

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



**DISEÑO DE BLOQUES PARA UN SISTEMA DE CORRECCIÓN DE
ERRORES POR MEDIO DE TURBOCÓDIGOS APLICADO A
TRANSMISIONES INALÁMBRICAS INFRARROJAS.**

Tesis para optar el Título de Ingeniero Electrónico

Presentado por:

IVÁN ARTURO LÓPEZ ZEGARRA

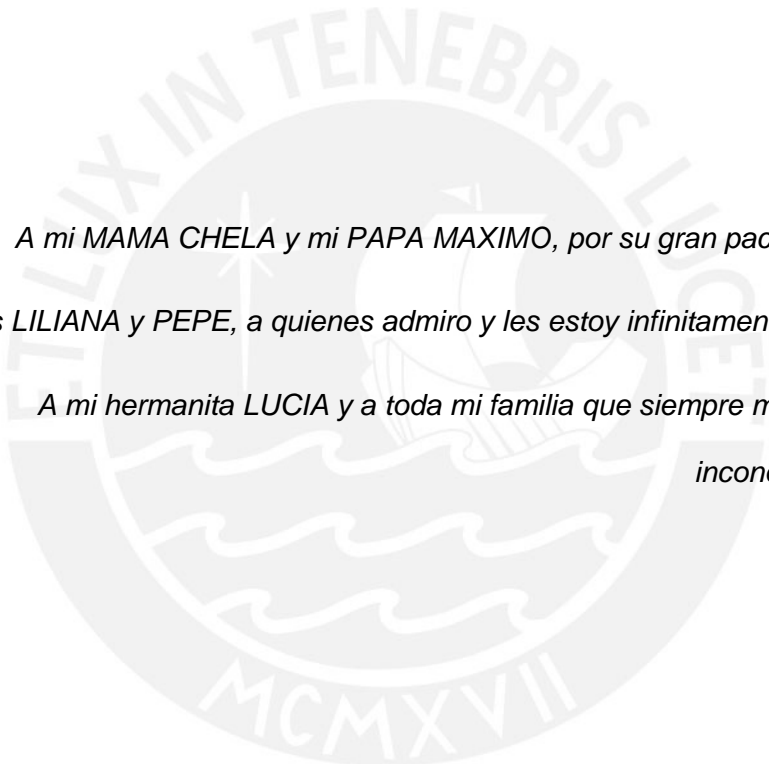
Lima - Perú

2007

RESUMEN

El presente trabajo es un desarrollo analítico sobre los códigos convolucionales de corrección de errores, específicamente enfocado en la codificación y decodificación de los turbo códigos. Al tratarse del primer estudio realizado dentro de la universidad con respecto a este tema, la presente tesis abarca tres aspectos introductorios como son: la aplicabilidad, complejidad matemática y diseño de bloques de un turbo codificador-decodificador. Desde el punto de vista de aplicabilidad se han estudiado los diversos problemas presentados en la transmisión de datos inalámbrica en banda infrarroja con respecto a la presencia de ruido e interferencia del medio por el que se propaga, así como también de que manera la aplicación de la turbo codificación-decodificación en este tipo de comunicación beneficiaría y mejoraría un sistema en términos de confiabilidad y consumo de potencia. Desde el punto de vista algorítmico-matemático se presenta el desarrollo de la base matemática utilizada para la estimación de la información ruidosa como producto de la turbo decodificación. Finalmente desde el punto de vista de diseño, se desarrolla el diseño circuital de los bloques que constituyen cada uno de los cálculos matemáticos que componen tanto el codificador como decodificador.

La presente tesis esta dividida en 5 capítulos. El primer capítulo describe el marco problemático relacionando el ruido presente en los enlaces infrarrojos. El segundo se centra en el marco teórico relacionando la turbo codificación-decodificación con la transmisión de datos en banda infrarroja. El tercer capítulo describe la base matemática necesaria para el diseño de un turbo codificador-decodificador. El cuarto capítulo presenta el diseño de cada uno de los bloques que constituyen tanto el codificador como el decodificador Log-MAP. Finalmente se muestran los resultados y conclusiones seguidos por las recomendaciones y proyecciones que se esperan como consecuencia de la tesis presentada.



A mi MAMA CHELA y mi PAPA MAXIMO, por su gran paciencia y amor.

A mis padres LILIANA y PEPE, a quienes admiro y les estoy infinitamente agradecido.

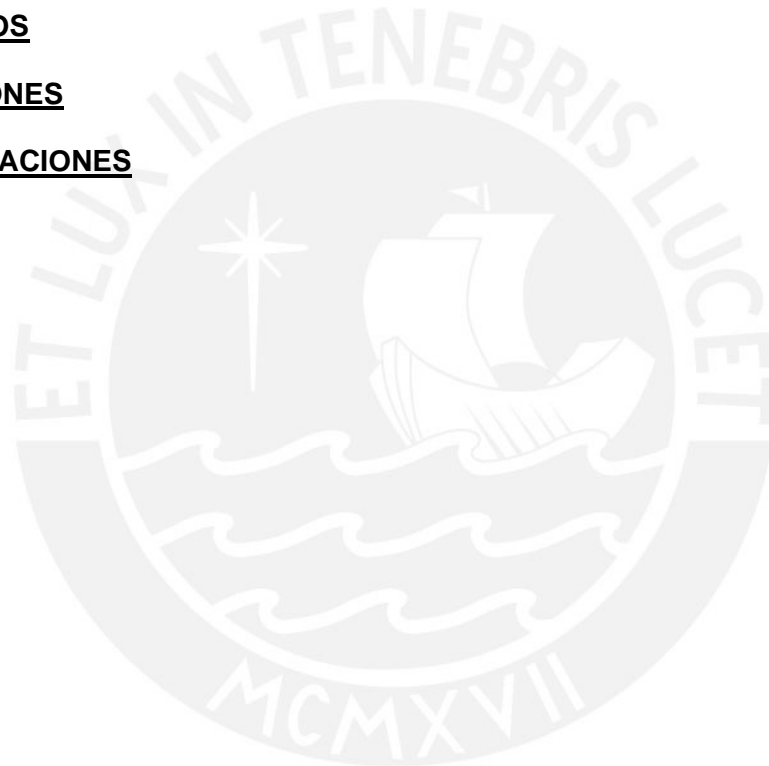
A mi hermanita LUCIA y a toda mi familia que siempre me ha apoyado incondicionalmente.

ÍNDICE

<u>INTRODUCCIÓN</u>	I
<u>CAPÍTULO 1: DECLARACIÓN DEL PROBLEMA</u>	
1.1. Variables medio ambiente general.	1
1.2. Variables medio ambiente específico.	2
1.3. Variables medio ambiente organizacional	3
1.4. Variables internas	4
1.4.1 Número de equipos electrónicos.	
1.4.2 Diversidad de tecnologías inalámbricas.	
1.4.3 Alcances de las transmisiones infrarrojas.	
1.4.4 Efectos del medio ambiente.	5
1.4.5 Técnicas de corrección de errores.	
1.4.6 Algoritmos aplicados a la turbo-decodificación.	
1.5. Declaración del marco problemático.	6
<u>CAPÍTULO 2: MARCO TEÓRICO.</u>	
2.1. Estado del Arte	7
2.1.1. Presentación del asunto de estudio	7
2.1.2. Estado actual de la investigación	9
2.1.2.1 Las comunicaciones inalámbricas	9
2.1.2.2 Las comunicaciones infrarrojas.	14
2.1.2.3 La codificación	18
2.1.2.4 Los Turbocódigos.	19
2.1.3. Síntesis.	21
2.2. Conceptualizaciones Generales	22
2.2.1 En comunicación infrarroja	22
2.2.2 Codificación	23
2.3. Modelo Teórico.	25
<u>CAPÍTULO 3: ALGORITMO DE TURBO DECODIFICACION Y DECODIFICACION.</u>	
3.1. Clasificación de los códigos concatenados.	26
3.2. Códigos convolucionales sistemático recursivos (RSC)	28
3.3. Turbo codificación.	32
3.4. Algoritmo Turbo decodificador.	34
3.4.1 Descripción.	34
3.4.2 Algoritmo MAP.	38
3.4.3 El Algoritmo Log-MAP	46
3.4.4 Codificación basado en el Algoritmo Log-MAP	49
3.5. Implementación del Algoritmo Log-MAP	51

CAPÍTULO 4: DISEÑO SOBRE FPGA DE LOS BLOQUES QUE CONSTITUYEN EL TURBO CODIFICADOR-DECODIFICADOR PROPUESTOS

4.1	Propuesta de turbo decodificador.	59
4.2	Diseño de bloques para el Algoritmo Log-MAP	59
4.2.1	Bloque general para el decodificador Log-MAP.	59
4.2.2	Módulo de cálculo de la métrica de transición (BMT)	61
4.2.3	Módulos de cálculo de las métricas de estado MEP y MER	67
4.2.3.1	Cálculo de la MEP.	67
4.2.3.2	Cálculo para la mínima métrica de estado (min_ME)	73
4.2.3.3	Almacenamiento de las MEPs	74
4.2.3.4	Cálculo de la MER	75
4.2.4	Módulo de cálculo de la Razón Logarítmica de Probabilidad (RLP)	78
4.2.5	Módulo de control para el decodificador log_MAP	81
	<u>RESULTADOS</u>	82
	<u>CONCLUSIONES</u>	85
	<u>RECOMENDACIONES</u>	86
	<u>FUENTES</u>	



INDICE DE FIGURAS

		pag
Figura 3.1	Estructuras para (a) PCCCs y (b)SCCCs	27
Figura 3.2	Ejemplo de codificador convolucional	28
Figura 3.3	Codificador convolucional sistemático recursivo (2,1,2) con razón $\frac{1}{2}$ y matriz generadora $G=[1\ 1\ 1; 1\ 0\ 1]$	31
Figura 3.4	Diagrama de trellis de un código RSC (2,1,2) con razón 1/2	32
Figura 3.5	Ejemplo de un turbo codificador de razón 1/3	33
Figura 3.6	Sistema de codificación y decodificación en un canal ruidoso	36
Figura 3.7	Turbo decodificador iterativo	37
Figura 3.8	Ilustración de las funciones de probabilidad en el trellis	40
Figura 3.9	Representación gráfica de los cálculos de α_k^m y β_k^m .	44
Figura 3.10	Flujo de decodificación del algoritmo log-MAP	52
Figura 3.11	Bloque de cálculo para la métrica de transición	55
Figura 3.12	Calculo de la MEP	55
Figura 3.13	Módulo de Operación-E en el bloque de cálculo de la MEP	56
Figura 3.14	Calculo de la MER	57
Figura 3.15	Diagrama de bloques para el cálculo de la RLP	58
Figura 4.1	Diagrama de bloques del decodificador log-MAP	60
Figura 4.2	Transiciones de estado para el tiempo k-1 hacia el tiempo k.	62
Figura 4.3	Diagrama de bloques del la unidad de cálculo BMEP	65
Figura 4.4	Trellis parcial para el calculo de una MEP	68
Figura 4.5	TDB de ajuste para la función f(z)	69
Figura 4.6	Circuito serial-paralelo para el ingreso de MEPs leídas de la RAM	70
Figura 4.7	Diagrama de tiempos para el calculo de la MEP	71
Figura 4.8	Sumador previo para determinar la MC	72
Figura 4.9	Circuito de Operación-E	73
Figura 4.10	Bloque de cálculo para determinar la mínima MEP por estación	74
Figura 4.11	Almacenamiento en RAMs para las MEPs	74
Figura 4.12	Trellis parcial para el calculo de MER_k^m	75
Figura 4.13	Operación de lectura de las MT RAMs para el cálculo de la MER	77
Figura 4.14	Módulo de Cálculo para la RLP	79
Figura 4.15	Diagrama de tiempos para el cálculo de las RLPs	80

ANEXOS

- ANEXO A : Jerarquías del codificador Log-MAP.
- ANEXO B : Entradas y Salidas de Intercaladotes.
- ANEXO C.1 : Diagrama Esquemático del Decodificador Log-MAP (DEC).
- ANEXO C.2 : Diagrama Esquemático del Módulo de Calculo para la Métrica de Transición (MT)
- ANEXO C.3 : Diagrama Esquemático del Módulo de Memoria para la Métrica de Transición (MT_RAM)
- ANEXO C.4 : Diagrama Esquemático del bloque general para la Métrica Progresiva (MEP_block)
- ANEXO C.5 : Diagrama Esquemático del Módulo de Cálculo para la Métrica Progresiva (MEP)
- ANEXO C.6 : Diagrama Esquemático del Módulo de Memoria para la Métrica Progresiva (MEP_RAM)
- ANEXO C.7 : Diagrama Esquemático del Módulo para la mínima Métrica Progresiva (minMEP)
- ANEXO C.8 : Diagrama Esquemático del bloque general para la Métrica Regresiva (MER_block)
- ANEXO C.9 : Diagrama Esquemático del Módulo de Cálculo para la Métrica Regresiva (MER)
- ANEXO C.10 : Diagrama Esquemático del Módulo para la mínima Métrica Progresiva (minMEP)
- ANEXO C.11 : Diagrama Esquemático del Módulo de Cálculo para la Razón Logarítmica de Probabilidad (RLP)
- ANEXO C.12 : Diagrama Esquemático del mínimo valor absoluto para la Relación Logarítmica de Probabilidad (ABS_min)
- ANEXO C.13 : Diagrama Esquemático del Intercalador
- ANEXO C.14 : Diagrama Esquemático del generador de código sistemático recursivo para el codificador (RSC)
- ANEXO C.15 : Diagrama Esquemático del Turbo codificador propuesto
- ANEXO D.1 : Diagrama de tiempos de las computaciones de la MT y la MEP
- ANEXO D.2 : Diagrama de tiempos de la transición de la MEP a la MER y RLP
- ANEXO D.3 : Diagrama de tiempos de las inicialización y señales de control

INTRODUCCIÓN

En las últimas décadas, el avance tecnológico en el área de las telecomunicaciones tiene un crecimiento vertiginoso y sostenido, gracias a los campos afines que la complementan cada vez con mayor fuerza, como la microelectrónica y las matemáticas. Ambas proporcionan una serie de ventajas, por un lado la miniaturización de circuitos digitales y analógicos mientras que por otro lado el estudio y desarrollo de nuevos algoritmos matemáticos permiten facilitar el diseño de dichos circuitos con menores recursos computacionales. El Perú está constantemente en la búsqueda de acortar la brecha tecnológica que nos llevan los países industrializados, en los cuales los campos mencionados son muy desarrollados y cuentan con fuertes sumas de dinero, con inversiones por parte del estado y las empresas privadas, abocados tan solo a la investigación. Siendo los productos finales equipos de comunicaciones con altos niveles de confiabilidad y robustez que permiten en la actualidad, una comunicación mas clara y fluida sin importar la zona geográfica ni las condiciones climáticas en las que una persona se encuentre.

Los sistemas de comunicación celular de tercera generación (3G) se caracterizan por proveer una gran variedad de servicios incluyendo comunicación multimedia. Esto requiere una transmisión de datos digitales con una baja tasa de error de bits. Sin embargo, debido a que toda batería tiene un tiempo de vida definido en los equipos inalámbricos, la potencia de transmisión debe ser la menor posible. Las bajas potencias hacen que el sistema sea más susceptible al ruido y a la interferencia. La codificación para control de errores es utilizado para incrementar la inmunidad al ruido

de un sistema de transmisión determinado. Como resultado, la potencia de transmisión puede ser disminuida para obtener la misma tasa de errores de bits comparado con un sistema que no utilice un esquema de codificación para la corrección de errores.

Dentro de las distintas bandas espectrales utilizadas para la transmisión inalámbrica, la banda infrarroja está recobrando nueva fuerza frente al auge de la Radio frecuencia, en la cual se han implementado la mayor cantidad de códigos de corrección de errores, desde aplicaciones para bluetooth hasta comunicaciones satelitales inter-espaciales. Es por esta razón que distintos sistemas de comunicación infrarroja vienen siendo mejorados debido a las grandes desventajas que tiene con respecto a su inmunidad al ruido producto del ambiente y la luz solar o de interiores.

De entre todos los sistemas de corrección de errores, los turbo códigos desde 1993 están revolucionando los distintos sistemas de transmisión, disminuyendo las tasas de errores de bits y los consumos de potencia requeridos para transmisiones a mayores distancias. Implementaciones se están realizando aplicadas en su mayoría a las comunicaciones celulares de tercera generación, sin embargo la propuesta de la presente tesis se basa en el estudio de la aplicabilidad de los turbo códigos en transmisiones infrarrojas, el estudio de los distintos algoritmos para la implementación de los turbo decodificadores y finalmente el diseño de los bloques constitutivos de dichos turbo decodificadores para posteriormente ser utilizados libres de licencias con motivos de estudios futuros, mejoras e implementaciones.

El propósito de esta tesis es marcar el punto de partida en el tema de los turbo códigos en nuestro país, iniciado por Jorge Lagos hace aproximadamente 4 años como miembro del grupo de microelectrónica. Sin el apoyo y guía de él la profundización en este tema no sería posible en nuestra universidad. El grupo de microelectrónica a cargo del Dr. Carlos Silva y el Ing. Cesar Saldaña se ha convertido en el centro de investigación por excelencia en el cual se empieza con la difusión de este novedoso sistema de corrección de errores en todo el Perú.

CAPITULO 1:
DECLARACION DEL PROBLEMA.

1.1 Variables medio ambiente general.

La Sociedad actual se está inclinando más a la adquisición de equipos que permitan el acceso a los diversos servicios en línea, al mismo tiempo la confiabilidad en la transmisión de información de los mismos exige que la tecnología enfocada a la mejora de este aspecto de la comunicación se desarrolle constantemente, siendo ésta una variable importante en el desarrollo de nuevas técnicas de transmisión inalámbricas.

El avance tecnológico en tecnologías de comunicación inalámbrica es una variable que determina que tan copado se encuentra el espectro electromagnético. Por tanto la aparición de un mayor número de usuarios de estos medios de comunicación, implica la asignación de mayores canales para transmisión de datos dentro del espectro electromagnético, por lo cual se buscan medios de transmisión diferentes para comunicaciones inalámbricas.

El medio ambiente se ve afectado en el aspecto de la radiación electromagnética, sin embargo el mayor énfasis está en la salud de las personas que tienen acceso a los servicios de telefonía inalámbricos. Por esta razón el desarrollo de transmisiones inalámbricas alternas es una propuesta para la disminución de estos problemas que podrían afectar la salud.

1.2 Variables medio ambiente específico

La carrera por buscar nuevas tecnología de transmisión y mejorarlas, es una tarea que avanza con rapidez. En la cual los clientes son una variable que limita el tiempo de desarrollo e investigación de las mismas. Estos clientes son las grandes empresas que tienen el capital necesario para la implementación de nuevas tecnologías investigadas en un laboratorio.

La velocidad en el desarrollo de nuevos métodos para mejorar la confiabilidad en las transmisiones inalámbricas, esta relacionado íntimamente con el trabajo en los laboratorios. Estos compiten de una manera profesional en la búsqueda de nuevas tecnologías para ser implementadas de acuerdo a las necesidades de los usuarios. Por lo cual la calidad de estos centros de investigación es una variable fundamental en el desarrollo de nuevas tecnologías para la comunicación inalámbrica.

Los productos finales que cuentan con las nuevas técnicas de control de errores y de codificación incorporadas, encuentran una aplicación en las áreas de la industria en las cuales la confiabilidad de la información es de vital importancia, y la presencia de ruido ambiental es un factor importante a corregir. Las oficinas y ambientes en los cuales la presencia de equipos electrónicos es elevado, y a su vez la necesidad de interconexión entre ellos es necesaria, se convierten en una variable a tomar en cuenta para el desarrollo de nuevos algoritmos de corrección de errores.

La normatividad como variable, estaría centrada en la estandarización de las diversas tecnologías aplicadas a la mejora de la confiabilidad de sistemas de transmisión inalámbrica.

1.3 Variables medio ambiente organizacional.

La Tecnología en si misma, constituye una variable importante en el desarrollo de nuevos sistemas para asegurar la confiabilidad de las transmisiones inalámbricas. De no contar con los medios necesarios, para la implementación de nuevos algoritmos de codificación y corrección de errores, se estaría limitando la capacidad de desarrollo de estas aplicaciones. De tal manera que la tecnología como variable organizacional, es la encargada de proporcionar los medios necesarios para hacer posible el desarrollo físico de equipos, que son producto de la investigación matemática y teórica de nuevos algoritmos de codificación y decodificación.

El marketing como variable organizacional, se enfoca desde el punto de vista de la investigación. El desarrollo de nuevas metodologías para combatir ruido en las transmisiones inalámbricas, es producto del constante intercambio científico. Este no sería posible si no existiesen congresos internacionales, en los cuales los últimos avances algorítmicos, matemáticos y físicos son expuestos para crear corrientes de desarrollo a mayor escala. En consecuencia este marketing como variable dependerá de la facilidad de divulgación de las nuevas metodologías a través de los congresos internacionales a nivel científico.

La variable organizacional de la cual dependen las demás, es la calidad de Recursos Humanos con el que se cuenta para el desarrollo de nuevas tecnologías. Mientras no exista una comunidad científica, con un conocimiento profundo en este caso de técnicas de corrección de errores, no se tendría un punto de partida para la creación, mejora e implementación de los mismos para hacer frente al ruido presente en las comunicaciones inalámbricas en la actualidad.

1.4 Variables internas

1.4.1 Número de equipos electrónicos.

Esto genera un problema en cuanto a la conectividad de los mismos. Mientras más equipos entren al mercado, mayor cantidad de cableado se necesita. Pues más información se transmite. A más equipamientos cabe la posibilidad de una mayor interferencia y mayor velocidad, por lo cual se requieren mejores técnicas de transmisión y por consiguiente mayor confiabilidad de las mismas.

1.4.2 Diversidad de tecnologías inalámbricas.

El problema generado es que la mayoría de las mismas están desarrolladas en radio frecuencia, por lo cual se van limitando los rangos disponibles del espectro y por tanto se encasillan las nuevas tecnologías a altas frecuencias. Esta variable interna está relacionada con la búsqueda de métodos alternativos de transmisión con iguales o mayores ventajas que las presentes en la actualidad.

1.4.3 Alcances de las transmisiones infrarrojas.

Esto no permite que sea ampliamente utilizado en las nuevas tecnologías de conectividad inalámbrica. Sin embargo de acuerdo al uso de mayores potencias de transmisión se pueden obtener resultados considerables. Esta variable determina el entorno en el cual serán utilizados los sistemas inalámbricos, la velocidad y las potencias a utilizar.

1.4.4 Efectos del medio ambiente

La fuente principal de ruido en las transmisiones infrarrojas en cuanto a conexiones inalámbricas de corta distancia, es la luz ambiental. La variable por analogía en cuanto a enlaces de mediana distancia, para conexiones no mayores de 2km, es el clima ambiental.

1.4.5 Técnicas de corrección de errores.

No hay una óptima implantación de estos códigos en sistemas de comunicaciones a cortas y medias distancias y menos en otros medios de transmisión inalámbrica que no sean radiofrecuencia.

1.4.6 Algoritmos aplicados a la turbo-decodificación.

Conforme avanza la tecnología nuevos algoritmos vienen siendo estudiados, por lo cual el problema generado está en el campo computacional de los mismos.

1.5 Declaración del marco problemático:

Dado el creciente número de sistemas inalámbricos, el aumento de información transmitida entre equipos electrónicos, y la mayor velocidad requerida por los mismos, hacen que la confiabilidad en estas transmisiones se haga un factor determinante en el desarrollo de nuevas tecnologías. La comunidad científica internacional está enfocada en mejorar esta confiabilidad sobretodo en sistemas de comunicaciones de larga distancia por radiofrecuencia o microondas. Por esta razón en redes de transmisión a pequeñas y a medianas distancias, no se desarrollan sistemas de corrección de errores tan avanzados como los turbo-códigos, y menos aun aplicadas a comunicaciones infrarrojas en las cuales el ruido luminoso a cortas distancias es análogo al ruido producto del desvanecimiento en largas distancias.

Los estudios recientes en cuanto a la implantación de turbo-códigos en tecnología celular de tercera generación, certifican la aplicación de los mismos como un método novedoso para dar mayor confiabilidad y seguridad en la transmisión de información.

CAPITULO 2:

MARCO TEÓRICO

2.1. Estado del arte:

2.1.1 Presentación del Asunto de Estudio

En los últimos años los equipos móviles digitales tales como Asistentes digitales personales (PDAs), teléfonos móviles, cámaras digitales, y laptops han penetrado al mercado. Todos estos equipos requieren un método de transferencia de datos de corto alcance lo suficientemente potentes y eficientes para comunicarse entre ellos, conectarse con impresoras o acceder a redes de área local (LANs). Con la finalidad de estar inmersos en una red mundial de interconexión, que con el paso de los años se expande exponencialmente, las tecnologías de transmisión de datos de larga distancia también requieren de la misma robustez y confiabilidad. Básicamente, los métodos de comunicación están basados en conexiones cableadas, enlaces por radio o enlaces ópticos. Desde que cada uno de ellos tiene distintas ventajas y desventajas, cada uno encuentra una mejor aplicación dentro de determinados productos.

La transmisión vía cable es un método muy bien establecido; “Universal Serial Bus” (USB) se ha convertido en una interfase ampliamente utilizada, gracias a su beneficios de velocidad, sin embargo está limitado por su dificultad en cuanto a la movilidad. Por tanto su mejor aplicación está en dispositivos fijos y de poca movilidad.

Por otro lado, la movilidad es una gran ventaja de los métodos de comunicación inalámbrica a corta distancia basados en radiofrecuencia tales como el Bluetooth, el

cual ha hecho su aparición recientemente en muchos dispositivos móviles. Las mejores aplicaciones en cuanto al Bluetooth se enmarcan en comunicaciones punto a punto o punto a multipunto a velocidades moderadas, donde la movilidad es un requerimiento importante. Por ejemplo para la transmisión de datos de audio desde su teléfono celular a sus auriculares. Contrariamente al USB y al Bluetooth, la transmisión infrarroja basada en los estándares de la “Infrared Data Association” (IrDA) permite establecer conexiones simples y muy rápidas a través de su característica de “apunta y dispara” (point-and-shoot). Al mismo tiempo que permite una alta tasa de transferencia lo que hace que las transmisiones basadas en IrDA encajen perfectamente en aplicaciones que requieren un alto desempeño en conexiones dedicadas punto a punto. Por ejemplo transferencia de imágenes entre teléfonos celulares y PCs.

Entre los factores mas importantes que afectan a las comunicaciones infrarrojas interiores, están el ruido producido por la iluminación ambiental, la potencia infrarroja irradiada y la distancia entre el transmisor y receptor infrarrojo. Mientras que en las comunicaciones infrarrojas de larga distancia son los efectos del clima atmosférico como neblina, nieve, humo, lluvia, resplandor térmico y tormentas de arena así como la potencia de transmisión y la distancia entre el transmisor y receptor, los factores causantes de la atenuación y el ruido en la señal de luz que contiene la información a través de la atmósfera. . Esto provoca que la relación señal a ruido (SNR) en el receptor pueda variar considerablemente produciendo un aumento en los errores durante la transmisión. En vías de mejorar los sistemas de transmisión infrarroja, esfuerzos importantes se están haciendo en los últimos años con respecto a la implantación de metodologías de corrección de errores, entre ellos los Turbo códigos, usadas en sistemas de radiofrecuencia.

El presente acápite organiza y presenta los avances e investigaciones realizadas, tanto en el campo de las comunicaciones inalámbricas, tecnologías de transmisión infrarroja, estándares utilizados y como los sistemas avanzados de corrección de errores (Turbo códigos) están siendo implantados en los distintos tipos de transmisión de datos.

2.1.2 Estado de la investigación

2.1.2.1 Las Comunicaciones Inalámbricas.

Si bien la era de la comunicación radio móvil tuvo sus inicios en 1897 cuando Marconi desarrollo el primer telégrafo inalámbrico, la modulación en amplitud (AM) no veía la gran gama de usos de la comunicación móvil hasta que las estaciones policiales de los Estados Unidos la adoptaron alrededor de los años de 1920.

La modulación en frecuencia (FM) se convierte en la técnica de modulación *de facto* para las comunicaciones móviles rápidamente, luego de que Edwin Armstrong demostrara su uso en 1935. El advenimiento de la segunda guerra mundial ayudó en la introducción a la era de las comunicaciones móviles debido a que las investigaciones realizadas en la tecnología de Radio frecuencia encontraron durante esos tiempos un mercado de consumo creciente a corto plazo.

Un hito importante en las comunicaciones inalámbricas nace en 1946 con la introducción del teléfono radio móvil. Previamente, los sistemas de comunicación inalámbrica no estaban interconectados a la red pública de telefonía (public switched telephone network PSTN), pero ahora el usuario móvil tiene acceso a los sistemas de línea telefónica comercial. Cada mercado para el sistema de radioteléfono era manejado y servido por medio de una gran torre de radio y un solo transmisor de alta

potencia, y los usuarios estaban limitados a comunicaciones FM *half-duplex*. En los años sesenta (60) se vio por primera vez un sistema público amplio de comunicación inalámbrica con designaciones de espectro para cada usuario, lo que se denominaba como *citizen band (CB) spectrum*. Las nuevas técnicas automáticas de troncalizado y las comunicaciones full duplex fueron también desarrolladas en aquellas épocas, pero estas mejoras pronto se sobresaturaron debido a la gran demanda de las comunicaciones móviles. En 1976 la ciudad de Nueva York, un mercado de alrededor 10 millones de personas, contó con los servicios de tan solo 12 canales radiomóviles. La lista de espera por el servicio era de cuatro mil, sin embargo el sistema móvil de BELL solo pudo acomodar a 543 abonados [1].

En respuesta a la creciente demanda de las comunicaciones inalámbricas, los laboratorios Bell empiezan a desarrollar el concepto de comunicación celular durante los años sesenta. El concepto Celular fue la solución a los problemas de la limitación de frecuencia del espectro y la capacidad de usuario. Este concepto dividió el mercado en una serie de pequeñas celdas, cada una de ellas proporciona servicio a solo una porción pequeña del mercado. Solo una porción del total de canales de frecuencia es asignado a cada estación base dentro de la célula, y las estaciones bases vecinas tenían diferente juego de canales de frecuencia. Las comunicaciones celulares son sistemas limitados por interferencia, por lo tanto esta técnica busca minimizar la interferencia mientras se maximiza la capacidad. Hasta aquí, el concepto “celular” permite el reuso de frecuencia. Los canales de frecuencia pueden ser reusados dentro del mercado, siempre y cuando la interferencia co-canal se encuentre dentro de los límites permitidos [1].

Sin embargo implementaciones prácticas del concepto “celular” no fueron factibles hasta finales de los años setenta. Una firma japonesa, “Nippon Telephone Company”, estableció el primer sistema celular analógico en el año 1979. Este sistema alojaba

30MHz del espectro para un total de 600 canales *full-duplex*. Europa no se encontraba muy por detrás en la implementación de sistemas analógicos, y en 1981 se desarrolló el Sistema Nórdico de Telefonía Móvil (NMT 450), y cuatro años mas tarde el Sistema Celular de Acceso Total Europeo (ETACS) fue introducido. Por otro lado el Sistema Avanzado de Telefonía Móvil (AMPS) fue introducido por Ameritech en Chicago, en 1983, cuando la comisión federal de comunicaciones (FCC) asignó 40MHz del espectro de frecuencia en la banda de los 800MHz. Este sistema contaba con 666 canales duplex, y cada canal ocupaba 60KHz del espectro. A pesar de su éxito, esta primera generación de sistemas analógicos sufrió de incompatibilidad y limitaciones de capacidad. [1].

En el inicio de la década del 90 se observó que la comunicación digital suplantaría a la analógica. Las comunicaciones digitales ofrecen numerosas ventajas prácticas y económicas con respecto a las técnicas analógicas. Mejoras continuas en el diseño de implementaciones de integración a gran escala (VLSI), contribuyeron en facilitar el camino para las comunicaciones digitales. Técnicas avanzadas de acceso múltiple como el Acceso Múltiple por División de Tiempo (TDMA) y el Acceso Múltiple por División de Código (CDMA) podrían solamente ser implementadas usando comunicaciones digitales. Estos sistemas de comunicación digital también soportan grandes capacidades de usuario gracias al empleo de técnicas de codificación avanzadas con las cuales el uso del espectro disponible es mas eficiente. Codificadores digitales de alta velocidad y bajo costo han hecho también que las comunicaciones digitales inalámbricas sean mas factibles. Finalmente, la corrección de errores es solo posible con la comunicación digital. La resistencia adicional a la interferencia y el decreciente requerimiento de potencia, resultado de la ganancia debido a la codificación con corrección de errores, hacen que los sistemas de comunicación digitales sean más atractivos que sus predecesores analógicos.

El primer sistema celular digital fue desplegado en Europa en 1993. El Sistema Global para Comunicaciones Móviles (GSM) fue desarrollado la *Conférence Européenne des Postes et Télécommunications* (CEPT) para cubrir la demanda de mas usuarios y asegurar la compatibilidad fuera de los límites nacionales. Contrariamente a los Estados Unidos, Europa había sufrido previamente los efectos de muchos estándares incompatibles. Alemania y Portugal usaban C-450, mientras que Francia Radiocom. 2000; Italia empleaba el Sistema Telefónico Radio Móvil (RMTS). Debido a la variedad de estándares en Europa, un nuevo espectro fue asignado para el sistema GSM en la banda de los 900MHz. GSM experimentó un enorme éxito en su despliegue debido a que proporcionó un mejor servicio, aumentó la capacidad de conexiones fuera del área local, y la habilidad de transmitir información y datos [2]

Los estándares celulares en Estados Unidos, sin embargo, crecieron y se fortalecieron de una manera sigilosa y diferente. Debido a que AMPS era el único estándar, no había la necesidad de asignar anchos de banda para estándares emergentes. Los nuevos estándares, basados en tecnología digital, eran forzados a ser compatibles con el estándar AMPS existente. Por consiguiente, estos estándares emergentes tenían que operar en modo dual, tanto para su operación analógica y digital. Una segunda generación de sistema analógico es el conocido como "Narrowband AMPS (NAMPS), fue esencialmente equivalente al AMPS excepto por el ancho de banda que usa cada usuario el cual es reducido por un factor de 3 a 10 KHz. Los Sistemas digitales de segunda generación fueron clasificados de acuerdo a las técnicas de múltiple acceso que emplean. Los sistemas digitales que utilizan la tecnología TDMA multiplexan una determinada cantidad de usuarios en la misma frecuencia. Los sistemas digitales que utilizan la tecnología CDMA toman una aproximación diferente, donde cada usuario en el sistema esta determinado por un código el cual es ortogonal con respecto al otro de diferente usuario. Cada señal de usuario es modulada con un código de expansión de ancho de banda antes de ser transmitida. Todos los usuarios en el sistema ocupaban

la misma banda de frecuencia al mismo tiempo. Las señales individuales pueden ser demoduladas en el receptor debido a la ortogonalidad y el bajo cruce correlativo de los códigos de expansión individuales [3].

Las comunicaciones inalámbricas personales toman muchas formas, pero los sistemas celulares claramente han tenido un rápido crecimiento en el mercado de las telecomunicaciones móviles. Para conocer la creciente demanda, nuevos espectros están siendo asignados en la banda de los 1.9 GHz para sistemas de comunicación personal (PCS). Los estándares PCS generalmente caen inmersos dentro de dos categorías: Sistemas de alto grado que soportan usuarios de altas velocidades en pequeñas células, y los sistemas de bajo grado los cuales son diseñados para bajas velocidades, baja potencia, y usuarios de baja complejidad residentes en microcélulas. PCS ofrece a los consumidores mayores servicios que los sistemas celulares tradicionales, en la actualidad las tecnologías empleadas en PCS son muy similares a aquellas existentes en los sistemas de segunda generación. [1] [3].

La búsqueda de una estandarización en las comunicaciones celulares a nivel mundial, da origen a lo que se conoce como la tecnología celular de tercera generación (3G). La cual se caracteriza además por la transferencia de información multimedia y de alta velocidad desde los 144Kbs hasta los 2Mbs. Una de las ventajas adicionales es la Mayor capacidad y eficiencia en el uso del espectro. El IMT-2000 (International Mobile Telecommunications-2000) es un estándar de la UIT (Unión Internacional de Telecomunicaciones) definido por un grupo de recomendaciones de la serie M, F, G y Q. Agrupa a una familia de sistemas con capacidades y servicios 3G. En el año 2002 la 3G entró en operación comercial en Corea y en Japón (NTT-DoCoMo). Las siglas UMTS son la abreviación de Universal Mobile Telecommunications System, y constituye la visión europea de sistemas con capacidades 3G como parte de la familia de estándares IMT-2000. UMTS es la evolución lógica de la comunidad GSM a la

tercera generación, por lo que está siendo mayoritariamente adoptado en la Unión Europea.

La interfaz de red definida para cdma2000 apoya la red de segunda generación de todos los operadores actuales, independientemente de la tecnología (cdmaOne, IS-136 TDMA o GSM). La TIA ha presentado esta norma ante la UIT como parte del proceso IMT- 2000/3G. Operando en modo TDD y/o FDD, cdma2000 ofrece velocidades desde 1,2 kbit/s hasta 2 Mbit/s, y soporte para canales de 1,25–3,75–7,5–11,25 y 15 MHz con una o múltiples portadoras. Los 3.6864 Mcps (cps/Chip determina el grado de ensanchamiento del espectro) de cdma2000 proporciona una capacidad superior al sistema en desarrollos de 10 y 20 MHz. No así cuando se trata de desarrollos de 5 MHz donde los 4.096 Mcps de W-CDMA proporciona un mejor rendimiento. Cdma2000 además agrega una banda de guardia de 640 kHz por lado para protección contra interferencia en canales adyacentes (interferencia co-canal). Cdma2000 opera con sincronismo entre el móvil y la estación base.

2.1.2.2 Las Comunicaciones infrarrojas

Las comunicaciones infrarrojas aparecen alrededor de los años 70 cuando Gfeller [4] presenta un “paper” en modelamiento de canal infrarrojo, y la manera como este puede ser usado a interconexiones dentro de un mismo ambiente. En este trabajo, el medio infrarrojo es presentado como una alternativa que haría frente a la comunicación cableada, pues típicamente los sistemas basados en radiofrecuencia no eran capaces de transmisión de altas tasas de transferencia. Desde los primeros días de las comunicaciones infrarrojas, una variedad de modelos de canal han sido presentados, sin embargo estos no eran ampliamente aceptados como los presentados en radiofrecuencia, en parte por porque se pensaban innecesarios los pequeños dispositivos y los costos asociados a la implementación de dichos sistemas. Nuevas técnicas en lentes y filtros se desarrollaron [5][6] que realzan la comunicación,

así también como desarrollo de enlaces infrarrojos internos con velocidades hasta de 155Mb/sec [7] , y hasta propuestas para utilizar sistemas que alcanzarían 1Gbs en redes ATM. Estas redes de comunicación son utilizadas en el mercado actual sobretodo para los sistemas de comunicaciones a medianas distancias no mayores a los 2 km. Un claro ejemplo en la aplicación de esta tecnología, es aplicado por la compañía “CableFree Solutions”, líder en el mercado en diseño y manufactura de sistemas de comunicación ópticos en espacio libre “Free Space Optical” (FSO). Su principal mercado es el Europeo debido a la gran demanda de este tipo de comunicación óptica.

Pero las comunicaciones infrarrojas han encontrado otras áreas de uso, especialmente en casas y oficinas. Todas las casas tienen televisión a control remoto el cual es controlado por medio de infrarrojo. Además, existe un estándar para las comunicaciones infrarrojas entre dos computadoras. Este estándar es conocido como IrDA .

El IrDA se formó en Junio de 1993 para desarrollar una tecnología de comunicación inalámbrica a corta distancia por medio de infrarrojo, la cual sea Interoperable, de bajo costo y de fácil uso. Estando presentes en la reunión inicial mas de 70 compañías con la finalidad definir una serie de especificaciones para la comunicación de datos vía infrarrojo.

A finales de Setiembre de 1993, IrDA seleccionó una de las tres métodos propuestos para definir la capa física definida por Hewlett-Packard [8]. Los tres métodos asumían la presencia de un UART (*universal asynchronous receiver-transmitter*) el cual sería usado para la transmisión infrarroja.

En noviembre de 1993, IrDA desarrolla un método de “token-passing” en el nivel de acceso al canal. Se derivó del HDLC. (High level Data Link Control) y a la vez operando en un modo de respuesta normal (NRM). Así como los otros métodos este define la forma como la información es empaquetada.

En Junio de 1994 se desarrolla la primera plataforma de especificaciones centrales del IrDA : IrPHY , IrLAP , y IrLMP las cuales están especificadas en [9][10][11]. A su vez conforme las investigaciones progresaban se desarrollaron protocolos de aplicaciones y servicios tales como : IrCOMM [12] el cual provee acceso a la plataforma IrDA por medio de los puertos paralelos y seriales; IrLAN [13] el cual provee acceso a LANs bajo la norma IEEE 802; IrTRAN-P [14] La cual permite el intercambio de imaganes entre cámaras digitales , PCs , impresiones fotográficas, etc. Extendiéndose estos protocolos conforme pasan los años, mejorando en cuanto a velocidad y seguridad de transmisión. Al margen de los estándares y especificaciones adoptados por la IrDA, el diseño de los enlaces infrarrojos va a determinar que tan eficiente es la transmisión de la información por este medio.

En [2] se menciona que los enlaces infrarrojos pueden emplear distintos diseños por lo que están clasificados de acuerdo a dos criterios. El primer criterio es el grado de direccionalidad del transmisor y el receptor. Enlaces Direccionados emplean transmisores y receptores direccionales, los cuales deben estar dispuestos de tal manera que estén unidos por una línea recta de visión. Mientras que los enlaces no-direccionados emplean un ancho ángulo de visión entre el transmisor y el receptor, aliviando la necesidad de precisión en el apuntamiento. Los diseños de enlaces direccionados maximizan la eficiencia en cuanto a la potencia transmitida, minimizando caminos de perdida y recepción de ruido luminoso del ambiente. Por otro lado, los enlaces no-direccionados pueden ser de gran utilidad sobretodo en dispositivos con mayor movilidad en cuanto a su uso, puesto que no requieren extrema precisión en cuando al direccionamiento. Sin embargo se encuentran los enlaces híbridos, los cuales combinan los grados de direccionalidad tanto del transmisor como del receptor.

El segundo criterio de clasificación se relaciona con el hecho que existan elementos que se interpongan en la línea de visión (“LOS” line of sight) entre el transmisor y el receptor. Enlaces LOS cuentan con un solo camino de transmisión, mientras que

enlaces sin LOS tienen la ventaja de que aprovechan el reflejo de alguna superficie adyacente para completar su camino de transmisión que ha sido interrumpido. El diseño con enlaces LOS maximiza la eficiencia de la potencia transmitida mientras que minimizan la distorsión por el efecto de multicaminos (multipath distortion). Diseños con enlaces No-LOS incrementan la robustez del enlace y la facilidad de uso, permitiendo al enlace operar a pesar de las barreras sean estas personas o objetos que se interpongan entre el receptor y el transmisor. La mayor robustez y facilidad de uso están implícitos en los diseños de enlaces no-LOS, los cuales comúnmente se conocen como enlaces difusos (Diffuse link).

De acuerdo a estos enlaces investigados hasta hoy cabe la posibilidad que en mayor o menor grado pueda existir pérdida de información, sin embargo en algunos casos las ventajas de utilizar comunicación infrarroja saltan a la vista, así tenemos que los sistemas de telemetría de corto alcance, como ejemplo presentado en [15], los cuales emplean dispositivos infrarrojos presentan algunas ventajas con respecto a los sistemas que emplean radio frecuencia. El ancho de banda utilizado por los enlaces infrarrojos no está regulado permitiendo esto usar elevadas velocidades de transmisión. Dado que la radiación infrarroja es bloqueada por las paredes o barreras opacas, no se producen interferencias entre los sistemas colocados en diferentes habitaciones o separados por paneles opacos [4].

Estos enlaces deben ser compactos, consumir poca potencia y ser robustos contra el ruido ambiental y la interferencia producida por otros usuarios. Sin embargo, los enlaces infrarrojos deben operar en la presencia de una fuerte radiación ambiental producida por la luz solar, iluminación incandescente y fluorescente. También el rango de operación del enlace [4] queda restringido debido a que la potencia de transmisión debe limitarse para evitar el consumo elevado y para prevenir daños en la vista. Por estos motivos la relación señal a ruido SNR en el receptor puede variar considerablemente con el consiguiente aumento de los errores producidos durante la transmisión.

2.1.2.3 La Codificación

A diferencia de la comunicación analógica, la comunicación digital posee la habilidad de detectar y corregir errores introducidos por el canal. “forward error correction” (FEC) juega un papel importante en el proceso de diseño de sistemas que se enfocan en el balance en el consumo de potencia, ancho de banda y confiabilidad de información transmitida.

Toda la teoría de codificación encuentra sus inicios en Claude Shannon, padre de la teoría de información moderna. El aporte más importante de Shannon incluye el desarrollo del teorema de una codificación de fuente sin ruido, el teorema de la tasa de distorsión y el teorema de codificación de canal. En 1948 Shannon publicó el trabajo el cual es el punto de partida y a la vez establece la máxima tasa teórica de información permisible en un determinado canal. Shannon mostró para una canal con ruido gaussiano blanco aditivo que la probabilidad de error puede ser atenuada al mínimo, proveyendo una tasa de datos menos o igual a la capacidad máxima de canal. Estas pruebas usaron códigos aleatorios generados y decodificación sub-óptima a la típica. Usando este método, Shannon fue capaz de mostrar que las regiones decodificadas por distintos códigos eran traslapadas solo en un número insignificante de lugares. Desafortunadamente, las pruebas de Shannon no nos daban con precisión la manera como se debería implementar la construcción de estos códigos de tal manera que se aproximen al límite máximo de capacidad de canal. Debido a la carencia de estructura, códigos aleatorios son notablemente difíciles de decodificar. Agregando estructuras al código se hace más factible el proceso de decodificación, pero los códigos estructurados son pobres cuando se compara con el límite teórico.

2.1.2.4 Los Turbocódigos.

La Codificación turbo convolucional representa una nueva y poderosa técnica de control de errores, la cual empezó a tener un significativo impacto a finales de los 90s, permitiendo la comunicación lo mas cerca posible a la capacidad de canal. La poderosa capacidad de corrección de errores de los turbo códigos ha sido reconocida y aceptada en casi todos los tipos de canales liderando el incremento de tasas de transmisión y mejorando la calidad de servicio de los sistemas de comunicaciones. En [16] se puede comprobar que los turbocodigos pueden operar a 0.1 dB del limite de Shannon (límite máximo de capacidad de canal), destacando esta técnica de corrección de errores entre las demás técnicas hasta ahora descubiertas.

Hasta el momento se han implementado turbo códigos digitales, sin embargo estos serán reemplazados en un futuro cercano por implementaciones analógicas según Sorin Adrian Barbulescu. Puesto que entre sus principales características se encuentran su bajo consumo de potencia y su alta velocidad de procesamiento, lo que hace ideal esta solución.

Los turbo códigos fueron introducidos en 1993 [17] y el primer MODEM diseñado para aplicaciones comerciales fue probado en 1997 ([18][19]). Desde ese entonces se han realizado numerosas investigaciones acerca de cómo aplicar los turbo códigos en comunicaciones espaciales, comunicaciones móviles ya sean por satélite o celulares, enlaces microondas, o implementadas en arquitecturas como OFDM y CDMA. Los turbo códigos sobrepasan ampliamente los esquemas de codificación conocidas hasta el momento según Federic Hirzel , puesto que la ganancia extra ofrecida por los turbo códigos también puede ser usada para reducir el ancho de banda y la potencia requerida para la transmisión.

Turbo codificación fue propuesta en 1993 por Berrou, Glavieux and Thitimajashima, quienes reportaron excelentes resultados de ganancia en [17], En la propuesta original de Berrou invocó a una versión modificada de la mínima tasa de error de bits clásica,

algoritmo de maximización a posterior (MAP : Maximun aposteriory algorithm) a través del trabajo de Bahl [20] que consistía en una estructura iterativa para la codificación de códigos constitutivos. Desde esa concepción de los turbo códigos un gran numero de trabajos se han venido realizando en esta área, enfocándose por ejemplo en reducir la complejidad del decodificador, como lo sugerían Robertson, Villerbrun and Hoehner [21]. Por otro lado Robertson y Worz sugirieron usar los códigos en unión con un esquema eficiente de modulación para aprovechar el ancho de banda. Un buen numero de autores han investigado en cuanto a la disminución de la complejidad e interpretación de la teoría básica de los turbocodigos, siendo uno de los trabajos de Pyndiah [22] el cual extiende el concepto de los turbocodigos a códigos concatenados en bloques paralelos.+

Jung y Nabhan [23] caracterizan el desempeño del código bajo limites de transmisiones de tramas cortas, las cuales son características en sistemas de habla. En colaboración con Blanz ellos también aplicaron los turbocodigos a sistemas CDMA usando detección mutua y diversidad de antena. Mientras que Barbulescue y Pietrobon en [24] Así como muchos autores mencionados anteriormente enfatizan en la importancia de los diseños en las etapas de mezcla (intercaladores). La mayor recopilación del aspecto teórico se ha encontrado en [25].

Se resalta que las recientes implementaciones están enfocadas mayormente al desarrollo bloques constitutivos de los turbo códigos con la finalidad de obtener mejores resultados en cuanto a velocidad de procesamiento y inmunidad al ruido. Asi como también están implementaciones en su mayoría constituyen simulaciones con fines de ser trasladadas a ASIC (circuitos de Aplicación Especifica) y Dispositivos lógicos programables (CPLD). Sin embargo implementaciones aplicativas se encuentran al 50% en lo que ahora conocemos como la tercera Generación de telefónica celular.

Entres las implementaciones recientes se encuentran: [26], [27] , [28] , [29] y [30].

En Junio del 2004 Altera ha lanzado el Megacore “Turbo Encoder & Decoder” siendo el acceso libre para todos los usuarios interesados en simular el comportamiento y eficiencia de los turbo códigos a diferentes escalas de intercaladores.

El grupo de microelectrónica de la Pontificia Universidad Católica del Perú, presentó dos trabajos relacionados íntimamente con el desarrollo de un turbodecodificador. Estos trabajos se basan en la parte algorítmica y fueron desarrollados haciendo uso de herramientas de diseño en microelectrónica a niveles de transistor. En el trabajo “Diseño de un intercalador programable de 8 bits para la decodificación analógica de Turbo Códigos” se enfatiza en la importancia de este bloque dentro del algoritmo de un turbodecodificador pues se basa en el principio de aleatorización de una señal para ser reutilizada en un bloque convolucional. Mientras que el trabajo “Diseño de un Módulo SISO para la decodificación analógica de Turbo Códigos” es de alguna manera el que enmarca el bloque principal de un turbodecodificador. Ambos trabajos fueron presentados en Marzo del 2005 con motivos del XI IBERCHIP 2005 celebrado en Bahia-Brasil.

2.1.3 Síntesis.

En cuanto al desarrollo tecnológico en las comunicaciones inalámbricas se ha avanzando en gran medida, tanto por medios infrarrojos como por radio frecuencia. Sin embargo la poca acogida del primer medio no significa que es menos importante este método de comunicación inalámbrica. Ambos medios tienen ventajas y desventajas.

Entre las ventajas más saltantes en lo que se refiere a la comunicación infrarroja, se encuentra la libertad del ancho de banda a usar, pues no está limitado como el espectro electromagnético. A esto se le suma la menor complejidad en cuanto a

métodos de modulación y la seguridad interna que este medio puede dar si es que nos restringimos a comunicaciones a cortas distancias, independizando zonas por medio de barreras físicas.

A pesar que el acoplamiento de ruido producto de la luz ambiental dentro del campo de trabajo de una transmisión infrarroja es considerable, y la distancia es un

2.2 Conceptualizaciones generales

2.2.1 En comunicación Infrarroja.

IrDa (Infrared Data Association) Organismo creado en 1993 para normar y regular los distintos protocolos en lo que a transmisión infrarroja se refiere. Se centra mayormente en crear estándares únicos para quienes intenten implementar dispositivos con interfases infrarrojas compatibles entre si.

Los enlaces infrarrojos pueden emplear distintos diseños por lo que están clasificados de acuerdo a dos diseños.

El primer criterio es el grado de direccionalidad del transmisor y receptor. Por tanto pueden ser direccionales (angulo de vista pequeño) o no direccionables (mayor angulo de vista).

El segundo criterio de clasificación se relaciona con el hecho que existan elementos que se interpongan en la línea de visión ("LOS" Line-Of-Sight) entre el transmisor y el receptor.

2.2.2 Codificación.

Uno de los parámetros que miden el desempeño de un sistema de comunicación digital es la tasa de error binaria (BER) para un cierto valor de la relación de energía de bit a densidad espectral de ruido (E_b/N_0). Generalmente, debido al ruido del canal de transmisión, la única manera de proporcionar un valor de BER aceptable es agregando códigos para detección y corrección de errores.

Código Hamming El código Hamming es clasificado como un código de bloque porque tiene como entrada un grupo (bloque) de m bits de datos, a los cuales se les agrega un grupo de k bits de paridad de acuerdo a reglas preestablecidas, dando como salida un grupo de n bits comúnmente llamado palabra código. Además es lineal, y es comúnmente utilizado en aplicaciones donde ocurre un solo error por bloque.

Códigos Cíclicos Son una subclase de los códigos de bloque lineales, los cuales tienen esquemas de decodificación eficientes, es decir con algoritmos relativamente simples. Se dice que un código es cíclico cuando cualquier desplazamiento en lazo cerrado de una palabra-código da como resultado otra palabra-código existente dentro del conjunto empleado para codificar los posibles mensajes

Códigos convolucionales Un código convolucional es visto como una máquina de estados finitos que consta de un registro de corrimiento de M etapas con conexiones preestablecidas a n sumadores módulo 2 y con un multiplexor que serializa la salida de los sumadores. Una secuencia de entrada, correspondiente a un mensaje de L bits produce una secuencia de salida codificada de tamaño $n(L+M)$ bits. De manera que la relación entre la entrada y la salida, conocida como índice de código se obtiene con $r=L/n(L+M)$.

Los códigos convolucionales son ampliamente usados en enlaces de radio de línea de vista y en comunicaciones móviles, tal como telefonía celular.

Turbo códigos Es una clase de código de corrección de errores convolucional, el cual consta de dos etapas las cuales a grandes rasgos se pueden resumir en:

El turbo codificador- El cual se basa en una división de dos caminos de datos, los cuales se envían en un solo canal luego de haber realizado un proceso de mezcla, tanto de los datos originales como de los bits de paridad generados.

El turbo decodificador- El cual se encarga de decodificar la información enviada y por medio de dos etapas de decodificación y “desmezclado”

Estimación del SNR El primer aspecto importante en el cual surge un gran impacto el uso de la turbo codificación es la relación señal a ruido, la cual disminuye debido a su capacidad de disminuir la tasa de error de bits (BER) en gran medida

Retardo Es una característica importante si se busca implementar un turbo codificador-decodificador, pues el retardo del sistema es producto de la complejidad en los algoritmos de codificación y decodificación empleados, los cuales pueden ser optimizados para obtener un menor retardo de procesamiento.

Complejidad de decodificación

No existe una sola manera de implementar un decodificador, pues de acuerdo a los estudios previos realizados se tienen una variedad de algoritmos de decodificación los cuales pueden ser usados, algunos de ellos son: MAP, Log-MAP, Max-Log-MAP, sliding MAP y el algoritmo SOVA.

2.3 Modelo Teórico

El desarrollo de un sistema de corrección de errores debe basarse en una metodología orientada a la mejora de las comunicaciones inalámbricas de corta y mediana distancia, tecnología infrarroja como aplicativo directo de accesible implantación y técnicas microelectrónicas para el diseño del Turbo codificador-decodificador en sí mismo.

Diversas transmisiones inalámbricas están orientadas a mejorar la calidad de comunicación entre dispositivos electrónicos, así como también aumentar la comodidad del usuario producto de la movilidad de los dispositivos que él esté usando. Las comunicaciones infrarrojas surgen como respuesta al problema de la conectividad a cortas distancias sin la necesidad de reportar algún espacio del espectro electromagnético ha ser utilizado. Permitiendo además que en diferentes habitaciones no existan interferencias en caso de redes internas y seguras. Dentro de las desventajas se encuentran la presencia del ruido en esta clase de comunicaciones producto de la luz ambiental en mayor o menor grado, así como también se limita la distancia máxima de transmisión.

De los diversos códigos de corrección de errores, los mas novedosos y actuales están orientados a aplicaciones en telefonía celular. Sin embargo se pueden implantar en sistemas de corto alcance como transmisiones infrarrojas, con el fin de aprovechar sus propiedades en cuanto benefician en mejorar la pérdida de información producto del ruido luminoso.

La microelectrónica como rama de la electrónica proporciona las herramientas necesarias para desarrollar algoritmos tan complejos como los turbocódigos, siendo estos aplicables a más de un tipo de comunicación que requiera de la disminución de ruido en su transmisión.

CAPITULO 3:

ALGORITMO DE TURBO CODIFICACION Y DECODIFICACION

3.1 Clasificación de los códigos concatenados.

Los códigos concatenados pueden ser clasificados tanto como códigos convolucionales concatenados paralelos (PCCCs), en los cuales dos codificadores operan los mismos bits de información, o como códigos convolucionales concatenado seriales (SCCCs), en los cuales la salida de uno de los codificadores es codificada por el otro. El término “turbo código” está frecuentemente asociado con los PCCCs, estos emplean dos o mas codificadores convolucionales sistemático-recursivos (RSC) unidos en forma paralela por medio de intercaladores pseudo aleatorios. La codificación sistemática es preferida para la concatenación paralela puesto que permite el fácil ocultamiento de los bits sistemáticos del segundo codificador. Los codificadores no recursivos tienen la desventaja de presentar una distancia de hamming baja [31], además presentan un peso menor con respecto a los códigos producidos por un codificador recursivo o realimentado, por esta razón para la implementación de un turbo codificador se utiliza los codificadores RSC. A diferencia de los esquemas de codificación convencionales en los cuales se busca incrementar la distancia libre de un código, en los turbo códigos se busca decrementar la multiplicidad de bajo peso hamming del código a la salida de los codificadores [31]. Los intercaladores pseudo aleatorios reducen la probabilidad de que ambos codificadores produzcan secuencias de paridad de bajo peso simultáneamente. Esta técnica permite que los PCCCs tengan un buen desempeño a pesar de su relativa corta distancia de los códigos que lo constituyen.

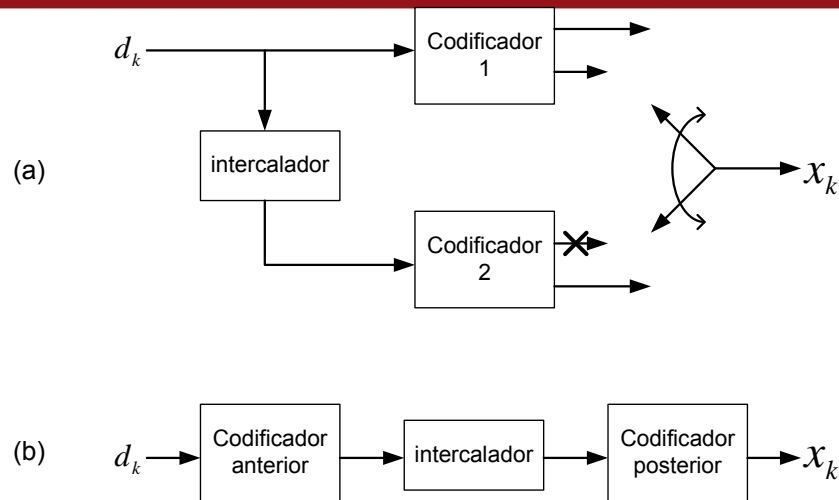


Figura 3.1.- Estructuras para (a) PCCCs y (b) SCCCs

Los SCCCs emplean los mismos componentes constitutivos que los PCCCs; sin embargo los codificadores están conectados serialmente por un intercalador pseudo aleatorio. La figura 3.1 muestra las estructuras de codificadores correspondientes a los PCCCs y SCCCs respectivamente. La característica de los SCCCs con diferencia a los PCCCs radica en que mientras la ganancia del intercalador está alrededor de $N-1$, la curva de BER para los SCCCs continúa decreciendo para $N-2$, $N-3$ etc, donde "N" es el peso de hamming para cada código generado, por tanto no sufre del mismo error que los PCCCs. Sin embargo el codificador primario en un SCCCs no necesita ser recursivo, mientras que el secundario debe ser recursivo de tal manera que se explote la capacidad de ganancia del intercalador [31]. PCCCs son usualmente preferidos en vez de los SCCCs debido a que presentan menor complejidad en el diseño para generar los mismos códigos constitutivos, además tienen menor BER que los SCCCs para bajos SNRs. [31]

3.2 Códigos convolucionales sistemático recursivos (RSC).

En la sección anterior se presentó un aspecto general entre los código concatenados paralelos y seriales, en esta sección se describe los códigos esenciales a ser utilizados en los turbo códigos.

Los códigos convolucionales pueden ser aplicados a secuencias de datos continuas, ofreciendo una alternativa frente a los códigos de bloques para transmisiones sobre un canal ruidoso. Un codificador con un lazo de realimentación genera códigos recursivos, los cuales tienen una respuesta al impulso infinita (IIR), mientras que los codificadores sin realimentación tienen una respuesta finita al impulso (FIR). Un código sistemático es el cual tiene como una de sus salidas a la entrada misma. Un codificador comúnmente utilizado es el binario convolucional no sistemático y no recursivo, como el mostrado en la figura 3.2. Las salidas son calculadas como una combinación lineal de las entradas actuales con un número finito de entradas pasadas.

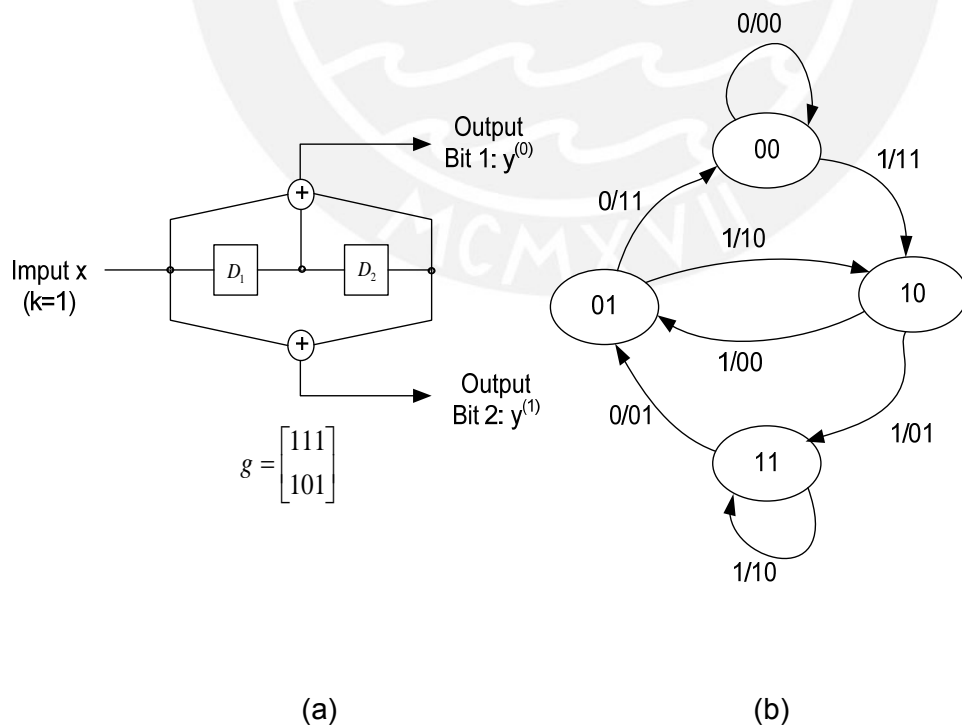


Figura 3.2 Ejemplo de codificador convolucional. (a) Diagrama esquemático, (b) Diagrama de estados.

El codificador de la figura 3.2 tiene dos elementos de memoria (flip-flops D1 y D2) y producen dos bits de información codificada para cada bit de información de entrada, por lo tanto se le conoce como un codificador binario convolucional de rango $\frac{1}{2}$ con $m=2$, donde “m” es el número de registros. Un codificador convolucional está generalmente caracterizado por el formato (n,k,m) , donde .

n = número de salidas del codificador
k= número de entradas del codificador
m= número de registros del codificador.

El rango de un codificador esta dado por la relación k/n el cual es una medida de eficiencia del código, por ejemplo el codificador mostrado en la figura 3.2 es $(2,1,2)$ con rango $\frac{1}{2}$.

Una manera conveniente de expresar las salidas de un codificador convolucional está caracterizada por medio de la matriz generadora G. Por ejemplo la matriz generadora del codificador de la figura 3.2 es:

$$G = \begin{bmatrix} g^{(0)} \\ g^{(1)} \end{bmatrix} = \begin{bmatrix} 111 \\ 101 \end{bmatrix}$$

Los dos elementos de la matriz generadora $g^{(0)}$ y $g^{(1)}$ son también conocidas como funciones de interconexión puesto que representan las conexiones entre las celdas de memoria y la compuertas XOR. El código Y correspondiente a la información de entrada x se obtiene a través de la secuencia de generación:

$$y_i^{(j)} = \sum_{l=0}^m x_{i-l} g_l^{(j)} \text{ mod } 2$$

Donde $Y=(y^{(0)}, y^{(1)})$ y m es el número de registros utilizados.

Un codificador convolucional es una máquina de Mealy, donde la salida es función del estado actual y la entrada actual. Técnicas gráficas, las cuales incluyen los diagramas de estado y los diagramas de trellis, pueden ser utilizados para el análisis de los códigos convolucionales. Una máquina de estado es la que se presenta en la figura 3.2(b) que muestra la transición de los estados y las correspondientes salidas codificadas. Debido a que hay 2 elementos de memoria en el codificador existen por tanto 4 estados en el diagrama de estados. El diagrama de Trellis es una extensión del diagrama de estado que además muestra el paso del tiempo. En esta sección se describe un diagrama de Trellis así como un código recursivo sistemático, ejemplos a ser aplicados en el diseño de un turbocodificador.

Como se mencionó anteriormente los códigos RSC son los comúnmente utilizados en turbo códigos. En la figura 3.3 se muestra un ejemplo de un código RSC (2,1,2) de rango $\frac{1}{2}$. En la figura, “x” representa la secuencia de entrada, y una de las salidas la cual es diferente a “x” es representada por “p”, que denota el bit de paridad. La matriz generadora G para el codificador RSC en la figura 3.3 es:

$$G = \begin{bmatrix} 111 \\ 101 \end{bmatrix} = \begin{bmatrix} g_0^{(1)} g_1^{(1)} g_2^{(1)} \\ g_0^{(2)} g_1^{(2)} g_2^{(2)} \end{bmatrix}$$

Nótese que las matrices generadoras de los dos codificadores en las figuras 3.2 y 3.3 son idénticas. Por tanto las técnicas utilizadas para analizar los códigos sistemáticos no recursivos pueden ser aplicadas a códigos RSC. Con la misma matriz generadora, la estructura del trellis es idéntica para los códigos RSC y los códigos sistemáticos no recursivos excepto por el hecho que algunas entradas y salidas correspondientes a determinadas transiciones del trellis serán diferentes. El diagrama de trellis para el codificador RSC de la figura 3.3 se muestra en la figura 3.4- En el diagrama de trellis, los nodos representan los estados del codificador. Las líneas muestran las posibles

transiciones a través de los estados producto de una determinada entrada. Las ramificaciones en el trellis que están etiquetadas muestran los símbolos generados por las transiciones.

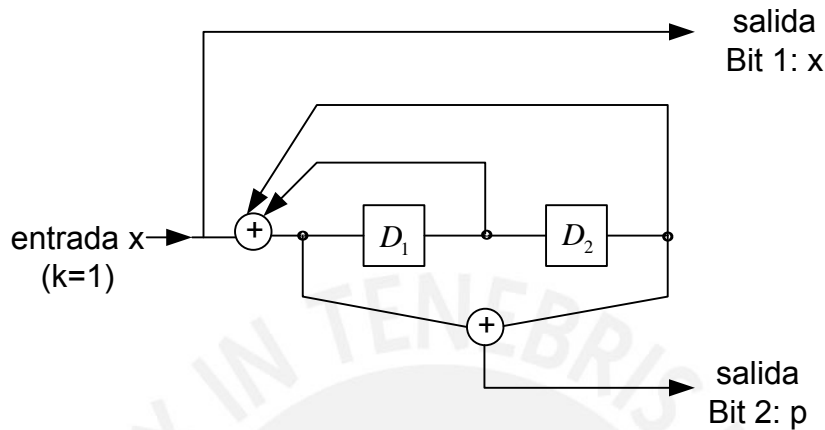


Figura 3.3: Codificador convolucional sistemático recursivo (2,1,2) con

$$\text{razón } \frac{1}{2} \text{ y matriz generadora } G = \begin{bmatrix} 111 \\ 101 \end{bmatrix}$$

El esquema usual es inicializar el codificador en el estado S_0 . Por lo tanto empezando en el estado inicial en S_0 para $t=0$, se empezará a grabar en el trellis todas las transiciones posibles. Después de que todas las entradas de la trama están codificadas, un número de bits de terminación son agregados y codificados para forzar al codificador en retornar al estado S_0 , esta sugerencia es tomada de [31]. Para los codificadores convolucionales no recursivos, una secuencia de '0' puede forzar el estado S_0 . Sin embargo en un código RSC se necesita resolver las ecuaciones de estado para encontrar la manera mas propicia de forzar el estado final S_0 .

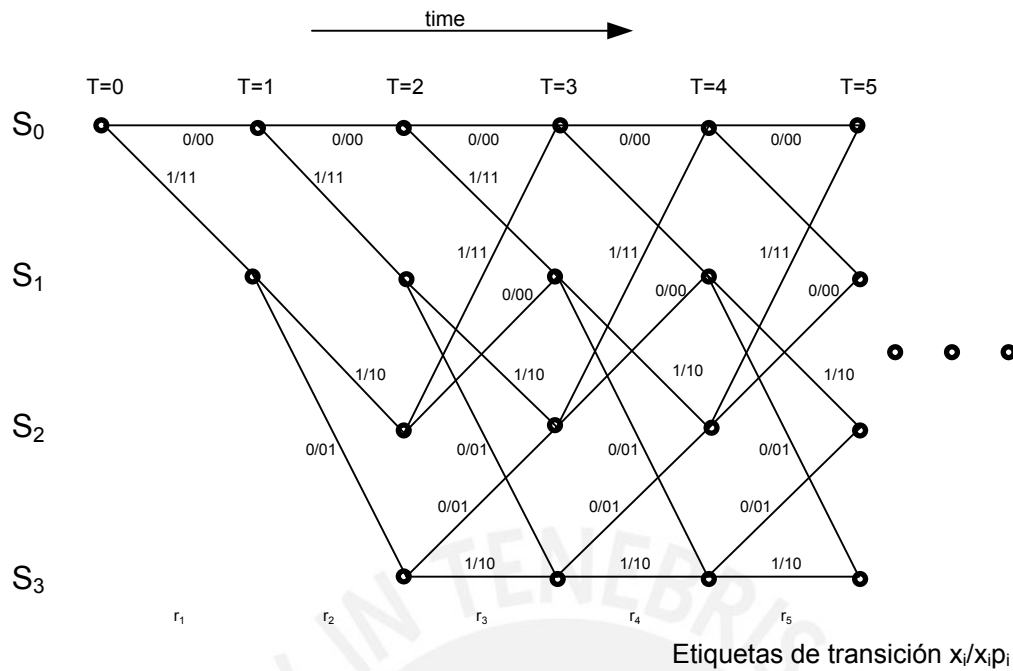


Figura 3.4 diagrama de trellis de un código RSC (2,1,2) con razón $\frac{1}{2}$. Tomado de [3]

3.3 La Turbo codificación

Un turbo codificador típico está compuesto por dos codificadores RSC concatenados en forma paralela que están comunicados por medio de un intercalador como se muestra en la figura 3.5. Cada codificador RSC es un componente del turbo codificador.

Una secuencia “x”, definida como la entrada sistemática, es alimentada a un codificador RSC de rango $\frac{1}{2}$ generando una secuencia de paridad “ p_1 ”, mientras que el segundo codificador RSC también de rango $\frac{1}{2}$ genera con los datos “x” intercalados una secuencia de paridad “ p_2 ”. El bit de información y los dos bits de paridad generados son multiplexados con la finalidad de crear la secuencia de salida “y”, de lo que resulta un turbo código de rango $\frac{1}{3}$.

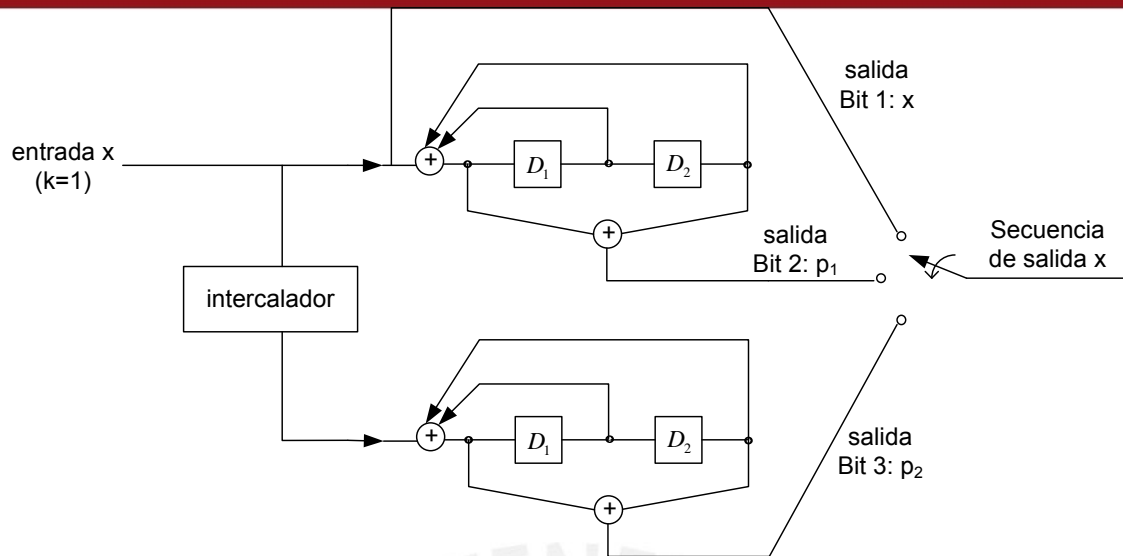


Figura 3.5 Ejemplo de un turbo codificador de razón 1/3. Tomado de [3]

El intercalador es un bloque de una sola entrada y una sola salida que recibe los datos de entrada en la secuencia original y permite que estos sean leídos por el segundo componente de codificación en un orden aleatorio diferente a la secuencia original de entrada. Cuando se reciben las tramas codificadas estas se decodifican en el lado del receptor, al momento de reordenar los datos intercalados se evitan los efectos de algunos patrones de error como son los errores de ráfaga. En [32] se reportó que la construcción y la longitud del intercalador afecta en el desempeño, a mayor longitud del intercalador se mejora el desempeño del código, siendo el diseño de los mismos un tema que se ha analizado en trabajos previos [34]. En términos de diseño de hardware, una memoria RAM puede ser utilizada para almacenar la secuencia intercalada. Un método eficiente de desarrollo en hardware es el uso de un generador pseudo aleatorio haciendo uso de realimentaciones polinomiales. La información es escrita en la memoria en un orden secuencial determinado y leída de la misma en un orden pseudo aleatorio producto del polinomio generador. Este tipo de intercalador será detallado con mayor profundidad en el desarrollo del diseño propuesto.

El intercalador de un turbo codificador presenta problemas al momento de forzar el estado inicial. So debido a los dos codificadores RSC al final de cada trama. Pues los bits de cola solo forzarán al primer codificador al estado So, sin embargo el intercalador presenta una limitante para que el segundo componente del turbo codificador pueda finalizar en So con los mismo bits de cola. Por esta razón, se recomienda en [35] que un Trellis empiece y termine en el estado So solo para el primer componente del turbo codificador. A pesar que el trellis para el segundo componente empiece en So, tiene la misma probabilidad de terminar en cualquier estado al final de la trama.

3.4 Algoritmo Turbo Decodificador.

3.4.1 Descripción.

La Turbo decodificación es una aplicación iterativa para el algoritmo de decodificación convolucional que sucesivamente genera una mejora en la estimación debido a la realimentación de la información transmitida [36]. En esta sección, se hace una introducción a las notaciones a utilizar y una breve descripción del algoritmo para la turbo decodificación. Se detallará con más precisión en la siguiente sección todo lo referente a los algoritmos que son utilizados para la turbo decodificación.

En primer lugar se considera un sistema de comunicación digital binario sobre un canal con ruido blanco aditivo gaussiano (AWGN – additive white gaussian noise) como el que se muestra en la figura 3.6. En esta figura y para la discusión posterior, se tienen las presentes definiciones y expresiones:

- 1.- “x” es la secuencia de bits de información con componentes de un campo finito que denotaré como $GF(2)$.

2.- “y” es la secuencia de símbolos codificados como resultado del turbo codificador. Cada símbolo consta de 3 bits: el bit sistemático x_k , el bit de paridad p_{1k} del codificador 1, el bit de paridad p_{2k} del codificador 2.

3.- “r” es la secuencia de símbolos recibida con $r_k = [x'_k, p'_{1k}, p'_{2k}]$, donde r_k es el símbolo recibido en el tiempo k en la trama. r_k está afectado por el ruido gaussiano el cual será representado por n_k, n_{p1}, n_{p2} las cuales se definen por una media igual a 0 y una varianza σ^2 según [27]. Además, bajo una modulación BPSK con señales antisolares ± 1 , los componentes x'_k, p'_{1k}, p'_{2k} son variables aleatorias con una media $2x_k - 1$, $2p_{1k} - 1$ y $2p_{2k} - 1$, respectivamente, y con varianza σ^2 . Los tres valores x'_k, p'_{1k}, p'_{2k} pueden representarse de la siguiente manera:

$$\begin{aligned} x'_k &= (2x_k - 1) + n_k \\ p'_{1k} &= (2p_{1k} - 1) + n_{p1} \quad \text{donde } x_k, p_{1k}, p_{2k} \in \{0,1\} \\ p'_{2k} &= (2p_{2k} - 1) + n_{p2}, \end{aligned}$$

4.- Como los tres términos $2x_k - 1$, $2p_{1k} - 1$ y $2p_{2k} - 1$ pueden solo tomar dos valores, 1 y -1, con igual probabilidad, por lo tanto la función de densidad de probabilidad gaussiana para x'_k esta representada por:

$$f(x'_k | 2x_k - 1 = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x'_k - 1)^2}{2\sigma^2}\right) \quad (3.1)$$

$$f(x'_k | 2x_k - 1 = -1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x'_k + 1)^2}{2\sigma^2}\right) \quad (3.2)$$

Y de igual manera la misma expresión sería para p_{1k}, p_{2k} .

5.- $r_{a \leq k \leq b}$ representa la secuencia de los símbolos recibidos desde el tiempo =a

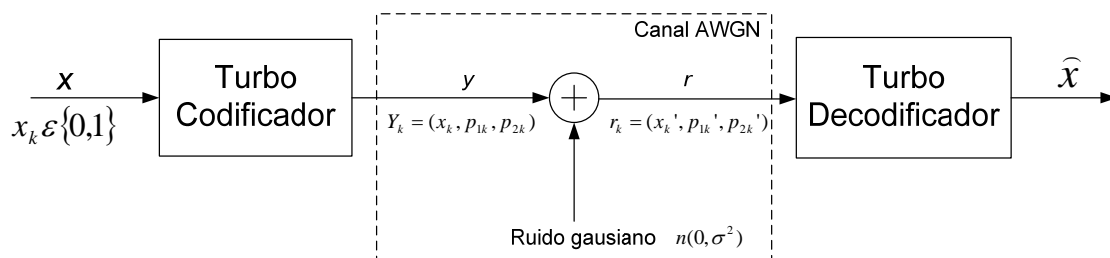
hasta b, esto quiere decir que $r_{a \leq k \leq b} = (r_a, r_{a+1}, r_{a+2}, \dots, r_{b-1}, r_b)$.

6.- \hat{x} es un estimado para la secuencia de información x. La razón logarítmica de probabilidad (RLP) es una herramienta útil para realizar la estimación.

Asumiendo que x_k es el bit del cual queremos determinar el valor. La RLP de la estimación de x_k denotado por $\Lambda_k(\hat{x})$ esta definido como el logaritmo de la razón de la probabilidad *a posteriori* (PAP) de que cada bit de información x_k sea 1 y la PAP de que sea 0. Para ello podemos utilizar un número arbitrario ϵ como la base del logaritmo.

$$\Lambda_k(\hat{x}) \equiv \log_{\epsilon} \frac{\Pr[x_k = 1|r]}{\Pr[x_k = 0|r]}$$

7.-Para una secuencia x, la secuencia intercalada es $\alpha(x)$, y la secuencia de-intercalada de x es $\alpha^{-1}(x)$.



El diagrama de bloques de un turbo decodificador se muestra en la figura 3.7. Para cada componente codificador en el turbo codificador hay un correspondiente componente decodificador en el lado del turbo decodificador. El proceso de decodificación es el siguiente. El componente decodificador DEC1 recibe la entrada sistemática x' la entrada de paridad p_1' y la información a priori desintercalada $\alpha^{-1}(L_e^{(2)})$. La información a priori es producida por el componente decodificador DEC2 y es inicializada en 0 durante la primera iteración. Después del proceso de decodificación, DEC1 genera un RLP denotado por $\Lambda^{(1)}$ y la información extrínseca $L_e^{(1)}$. Luego, DEC2 realiza el proceso de decodificación para las entradas sistemáticas intercaladas $\alpha(x')$, la paridad p_2' , y la información extrínseca a priori intercalada $\alpha(L_e^{(1)})$. Como resultado del proceso de decodificación, DEC2 genera una RLP denotada por $\Lambda^{(2)}$, y la información extrínseca $L_e^{(2)}$ es realimentada al DEC1 después de ser desintercalada, luego DEC1 repite el mismo proceso. Se debe notar que cuando la información extrínseca generada por el decodificador es no correlativa con la información a priori del codificador, el BER puede ser mejorado [37]. Por tanto la calidad de la secuencia estimada \hat{x} mejorará con cada iteración debido a la mejora de la información a priori proveniente del proceso de codificación de la etapa anterior.

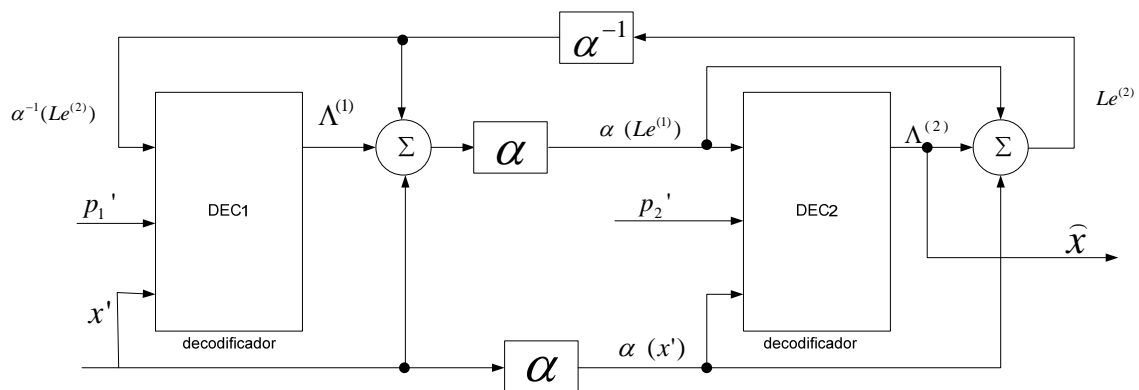


Figura 3.7 Turbo decodificador iterativo.

3.4.2 Algoritmo MAP

El algoritmo de decodificación para un componente decodificador del turbo decodificador debe manejar RLPs tanto en las entradas como en las salidas [32]. Existen 2 algoritmos que se adecuan a este requerimiento, el SOVA (proveniente del inglés: soft-in soft-out Viterbi Algorithm) , y el BCJR (Bahl-Cocke-Jelinek-Raviv Algorithm). El SOVA busca minimizar el error en un código por medio de la maximización de la probabilidad $p(r|\hat{x})$, el cual maximiza el camino más probable estimado para una determinada transición, es conocido en inglés como maximum likelihood algorithm (ML), El algoritmo BCJR busca maximizar la probabilidad a posteriori (PAP) $p(\hat{x}|r)$ de los bits individuales, y es también en inglés conocido como maximum a posteriori (MAP) algorithm [33].

El algoritmo MAP es óptimo para estimar un bit de información de entrada x_i en términos del BER. En esta sección se presentará a manera resumida los conceptos básicos y resultados del algoritmo MAP.

Como se mencionó anteriormente, el RLP del k-ésimo bit de entrada de una secuencia de entrada x esta definida por:

$$\Lambda(\hat{x}) \equiv \log_{\epsilon} \frac{\Pr[x_k = 1|r]}{\Pr[x_k = 0|r]} \quad (3.3)$$

$\Pr[x_k = 1|r]$ es la PAP en la cual el bit de información es igual a '1'. Considerando la transición de estado en la estructura del trellis, se puede expresar $\Pr[x_k = 1|r]$ como:

$$\Pr[x_k = 1|r] = \sum_{(s',s) \in S^+} \Pr(S_{k-1} = s', S_k = s|r) = \sum_{(s',s) \in S^+} \Pr(S_{k-1} = s', S_k = s, r) / \Pr(r) \quad (3.4)$$

Donde S^+ es el conjunto de todos los pares de estados en los cuales la transición del estado s' en el tiempo $k-1$ al estado s en el tiempo k es por medio de $x_k = 1$. Similar,

$$\Pr[x_k = 0|r] = \sum_{(s',s) \in S^-} \Pr(S_{k-1} = s', S_k = s|r) = \sum_{(s',s) \in S^-} \Pr(S_{k-1} = s', S_k = s, r) / \Pr(r) \quad (3.5)$$

Donde S^- es el conjunto de todos los pares de estados en los cuales la transición del estado s' en el tiempo $k-1$ al estado s en el tiempo k es por medio de $x_k = 0$. Por esta razón el RLP del k -ésimo bit de entrada de una secuencia x se obtiene:

$$\Lambda(\hat{x}) \equiv \log_{\varepsilon} \frac{\Pr[x_k = 1|r]}{\Pr[x_k = 0|r]} = \log_{\varepsilon} \left[\frac{\sum_{(s',s) \in S^+} \Pr(S_{k-1} = s', S_k = s, r)}{\sum_{(s',s) \in S^-} \Pr(S_{k-1} = s', S_k = s, r)} \right] \quad (3.6)$$

Si $\Lambda(\hat{x}) > 0$, decodificamos el bit de entrada x_k como 1; caso contrario, el bit de entrada se estima como 0.

$$x_k = \begin{cases} 1 & \text{si } \Lambda(\hat{x}) \geq 0 \\ 0 & \text{si } \Lambda(\hat{x}) \leq 0 \end{cases}$$

Por conveniencia, se denotara " $S_{k-1} = s'$ " y " $S_k = s$ " como S_{k-1} y S_k , respectivamente. Por medio de la regla de Bayes demostrada en [35] se puede particionar $\Pr(S_{k-1} = s', S_k = s, r)$ en 3 partes:

$$\Pr(S_{k-1} = s', S_k = s, r) = \Pr(S_{k-1}, r_{1 \leq j < k}) \Pr(r, S_k | S_{k-1}) \Pr(r_{k < j \leq n} | S_k) \quad (3.7)$$

Por tanto se pueden definir estas tres probabilidades de la siguiente manera:

$$\alpha_{k-1}(S_{k-1}) \equiv \Pr(S_{k-1}, r_{1 < j \leq k}) \tag{3.8}$$

$$\gamma_k(S_{k-1}, S_k) \equiv \Pr(r_k, S_k | S_{k-1}) \tag{3.9}$$

$$\beta_k(S_k) \equiv \Pr(r_{k < j \leq n} | S_k) \tag{3.10}$$

Donde $\alpha_{k-1}(S_{k-1})$ es la función de la información recibida previa a la estación en el tiempo k, $\gamma_k(S_{k-1}, S_k)$ es la función de la información recibida para la estación en el tiempo k, y $\beta_k(S_k)$ es la función de la información recibida después de la estación en el tiempo k. Esto se encuentra ilustrado en la figura 3.8:

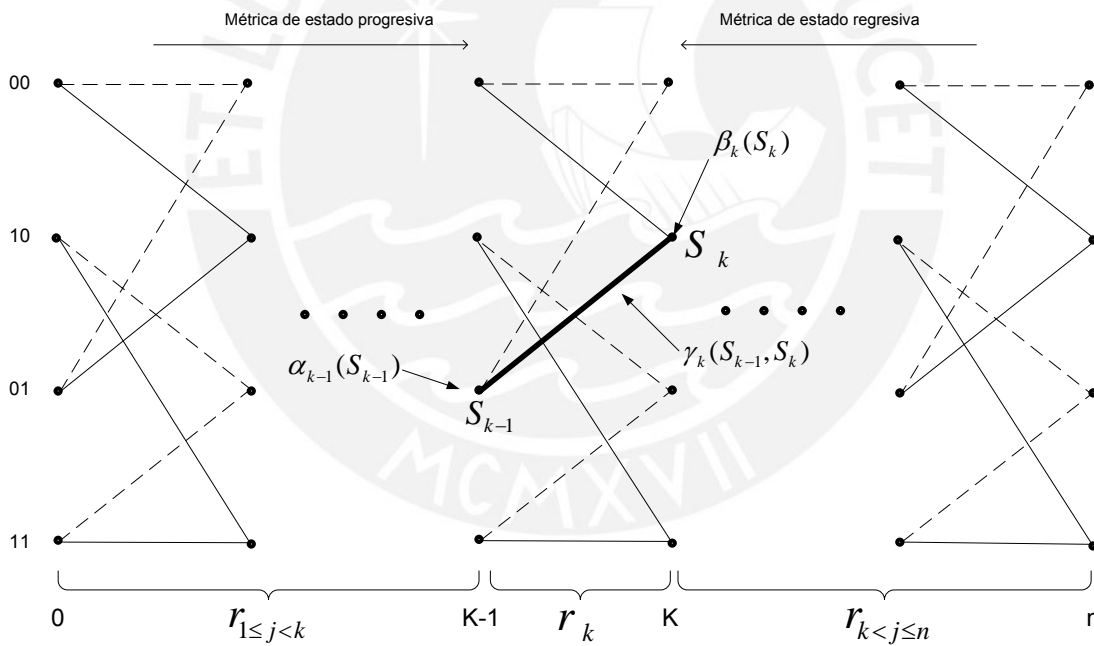


Figura 3.8 Ilustración de las funciones de probabilidad en el Trellis.

Con el fin de maximizar la PAP de bit de información estimado \hat{x}_k , todos los posibles estados del trellis deben ser considerados. Por lo tanto, las funciones de probabilidad α , γ y β deben ser calculadas para cada estado en cada estación [32]. Se usarán tres

parámetros, métrica de estado progresiva (MEP) α_k^m , métrica de transición (MT) $\gamma_k^{i,m}$ y métrica de estado regresiva (MER) β_k^m para caracterizar el nodo en la estación k y estado m en un trellis con bit de información de entrada i siendo '1' o '0'. Estas notaciones serán simplificadas como $\alpha_k(S_k)$, $\gamma_k(S_{k-1}, S_k)$, $\beta_k(S_k)$.

El cálculo de $\gamma_k^{i,m}$ involucra la estimación de la función de densidad de probabilidad gaussiana con una variancia de ruido σ^2 . La función $\gamma_k^{i,m}$ puede ser expresada como:

$$\gamma_k^{i,m} = \Pr(r_k, S_k | S_{k-1}) = p(x_k = i) p(x_k' | x_k) p(p_k' | p_k) \quad (3.11)$$

Con el fin de calcular $p(x_k' | x_k)$ y $p(p_k' | p_k)$, definimos RLPs de las medidas del canal de la señal recibida x_k' como $Lc(x_k')$ y p_k' como $Lc(p_k')$ de la manera siguiente:

$$Lc(x_k') \equiv \log_{\epsilon} \frac{p(x_k' | x_k = 1)}{p(x_k' | x_k = 0)} \quad (3.12)$$

$$Lc(p_k') \equiv \log_{\epsilon} \frac{p(p_k' | p_k = 1)}{p(p_k' | p_k = 0)} \quad (3.13)$$

Como se mencionó anteriormente x_k' y p_k' son variables gaussianas aleatorias con media ± 1 y variancia σ^2 . Refiriéndose a la ecuación (3.1) y haciendo uso de $p(x_k' | x_k = 1) \cong f(x_k' | 2x_k - 1) * \Delta$, donde la aproximación se torna en igualdad en el límite cuando $\Delta \rightarrow 0$.

$Lc(x'_k)$ puede ser expresado como:

$$Lc(x'_k) \equiv \log_{\varepsilon} \frac{p(x'_k | x_k = 1)}{p(x'_k | x_k = 0)} \cong \log_{\varepsilon} \frac{f(x'_k | 2x_k - 1 = 1)\Delta}{f(x'_k | 2x_k - 1 = 1)\Delta}$$

Como $\Delta \rightarrow 0$.

$$Lc(x'_k) = \frac{1}{\ln \varepsilon} \frac{2}{\sigma^2} x'_k \quad (3.14)$$

De igual manera

$$Lc(p'_k) \equiv \log_{\varepsilon} \frac{p(p'_k | p_k = 1)}{p(p'_k | p_k = 0)} = \frac{1}{\ln \varepsilon} \frac{2}{\sigma^2} p'_k \quad (3.15)$$

Se puede observar que las RLPs de las medidas de canal de las señales recibidas

x'_k y p'_k pueden ser obtenidas escalando cada información de entrada por $\frac{1}{\ln \varepsilon} \frac{2}{\sigma^2}$.

Resolviendo las ecuaciones (3.12) y (3.13) y usando las relaciones

$$p(x'_k | x_k = 0) = 1 - p(x'_k | x_k = 1) \quad \text{y} \quad p(x'_k | x_k = 1) = 1 - p(x'_k | x_k = 0) \quad \text{obtenemos}$$

$$p(x'_k | x_k = 0) = \left[\frac{1}{1 + \varepsilon^{Lc(x'_k)}} \right] ; \quad \text{Utilizamos } x_k \text{ para representar un señal binaria. Por}$$

tanto las ecuaciones pueden ser presentadas de la siguiente manera:

$$p(x'_k | x_k) = \left[\frac{\varepsilon^{x_k Lc(x'_k)}}{1 + \varepsilon^{Lc(x'_k)}} \right] \quad (3.16)$$

$$p(p'_k | p_k) = \left[\frac{\varepsilon^{p_k Lc(p'_k)}}{1 + \varepsilon^{Lc(p'_k)}} \right] \quad (3.17)$$

Por otro lado, se utilizará $La(x_k)$ para denotar la RLP a priori del bit de información de la siguiente manera:

$$La(x_k) \equiv \log_{\varepsilon} \frac{p(x_k = 1)}{p(x_k = 0)} \quad (3.18)$$

Resolviendo la ecuación anterior usando la relación $p(x_k = 1) = 1 - p(x_k = 0)$

$$p(x_k = i) = \frac{\varepsilon^{x_k La(x_k)}}{\varepsilon^{La(x_k)} + 1}, \quad i = 0 \text{ o } 1 \quad (3.19)$$

Combinando las ecuaciones (3.16) – (3.19), $\gamma_k^{i,m}$ se convierte en:

$$\gamma_k^{i,m} = p(x_k = i) p(x_k' | x_k) p(p_k' | p_k) = \left[\frac{\varepsilon^{x_k La(x_k)}}{\varepsilon^{La(x_k)} + 1} \right] \left[\frac{\varepsilon^{x_k Lc(x_k')}}{1 + \varepsilon^{Lc(x_k')}} \right] \left[\frac{\varepsilon^{p_k Lc(p_k')}}{1 + \varepsilon^{Lc(p_k')}} \right] \quad (3.20)$$

Donde (x_k, p_k) son los símbolos esperados de una transición específica de un estado $b(i,m)$ en el tiempo $k-1$ al estado m en el tiempo k , donde $b(i,m)$ es el estado previo del estado actual m debido a la correspondiente transición producto de la entrada i .

La MEP α_k^m puede ser calculada por medio de una repetición progresiva a través del trellis con el conocimiento del estado inicial en el tiempo 0 y $\gamma_k(S_{k-1}, S_k)$. Basados en el cálculo de la probabilidad, α_k^m es expresada como:

$$\alpha_k^m = \alpha_{k-1}^{b(0,m)} \gamma_k^{0,m} + \alpha_{k-1}^{b(1,m)} \gamma_k^{1,m} \quad (3.21)$$

cualquier numero, pero 0 es preferible para prevenir sobrecarga. Por esta razón se tiene la siguiente condición de inicialización:

Decoder 1: $\beta_n^0 = 1$ y $\beta_n^m = 0$ para $m \neq 0$;

Decoder 2: $\beta_n^m = 0$ para todo m ;

De las ecuaciones (3.6) y (3.8)-(3.10) obtenemos la siguiente ecuación:

$$\Lambda(\hat{x}_k) \equiv \log_{\varepsilon} \frac{p(x_k = 1|r)}{p(x_k = 0|r)} = \log_{\varepsilon} \left[\frac{\sum_{m,k} \alpha_{k-1}^{b(1,m)} \gamma_k^{1,m} \beta_k^m}{\sum_{m,k} \alpha_{k-1}^{b(0,m)} \gamma_k^{0,m} \beta_k^m} \right] \quad (3.23A)$$

La ecuación (3.23A) puede ser calculada usando las ecuaciones (3.20)-(3.22) derivadas a continuación. Por tanto del trabajo original de turbo decodificación presentado en [15] la RLP puede ser particionada en 3 términos :

$$\Lambda(\hat{x}_k) = Lc(x_k') + La(x_k) + L_{ek} \quad (3.23B)$$

$$Lc(x_k') = \frac{1}{\ln \varepsilon} \frac{2}{\sigma^2} x_k' \quad (3.14)$$

$$La(x_k) \equiv \log_{\varepsilon} \frac{p(x_k = 1)}{p(x_k = 0)} \quad (3.18)$$

En términos de implementación en hardware, las ecuaciones (3.23A) y (3.23B) son interpretadas de la siguiente manera: La salida del decodificador MAP es la RLP $\Lambda(\hat{x}_k)$. Esta salida está constituida por 3 partes: La RLP de la estimación del canal de la señal recibida x_k' , la RLP a priori de x_k y la información extrínseca L_{ek} . En contraste, los dos primeros términos $Lc(x_k')$ y $La(x_k)$ combinadas son conocidas como información intrínseca. El codificador MAP puede ser configurado para recibir la

secuencia de información en la forma de razones de máxima probabilidad y producir un estimado para la información en forma de RLPs también. El signo del bit $\Lambda(\hat{x}_k)$ denota la decisión final, y la magnitud del mismo representa la confiabilidad de esta decisión final.

3.4.3 El Algoritmo Log-MAP

Con el propósito de eliminar los cálculos no-lineales en el algoritmo MAP, se reconfigura de tal manera que toda la información sea expresada en logaritmos negativos. Posteriormente todas las multiplicaciones son convertidas en adiciones. Antes de hacer una revisión al algoritmo log-MAP, introduciremos una operación binaria denominada “Operación-E”. Esta operación está basada en el logaritmo Jacobiano y esta definida como en [31]:

$$E(a, b) \equiv -\log_{\varepsilon} (\varepsilon^{-a} + \varepsilon^{-b}) = \min(a, b) - f(|a - b|); \quad (3.24)$$

$$f(z) = c \ln(1 + e^{-z/c}), \quad z \geq 0 \quad (3.25)$$

$$c = \log_{\varepsilon} e \quad (3.26)$$

Donde la función $f(z)$ es una función de corrección no lineal y ε en teoría puede ser un número arbitrario positivo. Sin embargo, el limitado ancho de bit en hardware requiere una cuidadosa selección de $\log_{\varepsilon} e$. A manera de ejemplo consideremos una función de ajuste $f(z)$. Si una tabla de relación (TR) con 64 entradas y un ancho de 4bits se escoge para implementarla, la entrada z debe ser cuantizada con 6 bits, para un rango de 0 a 63. Como $f(z)$ es una función monótona decreciente, el valor máximo de $f(z)$ es obtenido cuando $z=0$. De esta manera, $f(z)_{\max} = c \ln(1 + e^{-z/c})|_{z=0} = c \ln 2$. Puesto

que la salida de $f(z)$ esta representada por solo 4 bits, $f(z)_{\max} = 15$. La ecuación $\epsilon^{\ln 2} = 15$ conlleva a un $c = 21.64$. finalmente, ϵ es 1.047 de la ecuación (3.26).

La operación-E puede extenderse para manejar múltiples términos. Puede ser expresada:

$$E(a_1, a_2, \dots, a_n) \equiv -\log_{\epsilon} (\epsilon^{-a_1} + \epsilon^{-a_2} + \dots + \epsilon^{-a_n}) \quad (3.27)$$

La operación-E con múltiples términos puede ser desarrollada recursivamente por medio del cálculo de dos términos en cada tiempo como se muestra:

$$\epsilon^{-a_1} + \epsilon^{-a_2} = \epsilon^{\log_{\epsilon}(\epsilon^{-a_1} + \epsilon^{-a_2})} = \epsilon^{-E(a_1, a_2)}$$

Similarmente: $\epsilon^{-a_1} + \epsilon^{-a_2} + \epsilon^{-a_3} = (\epsilon^{-a_1} + \epsilon^{-a_2}) + \epsilon^{-a_3}$

$$= \epsilon^{-E(a_1, a_2)} + \epsilon^{-a_3}$$

$$= \epsilon^{-E(-E(a_1, a_2), a_3)}$$

Y repitiendo el mismo proceso obtenemos,

$$E(a_1, a_2, \dots, a_n) = -\log_{\epsilon} (\epsilon^{-E(-E(-E(a_1, a_2), a_3), \dots, a_n)}) \quad (3.28)$$

De esta manera, la operación-E con múltiples términos puede ser calculada usando la ecuación (3.24) repetitivamente.

Tomando los logaritmos negativos en los tres parámetros, tenemos:

$$\Gamma_k^{i,m} \equiv -\log_{\epsilon} \gamma_k^{i,m} \quad (3.29)$$

$$A_k^m \equiv -\log_{\epsilon} \alpha_k^m \quad (3.30)$$

$$B_k^m \equiv -\log_{\epsilon} \beta_k^m \quad (3.31)$$

$$L_k \equiv -\Lambda(\hat{x}_k) \quad (3.32)$$

El algoritmo MAP se convierte en el algoritmo log-MAP:

$$A_k^m \equiv -\log_\varepsilon \alpha_k^m = -\log_\varepsilon (\alpha_{k-1}^{b(0,m)} \gamma_k^{0,m} + \alpha_{k-1}^{b(1,m)} \gamma_k^{1,m})$$

Calculando :

$$A_k^m = E(\Gamma_k^{0,m} + A_{k-1}^{b(0,m)}, \Gamma_k^{1,m} + A_{k-1}^{b(1,m)}) \quad (3.33)$$

$$= \min(\Gamma_k^{0,m} + A_{k-1}^{b(0,m)}, \Gamma_k^{1,m} + A_{k-1}^{b(1,m)}) - f(z) \quad (3.34)$$

Donde

$$z = \left| (\Gamma_k^{0,m} + A_{k-1}^{b(0,m)}) - (\Gamma_k^{1,m} + A_{k-1}^{b(1,m)}) \right|$$

De igual manera:

$$B_k^m = E(\Gamma_{k+1}^{0,f(0,m)} + B_{k+1}^{f(0,m)}, \Gamma_{k+1}^{1,f(1,m)} + B_{k+1}^{f(1,m)}) \quad (3.35)$$

$$= \min(\Gamma_{k+1}^{0,f(0,m)} + B_{k+1}^{f(0,m)}, \Gamma_{k+1}^{1,f(1,m)} + B_{k+1}^{f(1,m)}) - f(z) \quad (3.36)$$

Donde

$$z = \left| (\Gamma_{k+1}^{0,f(0,m)} + B_{k+1}^{f(0,m)}) - (\Gamma_{k+1}^{1,f(1,m)} + B_{k+1}^{f(1,m)}) \right|$$

La métrica de transición es expresada como:

$$\Gamma_k^{i,m} = -\log_\varepsilon \varepsilon^{x_k Lc(x_k) + p_k Lc(p_k)} - \log_\varepsilon \varepsilon^{x_k La(x_k)} - C_k \quad (3.37)$$

$$= -(x_k Lc(x_k) + p_k Lc(p_k) + x_k La(x_k) - C_k)$$

C_k es un factor común para todas las transiciones en la estación k en el trellis, por tanto puede ser ignorado del calculo de $\Gamma_k^{i,m}$.

Finalmente, la RLP del bit de información estimado es obtenido por:

$$L_k = \underset{m=0}{E} (A_{k-1}^{b(0,m)} + \Gamma_k^{0,m} + B_k^m) - \underset{m=0}{E} (A_{k-1}^{b(1,m)} + \Gamma_k^{1,m} + B_k^m) \quad (3.38)$$

Notar que el operador-E para múltiples términos puede ser aplicado para la ecuación (3.38). mientras todos las multiplicaciones y cálculos de exponenciales son eliminados, para hallar la RLP solo es necesario calcular las adiciones y sustracciones. De esta manera, el algoritmo log-MAP minimiza la complejidad computacional.

3.4.4 Codificación basado en el Algoritmo Log-MAP

En esta sección se explica el algoritmo de turbo decodificación basado en el log-MAP.

Tomando en cuenta la estructura mostrada en la figura 3.7. Se utiliza α para representar el intercalamiento y α^{-1} para el desintercalamiento. El primer decodificador log-MAP acepta tres entradas, la secuencia de bits sistemáticos recibidos x' , la secuencia de paridad p'_1 y la salida $\alpha^{-1}(Le^{(2)})$ proveniente del decodificador DEC2. las dos secuencias x' y p'_1 son convertidas a RLPs de las medidas de canal de las señales recibidas $Lc(x'_k)$ y $Lc(p'_k)$ por medio del escalamiento con el factor $\frac{1}{\ln \varepsilon} \frac{2}{\sigma^2}$.

Con estas entradas el algoritmo log-MAP es desarrollado en el decodificador DEC1. El decodificador log-MAP genera las RLP $\Lambda(x'_k)$ de la secuencia estimada, estos valores están dados por la ecuación (3.38), el cual esta denotado como $\Lambda^{(1)}$ en el diagrama de bloques. Este proceso por lo tanto se representa :

$$\left[\Lambda(x'_k), \Lambda(p'_{1k}), \alpha^{-1}(Le^{(2)}_k) \right] \Rightarrow \Lambda^{(1)}_k$$

Basado en la ecuación (3.23B), la información extrínseca $Le^{(1)}$ del DEC1 es:

$$Le^{(1)}_k = \Lambda^{(1)}_k - \Lambda(x'_k) - \alpha^{-1}(Le^{(2)}_k) \quad (3.39)$$

La ecuación (3.39) es la información RLP extrínseca para la estimación de la secuencia de información x . Nótese que esta es una función solo de la secuencia de paridad p_1' para la secuencia de entrada dada x . Por tanto el error en la estimación es independiente del error en la paridad de la secuencia p_2' . Entonces la información extrínseca generada por el DEC1 y la información intrínseca de p_2' son no correlativas.

Luego, la información extrínseca $Le^{(1)}$ del decodificador DEC1 y la secuencia de bits de información recibida x' son permutadas para aleatorizar el error de ráfaga en la secuencia. Las dos secuencias intercaladas $\alpha(x')$ y $\alpha(Le^{(1)})$ y la segunda secuencia de bits de paridad p_2' ingresan al decodificador DEC2. de manera similar se representa el proceso del DEC2 de la forma:

$$\left[\Lambda(\alpha(x'_k)), \Lambda(p_{2k}'), \alpha(Le^{(2)k}) \right] \Rightarrow \Lambda_k^{(2)}$$

Al RLP de los bits de información estimados del DEC2 se le subtrae la razón máxima de probabilidad de los bits de información intercalada x y la RLP a priori de la entrada para obtener la información extrínseca del DEC2.

$$Le^{(2)k} = \Lambda_k^{(2)} - \Lambda(\alpha(x'_k)) - \alpha(Le^{(1)k}) \quad (3.40)$$

Después que $Le^{(2)k}$ es desintercalado, se convierte en una nueva información a priori para el DEC1. La confiabilidad de la RLP a priori aumenta mientras aumente el número de iteraciones, y el decodificador genera una estimación de mayor confianza. El lazo continua hasta un numero fijo de iteraciones, y el signo de la RLP final es la

información estimada $\Lambda(\hat{x}_k)$ para ser examinado y decidir si el valor de los bit de información es '1' o '0'.

3.5 implementación del algoritmo Log-MAP

En esta sección se hará una descripción previa de la implementación y la discusión de los detalles se mostrarán en el capítulo 4. Las tareas principales del decodificador log-MAP son las siguientes:

- 1.-Cuantización: conversión de las entradas analógicas en digitales.
- 2.-Sincronización: detección de los límites de la trama y de los símbolo-códigos.
- 3.-Cálculo de la métrica de transición, cálculo de la métrica progresiva, y almacenamiento de datos.
- 4.-Cálculo de la métrica regresiva y cálculo de las RLP de los bits de la secuencia de información estimada.
- 5.-Generación de las salidas y las iteraciones de control.

En la figura 3.10 muestra el flujo del algoritmo de decodificación log-MAP, el cual realiza las tareas en el orden especificado.

La cuantización es un factor importante para alcanzar bajas tasas de error de bits (BER) en la turbo decodificación [31]. Con el fin de acomodar la señal recibida dentro de un rango de escala completo del cuantizador, la señal recibida necesariamente tiene que ser ajustada. Factores de escalamiento óptimos han sido obtenidos de [31].

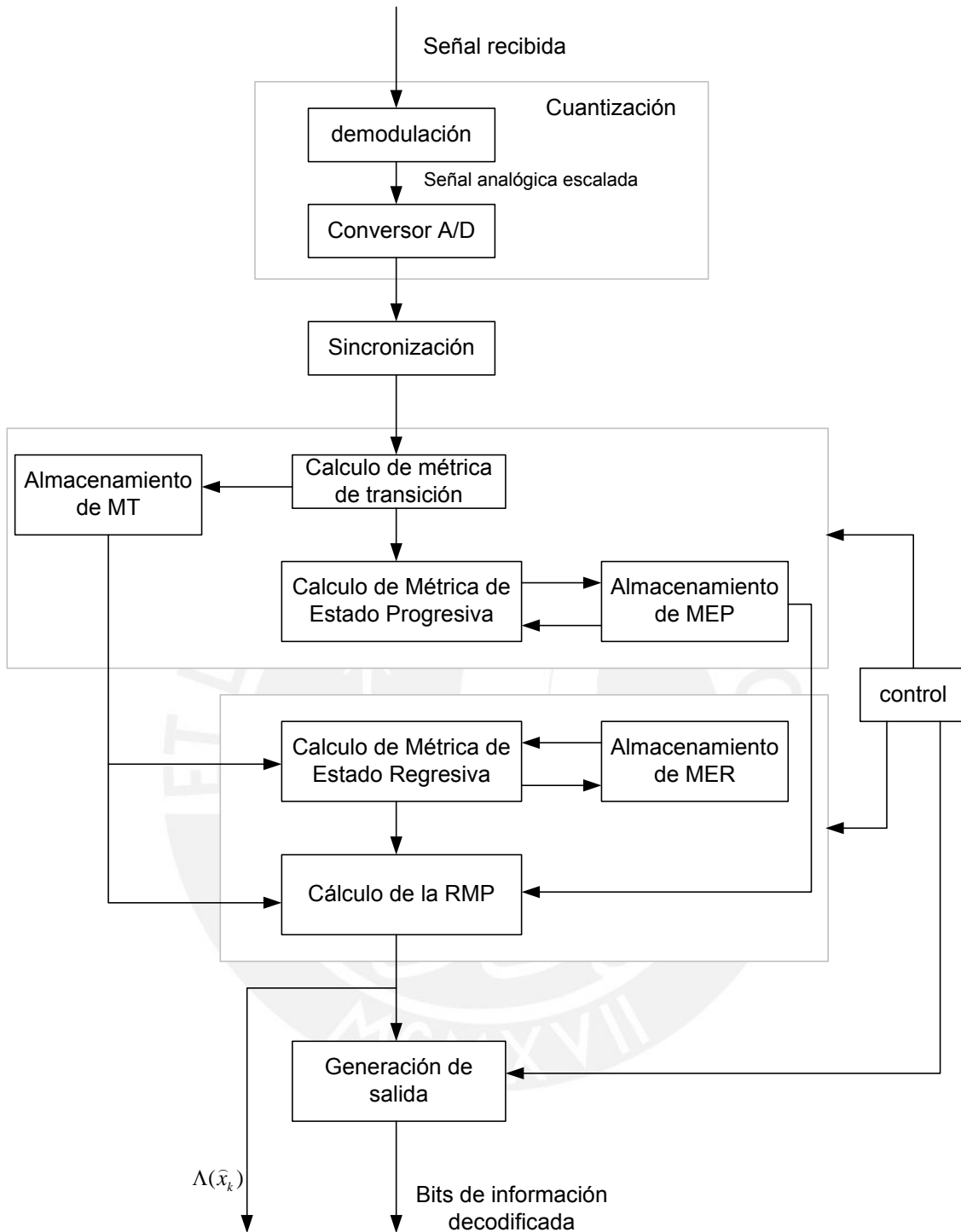


Figura 3.10 Flujo de decodificación del algoritmo log-MAP

El demodulador presentado en la figura 3.10 realiza un escalamiento de las señales analógicas recibidas las cuales por medio del conversor análogo-digital (ADC) son digitalizadas. En el turbo decodificador propuesto se asume que las señales ingresan directamente digitalizadas, después de haber sido demoduladas y cuantizadas a señales digitales de 7 bits. Además se asume que la etapa de sincronización que delimitan las tramas recibidas ya han sido definidas, por tanto el diseño del CORE que se describe a continuación se ha desarrollado desde la etapa del cálculo de la métrica de transición.

La métrica de transición dada en la ecuación (3.37) puede ser simplificada basada en la siguiente igualdad, donde “b” es un número binario y “r” un número real y según [37]:

$$-br = |r|(b \oplus u(r)) - (r + |r|)/2 \tag{3.41}$$

Donde para un número real r $u(r) \equiv \begin{cases} 1 & \text{cuando } r \geq 0 \\ 0 & \text{cuando } r < 0 \end{cases}$,

y para los números binarios b₁ y b₂, $b_1 \oplus b_2 \equiv \begin{cases} 1 & \text{cuando } b_1 \neq b_2 \\ 0 & \text{cuando } b_1 = b_2 \end{cases}$

Notar que u(r) es la función escalón unitario y \oplus es la operación XOR en diseño lógico. Además haciendo $A = \frac{1}{\ln \varepsilon} \frac{2}{\sigma^2}$, las ecuaciones (3.14) y (3.15) pueden denotarse:

$$Lc(x_k) = Ax_k' \quad \text{y} \quad Lc(p_k) = Ap_k'$$

Por tanto la ecuación (3.37) sería:

$$\Gamma_k^{i,m} = -x_k (Ax_k' + x_k L_a(x_k)) - p_k Ap_k' - C_k$$

Haciendo uso de la igualdad de la ecuación (3.41) y tomando en cuenta que x_k y p_k son números binarios, la métrica de transición puede ser expresada como:

$$\begin{aligned} \Gamma_k^{i,m} = & \left| L_a(x_k) + A^* x_k' \right| (x_k \oplus u(L_a(x_k) + A^* x_k')) \\ & - ((L_a(x_k) + A^* x_k') + |L_a(x_k) + A^* x_k'|) / 2 \\ & + |A^* p_k'| (p_k \oplus u(A^* p_k')) - (A^* p_k' + |A^* p_k'|) / 2 - C_k \end{aligned}$$

Puesto que solo las variables x_k y p_k dependen del estado m , se puede simplificar la expresión así como se muestra en [32] y [33]:

$$\Gamma_k^{i,m} = \left| L_a(x_k) + A^* x_k' \right| (x_k \oplus u(L_a(x_k) + A^* x_k')) + |A^* p_k'| (p_k \oplus u(A^* p_k')) - C_k - C_k^*$$

Los dos términos constantes son comunes para todos los estados en una estación k , y solo los valores relativos de las métricas de transición en una estación son significantes. Por lo tanto estos dos términos constantes C_k y C_k^* pueden ser ignorados en la implementación, con lo cual la ecuación final a implementar sería:

$$\Gamma_k^{i,m} = \left| L_a(x_k) + A^* x_k' \right| (x_k \oplus u(L_a(x_k) + A^* x_k')) + |A^* p_k'| (p_k \oplus u(A^* p_k')) \quad (3.42)$$

El diagrama de bloques mostrado en la figura 3.11 implementa la ecuación (3.42). El módulo ABS calcula el valor absoluto del canal de información. Basado en las salidas de las compuertas lógicas XOR, un multiplexor escoge la correspondiente métrica de transición hacia la salida.

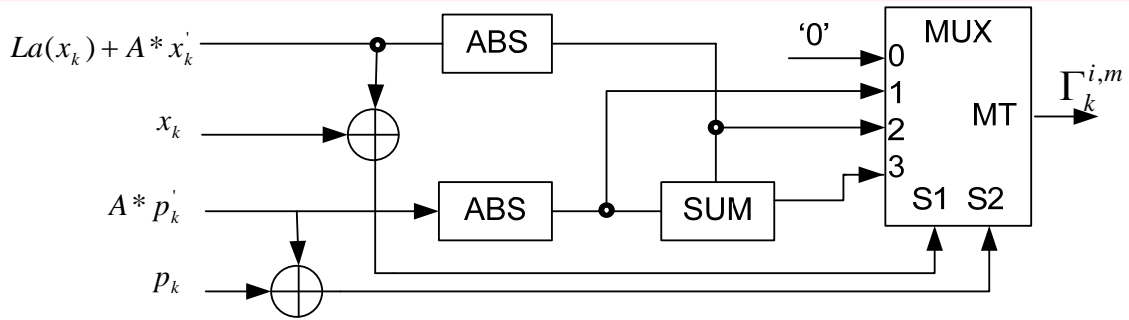


Figura 3.11: Bloque de cálculo para la métrica de transición.

Una vez que la métrica de transición es calculada para cada transición, esta es almacenada en una memoria RAM para ser usada posteriormente pero a la vez es utilizada al mismo tiempo para el cálculo de la métrica de estado progresiva (MEP) usando el diagrama de trellis de la figura 3.12. El diagrama de trellis mostrado en la figura 3.12 tiene una estructura en la cual se muestran los estados de origen y destino, y cuatro transiciones de interconexión.

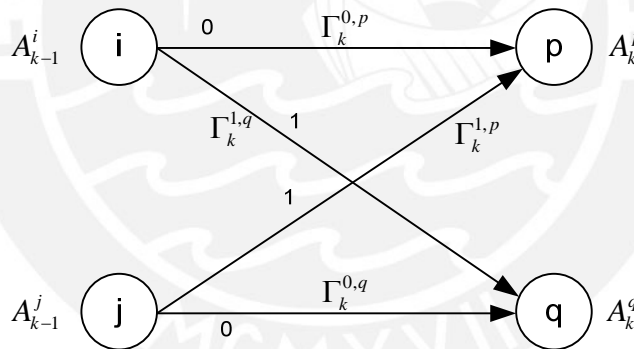


Figura 3.12: Calculo de la MEP.

En la figura, las transiciones etiquetadas con '1' ('0') corresponden al bit de información de entrada '1'('0'). Por otro lado se guarda la siguiente relación para un $(n,1,K)$ codificador RSC.

$$\begin{aligned}
 J &= i + 2^{k-2} \\
 P &= 2i \\
 Q &= 2i + 1
 \end{aligned}$$

La operación clave en el bloque para determinar las métricas de estado (ME) es la operación-E. El bloque suma las dos métricas de estado de los dos estados previos con su correspondiente métrica de transición y obtiene dos métricas de camino del estado actual. Hasta este punto, el proceso es idéntico al algoritmo de Viterbi. Estas dos métricas de camino son comparadas y la mínima es seleccionada. Sin embargo, el cálculo de la MEP no termina ahí. La función de corrección, la cual es el valor absoluto de la diferencia de las dos métricas de camino, debe ser calculada y utilizada para ajustar la métrica de estado. El diagrama de bloques de la operación-E se muestra en la figura 3.13. El modulo de la función $f(z)$ en la figura puede ser implementada usando una Tabla de búsqueda (TDB).

Para cada estado en la estación “k” es necesaria una operación-E. Con el fin de reducir el número de compuertas, se utiliza usualmente una implementación serial, esto significa que una MEP es calculada durante un periodo de reloj. En el presente diseño se tienen 16 estados por cada estación en el trellis, por lo tanto se usan 16 ciclos de reloj para el cálculo de las 16 MEPs para cada estación.

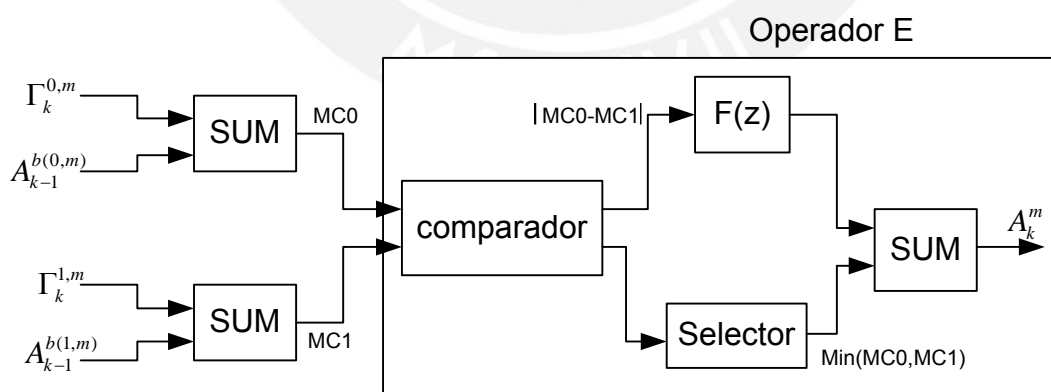


Fig 3.13: Módulo de Operación-E en el bloque de cálculo de la MEP

Después que las MEPs son calculadas para cada punto del trellis, las MERs son calculadas en el orden regresivo. Considerando el arreglo de estados de la figura 3.14, el objetivo del MER es calcular B_k^i y B_k^j para las métricas de estado dadas B_{k+1}^p y B_{k+1}^q , y las métricas de transición $\Gamma_{k+1}^{0,p}$, $\Gamma_{k+1}^{1,p}$, $\Gamma_{k+1}^{0,q}$ y $\Gamma_{k+1}^{1,q}$. Las métricas de transición son obtenidas de la memoria RAM en la cual han sido almacenadas durante el proceso de cálculo de dichas TM. El proceso de suma, comparación y selección y la operación-E son las mismas que las utilizadas para el calculo de las MEPs.

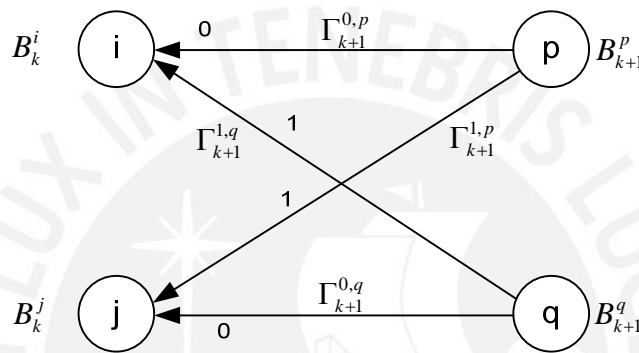


Figura 3.14: Calculo de la MER.

Mientras los cálculos de las MERs son desarrollados, las Razones Máximas de Probabilidad (RLPs) también son calculadas al mismo tiempo gracias al módulo RLP. Considerando el estado “i” en la estación “k” en la figura 3.14 como ejemplo tenemos que: Durante el calculo de B_k^i , el camino regresivo inmediato $B_{k+1}^p + \Gamma_{k+1}^{0,p}$ y $B_{k+1}^q + \Gamma_{k+1}^{1,p}$ son aplicados al módulo RLP. Al mismo tiempo, la MEP A_k^i para el estado “i” en la estación “k” es leído de la RAM correspondiente. El módulo RLP suma A_k^i a las dos métricas de camino regresivas respectivamente y ejecuta la operación-E individualmente. Desde que hay 16 estados en cada estación, la operación-E en el módulo RLP contiene 16 términos. Como se verifico en la sección 3.4.4 la operación-E con múltiples términos puede ser calculada recursivamente usando la ecuación (3.28). El diagrama de bloques del circuito se muestra en la figura 3.15. Durante cada ciclo de

reloj, la nueva métrica calculada y la salida del registro actual son realimentadas al circuito de la operación-E y el resultado es almacenado en el mismo registro para su posterior uso en el siguiente ciclo de reloj. Cada modulo RLP toma una métrica de camino en cada tiempo, mientras que en paralelo dos módulos RLP son necesarios para calcular $\sum_{m=0}^{2^v-1} E(A_{k-1}^{b(0,m)} + \Gamma_k^{0,m} + B_k^m)$ y $\sum_{m=0}^{2^v-1} E(A_{k-1}^{b(1,m)} + \Gamma_k^{1,m} + B_k^m)$ respectivamente. Puesto que hay $2^v=16$ estados por cada estación, la salida de la RLP del decodificador log-MAP se actualiza cada 16 ciclos de reloj.

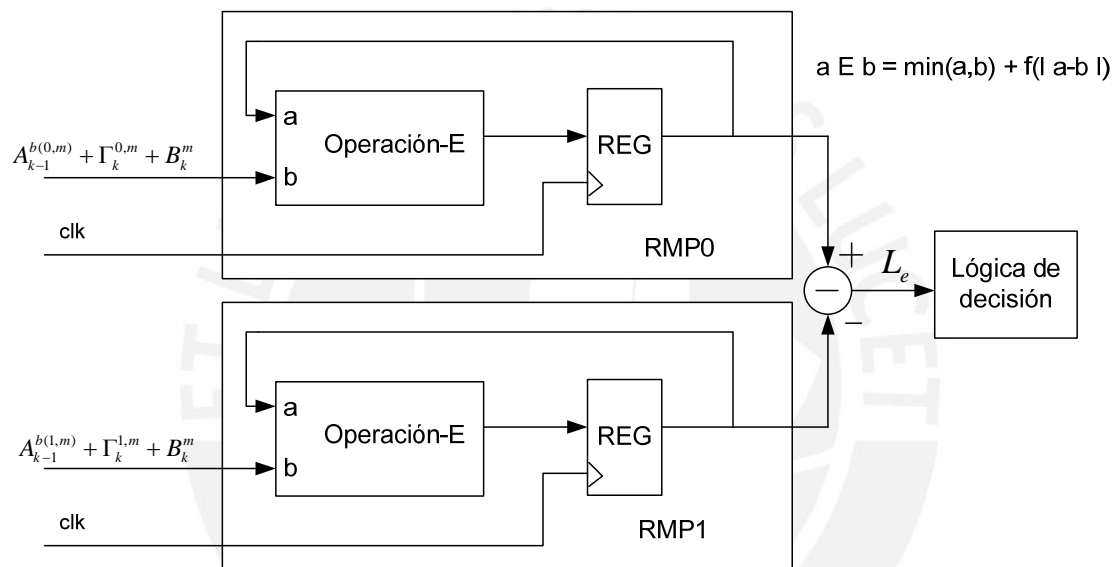


Figura 3.15: Diagrama de bloques para el cálculo de la RLP

CAPITULO 4:

DISEÑO SOBRE FPGA DEL TURBO CODIFICADOR- DECODIFICADOR

PROPUESTOS

4.1 Propuesta de turbo decodificador

En el capítulo anterior se describió el algoritmo Log-MAP. El cual es el centro del turbo decodificador. En este capítulo se describe y detalla cada uno de los bloques diseñados para la implementación de este algoritmo sobre un FPGA de Altera. En primer lugar se presenta el diagrama de bloques de la más alta jerarquía del decodificador Log-MAP. Posteriormente se muestran los distintos bloques que lo conforman. Finalmente se diseña un bloque de control para comprobar el funcionamiento de los distintos bloques mostrados.

4.2 Diseño de bloques para el Algoritmo Log-MAP.

En esta sección se detalla el diseño de cada uno de los bloques que conforman cada uno de los decodificadores Log-MAP, así como también el módulo de control independiente para cada uno de estos componentes.

4.2.1 Bloque general para el decodificador Log-MAP.

Los bloques que constituyen el turbo decodificador log-MAP se muestra en la figura 4.1 de los cuales se observa que los 4 componentes principales son el bloque de cálculo de la métrica de transición (MT), el bloque de cálculo de la métrica de estado progresiva (MEP), el bloque de cálculo de la métrica de estado regresiva (MER) y el bloque de cálculo de la razón logarítmica de probabilidad (RLP). El circuito de control es también parte del decodificador. De entre todos los bloques mencionados cabe

mencionar que hay 2 unidades de memoria RAM de 4k x 8 bits que son usadas para almacenar las MEPs y las MERs respectivamente, así como también 2 memorias RAM de 512 x 8 bits para el almacenamiento de las MTs.

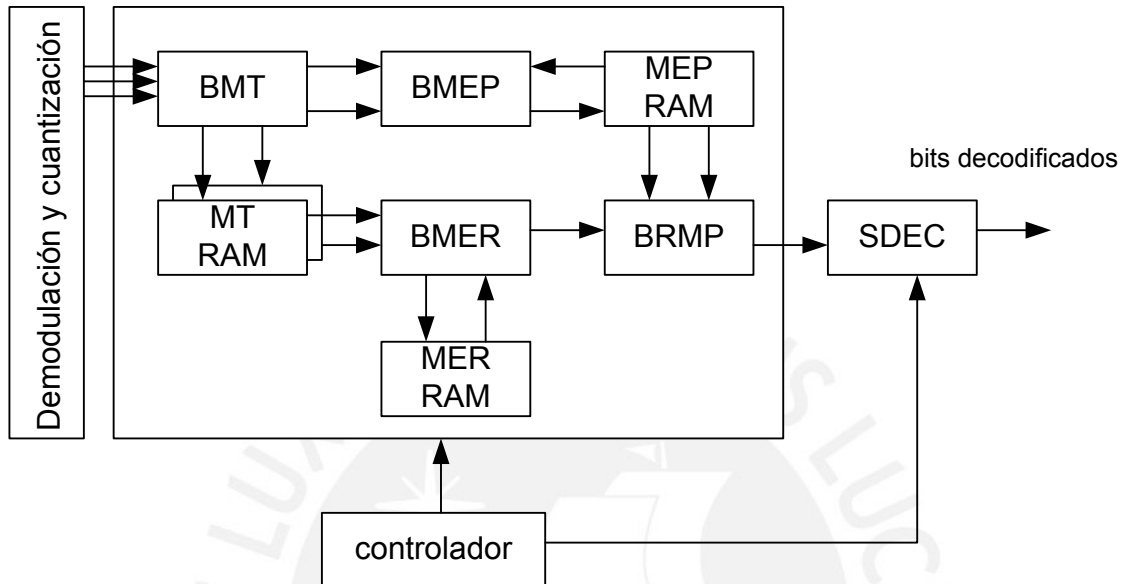


Figura 4.1 Diagrama de bloques del decodificador log-MAP

Todas las notaciones usadas en el capítulo 3 serán mantenidas con excepción de las siguientes definiciones:

1.- La secuencia de bits de información x_i' , bits de paridad $p1_i'$ y $p2_i'$ son cuantizados con $q=7$ bits en complemento a 2 por tanto están comprendidos en el rango de -64 a 63. la información cuantizada está denotada por lo tanto como sys , $par1$ y $par2$, las cuales corresponden a x_i' , $p1_i'$ y $p2_i'$ respectivamente. El diseño de esta tesis no implementa el demodulador ni el conversor análogo digital.

2.- L_a es la probabilidad *a priori* (PraP) proveniente del cálculo del decodificador complementario. Y e la misma es cuantizada con 8 bits en el rango de -128 a 127.

3.- A_k^m , B_k^m y $\Gamma_k^{i,m}$ usados en el capítulo 3 son reemplazados por MEP_k^m , MER_k^m y MT_k^m respectivamente.

Basado en la figura 4.1 se detalla a grandes rasgos la relación existente entre cada uno de los bloques de cálculo dentro de uno de los decodificadores Log-MAP:

En la unidad de cálculo BMT recibe 3 entradas: L_a , sys y $par1$ ($par2$) para el decodificador 1 (decodificador 2). Esta genera las métricas de transición (MTs) correspondientes, las cuales son alimentadas hacia la unidad de cálculo BMEP y además son almacenadas en dos memorias (MT RAM) al mismo tiempo. La unidad BMEP calcula las nuevas MEPs usando las MEPs almacenadas en las memorias (MEP RAM) de la estación anterior y las MTs correspondientes a la estación actual. Después que se han calculado todas las MEPs y las MTs para una trama completa, la unidad BMER empieza con el cálculo de las MERs en forma regresiva haciendo uso de las MERs almacenadas en las memorias (MER RAM) de la estación posterior y las MTs correspondientes a la estación actual. Al mismo tiempo, el unidad de cálculo BRLP genera las razones máximas de probabilidad (RLP) basadas en las MEPs, MERs y MTs calculadas para cada estación. Si esta operación constituye la última iteración del proceso de turbo decodificación, la información estimada es generada por el bloque SDEC.

4.2.2 Módulo de cálculo de la métrica de transición (BMT).

El cálculo de la métrica de transición esta basada en los valores de entradas recibidas y los símbolos de código esperados en el trellis. Los 16 estados en el tiempo "k" del trellis cambian hacia un estado posterior en el tiempo "k+1" de acuerdo a al bit de información que ingresa x_i' . La transición de estados de la estación "k-1" hacia la

estación “k” puede ser descrita por medio de 8 estructuras en forma de mariposa. En donde cada mariposa corresponde a dos estados de entrada y dos estados de salida los cuales se muestran en la figura 4.2. Los símbolos esperados para cada posible transición están etiquetados en cada una de las líneas de cada mariposa con el formato ‘ip’, donde ‘i’ es el bit de información entrante, y ‘p’ es el bit de paridad correspondiente al componente de codificación.

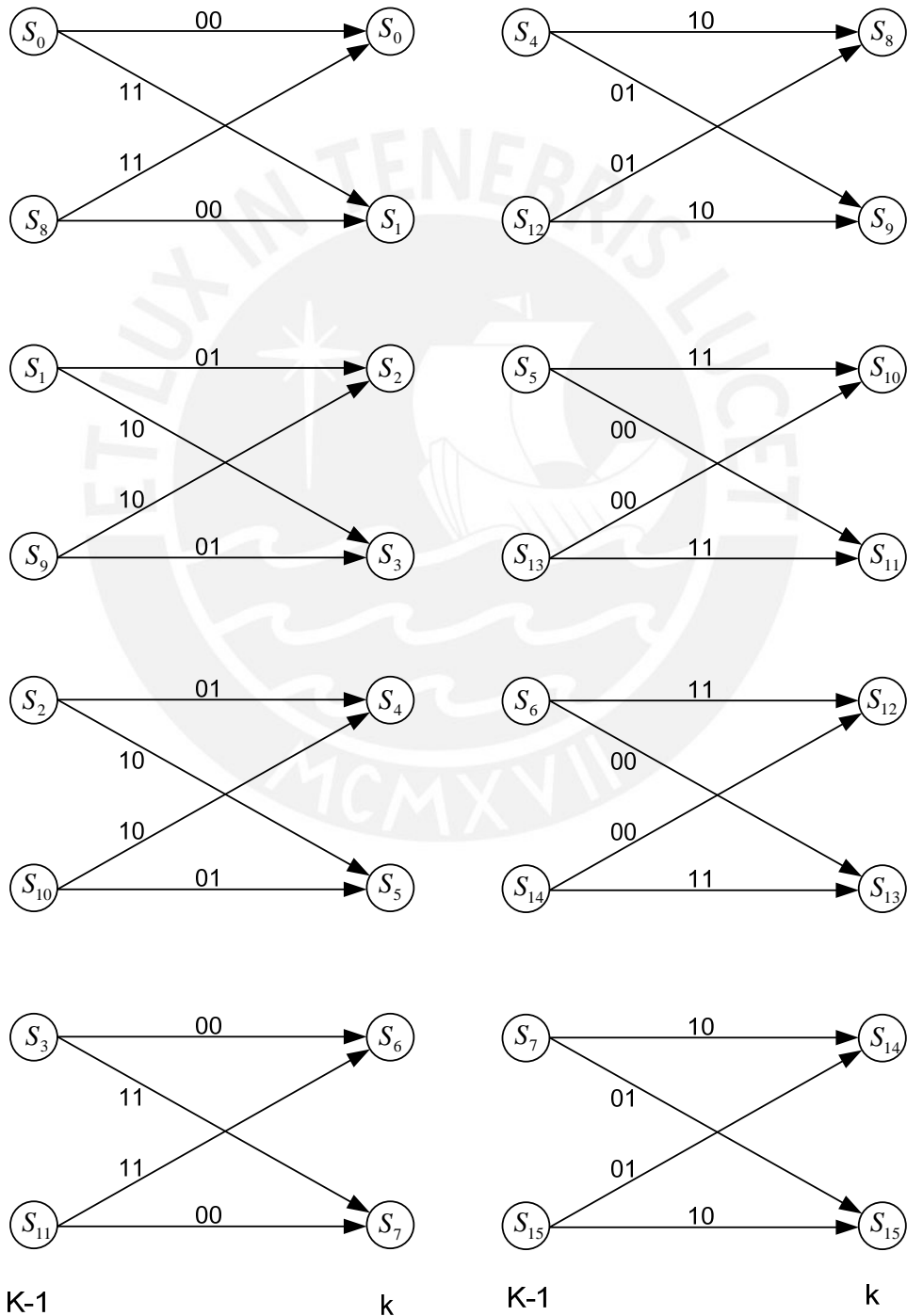


Figura 4.2: Transiciones de estado para el tiempo $k-1$ hacia el tiempo k .

En la sección 3.5, la ecuación (3.42) muestra el cálculo de las métricas de transición la cual en esta sección es descrita con la nueva notación con el fin de adaptarla a las variables utilizadas en el diseño con lo cual se tiene:

$$MT_k^{i,m} = |L_{-a_k} + sys_k|(i \oplus u(L_{-a_k} + sys_k)) + |par_k|(p^{i,m} \oplus u(par_k)) \quad (4.1)$$

La métrica de transición $MT_k^{i,m}$ es la métrica para la transición que tiene como destino el estado ‘m’ en la estación ‘k’ debido al bit de información ‘i’. La ecuación mostrada esta representada en una tabla de verdad. La tabla 4.1 es el punto de partida para el diseño del modulo BMT.

$i \oplus u(L_{-a_k} + sys_k)$	$p^{i,m} \oplus u(par_k)$	$MT_k^{i,m}$
0	0	0
0	1	$ par_k $
1	0	$ L_{-a_k} + sys_k $
1	1	$ L_{-a_k} + sys_k + par_k $

Tabla 4.1 Tabla de verdad de la unidad de cálculo BMT.

En la tabla anterior, $u(x)$ es la función escalón unitaria la cual entrega ‘1’ (‘0’) si $x \geq 0$ ($x < 0$). Como se mencionó anteriormente, la probabilidad a priori L_{-a_k} de la estación k es un número con signo de 8 bits que va desde -128 a 127, mientras que sys_k y par_k en una estación k es un número de 7 bits que va desde -64 hasta 63. Solo el bit de información ‘i’ y el bit de paridad ‘p’ estimados son específicos para cada transición, mientras que los otros parámetros son comunes para todas las 32 transiciones correspondientes para una estación ‘k’. Con el fin de limitar el número de bits para expresar una Métrica de estado (ME), se utiliza un número de 7 bits sin signo para representar un MT, por tanto el máximo valor posible alcanzado por un MT esta limitado a 127. Notar que el valor de una MT es siempre positivo, por tanto un valor

mayor implica que el camino correspondiente a esa transición es menos probable en el trellis. Debido a las limitaciones, el resultado de la suma es truncada si $L_a_k + sys_k$ es mayor a 127 o menor a -128. Si tanto $(i \oplus u(L_a_k + sys_k))$ como $(p^{i,m} \oplus u(par_k))$ son '1', la suma de $|L_a_k + sys_k|$ y $|par_k|$ también es limitada a 127. Un sumador con acarreo con tamaño de palabra de 7 bits es utilizado para realizar la operación. Cuando el bit de acarreo de salida es '0'. La suma de los dos datos es menor o igual a 127, por tanto no hay necesidad de limitar los bits. Sin embargo, si el bit de acarreo de salida es '1', lo que implica que la adición ha causado un desbordamiento, los 7 bits son puestos a '1' para limitar la salida resultante igual a 127. Se utilizan 7 compuertas lógicas OR de 2 entradas para implementar el circuito limitador. El diagrama de bloques de la operación que se describirá a continuación se muestra en la figura 4.3.

El bloque LIM1 limita el ancho de palabra a un rango deseado. El bloque ABS calcula el valor absoluto de los datos entrantes ya sean provenientes del LIM1 o de los bits de paridad. Un multiplexor determina la salida en complemento a 2 de la entrada o la entrada misma dependiendo del bit MSB de la misma. Si el MSB es '0', la salida del bloque ABS es la entrada misma, de otra manera el complemento a 2 de la entrada pasa como salida. Durante un periodo de reloj de datos DCLK, las cuatro entradas del MUX se mantienen debido a que las 3 entradas L_a , sys y par se mantienen iguales. Solo una de las cuatro entradas del MUX es seleccionada como salida del BMT basado en los símbolos esperados leídos desde la ROM.

Los símbolos esperados 'ip' para cada transición de estados son almacenados en memorias ROM. Donde entre los símbolos esperados, el que es diferente en cada estado es 'p' para $i=0$, en consecuencia, se usa dos memorias ROM de 16x1bit para almacenar 'p' cuando $i=0$ y $i=1$ respectivamente. Los bits de paridad 'p' son

almacenados en la ROM1 en las direcciones desde la 0 hasta la 15 para cuando $i=0'$. Los valores de 'p' para $i=1'$ son almacenados en la ROM2. El ROMs está controlado por un contador de 0 a 15 que trabaja con la señal del reloj de sistema del decodificador CLK.

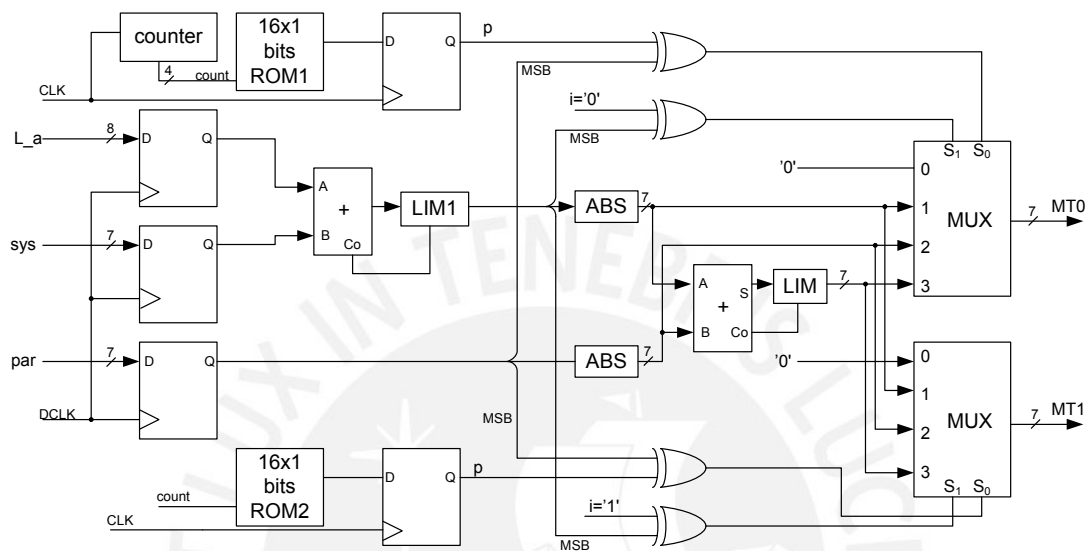


Figura 4.3 diagrama de bloques del la unidad de cálculo BMEP.

Puesto que las métricas de transición solo dependen de la información a priori. (los bits sistemáticos y los de paridad). Hay solo cuatro métricas de transición diferentes para cada estación k correspondientes, las cuales son $ip= 00 , 01, 10$ y 11 . Pero debido a que estas métricas también son usadas al mismo tiempo por la unidad de cálculo BMEP, un par de MTs deben ser calculadas, $MT_k^{0,m}$ y $MT_k^{1,m}$ necesitan ser halladas por lo tanto para ser usadas en el cálculo de la métrica de estado progresiva. Examinando la ecuación (4.1) y la estructura de la figura 4.2 revela que cada par de transiciones que terminan en estado $2m$ y estado $2m+1$ tienen los mismos MTs, Tomando la primera estructura mariposa de la figura 4.3 como ejemplo. Los estados de entrada son S_0 y S_8 , y los estado salida S_0 y S_1 . Las transiciones de estado desde S_0 hasta S_0 y desde S_8 hasta S_1 corresponden a los datos de entrada $ip=00$. Puesto

que las dos transiciones de estado tienen los mismos símbolos esperados, sus MTs son las mismas. Por tanto, solo se necesita calcular las MTs de las transiciones terminadas en los estados $2m$, donde $m=0, 1, 2, \dots, 7$. Con el fin de simplificar el diseño, un par de MTs, $MT_k^{0,2m}$ y $MT_k^{1,2m}$, cuyas transiciones terminan en el mismo estado $2m$ en la estación k son calculados en paralelo.

Como las métricas de estado también son necesarias para el cálculo de las métricas de estado regresivas. Las MTs calculadas son almacenadas en dos memorias RAM. Para disminuir el uso de compuertas en el diseño, solo cuatro métricas de transición son almacenadas $MT_k^{0,7}$ y $MT_k^{0,15}$, correspondientes a $ip=00, 01$, las cuales son almacenadas en la primera RAM. $MT_k^{1,15}$ y $MT_k^{1,7}$ son almacenadas en la segunda RAM y corresponden a $ip=10, 11$ respectivamente. En total, $2N$ elementos de almacenamiento se requieren para cada MT RAM, donde N es la longitud de la trama $N=256$ para el decodificador presentado, por tanto 1024 bytes de memoria se necesitan para almacenar todas las MTs. Con la finalidad de poder acceder a $MT_k^{0,m}$ y $MT_k^{1,m}$ al mismo tiempo, empleo dos memorias RAM asíncronas con un puerto estático simple y de 512 Bytes.

Puesto que las MTs almacenadas son las que se obtienen en el octavo y dieciseisavo ciclo de reloj CLK en cada estación, y puesto que coincide con la mitad del periodo de un ciclo del reloj de datos DCLK, las MTs a ser usadas están listas para ser leídas a partir del siguiente ciclo del reloj DCLK.

4.2.3 Módulos de cálculo de las métricas de estado MEP y MER.

La estructura para el bloque de la métrica de estado progresiva (BMEP) y para el bloque de la métrica de estado regresiva (BMER) son muy similares. Se explicará en esta sección la operación de la unidad de cálculo BMEP al detalle y posteriormente se describen las diferencias que existen con respecto a la unidad de cálculo BMER.

Cada nodo en el trellis necesita dos métricas de estado, la métrica de estado progresiva (MEP) y la métrica de estado regresiva (MER). Después de considerar la precisión en los cálculos y la complejidad de hardware que podría generar, se asigna un número de 8 bits sin signo a cada métrica de estado. Por tanto el rango de cada métrica es de 0 a 255.

4.2.3.1 Cálculo de la MEP.

Para cada nodo S_m , existen dos transiciones de entrada b_i (correspondiente al bit de información '0') y b_j (correspondiente al bit de información '1') partiendo de S_i y S_j respectivamente. La MEP de S_m es la operación-E de las dos métricas de camino (la MEP de S_i + la MT de b_i) y (la MEP de S_j + la MT de b_j). Se ilustra el proceso usando la estructura de mariposa mostrado en la figura 4.4. Suponiendo que se quiere calcular la MEP del estado S_6 en el tiempo k , MEP_k^6 . Partiendo de las relaciones de las estructuras en mariposa de la sección 3.5, se puede identificar dos transiciones entrantes y sus respectivos estados de partida S_3 y S_{11} . La MEP es calculada de la siguiente manera:

$$MEP_k^6 = E(MEP_{k-1}^3 + MT_k^{0,6}, MEP_{k-1}^{11} + MT_k^{1,6})$$

De acuerdo con la definición de la operación-E descrita en la sección 3.5, la siguiente ecuación es expresada como:

$$MEP_k^6 = \min(MEP_{k-1}^3 + MT_k^{0,6}, MEP_{k-1}^{11} + MT_k^{1,6}) - f(|(MEP_{k-1}^3 + MT_k^{0,6}) - (MEP_{k-1}^{11} + MT_k^{1,6})|)$$

y
$$f(z) = c \ln(a + e^{-z/c}) , z \geq 0$$

donde c es determinado de acuerdo al tamaño de la tabla TDB. El estudio sobre el valor a usar y el tamaño de la tabla están analizados en [34]. Donde se utiliza una TDB de 64x4 bits y un valor de c de 21.6.

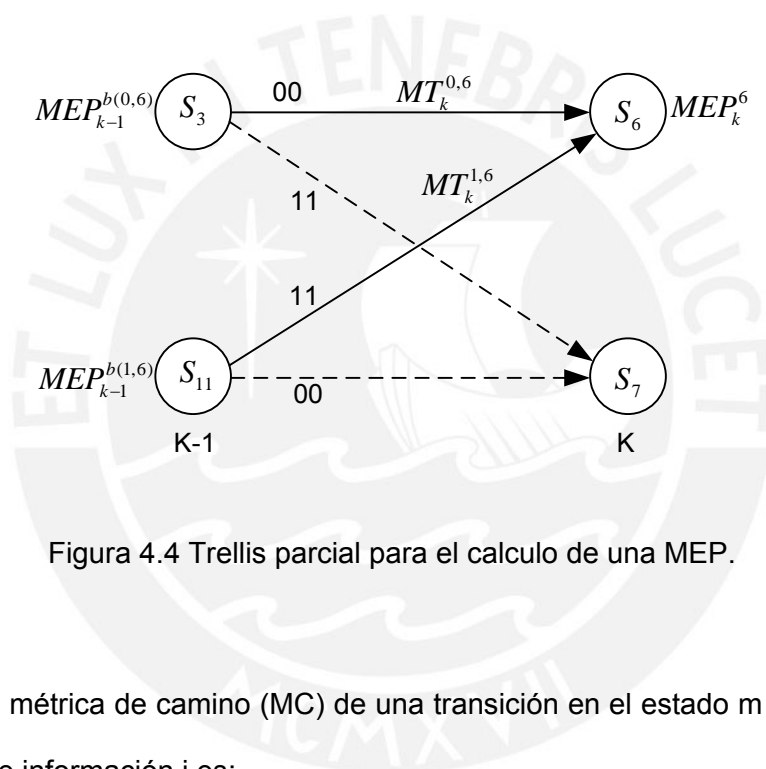


Figura 4.4 Trellis parcial para el calculo de una MEP.

Denotando la métrica de camino (MC) de una transición en el estado m para el tiempo k bajo el bit de información i es:

$$MC_k^{i,m} = MEP_{k-1}^{b(i,m)} + MT_k^{i,m} \tag{4.2}$$

Notar que $MEP_{k-1}^{b(i,m)}$ denota la MEP del estado en el tiempo k-1 el cual antecede al estado m cuando la entrada de información i. En general, un MEP para un nodo m en el tiempo k esta redefinido de la forma:

$$MEP_k^m = \min(MC_k^{0,m}, MC_{k-1}^{1,m}) - f(|MC_k^{0,m} - MC_k^{1,m}|) \tag{4.3}$$

Donde $f(z) = 21.6\ln(1 + e^{-z/21.6})$, $z \geq 0$ (4.4)

Puesto una adición de números positivos es mas simple que una sustracción, se hizo que $f(z)$ absorba el signo “menos”. Con lo cual el cálculo de la MEP se convierte en:

$$MEP_k^m = \min(MC_k^{0,m}, MC_{k-1}^{1,m}) + f(|MC_k^{0,m} - MC_k^{1,m}|) \quad (4.5)$$

Donde $f(z) = -21.6\ln(1 + e^{-z/21.6})$, $z \geq 0$

(4.6)

$f(z)$ en la ecuación (4.6) es siempre negativa teniendo como mínimo valor a $-21.6 \cdot \ln 2$ para $z=0$. Sin embargo para evitar el uso de números negativos para la TDB , se le suma el valor $21.6 \cdot \ln 2$ a $f(z)$. La ecuación queda modificada de la siguiente manera:

$$f(z) = -21.6\ln(1 + e^{-z/21.6}) + 21.6\ln 2, \quad z \geq 0 \quad (4.7)$$

Por lo tanto el resultado de $f(z)$ es mayor o igual que 0, donde datos de 4 bits sin signo se usaron para representar $f(z)$. Esto será relevante si los valores relativos, no absolutos de las MEPs son importantes. una TDB para $f(z)$ se muestra en la figura 4.5:

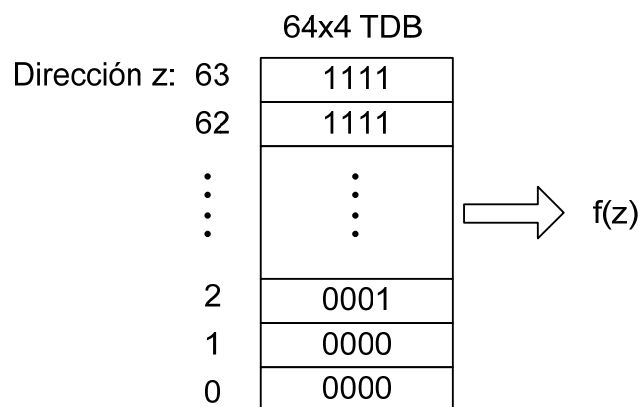


Figura 4.5: TDB de ajuste para la función $f(z)$

un diagrama de tiempos que revela como es calculado la MEP para un tiempo k y para un estado determinado.

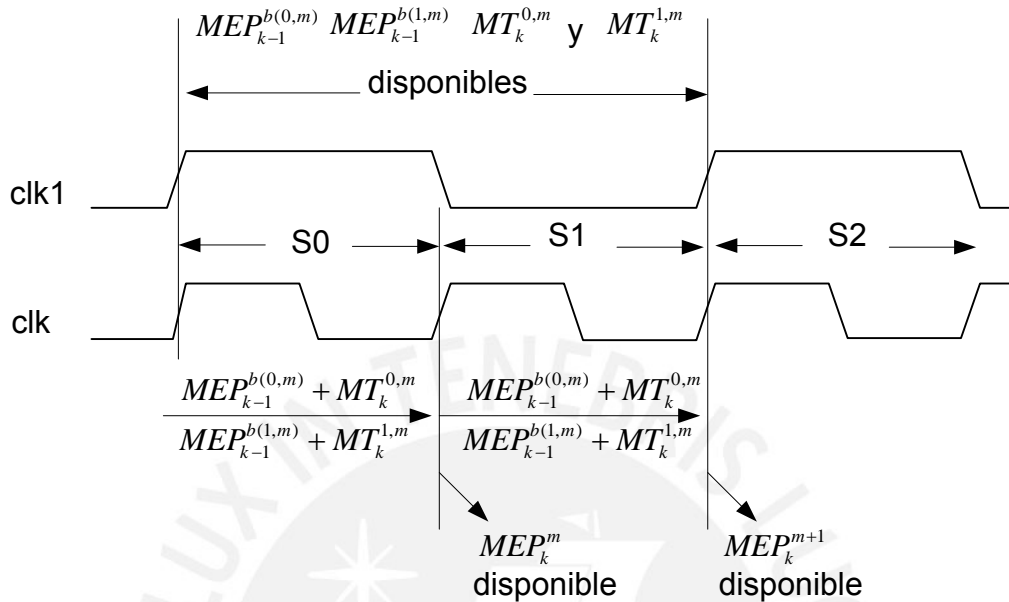


Figura 4.7 Diagrama de tiempos para el cálculo de la MEP.

Por tanto como se mostró en la ecuación (4.5), las métricas de camino deben ser calculadas antes de proceder con la operación-E. La figura 4.8 muestra como es calculada esta métrica de camino MC_k sumando MEP_{k-1} y MT_k . Puesto que en este proceso se suman dos números positivos directamente, el MEP_{k-1} puede ocasionar un desborde. Para hacer frente a este problema, se resta el mínimo MEP de la estación k-1 de $MEP_{k-1}^{b(i,m)}$, antes que la suma sea desarrollada. Esta operación previa será relevante si es que el valor relativo de las MEPs son importantes. Después de la adición, un circuito limita la suma a 8 bits. En caso haya un desborde, la suma se hace 255.

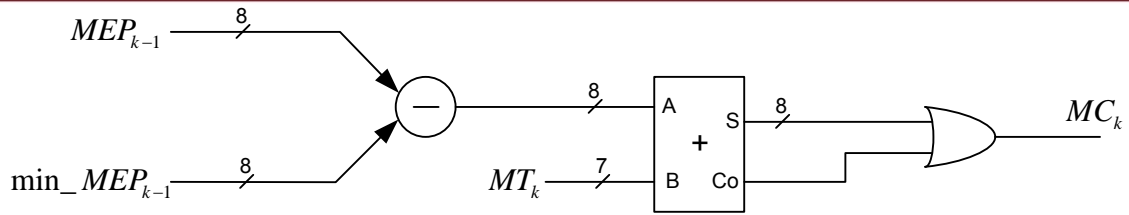


Figura 4.8: Sumador previo para determinar la MC.

La operación-E es el punto clave para el cálculo de las métricas de estado (tanto progresiva como regresiva). Por conveniencia se denota como MC0 y MC1 para representar las métricas de camino $MC_k^{0,m}$ y $MC_k^{1,m}$ respectivamente. Como se muestra en la ecuación (4.3), la operación-E incluye la búsqueda de la mínima métrica de camino y el cálculo del valor absoluto de la diferencia entre MC0 y MC1. Posteriormente gracias a la implementación de la tabla de búsqueda TDB, se puede realizar la función $f(z)$ la cual es sumada al mínimo valor de las métrica de camino para obtener una MEP. La figura 4.9 muestra como se ha diseñado el circuito para el cálculo de la operación-E. Tanto MC0 y MC1 son números de 8 bits sin signo. El Sumador1 compara las magnitudes de los dos datos adquiridos de la manera siguiente:

$$Co1='1' \text{ si } PM0 > PM1 \quad \text{y} \quad Co2='0' \text{ si } PM0 < PM1$$

El Sumador2 realiza la operación de substracción, $PM0 - PM1$, en complemento a 1. La compuerta XNOR se encarga de cambiar en signo si es que el resultado es negativo. Por tanto el nodo U es equivalente a $|PM0 - PM1|$. El circuito LIM limita las entradas a un rango de 63, y la salida de este circuito sirva para direccional z a través de la TDB. El MUX selecciona la mínima métrica de camino PM0 o PM1, para ser adicionado a $f(z)$, finalmente la suma es limitada a 255 por medio de una compuerta OR.

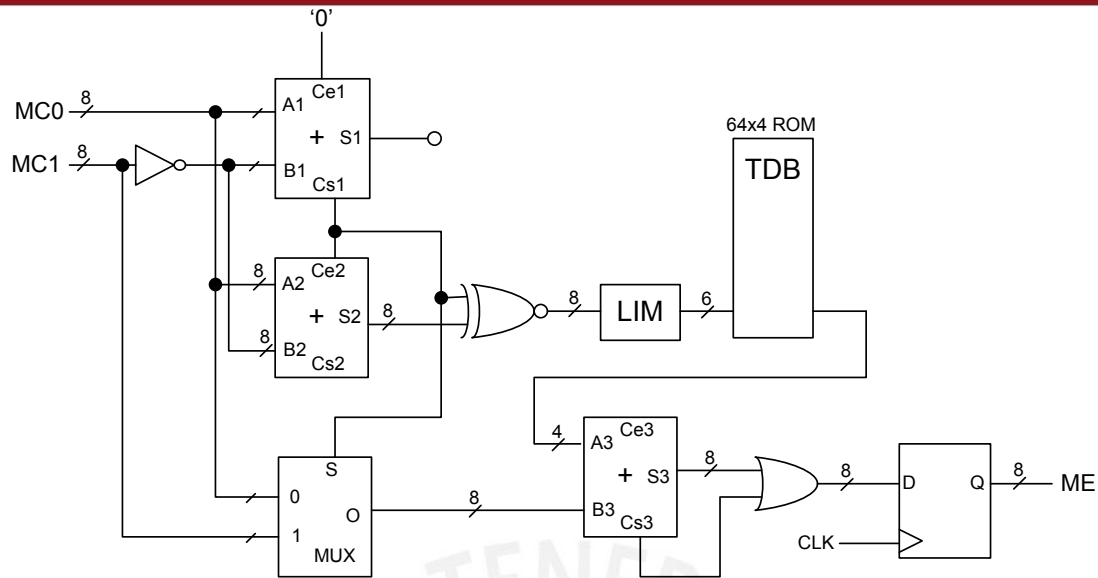


Figura 4.9: Circuito de Operación-E

4.2.3.2 Cálculo para la mínima métrica de estado (min_ME).

Cuando una nueva MEP es calculada, esta es almacenada en la MEP RAM y a la vez pasa al bloque de cálculo para determinar la mínima métrica de estado (min_MEP) el cual se muestra en la figura 4.10. La nueva MEP es comparada con la mínima MEP actual y almacenada en los flip-flops DFF2. El juego de flip-flops DFF1 tienen la función de almacenar la mínima MEP de la estación $k-1$ mientras que la mínima MEP es calculada para la estación actual k . Los flip-flops DFF1 son actualizados solo cuando el cálculo de las MEPs actuales ha terminado cada 16 ciclos del reloj CLK. Inicialmente, los flip-flop DFF2 son cargados al más alto valor, es decir a 255 para dar inicio al cálculo de la mínima MEP para la siguiente estación. Por otro lado para inicializar la MEPs en todo el trellis al comienzo de una nueva trama es necesario que el estado inicial sea S_0 en la estación 0 y que el valor de la MEP correspondiente MEP_0^0 , sea igual a 0. Para los otros 15 estados MEP_0^m , $m=1,2,\dots,15$, el valor sea 255.

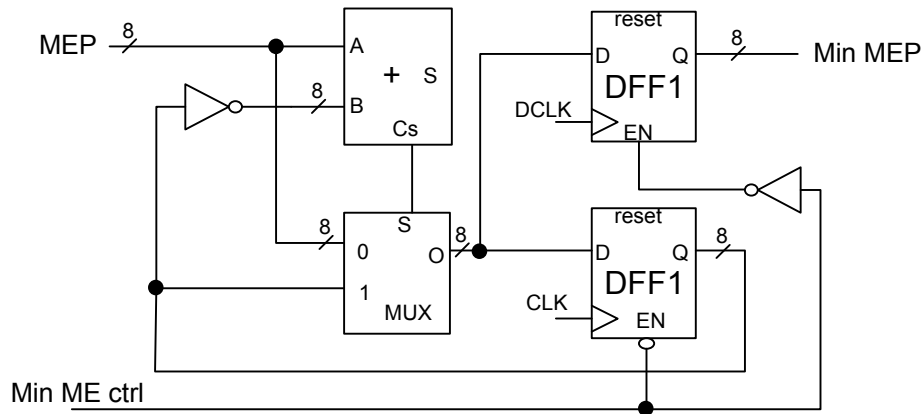


Figura 4.10 Bloque de cálculo para determinar la mínima MEP por estación.

4.2.3.3 Almacenamiento de las MEPs.

Para el cálculo de cada MEP en una estación k , es necesaria la información de la MEPs de la estación anterior $k-1$, por esta razón la forma como se almacena la información y como se lee para el cálculo de la nueva RAM se muestra en la figura 4.11. En la cual el proceso de escritura se realiza en forma ordenada para cada estación, es decir para una estación k , se almacenan las MEPs de manera ascendente. Para leer las MEPs de la estación anterior se tiene que leer en el orden descrito en la sección 4.3.2.1

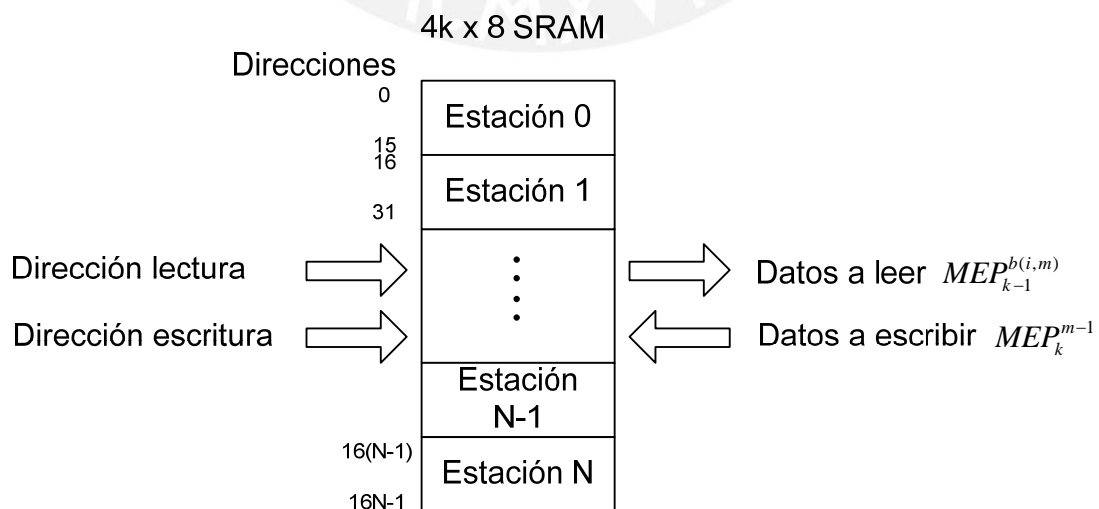


Figura 4.11: Almacenamiento en RAMs para las MEPs

4.2.3.4 Cálculo de la MER.

El cálculo de las MER (Métricas de Estado Regresivas) es el mismo que para las MEPs con excepción de los índices de tiempo. Una MER_k^m de un estado m en el tiempo k es calculada usando las dos MEPs en el tiempo k+1 ($MER_{k+1}^{f(0,m)}$ y $MER_{k+1}^{f(1,m)}$) y las dos MTs ($MT_{k+1}^{0,m}$ y $MT_{k+1}^{1,m}$). La ecuación que calcula una MER para un trellis parcial se deduce de la figura 4.12 y es:

$$MER_k^m = E(MT_{k+1}^{0,f(0,m)} + MER_{k+1}^{f(0,m)}, MT_{k+1}^{1,f(1,m)} + MER_{k+1}^{f(1,m)}) \tag{4.8}$$

Definiendo una métrica de camino regresiva (MCR) de una transición desde un estado m en el tiempo k hasta un estado anterior b(i,m) en el tiempo k-1 producto de una entrada i como $MCR_k^{i,m}$, se tiene:

$$MCR_k^{i,m} = MER_k^m, MT_k^{i,m} \tag{4.9}$$

Por tanto la ecuación (4.8) puede ser representada usando MCR:

$$MER_k^m = E(MCR_{k+1}^{0,f(0,m)}, MCR_{k+1}^{1,f(1,m)})$$

$$MER_k^m = \min(MCR_{k+1}^{0,f(0,m)}, MCR_{k+1}^{1,f(1,m)}) + f(|MCR_{k+1}^{0,f(0,m)} - MCR_{k+1}^{1,f(1,m)}|) \tag{4.10}$$

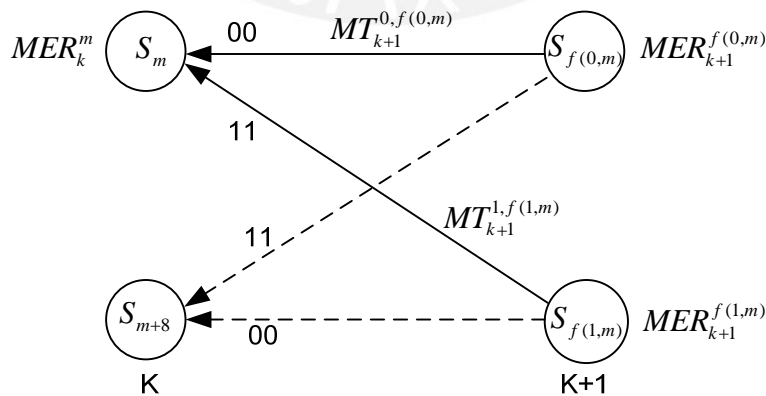


Figura 4.12. Trellis parcial para el calculo de MER_k^m

En la ecuación, $f(i,m)$ es el posible siguiente número de estado para el estado actual m debido al bit de entrada i . Notar que las mismas métricas de transición son utilizadas tanto para los cálculos de la MEP y la MER. La función de ajuste $f(z)$ es la misma que para el cálculo de la MEP, por tanto la misma TDB se utiliza.

De la ecuación (4.5) y (4.10), se puede decir que las unidades MER y MEP pueden tener exactamente el mismo circuito como estructura, con excepción de la memoria RAM que almacena los estados $f(i,n)$ s. La lógica de control es la encargada de decidir correctamente las direcciones $MER_{k+1}^{f(0,m)}$ y $MER_{k+1}^{f(1,m)}$ para calcular MER_k^m en la estación k desde la MER RAM. De acuerdo al trellis parcial de la figura 4.2, dos nuevos MERs con índices de estado m y $m+8$ en el tiempo k pueden ser calculados leyendo dos MERs consecutivos y dos MTs consecutivos de las RAMs en el tiempo $k+1$. Posteriormente la unidad de control debe generar las siguientes direcciones de lectura y escritura sobre la MER RAM.

Tabla 4.2 secuencia de direcciones para la lectura y escritura en la MER RAM.

Dirección de lectura para $k+1$	0	1	2	3	4	5	6	7	9	8	11	10	13	12	15	14
Dirección de escritura para k	0	8	1	9	2	10	3	11	12	4	13	5	14	6	15	7

Puesto que solo cuatro métricas de transición son almacenadas en las MT RAMs, las MTs tienen que ser leídas de las dos RAMs basadas en los símbolos esperados 'ip' para poder desarrollar el cálculo de la MER. Cuando la MEP a calcular es MER_k^m , los símbolos a esperar de las dos transiciones que parten del estado m pueden ser leídas desde las ROMs que almacenan los bits p de paridad como se muestra en la figura 4.3. Cuando $i='0'$, la salida p de la ROM1 sirve como la dirección para leer una MT de la memoria MT RAM0 en la estación $k+1$. la MT dada cuando $i='1'$ es leída de la MT RAM2 de la estación $k+1$ basada en la salida de la ROM2. Todos estos pasos son similares a los realizados para el cálculo de la MEP, en donde cada par de

transiciones terminadas en los estados m y $m+8$ tienen exactamente las mismas MTs. Por lo tanto las MTs partiendo desde la S_0 hasta la S_7 son necesarias para ser leídas y poder calcular la MEP correspondiente. La operación de lectura de las MTs desde las MT RAMs se muestra en la figura 4.13:

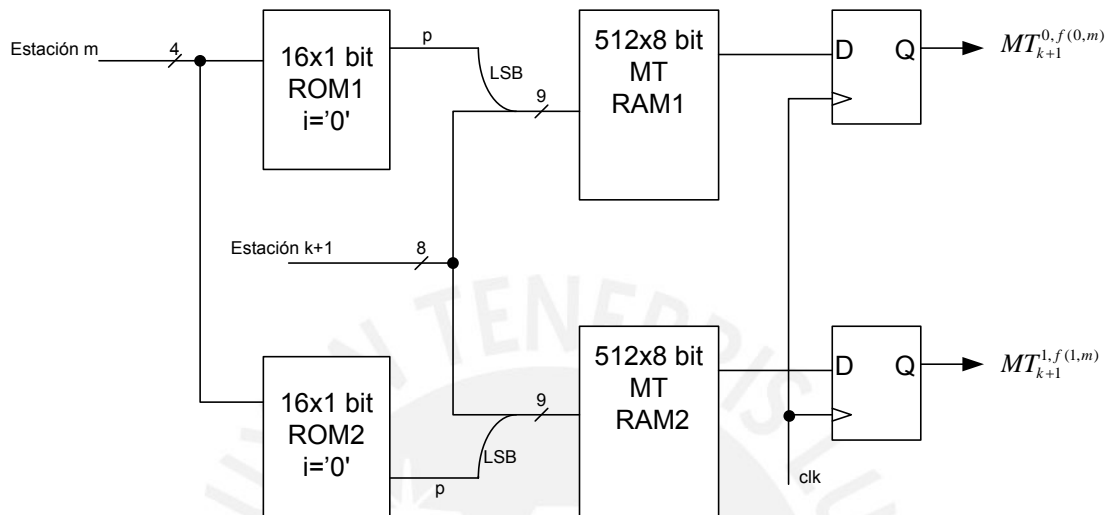


Figura 4.13: Operación de lectura de las MT RAMs para el cálculo de la MER.

Puesto que en el turbo decodificador completo, existen dos componentes decodificadores que hacen uso de las mismas estructuras, se tiene que la inicialización para el decodificador1 esta dada por: $MER_{255}^0 = 0$ y $MER_{255}^m = 255$, $m = 1, 2, \dots, 15$. mientras que como para el decodificador2 los datos de entrada están intercalados, no se puede asegurar que el último esta sea S_0 , por tanto todos los valores iniciales para la MER para la estación 255 será igual a 0.

Finalmente el tipo de RAM utilizada para almacenar las MERs es idéntica a las usadas para el almacenamiento de las MEPs. Por tanto tienen la misma organización de direcciones.

4.2.4 Módulo de cálculo de la Razón logarítmica de Probabilidad (RLP)

El cálculo de la Razón logarítmica de Probabilidad RLP_{k+1} de los bits de información estimados para la estación $k+1$ se rige por la siguiente ecuación:

$$RLP_{k+1} = RLP_{(k+1)0} - RLP_{(k+1)1} \quad (4.11)$$

$$RLP_{(k+1)0} = E_{m=0}^{15} (MEP_k^m + MT_{k+1}^{0,f(0,m)} + MER_{k+1}^{0,f(0,m)}) \quad (4.12)$$

$$RLP_{(k+1)1} = E_{m=0}^{15} (MEP_k^m + MT_{k+1}^{1,f(1,m)} + MER_{k+1}^{1,f(1,m)}) \quad (4.13)$$

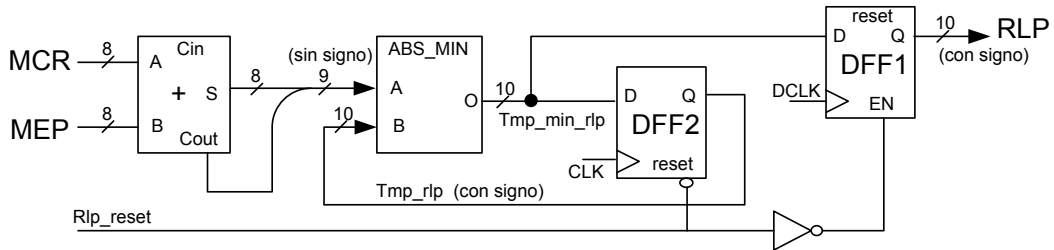
Donde $RLP_{(k+1)0}$ es la probabilidad a posteriori de que cada bit de información i sea '0' en el tiempo $k+1$, y $RLP_{(k+1)1}$ de que cada bit de información i sea '1' en el tiempo $k+1$.

Notar que la operación-E involucra 16 términos. Con propósitos de ilustración, se usa la métrica de razón logarítmica de probabilidad (MRLP) para expresar el cálculo de la $RLP_{(k+1)i}$:

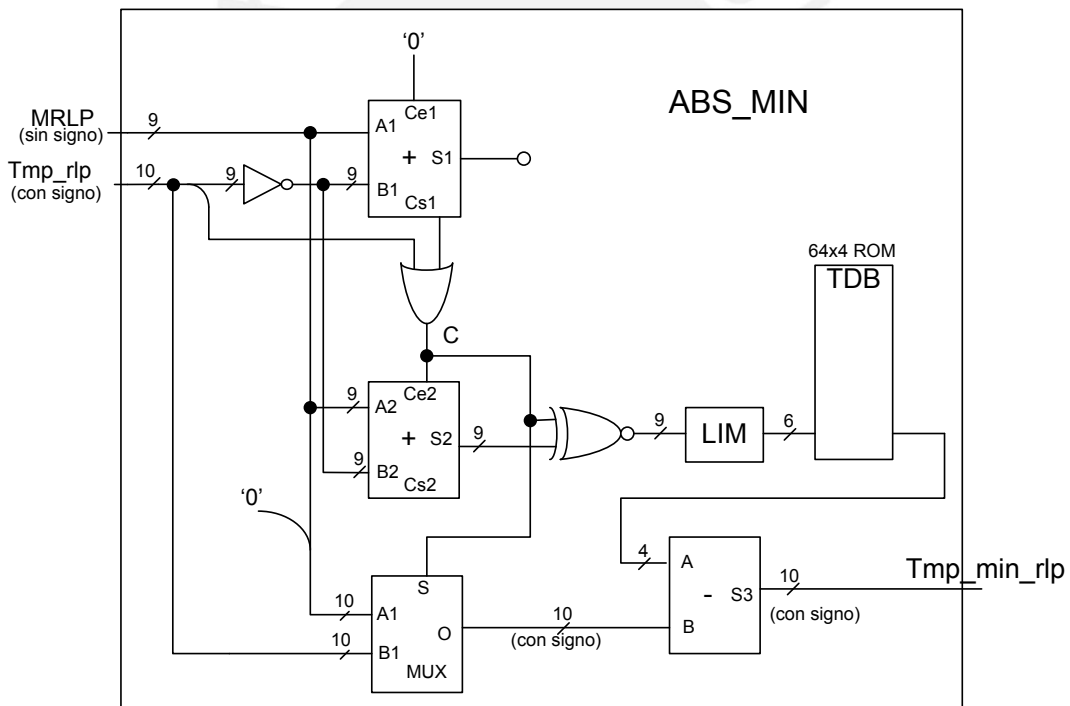
$$MRLP_{k+1}^{i,f(i,m)} \equiv MEP_k^m + MT_{k+1}^{i,f(i,m)} + MER_{k+1}^{i,f(i,m)} \equiv MEP_k^m + MCR_{k+1}^{i,f(i,m)} \quad (4.14)$$

El cálculo de la MRLP involucra una adición de una MCR y una MEP. Dos métricas de camino reversas $MCR_{k+1}^{0,f(0,m)}$ y $MCR_{k+1}^{1,f(1,m)}$ están disponibles conforme se va calculando la MER_k^m . Por tanto, solo una MEP_k^m del estado m en la estación k necesariamente es leída de la MEP RAM en cada ciclo del reloj CLK. El número de estado m sigue la secuencia 0, 8, 1, 9..., 7, 15, el cual sigue el mismo orden del cálculo de la MER. Como las dos MCRs se obtienen al mismo tiempo, $MRLP_{k+1}^{0,f(0,m)}$ y $MRLP_{k+1}^{1,f(1,m)}$ se calculan en paralelo en cada ciclo de reloj CLK. Por lo tanto se utilizan

dos módulos BRLP para calcular las dos RLPs y desarrollar las operaciones-E separadamente. Uno de los dos módulos se describe en la figura 4.14.



(a) diagrama bloque general



(b) circuito de cálculo de la diferencia y mínimo RMP

Figura 4.14: Módulo de Cálculo para la RLP

Como se menciona en la sección 3.4.3, La Operación E con múltiples términos puede ser calculada mediante el cálculo recursivo de dos términos por vez. (Ecuación 3.28). Por tanto, La “operación-E” del cálculo de la RLP se presenta en la figura 3.16b es

desarrollado por medio de un módulo que es similar al usado para el cálculo de la MEP (figura 4.7). Tan pronto como la MRLP esta disponible se compara con la MRLP almacenada en el flip-flop DFF2. El módulo ABS_MIN selecciona el menor de ellos, y ajusta el MRLP seleccionado de acuerdo a $f(z)$ (TBD figura 4.5) y este dato intermedio es almacenado nuevamente en DFF2 en cada ciclo de reloj CLK. Por tanto debido a que se tienen 16 estados, de las 16 MRLP que se calculan se obtiene solo un RLP para una estación $k+1$, siendo esta la menor de entre las 16. Es por ello que se hace uso del flip-flop DFF1 el cual se actualiza cada 16 ciclos de reloj CLK. Para efectos ilustrativos se muestra la figura 4.15 en la que se muestra el diagrama de tiempos para el cálculo de la RLP. Notar que el valor máximo para el RLP es 1023 (10 bits data).

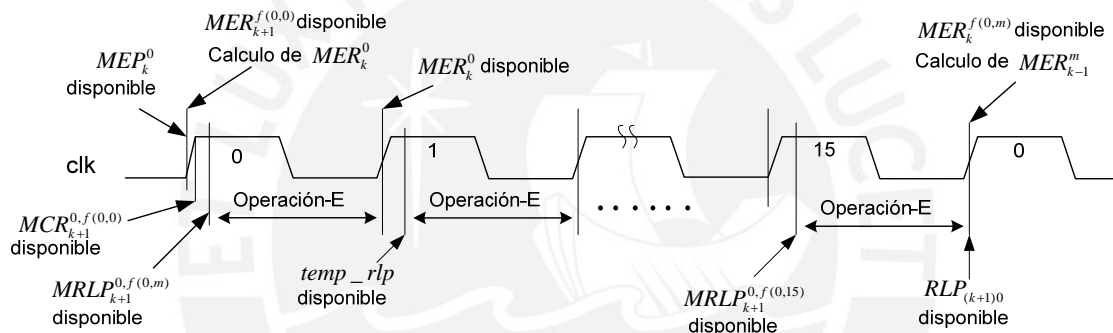


Figura 4.15 Diagrama de tiempos para el cálculo de las RLPs

Finalmente cuando $RLP_{(k+1)0}$ y $RLP_{(k+1)1}$ están determinados y disponibles cada 16 ciclos de reloj CLK, se resta $RLP_{(k+1)1}$ de $RLP_{(k+1)0}$ basando en la ecuación 4.11 para obtener $RLP_{(k+1)}$. Tener en cuenta que las RLPs son calculadas al mismo tiempo que las métricas de regresivas por tanto el orden en que los datos son entregados por el decodificador son inversos y por tanto esto se debe tomar en cuenta para el diseño de los intercalador y de-intercalador.

4.2.5 Módulo de control para el decodificador log_MAP.

El módulo de control que se desarrolló para la presente tesis genera señales de reset así como también de lectura y escritura de las memorias RAM. Sin embargo no es autónoma, y depende de datos ingresados manualmente con fines de hacer las simulaciones correspondientes de los distintos bloques básicos diseñados. Las señales correspondientes a este módulo de control se encuentran gráficamente expresadas en el anexo D.3 para propósitos de análisis en posteriores investigaciones. Por tanto este controlador es percibido como un control primitivo solo con la finalidad de obtener una simulación y comprobar que los bloques funcionan de acuerdo a lo esperado para el proceso de descodificación.

Cada proceso de decodificación está conformado por dos etapas: La primera etapa se encarga del cálculo tanto de la métrica de transición (MT) como de la métrica progresiva (MEP) para cada estación, en la segunda etapa se controla el cálculo de la métrica regresiva (MER) y la razón logarítmica de probabilidad (RLP) correspondiente a cada estación.

RESULTADOS

En la presente sección se resumen los resultados obtenidos tanto de diseño como de simulación, así como también los diagramas de tiempo de ejecución y consumos de espacio en el FPGA.

Resultados de simulación de bloques MT, MEP, MER y RLP.

El Anexo D.1 presenta la simulación obtenida del funcionamiento de los módulos de la métrica de transición (MT) y la métrica progresiva según lo expresado en las ecuaciones (3.33 y 3.37) :

$$A_k^m = E(\Gamma_k^{0,m} + A_{k-1}^{b(0,m)}, \Gamma_k^{1,m} + A_{k-1}^{b(1,m)}) \quad (3.33)$$

$$\Gamma_k^{i,m} = -(x_k Lc(x_k') + p_k Lc(p_k') + x_k La(x_k) - C_k) \quad (3.37)$$

En las cuales $\Gamma_k^{0,m} = \text{MTK_0}$, $\Gamma_k^{1,m} = \text{MTK_1}$, $A_k^m = \text{MEP_K}$, $A_{k-1}^{b(0,m)} = \text{MEP_0_km1}$

$A_{k-1}^{b(1,m)} = \text{MEP_1_km1}$, $x_k Lc(x_k') = \text{"sys"}$, $p_k Lc(p_k') = \text{"par"}$ y $x_k La(x_k) = \text{"La"}$. Cuyos

valores obtenidos para 16 ciclos de reloj se muestra en la Tabla-1 para las entradas "par"=1 y "sys"=2.

Tabla -1

CLK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
MTK_0	3			1			3		1		3				1	
MTK_1	0			1			0		1		0				1	
MEP_0	224		255			248				255				248		
MEP_1		249				212		255		248		255		236		255

Adicionalmente se comprueba que dichos cálculos siguen los patrones de transición de estados mostrados en la figura 3.8 y la Tabla-2 para la lectura y escritura de las memorias de manera sincronizada.

Tabla-2

Dirección de lectura para k+1	0	1	2	3	4	5	6	7	9	8	11	10	13	12	15	14
Dirección de escritura para k	0	8	1	9	2	10	3	11	12	4	13	5	14	6	15	7

El Anexo D.2 presenta la simulación obtenida del funcionamiento conjunto de los módulos MT, MEP y la métrica de transición regresiva (MER), de donde se obtiene la Tabla-3 y cumple con lo expresado en la ecuación (3.35)

Tabla-3

CLK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
MTK_0	57	54			57			54			57			54		
MTK_1	68	0			68			0			68			0		
MEP_K	138	158	230	153	195	222	152	205	234	181						
MER_K	79	64		79		64		79			255					

$$B_k^m = E(\Gamma_{k+1}^{0,f(0,m)} + B_{k+1}^{f(0,m)}, \Gamma_{k+1}^{1,f(1,m)} + B_{k+1}^{f(1,m)}) \quad (3.35)$$

En la cual $B_k^m = \text{MER_K}$. Notar que se muestra además la transición de un modo de cálculo progresivo a un modo de cálculo regresivo, por lo cual tanto las direcciones de memoria para la MEP siguen el mismo orden que las direcciones de memoria para la MER desde ese punto en adelante. Finalmente se sigue manteniendo el mismo patrón mostrado en la figura 3.8 y la Tabla-2

El Anexo D.3 presenta la simulación obtenida del funcionamiento conjunto de todos los bloques MT, MEP, MER y la relación logarítmica de probabilidad (RLP), según lo expresado en la ecuación (3.38)

$$L_k = \underset{m=0}{E} (A_{k-1}^{b(0,m)} + \Gamma_k^{0,m} + B_k^m) - \underset{m=0}{E} (A_{k-1}^{b(1,m)} + \Gamma_k^{1,m} + B_k^m) \quad (3.38)$$

En la cual se toman valores de prueba en las memorias de la MER para poder comprobar que la RLP denotada en la ecuación como L_k (representada como RMP1_k en el anexo) es coherente cada 16 ciclos de reloj. Notar que a diferencia de los valores obtenidos para MT, MEP y MER la RLP tiende a tomar valores cercanos al límite superior FF o al límite inferior 00 puesto que esta respuesta es la estimación final con el efecto del ruido ya corregido por el algoritmo Log-MAP. Dicho resultado para una iteración se muestra en la Tabla-4

Tabla-4

CLK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
par	-43															
sys	33															
MTK_0	32		48				32		48		32				48	
MTK_1	32		0				32		0		32				0	
MEP_K	249	241	255		248	241	152		205		234		91			
MER_K	79		64		79		64		79				255			
RMPL_K	FC															

Donde RMPL_K es la etiqueta para la RLP. Anexo D.3

Especificaciones de hardware del los bloques diseñados.

Producto de las compilaciones realizadas se obtuvieron los resultados mostrados en la

Tabla-5

Resultados de Compilación por bloques			
	Codificador	Intercalador	Decodificador
Software	Quartus II		
Versión	5.0 build 158 04/26/05 SJ Web Edition		
Familia	Stratix		
Dispositivo	EP1S10F484C5		
Tipo	Codificador RSC (3,1,5)		Decodificador Log-MAP
Rango	(1/3)		
Dimensión de intercalado	256 bits (ver anexo B)		
Elementos Lógicos	42	21	682
Bits de memoria	1,024	4,096	74,928
Relojes	clk= 160 ns	clk = 160 ns	clk=10 ns, dclk=160ns
Frecuencia de operación	6.25 Mbps	6.25 Mbps	100 Mbps
Periodo de inicialización	160 ns	160 ns	520 ns
Tiempo de respuesta por trama			82.38 us
Tamaño de trama			256 bytes

Tabla-5

CONCLUSIONES

Como respuesta a los objetivos propuestos para esta Tesis, se concluye que:

Sistemas matemáticos complejos como los Turbo Códigos pueden ser descritos e implementados sobre FPGAs partiendo de sus bloques básicos. Sin embargo debido a la complejidad computacional estas implementaciones estarían limitadas a una máxima velocidad de 6.5 Mbps. Consecuentemente la aplicación de este sistema de corrección de errores en transmisiones inalámbricas infrarrojas es viable y a la vez un buen punto de partida para futuros desarrollos.

A mayor complejidad de Turbo de-codificación, mayor el número de bancos de memoria a utilizar producto de las iteraciones. Por lo tanto los “bits de memoria” constituyen una variable fundamental en el diseño de Turbo Códigos en FPGAs.

La información algorítmico matemática recopilada y estudiada abarca los fundamentos básicos y necesarios para iniciar el diseño de sistemas de correcciones de errores basados en Turbo códigos.

RECOMENDACIONES

Partiendo del diseño de los bloques presentados en la presente Tesis, se recomienda implementar un sistema completo e integrado para la turbo codificación y de-codificación con la finalidad de demostrar de forma aplicativa en que factor se mejoraría la relación señal a ruido en una transmisión vía infrarrojo.

7 FUENTES

- [1] T. Rappaport, *Wireless Communications: Principles and Practice*. New Jersey: Prentice-Hall Inc., 1996
- [2] J. E. Padgett, C. G. Gunther, and T. Hattori, "Overview of wireless personal communication," *IEEE Communication Magazine*, vol. 33, pp. 28-41, Jan. 1995.
- [3] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. J.*, vol. 27, pp.378-423 y 623-656, 1948.
- [4] F.R. Gferller and U. Bapst, "Wireless In-House Data Communication via Diffuse Infrared Radiation", *Proceedings of IEE*, vol. 67, no.11, pp. 1474-1486, Noviembre 1979.
- [5] V. Pohl, V. Jungnickel, and C. von Helmont, "A Channel Model for Wireless Infrared Communication", *PIMRC 2000: 11th IEEE International Symposium on Personal, Indoor and Radio Communications*, vol. 1, pp. 297-303, 2000.
- [6] E. Simova and M. Kavehrad, "Light Shaping Diffusers for Indoor Wireless Infrared Communication via Holographic approach", *Proceeding of the International Society for Optical Engeneering conference*, vol.2689, pp. 284-291, February 1996.
- [7] J. R. V. Alvarez, F. J. L. Hernández, A. S. Galdón, R. P. Jiménez, and J. A. R. Borges, "infrared wireless DSSS system for indoor data communication links", *Proceedings of SPIE- Optical Wireless Communications II*, vol. 3850, pp. 92-99, September 1999.
- [8] P.D Brown, L. S. Moore, and D. C. York. "Low Power Optical Transceiver for portable Computing Devices, " *U.S. Patent N°. 5,075,792* , Hewlett-Packard Company, Dec 24, 1991.
- [9] IrDA "Serial Infrared Link Access Protocol (IrLAP)," v.1.1, June 1996.
- [10] IrDA "Link Management Protocol," v1.1 Jan 1996
- [11] IrDA "Serial Infrared Physical Layer Link Specification," v.1.3 Oct 1998
- [12] IrDA, "IrCOMM; Serial and Parallel Port Emulation over IR (Wire Replacement)," v.1.0 Nov 1995
- [13] IrDA, "LAN Access Extension for Link Management Protocol IrLAN," v.1.0 July 1997

- [14] IrDA “IrTran-P (Infrared Transfer Picture) Specification,” v.1.0 July 1997
- [15] L. Arnone, C. Gayoso, C. González, J. C. Garcia y J. Castiñeira ,“Diseño de un decodificador Viterbi para ser Utilizado en Enlaces Infrarrojos” IX Taller IBERCHIP, La Habana, Cuba; 26 a 28 de Marzo de 2003
- [16] S. Ten Brink “Rate one-half code for approaching the Shannon limit by 0.1 dB” Electronic Letters, Vol 36 N°15, pp 1293-1294, July 2000,
- [17] C. Berrou, A. Glavieux and P. Thinitimajshima, “Near Shannon Limit error-correcting coding and decoding: Turbo-codes”, ICC 1993, Geneva, Switzerland, pp. 1064-1070, May 1993.
- [18] S. A. Barbulescu, W. Farell, P.Gray, and M. Rice, “Bandwidth Efficient Turbo Coding for High Speed Mobile Satellite Communications”, in Proc. International Symposium on Turbo codes and Related Topics, (Brest, France), pp 119-129, Sep. 1997.
- [19] H. Feldman and D. V. Ramana , “An introduction to Inmarsat’s New Mobile Multimedia Service”, The Sixth International Mobile Satellite Conference, Ottawa, pp. 226-229 , June 1999
- [20] L. R. Bahl and J. Cocke and F. Jelinek and J. Raviv “Optimal Decoding of linear Codes for Minimising Symbol Error Rate” IEEE Transactions on information theory, vol20, pp.284-287, March 1974.
- [21] P. Robertson and E. Villeburn and P. Hoher, “ A comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain” Proceedings of the international Conference on Communications, pp 1009-1013, June 1995.
- [22] R. Pyndiah, “Iterative decoding of product codes: Block turbo codes” in international Symposium on turbocodes and related topics, pp.71-79 September 1997.
- [23] P. Jung and M. Nasshan, Performance evaluation of turbo-codes for short frame transmission systems” IEEE electronics Letters, vol. 30 pp. 111-112 January 1994.
- [24] A. Barbulescu and S. Pietrobon “Interleaver design for turbo codes” IEEE Electronics letters, pp.2107-2108, December 1994.
- [25] J. P. Woodard, L. Hanzo : “ Comparative Study of Turbo Decoding Techniques: An Overview, IEEE Transactions on Vehicular Technology, Nov 2000, Vol 49, N° 6 pp. 2208-2234.

- [26] Kena Aksoy y Umit Aygolu, "NEw Punctured Turbo Codes", *International Journal of Electronics and Communications*, 2003; 57, 3
- [27] "Hardware implemtation of Non- Binary Trubo Code for DVB/RCS ", Yimin Du , Thesis for the Degree of master of Apllied Science at Concordia Univesity. February 2003.
- [28] "VHDL Implementation of Turbo Codec" , Yanhui Tong , Thesis , University of Ottawa, april 2003.
- [29] Diseño de un Decodificador Analógico Basado en el Algoritmo MAP para el código de corrección de errores Binario (5,2) , J. Lagos Benites, E. Bances Purizaca, I. Lopez Zegarra, X Taller IBERCHIP Cartagena de Indias, Colombia; 10 a 12 de Marzo de 2004.
- [30] A VLSI Implemtation of Punctured Convolutional Codes , E.Garcia, M. E. Guzman, X Taller IBERCHIP Cartagena de Indias, Colombia; 10 a 12 de Marzo de 2004
- [24] Turbo Encoder & Decoder , Megacore Function User Guide , ALTERA , junio 2004.
- [31] Y. Wu, "*Implementation of parallel and serial concatenated codes.*" PhD dissertation, Virginia Polytechnic Institute and State University, Apr.2000.
- [32] William J. Ebel, "Turbo-codes: Algorithms and implementation," contract report submitted to Texas Instruments.
- [33] C. heegard. S. B. Wicker, Turbo Coding, Kluwer Academic Publishers, 1999.
- [34] Joseph M. Kahn and John R. Barry, "Wireless Infrared Communications" , *Proceedings of IEE*, vol. 85, no.2, pp. 265-298, Noviembre 1997.
- [35] P. Robertson and E. Villeburn and P. Hoher, " Optimal and Sub-Optimal maximum a posteriori algorithm suitable for turbo decoding" *European Trans. On Telecommun.* Vol.8, pp 119-125, Mar/Apr1997.
- [37] O. M. Collins, "The subtleties and intricacies of building a constraint length 15 convolutional decoder" *IEEE Trans. Commun.*, COM-40, pp. 1810-1819, 1992.
- [38] S.S. Pietrobon, "Implementation and performance of a turbo/MAP decoder," in *Int. J. Satell. Commun.*, vol.16, pp. 23-46, 1998
- [39] J. L. Tonfat Seclen, J. Benavides Aspiazu, J. L. Lagos Benites "Diseño de un Intercalador Programable de 8 Bits para la Decodificación Analógica de Turbo Códigos"