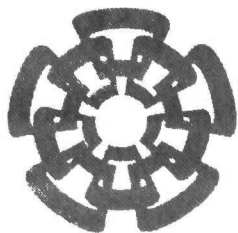


xx(13 1343.1)



CINVESTAV

Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Módulo codificador FEC en banda para elementos de red SONET/SDH a 2.5 Gbps y 10 Gbps

**CINVESTAV
IPN
ADQUISICION
DE LIBROS**

Tesis que presenta:

Baldemar Irineo Carrasco

para obtener el grado de:

Maestro en Ciencias

en la especialidad de:

Ingeniería Eléctrica

**CINVESTAV I.P.N.
SECCION DE INFORMACION
Y DOCUMENTACION**

Directores de Tesis

Dr. Deni Librado Torres Román

Dr. Arturo Veloz Guerrero

Guadalajara, Jalisco, Septiembre de 2006.

CLASIF. TK165.68 J75 2006
ADQUIS.: SSI-414
FECHA: 16-V-2007
PROCED. Don. 2007

ID: 130765-1001

Módulo codificador FEC en banda para elementos de red SONET/SDH a 2.5 Gbps y 10 Gbps

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Baldemar Irineo Carrasco
Ingeniero Electrónico

Instituto Tecnológico de Minatitlán 1996-2000

Becario de CONACYT, expediente no. 165077

Directores de Tesis

Dr. Deni Librado Torres Román
Dr. Arturo Veloz Guerrero

CINVESTAV del IPN Unidad Guadalajara, Septiembre de 2006.

Resumen

Los estándares SONET y SDH son utilizados en todo el mundo en redes de transmisión de datos a través de fibra óptica. En ellos se definen los formatos y velocidades de transmisión que van desde los 51.84 Mbps hasta los 40 Gbps.

Velocidades de transmisión de datos en el orden de Gbps acentúan las imperfecciones de las fibras ópticas y en consecuencia se induce un mayor número de errores en los datos que son transmitidos. La tasa de errores de los enlaces ópticos se incrementa y se degrada la calidad del servicio.

La unión internacional de telecomunicaciones (ITU) ha empleado la técnica digital conocida como corrección de errores hacia adelante o FEC, por sus siglas en inglés: *Forward Error Correction*, para formular una función llamada FEC en banda que permite mejorar la calidad de servicio en los enlaces ópticos.

FEC en banda emplea un código corrector de errores del tipo BCH para proteger los datos transmitidos y así mantener la tasa de errores en un nivel aceptable. Esta función ha sido definida para velocidades de transmisión a 2.5 Gbps, 10 Gbps y 40 Gbps.

La función FEC en banda ha sido anexada a la recomendación ITU-T G.707, en donde se definen las generalidades sobre la interfaz de nodo de red para SDH.

En este trabajo de tesis se presenta la especificación, el diseño, la verificación funcional y la síntesis lógica de un módulo digital que implementa la función FEC en banda para elementos de red SONET/SDH a 2.5 Gbps y 10 Gbps.

Agradecimientos:

A mis padres y mis hermanos por su amor y comprensión

A los profesores del área de Telecomunicaciones por su dedicación y sus enseñanzas

A los Doctores Deni Torres y Arturo Veloz por su confianza y apoyo

Al CONACYT por el apoyo económico

ÍNDICE

1	INTRODUCCIÓN	1
1.1	Planteamiento del problema.....	1
1.2	Objetivos	2
1.2.1	Objetivo general.....	2
1.2.2	Objetivos específicos	2
1.3	Implementación	3
1.4	Estilo de redacción	3
2	SONET / SDH	4
2.1	Introducción a SONET / SDH.....	4
2.2	Estructura de una trama SONET / SDH.....	4
2.3	FEC en banda	7
2.3.1	Descripción del código	7
2.3.2	Subfilas y <i>grupos</i>	8
2.3.3	Bytes P1	9
2.3.4	Bytes Q1	11
2.3.5	Bytes no incluidos en el proceso de codificación FEC en banda.....	11
2.3.6	Estados operacionales del módulo codificador FEC en banda	11
2.3.7	Compensación de los bytes B2	12
2.3.8	Generación de la señal MSF-AIS	13
2.3.9	Desempeño de FEC en banda	14
3	NOCIONES TEÓRICAS SOBRE CÓDIGOS CORRECTORES DE ERRORES ...	15
3.1	Introducción.....	15
3.2	Introducción al álgebra de campos de Galois	15
3.2.1	Definición de Campo	15
3.2.2	Inverso aditivo e inverso multiplicativo	16
3.2.3	Campos primos y extensiones de campos.....	16
3.2.4	Orden de un elemento de un campo finito	17
3.2.5	Elemento primitivo de un campo finito	17
3.2.6	Aritmética de campos binarios	17
3.2.7	Construcción de campos de Galois $GF(2^m)$	18
3.2.8	Polinomio Mínimo	19
3.2.9	Conjugados de un elemento de campo.....	19
3.3	Códigos de Bloque Lineales	20
3.3.1	Distancia mínima de un código de bloque.....	21
3.4	Códigos Cíclicos.....	21
3.4.1	Definición de Código Cíclico	22
3.4.2	Representación Polinomial	22
3.4.3	Polinomio generador de un código cíclico $C(n, k)$	22
3.4.4	Forma sistemática de un código cíclico	22
3.4.5	Circuito para implementar la codificación en un código cíclico (n, k)	23
3.4.6	Códigos Cíclicos Acortados.....	24
3.5	Códigos BCH	25
3.5.1	Descripción de los códigos BCH binarios	25
3.5.2	Polinomio generador de códigos BCH binarios.....	25

3.5.3	Propiedades de los códigos BCH binarios	26
3.5.4	Códigos BCH Binarios de longitud 63	26
3.5.5	Código BCH binario (8191, 8152).....	28
4	ESPECIFICACIÓN DE REQUERIMIENTOS.....	29
4.1	Introducción.....	29
4.2	Módulo codificador FEC en banda.....	29
4.2.1	Propósito del hardware	29
4.2.2	Restricciones principales del módulo	29
4.2.3	Suposiciones y dependencias.....	29
4.2.4	Requerimientos funcionales del módulo codificador.....	30
4.2.5	Requerimientos no funcionales del módulo codificador.....	32
5	ARQUITECTURA Y DISEÑO	34
5.1	Introducción.....	34
5.2	Señales de entradas y de salida del módulo codificador	34
5.3	Arquitectura del módulo codificador	35
5.4	Insertor de la señal FSI.....	38
5.5	Insertor de ceros.....	39
5.6	Calculador de bytes P1.....	40
5.6.1	Microarquitectura del bloque Calculador de bytes P1	40
5.6.2	Calculador de bytes P1 para el Grupo 1.....	43
5.6.3	Calculador de bytes P1 para el Grupo 2, Grupo 3 y Grupo 4	46
5.6.4	Registro de bytes P1 para el Grupo 1, Grupo 2, Grupo 3 y Grupo 4	48
5.6.5	Multiplexor del Grupo 1, Grupo 2, Grupo 3 y Grupo 4.....	49
5.6.6	Multiplexor de Grupos.....	50
5.6.7	Control del calculador de bytes P1	51
5.7	Buffer de datos.....	66
5.8	Insertor de bytes P1.....	67
5.9	Calculador de bytes B2	70
5.9.1	Máquina de estados para el cálculo del código de paridad B2.....	71
5.9.2	Máquina de estados para el envío del código de paridad B2 calculado	72
5.9.3	Señal de indicación de alarma MSF-AIS	72
5.10	Compensador de bytes B2	73
5.10.1	Señal de indicación de alarma MSF-AIS	74
5.11	Multiplexor de datos de salida	74
5.12	Bloque de control del módulo codificador.....	76
5.12.1	Señales de entrada que controlan al módulo codificador.....	77
5.12.2	Máquina de control	77
6	VERIFICACIÓN FUNCIONAL	79
6.1	Introducción.....	79
6.2	Verificación funcional.....	79
6.3	Casos de prueba del módulo codificador FEC en banda.....	81
6.3.1	Identificador de los casos de prueba	81
6.3.2	Descripción de los casos de prueba	81

6.3.3	Subcasos de prueba.....	84
6.4	Síntesis.....	85
7	RESULTADOS	86
7.1	Estadísticas de diseño de los bloques.....	86
7.2	Estadísticas de Verificación.....	87
7.3	Reporte de síntesis.....	88
8	CONCLUSIONES	91
	REFERENCIAS	93
	ANEXOS	94

ÍNDICE DE FIGURAS

FIGURA 2.1	ESTRUCTURA DE UNA TRAMA SONET STS-1.....	5
FIGURA 2.2	TARA DE TRANSPORTE (TOH) DE UNA TRAMA BÁSICA STS-1.....	6
FIGURA 2.3	ESTRUCTURA DE UNA TRAMA SONET STS-3.....	6
FIGURA 2.4	TARA DE TRANSPORTE DE UNA TRAMA SONET STS-3.....	7
FIGURA 2.5	BITS DE UNA FILA EN UNA TRAMA SONET / SDH ARREGLADOS PARA PROPÓSITOS DE FEC EN BANDA.....	8
FIGURA 2.6	GENERACIÓN DE LOS BYTES P1, A PARTIR DE LOS BITS DE CHEQUEO DE PARIDAD A_N	9
FIGURA 2.7	TARA DE SECCIÓN SOH (SDH) Y SU SISTEMA DE COORDENADAS S (A, B, C).....	10
FIGURA 2.8	MODELO DE REFERENCIA DE FEC EN BANDA.....	13
FIGURA 2.9	BER DE SALIDA TEÓRICO CONTRA BER DE ENTRADA (BCH-3).....	14
FIGURA 3.1	FORMA SISTEMÁTICA DE UNA PALABRA DE CÓDIGO.....	21
FIGURA 3.2	CIRCUITO DE CODIFICACIÓN PARA UN CÓDIGO CÍCLICO (N, K).....	24
FIGURA 5.1	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL MÓDULO CODIFICADOR.....	34
FIGURA 5.2	ARQUITECTURA DEL MÓDULO CODIFICADOR.....	36
FIGURA 5.3	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>INSERTOR DE LA SEÑAL FSI</i>	38
FIGURA 5.4	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>INSERTOR DE CEROS</i>	39
FIGURA 5.5	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>CALCULADOR DE BYTES P1</i>	40
FIGURA 5.6	MICROARQUITECTURA DEL BLOQUE <i>CALCULADOR DE BYTES P1</i>	41
FIGURA 5.7	FLUJOS DE DATOS ENTRANTES A LOS BLOQUES <i>CALCULADOR DE BYTES P1</i> E <i>INSERTOR DE BYTES P1</i> Y CONTENIDO DE LOS REGISTROS INTERNOS DE LOS BLOQUES <i>CALCULADORES DE BYTES P1</i>	42
FIGURA 5.8	CONTENIDO DE LOS BLOQUES <i>REGISTROS DE BYTES P1</i>	43
FIGURA 5.9	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>CALCULADOR DE BYTES P1 PARA EL GRUPO 1</i>	43
FIGURA 5.10	MICROARQUITECTURA DEL BLOQUE <i>CALCULADOR DE BYTES P1 PARA EL GRUPO 1</i>	44
FIGURA 5.11	DISEÑO DEL BLOQUE <i>A_N BITS GENERATOR SLICE 1</i> PARA EL GRUPO 1.....	45
FIGURA 5.12	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>CALCULADOR DE BYTES P1 PARA EL GRUPO K (2 ≤ K ≤ 4)</i>	46
FIGURA 5.13	MICROARQUITECTURA DEL BLOQUE <i>CALCULADOR DE BYTES P1 PARA EL GRUPO K</i>	47
FIGURA 5.14	DISEÑO DEL BLOQUE <i>A_N BITS GENERATOR SLICE 1</i> PARA EL GRUPO K (2 ≤ K ≤ 4).....	48
FIGURA 5.15	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>REGISTRO DE BYTES P1 PARA EL GRUPO K (1 ≤ K ≤ 4)</i>	48
FIGURA 5.16	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>MULTIPLEXOR DEL GRUPO K (1 ≤ K ≤ 4)</i>	49
FIGURA 5.17	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>MULTIPLEXOR DE GRUPOS</i>	50
FIGURA 5.18	DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>CONTROL DEL CALCULADOR DE BYTES P1</i>	52

FIGURA 5.19 DIAGRAMA DE TIEMPO DE LAS SEÑALES DE CONTROL DURANTE EL ENVÍO DE BYTES P1 DE LA FILA 9 EN EL MODO DE OPERACIÓN DE 2.5 GBPS	55
FIGURA 5.20 DIAGRAMA DE TIEMPO DE LAS SEÑALES DE CONTROL DURANTE EL ENVÍO DE BYTES P1 DE LA FILA 1 EN EL MODO DE OPERACIÓN DE 2.5 GBPS	56
FIGURA 5.21 DIAGRAMA DE TIEMPO DE LAS SEÑALES DE CONTROL DURANTE EL ENVÍO DE BYTES P1 DE LA FILA 9 EN EL MODO DE OPERACIÓN DE 10 GBPS	58
FIGURA 5.22 DIAGRAMA DE TIEMPO DE LAS SEÑALES DE CONTROL DURANTE LA ESCRITURA DE LOS BYTES P1 DE LA FILA 1 EN LOS BLOQUES <i>REGISTROS DE BYTES P1</i> EN EL MODO DE OPERACIÓN DE 10 GBPS	60
FIGURA 5.23 DIAGRAMA DE TIEMPO DE LAS SEÑALES DE CONTROL DURANTE LA LECTURA DE LOS BYTES P1 DE LA FILA 1 ALMACENADOS EN LOS BLOQUES <i>REGISTROS DE BYTES P1</i> EN EL MODO DE OPERACIÓN DE 10 GBPS	61
FIGURA 5.24 MÁQUINA DE ESTADOS PARA GENERAR LA SEÑAL DE CONTROL <i>SEND_P1_BYTES_GROUP_1</i>	62
FIGURA 5.25 MÁQUINA DE ESTADOS PARA GENERAR LA SEÑAL DE CONTROL <i>SEL_P1_BYTES_GROUP_1</i>	62
FIGURA 5.26 MÁQUINA DE ESTADOS PARA GENERAR LAS SEÑALES DE CONTROL <i>WRITE_P1_BYTE_A38_A26_GROUP_1, WRITE_P1_BYTE_A25_A13_GROUP_1,</i> <i>WRITE_P1_BYTE_A12_A0_GROUP_1, READ_P1_BYTE_A38_A26_STORED_GROUP_1,</i> <i>READ_P1_BYTE_A25_A13_STORED_GROUP_1 Y READ_P1_BYTE_A12_A0_STORED_GROUP_1</i>	63
FIGURA 5.27 MÁQUINA DE ESTADOS PARA GENERAR LA SEÑAL DE CONTROL <i>SEL_MUX_GROUP_1</i>	65
FIGURA 5.28 MÁQUINA DE ESTADOS PARA GENERAR LA SEÑAL DE CONTROL <i>SEL_MUX_GROUPS</i>	65
FIGURA 5.29 DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>BUFFER DE DATOS</i>	66
FIGURA 5.30 MICROARQUITECTURA DEL BLOQUE <i>BUFFER DE DATOS</i>	67
FIGURA 5.31 DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>INSERTOR DE BYTES P1</i>	68
FIGURA 5.32 MICROARQUITECTURA DEL BLOQUE <i>INSERTOR DE BYTES P1</i>	69
FIGURA 5.33 MÁQUINA DE ESTADOS PARA GENERAR LAS SEÑALES DE CONTROL <i>INSERT_BYTES E</i> <i>INSERT_MSF_AIS</i>	69
FIGURA 5.34 DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>CALCULADOR DE BYTES B2</i>	70
FIGURA 5.35 MÁQUINA DE ESTADOS PARA EL CÁLCULO DEL CÓDIGO DE PARIDAD B2	71
FIGURA 5.36 MÁQUINA DE ESTADOS PARA EL ENVÍO DEL CÓDIGO DE PARIDAD B2	72
FIGURA 5.37 DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>COMPENSADOR DE BYTES B2</i>	73
FIGURA 5.38 DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>MULTIPLEXOR DE DATOS DE SALIDA</i>	75
FIGURA 5.39 DIAGRAMA DE SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE DE CONTROL DEL MÓDULO CODIFICADOR	76
FIGURA 5.40 ESTADOS DE LA MÁQUINA DE CONTROL	78
FIGURA 6.1 ESTRUCTURA DE LA VERIFICACIÓN FUNCIONAL	79
FIGURA 6.2 IDENTIFICADOR DE LOS CASOS DE PRUEBA DEL MÓDULO CODIFICADOR FEC EN BANDA	81
FIGURA 6.3 IDENTIFICADOR DE LOS SUBCASOS DE PRUEBA DEL MÓDULO CODIFICADOR FEC EN BANDA	84
FIGURA 7.1 PARTE DE SCRIPT " <i>SYNTHESIS_SCRIPT</i> "	89
FIGURA 7.2 SCRIPT CON COMANDOS DE SÍNTESIS PARA EL BLOQUE <i>INSERTOR DE LA SEÑAL FSI</i>	90

ÍNDICE DE TABLAS

TABLA 2.1 JERARQUÍAS DIGITALES SONET/SDH	4
TABLA 2.2 COORDENADAS A Y B EN DONDE SE INSERTARÁN LOS BYTES P1 (A_N)	10
TABLA 3.1 POLINOMIOS MÍNIMOS DE LOS ELEMENTOS EN $GF(2^6)$	27
TABLA 3.2 CÓDIGOS BCH BINARIOS DE LONGITUD 63, SUS POLINOMIOS GENERADORES Y SUS PARÁMETROS ..	27
TABLA 5.1 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL MÓDULO CODIFICADOR	35
TABLA 5.2 MATRIZ DE SEGUIMIENTO DE REQUERIMIENTOS CONTRA FUNCIONES	37
TABLA 5.3 MATRIZ DE SEGUIMIENTO DE FUNCIONES CONTRA BLOQUES DE DISEÑO	38
TABLA 5.4 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>INSERTOR DE LA SEÑAL FSI</i> ..	39
TABLA 5.5 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>INSERTOR DE CEROS</i>	39
TABLA 5.6 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>CALCULADOR DE BYTES P1</i> ..	40
TABLA 5.7 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADAS Y SALIDAS DEL BLOQUE <i>CALCULADOR DE BYTES P1 PARA EL GRUPO 1</i>	44

TABLA 5.8 SUBCONJUNTOS DE BYTES P1 PARA DIFERENTES VALORES DE LA SEÑAL <i>SEL_P1_BYTES_GROUP_1</i>	46
TABLA 5.9 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>CALCULADOR DE BYTES P1 PARA EL GRUPO K (2 ≤ K ≤ 4)</i>	46
TABLA 5.10 CORRESPONDENCIA ENTRE LAS SEÑALES <i>BYTE_A</i> , <i>BYTE_B</i> , <i>BYTE_C</i> Y <i>BYTE_D</i> , Y EL FLUJO DE DATOS ENTRANTE <i>DATA_WITH_ZEROS</i>	47
TABLA 5.11 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>REGISTRO DE BYTES P1 PARA EL GRUPO K (1 ≤ K ≤ 4)</i>	49
TABLA 5.12 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>MULTIPLEXOR DEL GRUPO K (1 ≤ K ≤ 4)</i>	50
TABLA 5.13 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>MULTIPLEXOR DE GRUPOS</i>	51
TABLA 5.14 FLUJO SELECCIONADO PARA CADA VALOR DE LA SEÑAL DE ENTRADA <i>SEL_MUX_GROUPS</i>	51
TABLA 5.15 DESTINO DE LAS SEÑALES DE SALIDA DEL BLOQUE <i>CONTROL DEL CALCULADOR DE BYTES P1</i>	53
TABLA 5.16 CICLOS DE INSERCIÓN DE BYTES P1	54
TABLA 5.17 VALORES DE LAS SEÑALES DE CONTROL EN LOS DIFERENTES ESTADOS	64
TABLA 5.18 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>BUFFER DE DATOS</i>	67
TABLA 5.19 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>INSERTOR DE BYTES P1</i>	68
TABLA 5.20 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>CALCULADOR DE BYTES B2</i>	70
TABLA 5.21 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>COMPENSADOR DE BYTES B2</i>	73
TABLA 5.22 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE <i>MULTIPLEXOR DE DATOS DE SALIDA</i>	75
TABLA 5.23 DESCRIPCIÓN DE LAS SEÑALES DE ENTRADA Y DE SALIDA DEL BLOQUE DE CONTROL DEL MÓDULO CODIFICADOR	77
TABLA 6.1 MATRIZ DE SEGUIMIENTO DE REQUERIMIENTOS DE DISEÑO CONTRA CASOS DE PRUEBA	84
TABLA 7.1 ESTADÍSTICAS DE DISEÑO DE LOS BLOQUES	86
TABLA 7.2 ESTADÍSTICAS DE VERIFICACIÓN	87
TABLA 7.3 PORCENTAJES DE COBERTURA DE CÓDIGO PARA LOS BLOQUES DE DISEÑO	88
TABLA 7.4 REPORTES DE SÍNTESIS DE LOS BLOQUES	89

1 INTRODUCCIÓN

1.1 Planteamiento del problema

SONET (Red óptica síncrona, por sus siglas en inglés: *Synchronous Optical NETwork*) es un estándar internacional para la transmisión de información a altas velocidades por medio de fibra óptica, el cual se utiliza principalmente en Estados Unidos y en Canadá. La jerarquía digital síncrona SDH (*Synchronous Digital Hierarchy*) desciende de SONET y es una estructura de transporte que ha sido estandarizada por la unión internacional de telecomunicaciones ITU (*International Telecommunication Union*), se utiliza en Europa y en el resto del mundo (excepto Estados Unidos y Canadá que utilizan SONET).

Las velocidades de transmisión de datos en redes ópticas SONET/SDH van desde 51.84 Mbps hasta 40 Gbps, que es la velocidad más alta en la actualidad. A altas velocidades de transmisión de datos en SONET/SDH (orden de Gbps), las imperfecciones en las fibras ópticas tales como: no linealidad óptica, dispersión óptica, dispersión en el modo de polarización, etc., se hacen críticas e inducen mayores errores en los datos que son transmitidos. Los factores anteriores hacen que se incremente la tasa de errores (BER: *Bit Error Rate*) del enlace óptico.

Una forma de solucionar el problema del incremento en la tasa de errores en la transmisión de datos a altas velocidades en fibras ópticas, es reduciendo la distancia entre los dispositivos regeneradores de la señal óptica. Esta solución resulta extremadamente costosa, dado el costo de los dispositivos regeneradores, además de que los cables de fibra óptica necesitarían ser desenterrados para instalar los dispositivos regeneradores.

Otra forma menos costosa es hacer uso de técnicas digitales ya desarrolladas, para proveer un servicio de corrección de errores en los datos transmitidos y reducir la tasa de errores en los enlaces ópticos. Dicha técnica digital es conocida como corrección de errores hacia adelante o FEC (por sus siglas en inglés: *Forward Error Correction*).

La corrección de errores hacia adelante es una técnica utilizada en sistemas de transmisión digital de datos para proveer robustez contra canales ruidosos, al corregir hasta un cierto número de errores que se inducen en los datos. FEC agrega información redundante a los datos que son protegidos mediante un proceso llamado codificación.

Los sistemas con corrección de errores hacia adelante disponen de una gran variedad de códigos para realizar la codificación, entre los que se encuentran los códigos BCH, códigos Reed-Solomon, códigos convolucionales y códigos concatenados.

La recomendación ITU-T G.707 define las generalidades sobre la interfaz de nodo de red para la jerarquía digital síncrona SDH. La unión internacional de telecomunicaciones ITU agregó la funcionalidad **FEC en banda** en la versión 5.0 de esta recomendación cuya revisión fue efectuada en octubre del año 2000. FEC en banda está definido para tramas con velocidades de 2.5, 10 y 40 Gbps.

El término FEC en banda se aplica para sistemas en donde la tasa de transmisión de datos permanece constante. Las unidades de transmisión de información en estos sistemas disponen de localidades libres en donde es colocada la información redundante obtenida en la codificación.

El código corrector de errores empleado en FEC en banda es un código binario, sistemático, acortado y del tipo BCH, derivado del código padre (8191,8152). FEC en banda provee un servicio de corrección de errores a la capa de línea (SONET) / sección de multiplexación (SDH). Por lo tanto pueden diseñarse elementos de red para la capa de línea / sección de multiplexación los cuales requieran contar con la funcionalidad FEC en banda, tales como ADM's (*Add Drop Multiplexers*), DCS's (*Digital Cross Connect's*) y equipos terminadores de línea. Además, los equipos regeneradores, los cuales entran en la clasificación de elementos de red para la capa de sección (SONET) / sección de regeneración (SDH), también pueden contar con dicha funcionalidad.

Esta tesis presenta el diseño de un módulo que implementa la funcionalidad FEC en banda en tramas SONET/SDH, como lo indica la recomendación ITU-T G.707 (versión 5.0).

1.2 Objetivos

1.2.1 Objetivo general

El objetivo general de este proyecto de tesis es diseñar un módulo que realice la codificación FEC en banda para elementos de red SONET/SDH a velocidades de 2.5 Gbps y 10 Gbps. Dicho módulo podrá ser utilizado en el diseño de elementos de red de la capa de línea (SONET)/sección de multiplexación (SDH), así como en regeneradores que requieran contar con la funcionalidad FEC en banda.

1.2.2 Objetivos específicos

Los objetivos específicos son:

- Realizar la especificación de requerimientos para el módulo codificador FEC en banda.
- Realizar el diseño del módulo de tal forma que se satisfagan los requerimientos especificados.
- Realizar un plan de verificación a fin de comprobar que el dispositivo funciona correctamente.
- Verificar funcionalmente el módulo diseñado mediante los casos y subcasos de prueba definidos en el plan de verificación.
- Realizar la síntesis del módulo en el software **Synopsys**.

1.3 Implementación

El diseño del módulo codificador FEC en banda en SONET/SDH se realizará en el lenguaje de descripción de hardware VHDL (*Very High Speed Integrated Circuit Hardware Description Language*).

1.4 Estilo de redacción

El estilo de redacción a utilizar en el presente trabajo de tesis presenta las siguientes características:

- Los acrónimos se presentan en idioma inglés, tal y como se citan en los libros, estándares, artículos y páginas de internet que tratan sobre los temas abordados en el presente trabajo de tesis. No es común encontrar estos acrónimos traducidos al idioma español.
- El texto contenido en las figuras se presenta en idioma inglés, por ser este idioma el de mayor difusión para artículos técnicos.

2 SONET / SDH

2.1 Introducción a SONET / SDH

SONET (*Synchronous Optical NETwork*) es un estándar internacional para el transporte de información a altas velocidades por medio de fibra óptica. Fue formulado por la ANSI (*American National Standards Institute*) y es una tecnología usada principalmente en Estados Unidos y Canadá.

SDH (*Synchronous Digital Hierarchy*) descende de SONET, es una estructura para el transporte de información que ha sido estandarizada por la ITU en la recomendación ITU-T G.707 y que se utiliza en Europa y en el resto de los países en donde no se utiliza SONET.

SONET y SDH definen una jerarquía óptica síncrona con suficiente flexibilidad para transportar señales de diferentes capacidades.

2.2 Estructura de una trama SONET / SDH

En SONET, la señal básica es llamada STS-1 (por sus siglas en inglés: *Synchronous Transport Signal Level 1*), su velocidad de transmisión es de 51.84 Mbps. La señal básica en SDH es llamada STM-1 (por sus siglas en inglés: *Synchronous Transport Module Level 1*), su velocidad de transmisión es de 155.52 Mbps.

Las señales de alto nivel son obtenidas mediante la multiplexación síncrona de señales de bajo nivel, y son denotadas como señales STS-N o STM-M. La velocidad de la señal STS-N es N veces la velocidad de la señal básica STS-1. La velocidad de la señal SDH STM-M es M veces la velocidad de la señal básica STM-1. Una señal SONET STS-N es equivalente a la señal SDH STM-M, en donde $N = 3 \times M$ y $M \in \{1, 4, 16, 64, 256\}$

Los estándares de SONET/SDH definen las señales de alto nivel que pueden formarse a partir de las señales básicas. Para señales SONET STS-N, el estándar sólo reconoce los siguientes valores para N: 1, 3, 12, 48, 192 y 768. En la Tabla 2.1 se muestra la jerarquía SONET y su equivalente SDH.

SONET	SDH	Bit Rate	SONET Capacity	SDH Capacity
STS-1	STM-0	51.84 Mbps	28 DS1 or 1 DS3	21 E1
STS-3	STM-1	155.52 Mbps	84 DS1 or 3 DS3	63 E1 or 1 E4
STS-12	STM-4	622.08 Mbps	336 DS1 or 48 DS3	252 E1 or 4 E4
STS-48	STM-16	2488.32 Mbps	1344 DS1 or 48 DS3	1008 E1 or 16 E4
STS-192	STM-64	9953.28 Mbps	5376 DS1 or 192 DS3	4032 E1 or 64 E4
STS-768	STM-256	39813.12 Mbps	21504 DS1 or 768 DS3	16128 E1 or 256 E4

Tabla 2.1 Jerarquías digitales SONET/SDH

Una trama básica SONET STS-1 es mostrada en la Figura 2.1, consiste en bytes ordenados en 9 filas por 90 columnas formando un total de 810 bytes (6480 bits). La duración de la trama es de 125 microsegundos, por lo tanto su velocidad es de 51.84 Mbps. El orden de transmisión de los bytes en la trama STS-1 es fila por fila de izquierda a derecha. El bit más significativo (*MSB: Most Significant Bit*) de cada byte es transmitido primero.

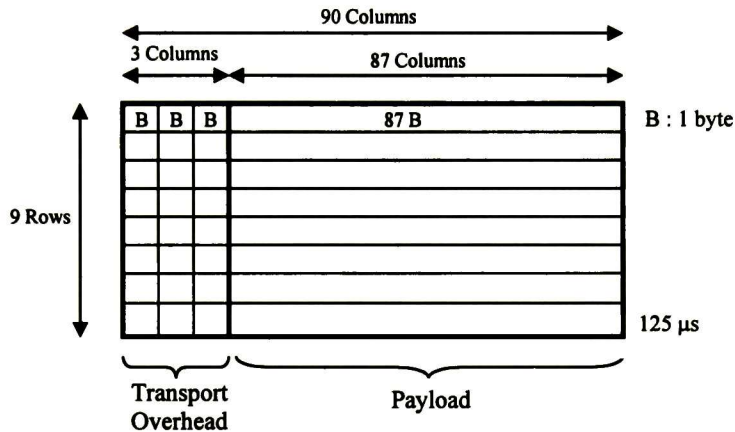


Figura 2.1 Estructura de una trama SONET STS-1

Las primeras 3 columnas de la trama básica SONET STS-1 mostrada en la Figura 2.1 forman la *tara de transporte (TOH: Transport OverHead)*. Las 87 columnas restantes forman la *carga útil o SPE (Synchronous Payload Envelope)* o simplemente *payload*, en donde se transportan señales DS3 o subseñales tales como DS2, DS1C, DS1 ó 2.048 Mbps.

En la Figura 2.2 se muestran detalladamente los bytes que forman la *tara de transporte (TOH: Transport OverHead)* en una trama STS-1. La *tara de sección (SOH: Section OverHead)* está formada por las primeras 3 filas de la *tara de transporte* y consta de 9 bytes. La *tara de línea (LOH: Line OverHead)* está formada por las 6 filas restantes de la *tara de transporte* y consta de 18 bytes. Estos bytes se utilizan para funciones de OAM&P (Operación, Administración, Mantenimiento y Provisionamiento) en los elementos de red de las *capas de línea y de sección*.

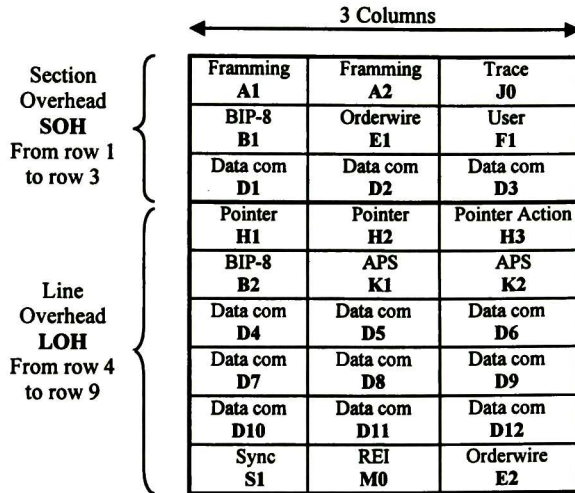


Figura 2.2 Tara de Transporte (TOH) de una trama básica STS-1

La señal de alto nivel STS-3 se obtiene mediante una multiplexación síncrona de 3 señales STS-1. Proceso que consiste en ir tomando un byte a la vez de cada señal STS-1 para así formar la señal STS-3. La estructura de la señal STS-3 es mostrada en la Figura 2.3 y consiste en bytes ordenados en 9 filas por 270 columnas. Las primeras 9 columnas forman la tara de transporte y las 261 columnas restantes forman la carga útil. La duración de las señales STS-N es constante e igual a 125 microsegundos, por tanto la velocidad de transmisión de la señal STS-3 debe ser 155.52 Mbps.

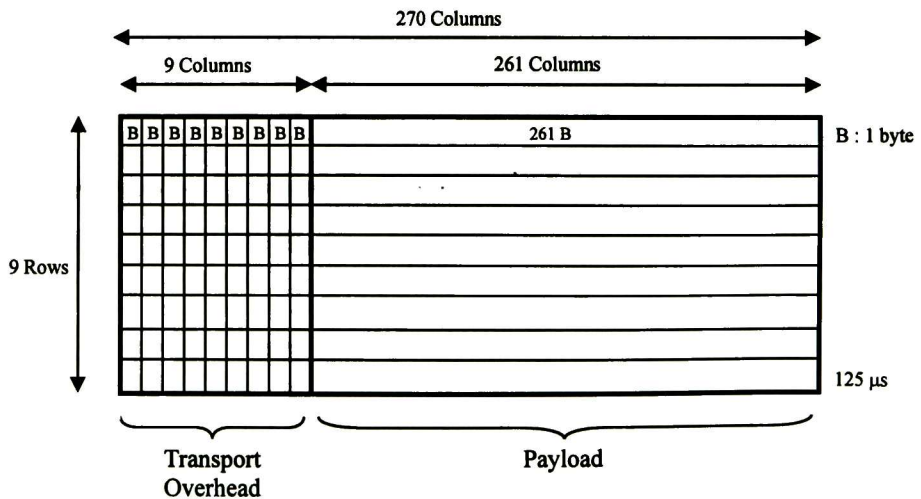


Figura 2.3 Estructura de una trama SONET STS-3

En la Figura 2.4 se muestran a detalle los bytes que forman la tara de transporte para una señal STS-3. Si se compara la tara de la señal STS-3 con la tara de la señal básica STS-1

(ver Figura 2.2) se observa que existen funciones OAM&P que hacen uso de 3 bytes y funciones que sólo emplean un byte. Los bytes restantes, o se encuentran indefinidos, o destinados para crecimiento de las funciones OAM&P, o son bytes que dependen del medio de transmisión (en SDH, además de la fibra óptica, también se ha definido su uso en enlaces satelitales, enlaces de radio y para interfaces eléctricas entre equipos) o son bytes reservados para uso nacional (para ser utilizados por los proveedores de las redes para implementar ciertas funcionalidades dentro de sus dominios de red).

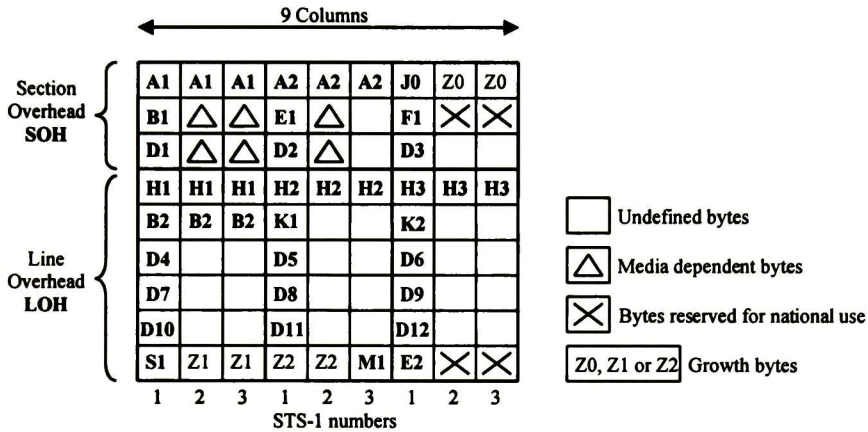


Figura 2.4 Tara de transporte de una trama SONET STS-3

2.3 FEC en banda

FEC en banda es una funcionalidad agregada a la recomendación ITU-T G.707 para proveer un servicio de corrección de errores a tramas SONET/SDH de velocidades STS-48/STM-16 (2.5 Gbps), STS-192/STM-64 (10 Gbps) y STS-768/STM-256 (40 Gbps). Lo anterior, con el fin de mejorar la tasa de errores (BER) en los enlaces vía fibra óptica, que tiende a degradarse a esas altas velocidades de transmisión.

2.3.1 Descripción del código

FEC en banda hace uso de un código binario, sistemático y acortado BCH (4359, 4320), derivado del código padre (8191, 8152), para codificar la información contenida en las tramas SONET / SDH. Dicho código tiene una capacidad de corrección de hasta 3 bits erróneos.

El polinomio generador del código es el siguiente: $g(X) = g_1(X) g_3(X) g_5(X)$, en donde:

$$g_1(X) = X^{13} + X^4 + X^3 + X + 1$$

$$g_3(X) = X^{13} + X^{10} + X^9 + X^7 + X^5 + X^4 + 1$$

$$g_5(X) = X^{13} + X^{11} + X^8 + X^7 + X^4 + X + 1$$

Se definió que la longitud del bloque de bits de información a codificar fuese de 4320 bits. Cada palabra de código es representada por el polinomio $C(X) = I(X) + R(X)$, en donde el polinomio $I(X) = a_{4358}X^{4358} + a_{4357}X^{4357} + \dots + a_{39}X^{39}$ representa al bloque de bits de información a codificar, y el polinomio $R(X) = a_{38}X^{38} + \dots + a_1X + a_0$ representa a los 39 bits de chequeo de paridad.

2.3.2 Subfilas y grupos

Cada fila de la trama SONET/SDH transporta $270 \times N$ bytes o bien $2160 \times N$ bits, en donde N es el orden de la trama SDH STM- N . Para propósitos de generar los bloques de bits de información que se codificarán, cada fila de la trama SONET / SDH es vista como un arreglo de bits de $8 \times N/16$ subfilas por 4320 columnas, como se muestra en la Figura 2.5. Cada una de estas subfilas se denomina "slice" y representa un bloque de 4320 bits de información a codificar.

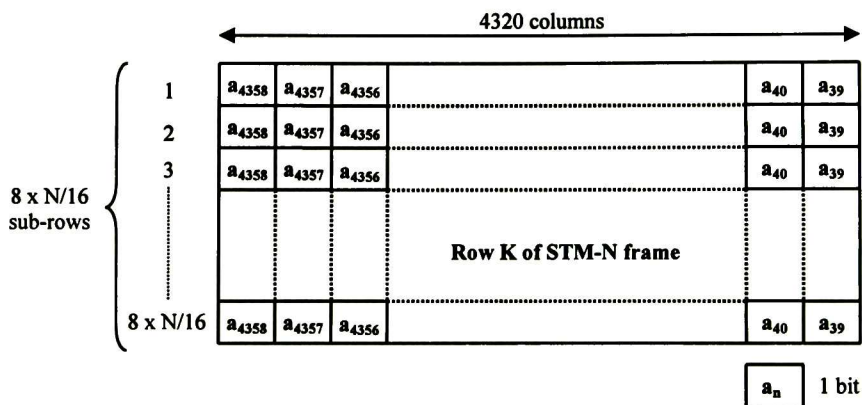


Figura 2.5 Bits de una fila en una trama SONET / SDH arreglados para propósitos de FEC en banda

El orden de transmisión de los bits en el arreglo en la Figura 2.5, con respecto al orden de transmisión de los bits en la fila de la trama SONET /SDH, es columna por columna de arriba hacia abajo. Ésta es una característica conocida como entrelazado de los bloques de información y proporciona al sistema de corrección de errores robustez para corregir errores del tipo ráfaga (*burst errors*).

De la Figura 2.5, cada fila de una trama SDH STM-16 es arreglada en $8 \times 1 = 8$ subfilas, y cada fila de una trama SDH STM-64 es arreglada en $8 \times 4 = 32$ subfilas.

Ocho subfilas sucesivas formarán un *grupo*. Por tanto, el arreglo de bits de la Figura 2.5 constará de 1 *grupo* ($M = 1$) para el caso de una trama SDH STM-16, de 4 *grupos* ($M = 1, 2, 3, 4$) para una trama SDH STM-64, y de 16 *grupos* ($M = 1, 2, \dots, 16$) para una trama SDH STM-256.

Cada subfila (*slice*) será codificada de forma independiente a las demás y el proceso de codificación generará 39 bits de chequeo de paridad a_n , en donde $0 \leq n \leq 38$.

2.3.3 Bytes P1

Al codificar las 8 subfilas de cada *grupo* se obtendrán 8 conjuntos de 39 bits de chequeo de paridad a_n , estos 8 conjuntos se mezclarán para formar 39 bytes de paridad P1(a_n). Los bits de chequeo de paridad con subíndice igual, formarán un byte P1. Por ejemplo, los 8 bits con subíndice 38 formarán el byte P1(a_{38}), en donde el bit más significativo será el bit de chequeo de paridad a_{38} calculado para la subfila 1. Este proceso se muestra en la Figura 2.6.

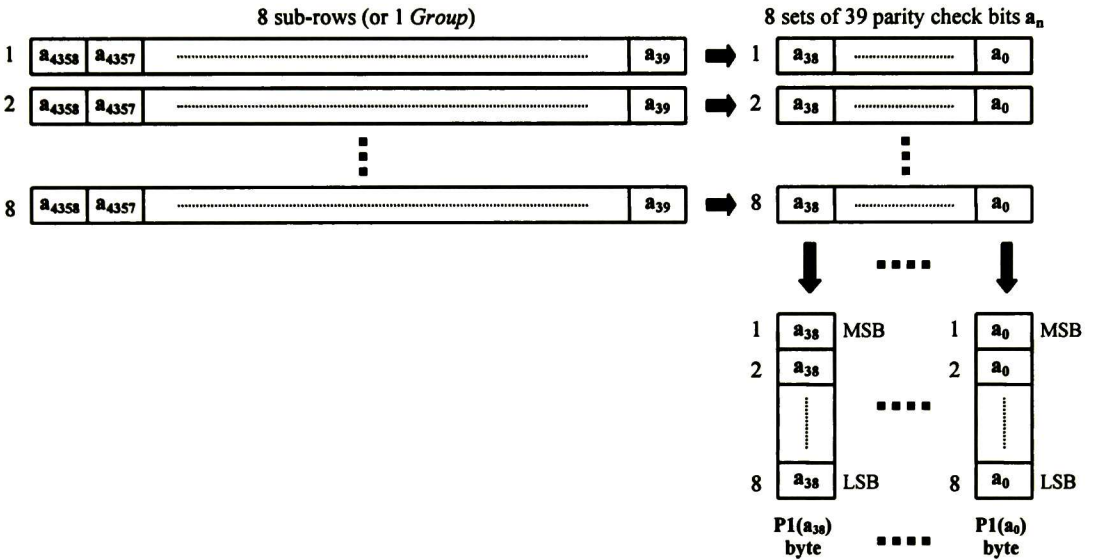


Figura 2.6 Generación de los bytes P1, a partir de los bits de chequeo de paridad a_n

Estos bytes se insertarán dentro de la tara de transporte (SONET) / tara de sección (SDH) de la trama SONET / SDH en las posiciones que se describirán a continuación.

En la recomendación ITU-T G.707 se ha definido un sistema de coordenadas de tres dimensiones $S(a, b, c)$ para ubicar a cada uno de los bytes de la tara de sección SOH (SDH). La coordenada a indica el número de fila, la coordenada b ($1 \leq b \leq 9$) indica una de 9 posibles multi-columnas y la coordenada c ($1 \leq c \leq N$) indica la posición dentro de la multi-columna. En la Figura 2.7 se muestra la tara de sección SOH (SDH) y el sistema de coordenadas mencionado.

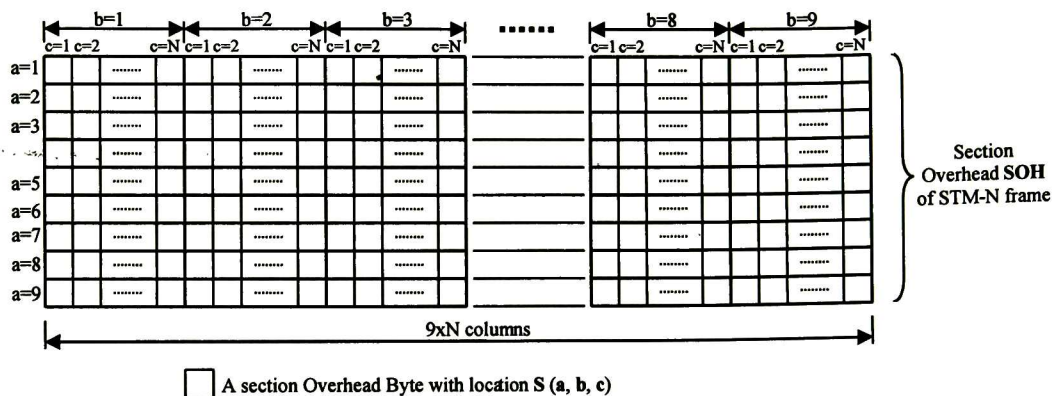


Figura 2.7 Tara de sección SOH (SDH) y su sistema de coordenadas S (a, b, c)

Las posiciones S (a, b, c) en las cuales se insertarán los 39 bytes $P1(a_n)$ de cada uno de los grupos se indica en la Tabla 2.2. En dicha tabla hace falta la coordenada c, ésta se calcula con la siguiente ecuación:

$$\text{Ecuación 2.1: } c = M \times 16 - n + 13 \times \text{Int} \left(\frac{n}{13} \right)$$

en donde M indica el grupo del cual fue calculado el byte $P1(a_n)$, n es el subíndice de los bits de chequeo de paridad (a_n) que transporta el byte $P1$, e $\text{Int} \left(\frac{n}{13} \right)$ es el cociente (parte entera) de la división indicada.

Row K	(a, b) for $P1(a_n)$ ($38 \geq n \geq 26$)	(a, b) for $P1(a_n)$ ($25 \geq n \geq 13$)	(a, b) for $P1(a_n)$ ($12 \geq n \geq 0$)
1	2, 1	2, 4	2, 6
2	3, 1	3, 4	3, 6
3	3, 7	3, 8	3, 9
4	5, 4	5, 5	5, 6
5	5, 7	5, 8	5, 9
6	6, 7	6, 8	6, 9
7	7, 7	7, 8	7, 9
8	8, 7	8, 8	8, 9
9	9, 1	9, 2	9, 3

Tabla 2.2 Coordenadas a y b en donde se insertarán los bytes $P1(a_n)$.

Así, el byte $P1(a_{38})$ correspondiente al grupo $M = 2$ de la fila 3 en una trama STS-192 / STM-64, será insertado en la posición S (3, 7, 20). De la Tabla 2.2: $a=3$ y $b=7$; de la Ecuación 2.1: $c = 2 \times 16 - 38 + 13 \times 2 = 20$.

2.3.4 Bytes Q1

Han sido definidos $N/16$ bytes denominados Q1 en tramas con FEC en banda, en donde N es el orden de la trama SDH STM- N , a fin de enviar información de control desde el módulo codificador hacia el módulo decodificador para efectos de sincronización.

Los $N/16$ bytes Q1 se encuentran ubicados en las posiciones $S(3, 9, c = 3 + 16x(M - 1))$, en donde $1 \leq M \leq N/16$. Por tanto, existe sólo un byte Q1 en las tramas STM-16 en la posición $S(3, 9, 3)$, 4 bytes Q1 en las tramas STM-64, y 16 bytes Q1 en las tramas STM-256.

El primer byte Q1 ubicado en $S(3, 9, 3)$ transporta una señal de dos bits llamada FSI (FEC Status Indicator) en los bits 7 y 8 (el bit 1 es el más significativo).

2.3.5 Bytes no incluidos en el proceso de codificación FEC en banda

En el proceso de codificación FEC en banda no se deben tomar en cuenta los siguientes bytes:

- Todos los bytes de la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH), exceptuando a los bytes Q1, los cuales sí deben considerarse en el proceso de codificación.
- Todas las posiciones correspondientes a los bytes P1.

Dichos bytes deben reemplazarse con valores “ceros” (bytes con valor 0x00) a la entrada de los calculadores de los bits de chequeo de paridad.

2.3.6 Estados operacionales del módulo codificador FEC en banda

El módulo codificador operará en tres estados:

- a) FEC habilitado
- b) FEC deshabilitado con retardo de codificación
- c) FEC deshabilitado sin retardo de codificación

El estado operacional del módulo codificador es controlado por la capa de administración de la red óptica.

Al operar el módulo codificador en el estado **a)** se hace necesario retardar los datos del flujo SONET/SDH a efectos de permitir calcular e insertar los bytes P1 dentro de la tara de transporte (SONET) / tara de sección (SDH). En el estado **b)**, el cálculo e inserción de los bytes P1 no se efectúa, pero se respeta el retardo introducido en el trayecto de los datos. En el estado **c)** no se realiza la codificación y no se introduce ningún retardo en el trayecto de

los datos. Como consecuencia, las transiciones hacia o desde el estado **c)** afectan el trayecto de los datos y por lo tanto tienen impacto en la sincronía (alineamiento) de los elementos de red.

Generación de la señal FSI

Cuando el módulo codificador FEC en banda se encuentra en el estado operacional **a)** se deberá transmitir la señal FSI = "01". Si se encuentra en el estado **b)** o en el **c)** se deberá transmitir la señal FSI = "00". Los valores de transmisión "10" y "11" se consideran inválidos.

Cambio del estado operacional

Para permitir la sincronización del módulo decodificador, la señal FSI debe cambiarse de "00" a "01" (o de "01" a "00") siete tramas antes de que la codificación quede habilitada (o quede deshabilitada); la codificación queda habilitada (o deshabilitada), entonces, a partir de la octava trama transmitida con el nuevo valor de la señal FSI.

2.3.7 Compensación de los bytes B2

2.3.7.1 Código de paridad B2

En una trama STM-N están definidos $3 \times N$ bytes B2, los cuales proveen una función de monitoreo de errores a la capa de sección de multiplexación MSOH. Esta función utiliza un código de paridad par con entrelazado de $N \times 24$ bits (BIP- $N \times 24$), que se calcula sobre todos los bytes de la trama STM-N previa, exceptuando a los bytes de la trama de sección de regeneración RSOH. El código de paridad BIP- $N \times 24$ es colocado en los bytes B2 de la trama actual.

2.3.7.2 Modelo de referencia de la función FEC en banda

En la Figura 2.8 se muestra el modelo de referencia de la función FEC en banda, la cual reside entre las funciones de la sección de multiplexación MSOH y las funciones de la sección de regeneración RSOH. Dentro de las funciones de la sección de multiplexación MSOH se encuentran el cálculo y la inserción de los bytes de paridad B2.

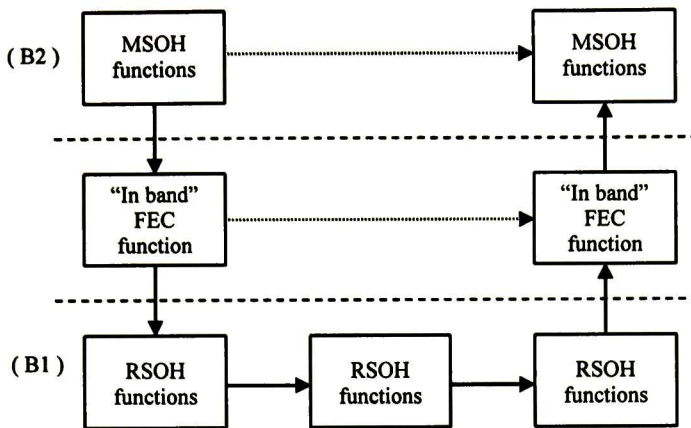


Figura 2.8 Modelo de referencia de FEC en banda

La función FEC en banda inserta los bytes de chequeo de paridad P1 dentro de la tara de sección de multiplexación MSOH, la cual está cubierta por el código de paridad B2. Por tanto es necesario compensar los bytes B2 para que incluyan a los bytes P1 insertados.

Esta compensación debe hacerse antes del proceso de codificación FEC en banda dado que los bytes B2 son tomados en cuenta en ese proceso.

2.3.8 Generación de la señal MSF-AIS

La señal MSF-AIS (*Multiplex Section FEC Alarm Indication Signal*) es una señal de indicación de alarma que se envía a otros elementos de red para indicar que se ha detectado una falla o un defecto en el flujo entrante.

La señal MSF-AIS se define como la inserción de valores “unos” (bytes con valor 0xFF) en toda la trama STM-N, excluyendo a casi toda la tara de sección de regeneración RSOH, en donde no se deben insertar valores “unos”. Las posiciones de la tara de sección de regeneración RSOH en donde se deben insertar valores “unos” son las correspondientes a los bytes P1 y bytes Q1.

La inserción de valores “unos” en los bytes P1 y Q1 es lo que distingue a la señal MSF-AIS de la señal MS-AIS (*Multiplex Section Alarm Indication Signal*). La señal MS-AIS se define como la inserción de valores “unos” en toda la trama STM-N, excluyendo a todos los bytes de la tara de sección de regeneración RSOH.

Dentro de las funciones de la sección de multiplexación MSOH, que se muestran en modelo de referencia de la Figura 2.8, se encuentra la generación de la señal MSF-AIS. Este hecho permite que la generación de la señal MSF-AIS se realice en la función FEC en banda de manera más sencilla, ya que solamente hay que insertar valores “unos” en los bytes P1 y bytes Q1 que se ubican en la tara de sección de regeneración RSOH.

2.3.9 Desempeño de FEC en banda

En la Figura 2.9 se muestra el desempeño teórico del código BCH-3 (4359, 4320) empleado en la función FEC en banda. El eje horizontal indica la tasa de errores BER (*Bit Error Rate*) de los datos a la entrada del sistema de corrección de errores y el eje vertical muestra la tasa teórica de los datos a la salida.

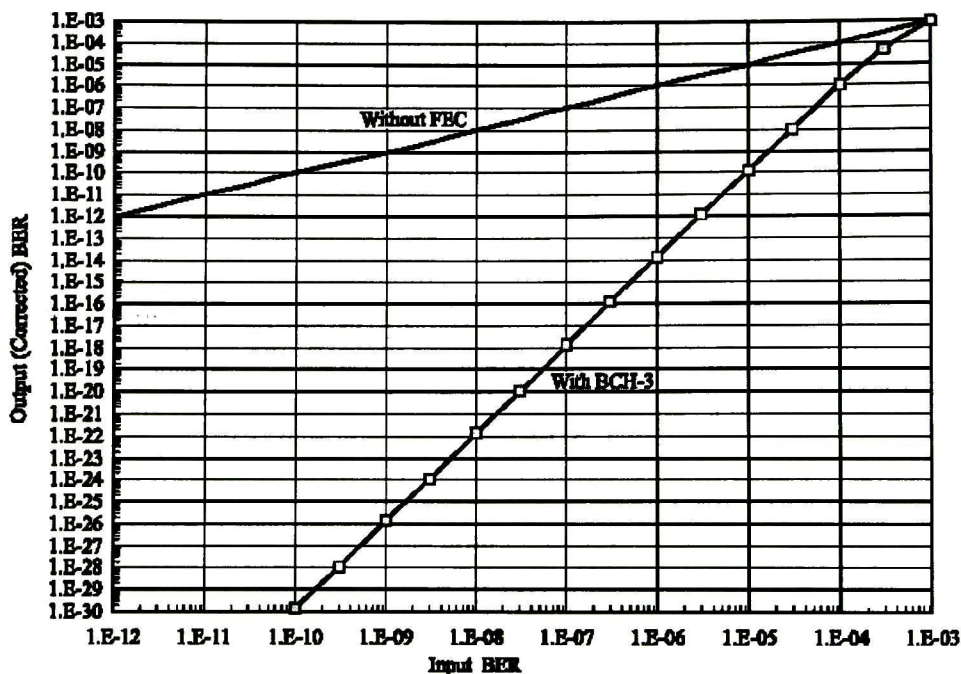


Figura 2.9 BER de salida teórico contra BER de entrada (BCH-3)

Para una tasa de errores a la entrada igual a 10^{-6} , la tasa de errores a la salida del sistema de corrección de errores se ve reducida a aproximadamente 10^{-14} . Para una tasa de errores a la entrada igual a 10^{-10} , la tasa de errores a la salida del sistema se reduce hasta aproximadamente 10^{-30} .

3 NOCIONES TEÓRICAS SOBRE CÓDIGOS CORRECTORES DE ERRORES

3.1 Introducción

Los códigos correctores de errores codifican la información que se desea transmitir a fin de que el lado receptor sea capaz de detectar o corregir errores que se pudiesen inducir en el medio de transmisión.

Existen dos clases principales de códigos correctores de errores: códigos de árbol (*tree codes*), y códigos de bloque (*block codes*). Dentro de los primeros están los códigos convolucionales; dentro de los segundos hay una gran variedad como: códigos de Hamming, códigos BCH, códigos Reed Solomon, etc.

Los códigos BCH (*Bose – Chaudhuri – Hocquenghem*) forman una gran clase de códigos correctores de múltiples errores aleatorios. Fueron descubiertos por *Hocquenghem* en 1959 [1] e independientemente por *Bose* y *Chaudhuri* en 1960 [2]. Pertenecen a la familia de los códigos cíclicos.

Las aplicaciones originales de los códigos BCH fueron restringidas a códigos binarios de longitud $2^m - 1$, en donde m es un entero. Estos fueron extendidos posteriormente a códigos no binarios con símbolos de los campos de *Galois* $GF(q)$. Dentro de los códigos BCH no binarios, la subclase más importante es la clase de los códigos *Reed-Solomon*.

El primer algoritmo de decodificación para códigos binarios BCH fue propuesto por *Peterson* en 1960 [3]. Desde entonces, el algoritmo de *Peterson* ha sido refinado por varios autores. Los algoritmos iterativo de *Berlekamp* [4] y de búsqueda de *Chien* [5] son considerados los más eficientes.

3.2 Introducción al álgebra de campos de Galois

3.2.1 Definición de Campo

Un campo es un conjunto de elementos cerrado con respecto a las operaciones de suma, resta, multiplicación y división. La suma y resta deben satisfacer las leyes conmutativas, asociativas y distributivas.

Definición: Sea F un conjunto de elementos en el cual dos operaciones binarias, llamadas suma “+” y multiplicación “·”, son definidas. El conjunto F junto con las dos operaciones binarias $+$ y \cdot es un *campo* si las siguientes condiciones son satisfechas:

- (i) F es un grupo conmutativo sobre la suma “+”. El elemento identidad con respecto a la suma es llamado el elemento *cero*, o identidad aditiva de F , y es denotado por 0.

- (ii) El conjunto de elementos en F es un grupo conmutativo sobre la multiplicación “ \cdot ” El elemento identidad con respecto a la multiplicación es llamado el elemento *unidad*, o identidad multiplicativa de F , y es denotado por 1.
- (iii) La Multiplicación es distributiva sobre la suma; esto es, para cualesquiera tres elementos a, b y c en F ,

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

3.2.2 Inverso aditivo e inverso multiplicativo

Un campo consiste de al menos dos elementos, la identidad aditiva y la identidad multiplicativa. El número de elementos en un campo es llamado el *orden* del campo. Un campo con un número finito de elementos es llamado un *campo finito*.

En un campo, el inverso aditivo de un elemento a es denotado por $-a$, y el inverso multiplicativo de a es denotado por a^{-1} (siempre y cuando $a \neq 0$). Por tanto, el restar el elemento b del elemento a equivale a sumar el inverso aditivo de b al elemento a , es decir, $a - b = a + (-b)$. Si b es un elemento distinto a cero, el dividir el elemento a entre el elemento b equivale a multiplicar a por el inverso multiplicativo de b , es decir, $a \div b = a \cdot b^{-1}$.

3.2.3 Campos primos y extensiones de campos

Sea p un número primo. El conjunto de elementos $\{0, 1, 2, \dots, p - 1\}$ es un grupo conmutativo sobre la suma módulo p y sobre la multiplicación módulo p . El conjunto $\{0, 1, 2, \dots, p - 1\}$ es un *campo* de orden p sobre la suma y multiplicación módulo p . Dado que este campo es construido sobre un número primo es llamado *campo primo* y es denotado por $\mathbf{GF}(p)$. Para $p = 2$, se obtiene el campo binario $\mathbf{GF}(2)$.

Para cualquier entero positivo m es posible extender el campo primo $\mathbf{GF}(p)$ a un campo de p^m elementos que es llamado un *campo de extensión* de $\mathbf{GF}(p)$ y es denotado por $\mathbf{GF}(p^m)$.

Códigos con símbolos dentro de los campos binarios $\mathbf{GF}(2)$ o sus extensiones $\mathbf{GF}(2^m)$ son los más ampliamente utilizados en la transmisión digital de datos y en sistemas de almacenamiento debido a que la información en estos sistemas está codificada universalmente en forma binaria por razones prácticas.

El *orden* de cualquier campo finito es un número primo. Los campos finitos también son llamados *campos de Galois*, en honor a su descubridor. Gran parte de la teoría de codificación, la construcción de códigos y la decodificación es construida sobre los campos finitos.

3.2.4 Orden de un elemento de un campo finito

Sea a un elemento diferente a cero en $\text{GF}(q)$. Dado que el número de elementos diferentes a cero en $\text{GF}(q)$ es cerrado con respecto a la multiplicación, las potencias del elemento a ,

$$a^1 = a, a^2 = a \cdot a, a^3 = a \cdot a \cdot a, \dots$$

deben ser también elementos diferentes a cero en $\text{GF}(q)$. Dado que $\text{GF}(q)$ tiene un número finito de elementos, las potencias de a dadas arriba no pueden ser todas distintas. Por tanto, en algún punto de la secuencia de las potencias de a debe existir una repetición, es decir, dados dos enteros positivos k y m tal que $m > k$ y $a^k = a^m$, se cumple que:

$$1 = a^{m-k}$$

Esto implica que debe existir un entero positivo n , lo más pequeño posible, tal que $a^n = 1$. Este entero n es llamado *el orden del elemento de campo a* . Entonces, la secuencia a^1, a^2, a^3, \dots se repite hasta que $a^n = 1$. Las potencias $a^1, a^2, a^3, \dots, a^{n-1}, a^n = 1$ son todas distintas, y forman un grupo cerrado bajo la multiplicación de $\text{GF}(q)$.

Sea a un elemento distinto a cero en un campo finito $\text{GF}(q)$, entonces se cumple que $a^{q-1} = 1$. Si el elemento a tiene un orden igual a n , entonces se cumple que n divide a $q - 1$.

3.2.5 Elemento primitivo de un campo finito

En un campo finito $\text{GF}(q)$, un elemento a distinto a cero se dice que es *primitivo* si el orden de a es $q - 1$. Por lo tanto, las potencias de un elemento primitivo generan todos los elementos distintos a cero de $\text{GF}(q)$. Cada campo finito tiene un elemento primitivo.

3.2.6 Aritmética de campos binarios

Un polinomio sobre $\text{GF}(2)$, $f(X)$, con variable X , es de la siguiente forma:

$$f(X) = f_0 + f_1X + f_2X^2 + \dots + f_nX^n$$

en donde $f_i = 0$ ó 1 para $0 \leq i \leq n$. Este es un polinomio con coeficientes del campo $\text{GF}(2)$, es decir dígitos binarios.

Los polinomios sobre $\text{GF}(2)$ pueden ser sumados, restados, multiplicados o divididos mediante operaciones módulo 2. Estos polinomios satisfacen las propiedades conmutativas, asociativas y distributivas.

3.2.6.1 División de dos polinomios sobre $\text{GF}(2)$

Suponiendo que el grado de $g(X)$ es distinto de cero. Sea $f(X)$ otro polinomio que divida a $g(X)$, se obtiene un par de polinomios sobre $\text{GF}(2)$: $q(X)$ llamado el cociente, y $r(X)$ llamado el residuo, tal que

$$f(X) = q(X)g(X) + r(X)$$

y el grado de $r(X)$ es menor que el grado de $g(X)$. Si el residuo $r(X)$ es idéntico a cero ($r(X) = 0$) se dice que $f(X)$ es divisible por $g(X)$ y que $g(X)$ es un factor de $f(X)$.

3.2.6.2 Polinomios irreducibles

Un polinomio $p(X)$ sobre $GF(2)$ de grado m se dice que es *irreducible* sobre $GF(2)$ si $p(X)$ no es divisible por ningún polinomio sobre $GF(2)$ de grado menor a m pero mayor que cero. Se ha probado que para cualquier $m \geq 1$ existe un polinomio irreducible de grado m . Estos polinomios presentan la propiedad de que son factores del polinomio $X^{2^m-1} + 1$.

3.2.6.3 Polinomios primitivos

Un polinomio irreducible $p(X)$ de grado m se dice que es *primitivo* si el entero positivo n más pequeño para el cual $p(X)$ divide a $X^n + 1$ es $n = 2^m - 1$.

3.2.7 Construcción de campos de Galois $GF(2^m)$

Sean los dos elementos 0 y 1 de $GF(2)$ y un nuevo símbolo α . Mediante la operación multiplicación “ \cdot ” se obtiene una secuencia de potencias de α como sigue:

$$\begin{aligned} 0 \cdot 0 &= 0, \\ 0 \cdot 1 &= 1 \cdot 0 = 0, \\ 1 \cdot 1 &= 1, \\ 0 \cdot \alpha &= \alpha \cdot 0 = 0, \\ 1 \cdot \alpha &= \alpha \cdot 1 = \alpha, \\ \alpha^2 &= \alpha \cdot \alpha, \\ \alpha^3 &= \alpha \cdot \alpha \cdot \alpha, \end{aligned}$$

$$\alpha^j = \alpha \cdot \alpha \cdot \dots \cdot \alpha \text{ (} j \text{ veces),}$$

Se sigue de la definición de multiplicación descrita arriba que:

$$\begin{aligned} 0 \cdot \alpha^j &= \alpha^j \cdot 0 = 0, \\ 1 \cdot \alpha^j &= \alpha^j \cdot 1 = \alpha^j, \\ \alpha^i \cdot \alpha^j &= \alpha^j \cdot \alpha^i = \alpha^{i+j}. \end{aligned}$$

Ahora se tiene el conjunto de elementos $F = \{0, 1, \alpha, \alpha^2, \dots, \alpha^j, \dots\}$ en el cual la operación multiplicación “ \cdot ” está definida.

Estableciendo las siguientes condiciones: (1) el conjunto F contiene sólo 2^m elementos y (2) este conjunto es cerrado con respecto a la operación multiplicación “ \cdot ”. Sea $p(X)$ un

polinomio primitivo de grado m sobre $\text{GF}(2)$. Asumiendo que $p(\alpha) = 0$, dado que $p(X)$ divide $X^{2^m-1} + 1$, se tiene que

$$X^{2^m-1} + 1 = q(X)p(X)$$

Si reemplazamos X por α , se obtiene

$$\alpha^{2^m-1} + 1 = q(\alpha)p(\alpha) = q(\alpha) \cdot 0 = 0$$

Sumando 1 a ambos lados de $\alpha^{2^m-1} + 1 = 0$ resulta en la siguiente igualdad:

$$\alpha^{2^m-1} = 1$$

Por tanto, bajo la condición de que $p(\alpha) = 0$, el conjunto F se hace finito y contiene los siguientes elementos:

$$F^* = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{2^m-2}\}$$

Los elementos de F^* distintos a cero son cerrados con respecto a la operación multiplicación “ \cdot ” y forman un grupo de orden $2^m - 1$. Estos elementos son conmutativos bajo la operación multiplicación “ \cdot ”.

Los $2^m - 1$ elementos de F^* distintos a cero, $\alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^m-2}$, son representados por $2^m - 1$ polinomios sobre $\text{GF}(2)$ con grado igual o menor a $m - 1$. Dichos polinomios toman la siguiente forma:

$$\alpha^i = a_i(\alpha) = a_{i,0} + a_{i,1}\alpha + a_{i,2}\alpha^2 + \dots + a_{i,m-1}\alpha^{m-1}$$

Estos polinomios forman un grupo cerrado bajo la suma “+”.

La multiplicación en F^* es distributiva sobre la suma en F^* . Por tanto, el conjunto $F^* = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{2^m-2}\}$ es un campo de Galois de 2^m elementos, $\text{GF}(2^m)$.

3.2.8 Polinomio Mínimo

Sea $\phi(X)$ el polinomio de grado más pequeño sobre $\text{GF}(2)$ tal que $\phi(\beta) = 0$. Este polinomio $\phi(X)$ es llamado polinomio mínimo del elemento β .

3.2.9 Conjugados de un elemento de campo

Sea β un elemento del campo de Galois $\text{GF}(2^m)$. Los elementos conjugados de β son todos los elementos β^{2^l} para $l \geq 1$. Los elementos del campo son finitos, por tanto, para un valor $l = e$ se cumple que $\beta^{2^e} = \beta$, además $e \leq m$. El polinomio mínimo del elemento β es el mismo que el de sus elementos conjugados.

3.3 Códigos de Bloque Lineales

Asumiendo que la salida de una fuente de información es una secuencia de dígitos binarios. En la codificación de bloques, esta información es segmentada en bloques de mensajes de longitud fija. Cada bloque de mensaje denotado por \mathbf{u} consiste de k dígitos de información, por tanto existen 2^k mensajes distintos. Un codificador transforma cada mensaje entrante \mathbf{u} en una secuencia \mathbf{v} de n dígitos (en donde $n > k$). Esta secuencia de n dígitos binarios es conocida como *palabra de código* o *vector de código* del mensaje \mathbf{u} .

Definición: Un código de bloque \mathbf{C} , de longitud n y 2^k palabras de código, es llamado un código lineal (n, k) si y sólo si sus 2^k palabras de código forman un subespacio de dimensión k del espacio vectorial de todas las secuencias binarias de longitud n sobre el campo $\mathbf{GF}(2)$.

Es posible encontrar k palabras de código linealmente independientes, $\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{k-1}$ en \mathbf{C} tal que cada palabra de código \mathbf{v} es una combinación lineal de estas k palabras de código:

$$\mathbf{v} = u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \dots + u_{k-1}\mathbf{g}_{k-1},$$

en donde u_i es 0 ó 1 para $0 \leq i < k$.

La matriz generadora \mathbf{G} del código de bloque \mathbf{C} es formada a partir de estas k palabras de código linealmente independientes:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix},$$

en donde $\mathbf{g}_i = (g_{i0}, g_{i1}, \dots, g_{i,n-1})$ para $0 \leq i < k$.

Si $\mathbf{u} = (u_0, u_1, \dots, u_{k-1})$ es el mensaje a ser codificado, la palabra de código correspondiente se obtiene como sigue:

$$\mathbf{v} = \mathbf{u} \cdot \mathbf{G} = (u_0, u_1, \dots, u_{k-1}) \cdot \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \cdots + u_{k-1}\mathbf{g}_{k-1}$$

Un código lineal (n, k) está completamente especificado por las k filas de su matriz generadora \mathbf{G} . Un codificador sólo necesita almacenar estas k filas y formar una combinación lineal de estas k filas basado en el mensaje de entrada: $\mathbf{u} = (u_0, u_1, \dots, u_{k-1})$.

Los códigos de bloque lineales poseen la propiedad de que las palabras de código pueden presentar una *estructura sistemática* como se muestra en la Figura 3.1, en donde una

palabra de código puede ser dividida en dos partes: la parte del mensaje y la parte de chequeo redundante.

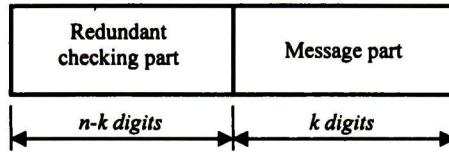


Figura 3.1 Forma sistemática de una palabra de código

3.3.1 Distancia mínima de un código de bloque

La *distancia mínima* es un parámetro de un código de bloque que determina la capacidad de un código para detectar y corregir errores.

Sea $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$ una secuencia de n dígitos binarios. El *peso de Hamming*, o simplemente *peso*, de \mathbf{v} se define como el número de dígitos diferentes a cero, y se denota como $w(\mathbf{v})$.

Sean \mathbf{v} y \mathbf{w} dos secuencias de n dígitos binarios. La *distancia de Hamming*, o simplemente *distancia*, entre \mathbf{v} y \mathbf{w} ($d(\mathbf{v}, \mathbf{w})$) es definida como el número de dígitos en que las dos secuencias difieren.

La distancia mínima de un código de bloque se define a continuación:

$$d_{min} = \min \{d(\mathbf{v}, \mathbf{w}) : \mathbf{v}, \mathbf{w} \in C, \mathbf{v} \neq \mathbf{w}\}$$

Un código de bloque con distancia mínima d_{min} garantiza la detección de todos los patrones de error que contienen un número de errores igual o menor a $d_{min} - 1$.

Un código de bloque con distancia mínima d_{min} garantiza la corrección de todos los patrones de error iguales o menor a $t = (d_{min} - 1)/2$. El parámetro t es llamado la *capacidad correctora de errores* del código.

3.4 Códigos Cíclicos

Los códigos cíclicos forman una importante subclase de los códigos lineales. Fueron estudiados primeramente por Prange en 1957.

Este tipo de códigos son atractivos debido a que su codificación puede ser implementada fácilmente mediante registros de corrimiento con conexiones retroalimentadas, además de que su estructura algebraica les permite encontrar métodos prácticos de decodificación.

3.4.1 Definición de Código Cíclico

Un código lineal $C(n, k)$ es llamado un *código cíclico* si cada corrimiento cíclico de un cierto vector de código es también un vector de código en C .

Si el vector de código $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$ es recorrido cíclicamente un lugar hacia la derecha, se obtiene otro vector de código: $\mathbf{v}^{(1)} = (v_{n-1}, v_0, \dots, v_{n-2})$.

Generalizando, al recorrer cíclicamente i lugares el vector $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$, se obtiene el vector $\mathbf{v}^{(i)} = (v_{n-i}, v_{n-i+1}, \dots, v_{n-1}, v_0, v_1, \dots, v_{n-i-1})$.

3.4.2 Representación Polinomial

La representación polinomial del vector de código $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$ es como sigue:

$$\mathbf{v}(X) = v_0 + v_1X + v_2X^2 + \dots + v_{n-1}X^{n-1}$$

3.4.3 Polinomio generador de un código cíclico $C(n, k)$

El polinomio generador $g(X)$ de un código cíclico satisface las siguientes propiedades:

- Es el código polinomial que tiene el grado mínimo.
- Los demás códigos polinomiales son múltiplos de $g(X)$.
- El grado de $g(X)$ es $n - k$, y es igual al número de dígitos de chequeo de paridad del código.
- El polinomio $g(X)$ es un factor de $X^n + 1$.

El polinomio generador $g(X)$ es de la siguiente forma:

$$g(X) = 1 + g_1X + g_2X^2 + \dots + g_{n-k-1}X^{n-k-1} + X^{n-k}$$

Cada código polinomial $\mathbf{v}(X)$ en un código cíclico (n, k) puede ser expresado en la siguiente forma:

$$\mathbf{v}(X) = \mathbf{u}(X)g(X) = (u_0 + u_1X + \dots + u_{k-1}X^{k-1})g(X)$$

en donde $\mathbf{u}(X)$, u_0, u_1, \dots, u_{k-1} , son los k dígitos de información a ser codificados, $\mathbf{v}(X)$ es el código polinomial correspondiente. Entonces, la codificación se realiza al multiplicar el mensaje polinomial $\mathbf{u}(X)$ por el polinomio generador $g(X)$.

3.4.4 Forma sistemática de un código cíclico

Sea $\mathbf{u} = (u_0, u_1, \dots, u_{k-1})$ el mensaje a ser codificado, el mensaje polinomial será:

$$\mathbf{u}(X) = u_0 + u_1X + \dots + u_{k-1}X^{k-1}$$

Multiplicando $\mathbf{u}(X)$ por X^{n-k} , obtenemos un polinomio de grado igual o menor a $n - 1$,

$$X^{n-k}\mathbf{u}(X) = u_0X^{n-k} + u_1X^{n-k+1} + \dots + u_{k-1}X^{n-1}$$

Dividiendo $X^{n-k} \mathbf{u}(X)$ por el polinomio generador $\mathbf{g}(X)$, tenemos

$$X^{n-k} \mathbf{u}(X) = \mathbf{a}(X)\mathbf{g}(X) + \mathbf{b}(X)$$

en donde $\mathbf{a}(X)$ y $\mathbf{b}(X)$ son el cociente y el residuo de la división euclidiana, respectivamente. Y dado que el grado de $\mathbf{g}(X)$ es $n - k$, el grado de $\mathbf{b}(X)$ debe ser igual o menor a $n - k - 1$:

$$\mathbf{b}(X) = b_0 + b_1X + \dots + b_{n-k-1}X^{n-k-1}$$

reordenando $X^{n-k} \mathbf{u}(X) = \mathbf{a}(X)\mathbf{g}(X) + \mathbf{b}(X)$, se obtiene el siguiente polinomio de grado igual o menor a $n - 1$:

$$\mathbf{b}(X) + X^{n-k} \mathbf{u}(X) = \mathbf{a}(X)\mathbf{g}(X)$$

Este polinomio es múltiplo del polinomio generador $\mathbf{g}(X)$ y por lo tanto es un código polinomial del código cíclico generado por $\mathbf{g}(X)$. Desarrollando $\mathbf{b}(X) + X^{n-k} \mathbf{u}(X)$, se tiene que:

$$\mathbf{b}(X) + X^{n-k} \mathbf{u}(X) = (b_0 + b_1X + \dots + b_{n-k-1}X^{n-k-1}) + (u_0X^{n-k} + u_1X^{n-k+1} + \dots + u_{k-1}X^{n-1})$$

y corresponde al vector de código:

$$(b_0, b_1, \dots, b_{n-k-1}, u_0, u_1, \dots, u_{k-1})$$

Este vector de código consiste de k dígitos de información sin alterar (u_0, u_1, \dots, u_{k-1}) y de $n - k$ dígitos de chequeo de paridad ($b_0, b_1, \dots, b_{n-k-1}$), por lo cual se encuentra en su forma sistemática. Los dígitos de chequeo de paridad son los coeficientes del residuo obtenido al dividir el mensaje polinomial $X^{n-k} \mathbf{u}(X)$ por el polinomio generador $\mathbf{g}(X)$.

Resumiendo, la codificación en forma sistemática consiste en los siguientes 3 pasos:

1. Multiplicar el mensaje $\mathbf{u}(X)$ por X^{n-k} .
2. Obtener el residuo $\mathbf{b}(X)$ (dígitos de chequeo de paridad) de dividir $X^{n-k} \mathbf{u}(X)$ por el polinomio generador $\mathbf{g}(X)$.
3. Combinar $\mathbf{b}(X)$ y $X^{n-k} \mathbf{u}(X)$ para obtener el código polinomial $\mathbf{b}(X) + X^{n-k} \mathbf{u}(X)$.

3.4.5 Circuito para implementar la codificación en un código cíclico (n, k)

La codificación en forma sistemática descrita en el punto anterior puede ser realizada con un circuito de división, el cual es simplemente un registro de corrimiento de $n - k$ etapas con conexiones de retroalimentación basadas en el polinomio generador $\mathbf{g}(X) = 1 + g_1X + g_2X^2 + \dots + g_{n-k-1}X^{n-k-1} + X^{n-k}$. Este circuito se muestra en la Figura 3.2.

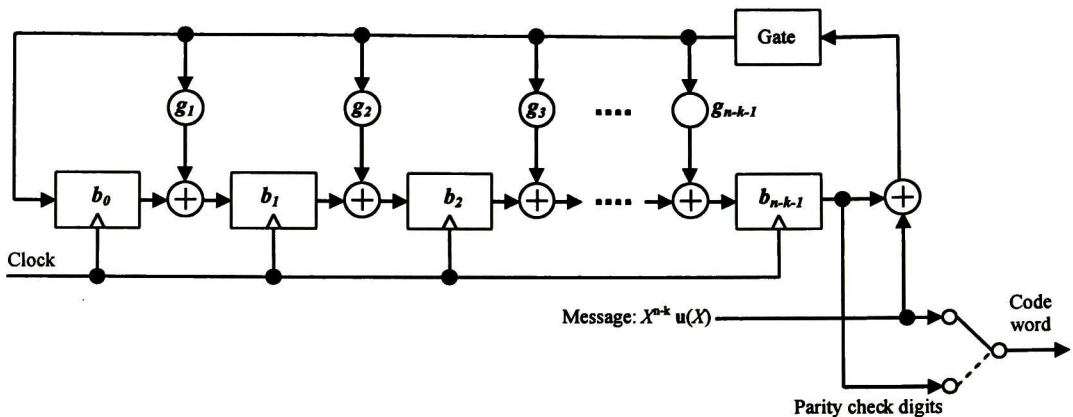


Figura 3.2 Circuito de codificación para un código cíclico (n, k)

La codificación se efectúa de la siguiente manera:

1. Con la compuerta en estado encendido, los k dígitos de información u_0, u_1, \dots, u_{k-1} son recorridos dentro del circuito de corrimiento y son enviados simultáneamente hacia el canal de comunicación. Realizar el corrimiento del mensaje $u(X)$ dentro del circuito de corrimiento es equivalente a multiplicar $u(X)$ por X^{n-k} . Una vez que los k dígitos del mensaje han entrado en el circuito de corrimiento, los $n-k$ dígitos que contienen en ese momento los registros forman el residuo de la división y representan los dígitos de chequeo de paridad.
2. La compuerta se lleva al estado apagado, con lo cual se rompe la conexión de retroalimentación.
3. Se recorren los dígitos de chequeo de paridad y se envían hacia el canal de comunicación. Los $n-k$ dígitos de chequeo de paridad $b_0, b_1, \dots, b_{n-k-1}$ junto con los k dígitos de información forman un vector de código en forma sistemática.

3.4.6 Códigos Cíclicos Acortados

Los códigos cíclicos se diseñan para poseer una determinada longitud. Cuando se diseña un sistema de corrección de errores se hace necesario usar algún código que posea una longitud específica n o bien que satisfaga un determinado número de bits de información k . Puede darse el caso de que no se encuentre un código con la longitud necesaria o con el número de bits de información necesarios, entonces se recurre a una técnica con la cual se acorta la longitud de un código para satisfacer las necesidades que demanda el sistema en el cual se aplicará.

Dado un código cíclico $C(n, k)$, si consideramos el conjunto de vectores para los cuales los l dígitos de información de mayor orden son iguales a cero, entonces estos 2^{k-l} vectores de código forman un subcódigo lineal de C .

Si los l dígitos de información son borrados de cada uno de estos vectores de código, se obtiene un conjunto de 2^{k-l} vectores de longitud $n-l$. Estos vectores forman un código

lineal $(n - l, k - l)$. Este código es llamado *código cíclico acortado*. Tiene la misma capacidad correctora de errores que el código $C(n, k)$ del cual fue derivado, el cual es conocido como código padre.

La codificación y decodificación para un código cíclico acortado puede realizarse con los mismos circuitos que emplea el código padre. Esto es debido a que los l dígitos de información eliminados no afectan el cálculo de los bits de chequeo de paridad para el caso de la codificación.

3.5 Códigos BCH

Los códigos BCH (Bose-Chaudhuri-Hocquenghem) forman una clase extensa y poderosa de códigos cíclicos correctores de múltiples errores aleatorios. Se consideran una generalización de los códigos de Hamming para corrección de múltiples errores.

Los códigos BCH son códigos con símbolos en el campo de Galois $GF(p^m)$, en donde p es un número primo. Los códigos BCH con símbolos en el campo de Galois $GF(2^m)$ se denominan códigos BCH binarios.

3.5.1 Descripción de los códigos BCH binarios

De la teoría de códigos BCH se tiene que: para cualquier entero $m \geq 3$ y $t < 2^{m-1}$ existe un código BCH binario con los siguientes parámetros:

Longitud del bloque: $n = 2^m - 1$

Número de dígitos de chequeo de paridad: $n - k \leq mt$

Distancia mínima del código: $d_{min} \geq 2t + 1$

Este código es capaz de corregir cualquier combinación igual o menor a t errores en un bloque de $n = 2^m - 1$ dígitos. Este código se conoce como código BCH corrector de t errores.

3.5.2 Polinomio generador de códigos BCH binarios

El polinomio generador de un código binario BCH se especifica en términos de sus raíces en el campo de Galois $GF(2^m)$ de la siguiente forma:

Sea α un elemento primitivo en el campo de Galois $GF(2^m)$. El polinomio generador $g(X)$ del código BCH binario corrector de t errores y de longitud $2^m - 1$, es el polinomio de menor grado sobre $GF(2)$ que tiene de raíces a los elementos: $\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t}$ ($g(\alpha^i) = 0$ para $1 \leq i \leq 2t$).

Sea $\phi_1(X)$ el polinomio mínimo del elemento α^1 . Entonces el polinomio generador $g(X)$ debe ser el mínimo común múltiplo de los polinomios mínimos $\phi_1(X), \phi_2(X), \dots, \phi_{2t}(X)$, esto es:

$$g(X) = MCM\{\phi_1(X), \phi_2(X), \dots, \phi_{2t}(X)\}$$

Los valores pares de i se pueden expresar como un producto de la siguiente forma:

$$i = i' \cdot 2^l$$

en donde i' es un número impar y $l \geq 1$. Entonces $\alpha^i = (\alpha^{i'})^{2^l}$ es un conjugado del elemento $\alpha^{i'}$ y por tanto α^i y $\alpha^{i'}$ tienen el mismo polinomio mínimo, esto es, $\phi_1(X) = \phi_{i'}(X)$.

Por tanto, cada potencia par del elemento α tiene el mismo polinomio mínimo que alguna potencia impar precedente. El polinomio generador $g(X)$ del código BCH binario corrector de t errores puede ser reducido a:

$$g(X) = MCM\{\phi_1(X), \phi_3(X), \dots, \phi_{2t-1}(X)\}$$

Dado que el grado de cada polinomio mínimo en la expresión anterior es igual o menor a m , el grado del polinomio generador $g(X)$ es a lo más mt . Esto implica que el número de dígitos de chequeo de paridad, $n - k$, del código es a lo más igual a mt .

Existen tablas en donde se listan los polinomios primitivos y los polinomios mínimos de los elementos de campo de Galois $GF(2^m)$, para diversos valores de m [7]. A partir de estos datos se diseñan los códigos BCH binarios para determinados parámetros.

3.5.3 Propiedades de los códigos BCH binarios

Sea un código BCH corrector de t errores, de longitud $n = 2^m - 1$ con polinomio generador $g(X)$. Se cumple que:

- $g(X)$ tiene a $\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t}$ como raíces: $g(\alpha^i) = 0$ para $1 \leq i \leq 2t$
- Dado que cierto código polinomial $v(X)$ es múltiplo de $g(X)$, $v(X)$ también tiene a $\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t}$ como raíces: $v(\alpha^i) = 0$ para $1 \leq i \leq 2t$.
- Un polinomio $v(X)$ de grado menor a $2^m - 1$ es un código polinomial si y sólo si tiene a $\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t}$ como raíces.

3.5.4 Códigos BCH Binarios de longitud 63

El campo de Galois $GF(2^6)$ es construido con el polinomio primitivo $p(X) = 1 + X + X^6$. Los polinomios mínimos de los elementos en $GF(2^6)$ se listan en la Tabla 3.1.

Elementos	Polinomios mínimos
$\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}$	$\phi_1(X) = 1 + X + X^6$
$\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{48}, \alpha^{33}$	$\phi_3(X) = 1 + X + X^2 + X^4 + X^6$
$\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^{40}, \alpha^{17}, \alpha^{34}$	$\phi_5(X) = 1 + X + X^2 + X^5 + X^6$
$\alpha^7, \alpha^{14}, \alpha^{28}, \alpha^{56}, \alpha^{49}, \alpha^{35}$	$\phi_7(X) = 1 + X^3 + X^6$
$\alpha^9, \alpha^{18}, \alpha^{36}$	$\phi_9(X) = 1 + X^2 + X^3$
$\alpha^{11}, \alpha^{22}, \alpha^{44}, \alpha^{25}, \alpha^{50}, \alpha^{37}$	$\phi_{11}(X) = 1 + X^2 + X^3 + X^5 + X^6$
$\alpha^{13}, \alpha^{26}, \alpha^{52}, \alpha^{41}, \alpha^{19}, \alpha^{38}$	$\phi_{13}(X) = 1 + X + X^3 + X^4 + X^6$
$\alpha^{15}, \alpha^{30}, \alpha^{60}, \alpha^{57}, \alpha^{51}, \alpha^{39}$	$\phi_{15}(X) = 1 + X^2 + X^4 + X^5 + X^6$
α^{21}, α^{42}	$\phi_{21}(X) = 1 + X + X^2$
$\alpha^{23}, \alpha^{46}, \alpha^{29}, \alpha^{58}, \alpha^{53}, \alpha^{43}$	$\phi_{23}(X) = 1 + X + X^4 + X^5 + X^6$
$\alpha^{27}, \alpha^{54}, \alpha^{45}$	$\phi_{27}(X) = 1 + X + X^3$
$\alpha^{31}, \alpha^{62}, \alpha^{61}, \alpha^{59}, \alpha^{55}, \alpha^{47}$	$\phi_{31}(X) = 1 + X^5 + X^6$

Tabla 3.1 Polinomios mínimos de los elementos en $GF(2^6)$

Usando la tabla anterior se pueden encontrar los polinomios generadores de todos los códigos BCH binarios de longitud 63. En la Tabla 3.2 se listan estos polinomios generadores junto con la capacidad correctora t del código y la longitud del bloque de bits a codificar, k .

Parámetros del código			Polinomio generador $g_t(X)$
n	k	t	
63	57	1	$g_1(X) = \phi_1(X) = 1 + X + X^6$
63	51	2	$g_2(X) = \phi_1(X)\phi_3(X) = (1 + X + X^6)(1 + X + X^2 + X^4 + X^6)$
63	45	3	$g_3(X) = g_2(X)\phi_5(X) = g_2(X)(1 + X + X^2 + X^5 + X^6)$
63	39	4	$g_4(X) = g_3(X)\phi_7(X) = g_3(X)(1 + X^3 + X^6)$
63	36	5	$g_5(X) = g_4(X)\phi_9(X) = g_4(X)(1 + X^2 + X^3)$
63	30	6	$g_6(X) = g_5(X)\phi_{11}(X) = g_5(X)(1 + X^2 + X^3 + X^5 + X^6)$
63	24	7	$g_7(X) = g_6(X)\phi_{13}(X) = g_6(X)(1 + X + X^3 + X^4 + X^6)$
63	18	10	$g_{10}(X) = g_7(X)\phi_{15}(X) = g_7(X)(1 + X^2 + X^4 + X^5 + X^6)$
63	16	11	$g_{11}(X) = g_{10}(X)\phi_{21}(X) = g_{10}(X)(1 + X + X^2)$
63	10	13	$g_{13}(X) = g_{11}(X)\phi_{23}(X) = g_{11}(X)(1 + X + X^4 + X^5 + X^6)$
63	7	15	$g_{15}(X) = g_{13}(X)\phi_{27}(X) = g_{13}(X)(1 + X + X^3)$

Tabla 3.2 Códigos BCH binarios de longitud 63, sus polinomios generadores y sus parámetros

Así, el código BCH binario (63, 45) tiene una capacidad correctora de 3 errores a lo más. La longitud del código es de 63 bits y la longitud de las palabras de información a codificar es de 45 bits. El polinomio generador de este código es: $g(X) = \phi_1(X)\phi_3(X)\phi_5(X) = 1 + X + X^2 + X^3 + X^6 + X^7 + X^9 + X^{15} + X^{16} + X^{17} + X^{18}$.

3.5.5 Código BCH binario (8191, 8152)

Este código BCH binario es el que se empleará en este trabajo de tesis. Dicho código consta de elementos del campo de Galois $GF(2^m)$, en donde $m = 13$. El polinomio primitivo que genera todos los elementos de este campo es el polinomio $p(X) = 1 + X + X^3 + X^4 + X^{13}$

Los códigos correctores de longitud $n = 2^{13} - 1 = 8191$ son obtenidos de manera similar a la descrita para los códigos de longitud 63.

El código corrector de 3 errores tiene como polinomio generador $g(X)$ al producto de los siguientes polinomios mínimos:

$$\begin{aligned}\phi_1(X) &= 1 + X + X^3 + X^4 + X^{13} \\ \phi_3(X) &= 1 + X^4 + X^5 + X^7 + X^9 + X^{10} + X^{13} \\ \phi_5(X) &= 1 + X + X^4 + X^7 + X^8 + X^{11} + X^{13}\end{aligned}$$

por tanto el polinomio generador $g(X) = \phi_1(X) \phi_3(X) \phi_5(X) = 1 + X^2 + X^3 + X^5 + X^6 + X^7 + X^8 + X^{10} + X^{11} + X^{12} + X^{13} + X^{15} + X^{17} + X^{20} + X^{21} + X^{23} + X^{24} + X^{26} + X^{28} + X^{29} + X^{30} + X^{31} + X^{33} + X^{35} + X^{36} + X^{37} + X^{39}$

El número de dígitos de chequeo de paridad $n - k$ es igual a 39, que es el grado del polinomio generador. Por tanto, las palabras de información tienen una longitud de $k = 8152$ dígitos binarios.

Se utiliza un código acortado derivado de este código padre (8191, 8152) para implementar un sistema de corrección de errores en la función FEC en banda en SONET/SDH. El código padre se acorta de forma que la longitud de las palabras de información sea de 4320 dígitos binarios. El código resultante se identifica como un código BCH binario acortado (4359, 4320), y la codificación se realiza de manera sistemática.

Al igual que para los códigos cíclicos, la codificación en forma sistemática en los códigos BCH binarios se efectúa mediante registros de corrimiento con conexiones de retroalimentación basadas en el polinomio generador del código.

4 ESPECIFICACIÓN DE REQUERIMIENTOS

4.1 Introducción

La especificación de requerimientos para un módulo de hardware es el documento que plasma las necesidades del cliente al grupo que diseñará el módulo. En él se definen claramente las características generales y técnicas que el cliente espera del módulo a diseñar. Dicha especificación debe escribirse en un lenguaje claro y sin ambigüedades de forma que sea entendido por ambas partes.

4.2 Módulo codificador FEC en banda

4.2.1 Propósito del hardware

El propósito del módulo de hardware a diseñar es el de implementar la función FEC en banda en SONET / SDH a velocidades de 2.5 Gbps y 10 Gbps, y obtener un bloque reutilizable para diseñar elementos de redes ópticas SONET / SDH que requieran contar con dicha función.

4.2.2 Restricciones principales del módulo

1. El módulo procesará tramas SONET/SDH de velocidades STS-48/STM-16 (2.5 Gbps) y STS-192/STM-64 (10 Gbps).
2. Los buses de entrada y salida de datos del módulo codificador FEC en banda tendrán un ancho de 128 bits.
3. La frecuencia del reloj con que operará el módulo codificador FEC en banda será de 19.44 MHz para tramas STS-48/STM-16, y de 77.76 MHz para tramas STS-192/STM-64.

4.2.3 Suposiciones y dependencias

1. Los datos de entrada vendrán acompañados de una señal llamada "*J0 detected*" proporcionada por dispositivos externos, y que indicará la presencia del byte J0 en el flujo de los datos de entrada.
2. La señal de reloj es proporcionada por dispositivos externos a velocidades dependientes del orden de trama a procesar.
3. Los buses de datos tienen a su bit más significativo situado a su izquierda (127 es el bit más significativo en un bus de 128 bits, que será referenciado como un bus de 127 a 0).
4. Señales de control del módulo codificador FEC en banda serán generadas por un dispositivo externo que será la interfaz hacia la capa de administración de la red óptica.

4.2.4 Requerimientos funcionales del módulo codificador

- R-FEC-ENC-01:** El módulo codificador FEC en banda procesará tramas SONET/SDH de órdenes STS-48/STM-16 y STS-192/STM-64.
- R-FEC-ENC-02:** Para la codificación FEC en banda se utilizará un código binario sistemático BCH acortado (4359,4320), derivado del código padre (8191,8152), el cual tiene una capacidad de corrección de hasta 3 bits erróneos.
- R-FEC-ENC-03:** El polinomio generador para el código binario sistemático BCH acortado utilizado en la codificación FEC en banda, será el siguiente: $g(X) = 1 + X^2 + X^3 + X^5 + X^6 + X^7 + X^8 + X^{10} + X^{11} + X^{12} + X^{13} + X^{15} + X^{17} + X^{20} + X^{21} + X^{23} + X^{24} + X^{26} + X^{28} + X^{29} + X^{30} + X^{31} + X^{33} + X^{35} + X^{36} + X^{37} + X^{39}$
- R-FEC-ENC-04:** El retardo máximo en el trayecto de datos que introducirá el módulo codificador FEC en banda será de 15 μ s.
- R-FEC-ENC-05:** Los bits de la fila k ($1 \leq k \leq 9$) de la trama SDH STM-N a procesar, serán ordenados en $8xN/16$ subfilas o *slices*, cada uno constará de 4320 bits como se muestra en la Figura 2.5.
- R-FEC-ENC-06:** Cada una de las $8xN/16$ subfilas que forman una fila de la trama SDH STM-N, será codificada de forma independiente y se obtendrán 39 bits de chequeo de paridad a_n ($0 \leq n \leq 38$) para cada una de ellas.
- R-FEC-ENC-07:** Cada 8 *slices* sucesivos formarán un *grupo*, por tanto, por cada fila de una trama STM-N se tendrán $N/16$ *grupos* (M : índice de grupo: $M \in \{1, 2, \dots, N/16\}$).
- R-FEC-ENC-08:** Por cada *grupo* codificado se obtendrán 8 conjuntos de 39 bits de chequeo de paridad, los cuales serán agrupados para formar 39 bytes $P1(a_n)$, en donde $0 \leq n \leq 38$.
- R-FEC-ENC-09:** Los bytes P1, calculados para la fila k ($1 \leq k \leq 9$) de la trama STM-N, serán insertados en las coordenadas $S(a,b,c)$. (a,b) se muestran en la Tabla 2.2, la coordenada $c = Mx16 - n + 13xInt(n/13)$, en donde M es el índice del *grupo* del cual se obtuvo el byte P1, n es el subíndice de los bits de paridad (a_n) que transporta el byte P1 e $Int(n/13)$ es el parte entera de la división indicada.

- R-FEC-ENC-10:** Para el cálculo de los bytes P1 se descartarán los bytes correspondientes al RSOH, incluyendo los bytes RSOH no definidos y exceptuando los bytes Q1. También se descartarán las posiciones correspondientes a los bytes P1. Los bytes descartados se considerarán como ceros para el proceso de codificación.
- R-FEC-ENC-11:** El módulo codificador FEC en banda insertará una palabra de control de 8 bits en el primer byte Q1 de la trama SONET/SDH, cuya posición es: *S* (3, 9, 3). Los bits 7 y 8 de dicho byte transportarán una señal de 2 bits llamada *FSI* (*FEC Status Indicator*) (el bit 8 es el bit menos significativo). Los bits del 1 al 6 transportarán valores '0' (este requerimiento fue tomado de la recomendación ITU-T G.707, en donde el bit 1 es el más significativo).
- R-FEC-ENC-12:** El módulo codificador FEC en banda tendrá 3 estados operacionales:
- a) *FEC habilitado*
 - b) *FEC deshabilitado con retardo de codificación*
 - c) *FEC deshabilitado sin retardo de codificación*
- R-FEC-ENC-13:** Cuando el módulo codificador FEC en banda se encuentre operando en el estado a) *FEC habilitado*, la señal *FSI* = "01" será transmitida.
- R-FEC-ENC-14:** Cuando el módulo codificador FEC en banda se encuentre operando en el estado b) *FEC deshabilitado con retardo de codificación* o en el estado c) *FEC deshabilitado sin retardo de codificación*, la señal *FSI* = "00" será transmitida.
- R-FEC-ENC-15:** El módulo codificador FEC en banda pasará del estado a) *FEC habilitado* al estado b) *FEC deshabilitado con retardo de codificación* o al estado c) *FEC deshabilitado sin retardo de codificación* al detectar la solicitud de cambio de estado en las entradas de control. El nuevo valor de *FSI* transmitido será "00" (indicando *FEC deshabilitado*). El proceso de codificación (cálculo e inserción de los bytes P1 y compensación de los bytes B2) permanecerá habilitado durante las primeras 7 tramas enviadas con *FSI* = "00", para propósitos de sincronización en el módulo decodificador FEC en banda. A partir de la octava trama el proceso de codificación deberá deshabilitarse.
- R-FEC-ENC-16:** El módulo codificador FEC en banda pasará del estado b) *FEC deshabilitado con retardo de codificación* o del estado c) *FEC deshabilitado sin retardo de codificación* al estado a) *FEC habilitado* al detectar la solicitud de cambio de estado en las

entradas de control. El nuevo valor de FSI transmitido será “01” (indicando FEC habilitado). El proceso de codificación permanecerá deshabilitado durante las primeras 7 tramas enviadas con FSI = “01”, para propósitos de sincronización en el módulo decodificador FEC en banda. A partir de la octava trama el proceso de codificación deberá habilitarse.

- R-FEC-ENC-17:** Los valores de transmisión $FSI = “10”$ y $FSI = “11”$ son inválidos.
- R-FEC-ENC-18:** Las transiciones entre los estados operacionales del módulo codificador FEC en banda serán controladas por la capa de administración de la red, por medio de señales de control entrantes al módulo.
- R-FEC-ENC-19:** Los bytes B2 de la trama SONET/SDH deberán ser compensados debido a la inserción de bytes P1 en la tara de sección de multiplexación MSOH. Los bytes B2 son tomados en cuenta en el proceso de codificación, por tanto deberán ser compensados antes de realizar el cálculo de los bits de chequeo de paridad.
- R-FEC-ENC-20:** La señal de indicación de alarma MSF-AIS (*Multiplex Section FEC Alarm Incoming Signal*) es especificada como todos “unos” (bytes con valor 0xFF) en toda la trama STM-N, excluyendo a la tara de sección de regeneración RSOH en donde se exceptúan a los bytes P1 y Q1, los cuales deberán contener “unos”.
- R-FEC-ENC-21:** Una señal de control le indicará al módulo codificador FEC en banda cuándo generar la señal de indicación de alarma MSF-AIS.
- R-FEC-ENC-22:** Para generar la señal de indicación de alarma MSF-AIS, el módulo codificador FEC en banda deberá insertar bytes con valor 0xFF en las posiciones de los bytes P1 y Q1 de la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH).

4.2.5 Requerimientos no funcionales del módulo codificador

- R-FEC-ENC-23:** El módulo debe diseñarse basándose en estándares internacionales.
- R-FEC-ENC-24:** El módulo deberá ser implementado en el lenguaje VHDL.
- R-FEC-ENC-25:** El módulo deberá verificarse funcionalmente.

- R-FEC-ENC-26:** El módulo descrito en VHDL deberá ser sintetizable, para ello se utilizará el software de síntesis **Synopsys**.
- R-FEC-ENC-27:** En la verificación funcional se emplearán archivos en formato CSV (*Comma Separated Values*) que contienen los estímulos de entrada.
- R-FEC-ENC-28:** La generación de los archivos CSV de entrada será realizada por una herramienta de software desarrollada en el Cinvestav Unidad Guadalajara [13].

5 ARQUITECTURA Y DISEÑO

5.1 Introducción

El módulo codificador FEC en banda para elementos de red SONET/SDH implementa la funcionalidad FEC en banda definida en el anexo A de la recomendación ITU-T G.707 versión 5.0. Procesa tramas SONET/SDH de órdenes STS-48/STM-16 (2.5 Gbps) y STS-192/STM-64 (10 Gbps).

Este módulo se podrá utilizar en el diseño de elementos de red para la capa de línea (SONET) / sección de multiplexación (SDH), tales como ADMs (Add/Drop Multiplexers), DCSs (Digital Cross-Connect Systems), así como elementos *Regeneradores* pertenecientes a la capa de sección (SONET)/sección de regeneración (SDH) que requieran contar con la funcionalidad FEC en banda.

El módulo codificador presenta las siguientes características:

- Capacidad de procesar flujos SONET/SDH STS-48/STM-16 (2.5 Gbps) y STS-192/STM-64 (10 Gbps).
- Buses de entrada y salida de datos con un ancho de 128 bits.
- Tres estados de operación: FEC habilitado, FEC deshabilitado con retardo de codificación y FEC deshabilitado sin retardo de codificación.
- La capa de administración de la red óptica controla el estado de operación del módulo codificador FEC en banda.

5.2 Señales de entradas y de salida del módulo codificador

En la Figura 5.1 se presenta el diagrama de señales de entrada y de salida del módulo codificador, y en la Tabla 5.1 se presenta la descripción de estas señales.

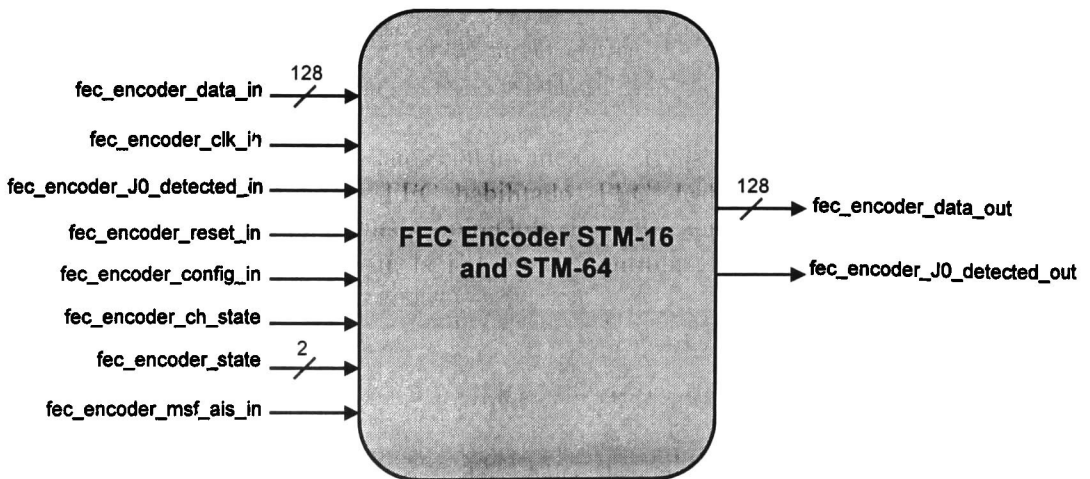


Figura 5.1 Diagrama de señales de entrada y de salida del módulo codificador

Entrada	Descripción
fec_encoder_data_in [127:0]	Bus de entrada de datos de 128 bits leído con el flanco de subida del reloj del sistema.
fec_encoder_clk_in	Reloj del sistema. En modo 2.5 Gbps, la frecuencia de este reloj será de 19.44 MHz y en modo 10 Gbps será de 77.76 MHz.
fec_encoder_J0_detected_in	Señal de sincronía. Cuando el byte de traza J0 se encuentra presente en el bus de entrada de datos, esta señal toma el valor "1".
fec_encoder_reset_in	Señal de reset del módulo codificador. Esta señal es activa en alto.
fec_encoder_config_in	Señal de configuración del módulo codificador. Con un valor "0" el módulo operará en el modo de 2.5 Gbps y con un valor "1" operará en el modo de 10 Gbps.
fec_encoder_ch_state	Señal de control para realizar un cambio en el estado operacional del módulo codificador. Esta señal es activa en alto.
fec_encoder_state [1:0]	Señal de indicación del estado operacional siguiente. Esta señal de dos bits es leída cuando se solicita un cambio en el estado operacional del módulo codificador. El valor "01" indica que el estado operacional siguiente será <i>FEC Habilitado</i> , el valor "10" indica que será <i>FEC deshabilitado con retardo de codificación</i> y el valor "11" indica que será <i>FEC deshabilitado sin retardo de codificación</i> . El valor "00" es inválido.
fec_encoder_msf_ais_in	Señal de control para la generación de la señal de indicación de alarma MSF-AIS. Esta señal es activa en alto.
Salida	Descripción
fec_encoder_data_out [127:0]	Bus de salida de datos con un ancho de 128 bits.
fec_encoder_J0_detected_out	Señal de sincronía. Cuando el byte de traza J0 se encuentra presente en el bus de salida de datos, esta señal toma el valor "1".

Tabla 5.1 Descripción de las señales de entrada y de salida del módulo codificador

5.3 Arquitectura del módulo codificador

En la Figura 5.2 se muestra la arquitectura propuesta para el módulo codificador FEC en banda. Está formada por 9 bloques, los cuales son:

- Insertor de la señal FSI (*FSI Insertor*)
- Compensador de bytes B2 (*B2 Bytes Compensator*)
- Insertor de ceros (*Zeros Insertor*)
- Calculador de bytes P1 (*P1 Bytes Calculator*)
- Buffer de datos (*Data Path Buffer*)
- Insertor de bytes P1 (*P1 Bytes Insertor*)
- Calculador de bytes B2 (*B2 Bytes Calculator*)
- Multiplexor de datos de salida (*Output Mux*)
- Control (*FEC Encoder Control*)

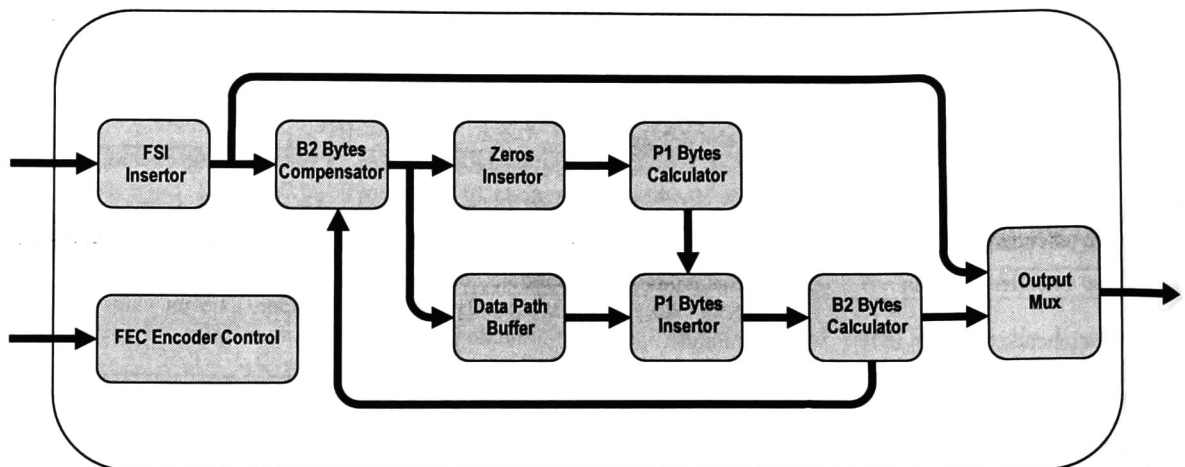


Figura 5.2 Arquitectura del módulo codificador

A continuación se listan las principales funciones del módulo codificador. Para cada función se indica el bloque o los bloques encargados de su ejecución.

- **Inserción de los bytes Q1:** Función ejecutada por el bloque *Insertor de la señal FSI*. Esta función debe ser ejecutada en primer término a fin de incluir los bytes Q1 en el proceso de codificación.
- **Exclusión de bytes no considerados en la codificación:** Función ejecutada por el bloque *Insertor de ceros*. Esta función debe ser ejecutada antes del cálculo de los bytes de paridad P1.
- **Cálculo de los bytes P1:** Función ejecutada por el bloque *Calculador de bytes P1*. La inserción de los bytes Q1 y la compensación de los bytes B2 deben ejecutarse previo al cálculo de los bytes P1, a fin de incluir esta información en el proceso de codificación.
- **Retardo en los datos:** Función ejecutada por el bloque *Buffer de datos*. Las posiciones destinadas para los bytes P1 obliga a tener que introducir un retardo en el trayecto de los datos a fin de permitir insertar esta información de chequeo de paridad.
- **Inserción de los bytes P1:** Función ejecutada por el bloque *Insertor de bytes P1*. Los bytes P1 a insertar son enviados por el bloque *Calculador de bytes P1*.
- **Compensación de bytes B2:** Función ejecutada por el bloque *Compensador de bytes B2* con ayuda del bloque *Calculador de bytes B2*. El bloque *Calculador de bytes B2* realiza el cálculo de los bytes B2 sobre el flujo de datos de salida del bloque *Insertor de bytes P1* y envía estos bytes calculados al bloque *Compensador de bytes B2* para su inserción. Los bytes B2 son compensados antes del cálculo de los bytes P1, debido a que los bytes B2 son incluidos en la codificación.
- **Generación de la señal MSF-AIS:** Función ejecutada por los bloques *Insertor de la señal FSI* e *Insertor de bytes P1*.

- **Control del estado operacional:** Función ejecutada por el bloque *Control*. El bloque *Control* genera todas las señales para controlar el resto de los bloques del módulo codificador.

En la Tabla 5.2 se presenta una matriz de seguimiento que permite comparar la relación entre los requerimientos de diseño y las funciones del módulo codificador FEC en banda.

	Inserción de bytes Q1	Exclusión de bytes en codificación	Cálculo de bytes P1	Retardo en el trayecto de datos	Inserción de bytes P1	Compensación de bytes B2	Generación de MSF-AIS	Control del estado operacional
R-FEC-ENC-01	●	●	●	●	●	●	●	●
R-FEC-ENC-02			●					
R-FEC-ENC-03			●					
R-FEC-ENC-04				●				
R-FEC-ENC-05			●					
R-FEC-ENC-06			●					
R-FEC-ENC-07			●					
R-FEC-ENC-08			●					
R-FEC-ENC-09					●			
R-FEC-ENC-10		●						
R-FEC-ENC-11	●							
R-FEC-ENC-12								●
R-FEC-ENC-13	●							●
R-FEC-ENC-14	●							●
R-FEC-ENC-15	●							●
R-FEC-ENC-16	●							●
R-FEC-ENC-17	●							
R-FEC-ENC-18								●
R-FEC-ENC-19						●		
R-FEC-ENC-20	●				●			
R-FEC-ENC-21								●
R-FEC-ENC-22	●				●			

Tabla 5.2 Matriz de seguimiento de requerimientos contra funciones

En la Tabla 5.3 se presenta una matriz de seguimiento para comparar la relación entre las funciones del módulo codificador FEC en banda y los bloques de diseño del mismo.

	Insertor de la señal FSI	Insertor de ceros	Calculador de bytes P1	Buffer de datos	Insertor de bytes P1	Calculador de bytes B2	Compensador de bytes B2	Multiplexor de datos de salida	Control
Inserción de bytes Q1	●								
Exclusión de bytes en codificación		●							
Cálculo de bytes P1			●						
Retardo en el trayecto de datos				●					

Inserción de bytes P1					●				
Compensación de bytes B2						●	●		
Generación de MSF-AIS	●				●				●
Control del estado operacional								●	●

Tabla 5.3 Matriz de seguimiento de funciones contra bloques de diseño

A continuación se explica a detalle cada uno de los bloques que integran el módulo codificador.

5.4 Insertor de la señal FSI

Este bloque es el encargado de insertar la señal de dos bits FSI (*FEC Status Indicator*) en el primer byte Q1, que se ubica dentro de la tara en las coordenadas $S(3,9,3)$. La señal FSI es insertada en los bits 7 y 8, los bits del 1 al 6 son puestos a “0” (el bit 1 es el bit más significativo). Los valores de transmisión “00” y “01”, para la señal FSI, son los únicos válidos. El valor de FSI a insertar es proporcionado por el bloque de Control.

Este bloque, además, contribuye a la generación de la señal de indicación de alarma MSF-AIS (*Multiplex Section FEC Alarm Indication Signal*) al insertar bytes “unos” (bytes con valor 0xFF) en todas las posiciones de los bytes Q1.

En la Figura 5.3 se presenta el diagrama de señales de entrada y de salida del bloque *Insertor de la señal FSI*, y en la Tabla 5.4 se presenta la descripción de estas señales.

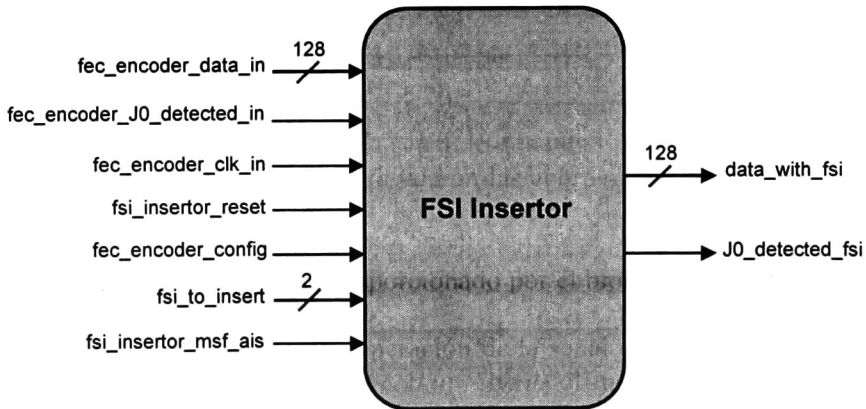


Figura 5.3 Diagrama de señales de entrada y de salida del bloque *Insertor de la señal FSI*

Entrada al bloque	Descripción	Origen
fec_encoder_data_in [127:0]	Bus de datos de entrada	Exterior
fec_encoder_J0_detected_in	Señal de sincronía	Exterior
fec_encoder_clk_in	Señal de reloj del sistema	Exterior

fsi_insertor_reset	Señal de reset activa en valor alto	Control
fec_encoder_config	Señal de configuración	Control
fsi_to_insert [1:0]	Señal FSI a insertar en el bloque	Control
fsi_insertor_msf_ais	Señal de control para la generación de MSF-AIS	Control
Salida del bloque		Destino
data_with_fsi [127:0]	Bus de datos de salida	<ul style="list-style-type: none"> • Compensador de bytes B2 • Multiplexor de datos de salida
J0_detected_fsi	Señal de sincronía	<ul style="list-style-type: none"> • Compensador de bytes B2 • Multiplexor de datos de salida

Tabla 5.4 Descripción de las señales de entrada y de salida del bloque *Insertor de la señal FSI*

5.5 Insertor de ceros

Los bytes no incluidos en el proceso de codificación FEC en banda son los siguientes:

- Todos los bytes de la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH), y exceptuando a los bytes Q1.
- Todos los bytes P1.

El bloque *Insertor de ceros* se encarga de realizar la inserción de “ceros” (bytes con valor 0x00) en las posiciones de los bytes que no serán incluidos en la codificación FEC en banda. En la Figura 5.4 se presenta el diagrama de señales de entrada y de salida del bloque *Insertor de ceros*, y en la Tabla 5.5 se presenta la descripción de estas señales.

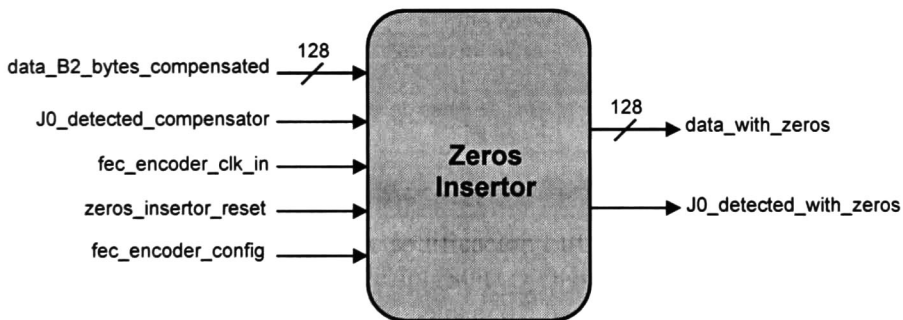


Figura 5.4 Diagrama de señales de entrada y de salida del bloque *Insertor de ceros*

Entrada al bloque	Descripción	Origen
data_B2_bytes_compensated [127:0]	Bus de datos de entrada	Compensador de bytes B2
J0_detected_compensator	Señal de sincronía	Compensador de bytes B2
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
zeros_insertor_reset	Señal de reset activa en valor alto	Control
fec_encoder_config	Señal de configuración	Control
Salida del bloque	Descripción	Destino
data_with_zeros [127:0]	Bus de datos de salida	Calculador de bytes P1
J0_detected_with_zeros	Señal de sincronía	Calculador de bytes P1

Tabla 5.5 Descripción de las señales de entrada y de salida del bloque *Insertor de ceros*

5.6 Calculador de bytes P1

Este bloque realiza el cálculo de los bytes P1 y los envía en el tiempo y en el orden adecuado hacia el bloque *Insertor de bytes P1*.

En la Figura 5.5 se presenta el diagrama de señales de entrada y de salida del bloque *Calculador de bytes P1*, y en la Tabla 5.6 se presenta la descripción de estas señales.

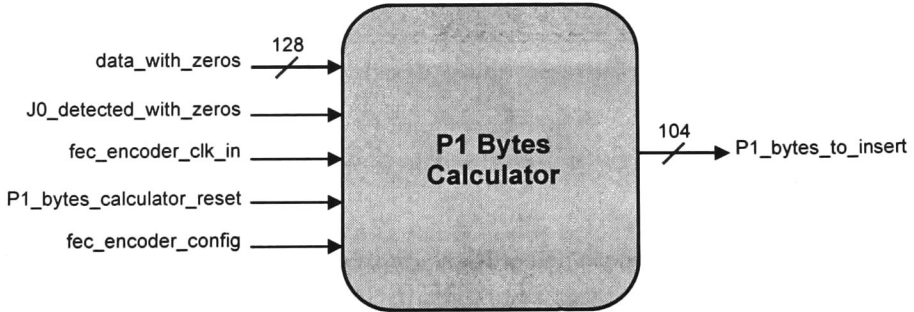


Figura 5.5 Diagrama de señales de entrada y de salida del bloque *Calculador de bytes P1*

Entrada al bloque	Descripción	Origen
data_with_zeros [127:0]	Bus de datos de entrada	Insertor de ceros
J0_detected_with_zeros	Señal de sincronía	Insertor de ceros
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
P1_bytes_calculator_reset	Señal de reset activa en valor alto	Control
fec_encoder_config	Señal de configuración	Control
Salida del bloque	Descripción	Destino
P1_bytes_to_insert [103:0]	Bus de datos de salida de bytes P1	Insertor de bytes P1

Tabla 5.6 Descripción de las señales de entrada y de salida del bloque *Calculador de bytes P1*

5.6.1 Microarquitectura del bloque Calculador de bytes P1

En la Figura 5.6 se presenta el diagrama de microarquitectura del bloque *Calculador de bytes P1*.

El flujo de datos de entrada *data_with_zeros* tiene un ancho de 128 bits, por tanto, la duración en ciclos de reloj de una trama STS-48 / STM-16 es de 2430 ciclos y para una trama STS-192 / STM-64 es de 9720 ciclos. Estos 128 bits son ordenados en 16 bytes numerados del 1 al 16, en donde el byte 1 contiene al bit más significativo, y por tanto está formado por los bits del 127 al 120. El byte 16 contiene al bit menos significativo y está formado por los bits del 7 a 0.

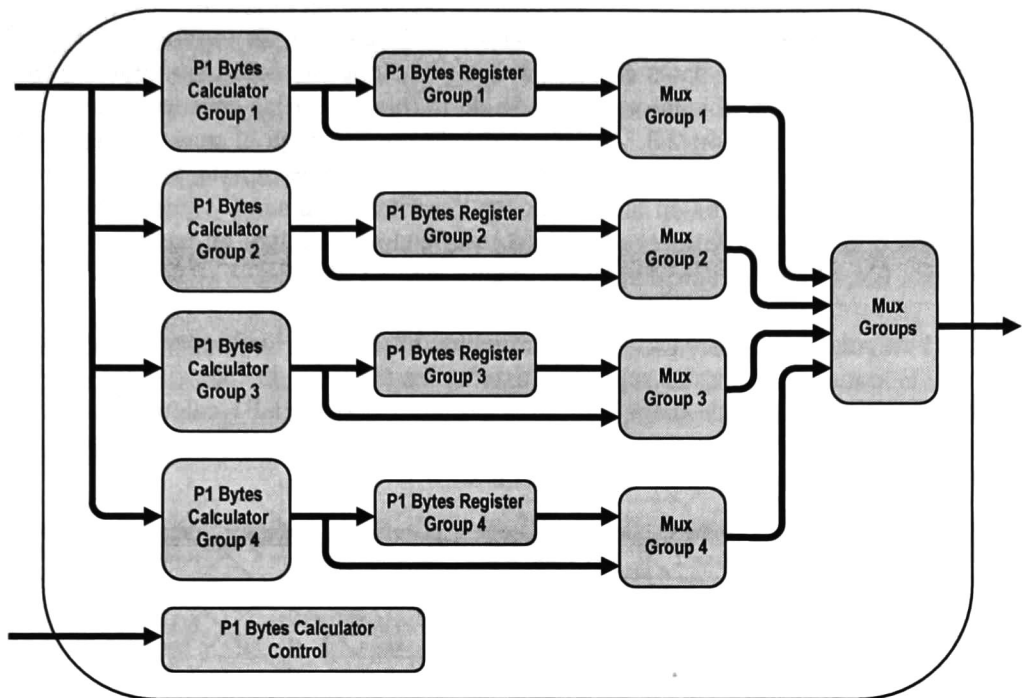


Figura 5.6 Microarquitectura del bloque *Calculador de bytes P1*

De acuerdo a lo expuesto en la sección 2.3.2, en el modo de 2.5 Gbps (STS-48 / STM-16) los bits de información para una fila son ordenados en 8 subfilas para formar el grupo 1. El flujo de datos entrante debe ser separado en 8 flujos para formar los flujos de bits para las subfilas. Cada flujo obtenido tendrá un ancho de 16 bits. Estos 8 flujos serán procesados por el bloque *Calculador de bytes para el Grupo 1*.

En el modo de 10 Gbps (STS-192 / STM-64) los bits de información para una fila son ordenados en 32 subfilas y 8 subfilas sucesivas formarán un grupo, por tanto se tendrán 4 grupos numerados del 1 al 4. El flujo de datos entrante debe ser separado en 32 flujos, y cada uno tendrá un ancho de 4 bits. Los 8 flujos para el grupo 1 son obtenidos a partir de los bytes 1, 5, 9 y 13 del flujo de entrada. Para el grupo 2, a partir de los bytes 2, 6, 10 y 14. Para el grupo 3, a partir de los bytes 3, 7, 11 y 15. Y finalmente, para el grupo 4 son obtenidos a partir de los bytes 4, 8, 12 y 16.

El bloque *Calculador de bytes P1 para el Grupo 1* funciona en ambos modos de operación. En el modo de 2.5 Gbps procesa los 16 bytes del flujo de datos entrante por cada ciclo de reloj y en el modo de 10 Gbps procesa 4 bytes del flujo de datos entrante por cada ciclo de reloj.

Los bloques *Calculador de bytes P1 para el Grupo 2, Grupo 3 y Grupo 4* sólo funcionan en el modo de operación de 10 Gbps (STS-192 / STM-64). Cada bloque procesa 4 bytes del flujo de datos entrante por cada ciclo de reloj.

Las posiciones definidas para los bytes P1 dentro de la tara de las tramas SONET/SDH obliga a almacenar el flujo de datos en *buffers*. Las posiciones de inserción de los bytes P1 calculados para la fila 9 se ubican casi al inicio de la misma, en las coordenadas S(9,1,c), S(9,2,c) y S(9,3,c) (ver sección 2.3.3). Esta fila es la que necesita el mayor retardo en el trayecto de los datos ya que es necesario procesar la fila por completo y posteriormente insertar los bytes P1 calculados en las posiciones descritas. El diseño y sincronización del resto de los bloques que conforman el módulo codificador se realizó en base al retardo de datos para esta fila, el cual se aproxima a la duración de una fila.

Los bytes P1 calculados son almacenados en registros internos en los bloques *Calculadores de bytes P1*. El contenido de estos registros se actualiza al inicio de cada fila con los bytes P1 calculados sobre el flujo de datos de la fila anterior. Los bytes P1 calculados para una cierta fila permanecen almacenados durante el tiempo de procesamiento del flujo de datos de la fila siguiente, como se muestra en la Figura 5.7.

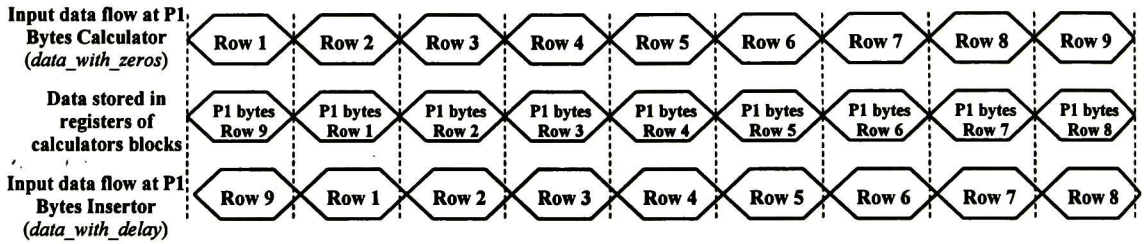


Figura 5.7 Flujos de datos entrantes a los bloques *Calculador de bytes P1* e *Insertor de bytes P1* y Contenido de los registros internos de los bloques *Calculadores de bytes P1*

El retardo introducido en el trayecto de los datos hace que el flujo de datos entrante al bloque *Insertor de bytes P1* se encuentre retardado aproximadamente una fila con respecto al flujo de datos entrante al bloque *Calculador de bytes P1*.

Por ejemplo, cuando el flujo de datos entrante al bloque *Calculador de bytes P1* coincide con los datos de la fila 4 y los registros internos de los bloques calculadores contienen los bytes P1 calculados para la fila 3, el flujo de datos entrante al bloque *Insertor de bytes P1* coincide con los datos de la fila 3. Este comportamiento en los flujos de datos y en el contenido de los registros internos sumado al hecho de que las posiciones de inserción de los bytes P1 de la fila 3 se ubican en la tara de la misma fila, permite que la inserción de los bytes P1 se realice como se describe a continuación: los bytes P1 almacenados en los registros internos de los bloques calculadores son enviados hacia el puerto de salida *P1_bytes_to_insert* del bloque *Calculador de bytes P1*, a través de los bloques multiplexores. Por este puerto el bloque *Insertor de bytes P1* recibe la información de paridad que insertará en el flujo de datos entrante.

El proceso que se ha descrito para la fila 3 se repite para las filas 5, 6, 7, 8 y 9, dado que las posiciones de inserción de los bytes P1 se ubican en la tara de las mismas filas.

Las posiciones de inserción de los bytes P1 para las filas restantes: 1, 2 y 4, se ubican en la tara de la fila siguiente. Por ejemplo, los bytes P1 de la fila 1 se insertan en la tara de la fila 2. Cuando el flujo de datos entrante al bloque *Insertor de bytes P1* coincide con los datos de la fila 2, los bytes P1 almacenados en los registros internos de los bloques calculadores corresponden a los de la fila 2 y no a los de la fila 1. Los bytes P1 para las filas 1, 2 y 4 deben ser guardados en otros registros para evitar su pérdida y permitir la inserción en las posiciones definidas. Por tanto, los bytes P1 para estas filas son guardados en registros externos a los bloques calculadores, en los bloques llamados *Registros de bytes P1*. En la Figura 5.8 se muestra como se actualiza el contenido de los *Registros de bytes P1*.

Para realizar la inserción, los bytes P1 son enviados desde los bloques *Registros de bytes P1* hacia el puerto de salida *P1_bytes_to_insert* del bloque *Calculador de bytes P1*. Este puerto de salida tiene un ancho de 104 bits, dado que los bytes P1 son insertados en subconjuntos de 13 bytes en el flujo de datos.

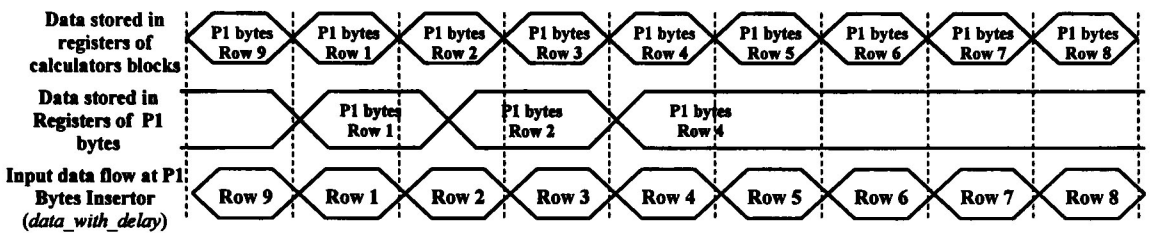


Figura 5.8 Contenido de los bloques *Registros de bytes P1*

A continuación se explican a detalle los bloques que forman el *Calculador de bytes P1*.

5.6.2 Calculador de bytes P1 para el Grupo 1

Este bloque realiza el cálculo de los bytes P1 sobre los bits de información que corresponden al Grupo 1. En la Figura 5.9 se presenta el diagrama de señales de entrada y de salida del bloque, y en la Tabla 5.4 se presenta la descripción de estas señales.

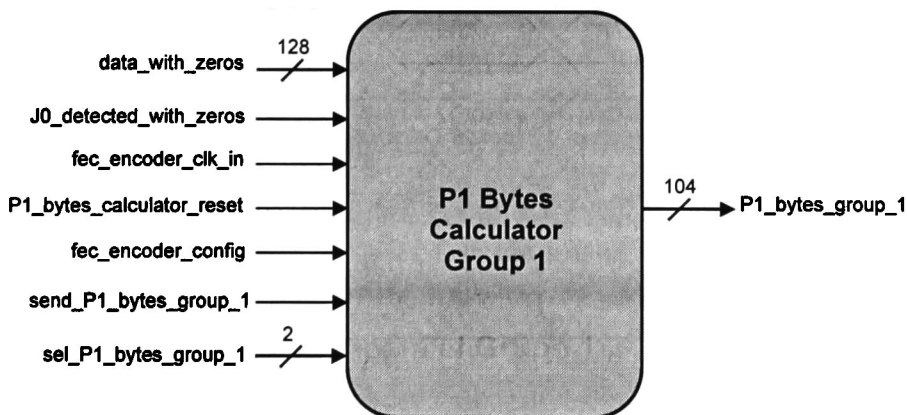


Figura 5.9 Diagrama de señales de entrada y de salida del bloque *Calculador de bytes P1 para el Grupo 1*

Entrada al bloque	Descripción	Origen
data_with_zeros [127:0]	Bus de datos de entrada	Insertor de ceros
J0_detected_with_zeros	Señal de sincronía	Insertor de ceros
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
P1_bytes_calculator_reset	Señal de reset activa en valor alto	Control
fec_encoder_config	Señal de configuración	Control
send_P1_bytes_group_1	Señal de control para el envío de bytes P1	Control del calculador de bytes P1
sel_P1_bytes_group_1 [1:0]	Señal de selección del subconjunto de bytes P1 a enviar	Control del calculador de bytes P1
Salida del bloque	Descripción	Destino
P1_bytes_group_1 [103:0]	Bus de datos de salida de bytes P1 para el grupo 1	Registro de bytes P1 para el grupo 1

Tabla 5.7 Descripción de las señales de entradas y salidas del bloque *Calculador de bytes P1 para el Grupo 1*

En la Figura 5.10 se presenta el diagrama de microarquitectura de este bloque.

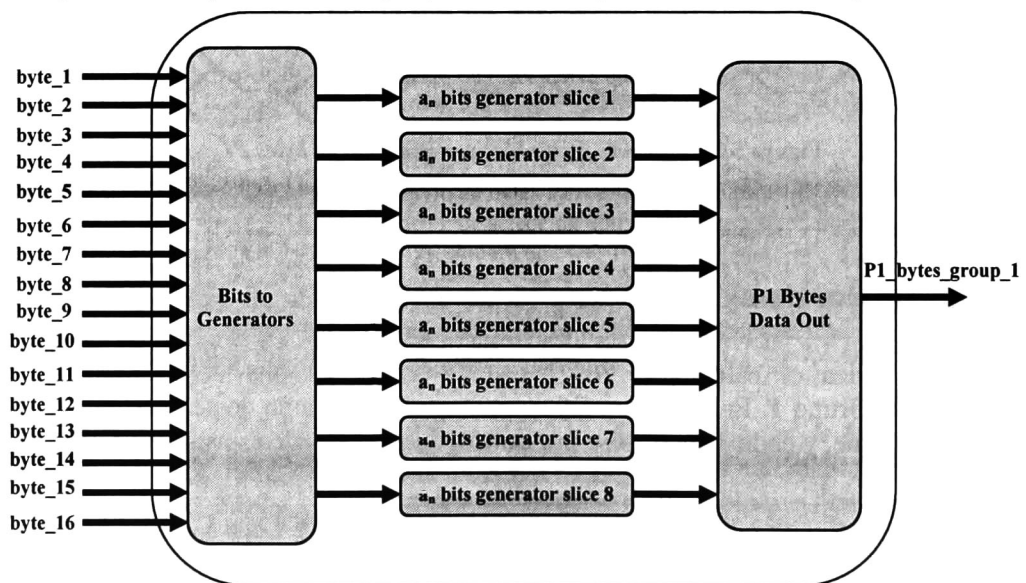


Figura 5.10 Microarquitectura del bloque *Calculador de bytes P1 para el Grupo 1*

El bloque *Bits to Generators* forma los 8 flujos de bits de información a partir del flujo de datos entrante *data_with_zeros*. En el modo de 2.5 Gbps en cada flujo se reciben 16 bits por ciclo de reloj y en el modo de 10 Gbps se reciben 4 bits por ciclo de reloj.

Cada flujo es procesado por un bloque *a_n bits generator*, procesa un total de 4320 bits de información y obtiene 39 bits de chequeo de paridad.

El diseño de los bloques *a_n bits generator* se basa en el circuito de codificación que se describe en la sección 3.4.5, en el cual la información a codificar se recibe de manera serial y el flujo de salida es serial e incluye tanto los bits de información como los bits de chequeo de paridad. Cada bloque *a_n bits generator* recibe la información a codificar de forma paralela. Procesa 16 bits por ciclo de reloj en el modo de 2.5 Gbps y 4 bits por ciclo de reloj en el modo de 10 Gbps. En la Figura 5.11 se muestra el diseño del bloque *a_n bits generator slice 1*. Este diseño aplica para los 8 bloques *a_n bits generator* para el grupo 1.

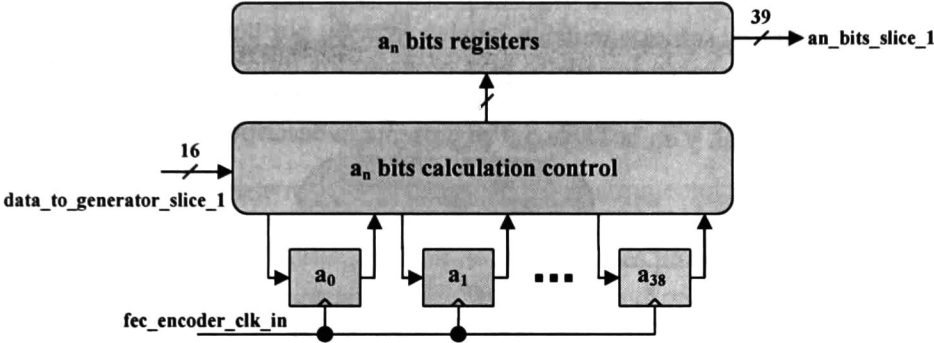


Figura 5.11 Diseño del bloque *a_n bits generator slice 1* para el grupo 1

El bloque *a_n bits calculation control* actualiza el valor de los 39 registros de corrimiento mediante una lógica combinacional que calcula el nuevo valor en base al valor actual de los registros, a los datos de entrada y al polinomio generador del código. Al término del procesamiento de los 4320 bits que forman la subfila o *slice*, este bloque inicializa los registros de corrimiento con un valor “cero” para calcular los bits de paridad de la subfila siguiente. Los bits de paridad obtenidos son almacenados en el bloque *a_n bits registers* y pueden ser leídos por bloques externos a través del puerto *an_bits_slice_1*. El contenido del bloque *a_n bits registers* se actualiza al término del procesamiento de cada subfila.

Por el puerto de salida *P1_bytes_group_1* se envían los bytes P1 en subconjuntos de 13 bytes. El primer subconjunto (S1) va del byte P1(a₁₂) al P1(a₀), el segundo subconjunto (S2) va del byte P1(a₂₅) al P1(a₁₃) y el tercer subconjunto (S3) va del byte P1(a₃₈) al P1(a₂₆).

El bloque *P1 Bytes Data Out* lee los bits de chequeo de paridad que obtienen los bloques *a_n bits generators*. A partir de ellos forma los bytes P1 y los envía hacia bloques externos mediante las señales de control *send_P1_bytes_group_1* y *sel_P1_bytes_group_1*. La señal *send_P1_bytes_group_1* es activa en valor alto. La señal *sel_P1_bytes_group_1* es una entrada de selección para escoger el subconjunto de bytes P1 que se enviará hacia bloques externos. En la Tabla 5.8 se muestra el subconjunto enviado para diferentes valores de esta señal de selección. El valor “00” se considera inválido.

Valor de la señal <i>sel_P1_bytes_group_1</i>	Subconjunto enviado hacia bloques externos
“01”	P1(a ₁₂) a P1(a ₀)

“10”	P1(a ₂₅) a P1(a ₁₃)
“11”	P1(a ₃₈) a P1(a ₂₆)

Tabla 5.8 Subconjuntos de bytes P1 para diferentes valores de la señal *sel_P1_bytes_group_1*

5.6.3 Calculador de bytes P1 para el Grupo 2, Grupo 3 y Grupo 4

Estos bloques calculadores son exactamente iguales entre si, por tanto se diseñó un bloque general a partir del cual se hacen instancias para procesar los flujos de bits de información para los grupos 2, 3 y 4. En la Figura 5.12 se presenta el diagrama de señales de entrada y de salida del bloque general, y en la Tabla 5.9 se presenta la descripción de estas señales.

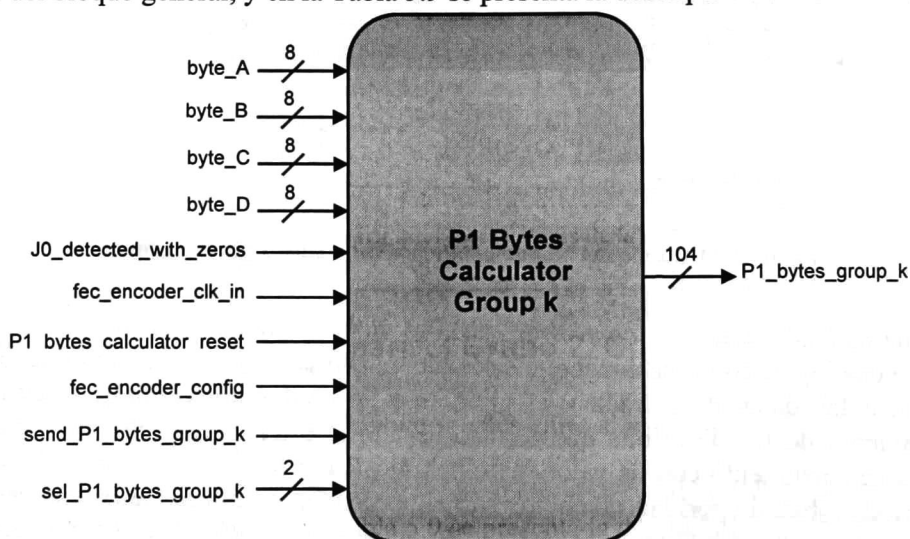


Figura 5.12 Diagrama de señales de entrada y de salida del bloque *Calculador de bytes P1 para el Grupo k* ($2 \leq k \leq 4$)

Entrada al bloque	Descripción	Origen
byte_A [7:0], byte_B [7:0], byte_C [7:0], byte_D [7:0]	Buses de datos de entrada	Insertor de ceros
J0_detected_with_zeros	Señal de sincronía	Insertor de ceros
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
P1_bytes_calculator_reset	Señal de reset activa en valor alto	Control
fec_encoder_config	Señal de configuración	Control
send_P1_bytes_group_k	Señal de control para el envío de bytes P1	Control del calculador de bytes P1
sel_P1_bytes_group_k [1:0]	Señal de selección del subconjunto de bytes P1 a enviar	Control del calculador de bytes P1
Salida del bloque	Descripción	Destino
P1_bytes_group_k [103:0]	Bus de datos de salida de bytes P1 para el grupo k	Registro de bytes P1 para el grupo k

Tabla 5.9 Descripción de las señales de entrada y de salida del bloque *Calculador de bytes P1 para el Grupo k* ($2 \leq k \leq 4$)

En la Tabla 5.10 se muestra la correspondencia entre las señales de entrada *byte_A*, *byte_B*, *byte_C* y *byte_D*, y los bytes del flujo de datos entrante *data_with_zeros*.

Señal de entrada	Calculador para el Grupo 2	Calculador para el Grupo 3	Calculador para el Grupo 4
<i>byte_A</i>	Byte 2	Byte 3	Byte 4
<i>byte_B</i>	Byte 6	Byte 7	Byte 8
<i>byte_C</i>	Byte 10	Byte 11	Byte 12
<i>byte_D</i>	Byte 14	Byte 15	Byte 16

Tabla 5.10 Correspondencia entre las señales *byte_A*, *byte_B*, *byte_C* y *byte_D*, y el flujo de datos entrante *data_with_zeros*

En la Figura 5.13 se presenta el diagrama de microarquitectura del bloque general. El bloque *Bits to Generators* forma los 8 flujos de bits de información a partir de los 4 bytes de datos de entrada. Cada flujo tiene un ancho de 4 bits y es procesado por un bloque *a_n bits generator*.

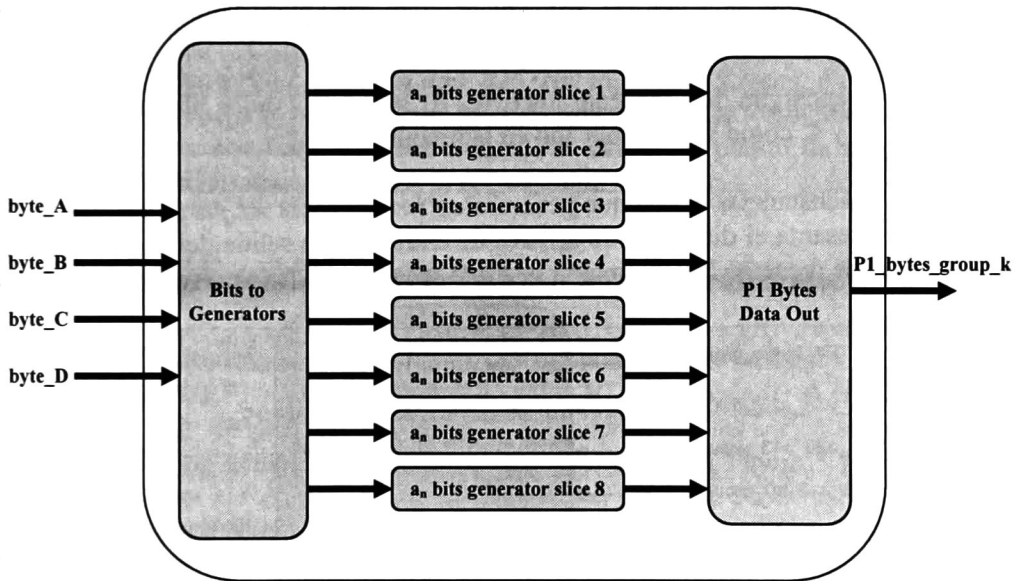


Figura 5.13 Microarquitectura del bloque *Calculador de bytes P1 para el Grupo k*

Los bloques *a_n bits generator* que integran el *Calculador de bytes P1 para el grupo k* presentan un diseño muy similar a los bloques que integran el *Calculador de bytes P1 para el grupo 1*. La diferencia está en que el bus de datos de entrada es de 4 bits ya que sólo funciona en el modo de 10 Gbps. En la Figura 5.14 se muestra el diseño de este bloque.

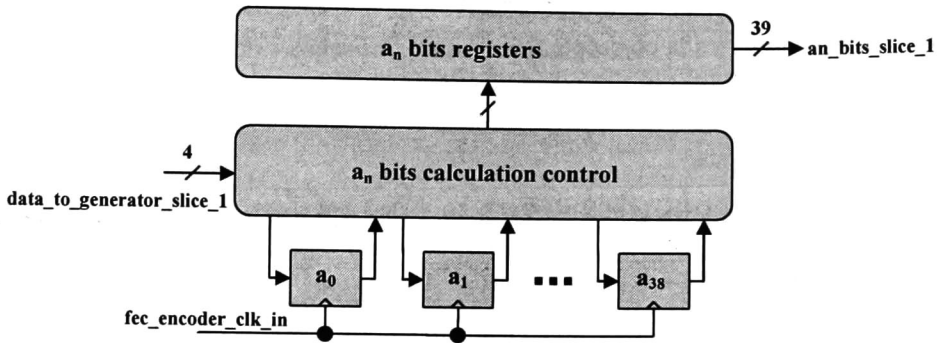


Figura 5.14 Diseño del bloque a_n bits generator slice 1 para el grupo k ($2 \leq k \leq 4$)

El bloque *P1 Bytes Data Out* es idéntico al descrito en la sección 5.6.2

5.6.4 Registro de bytes P1 para el Grupo 1, Grupo 2, Grupo 3 y Grupo 4

Estos bloques se encargan de almacenar los bytes P1 que calculan los bloques calculadores para las filas 1, 2 y 4, como se ha explicado en la sección 5.6.1.

Estos bloques son instancias del bloque general *Registro de bytes P1 para el Grupo k*. En la Figura 5.15 se presenta el diagrama de señales de entrada y de salida del bloque general, y en la Tabla 5.11 se presenta la descripción de estas señales.

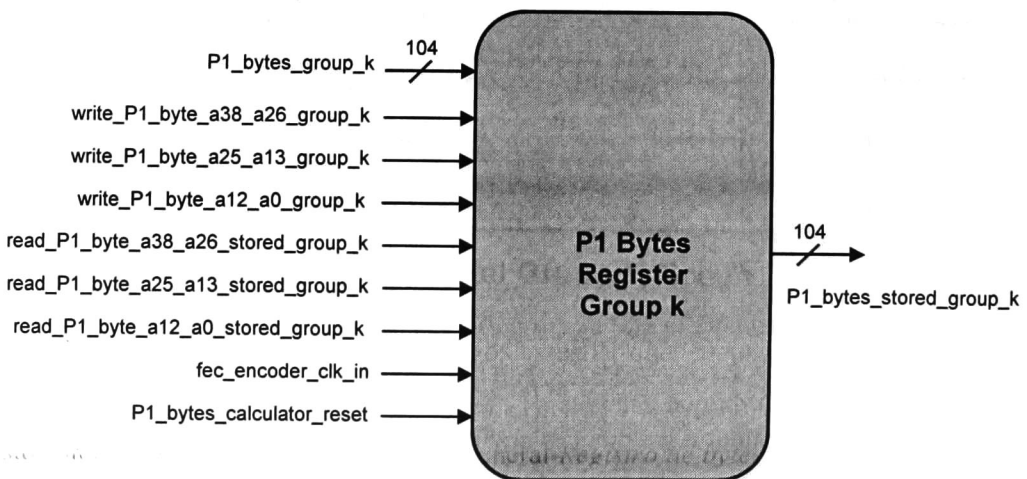


Figura 5.15 Diagrama de señales de entrada y de salida del bloque *Registro de bytes P1 para el Grupo k* ($1 \leq k \leq 4$)

Entrada al bloque	Descripción	Origen
P1_bytes_group_k [103:0]	Bus de datos de entrada de bytes P1 para el grupo k	Calculador de bytes P1 para el grupo k
write_P1_byte_a38_a26_group_k,	Señales de escritura para	Control del calculador

write_P1_byte_a25_a13_group_k, write_P1_byte_a12_a0_group_k	almacenar los bytes P1	de bytes P1
read_P1_byte_a38_a26_stored_group_k, read_P1_byte_a25_a13_stored_group_k, read_P1_byte_a12_a0_stored_group_k	Señales de lectura para enviar los bytes P1 almacenados	Control del calculador de bytes P1
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
P1_bytes_calculator_reset	Señal de reset activa en valor alto	Control
Salida del bloque	Descripción	Destino
P1_bytes_stored_group_k [103:0]	Bus de datos de salida de bytes P1 almacenados para el grupo k	Multiplexor del grupo k

Tabla 5.11 Descripción de las señales de entrada y de salida del bloque *Registro de bytes P1 para el Grupo k* ($1 \leq k \leq 4$)

Las señales de escritura *write_P1_byte_a38_a26_group_k*, *write_P1_byte_a25_a13_group_k* y *write_P1_byte_a12_a0_group_k* permiten almacenar los bytes presentes en el flujo de entrada *P1_bytes_group_k* en registros internos. Estas señales son activas en valor alto.

Las señales de lectura *read_P1_byte_a38_a26_stored_group_k*, *read_P1_byte_a25_a13_stored_group_k* y *read_P1_byte_a12_a0_stored_group_k* envían los bytes P1 almacenados en los registros internos hacia bloques externos a través del puerto de salida *P1_bytes_stored_group_k*. Estas señales son activas en valor alto.

5.6.5 Multiplexor del Grupo 1, Grupo 2, Grupo 3 y Grupo 4

Estos bloques multiplexan los flujos de datos entrantes provenientes de los bloques calculadores de bytes P1 y de los registros de bytes P1, para permitir el envío de los bytes P1 hacia el puerto de salida del bloque *Calculador de bytes P1*. Estos bloques son instancias del bloque general *Multiplexor del Grupo k*. En la Figura 5.16 se presenta el diagrama de señales de entrada y salida del bloque, y en la Tabla 5.12 se presenta la descripción de estas señales.

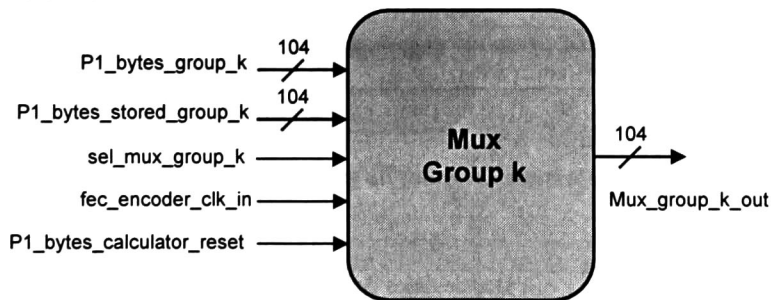


Figura 5.16 Diagrama de señales de entrada y de salida del bloque *Multiplexor del Grupo k* ($1 \leq k \leq 4$)

Entrada al bloque	Descripción	Origen
P1_bytes_group_k [103:0]	Bus de datos de entrada	Calculador de bytes P1

		para el grupo k
P1_bytes_stored_group_k [103:0]	Bus de datos de entrada	Registro de bytes P1 para el grupo k
sel_mux_group_k	Señal de selección del flujo de bytes P1	Control del calculador de bytes P1
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
P1_bytes_calculator_reset	Señal de reset activa en valor alto	Control
Salida del bloque	Descripción	Destino
Mux_group_k_out [103:0]	Bus de datos de salida de bytes P1 para el grupo k	Multiplexor de grupos

Tabla 5.12 Descripción de las señales de entrada y de salida del bloque *Multiplexor del Grupo k* ($1 \leq k \leq 4$)

La señal *sel_mux_group_k* es la entrada de selección que permite al bloque realizar su función de multiplexación sobre los flujos de datos. Con un valor bajo (“0”) se selecciona el flujo *P1_bytes_group_k* proveniente del bloque calculador. Con un valor alto (“1”) se selecciona el flujo *P1_bytes_data_stored_group_k* proveniente del bloque de registros.

5.6.6 Multiplexor de Grupos

Este bloque realiza la multiplexación de los 4 flujos provenientes de los bloques multiplexores de los grupos del 1 al 4. En la Figura 5.17 se presenta el diagrama de señales de entrada y de salida del bloque, y en la Tabla 5.13 se presenta la descripción de estas señales.

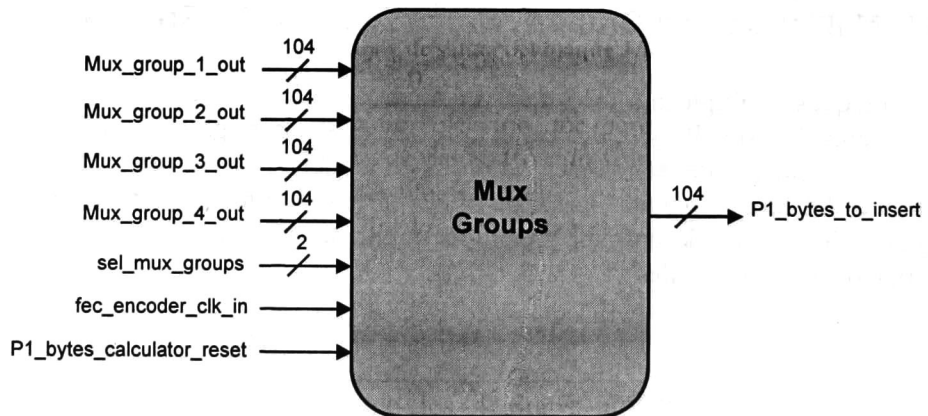


Figura 5.17 Diagrama de señales de entrada y de salida del bloque *Multiplexor de Grupos*

Entrada al bloque	Descripción	Origen
Mux_group_1_out [103:0]	Bus de datos de entrada	Multiplexor del grupo 1
Mux_group_2_out [103:0]	Bus de datos de entrada	Multiplexor del grupo 2
Mux_group_3_out [103:0]	Bus de datos de entrada	Multiplexor del grupo 3
Mux_group_4_out [103:0]	Bus de datos de entrada	Multiplexor del grupo 4
sel_mux_groups [1:0]	Señal de selección del flujo de bytes P1	Control del calculador de bytes P1
fec_encoder_clk_in	Señal de reloj del sistema	Exterior

P1_bytes_calculator_reset	Señal de reset activa en valor alto	Control
Salida del bloque	Descripción	Destino
Mux_group_k_out [103:0]	Bus de datos de salida de bytes P1 para el grupo k	Multiplexor de grupos

Tabla 5.13 Descripción de las señales de entrada y de salida del bloque *Multiplexor de Grupos*

La señal *sel_mux_groups*, que tiene un ancho de 2 bits, es la entrada de selección del bloque. En la Tabla 5.14 se muestra el flujo seleccionado para cada valor de la señal de entrada.

Valor de la señal <i>sel_mux_groups</i>	Flujo seleccionado
"00"	<i>Mux_group_1_out</i>
"01"	<i>Mux_group_2_out</i>
"11"	<i>Mux_group_3_out</i>
"10"	<i>Mux_group_4_out</i>

Tabla 5.14 Flujo seleccionado para cada valor de la señal de entrada *sel_mux_groups*

5.6.7 Control del calculador de bytes P1

Este bloque genera todas las señales de control para el resto de los bloques que integran el *Calculador de bytes P1*. Las máquinas de estados que generan las señales de control se diseñaron para sincronizar los flujos de datos entre los bloques que integran el *Calculador de bytes P1*, para así enviar los bytes P1 hacia el bloque *Insertor de bytes P1* en el orden y en el tiempo correcto. En la Figura 5.18 se presenta el diagrama de señales de entrada y de salida del bloque.

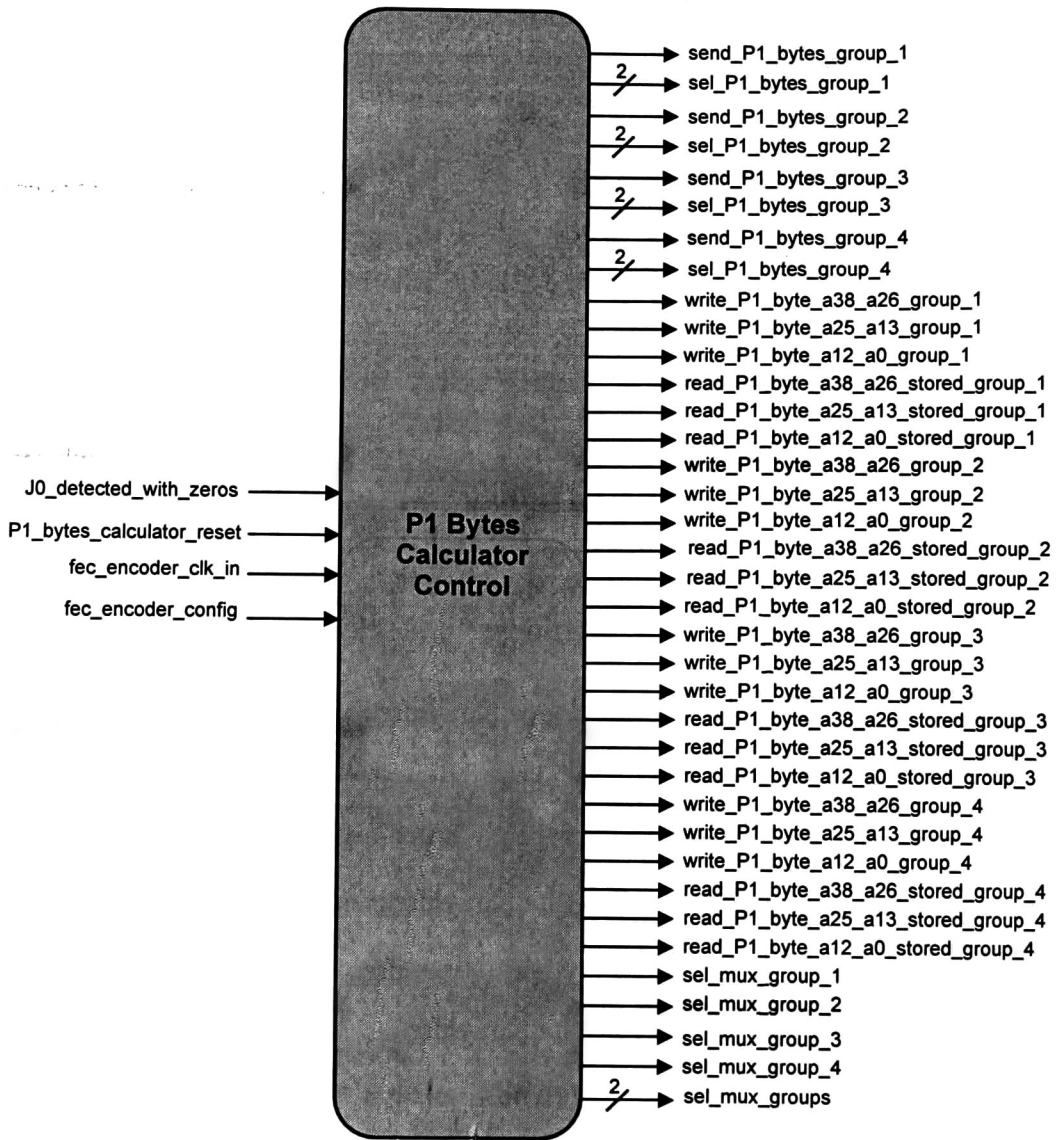


Figura 5.18 Diagrama de señales de entrada y de salida del bloque *Control del calculador de bytes P1*

En la Tabla 5.15 se detallan los bloques a los que se destinan las señales de salida del bloque *Control del calculador de bytes P1*.

Salida del bloque	Destino
send_P1_bytes_group_1 sel_P1_bytes_group_1 [1:0]	Calculador de bytes P1 para el grupo 1
send_P1_bytes_group_2 sel_P1_bytes_group_2 [1:0]	Calculador de bytes P1 para el grupo 2
send_P1_bytes_group_3 sel_P1_bytes_group_3 [1:0]	Calculador de bytes P1 para el grupo 3

send_P1_bytes_group_4 sel_P1_bytes_group_4 [1:0]	Calculador de bytes P1 para el grupo 4
write_P1_byte_a38_a26_group_1 write_P1_byte_a25_a13_group_1 write_P1_byte_a12_a0_group_1 read_P1_byte_a38_a26_stored_group_1 read_P1_byte_a25_a13_stored_group_1 read_P1_byte_a12_a0_stored_group_1	Registro de bytes P1 para el grupo 1
write_P1_byte_a38_a26_group_2 write_P1_byte_a25_a13_group_2 write_P1_byte_a12_a0_group_2 read_P1_byte_a38_a26_stored_group_2 read_P1_byte_a25_a13_stored_group_2 read_P1_byte_a12_a0_stored_group_2	Registro de bytes P1 para el grupo 2
write_P1_byte_a38_a26_group_3 write_P1_byte_a25_a13_group_3 write_P1_byte_a12_a0_group_3 read_P1_byte_a38_a26_stored_group_3 read_P1_byte_a25_a13_stored_group_3 read_P1_byte_a12_a0_stored_group_3	Registro de bytes P1 para el grupo 3
write_P1_byte_a38_a26_group_4 write_P1_byte_a25_a13_group_4 write_P1_byte_a12_a0_group_4 read_P1_byte_a38_a26_stored_group_4 read_P1_byte_a25_a13_stored_group_4 read_P1_byte_a12_a0_stored_group_4	Registro de bytes P1 para el grupo 4
sel_mux_group_1	Multiplexor del grupo 1
sel_mux_group_2	Multiplexor del grupo 2
sel_mux_group_3	Multiplexor del grupo 3
sel_mux_group_4	Multiplexor del grupo 4
sel_mux_groups [1:0]	Multiplexor de grupos

Tabla 5.15 Destino de las señales de salida del bloque *Control del calculador de bytes P1*

El diseño de este bloque toma en cuenta los ciclos de inserción de los bytes P1. En la Tabla 5.16 se presentan todos los ciclos de inserción para ambos modos de operación: 2.5 y 10 Gbps.

Subconjunto de bytes P1		Ciclos de inserción				
		Modo 2.5 Gbps	Modo 10 Gbps			
		Grupo 1	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Fila 1	P1(a ₃₈) a P1(a ₂₆)	270	1080	1081	1082	1083
	P1(a ₂₅) a P1(a ₁₃)	273	1092	1093	1094	1095
	P1(a ₁₂) a P1(a ₀)	275	1100	1101	1102	1103
Fila 2	P1(a ₃₈) a P1(a ₂₆)	540	2160	2161	2162	2163

	P1(a ₂₅) a P1(a ₁₃)	543	2172	2173	2174	2175
	P1(a ₁₂) a P1(a ₀)	545	2180	2181	2182	2183
Fila 3	P1(a ₃₈) a P1(a ₂₆)	546	2184	2185	2186	2187
	P1(a ₂₅) a P1(a ₁₃)	547	2188	2189	2190	2191
	P1(a ₁₂) a P1(a ₀)	548	2192	2193	2194	2195
Fila 4	P1(a ₃₈) a P1(a ₂₆)	1083	4332	4333	4334	4335
	P1(a ₂₅) a P1(a ₁₃)	1084	4336	4337	4338	4339
	P1(a ₁₂) a P1(a ₀)	1085	4340	4341	4342	4343
Fila 5	P1(a ₃₈) a P1(a ₂₆)	1086	4344	4345	4346	4347
	P1(a ₂₅) a P1(a ₁₃)	1087	4348	4349	4350	4351
	P1(a ₁₂) a P1(a ₀)	1088	4352	4353	4354	4355
Fila 6	P1(a ₃₈) a P1(a ₂₆)	1356	5424	5425	5426	5427
	P1(a ₂₅) a P1(a ₁₃)	1357	5428	5429	5430	5431
	P1(a ₁₂) a P1(a ₀)	1358	5432	5433	5434	5435
Fila 7	P1(a ₃₈) a P1(a ₂₆)	1626	6504	6505	6506	6507
	P1(a ₂₅) a P1(a ₁₃)	1627	6508	6509	6510	6511
	P1(a ₁₂) a P1(a ₀)	1628	6512	6513	6514	6515
Fila 8	P1(a ₃₈) a P1(a ₂₆)	1896	7584	7585	7586	7587
	P1(a ₂₅) a P1(a ₁₃)	1897	7588	7589	7590	7591
	P1(a ₁₂) a P1(a ₀)	1898	7592	7593	7594	7595
Fila 9	P1(a ₃₈) a P1(a ₂₆)	2160	8640	8641	8642	8643
	P1(a ₂₅) a P1(a ₁₃)	2161	8644	8645	8646	8647
	P1(a ₁₂) a P1(a ₀)	2162	8648	8649	8650	8651

Tabla 5.16 Ciclos de inserción de bytes P1

5.6.7.1 Señales de control para el envío de bytes P1 en el modo de operación de 2.5 Gbps

Se ha mencionado que la inserción de bytes P1 para fila 9 es la determina el retardo que se debe introducir en el trayecto de los datos. A fin de minimizar este retardo, los bytes P1 calculados para esta fila deben ser enviados hacia el bloque *Insertor de bytes P1* lo más pronto posible.

De acuerdo a la Tabla 5.16, los ciclos de inserción para los bytes P1 de la fila 9 en el modo de operación de 2.5 Gbps son los siguientes: el subconjunto S3 de bytes P1(a₃₈) a P1(a₂₆) debe ser insertado en el ciclo 2160, posteriormente el subconjunto S2 de bytes P1(a₂₅) a P1(a₁₃) en el ciclo 2161 y finalmente el subconjunto S1 de bytes P1(a₁₂) a P1(a₀) en el ciclo 2162.

En la Figura 5.19 se muestra el diagrama de tiempo en donde se detalla la generación de las señales de control que permiten el envío de los bytes P1 para la fila 9. Los últimos datos de la fila 9, y por consiguiente de la trama x , se hacen presentes en el flujo de datos *data_with_zeros* durante el ciclo 2429. A partir del inicio de la trama siguiente ($x + 1$) y durante todo el procesamiento de la fila 1, los registros internos del bloque calculador para el grupo 1 tienen almacenados los bytes P1 calculados para la fila anterior (fila 9 de la trama x). Para enviar estos bytes hacia el puerto de salida del bloque *Calculador de bytes*

P1 primeramente deben ser leídos de los registros internos mediante las señales de control *send_P1_bytes_group_1* y *sel_P1_bytes_group_1*.

La señal *send_P1_bytes_group_1* es puesta a un valor alto en los ciclos del 0 al 2 para solicitar el envío de los bytes P1. La señal *sel_P1_bytes_group_1* toma el valor "11" en el ciclo 0 para enviar el subconjunto S3 de bytes P1, luego toma el valor "10" en el ciclo 1 para enviar el subconjunto S2 de bytes P1 y por último toma el valor "01" en el ciclo 2 para enviar el subconjunto S1 de bytes P1. Los subconjuntos de bytes P1 se hacen presentes en el bus *P1_bytes_group_1* un ciclo después de su solicitud.

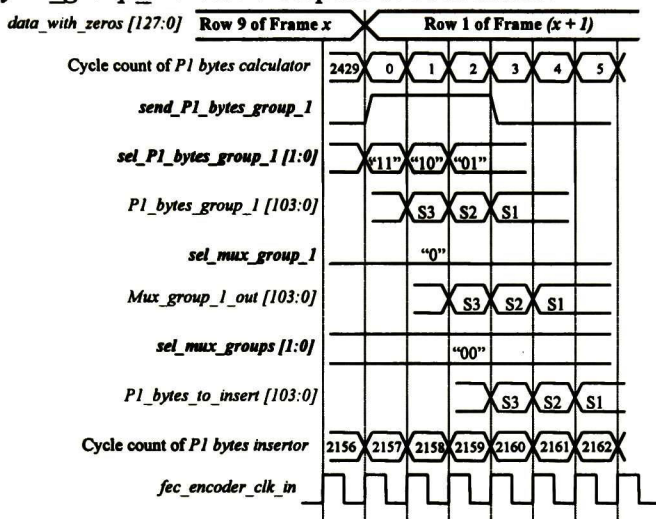


Figura 5.19 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 9 en el modo de operación de 2.5 Gbps

La entrada de selección *sel_mux_group_1* del bloque *Multiplexor del grupo 1* es puesta a un valor bajo para que el flujo de datos a su salida, *Mux_group_1_out*, corresponda al flujo entrante *P1_bytes_group_1*. El flujo de salida *Mux_group_1_out* se encuentra retrasado un ciclo de reloj con respecto al flujo entrante *P1_bytes_group_1* debido al diseño del bloque multiplexor.

La entrada de selección *sel_mux_groups* del bloque *Multiplexor de grupos* es puesta a valor "00" para escoger el flujo de datos entrante *Mux_group_1_out*. De esta forma, en el flujo de salida *P1_bytes_to_insert* se presentan los subconjuntos de bytes P1 en los ciclos del 3 al 5. Este flujo de salida se interconecta con el bloque *Insertor de bytes P1*, por tanto cuando el contador de ciclos en el bloque calculador coincide con el ciclo 3 y en el puerto *P1_bytes_to_insert* se encuentra presente el subconjunto S3 de bytes P1, el contador de ciclos en el bloque insertor debe coincidir con el ciclo 2160 para permitir insertar correctamente estos bytes en el flujo de datos. El desfase entre los contadores de ciclos del bloque calculador y del bloque insertor, para el modo de operación de 2.5 Gbps, es de 273 ciclos. Este desfase es el que determina el retardo que se introduce en el trayecto de los datos.

La generación de señales de control para las filas 3, 5, 6, 7 y 8 es similar al proceso descrito para la fila 9. Los diagramas de tiempo para estas filas se presentan en el Anexo A.

En la Figura 5.20 se muestra el diagrama de tiempo en donde se detalla la generación de las señales de control que permiten el envío de los bytes P1 para la fila 1. Los bytes P1 son almacenados en registros externos a los bloques calculadores, como se ha explicado en la sección 5.6.1.

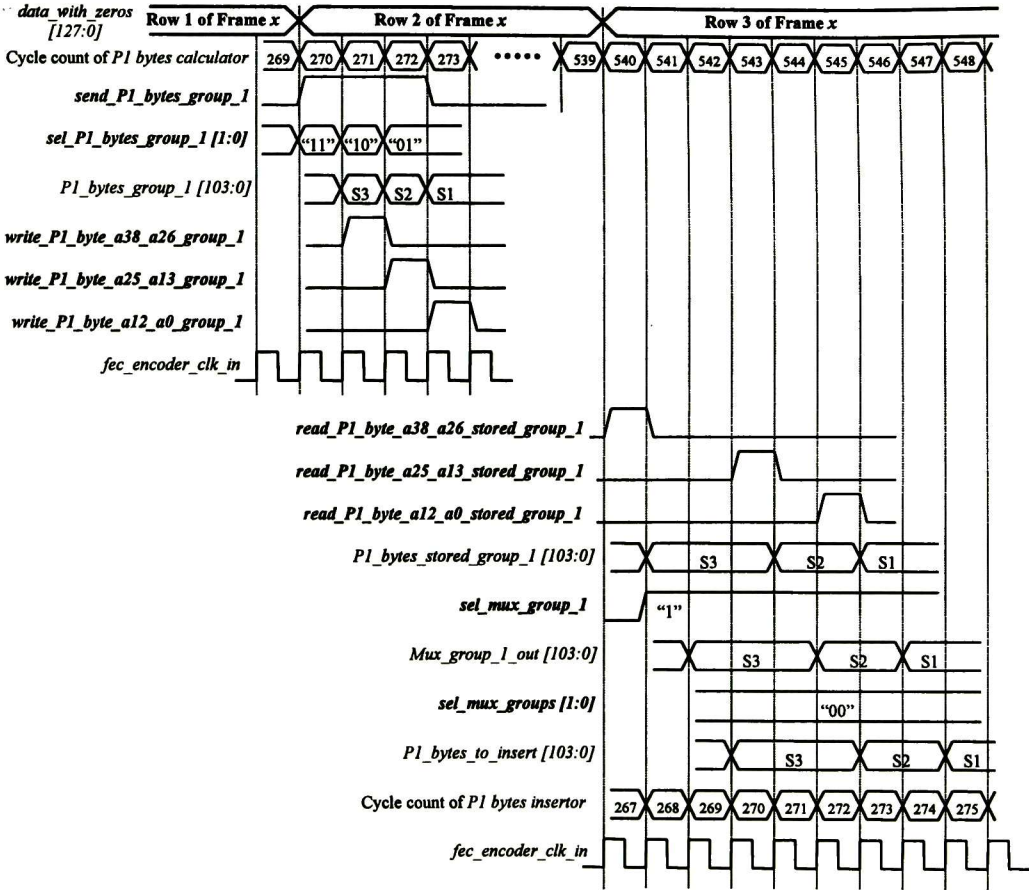


Figura 5.20 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 1 en el modo de operación de 2.5 Gbps

La señal *send_P1_bytes_group_1* es puesta a un valor alto del ciclo 270 al 272 para solicitar el envío de los bytes P1. La señal *sel_P1_bytes_group_1* toma el valor "11" en el ciclo 270 para enviar el subconjunto S3 de bytes P1, luego toma el valor "10" en el ciclo 271 para enviar el subconjunto S2 de bytes P1 y por último toma el valor "01" en el ciclo 272 para enviar el subconjunto S1 de bytes P1. Los subconjuntos de bytes P1 se hacen presentes en el bus *P1_bytes_group_1* un ciclo después de su solicitud.

Los bytes P1 son entonces almacenados en el bloque *Registro de bytes P1 para el grupo 1* mediante señales de escritura. La señal *write_P1_byte_a38_a26_group_1* es puesta a un valor alto en el ciclo 271 para escribir en los registros el subconjunto S3 de bytes P1, los

cuales se encuentran presentes en el bus *P1_bytes_group_1*. De igual forma, la señal *write_P1_byte_a25_a13_group_1* es puesta a un valor alto en el ciclo 272 para escribir el subconjunto S2 y la señal *write_P1_byte_a12_a0_group_1* es puesta a un valor alto en el ciclo 273 para escribir el subconjunto S1.

El subconjunto S3 de bytes P1 debe estar presente en el ciclo 270 en el bloque insertor. Tomando en cuenta el desfase entre los contadores de ciclos del bloque calculador y del bloque insertor, las señales de control restantes deber ser generadas como se detalla a continuación. La señal *read_P1_byte_a38_a26_stored_group_1* es puesta a un valor alto en el ciclo 540 para leer el subconjunto S3 de bytes P1 almacenado en el bloque *Registro de bytes para el grupo 1* y enviarlo por el bus *P1_bytes_stored_group_1*. De igual forma, la señal *read_P1_byte_a25_a13_stored_group_1* es puesta a un valor alto en el ciclo 543 para leer el subconjunto S2 y la señal *read_P1_byte_a12_a0_stored_group_1* es puesta a un valor alto en el ciclo 545 para leer el subconjunto S1. Los subconjuntos de bytes P1 se hacen presentes en el bus *P1_bytes_stored_group_1* un ciclo después de su solicitud.

La entrada de selección *sel_mux_group_1* del bloque *Multiplexor del grupo 1* es puesta a un valor alto para que el flujo de datos a su salida, *Mux_group_1_out*, corresponda al flujo entrante *P1_bytes_stored_group_1*. El flujo de salida *Mux_group_1_out* se encuentra retrasado un ciclo de reloj con respecto al flujo entrante *P1_bytes_stored_group_1* debido al diseño del bloque multiplexor.

La entrada de selección *sel_mux_groups* del bloque *Multiplexor de grupos* es puesta a valor "00" para escoger el flujo de datos entrante *Mux_group_1_out*. De esta forma, en el flujo de salida *P1_bytes_to_insert* se presenta el subconjunto S3 de bytes P1 en el ciclo 543, el subconjunto S2 en el ciclo 546 y el subconjunto S1 en el ciclo 548. Los subconjuntos son recibidos en el bloque *Insertor de bytes P1* en los ciclos 270, 273 y 275.

La generación de señales de control para la filas 2 y 4 es similar al proceso descrito para la fila 1. Los diagramas de tiempo para estas filas se presentan en el Anexo A.

5.6.7.2 Señales de control para el envío de bytes P1 en el modo de operación de 10 Gbps

Ya se ha descrito la generación de las señales de control en el modo de operación de 2.5 Gbps. Ahora se describirá la generación de las señales de control en el modo de operación de 10 Gbps. En este modo de operación, los bits de información de cada fila se separan en cuatro grupos, para cada uno de ellos se obtiene un conjunto de bytes P1. Los ciclos de inserción de bytes P1 para los diferentes grupos se presentan en la Tabla 5.16.

Para el caso de la fila 9, el primer subconjunto de bytes P1 que se inserta es el subconjunto S3 del grupo 1 en el ciclo 8640. En el ciclo 8641 se inserta el subconjunto S3 del grupo 2, en el ciclo 8642 el subconjunto S3 del grupo 3 y en el ciclo 8643 el subconjunto S3 del grupo 4. De manera similar son insertados los subconjuntos S2 del ciclo 8644 al 8647 y los subconjuntos S1 del ciclo 8648 al 8651.

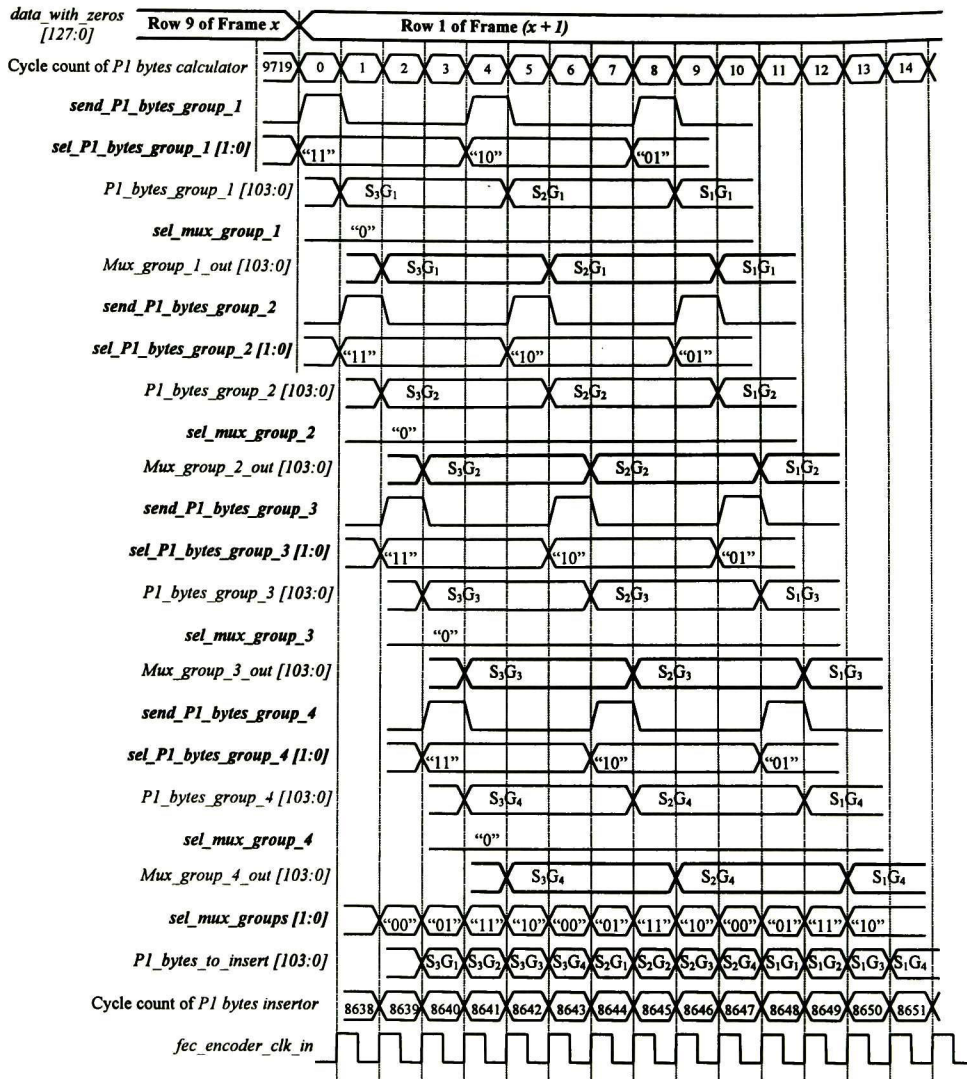


Figura 5.21 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 9 en el modo de operación de 10 Gbps

En la Figura 5.21 se muestra el diagrama de tiempo en donde se detalla la generación de las señales de control que permiten el envío de los bytes P1 para la fila 9. Las señales de control *send P1 bytes group 1*, *sel P1 bytes group 1*, *sel_mux_group 1* y *sel_mux groups* permiten que el subconjunto S3 de bytes P1 del grupo 1 se haga presente en el puerto de salida *P1 bytes to insert* cuando el contador del bloque calculador coincide con el ciclo 3. En este instante, el contador del bloque inserctor deberá coincidir con el ciclo 8640 para que el subconjunto S3 del grupo 1 sea insertado correctamente en el flujo de datos. El desfase entre el contador del bloque calculador y el contador del bloque inserctor, para el modo de operación de 10 Gbps, es de 1083 ciclos de reloj.

Las señales de control restantes deberán generarse de tal forma que los subconjuntos de bytes P1 de los cuatro grupos se hagan presentes en los ciclos correctos en el puerto de salida *P1_bytes_to_insert*. La generación de señales de control para las filas 3, 5, 6, 7 y 8 es similar al proceso descrito para la fila 9. Los diagramas de tiempo para estas filas se presentan en el Anexo A.

La generación de las señales de control para las filas 1, 2 y 4 implica la escritura de los bytes P1 en los bloques *Registros de bytes P1* y su posterior lectura para enviarlos por el puerto de salida *P1_bytes_to_insert*. En la Figura 5.22 se muestra el diagrama de tiempo en donde se detalla la generación de las señales de control que permiten la escritura de los bytes P1 para la fila 1 en los bloques *Registros de bytes P1* y en la Figura 5.23 se muestra el diagrama de tiempo en donde se detalla la generación de las señales de control que permiten la lectura de los bytes P1 almacenados y su envío hacia el puerto de salida *P1_bytes_to_insert*.

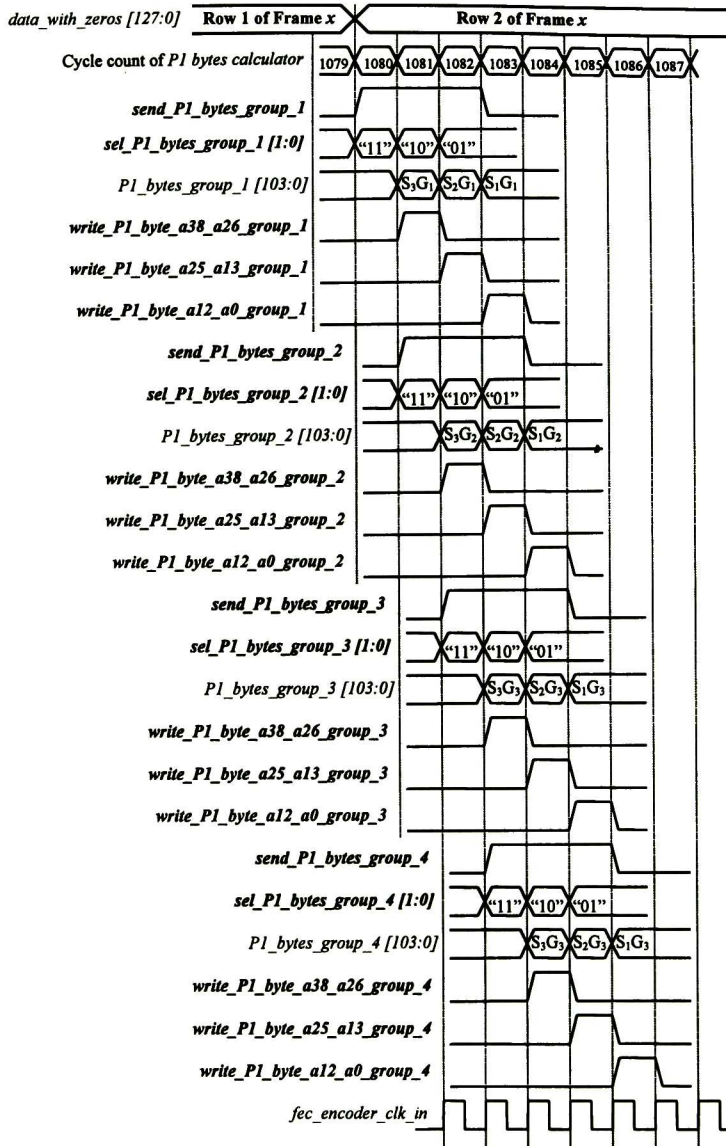


Figura 5.22 Diagrama de tiempo de las señales de control durante la escritura de los bytes P1 de la fila 1 en los bloques *Registros de bytes P1* en el modo de operación de 10 Gbps

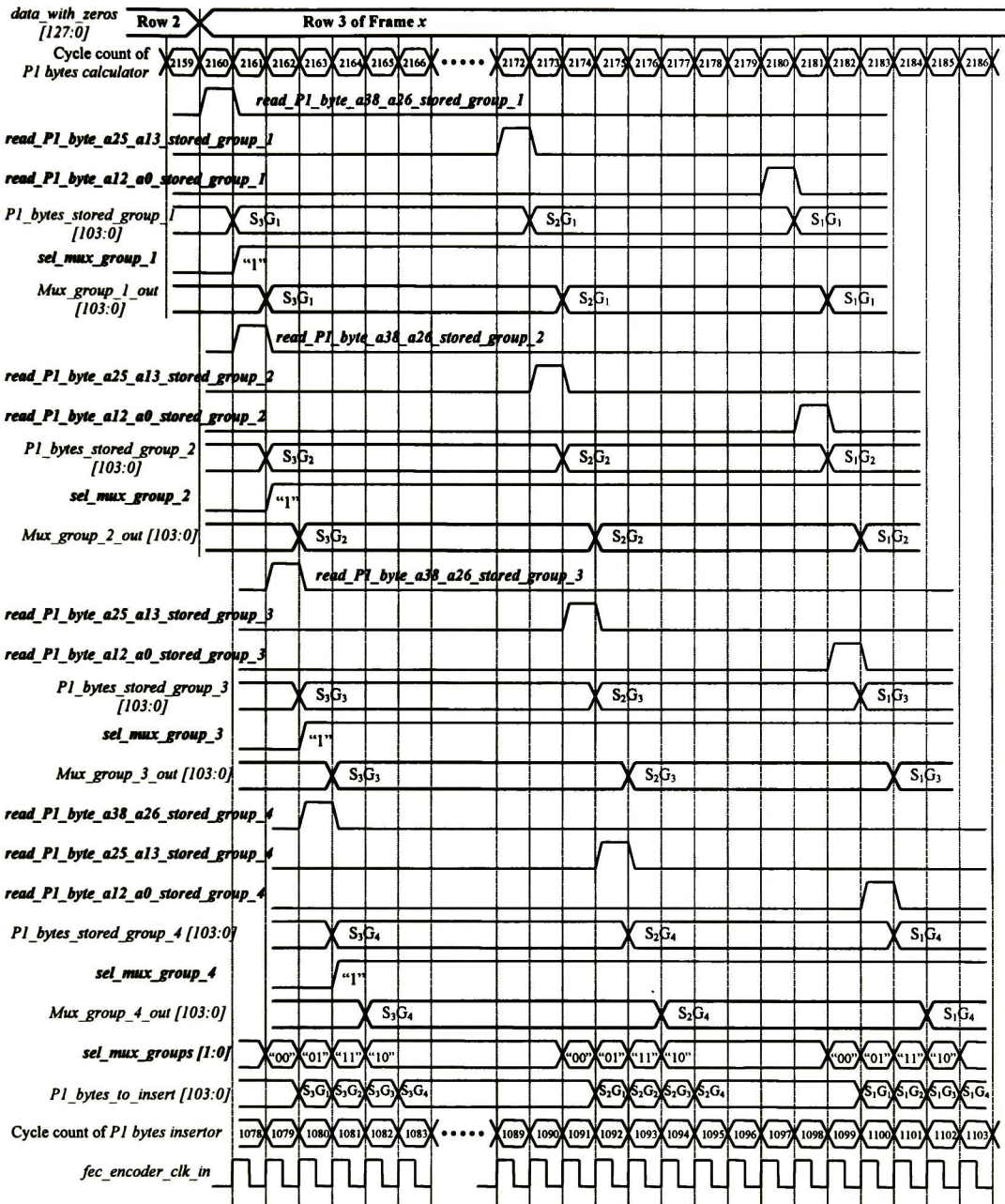


Figura 5.23 Diagrama de tiempo de las señales de control durante la lectura de los bytes P1 de la fila 1 almacenados en los bloques *Registros de bytes P1* en el modo de operación de 10 Gbps

La generación de señales de control para las filas 2 y 4 es similar al proceso descrito para la fila 1. Los diagramas de tiempo para estas filas se presentan en el Anexo A.

5.6.7.3 Máquinas de estados para la generación de las señales de control

5.6.7.3.1 Señal de control *send_P1_bytes_group_1*

La máquina de estados que genera la señal de control *send_P1_bytes_group_1* se muestra en la Figura 5.24.

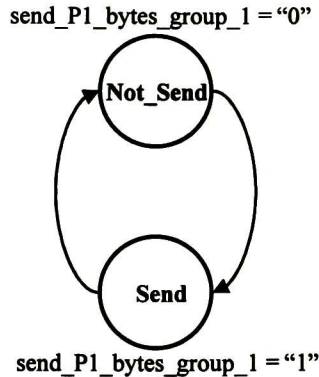


Figura 5.24 Máquina de estados para generar la señal de control *send_P1_bytes_group_1*

La máquina de estados es del tipo Moore, en donde la salida *send_P1_bytes_group_1* depende solamente del estado presente. En el estado *Not_Send* la salida toma el valor “0” y en el estado *Send* toma el valor “1”. El estado *Not_Send* es el estado inicial y las transiciones entre estados dependen del valor presente en el contador de ciclos y de la señal de configuración del módulo.

5.6.7.3.2 Señal de control *sel_P1_bytes_group_1*

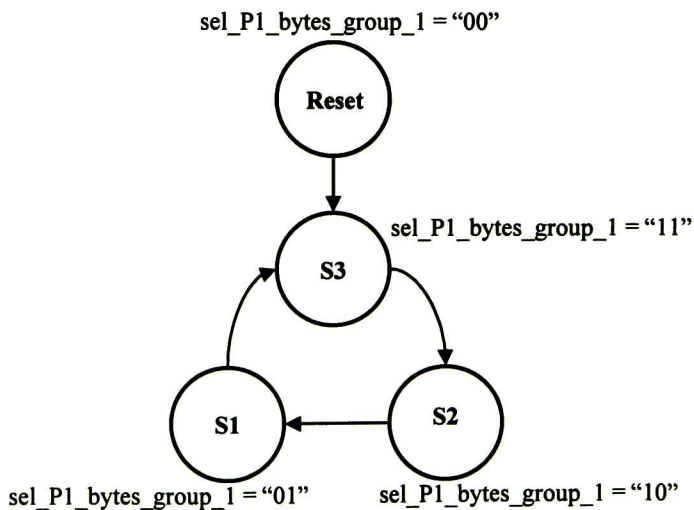


Figura 5.25 Máquina de estados para generar la señal de control *sel_P1_bytes_group_1*

La máquina de estados que genera la señal de control *sel_P1_bytes_group_1* se muestra en la Figura 5.25. Esta máquina es del tipo Moore, el valor que toma la señal de salida *sel_P1_bytes_group_1* en cada uno de los estados se muestra en la figura. Las transiciones entre estados dependen del valor presente en el contador de ciclos, de la señal de reset y de la señal de configuración del módulo.

5.6.7.3 Señales de control *write_P1_byte_a38_a26_group_1*, *write_P1_byte_a25_a13_group_1*, *write_P1_byte_a12_a0_group_1*, *read_P1_byte_a38_a26_stored_group_1*, *read_P1_byte_a25_a13_stored_group_1* y *read_P1_byte_a12_a0_stored_group_1*

La máquina de estados que genera estas señales de control se muestra en la Figura 5.26.

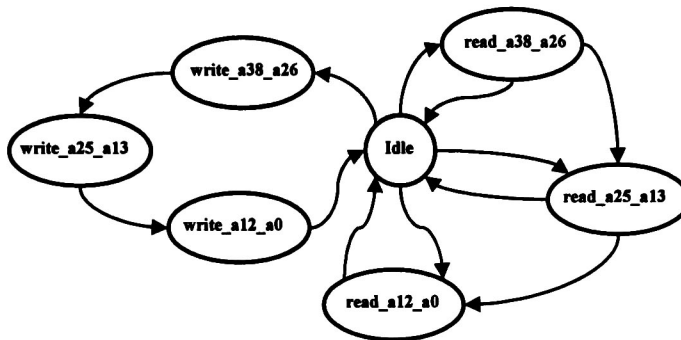


Figura 5.26 Máquina de estados para generar las señales de control *write_P1_byte_a38_a26_group_1*, *write_P1_byte_a25_a13_group_1*, *write_P1_byte_a12_a0_group_1*, *read_P1_byte_a38_a26_stored_group_1*, *read_P1_byte_a25_a13_stored_group_1* y *read_P1_byte_a12_a0_stored_group_1*

Esta máquina es del tipo Moore, en la Tabla 5.17 se muestran los valores que toman las señales de control en los diferentes estados.

Estado	Señales de control
Idle	<i>write_P1_byte_a38_a26_group_1</i> = "0" <i>write_P1_byte_a25_a13_group_1</i> = "0" <i>write_P1_byte_a12_a0_group_1</i> = "0" <i>read_P1_byte_a38_a26_stored_group_1</i> = "0" <i>read_P1_byte_a25_a13_stored_group_1</i> = "0" <i>read_P1_byte_a12_a0_stored_group_1</i> = "0"
<i>write_a38_a26</i>	<i>write_P1_byte_a38_a26_group_1</i> = "1" <i>write_P1_byte_a25_a13_group_1</i> = "0" <i>write_P1_byte_a12_a0_group_1</i> = "0" <i>read_P1_byte_a38_a26_stored_group_1</i> = "0" <i>read_P1_byte_a25_a13_stored_group_1</i> = "0" <i>read_P1_byte_a12_a0_stored_group_1</i> = "0"
<i>write_a25_a13</i>	<i>write_P1_byte_a38_a26_group_1</i> = "0" <i>write_P1_byte_a25_a13_group_1</i> = "1" <i>write_P1_byte_a12_a0_group_1</i> = "0" <i>read_P1_byte_a38_a26_stored_group_1</i> = "0"

	read_P1_byte_a25_a13_stored_group_1 = "0" read_P1_byte_a12_a0_stored_group_1 = "0"
write_a12_a0	write_P1_byte_a38_a26_group_1 = "0" write_P1_byte_a25_a13_group_1 = "0" write_P1_byte_a12_a0_group_1 = "1" read_P1_byte_a38_a26_stored_group_1 = "0" read_P1_byte_a25_a13_stored_group_1 = "0" read_P1_byte_a12_a0_stored_group_1 = "0"
read_a38_a26	write_P1_byte_a38_a26_group_1 = "0" write_P1_byte_a25_a13_group_1 = "0" write_P1_byte_a12_a0_group_1 = "0" read_P1_byte_a38_a26_stored_group_1 = "1" read_P1_byte_a25_a13_stored_group_1 = "0" read_P1_byte_a12_a0_stored_group_1 = "0"
read_a25_a13	write_P1_byte_a38_a26_group_1 = "0" write_P1_byte_a25_a13_group_1 = "0" write_P1_byte_a12_a0_group_1 = "0" read_P1_byte_a38_a26_stored_group_1 = "0" read_P1_byte_a25_a13_stored_group_1 = "1" read_P1_byte_a12_a0_stored_group_1 = "0"
read_a12_a0	write_P1_byte_a38_a26_group_1 = "0" write_P1_byte_a25_a13_group_1 = "0" write_P1_byte_a12_a0_group_1 = "0" read_P1_byte_a38_a26_stored_group_1 = "0" read_P1_byte_a25_a13_stored_group_1 = "0" read_P1_byte_a12_a0_stored_group_1 = "1"

Tabla 5.17 Valores de las señales de control en los diferentes estados

En el estado **Idle** todas las señales toman el valor "0". Para el resto de los estados, esta máquina se ha diseñado de forma que cada estado ejecute una operación de escritura o de lectura de bytes P1. Por ejemplo, el estado **write_a38_a26** ejecuta la operación de escritura del subconjunto S3 de bytes P1, para lo cual la señal *write_P1_byte_a38_a26_group_1* toma el valor "1" y las señales restantes el valor "0". El nombre de cada estado es un indicativo de la tarea que ejecuta y por consiguiente de los valores que deben tomar las señales de control. Las transiciones entre estados se obtienen de analizar las señales de control mostradas en las figuras de las secciones 5.6.7.1 y 5.6.7.2 y en el Anexo A.

5.6.7.3.4 Señal de control *sel_mux_group_1*

La máquina de estados que genera la señal de control *sel_mux_group_1* se muestra en la Figura 5.27. En el estado **Sel_Calc** la señal de control *sel_mux_group_1* toma el valor "0" y en el estado **Sel_Reg** toma el valor "1". El estado **Sel_Calc** es el estado inicial y las transiciones entre estados dependen del valor presente en el contador de ciclos y de la señal de configuración del módulo.

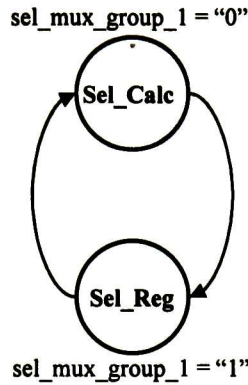


Figura 5.27 Máquina de estados para generar la señal de control *sel_mux_group_1*

5.6.7.3.5 Señal de control *sel_mux_groups*

La máquina de estados que genera la señal de control *sel_mux_groups* se muestra en la Figura 5.28.

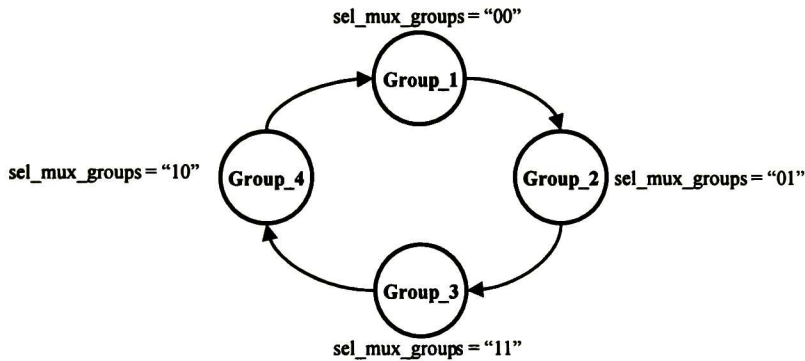


Figura 5.28 Máquina de estados para generar la señal de control *sel_mux_groups*

Los valores que toman las señales de control en cada uno de los estados se indican en la figura. Las transiciones entre estados dependen del valor presente en el contador de ciclos y de la señal de configuración del módulo.

5.6.7.3.6 Señales de control restantes

Analizando los diagramas de tiempo de la sección 5.6.7.2 y del anexo A, en donde se detalla la generación de las señales de control para el modo de operación de 10 Gbps, se puede observar que las variaciones en las señales de control del grupo 2 son idénticas a las variaciones que presentan las señales de control del grupo 1 y que la diferencia entre ambos conjuntos de señales es el retraso de un ciclo de reloj que presentan las señales del grupo 2

con respecto al grupo 1. Por tanto, las señales de control del grupo 2 se pueden obtener introduciendo un retardo de un ciclo de reloj en las señales de control del grupo 1. De igual forma, las señales de control de los grupos 3 y 4 se obtienen introduciendo un retardo de un ciclo de reloj en las señales de control del grupo al que preceden. Dado el comportamiento que presentan las señales de control, se simplifica la implementación de los bloques de control que generan estas señales. Por último, las señales de control para los grupos del 2 al 4 deben encontrarse deshabilitadas en el modo de operación de 2.5 Gbps.

5.7 Buffer de datos

Este bloque introduce un retardo en el trayecto de los datos para permitir la inserción de los bytes P1 calculados. En las secciones 5.6.7.1 y 5.6.7.2 se ha encontrado que el contador de ciclos del bloque *Insertor de bytes P1* se debe encontrar retrasado 273 ciclos con respecto al contador del bloque *Calculador de bytes P1*, esto para cuando se procesan tramas de orden STS-48 / STM-16. Y para cuando se procesan tramas de orden STS-192 / STM-64 el retraso debe ser de 1083 ciclos de reloj.

Se puede observar en el diagrama de arquitectura del módulo codificador, presentado en la Figura 5.2, que previo al bloque *Calculador de bytes P1* se encuentra el bloque *Insertor de Ceros*, y que el flujo de datos entrante al bloque *Buffer de datos* es el mismo que entra al bloque *Insertor de Ceros*. El contador de ciclos del bloque *Calculador de bytes P1* se encuentra retrasado un ciclo con respecto al contador de ciclos del bloque *Insertor de Ceros*, por tanto, el desfase entre los contadores de ciclos de los bloques *Insertor de bytes P1* e *Insertor de Ceros* debe ser de 274 ciclos y de 1084 ciclos de reloj para cuando se procesan tramas de orden STS-48 / STM-16 y STS-192 / STM-64, respectivamente.

El bloque *Buffer de datos* debe introducir un retardo de 274 ciclos de reloj en el flujo de datos entrante cuando se procesan tramas de orden STS-48 / STM-16, y un retardo de 1084 ciclos para tramas de orden STS-192 / STM-64. En la Figura 5.29 se presenta el diagrama de señales de entrada y de salida del bloque, y en la Tabla 5.18 se presenta la descripción de estas señales.

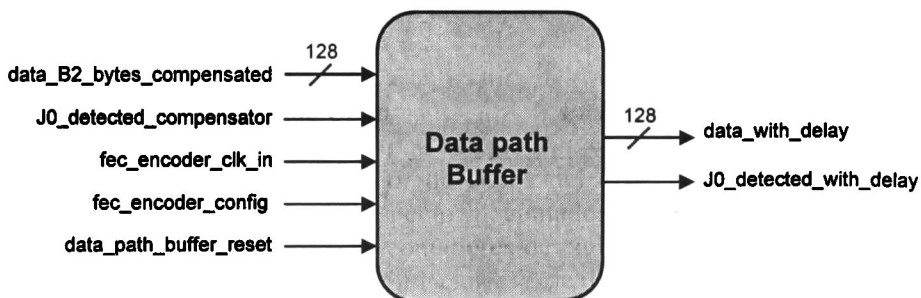


Figura 5.29 Diagrama de señales de entrada y de salida del bloque *Buffer de datos*

Entrada al bloque	Descripción	Origen
data_B2_bytes_compensated [127:0]	Bus de datos de entrada	Compensador de bytes B2
J0_detected_compensator	Señal de sincronía	Compensador de bytes B2

fec_encoder_clk_in	Señal de reloj del sistema	Exterior
fec_encoder_config	Señal de configuración	Control
data_path_buffer_reset	Señal de reset activa en valor alto	Control
Salida del bloque	Descripción	Destino
data_with_delay [127:0]	Bus de datos de salida	Insertor de bytes P1
J0_detected_with_delay	Señal de sincronía	Insertor de bytes P1

Tabla 5.18 Descripción de las señales de entrada y de salida del bloque *Buffer de datos*

En la Figura 5.30 se presenta el diagrama de microarquitectura del bloque *Buffer de datos*.

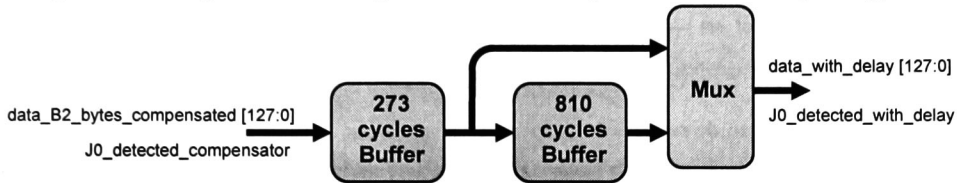


Figura 5.30 Microarquitectura del bloque *Buffer de datos*

El bloque *Buffer de 273 ciclos*, como su nombre lo indica, introduce un retardo de 273 ciclos de reloj en el trayecto de los datos. De igual forma, el bloque *Buffer de 810 ciclos* introduce un retardo de 810 ciclos de reloj.

En el modo de operación de 2.5 Gbps sólo operan el *Buffer de 273 ciclos* y el bloque *Multiplexor*. El bloque *Multiplexor* selecciona el flujo de datos a la salida del *Buffer de 273 ciclos*. Ambos bloques introducen un retardo total de 274 ciclos de reloj en el trayecto de los datos.

En el modo de operación de 10 Gbps, los tres bloques entran en funcionamiento. El flujo de datos a la salida del *Buffer de 273 ciclos* es el flujo de datos entrante al *Buffer de 810 ciclos*, el cual solamente opera en el modo de 10 Gbps. El bloque multiplexor selecciona el flujo de datos a la salida del *Buffer de 810 ciclos*. Los tres bloques introducen un retardo total de 1084 ciclos de reloj en el trayecto de los datos.

Así, se satisfacen los requerimientos de retardo en el trayecto de los datos para permitir la inserción de bytes P1 en ambos modos de operación del módulo codificador.

5.8 Insertor de bytes P1

El bloque *Insertor de bytes P1* realiza la inserción de los bytes P1 en el flujo de datos y contribuye a generar la señal de indicación de alarma MSF-AIS (*Multiplex Section FEC Alarm Indication Signal*). En la Figura 5.31 se presenta el diagrama de señales de entrada y de salida del bloque, y en la Tabla 5.19 se presenta la descripción de estas señales.

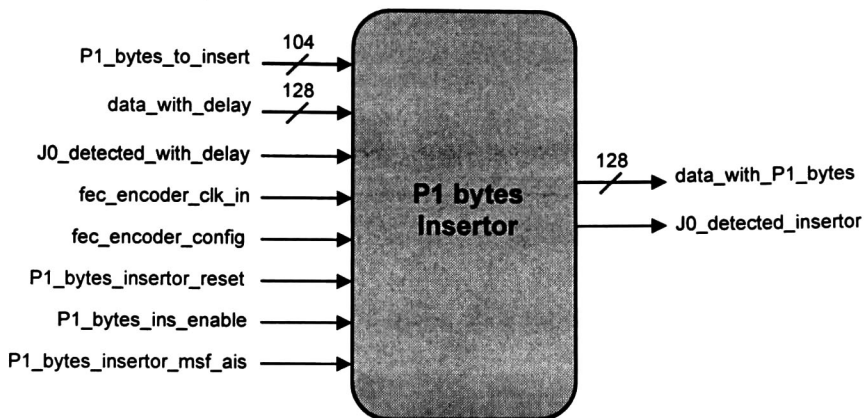


Figura 5.31 Diagrama de señales de entrada y de salida del bloque *Insertor de bytes P1*

Entrada al bloque	Descripción	Origen
P1_bytes_to_insert [103:0]	Bus de datos de entrada de bytes P1	Calculador de bytes P1
data_with_delay [127:0]	Bus de datos de entrada	Buffer de datos
J0_detected_with_delay	Señal de sincronía	Buffer de datos
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
fec_encoder_config	Señal de configuración	Control
P1_bytes_insertor_reset	Señal de reset activa en valor alto	Control
P1_bytes_ins_enable	Señal de habilitación de inserción de bytes P1	Control
P1_bytes_insertor_msf_ais	Señal de control para la generación de MSF-AIS	Control
Salida del bloque	Descripción	Destino
data_with_P1_bytes [127:0]	Bus de datos de salida	Calculador de bytes B2
J0_detected_insertor	Señal de sincronía	Calculador de bytes B2

Tabla 5.19 Descripción de las señales de entrada y de salida del bloque *Insertor de bytes P1*

Los ciclos de inserción de bytes P1 ya se han descrito en la Tabla 5.16. En cada uno de estos ciclos se deben insertar 13 bytes P1 en el flujo de datos entrante *data_with_delay*, desde el byte 4 al byte 16. En estos ciclos, los bytes del 1 al 3 pasarán sin ser modificados por el bloque insertor.

La señal de control *P1_bytes_ins_enable*, es una señal activa en valor alto que le indica al bloque que habilite la inserción de los bytes P1 en el flujo de datos.

La señal de control *P1_bytes_insertor_msf_ais*, es una señal activa en valor alto que le indica al bloque que habilite la generación de la señal de indicación de alarma MSF-AIS. Para contribuir a la generación de la señal MSF-AIS, el bloque *Insertor de bytes P1* debe insertar bytes “unos” en las posiciones de los bytes P1 que se ubican en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH).

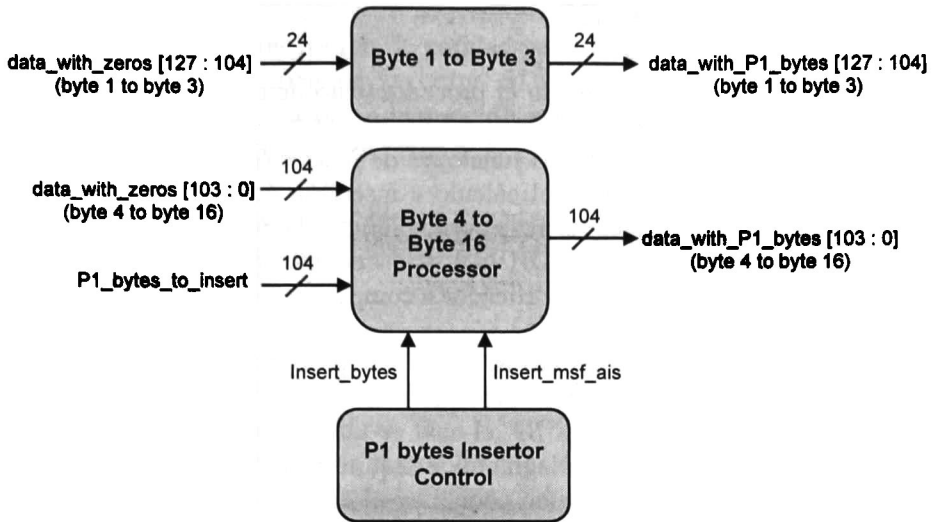


Figura 5.32 Microarquitectura del bloque *Insertor de bytes P1*

En la Figura 5.32 se presenta el diagrama de microarquitectura del bloque *Insertor de bytes P1*. El bloque *Control del Insertor de bytes P1* implementa una máquina de estados que genera las señales *Insert_bytes* e *Insert_msf_ais*. Estas dos señales controlan la operación del bloque *Procesador del byte 4 al byte 16*. Un valor “1” en la señal *Insert_bytes* le indica al bloque procesador que inserte los bytes P1 presentes en el bus *P1_bytes_to_insert*. Un valor “1” en la señal *Insert_msf_ais* le indica al bloque procesador que inserte bytes con valor 0xFF en el flujo de datos para así generar la señal MSF-AIS.

En la Figura 5.33 se muestra la máquina de estados del tipo Moore que genera estas señales de control.

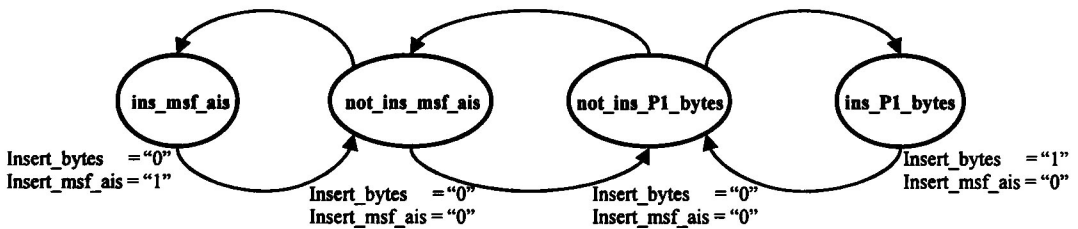


Figura 5.33 Máquina de estados para generar las señales de control *Insert_bytes* e *Insert_msf_ais*

Las transiciones entre estados dependen del valor presente en el contador de ciclos y del valor de las señales *fec_encoder_config*, *P1_bytes_ins_enable* y *P1_bytes_insertor_msf_ais*.

5.9 Calculador de bytes B2

En la sección 2.3.7 se ha fundamentado el procedimiento de compensación del código de paridad B2 (bytes B2). El módulo codificador FEC en banda recibe tramas SONET / SDH sobre las cuales ya se han efectuado las funciones de la sección de multiplexación MSOH (SDH), entre las cuales se encuentra el cálculo e inserción del código de paridad B2. El módulo codificador FEC en banda inserta información de chequeo de paridad (bytes P1) en posiciones de las tramas SONET / SDH que se encuentran protegidas por el código de paridad B2, lo que obliga al módulo codificador a compensar dicho código.

La compensación se efectúa en dos pasos. El primer paso consiste en realizar un nuevo cálculo del código de paridad B2 para proteger a los bytes P1 insertados. Esta tarea la realiza el bloque *Calculador de bytes B2*, el cual se ubica después del bloque *Insertor de bytes P1*, tal como se observa en el diagrama de arquitectura del módulo codificador en la Figura 5.2. El segundo paso consiste en actualizar el código de paridad B2 con el nuevo valor calculado. Esta tarea la ejecuta el bloque *Compensador de bytes B2* al insertar el resultado del cálculo obtenido por el bloque *Calculador de bytes B2*.

En la Figura 5.34 se presenta el diagrama de señales de entrada y de salida del bloque *Calculador de bytes B2*, y en la Tabla 5.20 se presenta la descripción de estas señales.

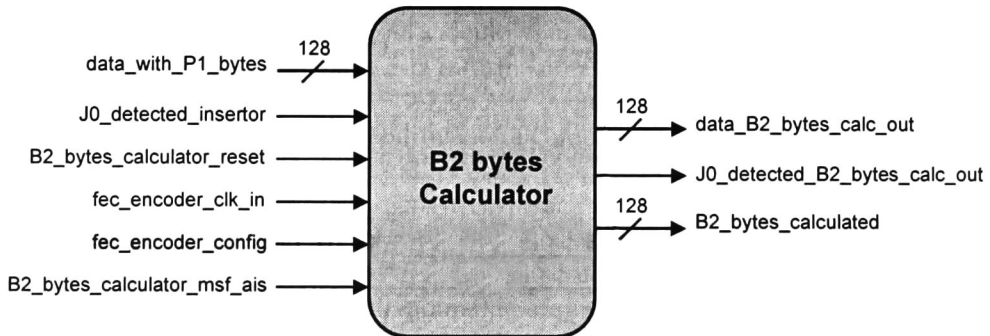


Figura 5.34 Diagrama de señales de entrada y de salida del bloque *Calculador de bytes B2*

Entrada al bloque	Descripción	Origen
data with P1 bytes [127:0]	Bus de datos de entrada	Insertor de bytes P1
J0 detected insertor	Señal de sincronía	Insertor de bytes P1
B2_bytes_calculator_reset	Señal de reset activa en valor alto	Control
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
fec_encoder_config	Señal de configuración	Control
B2_bytes_calculator_msf_ais	Señal de indicación de generación de MSF-AIS	Control
Salida del bloque	Descripción	Destino
data B2 bytes calc out [127:0]	Bus de datos de salida	Multiplexor de salida
J0 detected B2 bytes calc out	Señal de sincronía	Multiplexor de salida
B2_bytes_calculated [127:0]	Bus de datos de salida de bytes B2	Compensador de bytes B2

Tabla 5.20 Descripción de las señales de entrada y de salida del bloque *Calculador de bytes B2*

El código de paridad B2, es un código de paridad par con entrelazado de $N \times 24$ bits (BIP- $N \times 24$), en donde N es el orden de la trama SDH. En tramas de orden STM-16, el entrelazado del código es de 384 bits o bien de 48 bytes y en tramas de orden STM-64, el entrelazado es de 1536 bits o 192 bytes.

El ancho del bus de datos entrante al bloque calculador es de 16 bytes. El entrelazado del código, desde el punto de vista de ciclos de reloj, es de 3 ciclos para tramas STM-16 y de 12 ciclos para tramas STM-64.

5.9.1 Máquina de estados para el cálculo del código de paridad B2

La máquina de estados mostrada en la Figura 5.35 le permite al bloque *Calculador de bytes B2* realizar el cálculo del código de paridad. Esta máquina tiene un total de 12 estados y el estado inicial es *Set_1*. Por cada estado se tiene un registro de datos de 16 bytes, en el cual se almacena el cálculo parcial del código de paridad B2. Para tramas STM-16 la máquina recorre sólo los primeros tres estados ya que el entrelazado del código es de 48 bytes o de 3 ciclos de reloj. Para tramas STM-64 la máquina recorre el total de los estados ya que el entrelazado es de 192 bytes o de 12 ciclos de reloj. El cambio de estado se produce con cada ciclo de reloj siempre y cuando la información presente en el flujo de datos entrante deba incluirse en el cálculo del código de paridad.

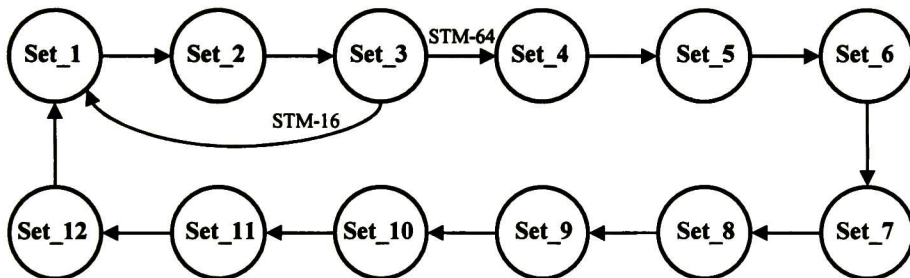


Figura 5.35 Máquina de estados para el cálculo del código de paridad B2

Al finalizar el procesamiento de una trama, el resultado del cálculo, que se encuentra en los registros de los estados, es almacenado en registros adicionales desde donde será leído y enviado hacia el bloque *Compensador de bytes B2* a través del puerto de salida *B2_bytes_calculated*. El contenido de los registros de los estados es inicializado a un valor de “ceros” al término de cada trama.

5.9.2 Máquina de estados para el envío del código de paridad B2 calculado

La máquina de estados mostrada en la Figura 5.36 le permite al bloque *Calculador de bytes B2* enviar el valor calculado del código de paridad B2 a través del puerto de salida *B2_bytes_calculated*.

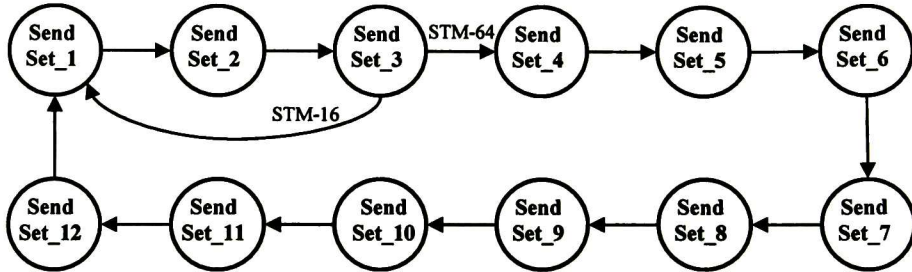


Figura 5.36 Máquina de estados para el envío del código de paridad B2

El diseño de la máquina de estados toma en cuenta la ubicación de los bytes B2 en el flujo de datos. En tramas de orden STS-48 / STM-16, los bytes B2 se ubican desde el ciclo 1080 al ciclo 1082 (48 bytes) y, en tramas de orden STS-192 / STM-64, se ubican desde el ciclo 4320 al ciclo 4331 (192 bytes).

El diseño del módulo codificador hace que exista un desfase entre el contador de ciclos del bloque *Calculador de bytes B2* y el contador de ciclos del bloque *Compensador de bytes B2*. En el modo de operación de 2.5 Gbps, el contador de ciclos del *Calculador de bytes B2* se encuentra retrasado 277 ciclos de reloj con respecto al contador del *Compensador de bytes B2* y, en el modo de operación de 10 Gbps, el retraso es de 1087 ciclos de reloj.

El desfase entre los contadores se toma en cuenta en el diseño de la máquina de estados para que los ciclos de envío de los bytes B2 coincidan con los ciclos de inserción en el bloque *Compensador de bytes B2*.

En el modo de operación de 2.5 Gbps, los bytes B2 son enviados del ciclo 803 al ciclo 805 y para ello la máquina de estados recorre sólo los primeros tres estados. En el modo de operación de 10 Gbps, los bytes B2 son enviados del ciclo 3233 al ciclo 3244 y la máquina de estados recorre el total de ellos.

5.9.3 Señal de indicación de alarma MSF-AIS

La señal de entrada *B2_bytes_calculator_msf_ais*, la cual es activa en valor alto, notifica al bloque *Calculador de bytes B2* de la generación de la señal de indicación de alarma MSF-AIS. Cuando se encuentra activa la generación de esta señal de alarma, el cálculo del código de paridad B2 y el envío de los bytes B2 deberán deshabilitarse. Estos procesos volverán a habilitarse hasta que se desactive la generación de esta señal.

5.10 Compensador de bytes B2

La función de este bloque consiste en actualizar el valor del código de paridad B2, el cual se ubica en los bytes B2 de las tramas SONET / SDH. El nuevo valor del código es calculado por el bloque *Calculador de bytes B2*.

Dado que las posiciones en donde se ubican los bytes B2 se encuentran protegidas por el proceso de codificación FEC en banda, el bloque *Compensador de bytes B2* debe actualizar estos bytes justo antes de que las tramas sean codificadas (cálculo e inserción de los bytes P1). Por esta razón, el bloque *Compensador de bytes B2* se localiza antes del bloque *Insertor de ceros* y del bloque *Buffer de datos*, como se observa en la Figura 5.2, en donde se presenta la arquitectura del módulo codificador FEC en banda.

En la Figura 5.37 se presenta el diagrama de señales de entrada y de salida del bloque *Compensador de bytes B2*, y en la Tabla 5.21 se presenta la descripción de estas señales.

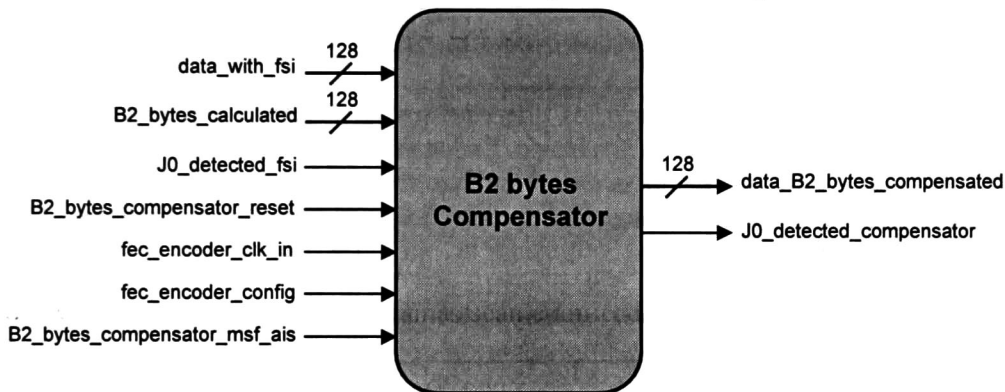


Figura 5.37 Diagrama de señales de entrada y de salida del bloque *Compensador de bytes B2*

Entrada al bloque	Descripción	Origen
data_with_fsi [127:0]	Bus de datos de entrada	Insertor de la señal FSI
B2_bytes_calculated [127:0]	Bus de datos de entrada de bytes B2	Calculador de bytes B2
J0_detected_fsi	Señal de sincronía	Insertor de la señal FSI
B2_bytes_compensator_reset	Señal de reset activa en valor alto	Control
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
fec_encoder_config	Señal de configuración	Control
B2_bytes_compensator_msf_ais	Señal de indicación de generación de MSF-AIS	Control
Salida del bloque	Descripción	Destino
data_B2_bytes_compensated [127:0]	Bus de datos de salida	<ul style="list-style-type: none"> • Insertor de Ceros • Buffer de datos
J0_detected_compensator	Señal de sincronía	<ul style="list-style-type: none"> • Insertor de Ceros • Buffer de datos

Tabla 5.21 Descripción de las señales de entrada y de salida del bloque *Compensador de bytes B2*

A través del puerto *B2_bytes_calculated* son recibidos los bytes B2 que serán insertados por el bloque. Los ciclos de inserción de los bytes B2 van del ciclo 1080 al 1082 para el modo de operación de 2.5 Gbps y del ciclo 4320 al 4331 para el modo de operación de 10 Gbps.

5.10.1 Señal de indicación de alarma MSF-AIS

Por medio de la señal de entrada *B2_bytes_compensator_msf_ais*, la cual es activa en valor alto, se le notifica al bloque *Compensador de bytes B2* la generación de la señal de indicación de alarma MSF-AIS. Cuando esto ocurre, el bloque compensador deshabilita temporalmente la inserción de los bytes B2. Una vez que se desactive la generación de la señal de indicación de alarma, el bloque compensador vuelve a habilitar la inserción de los bytes B2.

5.11 Multiplexor de datos de salida

En la sección 2.3.6 se han presentado los diferentes estados operacionales en que puede funcionar el módulo codificador FEC en banda. En los estados a) *FEC habilitado* y b) *FEC deshabilitado con retardo de codificación*, el módulo codificador introduce un retardo de codificación en el trayecto de los datos. Por el contrario, en el estado c) *FEC deshabilitado sin retardo de codificación* el módulo codificador no introduce el retardo de codificación en el trayecto de los datos.

La función del bloque *Multiplexor de datos de salida* es la de seleccionar el flujo de datos que se proporcionará como salida del módulo codificador. En la Figura 5.2 se observa la ubicación del bloque multiplexor dentro de la arquitectura del módulo codificador.

El flujo de datos a la salida del bloque *Insertor de la señal FSI* es uno de los dos flujos de datos que entran al bloque multiplexor. Este flujo es el que selecciona el bloque multiplexor para proporcionarlo a su salida cuando el módulo codificador opera en el estado c) *FEC deshabilitado sin retardo de codificación*. En este caso, el retardo total que introduce el módulo codificador sobre el flujo de datos es de sólo dos ciclos de reloj, lo que en unidades de tiempo equivale a un retardo de 102.88 ns en el modo de operación de 2.5 Gbps y a un retardo de 25.72 ns en el modo de operación de 10 Gbps.

El flujo de datos a la salida del bloque *Calculador de bytes B2* es el otro flujo de datos que entra al bloque multiplexor. Este flujo es el que selecciona el bloque multiplexor para proporcionarlo a su salida cuando el módulo codificador opera en el estado a) *FEC habilitado* o en el estado b) *FEC deshabilitado con retardo de codificación*. En estos casos, el retardo total que introduce el módulo codificador sobre el flujo de datos, cuando se encuentra configurado en el modo de 2.5 Gbps, es de 279 ciclos de reloj, lo que equivale a 14.35 μ s. Y cuando el módulo codificador se configura en el modo de 10 Gbps, el retardo total introducido sobre el flujo de datos es de 1089 ciclos de reloj, lo que equivale a 14.004 μ s.

Los retardos que se han descrito en los párrafos anteriores cumplen con el requerimiento R-FEC-ENC-04, en donde se especifica que el retardo máximo que puede introducir el módulo codificador FEC en banda sobre el trayecto de los datos es de 15 μ s.

En la Figura 5.38 se presenta el diagrama de señales de entrada y de salida del bloque multiplexor, y en la Tabla 5.22 se presenta la descripción de estas señales.

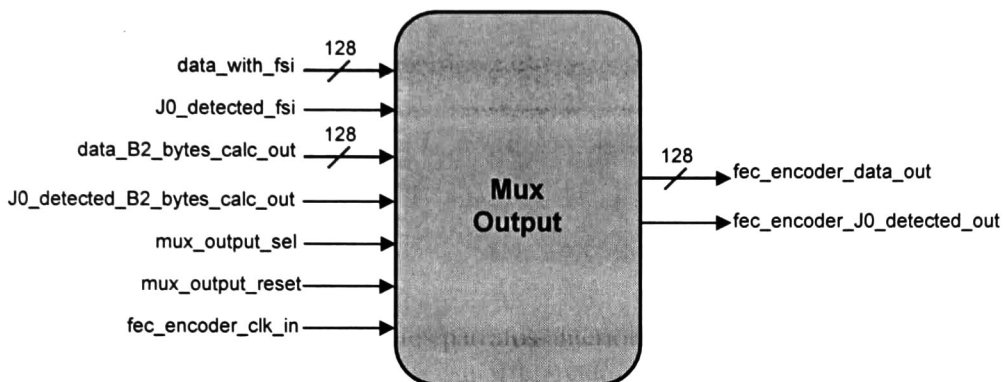


Figura 5.38 Diagrama de señales de entrada y de salida del bloque *Multiplexor de datos de salida*

Entrada al bloque	Descripción	Origen
data_with_fsi [127:0]	Bus de datos de entrada	Insertor de la señal FSI
J0 detected_fsi	Señal de sincronía	Insertor de la señal FSI
data_B2_bytes_calc_out [127:0]	Bus de datos de entrada	Calculador de bytes B2
J0 detected B2_bytes_calc out	Señal de sincronía	Calculador de bytes B2
mux_output_sel	Señal de selección del flujo de datos de salida	Control
mux_output_reset	Señal de reset activa en valor alto	Control
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
Salida del bloque	Descripción	Destino
fec_encoder_data_out [127:0]	Bus de datos de salida	Exterior
fec_encoder_J0_detected_out	Señal de sincronía	Exterior

Tabla 5.22 Descripción de las señales de entrada y de salida del bloque *Multiplexor de datos de salida*

La señal *mux_output_sel* es la entrada de selección del bloque multiplexor. Cuando esta señal toma un valor bajo (“0”) se seleccionan los datos que no contienen retardo de codificación, esto es, el flujo de datos *data_with_fsi* y la señal de sincronía *J0_detected_fsi* los cuales provienen del bloque *Insertor de la señal FSI*. Cuando la entrada de selección toma un valor alto (“1”) se seleccionan los datos que contienen retardo de codificación, esto es, el flujo de datos *data_B2_bytes_calc_out* y la señal de sincronía *J0_detected_B2_bytes_calc_out* los cuales provienen del bloque *Calculador de bytes B2*.

5.12 Bloque de control del módulo codificador

Este bloque controla al resto de los bloques que integran el módulo codificador FEC en banda. En la Figura 5.39 se presenta el diagrama de señales de entrada y de salida del bloque de control.

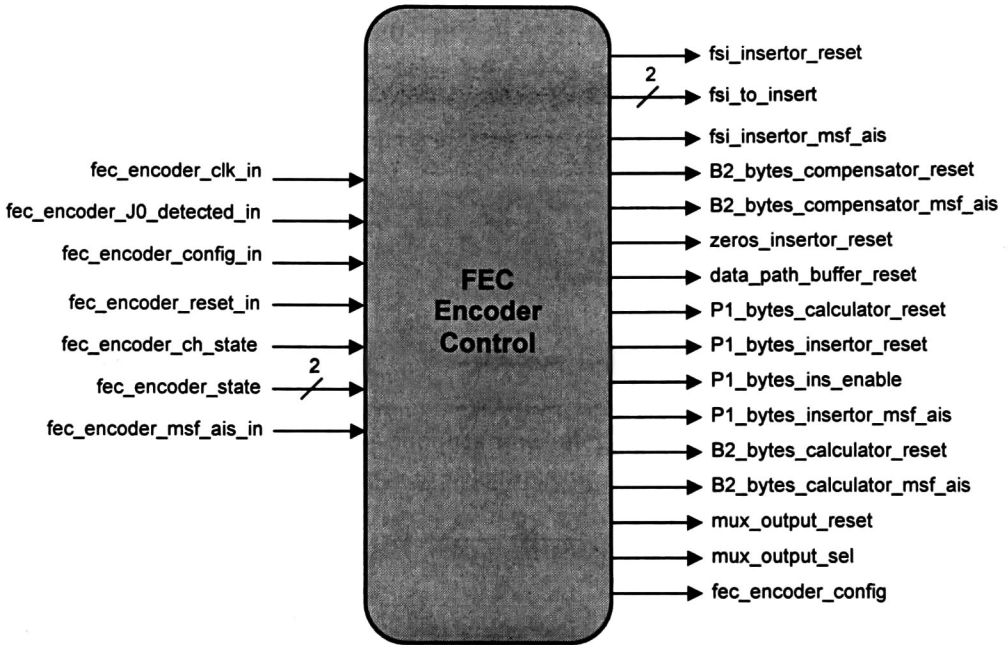


Figura 5.39 Diagrama de señales de entrada y de salida del bloque de control del módulo codificador

Entrada al bloque	Descripción	Origen
fec_encoder_clk_in	Señal de reloj del sistema	Exterior
fec_encoder_J0_detected_in	Señal de sincronía	Exterior
fec_encoder_config_in	Señal de configuración	Exterior
fec_encoder_reset_in	Señal de reset activa en valor alto	Exterior
fec_encoder_ch_state	Señal de control para solicitar un cambio en el estado operacional	Exterior
fec_encoder_state [1:0]	Señal de indicación del estado operacional siguiente	Exterior
fec_encoder_msf_ais_in	Señal de control para la generación de MSF-AIS	Exterior
Salida del bloque	Descripción	Destino
fsi_inserter_reset	Señal de reset activa en valor alto	Insertor de la señal FSI
fsi_to_insert [1:0]	Señal FSI a insertar	Insertor de la señal FSI
fsi_inserter_msf_ais	Señal de control para la generación de MSF-AIS	Insertor de la señal FSI
B2_bytes_compensator_reset	Señal de reset activa en valor alto	Compensador de bytes B2
B2_bytes_compensator_msf_ais	Señal de indicación de generación de MSF-AIS	Compensador de bytes B2
zeros_inserter_reset	Señal de reset activa en valor alto	Insertor de ceros
data_path_buffer_reset	Señal de reset activa en valor alto	Buffer de datos
P1_bytes_calculator_reset	Señal de reset activa en valor alto	Calculador de bytes P1

P1_bytes_insertor_reset	Señal de reset activa en valor alto	Insertor de bytes P1
P1_bytes_ins_enable	Señal de habilitación de inserción de bytes P1	Insertor de bytes P1
P1_bytes_insertor_msf_ais	Señal de control para la generación de MSF-AIS	Insertor de bytes P1
B2_bytes_calculator_reset	Señal de reset activa en valor alto	Calculador de bytes B2
B2_bytes_calculator_msf_ais	Señal de indicación de generación de MSF-AIS	Calculador de bytes B2
mux_output_reset	Señal de reset activa en valor alto	Multiplexor de datos de salida
mux_output_sel	Señal de selección del flujo de datos de salida	Multiplexor de datos de salida
fec_encoder_config	Señal de configuración programada	Hacia el resto de los bloques del módulo codificador

Tabla 5.23 Descripción de las señales de entrada y de salida del bloque de control del módulo codificador

5.12.1 Señales de entrada que controlan al módulo codificador

La señal de reset del módulo codificador es *fec_encoder_reset_in*, la cual es activa en valor alto (“1”).

La capa de administración de la red óptica gestiona cambios en el estado operacional del módulo codificador por medio de las señales de entrada *fec_encoder_ch_state* y *fec_encoder_state*.

La generación de la señal de indicación de alarma MSF-AIS es gestionada por medio de la señal de entrada *fec_encoder_msf_ais_in*.

La señal de entrada *fec_encoder_config_in* es la señal de configuración del módulo codificador. Cuando esta señal toma el valor “0” configura al módulo codificador para operar en el modo de 2.5 Gbps y cuando toma el valor “1” lo configura en el modo de 10 Gbps.

5.12.2 Máquina de control

Una máquina de control con estados y subestados genera las señales de reset, de habilitación, de activación de alarma, de configuración y de selección, que permiten a los bloques operar de manera correcta y sincronizada en cada estado operacional en que se configura al módulo codificador.

En la Figura 5.40 se muestran los estados posibles de la máquina de control. El estado inicial es el de *Reset*, y es el que toma la máquina de control cuando la señal de reset *fec_encoder_reset_in* se encuentra activa.

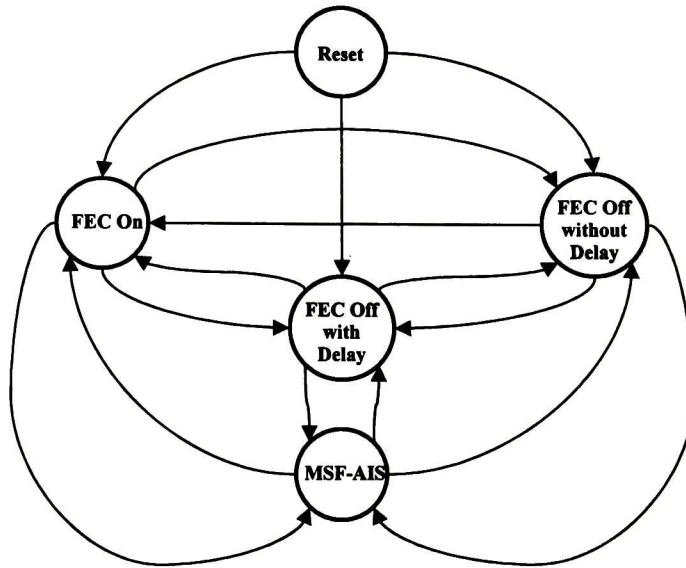


Figura 5.40 Estados de la máquina de control

Una vez que se ha desactivado la señal de reset, la máquina toma alguno de los siguientes estados: *FEC On*, *FEC Off with Delay* o *FEC Off without Delay*, dependiendo del valor de la señales de entrada *fec_encoder_ch_state* y *fec_encoder_state*.

El estado *MSF-AIS* es el que toma la máquina cuando por medio de la señal *fec_encoder_msf_ais_in* se notifica la generación de la señal de indicación de alarma MSF-AIS. Una vez que se da por terminada esta señal de indicación de alarma, la máquina de control regresa al estado en el que se encontraba previamente.

Las transiciones entre los estados *FEC On*, *FEC Off with Delay* y *FEC Off without Delay* se gestionan por medio de las señales de entrada *fec_encoder_ch_state* y *fec_encoder_state*.

Cada estado de la máquina de control tiene asociado un cierto número de subestados. Las transiciones entre estados implican también transiciones entre subestados. Los subestados realizan tareas específicas para el control de los bloques que componen al módulo codificador. En el Anexo B se presenta detalladamente el diseño de esta máquina de estados y subestados.

6 VERIFICACIÓN FUNCIONAL

6.1 Introducción

En este capítulo se presenta el plan de verificación para el módulo codificador FEC en banda para elementos de red SONET/SDH. Un plan de verificación realiza pruebas sobre el dispositivo diseñado a fin de comprobar que sea completamente funcional. Este plan se obtiene a partir de la especificación de requerimientos de diseño del dispositivo.

6.2 Verificación funcional

La verificación funcional es el proceso que comprueba el correcto funcionamiento de un dispositivo, mediante la inserción de estímulos de entrada y el análisis de los datos salientes. Dicho análisis es una evaluación de las diversas características que presenta el dispositivo y que se obtienen de la especificación de requerimientos de diseño.

La verificación funcional de un dispositivo se realiza basándose en un plan de verificación, el cual es un documento que define lo siguiente:

- *Pruebas aplicadas al diseño:* A cada característica del dispositivo, ya sea un comportamiento o una función, se debe realizar una prueba a fin de comprobar que el diseño funcione correctamente ante ciertos estímulos de entrada.
- *Camas de prueba para el diseño bajo prueba:* Se define el lenguaje en que se realizará la verificación y se escriben las camas de prueba que generarán los estímulos adecuados para la verificación.
- *Validación del diseño bajo prueba:* Se define la forma en que se realizará la validación del diseño bajo prueba.

En la Figura 6.1 se presenta la estructura de la verificación funcional que se utilizará en los casos de prueba del plan de verificación.

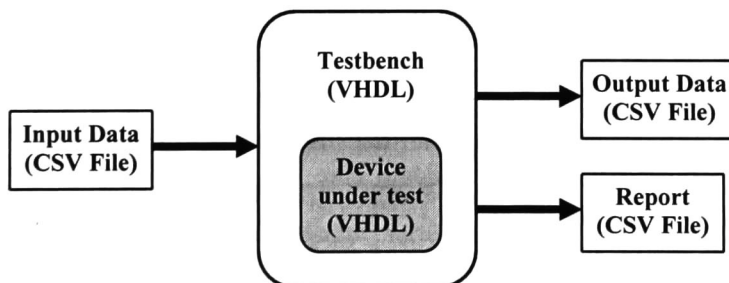


Figura 6.1 Estructura de la verificación funcional

El dispositivo bajo prueba o DUT (*Device Under Test*) es verificado funcionalmente por la cama de pruebas (*Testbench*), la cual está escrita en el lenguaje VHDL. La cama de pruebas inyecta los flujos de datos SONET / SDH al dispositivo bajo prueba a partir de la información escrita en un archivo de datos de entrada, el cual es del tipo CSV (*Comma*

Separated Values). La cama de pruebas proporciona además los estímulos de reset, habilitación, configuración y la señal de reloj al dispositivo bajo prueba.

Los flujos de datos a la salida del dispositivo bajo prueba son leídos por la cama de pruebas y son escritos en un archivo de datos de salida, también del tipo CSV. La cama de pruebas realiza la verificación funcional sobre el dispositivo y las discrepancias o errores que detecta son escritos en un archivo de reporte también del tipo CSV. La característica o funcionalidad a verificar se obtiene de la especificación de requerimientos.

Archivos de datos de entrada: Los archivos de datos de entrada son archivos del tipo CSV (*Comma Separated Values*), los cuales contienen campos de texto separados por el carácter coma (,). Los archivos de datos de entrada contienen tramas SONET / SDH y los campos de texto representan bytes en formato hexadecimal (01, F6, 28, 0E, etc.). Estos archivos pueden contener una o más tramas SONET / SDH que representan los estímulos de entrada para ejecutar los casos y subcasos de pruebas del plan de verificación. Para su identificación, el nombre de los archivos de datos de entrada comienza con el prefijo “I_data_”. Estos archivos son generados por una herramienta de software desarrollada en el Cinvestav del IPN unidad Guadalajara [13].

Archivos de datos de salida: Los archivos de datos de salida son similares a los archivos de datos de entrada. Para su identificación, el nombre de los archivos de datos de salida comienza con el prefijo “O_data_”. Estos archivos son escritos por la cama de pruebas y refleja el flujo de información que proporciona en su salida de datos el dispositivo bajo prueba.

Archivos de reporte: Los archivos de reporte son archivos del tipo CSV y son generados por la cama de pruebas. En estos archivos se reflejan los errores detectados al verificar alguna característica o funcionalidad del dispositivo bajo prueba. Los campos de texto contienen el error detectado, la trama en que se detecta el error y el tiempo de simulación.

Simulación: La simulación es un proceso que se ejecuta en herramientas de software y que permite introducir estímulos de datos y estímulos de control, de configuración y de habilitación al dispositivo bajo prueba, así como la lectura de señales de respuesta y datos a la salida del dispositivo.

Se puede realizar la verificación funcional del dispositivo bajo prueba de manera visual, analizando sus respuestas y comportamiento ante los datos y estímulos de entrada. Este proceso resulta prácticamente imposible cuando el dispositivo a probar es muy complejo. Las camas de prueba que ejecutan los casos y subcasos de prueba del plan de verificación permiten que las simulaciones se realicen de manera automatizada y de esta forma sean rápidas y confiables.

6.3 Casos de prueba del módulo codificador FEC en banda

En esta sección se describen los casos prueba que contempla el plan de verificación del módulo codificador FEC en banda para elementos de red SONET/SDH. En el Anexo C se presentan a detalle todos los casos y subcasos de prueba del plan de verificación.

6.3.1 Identificador de los casos de prueba

En la Figura 6.2 se muestra el identificador que se utiliza para nombrar a los casos de prueba.

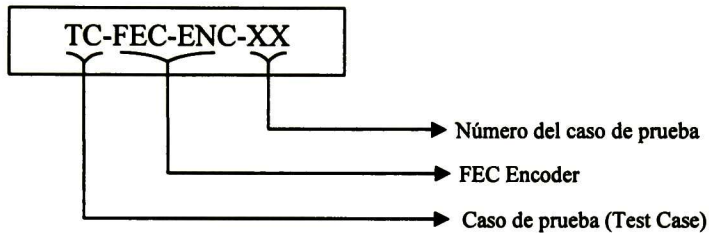


Figura 6.2 Identificador de los casos de prueba del módulo codificador FEC en banda

6.3.2 Descripción de los casos de prueba

A continuación se describen los diferentes casos de prueba contemplados en el plan de verificación del módulo codificador FEC en banda.

6.3.2.1 Caso de prueba TC-FEC-ENC-01

El presente caso de prueba verifica el funcionamiento del bloque *Insertor de la señal FSI*, esto es, que inserte correctamente la señal de dos bits FSI (*FEC Status Indicator*) en el primer byte Q1 cuando opera de manera normal, y que se inserte el valor 0xFF en todos los bytes Q1 cuando se habilita la generación de la señal de alarma MSF-AIS.

6.3.2.2 Caso de prueba TC-FEC-ENC-02

El presente caso de prueba verifica que el bloque *Insertor de ceros* realice la inserción de bytes con valor 0x00 en las posiciones de la trama SONET / SDH que no se consideran en el proceso de codificación FEC en banda.

6.3.2.3 Caso de prueba TC-FEC-ENC-03

El presente caso de prueba verifica el correcto funcionamiento del bloque *Calculador de bytes P1*.

Primero, se verificará que el flujo de datos SONET / SDH de 128 bits sea separado en $8 \times N/16$ flujos mediante un desentrelazado de $8 \times N/16$ bits, en donde N toma un valor igual a 16 cuando se opera en el modo de 2.5 Gbps (STS-48 / STM-16) y un valor igual a 64 cuando se opera en el modo de 10 Gbps (STS-192 / STM-64).

Segundo, se verificará que todos los bloques *calculadores de bits a_n* realicen correctamente el cálculo de dichos bits.

Tercero, se verificará que a partir de los bits a_n calculados para las subfilas de todos los grupos se formen correctamente los bytes P1.

Por último, se verificará que los bytes P1 sean enviados en el tiempo y en el orden adecuado hacia el puerto de salida del bloque *Calculador de bytes P1*, para que sean insertados por el bloque *Insertor de bytes P1*.

6.3.2.4 Caso de prueba TC-FEC-ENC-04

El presente caso de prueba verifica el correcto funcionamiento de los bloques *Insertor de bytes P1* y *Buffer de datos*. El presente caso de prueba tiene como objetivos:

- Verificar que el bloque *Insertor de bytes P1* realice la inserción de los bytes P1, que recibe del bloque *Calculador de bytes P1*, sobre el flujo entrante SONET / SDH en las posiciones de inserción que describe el requerimiento R-FEC-ENC-09. Lo anterior para cuando se encuentre habilitada la inserción de los bytes P1.
- Verificar que el bloque *Insertor de bytes P1* opere de manera *transparente* cuando se encuentre deshabilitada la inserción de los bytes P1, es decir, se debe verificar que el flujo saliente SONET / SDH sea igual al flujo entrante.
- Verificar que cuando se encuentre habilitada la generación de la señal de indicación de alarma MSF-AIS, el bloque *Insertor de bytes P1* inserte el valor 0xFF en todas las posiciones de inserción de bytes P1 en la tara de sección (SOH) / tara de sección de regeneración (RSOH) del flujo entrante SONET / SDH.
- Verificar que el bloque *Buffer de datos* introduzca el retardo en el flujo de datos SONET / SDH y en la señal *J0_detected*, en número de ciclos de reloj, para el cual fue diseñado.

6.3.2.5 Caso de prueba TC-FEC-ENC-05

El presente caso de prueba verifica el correcto funcionamiento de los bloques *Calculador de bytes B2* y *Compensador de bytes B2*. Tiene como objetivos:

- Verificar que el bloque *Calculador de bytes B2* realice correctamente el cálculo de los bytes B2 y los envíe en el tiempo y en el orden adecuado hacia el bloque *Compensador de bytes B2*. Lo anterior para cuando se encuentre deshabilitada la generación de la señal de indicación de alarma MSF-AIS.
- Verificar que el bloque *Compensador de bytes B2* inserte correctamente los bytes B2 que le envía el bloque *Calculador de bytes B2*. Lo anterior para cuando se encuentre deshabilitada la generación de la señal de indicación de alarma MSF-AIS.
- Verificar que cuando se encuentre habilitada la generación de la señal de indicación de alarma MSF-AIS, el bloque *Compensador de bytes B2* opere de forma transparente, es decir, que el flujo de datos de salida sea igual al flujo de datos de entrada.

6.3.2.6 Caso de prueba TC-FEC-ENC-06

El presente caso de prueba verifica el correcto funcionamiento del bloque *Multiplexor de Salida*.

6.3.2.7 Caso de prueba TC-FEC-ENC-07

El presente caso de prueba verifica el correcto funcionamiento del bloque *Control*. Tiene como objetivos:

- Verificar que la máquina de estados del bloque *Control* alcance todos los estados y subestados y realice las transiciones correctamente, en respuesta a las señales de entrada y señales internas presentes.
- Verificar que el bloque *Control* genere correctamente las señales de control para todos los bloques que integran el módulo codificador FEC en banda.

6.3.2.8 Caso de prueba TC-FEC-ENC-08

El presente caso de prueba realiza la verificación funcional a nivel *top* del módulo codificador FEC en banda. Tiene como objetivos:

- Verificar la correcta generación e inserción de la señal FSI y bytes Q1.
- Verificar el cálculo e inserción de los bytes P1.
- Verificar la compensación de los bytes de paridad B2.
- Verificar la generación de la señal de indicación de alarma MSF-AIS.
- Verificar la no violación en el máximo retardo permitido en el flujo de datos.
- Verificar que el módulo codificador actúe de manera transparente en los bytes restantes de las tramas SONET / SDH, los cuales no deben ser alterados por la función FEC en banda.

En la Tabla 6.1 se presenta una matriz de seguimiento de requerimientos de diseño contra casos de prueba

	TC-FEC-ENC-01	TC-FEC-ENC-02	TC-FEC-ENC-03	TC-FEC-ENC-04	TC-FEC-ENC-05	TC-FEC-ENC-06	TC-FEC-ENC-07	TC-FEC-ENC-08
R-FEC-ENC-01	●	●	●	●	●	●		●
R-FEC-ENC-02			●					●
R-FEC-ENC-03			●					●
R-FEC-ENC-04							●	●
R-FEC-ENC-05			●					●
R-FEC-ENC-06			●					●
R-FEC-ENC-07			●					●
R-FEC-ENC-08			●					●
R-FEC-ENC-09			●	●				●
R-FEC-ENC-10		●						●
R-FEC-ENC-11	●							●
R-FEC-ENC-12							●	●
R-FEC-ENC-13							●	●
R-FEC-ENC-14							●	●
R-FEC-ENC-15							●	●
R-FEC-ENC-16							●	●
R-FEC-ENC-17	●						●	●
R-FEC-ENC-18								●
R-FEC-ENC-19					●			●
R-FEC-ENC-20	●			●	●			●
R-FEC-ENC-21	●			●	●			●
R-FEC-ENC-22	●			●				●

Tabla 6.1 Matriz de seguimiento de requerimientos de diseño contra casos de prueba

6.3.3 Subcasos de prueba

El plan de verificación tiene en total 8 casos de prueba, cada uno de ellos está formado por uno o más subcasos de prueba, teniéndose en total 33 subcasos de prueba. En la Figura 6.3 se muestra el identificador para los subcasos de prueba.

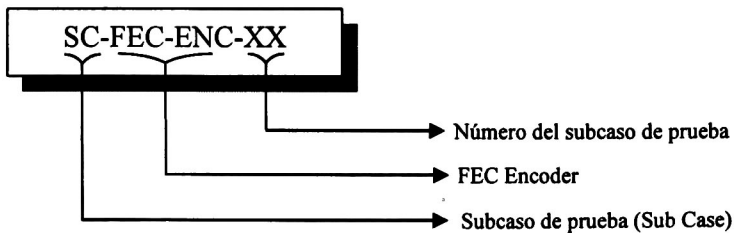


Figura 6.3 Identificador de los subcasos de prueba del módulo codificador FEC en banda

6.4 Síntesis

Un paso en el proceso de diseño digital es la síntesis de los dispositivos diseñados. Una vez que los dispositivos ya han sido probados funcionalmente debe comprobarse además, que estos sean completamente sintetizables, es decir, que la funcionalidad del dispositivo pueda ser implementada en alguna tecnología de circuitos integrados, por ejemplo en un FPGA o en un ASIC.

El proceso de síntesis lo realizan herramientas de software como **Synopsys**, en donde el código escrito en lenguaje de descripción de hardware (VHDL) es analizado y mapeado a componentes (compuertas, bloques lógicos, etc.) de alguna librería de tecnología específica (por ejemplo la librería *flex10k-3_fpga.db*). El mapeo realizado por el proceso de síntesis se refleja en un archivo *netlist* en donde se detallan los componentes y su interconexión. Las herramientas de síntesis optimizan el mapeo de tal forma que se utilice la menor área posible y que no se presenten violaciones en tiempo.

7 RESULTADOS

En este capítulo se presentan los resultados obtenidos durante la realización de este trabajo de tesis. Estos resultados incluyen las estadísticas de diseño de los bloques en lenguaje VHDL, las estadísticas de verificación que incluyen la implementación y ejecución de las camas de prueba, las estadísticas de cobertura de líneas de código de los bloques de diseño y, finalmente, los reportes de síntesis obtenidos en el software **Synopsys**.

7.1 Estadísticas de diseño de los bloques

En la Tabla 7.1 se presentan las siguientes estadísticas: número de componentes, número de entidades, número de procesos y número de líneas de código para cada uno de los bloques que integran el módulo codificador FEC en banda para elementos de red SONET/SDH.

Nombre del bloque	Número de componentes	Número de entidades	Número de procesos	Líneas de código
Insertor de la señal FSI	1	1	4	204
Insertor de ceros	1	1	5	399
Calculador de bytes P1	20	21	60	5434
Insertor de bytes P1	3	4	4	708
Buffer de datos	5	6	4	723
Calculador de bytes B2	1	1	8	500
Compensador de bytes B2	1	2	4	260
Multiplexor de datos de salida	2	3	2	474
Control	1	1	2	609

Tabla 7.1 Estadísticas de diseño de los bloques

El software **Visual HDL** se utilizó para la edición del código en VHDL. Este software es una herramienta gráfica que permite ahorrar esfuerzo de edición de código ya que automáticamente genera código para los bloques que realizan instancias de otros componentes. Por ejemplo, el bloque *Calculador de bytes P1* consta de 5434 líneas, 2434 de ellas son generadas automáticamente por el software.

7.2 Estadísticas de Verificación

En la Tabla 7.2 se presentan las siguientes estadísticas: número de subcasos de prueba, número de entidades y líneas de código de las camas de prueba y número de tramas simuladas para caso de prueba.

Identificador del Caso de prueba	Número de subcasos de prueba	Número de entidades	Líneas de código	Número de tramas simuladas
TC-FEC-ENC-01	2	2	631	60 tramas
TC-FEC-ENC-02	2	2	642	20 tramas
TC-FEC-ENC-03	15	14	12632	60 tramas
TC-FEC-ENC-04	4	4	2155	60 tramas
TC-FEC-ENC-05	4	4	1300	60 tramas
TC-FEC-ENC-06	2	3	672	60 tramas
TC-FEC-ENC-07	2	3	6056	380 tramas
TC-FEC-ENC-08	2	3	5890	380 tramas

Tabla 7.2 Estadísticas de verificación

Las simulaciones de los casos de prueba se realizaron en el software **ModelSim** en una computadora PC con Windows XP Home Edition, procesador Intel Pentium 4 a 2.0 GHz y 256 MB de memoria RAM.

Cada caso de prueba se ejecuta mediante una sola simulación en la cual se proporcionan los estímulos de entrada y señales de habilitación y configuración del bloque de forma tal que se estrese el dispositivo al máximo. Primero se verifica el bloque en el modo de operación de 2.5 Gbps y posteriormente a 10 Gbps. Los subcasos de prueba se implementan en entidades en VHDL, las cuales realizan la verificación funcional.

Las camas de prueba que implementan los casos de prueba TC-FEC-ENC-03, TC-FEC-ENC-07 y TC-FEC-ENC-08 constan de 12632, 6056 y 5890 líneas de código, respectivamente. Estas cantidades reflejan la complejidad de los bloques que son verificados. El total de líneas de código de las camas de prueba es de 29978 líneas, 1192 de ellas son generadas automáticamente por el software **Visual HDL**.

La herramienta de simulación **ModelSim** permite obtener las estadísticas de cobertura de código (*code coverage*) para los bloques de diseño que son simulados. En la Tabla 7.3 se presentan los porcentajes de líneas de código que fueron cubiertas durante la ejecución de los casos de prueba.

Nombre del bloque	Líneas de código	Porcentaje de cobertura de código
Insertor de la señal FSI	204	96.6 %
Insertor de ceros	399	97.7 %
Calculador de bytes P1	5434	97.7 %
Insertor de bytes P1	708	91.4 %
Buffer de datos	723	98 %
Calculador de bytes B2	500	96.4 %
Compensador de bytes B2	260	91.8 %
Multiplexor de datos de salida	474	97.1 %
Control	609	96.4 %

Tabla 7.3 Porcentajes de cobertura de código para los bloques de diseño

7.3 Reporte de síntesis

Los bloques fueron sintetizados en el software **Synopsys**, en una estación de trabajo **SUN Blade 100** con procesador UltraSPARC IIe de 500 MHz y 512 MB de memoria SDRAM.

La librería utilizada en la síntesis es “**tc6a_cbacore.db**”, la cual forma parte de las librerías que se instalan junto con el software **Synopsys**. Esta librería de tecnología corresponde a un ASIC y quien la provee es la compañía fundidora **TSMC**, con oficinas centrales en Taiwán.

En la Tabla 7.4 se presenta el reporte de síntesis de los bloques en donde se muestran el número de líneas de código, número de compuertas, máximo tiempo de arribo de datos y tiempo de violación. El periodo de la señal de reloj de los bloques se configura a 12.86 ns (frecuencia de reloj de 77.76 MHz) y corresponde al periodo de reloj para cuando los bloques operan en el modo de 10 Gbps.

Nombre del bloque	Líneas de código	Número de compuertas	Tiempo de arribo de datos (ns)	Tiempo de violación (ns)
Insertor de la señal FSI	204	1443	7.12	---
Insertor de ceros	399	1819	7.82	---

Calculador de bytes P1	5434	68235	12.19	---
Insertor de bytes P1	708	1704	7.75	---
Buffer de datos	723	*	*	*
Calculador de bytes B2	500	38143	7.19	---
Compensador de bytes B2	260	1780	7.18	---
Multiplexor de datos de salida	474	1417	0.64	---
Control	609	1343	8.06	---

Tabla 7.4 Reportes de síntesis de los bloques

Notas:

- No se presenta violación en tiempo
- * Durante la síntesis del bloque *Buffer de datos* el software Synopsys agotó la memoria de la estación de trabajo y no se concretó el proceso de optimización

Un solo *script* ejecuta la síntesis de todos los bloques diseñados. En la Figura 7.1 se muestra parte del texto del *script* “*synthesis_script*” en donde aparecen todos los comandos ejecutados para la síntesis de los bloques *Insertor de la señal FSI* e *Insertor de Ceros*.

```
#!/bin/csh -f
cd work_dir

#####
echo Synthesis of FSI Insertor Block
#####
cp ~/fec_encoder_project/code/fsi_insertor.vhd ./
cp ~/fec_encoder_project/synth/script/script_fsi_insertor ./running_script

echo Synthesis of FSI Insertor Block > ../synthesis_time.txt
date >> ../synthesis_time.txt

dc_shell -f running_script
date >> ../synthesis_time.txt
cp command.log ../log/command_fsi_insertor.log

#####
echo Synthesis of Zeros Insertor Block
#####
cp ~/fec_encoder_project/code/zeros_insertor.vhd ./
cp ~/fec_encoder_project/synth/script/script_zeros_insertor ./running_script

echo Synthesis of Zeros Insertor Block >> ../synthesis_time.txt
date >> ../synthesis_time.txt

dc_shell -f running_script
date >> ../synthesis_time.txt
cp command.log ../log/command_zeros_insertor.log

...
```

Figura 7.1 Parte de *script* “*synthesis_script*”

Para cada bloque se ha editado un script en donde se indican los comandos de síntesis que debe ejecutar la interfaz de línea de comandos (**dc shell**) del módulo **Design Compiler** de Synopsys. En el script “*synthesis_script*”, el llamado a la interfaz **dc shell** se realiza con el comando **dc_shell -f running_script**.

En la Figura 7.2 se muestra el contenido del script “*script_fsi_inserter*” editado para el bloque *Inserter de la señal FSI*.

```
/* synthesis script */

/* clear memory */
free -all

define_design_lib WORK -path ../work"

/* analyze files */
analyze -format vhdl -lib WORK fsi_inserter.vhd

/* elaborate designs */
elaborate fsi_inserter -arch fsi_inserter_arch -lib WORK -update

/* link libraries */
link

/* create clocks */
create_clock fec_encoder_clk_in -name "fec_encoder_clk_in" -period 12.86
set_dont_touch_network find(clock , "fec_encoder_clk_in" )
set_fix_hold find( clock, {"fec_encoder_clk_in"} )

/* check design */
check_design

/* compile design */
compile -map_effort high -verify -verify_effort high -boundary_optimization

/* save db and edif files */
write -f db -hierarchy -output ~/fec_encoder_project/synth/db/fsi_inserter_cbacore.db
write -f edif -hierarchy -output ~/fec_encoder_project/synth/edif/fsi_inserter_cbacore.edf

/* generate area report */
report_area > ~/fec_encoder_project/synth/report/area_fsi_inserter_cbacore.log

/* generate timing report */
report_timing -path full -delay max -max_paths 1 -nworst 1 >
~/fec_encoder_project/synth/report/timing_fsi_inserter_cbacore.log

/* end */
quit
```

Figura 7.2 Script con comandos de síntesis para el bloque *Inserter de la señal FSI*

El proceso de síntesis para cada bloque genera los siguientes resultados en archivos: reporte de área, reporte de tiempo, archivo de base de datos DB que contiene el esquemático del bloque sintetizado y archivo de base de datos EDIF (Electronic Database Interchange Format).

La información que contiene el archivo EDIF es conocido en el dominio del diseño electrónico como *netlist*, ya que se describe la interconexión de los diferentes componentes (compuertas, flip flops, latches, etc.) que se obtuvieron como resultado del proceso de síntesis.

8 CONCLUSIONES

En el presente trabajo de tesis se realizó la especificación, diseño, verificación y síntesis de un módulo que implementa la funcionalidad FEC en banda para elementos de red SONET/SDH a velocidades de 2.5 Gbps y 10 Gbps. Esta funcionalidad se describe en el anexo A de la recomendación ITU-T G.707 en la versión 5.0.

Como etapa inicial se realizó la especificación de requerimientos de diseño del módulo basándose en estándares internacionales y recomendaciones. En este documento se presentan las restricciones del diseño y los requerimientos funcionales del módulo. Un total de 22 requerimientos funcionales fueron obtenidos.

Un análisis de la especificación de requerimientos permitió identificar las tareas o funciones esenciales del módulo codificador FEC en banda. A partir de ellas se definieron los diferentes bloques en la arquitectura del módulo codificador FEC en banda.

El lenguaje de descripción de hardware VHDL se utilizó en el diseño del módulo codificador FEC en banda. El diseño de todos los bloques suma en total 9311 líneas de código. La edición fue realizada en la herramienta gráfica Visual HDL. Esta herramienta genera código de manera automática para entidades a nivel *top* que constan de varios sub-bloques. La tarea de diseño se centró en la escritura del código para estos sub-bloques y en definir y editar las señales de interconexión entre ellos.

Una vez concluido el diseño de los bloques se desarrolló un plan de verificación para comprobar el correcto funcionamiento del módulo codificador FEC en banda. Este plan cumple con la especificación de requerimientos de diseño.

Un total de 8 casos de prueba y 33 subcasos de prueba fueron definidos en el plan de verificación. Los casos de prueba del 1 al 7 verifican las funciones que ejecutan los bloques de diseño. El caso de prueba número 8 verifica el funcionamiento a nivel *top* del módulo codificador FEC en banda y por tanto es el caso de prueba de mayor complejidad.

Las camas de prueba que ejecutan los diferentes casos de prueba hacen uso de archivos con formato CSV con campos de texto separados por el caracter coma (,). Estos archivos contienen datos que representan tramas SONET / SDH y que son proporcionados como estímulos de entrada a los bloques verificados. Los datos a la salida de los bloques verificados son leídos y analizados por la cama de prueba y posteriormente son escritos en archivos con formato CSV. La edición de las camas de prueba se realizó en el software Visual HDL. Un total de 29978 líneas de código fueron obtenidas.

La simulación de los casos de prueba se realizó en el software ModelSim. Para cada caso de prueba se realizó un script que permite automatizar el proceso de simulación. Este software permitió obtener estadísticas de cobertura de líneas de código de los bloques diseñados. El porcentaje de cobertura de líneas código de los bloques está por arriba del 91%.

Por último se realizó la síntesis de los bloques de diseño en el software Synopsys en una estación de trabajo SUN Blade 100 con sistema operativo SUN Solaris 10.

Se realizaron scripts para automatizar el proceso de síntesis de cada uno de los bloques diseñados. La librería de tecnología utilizada en la síntesis es la librería de un ASIC, tc6a_cbacore.db.

Los resultados de síntesis incluyen archivos de reporte de área utilizada, reportes de tiempo, archivos con formato DB que contienen los esquemáticos de los diseños sintetizados y archivos con formato EDIF que contienen los *netlists*. Los bloques diseñados son completamente sintetizables y no presentan violaciones en tiempo.

Como resultado del presente trabajo de tesis, se ha obtenido una librería en lenguaje VHDL completamente sintetizable y funcional para el módulo codificador FEC en banda.

Dentro de los trabajos a futuro se plantea el uso de esta librería en el diseño de un elemento de red SONET/SDH con funcionalidad FEC en banda, y la implementación y validación de este diseño en un ASIC o en un FPGA.

El diseño del módulo complementario, el decodificador FEC en banda, forma parte de otro trabajo de tesis que actualmente se encuentra en proceso de terminación por un estudiante del área de telecomunicaciones de Cinvestav unidad Guadalajara.

Se plantea también la medición del desempeño real de FEC en banda en una red SONET/SDH y la medición del desempeño ante diferentes tipos de errores.

El desarrollo de este trabajo de tesis me ha permitido involucrarme y conocer el proceso de diseño digital. He visto la importancia del uso de herramientas de diseño a alto nivel y de síntesis lógica en la implementación de circuitos integrados. He aprendido a diseñar en el lenguaje de descripción de hardware VHDL, a utilizar herramientas de síntesis como Synopsys y metodologías de diseño y verificación funcional de dispositivos digitales.

Finalmente, este trabajo de tesis generó el artículo "*In band FEC encoder for SONET/SDH at 2.5 Gbit/s and 10 Gbit/s*" que fue presentado en la Novena Conferencia de Ingeniería Eléctrica (CIE), celebrada en septiembre de 2003 en la ciudad de México D.F..

REFERENCIAS

- [1] A. Hocquenghem, "Codes correcteurs d'erreurs", Chiffres, Vol. 2, pp. 147-156, 1959.
- [2] R.C. Bose and D.K. Ray-Chaudhuri, "On a class of error correcting binary group codes", Inf. Control, Vol. 3, pp. 68-79, March 1960.
- [3] W.W. Peterson, "Encoding and error correction procedures for the Bose-Chaudhuri codes" IEEE Trans. Inf. Theory, IT-6, pp. 459-470, September 1960.
- [4] E.R. Berlekamp, "On decoding binary Bose-Chaudhuri-Hocquenghem codes", IEEE Trans. Inf. Theory, IT-11, pp. 577-580, October 1965.
- [5] R.T. Chien, "Cyclic decoding procedure for the Bose-Chaudhuri-Hocquenghem codes", IEEE Trans. Inf. Theory, IT-10, pp. 357-363, October 1964.
- [6] S. Lin and D.J. Costello, Jr., Error control coding: fundamentals and applications, Prentice Hall, New Jersey, 1983.
- [7] W.W. Peterson, Error-Correcting Codes, MIT Press, pp 251-255, Cambridge, Mass., 1970.
- [8] International Telecommunication Union, ITU-T G.707 "Network Node Interface for the Digital Synchronous Hierarchy", October 2000.
- [9] International Telecommunication Union, ITU-T G.780 "Vocabulary of terms for synchronous digital hierarchy (SDH) networks and equipment", June 1999.
- [10] International Telecommunication Union, ITU-T G.783 "Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks" April 1997.
- [11] American National Standard for telecommunications, "Synchronous Optical Network (SONET) – Basic description including multiplex structure, rates and formats", Draft ANSI T1.105, 2001.
- [12] American National Standard for telecommunications, "Synchronous Optical Network (SONET) – In-band Forward Error Correction Code Specification", Draft ANSI T1.105.08, 2001.
- [13] E. Jiménez and D. Torres "SONET/SDH Tester".
- [14] W. Goralski, SONET A Guide to Synchronous Optical Network, McGraw Hill, New York, 1997.
- [15] J. Ashenden, The Designer's Guide to VHDL, Morgan Kaufmann Publishers, San Francisco, 1996.
- [16] J. Bergeron, Writing Test Benches: Functional verification of HDL models, Kluwer Academic, Boston, 2000.
- [17] P. Kurup, Logic Synthesis Using Synopsys, Kluwer Academic, Boston, 1997.
- [18] W.F. Lee, VHDL Coding and Logic Synthesis with Synopsys, Academic Press, San Diego CA, 2000.

ANEXOS

En esta sección se presentan los anexos del presente trabajo de tesis.

Se realizó un CD de documentación que contiene el código de los bloques de diseño del módulo codificador FEC en banda, el código de las camas de prueba, los resultados de síntesis de los bloques y los resultados de verificación funcional. Este CD se encuentra disponible para consulta con el auxiliar de investigación del área de telecomunicaciones o bien con los asesores de este trabajo de tesis.

La estructura de directorios del CD de documentación, y la explicación del contenido de cada carpeta y subcarpeta, se presenta en el archivo de texto *readme.txt*, ubicado en el directorio raíz de este CD.

A. Diagramas de tiempo de las señales de control del bloque control del calculador de bytes P1

En este anexo se presentan el resto de los diagramas de tiempo contemplados en el diseño del bloque control del calculador de bytes P1. Los diagramas de tiempo para las filas 1 y 9 se han presentado en la sección 5.6.7 del documento de tesis.

A.1 Diagramas de tiempo para el modo de operación de 2.5 Gbps

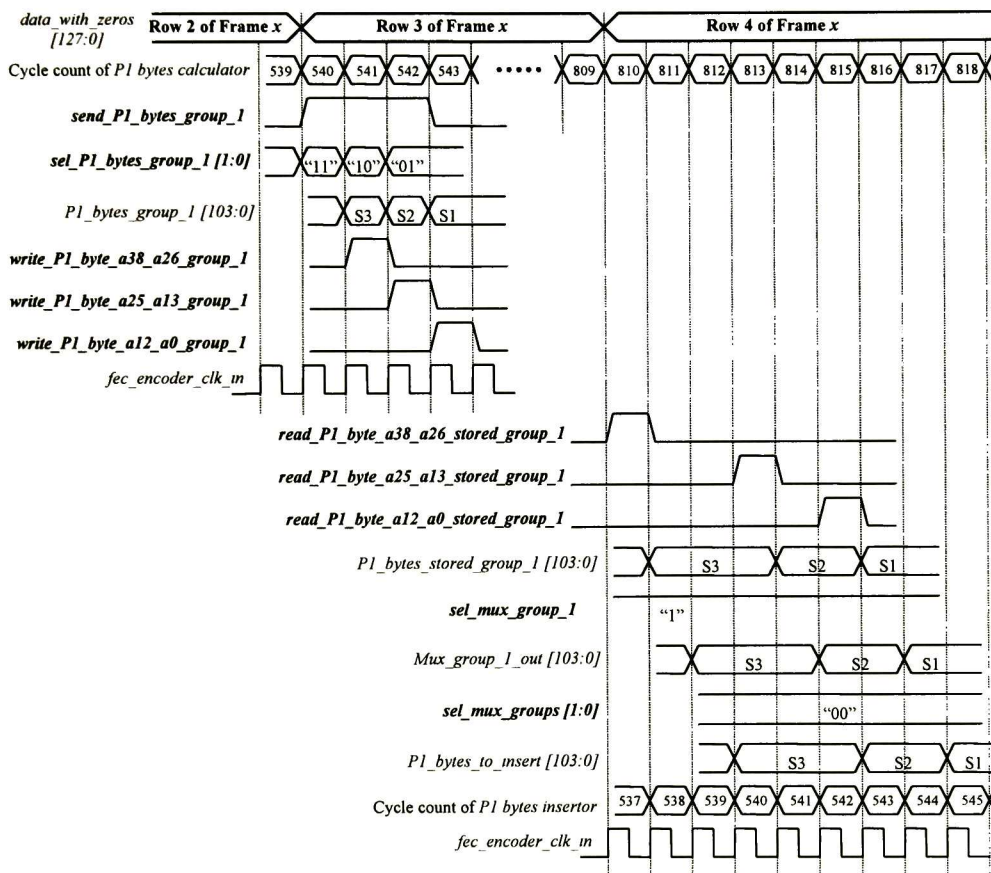


Figura A.1 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 2

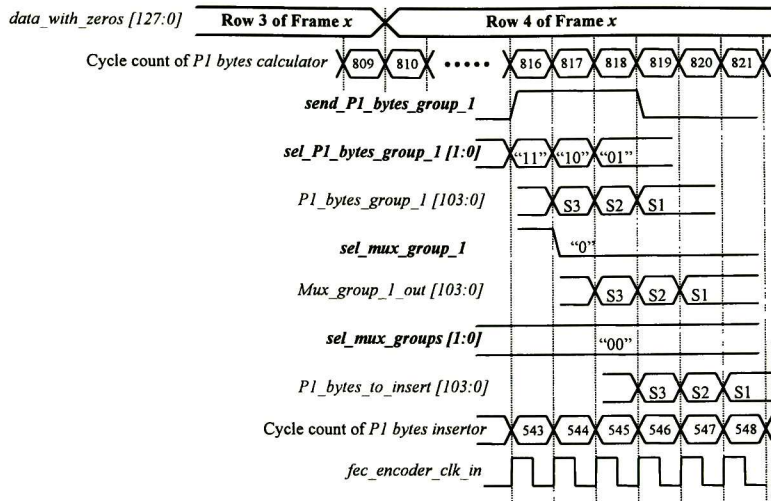


Figura A.2 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 3

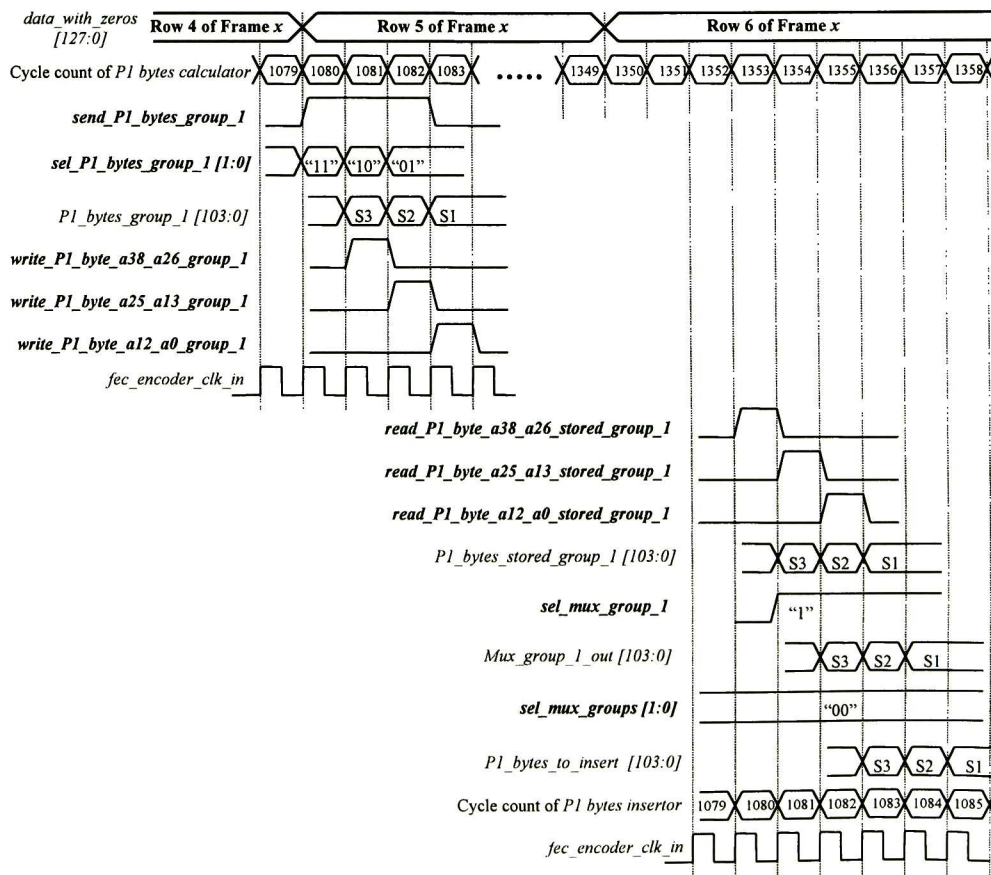


Figura A.3 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 4

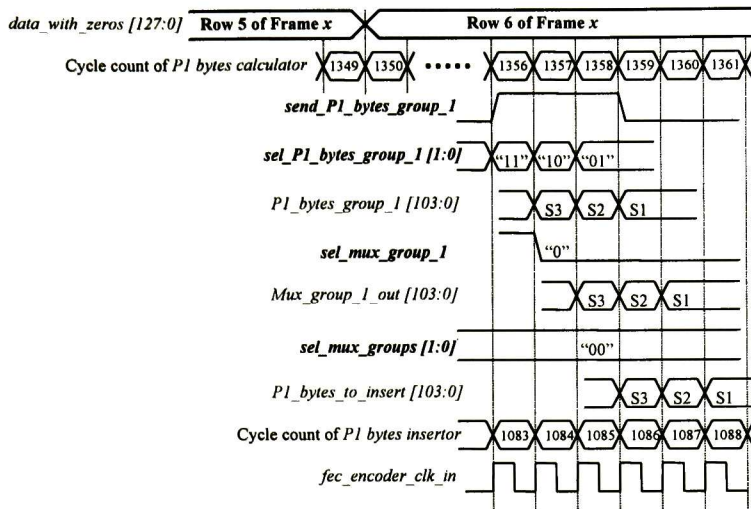


Figura A.4 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 5

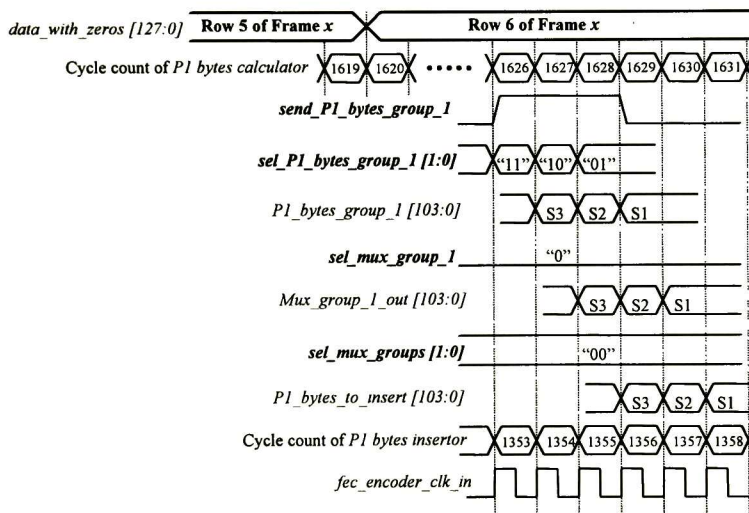


Figura A.5 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 6

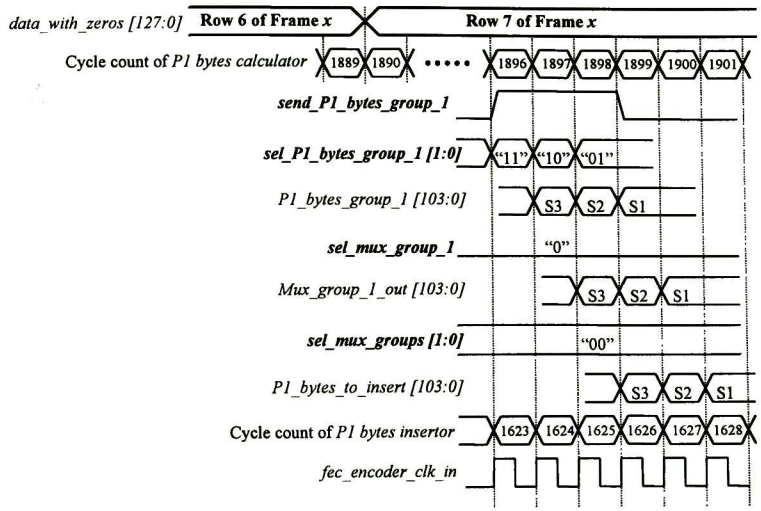


Figura A.6 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 7

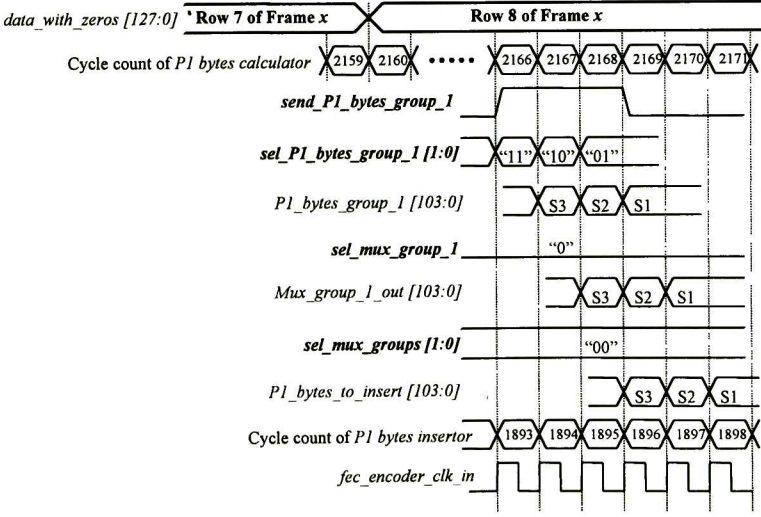


Figura A.7 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 8

A.2 Diagramas de tiempo para el modo de operación de 10 Gbps

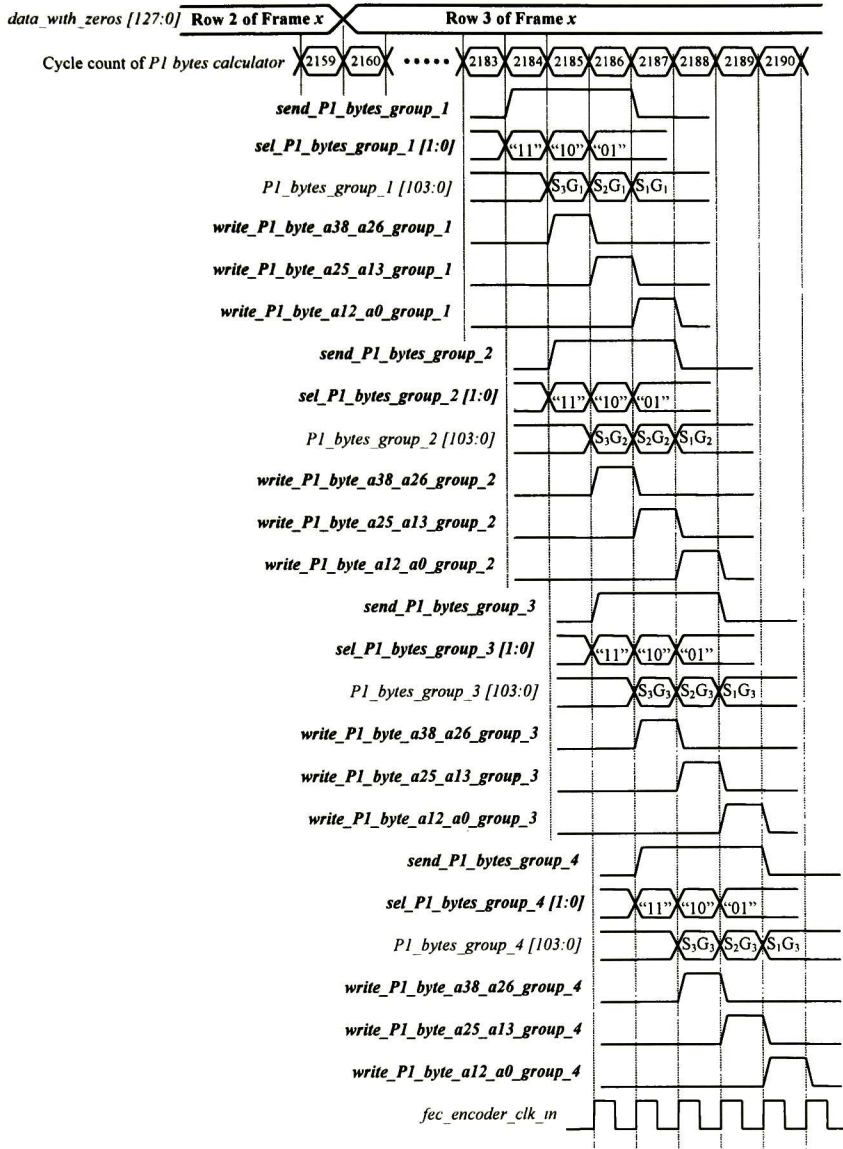


Figura A.8 Diagrama de tiempo de las señales de control durante la escritura de los bytes P1 de la fila 2 en los bloques *Registros de bytes P1*

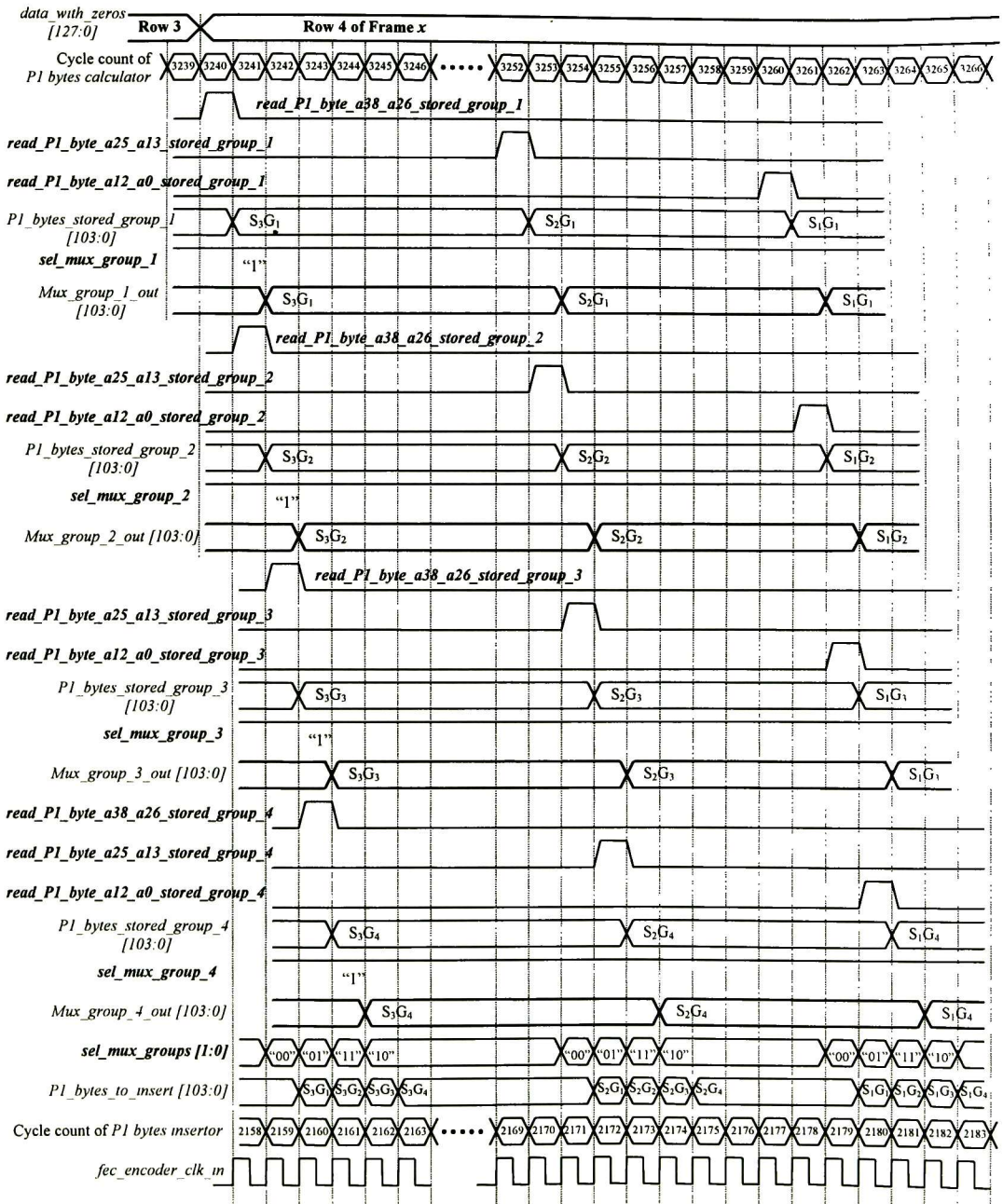


Figura A.9 Diagrama de tiempo de las señales de control durante la lectura de los bytes P1 de la fila 2 almacenados en los bloques *Registros de bytes P1*

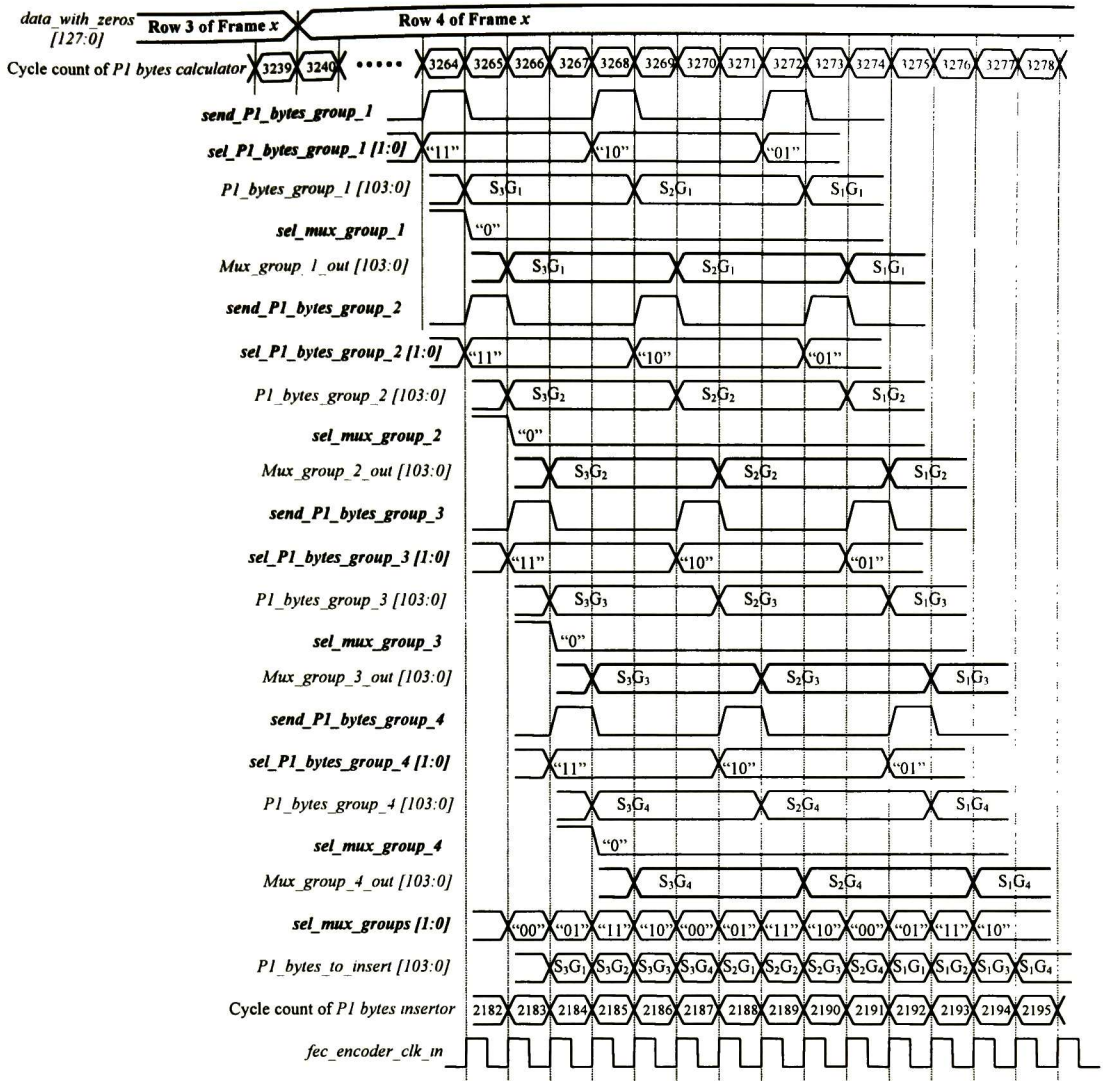


Figura A.10 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 3

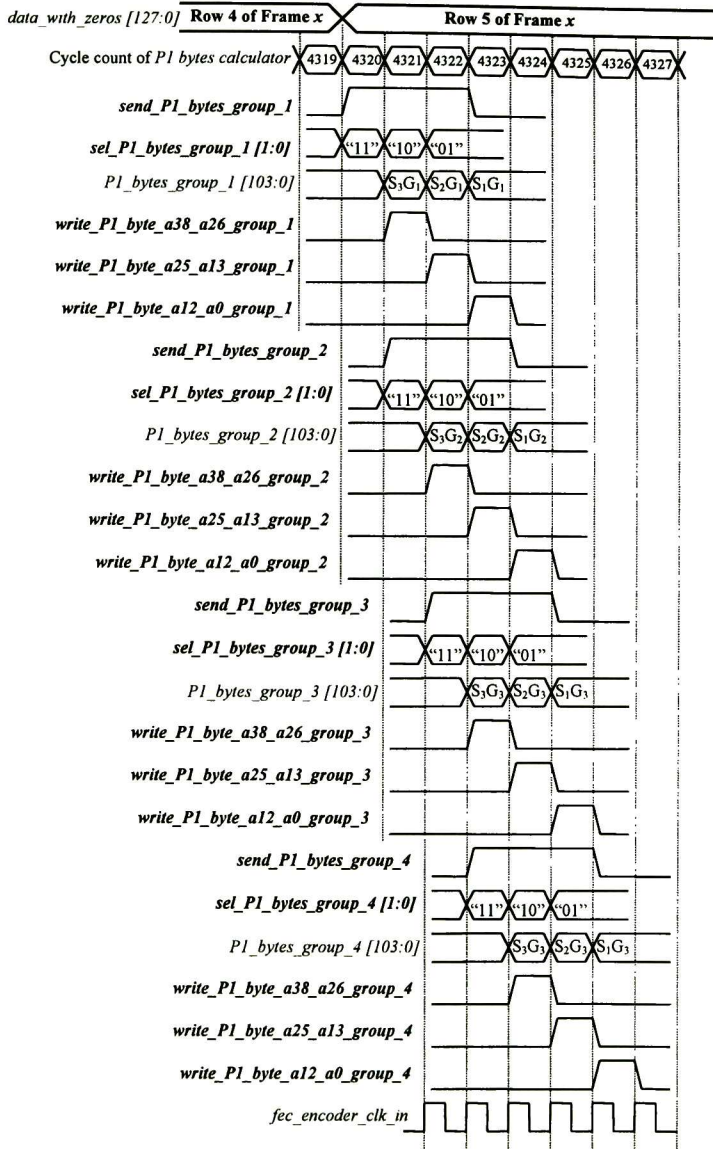


Figura A.11 Diagrama de tiempo de las señales de control durante la escritura de los bytes P1 de la fila 4 en los bloques *Registros de bytes P1*

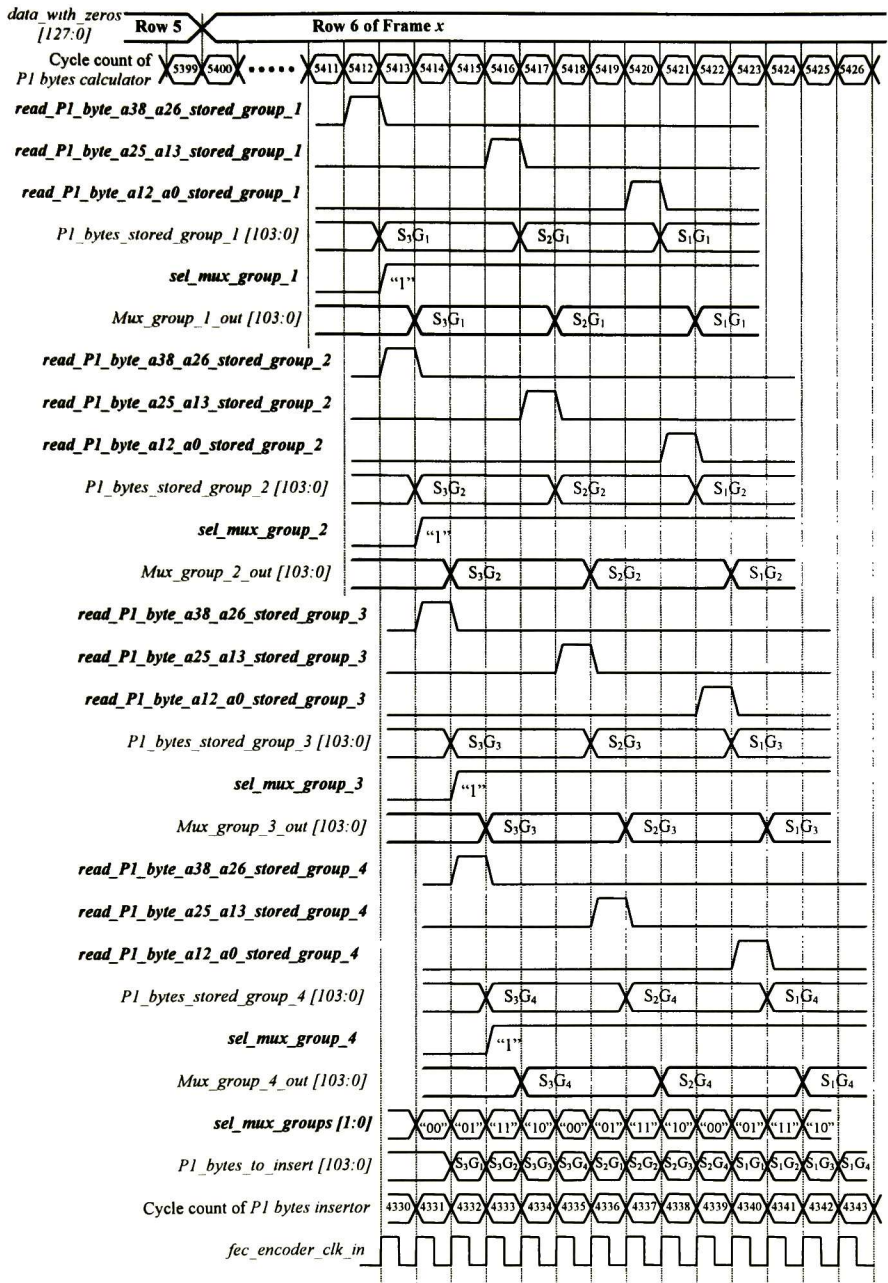


Figura A.12 Diagrama de tiempo de las señales de control durante la lectura de los bytes P1 de la fila 4 almacenados en los bloques *Registros de bytes P1*

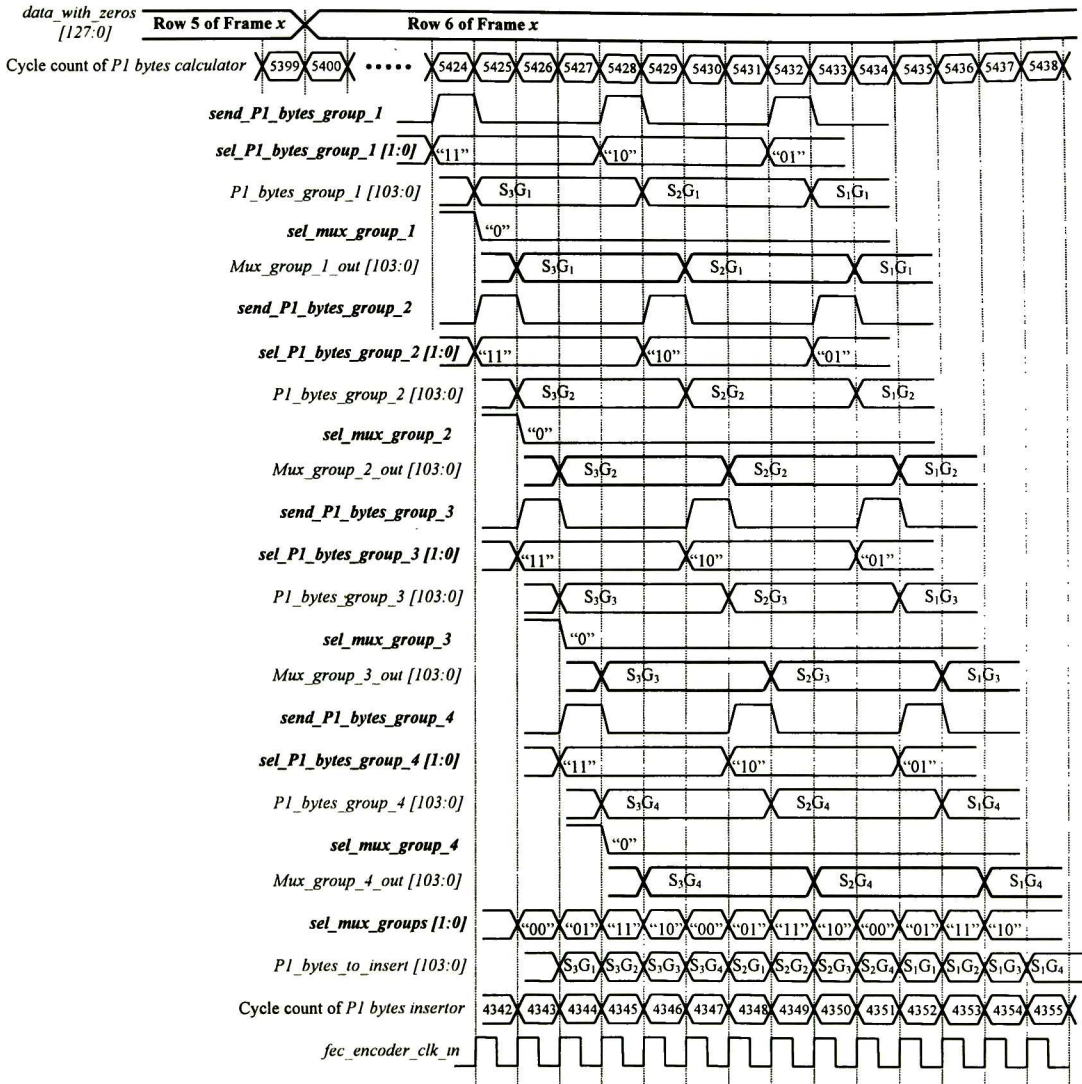


Figura A.13 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 5

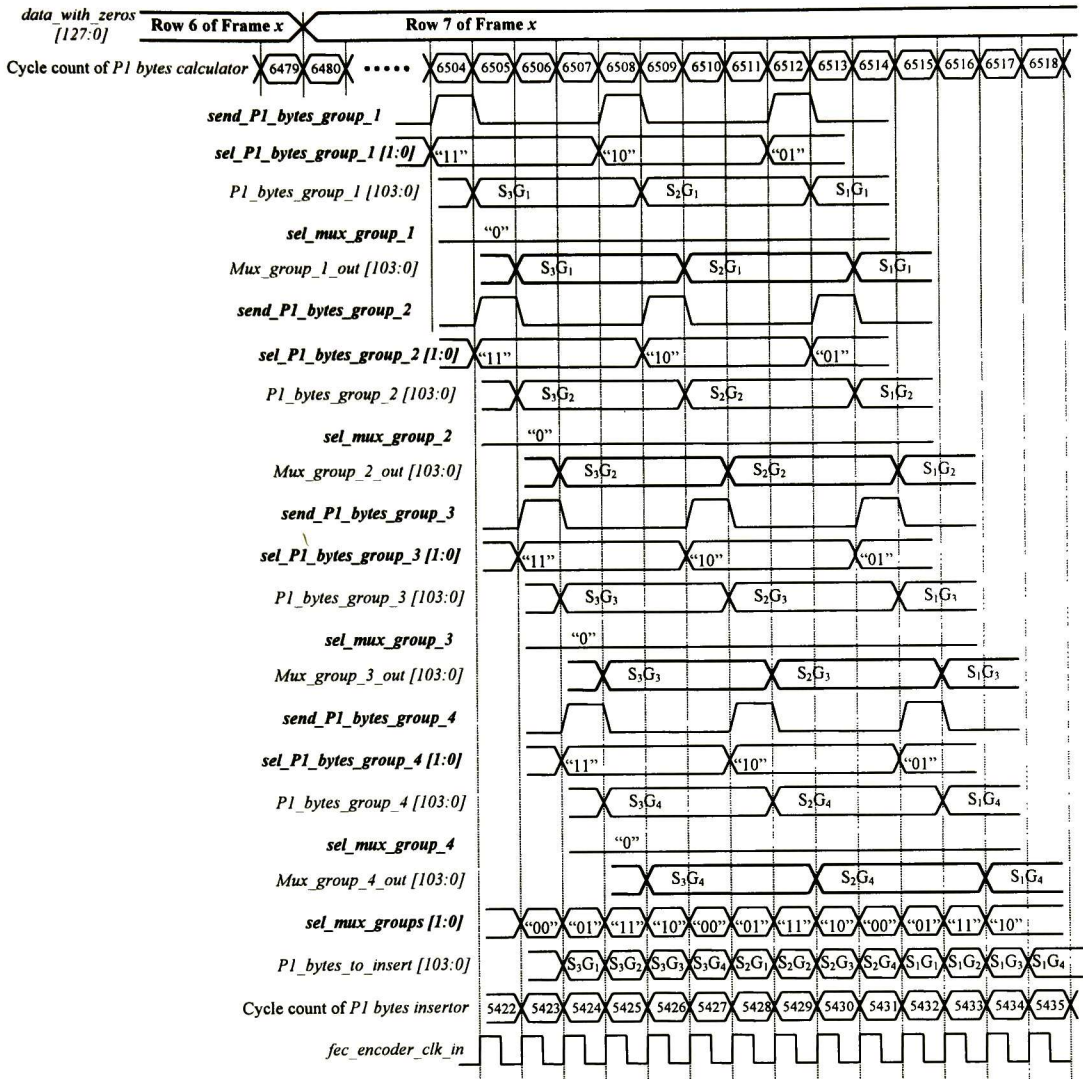


Figura A.14 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 6

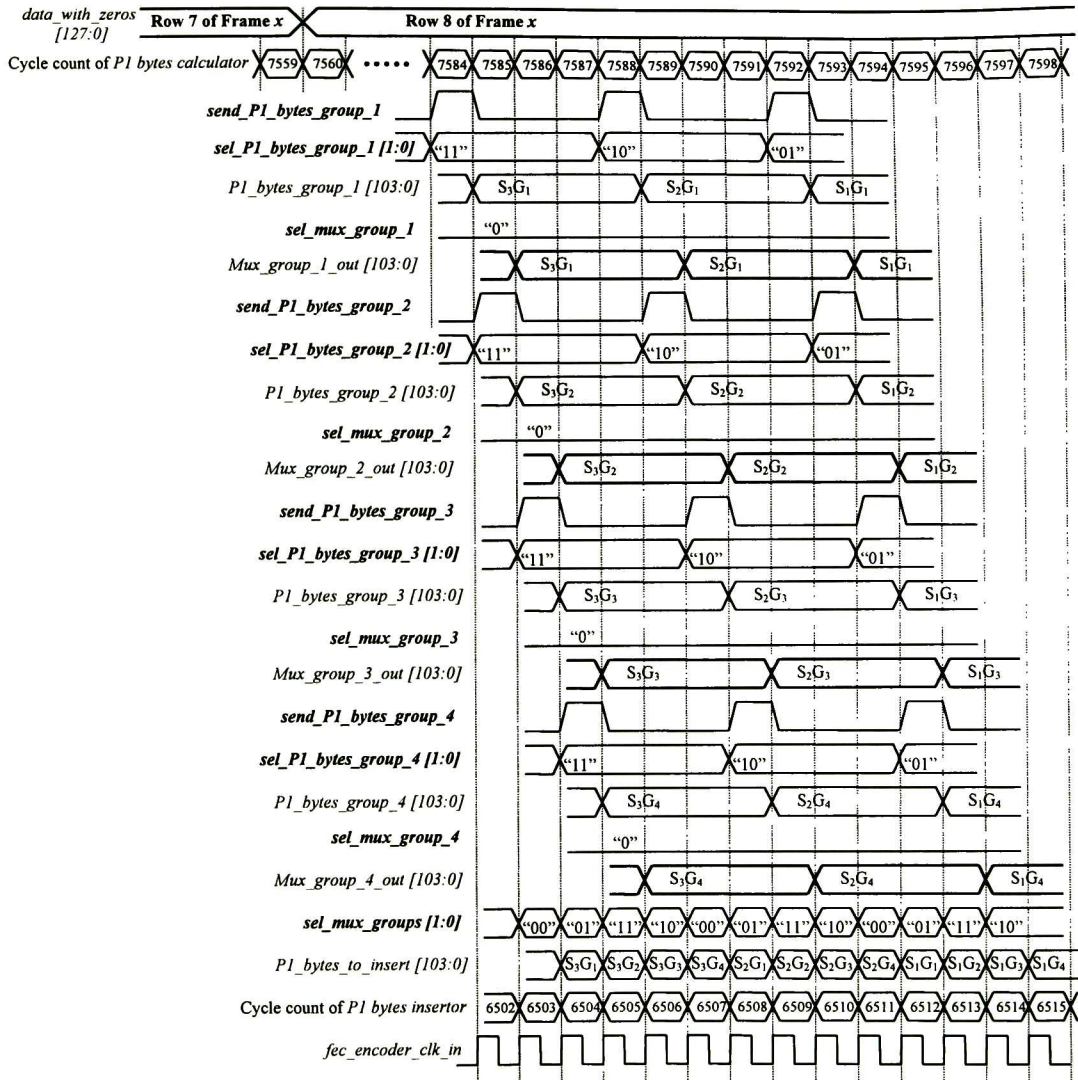


Figura A.15 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 7

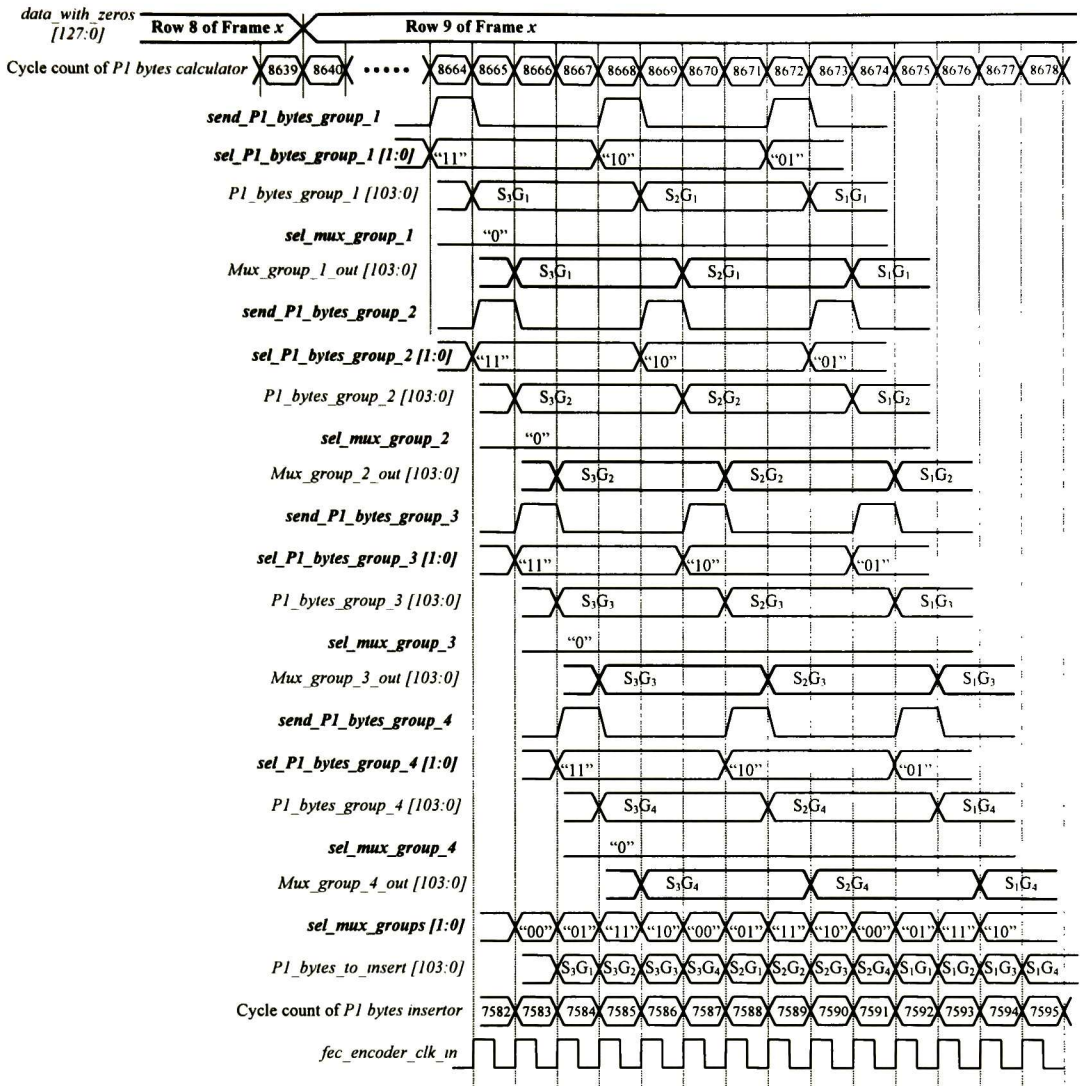


Figura A.16 Diagrama de tiempo de las señales de control durante el envío de bytes P1 de la fila 8

B. Diseño de los subestados de la máquina de control del bloque control del módulo codificador

B.1 Diseño de los subestados asociados al estado *FEC On*

El diseño de los subestados y sus transiciones se basa en las tareas que se deben ejecutar al realizar una transición entre dos estados de la máquina de control. A continuación se presentan todas las transiciones posibles hacia el estado *FEC On*.

B.1.1 Transición: *Reset* → *FEC On*

Esta transición del estado *Reset* hacia el estado *FEC On* implica un proceso de sincronización en el que se transmiten tramas con la señal FSI puesta al valor “01” y en el cual a partir de la octava trama con FSI = “01” se habilita el proceso de codificación (cálculo e inserción de bytes P1).

Condiciones iniciales de los bloques:

Todos los bloques se encuentran deshabilitados.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Habilitar el bloque <i>FSI Insertor</i> y poner la señal <i>fsi_to_insert</i> al valor “01”	-----
2. Habilitar el bloque <i>B2 Bytes Compensator</i>	<i>Enable B2 Comp</i>
3. Habilitar el bloque <i>Data Path Buffer</i>	<i>Enable Data Buffer</i>
4. Habilitar el bloque <i>P1 Bytes Insertor</i>	<i>Enable P1 Insertor</i>
5. Habilitar el bloque <i>B2 Bytes Calculator</i>	<i>Enable B2 Calc</i>
6. Habilitar el bloque <i>Mux Output</i> y poner su señal de selección <i>mux_output_sel</i> a “1” para seleccionar el flujo de datos con retardo de codificación	<i>Enable Mux</i>
7. Esperar a que se transmitan 7 tramas con la señal FSI puesta al valor “01”	<i>Waiting 7 Frames</i>
8. En la octava trama con FSI = “01”, habilitar el bloque <i>Zeros Insertor</i>	<i>Enable Zeros</i>
9. Habilitar el bloque <i>P1 bytes Calculator</i>	<i>Enable P1 Calc</i>
10. Habilitar la función de inserción del bloque <i>P1 bytes Insertor</i> poniendo la señal <i>P1 bytes ins enable</i> al valor “1”	<i>Enable P1 Insertion</i>
11. Esperar a que se solicite una transición de estado	<i>Waiting for State Change</i>

Tabla B.1 Tareas y subestados para la transición del estado *Reset* al estado *FEC On*

Notas:

La tarea 1 no tiene un subestado asociado. Esta tarea se ejecuta en el momento en que se satisface la condición para realizar la transición del estado *Reset* hacia el estado *FEC On*.

B.1.2 Transición: FEC Off with Delay → FEC On

En el estado *FEC Off with Delay* el proceso de codificación se encuentra desactivado. El módulo codificador FEC en banda inserta la señal FSI con un valor “00” e introduce el retardo de codificación sobre el flujo de datos SONET / SDH. Para efectuar la transición hacia el estado *FEC On* se tienen que transmitir tramas con la señal FSI puesta al valor “01” y a partir de la octava trama con FSI = “01” se debe habilitar el proceso de codificación.

Condiciones iniciales de los bloques:

Los bloques *FSI Insertor*, *B2 Bytes Compensator*, *Data Path Buffer*, *P1 Bytes Insertor*, *B2 Bytes Calculator* y *Mux Output* se encuentran habilitados.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Poner la señal <i>fsi_to_insert</i> a “01”	<i>Waiting for Cycle Count</i>
2. Esperar a que se transmitan 7 tramas con la señal FSI puesta al valor “01”	<i>Waiting 7 Frames</i>
3. En la octava trama con FSI = “01”, habilitar el bloque <i>Zeros Insertor</i>	<i>Enable Zeros</i>
4. Habilitar el bloque <i>P1 bytes Calculator</i>	<i>Enable P1 Calc</i>
5. Habilitar la función de inserción del bloque <i>P1 bytes Insertor</i> poniendo la señal <i>P1 bytes ins enable</i> al valor “1”	<i>Enable P1 Insertion</i>
6. Esperar a que se solicite una transición de estado	<i>Waiting for State Change</i>

Tabla B.2 Tareas y subestados para la transición del estado *FEC Off with Delay* al estado *FEC On*

Notas:

La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *FEC Off with Delay* hacia el estado *FEC On*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control es menor al ciclo de inserción de la señal FSI o bien si el contador de ciclos coincide con el fin de una trama. En estos casos el subestado siguiente de la máquina de control será *Waiting 7 Frames*. Ahora, si el valor del contador de ciclos no satisface las condiciones descritas anteriormente, el subestado siguiente será *Waiting for Cycle Count*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincida con el fin de una trama.

B.1.3 Transición: FEC Off without Delay → FEC On

En el estado *FEC Off without Delay* el proceso de codificación se encuentra desactivado. El módulo codificador FEC en banda inserta la señal FSI con un valor “00” y no introduce el retardo de codificación sobre el flujo de datos SONET / SDH. Para efectuar la transición hacia el estado *FEC On* hay que introducir el retardo de codificación sobre el flujo de los datos e insertar el valor “01” en la señal FSI. A partir de la octava trama con FSI = “01” se debe habilitar el proceso de codificación.

Condiciones iniciales de los bloques:

Los bloques *FSI Insertor* y *Mux Output* se encuentran habilitados y el bloque *Mux Output* selecciona el flujo de datos sin retardo de codificación (la señal de selección *mux_output_sel* tiene el valor “0”).

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Poner la señal de selección <i>mux_output_sel</i> del bloque <i>Mux Output</i> a un valor “1” para seleccionar el flujo de datos con retardo de codificación y poner la señal <i>fsi_to_insert</i> a “01”	<i>Waiting for Cycle Count</i>
2. Habilitar el bloque <i>B2 Bytes Compensator</i>	<i>Enable B2 Comp</i>
3. Habilitar el bloque <i>Data Path Buffer</i>	<i>Enable Data Buffer</i>
4. Habilitar el bloque <i>P1 Bytes Insertor</i>	<i>Enable P1 Insertor</i>
5. Habilitar el bloque <i>B2 Bytes Calculator</i>	<i>Enable B2 Calc</i>
6. Esperar a que se transmitan 7 tramas con la señal FSI puesta al valor “01”	<i>Waiting 7 Frames</i>
7. En la octava trama con FSI = “01”, habilitar el bloque <i>Zeros Insertor</i>	<i>Enable Zeros</i>
8. Habilitar el bloque <i>P1 bytes Calculator</i>	<i>Enable P1 Calc</i>
9. Habilitar la función de inserción del bloque <i>P1 bytes Insertor</i> poniendo la señal <i>P1_bytes_ins_enable</i> al valor “1”	<i>Enable P1 Insertion</i>
10. Esperar a que se solicite una transición de estado	<i>Waiting for State Change</i>

Tabla B.3 Tareas y subestados para la transición del estado *FEC Off without Delay* al estado *FEC On*

Notas:

La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *FEC Off without Delay* hacia el estado *FEC On*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control coincide con el fin de una trama. En este caso el subestado siguiente de la máquina de control será *Enable B2 Comp*. Ahora, en caso de que el contador de ciclos no coincida con el fin de una trama, el subestado siguiente será *Waiting for Cycle Count*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincida con el fin de una trama.

B.1.4 Transición: MSF-AIS → FEC On

El estado *MSF-AIS* es un estado de operación especial que toma la máquina de control para generar la señal de indicación de alarma MSF-AIS. Cuando el módulo codificador se encuentra en un estado de operación normal, ya sea *FEC On*, *FEC Off with Delay* o *FEC Off without Delay*, y se solicita la generación de la señal de indicación de alarma MSF-AIS, la máquina de control pasará al estado *MSF-AIS*. El estado y subestado de la máquina de control, previos a esta transición hacia el estado *MSF-AIS*, serán almacenados ya que a ellos se retornará una vez que se dé por terminada la generación de la señal de indicación de alarma.

Cuando ocurre la transición del estado *FEC On* hacia el estado *MSF-AIS* se debe deshabilitar la inserción de la señal FSI y de los bytes P1, así como el cálculo y compensación del código de paridad B2. Además se contribuye a generar la señal de indicación de alarma MSF-AIS al insertar bytes con valor 0xFF en las posiciones de los bytes P1 que se ubican en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH) y en todos los bytes Q1.

En la transición del estado *MSF-AIS* hacia el estado *FEC On* se tiene que cancelar la generación de la señal de indicación de alarma MSF-AIS y se deben habilitar las funciones de inserción de la señal FSI, de inserción de bytes P1, de cálculo y de compensación del código de paridad B2.

Condiciones iniciales de los bloques:

- Los bloques *FSI Insertor*, *B2 Bytes Compensator*, *Data Path Buffer*, *Zeros Insertor*, *P1 Bytes Calculator*, *P1 Bytes Insertor*, *B2 Bytes Calculator* y *Mux Output* se encuentran habilitados.
- La señal *fsi_insertor_msf_ais* del bloque *FSI Insertor* tiene el valor “1”, indicando que se encuentra habilitada la inserción de bytes con valor 0xFF en las posiciones de los bytes Q1 y, por tanto, se ha deshabilitado la inserción de la señal FSI.
- La señal *B2_bytes_compensator_msf_ais* del bloque *B2 Bytes Compensator* tiene el valor “1”, indicando que se encuentra deshabilitada la función de compensación del código de paridad B2.
- La señal *P1_bytes_insertor_msf_ais* del bloque *P1 Bytes Insertor* tiene el valor “1”, indicando que se encuentra habilitada la inserción de bytes con valor 0xFF en las posiciones de los bytes P1 ubicados en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH).
- La señal *B2_bytes_calculator_msf_ais* del bloque *B2 Bytes Calculator* tiene el valor “1”, indicando que se encuentra deshabilitado el cálculo del código de paridad B2.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Poner la señal <i>fsi_insertor_msf_ais</i> del bloque <i>FSI Insertor</i> a un valor “0” para cancelar la inserción de bytes con valor 0xFF en las posiciones de los bytes Q1 y para habilitar la inserción de la señal FSI	<i>Disable MSF-AIS FSI Insertor</i>

2. Poner la señal <i>B2_bytes_compensator_msf_ais</i> del bloque <i>B2 Bytes Compensator</i> a un valor "0" para habilitar la función de compensación del código de paridad B2	<i>Disable MSF-AIS B2 Comp</i>
3. Poner la señal <i>P1_bytes_insertor_msf_ais</i> del bloque <i>P1 Bytes Insertor</i> a un valor "0" para cancelar la inserción de bytes con valor 0xFF en las posiciones de los bytes P1 ubicados en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH)	<i>Disable MSF-AIS P1 Insertor</i>
4. Poner la señal <i>B2_bytes_calculator_msf_ais</i> del bloque <i>B2 Bytes Calculator</i> a un valor "0" para habilitar el cálculo del código de paridad B2	<i>Disable MSF-AIS B2 Calc</i>

Tabla B.4 Tareas y subestados para la transición del estado *MSF-AIS* al estado *FEC On*

Notas:

- La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *MSF-AIS* hacia el estado *FEC On*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control coincide con el fin de una trama. En este caso el subestado siguiente de la máquina de control será *Disable MSF-AIS B2 Comp*. Ahora, en caso de que el contador de ciclos no coincida con el fin de una trama, el subestado siguiente será *Disable MSF-AIS FSI Insertor*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincida con el fin de una trama.
- La transición del estado *FEC On* hacia el estado *MSF-AIS* se realiza sólo si la máquina de control se encuentra en el subestado *Waiting 7 Frames* o en el subestado *Waiting for State Change*.

B.1.5 Subestados asociados al estado *FEC On*

En la Figura B.1 se muestran los subestados asociados al estado *FEC On* y se detallan las transiciones entre estos subestados.

B.2 Diseño de los subestados asociados al estado *FEC Off with Delay*

A continuación se presentan todas las transiciones posibles hacia el estado *FEC Off with Delay*.

B.2.1 Transición: *Reset* → *FEC Off with Delay*

En esta transición del estado *Reset* hacia el estado *FEC Off with Delay* se tienen que habilitar y configurar los bloques para que el módulo codificador opere de manera correcta.

Condiciones iniciales de los bloques:

Todos los bloques se encuentran deshabilitados.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Habilitar el bloque <i>FSI Insertor</i> y poner la señal <i>fsi_to_insert</i> al valor "00"	-----
2. Habilitar el bloque <i>B2 Bytes Compensator</i>	<i>Enable B2 Comp</i>
3. Habilitar el bloque <i>Data Path Buffer</i>	<i>Enable Data Buffer</i>
4. Habilitar el bloque <i>P1 Bytes Insertor</i>	<i>Enable P1 Insertor</i>
5. Habilitar el bloque <i>B2 Bytes Calculator</i>	<i>Enable B2 Calc</i>
6. Habilitar el bloque <i>Mux Output</i> y poner su señal de selección <i>mux_output_sel</i> a "1" para seleccionar el flujo de datos con retardo de codificación	<i>Enable Mux</i>
7. Esperar a que se solicite una transición de estado	<i>Waiting for State Change</i>

Tabla B.5 Tareas y subestados para la transición del estado *Reset* al estado *FEC Off with Delay*

Notas:

La tarea 1 no tiene un subestado asociado. Esta tarea se ejecuta en el momento en que se satisface la condición para realizar la transición del estado *Reset* hacia el estado *FEC Off with Delay*.

B.2.2 Transición: *FEC Off without Delay* → *FEC Off with Delay*

En el estado *FEC Off without Delay*, el módulo codificador FEC en banda inserta la señal FSI con un valor "00" y no se introduce el retardo de codificación sobre el flujo de datos SONET / SDH.

Condiciones iniciales de los bloques:

El bloque *FSI Inserter* y *Mux Output* se encuentran habilitados y el bloque *Mux Output* selecciona el flujo de datos sin retardo de codificación (la señal de selección *mux_output_sel* tiene el valor “0”).

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Poner la señal de selección <i>mux_output_sel</i> del bloque <i>Mux Output</i> a un valor “1” para seleccionar el flujo de datos con retardo de codificación	<i>Waiting for Cycle Count</i>
2. Habilitar el bloque <i>B2 Bytes Compensator</i>	<i>Enable B2 Comp</i>
3. Habilitar el bloque <i>Data Path Buffer</i>	<i>Enable Data Buffer</i>
4. Habilitar el bloque <i>P1 Bytes Inserter</i>	<i>Enable P1 Inserter</i>
5. Habilitar el bloque <i>B2 Bytes Calculator</i>	<i>Enable B2 Calc</i>
6. Esperar a que se solicite una transición de estado	<i>Waiting for State Change</i>

Tabla B.6 Tareas y subestados para la transición del estado *FEC Off without Delay* al estado *FEC Off with Delay*

Notas:

La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *FEC Off without Delay* hacia el estado *FEC Off with Delay*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control coincide con el fin de una trama. En este caso el subestado siguiente de la máquina de control será *Enable B2 Comp*. Ahora, en caso de que el contador de ciclos no coincida con el fin de una trama, el subestado siguiente será *Waiting for Cycle Count*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincida con el fin de una trama.

B.2.3 Transición: *FEC On* → *FEC Off with Delay*

En el estado *FEC On* se encuentra habilitado el proceso de codificación. El módulo codificador inserta la señal FSI con un valor “01” e introduce el retardo de codificación sobre el flujo de datos SONET / SDH. Para efectuar la transición hacia el estado *FEC Off with Delay* se tienen que transmitir tramas con la señal FSI puesta al valor “00” y a partir de la octava trama con FSI = “00” se debe deshabilitar el proceso de codificación.

Condiciones iniciales de los bloques:

Los bloques *FSI Inserter*, *B2 Bytes Compensator*, *Data Path Buffer*, *Zeros Inserter*, *P1 Bytes Calculator*, *P1 Bytes Inserter*, *B2 Bytes Calculator* y *Mux Output* se encuentran habilitados.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Poner la señal <i>fsi_to_insert</i> a "00"	<i>Waiting for Cycle Count</i>
2. Esperar a que se transmitan 7 tramas con la señal FSI puesta al valor "00"	<i>Waiting 7 Frames</i>
3. En la octava trama con FSI = "00", deshabilitar el bloque <i>Zeros Insertor</i>	<i>Disable Zeros</i>
4. Deshabilitar el bloque <i>P1 bytes Calculator</i>	<i>Disable P1 Calc</i>
5. Deshabilitar la función de inserción del bloque <i>P1 bytes Insertor</i> poniendo la señal <i>P1 bytes ins enable</i> al valor "0"	<i>Disable P1 Insertion</i>
6. Esperar a que se solicite una transición de estado	<i>Waiting for State Change</i>

Tabla B.7 Tareas y subestados para la transición del estado *FEC On* al estado *FEC Off with Delay*

Notas:

La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *FEC On* hacia el estado *FEC Off with Delay*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control es menor al ciclo de inserción de la señal FSI o bien si el contador de ciclos coincide con el fin de una trama. En estos casos el subestado siguiente de la máquina de control será *Waiting 7 Frames*. Ahora, si el valor del contador de ciclos no satisface las condiciones descritas anteriormente, el subestado siguiente será *Waiting for Cycle Count*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincida con el fin de una trama.

B.2.4 Transición: MSF-AIS → FEC Off with Delay

En la transición del estado *MSF-AIS* hacia el estado *FEC Off with Delay* se tiene que cancelar la generación de la señal de indicación de alarma MSF-AIS y se deben habilitar las funciones de inserción de la señal FSI y el cálculo y compensación del código de paridad B2.

Condiciones iniciales de los bloques:

- Los bloques *FSI Insertor*, *B2 Bytes Compensator*, *Data Path Buffer*, *P1 Bytes Insertor*, *B2 Bytes Calculator* y *Mux Output* se encuentran habilitados.
- La señal *fsi_insertor_msf_ais* del bloque *FSI Insertor* tiene el valor "1", indicando que se encuentra habilitada la inserción de bytes con valor 0xFF en las posiciones de los bytes Q1 y, por tanto, se ha deshabilitado la inserción de la señal FSI.
- La señal *B2_bytes_compensator_msf_ais* del bloque *B2 Bytes Compensator* tiene el valor "1", indicando que se encuentra deshabilitada la función de compensación del código de paridad B2.
- La señal *P1_bytes_insertor_msf_ais* del bloque *P1 Bytes Insertor* tiene el valor "1", indicando que se encuentra habilitada la inserción de bytes con valor 0xFF en las posiciones de los bytes P1 ubicados en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH).

- La señal *B2_bytes_calculator_msf_ais* del bloque *B2 Bytes Calculator* tiene el valor “1”, indicando que se encuentra deshabilitado el cálculo del código de paridad B2.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Poner la señal <i>fsi_insertor_msf_ais</i> del bloque <i>FSI Insertor</i> a un valor “0” para cancelar la inserción de bytes con valor 0xFF en las posiciones de los bytes Q1 y para habilitar la inserción de la señal FSI	<i>Disable MSF-AIS FSI Insertor</i>
2. Poner la señal <i>B2_bytes_compensator_msf_ais</i> del bloque <i>B2 Bytes Compensator</i> a un valor “0” para habilitar la función de compensación del código de paridad B2	<i>Disable MSF-AIS B2 Comp</i>
3. Poner la señal <i>P1_bytes_insertor_msf_ais</i> del bloque <i>P1 Bytes Insertor</i> a un valor “0” para cancelar la inserción de bytes con valor 0xFF en las posiciones de los bytes P1 ubicados en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH)	<i>Disable MSF-AIS P1 Insertor</i>
4. Poner la señal <i>B2_bytes_calculator_msf_ais</i> del bloque <i>B2 Bytes Calculator</i> a un valor “0” para habilitar el cálculo del código de paridad B2	<i>Disable MSF-AIS B2 Calc</i>

Tabla B.8 Tareas y subestados para la transición del estado *MSF-AIS* al estado *FEC Off with Delay*

Notas:

- La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *MSF-AIS* hacia el estado *FEC Off with Delay*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control coincide con el fin de una trama. En este caso el subestado siguiente de la máquina de control será *Disable MSF-AIS B2 Comp*. Ahora, en caso de que el contador de ciclos no coincida con el fin de una trama, el subestado siguiente será *Disable MSF-AIS FSI Insertor*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincida con el fin de una trama.
- La transición del estado *FEC Off with Delay* hacia el estado *MSF-AIS* se realiza sólo si la máquina de control se encuentra en el subestado *Waiting 7 Frames* o en el subestado *Waiting for State Change*.

B.2.5 Subestados asociados al estado *FEC Off with Delay*

En la Figura B.2 se muestran los subestados asociados al estado *FEC Off with Delay* y se detallan las transiciones entre estos subestados.

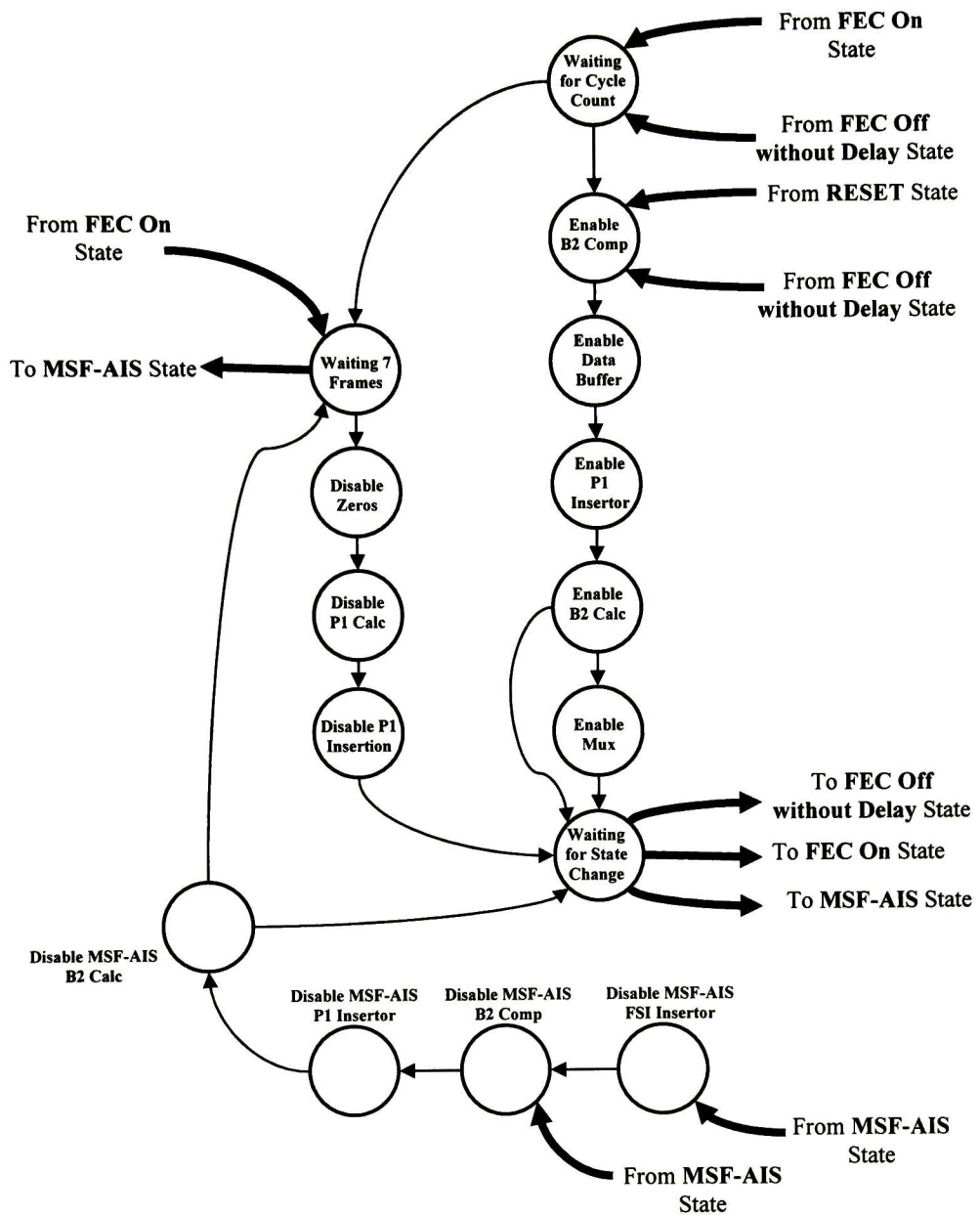


Figura B.2 Subestados asociados al estado FEC Off with Delay

B.3 Diseño de los subestados asociados al estado *FEC Off without Delay*

A continuación se presentan todas las transiciones posibles hacia el estado *FEC Off without Delay*.

B.3.1 Transición: *Reset* → *FEC Off without Delay*

En esta transición del estado *Reset* hacia el estado *FEC Off without Delay* se tienen que habilitar y configurar los bloques para que el módulo codificador opere de manera correcta.

Condiciones iniciales de los bloques:

Todos los bloques se encuentran deshabilitados.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Habilitar el bloque <i>FSI Inserter</i> y poner la señal <i>fsi_to_insert</i> al valor "00". Habilitar el bloque <i>Mux Output</i> y poner su señal de selección <i>mux_output_sel</i> a "0" para seleccionar el flujo de datos sin retardo de codificación	-----
2. Esperar a que se solicite una transición de estado	<i>Waiting for State Change</i>

Tabla B.9 Tareas y subestados para la transición del estado *Reset* al estado *FEC Off without Delay*

Notas:

La tarea 1 no tiene un subestado asociado. Esta tarea se ejecuta en el momento en que se satisface la condición para realizar la transición del estado *Reset* hacia el estado *FEC Off without Delay*.

B.3.2 Transición: *FEC Off with Delay* → *FEC Off without Delay*

En el estado *FEC Off with Delay*, el módulo codificador introduce el retardo de codificación sobre el flujo de datos SONET / SDH. En la transición del estado *FEC Off with Delay* hacia el estado *FEC Off without Delay* se deben deshabilitar los bloques que introducen el retardo de codificación y el flujo de datos SONET / SDH que se envía a la salida del módulo codificador no debe contener el retardo de codificación.

Condiciones iniciales de los bloques:

Los bloques *FSI Inserter*, *B2 Bytes Compensator*, *Data Path Buffer*, *P1 Bytes Inserter*, *B2 Bytes Calculator* y *Mux Output* se encuentran habilitados.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Deshabilitar los bloques <i>B2 Bytes Compensator</i> , <i>Data Path Buffer</i> , <i>P1 bytes Insertor</i> y <i>B2 Bytes Calculator</i> . Poner la señal de selección <i>mux_output_sel</i> del bloque <i>Mux Output</i> a un valor "0" para seleccionar el flujo de datos sin retardo de codificación	<i>Disable Delay Blocks</i>
2. Esperar a que se solicite una transición de estado	<i>Waiting for State Change</i>

Tabla B.10 Tareas y subestados para la transición del estado *FEC Off with Delay* al estado *FEC Off without Delay*

Notas:

La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *FEC Off with Delay* hacia el estado *FEC Off without Delay*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control coincide con el inicio de una trama. En este caso el subestado siguiente de la máquina de control será *Waiting for State Change*. Ahora, si el valor del contador de ciclos no coincide con el inicio de una trama, el subestado siguiente será *Disable Delay Blocks*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincida con el inicio de una trama.

B.3.3 Transición: *FEC On* → *FEC Off without Delay*

En el estado *FEC On*, el proceso de codificación se encuentra activado y el módulo codificador introduce el retardo de codificación sobre el flujo de datos SONET / SDH. En la transición del estado *FEC On* hacia el estado *FEC Off without Delay* se deben deshabilitar los bloques que introducen el retardo de codificación y que realizan el proceso de codificación. El flujo de datos SONET / SDH que se envía a la salida del módulo codificador no debe contener el retardo de codificación.

Condiciones iniciales de los bloques:

Los bloques *FSI Insertor*, *B2 Bytes Compensator*, *Data Path Buffer*, *Zeros Insertor*, *P1 Bytes Calculator*, *P1 Bytes Insertor*, *B2 Bytes Calculator* y *Mux Output* se encuentran habilitados.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Poner la señal <i>fsi_to_insert</i> a "00"	<i>Waiting for Cycle Count</i>
2. Esperar a que se transmitan 7 tramas con la señal FSI puesta al valor "00"	<i>Waiting 7 Frames</i>
3. Deshabilitar los bloques <i>B2 Bytes Compensator</i> , <i>Data Path Buffer</i> , <i>P1 bytes Insertor</i> y <i>B2 Bytes Calculator</i> . Poner la señal de selección <i>mux_output_sel</i> del bloque <i>Mux Output</i> a un valor "0" para seleccionar el flujo de datos sin retardo de codificación	<i>Disable Delay Blocks</i>

4. Esperar a que se solicite una transición de estado	<i>Waiting for State Change</i>
---	---------------------------------

Tabla B.11 Tareas y subestados para la transición del estado *FEC On* al estado *FEC Off without Delay*

Notas:

La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *FEC On* hacia el estado *FEC Off without Delay*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control es menor al ciclo de inserción de la señal FSI o bien si el contador de ciclos coincide con el fin de una trama. En estos casos el subestado siguiente de la máquina de control será *Waiting 7 Frames*. Ahora, si el valor del contador de ciclos no satisface las condiciones descritas anteriormente, el subestado siguiente será *Waiting for Cycle Count*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincida con el fin de una trama.

B.3.4 Transición: MSF-AIS → FEC Off without Delay

La transición del estado *FEC Off without Delay* hacia el estado *MSF-AIS* se realiza sólo si la máquina de control se encuentra en el subestado *Waiting 7 Frames* o en el subestado *Waiting for State Change*.

Si cuando se efectúa la transición del estado *MSF-AIS* hacia el estado *FEC Off without Delay*, el contador de ciclos del bloque de control coincide con el fin de una trama el subestado siguiente de la máquina de control será *Waiting 7 Frames* o bien *Waiting for State Change*. En caso contrario, el subestado siguiente de la máquina de control será *Waiting for Cycle Count*.

B.3.5 Subestados asociados al estado FEC Off without Delay

En la Figura B.3 se muestran los subestados asociados al estado *FEC Off without Delay* y se detallan las transiciones entre estos subestados.

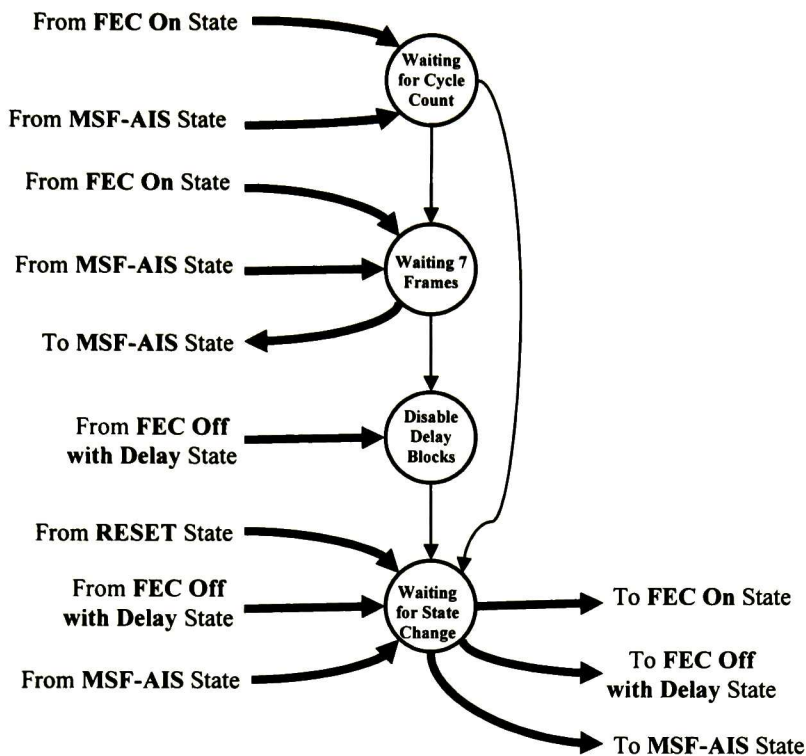


Figura B.3 Subestados asociados al estado FEC Off without Delay

B.4 Diseño de los subestados asociados al estado MSF-AIS

A continuación se presentan todas las transiciones posibles hacia el estado *MSF-AIS*.

B.4.1 Transición: FEC On → MSF-AIS

Cuando ocurre esta transición se debe deshabilitar la inserción de la señal FSI y de los bytes P1, así como el cálculo y compensación del código de paridad B2. Además se contribuye a generar la señal de indicación de alarma MSF-AIS al insertar bytes con valor 0xFF en las posiciones de los bytes P1 que se ubican en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH) y en todos los bytes Q1.

Condiciones iniciales de los bloques:

- Los bloques *FSI Insertor*, *B2 Bytes Compensator*, *Data Path Buffer*, *Zeros Insertor*, *P1 Bytes Calculator*, *P1 Bytes Insertor*, *B2 Bytes Calculator* y *Mux Output* se encuentran habilitados.
- La señal *fsi_insertor_msf_ais* del bloque *FSI Insertor* tiene el valor “0”, indicando que se encuentra habilitada la inserción de la señal FSI.

- La señal *B2_bytes_compensator_msf_ais* del bloque *B2 Bytes Compensator* tiene el valor “0”, indicando que se encuentra habilitada la función de compensación del código de paridad B2.
- La señal *P1_bytes_inserter_msf_ais* del bloque *P1 Bytes Inserter* tiene el valor “0”, indicando que se encuentra habilitada la inserción de bytes P1.
- La señal *B2_bytes_calculator_msf_ais* del bloque *B2 Bytes Calculator* tiene el valor “0”, indicando que se encuentra habilitado el cálculo del código de paridad B2.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Poner la señal <i>fsi_inserter_msf_ais</i> del bloque <i>FSI Inserter</i> a un valor “1” para insertar bytes con valor 0xFF en las posiciones de los bytes Q1 y para deshabilitar la inserción de la señal FSI	<i>Enable MSF-AIS FSI Inserter</i>
2. Poner la señal <i>B2_bytes_compensator_msf_ais</i> del bloque <i>B2 Bytes Compensator</i> a un valor “1” para deshabilitar la función de compensación del código de paridad B2	<i>Enable MSF-AIS B2 Comp</i>
3. Poner la señal <i>P1_bytes_inserter_msf_ais</i> del bloque <i>P1 Bytes Inserter</i> a un valor “1” para insertar bytes con valor 0xFF en las posiciones de los bytes P1 ubicados en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH) y deshabilitar la inserción de los bytes P1 calculados	<i>Enable MSF-AIS P1 Inserter</i>
4. Poner la señal <i>B2_bytes_calculator_msf_ais</i> del bloque <i>B2 Bytes Calculator</i> a un valor “1” para deshabilitar el cálculo del código de paridad B2	<i>Enable MSF-AIS B2 Calc</i>

Tabla B.12 Tareas y subestados para la transición del estado *FEC On* al estado *MSF-AIS*

Notas:

- La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *FEC On* hacia el estado *MSF-AIS*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control coincide con el fin de una trama. En este caso el subestado siguiente de la máquina de control será *Enable MSF-AIS B2 Comp*. Ahora, en caso de que el contador de ciclos no coincida con el fin de una trama, el subestado siguiente será *Enable MSF-AIS FSI Inserter*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincide con el fin de una trama.
- La transición del estado *FEC On* hacia el estado *MSF-AIS* se realiza sólo si la máquina de control se encuentra en el subestado *Waiting 7 Frames* o en el subestado *Waiting for State Change*.

B.4.2 Transición: *FEC Off with Delay* → *MSF-AIS*

Cuando ocurre esta transición se debe deshabilitar la inserción de la señal FSI y el cálculo y compensación del código de paridad B2. Además se contribuye a generar la señal de indicación de alarma MSF-AIS al insertar bytes con valor 0xFF en las posiciones de los

bytes P1 que se ubican en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH) y en todos los bytes Q1.

Condiciones iniciales de los bloques:

- Los bloques *FSI Inserter*, *B2 Bytes Compensator*, *Data Path Buffer*, *P1 Bytes Inserter*, *B2 Bytes Calculator* y *Mux Output* se encuentran habilitados.
- La señal *fsi_inserter_msf_ais* del bloque *FSI Inserter* tiene el valor “0”, indicando que se encuentra habilitada la inserción de la señal FSI.
- La señal *B2_bytes_compensator_msf_ais* del bloque *B2 Bytes Compensator* tiene el valor “0”, indicando que se encuentra habilitada la función de compensación del código de paridad B2.
- La señal *P1_bytes_inserter_msf_ais* del bloque *P1 Bytes Inserter* tiene el valor “0”, indicando que se encuentra deshabilitada la inserción de bytes con valor 0xFF en las posiciones de los bytes P1 ubicados en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH).
- La señal *B2_bytes_calculator_msf_ais* del bloque *B2 Bytes Calculator* tiene el valor “0”, indicando que se encuentra habilitado el cálculo del código de paridad B2.

Tareas a ejecutar y sus subestados asociados:

Tarea	Subestado
1. Poner la señal <i>fsi_inserter_msf_ais</i> del bloque <i>FSI Inserter</i> a un valor “1” para insertar bytes con valor 0xFF en las posiciones de los bytes Q1 y para deshabilitar la inserción de la señal FSI	<i>Enable MSF-AIS FSI Inserter</i>
2. Poner la señal <i>B2_bytes_compensator_msf_ais</i> del bloque <i>B2 Bytes Compensator</i> a un valor “1” para deshabilitar la función de compensación del código de paridad B2	<i>Enable MSF-AIS B2 Comp</i>
3. Poner la señal <i>P1_bytes_inserter_msf_ais</i> del bloque <i>P1 Bytes Inserter</i> a un valor “1” para insertar bytes con valor 0xFF en las posiciones de los bytes P1 ubicados en la tara de sección SOH (SONET) / tara de sección de regeneración RSOH (SDH)	<i>Enable MSF-AIS P1 Inserter</i>
4. Poner la señal <i>B2_bytes_calculator_msf_ais</i> del bloque <i>B2 Bytes Calculator</i> a un valor “1” para deshabilitar el cálculo del código de paridad B2	<i>Enable MSF-AIS B2 Calc</i>

Tabla B.13 Tareas y subestados para la transición del estado *FEC Off with Delay* al estado *MSF-AIS*

Notas:

- La tarea 1 se puede ejecutar al momento de realizar la transición desde el estado *FEC Off with Delay* hacia el estado *MSF-AIS*. Lo anterior ocurre si, cuando se satisface la condición de transición entre estados, el valor del contador de ciclos del bloque de control coincide con el fin de una trama. En este caso el subestado siguiente de la máquina de control será *Enable MSF-AIS B2 Comp*. Ahora, en caso de que el contador de ciclos no coincida con el fin de una trama, el subestado siguiente será *Enable MSF-AIS FSI Inserter*. La tarea 1 se ejecuta en este subestado una vez que el contador de ciclos coincida con el fin de una trama.

- La transición del estado *FEC Off with Delay* hacia el estado *MSF-AIS* se realiza sólo si la máquina de control se encuentra en el subestado *Waiting 7 Frames* o en el subestado *Waiting for State Change*.

B.4.3 Transición: *FEC Off without Delay* → *MSF-AIS*

La transición del estado *FEC Off without Delay* hacia el estado *MSF-AIS* se realiza sólo si la máquina de control se encuentra en el subestado *Waiting 7 Frames* o en el subestado *Waiting for State Change*.

B.4.4 Subestados asociados al estado *MSF-AIS*

En la Figura B.4 se muestran los subestados asociados al estado *MSF-AIS* y se detallan las transiciones entre estos subestados.

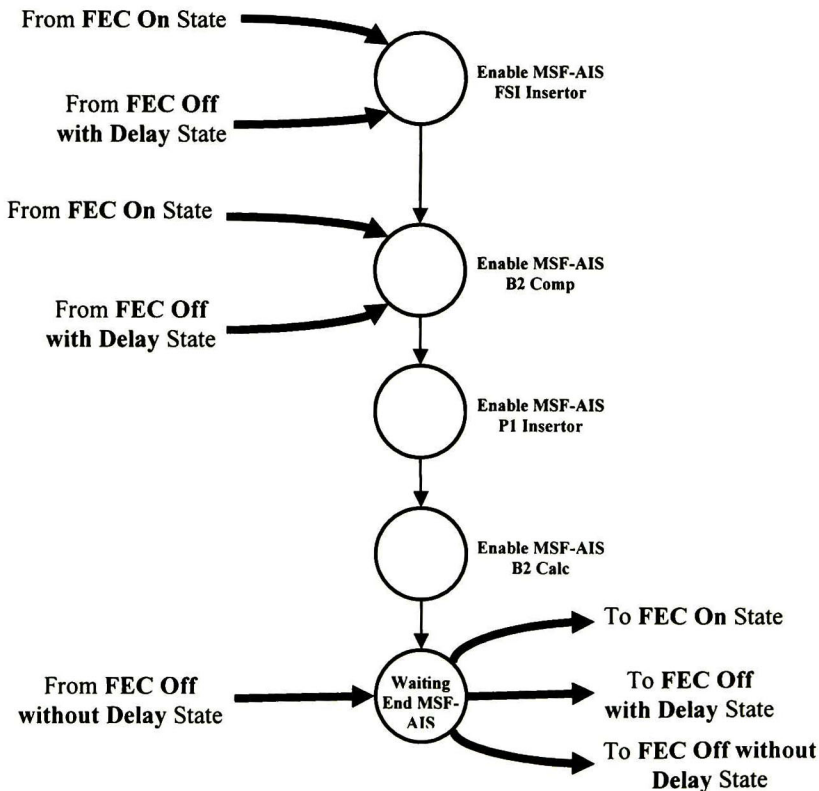


Figura B.4 Subestados asociados al estado *MSF-AIS*

C. Plan de verificación del módulo codificador FEC en banda

En este anexo se presenta el plan de verificación del módulo codificador FEC en banda. En el se describen detalladamente los casos y subcasos de prueba que se han definido para la verificación del módulo.

C.1 Caso de prueba TC-FEC-ENC-01

Descripción general:

El presente caso de prueba verifica el correcto funcionamiento del bloque *Insertor de la señal FSI*.

Cuando se encuentre deshabilitada la generación de la señal MSF-AIS, el bloque *Insertor de la señal FSI* realizará la inserción de la señal de dos bits FSI (*FEC Status Indicator*) en los bits 7 y 8 del primer byte Q1, cuyas coordenadas son $S(3,9,3)$, siendo el bit 8 el menos significativo. Los bits 1 al 6, de dicho byte Q1, serán transmitidos con valor "0". Los valores de transmisión válidos para la señal FSI son "00" y "01". Los valores de transmisión inválidos para la señal FSI, "10" y "11", no deberán ser insertados por el bloque y en su lugar se deberá insertar el valor "00", indicando que el módulo codificador se encuentra deshabilitado.

Cuando se encuentre habilitada la generación de la señal MSF-AIS, el bloque *Insertor de la señal FSI* insertará el valor 0xFF en todos los bytes Q1.

Requerimientos a verificar:

Los requerimientos que se verifican con el caso de prueba TC-FEC-ENC-01, son los siguientes:

- R-FEC-ENC-01
- R-FEC-ENC-11
- R-FEC-ENC-17
- R-FEC-ENC-20
- R-FEC-ENC-21
- R-FEC-ENC-22

Subcasos de Prueba:

El caso de prueba TC-FEC-ENC-01 está dividido en los siguientes subcasos de prueba:

- SC-FEC-ENC-01
- SC-FEC-ENC-02

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_tc_fec_enc_01.vhd*. Primero ejecuta el subcaso de prueba SC-FEC-ENC-01 y posteriormente el subcaso de prueba SC-FEC-ENC-02. La cama de prueba genera las señales de control para el bloque *Insertor de la señal FSI* e inyecta el flujo de datos de entrada a partir de los archivos *CSV I_data_sc_fec_enc_01.CSV* e *I_data_sc_fec_enc_02.CSV*.

En esta cama de prueba, hay un proceso que se encarga de analizar el flujo de datos a la salida del bloque *Insertor de la señal FSI* para verificar que su comportamiento funcional sea acorde a las señales de control presentes en los puertos de entrada. Así mismo se verifica la correcta generación de la señal *J0_detected* a la salida del bloque.

A continuación se describen los subcasos de prueba.

C.1.1 Subcaso de prueba SC-FEC-ENC-01

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Insertor de la señal FSI* a 2.5 Gbps (STS-48 / STM-16).

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
fec_encoder_data_in[127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_01.CSV</i>)
fec_encoder_J0_detected_in	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)
fsi_insertor_reset	0
fec_encoder_config	0 (STM-16)
fsi_to_insert	00 (tramas 1 a 4) 01 (tramas 5 a 8) 10 (tramas 9 a 12) 11 (tramas 13 a 16) 01 (tramas 17 a 30)
fsi_insertor_msf_ais	0 (tramas 1 a 16 y 25 a 30) 1 (tramas 17 a 24)

Tabla C.1 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-01

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo *CSV I_data_sc_fec_enc_01.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-48 / STM-16 con contenido aleatorio.

Respuestas:

El flujo de datos de salida del bloque *Insertor de la señal FSI* se escribe en el archivo *O_data_sc_fec_enc_01.CSV*.

En el archivo de reporte *O_asserts_tc_fec_enc_01.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: de las tramas 1 a 4 se deberá insertar el valor 0x00 en el byte FSI (FSI = "00"); de las tramas 5 a 8 se deberá insertar el valor 0x01 en el byte FSI (FSI = "01"); de las tramas 9 a 16 se deberá insertar el valor 0x00 en el byte FSI (los valores de FSI "10" y "11" son inválidos, en su lugar se insertará "00"); de las tramas 17 a 24 se deberá insertar el valor 0xFF en el byte FSI (se encuentra habilitada la generación de la señal MSF-AIS); de las tramas 25 a 30 se deberá insertar el valor 0x01 en el byte FSI (FSI = "01").

C.1.2 Subcaso de prueba SC-FEC-ENC-02

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Insertor de la señal FSI* a 10 Gbps (STS-192 / STM-64).

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
fec_encoder_data_in[127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_02.CSV</i>)
fec_encoder_J0_detected_in	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
fsi_insertor_reset	0
fec_encoder_config	1 (STM-64)
fsi_to_insert	00 (tramas 1 a 4) 01 (tramas 5 a 8) 10 (tramas 9 a 12) 11 (tramas 13 a 16) 01 (tramas 17 a 30)
fsi_insertor_msf_ais	0 (tramas 1 a 16 y 25 a 30) 1 (tramas 17 a 24)

Tabla C.2 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-02

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_sc_fec_enc_02.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio.

Respuestas:

El flujo de datos de salida del bloque *Insertor de la señal FSI* se escribe en el archivo *O_data_sc_fec_enc_02.CSV*.

En el archivo de reporte *O_asserts_tc_fec_enc_01.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: de las tramas 1 a 4 se deberá insertar el valor 0x00 en el byte FSI (FSI = "00"); de las tramas 5 a 8 se deberá insertar el valor 0x01 en el byte FSI (FSI = "01"); de las tramas 9 a 16 se deberá insertar el valor 0x00 en el byte FSI (los valores de FSI "10" y "11" son inválidos, en su lugar se insertará "00"); de las tramas 17 a 24 se deberá insertar el valor 0xFF en los 4 bytes Q1 (se encuentra habilitada la generación de la señal MSF-AIS); de las tramas 25 a 30 se deberá insertar el valor 0x01 en el byte FSI (FSI = "01").

C.2 Caso de prueba TC-FEC-ENC-02

Descripción general:

El presente caso de prueba verifica que el bloque *Insertor de ceros* realice la inserción de bytes con valor 0x00 en las posiciones de la trama SONET / SDH que no se consideran en el proceso de codificación FEC en banda.

Las posiciones de la trama SONET / SDH que no se consideran en el proceso de codificación FEC en banda son las siguientes:

- Todos los bytes de la tara de sección (SOH) / tara de sección de regeneración (RSOH) de la trama SONET / SDH, exceptuando a todos los bytes Q1, los cuales sí serán considerados en el proceso de codificación FEC en banda.
- Las posiciones correspondientes a todos los bytes P1.

Requerimientos a verificar:

Los requerimientos que se verifican con el caso de prueba TC-FEC-ENC-02, son los siguientes:

- R-FEC-ENC-01
- R-FEC-ENC-10

Subcasos de Prueba:

El caso de prueba TC-FEC-ENC-02 está dividido en los siguientes subcasos de prueba:

- SC-FEC-ENC-03

- SC-FEC-ENC-04

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_tc_fec_enc_02.vhd*. Primero ejecuta el subcaso de prueba SC-FEC-ENC-03 y posteriormente el subcaso de prueba SC-FEC-ENC-04. La cama de prueba genera las señales de control para el bloque *Insertor de ceros* e inyecta el flujo de datos de entrada a partir de los archivos *CSV I_data_sc_fec_enc_03.CSV* e *I_data_sc_fec_enc_04.CSV*.

En esta cama de prueba, hay un proceso que se encarga de analizar el flujo de datos a la salida del bloque *Insertor de ceros* para verificar su comportamiento funcional. Así mismo se verifica la correcta generación de la señal *J0_detected* a la salida del bloque.

Archivos de estímulos modificados:

Los archivos de estímulos originales *I_data_sc_fec_enc_03_original.CSV* e *I_data_sc_fec_enc_04_original.CSV* fueron obtenidos mediante una herramienta de software (referencia [13] del documento de tesis). Estas tramas generadas contienen bytes con valor 0x00 en posiciones en donde el bloque *Insertor de ceros* realizará su función; dada esta situación, no se sabrá si el bloque realmente ha insertado valores 0x00 en dichos bytes o simplemente actuó de manera transparente dejando sin alterar esos bytes. Por tanto se optó por modificar dichos archivos de estímulos mediante la inserción de valores diferentes a 0x00 en toda la tara de transporte / tara de sección de las tramas SONET / SDH, con el objeto de permitir al proceso de verificación comprobar que realmente se inserten valores 0x00 en las posiciones debidas.

La modificación de los archivos originales es realizada por un módulo escrito en lenguaje de descripción de hardware, en el archivo *I_data_modifier_tc_fec_enc_02.vhd*. Los archivos modificados son escritos en los archivos *I_data_sc_fec_enc_03.CSV* e *I_data_sc_fec_enc_04.CSV*.

A continuación se describen los subcasos de prueba.

C.2.1 Subcaso de prueba SC-FEC-ENC-03

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Insertor de ceros* a 2.5 Gbps (STS-48 / STM-16).

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_B2_bytes_compensated[127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_03.CSV</i>)

J0_detected_compensator	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)
zeros_insertor_reset	0
fec_encoder_config	0 (STM-16)

Tabla C.3 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-03

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_sc_fec_enc_03.CSV*. Consiste en 10 tramas SONET / SDH de orden STS-48 / STM-16 con contenido aleatorio y con la particularidad de que los bytes de la tara de transporte / tara de sección tienen valor diferente a 0x00.

Respuestas:

El flujo de datos de salida del bloque *Insertor de ceros* se escribe en el archivo *O_data_sc_fec_enc_03.CSV*.

En el archivo de reporte *O_asserts_tc_fec_enc_02.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.2.2 Subcaso de prueba SC-FEC-ENC-04

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Insertor de ceros* a 10 Gbps (STS-192 / STM-64).

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_B2_bytes_compensated [127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_04.CSV</i>)
J0_detected_compensator	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
zeros_insertor_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.4 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-04

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_sc_fec_enc_04.CSV*. Consiste en 10 tramas SONET / SDH de orden STS-192 /

STM-64 con contenido aleatorio y con la particularidad de que los bytes de la tara de transporte / tara de sección tienen valor diferente a 0x00.

Respuestas:

El flujo de datos de salida del bloque *Insertor de ceros* se escribe en el archivo *O_data_sc_fec_enc_04.CSV*.

En el archivo de reporte *O_asserts_tc_fec_enc_02.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3 Caso de prueba TC-FEC-ENC-03

Descripción general:

El presente caso de prueba verifica el correcto funcionamiento del bloque *Calculador de bytes P1*.

Primero, se verificará que el flujo de datos SONET / SDH de 128 bits sea separado en $8 \times N/16$ flujos mediante un desentrelazado de $8 \times N/16$ bits (ver requerimiento R-FEC-ENC-05), en donde N toma un valor igual a 16 cuando se opera en el modo de 2.5 Gbps (STS-48 / STM-16) y un valor igual a 64 cuando se opera en el modo de 10 Gbps (STS-192 / STM-64). Cada flujo obtenido representará el flujo de bits de información para cada una de las $8 \times N/16$ subfilas en que se arreglan los bits de cada fila de las tramas SONET / SDH. Cada 8 flujos consecutivos representarán el flujo de bits para un *grupo* de palabras de información, teniéndose en total $N/16$ grupos.

Segundo, se verificará que todos los bloques *calculadores de bits a_n* realicen correctamente el cálculo de dichos bits.

Tercero, se verificará que a partir de los bits a_n calculados para las subfilas de todos los grupos se formen correctamente los bytes P1.

Por último, se verificará que los bytes P1 sean enviados en el tiempo y en el orden adecuado hacia el puerto de salida del bloque *Calculador de bytes P1*, para que sean insertados por el bloque *Insertor de bytes P1*.

Requerimientos a verificar:

Los requerimientos que se verifican con el caso de prueba TC-FEC-ENC-03, son los siguientes:

- R-FEC-ENC-01
- R-FEC-ENC-02
- R-FEC-ENC-03

- R-FEC-ENC-05
- R-FEC-ENC-06
- R-FEC-ENC-07
- R-FEC-ENC-08
- R-FEC-ENC-09

Subcasos de Prueba:

El caso de prueba TC-FEC-ENC-03 está dividido en los siguientes subcasos de prueba:

- SC-FEC-ENC-05
- SC-FEC-ENC-06
- SC-FEC-ENC-07
- SC-FEC-ENC-08
- SC-FEC-ENC-09
- SC-FEC-ENC-10
- SC-FEC-ENC-11
- SC-FEC-ENC-12
- SC-FEC-ENC-13
- SC-FEC-ENC-14
- SC-FEC-ENC-15
- SC-FEC-ENC-16
- SC-FEC-ENC-17
- SC-FEC-ENC-18
- SC-FEC-ENC-19

Esquema de verificación:

Dada la complejidad del bloque *Calculador de bytes P1* se hace necesario verificar su comportamiento funcional mediante entidades que se encarguen de comprobar la ejecución de las tareas de sus bloques internos. Este tipo de verificación se conoce como de caja gris, en donde se tiene acceso a señales internas del bloque a verificar.

Una entidad ubicada en el exterior del bloque *Calculador de bytes P1* e implementada en el archivo *tb_tc_fec_enc_03.vhd* se encargará de generar los flujos de entrada SONET / SDH y de generar la señal de reloj y las señales de control. Los flujos se generan a partir de los archivos CSV de entrada: *I_data_tc_fec_enc_03_STM16.CSV* e *I_data_tc_fec_enc_03_STM64.CSV*. Cada archivo contiene 30 tramas SONET / SDH. Estos archivos fueron obtenidos al ejecutar el caso de prueba TC-FEC-ENC-02 con los siguientes archivos de entrada: *I_data_sc_fec_enc_01.CSV* (STM-16) e *I_data_sc_fec_enc_02.CSV* (STM-64).

Las tareas a verificar en los bloques internos parten de los requerimientos que debe satisfacer en forma global el bloque *Calculador de bytes P1*. Estas tareas internas serán verificadas por entidades que se ubicarán en el interior del bloque *Calculador de bytes P1*. En la Figura C.1 se muestra el esquema de verificación a nivel *top* del presente caso de prueba.

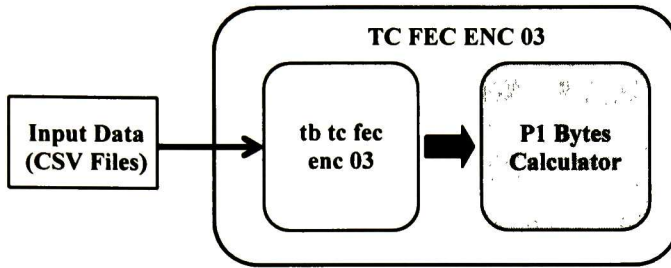


Figura C.1 Esquema de verificación para el caso de prueba TC-FEC-ENC-03

En la Figura C.2 se muestra el esquema de verificación implementado en el interior del bloque *Calculador de bytes P1*, en donde se observan los bloques encargados de ejecutar los subcasos de prueba 15, 16, 17, 18 y 19. Estos bloques verifican los datos a la salida de los bloques calculadores y el correcto funcionamiento de los bloques registros de bytes P1 y bloques multiplexores.

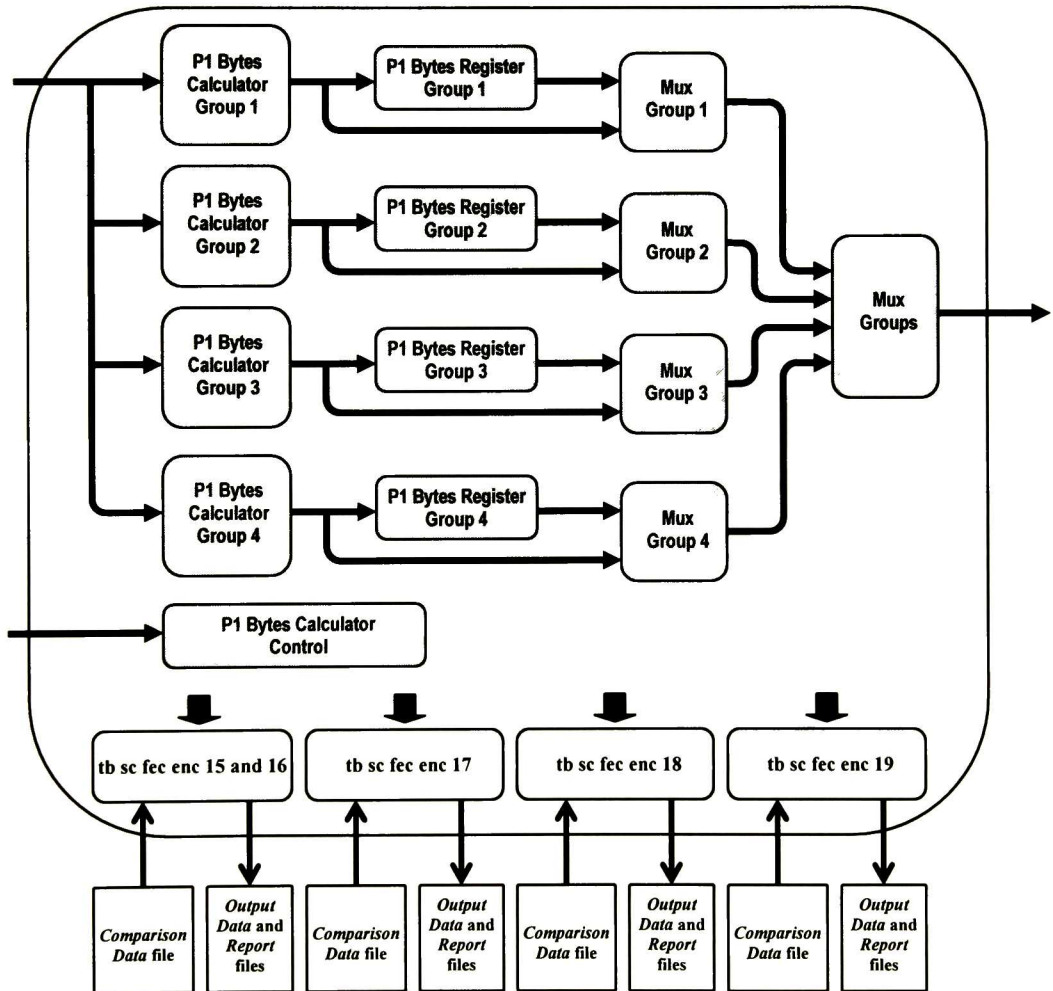


Figura C.2 Esquema de verificación en el interior del bloque *Calculador de bytes P1*

Los bloques que ejecutan los subcasos de prueba del 5 al 14 se ubican en el interior de los bloques *Calculador de bytes P1 para el grupo 1, grupo 2, grupo 3 y grupo*. De la Figura C.3 a la Figura C.6 se muestran los esquemas de verificación implementados en el interior de estos bloques.

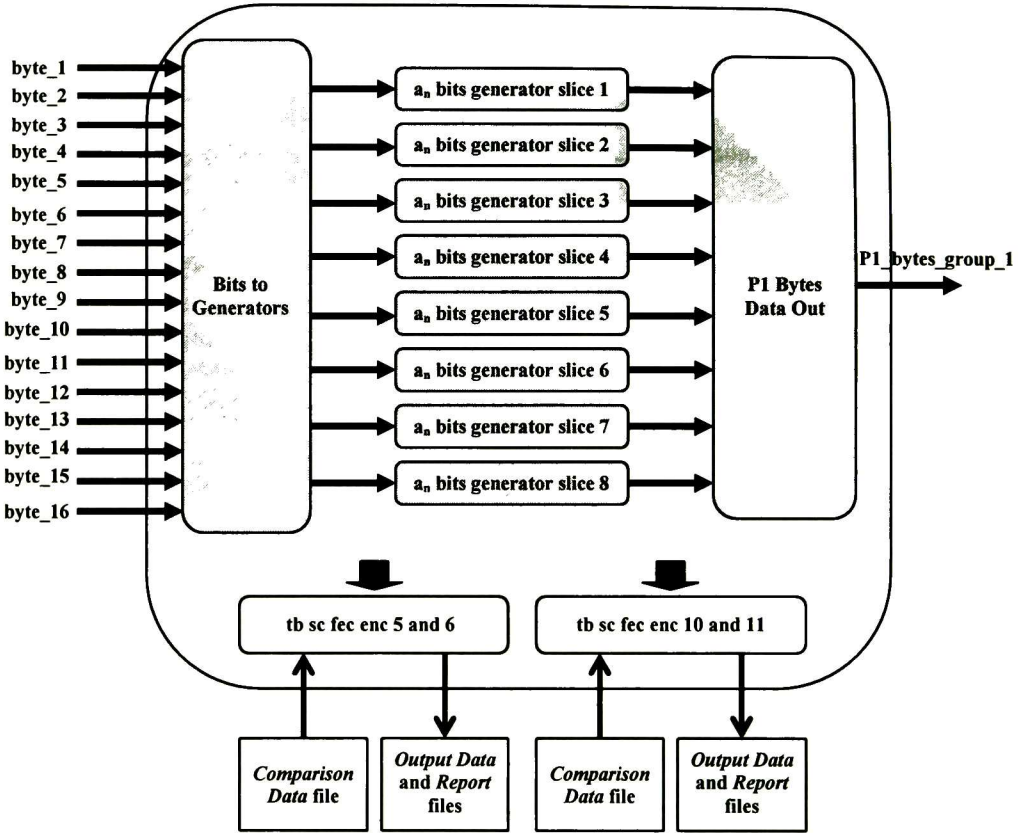


Figura C.3 Esquema de verificación en el interior del bloque *Calculador de bytes P1 para el grupo 1*

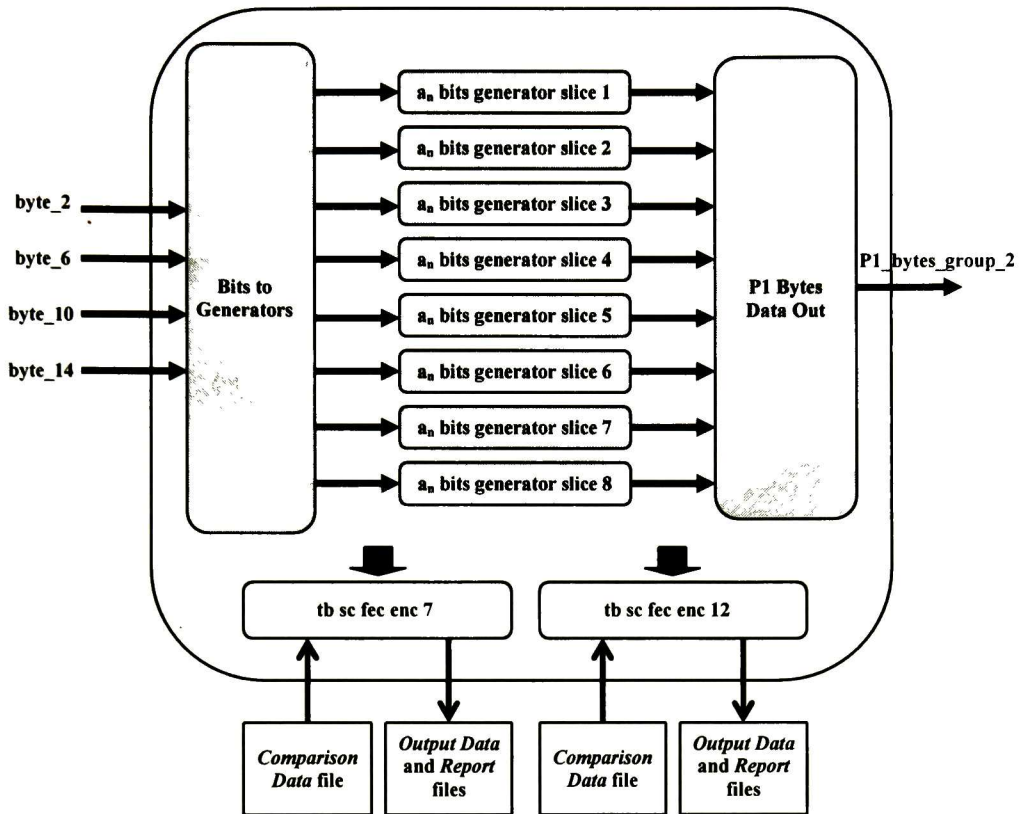


Figura C.4 Esquema de verificación en el interior del bloque *Calculador de bytes P1 para el grupo 2*

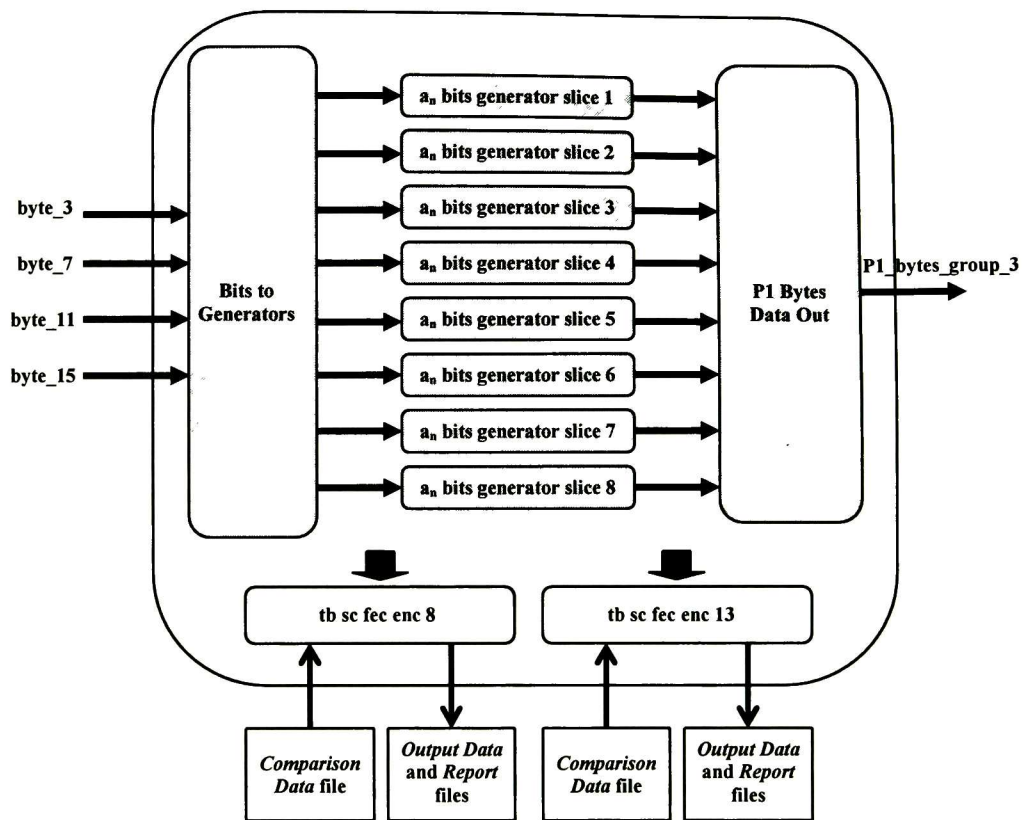


Figura C.5 Esquema de verificación en el interior del bloque *Calculador de bytes P1 para el grupo 3*

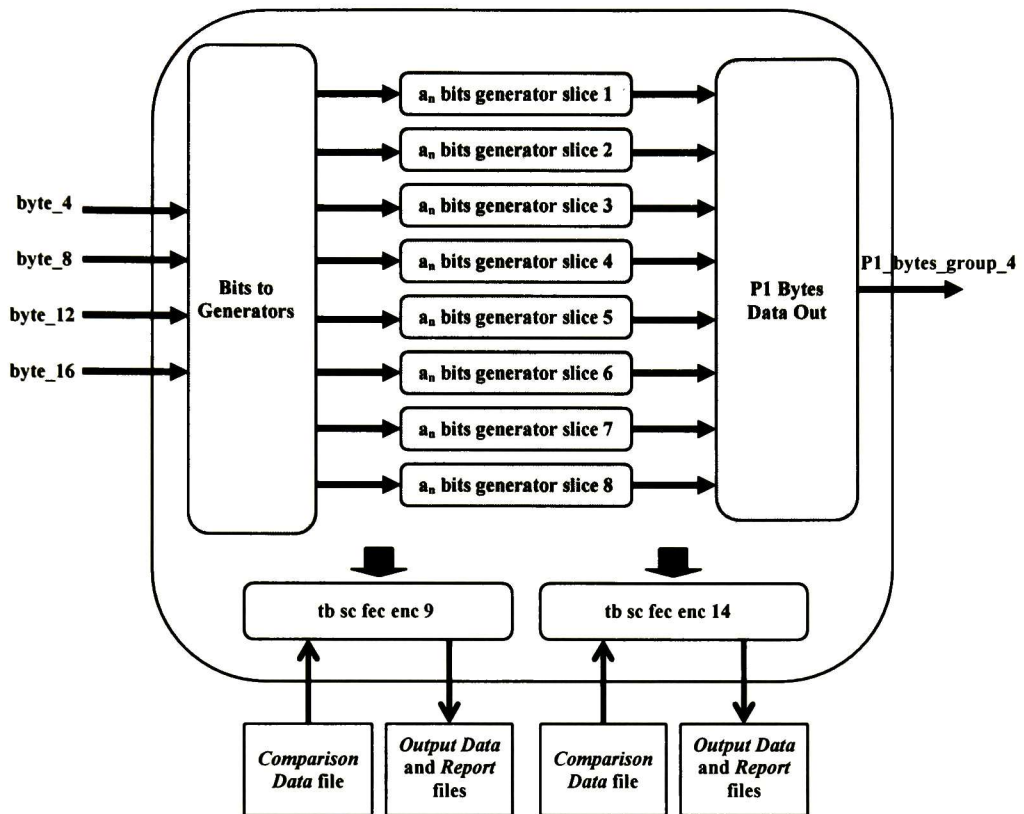


Figura C.6 Esquema de verificación en el interior del bloque *Calculador de bytes P1 para el grupo 4*

Archivos de comparación:

Las entidades de verificación harán uso de archivos de comparación que servirán para analizar los flujos de datos en los bloques internos. Estos archivos de comparación serán generados por el módulo escrito en lenguaje de descripción de hardware, en el archivo *C_data_generator_tc_fec_enc_03.vhd*. Este módulo es un emulador del comportamiento del bloque *Calculador de bytes P1* y a partir de los archivos CSV de entrada *I_data_tc_fec_enc_03_STM16.CSV* e *I_data_tc_fec_enc_03_STM64.CSV* genera los archivos de comparación para los subcasos de prueba del 5 al 19.

A continuación se describen los subcasos de prueba.

C.3.1 Subcaso de prueba SC-FEC-ENC-05

Descripción:

Verificar que a partir del flujo entrante SONET / SDH, el cual tiene un ancho de bus de 128 bits, el bloque *bits to generators* forme correctamente los 8 flujos de bits de información

que corresponderán a las 8 subfilas del *grupo 1*. Lo anterior para cuando el módulo codificador se encuentre configurado en el modo de 2.5 Gbps (STS-48 / STM-16).

Requerimientos a verificar:

- R-FEC-ENC-05
- R-FEC-ENC-06
- R-FEC-ENC-07

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_05and06.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-06.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
<i>data_with_zeros</i> [127:0]	Flujo de entrada SONET/SDH (<i>I_data_tc_fec_enc_03_STM16.CSV</i>)
<i>J0_detected_with_zeros</i>	Señal de indicación de J0
<i>fec_encoder_clk_in</i>	Reloj del sistema (19.44 MHz)
<i>P1_bytes_calculator_reset</i>	0
<i>fec_encoder_config</i>	0 (STM-16)

Tabla C.5 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-05

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_tc_fec_enc_03_STM16.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-48 / STM-16 con contenido aleatorio.

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_05.CSV*. Este archivo contiene campos de texto separados por el caracter coma y están formados por cuatro caracteres que representan a 16 bits en formato hexadecimal (0005, FE21, 78AE, etc.). Cada línea del archivo contiene los datos de comparación para una determinada fila de una trama SONET / SDH. Los datos de comparación se encuentran entrelazados, esto es, los primeros ocho campos contienen los datos para verificar los valores presentes en los flujos de las subfilas durante el primer de ciclos de reloj de la fila SONET / SDH (una fila dura 270 ciclos de reloj en el modo de 2.5 Gbps), los siguientes ocho campos contienen los datos de comparación para el segundo ciclo de reloj y así sucesivamente hasta llegar al último ciclo de reloj de la fila. En la Tabla C.6 se muestra el formato de este archivo.

		subfila												
		1	2	...	8	1	2	...	8	...	1	2	...	8
L í n e a	1	0000,	045F,	...	A876,	2309,	3490,	...	65E8,	...	1468,	5D91,	...	6CF4,
	2	2923,	2B2F,	...	0719,	5B38,	1EF8,	...	6370,	...	934B,	0453,	...	4AE4,
	3	ABD5,	DCFF,	...	AC94,	03DF,	FABD,	...	49E0,	...	C248,	0B2C,	...	DE80,
	4	11FC,	C1F3,	...	B23B,	F787,	5D48,	...	0B80,	...	3D8F,	9992,	...	3D8B,
	⋮	⋮	⋮	...	⋮	⋮	⋮	...	⋮	...	⋮	⋮	...	⋮
	N	B4AA,	1BE8,	...	3FE9,	F0F1,	0BB0,	...	E728,	...	6CF4,	934B,	...	A65E,

Tabla C.6 Formato del archivo de comparación *C_data_sc_fec_enc_05.CSV*

Respuestas:

Los 8 flujos verificados son escritos en el archivo *O_data_sc_fec_enc_05.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_05.CSV* se escriben los errores detectados durante la verificación de los flujos.

C.3.2 Subcaso de prueba SC-FEC-ENC-06

Descripción:

Verificar que a partir del flujo entrante SONET / SDH, el cual tiene un ancho de bus de 128 bits, el bloque *bits to generators* forme correctamente los 8 flujos de bits de información que corresponderán a las 8 subfilas del *grupo 1*. Lo anterior para cuando el módulo codificador se encuentre configurado en el modo de 10 Gbps (STS-192 / STM-64).

Requerimientos a verificar:

- R-FEC-ENC-05
- R-FEC-ENC-06
- R-FEC-ENC-07

Camara de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_05and06.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-05.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
<code>data_with_zeros[127:0]</code>	Flujo de entrada SONET/SDH (<i>I_data_tc_fec_enc_03_STM64.CSV</i>)
<code>J0_detected_with_zeros</code>	Señal de indicación de J0
<code>fec_encoder_clk_in</code>	Reloj del sistema (77.76 MHz)
<code>PI_bytes_calculator_reset</code>	0

fec_encoder_config	1 (STM-64)
--------------------	------------

Tabla C.7 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-06

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_tc_fec_enc_03_STM64.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio.

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_06.CSV*. Cada campo de texto, a diferencia del archivo de comparación del subcaso SC-FEC-ENC-05, sólo contiene un caracter y representa a 4 bits en formato hexadecimal (5, C, F, 9, etc.).

Respuestas:

Los 8 flujos verificados son escritos en el archivo *O_data_sc_fec_enc_06.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_06.CSV* se escriben los errores detectados durante la verificación de los flujos.

C.3.3 Subcaso de prueba SC-FEC-ENC-07

Descripción:

En el modo de operación 10 Gbps (STS-192 / STM-64), verificar que, a partir del flujo entrante SONET / SDH de 128 bits, el bloque *bits to generators* forme correctamente los 8 flujos de bits de información que corresponderán a las 8 subfilas del *grupo 2*.

Requerimientos a verificar:

- R-FEC-ENC-05
- R-FEC-ENC-06
- R-FEC-ENC-07

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_07.vhd*.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_with_zeros[127:0]	Flujo de entrada SONET/SDH

	<i>(I_data_tc_fec_enc_03_STM64.CSV)</i>
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.8 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-07

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_tc_fec_enc_03_STM64.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio.

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_07.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-06.

Respuestas:

Los 8 flujos verificados son escritos en el archivo *O_data_sc_fec_enc_07.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo *O_asserts_sc_fec_enc_07.CSV* se escriben los errores detectados durante la verificación de los flujos.

C.3.4 Subcaso de prueba SC-FEC-ENC-08

Descripción:

En el modo de operación 10 Gbps (STS-192 / STM-64), verificar que, a partir del flujo entrante SONET / SDH de 128 bits, el bloque *bits to generators* forme correctamente los 8 flujos de bits de información que corresponderán a las 8 subfilas del grupo 3.

Requerimientos a verificar:

- R-FEC-ENC-05
- R-FEC-ENC-06
- R-FEC-ENC-07

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_08.vhd*.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_with_zeros[127:0]	Flujo de entrada SONET/SDH (<i>I_data_tc_fec_enc_03_STM64.CSV</i>)
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.9 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-08

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_tc_fec_enc_03_STM64.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio.

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_08.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-06.

Respuestas:

Los 8 flujos verificados son escritos en el archivo *O_data_sc_fec_enc_08.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo *O_asserts_sc_fec_enc_08.CSV* se escriben los errores detectados durante la verificación de los flujos.

C.3.5 Subcaso de prueba SC-FEC-ENC-09

Descripción:

En el modo de operación 10 Gbps (STS-192 / STM-64), verificar que, a partir del flujo entrante SONET / SDH de 128 bits, el bloque *bits to generators* forme correctamente los 8 flujos de bits de información que corresponderán a las 8 subfilas del grupo 4.

Requerimientos a verificar:

- R-FEC-ENC-05
- R-FEC-ENC-06
- R-FEC-ENC-07

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_09.vhd*.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_with_zeros[127:0]	Flujo de entrada SONET/SDH (<i>I_data_tc_fec_enc_03_STM64.CSV</i>)
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.10 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-09

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_tc_fec_enc_03_STM64.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio.

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_09.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-06.

Respuestas:

Los 8 flujos verificados son escritos en el archivo *O_data_sc_fec_enc_09.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo *O_asserts_sc_fec_enc_09.CSV* se escriben los errores detectados durante la verificación de los flujos.

C.3.6 Subcaso de prueba SC-FEC-ENC-10

Descripción:

Verificar que los bloques *Generadores de bits a_n para el grupo 1* realicen correctamente el cálculo de los bits de paridad a_n ($0 \leq n \leq 38$) sobre los 8 flujos de bits de información entrantes correspondientes a las 8 subfilas del *grupo 1*. Lo anterior para cuando el módulo codificador se encuentre configurado en el modo de 2.5 Gbps (STS-48 / STM-16).

Requerimientos a verificar:

- R-FEC-ENC-02

- R-FEC-ENC-03
- R-FEC-ENC-08

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_10and11.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-11.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_to_generator_slice_1[15:0] data_to_generator_slice_2[15:0] data_to_generator_slice_3[15:0] data_to_generator_slice_4[15:0] data_to_generator_slice_5[15:0] data_to_generator_slice_6[15:0] data_to_generator_slice_7[15:0] data_to_generator_slice_8[15:0]	Flujos proporcionados por un bloque previo
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	0 (STM-16)

Tabla C.11 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-10

Estímulos:

Flujos de bits que son proporcionados por el bloque *bits to generators* (ver subcaso de prueba SC-FEC-ENC-05).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_10.CSV*. Este archivo contiene campos de texto separados por el caracter coma y compuestos por 39 dígitos binarios (0 o 1) que representan a 39 bits ordenados del más significativo al menos significativo. Cada línea del archivo contiene 8 campos de texto que representan los datos de comparación para una determinada fila de una trama SONET / SDH. El primer campo de texto de la línea contiene el dato de comparación que permite verificar el cálculo de los bits a_n que se realiza sobre la subfila 1, y así hasta llegar al octavo campo de texto que contiene el dato para verificar el cálculo sobre la subfila 8.

Respuestas:

Los bits a_n calculados por los bloques *Generadores de bits a_n para el grupo 1* son escritos en el archivo *O_data_sc_fec_enc_10.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_10.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3.7 Subcaso de prueba SC-FEC-ENC-11

Descripción:

Verificar que los bloques *Generadores de bits a_n para el grupo 1* realicen correctamente el cálculo de los bits de paridad a_n ($0 \leq n \leq 38$) sobre los 8 flujos de bits de información entrantes correspondientes a las 8 subfilas del *grupo 1*. Lo anterior para cuando el módulo codificador se encuentre configurado en el modo de 10 Gbps (STS-192 / STM-64).

Requerimientos a verificar:

- R-FEC-ENC-02
- R-FEC-ENC-03
- R-FEC-ENC-08

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_10and11.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-10.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_to_generator_slice_1[15:0] data_to_generator_slice_2[15:0] data_to_generator_slice_3[15:0] data_to_generator_slice_4[15:0] data_to_generator_slice_5[15:0] data_to_generator_slice_6[15:0] data_to_generator_slice_7[15:0] data_to_generator_slice_8[15:0]	Flujos proporcionados por un bloque previo
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.12 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-11

Estímulos:

Flujos de bits que son proporcionados por el bloque *bits to generators* (ver subcaso de prueba SC-FEC-ENC-06).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_11.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-10.

Respuestas:

Los bits a_n calculados por los bloques *Generadores de bits a_n para el grupo 1* son escritos en el archivo *O_data_sc_fec_enc_11.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_11.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3.8 Subcaso de prueba SC-FEC-ENC-12

Descripción:

En el modo de operación 10 Gbps (STS-192 / STM-64), verificar que los bloques *Generadores de bits a_n para el grupo 2* realicen correctamente el cálculo de los bits de paridad a_n ($0 \leq n \leq 38$) sobre los 8 flujos de bits de información entrantes correspondientes a las 8 subfilas del *grupo 2*.

Requerimientos a verificar:

- R-FEC-ENC-02
- R-FEC-ENC-03
- R-FEC-ENC-08

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_12.vhd*.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_to_generator_slice_1[3:0] data_to_generator_slice_2[3:0] data_to_generator_slice_3[3:0] data_to_generator_slice_4[3:0] data_to_generator_slice_5[3:0] data_to_generator_slice_6[3:0] data_to_generator_slice_7[3:0] data_to_generator_slice_8[3:0]	Flujos proporcionados por un bloque previo
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)

PI bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.13 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-12

Estímulos:

Flujos de bits que son proporcionados por el bloque *bits to generators* (ver subcaso de prueba SC-FEC-ENC-07).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_12.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-10.

Respuestas:

Los bits a_n calculados por los bloques *Generadores de bits a_n para el grupo 2* son escritos en el archivo *O_data_sc_fec_enc_12.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_12.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3.9 Subcaso de prueba SC-FEC-ENC-13

Descripción:

En el modo de operación 10 Gbps (STS-192 / STM-64), verificar que los bloques *Generadores de bits a_n para el grupo 3* realicen correctamente el cálculo de los bits de paridad a_n ($0 \leq n \leq 38$) sobre los 8 flujos de bits de información entrantes correspondientes a las 8 subfilas del *grupo 3*.

Requerimientos a verificar:

- R-FEC-ENC-02
- R-FEC-ENC-03
- R-FEC-ENC-08

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_13.vhd*.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_to_generator_slice_1[3:0] data_to_generator_slice_2[3:0] data_to_generator_slice_3[3:0] data_to_generator_slice_4[3:0] data_to_generator_slice_5[3:0] data_to_generator_slice_6[3:0] data_to_generator_slice_7[3:0] data_to_generator_slice_8[3:0]	Flujos proporcionados por un bloque previo
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.14 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-13

Estímulos:

Flujos de bits que son proporcionados por el bloque *bits to generators* (ver subcaso de prueba SC-FEC-ENC-08).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_13.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-10.

Respuestas:

Los bits a_n calculados por los bloques *Generadores de bits a_n para el grupo 3* son escritos en el archivo *O_data_sc_fec_enc_13.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_13.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3.10 Subcaso de prueba SC-FEC-ENC-14

Descripción:

En el modo de operación 10 Gbps (STS-192 / STM-64), verificar que los bloques *Generadores de bits a_n para el grupo 4* realicen correctamente el cálculo de los bits de paridad a_n ($0 \leq n \leq 38$) sobre los 8 flujos de bits de información entrantes correspondientes a las 8 subfilas del *grupo 4*.

Requerimientos a verificar:

- R-FEC-ENC-02
- R-FEC-ENC-03
- R-FEC-ENC-08

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_14.vhd*.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_to_generator_slice_1[3:0] data_to_generator_slice_2[3:0] data_to_generator_slice_3[3:0] data_to_generator_slice_4[3:0] data_to_generator_slice_5[3:0] data_to_generator_slice_6[3:0] data_to_generator_slice_7[3:0] data_to_generator_slice_8[3:0]	Flujos proporcionados por un bloque previo
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.15 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-14

Estímulos:

Flujos de bits que son proporcionados por el bloque *bits to generators* (ver subcaso de prueba SC-FEC-ENC-09).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_14.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-10.

Respuestas:

Los bits a_n calculados por los bloques *Generadores de bits a_n para el grupo 4* son escritos en el archivo *O_data_sc_fec_enc_14.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_14.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3.11 Subcaso de prueba SC-FEC-ENC-15

Descripción:

El presente subcaso de prueba tiene como objetivos:

- Verificar que el bloque *Calculador de bytes P1 para el grupo 1* forme correctamente los bytes P1 y los envíe en el tiempo y en el orden adecuado hacia los bloques *Multiplexor para el grupo 1* y *Registro de bytes P1 para el grupo 1* a través del puerto *P1_bytes_group_1*. Los bytes P1 son formados a partir de los bits a_n calculados por los bloques internos *Generadores de bits a_n para el grupo 1* (ver subcaso de prueba SC-FEC-ENC-10).
- Verificar que el bloque *Registro de bytes P1 para el grupo 1* realice el almacenamiento de los bytes P1 para las filas 1, 2 y 4, y su posterior envío en el tiempo y en el orden adecuado hacia el bloque *Multiplexor para el grupo 1*.
- Verificar que cuando el bloque *Multiplexor para el grupo 1* reciba el flujo de los bytes P1 del grupo 1, los envíe hacia el bloque *Multiplexor de grupos*. El bloque *Multiplexor para el grupo 1* recibe los bytes P1 de las filas 3, 5, 6, 7, 8 y 9 desde el bloque *Calculador de bytes P1 para el grupo 1*, y recibe los bytes P1 de las filas 1, 2 y 4 desde el bloque *Registro de bytes P1 para el grupo 1*.
- Verificar que cuando el bloque *Multiplexor de grupos* reciba el flujo de los bytes P1 del grupo 1 desde el bloque *Multiplexor para el grupo 1*, estos sean enviados por el puerto de salida *P1_bytes_to_insert*.

Lo anterior para cuando el módulo codificador se encuentre configurado en el modo de 2.5 Gbps (STS-48 / STM-16).

Requerimientos a verificar:

- R-FEC-ENC-08
- R-FEC-ENC-09

Camara de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_15.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-16.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
an_bits_slice_1[38:0] an_bits_slice_2[38:0] an_bits_slice_3[38:0] an_bits_slice_4[38:0] an_bits_slice_5[38:0] an_bits_slice_6[38:0] an_bits_slice_7[38:0] an_bits_slice_8[38:0]	Flujos internos del <i>Calculador de bytes P1</i> para el grupo 1
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)

P1 bytes calculator reset	0
fec_encoder_config	0 (STM-16)

Tabla C.16 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-15

Estímulos:

Flujos de bits internos del *Calculador de bytes P1 para el grupo 1* (ver subcaso de prueba SC-FEC-ENC-10).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_15.CSV*. Este archivo contiene campos de texto separados por el caracter coma y compuestos por 26 caracteres que representan a 13 bytes en formato hexadecimal. Cada línea del archivo contiene 3 campos de texto que representan a los bytes P1 calculados para una determinada fila de una trama SONET / SDH. El primer campo de la línea contiene el primer subconjunto de bytes P1 ordenado del byte P1(a₃₈) al P1(a₂₆), el segundo campo contiene el segundo subconjunto de bytes P1 ordenado del byte P1(a₂₅) al P1(a₁₃) y el tercer campo contiene el tercer subconjunto de bytes P1 ordenado del byte P1(a₁₂) al P1(a₀).

Respuestas:

Los bytes P1 que forma y envía el bloque *Calculador de bytes P1 para el grupo 1* son escritos en el archivo *O_data_sc_fec_enc_15.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_15.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3.12 Subcaso de prueba SC-FEC-ENC-16

Descripción:

El presente subcaso de prueba tiene como objetivos:

- Verificar que el bloque *Calculador de bytes P1 para el grupo 1* forme correctamente los bytes P1 y los envíe en el tiempo y en el orden adecuado hacia los bloques *Multiplexor para el grupo 1* y *Registro de bytes P1 para el grupo 1* a través del puerto *P1_bytes_group_1*. Los bytes P1 son formados a partir de los bits a_n calculados por los bloques internos *Generadores de bits a_n para el grupo 1* (ver subcaso de prueba SC-FEC-ENC-11).
- Verificar que el bloque *Registro de bytes P1 para el grupo 1* realice el almacenamiento de los bytes P1 para las filas 1, 2 y 4, y su posterior envío en el tiempo y en el orden adecuado hacia el bloque *Multiplexor para el grupo 1*.

- Verificar que cuando el bloque *Multiplexor para el grupo 1* reciba el flujo de los bytes P1 del *grupo 1*, los envíe hacia el bloque *Multiplexor de grupos*. El bloque *Multiplexor para el grupo 1* recibe los bytes P1 de las filas 3, 5, 6, 7, 8 y 9 desde el bloque *Calculador de bytes P1 para el grupo 1*, y recibe los bytes P1 de las filas 1, 2 y 4 desde el bloque *Registro de bytes P1 para el grupo 1*.
- Verificar que cuando el bloque *Multiplexor de grupos* reciba el flujo de los bytes P1 del *grupo 1* desde el bloque *Multiplexor para el grupo 1*, estos sean enviados por el puerto de salida *P1_bytes_to_insert*.

Lo anterior para cuando el módulo codificador se encuentre configurado en el modo de 10 Gbps (STS-192 / STM-64).

Requerimientos a verificar:

- R-FEC-ENC-08
- R-FEC-ENC-09

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_16.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-15.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
an_bits_slice_1[38:0] an_bits_slice_2[38:0] an_bits_slice_3[38:0] an_bits_slice_4[38:0] an_bits_slice_5[38:0] an_bits_slice_6[38:0] an_bits_slice_7[38:0] an_bits_slice_8[38:0]	Flujos internos del <i>Calculador de bytes P1</i> para el grupo 1
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.17 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-16

Estímulos:

Flujos de bits internos del *Calculador de bytes P1 para el grupo 1* (ver subcaso de prueba SC-FEC-ENC-11).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_16.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-15.

Respuestas:

Los bytes P1 que forma y envía el bloque *Calculador de bytes P1 para el grupo 1* son escritos en el archivo *O_data_sc_fec_enc_16.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_16.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3.13 Subcaso de prueba SC-FEC-ENC-17

Descripción:

El presente subcaso de prueba aplica cuando el módulo codificador se encuentra configurado en el modo de 10 Gbps (STS-192 / STM-64) y tiene como objetivos:

- Verificar que el bloque *Calculador de bytes P1 para el grupo 2* forme correctamente los bytes P1 y los envíe en el tiempo y en el orden adecuado hacia los bloques *Multiplexor para el grupo 2* y *Registro de bytes P1 para el grupo 2* a través del puerto *P1_bytes_group_2*. Los bytes P1 son formados a partir de los bits a_n calculados por los bloques internos *Generadores de bits a_n para el grupo 2* (ver subcaso de prueba SC-FEC-ENC-12).
- Verificar que el bloque *Registro de bytes P1 para el grupo 2* realice el almacenamiento de los bytes P1 para las filas 1, 2 y 4, y su posterior envío en el tiempo y en el orden adecuado hacia el bloque *Multiplexor para el grupo 2*.
- Verificar que cuando el bloque *Multiplexor para el grupo 2* reciba el flujo de los bytes P1 del *grupo 2*, los envíe hacia el bloque *Multiplexor de grupos*. El bloque *Multiplexor para el grupo 2* recibe los bytes P1 de las filas 3, 5, 6, 7, 8 y 9 desde el bloque *Calculador de bytes P1 para el grupo 2*, y recibe los bytes P1 de las filas 1, 2 y 4 desde el bloque *Registro de bytes P1 para el grupo 2*.
- Verificar que cuando el bloque *Multiplexor de grupos* reciba el flujo de los bytes P1 del *grupo 2* desde el bloque *Multiplexor para el grupo 2*, estos sean enviados por el puerto de salida *P1_bytes_to_insert*.

Requerimientos a verificar:

- R-FEC-ENC-08
- R-FEC-ENC-09

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_17.vhd*.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
an_bits_slice_1[38:0] an_bits_slice_2[38:0] an_bits_slice_3[38:0] an_bits_slice_4[38:0] an_bits_slice_5[38:0] an_bits_slice_6[38:0] an_bits_slice_7[38:0] an_bits_slice_8[38:0]	Flujos internos del <i>Calculador de bytes P1</i> para el grupo 2
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.18 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-17

Estímulos:

Flujos de bits internos del *Calculador de bytes P1 para el grupo 2* (ver subcaso de prueba SC-FEC-ENC-12).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_17.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-15.

Respuestas:

Los bytes P1 que forma y envía el bloque *Calculador de bytes P1 para el grupo 2* son escritos en el archivo *O_data_sc_fec_enc_17.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_17.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3.14 Subcaso de prueba SC-FEC-ENC-18

Descripción:

El presente subcaso de prueba aplica cuando el módulo codificador se encuentra configurado en el modo de 10 Gbps (STS-192 / STM-64) y tiene como objetivos:

- Verificar que el bloque *Calculador de bytes P1 para el grupo 3* forme correctamente los bytes P1 y los envíe en el tiempo y en el orden adecuado hacia los bloques *Multiplexor para el grupo 3* y *Registro de bytes P1 para el grupo 3* a través del puerto *P1_bytes_group_3*. Los bytes P1 son formados a partir de los bits a_n

calculados por los bloques internos *Generadores de bits a_n para el grupo 3* (ver subcaso de prueba SC-FEC-ENC-13).

- Verificar que el bloque *Registro de bytes P1 para el grupo 3* realice el almacenamiento de los bytes P1 para las filas 1, 2 y 4, y su posterior envío en el tiempo y en el orden adecuado hacia el bloque *Multiplexor para el grupo 3*.
- Verificar que cuando el bloque *Multiplexor para el grupo 3* reciba el flujo de los bytes P1 del grupo 3, los envíe hacia el bloque *Multiplexor de grupos*. El bloque *Multiplexor para el grupo 3* recibe los bytes P1 de las filas 3, 5, 6, 7, 8 y 9 desde el bloque *Calculador de bytes P1 para el grupo 3*, y recibe los bytes P1 de las filas 1, 2 y 4 desde el bloque *Registro de bytes P1 para el grupo 3*.
- Verificar que cuando el bloque *Multiplexor de grupos* reciba el flujo de los bytes P1 del grupo 3 desde el bloque *Multiplexor para el grupo 3*, estos sean enviados por el puerto de salida *P1_bytes_to_insert*.

Requerimientos a verificar:

- R-FEC-ENC-08
- R-FEC-ENC-09

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_18.vhd*.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
an_bits_slice_1[38:0] an_bits_slice_2[38:0] an_bits_slice_3[38:0] an_bits_slice_4[38:0] an_bits_slice_5[38:0] an_bits_slice_6[38:0] an_bits_slice_7[38:0] an_bits_slice_8[38:0]	Flujos internos del <i>Calculador de bytes P1</i> para el grupo 3
J0_detected_with_zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.19 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-18

Estímulos:

Flujos de bits internos del *Calculador de bytes P1 para el grupo 3* (ver subcaso de prueba SC-FEC-ENC-13).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_18.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-15.

Respuestas:

Los bytes P1 que forma y envía el bloque *Calculador de bytes P1 para el grupo 3* son escritos en el archivo *O_data_sc_fec_enc_18.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_18.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.3.15 Subcaso de prueba SC-FEC-ENC-19

Descripción:

El presente subcaso de prueba aplica cuando el módulo codificador se encuentra configurado en el modo de 10 Gbps (STS-192 / STM-64) y tiene como objetivos:

- Verificar que el bloque *Calculador de bytes P1 para el grupo 4* forme correctamente los bytes P1 y los envíe en el tiempo y en el orden adecuado hacia los bloques *Multiplexor para el grupo 4* y *Registro de bytes P1 para el grupo 4* a través del puerto *P1_bytes_group_4*. Los bytes P1 son formados a partir de los bits a_n calculados por los bloques internos *Generadores de bits a_n para el grupo 4* (ver subcaso de prueba SC-FEC-ENC-14).
- Verificar que el bloque *Registro de bytes P1 para el grupo 4* realice el almacenamiento de los bytes P1 para las filas 1, 2 y 4, y su posterior envío en el tiempo y en el orden adecuado hacia el bloque *Multiplexor para el grupo 4*.
- Verificar que cuando el bloque *Multiplexor para el grupo 4* reciba el flujo de los bytes P1 del *grupo 4*, los envíe hacia el bloque *Multiplexor de grupos*. El bloque *Multiplexor para el grupo 4* recibe los bytes P1 de las filas 3, 5, 6, 7, 8 y 9 desde el bloque *Calculador de bytes P1 para el grupo 4*, y recibe los bytes P1 de las filas 1, 2 y 4 desde el bloque *Registro de bytes P1 para el grupo 4*.
- Verificar que cuando el bloque *Multiplexor de grupos* reciba el flujo de los bytes P1 del *grupo 4* desde el bloque *Multiplexor para el grupo 4*, estos sean enviados por el puerto de salida *P1_bytes_to_insert*.

Requerimientos a verificar:

- R-FEC-ENC-08
- R-FEC-ENC-09

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_19.vhd*.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
an_bits_slice_1[38:0]	Flujos internos del <i>Calculador de bytes P1</i> para el grupo 4
an_bits_slice_2[38:0]	
an_bits_slice_3[38:0]	
an_bits_slice_4[38:0]	
an_bits_slice_5[38:0]	
an_bits_slice_6[38:0]	
an_bits_slice_7[38:0]	
an_bits_slice_8[38:0]	
J0 detected with zeros	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1 bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.20 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-19

Estímulos:

Flujos de bits internos del *Calculador de bytes P1 para el grupo 4* (ver subcaso de prueba SC-FEC-ENC-14).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_19.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-15.

Respuestas:

Los bytes P1 que forma y envía el bloque *Calculador de bytes P1 para el grupo 4* son escritos en el archivo *O_data_sc_fec_enc_19.CSV* siguiendo el formato del archivo de comparación del presente subcaso de prueba.

En el archivo de reporte *O_asserts_sc_fec_enc_19.CSV* se escriben los errores detectados durante la verificación funcional de los bloques, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.4 Caso de prueba TC-FEC-ENC-04

Descripción general:

El presente caso de prueba verifica el correcto funcionamiento de los bloques *Insertor de bytes P1* y *Buffer de datos*. El presente caso de prueba tiene como objetivos:

- Verificar que el bloque *Insertor de bytes P1* realice la inserción de los bytes P1, que recibe del bloque *Calculador de bytes P1*, sobre el flujo entrante SONET / SDH en las posiciones de inserción que describe el requerimiento R-FEC-ENC-09. Lo anterior para cuando se encuentre habilitada la inserción de los bytes P1.

- Verificar que el bloque *Insertor de bytes P1* opere de manera *transparente* cuando se encuentre deshabilitada la inserción de los bytes P1, es decir, se debe verificar que el flujo saliente SONET / SDH sea igual al flujo entrante.
- Verificar que cuando se encuentre habilitada la generación de la señal de indicación de alarma MSF-AIS, el bloque *Insertor de bytes P1* inserte el valor 0xFF en todas las posiciones de inserción de bytes P1 en la tara de sección (SOH) / tara de sección de regeneración (RSOH) del flujo entrante SONET / SDH.
- Verificar que el bloque *Buffer de datos* introduzca el retardo en el flujo de datos SONET / SDH y en la señal *J0_detected*, en número de ciclos de reloj, para el cual fue diseñado.

Requerimientos a verificar:

Los requerimientos que se verifican con el caso de prueba TC-FEC-ENC-04, son los siguientes:

- R-FEC-ENC-01
- R-FEC-ENC-09
- R-FEC-ENC-20
- R-FEC-ENC-21
- R-FEC-ENC-22

Subcasos de Prueba:

El caso de prueba TC-FEC-ENC-04 está dividido en los siguientes subcasos de prueba:

- SC-FEC-ENC-20
- SC-FEC-ENC-21
- SC-FEC-ENC-22
- SC-FEC-ENC-23

Esquema de verificación:

El esquema de verificación para el presente caso de prueba se muestra en la Figura C.7. En este esquema se incluyen los bloques *Insertor de ceros* y *Calculador de bytes P1*, que se verifican en los casos de prueba TC-FEC-ENC-02 y TC-FEC-ENC-03, y que al funcionar conjuntamente con los bloques *Insertor de bytes P1* y *Buffer de datos* realizan el cálculo e inserción de los bytes P1 sobre el flujo entrante SONET / SDH.

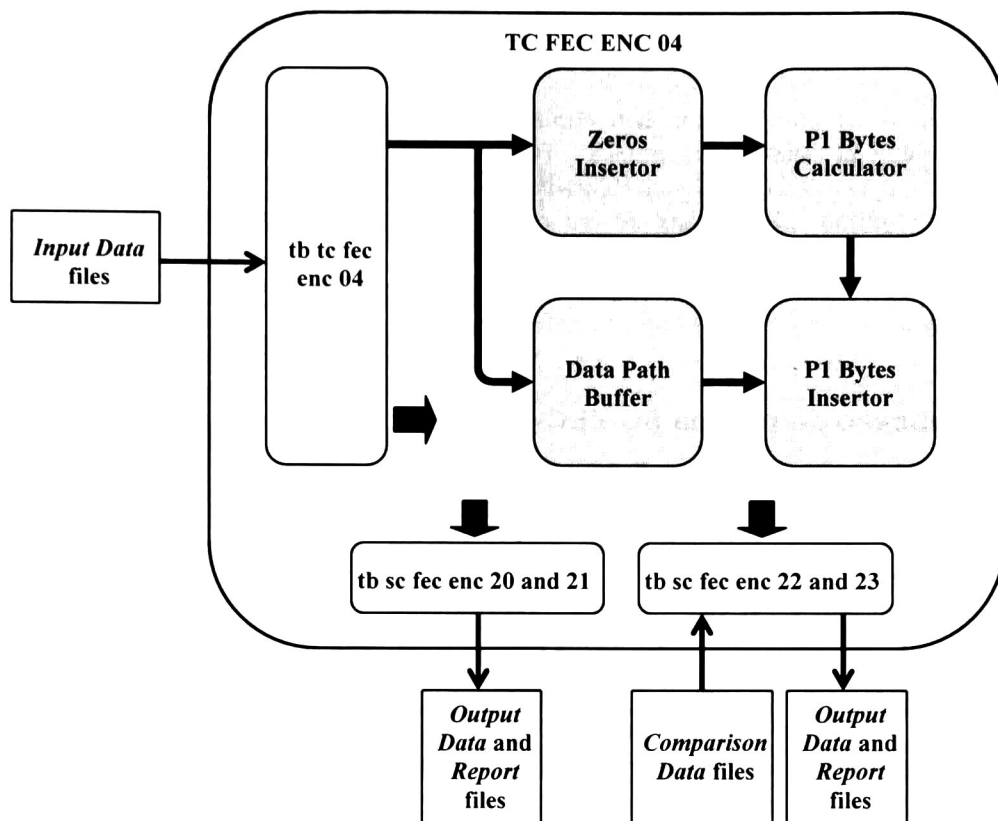


Figura C.7 Esquema de verificación para el caso de prueba TC-FEC-ENC-04

La entidad *tb_tc_fec_enc_04.vhd* se encarga de generar los flujos de entrada SONET / SDH a partir de los archivos CSV de entrada: *I_data_tc_fec_enc_04_STM16.CSV* e *I_data_tc_fec_enc_04_STM64.CSV*. Además, genera las señales de configuración, reset, habilitación y de reloj para los demás bloques.

Los archivos CSV de entrada son copias de los archivos utilizados en el caso de prueba TC-FEC-ENC-01: *I_data_sc_fec_enc_01.CSV* (STM-16) e *I_data_sc_fec_enc_02.CSV* (STM-64).

Los subcasos de prueba SC-FEC-ENC-20 y SC-FEC-ENC-21 verifican el funcionamiento del bloque *Buffer de datos* a 2.5 Gbps y 10 Gbps, respectivamente.

Los subcasos de prueba SC-FEC-ENC-22 y SC-FEC-ENC-23 verifican el funcionamiento del bloque *Insertor de bytes P1* a 2.5 Gbps y 10 Gbps, respectivamente.

Los subcasos de prueba SC-FEC-ENC-22 y SC-FEC-ENC-23 hacen uso de los archivos de comparación *C_data_sc_fec_enc_22.CSV*, *C_data_sc_fec_enc_23_group1.CSV*, *C_data_sc_fec_enc_23_group2.CSV*, *C_data_sc_fec_enc_23_group3.CSV* y *C_data_sc_fec_enc_23_group4.CSV*.

23_group4.CSV los cuales contienen los valores de bytes P1 que se espera inserte el bloque *Insertor de bytes P1* cuando se encuentre habilitada la inserción de bytes P1.

Estos archivos de comparación son copias de los siguientes archivos de comparación utilizados en el caso de prueba TC-FEC-ENC-03: *C_data_sc_fec_enc_15.CSV*, *C_data_sc_fec_enc_16.CSV* (bytes P1 para el grupo 1), *C_data_sc_fec_enc_17.CSV* (bytes P1 para el grupo 2), *C_data_sc_fec_enc_18.CSV* (bytes P1 para el grupo 3) y *C_data_sc_fec_enc_19.CSV* (bytes P1 para el grupo 4).

A continuación se describen los subcasos de prueba.

C.4.1 Subcaso de prueba SC-FEC-ENC-20

Descripción:

Verificar que el bloque *Buffer de datos* introduzca un retardo de 274 ciclos de reloj en el flujo SONET / SDH y en la señal *J0_detected*, cuando el bloque se encuentra configurado en el modo de 2.5 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_20and21.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-21.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_B2_bytes_compensated[127:0]	Flujo de entrada SONET/SDH (<i>I_data_tc_fec_enc_04_STM16.CSV</i>)
J0_detected_compensator	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)
data_path_buffer_reset	0
fec_encoder_config	0 (STM-16)

Tabla C.21 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-20

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_tc_fec_enc_04_STM16.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-48 / STM-16 con contenido aleatorio.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_20.CSV*.

En el archivo de reporte *O_asserts_sc_fec_enc_20.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.4.2 Subcaso de prueba SC-FEC-ENC-21

Descripción:

Verificar que el bloque *Buffer de datos* introduzca un retardo de 1084 ciclos de reloj en el flujo SONET / SDH y en la señal *J0_detected*, cuando el bloque se encuentra configurado en el modo de 10 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_20and21.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-20.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_B2_bytes_compensated[127:0]	Flujo de entrada SONET/SDH (<i>I_data_tc_fec_enc_04_STM64.CSV</i>)
J0_detected_compensator	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
data_path_buffer_reset	0
fec_encoder_config	1 (STM-64)

Tabla C.22 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-21

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_tc_fec_enc_04_STM64.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_21.CSV*.

En el archivo *O_asserts_sc_fec_enc_21.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.4.3 Subcaso de prueba SC-FEC-ENC-22

Descripción:

Verificar el correcto funcionamiento del bloque *Insertor de bytes P1*, cuando se encuentra configurado en el modo de 2.5 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_22and23.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-23.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_with_delay[127:0]	Flujo proveniente del bloque <i>Buffer de datos</i>
J0_detected_with_delay	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)
P1_bytes_insertor_reset	0
fec_encoder_config	0 (STM-16)
P1_bytes_ins_enable	1 (tramas 1 a 10) 0 (tramas 11 a 30)
P1_bytes_insertor_msf_ais	0 (tramas 1 a 20) 1 (tramas 21 a 30)

Tabla C.23 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-22

Estímulos:

El flujo de entrada al bloque *Insertor de bytes P1* es el flujo de salida del bloque *Buffer de datos* (ver subcaso de prueba SC-FEC-ENC-20).

Archivos de comparación:

El archivo de comparación para el presente subcaso de prueba es *C_data_sc_fec_enc_22.CSV*. El formato de este archivo es idéntico al del subcaso SC-FEC-ENC-15 y contiene los datos contra los cuales se compararán los bytes P1 del flujo de salida.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_22.CSV*.

En el archivo de reporte *O_asserts_sc_fec_enc_22.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: en las tramas 1 a 10 se deberá insertar correctamente los bytes P1 que envía el bloque *Calculador de bytes P1*; en las tramas 11 a 20 el bloque deberá actuar de manera *transparente*; en las tramas 21 a 30 se deberá insertar el valor 0xFF en todos los bytes P1 de la tara de sección (SOH) / tara de sección de regeneración (RSOH), ya que se encuentra habilitada la generación de la señal de indicación de alarma MSF-AIS.

C.4.4 Subcaso de prueba SC-FEC-ENC-23

Descripción:

Verificar el correcto funcionamiento del bloque *Insertor de bytes P1*, cuando se encuentra configurado en el modo de 10 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_22and23.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-22.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_with_delay[127:0]	Flujo proveniente del bloque <i>Buffer de datos</i>
J0_detected_with_delay	Señal de indicación de J0
fec_encoder_clk in	Reloj del sistema (77.76 MHz)
P1_bytes_insertor_reset	0
fec_encoder_config	1 (STM-64)
P1_bytes_ins_enable	1 (tramas 1 a 10) 0 (tramas 11 a 30)
P1_bytes_insertor_msf_ais	0 (tramas 1 a 20) 1 (tramas 21 a 30)

Tabla C.24 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-23

Estímulos:

El flujo de entrada al bloque *Insertor de bytes P1* es el flujo de salida del bloque *Buffer de datos* (ver subcaso de prueba SC-FEC-ENC-21).

Archivos de comparación:

Los archivos de comparación para el presente subcaso de prueba son *C_data_sc_fec_enc_23_group1.CSV*, *C_data_sc_fec_enc_23_group2.CSV*, *C_data_sc_fec_enc_23_group3.CSV* y *C_data_sc_fec_enc_23_group4.CSV*. El formato de estos archivos es idéntico al del subcaso SC-FEC-ENC-15 y en ellos se encuentran los datos contra los cuales se compararán los bytes P1 del flujo de salida.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_23.CSV*.

En el archivo *O_asserts_sc_fec_enc_23.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: en las tramas 1 a 10 se deberá insertar correctamente los bytes P1 que envía el bloque *Calculador de bytes P1*; en las tramas 11 a 20 el bloque deberá actuar de manera *transparente*; en las tramas 21 a 30 se deberá insertar el valor 0xFF en todos los bytes P1 de la tara de sección (SOH) / tara de sección de regeneración (RSOH), ya que se encuentra habilitada la generación de la señal de indicación de alarma MSF-AIS.

C.5 Caso de prueba TC-FEC-ENC-05

Descripción general:

El presente caso de prueba verifica el correcto funcionamiento de los bloques *Calculador de bytes B2* y *Compensador de bytes B2*. Tiene como objetivos:

- Verificar que el bloque *Calculador de bytes B2* realice correctamente el cálculo de los bytes B2 y los envíe en el tiempo y en el orden adecuado hacia el bloque *Compensador de bytes B2*. Lo anterior para cuando se encuentre deshabilitada la generación de la señal de indicación de alarma MSF-AIS.
- Verificar que el bloque *Compensador de bytes B2* inserte correctamente los bytes B2 que le envía el bloque *Calculador de bytes B2*. Lo anterior para cuando se encuentre deshabilitada la generación de la señal de indicación de alarma MSF-AIS.
- Verificar que cuando se encuentre habilitada la generación de la señal de indicación de alarma MSF-AIS, el bloque *Compensador de bytes B2* opere de forma transparente, es decir, que el flujo de datos de salida sea igual al flujo de datos de entrada.

Requerimientos a verificar:

Los requerimientos que se verifican con el caso de prueba TC-FEC-ENC-05, son los siguientes:

- R-FEC-ENC-01
- R-FEC-ENC-19
- R-FEC-ENC-20
- R-FEC-ENC-21

Subcasos de Prueba:

El caso de prueba TC-FEC-ENC-05 está dividido en los siguientes subcasos de prueba:

- SC-FEC-ENC-24
- SC-FEC-ENC-25
- SC-FEC-ENC-26
- SC-FEC-ENC-27

Esquema de verificación:

El esquema de verificación para el presente caso de prueba se muestra en la Figura C.8. En este esquema se incluyen los bloques *Insertor de Ceros*, *Calculador de bytes P1*, *Buffer de datos* e *Insertor de bytes P1*, que se verifican en los casos de prueba TC-FEC-ENC-02, TC-FEC-ENC-03 y TC-FEC-ENC-04, y que al funcionar conjuntamente con los bloques *Compensador de bytes B2* y *Calculador de bytes B2* realizan la compensación de los bytes B2 debido a la inserción de los bytes P1.

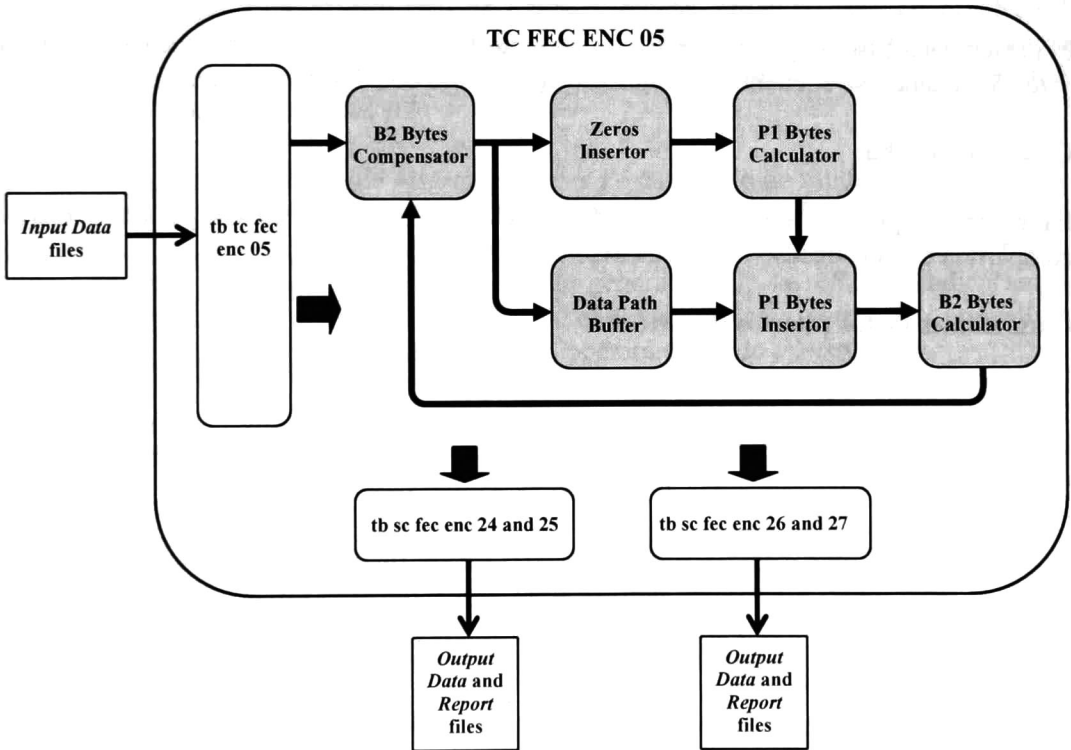


Figura C.8 Esquema de verificación del caso de prueba TC-FEC-ENC-05

La entidad *tb_tc_fec_enc_05.vhd* se encarga de generar los flujos de entrada SONET / SDH a partir de los archivos CSV de entrada: *I_data_tc_fec_enc_05_STM16.CSV* e *I_data_tc_fec_enc_05_STM64.CSV*. Además, genera las señales de configuración, reset, habilitación y de reloj para los demás bloques.

Los archivos CSV de entrada son copias de los archivos utilizados en el caso de prueba TC-FEC-ENC-01: *I_data_sc_fec_enc_01.CSV* (STM-16) e *I_data_sc_fec_enc_02.CSV* (STM-64).

Los subcasos de prueba SC-FEC-ENC-24 y SC-FEC-ENC-25 verifican el funcionamiento del bloque “*Calculador de bytes B2*” a 2.5 Gbps y 10 Gbps, respectivamente.

Los subcasos de prueba SC-FEC-ENC-26 y SC-FEC-ENC-27 verifican el funcionamiento del bloque “*Compensador de bytes B2*” a 2.5 Gbps y 10 Gbps, respectivamente.

A continuación se describen los subcasos de prueba.

C.5.1 Subcaso de prueba SC-FEC-ENC-24

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Calculador de bytes B2*, cuando se encuentra configurado en el modo de 2.5 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_24and25.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-25.

Condiciones de los puertos de entrada:

Insertor de bytes P1

Puertos de entrada	Valor
<i>data_with_delay</i> [127:0]	Flujo proveniente del bloque <i>Buffer de datos</i>
<i>J0_detected_with_delay</i>	Señal de indicación de J0
<i>fec_encoder_clk_in</i>	Reloj del sistema (19.44 MHz)
<i>P1_bytes_insertor_reset</i>	0
<i>fec_encoder_config</i>	0 (STM-16)
<i>P1_bytes_calculator_msf_ais</i>	0 (tramas 1 a 15) 1 (tramas 16 a 20) 0 (tramas 21 a 30)
<i>P1_bytes_calculator_ins_enable</i>	1 (tramas 1 a 5) 0 (tramas 6 a 10) 1 (tramas 11 a 25) 0 (tramas 26 a 30)

Tabla C.25 Condiciones de los puertos de entrada del bloque *Insertor de bytes P1* para el subcaso de prueba SC-FEC-ENC-24

Calculador de bytes B2

Puertos de entrada	Valor
data_with_P1_bytes[127:0]	Flujo proveniente del bloque <i>Insertor de bytes P1</i>
J0_detected_insertor	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)
B2_bytes_calculator_reset	0
fec_encoder_config	0 (STM-16)
B2_bytes_calculator_msf_ais	0 (tramas 1 a 15) 1 (tramas 16 a 20) 0 (tramas 21 a 30)

Tabla C.26 Condiciones de los puertos de entrada del bloque *Calculador de bytes B2* para el subcaso de prueba SC-FEC-ENC-24

Estímulos:

El flujo de entrada para el esquema de verificación del presente caso de prueba está contenido en el archivo *CSV I_data_tc_fec_enc_05_STM16.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-48 / STM-16 con contenido aleatorio. Este flujo es inyectado al bloque *Compensador de bytes B2*.

El flujo de entrada para el bloque *Insertor de bytes P1* es el flujo de salida del bloque *Buffer de datos*. En las tramas 1 a 5, 11 a 15 y 21 a 25 el bloque *Insertor de bytes P1* insertará los bytes P1 que le envía el bloque *Calculador de bytes P1*. En las tramas 16 a 20 el bloque *Insertor de bytes P1* funcionará en el modo de generación de la señal de indicación de alarma MSF-AIS. En las tramas 6 a 10 y 26 a 30 el bloque *Insertor de bytes P1* no realizará la inserción de los bytes P1 calculados, es decir, operará en modo *transparente*.

El flujo de entrada para el bloque *Calculador de bytes B2* es el flujo de salida del bloque *Insertor de bytes P1*.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_24.CSV*.

En el archivo de reporte *O_asserts_sc_fec_enc_24.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: en las tramas 1 a 15 y 21 a 30, el bloque *Calculador de bytes B2* realizará el cálculo de los bytes B2 y los enviará hacia el bloque *Compensador de bytes B2*; en las tramas 16 a 20 el bloque *Calculador de bytes B2* no realizará el cálculo de los bytes B2, dado que en estas tramas se encuentra habilitada la generación de la señal de indicación de alarma MSF-AIS. En todos los casos, el bloque

debe operar en modo *transparente*, es decir, el flujo de salida debe ser igual al flujo de entrada.

C.5.2 Subcaso de prueba SC-FEC-ENC-25

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Calculador de bytes B2*, cuando se encuentra configurado en el modo de 10 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_24and25.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-24.

Condiciones de los puertos de entrada:

Insertor de bytes P1

Puertos de entrada	Valor
data_with_delay[127:0]	Flujo proveniente del bloque <i>Buffer de datos</i>
J0_detected_with_delay	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
P1_bytes_insertor_reset	0
fec_encoder_config	1 (STM-64)
P1_bytes_calculator_msf_ais	0 (tramas 1 a 15) 1 (tramas 16 a 20) 0 (tramas 21 a 30)
P1_bytes_calculator_ins_enable	1 (tramas 1 a 5) 0 (tramas 6 a 10) 1 (tramas 11 a 25) 0 (tramas 26 a 30)

Tabla C.27 Condiciones de los puertos de entrada del bloque *Insertor de bytes P1* para el subcaso de prueba SC-FEC-ENC-25

Calculador de bytes B2

Puertos de entrada	Valor
data_with_P1_bytes[127:0]	Flujo proveniente del bloque <i>Insertor de bytes P1</i>
J0_detected_insertor	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)
B2_bytes_calculator_reset	0
fec_encoder_config	1 (STM-64)
B2_bytes_calculator_msf_ais	0 (tramas 1 a 15)

	1 (tramas 16 a 20)
	0 (tramas 21 a 30)

Tabla C.28 Condiciones de los puertos de entrada del bloque *Calculador de bytes B2* para el subcaso de prueba SC-FEC-ENC-25

Estímulos:

El flujo de entrada para el esquema de verificación del presente caso de prueba está contenido en el archivo CSV *I_data_tc_fec_enc_05_STM64.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio. Este flujo es inyectado al bloque *Compensador de bytes B2*.

El flujo de entrada para el bloque *Insertor de bytes P1* es el flujo de salida del bloque *Buffer de datos*. En las tramas 1 a 5, 11 a 15 y 21 a 25 el bloque *Insertor de bytes P1* insertará los bytes P1 que le envía el bloque *Calculador de bytes P1*. En las tramas 16 a 20 el bloque *Insertor de bytes P1* funcionará en el modo de generación de la señal de indicación de alarma MSF-AIS. En las tramas 6 a 10 y 26 a 30 el bloque *Insertor de bytes P1* no realizará la inserción de los bytes P1 calculados, es decir, operará en modo *transparente*.

El flujo de entrada para el bloque *Calculador de bytes B2* es el flujo de salida del bloque *Insertor de bytes P1*.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_25.CSV*.

En el archivo *O_asserts_sc_fec_enc_25.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: en las tramas 1 a 15 y 21 a 30, el bloque *Calculador de bytes B2* realizará el cálculo de los bytes B2 y los enviará hacia el bloque *Compensador de bytes B2*; en las tramas 16 a 20 el bloque *Calculador de bytes B2* no realizará el cálculo de los bytes B2, dado que en estas tramas se encuentra habilitada la generación de la señal de indicación de alarma MSF-AIS. En todos los casos, el bloque debe operar en modo *transparente*, es decir, el flujo de salida debe ser igual al flujo de entrada.

C.5.3 Subcaso de prueba SC-FEC-ENC-26

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Compensador de bytes B2*, cuando se encuentra configurado en el modo de 2.5 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_26and27.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-27.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
<i>data_with_fsi</i> [127:0]	Flujo de entrada SONET/SDH (<i>I_data_tc_fec_enc_05_STM16.CSV</i>)
<i>J0_detected_fsi</i>	Señal de indicación de J0
<i>fec_encoder_clk_in</i>	Reloj del sistema (19.44 MHz)
<i>B2_bytes_compensator_reset</i>	0
<i>fec_encoder_config</i>	0 (STM-16)
<i>B2_bytes_compensator_msf_ais</i>	0 (tramas 1 a 15) 1 (tramas 16 a 20) 0 (tramas 21 a 30)

Tabla C.29 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-26

Estímulos:

El flujo de entrada para el bloque *Compensador de bytes B2* está contenido en el archivo CSV *I_data_tc_fec_enc_05_STM16.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-48 / STM-16 con contenido aleatorio.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_26.CSV*.

En el archivo *O_asserts_sc_fec_enc_26.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: en las tramas 1 a 15 y 21 a 30, el bloque *Compensador de bytes B2* realizará la compensación de los bytes B2 a partir de los datos que recibe del bloque *Calculador de bytes B2*; en las tramas 16 a 20 el bloque *Compensador de bytes B2* operará de modo *transparente*, dado que en estas tramas se encuentra habilitada la generación de la señal de indicación de alarma MSF-AIS.

C.5.4 Subcaso de prueba SC-FEC-ENC-27

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Compensador de bytes B2*, cuando se encuentra configurado en el modo de 10 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_26and27.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-26.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
<code>data_with_fsi[127:0]</code>	Flujo de entrada SONET/SDH (<i>I_data_tc_fec_enc_05_STM64.CSV</i>)
<code>J0_detected_fsi</code>	Señal de indicación de J0
<code>fec_encoder_clk_in</code>	Reloj del sistema (77.76 MHz)
<code>B2_bytes_compensator_reset</code>	0
<code>fec_encoder_config</code>	1 (STM-64)
<code>B2_bytes_compensator_msf_ais</code>	0 (tramas 1 a 15) 1 (tramas 16 a 20) 0 (tramas 21 a 30)

Tabla C.30 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-27

Estímulos:

El flujo de entrada para el bloque *Compensador de bytes B2* está contenido en el archivo CSV *I_data_tc_fec_enc_05_STM64.CSV*. Consiste en 30 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_27.CSV*.

En el archivo *O_asserts_sc_fec_enc_27.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: en las tramas 1 a 15 y 21 a 30, el bloque *Compensador de bytes B2* realizará la compensación de los bytes B2 a partir de los datos que recibe del bloque *Calculador de bytes B2*; en las tramas 16 a 20 el bloque *Compensador de bytes B2* operará de modo *transparente*, dado que en estas tramas se encuentra habilitada la generación de la señal de indicación de alarma MSF-AIS.

C.6 Caso de prueba TC-FEC-ENC-06

Descripción general:

El presente caso de prueba verifica el correcto funcionamiento del bloque *Multiplexor de Salida*.

Requerimientos a verificar:

Los requerimientos que se verifican con el caso de prueba TC-FEC-ENC-06, son los siguientes:

- R-FEC-ENC-01

Subcasos de Prueba:

El caso de prueba TC-FEC-ENC-06 está dividido en los siguientes subcasos de prueba:

- SC-FEC-ENC-28
- SC-FEC-ENC-29

Esquema de verificación:

El esquema de verificación para el presente caso de prueba se muestra en la Figura C.9.

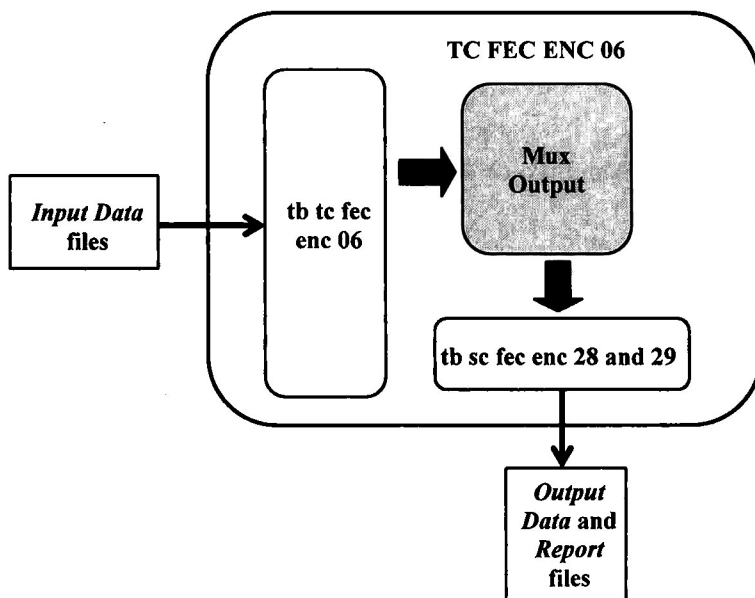


Figura C.9 Esquema de verificación del caso de prueba TC-FEC-ENC-06

La entidad *tb_tc_fec_enc_06.vhd* se encarga de generar los dos flujos de entrada SONET / SDH para el bloque *Multiplexor de salida*, a partir de archivos CSV de entrada. Además, genera las señales de selección, indicación, reset y de reloj.

Los subcasos de prueba SC-FEC-ENC-28 y SC-FEC-ENC-29 verifican el funcionamiento del bloque *Multiplexor de salida* a 2.5 Gbps y 10 Gbps, respectivamente.

A continuación se describen los subcasos de prueba.

C.6.1 Subcaso de prueba SC-FEC-ENC-28

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Multiplexor de salida*, cuando se encuentra configurado en el modo de 2.5 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_28and29.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-29.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
data_with_fsi[127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_28_FSI.CSV</i>)
data_B2_bytes_calc_out[127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_28_B2.CSV</i>)
J0_detected_fsi	Señal de indicación de J0
J0_detected_B2_bytes_calc_out	Señal de indicación de J0
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)
mux_output_reset	0
mux_output_sel	0 (trayecto sin retardo) 1 (trayecto con retardo)

Tabla C.31 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-28

Estímulos:

Los flujos de entrada para el bloque *Multiplexor de salida* se obtienen a partir de los archivos CSV: *I_data_sc_fec_enc_28_FSI.CSV* e *I_data_sc_fec_enc_28_B2.CSV*. Ambos archivos consisten en 30 tramas SONET / SDH de orden STS-48 / STM-16 con contenido aleatorio.

La entidad *tb_tc_fec_enc_06.vhd* genera de manera independiente las señales de indicación y de selección *J0_detected_fsi*, *J0_detected_B2_bytes_calc_out* y *mux_output_sel*, de forma que tomen los valores “0” y “1” alternadamente y que su duración en uno u otro valor sea variable. Esto con el fin de que estas señales presenten variaciones significativas durante la ejecución del subcaso de prueba.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_28.CSV*.

En el archivo *O_asserts_sc_fec_enc_28.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: cuando la entrada de selección del bloque *Multiplexor de salida* sea “0”, el flujo de datos de salida será igual al flujo entrante *data_with_fsi* y la señal de salida *fec_encoder_J0_detected_out* será igual a la señal de entrada *J0_detected_fsi*; cuando la entrada de selección del bloque *Multiplexor de salida* sea “1”, el flujo de datos de salida será igual al flujo entrante *data_B2_bytes_calc_out* y la señal de salida *fec_encoder_J0_detected_out* será igual a la señal de entrada *J0_detected_B2_bytes_calc_out*.

C.6.2 Subcaso de prueba SC-FEC-ENC-29

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Multiplexor de salida*, cuando se encuentra configurado en el modo de 10 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_28and29.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-28.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
<i>data_with_fsi</i> [127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_29_FSI.CSV</i>)
<i>data_B2_bytes_calc_out</i> [127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_29_B2.CSV</i>)
<i>J0_detected_fsi</i>	Señal de indicación de J0
<i>J0_detected_B2_bytes_calc_out</i>	Señal de indicación de J0
<i>fec_encoder_clk_in</i>	Reloj del sistema (77.76 MHz)
<i>mux_output_reset</i>	0
<i>mux_output_sel</i>	0 (trayecto sin retardo) 1 (trayecto con retardo)

Tabla C.32 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-29

Estímulos:

Los flujos de entrada para el bloque *Multiplexor de salida* se obtienen a partir de los archivos CSV: *I_data_sc_fec_enc_29_FSI.CSV* e *I_data_sc_fec_enc_29_B2.CSV*. Ambos archivos consisten en 30 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio.

La entidad *tb_tc_fec_enc_06.vhd* genera de manera independiente las señales de indicación y de selección *J0_detected_fsi*, *J0_detected_B2_bytes_calc_out* y *mux_output_sel*, de forma que tomen los valores “0” y “1” alternadamente y que su duración en uno u otro valor sea variable. Esto con el fin de que estas señales presenten variaciones significativas durante la ejecución del subcaso de prueba.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_29.CSV*.

En el archivo *O_asserts_sc_fec_enc_29.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

El comportamiento funcional esperado es el siguiente: cuando la entrada de selección del bloque *Multiplexor de salida* sea “0”, el flujo de datos de salida será igual al flujo entrante *data_with_fsi* y la señal de salida *fec_encoder_J0_detected_out* será igual a la señal de entrada *J0_detected_fsi*; cuando la entrada de selección del bloque *Multiplexor de salida* sea “1”, el flujo de datos de salida será igual al flujo entrante *data_B2_bytes_calc_out* y la señal de salida *fec_encoder_J0_detected_out* será igual a la señal de entrada *J0_detected_B2_bytes_calc_out*.

C.7 Caso de prueba TC-FEC-ENC-07

Descripción general:

El presente caso de prueba verifica el correcto funcionamiento del bloque *Control*. Tiene como objetivos:

- Verificar que la máquina de estados del bloque *Control* alcance todos los estados y subestados y realice las transiciones correctamente, en respuesta a las señales de entrada y señales internas presentes.
- Verificar que el bloque *Control* genere correctamente las señales de control para todos los bloques que integran el módulo codificador FEC en banda.

Requerimientos a verificar:

Los requerimientos que se verifican con el caso de prueba TC-FEC-ENC-07, son los siguientes:

- R-FEC-ENC-04
- R-FEC-ENC-12
- R-FEC-ENC-13
- R-FEC-ENC-14
- R-FEC-ENC-15
- R-FEC-ENC-16
- R-FEC-ENC-17
- R-FEC-ENC-18
- R-FEC-ENC-21

Subcasos de Prueba:

El caso de prueba TC-FEC-ENC-07 está dividido en los siguientes subcasos de prueba:

- SC-FEC-ENC-30
- SC-FEC-ENC-31

Esquema de verificación:

El esquema de verificación para el presente caso de prueba se muestra en la Figura C.10.

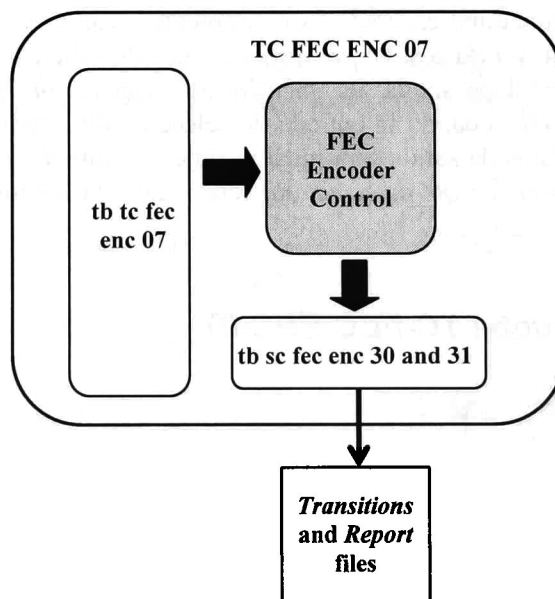


Figura C.10 Esquema de verificación del caso de prueba TC-FEC-ENC-07

La entidad *tb_tc_fec_enc_07.vhd* genera los estímulos para el bloque *Control*. Entre estos estímulos se encuentran las señales para realizar el cambio del estado operacional del módulo codificador y la señal de habilitación para la generación de la señal de indicación de alarma MSF-AIS. Estas señales son generadas de tal forma que permitan verificar que la máquina de control alcance todos los estados y subestados posibles.

Los subcasos de prueba SC-FEC-ENC-30 y SC-FEC-ENC-31 verifican el funcionamiento del bloque *Control* a 2.5 Gbps y 10 Gbps, respectivamente. El tiempo de simulación para cada subcaso de prueba corresponde a la duración de 190 tramas SONET / SDH.

A continuación se describen los subcasos de prueba.

C.7.1 Subcaso de prueba SC-FEC-ENC-30

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Control*, cuando se encuentra configurado en el modo de 2.5 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_30and31.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-31.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
fec_encoder_J0_detected_in	Señal de indicación de J0
fec_encoder_config_in	0 (STM-16)
fec_encoder_reset_in	0, 1
fec_encoder_ch_state	Señal de indicación de J0
fec_encoder_state	"01", "10", "11"
fec_encoder_msf_ais_in	0, 1
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)

Tabla C.33 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-30

Estímulos:

Los estímulos son generados por la entidad *tb_tc_fec_enc_07.vhd*.

Respuestas:

En el archivo de reporte *O_asserts_sc_fec_enc_30.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

En el archivo de transiciones *O_transitions_sc_fec_enc_30.CSV* se registran todas las transiciones que ocurren en la señal *estado* y en la señal *subestado* de la máquina de control.

C.7.2 Subcaso de prueba SC-FEC-ENC-31

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del bloque *Control*, cuando se encuentra configurado en el modo de 10 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_30and31.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-30.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
<i>fec_encoder_J0_detected_in</i>	Señal de indicación de J0
<i>fec_encoder_config_in</i>	1 (STM-64)
<i>fec_encoder_reset_in</i>	0, 1
<i>fec_encoder_ch_state</i>	Señal de indicación de J0
<i>fec_encoder_state</i>	“01”, “10”, “11”
<i>fec_encoder_msf_ais_in</i>	0, 1
<i>fec_encoder_clk_in</i>	Reloj del sistema (77.76 MHz)

Tabla C.34 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-31

Estímulos:

Los estímulos de control son generados por la entidad *tb_tc_fec_enc_07.vhd*.

Respuestas:

En el archivo de reporte *O_asserts_sc_fec_enc_31.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

En el archivo de transiciones *O_transitions_sc_fec_enc_31.CSV* se registran todas las transiciones que ocurren en la señal *estado* y en la señal *subestado* de la máquina de control.

C.8 Caso de prueba TC-FEC-ENC-08

Descripción general:

El presente caso de prueba realiza la verificación funcional a nivel *top* del módulo codificador FEC en banda.

Tiene como objetivos:

- Verificar la correcta generación e inserción de la señal FSI y bytes Q1.
- Verificar el cálculo e inserción de los bytes P1.
- Verificar la compensación de los bytes de paridad B2.
- Verificar la generación de la señal de indicación de alarma MSF-AIS.
- Verificar la no violación en el máximo retardo permitido al flujo de datos.
- Verificar que el módulo codificador actúe de manera transparente en los bytes restantes de las tramas SONET / SDH, los cuales no deben ser alterados por la función FEC en banda.

Requerimientos a verificar:

Los requerimientos que se verifican con el caso de prueba TC-FEC-ENC-08, son los siguientes:

- R-FEC-ENC-01
- R-FEC-ENC-02
- R-FEC-ENC-03
- R-FEC-ENC-04
- R-FEC-ENC-05
- R-FEC-ENC-06
- R-FEC-ENC-07
- R-FEC-ENC-08
- R-FEC-ENC-09
- R-FEC-ENC-10
- R-FEC-ENC-11
- R-FEC-ENC-12
- R-FEC-ENC-13
- R-FEC-ENC-14
- R-FEC-ENC-15
- R-FEC-ENC-18
- R-FEC-ENC-19
- R-FEC-ENC-21
- R-FEC-ENC-22

Subcasos de Prueba:

El caso de prueba TC-FEC-ENC-08 está dividido en los siguientes subcasos de prueba:

- SC-FEC-ENC-32

- SC-FEC-ENC-33

Esquema de verificación:

El esquema de verificación del presente caso de prueba es de caja negra ya que no se tiene acceso a bloques y señales internas del módulo codificador FEC en banda, sino solamente se analiza el flujo de datos SONET / SDH a su salida. En la Figura C.11 se muestra este esquema de verificación.

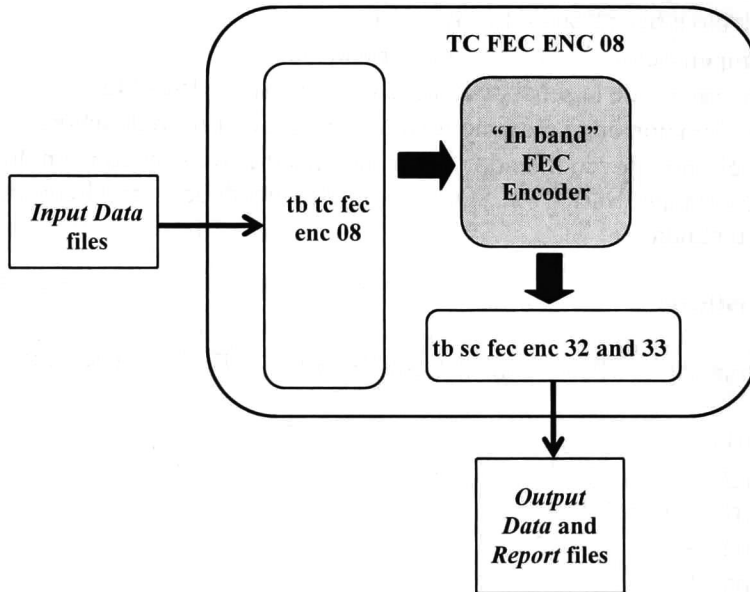


Figura C.11 Esquema de verificación del caso de prueba TC-FEC-ENC-08

La entidad *tb_tc_fec_enc_07.vhd* se encarga de generar los flujos de entrada SONET / SDH a partir de los archivos CSV de entrada: *I_data_sc_fec_enc_32.CSV* e *I_data_sc_fec_enc_33.CSV*. Además genera los estímulos de entrada de forma que se verifique el módulo codificador en todos los estados operacionales posibles. Estos estímulos son generados de igual forma que en el caso de prueba TC-FEC-ENC-07 para hacer que la máquina de control alcance todos los estados y subestados posibles.

Los subcasos de prueba SC-FEC-ENC-32 y SC-FEC-ENC-33 verifican el funcionamiento del módulo codificador FEC en banda a 2.5 Gbps y 10 Gbps, respectivamente. El tiempo de simulación para cada subcaso de prueba corresponde a la duración de 190 tramas SONET / SDH.

La cama de prueba es compleja y consta de varios procesos que verifican las funciones que ejecuta el módulo codificador FEC en banda en base al estado operacional presente.

A continuación se describen los subcasos de prueba.

C.8.1 Subcaso de prueba SC-FEC-ENC-32

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del módulo codificador de FEC en banda, cuando se encuentra configurado en el modo de 2.5 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_32and33.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-33.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
fec_encoder_data_in[127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_32.CSV</i>)
fec_encoder_J0_detected_in	Señal de indicación de J0
fec_encoder_config_in	0 (STM-16)
fec_encoder_reset_in	0, 1
fec_encoder_ch_state	Señal de indicación de J0
fec_encoder_state	“01”, “10”, “11”
fec_encoder_msf_ais_in	0, 1
fec_encoder_clk_in	Reloj del sistema (19.44 MHz)

Tabla C.35 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-32

Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_sc_fec_enc_32.CSV*. Consiste en 190 tramas SONET / SDH de orden STS-48 / STM-16 con contenido aleatorio.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_32.CSV*.

||En el archivo *O_asserts_sc_fec_enc_32.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.

C.8.2 Subcaso de prueba SC-FEC-ENC-33

Descripción:

El presente subcaso de prueba verifica el correcto funcionamiento del módulo codificador FEC en banda, cuando se encuentra configurado en el modo de 10 Gbps.

Cama de prueba:

La cama de prueba fue implementada en el archivo *tb_sc_fec_enc_32and33.vhd*. Es compartida con el subcaso de prueba SC-FEC-ENC-32.

Condiciones de los puertos de entrada:

Puertos de entrada	Valor
fec_encoder_data_in[127:0]	Flujo de entrada SONET/SDH (<i>I_data_sc_fec_enc_33.CSV</i>)
fec_encoder_J0_detected_in	Señal de indicación de J0
fec_encoder_config_in	1 (STM-64)
fec_encoder_reset_in	0, 1
fec_encoder_ch_state	Señal de indicación de J0
fec_encoder_state	"01", "10", "11"
fec_encoder_msf_ais_in	0, 1
fec_encoder_clk_in	Reloj del sistema (77.76 MHz)

Tabla C.36 Condiciones de los puertos de entrada para el subcaso de prueba SC-FEC-ENC-33

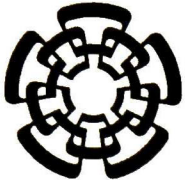
Estímulos:

El flujo de entrada para el presente subcaso de prueba se encuentra en el archivo CSV *I_data_sc_fec_enc_33.CSV*. Consiste en 190 tramas SONET / SDH de orden STS-192 / STM-64 con contenido aleatorio.

Respuestas:

El flujo de salida es escrito en el archivo *O_data_sc_fec_enc_33.CSV*.

En el archivo *O_asserts_sc_fec_enc_33.CSV* se escriben los errores detectados durante la verificación funcional del bloque, indicando la trama y el tiempo de simulación en el cual ocurrieron.



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N. UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Módulo codificador FEC en banda para elementos de red SONET/SDH a
2.5 Gbps y 10 Gbps

del (la) C.

Baldemar IRINEO CARRASCO

el día 08 de Septiembre de 2006.

Dr. Deni Librado Torres Román
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Ramón Parra Michel
Investigador CINVESTAV 2A
CINVESTAV Unidad Guadalajara

Dr. Luis Fernando González Pérez
Director del Centro de Diseño
Electrónico
Instituto Tecnológico y de Estudios
Superiores de Monterrey



CINVESTAV
BIBLIOTECA CENTRAL



SSIT000008746