamiento temporal de los coeficientes del FFE. (d) Desempeño del FFE. . . . .	161
6.20. Comportamiento de los ecualizadores. (a) y (b) Respuesta del FFE. (c) y (d) Respuesta del DFFE. (e) y (f) Respuesta del canal equivalente en la entrada del detector de umbral. . . . .	162
6.21. Comportamiento de los ecualizadores. (a) Histograma de la señal de entrada del FFE. (b) Error de estimación. (c) y (d) Comportamiento temporal de los coeficientes del FFE y DFFE. (e) Desempeño del FFE-DFFE. . . . .	163
6.22. Curvas de BER para el canal de la Fig. 6.17. Comparación del desempeño entre el FFE y FFE-DFFE. . . . .	164



# ÍNDICE DE TABLAS

3.1. <i>Complejidad del DFE paralelo implementando la técnica mirar adelante [22].</i>	68
3.2. <i>Complejidad del DFE paralelo implementando la arquitectura CSA [28]. . .</i>	70
3.3. <i>Complejidad del DFE paralelo implementando la técnica de segmentación de lazo de multiplexores anidados [31]. . . . .</i>	73
3.4. <i>Complejidad del DFE paralelo implementando la técnica de pre-cálculo parcial [27]. . . . .</i>	75
3.5. <i>Comparación de complejidad entre diferentes arquitecturas paralelas del DFE para una modulación 2-PAM y un canal con gran memoria (<math>L \gg 1</math>).</i>	76
4.1. <i>Canal utilizado para verificar el funcionamiento del DFFE-VA de la Fig. 4.15 . . . . .</i>	111
5.1. <i>Complejidad de la arquitectura del DFFE paralelo para 2-PAM y <math>R &gt; L</math>. .</i>	122
5.2. <i>Comparación de complejidad entre el DFFE paralelo y arquitecturas DFE para 2-PAM con <math>R = L + 1</math> para <math>L \gg 1</math> . . . . .</i>	123
5.3. <i>Comparación de complejidad entre el DFFE paralelo y arquitecturas DFE para M-PAM con <math>R = L + 1</math> para <math>L \gg 1</math> . . . . .</i>	123
5.4. <i>Resultados de síntesis para la arquitectura del DFFE paralelo para 2-PAM y <math>R = L + 1</math> en Tecnología CMOS 28nm . . . . .</i>	126

## Índice de Tablas

---

5.5. Complejidad de la arquitectura paralela del SBVD para 2-PAM [48] . . . . .	130
5.6. Complejidad de arquitecturas paralelas del DFE para 2-PAM . . . . .	130
6.1. <i>Detalle de comandos del RF.</i> . . . . .	143
6.2. <i>Reporte de síntesis donde se detalla el número de celdas discriminado por jerarquía.</i> . . . . .	150
6.3. <i>Reporte de síntesis donde se detalla el número de “Slice” utilizados en el diseño considerando la etapa de implementación.</i> . . . . .	152
6.4. <i>Reporte de síntesis donde se detalla el número de LUTs utilizados en el diseño considerando la etapa de implementación.</i> . . . . .	153
6.5. <i>Reporte de síntesis donde se detalla el número de BRAM y DPS utilizados en el diseño.</i> . . . . .	153

# PUBLICACIONES

### Conferencias

**2011** A. L. Pola, D. E. Crivelli, J. E. Cousseau, O. E. Agazzi, and M. R. Hueda, “A new low complexity iterative equalization architecture for high-speed receivers on highly dispersive channels: Decision feedforward equalizer (DFFE)”, in *2011 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 133–136, IEEE, May 2011.

Disponible en: <http://dx.doi.org/10.1109/ISCAS.2011.5937519>

**2013** A. L. Pola, J. E. Cousseau, O. E. Agazzi, and M. R. Hueda, “Efficient decision feedforward equalizer with parallelizable architecture”, in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2771–2774, 2013.

Disponible en: <http://dx.doi.org/10.1109/ISCAS.2013.6572453>

---

## Revista

**2013** A. L. Pola, J. E. Cousseau, O. E. Agazzi, and M. R. Hueda, “A Low-Complexity Decision Feedforward Equalizer Architecture for High-Speed Receivers on Highly Dispersive Channels”, *Journal of Electrical and Computer Engineering*, vol. 2013, May 2013.

Disponibile en: <http://dx.doi.org/10.1155/2013/587108>

### Citas

#### ■ En publicaciones:

- 2012** Adam Healey and Chad Morgan, “A comparison of 25 gbps NRZ & PAM-4 modulation used in legacy & premium backplane channels”, DesignCom 2012, 2012.
- 2012** Adam Healey and Chad Morgan, “A comparison of 25 gbps NRZ & PAM-4 modulation used in legacy & premium backplane channels”, White paper, 2012.
- 2013** K. Wu and P. Liu, “A novel signaling technique for high-speed backplane transceiver,” in *ISOCC 2013 Conference*, pp. 148–151, 2013.
- 2014** K. Wu, P. Liu, and Q. Yu, “A novel signaling technique for high-speed wireline backplane transceiver: Four phase-shifted sinusoid symbol (PSS-4),” in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2141–2144, June 2014.

#### ■ En patentes:

- 2013** E. V. Chmelar, “Adaptation using error signature analysis in a communication system,” Dec. 24 2013. US Patent 8.615.062.
- 2014** C. Palusa, T. Prokop, A. B. Healey, and Y. Liu, “Decision feedforward equalization,” July 22 2014. US Patent 8.787.439.
- 2015** E. V. Chmelar, “Error signature analysis for data and clock recovery in a communication system,” Apr. 21 2015. US Patent 9.014.313.

# SIGLAS

Sigla	Inglés	Español
ASIC	Application Specific Integrated Circuit	Circuito Integrado de Aplicación Específica
AWGN	Additive White Gaussian Noise	Ruido Aditivo Blanco Gaussiano
BER	Bit Error Rate	Tasa de Error de Bit
BRAM	Block RAM	Bloque RAM
CAGR	Compound Annual Growth Rate	Tasa de Crecimiento Anual Compuesta
CCSA	Concurrent Carry Save Adder	Sumador Acarrear-Guardar Concurrente
CD	Chromatic Dispersion	Dispersión Cromática
CLB	Configurable Logic Block	Bloque Lógico Configurable
CMOS	Complementary Metal Oxide Semiconductor	Semiconductor Complementario de Óxido Metálico
CP	Critical Path	Camino Crítico
CPU	Central Processing Unit	Unidad de Procesamiento Central

*Continúa en la página siguiente*

## Siglas

Sigla	Inglés	Español
CR	Carrier Recovery	Sincronismo de Portadora
CSA	Carry Save Adder	Sumador Acarrear-Guardar
DCT	Discrete Cosine Transform	Transformada Discreta del Coseno
DFE	Decision Feedback Equalizer	Ecualizador Realimentado por Decisión
DFFE	Decision FeedForward Equalizer	Ecualizador Directo Asistido por Decisiones
FBF	FeedBack Filter	Filtro Realimentado
FEXT	Far-End Crosstalk	Diafonía de Extremo
FFF	Feed Forward Filter	Filtro Directo
FFT	Fast Fourier Transform	Transformada Rápida de Fourier
FIR	Finite Impulse Response	Respuesta al Impulso Finita
FPGA	Field Programmable Gate Array	Arreglo de Compuertas Lógicas Programables
FSK	Frequency Shift Keying	Desplazamiento de Frecuencia
FSM	Finite State Machine	Máquina de Estado Finita
GNG	Gaussian Noise Generator	Generador de Ruido Gaussiano
GPIO	General Purpose Input/Output	Entrada/Salida de Propósito General
HDL	Hardware Description Language	Lenguaje de Descripción de Hardware
IB	Iteration Bound	Límite de Iteración
IC	Integrated Circuit	Circuito Integrado
IIR	Infinite Impulse Response	Respuesta al Impulso Infinita
IoT	Internet of Things	Internet de las cosas
IP	Internet Protocol	Protocolo de Internet
ISI	Intersymbol Interference	Interferencia Inter-Símbolo
IT	Information Technology	Tecnología de Información

*Continúa en la página siguiente*



<b>Sigla</b>	<b>Inglés</b>	<b>Español</b>
LB	Loop Bound	Límite de Lazo
LE	Lineal Equalizer	Ecualizador Lineal
LMS	Least Mean Square	Mínimo Cuadrado Medio
LSB	Least Significant Bit	Bit Menos Significativo
LTI	Linear Time Invariant	Lineal Invariante en el Tiempo
LUT	Look-Up Tables	Tabla de Funciones
MDM	MicroBlaze Debug Module	Módulo de Depuración MicroBlaze
MF	Matched Filter	Filtro Apareado
MIMO	Multiple-Input Multiple-Output	Múltiple-Entrada Múltiple-Salida
MLSD	Maximum Likelihood Sequence Detector	Detector de Secuencia de Máxima Verosimilitud
MMCM	Mixed-Mode Clock Manager	Mestor de Reloj de Modo Mixto
MMF	Multi Mode Fiber	Fibra Multi Modo
MMSE	Minimum Mean Squared Error	Minimización del Error Cuadrático Medio
MSB	Most Significant Bit	Bit Más Significativo
MSE	Mean Squared Error	Error Cuadrático Medio
PAM	Pulse Amplitude Modulation	Modulación por Amplitud de Pulso
PCB	Printed Circuit Board	Placa de Circuito Impreso
PDF	Probability Density Function	Función de Densidad de Probabilidad
PMD	Polarization Mode Dispersion	Dispersión por Modo de Polarización
PRBS	Pseudo-Random Binary Sequence	Secuencia Binaria Pseudo-Aleatoria
PSK	Phase Shift Keying	Desplazamiento de Fase
QAM	Quadrature Amplitude Modulation	Modulación de Amplitud en Cuadratura
RF	Register File	Registro de Archivos

*Continúa en la página siguiente*

## Siglas

Sigla	Inglés	Español
RISC	Reduced Instruction Set Computer	Conjunto de Instrucciones Reducidas de Computadora
RLS	Recursive Least Squares	Recursivo de Mínimos Cuadrados
RTL	Register Transfer Level	Nivel de Transferencia de Registro
SBVD	Sliding Block Viterbi Decoder	Decodificador de Viterbi de Bloque Deslizante
SFP	Small Form-factor Pluggable	Factor de Forma Pequeño Conectable
SISO	Single-Input Single-Output	Simple-Entrada Simple-Salida
SNR	Signal to Noise Ratio	Relación Señal a Ruido
SVT	Standard Voltage Threshold	Umbral de Voltaje Estándar
ToR	Top of Rack	Superior del Rack
TR	Timing Recovery	Sincronismo de Símbolo
UART	Universal Asynchronous Receiver-Transmitter	Transmisor/Receptor Asincrónico Universal
UTP	Unshielded twisted pair	Par Trenzado sin Blindaje
VA	Viterbi Algotihm	Algoritmo de Viterbi
VLSI	Very Large Scale Integration	Gran Escala de Integración
ZF	Zero Forcing	Forzado a Cero
ZF-LE	Zero Forcing-Lineal Equalizer	Ecualizador Lineal de Forzado a Cero

# CAPÍTULO

## 1

# INTRODUCCIÓN

***Síntesis:** En el presente capítulo se introducen los diferentes aspectos que han motivado el desarrollo de esta Tesis, haciendo énfasis en la dificultad que presenta el diseño de sistemas de comunicaciones digitales de alta velocidad. Luego, se revisa el Estado del Arte en donde es posible tomar contacto con las problemáticas actuales del área de desarrollo de este trabajo. Finalmente, se presentan los objetivos y las contribuciones realizadas a lo largo de esta Tesis.*

### 1.1. Introducción

La aparición del transistor en 1947 en los Laboratorios BELL, permitió un incremento vertiginoso en los avances de los sistemas electrónicos, logrando el diseño de circuitos integrados (*Integrated Circuit - IC*) extremadamente compactos. Esto abrió las puertas para el desarrollo de circuitos de gran escala de integración (*Very Large Scale Integration - VLSI*) jugando un papel fundamental sobre los sistemas con los que convivimos hoy en día. En 1975, uno de los científicos más reconocidos del siglo XX, Gordon Moore, predijo que el número de transistores por unidad de superficie en circuitos integrados se duplicaría cada 24 meses [1]. Sorpresivamente esto se sigue cumpliendo y un ejemplo de esta evolución son las unidades de procesamiento central (*Central Processing Unit - CPU*). Como consecuencia de estos avances, han surgido innumerables aplicaciones en donde los ingenieros y científicos han podido probar su capacidad de invención produciendo nuevas áreas de desarrollo. Algunas de las aplicaciones se resumen en la Fig. 1.1, de las cuales podemos resaltar sistemas multimedia, telefonía móvil, redes de datos, computadoras portátiles, procesamiento de imágenes, sistemas de comunicaciones digitales, aplicaciones satelitales, entre otras. Un factor común que posee la mayoría de las aplicaciones es el procesamiento de señales en tiempo real, el cual exige aún más el ingenio de los investigadores.

Tradicionalmente, el diseño de los algoritmos para estas aplicaciones y su implementación en VLSI siguen un proceso de evolución paralela cuyos caminos se ven influenciados por diferentes actores. Mientras los diseñadores de algoritmos tienden a enfatizar la mejora del rendimiento, los diseñadores de hardware <sup>1</sup> buscan flexibilidad en la implementación de acuerdo a la tecnología empleada. Estos objetivos complementarios resultan en un largo proceso iterativo entre ambos equipos de diseño para poder converger a una arquitectura implementable y que cumpla con los requisitos de desempeño. Un ejemplo del flujo de desarrollo de la implementación de una aplicación de un sistema de comunicaciones digitales se muestra en la Fig. 1.2. Una aplicación completa de procesamiento de señales se suele dividir en el diseño de software y hardware. El primer paso del flujo de diseño es la definición de los requisitos y especificaciones, que por lo general son velocidad de muestreo, una medida cuantitativa de rendimiento en presencia de ruido, máxima velocidad de

---

<sup>1</sup>Entiéndase como diseñador de hardware a la persona que implementa el algoritmo utilizando algún lenguaje de descripción de hardware (*Hardware Description Language - HDL*).

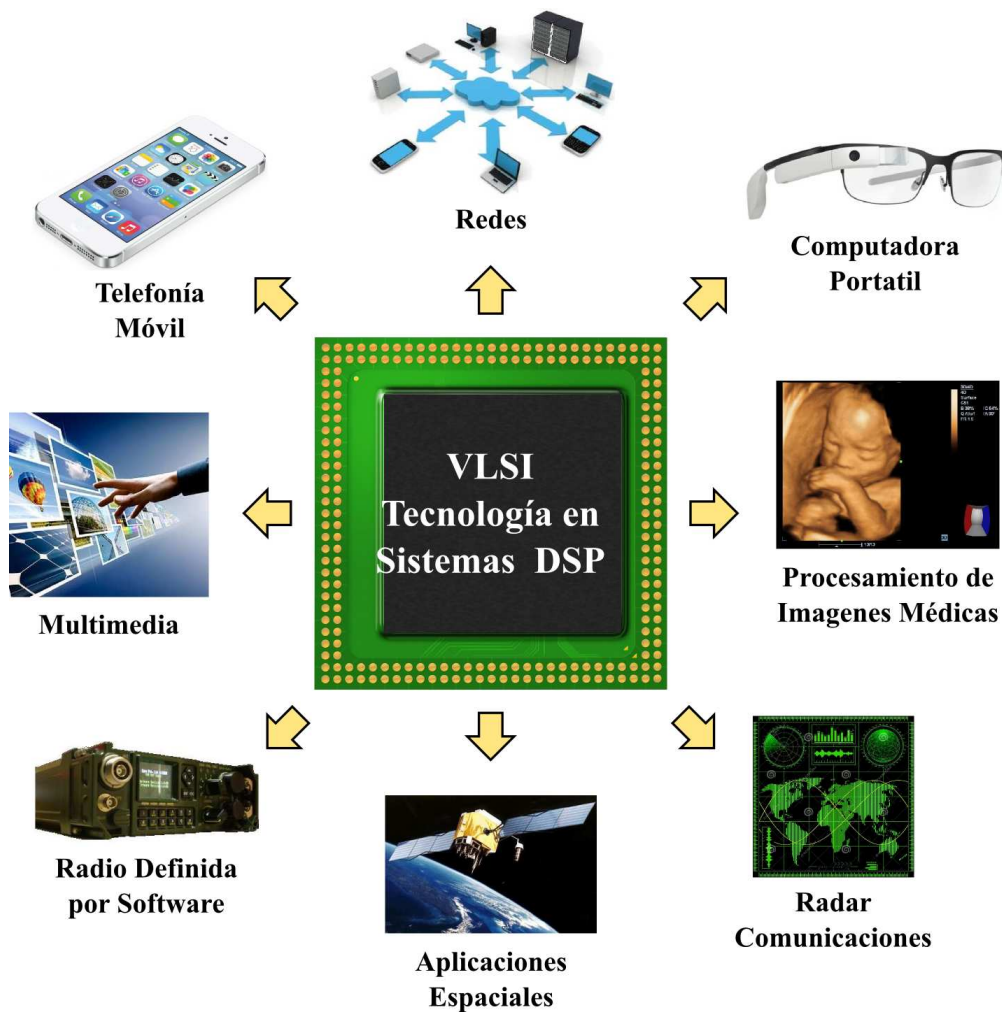


Figura 1.1: Algunas aplicaciones donde la tecnología VLSI juega un papel crítico en la realización de sistemas de procesamiento de señales [2].

datos soportada en bits por segundo (*bits per second - bps*) en el transmisor, la tasa de error de bit (*Bit Error Rate - BER*) permitida, el ancho de banda del canal, las frecuencias de portadora e intermedias, los límites de potencia, las interfaces, etc. Los desarrolladores de algoritmos toman estos requisitos y exploran las técnicas de comunicaciones digitales que pueden satisfacer las especificaciones indicadas. Estos algoritmos son generalmente codificados con las herramientas de modelado de comportamiento empleando funciones (por ejemplo, MATLAB o C++). Un paso crítico es la partición de la aplicación en hardware y software. La división es impulsada por factores como el rendimiento, costo, la disipación de energía, tiempo en el mercado, etc. Esta separación deriva en el desarrollo del código en punto fijo. Para la parte de software, el diseñador toma el código de punto flotante que está diseñado en el modelo inicial y lo convierte en formato de punto fijo.

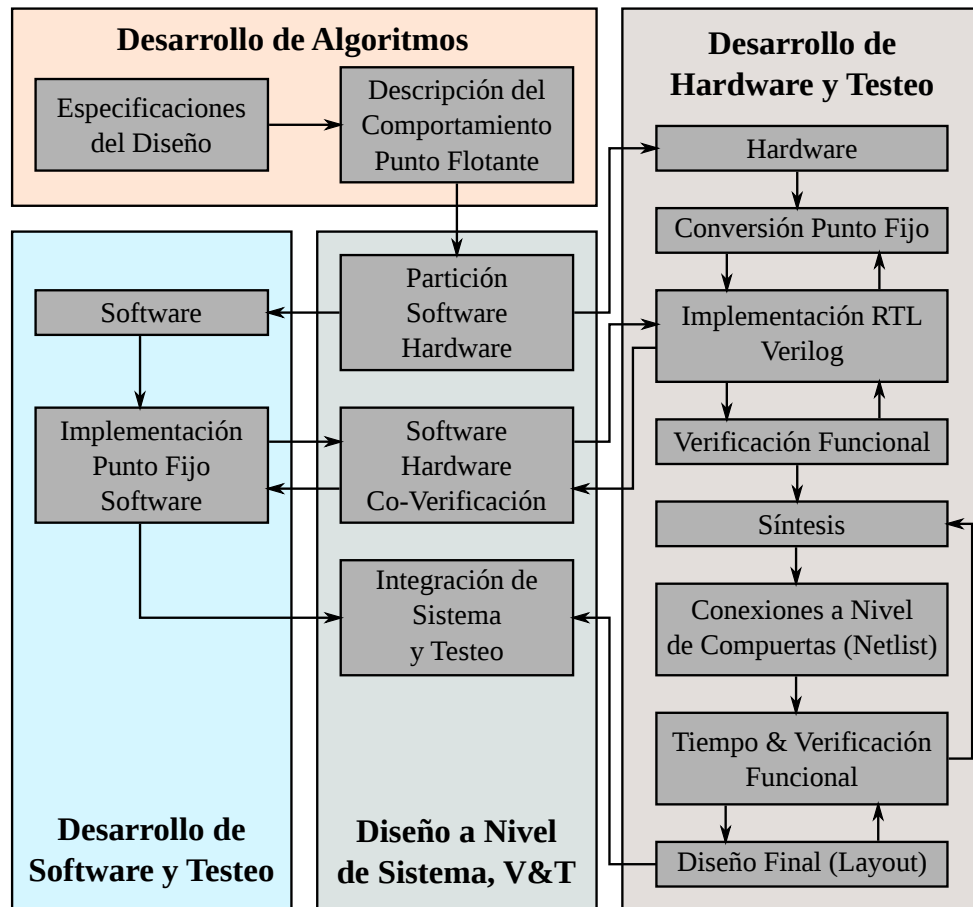


Figura 1.2: Flujo de diseño de sistemas digitales [2].

Es decir, cambia todas las variables y constantes anteriormente diseñados con formato de doble o simple precisión a variables de punto fijo estándar del tipo “char” (8 bits), “short” (16 bits) e “int” (32 bits), respectivamente. En comparación con la conversión de punto fijo para el mapeo en un software, la conversión a punto fijo para el mapeo de hardware requiere más deliberación. Esto es porque cada bit adicional en hardware implica recursos adicionales, y la búsqueda del tamaño óptimo de todas las variables requiere varias iteraciones con diferentes precisiones de configuración de las variables. Después de convertir el código al formato de punto fijo requerido, siempre se recomienda verificar por medio de simulaciones ambas partes y caracterizar el algoritmo para diferentes escenarios con ruido. Luego, el diseñador de hardware explora diferentes arquitecturas con el objetivo de buscar la implementación óptima y se codifica en un lenguaje de descripción de hardware (*Hardware Description Language - HDL*) (por ejemplo, Verilog o VHDL). El siguiente paso es verificar que la implementación siga cumpliendo con los objetivos del

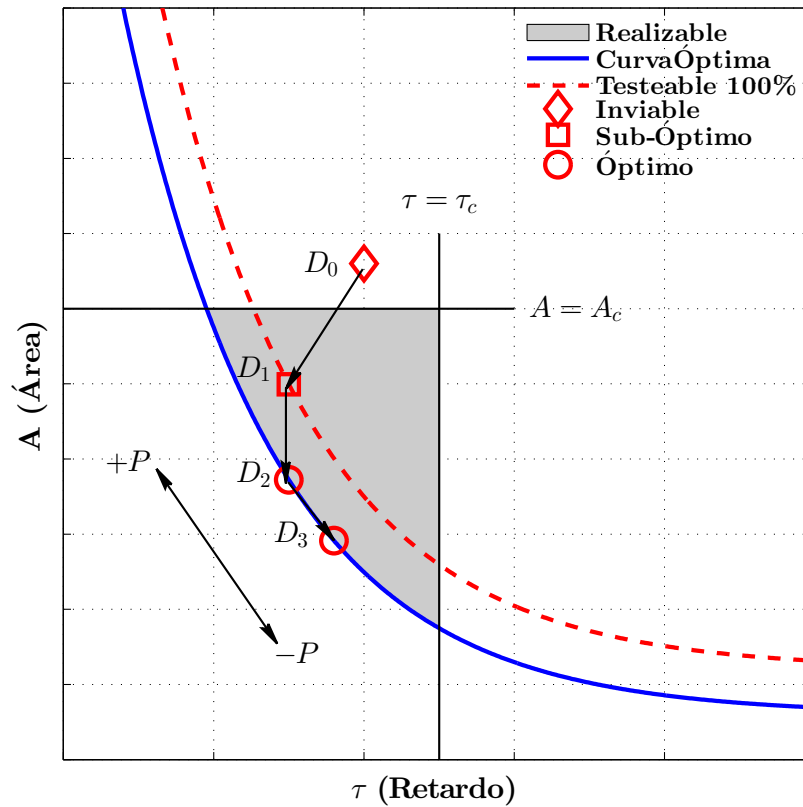


Figura 1.3: Relación entre el área, retardo, potencia y capacidad de prueba en el diseño digital de un sistema [3].

algoritmo, esto se denomina apareamiento de vectores (*vector matching*). Por lo general, se emplea el mismo simulador de punto fijo. El código RTL (*Register Transfer Level - RTL*) verificado se sintetiza a continuación, en una tecnología de destino mediante una herramienta de síntesis. Si la tecnología es un arreglo de compuertas lógicas programable (*Field Programmable Gate Array - FPGA*), la herramienta de síntesis también genera la disposición final. En el caso de un circuito integrado de aplicación específica (*Application Specific Integrated Circuit - ASIC*), se genera una lista de conexiones a nivel de compuertas (*gates*). El diseñador puede realizar la verificación de tiempo (*timing*) y luego utilizar las herramientas para el diseño final (*layout*). El último paso es la integración del software y el hardware en la aplicación final.

Haciendo énfasis en el flujo de diseño del hardware, el diseñador debe considerar los siguientes objetivos críticos [2]

- Área ocupada por compuertas lógicas (*logic gates*) e interconexiones.

- El retardo del camino crítico (*critical path*) de la ruta más larga a través de la lógica.
- El grado de capacidad de prueba (*testability*) del circuito, medida en términos del porcentaje de las fallas cubiertas por un conjunto específico de vectores de prueba, para un modelo de falla apropiado.
- Disipación de potencia de las compuertas lógicas.

El arte del diseño digital es encontrar el equilibrio óptimo entre estos cuatro pilares. Para poder entender la dificultad de equilibrar estos objetivos se plantea un caso típico de diseño, el cual se ejemplifica en la Fig. 1.3 [3]. En cualquier diseño digital se definen los límites de área ( $A_c$ ) y retardos (*delay* -  $\tau_c$ ) máximos según los requerimientos, lo cual demarca un área o región. Cualquier punto dentro de esta zona genera un circuito realizable. Normalmente, el diseño comienza como un diseño no realizable ( $D_0$ ) fuera de los límites establecidos, que está muy alejado de la curva óptima (línea continua). El diseño inicial puede mejorarse rediseñando ciertos parámetros para mejorar tanto área como retardo, donde  $D_1$  es realizable pero sub-óptimo mientras que  $D_2$  y  $D_3$  son realizables y óptimos. Cualquier movimiento sobre esta área afecta directamente la potencia. Por ejemplo, si nos movemos hacia la izquierda por la curva óptima (mayor área y menor retardo) incrementamos la potencia, ya que se asocia con un incremento en la complejidad. En dirección contraria reducimos la potencia, ya que lo asociamos con una reducción en la velocidad del sistema. Por último, tales diseños tiene su propia curva de capacidad de prueba, generalmente más alta que la curva óptima del diseño, como se ilustra en la Fig. 1.3 por la línea de trazo. Diseños sobre esta curva es probable que tengan cierto grado de redundancia en los circuitos implementados [3].

Todo lo mencionado, es generalmente cierto cuando los requerimientos de velocidad, potencia y área se encuentran dentro de las capacidades de la tecnología. Esto permite desacoplar o minimizar la interacción a nivel de sistema y el diseño VLSI y concentrarnos en optimizar los aspectos críticos del diseño. Sin embargo, si los requerimientos sobrepasan los límites permitidos por la tecnología, es necesario evaluar nuevamente los algoritmos haciendo que el ciclo de diseño vuelva a comenzar. En síntesis, esto nos lleva a que los diseñadores de algoritmos tengan que abordar los problemas teniendo en cuenta aspectos de implementación críticos que pueden definir la eficiencia y desempeño del diseño final del sistema.



### 1.1.1. La Necesidad de Incrementar la Velocidad de los Receptores

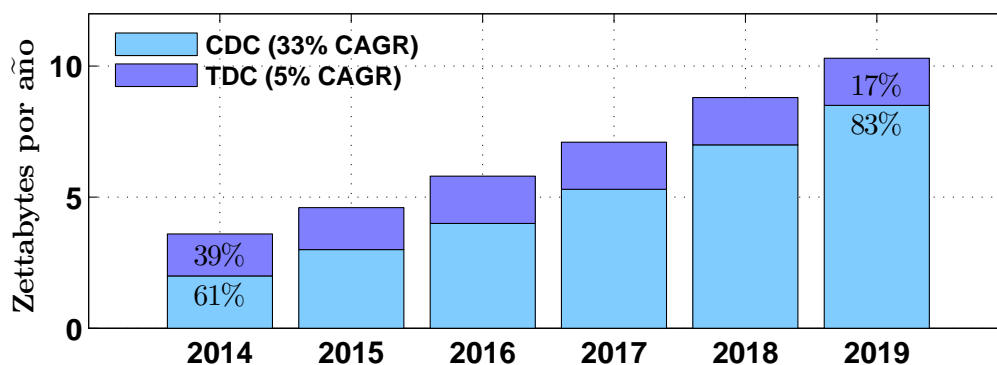
En los últimos años, la industria de las telecomunicaciones ha visto evolucionar el concepto de procesamiento en la nube, de una tecnología emergente a una solución de red fuertemente establecida que está ganando aceptación y despliegue generalizado. Un claro ejemplo de esto son las empresas y organizaciones gubernamentales que están pasando gran parte de su carga de trabajo crítico a la nube. Para los consumidores, los servicios en la nube ofrecen un acceso simple a los contenidos y servicios, en varios dispositivos, entregados en casi cualquier lugar en donde el usuario tenga acceso a la red. Las principales motivaciones cualitativas para la adopción de la nube incluyen una entrega más rápida de servicios y datos, el aumento de rendimiento de las aplicaciones y la mejora en la eficiencia operativa. Uno de los aspectos que preocupa a muchos usuarios que utilizan aplicaciones basadas en la nube es la seguridad y la integración con los actuales entornos de tecnología de información (*Information Technology - IT*). Sin embargo, este punto de incertidumbre no ha detenido el continuo incremento en la gama de servicios de consumo y negocios que hoy están disponibles. Las perspectivas de los servicios en la nube varían según las necesidades del cliente (por ejemplo, la privacidad, la movilidad y el acceso de múltiples dispositivos) y de apoyo a las oportunidades a corto plazo, así como las prioridades estratégicas a largo plazo para los operadores de redes, tanto públicas como privadas.

Cuantitativamente, el impacto del cómputo en la nube en el tráfico del centro de datos (*data center*) es claro. La mayor parte del tráfico de Internet se ha originado o terminado en un centro de datos desde el año 2008, cuando el tráfico de redes de pares (*peer-to-peer*<sup>2</sup>) dejó de dominar las aplicaciones de Internet. El tráfico del centro de datos continuará dominando el tráfico de Internet en el futuro previsible, pero la naturaleza del tráfico en estos lugares está experimentando una transformación fundamental provocada por aplicaciones en la nube, los servicios y la infraestructura.

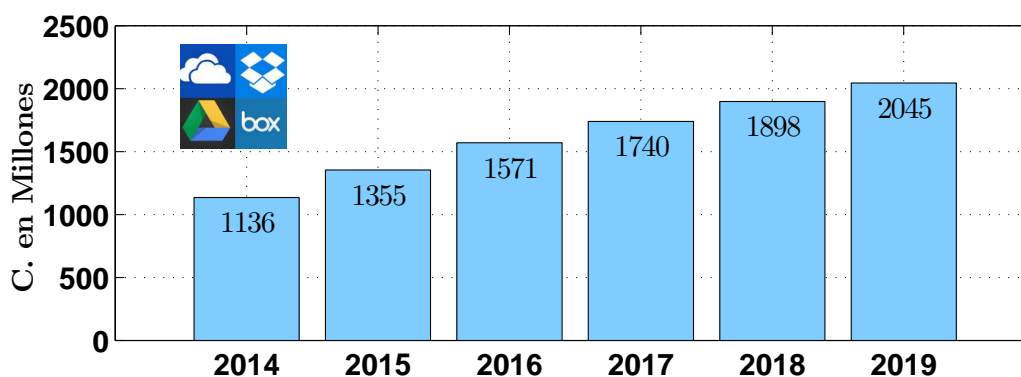
La importancia de la evolución global de la nube se destaca en las proyecciones de tráfi-

---

<sup>2</sup>Una red peer-to-peer es una red de ordenadores en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red.



(a)



(b)

Figura 1.4: *Proyecciones al 2019 expuestas por CISCO [4] en donde se muestra (a) el incremento del flujo de datos en los centros de datos tradicionales y nube (b) el número de usuarios que utilizan las aplicaciones de almacenamiento en la nube.*

co, donde más de las cuartas quintas partes del tráfico del centro de datos, serán generadas por la nube [4]. La Fig. 1.4(a) muestra la tasa de crecimiento anual compuesta (*Compound Annual Growth Rate - CAGR*) del tráfico IP (*Internet Protocol - IP*) en los centros de datos tradicionales y la nube (*cloud data center*). La diferencia entre los dos conceptos radica en la ubicación de los servicios. En el caso de los centros de datos tradicionales, concentran todo el almacenamiento dentro de una misma red local de una organización. Para la nube, el almacenamiento se realiza en la misma Internet gestionando el servicio por intermedio de múltiples redes locales. La tendencia al uso del almacenamiento en la nube tiene perspectivas importantes tal como se refleja en la Fig. 1.4(b).

El crecimiento vertiginoso del tráfico en los centros de datos y servicios en la nube se debe en gran medida al incremento de datos que generan no sólo las personas o usuarios, sino también las máquinas y otros dispositivos o cosas (por ejemplo, sensores en avio-

nes, automóviles y edificios), denominando todo esto el Internet de las cosas (*Internet of Things - IoT*). Según estudios y predicciones [4] el volumen de información crecerá mas de doscientas veces en los próximos cinco años.

Bajo estas condiciones, los centros de datos se encuentran sometidos a altas exigencias técnicas que pretenden restricciones de tiempos muy acotados. Tal como se estipula, en un futuro no muy lejano [5], la infraestructura de estos sistemas buscan migrar por completo a fibra óptica. Sin embargo, para satisfacer las demandas actuales, los investigadores han propuesto soluciones de procesamiento de señales que puedan utilizar los medios de transmisión que hoy en día componen un centro de datos complejo. Típicamente podemos encontrar

- **Fibra óptica:** Se utiliza ampliamente en los sistemas de telecomunicaciones, ya que permiten enviar gran cantidad de datos a gran distancia, con velocidades similares a las de radio y superiores a las de cable de cobre. La muy alta velocidad ( $>10\text{Gbps}$ ), exagera los efectos de la dispersión cromática (*Chromatic Dispersion - CD*), polarización modal (*Polarization Mode Dispersion - PMD*), y otras limitaciones del canal [6], las cuales se ven traducidas en un fenómeno denominado interferencia inter-símbolo (*Intersymbol Interference - ISI*). En el caso de los centros de datos la distancia que se busca satisfacer no supera los 2 km [5] y se suele utilizar en la interconexión entre los rack (en el *Capítulo 4* se ampliará la forma de conectar diferentes partes de un centro de datos).
- **Cable de cobre:** Las principales limitaciones son la distancia, ancho de banda y tasa de datos. La atenuación es una función fuertemente dependiente de la frecuencia [7, 8]. La interferencia y el ruido externo también son factores importantes, por eso se utilizan coberturas externas y el trenzado, donde los segmentos no alcanzan distancias superiores a 100 metros. En los centros de datos se suele utilizar en la interconexión de los diferentes servidores y el switch dentro del rack.

El factor común que presentan estos medios de transmisión es la interferencia inter-símbolo. El desafío se centra en proponer arquitecturas equalizadoras que permitan afrontar las altas tasas de datos y compensar la distorsión introducida por el canal. Para ello, las técnicas de paralelización juegan un papel indispensable, con el objetivo de lograr una relación aceptable entre complejidad y desempeño.

### 1.1.2. La Complejidad de Implementación

Uno de los principales efectos que degrada la performance de un sistema de comunicaciones digitales es la ISI [6], la cual afecta todos los sistemas de modulación de pulso. Esto incluye modulación por desplazamiento de frecuencia (*Frequency Shift Keying - FSK*), modulación por desplazamiento de fase (*Phase Shift Keying - PSK*), y la modulación de amplitud en cuadratura (*Quadrature Amplitude Modulation - QAM*). Este fenómeno se origina por la distorsión de amplitud y fase que experimenta la señal transmitida durante su propagación por el canal de comunicaciones [9]. El reconocimiento de este problema se remonta a finales del siglo XIX [10], cuando Oliver Heaviside estudió en detalle la acción distorsionante de las líneas de transmisión y especificó las condiciones para una línea sin distorsión [11]. Una línea sin distorsión se lograba insertando inductancias uniformemente espaciadas en serie. Esta carga de la línea fue uno de los primeros métodos utilizados para eliminar la distorsión del canal o, de manera equivalente, para ecualizar el canal. En 1928, Otto J. Zobel [12] publicó un extenso trabajo que describe el uso de filtros RLC para corregir la distorsión de canal. Estos filtros proporcionan corrección de amplitud satisfactoria, pero se introduce distorsión de fase. Sin embargo, durante un buen tiempo, estos filtros proporcionaron un buen rendimiento a través de circuitos de telefonía, ya que el oído humano es relativamente insensible a las distorsiones de fase.

Durante los últimos 50 años, sin embargo, ecualizadores de canal más sofisticados se han desarrollado para hacer frente a la ISI, impulsados por las altas tasas de datos de los sistemas de comunicaciones digitales a través de canales de ancho de banda limitado [13, 14]. Para hacer frente a las imperfecciones del canal, se han propuesto diferentes técnicas de ecualización. El método de ecualización más atractivo por su baja complejidad en la implementación es el ecualizador lineal (*Lineal Equalizer - LE*) [15, 16], el cual consiste en un filtro que aplanar la respuesta del canal pero su desempeño decrece frente a alto ruido. Por otro lado, las técnicas que mejor rendimiento presentan frente a la ISI son el ecualizador realimentado por decisión (*Decision Feedback Equalizer - DFE*) [17, 18] y el detector de secuencia de máxima verosimilitud (*Maximum Likelihood Sequence Detector - MLSD*).

El DFE se caracteriza por tener una muy buena relación entre complejidad y desempeño en comparación con otras arquitecturas ecualizadoras. Tanto en el LE como el DFE

es posible emplear diferentes criterios de optimización de los coeficientes de los filtros, en donde podemos resaltar los criterios de forzado a cero (*Zero Forcing - ZF*) y minimización del error cuadrático medio (*Minimum Mean Squared Error - MMSE*). Un aspecto crucial de estos ecualizadores es que son sub-óptimos en términos de probabilidad de error. Algunas debilidades inherentes al diseño tales como la restricción de longitud del filtro, amplificación del ruido utilizando el criterio ZF, o la propagación de error en el DFE, restringen las aplicaciones de estos ecualizadores. El aspecto favorable que presentan estos ecualizadores es la relativa baja complejidad comparada con las técnicas óptimas de ecualización.

En las comunicaciones digitales, un criterio de evaluación es la tasa de error de bit de los sistemas. El receptor óptimo que busca minimizar la BER es el MLSD. No remueve la ISI, sino que trata de encontrar la secuencia más probable de símbolos transmitidos. El algoritmo de Viterbi (*Viterbi Algotihm - VA*) [19, 20, 21] es una de las técnicas más empleadas para la implementación del MLSD. La desventaja que presenta esta metodología es el incremento exponencial de la complejidad a medida que aumenta la memoria del canal.

En aplicaciones comerciales, los receptores que utilizan estas técnicas digitales, se suelen implementar en tecnología semiconductor complementario de óxido metálico (*Complementary Metal Oxide Semiconductor - CMOS*) de 28nm [13]. La implementación serial de estas técnicas en sistemas complejos de procesamiento digital, se ve limitada por la frecuencia máxima que puede alcanzar el reloj ( $\approx 1\text{GHz}$ ), lo cual hace prohibitivo su utilización para altas tasas de datos. Es por ello, que es indispensable plantear técnicas de procesamiento paralelo que solucionen estas limitaciones. En primer lugar, en el MLSD la complejidad crece exponencialmente con la memoria del canal, independientemente si utilizamos o no técnicas de paralelización. Por otro lado, el lazo de realimentación dentro del DFE determina el límite superior de velocidad alcanzable en la implementación hardware del caso serial. En otras palabras, el rendimiento del DFE está limitada por la velocidad del lazo de realimentación. Esto nos lleva a buscar técnicas de paralelización del DFE que satisfagan la necesidad de velocidad y baja complejidad. Las técnicas de paralelización actuales (detalladas en la siguiente sección) proponen métodos que continúan teniendo un incremento exponencial de la complejidad dependientes de la memoria del canal.

En resumen, la problemática en los receptores de alta velocidad que utilizan el DFE

como técnica de ecualización para compensar la ISI del canal, está asociada a que

- No hay arquitecturas de implementación estándar del DFE que posean una complejidad reducida en canales muy dispersivos. Las soluciones propuestas tienen un incremento exponencial de la complejidad en relación con la memoria del canal.
- Las técnicas de paralelización del DFE no son totalmente directas (*forward*), lo que agrega un factor de dificultad adicional al diseñador para solucionar los problemas de tiempo (*timing*).

Es por ello que se abre una línea de investigación en la búsqueda de nuevas técnicas de ecualización que permitan reducir la complejidad de implementación y sea paralelizable. Precisamente éste es el camino que recorre la presente Tesis. En la siguiente sección revisaremos las diferentes propuestas que resuelven la problemática de implementar en paralelo arquitecturas recursivas, haciendo foco principalmente en la complejidad de implementación.

## 1.2. Estado del Arte

El incremento de la velocidad de los receptores en sistemas de comunicaciones digitales se encuentra acompañado de una mayor complejidad en la implementación, ya que es necesario aplicar algoritmos y estrategias de diseño que permitan cumplir con este requisito. El desarrollo de la presente Tesis centra los estudios en el ecualizador realimentado por decisiones, en donde la justificación de la elección de este ecualizador se detalla en el *Capítulo 2*. El DFE se caracteriza por tener una buena relación entre desempeño y complejidad pero la dificultad de implementación en receptores de alta velocidad, radica en el lazo realimentado. Esto hace, que la aplicación de técnicas como segmentación (*pipeline*), “retiming” y procesamiento en paralelo resulten complejas y en ocasiones no puedan ser utilizadas. Estas dificultades han inspirado a diversos autores a proponer soluciones novedosas, de las cuales se realiza un resumen de los conceptos básicos en las siguientes secciones.

### 1.2.1. Segmentación y Paralelismo

Para el incremento en la velocidad de procesamiento del DFE se pueden utilizar diferentes técnicas que resuelven las limitaciones del lazo realimentado. Por un lado, es posible aplicar segmentación (*pipeline*) para reducir el camino crítico (*critical path*) e incrementar la velocidad de cómputo en el diseño del ecualizador [22]. Con el mismo objetivo, paralelizar el ecualizador favorece el desempeño [23], no así la complejidad. Es de interés observar que el procesamiento en paralelo y la técnica de segmentación pueden operar en simultáneo, es decir, si un cómputo puede ser segmentado, este puede ser también procesado en paralelo [24]. Estas ventajas del procesamiento pueden mitigar las limitaciones de velocidad en el lazo realimentado usando arquitecturas paralelas basadas en la técnica mirar adelante de lazo de multiplexor segmentado (*look-ahead pipelined multiplexer loop*) [25, 26, 27] y pre-cálculo parcial de las decisiones [27]. En [28] se implementa el concepto de pre-cálculo de las operaciones, usando una red de multiplexores para la selección de la decisión correcta e incorporando sumadores CSA concurrentes (*Concurrent Carry Save Adder - CCSA*) para realizar la suma de los coeficientes.

La reducción de complejidad es el enfoque principal en todas las implementaciones, donde utilizando el algoritmo de procesamiento de bloque incremental (*incremental block processing algorithms*) [24] sobre la base de arquitecturas como mirar adelante de lazo de multiplexor segmentado, se busca hacer frente a este problema [29, 30, 31]. Sin embargo, la complejidad de las arquitecturas paralelas del DFE basadas en estas técnicas aún crecen exponencialmente con la memoria del canal. Trabajos recientes [32, 33], presentan la técnica concurrente mirar adelante (*Concurrent look-ahead*) para tasas de datos muy altas. Este esquema reduce la complejidad del hardware en comparación con la técnica antes mencionada pero el lazo realimentado no es eliminado.

### 1.2.2. Cancelación Iterativa

En los últimos años, la cancelación iterativa de interferencia y turbo ecualización están siendo consideradas como temas de interés [34]. Por ejemplo, la cancelación iterativa es propuesta en [35, 36, 37, 38] donde se introducen ecualizadores no-lineales para canales con ISI. Esta técnica utiliza un algoritmo iterativo para cancelar sucesivamente la ISI de un bloque de datos recibidos. El algoritmo genera decisiones del símbolo cuya fiabilidad

aumenta continuamente con cada iteración [35, 36, 37]. Bajo la misma idea, se puede emplear múltiples DFEs con el objetivo de tomar múltiples decisiones blandas (*soft-decision*) y promediarlas para obtener una métrica con el fin de estimar la decisión final dura (*hard-decision*) [38]. De acuerdo con estos autores, hasta ahora estas técnicas no se han aplicado para crear implementaciones segmentadas y de procesamiento paralelo en aplicaciones de ultra-alta velocidad a pesar de sus novedosas características.

### 1.2.3. Decisiones Tentativas

El concepto de decisiones tentativas radica en la idea de utilizar decisiones parciales para estimar la decisión final. Es importante destacar que las decisiones tentativas han sido utilizadas en el pasado para la cancelación de interferencia de diafonía de extremo (*Far-End Crosstalk - FEXT*) [39]. Particularmente, esta propuesta opera en conjunto con la técnica de codificación Tomlinson-Harashima para la compensación de la ISI. Para canales dinámicos, esta técnica necesita un canal de retorno (*feedback*) y una etapa de entrenamiento para que pueda adaptar los coeficientes en el transmisor.

### 1.2.4. Propuesta

En base a la discusión del estado del arte, la aplicación del DFE y MLSD se limita a canales con moderada ISI. Como consecuencia, hay una necesidad de reducir complejidad en los receptores de alta velocidad que pueden funcionar de manera eficiente en canales con gran ISI. Esta Tesis propone un ecualizador no lineal que utiliza un criterio de iteratividad para compensar la distorsión del canal que denominamos Ecualizador Directo Asistido por Decisiones (*Decision FeedForward Equalizer - DFFE*). Las características principales de esta novedosa arquitectura son

- Emplea decisiones tentativas para compensar la ISI.
- La complejidad se incrementa cuadráticamente con la memoria del canal.
- La implementación es totalmente directa (*forward*).
- El desempeño del DFFE es similar al DFE, esto se demuestra teniendo en cuenta la probabilidad de error del DFE, en donde existen numerosas publicaciones que estudian este concepto [40, 41, 42, 43, 44].



## 1.3. Objetivos y Resultados

Esta Tesis se enfoca en los problemas previamente mencionados, esto es (i) proponer una nueva arquitectura ecualizadora totalmente directa (*forward*) y (ii) brindar una solución de complejidad reducida de implementación paralela. Para el primer caso, se propone una arquitectura ecualizadora que elimina el lazo realimentado del DFE. Esta estructura es fácilmente paralelizable y presenta un incremento cuadrático de la complejidad en relación a la memoria del canal. Por último, también se combina esta técnica con otro algoritmo de detección, como lo es el algoritmo de Viterbi con el mismo fin de reducir complejidad en el receptor. En las siguientes secciones se explican brevemente los resultados obtenidos que se desarrollan en los *Capítulos 4-6*.

### 1.3.1. Nueva Arquitectura Ecualizadora Iterativa de Baja Complejidad

Con la finalidad de reducir la complejidad de implementación en receptores de alta velocidad, se propone una arquitectura totalmente directa (*forward*) denominada Ecualizador Directo Asistido por Decisiones (*Decision FeedForward Equalizer - DFFE*) [45, 46]. En el *Capítulo 4* se realiza el desarrollo del modelo matemático para describir el comportamiento del algoritmo de ecualización propuesto. El análisis determina que el desempeño del DFFE es similar al DFE. A esta conclusión se llega evaluando en forma analítica la probabilidad de error y corroborando los resultados por medio de simulaciones en computadora. Empleando el mismo mecanismo de verificación, se realiza otra propuesta que utiliza el algoritmo de Viterbi en conjunto con el DFFE (*DFFE-VA*) [47] con el objetivo de reducir la complejidad final del receptor. La finalidad de esto es mejorar las primeras decisiones tentativas y de esta forma reducir el número final de iteraciones del receptor. El resultado es un desempeño similar al DFE.

### 1.3.2. Complejidad e Implementación Paralela del DFFE

Se propone una arquitectura paralela del DFFE que tiene la ventaja de la implementación directa del ecualizador. En el *Capítulo 5* se desarrolla un análisis teórico de complejidad y se resaltan las ventajas que posee el DFFE para aplicaciones de alta velo-

cidad frente a otras estructuras ecualizadoras. La característica principal que destaca al DFFE sobre otras opciones es el crecimiento cuadrático de la complejidad en relación a la memoria del canal. Este punto se verifica por medio de resultados de síntesis considerando un diseño del DFFE para VLSI. Por otro lado, también se evalúa la complejidad para el DFFE-VA, en donde se considera la implementación del VA aplicando la técnica propuesta en [48]. El resultado de incorporar estas dos alternativas genera una reducción en la complejidad del receptor, en donde al final del *Capítulo 5* se realiza una comparación con otras arquitecturas.

### 1.3.3. Implementación en FPGA del DFFE

La nueva arquitectura ecualizadora se implementa en FPGA con el objetivo de corroborar todas las evaluaciones realizadas en la presente Tesis. La implementación consiste en el desarrollo del DFFE y el ecualizador lineal en paralelo, en conjunto con el algoritmo de adaptación de mínimo cuadrado medio (*Least Mean Square - LMS*) [9]. Para la verificación, se utiliza el criterio típico del diseño digital que consiste en el análisis a nivel de sistema (simulaciones) y emulación de un caso de uso. Para lograr este último objetivo se implementan en FPGA un generador de secuencia binaria pseudo-aleatoria (*Pseudo-Random Binary Sequence - PRBS*), un generador de ruido Gaussiano y un contador de BER. Todo el sistema es controlado por medio de un microprocesador embebido. Los detalles se describen en el *Capítulo 6*.

## 1.4. Organización

La organización de la presente Tesis es la siguiente. En el *Capítulo 2* se brinda una introducción de las estructuras ecualizadoras clásicas, la evaluación de desempeño de cada una de ellas y la justificación de la elección del DFE como estructura ecualizadora de referencia. En el *Capítulo 3* se analizan las técnicas y estrategias de paralelización de arquitecturas recursivas como el DFE y se evalúa la complejidad de implementación de las propuestas actuales. En el *Capítulo 4* se introduce y estudia la nueva arquitectura ecualizadora de baja complejidad. En el *Capítulo 5* se analiza la complejidad de las propuestas de los nuevos esquemas de ecualización. En el *Capítulo 6* se detalla la implementación en FPGA del DFFE en paralelo en conjunto con otros algoritmos. Finalmente, en el *Capítulo 7* se presentan las principales conclusiones.

# CAPÍTULO

## 2

# ECUALIZACIÓN DE CANAL

***Síntesis:** Este capítulo se enfoca en la elección de una técnica de ecualización tomando como referencia los parámetros de desempeño y complejidad. Se plantea un modelo de canal y las técnicas de ecualización lineal y no lineal. Se evalúan los criterios de estimación de los coeficientes óptimos de los ecualizadores considerando los criterios de forzado a cero y error cuadrático medio. La elección de la mejor técnica de ecualización, se analiza estudiando el desempeño por medio de la estimación teórica y resultados de simulación. La evaluación de estas técnicas de ecualización permite concluir que, en las condiciones de aplicación previstas, el ecualizador realimentado por decisiones es el que mejor relación tiene entre complejidad y desempeño.*

### 2.1. Introducción

En la actualidad, los sistemas de comunicaciones digitales, juegan un papel primordial en el desarrollo de nuevas aplicaciones que manejan un volumen de información extremadamente alto. El mejor aprovechamiento del ancho de banda y las altas tasas de transferencias de datos digitales, hacen de estos sistemas digitales, una herramienta fundamental en el campo de las comunicaciones.

En la Fig. 2.1 se observa un sistema básico de transmisión digital. La fuente de información se codifica para generar símbolos, los cuales se modulan y transmiten a través del canal. En el receptor, la señal recibida es demodulada y decodificada para obtener la información deseada. El principal objetivo del sistema codificador/modulador es minimizar la tasa de error de bit, esto es, procurar que la información detectada sea lo más parecida a la transmitida. El canal de transmisión introduce ruido y genera ISI, lo cual limita el desempeño del sistema de comunicación. El ruido es una señal no deseada que corrompe la señal generando posibles errores en el receptor. Por otro lado, la ISI se origina por las imperfecciones del canal que distorsionan la señal transmitida. Esta distorsión, que es más seria para altas velocidades de transmisión, degrada el desempeño del receptor originando un aumento en la tasa de error de bit.

Uno de los objetivos fundamentales que tienen los diseñadores de sistemas de comunicaciones digitales es incrementar la velocidad de transmisión. Para esto es preciso combatir el ruido y la ISI. Los problemas del ruido pueden reducirse empleando sistemas de codificación de canal (por ejemplo, códigos de bloques, códigos convolucionales, etc.), los cuales permiten reducir la tasa de error. Este beneficio, sin embargo, puede ser seriamente opacado por los problemas originados por la ISI. Los efectos de la ISI pueden minimizarse por medio de filtros que compensen las imperfecciones del canal. Estos filtros, que se denominan *ecualizadores*, dependen fundamentalmente del canal utilizado, por lo cual su respuesta debe ser conocida de antemano. En la mayoría de las aplicaciones prácticas, sin embargo, el canal no es conocido a priori. Por este motivo, es necesario diseñar un filtro que adapte su respuesta en forma dinámica para cada conexión.

El objetivo de este capítulo es analizar el desempeño de las técnicas de detección subóptimas (detección símbolo a símbolo) considerando los efectos de la ISI y ruido aditivo blanco Gaussiano (*Additive White Gaussian Noise - AWGN*). Es sabido que un medida

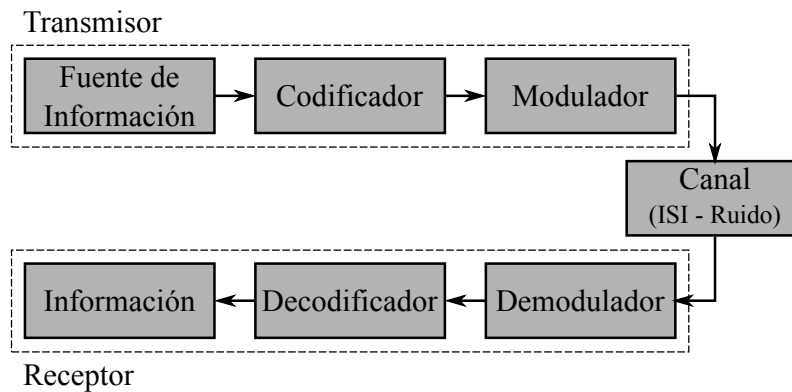


Figura 2.1: *Diagrama en bloque de un sistema de comunicaciones digitales típico.*

significativa del rendimiento de los sistemas de comunicaciones digitales es la probabilidad de error, pero para el caso que se desea analizar (ISI y AWGN) es computacionalmente complejo arribar a una expresión cerrada de esta métrica. Por tal motivo, los resultados de desempeño se expresan en términos de la relación señal a ruido (*Signal to Noise Ratio - SNR*) [6]. Además, la elección de estudiar las técnicas de detección sub-óptimas se debe a la baja complejidad que presentan en comparación con las técnicas de detección óptimas (detector de secuencias). Estas últimas, tienen la desventaja que la complejidad crece exponencialmente en relación a la memoria del canal. En base a estos criterios, en la Fig. 2.2 se muestra una clasificación de las técnicas de ecualización más utilizadas en la industria y se resaltan aquellas que forman parte del desarrollo de la presente Tesis. Los ecualizadores se clasifican según el tipo, la estructura y el algoritmo de adaptación.

- **Tipos:** El ecualizador lineal es ampliamente utilizado en la práctica, el cual consiste en un filtro que aplana la respuesta del canal. Los ecualizadores no lineales, se caracterizan por tener un mejor desempeño frente a la ISI y se destaca el ecualizador realimentado por decisiones. El DFE posee una muy buena relación complejidad/-desempeño en comparación con otras arquitecturas ecualizadoras.
- **Estructuras:** La implementación del LE y DFE con estructuras transversales posee el beneficio de tener una complejidad computacional reducida en comparación con otras alternativas.
- **Algoritmos de adaptación:** El algoritmo de mínimo cuadrado medio se destaca sobre otras técnicas, por su baja complejidad de implementación, lo cual lo convierte

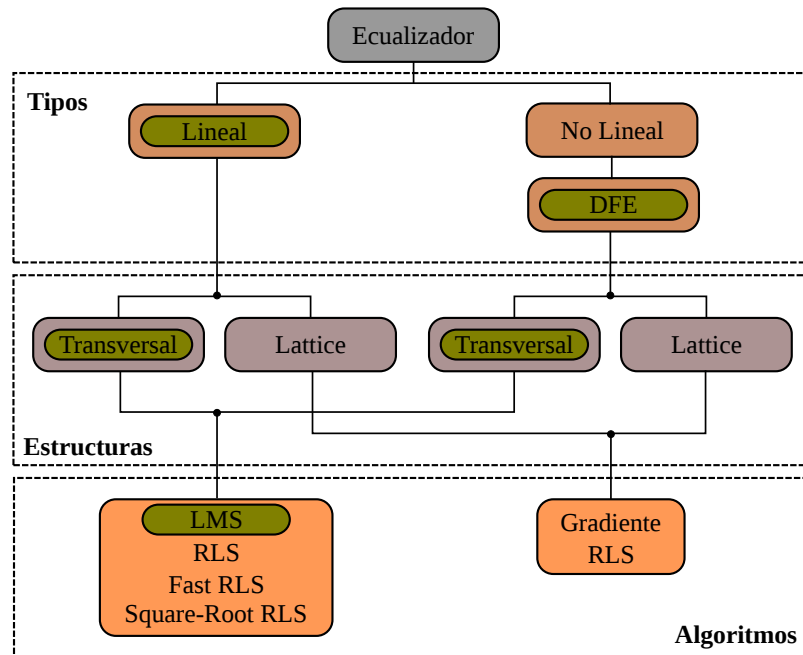


Figura 2.2: Se muestra una clasificación de las diferentes técnicas de ecualización según el tipo, estructura y algoritmo de adaptación. Los tipos se clasifican en Lineal y No lineal (DFE). Entre los algoritmos de adaptación se destacan el LMS, recursivo de mínimos cuadrados (Recursive Least Squares - RLS), rápido RLS (Fast RLS) y raíz cuadrada RLS [9].

en uno de los más utilizados en la industria de las comunicaciones.

A continuación se analizan los conceptos teóricos necesarios para abordar el desarrollo de la problemática que la presente Tesis Doctoral busca resolver.

## 2.2. Modelo de Canal

La Interferencia Inter-símbolo es un impedimento práctico común que se encuentra en muchos sistemas de transmisión y almacenamiento, incluyendo módems de banda vocal, discos rígidos, canales de radio móviles, canales de microondas, e incluso de fibra óptica. El efecto de la ISI es analizado considerando un sistema de comunicaciones con modulación por amplitud de pulso (*Pulse Amplitude Modulation - PAM*) en banda base. En la Fig. 2.3 se ilustra un sistema básico de transmisión digital, el cual esta formado por las etapas de transmisión, canal y recepción.

La señal transmitida se expresa como

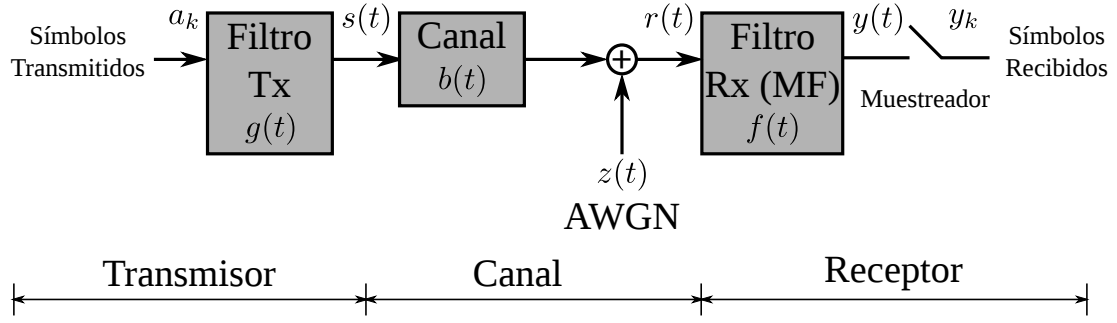


Figura 2.3: Sistema básico de transmisión digital, el cual consiste en un filtro transmisor, canal, ruido aditivo Gaussiano, filtro receptor o filtro apareado (*Matched Filter - MF*) y muestreador.

$$s(t) = \sum_{m=-\infty}^{\infty} a_m g(t - mT) \quad (2.1)$$

donde  $T$  es el período de símbolo,  $a_m$  es el símbolo transmitido y  $g(t)$  la respuesta al impulso del filtro transmisor. El canal es modelado con un filtro lineal invariante en el tiempo (*Linear Time Invariant - LTI*) con respuesta al impulso  $b(t)$ , seguido de ruido aditivo Gaussiano de media cero  $z(t)$  y densidad espectral de potencia  $N_0$ . Por lo tanto, la forma de onda a la entrada del receptor queda definida por

$$r(t) = s(t) * b(t) + z(t) \quad (2.2)$$

$$r(t) = \sum_{m=-\infty}^{\infty} a_m h(t - mT) + z(t) \quad (2.3)$$

donde  $h(t) = g(t) * b(t)$  es la convolución entre  $g(t)$  y  $b(t)$ . Las muestras de entrada al receptor son filtradas por el filtro receptor o filtro apareado (*Matched Filter - MF*) con respuesta al impulso  $f(t)$ . La salida del MF se la representa como

$$y(t) = r(t) * f(t) + n(t) \quad (2.4)$$

$$y(t) = \sum_{m=-\infty}^{\infty} a_m p(t - mT) + n(t) \quad (2.5)$$

donde  $n(t)$  es la versión filtrada del ruido recibido ( $n(t) = f(t) * z(t)$ ) y  $p(t) = g(t) * b(t) * f(t)$  es la forma general de pulso (*overall pulse shape*) compuesta por la cascada del filtro transmisor  $g(t)$ , el canal  $b(t)$  y el filtro receptor  $f(t)$ . La salida del filtro receptor es

muestreada en instantes de tiempo  $t = kT$ , con período de muestreo  $kT$ . En consecuencia, cada muestra queda definida por la secuencia

$$y(kT) = \sum_{m=-\infty}^{\infty} a_m p(kT - mT) + n(kT) \quad (2.6)$$

Reordenando los términos de (2.6) definimos

$$y(kT) = \underbrace{a_k p(0)}_{\text{Símbolo}} + \underbrace{\sum_{\substack{m=-\infty \\ m \neq k}}^{\infty} a_m p(kT - mT)}_{\text{ISI}} + \underbrace{n(kT)}_{\text{Ruido}} \quad (2.7)$$

donde se interpreta que la salida del muestreador consiste en un símbolo transmitido afectado por un factor dependiente de la respuesta al impulso del canal, seguido de ISI y ruido. Los dos últimos términos de (2.7) generan distorsión sobre la señal deseada y afectan la detección. Cada término de la ISI es proporcional a la respuesta al impulso del canal, en donde la ausencia de ISI ( $p(kT) = 0$  para todo  $k \neq 0$ ) da  $y(kT) = a_k p(0) + n(0)$ . La Ec. (2.7) se escribe en tiempo discreto como

$$y_k = \sum_{m=-\infty}^{\infty} a_m p_{k-m} + n_k \quad (2.8)$$

donde  $y_k = y(kT)$ ,  $p_k = p(kT)$  y  $n_k = n(kT)$ .

Considerando (2.8) y  $f(t) = h^*(t)$ , la respuesta al impulso del canal equivalente es

$$p_k = \int_{-\infty}^{\infty} h(t) h^*(t - kT) dt \quad (2.9)$$

también llamado autocorrelación muestreada (*sampled autocorrelation*) del pulso recibido. La transformada de Fourier de  $p(k)$  esta dada por

$$S_h(e^{j\omega T}) = \frac{1}{T} \sum_{m=-\infty}^{\infty} |H(\omega - 2\pi m/T)|^2 \quad (2.10)$$

denominado espectro plegado (*folded spectrum*) del pulso recibido. Por último, el ruido para el canal equivalente se modela como [6]

$$S_n(e^{j\omega T}) = 2N_0 S_h(e^{j\omega T}) \quad (2.11)$$

El modelo de canal equivalente en tiempo discreto se observa en la Fig. 2.4, en donde se considera la notación de la transformada  $Z$ .



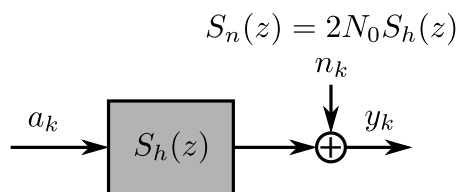


Figura 2.4: Modelo de canal equivalente en tiempo discreto en banda base.

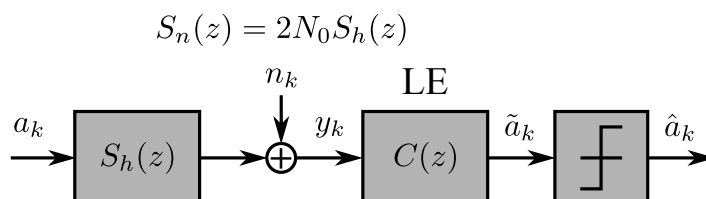


Figura 2.5: Modelo de sistema del ecualizador lineal, el cual incluye el modelo de canal equivalente en tiempo discreto.

Es de interés resaltar, que en un sistema de transmisión digital el desarrollador puede diseñar los filtros transmisor y receptor ( $g(t)$  y  $f(t)$ ), pero no el canal  $b(t)$ . Las respuestas al impulso de  $g(t)$  y  $f(t)$  suelen elegirse para forzar la ISI a cero, de modo que la forma general de pulso satisfaga el criterio de Nyquist [6]. Una dificultad con una estrategia de este tipo, es que el canal rara vez se conoce. Además, incluso cuando se conoce el canal, los filtros necesarios para satisfacer exactamente el criterio de Nyquist pueden ser difíciles o costosos de implementar. En la práctica, la forma general de pulso rara vez cumple con el criterio de Nyquist.

## 2.3. Ecuadorador Lineal

En esta sección se detalla el ecualizador lineal de canal, una de las técnicas más utilizadas para combatir la ISI y se describen distintos criterios para obtener los coeficientes óptimos del ecualizador. El aspecto más importante de estos criterios es la necesidad de conocer de antemano la respuesta del canal, lo cual no siempre es posible en la práctica. Un modelo de sistema del ecualizador lineal en conjunto con el canal se muestra en la Fig. 2.5.

El LE [9] posee un desempeño sub-óptimo en comparación con otras arquitecturas ecualizadoras (DFE y MLSD), pero tiene la ventaja que la complejidad de cómputo, crece linealmente con la memoria de canal  $L$ . Esto lo convierte en uno de los sistemas

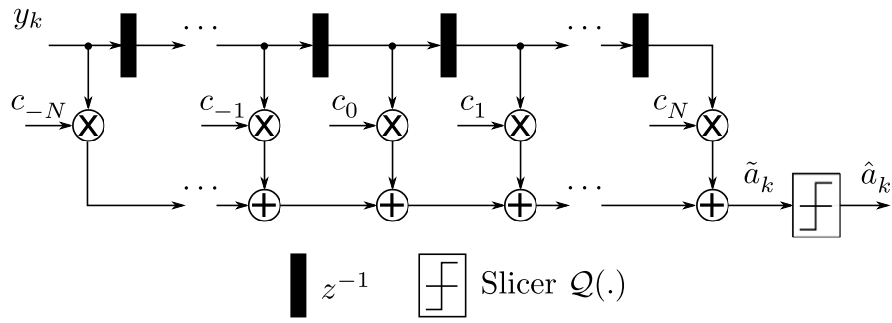


Figura 2.6: *Ecualizador Lineal implementado con un filtro transversal.*

mas utilizados en la industria de las comunicaciones. La estructura mas utilizada para la implementación del LE, es el filtro transversal, donde un esquema detallado se muestra en la la Fig. 2.6. La entrada al LE es la señal muestreada  $y_k$  y la salida es el símbolo estimado  $\tilde{a}_k$ . El filtrado de la señal de entrada queda definido como

$$\tilde{a}_k = \sum_{j=-N}^N c_j y_{k-j} \quad (2.12)$$

donde  $c_j$  son los  $2N + 1$  coeficientes del filtro y  $N$  representa el retardo de la salida en intervalos de símbolo. La estimación  $\tilde{a}_k$  es la entrada al detector de umbral  $\mathcal{Q}(\cdot)$  (*Slicer*), el cual detecta el símbolo transmitido. En este caso particular, donde se considera una modulación PAM antipodal, el detector de umbral entrega a la salida un símbolo  $\pm 1$  ( $\hat{a}_k$ ). Si la decisión  $\hat{a}_k$  no es igual al símbolo transmitido  $a_k$ , entonces se genera un error.

Existen publicaciones que analizan los criterios de optimización de los coeficientes  $c_j$ , de donde se destacan [15, 16, 49]. Dado que la medida más significativa del rendimiento de un sistema de comunicación digital es la probabilidad de error, es deseable elegir los coeficientes que minimicen esta métrica. Sin embargo, la probabilidad de error, es una función altamente no lineal de  $c_j$ . En consecuencia, la probabilidad de error como una métrica de rendimiento para la optimización de los coeficientes del ecualizador, es computacionalmente compleja<sup>1</sup>. Dos criterios han encontrado un amplio uso en la optimización de los coeficientes del ecualizador ( $c_j$ ). El criterio de distorsión de pico (*peak distortion criterion*) o forzado a cero (*Zero-Forcing - ZF*) se limita a eliminar completamente la ISI sin tener en cuenta el aumento de ruido a la salida del ecualizador. Por otro lado, el criterio de error cuadrático medio (*Mean Square Error - MSE*) minimiza el error cuadrático

<sup>1</sup>Excepto en el caso del canal puramente Gaussiano, el cual usualmente se toma como idealización de referencia.

medio entre la entrada del detector de umbral y el símbolo transmitido. A continuación se hace una breve descripción de estos conceptos.

### 2.3.1. Criterio de Forzado a Cero

La distorsión de pico es definida como el peor caso de ISI en la salida del ecualizador. La minimización de esta métrica es llamada criterio de distorsión de pico.

El modelo de canal en tiempo discreto, tiene una respuesta al impulso  $p_k$  y el ecualizador tiene una respuesta al impulso  $c_k$ . Ambos, pueden ser representados por un único filtro equivalente con respuesta al impulso

$$q_n = c_n * p_n \quad (2.13)$$

$$q_n = \sum_{j=-\infty}^{\infty} c_j p_{n-j} \quad (2.14)$$

donde  $q_n$  es la convolución entre  $c_n$  y  $p_n$ . Suponiendo que el ecualizador tiene infinitos coeficientes y considerando el modelo de sistema de la Fig. 2.5, la salida del ecualizador en el instante  $k^{th}$  se representa como

$$\tilde{a}_k = a_k q_0 + \sum_{\substack{m=-\infty \\ m \neq k}}^{\infty} a_m q_{k-m} + \sum_{j=-\infty}^{\infty} c_j n_{k-j} \quad (2.15)$$

La Ec. (2.15) se compone de una versión escalada del símbolo deseado, donde  $q_0$  es el valor normalizado a uno (1) del filtro equivalente de canal, ISI y ruido. El valor pico de la ISI se estima como

$$\mathcal{D}(\mathbf{c}) = \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} |q_n| = \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \left| \sum_{j=-\infty}^{\infty} c_j p_{n-j} \right| \quad (2.16)$$

La ISI se elimina cuando  $\mathcal{D}(\mathbf{c}) = 0$  y se cumple la condición

$$q_n = \sum_{j=-\infty}^{\infty} c_j p_{n-j} = \begin{cases} 1 & (n = 0) \\ 0 & (n \neq 0) \end{cases} \quad (2.17)$$

La transformada  $Z$  de (2.17) es

$$Q(z) = C(z)S_h(z) = 1 \quad (2.18)$$

o simplemente

$$C(z) = \frac{1}{S_h(z)} \quad (2.19)$$

donde  $C(z)$  es la transformada  $Z$  de  $c_j$ . Notar que el ecualizador, con función de transferencia  $C(z)$ , es simplemente el filtro inverso del modelo de filtro lineal equivalente  $S_h(z)$ . En otras palabras, la completa eliminación de la ISI requiere el uso de un filtro inverso para  $S_h(z)$ . Este filtro es llamado filtro de forzado a cero.

El MSE de la señal error  $e_k$  es una métrica que se utiliza para medir el desempeño del ecualizador. La MSE es

$$\varepsilon^2 = E\{|e_k|^2\} = E\{|a_k - \tilde{a}_k|^2\} \quad (2.20)$$

donde  $E\{\cdot\}$  es el valor esperado,  $a_k$  es el símbolo transmitido en  $k^{th}$  intervalo y  $\tilde{a}_k$  es la estimación del símbolo transmitido a la salida del ecualizador. El MSE en términos de la potencia espectral del ecualizador lineal de forzado a cero (*Zero Forcing-Linear Equalizer - ZF-LE*) [6] es

$$\varepsilon_{ZF-LE}^2 = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{N_0}{S_h(e^{j\omega T})} d\omega \quad (2.21)$$

donde  $N_0$  es la densidad espectral de potencia del ruido y  $S_h(e^{j\omega T})$  la potencia espectral del canal equivalente (definida en la *Sección 2.2*). Por conveniencia matemática, se supone que la señal recibida esta normalizada. Esto implica que  $q_0 = 1$  y  $E\{|a_k|^2\} = 1$ . Por último, la relación señal a ruido (*Signal to Noise Ratio - SNR*) a la salida del ecualizador es la inversa de la varianza de ruido  $\varepsilon_{ZF-LE}^2$  y se define como

$$SNR_{ZF-LE} = \left[ \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{N_0}{S_h(e^{j\omega T})} d\omega \right]^{-1} \quad (2.22)$$

Si el espectro del canal equivalente  $S_h(e^{j\omega T})$  posee algún cero, el integrando tiende a infinito y la  $SNR_{ZF-LE}$  tiende a cero. En conclusión, el rendimiento del ecualizador es pobre si el espectro del canal posee valores nulos o muy pequeños.

En la práctica, la implementación de este tipo de ecualizador, nos lleva a tener un número finito de coeficientes. Esto implica que la ISI no siempre es compensada en su totalidad, lo que significa que existe a la salida del ecualizador ISI residual. En consecuencia, se busca minimizar  $\mathcal{D}(\mathbf{c})$  en (2.16) con respecto a los coeficiente  $c_j$ , para obtener un desempeño adecuado.

### 2.3.2. Criterio de Error Cuadrático Medio

El criterio de ZF elimina completamente la ISI, pero tiene el problema que amplifica el ruido a la entrada del detector umbral. El criterio de error cuadrático medio no elimina completamente la ISI sino busca minimizar el error cuadrático medio entre los símbolos transmitidos y la señal a la salida del filtro ecualizador  $e_k$ . Al igual que en el caso del criterio ZF, se utiliza la MSE como una medida del rendimiento, definida como

$$\varepsilon^2 = E\{|e_k|^2\} = E\{|a_k - \tilde{a}_k|^2\} \quad (2.23)$$

Los coeficientes del ecualizador  $c_j$  se calculan para minimizar la MSE  $\varepsilon_{MMSE-LE}^2$  a la salida del ecualizador considerando infinitos coeficientes. En este caso, la estimación de  $\tilde{a}_n$  se expresa como

$$\tilde{a}_k = \sum_{j=-\infty}^{\infty} c_j y_{k-j} \quad (2.24)$$

Reemplazando (2.24) en la expresión para  $\varepsilon^2$  de (2.23) y expandiendo el resultado, se deriva la función de transferencia para este tipo de ecualizador [6]

$$C(z) = \frac{1}{S_h(z) + N_0} \quad (2.25)$$

Se considera como solución válida a la expresión anterior si el ruido es aditivo y blanco Gaussiano (*Additive White Gaussian Noise - AWGN*) y los símbolos transmitidos son no correlacionados, es decir

$$E\{a_k a_n^*\} = \begin{cases} E_a & k = n \\ 0 & k \neq n \end{cases} \quad (2.26)$$

Observando (2.25), la única diferencia entre la función de transferencia del ecualizador ZF y MSE es el factor  $N_0$ . Se puede demostrar que, cuando  $N_0$  es muy pequeño en comparación con la señal, los coeficientes estimados del ecualizador para ambos criterios son similares. La MSE en términos de la potencia espectral para el ecualizador lineal considerando el mínimo MSE (MMSE-LE) [6] es (con  $E_a = 1$ )

$$\varepsilon_{MMSE-LE}^2 = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{N_0}{S_h(e^{j\omega T}) + N_0} d\omega \quad (2.27)$$

y la SNR a la salida del ecualizador es

$$SNR_{MMSE-LE} = \frac{1}{\varepsilon_{MMSE-LE}^2} - 1 \quad (2.28)$$

### 2.3.3. Ecualizador de Espaciamiento Fraccionado

En el ecualizador lineal analizado en las secciones anteriores, los coeficientes se consideran espaciados a tasa de símbolo, por ejemplo  $1/T$ . El espaciamiento de los coeficientes es óptimo, si el ecualizador se encuentra precedido por el filtro apareado. Si las características del canal son desconocidas, el filtro receptor es apareado con el pulso de la señal transmitida y el tiempo de muestreo es optimizado para este filtro sub-óptimo. En general, este enfoque conduce a un rendimiento del ecualizador que es muy sensible a la elección del tiempo de muestreo. En contraste con el ecualizador a tasa de símbolo, el ecualizador de espaciamiento fraccionado (*Fractional Space Equalizer - FSE*), se basa en el muestreo de la señal entrante al menos tan rápido como la tasa de Nyquist. Típicamente, en aplicaciones comerciales, la separación de las muestras es  $T/2$ .

En base a lo discutido en secciones anteriores, se concluye que el ecualizador lineal tiene un buen desempeño cuando las características espectrales del canal no presentan regiones nulas. En general, el canal con regiones espectrales nulas resulta en un incremento del ruido a la salida del ecualizador lineal. Esta limitación básica del ecualizador lineal para hacer frente a la ISI severa, es uno de los motivos por el cual genera interés en utilizar otras técnicas no lineales de ecualización con baja complejidad computacional. Tal es el caso del ecualizador realimentado por decisiones que resuelve este problema de amplificación de ruido. El DFE se describirá brevemente en la *Sección 2.4*.

## 2.4. Ecualizador Realimentado por Decisiones

Los ecualizadores lineales presentan el problema de introducir un efecto de amplificación de ruido. Este efecto se hace más notorio cuando los canales presentan zonas del espectro cercanas a cero. El ecualizador realimentado por decisiones (*Decision Feedback Equalizer - DFE*) [9] resuelve este inconveniente (aunque no en forma óptima) utilizan-

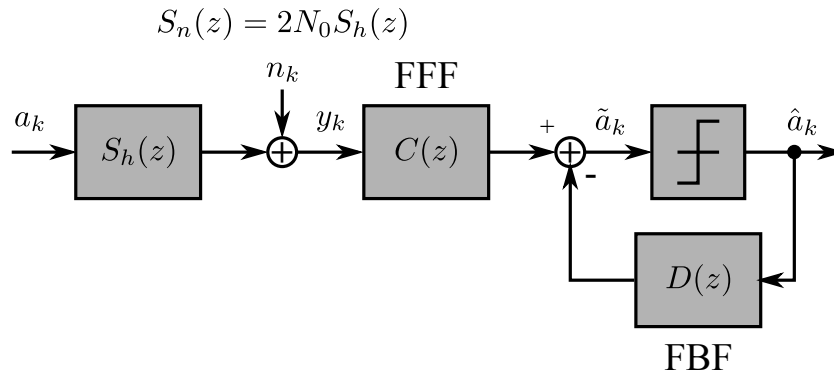


Figura 2.7: Modelo de sistema del ecualizador realimentado por decisiones (DFE), el cual incluye el modelo de canal equivalente en tiempo discreto.

do una estructura no lineal. El DFE consiste en dos filtros, un filtro directo y un filtro realimentado, como se muestra en el modelo de sistema en la Fig. 2.7.

La entrada del filtro directo (*Feed Forward Filter - FFF*) es la secuencia de símbolos transmitidos con ISI y el ruido aditivo. El filtro realimentado (*FeedBack Filter - FBF*) toma como entrada la secuencia de decisiones de los símbolos previamente detectados. Básicamente, el filtro realimentado es usado para eliminar parte de la ISI del símbolo actual, causada por los símbolos previamente detectados.

### 2.4.1. Caracterización del DFE

En la Fig. 2.8 se describe un esquema de implementación típico del DFE. La señal a la entrada del detector de umbral (*Slicer*) se la representa como

$$\tilde{a}_k = \sum_{j=-N_f}^0 c_j y_{k-j} + \sum_{j=1}^{N_b} d_j \hat{a}_{k-j} \quad (2.29)$$

donde  $\hat{a}_n$  es el símbolo detectado a la salida del detector de umbral,  $c_j$  los coeficientes del filtro directo y  $d_j$  los coeficientes del filtro realimentado. Se considera que se tienen  $N_f + 1$  coeficientes en el filtro directo y  $N_b$  coeficientes en el filtro realimentado.

El funcionamiento conceptual del DFE se describe en la Fig. 2.9(a), donde las muestras con índice de tiempo positivo son llamadas post-cursores y con índice negativo pre-cursores. La Fig. 2.9(a) representa la respuesta al impulso del canal a la entrada del DFE (es decir, a la entrada del FFF). El objetivo del filtro directo es compensar la ISI de los pre-cursores. El canal equivalente a la salida del FFF, queda definido por la convolución

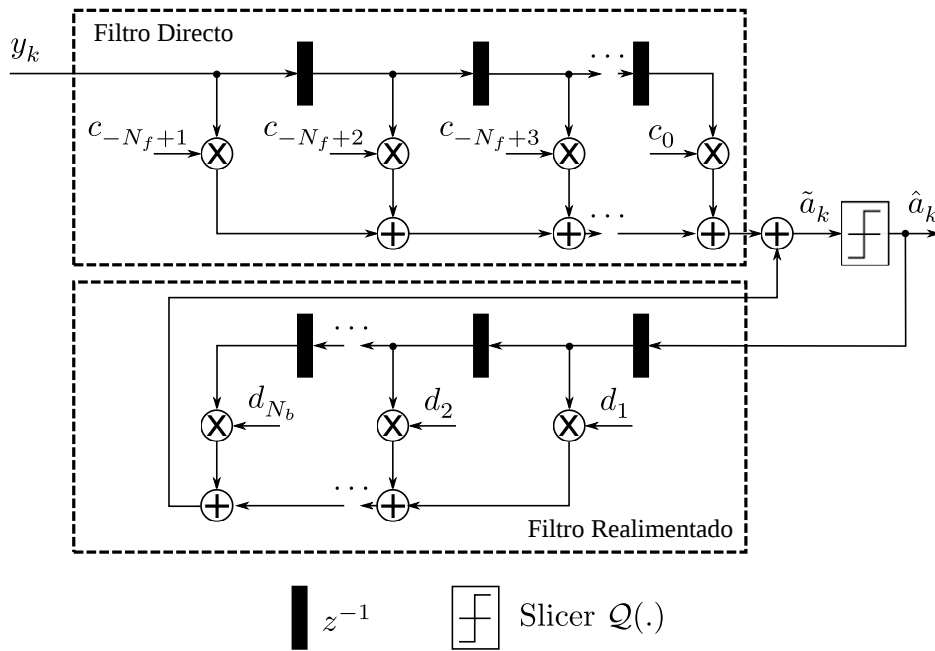


Figura 2.8: Ecuador realimentado por decisiones (DFE).

entre  $p_j$  y  $c_j$  como se ilustra en la Fig. 2.9(b). Idealmente, se busca que el canal equivalente tenga valores muy pequeños (nulos) de los pre-cursosores y una función de transferencia de fase mínima [6]. El papel del FFF es, convertir el componente de canal de fase máxima a fase mínima reflejando polos y ceros dentro del círculo unitario. Otra interpretación de este resultado es que, entre todas las respuestas causales con la misma magnitud de la transformada de Fourier, la respuesta de fase mínima tiene energía máxima cerca de  $k = 0$ . Por lo tanto, en este sentido, la respuesta de fase mínima minimiza la energía de la ISI que debe ser cancelada por el FBF. De esta forma, el FBF solo tiene que compensar la ISI de post-cursosores. Es importante resaltar, que en la práctica, es habitual encontrar a los filtros directos implementados como un FSE, en donde el filtro realimentado opera al período de símbolo  $T$ .

Un problema típico en el DFE, es el error en la decisión. Esto causa una estimación incorrecta de los post-cursosores del canal. El resultado es que un solo error provoca una reducción en el margen de ruido para un número de decisiones futuras. Este fenómeno se denomina propagación de errores, y se traduce en un aumento en la tasa de error. La propagación de errores se analizará con mas detalle en la *Sección 2.4.2*.

Los criterios de análisis de desempeño del DFE son el forzado a cero (ZF-DFE) y el error cuadrático medio (MSE-DFE). El ZF-DFE fuerza a cero la ISI a la entrada del



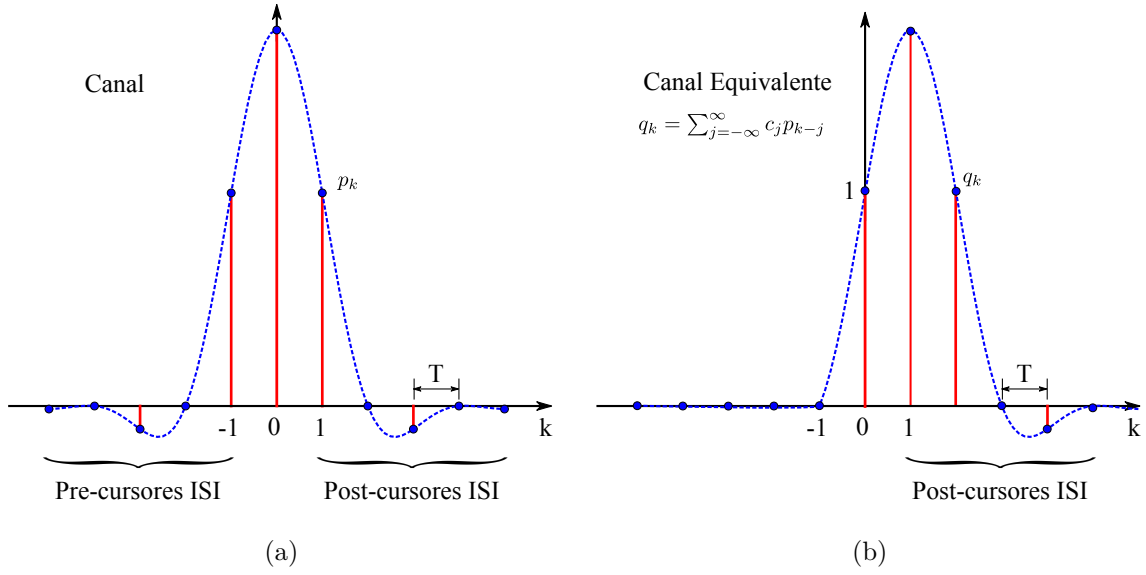


Figura 2.9: (a) Representación del canal con pre y post cursores antes de ser filtrado por el LE. (b) Canal equivalente después de ser filtrado por el FFF.

detector de umbral, mientras que el MSE-DFE minimiza la varianza del error del detector de umbral. Ambos criterios fueron analizados en detalle por Price (1972) [50] y Salz (1973) [51], respectivamente. Al igual que en la *Sección 2.3*, se considera la MSE como una medida de desempeño del DFE. Se puede demostrar [6], que la MSE del DFE utilizando el criterio de forzado a cero (ZF-DFE) y el mínimo error cuadrático medio (MMSE-DFE) son

$$\varepsilon_{ZF-DFE}^2 = \exp \left\{ \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \ln \left( \frac{N_0}{S_h(e^{j\omega T})} \right) d\omega \right\} \quad (2.30)$$

$$\varepsilon_{MMSE-DFE}^2 = \exp \left\{ \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \ln \left[ \frac{N_0}{S_h(e^{j\omega T}) + N_0} \right] d\omega \right\} \quad (2.31)$$

En ambos casos, se considera que la señal de entrada al DFE esta normalizada y no se cometen errores en la detección. Por último, la SNR para ambos criterios es

$$SNR_{ZF-DFE} = \exp \left\{ \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \ln \left( \frac{N_0}{S_h(e^{j\omega T})} \right) d\omega \right\}^{-1} \quad (2.32)$$

$$SNR_{MMSE-DFE} = \frac{1}{\varepsilon_{MMSE-DFE}^2} - 1 \quad (2.33)$$

### 2.4.2. Propagación del Error en el DFE

Un problema potencial con el DFE es que los errores de decisión en la salida del detector de umbral causarán una estimación equivocada de la ISI. El resultado es que un

solo error provoca una reducción en el margen del ruido para un número de decisiones futuras. Este fenómeno se denomina propagación de errores, y se traduce en una tasa de error mayor que el que puede predecir sobre la base de cálculos de SNR solamente.

En esta sección consideramos la naturaleza de la propagación de errores en el DFE, encontrando en particular, los límites superiores de la probabilidad de error que demuestran que los efectos suelen ser insignificantes en comparación con los beneficios de la mejora de ruido reducido [6]. Podemos modelar el fenómeno de propagación de error eliminando la hipótesis que  $a_k = \hat{a}_k$ , y reescribiendo la señal a la entrada del detector de umbral como

$$y_k = a_k + \sum_{j=1}^{\infty} d_j a_{k-j} - \sum_{j=1}^{\infty} d_j \hat{a}_{k-j} + z_k \quad (2.34)$$

donde  $z_k$  es el valor de ruido. Esto puede ser reescrito como

$$y_k = a_k + v_k + z_k \quad (2.35)$$

donde el término medio es la ISI residual debido a la cancelación incorrecta de las muestras con ISI post-cursor dadas por

$$v_k = \sum_{j=1}^L d_j w_{k-j}, \quad w_k = a_k - \hat{a}_k \quad (2.36)$$

y asumiendo un número finito de coeficientes ( $L$ ) en el filtro realimentado. A los efectos de la comprensión de este fenómeno, se tomará como ejemplo el caso de banda base 2-PAM antipodal, de manera que  $a_k = \pm 1$  y  $w_k$  asume los valores  $[\pm 2, 0]$ . Para este caso, el detector de umbral considera un umbral en cero, y podemos calcular fácilmente la probabilidad para ambos tipos de error en el instante  $k$ . El primer tipo de error se produce cuando se transmite un símbolo  $a_k = 1$  y la entrada al detector de umbral es negativa. Por lo tanto, la probabilidad de que ocurra  $w_k = 2$  es

$$Pr\{w_k = 2\} = p \cdot Pr\{1 + v_k + z_k < 0\} \quad (2.37)$$

donde  $p = Pr\{a_k = 1\}$  es la probabilidad de transmitir el símbolo  $a_k = 1$ . Del mismo modo, la probabilidad de que ocurra  $w_k = -2$  es

$$Pr\{w_k = -2\} = (1 - p) \cdot Pr\{-1 + v_k + z_k > 0\} \quad (2.38)$$

---

## 2.4. Ecuador Realimentado por Decisiones

La intuición detrás de este resultado es que no importa cuán grande es la ISI, que en realidad reduce la probabilidad de error para una polaridad de símbolo de datos, y si los símbolos son igualmente probables entonces la probabilidad de error puede no ser peor que  $1/2$  para cualquier ISI residual.

Por supuesto, la conclusión de que la probabilidad de error no es peor que un medio es de poco valor, ya que podríamos lanzar una moneda en el receptor y hacerlo igual de bien. Con el fin de obtener resultados más fuertes, asumamos que los símbolos de datos  $a_k$  y las muestras de ruido  $z_k$  son independientes. Sabemos que las muestras de ruido serán independiente cuando  $L \rightarrow \infty$  para el diseño óptimo del filtro directo, y debe ser aproximadamente cierto para  $L$  suficientemente grande. Entonces podemos ver que  $Pr\{w_k\}$  depende sólo de  $w_{k-j}, 1 \leq j \leq L$ , y por lo tanto  $w_k$  es una cadena de Markov con  $3^L$  estados. Mientras que la probabilidad de estado estacionario de los estados de esta cadena se puede calcular en [18], podemos desarrollar fácilmente un modelo simple para esta cadena que da un límite superior en la probabilidad de error. Esto permitirá describir conceptualmente el fenómeno de la propagación de errores [41].

Supongamos que, en ausencia de ISI la probabilidad de error es  $P_{e,0}$

$$P_{e,0} = Pr\{w_k \neq 0 | w_{k-j} = 0, 1 \leq j \leq L\} \quad (2.39)$$

Por otro lado, se considera como peor caso cuando si hay ISI residual y la probabilidad de error es  $1/2$

$$Pr\{w_k \neq 0 | w_{k-j} \neq 0 \forall j \in (1, 2, \dots, L)\} = 1/2 \quad (2.40)$$

Definir una cadena de Markov  $X_k$ , donde  $X_k$  es un recuento del número de decisiones correctas sucesivas que se han hecho, pero sin incluir el tiempo  $k$ . Es decir,  $X_k = n$  si  $w_{k-1} = w_{k-2} = \dots = w_{k-n} = 0$  y  $w_{k-n-1} \neq 0$ . Ahora tenemos el siguiente modelo para un evento de propagación de error. Si  $X_n \geq L$ , no hay interferencia residual porque no ha habido errores en la memoria del filtro realimentado del DFE. Supongamos que  $X_n \geq L$  pero se comete un error de todos modos en el instante  $k$ ,  $w_n \neq 0$ , debido al ruido aditivo, resultando en  $X_{k+1} = 0$ . Entonces, de acuerdo a nuestro modelo de peor caso, los errores se harán con una probabilidad  $1/2$  hasta el momento en que  $L$  decisiones correctas consecutivas se hayan hecho, momento en el cual volvemos al estado de cero ISI

residual y la probabilidad de error retorna a  $P_{e,0}$ . Supongamos que en promedio se tarda  $K$  incrementos de tiempo hasta que se hayan hecho  $L$  decisiones correctas en una fila, donde  $K$  es la duración media de un evento de propagación de error. Dado que la probabilidad de error es  $1/2$  durante los eventos, resulta la propagación de error en un promedio de errores  $K/2$  por cada error debido al ruido aleatorio. Por lo tanto, la probabilidad de error teniendo en cuenta ISI es

$$P_e = \left(\frac{1}{2}K + 1\right) P_{e,0} \quad (2.41)$$

En realidad, se espera que la probabilidad de error sea menor que esto, ya que la probabilidad de error durante un evento de error será inferior a  $1/2$ . Este límite se puede hacer más rigurosa, ver [41].

Este planteo, a pesar de que es el peor de los casos, da una considerable información sobre el mecanismo de propagación de errores. Esto demuestra que los eventos de error terminarán cuando se tomen  $L$  decisiones correctas, y que esto sucede en un tiempo relativamente corto ya que la probabilidad de error no es peor que  $1/2$  sin importa cuán grande sea la ISI. Este argumento depende en gran medida de considerar que los datos son igualmente probables, y de hecho hay, considerando el peor de los casos, secuencias de datos para los que el evento de error puede persistir mucho más tiempo de lo que se ha predicho aquí. Esto sugiere que es importante asegurar que los datos sean aleatorios y es deseable mantener  $L$  tan pequeño como sea posible consistente con la obtención de la mayor parte de los beneficios del DFE.

Podemos determinar  $K$  en (2.41) observando que  $K$  es el número medio de lanzamientos de una moneda justo antes obtener  $L$  caras en una fila. Donde

$$K = \frac{1 - 1/2^L}{1/2^{L+1}} = 2(2^L - 1) \quad (2.42)$$

Sustituyendo en (2.41), obtenemos

$$P_e = 2^L P_{e,0} \quad (2.43)$$

y la probabilidad de error se multiplica por un factor de  $2^L$  debida a la propagación de error. Si  $L$  es bastante modesto, este factor es compensado por el beneficio del DFE en la reducción de  $P_{e,0}$ . Por ejemplo,  $L = 3$  genera un incremento en un orden de magnitud en

la probabilidad de error debido al error de propagación, y el DFE debe reducir  $P_{e,0}$  sólo medio dB para dar lugar a una reducción neta de probabilidad de error.

## 2.5. Análisis Teórico de Desempeño

En esta sección se compara el desempeño entre las diferentes técnicas de ecualización utilizando los criterios de forzado a cero y error cuadrático medio [52, 53, 54]. Se adopta un canal Gaussiano con ISI y modulación PAM (antipodal). Además, el modelo de señal en tiempo discreto a la salida del muestreador es

$$y_k = \sum_{m=0}^L a_m p_{k-m} + n_k \quad (2.44)$$

El símbolo transmitido  $a_m$  es una variable aleatoria con media cero. La respuesta al impulso del canal  $p_k$  es finita de longitud  $L+1$  y energía finita ( $\sum_{m=0}^L |p_m|^2 < \infty$ ). Además,  $p_k$  contempla los filtros transmisor, canal y receptor. El ruido  $n_k$  se lo asume Gaussiano con media cero e independiente de la entrada. Los filtros del ecualizador poseen una respuesta al impulso finita. Por último, la probabilidad de error teórica para la modulación PAM esta definido por [9]

$$P_e = \mathcal{Q}(\sqrt{2\gamma}) = \mathcal{Q}\left(\sqrt{2\frac{E_a}{N_0}}\right) \quad (2.45)$$

donde  $\gamma$  representa la SNR. Para facilitar la interpretación de las señales que se consideran en el análisis, se reescribe la expresión de la señal recibida (2.44) en forma vectorial como

$$\mathbf{y}_k = \mathbf{P}\mathbf{a}_k + \mathbf{n}_k \quad (2.46)$$

donde  $\mathbf{P}$  es la matriz de tamaño  $N_f \times (N_f + L)$ ,  $\mathbf{a}_k$  el vector de símbolos transmitidos y  $\mathbf{n}_k$  el vector de ruido. Mas específicamente

$$\begin{aligned}
 \begin{bmatrix} y_{k+N_f-1} \\ y_{k+N_f-2} \\ \vdots \\ y_k \end{bmatrix} &= \begin{bmatrix} p_0 & p_1 & \dots & p_L & 0 & 0 & \dots & 0 \\ 0 & p_0 & p_1 & \dots & p_L & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & p_0 & p_1 & \dots & p_L \end{bmatrix} \begin{bmatrix} a_{k+N_f-1} \\ a_{k+N_f-2} \\ \vdots \\ a_{k-L} \end{bmatrix} \\
 &+ \begin{bmatrix} n_{k+N_f-1} \\ n_{k+N_f-2} \\ \vdots \\ n_k \end{bmatrix} \tag{2.47}
 \end{aligned}$$

donde  $N_f$  representa el número de coeficientes del filtro directo del ecualizador. La salida del ecualizador se expresa como

$$\tilde{a}_k = \mathbf{w}^T \mathbf{y}_k \tag{2.48}$$

donde  $\mathbf{w}$  es el vector de coeficientes del filtro. Por definición [9], la SNR a la entrada del detector de umbral se representa como

$$SNR = \frac{E_a}{E\{|e_k|^2\}} = \frac{E_a}{E\{|a_k - \tilde{a}_k|^2\}} = \frac{E_a}{\varepsilon^2} \tag{2.49}$$

donde  $\varepsilon^2$  representa la MSE a la salida del ecualizador considerando la ISI residual y ruido. Empleando el principio de ortogonalidad [6], la MSE se minimiza cuando  $E\{e_k \mathbf{y}_k^*\} = 0$ , por lo tanto la MSE para el LE y el DFE queda definida como

$$E\{a_k \mathbf{y}_k^*\} - \mathbf{w} E\{\mathbf{y}_k \mathbf{y}_k^*\} = 0 \tag{2.50}$$

$$\mathbf{R}_{\mathbf{a}\mathbf{y}} - \mathbf{w} \mathbf{R}_{\mathbf{y}\mathbf{y}} = 0 \tag{2.51}$$

donde  $\mathbf{R}_{\mathbf{y}\mathbf{y}}$  y  $\mathbf{R}_{\mathbf{a}\mathbf{y}}$  representan la auto-correlación de la señal de entrada al ecualizador ( $y_k$ ) y la correlación cruzada de los símbolos transmitidos con la señal recibida en el ecualizador, respectivamente. Por lo tanto, los coeficientes óptimos son

$$\mathbf{w} = \mathbf{R}_{\mathbf{a}\mathbf{y}} \mathbf{R}_{\mathbf{y}\mathbf{y}}^{-1} \tag{2.52}$$

Las variables que consideran los coeficientes y señales dependiendo cada ecualizador se expresan como

$$\mathbf{y}_k = \begin{cases} \mathbf{y}_k & , \mathbf{y}_{LE,k} \\ [\mathbf{y}_k^T, \hat{\mathbf{x}}_k^T]^T & , \mathbf{y}_{DFE,k} \end{cases} \quad (2.53)$$

$$\mathbf{w} = \begin{cases} [c_{-N_f+1}, \dots, c_0]^T & , \mathbf{w}_{LE} \\ [c_{-N_f+1}, \dots, c_0, d_1, \dots, d_{N_b}]^T & , \mathbf{w}_{DFE} \end{cases} \quad (2.54)$$

$$\mathbf{R}_{\mathbf{ay}} = \begin{cases} \mathbf{R}_{\mathbf{ay}} & , \mathbf{R}_{\mathbf{ay}LE} \\ [\mathbf{R}_{\mathbf{ay}}, \mathbf{0}_{N_b}] & , \mathbf{R}_{\mathbf{ay}DFE} \end{cases} \quad (2.55)$$

$$\mathbf{R}_{\mathbf{yy}LE} = \mathbf{E}_a \mathbf{P} \mathbf{P}^* + \frac{N_0}{2} \mathbf{R}_{\mathbf{nn}} \quad (2.56)$$

$$\mathbf{R}_{\mathbf{yy}DFE} = \begin{bmatrix} \mathbf{R}_{\mathbf{yy}} & \mathbf{E}\{\mathbf{y}_k \hat{\mathbf{a}}_k^*\} \\ \mathbf{E}\{\hat{\mathbf{a}}_k \mathbf{y}_k^*\} & \mathbf{E}_a \mathbf{I}_{N_b} \end{bmatrix} \quad (2.57)$$

donde  $N_b$  es el número de coeficientes del filtro realimentado del ecualizador DFE,  $\hat{\mathbf{a}}_k$  son las decisiones a la salida del DFE,  $\mathbf{R}_{\mathbf{nn}}$  la auto-correlación del ruido,  $\mathbf{0}_{N_b}$  vector de ceros de longitud  $N_b$  e  $\mathbf{I}_{N_b}$  la matriz identidad. Bajo todas estas consideraciones, el mínimo MSE y la SNR a la salida del ecualizador, se representan como

$$\varepsilon_{MMSE-LE}^2 = \mathbf{E}_a - \mathbf{w}_{LE} \mathbf{R}_{\mathbf{ay}LE} \quad (2.58)$$

$$\varepsilon_{MMSE-DFE}^2 = \mathbf{E}_a - \mathbf{w}_{DFE} \mathbf{R}_{\mathbf{ay}DFE} \quad (2.59)$$

$$SNR_{MMSE-LE} = \frac{\mathbf{E}_a}{\mathbf{E}_a - \mathbf{w}_{LE} \mathbf{R}_{\mathbf{ay}LE}} - 1 \quad (2.60)$$

$$SNR_{MMSE-DFE} = \frac{\mathbf{E}_a}{\mathbf{E}_a - \mathbf{w}_{DFE} \mathbf{R}_{\mathbf{ay}DFE}} - 1 \quad (2.61)$$

Finalmente, para el criterio de forzado a cero se considera que  $N_0 \rightarrow 0$  para la estimación de los coeficientes óptimos y la SNR a la salida del ecualizador se la expresa como

$$SNR_{ZF-LE} = \frac{E_a}{E_a - \mathbf{w}_{LE} \mathbf{R}_{ayLE} + N_0/2|\mathbf{w}_{LE}|^2} \quad (2.62)$$

$$SNR_{ZF-DFE} = \frac{E_a}{E_a - \mathbf{w}_{DFE} \mathbf{R}_{ayDFE} + N_0/2|\mathbf{c}|^2} \quad (2.63)$$

En el siguiente ejemplo, se muestra el funcionamiento y desempeño del ecualizador LE y DFE empleando los criterios ZF y MSE implementando el análisis teórico anterior. Los coeficientes óptimos son calculados de (2.52) y el desempeño se estima reemplazando la SNR (Ec. (2.60)-(2.63)) en (2.45).

### 2.5.1. Ejemplo

El modelo de sistema del LE y DFE es igual al contemplado en las Fig. 2.5 y 2.7, respectivamente. La Fig. 2.10 muestra el canal equivalente en tiempo discreto ( $p_m = 0,407\delta_{m+1} + 0,815\delta_m + 0,407\delta_{m-1}$ ) normalizado ( $\sum_{m=0}^L |p_m|^2 = 1$ ) y el espectro plegado del canal. Se emplea un número de coeficientes finitos de los filtros de los ecualizadores, donde  $N_f = 31$  para el LE y  $N_f = 15$  y  $N_b = 15$  para el DFE. Además, en el análisis del desempeño del DFE, no se consideran errores en las decisiones. Por último, las gráficas de comportamiento de los ecualizadores se realizan para una SNR de 15dB.

La respuesta del LE se observa en Fig. 2.11(a) y 2.11(b). Notar que el incremento de la amplitud del espectro se da en aquellas zonas donde éste se ve disminuido. En las Fig. 2.11(c) y 2.11(d) muestra que la convolución entre el canal equivalente y los coeficientes del LE ( $C(z)$ ) es aproximadamente el impulso unitario. Los valores de los coeficientes del canal distintos de cero, es consecuencia de la ISI residual debido a la longitud finita del filtro del ecualizador. El comportamiento del LE empleando el criterio MSE se ilustra en la Fig. 2.12, donde a diferencia del caso anterior, se observa ISI residual (Fig. 2.12(c)) pero con menor amplificación del espectro en las zonas que éste se aproxima a cero. La comparación del desempeño para ambos criterios (ZF y MSE) se grafica en la Fig. 2.15.

Las gráficas de las Fig. 2.13 y 2.14 muestran el comportamiento del DFE para los criterios ZF y MSE, respectivamente. El efecto de amplificación del ruido en el filtro directo se ve atenuado, ya que solo compensa la ISI causada por los pre-cursos del canal. En la Fig. 2.15 se evidencia el mejor desempeño del DFE en comparación con el LE para este tipo de canal. Finalmente, se simuló el funcionamiento del LE y DFE utilizando



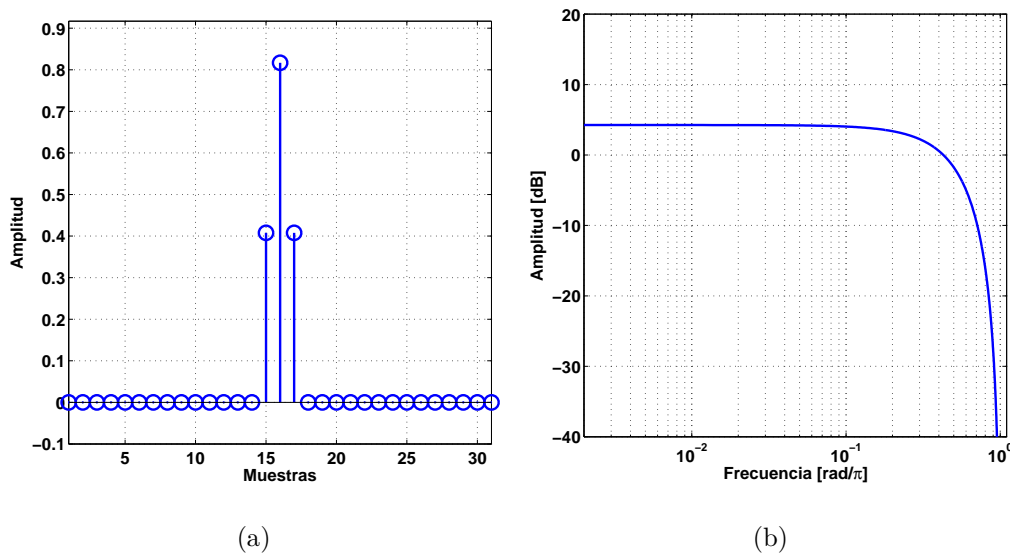


Figura 2.10: (a) *Canal Equivalente en Tiempo Discreto* ( $p_m = 0,4\delta_{m+1} + 0,8\delta_m + 0,4\delta_{m-1}$ ).  
 (b) *Espectro plegado del canal equivalente*

el criterio MSE, con el objetivo de verificar la correcta estimación del desempeño. Como se puede apreciar en la Fig. 2.15, el DFE presenta ventajas significativas de desempeño en comparación con el LE cuando el canal tiene regiones del espectro nulas o cercanas a cero.

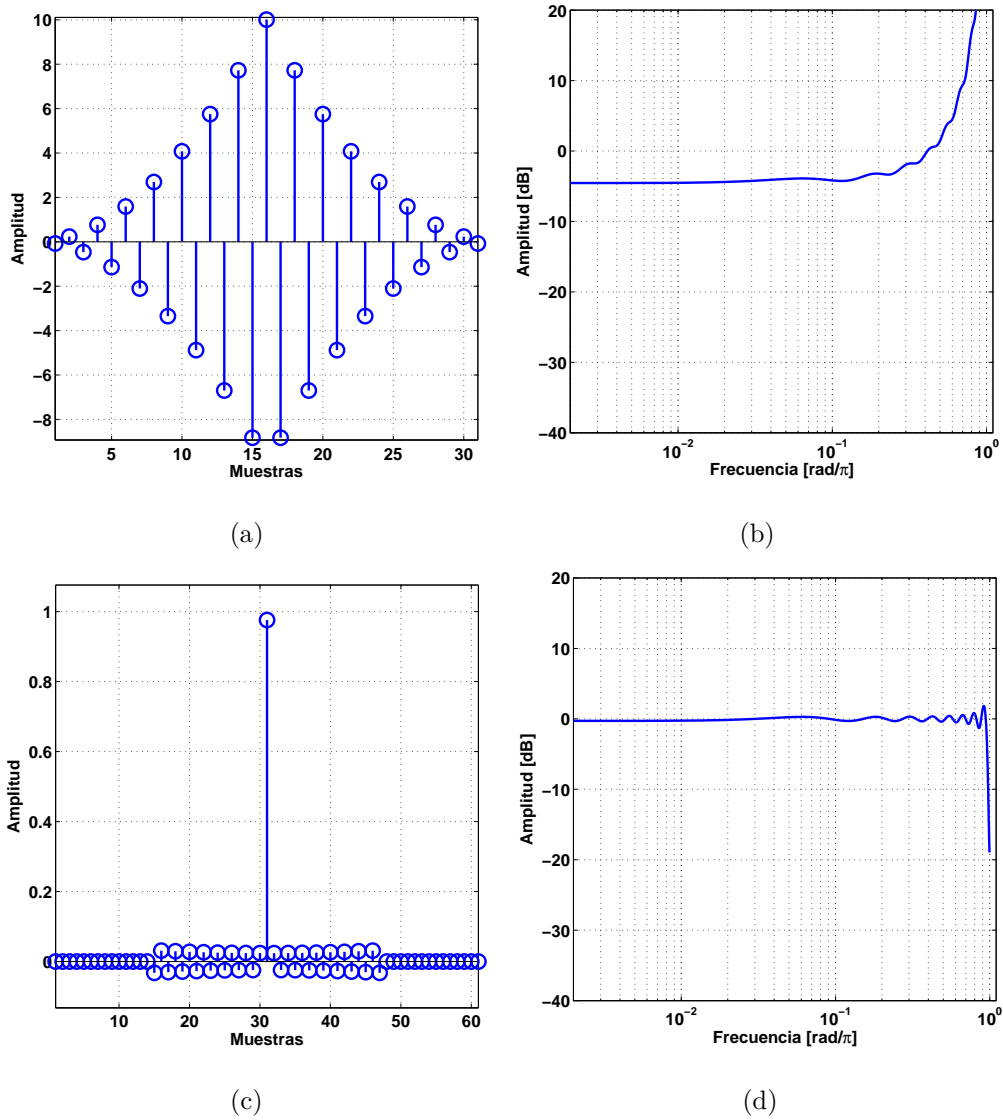


Figura 2.11: Comportamiento del ZF-LE. (a) y (b) Respuesta del LE. (c) y (d) Convolución del canal equivalente con los coeficientes del ecualizador.

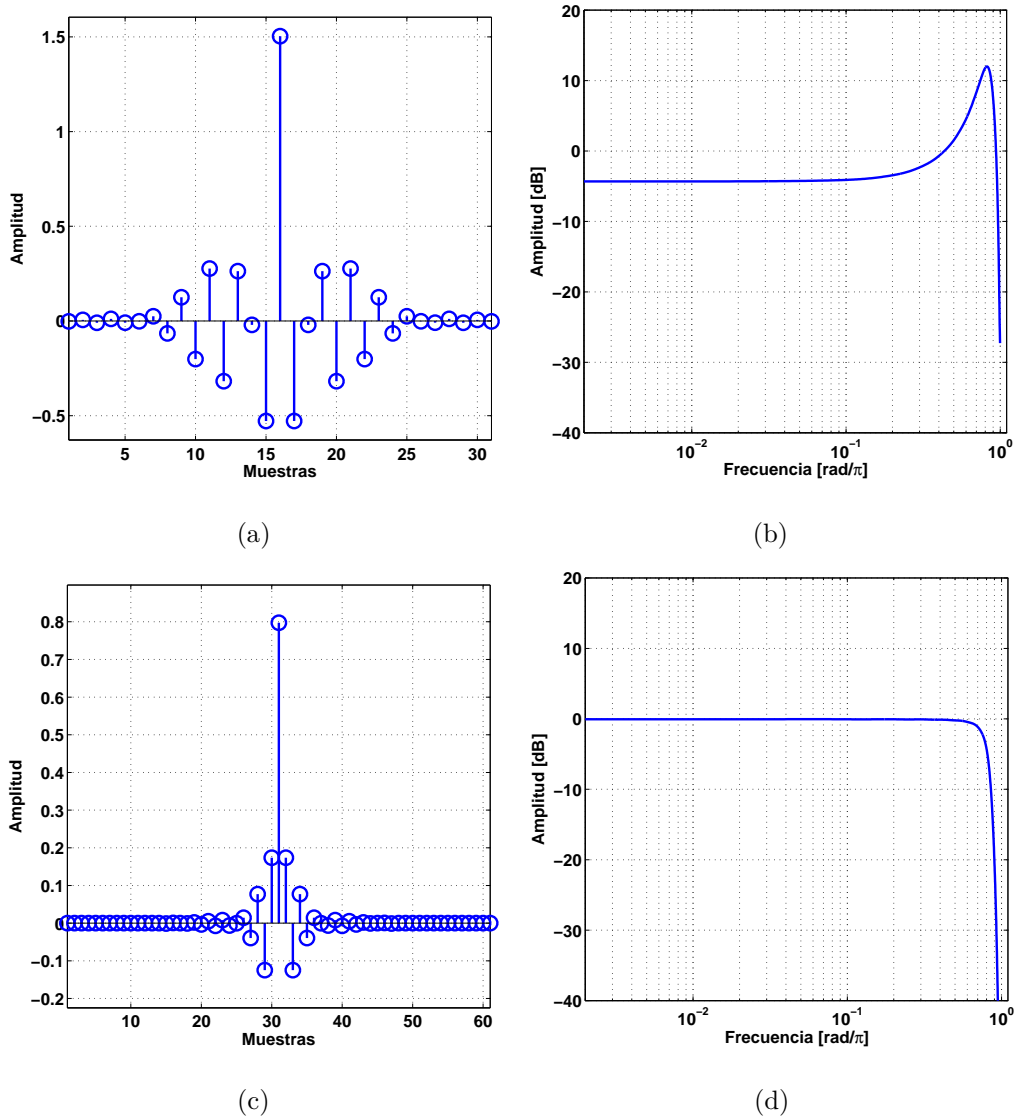


Figura 2.12: Comportamiento del MMSE-LE. (a) y (b) Respuesta del LE. (c) y (d) Convolución del canal equivalente con los coeficientes del ecualizador.

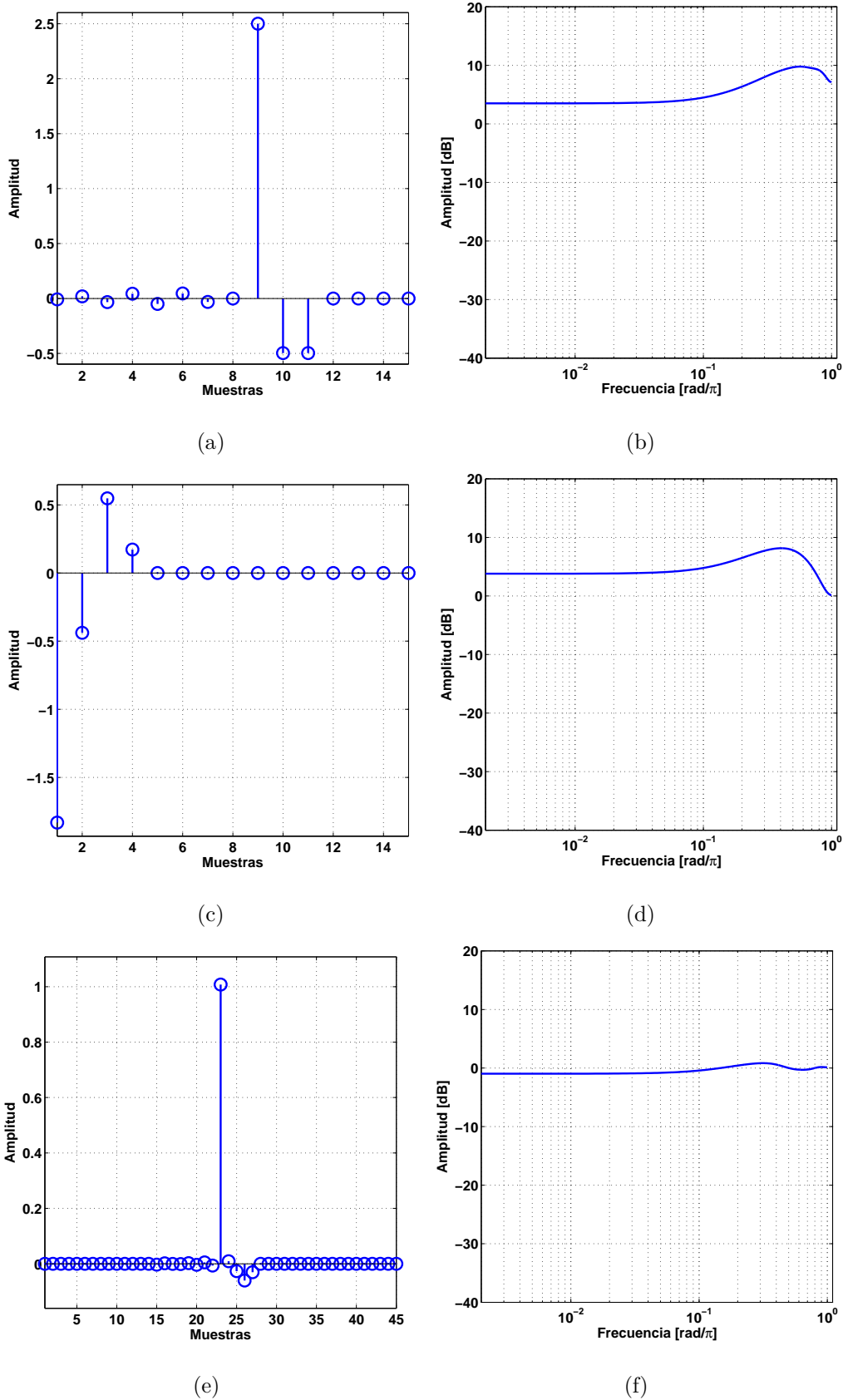


Figura 2.13: Comportamiento del ZF-DFE. (a) y (b) Respuesta del FFF. (c) y (d) Respuesta del FBF. (e) y (f) Respuesta del canal equivalente a la entrada del detector de umbral.

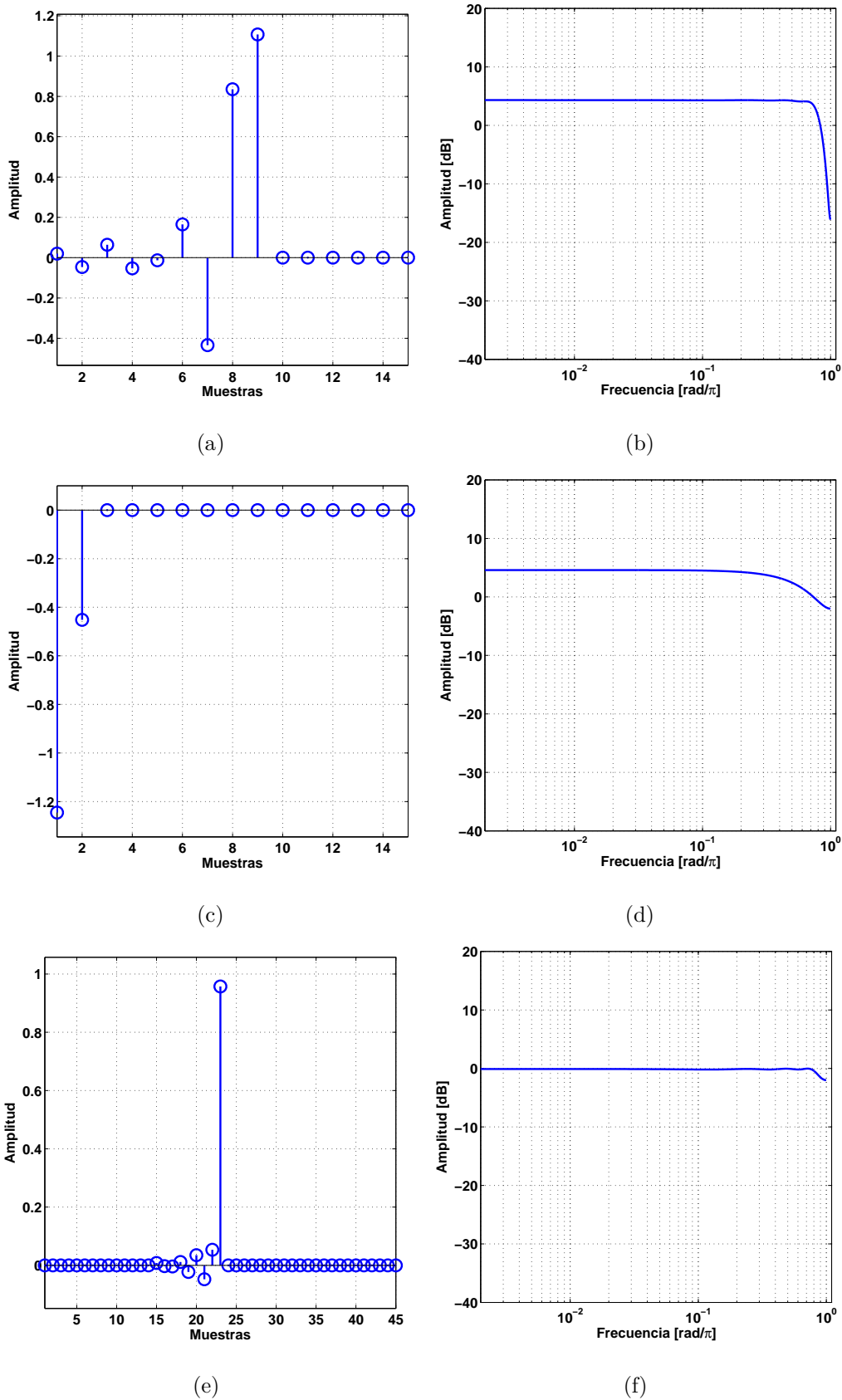


Figura 2.14: Comportamiento del MMSE-DFE. (a) y (b) Respuesta del FFF. (c) y (d) Respuesta del FBF. (e) y (f) Respuesta del canal equivalente a la entrada del detector de umbral.

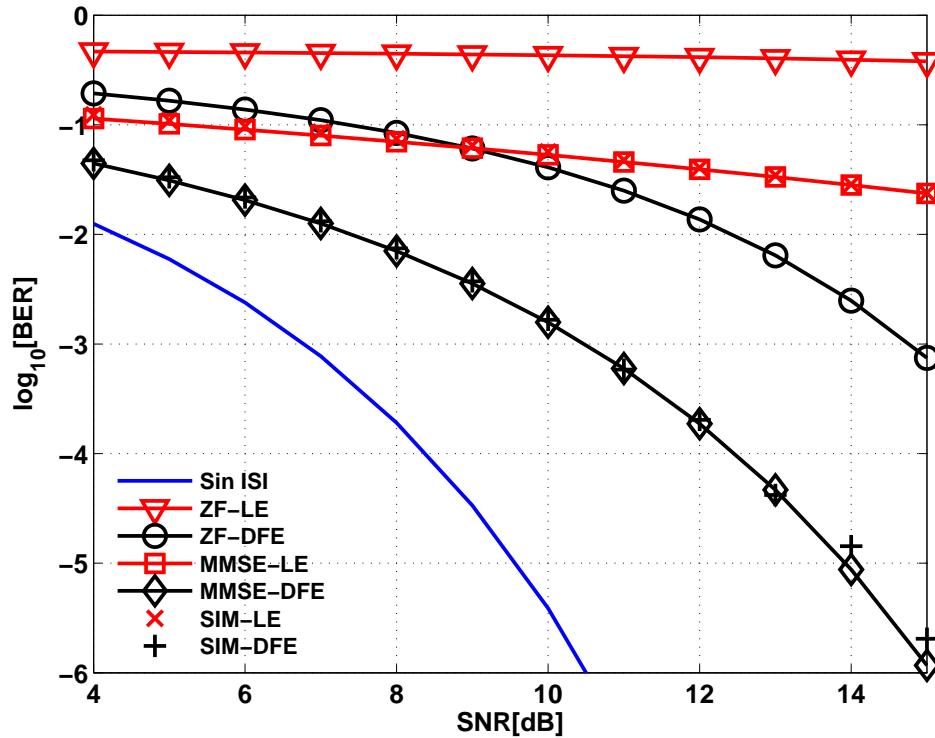


Figura 2.15: Curvas de BER para el canal de la Fig. 2.10. Comparación del desempeño del ecualizador LE y DFE para los criterios ZF y MSE.

## 2.6. Resultados Numéricos en Canales Exponenciales

En esta sección se analizará el desempeño del ecualizador lineal y DFE considerando los criterios ZF y MSE para canales exponenciales ( $p_k = \alpha^k \mu_k$ ). Tal como se estudiará en el siguiente capítulo, canales con estas características hacen que la complejidad de implementación en estructuras no lineales (como por ejemplo, el DFE y MLSD), crezcan exponencialmente con la memoria del canal.

Los filtros de los ecualizadores LE y DFE se configuran igual que en la Sección 2.5. Donde el número de coeficientes es  $N_f = 31$  para el LE y  $N_f = 15$  y  $N_b = 15$  para el DFE. En el análisis del desempeño del DFE, no se considera errores en las decisiones. Por último, las gráficas de comportamiento de los ecualizadores se realizan para una SNR de 15dB. La Fig. 2.16 muestra el canal equivalente en tiempo discreto y el espectro plegado del canal para un  $\alpha = 0,7$ . Este tipo de canal, es considerado en el resto de los capítulos de la presente Tesis doctoral.

Las Fig. 2.17-2.20 muestran el comportamiento de los ecualizadores considerando los

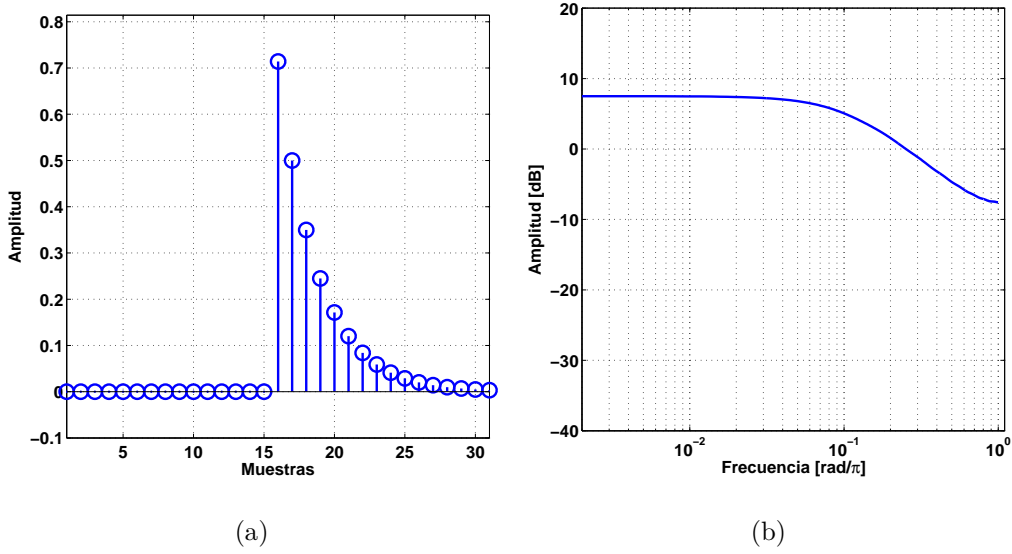


Figura 2.16: (a) *Canal Equivalente en Tiempo Discreto* ( $p_k = \alpha^k \mu_k$  con  $\alpha = 0,6/L = 30$ ). (b) *Espectro plegado del canal equivalente.*

criterios ZF y MSE. En todos los casos, se observa una mejora significativa del comportamiento de los ecualizadores en comparación con el canal evaluado en la sección anterior. Además, el comportamiento de los ecualizadores considerando los criterios ZF y MSE son similares. Todo esto se debe, a que el canal no presenta zonas del espectro cercanas a cero o nulas. Esta conclusión se observa en la Fig. 2.21. La mejora del desempeño del DFE en comparación con el LE, se hace evidente cuando se incrementa la memoria del canal. Las gráficas de la Fig. 2.22 muestran el desempeño del LE y DFE utilizando solamente el criterio MSE para canales exponenciales. El canal se configura con  $\alpha = 0,6/L = 22$  (Fig. 2.22(a)),  $\alpha = 0,82/L = 62$  (Fig. 2.22(b)),  $\alpha = 0,92/L = 122$  (Fig. 2.22(c)) y  $\alpha = 0,95/L = 202$  (Fig. 2.22(d)). El número de coeficiente del ecualizador lineal es  $N_f = 31$  para todos los casos y el número de coeficientes para el DFE es  $N_f = 15$  y  $N_b = [11, 31, 61, 101]$ , respectivamente.

En resumen, el DFE presenta un mejor desempeño que el LE considerando un canal exponencial. Es por esto, que se elige al DFE como técnica de ecualización para el análisis de la problemática que la presente Tesis doctoral busca resolver.

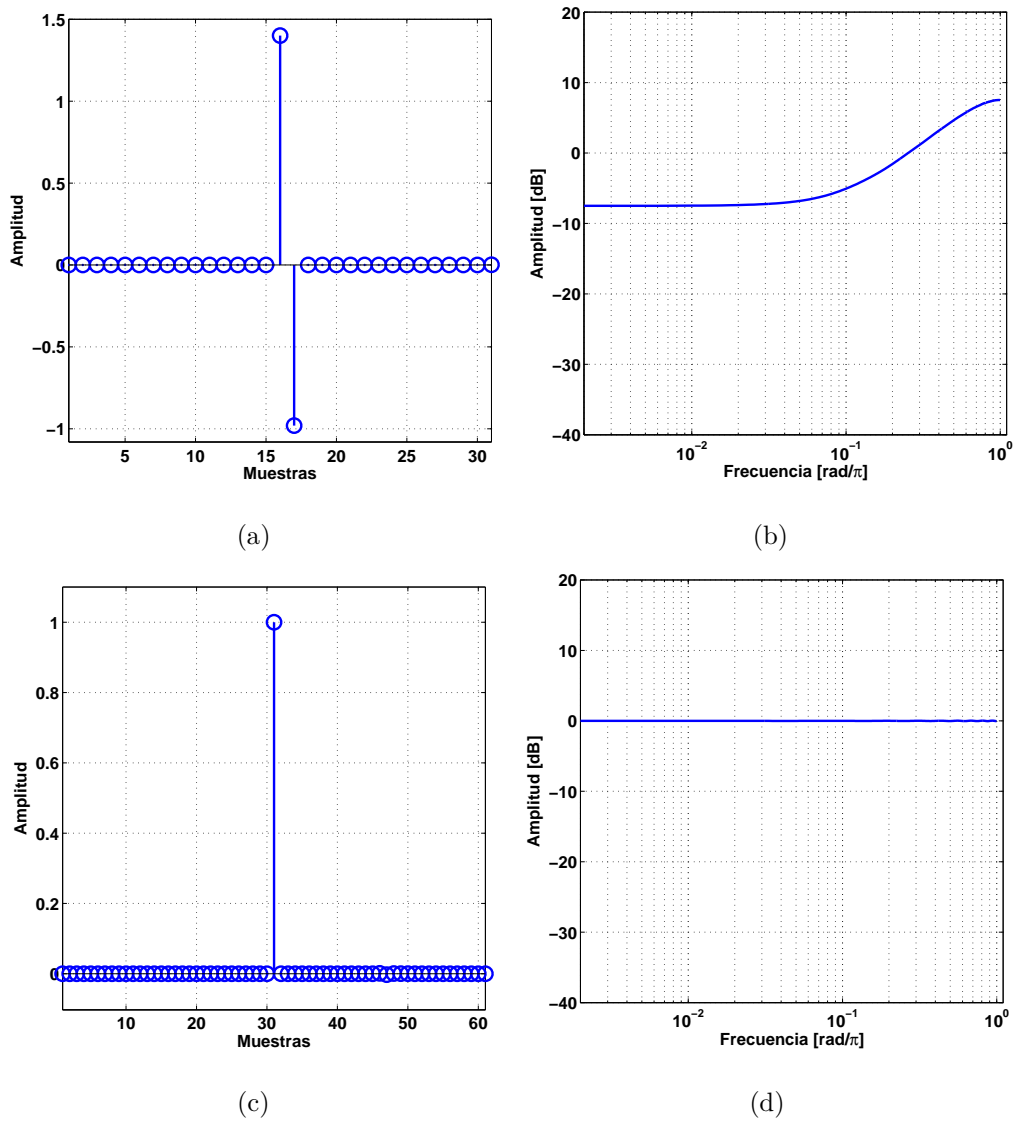


Figura 2.17: Comportamiento del ZF-LE. (a) y (b) Respuesta del LE. (c) y (d) Convulsión del canal equivalente con los coeficientes del ecualizador.



## 2.6. Resultados Numéricos en Canales Exponenciales

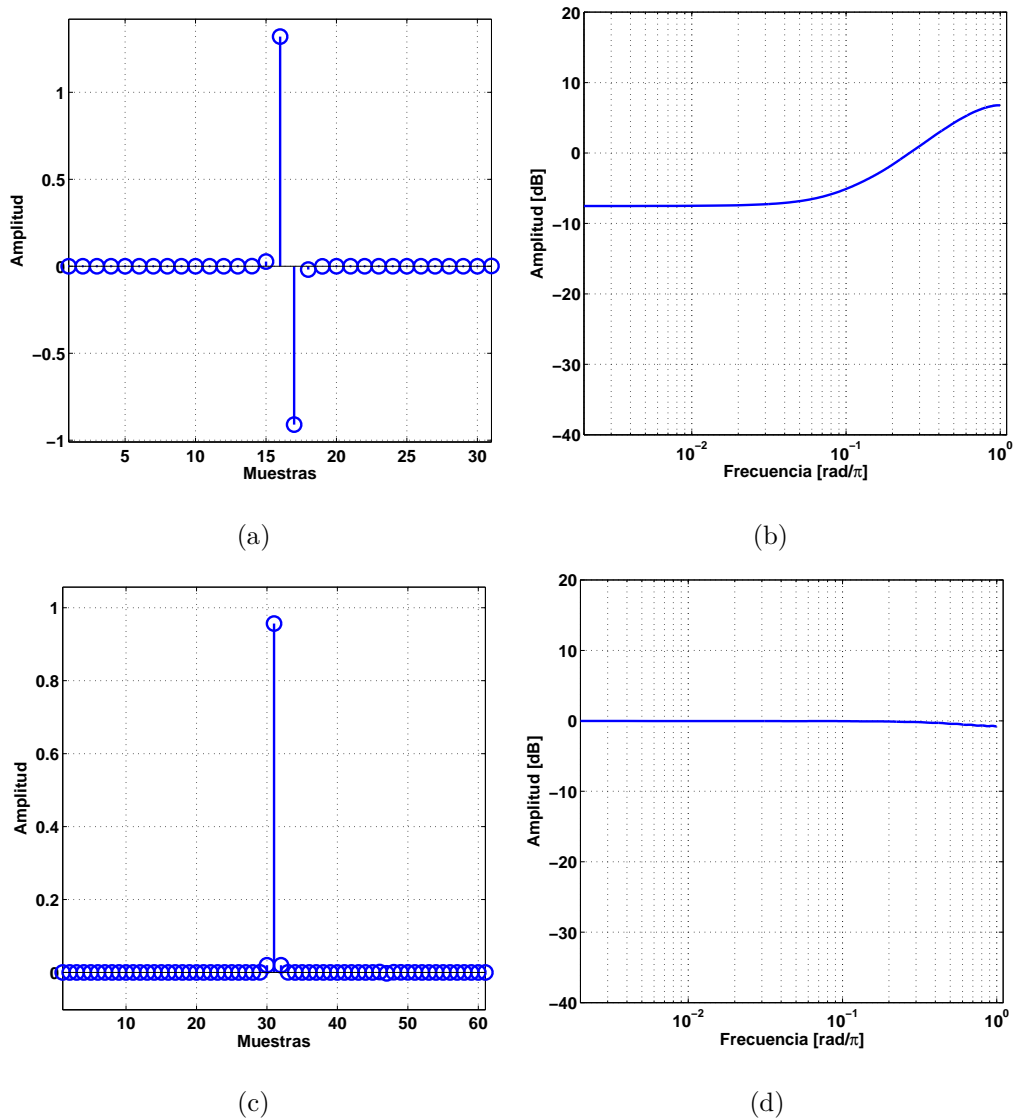


Figura 2.18: *Comportamiento del MMSE-LE. (a) y (b) Respuesta del LE. (c) y (d) Convolución del canal equivalente con los coeficientes del ecualizador.*

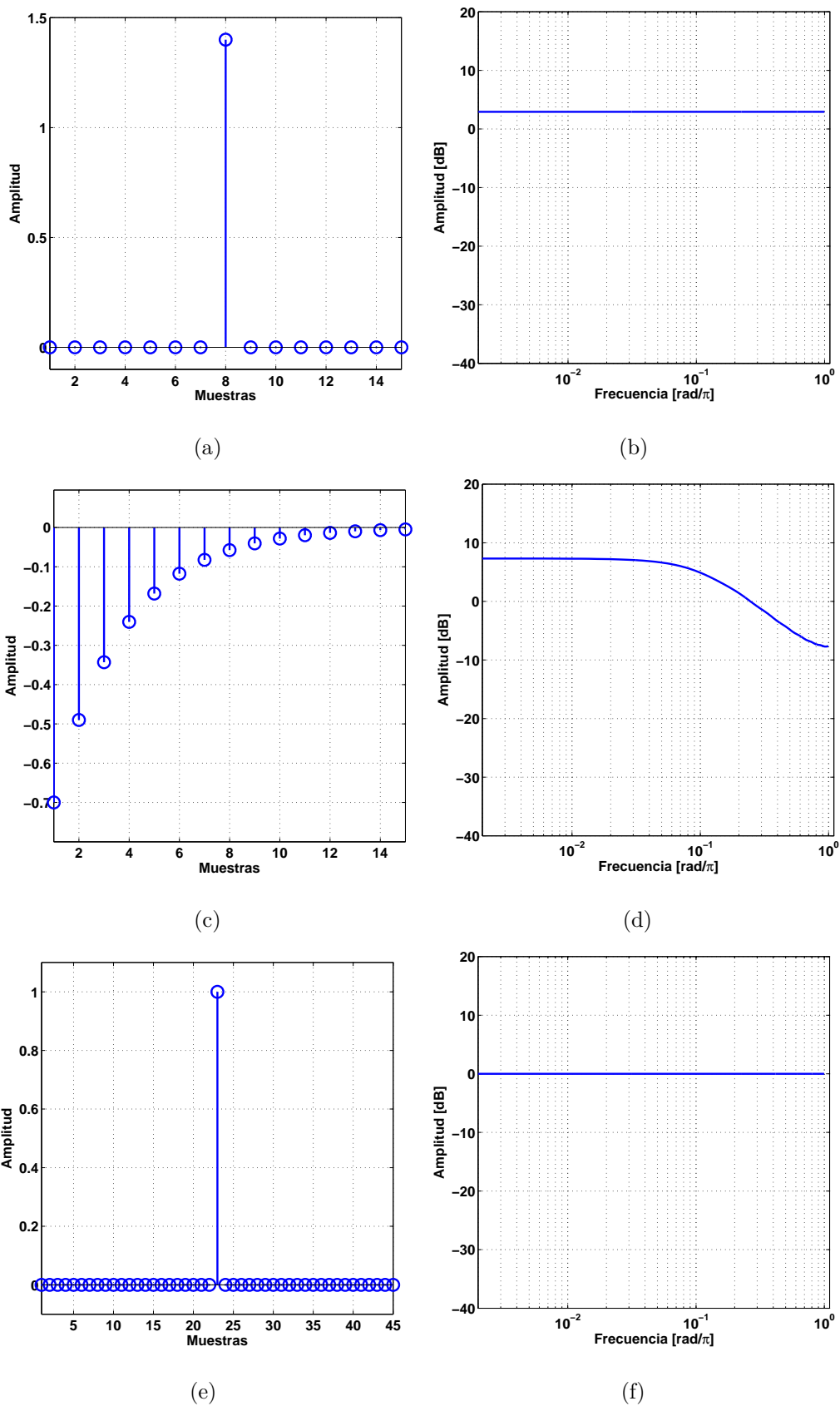


Figura 2.19: Comportamiento del ZF-DFE. (a) y (b) Respuesta del FFF. (c) y (d) Respuesta del FBF. (e) y (f) Respuesta del canal equivalente a la entrada del detector de umbral.

## 2.6. Resultados Numéricos en Canales Exponenciales

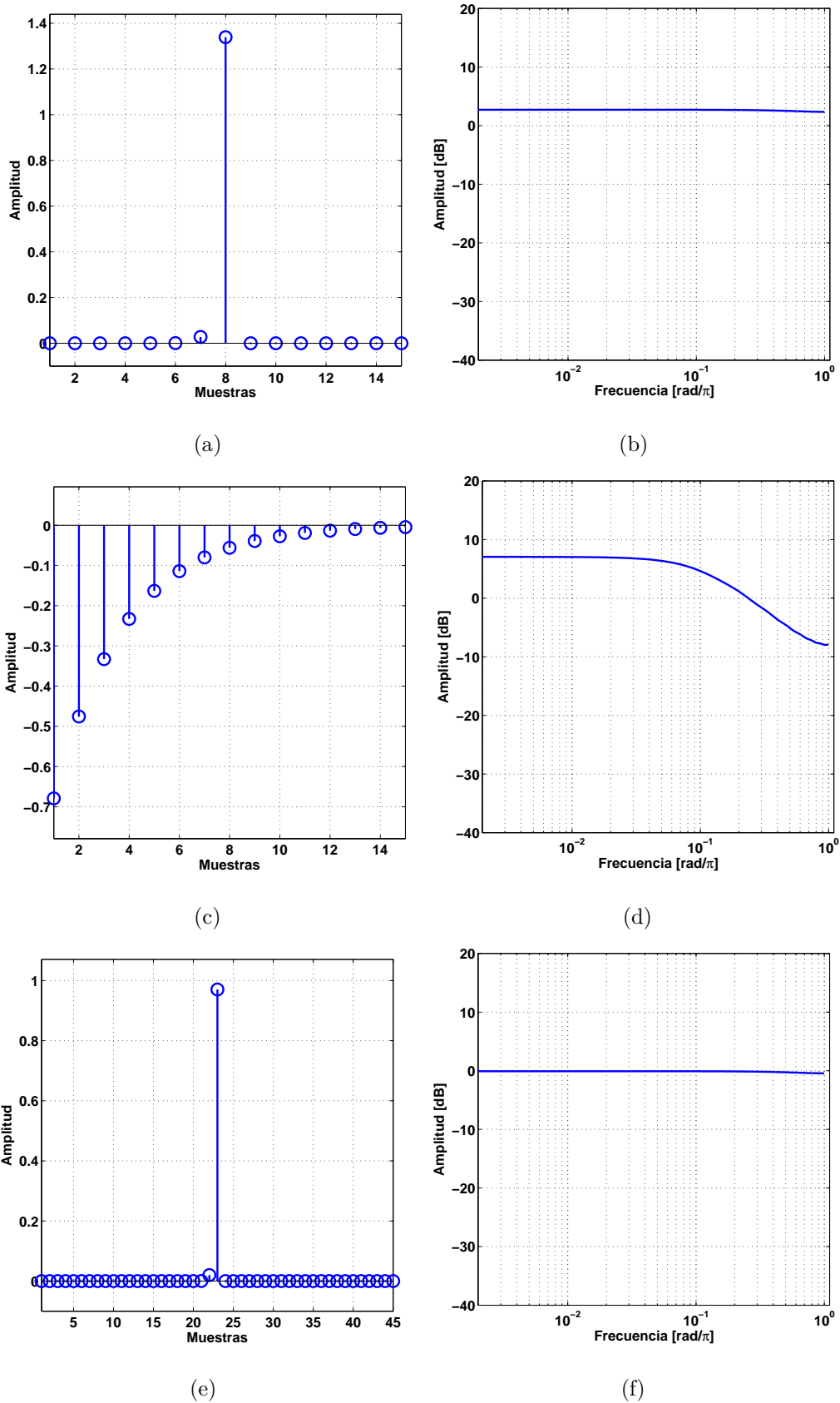


Figura 2.20: *Comportamiento del MMSE-DFE. (a) y (b) Respuesta del FFF. (c) y (d) Respuesta del FBF. (e) y (f) Respuesta del canal equivalente a la entrada del detector de umbral.*

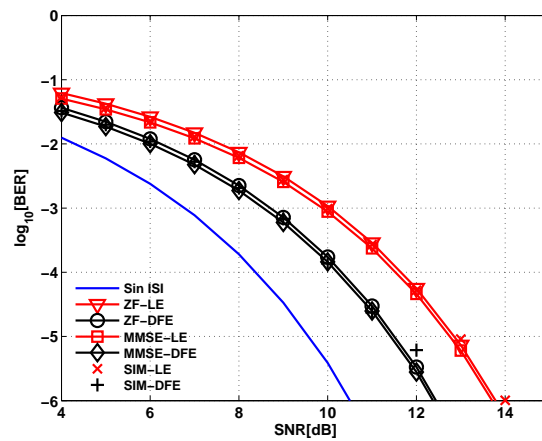


Figura 2.21: Curvas de BER para el canal de la Fig. 2.16. Comparación del desempeño del LE y DFE para los criterios ZF y MSE.

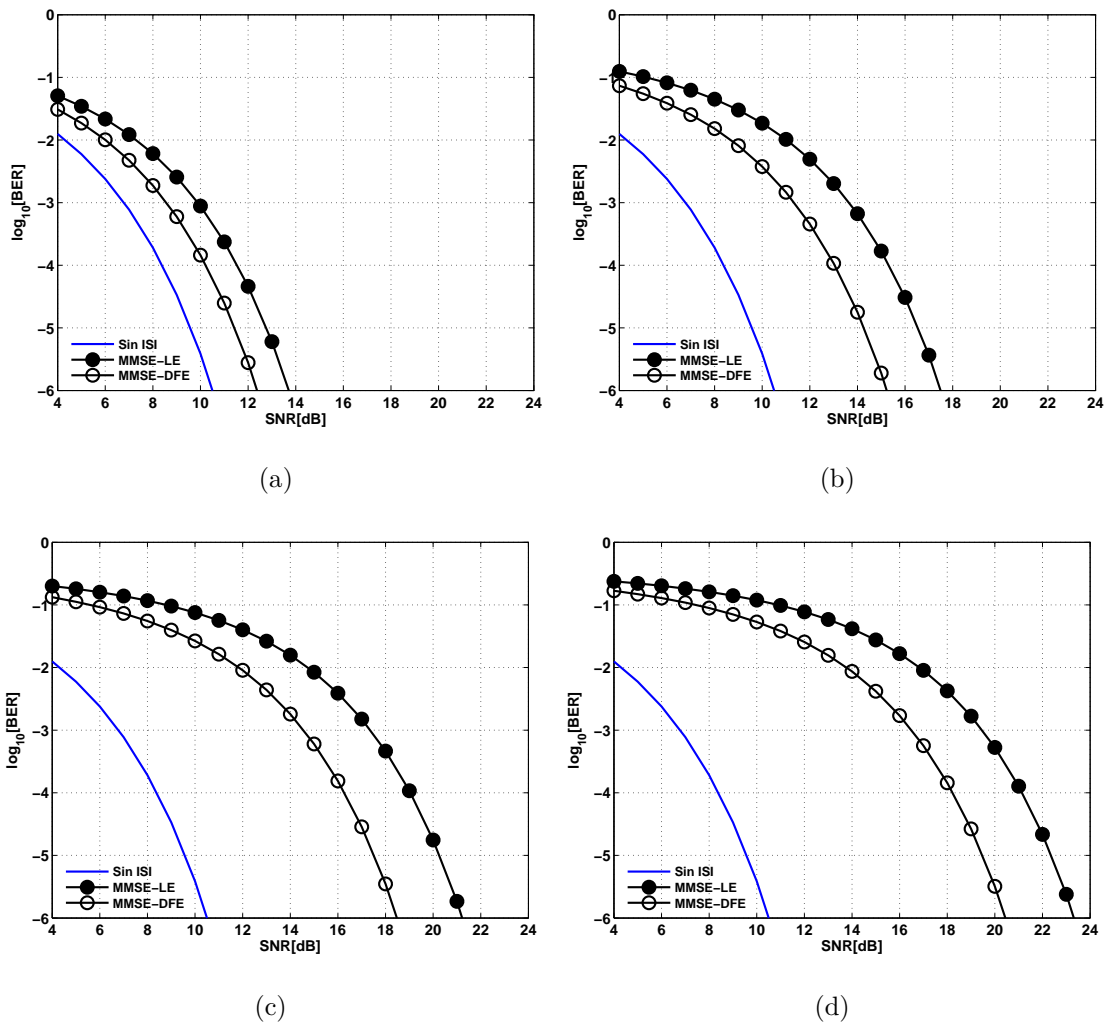


Figura 2.22: Comparación del desempeño del LE y DFE utilizando el criterio MSE. Se considera un canal exponencial  $p_k = \alpha^k \mu_k$  con (a)  $\alpha = 0,6/L = 22$ , (b)  $\alpha = 0,82/L = 62$ , (c)  $\alpha = 0,92/L = 122$  y (d)  $\alpha = 0,95/L = 202$ .

## **2.7. Conclusión Parcial**

En este capítulo se expusieron dos de las principales limitaciones para la transmisión digital de información de los canales, el ruido y la ISI. Se presentó dos estructuras ecualizadoras para compensar estos efectos del canal en transmisiones digitales. Se analizaron los criterios de forzado a cero y el error cuadrático medio para la estimación de los coeficientes óptimos del ecualizador. En base a estimaciones y simulaciones se verificó la reducción del ruido obtenido con el DFE. En consecuencia, resultó en una mejora significativa del desempeño para los canales cuya respuesta en frecuencia presenta valores cercanos a cero.

La principal motivación de los receptores obtenidos con estos criterios subóptimos es que los mismos poseen una complejidad de implementación significativamente inferior que la requerida por el receptor óptimo, como se hará evidente en esta Tesis doctoral. En aplicaciones de muy alta velocidad para canales con respuesta equivalente con un gran número de post-cursos, la implementación del DFE comienza a presentar limitaciones inherentes al lazo realimentado. Para solucionar este inconveniente se emplean técnicas de paralelización, tema que se comienza a desarrollar en el capítulo siguiente.



# CAPÍTULO

## 3

# ARQUITECTURAS DE ALTA VELOCIDAD DE ECUALIZADORES NO LINEALES

***Resumen:** El presente capítulo estudia la implementación de arquitecturas de ecualizadores no lineales de alta velocidad, analizando las metodologías de incremento de velocidad en sistemas realimentados. El desarrollo parte desde los conceptos de análisis de tiempo hasta la metodología de implementación de las técnicas “retiming”, segmentación y paralelización sobre este tipo de sistemas. Además, se presentan de manera resumida las arquitecturas paralelas propuestas del DFE comparando la complejidad de implementación en base a la longitud de la memoria del canal. Todos estos conceptos demuestran que existe la necesidad de emplear una nueva técnica de ecualización que elimine el cuello de botella generado por el lazo realimentado y sea de complejidad reducida.*

### 3.1. Introducción

La llegada de la integración a muy gran escala, ha reducido el costo de los dispositivos de hardware. Esto ha dado más flexibilidad a los diseñadores de sistemas, para implementar aplicaciones computacionalmente más complejas en pequeños dispositivos. Mayor rendimiento y altas velocidades de datos se han convertido en los objetivos de los desarrollos de hoy en día. Desde la perspectiva del hardware, la segmentación (*pipeline*), “retiming”<sup>1</sup> y procesamiento en paralelo son algunos métodos que ayudan al diseñador de hardware a alcanzar estos objetivos.

En términos generales, los sistemas de procesamiento de señales digitales se pueden clasificar como sistemas directos (*feedforward*) o realimentados (*feedback*). En los sistemas directos, los datos fluyen de la entrada a la salida y ningún valor en el sistema se alimenta de nuevo hacia la entrada. Los filtros de respuesta al impulso finita (*Finite Impulse Response - FIR*) son sistemas directos y son fundamentales para el procesamiento de señales. La mayoría de los algoritmos de procesamiento de señales, tales como la transformada rápida de Fourier (*Fast Fourier Transform - FFT*) y transformada discreta del coseno (*Discrete Cosine Transform - DCT*) son directos. En los sistemas directos, la sincronización se puede mejorar por segmentación (*pipeline*) de múltiples etapas en el diseño de hardware.

Sistemas recursivos, tales como los filtros de respuesta al impulso infinita (*Infinite Impulse Response - IIR*), también son muy utilizados en el procesamiento de señales digitales. Los algoritmos recursivos realimentados se utilizan, por ejemplo, para recuperación de sincronismo de símbolo (*Timing Recovery - TR*) y recuperación de sincronismo de portadora (*Carrier Recovery - CR*) en los receptores digitales. En estos sistemas, la implementación de técnicas como segmentación, “retiming” y procesamiento en paralelo, por lo general no es simple. Lo crítico de estos sistemas es la sincronización de las muestras, ya que consideran valores anteriores de salida para la estimación de la muestra actual.

Con la restricción impuesta por los osciladores, los cuales alcanzan frecuencias de varios GHz, es indispensable utilizar estrategias que permitan incrementar la velocidad en los receptores de comunicación, sin generar un incremento excesivo en la complejidad. Entre

---

<sup>1</sup>“Retiming” [24] es una técnica de transformación utilizada para cambiar las ubicaciones de elementos de retardo en un sistema sin afectar las características de entrada/salida del circuito.



las técnicas de ecualización conocidas que se expusieron en el *Capítulo 2*, el DFE posee una buena relación entre desempeño y complejidad para canales con una dispersión moderada. La dificultad de implementar el DFE en receptores de alta velocidad, radica en el lazo realimentado. Esto hace, que la aplicación de técnicas como segmentación, “retiming” y procesamiento en paralelo resultan complejas y en ocasiones no puedan ser utilizadas.

Este capítulo analiza los conceptos básicos de estas técnicas enfocando el desarrollo en la implementación del DFE en paralelo. Además, se plantean

- Las limitaciones de velocidad del DFE a consecuencia del lazo realimentado.
- La complejidad de implementación en paralelo aplicando técnicas tradicionales.
- La evaluación de técnicas alternativas propuestas por otros autores.
- El análisis de complejidad de los recursos utilizados.

Todos estos aspectos dejan a la luz los problemas que esta Tesis busca resolver.

## 3.2. Diseño Digital de Sistemas Realimentados

En el diseño digital de sistemas realimentados, la aplicación de técnicas que permitan incrementar las velocidades de procesamiento, son sumamente complejas. Esto se debe al cuello de botella que genera el lazo realimentado. Antes de abordar las arquitecturas paralelas del DFE que permiten incrementar la velocidad de cómputo, se revisarán los conceptos básicos de análisis de tiempos sobre este tipo de arquitecturas y las estrategias de implementación que rompen esta limitación.

### 3.2.1. Análisis de Tiempos

El análisis de tiempos en los sistemas digitales, principalmente en los sistemas recursivos, permite al diseñador conocer los límites del circuito implementado. El análisis consiste en contabilizar los tiempos de propagación sobre los diferentes componentes en un mismo camino. Definiendo como componentes a las operaciones de suma y producto, así como también a los multiplexores. Para explicar algunos de los conceptos más importantes relacionados con el análisis de tiempos en los diseños digitales realimentados, utilizaremos

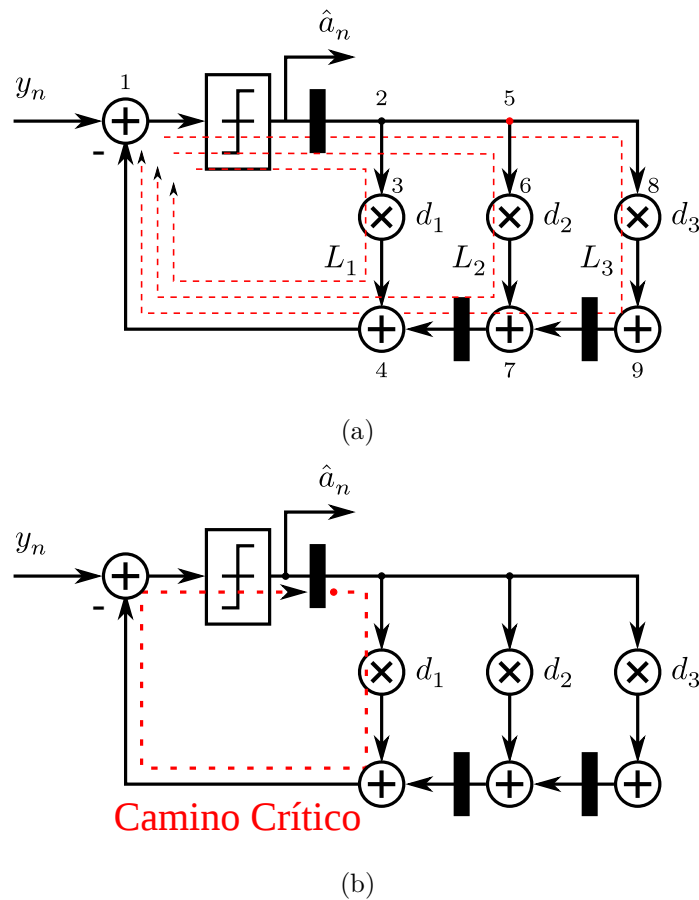


Figura 3.1: Ejemplo de implementación serial del DFE para tres (3) coeficientes del filtro realimentado. (a) Detalle de los lazos o bucles en el DFE. (b) En línea de trazo se resalta el camino crítico en el diseño.

como ejemplo el ecualizador DFE serial para una modulación 2-PAM mostrado en la Fig. 3.1. Además, por el esquema de modulación utilizado, reemplazamos los multiplicadores por multiplexores 2-a-1, considerando que se tiene almacenado el valor positivo y negativo de los coeficientes. También, en las estimaciones de los tiempos de propagación no se incluyen los retardos de pista ni el detector de umbral, ya que se busca un análisis mas general sobre el diseño. Con estas condiciones, los ejemplos de tiempos que se ilustran a continuación, son planteados tomando como base un proceso CMOS de  $28nm$ , donde los tiempos aproximados de cada componente son  $T_{mux} \approx 0,05ns$  para el multiplexor 2-a-1 y  $T_{add} \approx 0,10ns$  para el sumador.

El esquema de la Fig. 3.1(a) se compone de tres (3) coeficientes del filtro realimentado, donde la distribución de los componentes generan tres lazos  $L_1$ ,  $L_2$  y  $L_3$  (líneas de trazo de color rojo). Un lazo o bucle se define como el camino que comienza y termina en el

mismo nodo. En la Fig. 3.1(a), por ejemplo,  $L_1$  consta de los nodos  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ . Una métrica para determinar el largo del lazo es el límite de lazo (*Loop Bound - LB*) que se define como  $T_i/D_i$ , donde  $T_i$  es el tiempo de cálculo del lazo considerando todos los retardos de los componentes en el camino y  $D_i$  es el número de registros en el lazo. Para el circuito de la Fig. 3.1(a), el límite de lazo en cada bucle es

$$LB_{L_1} = \frac{T_1}{D_1} = \frac{T_{mux} + 2T_{add}}{1} = 0,25ns \quad (3.1)$$

$$LB_{L_2} = \frac{T_2}{D_2} = \frac{T_{mux} + 3T_{add}}{2} = 0,175ns \quad (3.2)$$

$$LB_{L_3} = \frac{T_3}{D_3} = \frac{T_{mux} + 4T_{add}}{3} = 0,15ns \quad (3.3)$$

El límite de iteración (*Iteration Bound - IB*), es el máximo límite de lazo entre todos los lazos del sistema. En particular, para el ejemplo analizado

$$IB = \max\{LB_{L_1}, LB_{L_2}, LB_{L_3}\} = LB_{L_1} \quad (3.4)$$

$$= \frac{T_{mux} + 2T_{add}}{1} = 0,25ns \quad (3.5)$$

Bajo estas consideraciones, decimos que el lazo crítico (*critical loop*) del diseño es  $L_1$ . Por último, uno de los conceptos más importantes para los diseñadores digitales es el camino crítico (*Critical Path - CP*). El CP se define como el camino con más retardo de tiempo sin considerar registros en ese camino. Es decir, el CP se lo analiza entre puerto de entrada y registro, entre registro y registro o entre registro y puerto de salida. Cualquiera sea el caso, solamente computa el tiempo de propagación de la lógica. En la Fig. 3.1(b), se muestra el CP para el DFE serial, el cual inicia en el registro y termina en el mismo registro. Puntualmente, considera un multiplexor y dos sumadores, donde el tiempo calculado para este camino es  $T_{CP} \approx 0,25ns$ . Notar que el límite de iteración es el mismo que el camino crítico, esto es una característica de los sistemas recursivos.

Como resultado de este análisis de tiempos conceptual, la frecuencia máxima del reloj con la que puede operar el DFE serial, es aproximadamente  $4GHz$ . Esta limitación en la frecuencia máxima de reloj, para aplicaciones de alta velocidad ( $> 10GHz$ ), obliga al diseñador digital a utilizar técnicas como segmentación, “retiming” y paralelización para alcanzar sus objetivos. A continuación se desarrollan estos conceptos, que son las bases para el diseño digital de receptores de alta velocidad.

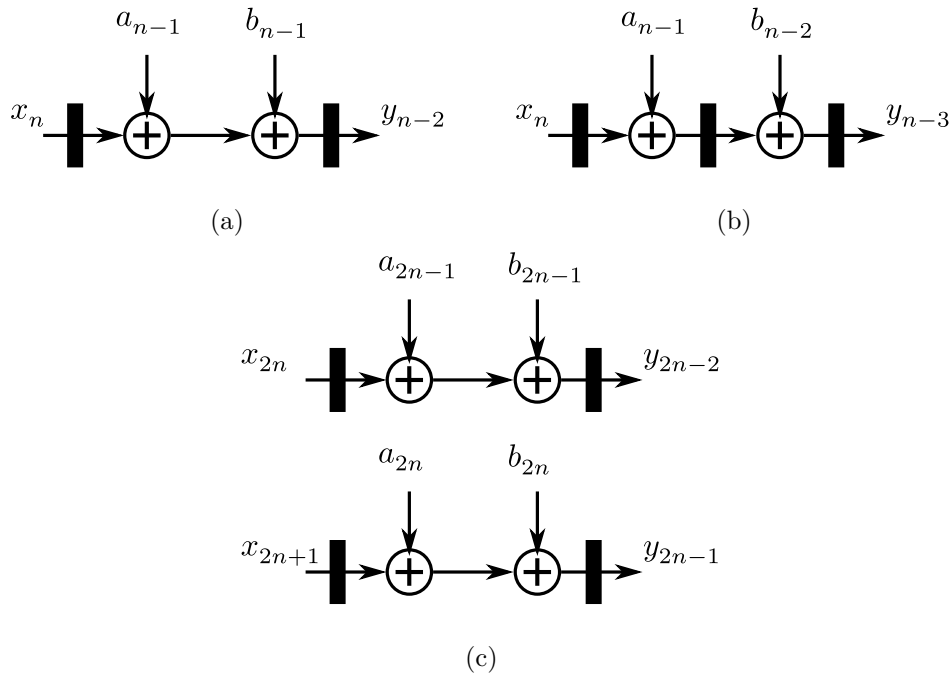


Figura 3.2: Esquemas de transformación de segmentación y paralelismo. (a) Camino de dato. (b) Estructura de segmentación de dos (2) niveles. (c) Estructura en procesamiento en paralelo de dos (2) niveles.

### 3.2.2. Segmentación y Paralelismo

La transformación de segmentación (*pipelining transformation*) [24] permite reducir el camino crítico (*critical path*). Esto es aprovechado para incrementar la frecuencia del reloj, la velocidad de muestreo o reducir la potencia del sistema a la misma velocidad. En el procesamiento en paralelo [24], múltiples salidas son calculadas de manera simultánea en un mismo período de reloj. Por lo tanto, la velocidad de muestreo efectiva se incrementa por el nivel de paralelismo. Similar a la segmentación, el procesamiento en paralelo también se puede utilizar para reducir el consumo. En la Fig. 3.2 se muestran dos ejemplos de implementación de estas técnicas. La implementación del procesamiento en segmentación se observa en la Fig. 3.2(b), donde  $z^{-1}$  representa un registro (*register*) o retardo (*delay*). Este es utilizado para reducir el camino crítico. La Fig. 3.2(c) representa un procesamiento en paralelo, donde se duplica el hardware con el mismo objetivo.

El camino crítico a través de una nube combinacional en un sistema de alimentación directa, se puede cambiar por la adición de registros usando el criterio de segmentación. En un sistema de alimentación directa la salida depende sólo de muestras de entrada actuales y anteriores. En estas estructuras los registros solo añaden latencia. Si se agregan

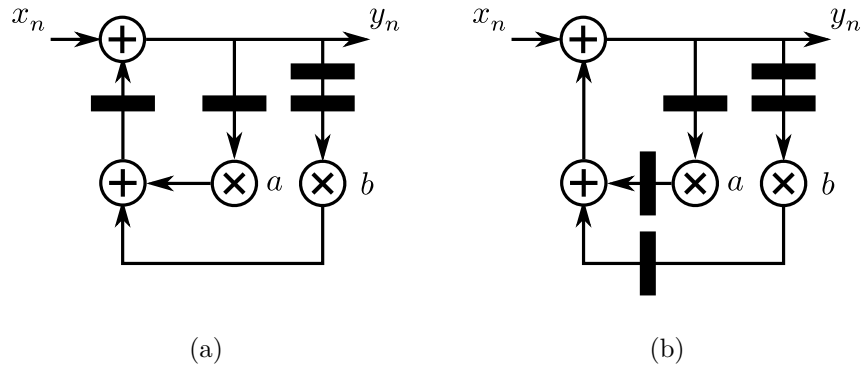


Figura 3.3: Ejemplo de reducción del camino crítico utilizando “retiming”. (a) Camino de dato original. (b) “Retiming” de los registros.

registros ( $D$ ), la función de transferencia del sistema se multiplica por  $z^{-D}$ . En general, en un sistema con  $D$  etapas de segmentación, el número de elementos de retardo desde cualquier camino de entrada a salida es  $D - 1$  mayor que en la misma ruta de acceso en la lógica original. En un sistema realimentado, los registros no pueden ser simplemente agregados, ya que los retardos modifican la función de transferencia del sistema, dando como resultado un cambio en el orden de la ecuación en diferencia.

Es de interés señalar, que las técnicas de procesamiento paralelo y segmentación son duales una de la otra, y si un cómputo puede ser segmentado, también puede ser procesado en paralelo. Ambas técnicas explotan la concurrencia en el cálculo de diferentes maneras. Por un lado, un conjunto de cálculos se ejecutan intercalados en forma independiente en un procesamiento segmentado, al mismo tiempo que estos se calculan utilizando hardware duplicado en el modo de procesamiento en paralelo.

### 3.2.3. Retiming

“Retiming” [24] es una técnica de transformación utilizada para cambiar las ubicaciones de elementos de retardo en un sistema sin afectar las características de entrada/salida del circuito. La segmentación y “retiming” son dos aspectos diferentes del diseño digital. En la segmentación, se agregan registros adicionales que cambian la función de transferencia del sistema. Mientras que en el “retiming”, los registros son reubicados en un diseño. Por lo general, el “retiming” tiene varios fines, donde se destacan

- **Reducir el camino crítico:** Se puede utilizar para aumentar la frecuencia de reloj de un circuito mediante la reducción del tiempo de cálculo del camino crítico.

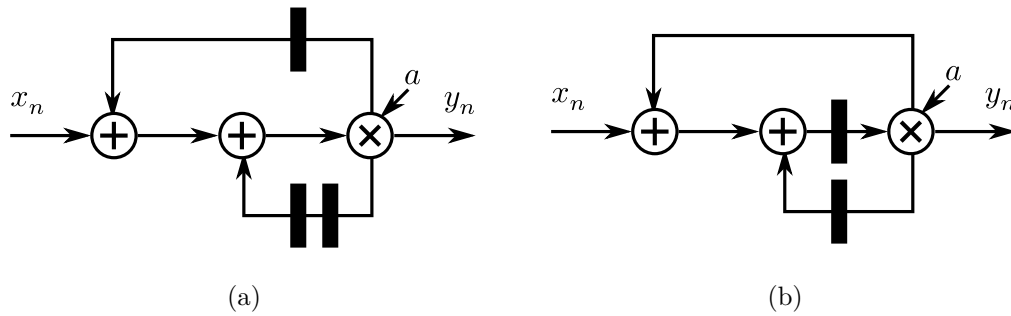


Figura 3.4: Ejemplo de reducción del camino crítico y reducción de complejidad utilizando “retiming”. (a) Camino de dato original. (b) Se modifica la posición de los registros reduciendo la complejidad.

Recordar que el camino crítico se define como la ruta con el tiempo de cálculo más largo entre todos los caminos que contienen cero retrasos, y el tiempo de cálculo del camino crítico es la cota inferior del período de reloj del circuito.

El camino crítico del filtro en la Fig. 3.3(a) pasa a través de un multiplicador ( $T_{mult} \approx 0,20ns$ ) y un sumador ( $T_{add} \approx 0,10ns$ ). Considerando el tiempo de propagación de estas operaciones para una tecnología CMOS de 28nm, el retardo del camino crítico es  $0,30ns$ . Aplicando “retiming”, en la Fig. 3.3(b) el camino crítico atraviesa dos (2) sumadores, donde el retardo ahora es  $0,20ns$ . Como resultado, este filtro puede operar con una frecuencia de reloj cercano a  $5GHz$ .

- **Reducir el número de registros:** Se pueden mover los registros existentes en el diseño con el objetivo de eliminar elementos redundantes. La Fig. 3.4(a) muestra un filtro con dos registros en el lazo realimentado inferior y un registro en el lazo superior. El “retiming” busca minimizar el uso de registros, reubicándolos en el filtro, tal como se observa en la Fig. 3.4(b). El resultado es una reducción en la complejidad del diseño.
- **Minimizar el uso de energía:** Se puede utilizar para reducir el consumo de energía de un circuito mediante la reducción de conmutación, que puede conducir a la disipación de potencia dinámica en circuitos estáticos CMOS. La colocación de los registros en las entradas de los nodos con grandes capacidades puede reducir las actividades de conmutación en estos nodos, que pueden conducir a soluciones de bajo consumo.

- **Aumentar la capacidad de prueba:** La adecuada colocación de los registros facilita la verificación del diseño [55, 56].

### 3.3. Arquitectura Paralela Directa del DFE

La implementación en hardware del DFE para alta velocidad se puede lograr mediante diferentes técnicas [24], algunas de las cuales se describieron en la sección anterior. La segmentación se emplea con éxito para aumentar la velocidad de cálculo en los diseños de ecualización [23]. Otro enfoque, para lograr incrementar la velocidad del receptor, es la implementación del ecualizador en paralelo [57]. Para sistemas directos (*forward*) el aprovechamiento de la segmentación y el procesamiento paralelo para aplicaciones de alta velocidad es sencillo. Sin embargo, para sistemas recursivos, como es el caso del DFE, no son fácilmente implementables debido al lazo realimentado. Para un filtro realimentado, el enfoque “retiming” se puede utilizar para mover los elementos de retardo de un camino más corto a una trayectoria más larga en el lazo. Sin embargo, el enfoque del “retiming” no puede alcanzar el límite de iteración<sup>2</sup> en la mayoría de los casos. A fin de lograr el límite de iteración, podemos desenrollar el lazo, que se conoce como el esquema de despliegue (*unfolding scheme*) [24], y luego aplicar el enfoque “retiming” a la arquitectura VLSI desplegada.

Para obtener la salida del DFE en paralelo, primero debemos considerar la expresión de un sistema simple-entrada simple-salida (*Single-Input Single-Output - SISO*). Planteando la salida del detector de umbral del DFE considerando solamente el filtro realimentando para distintos instantes de tiempo, obtenemos

---

<sup>2</sup>Recordemos que el límite de iteración (*Iteration Bound - IB*) (ver Sección 3.2) se define como el máximo límite de lazo entre todos los lazos del sistema. Donde el límite de lazo (*Loop Bound - LB*) es el tiempo de cálculo del lazo considerando todos los retardos de los componentes en el camino y el número de registros en el lazo.

$$\hat{a}_0 = \mathcal{Q} \left( y_0 - \sum_{m=1}^{N_b} \hat{a}_{0-m} d_m \right) \quad (3.6)$$

$$\hat{a}_1 = \mathcal{Q} \left( y_1 - \sum_{m=1}^{N_b} \hat{a}_{1-m} d_m \right) \quad (3.7)$$

$$\hat{a}_2 = \mathcal{Q} \left( y_2 - \sum_{m=1}^{N_b} \hat{a}_{2-m} d_m \right) \quad (3.8)$$

$$\begin{array}{ccc} \vdots & \vdots & \\ \hat{a}_k & = \mathcal{Q} \left( y_k - \sum_{m=1}^{N_b} \hat{a}_{k-m} d_m \right) & (3.9) \end{array}$$

donde  $\hat{a}_{k-m} = 0$  para  $k < m$ . Para obtener la estructura en paralelo, el sistema SISO debe ser convertido a un sistema de múltiple-entrada múltiple-salida (*Multiple-Input Multiple-Output - MIMO*). El DFE en paralelo queda definido como

$$\hat{a}_{n+0} = \mathcal{Q} \left( y_{n+0} - \sum_{m=1}^{N_b} \hat{a}_{n+0-m} d_m \right) \quad (3.10)$$

$$\hat{a}_{n+1} = \mathcal{Q} \left( y_{n+1} - \sum_{m=1}^{N_b} \hat{a}_{n+1-m} d_m \right) \quad (3.11)$$

$$\hat{a}_{n+2} = \mathcal{Q} \left( y_{n+2} - \sum_{m=1}^{N_b} \hat{a}_{n+2-m} d_m \right) \quad (3.12)$$

$$\begin{array}{ccc} \vdots & \vdots & \\ \hat{a}_{n+l} & = \mathcal{Q} \left( y_{n+l} - \sum_{m=1}^{N_b} \hat{a}_{n+l-m} d_m \right) & (3.13) \end{array}$$

donde  $l$  identifica cada salida para  $l = 0, 1, 2, \dots, P-1$  y  $n$  el tiempo que considera las  $P$  nuevas muestras ( $n = 0P, 1P, 2P, \dots$ ). Tomando a  $k$  como los ciclos de reloj ( $k = 0, 1, \dots$ ),  $n$  se reescribe como  $n = Pk$ . Por lo tanto, el símbolo detectado en cada salida del DFE paralelo para cada ciclo de reloj es

$$\hat{a}_{Pk+l} = \mathcal{Q} \left( y_{Pk+l} - \sum_{m=1}^{N_b} \hat{a}_{Pk+l-m} d_m \right) \quad \begin{cases} l = 0, 1, \dots, P-1 \\ k = 0, 1, \dots \end{cases} \quad (3.14)$$

La Ec. (3.14) deja en evidencia que, para un instante  $k$ , un símbolo detectado en cualquier salida del paralelismo depende de otras salidas en el mismo instante de tiempo. Esto genera la concatenación de operaciones de suma y multiplicación, incrementando el camino



### 3.3. Arquitectura Paralela Directa del DFE

---

crítico. En las siguientes expresiones examinamos el caso para un DFE paralelo con  $P = 4$  y  $N_b = 3$ . Los símbolos detectados a la salida del DFE en cada instante de tiempo quedan definidos por

$$\hat{a}_{4k+0} = \mathcal{Q}(y_{4k+0} - \hat{a}_{4k-1}d_1 - \hat{a}_{4k-2}d_2 - \hat{a}_{4k-3}d_3) \quad (3.15)$$

$$\hat{a}_{4k+1} = \mathcal{Q}(y_{4k+1} - \hat{a}_{4k+0}d_1 - \hat{a}_{4k-1}d_2 - \hat{a}_{4k-2}d_3) \quad (3.16)$$

$$\hat{a}_{4k+2} = \mathcal{Q}(y_{4k+2} - \hat{a}_{4k+1}d_1 - \hat{a}_{4k+0}d_2 - \hat{a}_{4k-1}d_3) \quad (3.17)$$

$$\hat{a}_{4k+3} = \mathcal{Q}(y_{4k+3} - \hat{a}_{4k+2}d_1 - \hat{a}_{4k+1}d_2 - \hat{a}_{4k+0}d_3) \quad (3.18)$$

La Fig. 3.5 representa las ecuaciones anteriores, donde cada salida define un módulo (DFE0, DFE1, DFE2, DFE3). Estos módulos representan los filtros realimentados del DFE. Además, en la Fig. 3.5, se remarca uno de los caminos críticos en línea de trazo de color rojo. Este inicia en el registro del módulo DFE0 y finaliza en la salida del módulo DFE3. Para estimar el camino crítico del DFE, se consideran las mismas restricciones planteadas en la *Sección 3.2*. Donde se emplea una modulación 2-PAM y los multiplicadores son reemplazados por multiplexores 2-a-1 (considerando que se tiene almacenado el valor positivo y negativo de los coeficientes). Con estas condiciones, el retardo en el camino crítico es

$$T_{CP} = 2(T_{mux} + N_b T_{add}) = 2(0,05ns + 3 * 0,10ns) = 0,70ns \quad (3.19)$$

El valor estimado del camino crítico determina, que para la arquitectura planteada en la Fig. 3.5, la frecuencia máxima del reloj debe ser  $\approx 1,4GHz$ . Con esta frecuencia y un paralelismo de  $P = 4$ , la tasa máxima de datos alcanzable es de  $4,4GS/s$ .

Para canales muy dispersivos (como los discutidos en el *Capítulo 2*), donde el número de coeficientes en el filtro realimentado del DFE es superior a  $N_b > 10$ , el camino crítico pasa a ser un factor limitante para alcanzar altas velocidades ( $> 10GS/s$ ) en el receptor. Por ejemplo, en la Fig. 3.6 se observan dos curvas que representan la frecuencia máxima del reloj (azul), que puede utilizar la arquitectura paralela del DFE condicionada al camino crítico y el paralelismo (verde), necesario para alcanzar una tasa máxima de datos de  $40GS/s$  en función del número de coeficientes del filtro realimentado. Obsérvese, que para valores de  $N_b > 10$ , el paralelismo necesario es superior a  $P > 100$ , haciendo que la

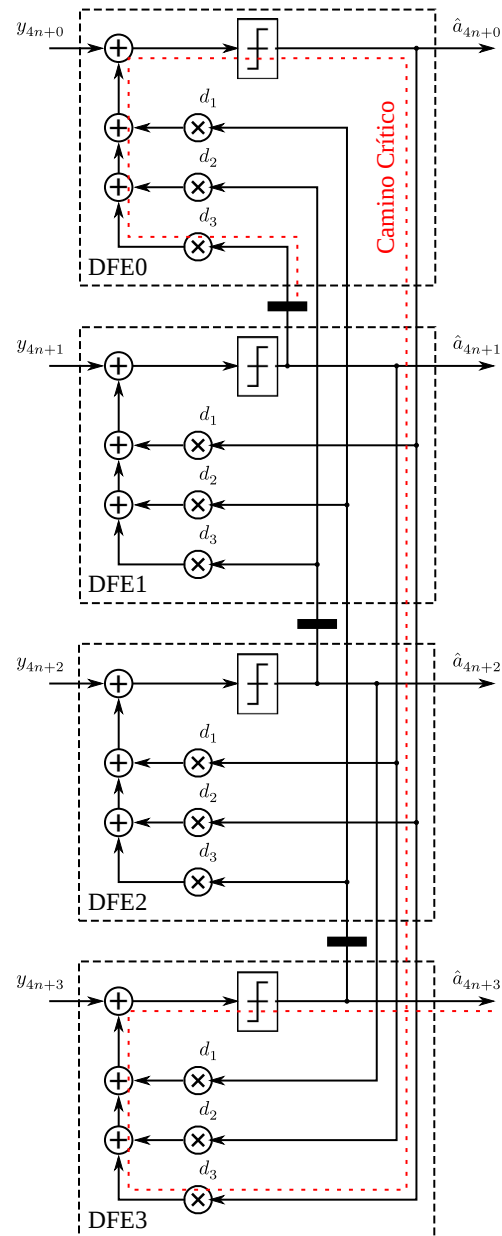


Figura 3.5: Diagrama detallado de la arquitectura del DFE en paralelo para  $P = 4$  y  $N_b = 3$ . En línea de trazo de color roja se marca uno de los caminos críticos.

implementación directa del DFE en paralelo sea muy compleja. Este análisis nos muestra la necesidad de buscar técnicas que resuelvan el camino crítico del DFE en paralelo para canales muy dispersivos. Tal como se desarrollará en la sección siguiente, existen técnicas de implementación en alta velocidad de cancelación no lineal que resuelven esta problemática, a costa de tener un crecimiento exponencial en la complejidad.

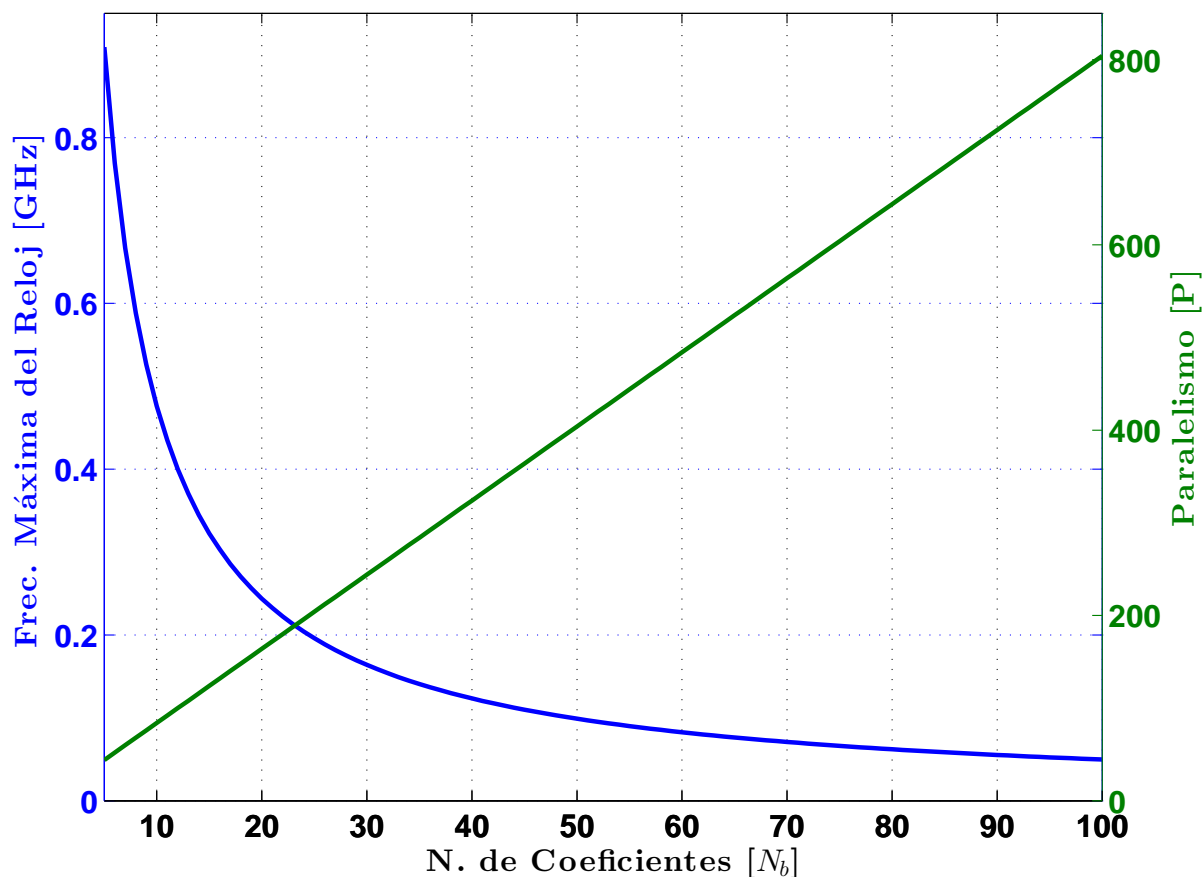


Figura 3.6: Curvas de frecuencia máxima del reloj y paralelismo en función del número de coeficientes del filtro realimentado del DFE para obtener una velocidad máxima de 40GS/s. La restricción de la frecuencia máxima del reloj es limitada por el camino crítico.

### 3.4. Técnicas para Implementación en Alta Velocidad de Cancelación No Lineal

La necesidad de implementar el DFE en receptores de alta velocidad, ha llevado a los investigadores a proponer diferentes técnicas que resuelven el cuello de botella generado por el lazo realimentado. Como se discutirá a continuación, la complejidad de estas alternativas crece exponencialmente con la memoria del canal. Esto hace que para canales con gran memoria ( $L \gg 1$ ), la implementación de estas técnicas no sea factible.

#### 3.4.1. Mirar Adelante

Mirar adelante (*look-ahead*) [22] es una de las técnicas más utilizadas para implementar la transformación de segmentación en los filtros IIR. Consiste en calcular todas las posibles

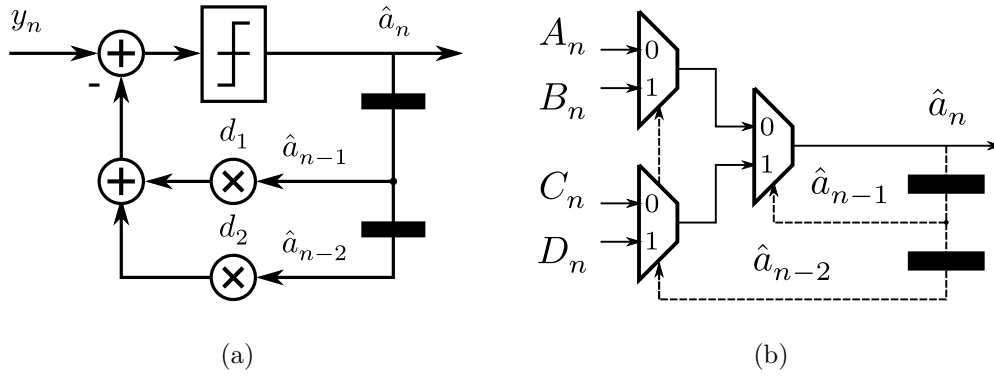


Figura 3.7: (a) Estructura tradicional del DFE considerando dos (2) coeficientes en el filtro realimentado. (b) Transformación del DFE aplicando la técnica mirar adelante.

decisiones futuras y seleccionar una de ellas a partir de decisiones pasadas.

Para comprender el concepto, se considera un DFE con dos (2) coeficientes en el filtro realimentado (Fig. 3.7(a)). La expresión que caracteriza el diseño es

$$\hat{a}_n = \mathcal{Q}(y_n - \hat{a}_{n-1}d_1 - \hat{a}_{n-2}d_2), \quad (3.20)$$

donde  $\hat{a}_n$  es la decisión a la salida del detector de umbral,  $y_n$  es la muestra recibida,  $d_1$  y  $d_2$  son los coeficientes del filtro realimentado y  $\mathcal{Q}(\cdot)$  representa el detector de umbral ( $\pm 1$ ). La expresión de la Ec. (3.20), se reformula contemplando todas las posibles salidas pre-calculadas como [22]

$$\hat{a}_n = A_n\hat{a}_{n-2}\hat{a}_{n-1} + B_n\bar{\hat{a}}_{n-2}\hat{a}_{n-1} + C_n\hat{a}_{n-2}\bar{\hat{a}}_{n-1} + D_n\bar{\hat{a}}_{n-2}\bar{\hat{a}}_{n-1} \quad (3.21)$$

donde  $\bar{\hat{a}}_n$  es el complemento de las decisiones que controlan los multiplexores y  $A_n, B_n, C_n$  y  $D_n$  son todas las posibles nuevas decisiones calculadas con la muestra de entrada actual definidas como

$$A_n = \mathcal{Q}(y_n - d_1 - d_2) \quad B_n = \mathcal{Q}(y_n - d_1 + d_2) \quad (3.22)$$

$$C_n = \mathcal{Q}(y_n + d_1 - d_2) \quad D_n = \mathcal{Q}(y_n + d_1 + d_2) \quad (3.23)$$

La Fig. 3.7(b) muestra el resultado de la conversión del DFE a la arquitectura mirar adelante. Por ejemplo, para las decisiones  $\hat{a}_{n-2} = +1$  y  $\hat{a}_{n-1} = -1$ , la decisión de salida es  $\hat{a}_n = B_n$ .

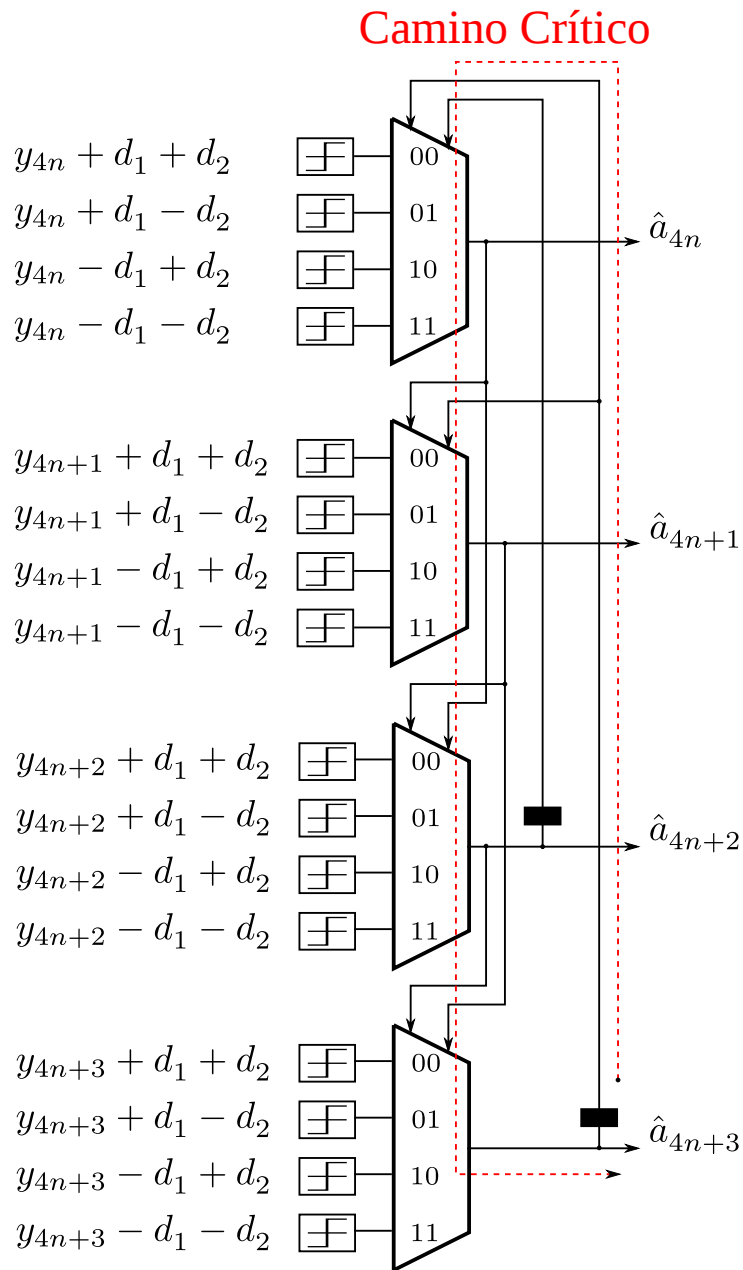


Figura 3.8: Arquitectura paralela del DFE implementando la técnica mirar adelante (look-ahead) para un paralelismo  $P = 4$  y  $N_b = 2$ . La línea de trazo de color rojo marca el camino crítico de la arquitectura.

Como resultado de este planteamiento, esta técnica deja solamente multiplexores en el lazo realimentado. Por lo tanto, el límite de iteración considera un solo multiplexor [22], permitiendo el incremento de frecuencia del reloj del sistema. Por ejemplo, para un DFE serial, para una modulación 2-PAM, aplicando la técnica mirar adelante, el límite de iteración es

Tabla 3.1: Complejidad del DFE paralelo implementando la técnica mirar adelante [22].

Componentes	DFE
Sumadores	$2^{N_b} P$
Registros	$2^{N_b} P$
Multiplexores 2-a-1	$(2^{N_b} - 1)P$

$$IB = \text{Log}_2(M)T_{mux} = \text{Log}_2(2)0,05ns = 0,05ns \quad (3.24)$$

Bajo estas condiciones, y sin considerar los posibles caminos críticos en la implementación de las operaciones de suma para las decisiones futuras, el DFE serial ( $P = 1$ ) con estas características puede operar a una tasa de datos máxima de  $20GS/s$ .

Para alcanzar velocidades mayores, es necesario aplicar paralelización. Esta práctica trae ciertas consecuencias al diseño, donde el principal inconveniente es el incremento exponencial de la complejidad con relación a la memoria del canal. En la Tabla 3.1 se detallan las expresiones de complejidad, donde se observa esta dependencia.

Otro factor clave en el diseño es el camino crítico, el cual tiene una dependencia directa con el número de coeficiente y el paralelismo. Un posible peor caso de camino crítico se resalta en la Fig. 3.8 en línea de trazos de color rojo. La arquitectura tiene un paralelismo  $P = 4$  y considera  $N_b = 2$  coeficientes del filtro realimentado del DFE. Como se aprecia en la Fig. 3.8, el camino crítico involucra todo los multiplexores a lo largo del paralelismo. A medida que se incremente el número de coeficientes, mas grande serán los multiplexores y peor el tiempo de propagación. Para este ejemplo, el tiempo estimado del camino crítico es

$$CP = (\text{Log}_2(2^{N_b})P)T_{mux} = (\text{Log}_2(2^2)4)0,05ns = 0,40ns \quad (3.25)$$

El resultado expresado en la Ec. (3.25) indica que para una arquitectura paralela del DFE utilizando la técnica mirar adelante para  $P = 4$  y  $N_b = 2$ , la frecuencia máxima del reloj tiene que ser  $2,5GHz$ . Con esta referencia de reloj es posible alcanzar una tasa de datos de  $10GS/s$ . Una forma de mejorar este camino crítico es aplicando “retiming”. La arquitectura Concurrente CSA [28], aplica este concepto y considera la reducción del

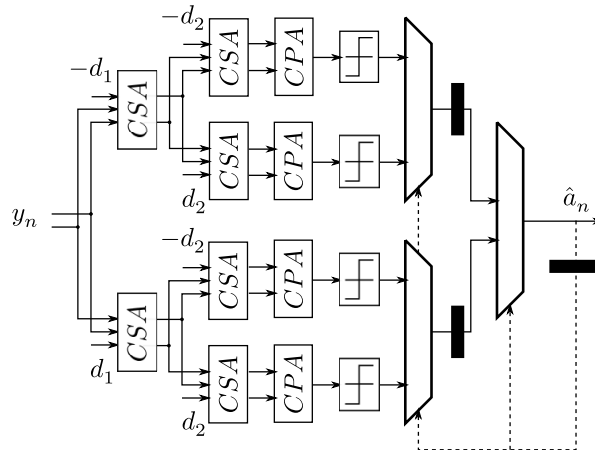


Figura 3.9: Esquema de implementación del DFE serial proponiendo la arquitectura CSA [28]

tiempo de propagación en las operaciones de suma, como se explica a continuación.

### 3.4.2. Arquitectura Concurrente CSA

La arquitectura CSA [28] es propuesta para reducir el camino crítico asociado a la implementación mostrada en la Fig. 3.8. Aplica la técnica “retiming” para cortar el camino crítico en los multiplexores y emplea un tipo de sumador llamado sumador acarrear-guardar (*Carry Save Adder - CSA*) distribuido en árbol. Un esquema serial del DFE utilizando esta técnica es mostrado en la Fig. 3.9, donde el límite de iteración sigue siendo igual al anterior. La ventaja que presenta es que, para el esquema paralelo del DFE mostrado en la Fig. 3.8, la frecuencia máxima del reloj puede llegar a  $4GHz$  y una tasa de datos de aproximadamente  $16GS/s$ .

Por otro lado, el incremento en la complejidad de los componentes sigue siendo exponencial. Un detalle de las expresiones de complejidad de cada uno de los componentes de esta arquitectura, se muestra en la Tabla 3.2.

### 3.4.3. Segmentación de Lazo de Multiplexores Anidados

La segmentación de lazo de multiplexores anidados (*pipelining nested multiplexer loops*) [29, 30, 31] parte del principio de mirar adelante y plantea la estimación de las decisiones futuras en varios niveles. Esto se logra conectando en cascada multiplexores, los cuales son controlados utilizando decisiones futuras y pasadas. Iterando sobre la Ec.

Tabla 3.2: Complejidad del DFE paralelo implementando la arquitectura CSA [28].

Componentes	DFE
Sumadores	$(2^{N_b+1} - 2)P$
Registros	$((2^{N_b+1} - 2) + N_b)P$
Multiplexores 2-a-1	$(2^{N_b} - 1)P$

(3.21), se incrementa el número de niveles (*Stages* -  $M$ ). Para esto, se reagrupan los términos en la Ec. (3.21)

$$\hat{a}_n = (A_n \hat{a}_{n-2} + B_n \bar{\hat{a}}_{n-2}) \hat{a}_{n-1} + (C_n \hat{a}_{n-2} + D_n \bar{\hat{a}}_{n-2}) \bar{\hat{a}}_{n-1} \quad (3.26)$$

y se expande para  $\hat{a}_{n-1}$

$$\hat{a}_{n-1} = (A_{n-1} \hat{a}_{n-3} + B_{n-1} \bar{\hat{a}}_{n-3}) \hat{a}_{n-2} + (C_{n-1} \hat{a}_{n-3} + D_{n-1} \bar{\hat{a}}_{n-3}) \bar{\hat{a}}_{n-2} \quad (3.27)$$

Reemplazando la Ec. (3.27) en la Ec. (3.26), obtenemos la expresión para dos niveles de multiplexores

$$\begin{aligned} \hat{a}_n = & A_n \hat{a}_{n-2} (A_{n-1} \hat{a}_{n-3} + B_{n-1} \bar{\hat{a}}_{n-3}) + B_n \bar{\hat{a}}_{n-2} (C_{n-1} \hat{a}_{n-3} + D_{n-1} \bar{\hat{a}}_{n-3}) + \\ & C_n \hat{a}_{n-2} (\bar{A}_{n-1} \hat{a}_{n-3} + \bar{B}_{n-1} \bar{\hat{a}}_{n-3}) + D_n \bar{\hat{a}}_{n-2} (\bar{C}_{n-1} \hat{a}_{n-3} + \bar{D}_{n-1} \bar{\hat{a}}_{n-3}) \end{aligned} \quad (3.28)$$

Observamos que el control de los multiplexores se realiza con decisiones pre-calculadas ( $A_{n-1}$ ,  $B_{n-1}$ ,  $C_{n-1}$  y  $D_{n-1}$ ). Esto hace que solamente un nivel de multiplexores sean controlados por las decisiones pasadas, reduciendo el límite de iteración. El esquema de la Fig. 3.10, ejemplifica lo mostrado en la Ec. (3.28). Consiste en dos niveles de multiplexores (2-stages), donde el primer nivel es controlado por decisiones pre-calculadas y el segundo nivel por decisiones pasadas. El límite de iteración para el ejemplo, se calcula considerando el número de niveles ( $M = 2$ ) y el número de pre-cálculos ( $L = 2^{N_b}$ ) [31]

$$IB = \frac{\log_2(L) + 1}{M + \log_2(L) - 1} T_{mux} = \frac{\log_2(2^2) + 1}{2 + \log_2(2^2) - 1} 0,05ns = 0,05ns \quad (3.29)$$

donde  $T_{mux}$  es el tiempo de propagación del multiplexor. Como se aprecia en la Ec. (3.29), el incremento en el número de niveles disminuye el tiempo de propagación con la



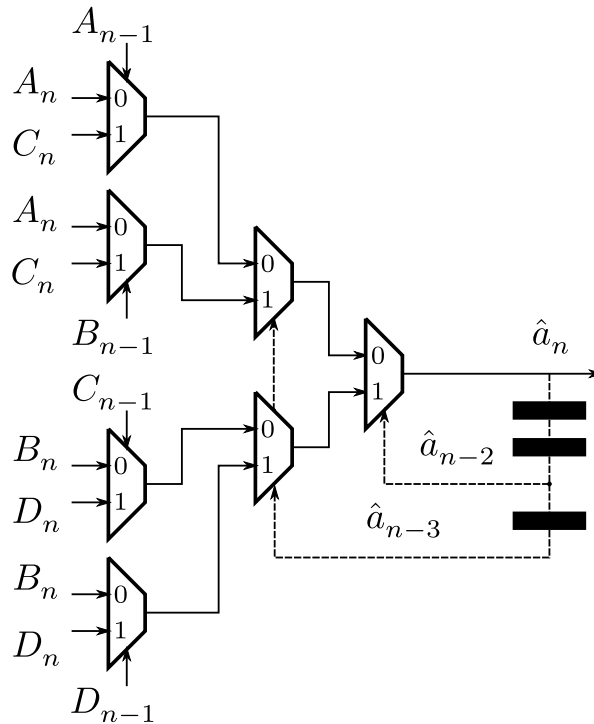


Figura 3.10: Esquema de implementación del DFE utilizando el criterio de segmentación de lazo de multiplexores anidados. Considera dos coeficientes del filtro realimentado ( $N_b = 2$ ) y dos niveles de multiplexores.

desventaja de incrementar la complejidad del hardware, principalmente en la cantidad de multiplexores. Este incremento ocurre con un factor  $2^{N_b}$  (considerando una modulación PAM).

El incremento excesivo de la complejidad de los multiplexores se puede reducir empleando el procesamiento de bloque incremental (*incremental block processing*) [24, 31]. El concepto se basa en la paralelización de un filtro IIR de  $N^{th}$  orden. Las primeras  $N$  muestras de salida se calculan independientes y las restantes ( $P - N$ ), se calculan usando las salidas previas. En la Fig. 3.11, se representa la arquitectura del DFE paralelo utilizando este criterio de optimización para un paralelismo de  $P = 4$ , dos coeficientes ( $N_b = 2$ ) del filtro realimentado y cuatro niveles ( $M = 4$ ). En línea de trazo de color rojo se resalta uno de los posibles caminos críticos, sin considerar las operaciones de suma de las decisiones pre-calculadas. La estimación del camino crítico resaltado en la Fig. 3.11 es

$$CP = (7)T_{mux} = (7)0,05ns = 0,35ns \quad (3.30)$$

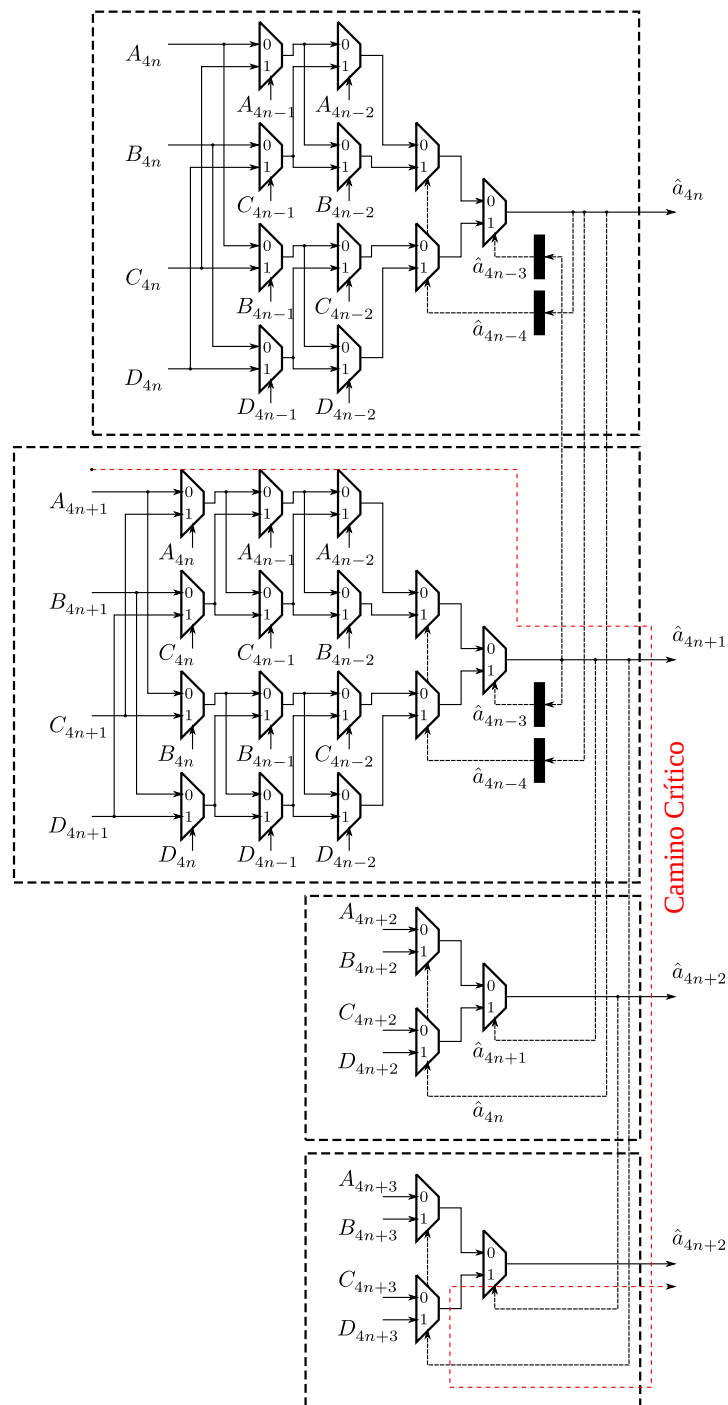


Figura 3.11: Arquitectura paralela del DFE aplicando la técnica de segmentación de lazo de multiplexores anidados [31] para un paralelismo de  $P = 4$ , dos coeficientes ( $N_b = 2$ ) y cuatro niveles ( $M = 4$ ). En línea de trazo de color rojo se resalta uno de los caminos críticos de la arquitectura.

Como consecuencia, la frecuencia máxima del reloj para este ejemplo es aproximadamente  $2,8GHz$  y una tasa de datos de  $\approx 12GS/s$ .

Tabla 3.3: *Complejidad del DFE paralelo implementando la técnica de segmentación de lazo de multiplexores anidados [31].*

Componentes	DFE
Sumadores	$2^{N_b} P$
Registros	$N_b^2 + LP$
Multiplexores 2-a-1	$[(ML - 1) - ((N_b - 1)L)/2]N_b + (L - 1)(P - N_b)$

Por último, la complejidad de la arquitectura segmentación de lazo de multiplexores anidados se detalla en la Tabla 3.3, donde el número de sumadores y multiplexores para el DFE paralelo se extrae de [31]. Los registros son estimados considerando la distribución de la arquitectura.

#### 3.4.4. Esquema de Pre-Cálculo Parcial

El esquema de pre-cálculo parcial (*partial pre-computation scheme*) [26, 27], combina los conceptos de las técnicas mirar adelante (*look-ahead*) y segmentación de lazo de multiplexores anidados. Divide el cálculo de todas las posibles decisiones futuras en dos redes de multiplexores, los cuales son controlados por decisiones anteriores. Esta división de los cálculos reduce el número de operaciones aritméticas y el número de multiplexores.

Los coeficientes se dividen en dos grupos. Si  $N_b$  es par un grupo tendrá  $L = N_b/2$  coeficiente, en caso contrario  $L = (N_b - 1)/2$ . De la combinación matemática de la muestra recibida  $y_k$  y el grupo de coeficientes  $d_1, d_2, \dots, d_L$ , se obtienen las entradas a la primera red de multiplexores. Las entradas a la segunda red de multiplexores resulta de la combinación de los coeficientes  $d_{L+1}, d_{L+2}, \dots, d_{N_b}$ . La salida de esta última suma parcial, se suma con el pre-cálculo de la red anterior. Por ejemplo, considerando un DFE serial con cuatro (4) coeficientes, los cuales se reagrupan en  $[d_1, d_2]$  y  $[d_3, d_4]$ . El esquema equivalente aplicando esta técnica, se observa en la Fig. 3.12. En este tipo de arquitectura, el límite de iteración queda determinado tanto por los multiplexores como por los sumadores. En el ejemplo de la Fig. 3.12, el límite de iteración es

$$IB = \frac{1}{N_b/2 + 1} T_{add} + (\log_2 M) T_{mux} = \frac{1}{3} 0,10ns + Loh_2(2) 0,05ns = 0,083ns \quad (3.31)$$

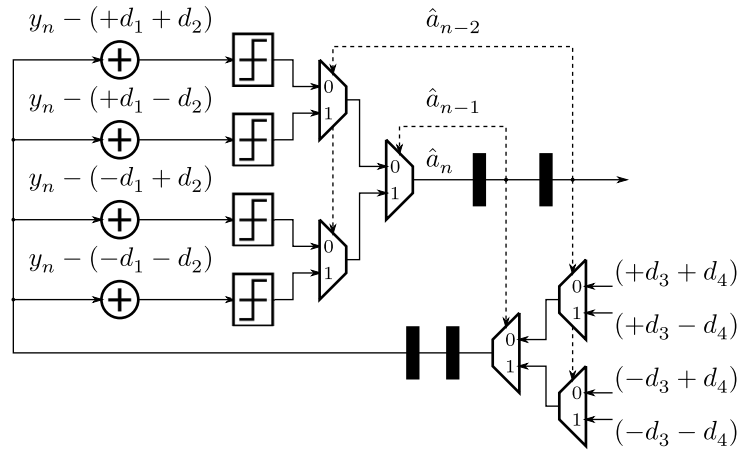


Figura 3.12: Esquema de implementación del DFE serial utilizando la técnica de pre-cálculo parcial [27].

donde  $T_{add}$  y  $T_{mux}$  representan el tiempo de propagación del sumador y del multiplexor, respectivamente. Considerando, que en los sistemas realimentados, el límite de iteración suele ser el camino crítico del diseño (Sección 3.2), la tasa de dato máxima para el ejemplo de la Fig. 3.12 es  $12GS/s$ .

A diferencia de la arquitectura de segmentación de lazo de multiplexores anidados, el esquema de pre-cálculo parcial presenta una distribución modular. Esto facilita al diseñador la implementación en paralelo. En la Fig. 3.13 se muestra el diseño de esta arquitectura para un paralelismo de  $P = 4$  y cuatro (4) coeficientes ( $N_b = 4$ ). La línea de trazo en color rojo remarca el camino crítico [27], el cual es

$$CP = 3 \frac{N_b}{2} T_{mux} + T_{add} = (6)0,05ns + 0,10ns = 0,40ns \quad (3.32)$$

Por lo tanto, el ejemplo planteado puede operar con una frecuencia de reloj máxima de  $2,5GHz$  y una tasa de dato de  $10GS/s$ . El esquema de pre-cálculo tiene la ventaja de poder aplicar la estrategia de “retiming” para poder mejorar el desempeño, a costa de incrementar la complejidad.

La complejidad de la arquitectura se la detalla en la Tabla 3.4, donde el número de sumadores y multiplexores para el DFE paralelo se extrae de [27]. Los registros son estimados considerando la distribución de la arquitectura. A diferencia de las arquitecturas anteriores, el incremento de la complejidad continúa creciendo exponencialmente con la memoria del canal, pero con un factor  $N_B/2$ . Esto sigue siendo crítico en los casos que

### 3.4. Técnicas para Implementación en Alta Velocidad de Cancelación No Lineal

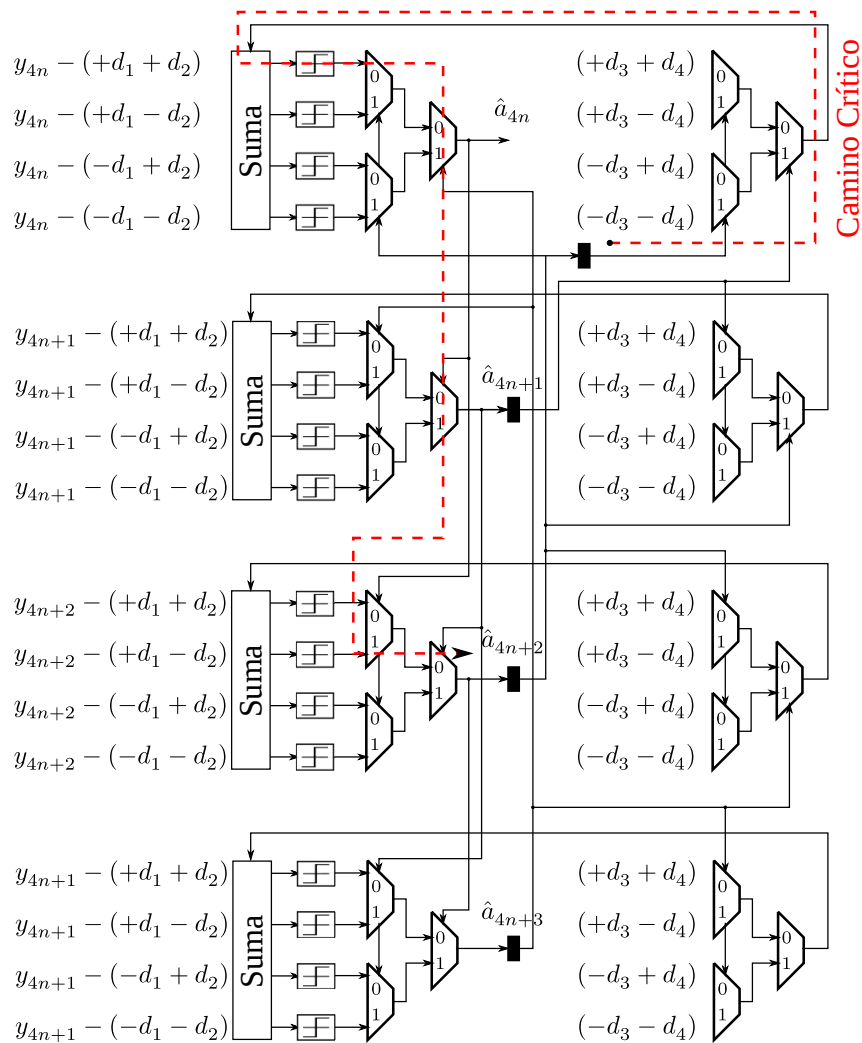


Figura 3.13: Arquitectura paralela del DFE implementando la técnica de pre-cálculo parcial para un paralelismo de  $P = 4$  y  $N_b = 4$ . En línea de trazo de color rojo se resalta un posible camino crítico del diseño

Tabla 3.4: Complejidad del DFE paralelo implementando la técnica de pre-cálculo parcial [27].

Componentes	DFE
Sumadores	$2^{N_b/2}2P$
Registros	$2^{N_b/2}(P + 1)$
Multiplexores 2-a-1	$(2^{N_b/2} - 1)2P$

$N_b \gg 1$ .

Tabla 3.5: *Comparación de complejidad entre diferentes arquitecturas paralelas del DFE para una modulación 2-PAM y un canal con gran memoria ( $L \gg 1$ ).*

Componentes	DFE [22]	DFE [27]	DFE [31]	DFE [28]
Sumadores	$2^L P$	$2^{L/2} 2P$	$2^L P$	$2^{L+1} P$
Registros	$\sim 2^L P$	$\sim 2^{L/2} (P + 1)$	$L^2 + 2^L P$	$(2^L + L) P$
Multiplexores 2-a-1	$(2^L - 1) P$	$(2^{L/2} - 1) 2P$	$2^L L (P - L/2 + P/L - 1)$	$2^L P$

### 3.5. Comparación de Complejidad

Las cuatro arquitecturas del DFE en paralelo presentadas en la *Sección 3.4* tienen como objetivo alcanzar altas velocidades de procesamiento. El concepto básico de los diseños es implementar la técnica mirar adelante, la cual estima todas las decisiones futuras en base a la muestra de entrada actual. Además, elige la decisión de salida utilizando decisiones anteriores para controlar multiplexores anidados. Este criterio, tiene la ventaja de eliminar del lazo realimentado las operaciones de suma, las cuales son muy costosas en términos de tiempo. La principal desventaja que presenta este tipo de diseño es el incremento exponencial de la complejidad en función de la memoria del canal. El procesamiento de bloque incremental y el pre-cálculo parcial de las decisiones futuras, son alternativas que buscan resolver esta problemática.

La Tabla 3.5 muestra una comparación de complejidad entre las arquitecturas del DFE paralelo propuestas en [22, 27, 31, 28]. La comparación se realiza considerando el número de sumadores de dos entradas y una salida, registros y multiplexores 2-a-1. Además, se contempla una modulación 2-PAM y un canal con gran memoria ( $L \gg 1$ ). En la Fig. 3.14 se grafica la comparación del número de componentes en función del número de coeficientes del filtro realimentado. Estos cálculos se realizan considerando un factor de paralelismo  $P = 16$ . La gráfica deja en evidencia el crecimiento exponencial de la complejidad de los componentes en relación con la longitud del canal.

Desde la perspectiva del diseñador, las arquitecturas paralelas del DFE expuestas en esta sección, son complejas de implementar. Esto se debe a la dependencia temporal de las muestras entre cada salida del esquema paralelo y la existencia del lazo realimentado. Lo cual viene acompañado con problemas de camino crítico, que llevan al diseñador a

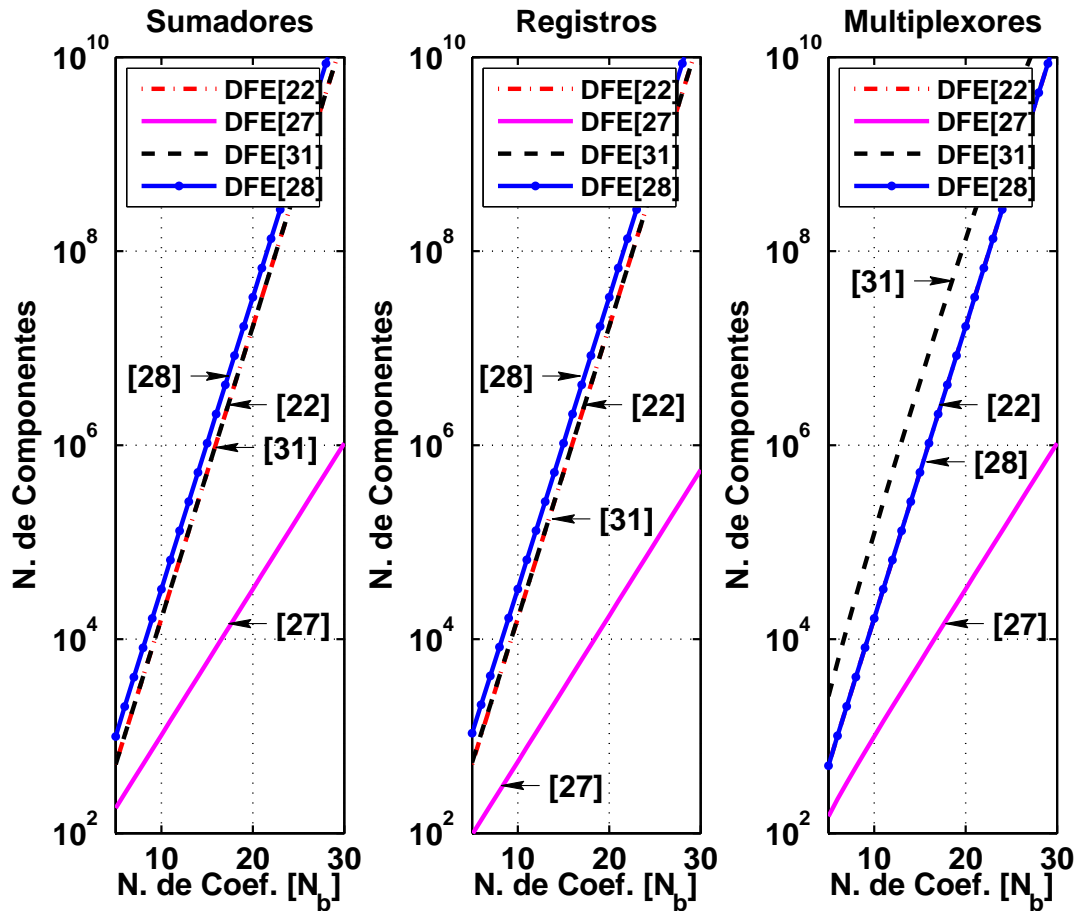


Figura 3.14: Número de sumadores, registros y multiplexores 2-a-1 en función del número de coeficientes del filtro realimentado ( $L$ ) para diferentes arquitecturas del DFE paralelo propuestos en [22, 27, 31, 28]. Factor de paralelismo:  $P = 16$ . Formato de modulación: 2-PAM.

aplicar estrategias de implementación que incrementan la complejidad.

Estos resultados nos muestran la necesidad de buscar nuevas alternativas de ecualización que reduzcan la complejidad de implementación para canales con muy larga memoria. Así como también, sean simples de desarrollar y eliminen por completo el lazo realimentado. La presente Tesis propone una nueva técnica de ecualización que resuelve todas estas dificultades, denominado ecualizador de alimentación directa de decisiones (*Decision FeedForward Equalizer - DFFE*).

### 3.6. Conclusión Parcial

En este capítulo se expusieron las limitaciones fundamentales de implementación del DFE en paralelo. Se explicaron las técnicas tradicionales que permiten incrementar la velocidad de los receptores de comunicación, entre las que se destacan segmentación, “re-timing” y procesamiento en paralelo. La implementación de estas técnicas no elimina las limitaciones intrínsecas en el lazo realimentado del DFE, haciendo evidente la necesidad de evaluar las propuestas actuales que resuelven este inconveniente. Entre las técnicas propuestas por diferentes autores, se destaca el mirar adelante (*look-ahead*) que pre-calcula todas las posibles decisiones futuras en base a decisiones anteriores. Bajo el mismo concepto, se muestran diferentes alternativas que buscan disminuir la complejidad de implementación de estos sistemas. En todos los casos, la complejidad de estas técnicas tienen un crecimiento exponencial dependiente de la memoria del canal. Estos aspectos de las arquitecturas del DFE paralelo hacen ver la necesidad de buscar nuevas alternativas de ecualización que operen con un desempeño similar al DFE, pero que la complejidad no crezca exponencialmente.

La técnica de ecualización orientada a resolver todas estas dificultades, se denomina ecualizador directo asistido por decisiones (*Decision FeedForward Equalizer - DFFE*). El siguiente capítulo desarrolla los conceptos teóricos y se analiza la complejidad del DFFE.



# CAPÍTULO

## 4

### NUEVA ARQUITECTURA ECUALIZADORA ITERATIVA DE BAJA COMPLEJIDAD

**Resumen:** *En este capítulo se presenta una nueva arquitectura ecualizadora iterativa de baja complejidad denominada ecualizador directo asistido por decisiones (Decision FeedForward Equalizer - DFFE). El DFFE utiliza decisiones tentativas para mejorar la estimación de la interferencia inter-símbolo del canal. Se realiza un análisis teórico del desempeño, mostrando que el comportamiento es similar al DFE. Las métricas utilizadas para validar el desempeño son la probabilidad de error de bit y la información mutua. Todos los resultados numéricos se corroboran empleando un simulador. Por último, se propone otra alternativa de estructura ecualizadora que interactúa con un detector de secuencia de máxima verosimilitud implementado con el algoritmo de Viterbi con el objetivo de mejorar la estimación de las primeras decisiones tentativas del DFFE. Esto trae como consecuencia una reducción en la complejidad final del receptor.*

### 4.1. Introducción

En el *Capítulo 3*, se analizó la necesidad de utilizar diferentes estrategias para implementar el DFE en sistemas de alta velocidad. En sistemas recursivos, tal como lo es el DFE, aplicar técnicas como segmentación, retiming y paralelismo para aumentar la velocidad en el receptor, suele ser muy difícil por el cuello de botella generado por el lazo realimentado. Existen técnicas que buscan solucionar esta limitación, por ejemplo, la técnica Mirar Adelante, a costa de un incremento exponencial en la complejidad dependiente de la longitud del canal. Esto implica que para canales muy dispersivos, la utilización de arquitecturas que emplean esta técnica no sea posible de implementar. Por lo tanto, la aplicación del DFE está limitada a canales con ISI moderada. Como consecuencia, hay una necesidad por receptores de complejidad reducida que puedan operar de manera eficiente en los canales con gran ISI.

La presente Tesis propone una nueva arquitectura ecualizadora iterativa de baja complejidad para receptores de alta velocidad que resuelve estas necesidades. El ecualizador directo asistido por decisiones (*Decision FeedForward Equalizer - DFFE*) permite obtener un rendimiento similar al DFE con una arquitectura paralelizable, cuya complejidad aumenta cuadráticamente con la memoria del canal. Para los canales con gran ISI, esto resulta en una dramática reducción de la complejidad si se compara con el DFE. La idea central detrás del DFFE, es la iteración de decisiones tentativas para mejorar la precisión de la estimación de la ISI.

En este capítulo se realiza un análisis teórico del desempeño del DFFE, con el objetivo de mostrar que el comportamiento es similar al DFE. Para tal fin, se utilizan los conceptos de propagación de error del DFE y teoría de la información [58]. Se propone una expresión de probabilidad de error, con el objetivo de facilitar al diseñador la estimación del número de iteraciones en el DFFE. Estas estimaciones se corroboran por medio de resultados numéricos empleando un simulador, donde se compara el desempeño del DFFE y DFE. Con los mismos objetivos y utilizando similares herramientas, es decir empleando análisis teórico y resultados de simulaciones, se propone una segunda alternativa de arquitectura del DFFE que integra el algoritmo de Viterbi con el objetivo de reducir la complejidad final del DFFE. Para ambos casos, se mostrará en el *Capítulo 5* que la complejidad de implementación es menor que otras propuestas de arquitecturas paralelas del DFE.

## 4.2. Ecuador Directo Asistido por Decisiones

El DFFE [45, 46] es una estructura ecualizadora que elimina el lazo realimentado de las decisiones, logrando alcanzar un desempeño similar al tradicional DFE. Para simplificar el análisis, consideraremos la dispersión del canal solamente con la ISI generada por los post-cursos. Posteriormente, los resultados de simulación se realizarán utilizando un canal con ISI con pre y post-cursos combinando el DFFE con un ecualizador directo. Tenemos  $y_n$ ,  $\hat{a}_n^{(i)}$  y  $L$  que son las muestras de entrada al DFFE, las decisiones tentativas en la  $i^{th}$  iteración y la memoria del canal, respectivamente. En la primera iteración,  $i = 0$ , obtenemos la primera decisión tentativa sin ninguna cancelación de la interferencia

$$\hat{a}_n^{(0)} = \mathcal{Q}(y_n), \quad (4.1)$$

donde  $\mathcal{Q}(\cdot)$  es la función detector de umbral. Esta decisión tentativa es usada luego para cancelar la ISI de los post-cursos introducido por el primer símbolo pasado y por lo tanto, mejorando la precisión de la detección. Utilizando los retardos correspondientes de tiempo, podemos obtener la decisión tentativa en la segunda iteración como sigue

$$\hat{a}_n^{(1)} = \mathcal{Q}\left(y_n - f_1(\hat{a}_{n-1}^{(0)})\right), \quad (4.2)$$

donde  $f_k(\cdot)$  con  $0 < k < L$  representa la ISI parcial de los post-cursos causada por los  $k$  símbolos pasados. Este proceso se repite al menos hasta  $L$  decisiones tentativas consecutivas. En este punto la decisión final puede ser obtenida desde

$$\hat{a}_n = \hat{a}_n^{(L)} = \mathcal{Q}\left(y_n - f_L(\hat{a}_{n-1}^{(L-1)}, \dots, \hat{a}_{n-L}^{(0)})\right), \quad (4.3)$$

donde  $f_L(\cdot)$  es la ISI total de los post-cursos del canal. Como se verá mas adelante, basándonos en la teoría de la información [58], mostraremos que la confiabilidad de las decisiones tentativas  $\hat{a}_n^{(i)}$  mejoran con el incremento del número de iteraciones  $i$ . En este sentido, tanto la precisión de la estimación de la interferencia y el desempeño del DFFE mejoran con el número de iteraciones. Resultados numéricos derivados de simulaciones en computadora demuestran que el DFFE puede alcanzar un desempeño similar al DFE en canales altamente dispersivos. Además, ya que las decisiones tentativas son usadas en lugar de las decisiones finales para la estimación de los post-cursos de la ISI, es posible

## Capítulo 4. Nueva Arquitectura Ecualizadora Iterativa de Baja Complejidad

implementar el DFFE de forma directa, lo que nos lleva de una manera muy simple a una implementación paralela. Mostraremos que la complejidad computacional del DFFE crece cuadráticamente con  $L$ . Este resultado es una drástica reducción de la complejidad en comparación con otras arquitecturas paralelas del DFE, donde la complejidad crece exponencialmente con  $L$ . Esta relación entre desempeño y complejidad hace del DFFE una excelente alternativa para implementaciones en receptores de muy alta velocidad en canales altamente dispersivos.

Como fue expresado en el párrafo anterior, el uso iterativo de las decisiones tentativas para estimar la ISI de los post-cursos es la clave del DFFE. En la siguiente sección, analizaremos como es el procesamiento de la señal de entrada en cada iteración y la arquitectura del DFFE.

### 4.2.1. Arquitectura del DFFE

La arquitectura del DFFE contempla los criterios de segmentación y retiming al igual que en el caso de los filtros directos. Esto se debe a que no posee lazos realimentados que dificulten la aplicación de estas técnicas. Como veremos en el *Capítulo 5*, la paralelización consiste en instanciar múltiples módulos DFFEs sin tener una dependencia temporal de las muestras entre cada uno de ellos. Esta característica ayuda al diseñador a desarrollar la arquitectura del DFFE de forma rápida y eficiente.

Para comprender la disposición de los componentes, plantearemos un caso de uso. Las muestras en el receptor se las representa como

$$y_n = a_n + \sum_{k=1}^L a_{n-k}d_k + z_n, \quad (4.4)$$

donde  $d_k$  con  $k = 1, \dots, L$  son los coeficientes de los post-cursos de la ISI,  $a_n$  es el símbolo transmitido (por ejemplo.,  $a_n \in \{\pm 1\}$ ), y  $z_n$  es el ruido Gaussiano con potencia  $\sigma^2$ . Partiendo de (4.3) y asumiendo que el canal es conocido en el receptor (para simplificar, inicialmente, la estimación del canal es perfecta), el símbolo detectado por el DFFE en el instante de tiempo  $n$  puede definirse como

$$\hat{a}_n = \hat{a}_n^{(R-1)} = \mathcal{Q} \left( y_n - \sum_{k=1}^L \hat{a}_{n-k}^{(R-1-k)} d_k \right), \quad (4.5)$$

## 4.2. Ecuador Directo Asistido por Decisiones

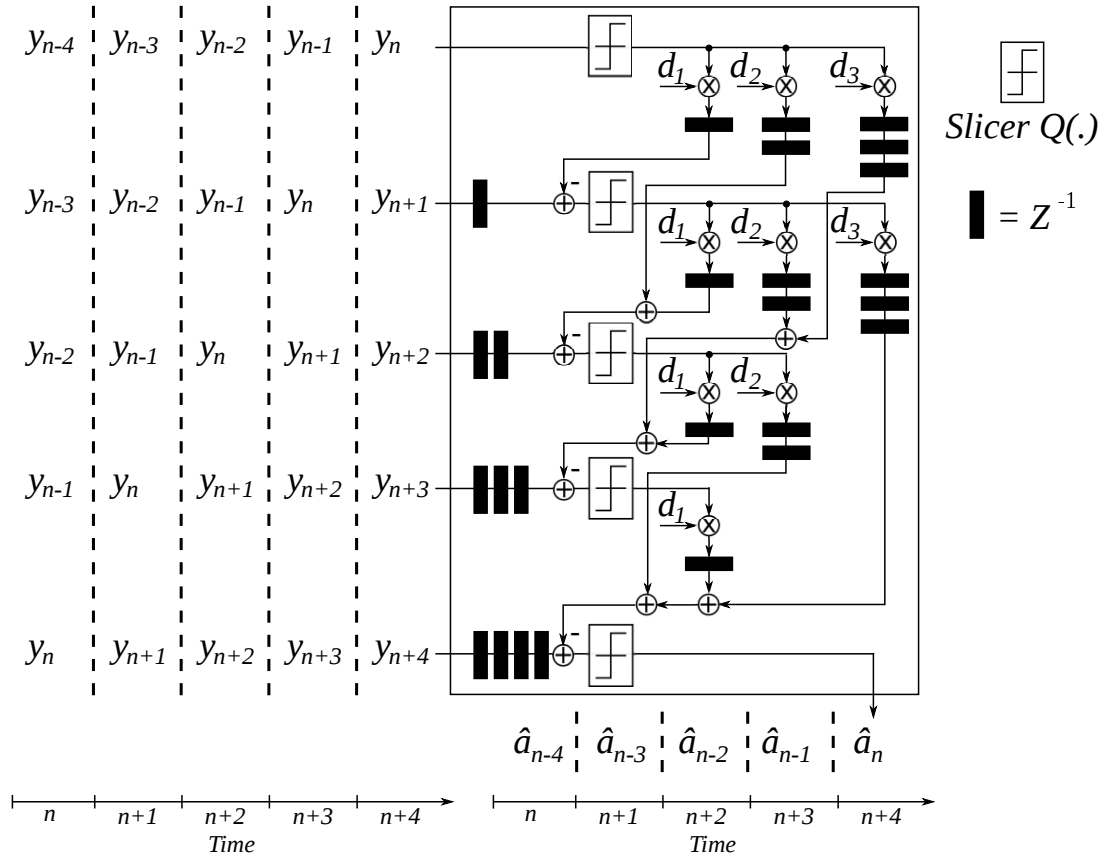


Figura 4.1: Ejemplo de un DFFE con tres (3) coeficientes ( $L = 3$ ) y  $R = 5$  iteraciones. Notar que la latencia entre la señal de entrada y la decisión es  $R - 1$ .

donde  $R$  es el número total de iteraciones, y  $R > L$ . Las primeras  $L$  decisiones tentativas pueden ser calculadas iterativamente como sigue

$$\hat{a}_n^{(i)} = \mathcal{Q} \left( y_n - \sum_{k=1}^i \hat{a}_{n-k}^{(i-k)} d_k \right), \quad 1 \leq i < L, \quad (4.6)$$

con  $\hat{a}_n^{(0)} = \mathcal{Q}(y_n)$  para  $i = 0$ .

La Fig. 4.1 muestra la arquitectura del DFFE para un canal con memoria  $L = 3$  y  $R = 5$ . Notar que la decisión final  $\hat{a}_n = \hat{a}_n^{(4)}$  usa decisiones tentativas pasadas para estimar la interferencia de los post-cursos, y no decisiones finales previas como en el DFE. Como se mostrará mas adelante, esto permite una implementación paralela directa del DFFE.

### 4.3. Confiabilidad de las Decisiones Tentativas

El concepto básico del DFFE se centra en utilizar decisiones tentativas para mejorar la estimación del símbolo recibido. El incremento en el número de iteraciones mejora el desempeño del DFFE. Para justificar este comportamiento, utilizaremos como métrica la información mutua (*mutual information*) [58, 59]. A continuación, se exponen los conceptos de este criterio que ayudarán a entender los posteriores análisis.

#### 4.3.1. Información Mutua

En general, una fuente de información genera cualquier conjunto de  $M$  símbolos diferentes, que son considerados como representaciones de una variable discreta aleatoria  $X$  que adopta cualquier valor en el rango  $\Omega_x = (x_1, x_2, \dots, x_M)$ . Cada símbolo  $x_i$  tiene probabilidad  $P(x_i)$  de ser transmitido y contiene información  $I_i$ . Las probabilidades de símbolos deben estar de acuerdo con el hecho de que al menos uno de ellos se transmitirá, por lo que

$$\sum_{i \in \Omega_x} P(x_i) = 1 \quad (4.7)$$

La distribución de probabilidad de símbolo de la fuente es estacionaria, y los símbolos son independientes y se transmiten a una velocidad de  $r$  símbolos por segundo. Esta descripción corresponde a una fuente discreta sin memoria. Cada símbolo contiene la información  $I_i$  para que el conjunto  $(I_1, I_2, \dots, I_M)$  pueda ser visto como una variable aleatoria discreta con información promedio

$$H_b(X) = \sum_{i \in \Omega_x} P(x_i) I_i = \sum_{i \in \Omega_x} P(x_i) \log_2 \frac{1}{P(x_i)} \quad (4.8)$$

La función así definida se llama entropía (*entropy*) de la fuente. En otras palabras, la entropía es la información promedio asociada a la fuente discreta sin memoria. Cuando se utiliza base 2, la entropía se mide en bits por símbolos

$$H(X) = \sum_{i \in \Omega_x} P(x_i) I_i = \sum_{i \in \Omega_x} P(x_i) \log_2 \frac{1}{P(x_i)} \quad (4.9)$$

La probabilidad de ocurrencia de un símbolo de salida dado  $y_j$  es  $P(y_j)$ . Sin embargo, si se conoce el símbolo transmitido  $x_i$ , entonces la probabilidad condicional relacionada

---

### 4.3. Confiabilidad de las Decisiones Tentativas

---

con el símbolo de salida, se convierte en  $P(y_j/x_i)$ . De la misma manera, la probabilidad de un símbolo de entrada dado, inicialmente por  $P(x_i)$ , puede también ser definido si se conoce la salida actual. Por lo tanto si el símbolo  $y_j$  recibido, aparece en la salida del canal, entonces la probabilidad condicional del símbolo de entrada se convierte en  $P(x_i/y_j)$ .

La probabilidad  $P(x_i)$  se conoce como la probabilidad a priori; es decir, que es la probabilidad que caracteriza al símbolo de entrada antes de que se conozca la presencia de cualquier símbolo de salida. Normalmente, esta probabilidad es igual a la probabilidad de que el símbolo de entrada haya sido transmitido por la fuente. La probabilidad  $P(x_i/y_j)$  es una estimación del símbolo  $x_i$  después de saber que un símbolo dado  $y_j$  apareció en la salida del canal, y se llama la probabilidad a posteriori. La entropía definida en la Ec. (4.9) es llamada entropía a priori. La entropía a posteriori (o entropía condicional) se la expresa como

$$H(X/y_j) = \sum_{i \in \Omega_x} P(x_i/y_j) \log_2 \frac{1}{P(x_i/y_j)} \quad (4.10)$$

donde la entropía condicional es definida por

$$H(X/Y) = \sum_{i \in \Omega_x} \sum_{j \in \Omega_y} P(x_i, y_j) \log_2 \frac{1}{P(x_i/y_j)} \quad (4.11)$$

donde  $\Omega_y$  representa el alfabeto de  $y_j$ .

La información mutua [59] para un par de entrada-salida  $(x_i, y_j)$  de un canal dado, mide la información transferida cuando  $x_i$  se envía y se recibe  $y_j$ , y se define como

$$I(x_i, y_j) = \log_2 \frac{P(x_i/y_j)}{P(x_i)} \text{bits} \quad (4.12)$$

En un canal sin ruido,  $y_j$  esta relacionado únicamente con el correspondiente  $x_i$ , por lo que constituyen un par de entrada-salida  $(x_i, y_j)$ , donde  $P(x_i/y_j) = 1$  y  $I(x_i, y_j) = \log_2 \frac{1}{P(x_i)}$  bits. Es decir, la información transferida es igual a la auto-información que corresponde a la entrada  $x_i$ . En un canal muy ruidoso, la salida  $y_j$  y la entrada  $x_i$  serían completamente no correlacionadas, y así  $P(x_i/y_j) = P(x_i)$  y también  $I(x_i, y_j) = 0$ , es decir, no hay transferencia de la información. En general, un canal dado operará entre estos dos extremos.

Generalizando, la información mutua [59] definida para todos los pares de entrada-salida de un canal dado es

$$I(X, Y) = \sum_{i \in \Omega_x} \sum_{j \in \Omega_y} P(x_i, y_j) I(x_i, y_j) = \sum_{i \in \Omega_x} \sum_{j \in \Omega_y} P(x_i, y_j) \log_2 \frac{P(x_i/y_j)}{P(x_i)} \text{bps} \quad (4.13)$$

Este cálculo se realiza sobre los alfabetos de entrada ( $\Omega_x$ ) y salida ( $\Omega_y$ ). La información mutua mide la cantidad media de información de la fuente obtenida de cada símbolo de salida.

La información mutua, puede ser expresada por medio de la entropía como [59]

$$I(X, Y) = H(X) - H(X/Y) \quad (4.14)$$

donde  $H(X/Y)$  (ver (4.11)) es usualmente llamada equivocación. En un sentido, la equivocación puede ser vista como la pérdida de información en el canal ruidoso, y es una función de la probabilidad condicional hacia atrás<sup>1</sup>  $P(x_i/y_j)$  (es decir, se refiere a la matriz de probabilidad condicional que define el canal). La observación de un símbolo  $y_j$  de salida ofrece  $H(X) - H(X/Y)$  bits de información. Esta diferencia es la información mutua del canal.

Por último, y como mencionamos en párrafos anteriores,  $I(X, Y)$  es la información de  $X$  contenida en  $Y$ . Por propiedad de simetría, la información mutua se puede expresar considerando la información de  $Y$  contenida en  $X$ . Ya que,  $P(x_i/y_j)P(y_j) = P(y_j/x_i)P(x_i)$ , la información mutua se ajusta a la condición  $I(X, Y) = I(Y, X)$  e intercambiando la entrada y salida también es cierto que

$$I(X, Y) = H(Y) - H(Y/X) \quad (4.15)$$

Tomando de referencia estos conceptos, la siguiente sección analiza la información mutua en cada iteración del DFFE.

### 4.3.2. Estimación de Confiabilidad

Partiendo de los conceptos expuestos en la *Sección 4.3.1*, analizaremos la información mutua entre el símbolo transmitido  $a_n$  y la decisión tentativa en la  $i^{\text{th}}$  iteración,  $\hat{a}_n^{(i)}$ . De la Ec. (4.14), la información mutua es

---

<sup>1</sup>Usualmente,  $P(y_j/x_i)$  son llamadas probabilidades condicionales hacia adelante (*forward conditional probabilities*) y  $P(x_i/y_j)$  son conocidas como probabilidades condicionales hacia atrás (*backward conditional probabilities*).



$$I(a_n, \hat{a}_n^{(i)}) = H(a_n) - H(a_n/\hat{a}_n^{(i)}) \quad (4.16)$$

donde  $H(\cdot)$  y  $H(\cdot/\cdot)$  es la entropía y la entropía condicional[58], respectivamente. Para facilitar el análisis y escribir la información mutua en términos de la probabilidad de transmisión (canal) y la distribución de entrada, aplicamos la propiedad de simetría definiendo a  $I(a_n, \hat{a}_n^{(i)})$  como

$$I(a_n, \hat{a}_n^{(i)}) = H(\hat{a}_n^{(i)}) - H(\hat{a}_n^{(i)}/a_n) \quad (4.17)$$

donde la información mutua pasa a ser la información de  $\hat{a}_n^{(i)}$  contenida en  $a_n$ .

De acuerdo a lo expresado en la *Sección 4.3.1*,  $Pr\{a_n = \Omega_j\}$  es la probabilidad de transmitir un símbolo  $a_n \in \Omega_j$  (por ejemplo, para 2-PAM  $\Omega_j = (+1, -1)$ ),  $Pr\{\hat{a}_n^{(i)} = \Omega_k\}$  determina la probabilidad de detectar un símbolo  $\hat{a}_n^{(i)} \in \Omega_k$  en la  $i^{th}$  iteración. La entropía de  $\hat{a}_n^{(i)}$  se calcula según la Ec. (4.9) como

$$H(\hat{a}_n^{(i)}) = \sum_{k \in \Omega_k} P(\hat{a}_n^{(i)}) \log_2 \frac{1}{P(\hat{a}_n^{(i)})} \quad (4.18)$$

y la entropía condicional es

$$H(\hat{a}_n^{(i)}/a_n) = \sum_{k \in \Omega_k} \sum_{j \in \Omega_j} P(\hat{a}_n^{(i)}, a_n) \log_2 \frac{1}{P(\hat{a}_n^{(i)}/a_n)} \quad (4.19)$$

donde  $P(\hat{a}_n^{(i)}, a_n)$  es la probabilidad conjunta de detectar un símbolo  $\hat{a}_n^{(i)}$  a la salida y haber transmitido un símbolo  $a_n$  y  $P(\hat{a}_n^{(i)}/a_n)$  es la probabilidad condicional de detectar el símbolo  $\hat{a}_n^{(i)}$  si se transmitió el símbolo  $a_n$ . Reemplazando (4.18) y (4.19) en (4.17) obtenemos

$$I(a_n, \hat{a}_n^{(i)}) = \sum_{k \in \Omega_k} P(\hat{a}_n^{(i)}) \log_2 \frac{1}{P(\hat{a}_n^{(i)})} - \sum_{k \in \Omega_k} \sum_{j \in \Omega_j} P(\hat{a}_n^{(i)}, a_n) \log_2 \frac{1}{P(\hat{a}_n^{(i)}/a_n)} \quad (4.20)$$

Considerando las siguientes igualdades

$$\begin{aligned} Pr\{\hat{a}_n^{(i)} = \Omega_k, a_n = \Omega_j\} &= Pr\{\hat{a}_n^{(i)} = \Omega_k/a_n = \Omega_j\}Pr\{a_n = \Omega_j\} \\ &= Pr\{a_n = \Omega_j/\hat{a}_n^{(i)} = \Omega_k\}Pr\{\hat{a}_n^{(i)} = \Omega_k\} \end{aligned} \quad (4.21)$$

$$Pr\{\hat{a}_n^{(i)} = \Omega_k\} = \sum_{j \in \Omega_j} Pr\{\hat{a}_n^{(i)} = \Omega_k/a_n = \Omega_j\}Pr\{a_n = \Omega_j\} \quad (4.22)$$

$$Pr\{a_n = \Omega_j\} = \sum_{k \in \Omega_k} Pr\{a_n = \Omega_j/\hat{a}_n^{(i)} = \Omega_k\}Pr\{\hat{a}_n^{(i)} = \Omega_k\} \quad (4.23)$$

y operando matemáticamente sobre (4.20), resulta

$$\begin{aligned} I(a_n, \hat{a}_n^{(i)}) &= \sum_{k \in \Omega_k} \sum_{j \in \Omega_j} P(\hat{a}_n^{(i)}/a_n) P(a_n) \log_2 \frac{1}{P(\hat{a}_n^{(i)})} - \\ &\quad \sum_{k \in \Omega_k} \sum_{j \in \Omega_j} P(\hat{a}_n^{(i)}/a_n) P(a_n) \log_2 \frac{1}{P(\hat{a}_n^{(i)}/a_n)} \end{aligned} \quad (4.24)$$

$$= \sum_{k \in \Omega_k} \sum_{j \in \Omega_j} P(\hat{a}_n^{(i)}/a_n) P(a_n) \left[ \log_2 \frac{1}{P(\hat{a}_n^{(i)})} - \log_2 \frac{1}{P(\hat{a}_n^{(i)}/a_n)} \right] \quad (4.25)$$

$$= \sum_{k \in \Omega_k} \sum_{j \in \Omega_j} P(\hat{a}_n^{(i)}/a_n) P(a_n) \log_2 \frac{P(\hat{a}_n^{(i)}/a_n)}{P(\hat{a}_n^{(i)})} \quad (4.26)$$

Por último, reemplazando (4.22) en (4.26), podemos expresar la información mutua en términos de la probabilidad de transmisión y la distribución de entrada como

$$I(a_n, \hat{a}_n^{(i)}) = \sum_{k \in \Omega_k} \sum_{j \in \Omega_j} P(\hat{a}_n^{(i)}/a_n) P(a_n) \log_2 \frac{P(\hat{a}_n^{(i)}/a_n)}{\left[ \sum_{j \in \Omega_j} P(\hat{a}_n^{(i)}/a_n) P(a_n) \right]} \quad (4.27)$$

La Ec. (4.27) nos brinda una relación entre el símbolo transmitido y el detectado en cada iteración del DFFE. Si consideramos la modulación 2-PAM, la probabilidad de detección de la señal de entrada en el detector de umbral queda definida como

$$Pr\{\hat{a}_n^{(i)} = +1\} = Pr\{y_n^{(i)} \geq 0\} \quad (4.28)$$

$$Pr\{\hat{a}_n^{(i)} = -1\} = Pr\{y_n^{(i)} < 0\} \quad (4.29)$$

y sustituyendo en (4.27), obtenemos

$$I(a_n, y_n^{(i)}) = \sum_{k \in \Omega_k} \sum_{j \in \Omega_j} P(y_n^{(i)}/a_n) P(a_n) \log_2 \frac{P(y_n^{(i)}/a_n)}{\left[ \sum_{j \in \Omega_j} P(y_n^{(i)}/a_n) P(a_n) \right]} \quad (4.30)$$

donde  $y_n^{(i)}$  es la señal de entrada al detector de umbral en la  $i^{\text{th}}$  iteración. Tal como se analizará en la *Sección 4.4*,  $y_n^{(i)}$  está relacionada con las características del canal y con el error en las decisiones de las  $(i - 1)^{\text{th}}$  iteraciones anteriores. A continuación se presentan ejemplos numéricos de simulaciones, para canales con diferentes tamaños de memoria, donde la información mutua provee una clara indicación del desempeño de la arquitectura propuesta.

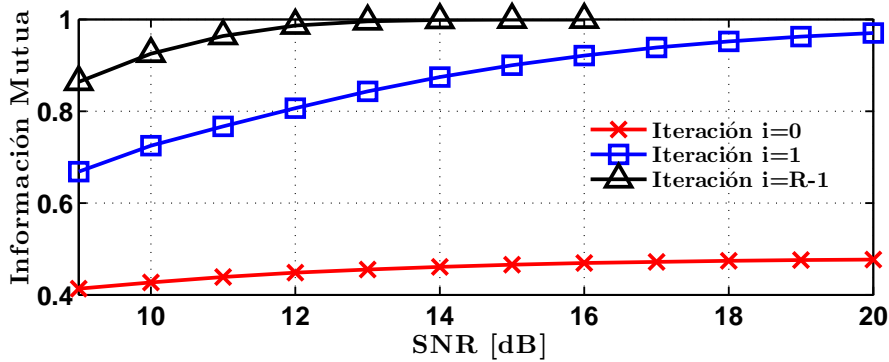
#### 4.3.3. Resultados Numéricos

Teniendo en cuenta el desarrollo anterior, se procede a mostrar en la Fig. 4.2(a) los resultados de simulación de información mutua en función de la SNR, definida como  $SNR = E|a_n|^2/\sigma^2$ . Consideramos  $a_n \in \pm 1$  y un modelo de canal con ISI post-cursor definido por

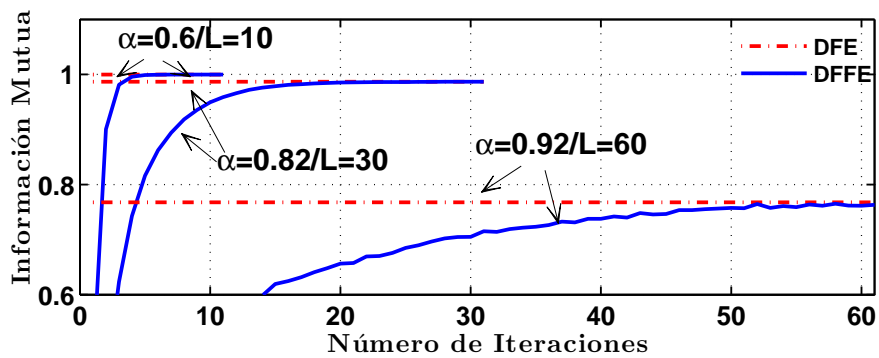
$$d_k = \begin{cases} \alpha^k & 0 < k \leq L \\ 0 & \text{otros} \end{cases}, \quad (4.31)$$

donde  $\alpha$  es un número positivo menor que uno.

En la Fig. 4.2(a) consideramos  $\alpha = 0,6$  con  $L = 10$  y un DFFE con  $R = 11$  iteraciones. Observar que la información mutua crece cuando la SNR aumenta. Para un caso límite, notar que  $I(a_n, \hat{a}_n^{(R-1)}) = I(a_n, \hat{a}_n) \rightarrow 1$  para  $SNR \rightarrow \infty$ . Para un valor dado de SNR, la mínima información mutua se verifica en la primera iteración ( $i = 0$ ). Esto puede ser comprendido desde (4.1), en la cual se obtiene la primera decisión tentativa directamente desde la muestra recibida sin ninguna cancelación de la interferencia. Sin embargo, aunque la fiabilidad de  $\hat{a}_n^{(0)}$  es baja, algo de la información del símbolo transmitido  $a_n$  es contenido en  $\hat{a}_n^{(0)}$ . Mas precisamente, esto es explotado en la segunda iteración ( $i = 1$ ), en el que se observa que la confiabilidad en  $\hat{a}_n^{(1)}$  se ha mejorado como resultado de la cancelación parcial de los post-cursos de la ISI causada por  $\hat{a}_n^{(0)}$ . Este proceso se repite en las sucesivas iteraciones hasta la última iteración  $i = R-1$ . En este punto, el DFFE puede suministrar la decisión final  $\hat{a}_n = \hat{a}_n^{(R-1)}$  con una alta confiabilidad. La Fig. 4.2(b) muestra la información mutua en función del numero de iteraciones para varios canales con ISI post-cursor con  $SNR = 15dB$ . Los valores de  $\alpha = 0,6, 0,82$  y  $0,92$  con  $L = 10, 30$  y  $60$ , respectivamente. En todos los casos, puede observarse que la confiabilidad de las decisiones tentativas mejora



(a)



(b)

Figura 4.2: Confiabilidad de las decisiones tentativas del DFFE. (a) Información mutua en función de la SNR para  $\alpha = 0,6$ ,  $L = 10$  y  $R = 11$ . (b) Información mutua en función del número de iteraciones para diferentes canales con  $SNR = 15dB$ .

con el número de iteraciones. En particular, notar que la confiabilidad de las decisiones del DFFE en  $R > L$  tiende a alcanzar al DFE. Este resultado sugiere que el desempeño del DFE y del DFFE con iteraciones  $R > L$  podrían ser similares. Este último concepto se demuestra en la Sección 4.4, donde analizaremos el desempeño del DFFE considerando el efecto de las decisiones incorrectas en las iteraciones anteriores.

## 4.4. Evaluación de Desempeño

De la Ec. (4.4) y (4.5), la señal de entrada a la  $i^{th}$  iteración,  $y_n^{(i)}$ , puede ser expresada como

$$y_n^{(i)} = \begin{cases} a_n + \sum_{k=1}^L a_{n-k} d_k + z_n & , \quad i = 0; \\ a_n + \sum_{k=1}^L a_{n-k} d_k - \sum_{k=1}^i \hat{a}_{n-k}^{(i-k)} d_k + z_n & , \quad 0 < i < L; \\ a_n + \sum_{k=1}^L a_{n-k} d_k - \sum_{k=1}^L \hat{a}_{n-k}^{(i-k)} d_k + z_n & , \quad i \geq L. \end{cases} \quad (4.32)$$

Siendo  $\Psi_n^{(i)}$  el vector de estados del DFFE en la  $i^{th}$  iteración definido por

$$\Psi_n^{(i)} = \begin{cases} (a_{n-1}, a_{n-2}, \dots, a_{n-L}) & , \quad i = 0; \\ (a_{n-1}, a_{n-2}, \dots, a_{n-L}, \hat{a}_{n-1}^{(i-1)}, \hat{a}_{n-2}^{(i-2)}, \dots, \hat{a}_{n-i}^{(0)}) & , \quad 0 < i < L; \\ (a_{n-1}, a_{n-2}, \dots, a_{n-L}, \hat{a}_{n-1}^{(i-1)}, \hat{a}_{n-2}^{(i-2)}, \dots, \hat{a}_{n-L}^{(0)}) & , \quad i \geq L. \end{cases} \quad (4.33)$$

donde definimos a  $N_i$  como la dimensión del vector de estados  $\Psi_n^{(i)}$ . Por lo tanto, observar que

$$\Psi_n^{(i)} \in \{ \psi^{(i,0)}, \psi^{(i,1)}, \dots, \psi^{(i,2^{N_i}-1)} \}, \quad (4.34)$$

donde  $\psi^{(i,0)} = (+1, +1, \dots, +1)$ ,  $\psi^{(i,1)} = (+1, +1, \dots, -1), \dots$ ,  $\psi^{(i,2^{N_i}-1)} = (-1, -1, \dots, -1)$ , son vectores  $(N_i \times 1)$ . La señal de entrada al detector de umbral en la  $i^{th}$  iteración definida por (4.32) puede ser reescrita como

$$y_n^{(i)} = g(a_n, \Psi_n^{(i)}) + z_n, \quad (4.35)$$

donde

$$g(a_n, \Psi_n^{(i)}) = \begin{cases} a_n + \sum_{k=1}^L a_{n-k} d_k & , \quad i = 0; \\ a_n + \sum_{k=1}^L a_{n-k} d_k - \sum_{k=1}^i \hat{a}_{n-k}^{(i-k)} d_k & , \quad 0 < i < L; \\ a_n + \sum_{k=1}^L a_{n-k} d_k - \sum_{k=1}^L \hat{a}_{n-k}^{(i-k)} d_k & , \quad i \geq L. \end{cases} \quad (4.36)$$

Entonces, la función de densidad de probabilidad (*Probability Density Function - PDF*) dado el símbolo de transmisión  $a_n$  puede ser expresada como

$$f_{y|a} \left( y_n^{(i)} | a_n \right) = \sum_{k=0}^{2^{N_i}-1} f_{y|a, \Psi} \left( y_n^{(i)} | a_n, \psi^{(i,k)} \right) P \left( \psi^{(i,k)} \right), \quad (4.37)$$

donde  $P \left( \psi^{(i,k)} \right) = Pr \left\{ \Psi_n^{(i)} = \psi^{(i,k)} \right\}$  y

$$f_{y|a, \Psi} \left( y_n^{(i)} | a_n, \psi^{(i,k)} \right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2} \left( y_n^{(i)} - g(a_n, \psi^{(i,k)}) \right)^2}. \quad (4.38)$$

La probabilidad de error de símbolo en la  $i^{th}$  iteración es

$$P_e^{(i)} = Pr \{ y_n^{(i)} < 0 | a_n = +1 \} Pr \{ a_n = +1 \} + Pr \{ y_n^{(i)} \geq 0 | a_n = -1 \} Pr \{ a_n = -1 \}. \quad (4.39)$$

Notar que  $Pr \{ y_n^{(i)} < 0 | a_n = +1 \}$  y  $Pr \{ y_n^{(i)} \geq 0 | a_n = -1 \}$  se puede calcular usando la PDF dada por (4.37).

#### 4.4.1. Ejemplo

En las siguientes ecuaciones consideramos un canal con post-cursos con  $L = 1$  y  $d_1 = 1$  (por ejemplo, un canal duo-binario). En la primer iteración, obtenemos

$$\Psi_n^{(0)} = (a_{n-1}), \quad (4.40)$$

$$g(a_n, \Psi_n^{(0)}) = a_n + a_{n-1}. \quad (4.41)$$

Tener en cuenta que  $N_i = 1$  y

$$\Psi_n^{(0)} \in \{ \psi^{(0,0)}, \psi^{(0,1)} \} \quad (4.42)$$

con  $\psi^{(0,0)} = (+1)$  y  $\psi^{(0,1)} = (-1)$ . La transmisión de símbolos se suponen independientes e idénticamente distribuidas con

$$Pr \{ a_n = +1 \} = Pr \{ a_n = -1 \} = \frac{1}{2} \quad \forall n. \quad (4.43)$$

En esta situación, desde (4.40) y (4.42) notar que

$$P \left( \psi^{(0,k)} \right) = \frac{1}{2}, \quad k = 0, 1. \quad (4.44)$$

La probabilidad de error  $P_e^{(0)}$  se puede derivar de (4.39) y

$$\begin{aligned}
f_{y|a}(y_n^{(0)}|a_n) &= \sum_{k=0}^1 f_{y|a,\Psi}(y_n^{(0)}|a_n, \psi^{(0,k)}) P(\psi^{(0,k)}) \\
&= \frac{1}{2} \sum_{k=0}^1 f_{y|a,\Psi}(y_n^{(0)}|a_n, \psi^{(0,k)}).
\end{aligned} \tag{4.45}$$

entonces

$$\begin{aligned}
P_e^{(0)} &= \frac{1}{2} \left[ \frac{1}{2} Q\left(\frac{2}{\sigma}\right) + \frac{1}{2} Q\left(\frac{0}{\sigma}\right) + \frac{1}{2} \left[ 1 - Q\left(\frac{-2}{\sigma}\right) \right] + \frac{1}{2} \left[ 1 - Q\left(\frac{0}{\sigma}\right) \right] \right] \\
&= \frac{1}{2} \left[ Q\left(\frac{2}{\sigma}\right) + Q\left(\frac{0}{\sigma}\right) \right]
\end{aligned} \tag{4.46}$$

donde

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt. \tag{4.47}$$

En la Fig. 4.3 se grafican las distribuciones de probabilidad en las primeras cuatro (4) iteraciones del DFFE considerando una  $SNR = 15dB$  para el canal duo binario. La Fig. 4.3(a) ejemplifica el concepto de (4.45), donde queda en evidencia que la  $P_e^{(0)}$  se encuentra directamente relacionada con las características del canal.

En la segunda iteración, obtenemos

$$\Psi_n^{(1)} = (a_{n-1}, \hat{a}_{n-1}^{(0)}), \tag{4.48}$$

$$g(a_n, \Psi_n^{(1)}) = a_n + a_{n-1} - \hat{a}_{n-1}^{(0)}. \tag{4.49}$$

En este caso, observar que  $N_i = 2$  y

$$\Psi_n^{(1)} \in \{\psi^{(1,0)}, \psi^{(1,1)}, \psi^{(1,2)}, \psi^{(1,3)}\} \tag{4.50}$$

con  $\psi^{(1,0)} = (+1, +1)$ ,  $\psi^{(1,1)} = (+1, -1)$ ,  $\psi^{(1,2)} = (-1, +1)$ , y  $\psi^{(1,3)} = (-1, -1)$ . De (4.43) y (4.48), obtenemos

$$\begin{aligned}
Pr\{\Psi_n^{(1)}\} &= Pr\{a_{n-1}, \hat{a}_{n-1}^{(0)}\} \\
&= Pr\{\hat{a}_{n-1}^{(0)}|a_{n-1}\} Pr\{a_{n-1}\} \\
&= \frac{1}{2} Pr\{\hat{a}_{n-1}^{(0)}|a_{n-1}\}.
\end{aligned} \tag{4.51}$$

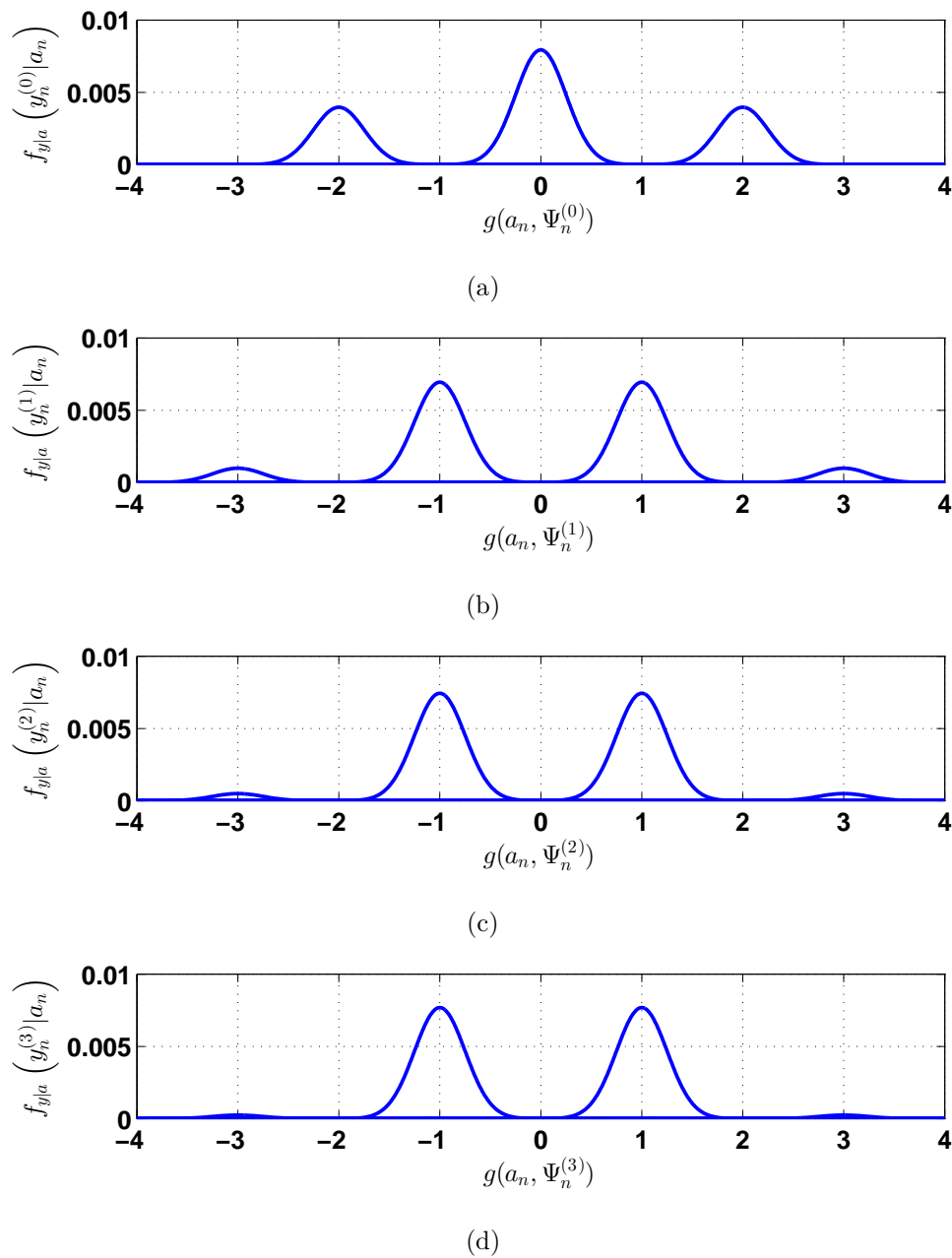


Figura 4.3: Las gráficas representan la distribución de probabilidad a la entrada del detector de umbral en las cuatro (4) primeras iteraciones del DFFE considerando el canal duo binario para una  $SNR = 15dB$ .

Sabiendo que

$$Pr \{ \hat{a}_{n-1}^{(0)} | a_{n-1} \} = P_e^{(0)}, \quad \hat{a}_{n-1}^{(0)} \neq a_{n-1}, \quad (4.52)$$

con  $P_e^{(0)}$  siendo la probabilidad de error de la primera iteración, la probabilidad (4.51) resulta



$$P(\psi^{(1,0)}) = P(\psi^{(1,3)}) = \frac{1}{2}(1 - P_e^{(0)}), \quad (4.53)$$

$$P(\psi^{(1,1)}) = P(\psi^{(1,2)}) = \frac{1}{2}P_e^{(0)}, \quad (4.54)$$

Por otra parte, teniendo en cuenta que

$$g(a_n, \psi^{(1,0)}) = g(a_n, \psi^{(1,3)}) = a_n, \quad (4.55)$$

$$g(a_n, \psi^{(1,1)}) = a_n + 2 \quad (4.56)$$

$$g(a_n, \psi^{(1,2)}) = a_n - 2 \quad (4.57)$$

podemos derivar la PDF como

$$f_{y|a, \Psi}(y_n^{(1)}|a_n, \psi^{(1,0)}) = f_{y|a, \Psi}(y_n^{(1)}|a_n, \psi^{(1,3)}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_n^{(1)} - a_n)^2} \quad (4.58)$$

$$f_{y|a, \Psi}(y_n^{(1)}|a_n, \psi^{(1,1)}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_n^{(1)} - a_n - 2)^2} \quad (4.59)$$

$$f_{y|a, \Psi}(y_n^{(1)}|a_n, \psi^{(1,2)}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_n^{(1)} - a_n + 2)^2} \quad (4.60)$$

donde un ejemplo de la distribución aplicando este concepto se grafica en la Fig. 4.3(b). Resulta de interés resaltar el cambio de la distribución de probabilidad a consecuencia de la compensación de la primera iteración. Por lo tanto, la  $P_e^{(1)}$  queda definida por

$$\begin{aligned} P_e^{(1)} &= \frac{1}{2} [Pr\{y_n^{(1)} < 0|a_n = +1\} + Pr\{y_n^{(1)} \geq 0|a_n = -1\}] \\ &= (1 - P_e^{(0)}) Q\left(\frac{1}{\sigma}\right) + \frac{1}{2}P_e^{(0)} \left[1 - Q\left(\frac{1}{\sigma}\right)\right] + \frac{1}{2}P_e^{(0)} Q\left(\frac{3}{\sigma}\right) \\ &= Q\left(\frac{1}{\sigma}\right) + P_e^{(0)} \left[\frac{1}{2} - \frac{3}{2}Q\left(\frac{1}{\sigma}\right) + \frac{1}{2}Q\left(\frac{3}{\sigma}\right)\right] \end{aligned} \quad (4.61)$$

Generalizando, para  $i > 0$  es posible mostrar que

$$P(\psi^{(i,0)}) = P(\psi^{(i,3)}) = \frac{1}{2}(1 - P_e^{(i-1)}), \quad (4.62)$$

$$P(\psi^{(i,1)}) = P(\psi^{(i,2)}) = \frac{1}{2}P_e^{(i-1)}. \quad (4.63)$$

Además, teniendo en cuenta que

$$g(a_n, \psi^{(i,0)}) = g(a_n, \psi^{(i,3)}) = a_n, \quad (4.64)$$

$$g(a_n, \psi^{(i,1)}) = a_n + 2, \quad (4.65)$$

$$g(a_n, \psi^{(i,2)}) = a_n - 2, \quad (4.66)$$

es posible ver que

$$f_{y|a, \Psi} (y_n^{(i)} | a_n, \psi^{(i,0)}) = f_{y|a, \Psi} (y_n^{(i)} | a_n, \psi^{(i,3)}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2} (y_n^{(i)} - a_n)^2}, \quad (4.67)$$

$$f_{y|a, \Psi} (y_n^{(i)} | a_n, \psi^{(i,1)}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2} (y_n^{(i)} - a_n - 2)^2}, \quad (4.68)$$

$$f_{y|a, \Psi} (y_n^{(i)} | a_n, \psi^{(i,2)}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2} (y_n^{(i)} - a_n + 2)^2}. \quad (4.69)$$

Por lo tanto, considerando las expresiones (4.42)-(4.69), es posible mostrar que la probabilidad de error ( $P_e^{(i)}$ ) en la  $i^{th}$  iteración es

$$\begin{aligned} P_e^{(i)} &= \frac{1}{2} [Pr\{y_n^{(i)} < 0 | a_n = +1\} + Pr\{y_n^{(i)} \geq 0 | a_n = -1\}] \\ &= (1 - P_e^{(i-1)}) Q\left(\frac{1}{\sigma}\right) + \frac{1}{2} P_e^{(i-1)} \left[1 - Q\left(\frac{1}{\sigma}\right)\right] + \frac{1}{2} P_e^{(i-1)} Q\left(\frac{3}{\sigma}\right) \\ &= Q\left(\frac{1}{\sigma}\right) + P_e^{(i-1)} \left[\frac{1}{2} - \frac{3}{2} Q\left(\frac{1}{\sigma}\right) + \frac{1}{2} Q\left(\frac{3}{\sigma}\right)\right] \end{aligned} \quad (4.70)$$

Operando en forma recursiva la probabilidad de error (4.70), de la siguiente forma

$$P_e^{(1)} = Q\left(\frac{1}{\sigma}\right) + P_e^{(0)} \underbrace{\left[\frac{1}{2} - \frac{3}{2} Q\left(\frac{1}{\sigma}\right) + \frac{1}{2} Q\left(\frac{3}{\sigma}\right)\right]}_A = Q\left(\frac{1}{\sigma}\right) + P_e^{(0)} A \quad (4.71)$$

$$P_e^{(2)} = Q\left(\frac{1}{\sigma}\right) + P_e^{(1)} A = Q\left(\frac{1}{\sigma}\right) [1 + A] + P_e^{(0)} A^2 \quad (4.72)$$

$$P_e^{(3)} = Q\left(\frac{1}{\sigma}\right) + P_e^{(2)} A = Q\left(\frac{1}{\sigma}\right) [1 + A + A^2] + P_e^{(0)} A^3 \quad (4.73)$$

$$\vdots \quad (4.74)$$

$$P_e^{(i)} = Q\left(\frac{1}{\sigma}\right) + P_e^{(i-1)} A = Q\left(\frac{1}{\sigma}\right) [1 + A + A^2 + \dots + A^{i-1}] + P_e^{(0)} A^i \quad (4.75)$$

obtenemos  $P_e^{(i)}$  como

$$P_e^{(i)} = Q\left(\frac{1}{\sigma}\right) \left[ \frac{A^i - 1}{A - 1} \right] + P_e^{(0)} A^i \quad (4.76)$$

Para alta SNR (por ejemplo,  $1/\sigma \gg 1$ ) y  $i \gg 1$ ,  $A^i \rightarrow 0$  y  $A \rightarrow 1/2$ , entonces reemplazando en (4.76)

$$P_e^{(i)} \approx 2Q\left(\frac{1}{\sigma}\right), \quad i \gg 1. \quad (4.77)$$

Dado que la probabilidad de error del DFE con la propagación del error viene dado por [6]

$$P_e^{DFE} \simeq 2^L Q\left(\frac{1}{\sigma}\right), \quad (4.78)$$

de (4.77) podemos concluir que, para un número de iteraciones suficientemente grande, el rendimiento del DFFE en presencia de un canal duo-binario ( $L = 1$ ) se reduce a la alcanzada por el DFE con propagación de error. Como veremos más adelante, el número apropiado de iteraciones depende en gran medida tanto de la potencia de ruido como de la dispersión del canal. Por último, es posible tener en cuenta que las conclusiones derivadas de este ejemplo se pueden ampliar para canales con mayor memoria  $L > 1$ .

#### 4.4.2. Impacto de la Estimación Incorrecta del Canal

Dado que el DFFE es una solución atractiva para canales con gran memoria ( $L \gg 1$ ), es posible mostrar que el impacto de la estimación incorrecta del canal es similar para ambos ecualizadores (DFE y DFFE). La muestra de entrada al receptor  $y_n$ , tanto para el DFFE como para el DFE, puede expresarse como

$$y_n = a_n + \sum_{k=1}^L a_{n-k} d_k + z_n, \quad (4.79)$$

donde  $d_k$  y  $k = 1, \dots, L$  son los post-cursores de la ISI,  $a_n$  es el símbolo transmitido, y  $z_n$  es ruido Gaussiano con potencia  $\sigma^2$ . La señal (4.79) puede ser reescrita como

$$\begin{aligned} y_n &= a_n + \sum_{k=1}^L a_{n-k} d_k + z_n, \\ &= a_n + \sum_{k=1}^L a_{n-k} \hat{d}_k + \sum_{k=1}^L a_{n-k} \Delta_k + z_n \end{aligned} \quad (4.80)$$

## Capítulo 4. Nueva Arquitectura Ecuatorial Iterativa de Baja Complejidad

donde  $\hat{d}_k$  y  $\Delta_k$  denotan los coeficientes estimados en el receptor y el error de estimación, respectivamente (por ejemplo,  $d_k = \hat{d}_k + \Delta_k$ ). Partiendo del teorema central del límite y dado que  $L \gg 1$  y los símbolos  $a_n$  son asumidos independientes e idénticamente distribuidos (iid), el término

$$r_n = \sum_{k=1}^L a_{n-k} \Delta_k \quad (4.81)$$

puede ser modelado como una variable aleatoria Gaussiana con media cero con varianza  $\sigma_r^2$ . Por consiguiente, la señal a la entrada del receptor con estimación incorrecta del canal puede ser vista como

$$y_n = a_n + \sum_{k=1}^L a_{n-k} \hat{d}_k + \tilde{z}_n, \quad (4.82)$$

donde

$$\tilde{z}_n = r_n + z_n \quad (4.83)$$

es ruido Gaussiano de media cero y potencia  $\sigma_r^2 + \sigma^2$ . Por lo tanto, en base a la suposición genérica de (4.81), (4.82) y (4.83) concluimos que el impacto de la estimación incorrecta del canal sobre el desempeño del DFE y DFFE son similares.

### 4.4.3. Resultados Numéricos

Una estimación teórica basada en la probabilidad de error proporciona una herramienta eficaz para el diseño de los parámetros del DFFE. El proceso de diseño es simple y consta de dos pasos principales

- Estimar el número de coeficientes de los filtros directo y realimentado de acuerdo a la respuesta del canal (de manera similar al diseño del DFE).
- Estimar el número de iteraciones del DFFE en base a la evaluación del desempeño. Esta tarea puede lograrse también mediante el uso de simulaciones por computadora. Como punto inicial, se debe fijar  $R = L + 1$ .

La Fig. 4.4 ilustra la curva de BER como una función de la SNR y el número de iteraciones. En este caso, usamos un canal con ISI post-cursor definido por  $d_k = \alpha^k, 0 <$

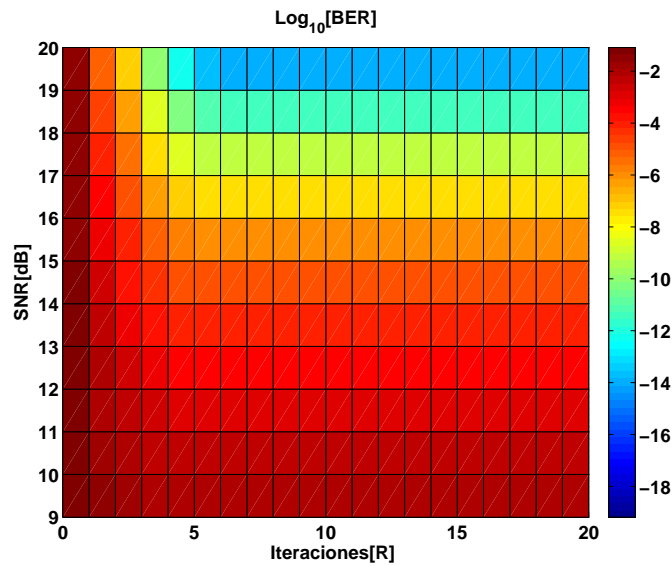


Figura 4.4: BER en función de la SNR y el número de iteraciones. Canal con post-cursores  $\alpha = 0,5/L = 6$ .

$k \leq L$  con  $\alpha = 0,5$ ,  $L = 6$ , y  $R = 20$ . Podemos observar que el desempeño del DFFE para  $R > 6$  es similar en todas las iteraciones. Por consiguiente, concluimos que el DFFE con  $R = L + 1$  consigue el mismo desempeño que el DFE tradicional, como se puede verificar en la Fig. 4.5. Para el DFFE, notar la excelente igualdad entre los valores derivados de simulaciones por computadora y la predicción teórica dadas por (4.39).

El desempeño del DFE y un DFFE adaptivo con  $R = L + 1$  iteraciones en presencia de diferentes canales dispersivos se evalúa en la Fig. 4.6. Consideramos cuatro canales:  $\alpha = 0,6, 0,82, 0,92$ , y  $0,95$  con  $L = 10, 30, 60$ , y  $100$ , respectivamente. El DFFE adaptivo se implementó con el algoritmo LMS [6] usando la decisión final para estimar la señal error. En todos los casos, se puede observar que el DFFE y el DFE alcanzan esencialmente el mismo desempeño. Este resultado es coincidente con el análisis teórico presentado en la Sección 4.4.2, donde se mostró el impacto de la estimación incorrecta del canal sobre el desempeño del DFE y DFFE.

## 4.5. Desempeño del DFFE en un Canal de Cobre

El contenido de esta sección pretende mostrar un entorno de trabajo real, en el cual se puede aplicar la estructura ecualizadora que esta Tesis propone. El entorno que se

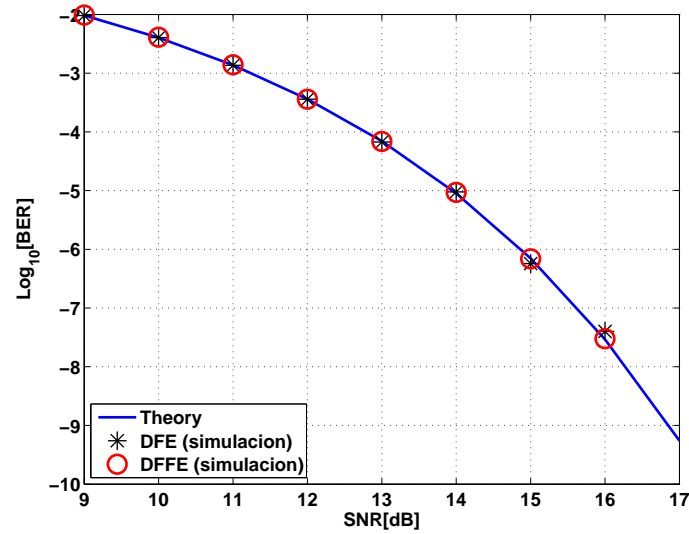


Figura 4.5: Desempeño del DFFE con  $R = 7$  y el DFE. Canal con post-cursos  $\alpha = 0,5/L = 6$ .

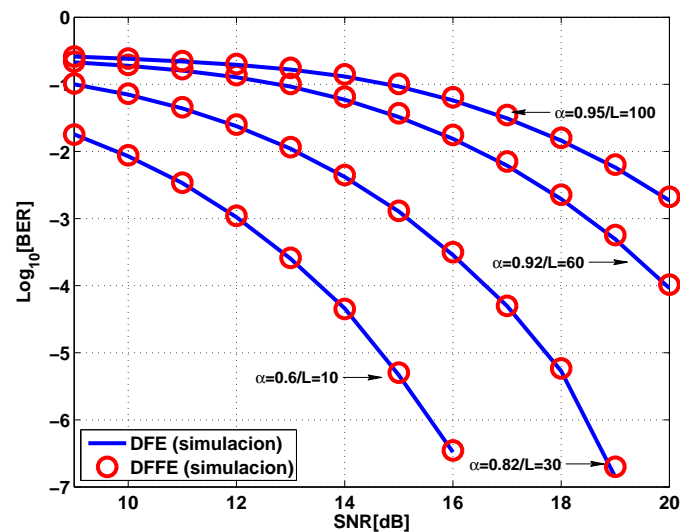


Figura 4.6: Desempeño del DFE y el DFFE adaptivo con  $R = L+1$  para diferentes canales con ISI post-cursor.

propone de evaluación es un centro de datos (*data center*), donde los canales que es posible encontrar son un excelente ejemplo de aplicación. Además, se detalla la herramienta utilizada para verificar, por medio de simulaciones, el desempeño del DFFE.

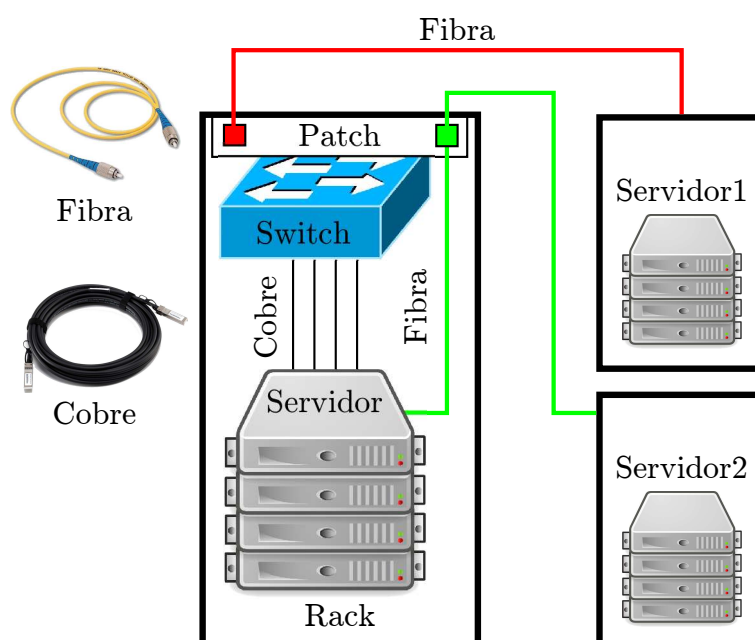


Figura 4.7: Esquemático general de conexiones entre distintos dispositivos que componen un rack de un centro de datos.

### 4.5.1. Escenario

La tecnología de interconexión de enlace serie ha tomado gran relevancia en la última década con la flexibilidad que disponen los nuevos dispositivos inteligentes para conectarse a la red. Los usos básicos son para almacenamiento y compartir música, fotos y vídeos a través de una interfaz fácil de usar, a relativamente bajo costo. Para brindar flexibilidad y disponibilidad de acceso a estas necesidades, los centros de datos (*Data Centers*) juegan un papel fundamental. Los centros de datos se componen básicamente de dispositivos ruteadores, de almacenamiento y sistemas que brindan diferentes servicios a los usuarios, montados en un rack. Esto es un gabinete que tiene la capacidad de contener estos dispositivos en forma ordenada y brinda un espacio físico de interconexión. El medio físico utilizado para conectar los diferentes dispositivos puede ser backplane, cable de cobre y fibra óptica.

- El backplane es una placa de circuito impreso (*Printed Circuit Board - PCB*) que conecta mecánicamente componentes electrónicos usando pistas conductoras a partir de hojas de cobre laminadas sobre un sustrato no conductor eléctricamente. Se la utiliza para la conexión entre placas dentro del rack.

## Capítulo 4. Nueva Arquitectura Ecualizadora Iterativa de Baja Complejidad

---

- El cable de cobre se lo emplea para conexiones Intra e Inter rack. Se suele utilizar configuración tales como cable de par trenzado sin blindaje (*Unshielded twisted pair* - *UTP*) y Twinax. La diferencia entre ambos radica en el alcance máximo que se puede lograr, donde es habitual utilizar cable UTP para conexiones entre racks y Twinax para conexiones inter rack.
- La fibra óptica utilizada para conexiones entre racks logrando un mayor alcance que el cable de cobre.

En la Fig. 4.7 se observa la forma de interconexión básica entre algunos dispositivos que componen un centro de datos, tal como se detalló en la descripción anterior. En la *Sección 4.5.3* se mostrará un ejemplo de canal práctico, considerando un canal de cobre. La aplicación de técnicas de ecualización en estos enlaces para compensar la ISI, se encuentra limitada a la aplicación del LE y DFE. Esto se debe a que el número de post-cursos del canal, en la mayoría de los casos, son muy grande ( $> 20$ ) lo que hace prohibitiva la implementación de técnicas óptimas como el MLSDF por su complejidad. Además, como se ha desarrollado en el *Capítulo 3*, las alternativas existentes también tienen un incremento exponencial de la complejidad. Esto convierte al DFFE es una excelente opción para este tipo de entorno.

### 4.5.2. Simulador

Los resultados numéricos obtenidos en esta Tesis se basaron en un simulador desarrollado en el lenguaje de programación C++. La arquitectura del simulador, la cual se detalla en la Fig. 4.8, consiste en un sistema de comunicaciones digital típico compuesto por

- **Transmisor:** Genera una señal modulada en amplitud (PAM).
- **Canal:** Representado con un filtro FIR con la capacidad de cargar la respuesta al impulso del canal a simular.
- **Generador de ruido Gaussiano**
- **Generador de Reloj:** Utilizado para simular el comportamiento de los registros, dando la posibilidad de evaluar el efecto de los diferentes retardos en la cadena de



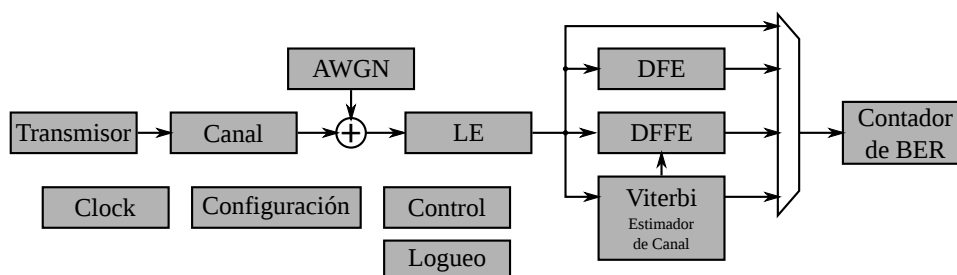


Figura 4.8: Diagrama en bloques de la arquitectura del simulador utilizado para determinar el desempeño de los ecualizadores. El lenguaje de programación en el cual se desarrolló el simulador fue C++.

recepción. Esto tiene gran importancia al momento de realizar la adaptación de los coeficientes de los ecualizadores.

- **Receptor:** Encargado de realizar el procesamiento de la señal recibida aplicando diferentes técnicas de ecualización. Para todos los casos, es posible utilizar el algoritmo de adaptación LMS [9]. Particularmente, el VA [19, 20, 21] utiliza un estimador de canal o el DFFE de donde extrae los coeficientes para la estimación de las métricas de rama (*branch metric*).
- **Bloques Extras:** Estos son el bloque de configuración, control, logueo y contador de BER. Son sistemas adicionales que permiten configurar las constantes de los diferentes algoritmos, selección de la técnica de ecualización que se quiere simular, control del algoritmo de adaptación y logueo de variables internas.

Esta herramienta permitió la estimación del desempeño del DFFE y compararlas con otras técnicas de ecualización conocidas. En la siguiente sección se muestran los resultados obtenidos para dos ejemplos prácticos.

### 4.5.3. Ejemplo

En la *Sección 4.2* se desarrollaron los conceptos básicos del ecualizador iterativo, donde se plantea una arquitectura capaz de obtener un desempeño similar al DFE. En esta sección se esboza un ejemplo práctico considerando un canal de cobre. Lo que se pretende mostrar es la flexibilidad de configuración del DFFE que hace posible obtener el desempeño deseado. En los resultados numéricos que a continuación se mostrarán, la arquitectura

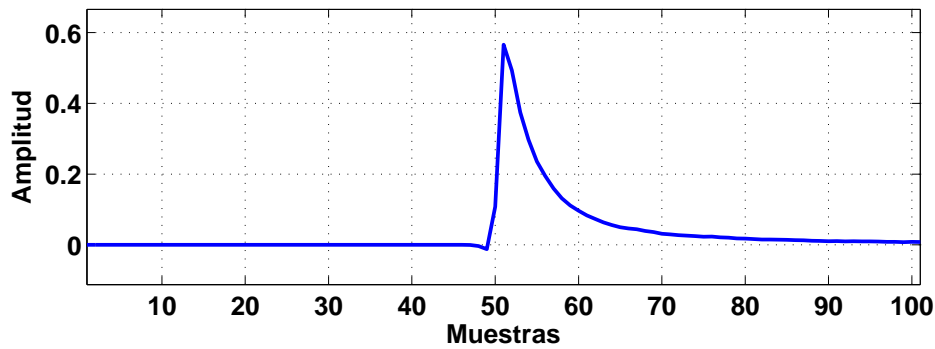


Figura 4.9: *Respuesta al impulso de un canal de cobre muestreado a una frecuencia de 32GHz.*

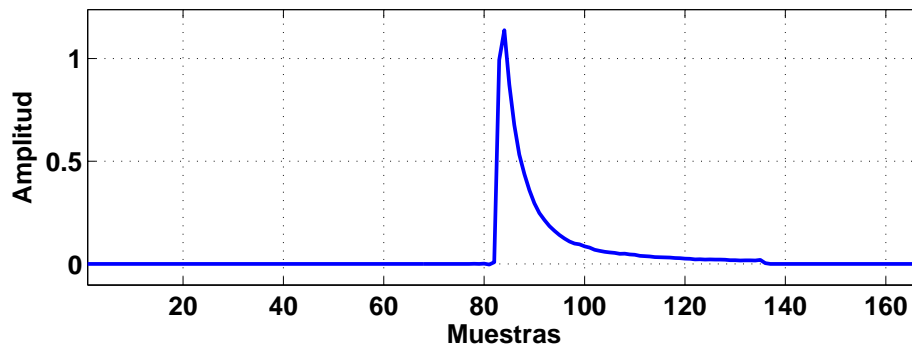
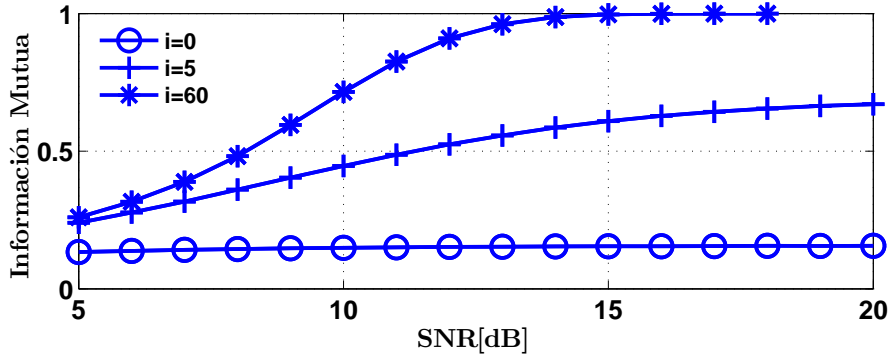


Figura 4.10: *Respuesta al impulso del canal equivalente después de ser compensado parcialmente por el ecualizador lineal.*

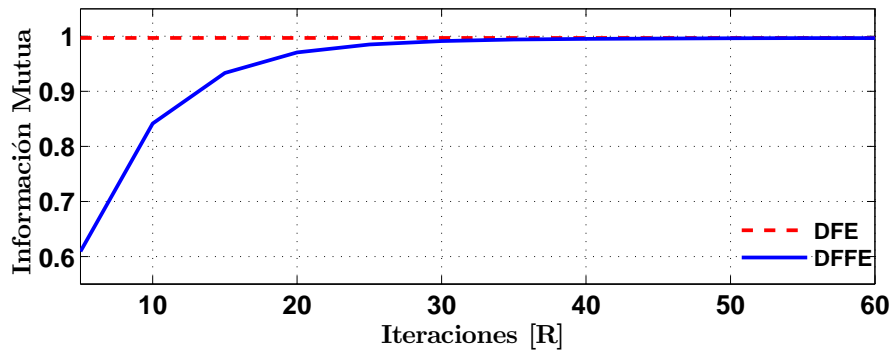
del receptor contempla un ecualizador lineal a la entrada y utiliza el algoritmo de adaptación LMS para la adaptación de los coeficientes de todos los algoritmos de ecualización (ver Fig. 4.8). Para la estimación de desempeño de los algoritmos de ecualización se toma como referencia un canal normalizado ( $\sum_{k=0}^L |q_k|^2 = 1$ ) y la  $SNR = E\{|a_n|^2\}/\sigma^2$ .

En primer lugar, evaluaremos el desempeño del DFFE para una canal de cobre. La respuesta al impulso que se considera pertenece a un cable Twinax de 5 metros con conector SFP (*Small Form-factor Pluggable - SFP*). Es ideal para conectar los servidores con los swith a nivel superior del rack (*Top of Rack - ToR*). En la Fig. 4.7, la cual representa un esquema simplificado de la interconexión de los racks de un centro de datos, se muestra en que lugar se utiliza este tipo de cable.

En este ejemplo solo consideraremos la arquitectura del DFFE desarrollada en la *Sección 4.2*. Esta consiste únicamente en iteración. Recordemos que



(a)



(b)

Figura 4.11: Confiabilidad de las decisiones tentativas del DFFE para un canal de cobre. (a) Información mutua en función de la SNR para tres (3) iteraciones diferentes. (b) Información mutua en función del número de iteración para una SNR = 15dB.

$$\hat{a}_n^{(i)} = \begin{cases} \mathcal{Q}(y_n) & , \quad i = 0 \\ \mathcal{Q}\left(y_n - \sum_{k=1}^i \hat{a}_{n-k}^{(i-k)} d_k\right) & , \quad 0 < i < L \\ \mathcal{Q}\left(y_n - \sum_{k=1}^L \hat{a}_{n-k}^{(R-1-k)} d_k\right) & , \quad i \geq L \end{cases} \quad (4.84)$$

donde  $i = 0$  es la iteración inicial,  $0 < i < L$  considera que las iteraciones son menores al número de coeficientes del canal ( $R < L$ ) y  $i \geq L$  ocurre cuando las iteraciones son mayores que el número de coeficientes del canal ( $R \geq L$ ). Bajo estas restricciones se plantea una arquitectura del DFFE con el número de iteraciones igual a  $R = L + 1$ . Para este ejemplo,  $L$  representa el número de post-cursores del canal equivalente.

El estudio del diseño del DFFE para este ejemplo se realiza en base a un canal de cobre, detallado en los párrafos anteriores, con una respuesta al impulso que se observa en la Fig. 4.9 y muestreado a  $32GHz$ . Tiene la particularidad de tener un gran número

## Capítulo 4. Nueva Arquitectura Ecualizadora Iterativa de Baja Complejidad

de coeficientes, con lo que utilizar una arquitectura de baja complejidad es prioritario para la implementación. Después de ser ecualizado el canal por intermedio del LE, el canal equivalente a la entrada del DFFE tiene una respuesta al impulso casi exponencial, esto se aprecia en la Fig. 4.10. Se considera que el número de post-cursos para el canal equivalente es aproximadamente igual a  $L = 60$ .

Siguiendo las reglas de diseño planteadas en el capítulo anterior, en primer lugar calculamos la información mutua. Recordemos que

$$I(a_n, \hat{a}_n^{(i)}) = H(\hat{a}_n^{(i)}) - H(\hat{a}_n^{(i)}/a_n) \quad (4.85)$$

donde  $H(\cdot)$  y  $H(\cdot/\cdot)$  es la entropía y la entropía condicional, respectivamente. Esto nos permite estimar el número de iteraciones que se necesitaran en el diseño del DFFE. Se evalúa esta métrica para diferente número de iteraciones en función de la  $SNR$ , esto se observa en la Fig. 4.11(a). El análisis da como resultado que con  $R = 60$  iteraciones se alcanza un valor máximo de información mutua. La Fig. 4.11(b) certifica esta estimación de las iteraciones, ya que relaciona la información mutua en función de las iteraciones para una  $SNR = 15dB$ . En esta última gráfica, se aprecia que, tanto el DFE como el DFFE, alcanzan el mismo comportamiento.

La mejora del desempeño del DFFE en función de las iteraciones, también se hace visible analizando las curvas de BER. En la Fig. 4.12(a) se detallan cinco diferentes configuraciones del DFFE para  $R = [20, 30, 40, 50, 60]$  iteraciones y se comparan con el DFE. Al igual que en el análisis de información mutua, la configuración con  $R = 60$  tiene el mejor desempeño, este último se resume en la Fig. 4.12(b).

En conclusión, para este ejemplo, la arquitectura del DFFE se la puede definir con un número de iteraciones muy próximas al número de coeficientes del canal ( $R = L + 1$ ).

### 4.6. DFFE más Algoritmo de Viterbi

En las secciones anteriores, se mostró el desempeño del DFFE por medio del análisis teórico y resultados de simulaciones considerando un canal de cobre. Todos los resultados concluyen que el desempeño del DFFE es similar al DFE bajo las mismas condiciones de operación. La ventaja que presenta la arquitectura del DFFE es que la implementación es totalmente directa (*forward*), esto nos permite una mayor flexibilidad para interactuar

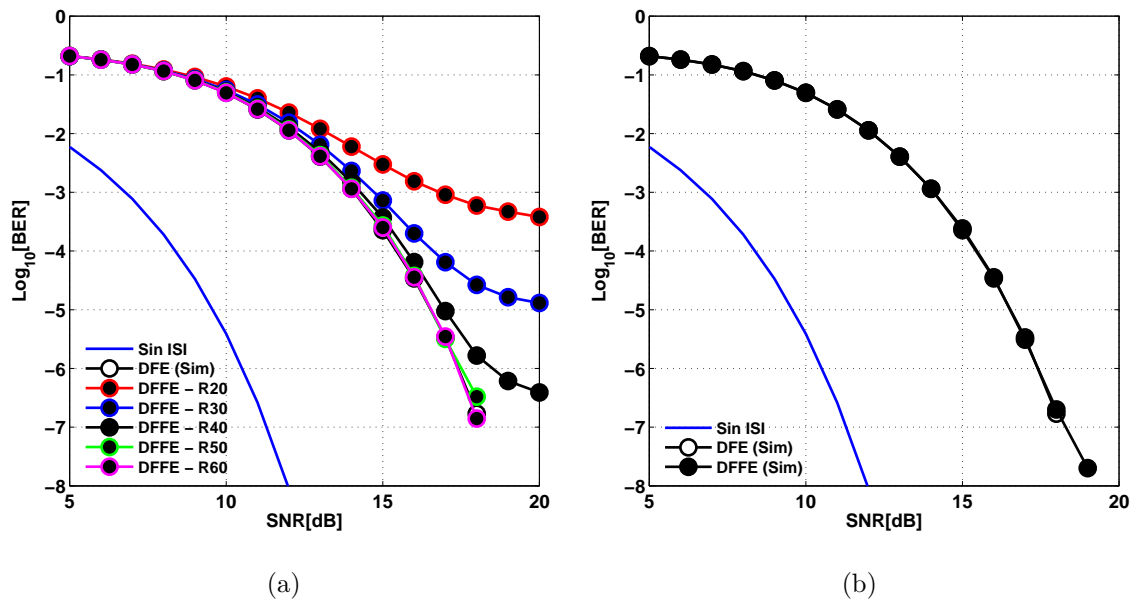


Figura 4.12: Curvas de desempeño (BER) del DFFE comparado con el DFE considerando un canal de cobre detallado en la Fig. 4.9. (a) Comparativa de curvas de desempeño considerando diferente número de iteraciones ( $R = [20, 30, 40, 50, 60]$ ). (b) Curvas de desempeño del DFE y DFFE considerando  $R = 60$  iteraciones.

con otros tipos de algoritmos. Un caso interesante de análisis es combinar el DFFE con algún otro tipo de algoritmo de detección, como por ejemplo el detector de secuencia de máxima verosimilitud (*Maximum Likelihood Sequence Detector - MLSD*). El MLSD es un detector óptimo que se emplea en sistemas de comunicaciones digitales y una de las alternativas de implementación es utilizando el algoritmo de Viterbi. La desventaja que presenta esta técnica es el crecimiento exponencial de la complejidad en función del número de estados con los que opera el algoritmo de Viterbi dependiente de la memoria del canal. Es por ello, que resulta de interés evaluar el comportamiento en conjunto del DFFE y el VA de estados reducidos [47], con el fin de obtener una estructura ecualizadora de baja complejidad pero con un desempeño similar al DFE.

En esta sección, se emplea el análisis teórico y resultados de simulación, para mostrar la interacción entre el algoritmo de Viterbi y el DFFE para reducir la complejidad de implementación del receptor. El VA compensa parcialmente la ISI del canal y reduce la probabilidad de error en las primeras decisiones tentativas del DFFE. Para simplificar el análisis, solamente se considera la ISI generada por los post-cursos del canal.

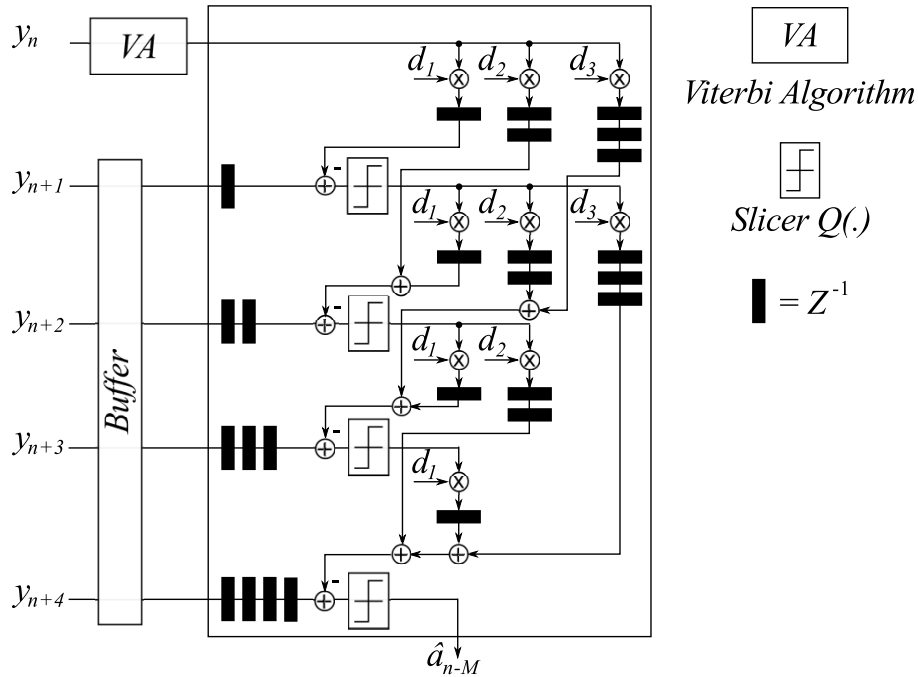


Figura 4.13: Ejemplo de tres (3) coeficientes DFFE-VA con  $L = 3$ ,  $L_v = 1$ ,  $R_v = 1$ ,  $R = 5$  y  $N = 2$ .

#### 4.6.1. Arquitectura del DFFE-VA

En el Sección 4.2 se introdujeron los conceptos del DFFE, donde se expone que la base del funcionamiento consiste es la utilización de decisiones tentativas para compensar la distorsión del canal. Repasando el criterio, las muestras de entrada al receptor DFFE  $y_n$  pueden ser modeladas como

$$y_n = a_n + \sum_{k=1}^L a_{n-k}d_k + z_n, \quad (4.86)$$

donde  $d_k$  es la ISI producida por post-cursos, con  $k = 1, \dots, L$  y  $L$  la memoria del canal,  $a_n$  es el símbolo transmitido (ejemplo  $a_n \in \{\pm 1\}$ ), y  $z_n$  es ruido blanco Gaussiano con varianza  $\sigma^2$ . Se asume que el canal es conocido en el receptor (ejemplo, las estimación del canal es perfecta). Sea  $\hat{a}_n^{(i)}$  la decisión tentativa en la  $i$ -th iteración. En la primera iteración ( $i = 0$ ), tenemos que la decisión tentativa no posee ninguna cancelación de interferencia, que puede escribirse como

$$\hat{a}_n^{(0)} = Q(y_n), \quad (4.87)$$

donde  $\mathcal{Q}(\cdot)$  es la función detector de umbral. Esta decisión tentativa entonces se puede utilizarse para cancelar la ISI postcursor introducida por el primer símbolo pasado, y por lo tanto, para mejorar la precisión de la detección. Usando los retardos de tiempo apropiados, podemos obtener la decisión tentativa de la segunda iteración tal como se expresa en

$$\hat{a}_n^{(1)} = \mathcal{Q}\left(y_n - \hat{a}_{n-k}^{(0)} d_0\right). \quad (4.88)$$

Este proceso se repite al menos hasta  $R-1$  decisiones tentativas consecutivas. En este punto, la decisión final puede obtenerse a partir

$$\hat{a}_n = \hat{a}_n^{(R-1)} = \mathcal{Q}\left(y_n - \sum_{k=1}^L \hat{a}_{n-k}^{(R-1-k)} d_k\right), \quad (4.89)$$

donde  $R$  es el número total de iteraciones. Podemos ver que la fiabilidad de las decisiones tentativas  $\hat{a}_n^{(i)}$  se incrementa a medida que el número de iteraciones aumenta [45]. De esta manera, tanto la exactitud de la estimación de la interferencia y el rendimiento del DFFE mejoran. Con el fin de mejorar la decisión inicial del DFFE, extendemos el concepto utilizando el VA. Los parámetros a considerar para el diseño de esta arquitectura son  $L_v$  el número de coeficientes compensados por el VA,  $R_v$  el número de iteraciones que utilizan decisiones de VA y  $N$  el número de estados del VA ( $2^{L_v}$ ). Después de las primeras  $R_v$  iteraciones obtenemos las correspondientes primeras  $R_v$  decisiones tentativas con cancelación de interferencia. Estas decisiones se expresan como

$$\hat{a}_n^{(R_v-1)} = \mathcal{VA}(y_n), \quad (4.90)$$

donde  $\mathcal{VA}(\cdot)$  es la función del algoritmo de Viterbi, con  $L_v < L$  y  $N = 2^{L_v}$  estados. Las decisiones tentativas provenientes del VA, se utilizan para cancelar la ISI de post-cursor introducida por los primeros  $R_v$  símbolos pasados, y por lo tanto, para mejorar la precisión de la detección. Siguiendo este enfoque, las decisiones tentativas pueden ser representadas como

$$\hat{a}_n^{(i)} = \begin{cases} \mathcal{VA}(y_n) & 0 \leq i < R_v \\ \mathcal{Q}\left(y_n - \sum_{k=1}^L \hat{a}_{n-k}^{(i-k)} d_k\right) & R_v \leq i < R \end{cases}. \quad (4.91)$$

## Capítulo 4. Nueva Arquitectura Ecualizadora Iterativa de Baja Complejidad

La Fig. 4.13 ilustra la propuesta de arquitectura DFFE-VA para un canal con memoria  $L = 3$ ,  $L_v = 1$ ,  $N = 2$ ,  $R_v = 1$  y  $R = 5$ . A diferencia de la anterior arquitectura DFFE [45], hemos incluido un buffer con la misma profundidad  $M$  como el VA. Este buffer permite sincronizar las muestras de entrada con las decisiones del VA. Por las características de implementación directa (*forward*) del DFFE-VA, es apto para aplicaciones de alta velocidad donde la paralelización es indispensable. En la *Sección 5.3.1* se profundiza el diseño en paralelo del ecualizador.

### 4.6.2. Decisiones Tentativas Proporcionadas por VA

En los canales altamente dispersivos, la probabilidad de error del DFFE en las primeras decisiones tentativas suele ser grande. Por lo tanto, hay dos opciones para compensar la dispersión del canal

- Aumentar el número de iteraciones.
- Utilizar las decisiones del VA en las primeras iteraciones del DFFE.

Claramente, la última alternativa reduce el número de iteraciones en el DFFE, ya que la probabilidad de error obtenida en la salida del VA es inferior a la estimada con el DFFE solamente.

Para un diseño adecuado del DFFE-VA, es importante estimar el número de iteraciones  $R_v$  que utilizan las decisiones tentativas proporcionados por el VA. Como una primera aproximación, es posible asumir que valores altos de  $R_v$  mejorarían el rendimiento del receptor DFFE-VA. Sin embargo, esto es particularmente cierto sólo si la probabilidad de error de las primeras  $R_v$  iteraciones es menor que en el VA.

Por ejemplo, en la Fig. 4.14 se grafica la BER en función del número de iteraciones que usan las decisiones tentativas proporcionadas por el VA ( $R_v$ ) para una SNR de  $15dB$ . Estas curvas que muestran los primeros resultados de la arquitectura propuesta se trazan considerando una modulación por desplazamiento de fase binaria (BPSK) y el canal con post-cursos definidos en la Tabla 4.1 ( $L = 10$ ). Además, el receptor DFFE-VA considera  $R = 11, 20, 30, 40, 50$  y  $60$  iteraciones en combinación con un VA de 4 estados ( $N = 4$ ).

En la Fig. 4.14,  $R_v = 0$  denota un DFFE sin VA, mientras que  $R_v = q$  ( $q > 0$ ) significa que las decisiones del VA se utilizan en las primeras  $q$  iteraciones. Hay que tener en cuenta



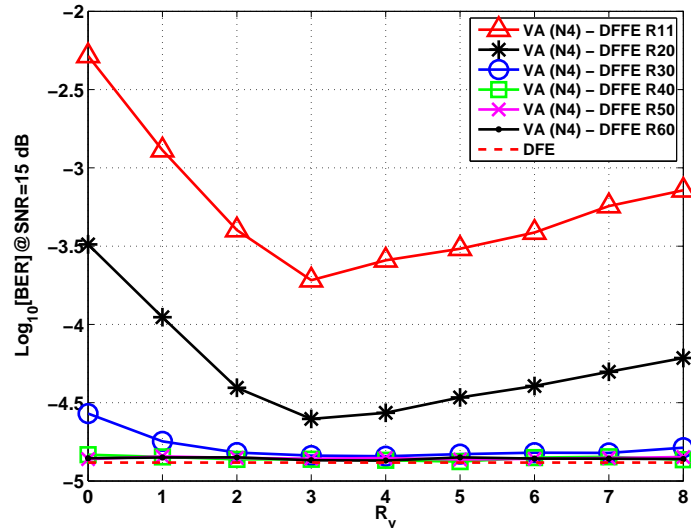


Figura 4.14: BER en función del número de iteraciones que utilizan decisiones tentativas proporcionadas por el VA ( $R_v$ ).

Tabla 4.1: Canal utilizado para verificar el funcionamiento del DFFE-VA de la Fig. 4.15

$L = 10$					
$d_0$	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$
1,0	0,6160	0,3251	-0,3581	0,1896	-0,0114
$d_6$	$d_7$	$d_8$	$d_9$	$d_{10}$	
-0,0095	-0,0195	0,0069	0,0151	0,0034	

que el DFFE-VA con  $R \geq 30$  en combinación con un VA de 4 estados con  $R_v = 3$ , logra un rendimiento similar al típico DFE. Por otro lado, cabe destacar que el rendimiento del DFFE-VA se degrada a altos valores de  $R_v$  cuando el número de iteraciones es de  $R \leq 20$ . Esto puede ser entendido por el hecho de que la fiabilidad de las decisiones provistas por las iteraciones del DFFE son mejores que las decisiones estimadas por el VA. En este ejemplo, la mejor solución de compromiso entre rendimiento y complejidad de la arquitectura DFFE-VA se consigue con  $R = 30$ .

Una forma de tener una mejor interpretación de este comportamiento, es comparar la BER para las diferentes configuraciones del DFFE-VA mencionadas anteriormente. La Fig. 4.15 ilustra la BER en función de la SNR para el canal de ISI post-cursor que se describe en la Tabla 4.1. La Fig. 4.15 representa el rendimiento del DFFE con  $R = 30$

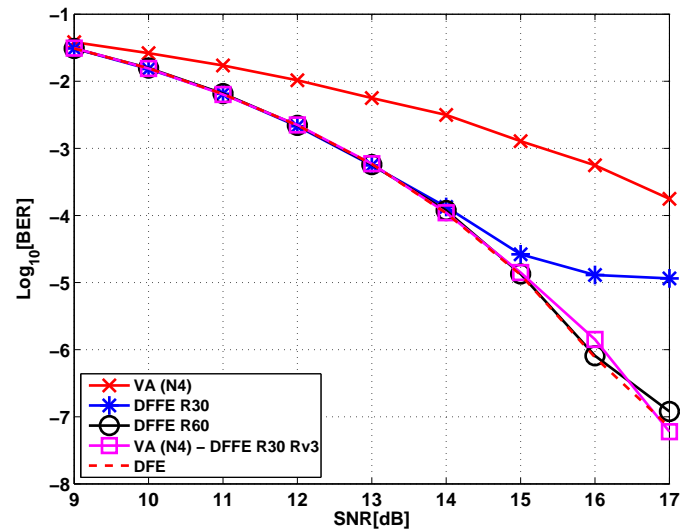


Figura 4.15: Curvas de BER en función de la SNR para el DFFE ( $R = 30$  y  $60$ ), VA ( $N = 4$ ), y DFFE-VA ( $R = 30$ ,  $R_v = 3$ ,  $N = 4$ ).

y 60 iteraciones, VA de 4 estados, y la propuesta del DFFE-VA con  $N = 4$ ,  $R = 30$ , y  $R_v = 3$ . Notar que para los casos aislados del VA de  $N = 4$ -estados y el DFFE con  $R = 30$  iteraciones, el desempeño es inferior al DFE. Sin embargo, la combinación propuesta de estado reducido del VA con un bajo número de iteraciones DFFE obtiene un rendimiento similar al DFE. La configuración que logra un buen equilibrio entre rendimiento y complejidad es DFFE-VA con  $R = 30$ ,  $R_v = 3$  y  $N = 4$ . Esto demuestra la flexibilidad que ofrece la arquitectura DFFE, compatible con diferentes técnicas de detección. Además, y como se mostrará en el *Capítulo 5*, esta combinación de algoritmos permite reducir la complejidad de implementación del receptor.

## 4.7. Desempeño del DFFE-VA en un Canal de Fibra Multimodo

En la *Sección 4.5* se describió el entorno de trabajo en donde se puede aplicar el receptor propuesto. El escenario consiste en un centro de datos el cual utiliza distintos tipos de medios físicos para conectar los diferentes componentes que lo integran. Principalmente podemos encontrar conexiones de cable de cobre y de fibra óptica. El cable de cobre se utiliza en distancias cortas y es empleado para las conexiones inter-rack. En contra parte, la fibra óptica se la encuentra en la comunicación entre los rack, en donde las distancias a recorrer van desde algunas decenas de metros hasta 2km. Las distancias a conectar dependen de la distribución del centro de datos.

Bajo este entorno, se analizará el caso para un canal de fibra óptica, en particular fibra multimodo (*Multi Mode Fiber - MMF*) muestreado a  $10GHz$ . El canal no presenta un número grande de coeficientes pero se requiere un número de iteraciones mucho mayor al número de coeficientes ( $R \gg L + 1$ ). Para optimizar los recursos de implementación, se propone utilizar la configuración del DFFE que incluye el algoritmo de Viterbi. Recordando el diseño de la arquitectura propuesta en la *Sección 4.6* tenemos que

$$\hat{a}_n^{(i)} = \begin{cases} \mathcal{VA}(y_n) & , \quad 0 \leq i < R_v \\ \mathcal{Q} \left( y_n - \sum_{k=1}^i \hat{a}_{n-k}^{(i-k)} d_k \right) & , \quad R_v < i < L \\ \mathcal{Q} \left( y_n - \sum_{k=1}^L \hat{a}_{n-k}^{(R-1-k)} d_k \right) & , \quad i \geq L \end{cases} \quad (4.92)$$

donde las primeras  $R_v$  iteraciones utilizan decisiones tentativas provenientes del VA. Bajo estos conceptos, se plantean los siguientes análisis de desempeño del DFFE-VA.

El primer análisis se desarrolla sin considerar el VA y buscando el número de iteración que genere un desempeño del DFFE similar al DFE. La evaluación de esta métrica se realiza sobre el canal de fibra multimodo con la respuesta al impulso que se detalla en la Fig. 4.16. A diferencia del ejemplo anterior, la respuesta al impulso del canal equivalente no posee un gran número de post-cursos, tal como se observa en la Fig. 4.17. Aplicando el concepto de información mutua en cada iteración, determinamos el número de iteración del DFFE. La Fig. 4.18(a) muestra la información mutua en función de la  $SNR$ , en donde se observa que para  $R = 60$  se alcanza un desempeño similar al DFE. Similar resultado

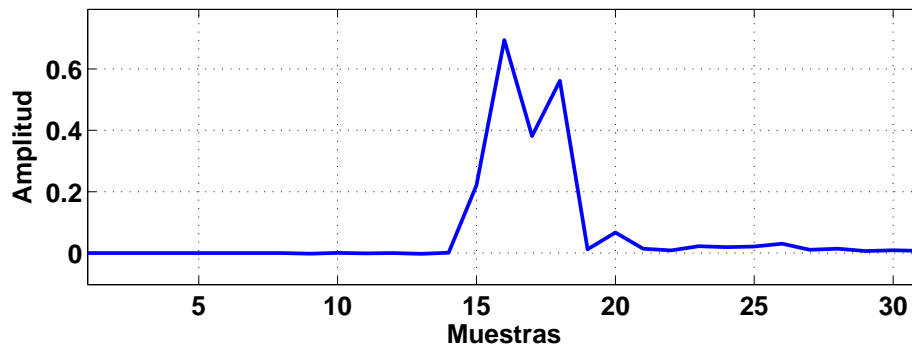


Figura 4.16: Respuesta la impulso de un canal de fibra multimodo muestreado a 10GHz

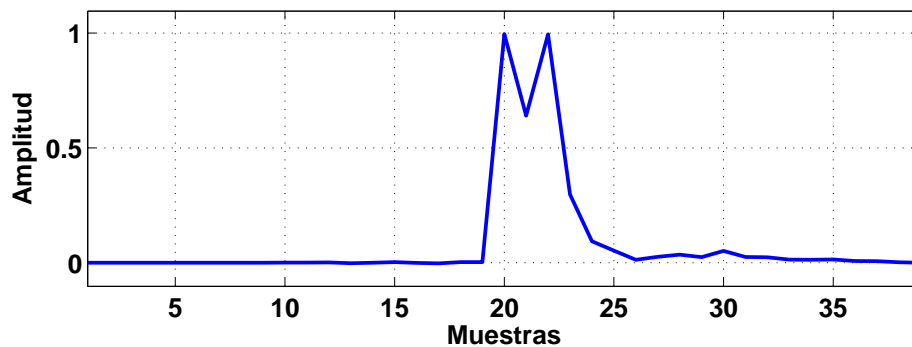
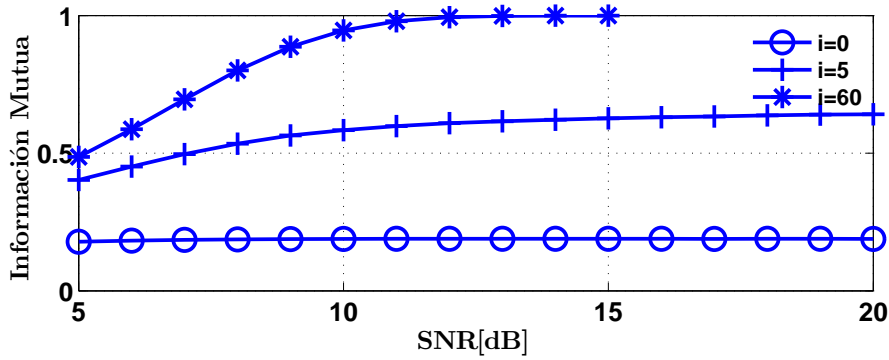


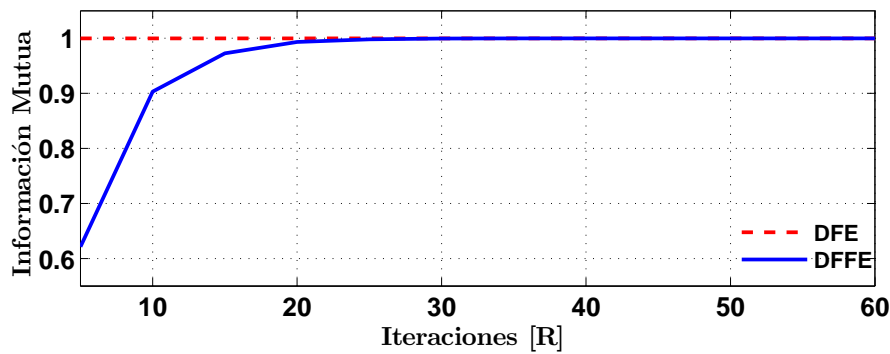
Figura 4.17: Respuesta la impulso del canal equivalente después de ser compensado parcialmente por el ecualizador lineal

se obtiene en la Fig. 4.18(b) al estimar la información mutua en función del número de iteraciones para una  $SNR = 14dB$ . Los resultados conducen a que el número de iteraciones es mayor que el número de coeficientes del canal ( $R \gg L + 1$ ). La mejora del desempeño del DFFE en función de las iteraciones, también se hace visible analizando las curvas de BER. En la Fig. 4.19(a) se detallan cinco diferentes configuraciones del DFFE para  $R = [20, 30, 40, 50, 60]$  iteraciones y se comparan con el DFE. Al igual que en el análisis de información mutua, la configuración con  $R = 60$  tiene el mejor desempeño, este último se resume en la Fig. 4.19(b).

En base a los resultados obtenidos y buscando reducir la complejidad final del ecualizador, se analiza la posibilidad de incorporar el algoritmo de Viterbi en las primeras  $R_v$  decisiones tentativas. Para determinar este valor, en la Fig. 4.20 se trazan las curvas de desempeño en función de las iteraciones que usan decisiones tentativas provenientes del VA para una  $SNR = 14dB$ . Además, se considera en el análisis que el VA utiliza cuatro estados ( $N = 4$ ). En la Fig. 4.20, los primeros tres casos que emplean  $VA(N4)$  y



(a)



(b)

Figura 4.18: Confiabilidad de las decisiones tentativas del DFFE para un canal de fibra multi-modo. (a) Información mutua en función de la SNR para tres (3) iteraciones diferentes. (b) Información mutua en función del número de iteración para una SNR = 14dB.

$R = [10, 20, 30]$  muestran un incremento significativo en el rendimiento a medida que se incrementa el valor de  $R_v$ . También se observa que para un número mayor de iteraciones ( $R = [40, 50, 60]$ ) el aporte del VA es despreciable. De lo cual se concluye, que para una configuración del DFFE-VA de  $N = 4$ ,  $R_v = 4$  y  $R = 20$ , se alcanza un desempeño similar al DFE con un reducción significativa en el número de iteraciones. Esto implica una disminución en la complejidad de implementación, donde este análisis se desarrolla en el siguiente capítulo.

Por último, y en base al análisis realizado, la Fig. 4.21 muestra las curvas de BER del DFE, el VA de cuatro estado ( $N = 4$ ), el DFFE con  $R = [20, 60]$  y DFFE-VA con  $N = 4$ ,  $R_v = 4$  y  $R = [20, 60]$ . Tal como se puede apreciar, el DFFE con  $R = 20$  y el VA de cuatro estados tienen un desempeño sub-óptimo para una  $SNR > 12dB$ . Combinando ambos algoritmos (DFFE-VA con  $N = 4$ ,  $R_v = 4$  y  $R = 20$ ) se obtiene el mismo desempeño

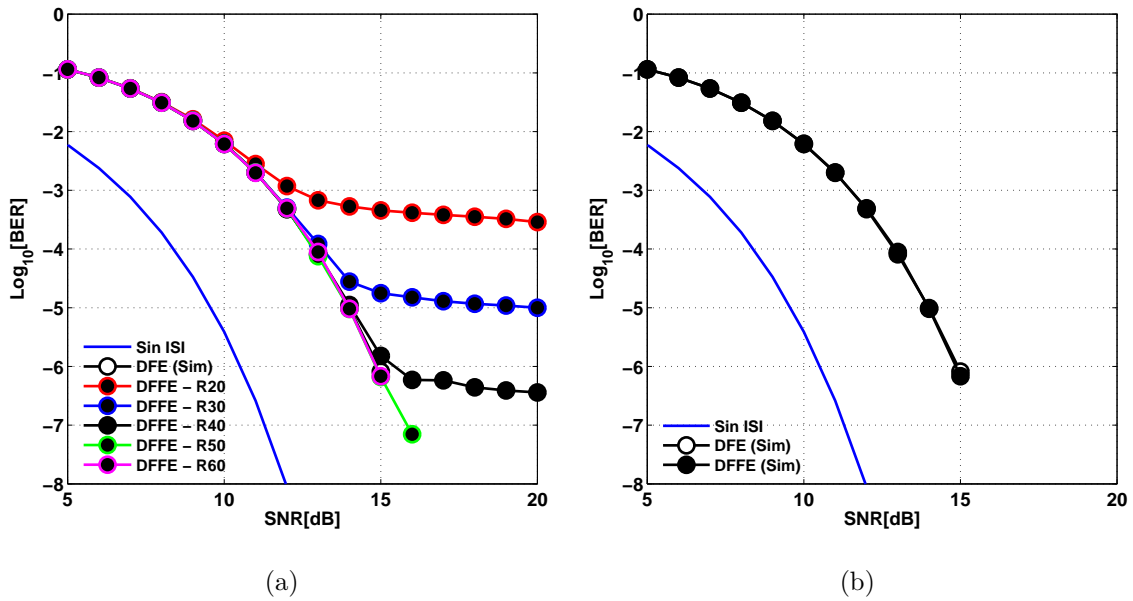


Figura 4.19: Curvas de desempeño (BER) del DFFE comparado con el DFE considerando un canal MMF detallado en la Fig. 4.16. (a) Comparativa de curvas de desempeño considerando diferente número de iteraciones ( $R = [20, 30, 40, 50, 60]$ ). (b) Curvas de desempeño del DFE y DFFE considerando  $R = 60$  iteraciones.

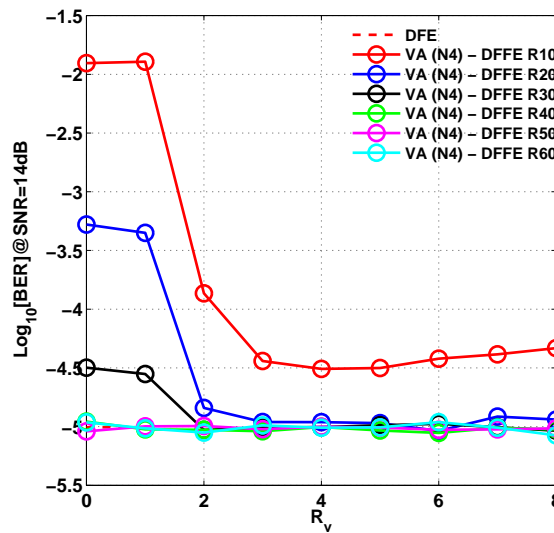


Figura 4.20: Desempeño del DFFE en función del número de iteraciones que utilizan decisiones tentativas provenientes del algoritmo de Viterbi ( $R_v$ )

que el DFE pero reduciendo significativamente la cantidad de iteraciones. Esto implica, al mismo tiempo, una reducción en la complejidad.

En conclusión, la flexibilidad que presenta el DFFE para operar en conjunto con otras técnicas de ecualización, lo convierten en un algoritmo apto para cualquier tipo

## 4.7. Desempeño del DFFE-VA en un Canal de Fibra Multimodo

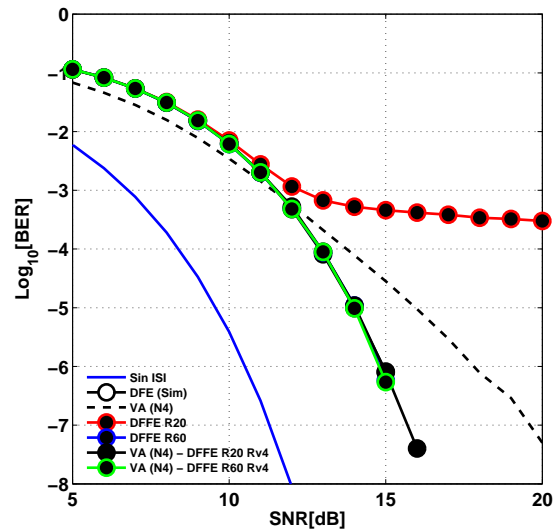


Figura 4.21: Curvas de desempeño del DFE, VA con  $N = 4$ , DFFE con  $R = [20, 60]$  y DFFE-VA con  $N = 4$ ,  $R = [20, 60]$  y  $R_V = 4$ .

de aplicación. Principalmente, en receptores de muy alta velocidad, ya que presenta un diseño de implementación directa (*forward*).

### 4.8. Conclusión Parcial

En este capítulo se desarrolló el análisis teórico del ecualizador directo asistido por decisiones (*Decision FeedForward Equalizer - DFFE*). El DFFE utiliza decisiones tentativas para compensar la distorsión del canal. Se utilizó el concepto de información mutua para mostrar que el incremento de las iteraciones favorece el desempeño. La ventaja que presenta esta técnica, es que no posee lazo realimentado, lo cual la convierte en una excelente alternativa para aplicaciones de alta velocidad. Además, y como se verá en el *Capítulo 5*, la complejidad del DFFE crece cuadráticamente con la memoria del canal a diferencia de otras alternativas ecualizadores.

Para facilitar la configuración de los parámetros del DFFE, se propuso una metodología de estimación de la probabilidad de error de bits que permite al diseñador estimar las iteraciones óptimas del ecualizador. Así como también, se empleó resultados de simulación para corroborar las estimaciones de desempeño del DFFE, dando como resultado un comportamiento similar al DFE. El escenario utilizado para tal fin fue un canal de cobre, típico en los centros de datos.

En conjunto conjunto con este análisis, y utilizando la misma metodología planteada en el párrafo anterior, se propuso una arquitectura alternativa que combina los beneficios del DFFE con la eficiencia de detección del detector de secuencia de máxima verosimilitud. El concepto radica en implementar esta técnica de detección empleando el algoritmo de Viterbi pero con estados reducidos. Es decir, compensar parcialmente la ISI del canal para reducir la probabilidad de error en la detección de bit en las primeras decisiones tentativas del DFFE. El efecto que esto logra, es reducir el número de iteraciones del DFFE. El análisis de la reducción de complejidad del receptor utilizando esta propuesta se detalla en el *Capítulo 5*.

En el siguiente capítulo se analiza la complejidad del DFFE y DFFE-VA frente a otras arquitecturas paralela del DFE y se proponen esquemas paralelos de ambos sistemas.



# CAPÍTULO

## 5

# COMPLEJIDAD E IMPLEMENTACIÓN PARALELA DEL DFFE

***Resumen:** En el capítulo se presenta el estudio de la complejidad de implementación de la nueva arquitectura ecualizadora, así como también propone un diseño paralelo posible de ser implementados en receptores de alta velocidad. La complejidad del ecualizador se compara con otras arquitecturas paralelas del DFE y las estimaciones de complejidad se verifican por medio de resultados de síntesis utilizando una herramienta de diseño digital. El resultado a destacar, es que considerando la memoria del canal, la complejidad del DFFE crece cuadráticamente a diferencia de las otras propuestas donde el incremento es exponencial.*

### 5.1. Introducción

En este capítulo se estudiará la complejidad para las dos arquitecturas propuestas en el capítulo anterior para aplicaciones de alta velocidad, donde la paralelización juega un papel fundamental en el desarrollo. Además, se compara la complejidad de implementación de las arquitecturas paralelas del DFE con respecto al DFFE, dejando en evidencia la reducción de los recursos utilizados para aquellos canales que tienen gran memoria. Con el objetivo de corroborar las estimaciones teóricas de complejidad, se generan resultados de síntesis para un ASIC. Esto se realiza para diferentes configuraciones del DFFE solamente considerando los coeficientes del filtro realimentado.

### 5.2. Análisis de la Arquitectura DFFE

En esta sección se estudia la complejidad del DFFE en una implementación en paralelo, donde se muestra como cambia el número de recursos en relación con la memoria del canal. La baja complejidad que caracteriza al DFFE se compara con otras arquitecturas paralelas del DFE detalladas en el *Capítulo 3*, con el objetivo de verificar este supuesto. Por último, se muestran resultados de síntesis para diferentes configuraciones del DFFE para aplicaciones ASIC. Hay que aclarar que solamente se considera, en esta primera etapa, la complejidad del filtro realimentado. Esto es válido, ya que existen diversas aplicaciones que implementan filtros directos en forma paralela de manera eficiente [60] y se pueden combinar con cualquiera de las implementaciones del DFFE paralelo.

#### 5.2.1. Procesamiento en Paralelo

En la *Sección 3.2* se desarrollaron las diferentes estrategias que permiten incrementar la velocidad de procesamiento en un sistema digital de comunicaciones. De las estudiadas, se destacan la segmentación (*pipeline*) y el paralelismo. Ambos métodos se aplican fácilmente a sistemas directos pero presentan una alta complejidad al momento de implementarlas en sistemas que contienen lazos realimentados. Un ejemplo de este tipo de sistema es el ecualizador realimentado por decisiones. Existen diferentes propuestas que combaten esta dificultad a costa de un incremento exponencial de la complejidad. Una solución a esta problemática es el DFFE, que soluciona el cuello de botella creado por el lazo realimentado

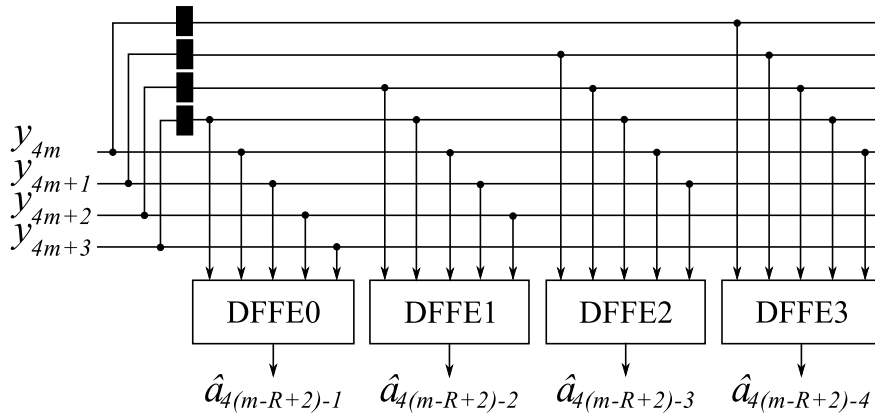


Figura 5.1: Arquitectura paralela del DFFE para  $P = 4$ ,  $L = 3$  y  $R = 5$ . Los bloques DFFEn son detallados en la Fig. 5.2.

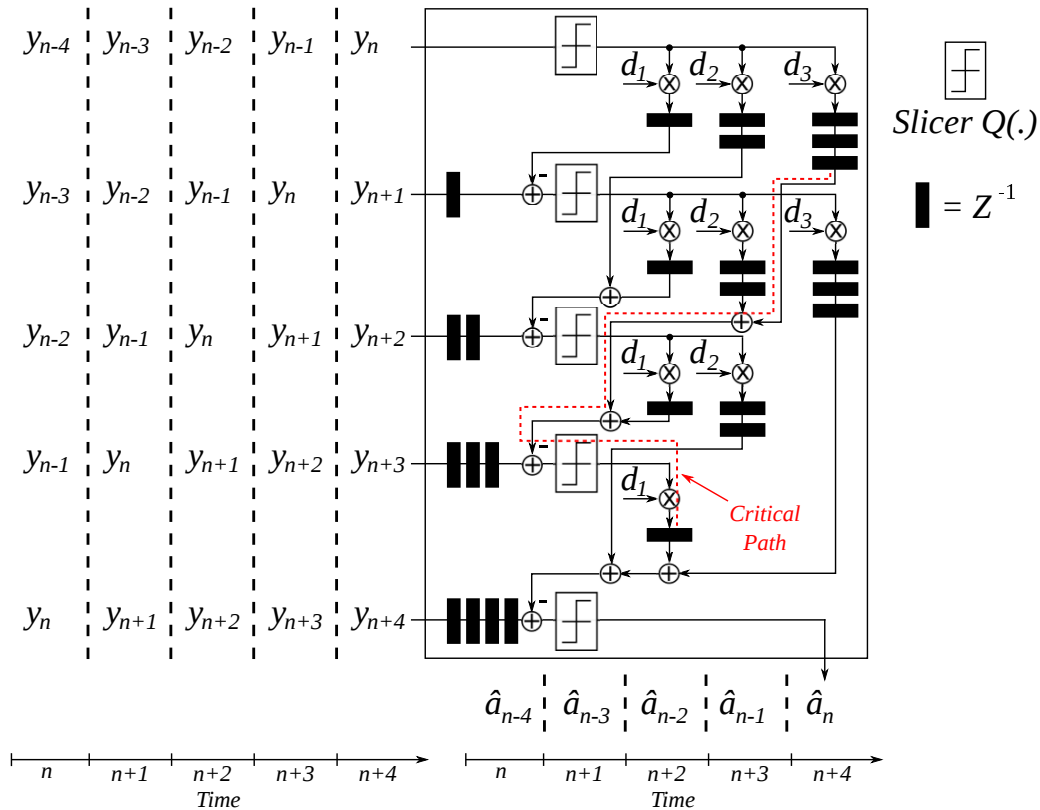


Figura 5.2: Ejemplo de un DFFE con tres (3) coeficientes ( $L = 3$ ) y  $R = 5$  iteraciones. a) Notar que la latencia entre la señal de entrada y la decisión es  $R - 1$ . b) La línea de trazo de color roja denota el camino crítico

del DFE, utilizando decisiones tentativas de una manera directa (*forward*). La aplicación de las técnicas de segmentación y paralelismo, se aplican directamente sobre el diseño del DFFE sin dificultad.

Tabla 5.1: Complejidad de la arquitectura del DFFE paralelo para 2-PAM y  $R > L$ .

Componentes	DFFE
Sumadores	$L(R - L/2 - 1/2)P$
Registros	$((R - 1)R/2 + (R - L)(L + 1)L/2 + (L^2 - 1)L/6)P$
Multiplexores 2-a-1	$L(R - L/2 - 1/2)P$

Las arquitecturas paralelas del DFE que buscan minimizar las limitaciones creadas por el lazo realimentado [22, 27, 31, 28], tienen la particularidad de compartir decisiones entre los diferentes bloques que componen el receptor. Esto genera un incremento en la complejidad por parte del ruteo de las señales, dificultando la implementación para sistemas de alta velocidad. Esta característica es solucionada por el DFFE, ya que independiza cada uno de los bloques y permite al diseñador incrementar en forma simple el paralelismo del receptor. Un ejemplo de implementación en paralelo se muestra en la Fig. 5.1, donde se observa un factor de paralelismo  $P = 4$ , el número de iteraciones  $R = 5$  y el número de coeficientes  $L = 3$ . El detalle de cada uno de los bloques DFFEn se especifican en la Fig. 5.2. Usando esta arquitectura, la velocidad de datos y el rendimiento puede incrementarse por un factor  $P$  con el crecimiento de la complejidad lineal en  $P$ .

### 5.2.2. Complejidad del DFFE

En base a la arquitectura del DFFE (ver Fig. 5.2), se estima la complejidad considerando solamente los componentes como unidad sin tener en cuenta el ruteo. En la Tabla 5.1 se detalla el número de sumadores, registros, y multiplexores para el DFFE, calculados bajo los siguientes supuestos. Los multiplicadores fueron considerados multiplexores 2-a-1. Se supone que tanto los valores positivos y negativos de los coeficientes  $d_k$  están disponibles. Es una suposición correcta para las decisiones binarias con valores  $\pm 1$  (por ejemplo, 2-PAM [6]). El número de sumadores para el DFFE fue estimado suponiendo que el componente básico es un sumador de dos entradas.

La comparación de complejidad entre las diferentes arquitecturas paralelas del DFE [22, 27, 31, 28] y el DFFE, se realiza considerando un canal con una memoria mucho mas grande que uno ( $L \gg 1$ ). La condición planteada, reduce las expresiones de complejidad

## 5.2. Análisis de la Arquitectura DFFE

Tabla 5.2: Comparación de complejidad entre el DFFE paralelo y arquitecturas DFE para 2-PAM con  $R = L + 1$  para  $L \gg 1$

Comp.	DFFE	DFE [22]	DFE [27]	DFE [31]	DFE [28]
Sumadores	$L^2P/2$	$2^L P$	$2^{L/2} 2P$	$2^L P$	$2^{L+1} P$
Registros	$L^3P/6$	$\sim 2^L P$	$\sim 2^{L/2}(P + 1)$	$L^2 + 2^L P$	$(2^L + L)P$
Mux 2-a-1	$L^2P/2$	$(2^L - 1)P$	$(2^{L/2} - 1)2P$	$2^L L(P - \frac{L}{2} + \frac{P}{L} - 1)$	$2^L P$

Tabla 5.3: Comparación de complejidad entre el DFFE paralelo y arquitecturas DFE para  $M$ -PAM con  $R = L + 1$  para  $L \gg 1$

Componente	DFFE	DFE [27]	DFE [31]
Sumadores	$L^2P/2$	$M^{L/2} 2P$	$M^L P$
Registros	$L^3P/6$	$\sim M^{L/2}(P + 1)$	$L^2 + M^L P$
Multiplexores 2-to-1	$(M - 1)L^2P/2$	$(M^{L/2} - 1)2P$	$M^L L(P - \frac{L}{2} + \frac{P}{L} - 1)$

a lo detallado en la Tabla 5.2. El número de sumadores y multiplexores 2-a-1 para los esquemas paralelos del DFE fueron extraídos de [22, 27] y [31], mientras que se estimó el número de registros en base a las arquitecturas.

La Fig. 5.3 muestra el número de los tres tipos de componentes en función del número de coeficientes del filtro realimentado. La diferencia más importante entre el DFFE y las propuestas del DFE paralelo es que, el primero no utiliza técnicas de mirar adelante (look-ahead) o lazo de multiplexor (multiplexer loop), y esto reduce la complejidad de la implementación. En todos los casos, los beneficios del DFFE son evidentes en la presencia de canales altamente dispersivos (es decir,  $L \gg 1$ ). Una comparación de la complejidad de  $M$ -PAM se muestra en la Tabla 5.3. Observamos que el DFFE todavía proporciona una reducción significativa de la complejidad con respecto a las arquitecturas DFE [27] y [31]. Notar que en  $M$ -PAM, las operaciones de multiplicación se logran mediante el uso de  $M - 1$  multiplexores 2-a-1. Esta conclusión se puede extender a  $M$ -QAM donde la complejidad de ambos, DFE y DFFE, es aproximadamente dos veces la obtenida con  $M$ -PAM.

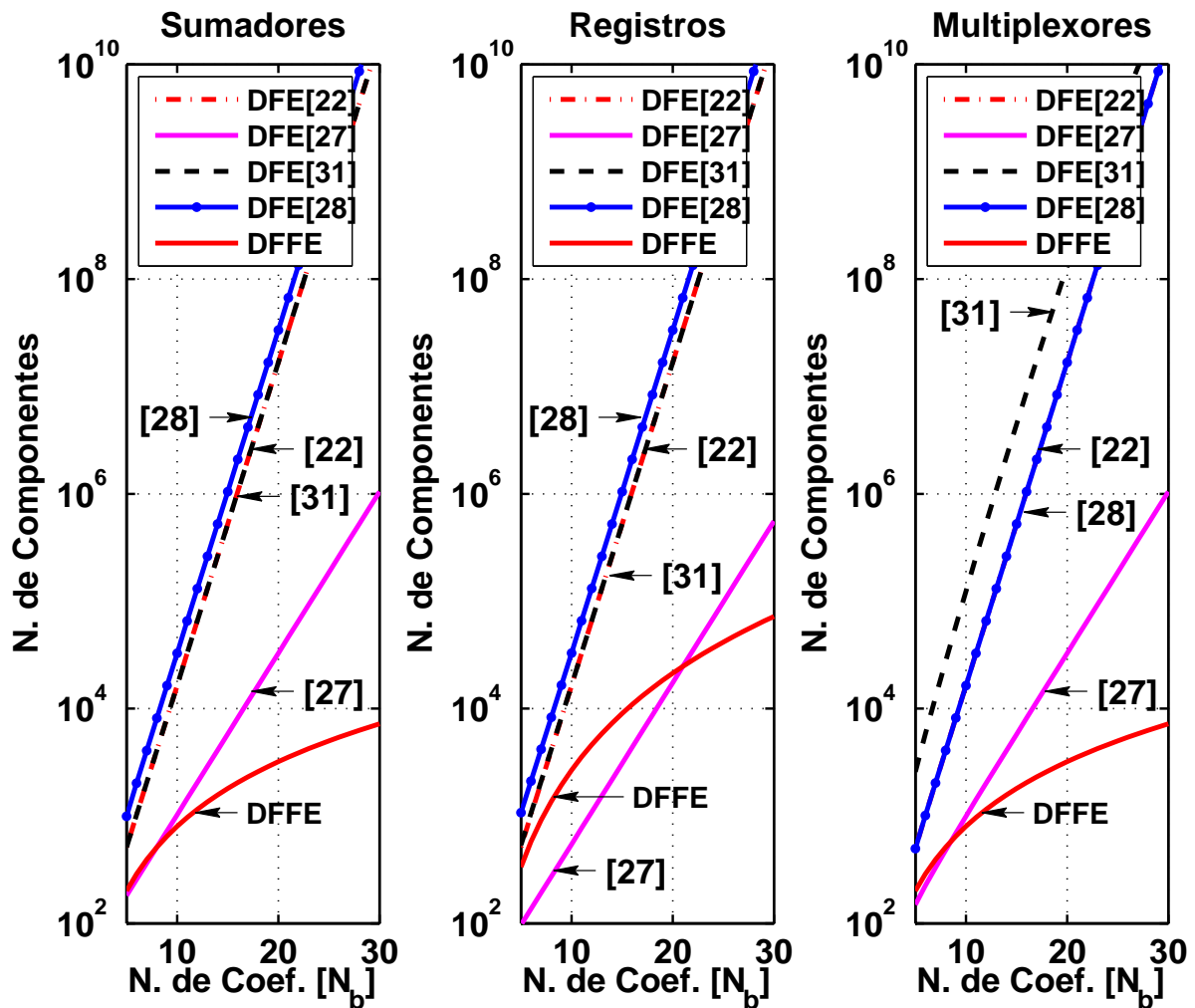


Figura 5.3: Número de sumadores, registros y multiplexores 2-a-1 en función del número de coeficientes del filtro realimentado  $L$ , para el DFFE paralelo con  $R = L + 1$  y las arquitecturas DFE propuestas en [22, 27, 31] y [28]. Factor de paralelización:  $P = 16$ . Formato de modulación: 2-PAM

### 5.2.3. Comparación con Resultados de Síntesis

La estimación de complejidad realizada en la *Sección 5.2.2* es una aproximación general de los componentes que utiliza la arquitectura propuesta. En un entorno más real, haciendo referencia a la implementación, es necesario considerar otros aspectos adicionales en la estimación de complejidad. En esta sección se realiza una comparación entre las estimaciones teóricas y resultados de síntesis de varias configuraciones del DFFE. Las consideraciones que se tuvieron en cuenta fueron las siguientes

- Se consideró un ASIC para la implementación del DFFE en un receptor de  $10Gb/s$

y modulación 2-PAM. La arquitectura DFFE fue sintetizada con éxito (es decir, no hay problemas de tiempo) mediante el uso de la tecnología CMOS de 28nm con transistores de umbral de voltaje estándar (*Standard Voltage Threshold - SVT*).

- El DFFE se constituyó para  $L = [5, 10, 30]$  coeficientes, con dos factores de paralelización  $P = 16$  ( $f_{clock} = 625,0$  MHz) y  $P = 32$  ( $f_{clock} = 312,5$  MHz). En todos los casos, el número de iteraciones contemplado en la estimación guarda la relación  $R = L + 1$ .
- Las operaciones de multiplicación fueron implementadas mediante el uso de multiplexores 2-a-1.
- Se definió un número finito de bits para representar las muestras de entrada y los coeficientes. El número de bits de las muestras de entrada ( $N_i$ ) y los coeficientes ( $N_c$ ) se han derivado de las simulaciones por computadora para diferentes canales con memoria  $L = [5, 10, 30]$ . Todas las operaciones internas en el DFFE se consideraron a máxima resolución, para evitar degradación en el desempeño. Se utilizaron dos configuraciones de resoluciones de las palabras de entrada según el canal. Para los casos  $L = 5$  y  $10$  se empleó  $N_c = 7$  y  $N_i = 7$  y con  $L = 30$  el número de bits de las muestras de entrada se incrementó a  $N_i = 8$ . La Fig. 5.4 compara el desempeño del DFE con las dos configuraciones de la muestra de entrada al DFFE, donde se observa que los comportamientos son muy similares. Los sumadores se implementaron con propagación de acarreo (*carry save adder*), por lo tanto  $N_c + \log_2(L)$  bits se requieren para representar la muestra en la entrada del detector de umbral.
- El detector de umbral utiliza el bit más significativo (*Most Significant Bit - MSB*) de la muestra de entrada para controlar los multiplexores que seleccionan los coeficientes positivo o negativo.

En la Tabla 5.4 se muestra el número total de celdas y componentes estimados normalizados para un paralelismo de  $P = 16$  y memoria del canal de  $L = 5$ . El número de celdas se deriva de la síntesis realizada con una herramienta de diseño digital. Este número incluye elementos extras como buffers que la herramienta agrega al diseño con el fin de compensar el retardo de cada una de las pistas a consecuencia del ruteo. Notar que estos resultados concuerdan muy bien con los valores esperados derivados del análisis de

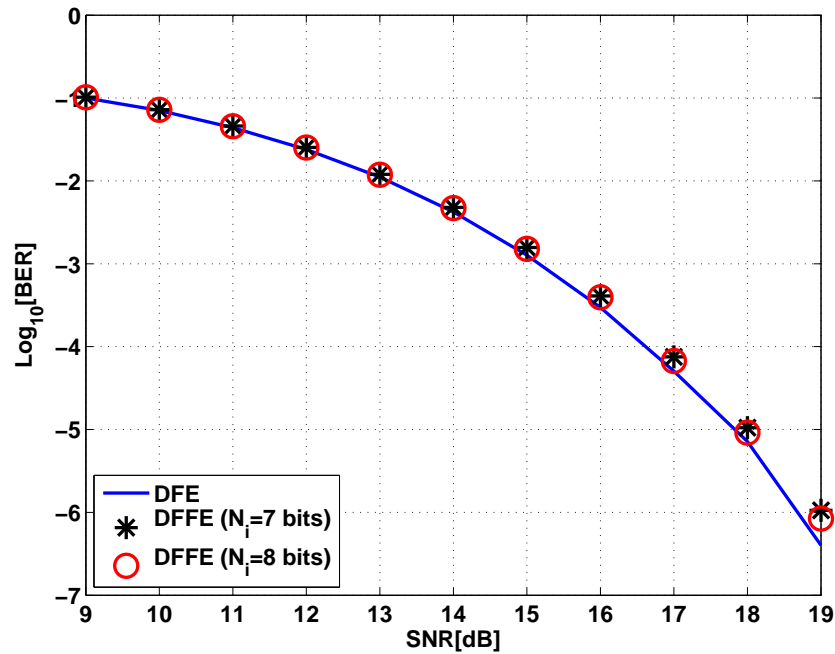


Figura 5.4: Curva BER en función de la SNR para le DFFE con  $L = 30$ ,  $R = 31$ ,  $N_i = 7/8$  bits, y  $N_c = 7$  bits.

Tabla 5.4: Resultados de síntesis para la arquitectura del DFFE paralelo para 2-PAM y  $R = L + 1$  en Tecnología CMOS 28nm

$f_{clock}$ (MHz)	$P$	$L$	N. de Celdas <sup>†</sup>	N. Componentes <sup>†</sup>
625,0	16	5	1,00	1,00
625,0	16	10	5,19	4,18
312,5	32	5	1,96	2,00
312,5	32	10	10,03	9,62
312,5	32	30	180,65	159,00

<sup>†</sup> El número total de celdas y componentes están normalizados para los valores  $P = 16$  y  $L = 5$ .

complejidad desarrollado en la Sección 5.2.2, es decir, la complejidad aumenta linealmente con el factor de paralelización ( $P$ ) y cuadráticamente con la memoria de canal  $L$ .



### 5.2.4. Análisis de Camino Crítico

La velocidad de las diferentes arquitecturas DFE se relacionan con sus caminos críticos. Las arquitecturas paralelas del DFE existentes de [22, 31, 28] son más rápidos que el DFFE. Sin embargo, no se consideran para una comparación de velocidad como resultado de su prohibitiva alta complejidad de implementación en presencia de canales con alta ISI ( $L \gg 1$ ). El análisis de camino crítico y complejidad de las arquitecturas mencionadas se detalló en el *Capítulo 3*. Por otro lado, el camino crítico de la solución DFE menos complejo propuesto en [27] está dada por

$$T_{DFE-[27]} \approx \frac{1}{L/2 + 1} T_{add} + \log_2(M) T_{mux} \quad (5.1)$$

para  $M$ -PAM, donde  $T_{mux}$  y  $T_{add}$  son los tiempos de propagación del multiplexor y sumador, respectivamente. Tenga en cuenta que  $T_{DFE-[27]}$  es independiente de la memoria del canal  $L$ . Por ejemplo, para la tecnología CMOS de 28nm,  $T_{mux} \approx 0,05\text{ns}$  y  $T_{add} \approx 0,10\text{ns}$ , de tal manera las tasas máximas de datos con  $P = 1$  para 2-PAM y 4-PAM son  $\sim 17,8$  y  $18,8\text{Gb/s}$ , respectivamente.

El camino crítico para el DFFE se muestra en la Fig. 5.2. Observe que el retraso del camino crítico esta dado por

$$T_{DFFE} \approx LT_{add} + \log_2(M) T_{mux} \quad (5.2)$$

y aumenta linealmente con la memoria del canal. Como se muestra en la *Sección 5.2.3*, no se observaron problemas de tiempo con  $L = 30$  y  $P = 32$  para 2-PAM con  $f_{clock} = 312,5\text{MHz}$  mediante el uso de tecnología CMOS 28nm. Por lo tanto, las tasas máximas de datos obtenidos por el DFFE para 2-PAM y 4-PAM son 10 y  $\sim 20\text{Gb/s}$ , respectivamente. Desde  $L \gg 1$  y  $T_{mux} < T_{add}$ , tenga en cuenta que  $T_{DFFE}$  está dominado por  $LT_{add}$ . Por tal motivo, el impacto del aumento del tamaño de la constelación ( $2 \rightarrow 4$ ) en el camino crítico será pequeño. Por otro lado, para  $L = 30$  la complejidad relativa del DFE [27] con  $P = 1$  ( $\propto 2M^{L/2}$ ) con respecto al DFFE con  $P = 32$  ( $\propto 32(M - 1)L^2$ ) es

- $2 \times 2^{(30/2)} / (32 \times 30^2) = 2,28$  para 2-PAM.
- $2 \times 4^{(30/2)} / (32 \times 3 \times 30^2) = 2,49 \times 10^4$  para 4-PAM.

En consecuencia, el DFFE es capaz de ofrecer altas velocidades de datos (por ejemplo,  $> 10Gb/s$ ) mediante el uso de la tecnología CMOS existentes con la complejidad de implementación menor que el derivado de la menor complejidad del DFE paralelo propuesto en [27].

### 5.3. Reducción de Complejidad en el DFFE

En la *Sección 5.2* se desarrolló el análisis de una nueva arquitectura ecualizadora iterativa denominada ecualizador directo asistido por decisiones. Una de las ventajas que presenta esta arquitectura es el crecimiento cuadrático de la complejidad a diferencia de otras arquitecturas paralelas del DFE que crecen exponencialmente con la longitud de la memoria del canal. Otra de las ventajas que destaca al DFFE por encima de otras arquitecturas es que posee una implementación modular y no necesita compartir muestras entre estos módulos. Esto facilita la implementación en paralelo y disminuye el ruteo, lo cual lo convierte en una excelente opción para aplicaciones de muy alta velocidad.

Esta sección tiene el objetivo de aplicar las ventajas del MLS (implementado con el algoritmo de Viterbi) como técnica de reducción de complejidad del DFFE. Se mostrarán los conceptos de aplicación de ambas técnicas y una posible implementación en paralelo. Por último, se compara la complejidad de diferentes técnicas de ecualización con esta nueva propuesta, donde el resultado que se observará es una reducción significativa en la complejidad de implementación.

#### 5.3.1. Complejidad del DFFE-VA

La principal característica que resalta al DFFE sobre las otras arquitecturas del DFE es la implementación totalmente directa (*forward*). Esta cualidad, facilita la implementación de técnicas de segmentación (*pipeline*) y paralelización con el fin de incrementar la tasa de datos en receptores de alta velocidad. En esta sección nos hemos enfocado en la implementación conjunta del DFFE con el algoritmo de Viterbi con el fin de reducir la complejidad de implementación del receptor. De las arquitecturas paralelas de VA, una de las más destacadas por su baja complejidad es el decodificador de Viterbi de bloque deslizante (*Sliding Block Viterbi Decoder - SBVD*) [48] en donde una implementación de

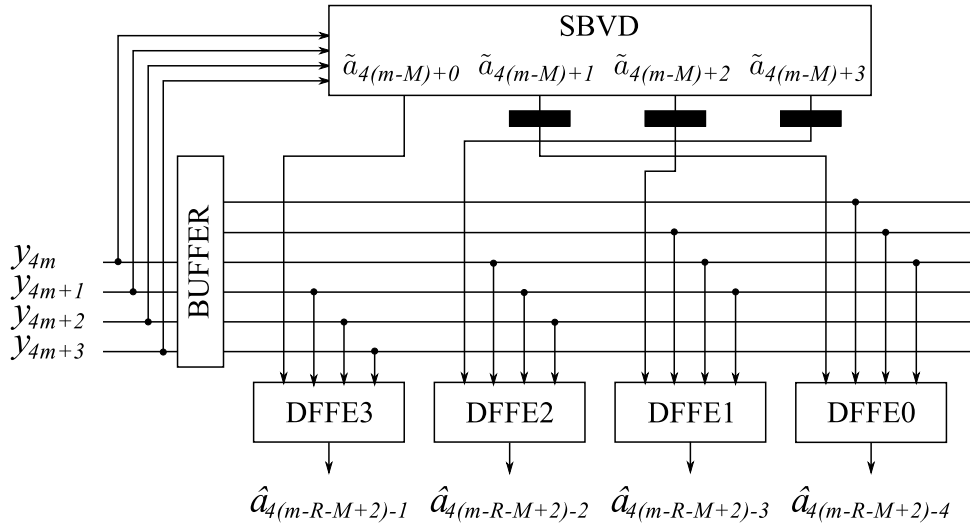


Figura 5.5: Arquitectura paralela del DFFE y SBVD para  $N = 2$ ,  $L = 3$  y  $P = 4$ . Los bloques DFFEn y SBVD son mostrados en la Fig. 4.13 y [48], respectivamente.

esta técnica se detalla en [13]. Un ejemplo de implementación paralela del DFFE-VA se muestra en la Fig. 5.5 para una configuración  $L = 3$ ,  $R = 4$ ,  $R_v = 1$ ,  $N = 2$  y  $P = 4$ . Los bloques DFFEn y SBVD se detallan en la Fig. 5.2 y [48], respectivamente.

El análisis de complejidad del DFFE se realizó en la Sección 5.2.2, en donde se mostró que el incremento de complejidad de implementación crece cuadráticamente con la memoria del canal. En esta sección se mostrará la reducción de complejidad del receptor al aplicar el algoritmo de Viterbi como técnica de detección adicional. Recordemos que, la estimación de complejidad se basa en el conteo de componentes básicos como el número de sumadores, registros, multiplexores y comparadores. En las Tablas 5.1 y 5.5 se detallan las expresiones de complejidad de cada uno de estos componentes para la aplicación del DFFE-VA, donde el VA se implementa mediante el uso de la técnica de SBVD paralelo [48]. La estimación de complejidad se realiza bajo los siguientes supuestos

- Complejidad del DFFE:** Los multiplicadores mostrados en la Fig. 5.2 se consideran que son multiplexores 2-a-1. Se supone que tanto los valores positivos y negativos de los coeficientes  $d_k$  están disponibles. Esta es una suposición correcta para decisiones binarias  $\pm 1$  (por ejemplo, binario de modulación de amplitud de pulso (PAM) [6]). El número de sumadores para el DFFE se estima suponiendo que el componente básico es un sumador de dos entradas.

Tabla 5.5: Complejidad de la arquitectura paralela del SBVD para 2-PAM [48]

Componente	SBVD
Sumadores	$N(2B + 1 + D(D + 6))$
Registros	$B(B + 6N) + D^2(N(\frac{2}{D} + \frac{1}{2}) + \frac{\text{Log}_2(N)}{D} + \frac{1}{4})$
Multiplexores 2-a-1	$D(\frac{DN}{2} + 3N + 3) + N - 1$
Comparadores	$D(\frac{DN}{2} + 3N) + N - 1$

Tabla 5.6: Complejidad de arquitecturas paralelas del DFE para 2-PAM

Comp.	DFE [31]	DFE [28]
Sumadores	$2^L P$	$2^{L+1} P$
Registros	$L^2 + 2^L P$	$(2^L + L)P$
Mux 2-a-1	$2^L L(P - \frac{L}{2} + \frac{P}{L} - 1)$	$2^L P$

- Complejidad de SBVD:** La definición de la complejidad para esta arquitectura paralela está regida por el número de estados del VA ( $N$ ), la longitud del bloque de entrada ( $B$ ) y la longitud decodificada ( $D$ ). Los parámetros  $D$  y  $B$  se los considera iguales al paralelismo  $P$ .

Bajo estas condiciones, la Fig. 5.6 muestra el número de componentes básicos de cuatro ecualizadores diferentes, de los cuales en la Tabla 5.6 se detalla la complejidad de las arquitecturas paralelas del DFE propuestas en [31] y [28]. La comparación mostrada en la Fig. 5.6 considera que el número de coeficientes del filtro realimentado es  $L = 10$  y el factor de paralelismo es  $P = 16$ . El número de iteraciones del DFFE es  $R = 60$ , mientras que el esquema DFFE-VA se configura con  $R = 30$ ,  $R_v = 3$  y  $N = 4$ .

La diferencia más importante entre el DFFE-VA y las arquitecturas DFE paralelas es que el primero no utiliza técnicas de mirar adelante (*look-ahead*) o bucle multiplexor (*multiplexer loop*), y reduce la complejidad de la implementación. Además, tanto el DFFE y la técnica SBVD poseen una implementación directa (*forward*), lo cual las convierte en una excelente opción para aplicaciones de alta velocidad.

### 5.3. Reducción de Complejidad en el DFFE

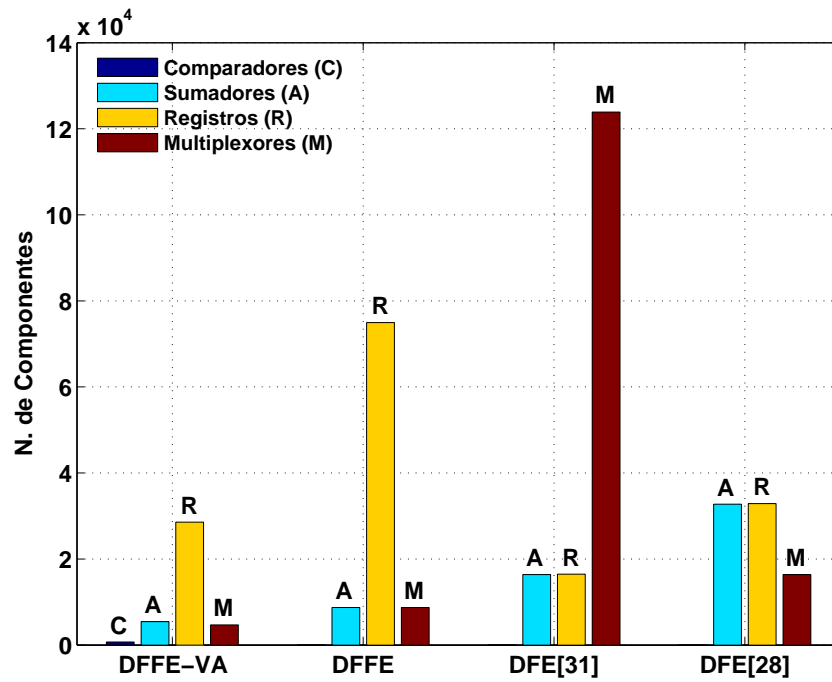


Figura 5.6: Número de componentes básicos para diferentes tipos de ecualizadores, donde el número de coeficientes del filtro realimentado es  $L = 10$  y paralelismo  $P = 16$ . El número de iteraciones del DFFE es  $R = 60$  y DFFE-VA utiliza  $R = 30$ ,  $R_v = 3$  y  $N = 4$  estados del VA.

### 5.4. Conclusión Parcial

En este capítulo se analizó la complejidad de dos arquitecturas propuestas. En primer lugar, se detalló un diseño paralelo del DFFE, en donde se resalta que el incremento de la complejidad del sistema crece linealmente con el paralelismo y cuadráticamente con el canal. Esta cualidad resalta al DFFE frente a otras propuestas, donde el incremento de la complejidad es dependiente de la memoria del canal en un sentido exponencial. Otra ventaja importante es la no interconexión de señales entre los módulos DFFE, esto disminuye el ruteo y permite un diseño simple de la arquitectura del receptor. Las estimaciones de complejidad teóricas se compararon con resultados de síntesis, en donde el resulta de interés resaltar que ambas estimaciones son muy similares.

Por otro lado, se evaluó la complejidad del DFFE en conjunto con el MLSD implementado con el algoritmo de Viterbi. El funcionamiento consiste en utilizar el VA de estados reducidos que mejora la estimación de las primeras decisiones tentativas del DFFE. Esto genera una reducción del número de iteraciones del DFFE manteniendo un desempeño similar al DFE. Por lo tanto, la implementación conjunta del DFFE con el VA lleva a una reducción significativa de la complejidad comparada con otras arquitecturas paralelas del DFFE. En la implementación en paralelo del algoritmo de Viterbi se consideró la técnica decodificador de Viterbi de bloque deslizante, la cual es muy utilizada en sistemas de comunicaciones por sus características de implementación directa (*forward*) y baja complejidad.

Todas estas ventajas no llevan a concluir que el DFFE y DFFE-VA son dos arquitecturas excelentes para aplicaciones de muy alta velocidad. En el siguiente capítulo se detalla la implementación en FPGA del DFFE, en donde se propone una diseño digital del DFFE paralelo en conjunto con un ecualizador lineal y el algoritmo de adaptación LMS.

# CAPÍTULO

## 6

# IMPLEMENTACIÓN EN FPGA DEL DFFE

**Resumen:** En el capítulo se presenta la implementación del DFFE en conjunto con otros algoritmos en una FPGA. Aprovechando las ventajas de este dispositivo, se desarrolló un sistema que consta de un FFE y el DFFE, ambos implementados en paralelo y con el algoritmo de adaptación LMS. El entorno de verificación incluye un procesador integrado (MicroBlaze) y diferentes módulos que permiten corroborar el desempeño del sistema desarrollado. La arquitectura presenta un paralelismo de ocho y una frecuencia de reloj de 50MHz alcanzando una tasa de datos de 400Mbps. Todo el entorno de trabajo se planteó para la placa de evaluación KC705 que tiene montado una FPGA de la familia Xilinx Kintex7 (XC7K325T-2ffg900). Además, se utilizaron las herramientas Vivado 2013.4 y Xilinx SDK 2015.4 para la síntesis del código HDL y compilación del software, respectivamente. En la segunda parte del capítulo, se realiza una evaluación de los recursos utilizados por la arquitectura propuesta analizando los reportes de síntesis. Además, se contrasta el desempeño de los ecualizadores implementados en la FPGA con los desarrollados en el simulador de punto fijo.

### 6.1. Introducción

En los capítulos anteriores se estudió una nueva estructura ecualizadora que tiene un desempeño similar al DFE y se caracteriza por tener una implementación totalmente directa (*forward*). Además, se analizó la complejidad del DFFE desde el aspecto teórico y por medio de síntesis orientadas a implementación en ASIC. La evaluación de este último punto nos llevó a concluir que la complejidad del DFFE crece cuadráticamente con la memoria del canal. Es importante resaltar que la evaluación de la síntesis para ASIC solo considera al DFFE. Es por ello que, para completar la evaluación, resulta de interés desarrollar una aplicación que contenga los elementos básicos de un receptor de comunicaciones.

Con este objetivo, en este capítulo se presenta la implementación del DFFE en conjunto con otros algoritmos en una FPGA. La FPGA brinda la flexibilidad de realizar pruebas de concepto a muy bajo costo y en corto plazo, convirtiéndola en una herramienta esencial para la verificación de sistemas digitales. Aprovechando las ventajas de este dispositivo, se implementa un sistema que consta de un FFE y el DFFE, ambos implementados en paralelo y con el algoritmo de adaptación LMS. El entorno de verificación incluye un procesador integrado (*MicroBlaze*) y diferentes módulos que permiten corroborar el desempeño del sistema desarrollado. El sistema propuesto consta de un paralelismo de ocho (8) y una frecuencia de reloj para la etapa de procesamiento de señal de  $50\text{MHz}$ , alcanzando una tasa de datos de  $400\text{Mbps}$ . Todo el entorno de trabajo se planteó para la placa de evaluación KC705 [61] que tiene montado una FPGA de la familia Xilinx Kintex7 (*XC7K325T-2ffg900*). Además, se utilizaron las herramientas *Vivado 2013.4* y *Xilinx SDK 2015.4* para la síntesis del código HDL y compilación del software, respectivamente. En conjunto con estas dos etapas, se plantea un entorno de usuario en el lenguaje Python. El usuario tiene la flexibilidad de interactuar por medio de un terminal haciendo configuraciones básicas o ingresar líneas de código en el software.

La verificación funcional de los algoritmos se realiza comparando el desempeño para diferentes escenarios entre el simulador de punto fijo y lo emulado en FPGA. En este entorno de trabajo, la siguiente sección resume el flujo de diseño utilizado en la implementación en FPGA.



## 6.2. Flujo de Diseño

El flujo de diseño de la arquitectura implementada sigue los lineamientos especificados en el *Capítulo 1*. Esto consiste en el desarrollo de simuladores en lenguaje de programación C++ utilizando la librería SystemC. El objetivo de esto es modelar de forma simple y dinámica el comportamiento temporal de los registros, así como también evaluar el desempeño del sistema bajo diferentes escenarios. Los simuladores se diseñan para trabajar con resolución de punto flotante y punto fijo [2].

En el primer caso, el modelo de punto flotante, permite determinar en forma precisa si el esquema seleccionado es correcto. El simulador utilizado para esta primera etapa es el mismo que se empleó para la verificación de desempeño en el *Capítulo 4*. El modelo de punto fijo, consiste en un simulador que representa en forma casi idéntica<sup>1</sup> al sistema que se implementa en la FPGA. La arquitectura de punto fijo se caracteriza por representar las variables con un número finito de bits. A diferencia del simulador utilizado en el *Capítulo 4*, en el cual se define una resolución de las muestras de entrada y cada una de las operaciones del DFFE, en este simulador se tienen definidas todas las etapas del sistema. Es decir, se tiene una representación en resolución finita del canal, del FFE, del DFFE y del algoritmo de adaptación de los coeficientes.

El simulador de punto fijo, es empleado para realizar el apareamiento de vectores (*vector matching*) con la finalidad de corroborar el correcto diseño de la arquitectura desarrollada en HDL, específicamente en Verilog. Posteriormente a las verificaciones mencionadas, se utiliza la herramienta de diseño digital Vivado 2013.4 de la empresa Xilinx configurándola para la FPGA Kintex7 [61]. Con esta herramienta se realiza toda la etapa de implementación que consiste en la síntesis del código Verilog, mapeo a compuertas lógicas, colocación (*placement*) de cada compuerta en una posición determinada de la FPGA y el encaminamiento (*routing*) de las señales. Por último, se genera el binario (*bitstream*) que detalla cuales compuertas lógicas y bloques específicos de la FPGA son conectados.

---

<sup>1</sup>**Casi Idéntica:** Existen diferencias con la arquitectura del generador de PRBS, generador de ruido Gaussiano y contador de BER. En particular, se emplean las funciones definidas en C++ para generar la PRBS y el generador de ruido Gaussiano. Por otro lado, se realiza la correlación de vectores para estimar el retardo de los símbolos transmitidos y símbolos detectados para calcular la BER. La calibración de estos módulos implementados en la FPGA se realizó en base a escenarios típicos y comparándolos con resultados de simulación.

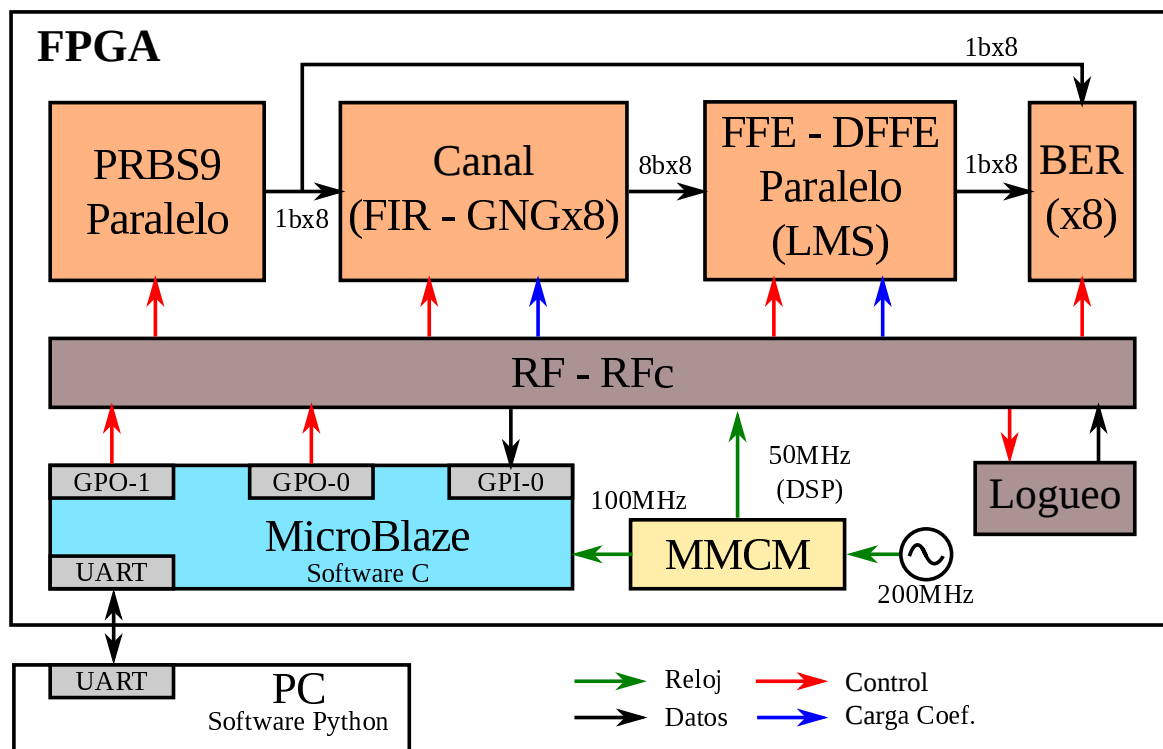


Figura 6.1: Diagrama en bloques de la arquitectura implementada en la FPGA.

La arquitectura implementada, la cual se especifica en las siguientes secciones, incluye un procesador que utiliza un software desarrollado en lenguaje de programación C. La compilación del programa se realiza con la herramienta Xilinx SDK 2015.4 (*Software Development Kit - SDK*). Esta herramienta genera un archivo de inicio que es utilizado por el procesador en la FPGA. Por último, se descarga a la FPGA el archivo compilado.

Antes de mostrar los reportes de implementación obtenidos con estas herramientas, en la sección siguiente se detalla cada uno de los módulos utilizados en el diseño de prueba.

### 6.3. Arquitectura Implementada

En esta sección se realiza un detalle de todos los módulos implementados en la FPGA. La Fig. 6.1 es el diagrama en bloques del sistema completo, en donde se observa la interacción de los bloques principales. Es importante destacar, que el código Verilog se escribió siguiendo las reglas de diseño de la FPGA, lo cual permite utilizar en forma eficiente los recursos del dispositivo. El camino principal o camino de datos (*data path*) consta de los módulos PRBS9, canal, contador de BER y ecualizadores. Los primeros

tres módulos tienen la finalidad de generar el entorno de trabajo (PRBS9 y canal) así como también el de verificar el desempeño (BER). El módulo de ecualizadores consta del FFE, DFFE y el algoritmo de adaptación LMS. Los restantes módulos forman parte de la estructura de configuración, control y logueo, considerando como elemento central un procesador integrado para el control general del sistema. A continuación se describen cada uno de los módulos mencionados.

### 6.3.1. Procesador MicroBlaze

El procesador integrado de núcleo blando MicroBlaze [62] contiene una computadora con un conjunto de instrucciones reducido (*Reduced Instruction Set Computer - RISC*) optimizado para su aplicación en FPGA. En la Fig. 6.2 se muestra un diagrama en bloques funcional del procesador con las interfaces utilizadas para el presente diseño.

La estructura del procesador consiste en tres puertos de entrada/salida de propósito general (*General Purpose Input/Output - GPIO*) utilizados como interfaces de comunicación entre el procesador y el registro de archivos (*Register File - RF*). El propósito de estos puertos es escribir en el RF los parámetros de configuración, control y ejecución del logueo de las señales dentro de los módulos que integran el camino de datos (*data path*). Por otro lado, se utiliza una interfaz de transmisor/receptor asíncrona universal (*Universal Asynchronous Receiver-Transmitter - UART*) con la finalidad de transferir desde la computadora/FPGA o FPGA/computadora los comandos que se enviarán posteriormente por los puertos GPIO. Además, el procesador consta de un controlador de memoria (*local memory*), un controlador de periféricos (*AXI Interconnect*), un controlador de reloj (*clocking*), un módulo de depuración (*MicroBlaze Debug Module - MDM*) y una unidad de reinicio (*reset*) general del procesador (*processor system reset*). La funcionalidad de cada una de estas unidades es similar a la de procesador tradicional y se describen en [62].

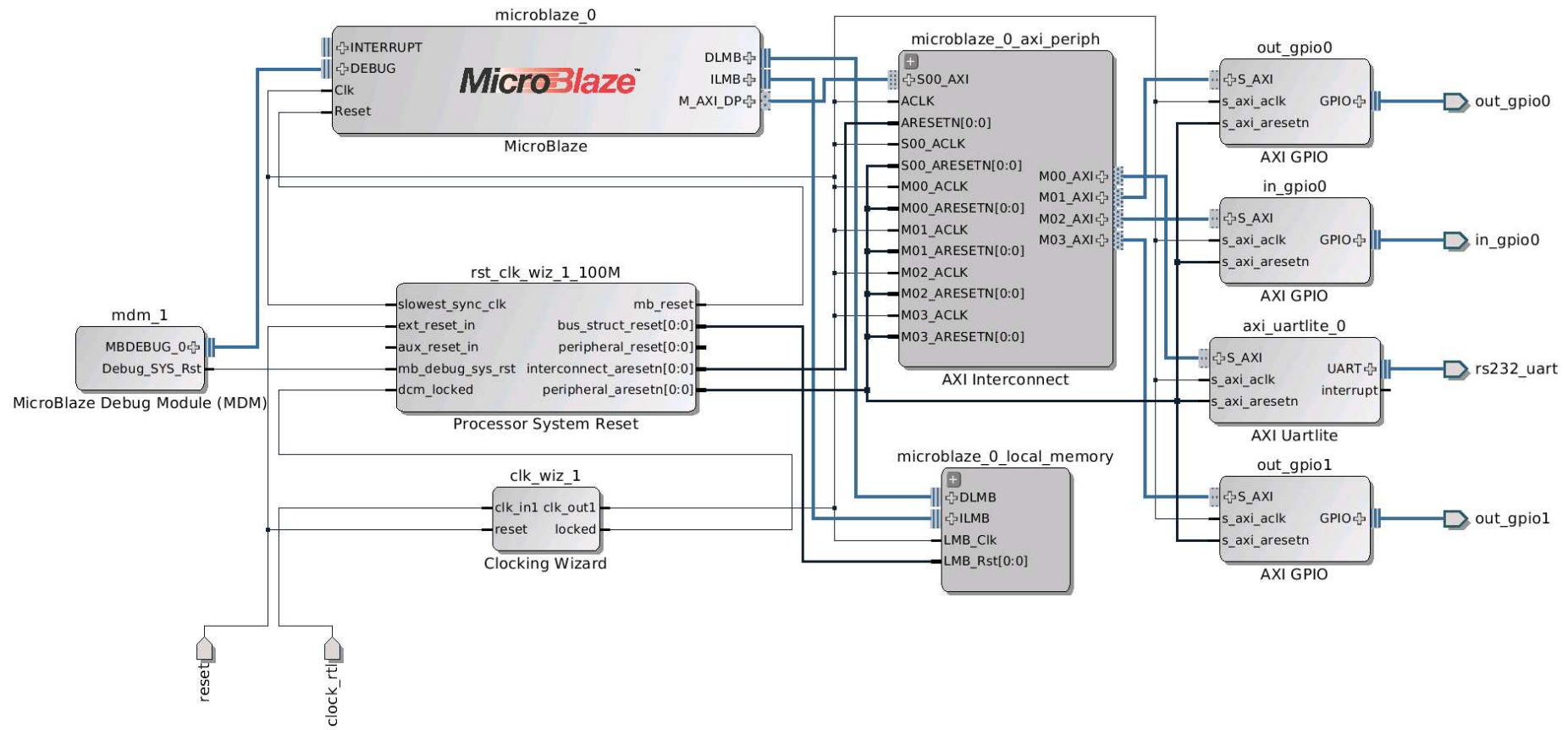


Figura 6.2: Diagrama en bloques del procesador MicroBlaze. Se detallan los módulos que son empleados en el presente diseño.

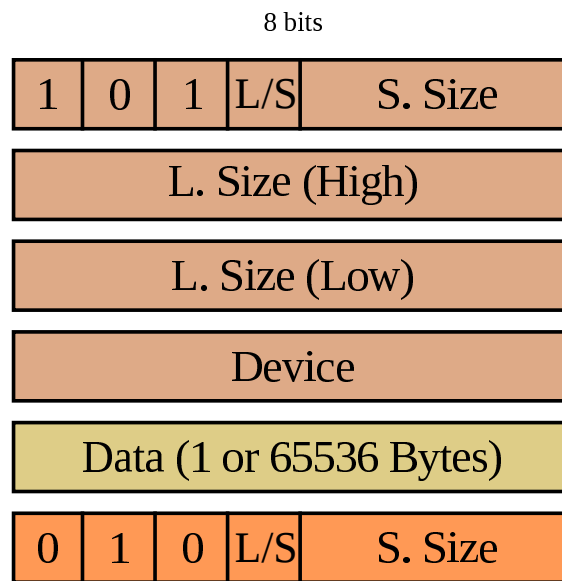


Figura 6.3: Estructura de la trama utilizada para la comunicación entre la computadora y la FPGA transferida por la interfaz UART.

La transferencia de datos entre la computadora y la FPGA se realiza utilizando un protocolo de comunicación personalizado (*custom*), el cual se compone de una cabecera, datos (comandos o logueo) y cola tal como se observa en la Fig. 6.3. A continuación se realiza una breve descripción de cada uno de los campos de la trama.

- **Cabecera:** Consta de cuatro octetos en donde se define una bandera de inicio (3 bits), el tipo de trama (1 bit, 0: trama corta y 1: trama larga), el número de octetos totales de los datos (20 bits) y el dispositivo o comando que se desea configurar o ejecutar (8 bits).
- **Datos:** Parte de la trama en donde se transporta la información.
- **Cola:** Consta de un octeto en donde se define la bandera de fin de trama (3 bits), el tipo de trama (1 bit) y la longitud de la trama en caso de ser una trama corta (4 bits).

Por último, se emplea un software desarrollado en lenguaje de programación C y otro en Python, los cuales se ejecutan en forma simultánea en la FPGA y en la computadora, respectivamente. A continuación se describe brevemente la funcionalidad de cada uno de ellos.

- **Programa en C:** El programa trabaja bajo el concepto de petición y respuesta, en donde se espera recibir un comando por intermedio de una trama enviada por la interfaz UART desde la computadora y ejecuta una tarea. Esta tarea consiste en escribir en el puerto GPO los datos necesarios para cumplir con la petición y leer la respuesta proveniente desde el hardware en el puerto GPI, en caso de ejecutar un logueo de variables. En todas las peticiones se genera una respuesta para asegurar la ejecución completa de la tarea.
- **Programa en Python:** El programa arma las tramas según las tareas que se desean ejecutar y las envía por el puerto UART. Permite la interacción con el usuario ingresando líneas de comandos en el mismo código o configurando diferentes opciones desde una terminal. En el caso de ejecutar una tarea de logueo, genera archivos de texto con los datos formato decimal. Además, controla la secuencia de inicio y habilitación de cada módulo para la ejecución de diferentes escenarios.

Con el objetivo de una mejor interpretación del funcionamiento de cada uno de los programas, la Fig. 6.4 ejemplifica el logueo de variables en los módulos DSP. Es importante destacar, que la finalidad del RF es permitir que se utilicen los GPIOs para múltiples tareas, ya que el RF es un banco de registros con múltiples funcionalidades como se detalla en la *Sección 6.3.3*. El primer paso en la tarea de logueo es armar la trama de petición en el programa en Python y enviarla por la interfaz UART. El procesador interpreta la trama y escribe en el GPO0 el comando de inicio de logueo (*Run*), el cual se guarda en el RF. El registro “Run” está conectado en el DSP directo al módulo de logueo que comienza a almacenar en todas las memorias BRAM los valores de las variables del DSP. Una vez llena la memoria (*Done*), el procesador selecciona cual de las memorias de almacenamiento se va a leer (*Mem.0*) y comienza la extracción de datos escribiendo cada dirección de memoria (*Add 0 - Add N*) en el puerto GPO0. Por cada dirección de memoria, se obtiene un dato (*Data*) que se escribe en el GPIO para ser almacenado en la memoria interna del procesador. Al finalizar la extracción de los datos, el procesador arma la trama y envía el logueo (*Logs*) por el puerto UART nuevamente a la PC. Esta metodología de trabajo se aplica a las diferentes tareas de configuración y control, necesarias para el funcionamiento de los diferentes módulos que integran el DSP.

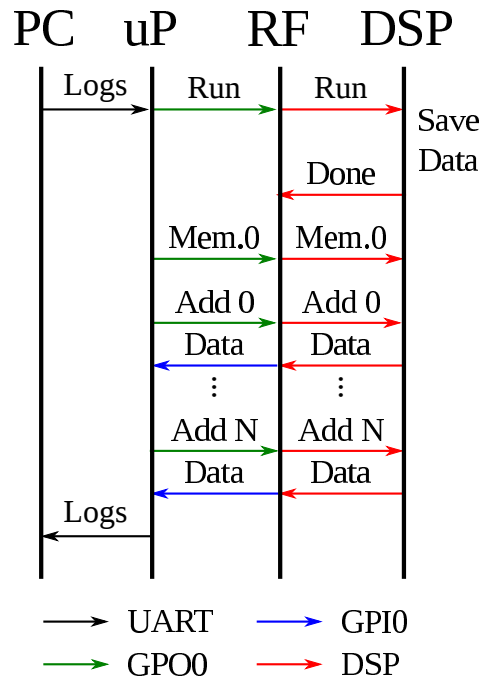


Figura 6.4: Diagrama de flujo del logueo de variables donde se observa como opera cada una de las interfaces de cada módulo.

### 6.3.2. Controlador de Reloj

El controlador de reloj en el diseño propuesto, es un bloque propietario llamado gestor de reloj de modo mixto (*Mixed-Mode Clock Manager - MMCM*) [63] encargado de generar la frecuencia de reloj para los diferentes módulos que integran la arquitectura propuesta. Las señales de reloj se generan tomando como referencia el cristal de la placa de desarrollo, cuya frecuencia es de  $200MHz$ . El MMCM genera dos frecuencias extras de  $100MHz$  y  $50MHz$  que se utilizan para el procesador y el resto de los módulos, respectivamente.

### 6.3.3. Registro de Archivo

El registro de archivo (*Register File - RF*) consiste en un banco de registros que son cargados por medio del GPO0 desde el procesador. La finalidad es almacenar los distintos comandos de control y logueo en registros que se conectan directamente a cada uno de los módulos que integran la arquitectura implementada. Se agrega además, otro RF con el objetivo de configurar los coeficientes de los filtros facilitando la verificación de los módulos del camino principal (*data path*). En la Fig. 6.5 se observan las salidas del RF y como se conecta con el procesador.

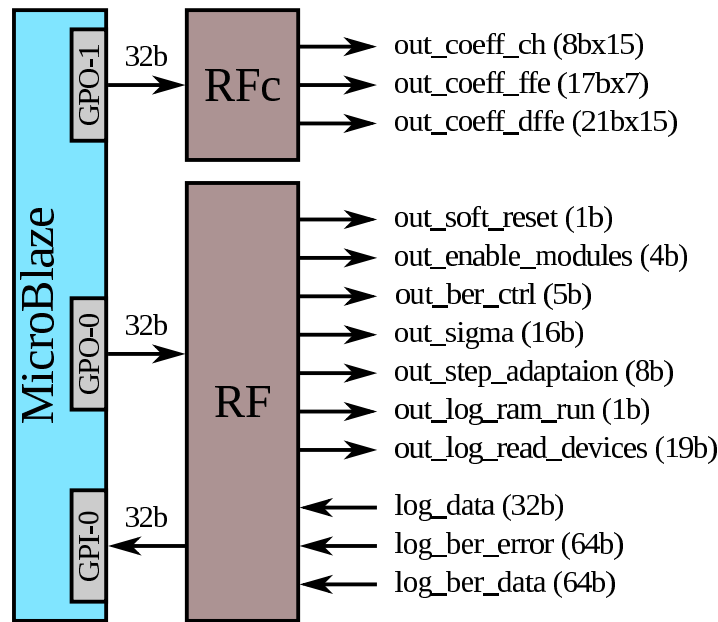


Figura 6.5: Descripción de las señales que componen el RF.

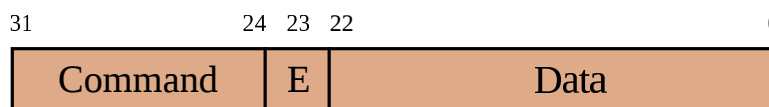


Figura 6.6: Detalle del GPO utilizado para la carga de comandos en el RF.

El funcionamiento consiste en la escritura de un registro de entrada del RF desde el GPO0 seleccionando el comando que va a ejecutar. Los 32 bits del GPO0 se dividen en selección del comando (31:24), habilitador de escritura (23) y dato a escribir en el registro de comandos (22:0), tal como se observa en la Fig. 6.6. El bit de habilitación de escritura (23) tiene la finalidad de asegurar que las señales que lleguen a los registros del RF se encuentren estables y cargue el valor correcto. Por ejemplo, para cargar el valor uno (1) en el registro de “soft reset” se tiene que escribir en el GPO0 la secuencia  $0 \times 01000001$ ,  $0 \times 01800001$  y  $0 \times 01000001$  (expresado en notación hexadecimal). Esta misma disposición de los bits se utiliza para el GPO1, donde la diferencia radica en que escribe los coeficientes de los filtros en el RFc. Para el caso particular del logueo de variables o la lectura de los errores y bits contados para la estimación de la BER, se utiliza el GPI0 como puerto de lectura. Los 32 bits del GPI0 se emplean para datos.

Por último, en la Tabla 6.1 se detallan los registros del RF describiendo la funcionalidad de cada comando.



Tabla 6.1: Detalle de comandos del RF.

Id.	Descripción	Bits	Detalle
<b>micro_dir</b>			
0	gpio_return_select	8	Selecciona el tipo de logueo que se quiere extraer.
1	out_soft_reset	1	Reinicio ( <i>reset</i> ) de los módulos del DSP.
2	out_enables_module	4	Habilita los módulos del DSP. 3: DFFE – 2: FFE – 1: GNG – 0: PRBS
3	out_ber_ctrl	5	Control del módulo de BER. 4-2: Elige cual módulo de BER leer – 1: Registra los errores y bits contados – 0: Habilita los módulos de BER
4	out_sigma_awgn	16	Valor del sigma utilizado para generar el nivel de ruido.
5	out_step_adaptation	8	Paso de adaptación del algoritmo LMS. 7-4: Paso del FFE – 3-0: Paso del DFFE
6	log_ram_run_from_micro	1	Habilita el logueo
7	log_read_devices	19	Control de lectura de las memorias de logueo. 18-4: Dirección de lectura – 3: Selección del bloque de datos a leer – 2-0: Selección de la memoria que se desea leer.
<b>gpio_return_select</b>			
0	in_log_capture_data	32	Datos de la memoria de logueo
1	in_ber_error_counter	32	LSB (31:0) de los errores del contador de BER
2	in_ber_error_counter	32	MSB (63:32) de los errores del contador de BER
3	in_ber_data_counter	32	LSB (31:0) de los bits del contador de BER
4	in_ber_data_counter	32	MSB (63:32) de los bits del contador de BER

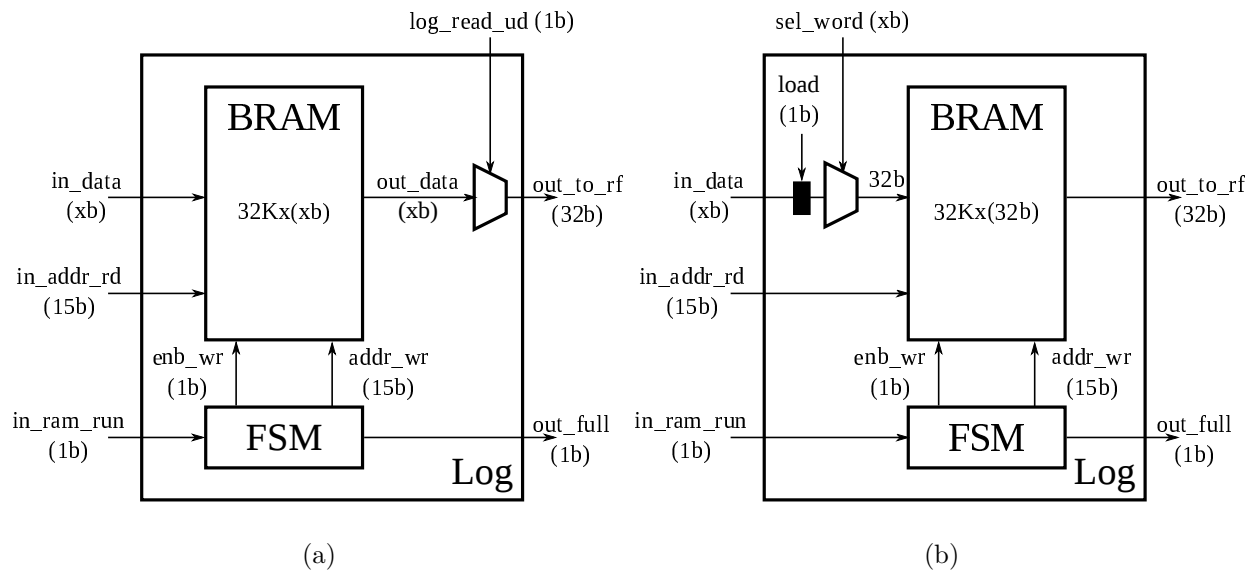


Figura 6.7: Configuraciones de las memorias de logueo de variables. (a) Configuración utilizada en el logueo de la señal de salida del canal con ruido, salida del FFE y el error de detección. (b) Configuración utilizada en el logueo de los coeficientes de los ecualizadores.

### 6.3.4. Módulo de Logueo

El módulo de logueo consiste en un bloque de memoria RAM (*block RAM - BRAM*) [64] de doble puerto, el cual almacena el valor de las variables de los módulos del camino de datos (*data path*). La instancia de cada BRAM incluye una máquina de estado finita (*Finite State Machine - FSM*) que controla el almacenamiento completo de los datos dentro de la memoria. Las señales que se pueden loguear son la salida de la PRBS9, la salida del canal con ruido, la salida del FFE, la salida del DFFE, el error de detección y los coeficientes de ambos ecualizadores. Dependiendo la señal, se utiliza una configuración diferente de la memoria de logueo. Para las señales de salida del FFE, de salida del canal con ruido y del error de detección se emplea la configuración que se observa en la Fig. 6.7(a). La particularidad de esto es que se escribe en cada fila de la memoria las ocho (8) palabras que integran cada una de las señales. Hay que destacar que cada palabra no supera los 8 bits. El multiplexor de salida selecciona los MSB (63:32) o LSB (31:0) de cada fila con el objetivo de generar una salida de la memoria de datos de 32 bits. En el caso de los coeficientes de los ecualizadores se utiliza la configuración que se detalla en la Fig. 6.7(b). A diferencia de la anterior, cada fila de la memoria representa un coeficiente diferente. Esto se logra copiando a un registro todos los coeficientes para luego almacenar

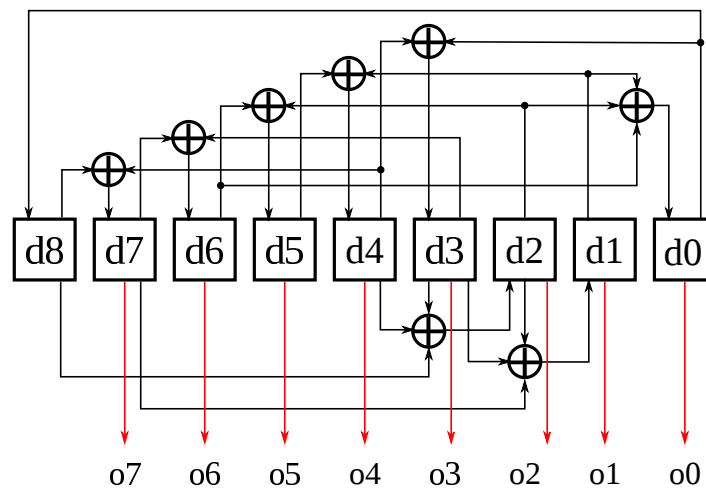


Figura 6.8: Esquemático del módulo PRBS9 paralelo, detallando las operaciones lógicas XORs utilizadas en la actualización de los registros.

cada uno de ellos en filas distintas. Por último, la salida de la PRBS9 y la salida del DFFE se almacenan en una tercer configuración que no contiene ningún multiplexor a la entrada o salida. El número de bits de cada fila de esta memoria es de 16 bits (15-8: DFFE - 7-0: PRBS9).

### 6.3.5. Módulo PRBS9 Paralelo

El módulo PRBS9 paralelo es el primero en el camino de datos (*data path*), el cual genera los bits que se transmiten hacia el canal. El diseño de la PRBS9 se basa en [65] donde se propone un esquema funcional para este tipo de operación, en donde se considera un paralelismo de ocho (8). La implementación consiste en un banco de registros que se actualizan según la operación lógica XOR tal como se observa en la Fig. 6.8. La modulación utilizada es PAM antipodal, por lo que se mapea el bit “1” como el símbolo +1 y el bit “0” como el símbolo -1.

### 6.3.6. Módulo de Canal

El módulo de canal consta de un banco de filtros en paralelo y generadores de ruido Gaussiano (*Gaussian Noise Generator - GNG*) conectados como se muestra en la Fig. 6.9. El banco de filtros consiste en filtros FIR de quince (15) coeficientes que tiene la finalidad de representar la dispersión del canal. El criterio de diseño de las operaciones aritméticas

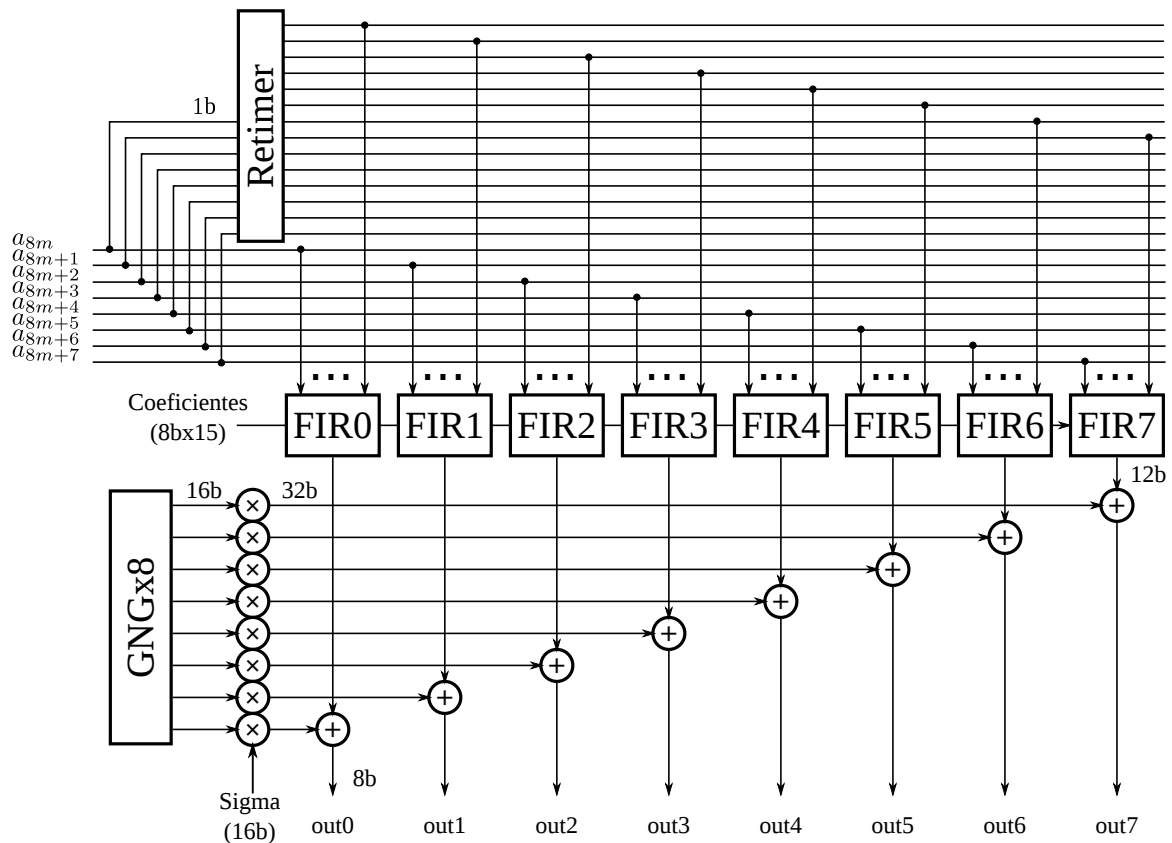


Figura 6.9: Diagrama en bloques del banco de filtros y generadores de ruido Gaussianos utilizados para generar los efectos de dispersión y ruido en el canal.

es máxima resolución y se aplica truncado y saturación a 8 bits a la salida del canal.

El paralelismo de entrada es ocho y se utiliza un “retimer” para sincronizar las muestras para cada uno de los filtros. Todos los filtros se cargan con el mismo valor de coeficientes utilizando el puerto de configuración proveniente del módulo RFc. A la salida de los filtros se suma ruido, el cual es generado por los módulos GNGs instanciados en forma independientes. Esto quiere decir, que existen ocho instancias del GNG los cuales tienen semillas distintas y se multiplica por un factor “sigma” que determina el nivel de ruido. Este parámetro se ajusta en forma dinámica utilizando el puerto del módulo RF, el cual el usuario puede modificarlo en cualquier instante de tiempo.

La implementación de los filtros se realizan considerando los conceptos básicos de paralelización especificados en [24]. Por otro lado, el módulo GNG se implementó tal como se describe en [66].

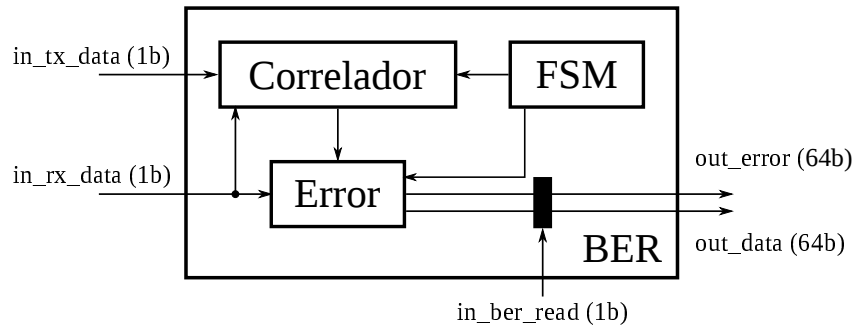


Figura 6.10: Diagrama en bloques del módulo de BER.

### 6.3.7. Módulo de BER

El módulo de BER tiene la finalidad de contar los errores que se producen cuando se toma una decisión incorrecta en el detector de umbral causada por la dispersión o el ruido en el canal. El diagrama en bloques se describe en la Fig. 6.10, el cual consiste en un correlador, un comparador (Error) y una FSM. Este módulo se instancia ocho veces (un módulo por cada salida del paralelismo) y para contabilizar la BER total se suman los errores y los bits contados de todos los módulos.

El correlador tiene como objetivo correlacionar las muestras transmitidas con las decisiones estimadas a la salida del DFFE calculando el retardo entre ellas. El correlador consiste en un banco de registros de longitud 1024 que acumula en cada posición  $511^2$  comparaciones. Esta operación se realiza utilizando la compuerta lógica XOR. Una vez completado el banco de registros, se determina cual es la posición con menor número de errores, estableciendo como salida del correlador este punto. Luego de determinar el retardo, el bloque de error comienza a acumular los errores entre los bits transmitidos y detectados y el número de bits contados. La secuencia de eventos mencionados es controlado por una FSM.

Por último, el acumulador de errores y bits contados está representado con una resolución de 64 bits. Antes de leer estos dos valores, se copian a un registro de salida por medio de una señal de habilitación. La copia de los datos evita un cambio en los valores registrados al momento de hacer la lectura desde el procesador.

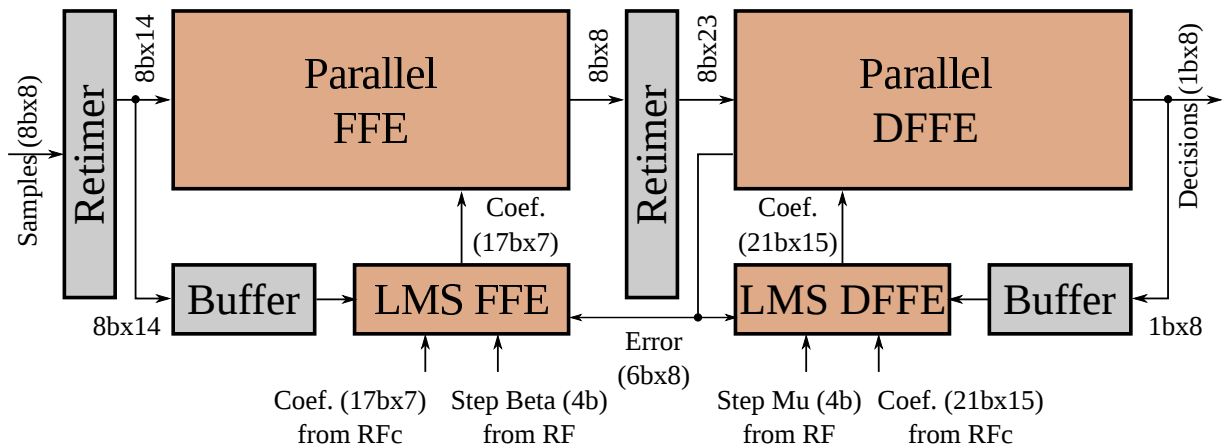


Figura 6.11: Diagrama en bloques de los ecualizadores.

### 6.3.8. Módulo de Ecualizadores

El módulo de ecualizadores consiste en la implementación del FFE y DFFE con un paralelismo de ocho (8) en conjunto con el algoritmo de adaptación LMS para ambos ecualizadores, donde la distribución de los bloques se muestra en la Fig. 6.11. Las muestras de entrada provenientes del canal se ordenan utilizando un “retimer”, para luego ser procesadas por el FFE. El FFE paralelo se implementa de forma similar que en [60], con la diferencia que las operaciones aritméticas se realizan con multiplicadores y sumadores. En la implementación en FPGA es recomendable utilizar este tipo de criterio, ya que estos dispositivos poseen bloques DSP [67] dedicados para tal fin.

La salida del FFE pasa a otro “retimer” que ordena nuevamente las muestras para ser procesadas por el DFFE. La estructura de implementación del DFFE paralelo es igual a la detallada en el *Capítulo 5* con diferencias en el criterio de diseño utilizado para los resultados de síntesis de la *Sección 5.2.3*. La diferencia entre ambas implementaciones se encuentra en los “buffers” dentro de los bloques DFFE. Para este caso se utilizan registros de desplazamiento (*shift register*), ya que la FPGA posee estos bloques dedicados haciendo que sea eficiente la utilización de los recursos. En el caso de los productos entre los coeficientes y las decisiones tentativas, se instancian multiplexores que eligen entre el valor positivo y negativo del coeficiente.

Por otro lado, ambos ecualizadores trabajan con el algoritmo de adaptación LMS [6]. El LMS utiliza el error de estimación calculado a la salida del DFFE. Para cada uno de

<sup>2</sup>Acumula 511 comparaciones porque la secuencia de PRBS9 se repite con esa periodicidad

los casos, se emplea un “buffer” que permite sincronizar las muestras (muestras del canal y decisiones) que se utilizan en la adaptación de los coeficientes. La profundidad de estos bloques depende directamente con el número de retardos que se encuentren en el camino principal (*data path*). El diseño propuesto tiene la capacidad de cargar en forma independiente los coeficientes de cada uno de los ecualizadores, permitiendo realizar pruebas de desempeño sin la necesidad de utilizar el algoritmo LMS. La inicialización de los coeficientes se realiza por medio del módulo RFc. Además, los bloques LMS trabajan con pasos de adaptación independientes que pueden ser modificados en cualquier instante de tiempo pudiendo aplicar la técnica de cambio de velocidad (*gear shift*) [6]. La aplicación del paso de adaptación se realiza por medio de corrimientos desde  $2^{-1}$  hasta  $2^{-15}$  implementados sobre una tabla.

Teniendo en cuenta todos los conceptos de diseño analizados en esta sección, a continuación se evalúa la complejidad de implementación de la arquitectura propuesta. Esto involucra el análisis de reportes de síntesis y de implementación, los cuales muestran los recursos utilizados en la FPGA.

## 6.4. Reporte de Síntesis e Implementación

Antes de analizar los reportes de síntesis, es necesario conocer brevemente la FPGA en la cual se implementó el sistema. La placa de desarrollo utilizada es la KC705 [61], la cual tiene montada una FPGA Kintex 7 de última generación de tecnología CMOS 28nm. La placa KC705 posee diferentes interfaces de las cuales podemos destacar Ethernet, UART y JTAG. Para la generación de estímulos externos, tiene montados llaves (*switch*) y pulsadores<sup>3</sup>. De los recursos disponibles en la FPGA Kintex7, la unidad principal es el bloque lógico configurable (*Configurable Logic Block - CLB*) [68]. La función principal del CLB es brindar el acceso de la lógica digital a la matriz de encaminamiento (*matrix routing*). El CLB se compone de dos unidades básicas llamadas “Slice” que contienen compuertas lógicas, bloques aritméticos, bloques de memoria, registros y generadores de funciones lógicas (*look-up tables - LUT*) [68]. Estas unidades son las que reporta la herramienta de síntesis en las estimaciones de complejidad.

El flujo de diseño se realizó con la herramienta Vivado 2013.4 de la empresa Xilinx. Una

---

<sup>3</sup>A los pulsadores se le dió funcionalidad de reinicio en el diseño propuesto.

Tabla 6.2: Reporte de síntesis donde se detalla el número de celdas discriminado por jerarquía.

Instancia	Celdas
Top	159125
u_microblaze	9627
u_register_file_prmt	1183
u_register_file	1034
u_clock_manager	5
u_top_phy	141987
u_ber_checker_top	20666
u_gng_parallel0	6761
u_parallel_prbs_tx	17
u_fir_channel_parallel	2095
u_top_rx	110889
u_dffe_filter_parallel	106982
u_dffe_adap	43160
u_ffe_filter_parallel	1584
u_ffe_adap	910

primera aproximación de la complejidad de la arquitectura implementada es el reporte de celdas [69] obtenidos en la etapa de síntesis. Este consiste en el conteo de instancias de diferentes componentes como ser registros (*flip-flop*), LUTs, I/O “buffers”, RAMs y DSPs, dando una primera aproximación de la complejidad de cada uno de los módulos implementados. En la Tabla 6.2 se detalla el número de celdas utilizadas para cada una de las jerarquías que componen el diseño propuesto. Para una mejor interpretación de la Tabla 6.2, en la Fig. 6.12 se grafica el número de celdas por jerarquía. En la Fig. 6.12(a) se representa el nivel superior (*top level*) del diseño, el cual involucra el procesador (*uP*), el registro de archivo en conjunto con el controlador de reloj (*RF-Clk*) y la jerarquía que involucra el resto de los módulos (*top-phy*). Sobre esta última jerarquía, en la Fig. 6.12(b) se detallan los módulos de contador de BER, generadores de ruido (*GNG*), PRBS9 y



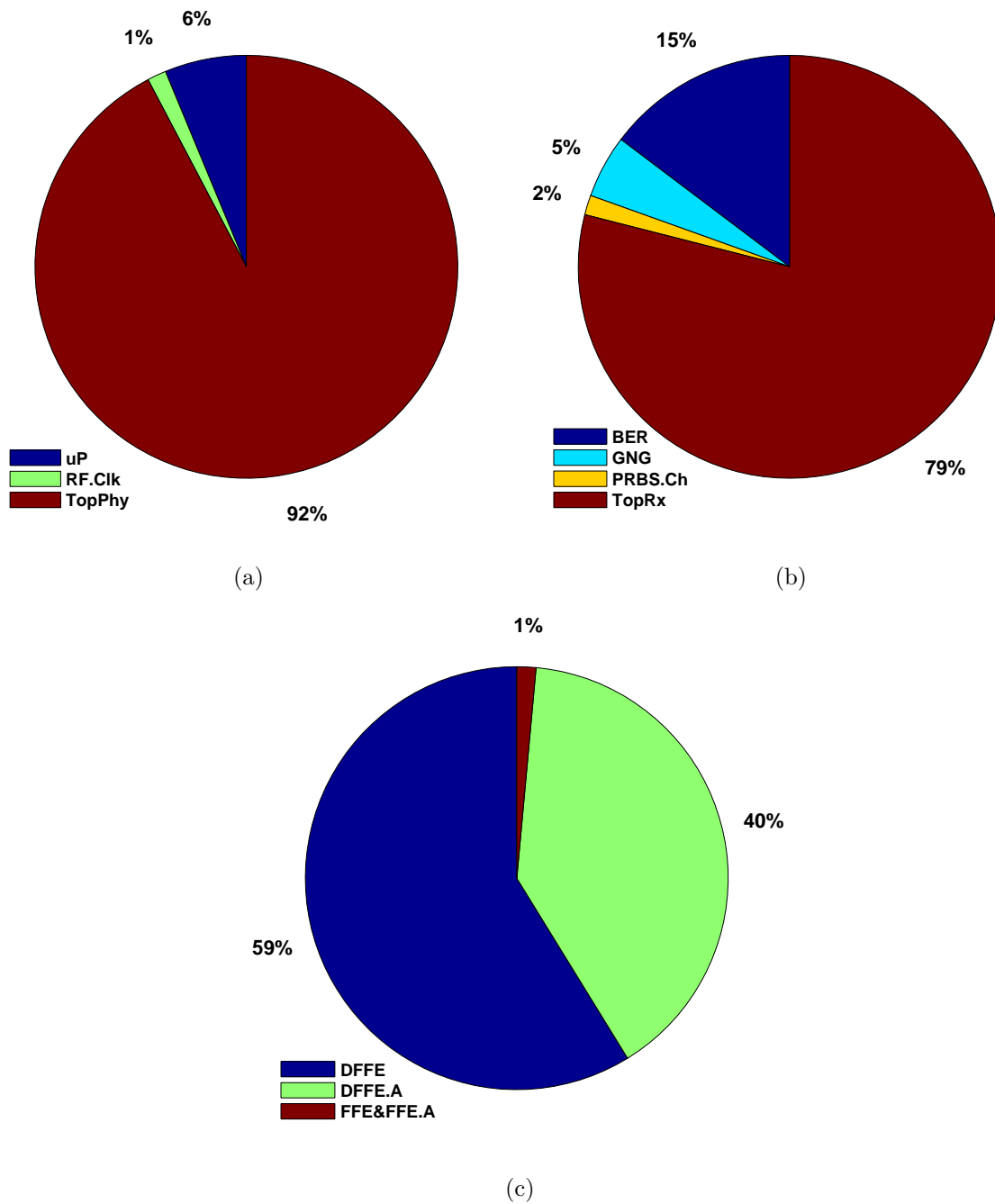


Figura 6.12: Comparación de complejidad de las diferentes jerarquías que integran el diseño implementado en base al número de celdas detalladas en la Tabla 6.2. (a) Nivel superior (*top-level*). (b) Nivel de integración de módulos (*top-phy*). (c) Nivel de ecualización (*top-rx*).

canal (*PRBS-Ch*) y la jerarquía que involucra las etapas de ecualización en conjunto con el algoritmo de adaptación (*top-rx*). Finalmente, en la Fig. 6.12(c) se detalla la etapa de ecualización.

## Capítulo 6. Implementación en FPGA del DFFE

---

Tabla 6.3: Reporte de síntesis donde se detalla el número de “Slice” utilizados en el diseño considerando la etapa de implementación.

Instancia	Usados	Disponible	%
Slice	24540	50950	48.16
LUT as Logic	56508	203800	27.72
using O5 output only	24		
using O6 output only	38866		
using O5 and O6	17618		
LUT as Memory	14668	64000	22.91
LUT as Distributed RAM	64		
using O5 and O6	64		
LUT as Shift Register	14604		
using O5 output only	14256		
using O6 output only	179		
using O5 and O6	169		
LUT Flip Flop Pairs	67322	203800	33.03
fully used LUT-FF pairs	19574		
LUT-FF pairs with unused LUT	10426		
LUT-FF pairs with unused Flip Flop	37322		
Unique Control Sets	584		

En la etapa de implementación, el reporte de utilización de recursos nos brinda una perspectiva de la complejidad del sistema calificado la utilización de “Slice”, los cuales se detallan en la Tabla 6.3. Este reporte también considera los recursos que se emplean para el encaminamiento (*routing*), lo cual es clave en los diseños de FPGA. Notar que el 48,16 % de los “Slice” son empleados con diferentes finalidades. Del total de la LUTs utilizadas, el 27,72 % operan como lógica combinacional y el 22,91 % como memoria. Haciendo foco principalmente en las LUTs y los registros (dos componente importantes en la FPGA), en la Tabla 6.4 se discrimina el número de instancias de cada uno de ellos. Observar que para

## 6.4. Reporte de Síntesis e Implementación

Tabla 6.4: Reporte de síntesis donde se detalla el número de LUTs utilizados en el diseño considerando la etapa de implementación.

Instancia	Usados	Disponible	%
Slice LUTs	71176	203800	34.92
LUT as Logic	56508	203800	27.72
LUT as Memory	14668	64000	22.91
LUT as Distributed RAM	64		
LUT as Shift Register	14604		
Slice Registers	49739	407600	12.2
Register as Flip Flop	49739	407600	12.2
F7 Muxes	3161	101900	3.1
F8 Muxes	1446	50950	2.83

Tabla 6.5: Reporte de síntesis donde se detalla el número de BRAM y DPS utilizados en el diseño.

Instancia	Usados	Disponible	%
Block RAM	234	445	52.58
DSPs	136	840	16.19

el caso de LUTs como memoria, un gran porcentaje está representado como registro de desplazamiento (*shift register*). Esto justifica la descripción de la arquitectura desarrollada en la sección anterior, donde se menciona la utilización de este tipo de estructura para un mejor aprovechamiento de los recursos. Por último, en la Tabla 6.5 se contabiliza el número de BRAMs y DSPs. Hay que resaltar que el número de bloques de memoria es elevado, esto se debe a la capacidad de logueo que presenta el diseño con el fin de facilitar la revisión de cada uno de los módulos implementados.

Acompañado con el reporte del número de “Slice”, otro parámetro importante a considerar en cualquier diseño en FPGA es el tiempo de holgura (*slack time*). En el caso de ser un valor negativo identifica un problema de exceso de tiempo entre dos puntos, los cuales pueden ser un puerto de entrada y registro, registro y registro o registro y puerto

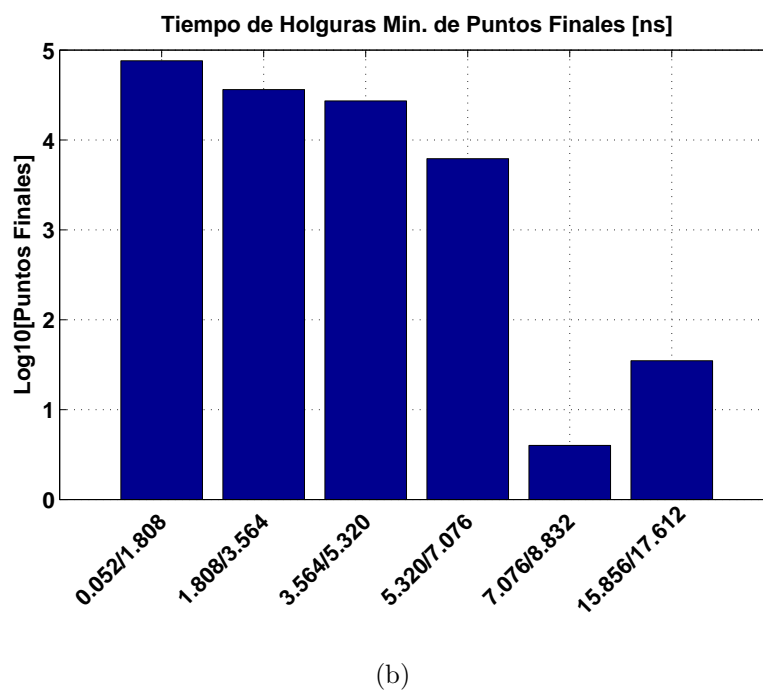
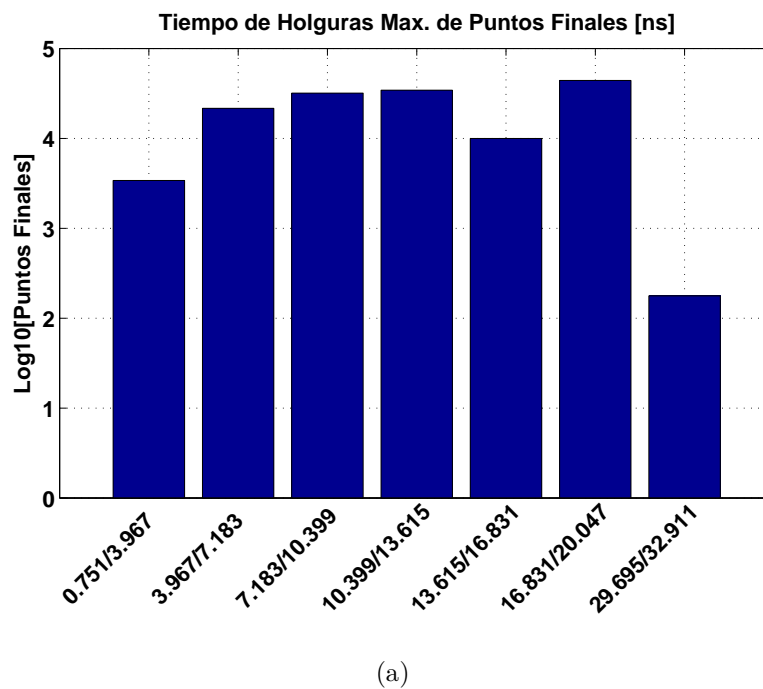


Figura 6.13: (a) *Histograma de tiempo de holgura de configuración (setup)*. (b). *Histograma de tiempo de holgura de retención (hold)*.

de salida. El exceso de tiempo entre dos punto expresa que el tiempo de propagación de la señal entre dos elementos definidos es mayor que el período de reloj. Una forma sencilla de observar estos valores es por medio de histogramas que clasifican el tiempo de holgura de configuración/retención (*setup/hold*) [70, 71]. El tiempo de holgura máximo de punto

final<sup>4</sup> (*maximum slack time endpoint*) esta asociado con el tiempo de configuración (*setup*) mostrado en la Fig. 6.13(a). Del mismo modo, el tiempo de holgura mínimo de punto final (*minimum slack time endpoint*) representa el tiempo de retención<sup>5</sup> (*hold*), el cual se grafica en la Fig. 6.13(b). Es importante resaltar que en ningún caso se observan valores negativos de tiempo.

Por último, en la Fig. 6.14 se muestra un detalle de la distribución de los componentes y encaminamiento (*routing*) de las señales. Esto permite tener una mejor referencia de la complejidad detalla en las tablas anteriores. En la Fig. 6.15 se observa la implementación sin considerar el conexionado identificando con diferentes colores los módulos principales. El color rojo representa el DFFE paralelo incluyendo la etapa de adaptación de coeficientes, el color verde remarca el FFE y el algoritmo de adaptación, el color amarillo identifica el procesador MicroBlaze y el color celeste el resto de los módulos.

En conclusión, los reportes de tiempo demuestran una implementación correcta de la arquitectura, ya que no se identifican tiempos de holgura negativos. Desde el punto de vista de los recursos, se aprovecharon los bloques de registro de desplazamiento (*shift register*) como elemento principal para el diseño del DFFE, tal como se observa en las Tabla 6.4. También es importante resaltar la flexibilidad que brinda la utilización de algoritmos de estructuras directas (*forward*), facilitando el encaminamiento (*routing*) de las señales sin generar problemas de congestiones (*routing*) en los distintos sectores de la FPGA.

---

<sup>4</sup>**Ejemplo de punto final (endpoint):** Para los circuitos internos, el camino de datos (*data path*) es la trayectoria entre el lanzamiento y la captura de dispositivos secuenciales (“A-B”) [71]. El punto de inicio del camino de dato (*data path startpoint*) es el puerto de reloj del dispositivo secuencial “A”. El punto final del camino de dato (*data path endpoint*) puerto de entrada de dato del dispositivo secuencial “B”.

<sup>5</sup>El detalle de los parámetros de configuración/retención (*setup/hold*) se especifican en [70].

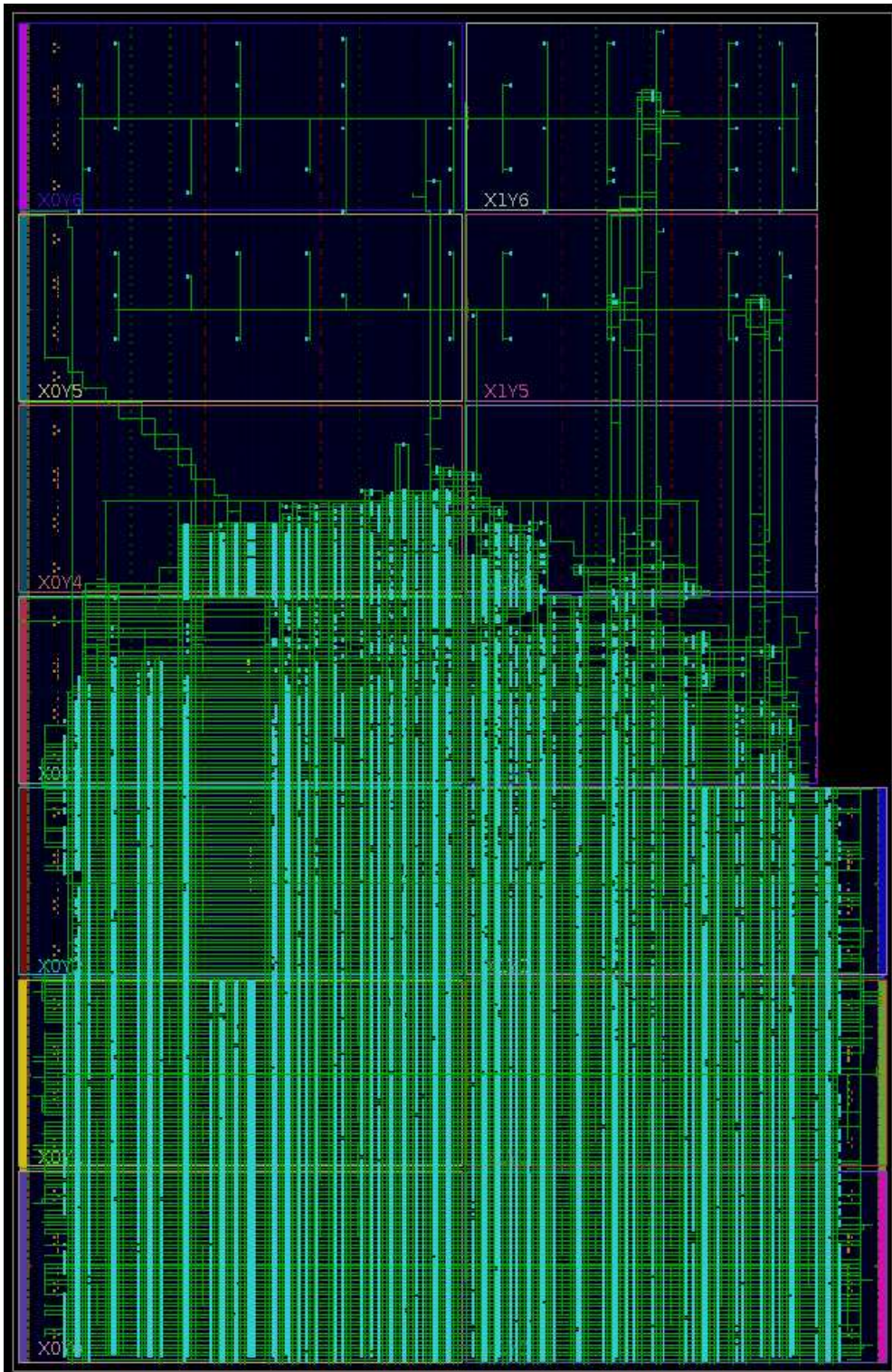


Figura 6.14: Resultado de la implementación de la arquitectura propuesta detallando el conexionado de las señales.

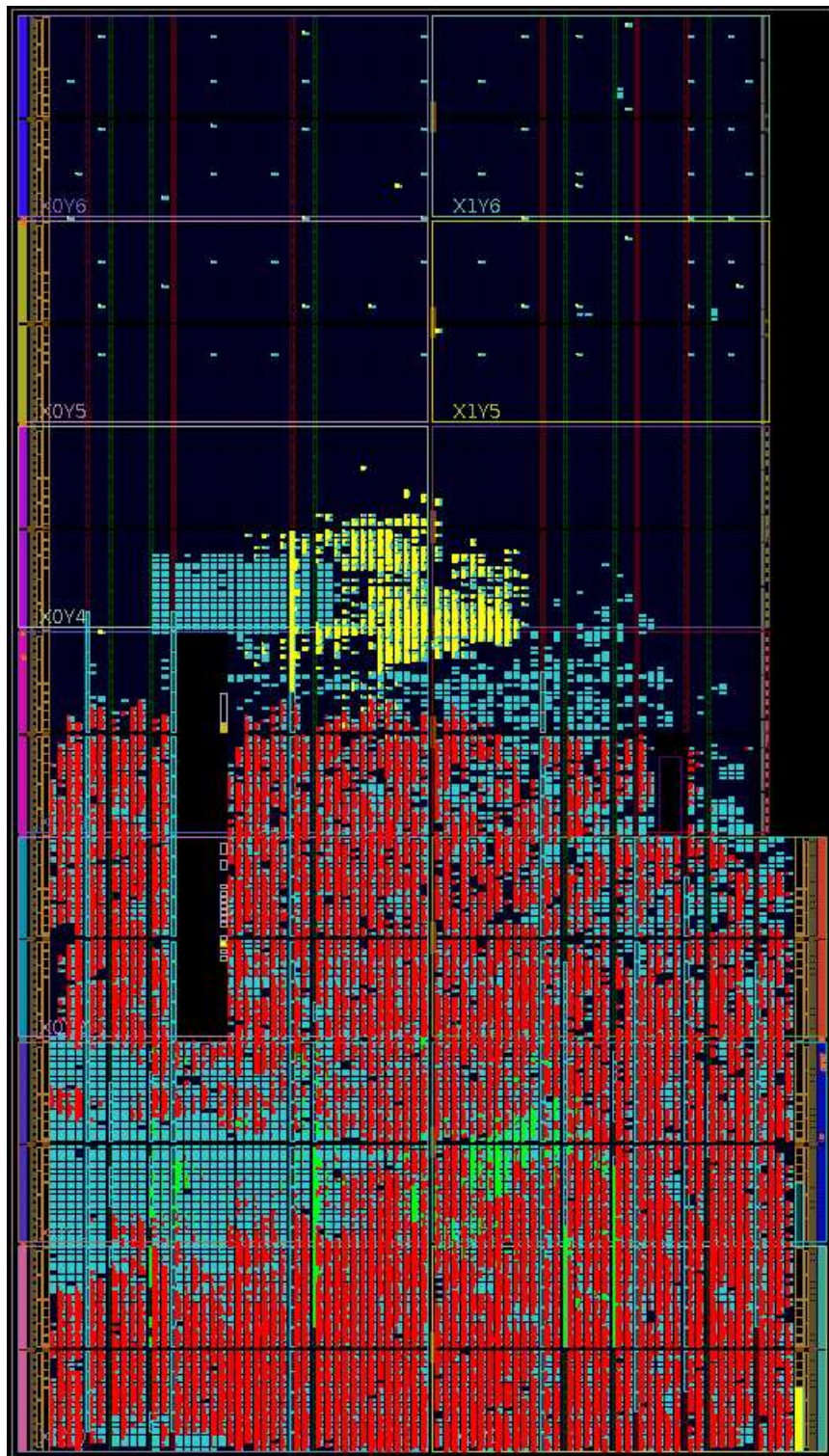


Figura 6.15: Resultado de la implementación de la arquitectura propuesta solamente considerando las instancias de los bloques lógicos de la FPGA. El color rojo representa el DFFE paralelo incluyendo la etapa de adaptación de coeficientes, el color verde remarca el FFE y el algoritmo de adaptación, el color amarillo identifica el procesador MicroBlaze y el color celeste el resto de los módulos.

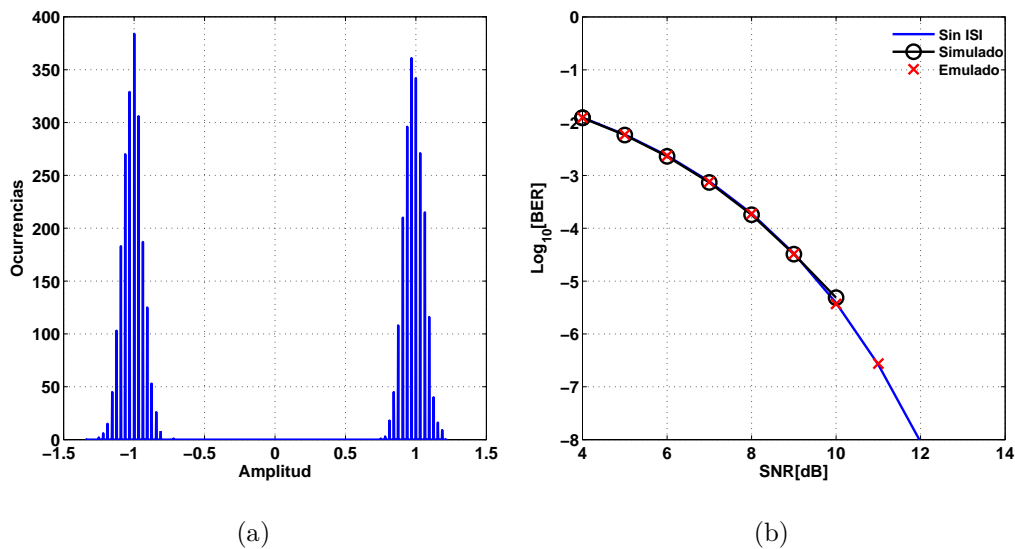


Figura 6.16: *Comportamiento de la señal a la entrada de la etapa de ecualización. (a) Histograma de la entrada del detector de umbral de la última iteración del DFFE. (b) Curva de BER del canal Gaussiano.*

## 6.5. Mediciones

El objetivo de esta sección es corroborar el funcionamiento de la arquitectura implementada en la FPGA comparando las mediciones con simulaciones realizadas en el simulador de punto fijo. La verificación consiste en la ejecución y análisis de tres escenarios de prueba para evaluar la calibración del nivel de ruido, desempeño del FFE y el desempeño del FFE-DFFE.

### 6.5.1. Calibración del Nivel de Ruido

El primer paso consiste en calibrar el nivel de ruido del canal en la FPGA cargando los coeficientes del filtro FIR del canal con el impulso unitario y definiendo un desvío estándar ( $\sigma$ ) para una  $SNR = 20dB$  ( $SNR = E|a_n|^2/\sigma^2$ ). Los ecualizadores no se adaptan haciendo solamente que opere el detector de umbral de la última iteración del DFFE. Para este escenario, se habilitan los módulos PRBS9, canal y contadores de BER. En la Fig. 6.16 se muestra el comportamiento de la señal a la entrada de la etapa de ecualización. El histograma de la Fig. 6.16(a) verifica la modulación (2-PAM antipodal) y el funcionamiento de los módulos de ruido Gaussiano implementados en la arquitectura propuesta. Además, en la Fig. 6.16(b) se grafica la curva de BER comparando la curva teórica del



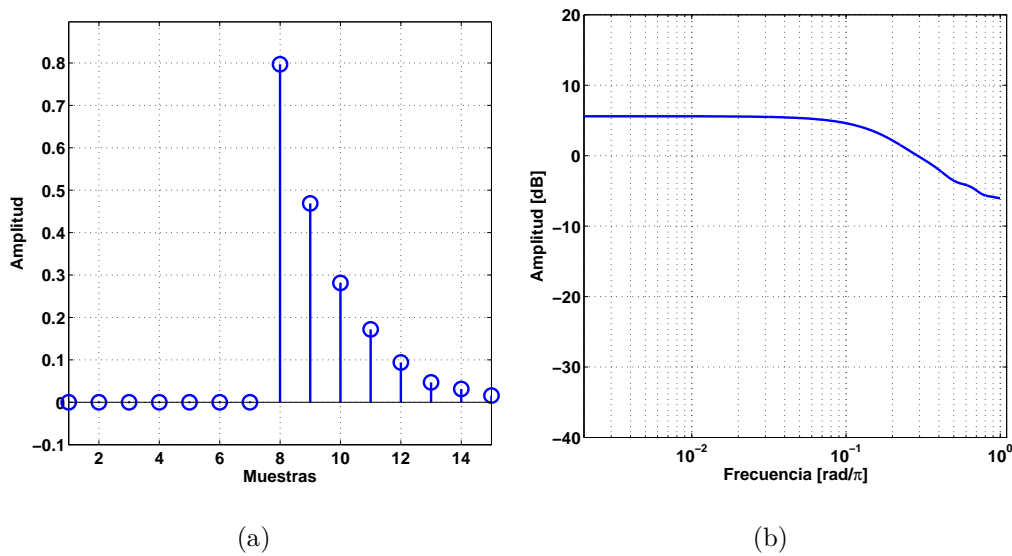


Figura 6.17: (a) *Respuesta al impulso del canal* ( $p_k = \alpha^k \mu_k$  con  $\alpha = 0,6/L = 15$ ). (b) *Espectro plegado del canal equivalente*.

canal sin ISI, el desempeño del simulador de punto fijo y las mediciones en la FPGA. En todos los casos mencionados se aprecia un comportamiento similar de los sistemas.

### 6.5.2. Desempeño del FFE

El siguiente paso es evaluar el desempeño del FFE. La arquitectura implementada esta diseñada para habilitar los módulos sin estar condicionados entre ellos. Este concepto facilita la verificación en la etapa de diseño. El escenario propuesto toma como base lo desarrollado en el *Capítulo 2*, donde se utiliza un canal exponencial con respuesta al impulso ( $p_k = \alpha^k \mu_k$  con  $\alpha = 0,6/L = 15$ ) y espectro plegado detallados en la Fig. 6.17. Notar que la respuesta al impulso del canal es similar a la utilizada en la Sección 2.6, con la diferencia que los valores de amplitud se encuentran definidos para una resolución finita. Para el escenario elegido se considera una  $SNR = 15dB$  y se emplea el algoritmo de adaptación LMS en conjunto con la técnica de cambio de velocidad (*gear shift*) ( $\beta = [2^{-11}, 2^{-13}]$ )

El respuesta del FFE en el escenario propuesta se observa en la Fig. 6.18 alcanzando un comportamiento similar al estudiado en el *Capítulo 2*. En la Fig. 6.19 se muestran diferentes logueos medidos en la FPGA. En la Fig. 6.19(a) se grafica el histograma de la señal de entrada al ecualizador, en la Fig. 6.19(b) el error entre la decisión y la señal de entrada al detector de umbral y en la Fig. 6.19(c) el comportamiento temporal de los

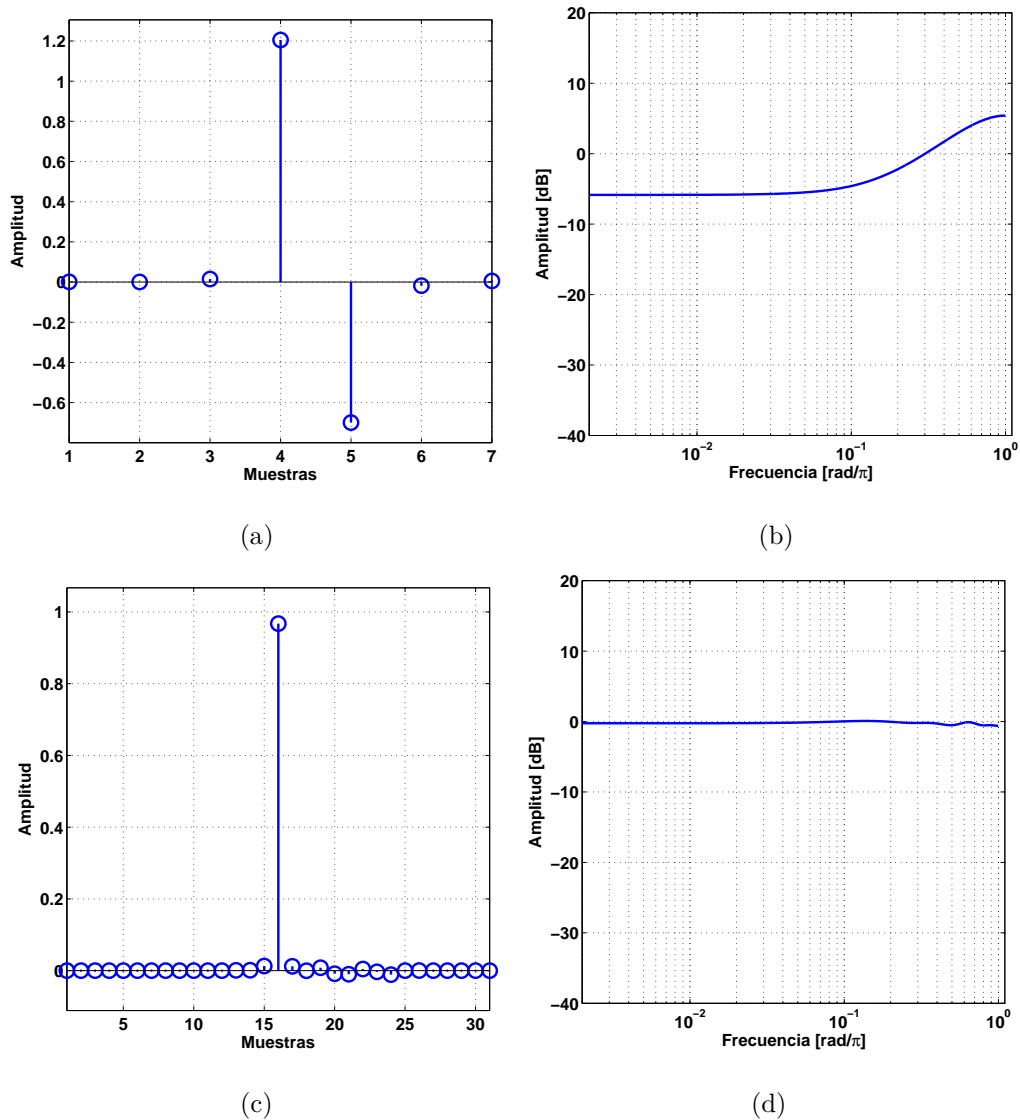


Figura 6.18: *Comportamiento del FFE. (a) y (b) Respuesta del FFE. (c) y (d) Respuesta del canal equivalente en la entrada del detector de umbral.*

coeficientes del ecualizador. Por último, en la Fig. 6.19(d) se compara el desempeño del FFE entre el simulador de punto fijo y la emulación en la FPGA. En conclusión, ambos sistemas se comportan de forma similar.

### 6.5.3. Desempeño del FFE-DFFE

Como etapa final, se analiza el desempeño de ambos ecualizadores, considerando el mismo canal detallado en la Fig. 6.17 con una  $SNR = 20dB$ . A diferencia del caso anterior, se agrega el DFFE con los pasos de adaptación calibrados en  $\mu = [2^{-10}, 2^{-12}]$ . En la Fig. 6.20 se grafica el comportamiento de los coeficientes de los ecualizadores y en la Fig. 6.21

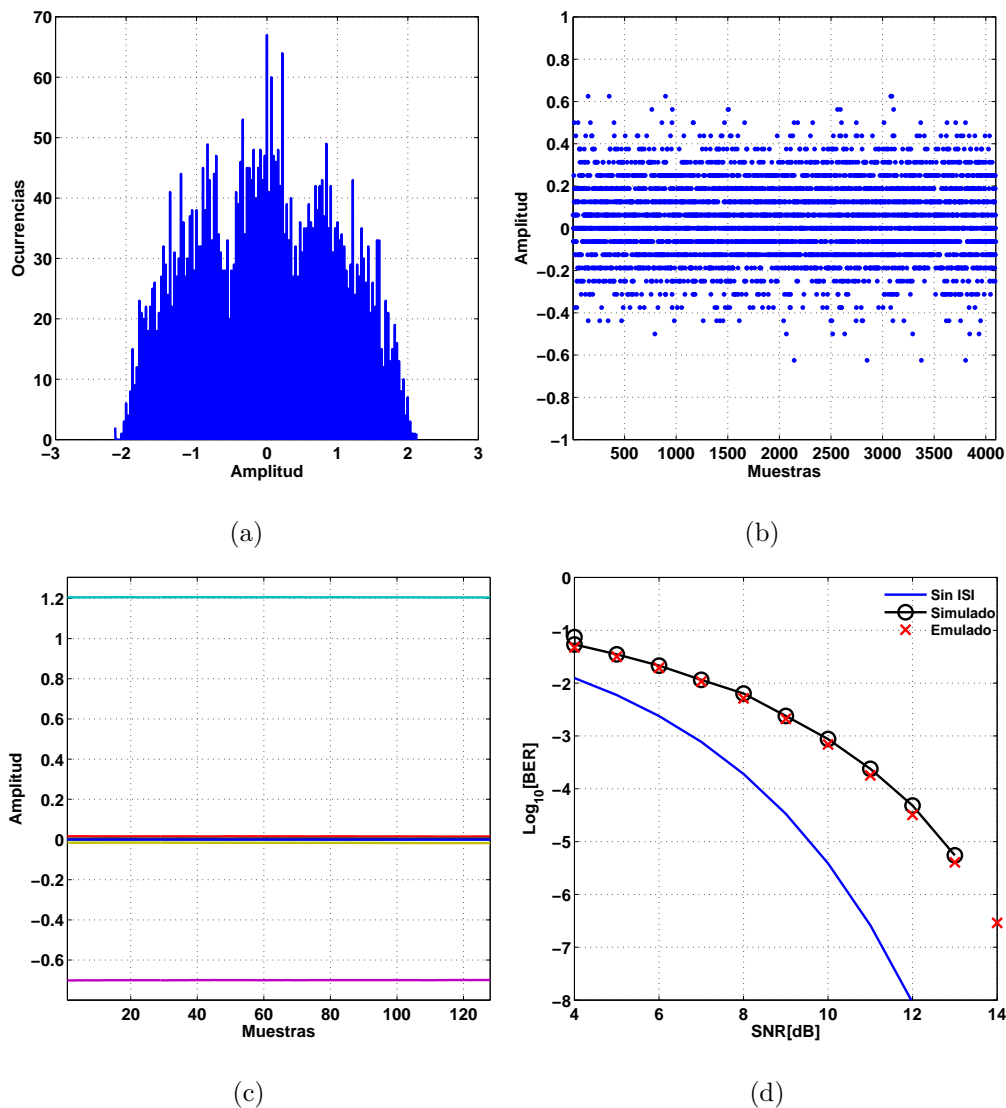


Figura 6.19: *Comportamiento del FFE. (a) Histograma de la señal de entrada al FFE. (b) Señal de error estimada entre el símbolo detectado y la señal de entrada al detector de umbral. (c) Comportamiento temporal de los coeficientes del FFE. (d) Desempeño del FFE.*

se observan diferentes señales logueadas desde la FPGA. Notar que en la Fig. 6.21(e) el desempeño del FFE-DFFE emulado en la FPGA es similar al obtenido en el simulador de punto fijo. Por último, en la Fig. 6.22 se muestra una comparación del desempeño entre el FFE y FFE-DFFE. Observar que, al igual que en el *Capítulo 2*, el DFFE mejora el desempeño del receptor. En conclusión, la arquitectura implementada agrega un marco práctico al desarrollo del DFFE logrando un desempeño similar que el DFE analizado en los capítulos anteriores.

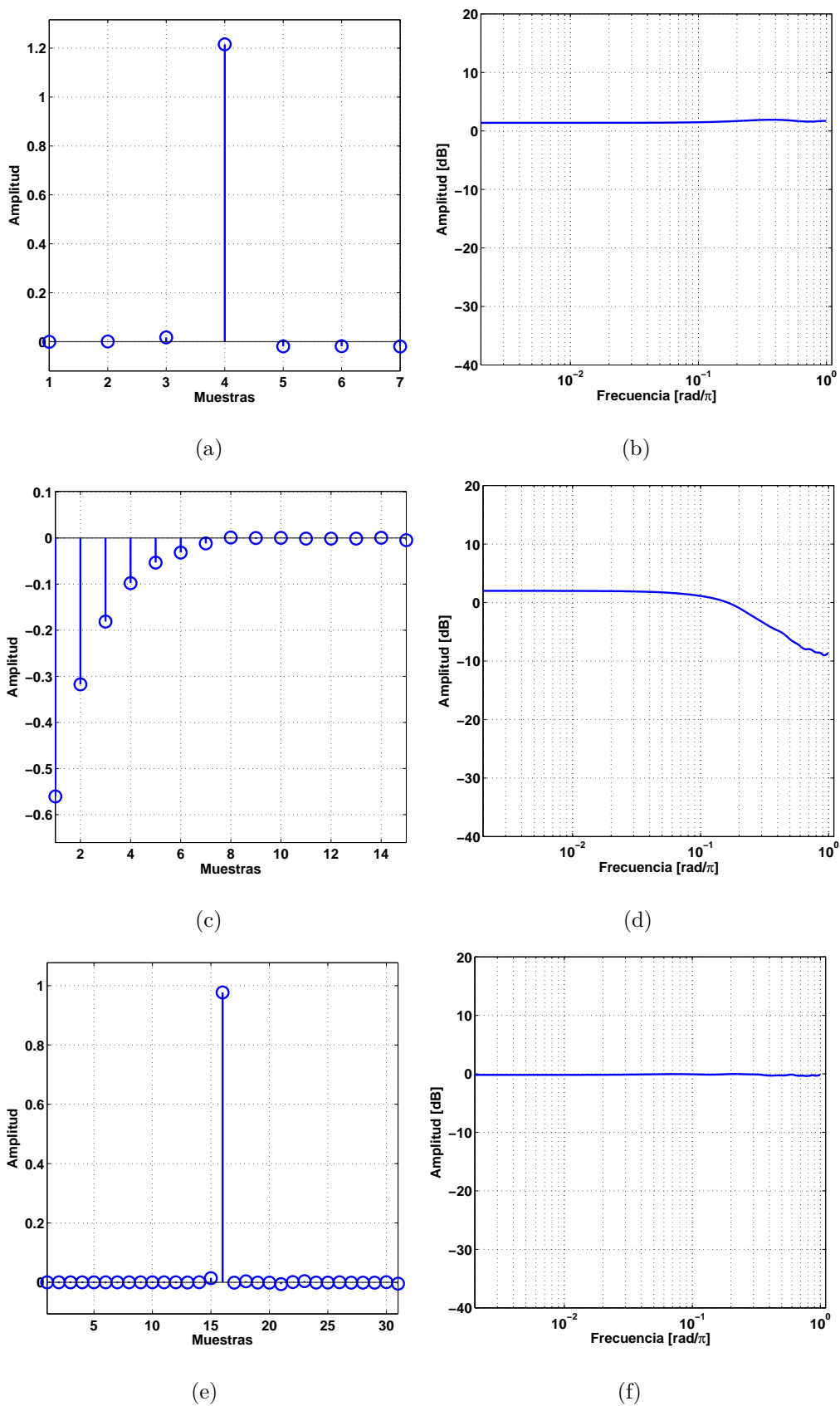


Figura 6.20: Comportamiento de los ecualizadores. (a) y (b) Respuesta del FFE. (c) y (d) Respuesta del DFFE. (e) y (f) Respuesta del canal equivalente en la entrada del detector de umbral.

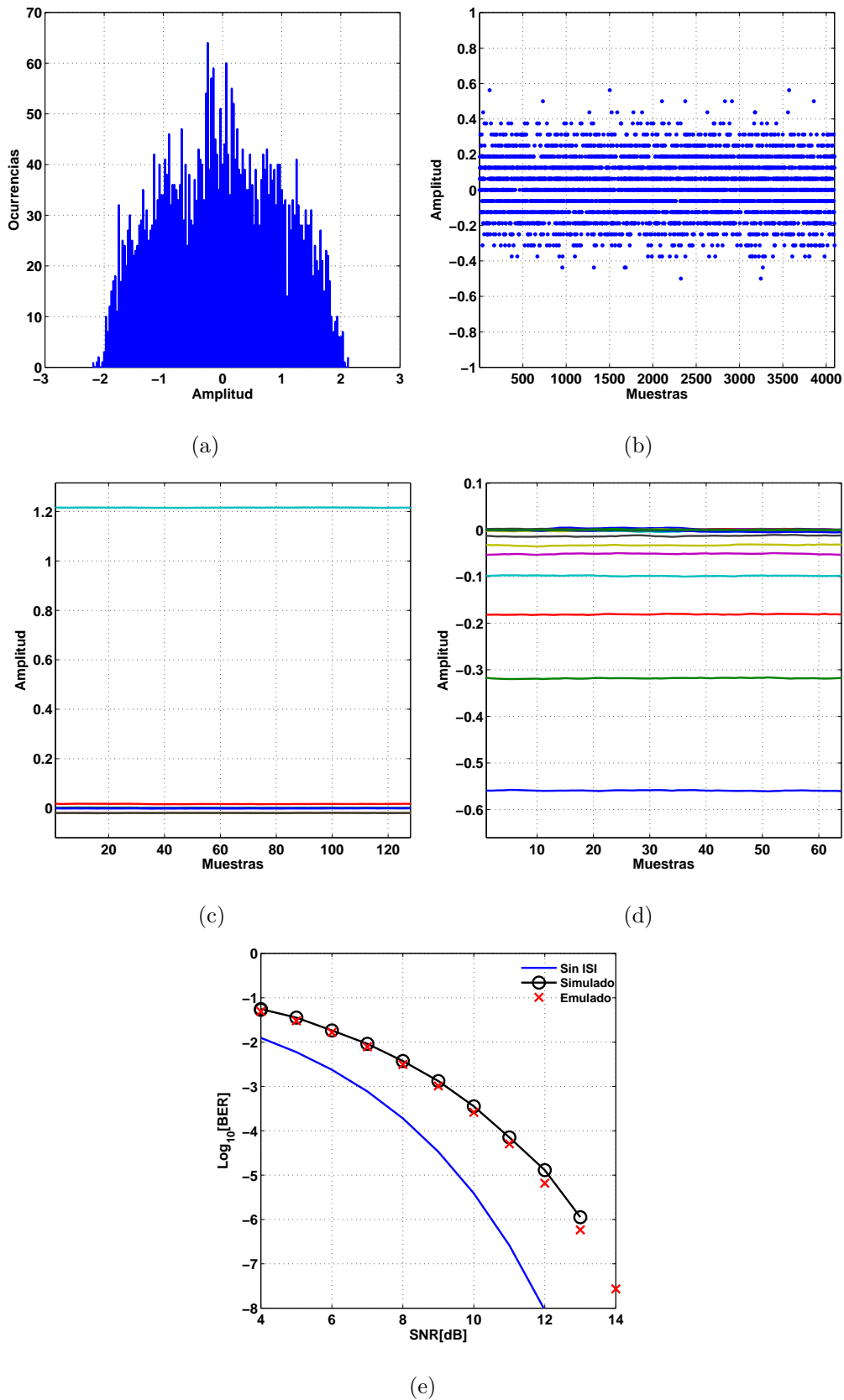


Figura 6.21: Comportamiento de los ecualizadores. (a) Histograma de la señal de entrada del FFE. (b) Error de estimación. (c) y (d) Comportamiento temporal de los coeficientes del FFE y DFFE. (e) Desempeño del FFE-DFFE.

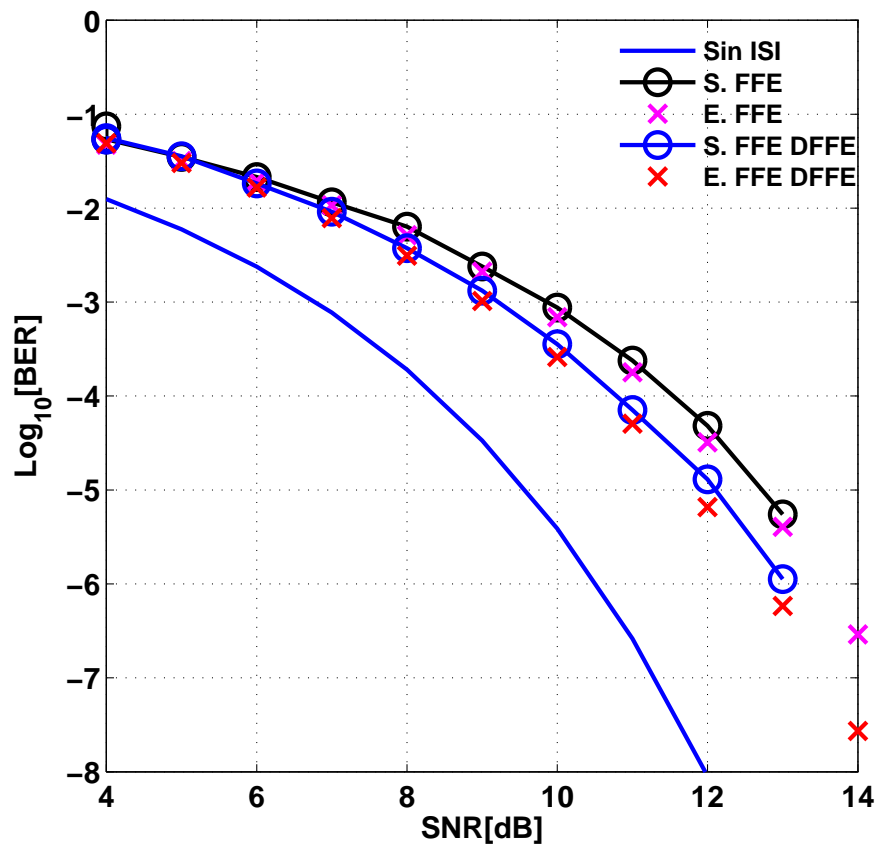


Figura 6.22: Curvas de BER para el canal de la Fig. 6.17. Comparación del desempeño entre el FFE y FFE-DFFE.

## 6.6. Conclusión Parcial

En este capítulo se detalló la implementación en paralelo del DFFE en una FPGA. El sistema completo consta de un generador de PRBS9, un canal que se representa por medio de un FIR y generadores de ruido Gaussiano y un contador BER. Estos módulos son utilizados para la verificación funcional de los ecualizadores FFE y DFFE implementados en paralelo, los cuales emplean el algoritmo de adaptación LMS. En complemento con estos módulos, se utiliza un procesador integrado MicroBlaze para el control y logueo de los módulos antes mencionados. Además, interactúa con la interfaz de usuario por medio del puerto UART. La arquitectura presenta un paralelismo de ocho y una frecuencia de reloj de  $50MHz$  alcanzando una tasa de datos de  $400Mbps$ . Todos estos módulos, los cuales se desarrollaron en el lenguaje de descripción de hardware Verilog, se sintetizaron con la herramienta Vivado 2013.4 para la plataforma KC705. Por otro lado, el procesador opera con un software elaborado en lenguaje de programación C y compilado con la herramienta SDK 2015.4.

Haciendo foco en la implementación, se analizaron los reportes de síntesis que brinda la herramienta Vivado 2013.4 corroborando la complejidad del diseño propuesto así como también las ventajas de utilizar estructuras de implementación directa (*forward*). La importancia de esto radica en la no existencia de problemas de tiempo (*timing*) ni de congestión de encaminamiento (*routing*), siendo estos factores clave en el análisis de arquitecturas de receptores de alta velocidad. Por último, y tomando como referencia el simulador de punto fijo, se realizaron distintos escenarios de evaluación de desempeño para corroborar el funcionamiento de los algoritmos implementados en la FPGA. Las mediciones arrojaron como resultado que ambos sistemas se comportan de manera similar.

Para finalizar con el estudio del DFFE, en el siguiente capítulo se elaboran las conclusiones finales y los trabajos futuros.





# CAPÍTULO

7

## CONCLUSIONES

### 7.1. Discusión Final

La aparición del transistor en 1947 en los Laboratorios BELL, permitió un incremento vertiginoso en los avances de los sistemas electrónicos, logrando el diseño de circuitos integrados extremadamente compactos. Esto abrió las puertas para el desarrollo de circuitos de gran escala de integración jugando un papel fundamental sobre los sistemas con los que convivimos hoy en día. Como consecuencia de estos avances, han surgido innumerables aplicaciones generando que los ingenieros y científicos hayan podido probar su capacidad de invención produciendo nuevas áreas de desarrollo.

En los últimos años, la industria de las telecomunicaciones ha visto evolucionar el concepto de procesamiento en la nube, de una tecnología emergente a una solución de red fuertemente establecida que está ganando aceptación y despliegue generalizado. Un claro ejemplo de esto son las empresas y organizaciones gubernamentales que están pasando gran parte de su carga de trabajo crítico a la nube. Para los consumidores, servicios en la nube ofrecen un acceso simple a los contenidos y servicios, en varios dispositivos, entregados en casi cualquier lugar en donde el usuario tenga acceso a la red. Las principales motivaciones cualitativas para la adopción de la nube incluyen una entrega más rápida de servicios y datos, el aumento de rendimiento de las aplicaciones y la mejora en la eficiencia operativa. Bajo estas condiciones, los centros de datos se encuentran sometidos a altas exigencias técnicas que pretenden restricciones de tiempos muy acotados. Tal como se estipula, en un futuro no muy lejano [5], la infraestructura de estos sistemas buscan migrar por completo a fibra óptica. Sin embargo, para satisfacer las demandas actuales, los investigadores han propuesto soluciones de procesamiento de señales que puedan utilizar los medios de transmisión que hoy en día componen un centro de datos complejo. Típicamente podemos encontrar fibra óptica y cable de cobre.

Uno de los fenómenos que se presenta en cualquier medio de transmisión es la ISI [6], siendo el principal obstáculo para la transmisión digital de alta velocidad confiable sobre canales de banda limitada. Para hacer frente a las imperfecciones del canal, se han propuesto diferentes técnicas de ecualización. El método de ecualización más atractivo por su baja complejidad en la implementación es el ecualizador lineal [15, 16]. Por otro lado, las técnicas que mejor rendimiento presentan frente a la ISI son el ecualizador realimentado por decisión [17, 18] y el detector de secuencia de máxima verosimilitud. En aplicaciones

comerciales, los receptores que utilizan estas técnicas digitales, se suelen implementar en tecnología CMOS de 28nm [13]. En implementaciones seriales, en donde la frecuencia máxima que puede alcanzar el reloj en sistemas complejos de procesamiento digital está limitado a 1GHz, hace prohibitivo su utilización para altas tasas de datos. Es por ello, que es indispensable plantear técnicas de procesamiento paralelo que solucionen estas limitaciones. En primer lugar, en el MLSD la complejidad crece exponencialmente con la memoria del canal, independientemente si utilizamos o no técnicas de paralelización. Por otro lado, el lazo de realimentación dentro del DFE determina el límite superior de velocidad alcanzable en la implementación hardware en el caso serial. En otras palabras, el rendimiento del DFE está limitada por la velocidad del lazo de realimentación. Esto nos lleva a buscar técnicas de paralelización del DFE que satisfagan la necesidad de velocidad y baja complejidad. Los métodos que buscan solucionar estas dificultades de implementación se clasificaron en la segmentación y paralelismo, cancelación iterativa y decisiones tentativas detallados en el *Capítulo 2*. Estos métodos continúan teniendo un incremento exponencial de la complejidad dependientes de la memoria del canal, tal como se analizó en el *Capítulo 3*.

En resumen, la problemática en los receptores de alta velocidad que utilizan el DFE como técnica de ecualización para compensar la ISI del canal, está asociada a que

- No hay arquitecturas de implementación estándar del DFE que posean una complejidad reducida en canales muy dispersivos. Las soluciones propuestas tienen un incremento exponencial de la complejidad en relación con la memoria del canal.
- Las técnicas de paralelización del DFE no son totalmente directas (*forward*), lo que agrega un factor de dificultad adicional al diseñador para solucionar los problemas de tiempo (*timing*).

Es por ello que esta Tesis propone una técnica de ecualización orientada a resolver todas estas dificultades, denomina ecualizador directo asistido por decisiones (*Decision FeedForward Equalizer - DFFE*). El *Capítulo 4* desarrolla el análisis teórico del ecualizador propuesto, el cual utiliza decisiones tentativas para compensar la distorsión del canal. Se utilizó el concepto de información mutua para mostrar que el incremento de las iteraciones favorece el desempeño. La ventaja que presenta esta técnica, es que no posee lazo

realimentado, lo cual la convierte en una excelente alternativa para aplicaciones de alta velocidad. Para facilitar la configuración de los parámetros del DFFE, se propuso una metodología de estimación de la probabilidad de error de bits que permite al diseñador estimar las iteraciones óptimas del ecualizador. Así como también, se empleó resultados de simulación para corroborar las estimaciones de desempeño del DFFE, dando como resultado un comportamiento similar al DFE. El escenario utilizado para tal fin fue un canal de cobre, típico en los centros de datos. Como se analizó en el *Capítulo 5*, la ventaja que presenta el DFFE es que la complejidad crece cuadráticamente con la memoria del canal a diferencia de otras alternativas ecualizadores. Acompañando este beneficio es importante resaltar la no interconexión de señales entre los módulos DFFEn, esto disminuye el encañamiento (*routing*) y permite un diseño simple de la arquitectura del receptor. Las estimaciones de complejidad teóricas se compararon con resultados de síntesis para una implementación en ASIC, en donde se verificó que ambas estimaciones son muy similares.

En conjunto con este análisis, y utilizando la misma metodología planteada en el párrafo anterior, se propuso una arquitectura alternativa que combina los beneficios del DFFE con la eficiencia de detección del detector de secuencia de máxima verosimilitud. El concepto radica en implementar esta técnica de detección empleando el algoritmo de Viterbi pero con estados reducidos. Es decir, compensar parcialmente la ISI del canal para reducir la probabilidad de error en la detección de bit en las primeras decisiones tentativas del DFFE. El efecto que esto logra, es reducir el número de iteraciones del DFFE, lo que lleva a una reducción complejidad del receptor, tal como se detalló en el *Capítulo 5*. Todas estas ventajas hacen del DFFE y DFFE-VA dos arquitecturas excelentes para aplicaciones de muy alta velocidad.

Finalmente, y con el objetivo de darle un marco práctico a la propuesta realizada en esta Tesis, en el *Capítulo 6* se detalló la implementación en paralelo del DFFE en FPGA. El entorno de verificación incluye un procesador integrado (*MicroBlaze*) y diferentes módulos que permiten corroborar el desempeño del sistema desarrollado. La arquitectura presenta un paralelismo de ocho y una frecuencia de reloj de  $50MHz$  alcanzando una tasa de datos de  $400Mbps$ . Los reportes de tiempo demuestran una implementación correcta de la arquitectura, ya que no se identifican tiempos de holgura negativos. Desde el punto de vista de los recursos, se aprovecharon los bloques de registro de desplazamiento (*shift register*) como elemento principal para el diseño del DFFE, tal como se observa en las

Tabla 6.4. También es importante resaltar la flexibilidad que brinda la utilización de algoritmos de estructuras directas (*forward*), facilitando el encaminamiento (*routing*) de las señales sin generar problemas de congestiones en los distintos sectores de la FPGA. Dos factores claves en el análisis de arquitecturas de receptores de alta velocidad. A nivel funcional, el desempeño del DFFE paralelo implementado en FPGA fue similar al obtenido en entornos de simulación. De esta forma, se validó los aspectos teóricos y funcionales del DFFE.

Todo lo antes mencionado nos lleva a concluir que el DFFE es una arquitectura versátil para ser implementada en cualquier plataforma (ASIC o FPGA) ya que brinda las ventajas de una estructura directa (*forward*) facilitando el desarrollo al diseñador. Además, el desempeño es similar al DFE, el cual es uno de los algoritmos más utilizados en receptores de alta velocidad. Por lo tanto, el DFFE es una excelente opción para este tipo de sistemas.

## 7.2. Trabajo Futuro

Para continuar con el desarrollo, se abren dos líneas de trabajos que se resumen a continuación

- El tráfico en los centros de datos y los servicios en la nube continuarán creciendo en forma vertiginosa [4], todo a consecuencia del Internet de las cosas (IoT). Para lograr satisfacer esta demanda será fundamental utilizar modulaciones de un orden mayor para incrementar la tasa de información. De este modo, el trabajo futuro se centra en la extrapolación del DFFE a modulaciones de mayor dimensionalidad con el objetivo de corroborar la eficiencia en complejidad y desempeño.
- Por otro lado, la distorsión no lineal en sistemas de comunicación óptica ha atraído una atención significativa en los últimos años [72]. En este entorno, el DFE basado en filtros no lineales ha tomado una importante participación [73]. El filtro de Volterra es una popular herramienta de procesamiento de señales no lineales y, a menudo se utiliza para la compensación de la distorsión no lineal [72]. Por lo tanto, una solución que ha llamado la atención de los investigadores es la combinación de ambas arquitecturas [74]. Es por ello, que otra línea de trabajo es evaluar el desempeño y complejidad del DFFE en este tipo de receptores.



# BIBLIOGRAFÍA

- [1] G. Moore, “Cramming more components onto integrated circuits,” *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998. 00439.
- [2] S. A. Khan, *Digital Design of Signal Processing Systems: A Practical Approach*. Chichester, West Sussex, U.K. ; Hoboken, N.J: John Wiley & Sons, edición: 1 ed., Feb. 2011.
- [3] G. D. Hachtel and F. Somenzi, *Logic Synthesis and Verification Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1st ed., 2000.
- [4] “Cisco Global Cloud Index: Forecast and Methodology, 2014–2019 White Paper - Cisco.” 00000.
- [5] OIF, “Oif next generation interconnect framework,” *OIF*, vol. OIF-FD-Client-400G-1T-01.0, Apr. 2013.
- [6] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital Communication*. KAP, third ed., 2004.
- [7] J. Liu and X. Lin, “Equalization in high-speed communication systems,” *IEEE Circuits and Systems Magazine*, vol. 4, no. 2, pp. 4–17, 2004. 00098.

- [8] Ali Enteshari and Mohsen Kavehrad, “40-100 gbps transmission over copper, myth and realities,” *DesignCom 2009*, 2009. 00000.
- [9] J. Proakis and M. Salehi, *Digital Communications*. Boston: McGraw-Hill Higher Education, edición: 5 ed., Nov. 2007. 00002.
- [10] V. Belevitch, “Summary of the history of circuit theory,” *Proceedings of the IRE*, vol. 50, pp. 848–855, May 1962. 00058.
- [11] O. Heaviside, *Electromagnetic theory*. London, "The Electrician"Pub. Co., 1893. 00000.
- [12] O. J. Zobel, *BSTJ 7: 3. July 1928: Distortion Correction in Electrical Circuits with Constant Resistance Recurrent Networks. (Zobel, Otto J.)*. Bell System Technical Journal, July 1928. 00000.
- [13] O. Agazzi, M. Hueda, D. Crivelli, H. Carrer, A. Nazemi, G. Luna, F. Ramos, R. Lopez, C. Grace, B. Kobeissy, C. Abidin, M. Kazemi, M. Kargar, C. Marquez, S. Ramprasad, F. Bollo, V. Posse, S. Wang, G. Asmanis, G. Eaton, N. Swenson, T. Lindsay, and P. Voois, “A 90 nm CMOS DSP MLSD transceiver with integrated AFE for electronic dispersion compensation of multimode optical fibers at 10 gb/s,” *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 12, pp. 2939–2957, 2008.
- [14] D. E. Crivelli, H. S. Carrer, and M. R. Hueda, “Adaptive digital equalization in the presence of chromatic dispersion, PMD, and phase noise in coherent fiber optic systems,” *Globecom'04*, Dec. 2004. Paper SP08-3.
- [15] R. W. Lucky, “Automatic equalization for digital communication,” *Bell System Technical Journal*, vol. 44, no. 4, pp. 547–588, 1965.
- [16] R. W. Lucky, “Techniques for adaptive equalization of digital communication systems,” *Bell System Technical Journal*, vol. 45, no. 2, pp. 255–286, 1966.
- [17] M. E. Austin, *Decision-Feedback Equalization for Digital Communication Over Dispersive Channels*. Massachusetts Inst. of Tech Lexington Lincoln Lab., 1 ed., Aug. 1967.



- [18] C. Belfiore and J. Park, J.H., “Decision feedback equalization,” *Proceedings of the IEEE*, vol. 67, pp. 1143–1156, Aug. 1979.
- [19] G. Forney, “Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference,” *IEEE Transactions on Information Theory*, vol. 18, pp. 363–378, May 1972.
- [20] J. K. Omura, “Optimal receiver design for convolutional codes and channels with memory via control theoretical concepts,” *Information Sciences*, vol. 3, pp. 243–266, July 1971. 00038.
- [21] G. Ungerboeck, “Adaptive maximum-likelihood receiver for carrier-modulated data-transmission systems,” *IEEE Transactions on Communications*, vol. 22, pp. 624–636, May 1974.
- [22] S. Kasturia and J. Winters, “Techniques for high-speed implementation of nonlinear cancellation,” *Selected Areas in Communications, IEEE Journal on*, vol. 9, no. 5, pp. 711–717, 1991.
- [23] K. Parhi and D. Messerschmitt, “Pipeline interleaving and parallelism in recursive digital filters. i. pipelining using scattered look-ahead and decomposition,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 7, pp. 1099–1117, 1989.
- [24] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. Wiley-Interscience, Jan. 1999.
- [25] K. Parhi, “Pipelining in algorithms with quantizer loops,” *Circuits and Systems, IEEE Transactions on*, vol. 38, no. 7, pp. 745–754, 1991.
- [26] C. Lin and A. Wu, “Low cost decision feedback equalizer (DFE) design for giga-bit systems,” in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP ’05). IEEE International Conference on*, vol. 3, pp. iii/1001–iii/1004 Vol. 3, 2005.
- [27] C. Lin, A. Wu, and F. Li, “High-Performance VLSI architecture of decision feedback equalizer for gigabit systems,” *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 53, no. 9, pp. 911–915, 2006.

- [28] C. Lin, Y. Lin, S. Jou, and M. Shiou, "Concurrent digital adaptive decision feedback equalizer for 10GBase-LX4 ethernet system," in *IEEE Custom Integrated Circuits Conference, 2007. CICC '07*, pp. 289–292, IEEE, Sept. 2007.
- [29] K. Parhi, "Pipelining of parallel multiplexer loops and decision feedback equalizers," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 5, pp. V–21–4 vol.5, 2004.
- [30] K. Parhi, "Design of multigigabit multiplexer-loop-based decision feedback equalizers," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 489–493, 2005.
- [31] D. Oh and K. Parhi, "Low complexity design of high speed parallel decision feedback equalizers," in *Application-specific Systems, Architectures and Processors, 2006. ASAP '06. International Conference on*, pp. 118–124, 2006.
- [32] Y. Lin, M. Shiue, and S. Jou, "10Gbps decision feedback equalizer with dynamic lookahead decision loop," in *IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009*, pp. 1839–1842, IEEE, May 2009.
- [33] Y. Lin, S. Jou, and M. Shiue, "High throughput concurrent lookahead adaptive decision feedback equaliser," *Circuits, Devices Systems, IET*, vol. 6, pp. 52–62, Jan. 2012.
- [34] J. Andrews, "Interference cancellation for cellular systems: a contemporary overview," *Wireless Communications, IEEE*, vol. 12, no. 2, pp. 19–29, 2005.
- [35] A. Chan and G. Wornell, "A class of block-iterative equalizers for intersymbol interference channels: fixed channel results," *Communications, IEEE Transactions on*, vol. 49, pp. 1966–1976, Nov. 2001.
- [36] A. M. Chan and G. W. Wornell, "A new class of efficient Block-Iterative interference cancellation techniques for digital communication receivers," *The Journal of VLSI Signal Processing*, vol. 30, no. 1, pp. 197–215, 2002.

- [37] Y. Liang, S. Sun, and C. K. Ho, “Block-iterative generalized decision feedback equalizers for large MIMO systems: algorithm design and asymptotic performance analysis,” *Signal Processing, IEEE Transactions on*, vol. 54, pp. 2035–2048, June 2006.
- [38] B. Cardiff, B. Gaffney, and A. Fagan, “Multiple decision feedback equalizers for vector systems with complexity/performance tradeoff,” in *Telecommunications, 2009. ICT '09. International Conference on*, pp. 303–308, May 2009.
- [39] J. Chen, Y. Gu, and K. Parhi, “Novel FEXT cancellation and equalization for high speed ethernet transmission,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, no. 6, pp. 1272–1285, 2009.
- [40] D. George, R. Bowen, and J. Storey, “An adaptive decision feedback equalizer,” *Communication Technology, IEEE Transactions on*, vol. 19, pp. 281–293, June 1971.
- [41] D. Duttweiler, J. Mazo, and D. Messerschmitt, “An upper bound on the error probability in decision-feedback equalization,” *Information Theory, IEEE Transactions on*, vol. 20, pp. 490–497, July 1974.
- [42] J. O’Reilly and A. de Oliveira Duarte, “Error propagation in decision feedback receivers,” *Communications, Radar and Signal Processing, IEE Proceedings F*, vol. 132, pp. 561–566, Dec. 1985.
- [43] S. Altekari and N. Beaulieu, “Upper bounds to the error probability of decision feedback equalization,” *Information Theory, IEEE Transactions on*, vol. 39, pp. 145–156, Jan. 1993.
- [44] J. Smee and N. Beaulieu, “Error-rate evaluation of linear equalization and decision feedback equalization with error propagation,” *Communications, IEEE Transactions on*, vol. 46, pp. 656–665, May 1998.
- [45] A. L. Pola, D. E. Crivelli, J. E. Cousseau, O. E. Agazzi, and M. R. Hueda, “A new low complexity iterative equalization architecture for high-speed receivers on highly dispersive channels: Decision feedforward equalizer (DFFE),” in *2011 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 133–136, IEEE, May 2011.

- [46] A. L. Pola, J. E. Cousseau, O. E. Agazzi, and M. R. Hueda, “A Low-Complexity Decision Feedforward Equalizer Architecture for High-Speed Receivers on Highly Dispersive Channels,” *Journal of Electrical and Computer Engineering*, vol. 2013, May 2013. 00000 Cited by 0000.
- [47] A. L. Pola, J. E. Cousseau, O. E. Agazzi, and M. R. Hueda, “Efficient decision feedforward equalizer with parallelizable architecture,” in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2771–2774, 2013. 00001 Cited by 0000.
- [48] P. Black and T. Meng, “A 1-Gb/s, four-state, sliding block viterbi decoder,” *Solid-State Circuits, IEEE Journal of*, vol. 32, no. 6, pp. 797–805, 1997.
- [49] B. Widrow, *Adaptive filters I: fundamentals*. Systems Theory Laboratory, Stanford Electronics Laboratories, Stanford University, 1966.
- [50] R. Price, “Nonlinearly feedback equalized pam versus capacity for noisy filter channels,” in *Proc. Int. Conf. Communication*, pp. 22–12, 1972.
- [51] J. Salz, “Optimum mean-square decision feedback equalization,” *Bell System Technical Journal*, vol. 52, no. 8, pp. 1341–1373, 1973. 00249 Cited by 0245.
- [52] N. Al-Dhahir and J. M. Cioffi, “MMSE decision-feedback equalizers: finite-length results,” *Information Theory, IEEE Transactions on*, vol. 41, pp. 961–975, July 1995.
- [53] J. Cioffi, G. Dudevoir, M. Vedat Eyuboglu, and J. Forney, G.D., “MMSE decision-feedback equalizers and coding. I. Equalization results,” *IEEE Transactions on Communications*, vol. 43, pp. 2582–2594, Oct. 1995. 00546.
- [54] J. Cioffi, G. Dudevoir, M. Eyuboglu, and G. Forney, “MMSE decision-feedback equalizers and coding. II. Coding results,” *IEEE Transactions on Communications*, vol. 43, pp. 2595–2604, Oct. 1995. 00130.
- [55] S. Dey and S. Chakradhar, “Retiming sequential circuits to enhance testability,” in *12th IEEE VLSI Test Symposium, 1994. Proceedings*, pp. 28–33, Apr. 1994. 00046.

- [56] A. El-Maleh, T. Marchok, J. Rajski, and W. Maly, “Behavior and testability preservation under the retiming transformation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, pp. 528–543, May 1997. 00028.
- [57] L. Lucke and K. Parhi, “Parallel processing architectures for rank order and stack filters,” in , *International Conference on Application-Specific Array Processors, 1993. Proceedings*, pp. 65–76, Oct. 1993. 00048.
- [58] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 99th ed., Aug. 1991.
- [59] “Wiley: Essentials of Error-Control Coding - Jorge Castiñeira Moreira, Patrick Guy Farrell.” 00201.
- [60] D. E. Crivelli, M. R. Hueda, H. S. Carrer, M. Del Barco, R. R. Lopez, P. Gianni, J. Finochietto, N. Swenson, P. Voois, and O. E. Agazzi, “Architecture of a single-chip 50 Gb/s DP-QPSK/BPSK transceiver with electronic dispersion compensation for coherent optical channels,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 4, pp. 1012–1025, 2014. 00007.
- [61] X. UG810, “Kc705 evaluation board for the kintex-7 fpga,” 2015.
- [62] X. UG081, “Microblaze processor reference guide,” 2006.
- [63] X. UG472, “7 series fpgas clocking resources - user guide,” 2015.
- [64] X. UG473, “7 series fpgas memory resources,” 2014.
- [65] J. J. O’Reilly, “Series-parallel generation of m-sequences,” *Radio and Electronic Engineer*, vol. 45, pp. 171–176, Apr. 1975. 00032.
- [66] O. Guangxi Liu, “Gaussian noise generator core specification,” 2015.
- [67] X. UG479, “7 series dsp48e1 slice user guide,” 2014.
- [68] X. UG474, “7 series fpgas configurable logic block,” 2014.
- [69] X. UG835, “Vivado design suite tcl command reference guide,” 2013.

## Bibliografia

---

- [70] X. UG906, “Vivado design suite user guide: Design analysis and closure techniques,” 2012.
- [71] X. UG903, “Vivado design suite user guide: Using constraints,” 2012.
- [72] J. Pan and C. H. Cheng, “Nonlinear Electrical Compensation for the Coherent Optical OFDM System,” *Journal of Lightwave Technology*, vol. 29, pp. 215–221, Jan. 2011. 00029.
- [73] G. Shulkind and M. Nazarathy, “Nonlinear digital back propagation compensator for coherent optical ofdm based on factorizing the volterra series transfer function,” *Opt. Express*, vol. 21, pp. 13145–13161, Jun 2013.
- [74] X. Han and C.-H. Cheng, “Nonlinear filter based decision feedback equalizer for optical communication systems,” *Opt. Express*, vol. 22, pp. 8712–8719, Apr 2014.