

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERA ELÉCTRICA



**“Diseño y Construcción de un Analizador de  
Señalización por Canal Común #7”**

PRESENTADO POR:

**EDUAR YEINS BARILLAS GUANDIQUE  
YIMI ALEXANDER GALLEGOS INTERIANO  
ROBERTO EDUARDO SARAVIA GUTIÉRREZ**

PARA OPTAR AL TÍTULO DE:  
**INGENIERO ELECTRICISTA**

CIUDAD UNIVERSITARIA, MAYO DEL 2007.

**UNIVERSIDAD DE EL SALVADOR**

RECTORA :

**DRA. MARÍA ISABEL RODRÍGUEZ**

SECRETARIA GENERAL :

**LICDA. ALICIA MARGARITA RIVAS DE RECINOS**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA**

DECANO :

**ING. MARIO ROBERTO NIETO LOVO**

SECRETARIO :

**ING. OSCAR EDUARDO MARROQUÍN HERNÁNDEZ**

**ESCUELA DE INGENIERÍA ELÉCTRICA**

DIRECTOR :

**ING. LUIS ROBERTO CHÉVEZ PAZ**

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

**INGENIERO ELECTRICISTA**

Título :

**“Diseño y Construcción de un Analizador de  
Señalización por Canal Común #7”**

Presentado por :

EDUAR YEINS BARILLAS GUANDIQUE  
YIMI ALEXANDER GALLEGOS INTERIANO  
ROBERTO EDUARDO SARAVIA GUTIERREZ

Trabajo de Graduación aprobado por:

Docente Director :

Ing. Werner David Meléndez Valle

San Salvador, Mayo de 2007.

Trabajo de Graduación Aprobado por:

Docente Director:

**Ing. Werner David Meléndez Valle**

## RESUMEN

---

El objetivo principal del presente trabajo es el diseño y la implementación de un instrumento virtual para monitoreo de señalización #7. Para dar cumplimiento a este objetivo, el trabajo se ha desarrollado con la siguiente estructura: Generalidades e información básica del sistema de señalización #7, sistema de señalización #7, diseño del circuito eléctrico interfaz, implementación y fallas y desarrollo del software de decodificación.

En el primer capítulo de este documento se desarrollan algunos conceptos básicos sobre la naturaleza del sistema de señalización además de las especificaciones técnicas del tren PCM, en este se presentan conceptos tales como codificación y cuantificación, principios de multiplexación, estructura de la trama primaria E1, entre otros.

En el capítulo dos, se describe en detalle todo el sistema de señalización #7, además del protocolo ISUP en el que se enfoca principalmente nuestra investigación, primero analizamos las características principales de la estructura de la red de señalización en el que se incluyen los elementos que constituyen una red tales como puntos de señalización y los tipos de enlace, posteriormente se describen las características del modelo de aproximación por capas del sistema de señalización #7, los tipos de mensajes, una breve descripción de los principales protocolos y un desglose más detallado del MTP (parte de transferencia de mensajes) y por último en este capítulo se describe en detalle el nivel del protocolo de señalización #7 conocido como ISUP.

El capítulo tres se describen los procesos para la construcción de una interfase capaz de muestrear y monitorear a través de tráfico real mensajes de nivel cuatro contenidos en la interconexión de dos centrales telefónicas, diseñando un instrumento basado en tecnología TTL, circuitos dedicados y auxiliado con microcontroladores.

En el capítulo cuatro, se describen los problemas que se encontraron cuando se realizaron las pruebas del hardware en centrales telefónicas, además se proponen soluciones para mejorar los diseños de proyectos o estudios futuros.

El capítulo cinco, se detallan las herramientas necesarias en software para la realización del sistema por computadora que controla el funcionamiento del hardware, y que implementa el protocolo para la decodificación de los paquetes capturados por el instrumento. Se describen técnicas de programación así como librerías utilizadas para la programación de la interfaz gráfica, algoritmos principales y clases usadas.

## PROLOGO

---

En este trabajo se ha desarrollado una interfase con la cual se puede monitorear la señalización entre dos centrales telefónicas que se comuniquen por medio del protocolo de señalización #7, la interfase se encarga de recolectar los datos y los envía a una computadora, en la cual por medio de un software desarrollado en Python los datos son decodificados y presentados en pantalla.

Para desarrollar esta interfase, es necesario establecer las bases para un conocimiento sólido de la Arquitectura del sistema de señalización # 7, profundizando en el conocimiento de las capas MTP e ISUP del protocolo y así poder diseñar un hardware y un software que sea capaz de capturar, decodificar y mostrar en pantalla los tipos de mensaje en tiempo real.

El diseño del hardware se realizó utilizando: El CI DS2187, que se encarga de convertir de HDB3 a TTL, el DS2143 que se encarga de realizar la alineación de tramas y colocar los datos en sus respectivas posiciones para su posterior captura, un microcontrolador es utilizado para la comunicación con el DS2143, para la captura de los datos y para la comunicación con el DLPUSB245, este último es utilizado para la comunicación del software con el microcontrolador y para el envío de los datos hacia la computadora. La interfase se elaboró tomando en cuenta todas las recomendaciones ITU-T.

El software se desarrolló en python, ya que es open source license, de fácil aprendizaje, de fácil desarrollo de aplicaciones y tiene la ventaja de que corre en Linux y en Windows.

## TABLA DE CONTENIDOS

---

<i>Capítulo</i>	<i>Página</i>
<b>1 GENERALIDADES E INFORMACION BASICA DE SEÑALIZACIÓN #7</b> ..15	
1.1 INTRODUCCION A LAS REDES DE SEÑALIZACIÓN .....	15
1.1.1 Concepto de señalización.....	15
1.1.2 Tipos de señalización.....	15
1.1.3 Señalización fuera de banda.....	16
1.1.4 Objetivos de la señalización fuera de banda .....	16
1.2 LA INTERFAZ E1 2048 kbps.....	16
1.3 SEÑAL PCM .....	17
1.3.1 Muestreo .....	18
1.3.2 Cuantificación y codificación .....	18
1.3.3 PCM FRAMING.....	20
1.3.3.1 Principios de multiplexación.....	20
1.3.3.2 La trama primaria.....	20
1.3.3.3 Alineación de trama .....	21
1.3.3.4 Señal de alineación de trama.....	21
1.3.3.5 Sincronización de trama.....	22
1.3.4 Señalización R2 .....	23
1.3.4.1 Chequeo de redundancia cíclica (CRC).....	23
1.3.4.2 Método CRC-4.....	24
1.3.5 ALARMAS .....	27
1.3.5.1 Alarmas remotas .....	27
1.3.5.2 Indicación alarma remota.....	27
1.3.5.3 Señal de indicación de alarma (AIS) .....	28
1.3.6 Códigos de línea.....	28
1.3.6.1 Código AMI .....	29
1.3.6.2 Código HDB3 .....	29
1.3.7 Recomendaciones ITU-T G.703 (PCM 2048Kbps).....	30
1.4 CONCLUSIONES DEL CAPITULO I .....	31
1.5 REFERENCIA BIBLIOGRÁFICA .....	32
<b>2 SISTEMA DE SEÑALIZACIÓN # 7</b> .....	33
2.1 ESTRUCTURA DE LA RED DE SEÑALIZACION .....	33
2.1.1 Punto de Servicio de Conmutación (SSP) .....	34
2.1.2 Punto de Transferencia de Señalización (STP).....	35
2.1.3 Punto de control de servicio (SCP).....	37

2.1.4	Enlaces de datos de señalización .....	38
2.1.5	Rutas .....	40
2.1.6	Point Codes .....	41
2.1.7	Rendimiento del enlace.....	41
2.1.8	Interfase de enlace físico.....	41
2.2	MODELO EQUIVALENTE DEL C7 .....	42
2.2.1	Nivel uno – capa física.....	43
2.2.2	Nivel dos – nivel de enlace de datos .....	43
2.2.3	Nivel tres – nivel de red .....	44
2.2.4	Nivel cuatro – parte usuaria .....	45
2.3	UNIDADES DE SEÑALIZACION .....	45
2.3.1	Generalidades de conmutación en la red.....	45
2.3.2	Estructura de paquetes. ....	46
2.3.2.1	Unidades de señalización de relleno (FISUs): .....	47
2.3.2.2	Unidades de señalización del estado del enlace (LSSUs):.....	49
2.3.2.3	Unidades de señalización de mensajes (MSUs):.....	50
2.3.3	Primitivas .....	51
2.4	INTRODUCCIÓN A LOS PROTOCOLOS DEL SISTEMA DE SEÑALIZACIÓN #7 .....	52
2.4.1	Parte de Transferencia de mensajes (MTP) .....	52
2.4.2	Capa de Adaptación del usuario a MTP2 (M2UA). ....	53
2.4.3	Capa de Adaptación del usuario a MTP3 (M3UA). ....	53
2.4.4	Protocolo de Transmisión de Señalización de Control (SCTP).....	53
2.4.5	Parte de Control de Conexión de la Señalización (SCCP).....	53
2.4.6	Parte de Usuario del ISDN (ISUP). ....	54
2.4.7	Parte de Usuario del ISDN de Banda Ancha (BISUP). ....	54
2.4.8	Parte de Usuario para Telefonía (TUP). ....	54
2.4.9	Parte de Aplicación de Capacidad de Transacción (TCAP). ....	54
2.5	PROTOCOLO MTP .....	55
2.5.1	Generalidades.....	55
2.5.2	Objetivos.....	55
2.5.3	Características generales.....	55
2.5.3.1	Método de descripción.....	55
2.5.4	Descripción del MTP (nivel dos).....	56
2.5.4.1	Estructura del MTP nivel dos (Unidades de Señalización). ....	58
2.5.4.2	Método de control básico de error. ....	61
2.5.5	MTP nivel tres.....	67
2.5.5.1	Estructura de los mensajes. ....	68
2.5.5.1.1	Discriminación de mensajes.....	68
2.5.5.1.2	Distribución de mensajes .....	69
2.5.5.1.3	Encaminamiento de mensajes .....	69
2.5.5.1.4	Selección de enlace .....	71
2.5.5.2	Administración de red.....	71
2.5.5.3	Procedimientos de administración de señalización de red.....	72
2.5.5.4	Procedimientos de administración de enlace .....	73
2.5.5.5	Procedimientos de administración de tráfico .....	74



2.5.5.6	Estructura del mensaje de administración de tráfico.....	75
2.5.5.7	Procedimientos de administración de rutas.....	77
2.6	PROTOCOLO ISUP.....	78
2.6.1	Consideraciones generales.....	78
2.6.1.1	Etiqueta de encaminamiento.....	79
2.6.1.2	Código de identificación de circuito.....	80
2.6.1.3	Código de tipo de mensaje.....	80
2.6.1.4	Parte obligatoria de longitud fija.....	82
2.6.1.5	Parte obligatoria de longitud variable.....	82
2.6.1.6	Parte facultativa.....	82
2.6.2	Principios de formación de los mensajes.....	83
2.6.2.1	Formatos y códigos de los parámetros.....	83
2.6.3	Principales mensajes usados para el establecimiento de una llamada básica. 85	
2.6.3.1	IAM.....	85
2.6.3.2	ACM.....	86
2.6.3.3	ANM.....	87
2.6.3.4	REL.....	87
2.6.3.5	RLC.....	88
2.6.4	Ejemplo del establecimiento de una llamada Básica.....	88
2.7	CONCLUSIONES DEL CAPITULO II.....	91
2.8	REFERENCIA BIBLIOGRÁFICA.....	92
<b>3</b>	<b>DISEÑO DEL CIRCUITO DE INTERFAZ.....</b>	<b>93</b>
3.1	INTRODUCCION.....	93
3.2	DIAGRAMA DE BLOQUES GENERAL.....	93
3.3	CIRCUITO DE ACOUPLE.....	94
3.4	INTERFAZ DE LINEA E1 (DS2187).....	95
3.4.1	Detector de picos y deslizamientos:.....	96
3.4.2	Extracción de reloj:.....	96
3.4.3	Supresión de código Cero.....	97
3.4.4	Detección de alarma.....	97
3.5	ETAPA DE FRAMER.....	98
3.5.1	PUERTO PARALELO.....	99
3.5.2	CONTROL Y PRUEBA DE REGISTROS.....	101
3.5.3	REGISTROS DE ESTADO Y DE INFORMACION.....	104
3.5.4	RELOJ DE CANAL Y RELOJ DE BLOQUE DE REGISTROS.....	105
3.6	ETAPA DE CONVERSIÓN SERIE PARALELO.....	106
3.6.1	Latch de salida.....	107
3.7	ETAPA DEL PIC18F452.....	108
3.7.1	Descripción General del microcontrolador PICF18F452.....	108
3.7.2	Arquitectura Interna.....	109
3.7.2.1	Flujo de instrucciones PIPELINE.....	109
3.7.2.2	Procesador tipo RISC.....	110

3.7.2.3	Arquitectura Harvard .....	111
3.7.2.4	Arquitectura Ortogonal .....	112
3.7.2.5	Arquitectura basada en bancos de registros .....	112
3.7.3	Lógica de Interrupciones.....	113
3.7.4	Módulos de interés .....	116
3.7.4.1	Puerto A .....	116
3.7.4.2	Puerto B .....	116
3.7.4.3	Puerto C .....	116
3.7.4.4	Puerto D .....	116
3.7.5	Pin de interrupción externa (RB0/INT) .....	116
3.7.6	Flujograma del controlador principal en el PIC18F452 .....	117
3.7.7	Lógica de Interrupciones en el PIC18F452.....	118
3.7.8	Sub-rutinas para escritura en chip Framer DS2143 .....	119
3.8	INTERFAZ USB .....	120
3.8.1	DLP-USB245M .....	120
3.8.2	Configuración de Pines .....	121
3.8.3	Diferentes tipos de polarización.....	124
3.8.3.1	Alimentación externa (sistema a 5V).....	125
3.8.3.2	Alimentación externa (sistema con interfase lógica a 3.3V). .....	126
3.8.4	DIAGRAMAS DE TIEMPO .....	126
3.8.4.1	Diagrama de temporización para el ciclo de lectura .....	126
3.8.4.2	Diagrama de temporización para el ciclo de escritura .....	127
3.9	CONCLUSIONES DEL CAPITULO III.....	128
3.10	REFERENCIA BIBLIOGRÁFICA.....	128
<b>4</b>	<b>IMPLEMENTACION Y FALLAS .....</b>	<b>129</b>
4.1	INTRODUCCION .....	129
4.2	PROBLEMAS PRESENTES EN LA IMPLEMENTACION.....	129
4.3	CRITERIOS DE DISEÑO DEL CIRCUITO IMPRESO .....	131
4.3.1	Como evitar Interferencias.....	131
4.3.2	Desacople de circuitos dentro de una misma tarjeta .....	132
4.3.3	Consideraciones para el diseño de la tableta.....	133
4.3.3.1	Masas independientes: .....	133
4.3.3.2	Chasis del equipo: .....	133
4.3.3.3	Tamaño de las pistas .....	133
4.3.3.4	Capacitor de Bulk .....	134
4.4	POSIBLES CAUSAS DE FALLAS .....	134
4.4.1	Alta concentración de Ruido.....	134
4.4.2	Desviación de frecuencia de fase (Jitter). .....	134
4.5	MODIFICACIONES PROPUESTAS .....	135
4.5.1	Circuito de acople .....	135
4.5.2	Ruido interno generado por oscilador.....	136
4.5.3	Lectura de dos canales adyacentes.....	136
4.5.4	Sincronización de dos canales .....	137

4.5.5	PROPUESTA DE MEJORA DEL CIRCUITO.....	137
4.6	CONCLUSIONES DEL CAPITULO IV .....	139
4.7	REFERENCIA BIBLIOGRÁFICA.....	139
<b>5</b>	<b>DISEÑO DE SOFTWARE.....</b>	<b>140</b>
5.1	INTRODUCCIÓN .....	140
5.2	EL DRIVER FTD2XX .....	140
5.3	INTERFASES DE PROGRAMACION DEL CONTROLADOR FTD2XX..	141
5.4	FLUJO DE DATOS A TRAVES DEL PUERTO USB .....	141
5.5	LENGUAJE DE PROGRAMACION "PYTHON" .....	143
5.5.1	PROGRAMACIÓN ORIENTADA A OBJETOS.....	143
5.5.2	PROGRAMACIÓN MULTI-THREADING EN PYTHON .....	144
5.5.2.1	SINCRONIZMO DE THREADINGS .....	146
5.5.3	LIBRERÍA DE OBJETOS GRÁFICOS: <i>WXPYTHON</i> .....	146
5.5.4	Patrón MVP .....	148
5.6	Estructura del Programa de Descodificación.....	150
5.6.1	Flujogramas del software.....	150
5.6.1.1	Descripción general del software.....	150
5.6.1.2	Descripción de la clase colectora de datos:.....	151
5.6.1.3	Descodificación Capa Dos.....	152
5.6.1.4	Codificación ISUP. ....	155
5.6.1.5	Captura en modo Traceo de Llamadas.....	164
5.6.1.6	Interfaz Grafica de Usuario.....	168
5.7	CONCLUSIONES DEL CAPITULO V.....	171
5.8	REFERENCIA BIBLIOGRÁFICA.....	171

## LISTA DE FIGURAS

---

<i><b>Figura</b></i>	<i><b>Página</b></i>
Figura 1.1 <i>Muestreo de la señal de voz</i> .....	18
Figura 1.2 <i>Niveles de cuantificación (Ley A) y característica ruido por cuantificación</i> ...	19
Figura 1.3 <i>Multiplexación</i> .....	20
Figura 1.4 <i>Trama primaria</i> .....	20
Figura 1.5 <i>Trama PCM30</i> .....	21
Figura 1.6 <i>Señal de alineación de trama</i> .....	21
Figura 1.7 <i>Diagrama de Sincronización de trama</i> .....	22
Figura 1.8 <i>Señalización utilizando multiplexores E&amp;M</i> .....	23
Figura 1.9 <i>Ejemplos de sincronización incorrecta</i> .....	24
Figura 1.10 <i>Diagrama de la transmisión de residuos en el método CRC-4</i> .....	25
Figura 1.11 <i>Indicación de alarma remota</i> .....	27
Figura 1.12 <i>Caídas en los niveles de voltaje de la señal</i> .....	29
Figura 1.13 <i>Tren con codificación AMI</i> .....	29
Figura 1.14 <i>Tren codificación HDB3</i> .....	30
Figura 1.15 <i>Máscara del pulso del tren PCM</i> .....	31
Figura 2.1 <i>Red típica C7, con múltiples enlaces</i> .....	34
Figura 2.2 <i>Relaciones del SSP en la red señalización #7</i> .....	35
Figura 2.3 <i>Relaciones del STPs en las redes de señalización # 7</i> .....	36
Figura 2.4 <i>Relaciones del SCP en la red de señalización #7</i> .....	37
Figura 2.5 <i>Diferentes tipos de enlaces de acuerdo a su disposición en la red</i> .....	39
Figura 2.6 <i>Modelo de pila SS7</i> .....	42
Figura 2.7 <i>Estructura de la unidad de señalización de relleno</i> .....	47
Figura 2.8 <i>Estructura de la unidad de señalización del estado del enlace</i> .....	49
Figura 2.9 <i>Estructura de la unidad de señalización de mensajes</i> .....	50
Figura 2.10 <i>Estructura de las primitivas</i> .....	51
Figura 2.11 <i>Componentes utilizados en el Nivel dos del MTP</i> .....	58
Figura 2.12 <i>Petición de retransmisión</i> .....	61
Figura 2.13 <i>Secuencia de transmisión exitosa</i> .....	62
Figura 2.14 <i>Componentes del LSSU</i> .....	63
Figura 2.15 <i>Estructura de la etiqueta de encaminamiento</i> .....	68
Figura 2.16 <i>Componentes del mensaje UPU</i> .....	69
Figura 2.17 <i>Estructura del PC internacional</i> .....	70
Figura 2.18 <i>Mensaje de mantenimiento de red</i> .....	73
Figura 2.19 <i>Mensaje señalización-datos-enlace-conexión</i> .....	74
Figura 2.20 <i>Mensaje de orden de intercambio (COO)</i> .....	75
Figura 2.21 <i>Declaración de retorno de intercambio</i> .....	76
Figura 2.22 <i>Transferencia-prohibida y transferencia-cluster-prohibida</i> .....	77
Figura 2.23 <i>Estructura del mensaje ISUP dentro de las MSUs</i> .....	79

Figura 2.24 <i>Etiqueta de encaminamiento.</i> .....	79
Figura 2.25 <i>Campo de identificación de circuito.</i> .....	80
Figura 2.26 <i>Formato General de mensaje ISUP.</i> .....	85
Figura 2.27 <i>Formato del IAM.</i> .....	86
Figura 2.28 <i>Formato del ACM.</i> .....	87
Figura 2.29 <i>Formato de la estructura de un ANM.</i> .....	87
Figura 2.30 <i>Formato del mensaje REL.</i> .....	88
Figura 2.31 <i>Estructura del RLC.</i> .....	88
Figura 2.32 <i>Ejemplo del establecimiento de una llamada</i> .....	89
Figura 3.1 <i>Diagrama de bloques de circuito interfaz.</i> .....	94
Figura 3.2 <i>Circuito de acoplamiento.</i> .....	95
Figura 3.3 <i>Diagrama de bloques interno del DS2187.</i> .....	95
Figura 3.4 <i>Diagrama del circuito de Interfaz de línea.</i> .....	97
Figura 3.5 <i>Diagrama de tiempo de las señales de salida del DS2187.</i> .....	97
Figura 3.6 <i>Diagrama de bloques del DS2143.</i> .....	99
Figura 3.7 <i>Tiempo de escritura del DS2143</i> .....	100
Figura 3.8 <i>Tiempo de lectura del DS2143</i> .....	100
Figura 3.9 <i>Diagrama de tiempo de RCHCLK y RCHBLK.</i> .....	105
Figura 3.10 <i>Diagrama del circuito del framer.</i> .....	106
Figura 3.11 <i>Circuito de la etapa de conversión serie paralelo</i> .....	106
Figura 3.12 <i>Diagrama de ciclos de Instrucción del PIC18F452</i> .....	110
Figura 3.13 <i>Estructura von Neumann</i> .....	111
Figura 3.14 <i>Estructura harvard</i> .....	111
Figura 3.15 <i>Mapa de memoria del PIC18F452</i> .....	112
Figura 3.16 <i>Mapa registros de funciones especiales.</i> .....	113
Figura 3.17 <i>Lógica de Interrupciones</i> .....	114
Figura 3.18 <i>Algoritmos de programa principal.</i> .....	117
Figura 3.19 <i>Algoritmos de rutinas de atención a las dos fuentes de interrupción.</i> .....	118
Figura 3.20 <i>Subrutinas de escritura para el DS2143.</i> .....	119
Figura 3.21 <i>Instalación del controlador.</i> .....	121
Figura 3.22 <i>Alimentación básica desde el bus USB.</i> .....	124
Figura 3.23 <i>Alimentación externa (sistema a 5V).</i> .....	125
Figura 3.24 <i>Alimentación externa (sistema con interfase lógica a 3.3V).</i> .....	126
Figura 3.25 <i>Diagrama de tiempo para el ciclo de lectura.</i> .....	126
Figura 3.26 <i>Diagrama de tiempo para el ciclo de escritura.</i> .....	127
Figura 4.1 <i>Esquema inicial de conexión.</i> .....	130
Figura 4.2 <i>Esquema de conexión final.</i> .....	131
Figura 4.3 <i>Condensadores no polarizados</i> .....	132
Figura 4.4 <i>Condensadores polarizados</i> .....	132
Figura 4.5 <i>Reloj ideal y reloj con Jitter.</i> .....	134
Figura 4.6 <i>Análisis de resistencias equivalentes.</i> .....	135
Figura 4.7 <i>Esquema de conexión de puentes, para modificar la impedancia equivalente de línea.</i> .....	135
Figura 4.8 <i>Diagrama de la Solución.</i> .....	136
Figura 4.9 <i>Diagrama de Tiempo de la solución.</i> .....	137
Figura 4.10 <i>Diagrama eléctrico de la propuesta de mejora.</i> .....	138

Figura 5.1 <i>Implementación de la clase DataColector..</i>	144
Figura 5.2 <i>Implementación de un objeto tipo threading, derivado de su clase base.</i>	145
Figura 5.3 <i>Creación de un objeto hilo.</i>	146
Figura 5.4 <i>Código ejemplificando uso de la librería para crear objetos gráficos.</i>	147
Figura 5.5 <i>Creación de un objeto tipo ventana.</i>	147
Figura 5.6 <i>Creación de un objeto tipo ventana y su botón.</i>	148
Figura 5.7 <i>Esquema de framework usado para la implementación del software.</i>	149
Figura 5.8 <i>Esquema general de funcionamiento en la etapa de adquisición.</i>	150
Figura 5.9 <i>Lógica de la recolección de datos y determinación de mensajes.</i>	151
Figura 5.10 <i>Lógica de la descodificación de nivel dos.</i>	154
Figura 5.11 <i>Ejemplo de tablas de traducción implementadas para la descodificación ISUP.</i>	155
Figura 5.12 <i>Salida en pantalla esperada de la descodificación ISUP.</i>	156
Figura 5.13 <i>Lógica resumida de la decodificación ISUP.</i>	157
Figura 5.14. <i>Lógica de la extracción de los parámetros obligatorios fijos</i>	158
Figura 5.15 <i>Lógica para la extracción de punteros.</i>	159
Figura 5.16 <i>Lógica para la extracción de parámetros obligatorios variables.</i>	160
Figura 5.17 <i>Lógica para la extracción de parámetros facultativos.</i>	161
Figura 5.18 <i>Lógica general aproximada para la descodificación de los parámetros de un mensaje</i>	163
Figura 5.19 <i>Ventana de capturas de llamadas.</i>	164
Figura 5.20 <i>Lógica de descodificación para captura de llamadas.</i>	165
Figura 5.21 <i>Algoritmo para Rastreo de llamadas (Primera Parte).</i>	166
Figura 5.22 <i>Algoritmo para Rastreo de llamadas (Segunda Parte).</i>	167
Figura 5.23 <i>Captura de pantalla de Interfaz Grafica de Usuario.</i>	169
Figura 5.24 <i>Algoritmo para configuración de Interfaz Grafica de Usuario.</i>	170

## LISTA DE CUADROS

---

<i>Tabla</i>	<i>Página</i>
Tabla 1.1 <i>Características comunes de los sistemas primarios.</i> .....	17
Tabla 1.2 <i>Características del el sistemas de comunicaciones EI y TI.</i> .....	17
Tabla 1.3 <i>Bits de alineación de trama.</i> .....	21
Tabla 1.4 <i>Bits de NFAS.</i> .....	22
Tabla 1.5 <i>parámetros correspondientes a la mascara del pulso.</i> .....	30
Tabla 2.1 <i>Algunos valores que toma H0.</i> .....	73
Tabla 2.2 <i>Tipos de mensaje ISUP.</i> .....	81
Tabla 2.3 <i>Parámetros en los mensajes ISUP</i> .....	83
Tabla 2.3 <i>Parámetros en los mensajes ISUP (Continuación).</i> .....	84
Tabla 3.1 <i>Registro 1 de control de recepción (dirección=10 Hex)</i> .....	101
Tabla 3.2 <i>Criterio de Sincronización/ re sincronización.</i> .....	102
Tabla 3.3 <i>Registro 2 de control de recepción (dirección=11 Hex)</i> .....	102
Tabla 3.4 <i>Registro común de control (dirección =14 hex)</i> .....	103
Tabla 3.5 <i>Descripción de los pines del USB245M</i> .....	122
Tabla 3.6 <i>Tiempos máximos y mínimos para el ciclo de lectura.</i> .....	127
Tabla 3.7 <i>Tiempos máximos y mínimos para el ciclo de escritura.</i> .....	127
Tabla 5.1 <i>Atributos de la Lógica de la recolección de los datos</i> .....	152
Tabla 5.2 <i>Atributos de la Lógica de decodificación de nivel dos.</i> .....	153
Tabla 5.3 <i>Atributos de la Lógica de decodificación de parámetros de usuario ISUP.</i> ..	162
Tabla 5.4 <i>Atributos de la Lógica de decodificación en la captura de llamadas.</i> .....	168

## CAPITULO I

### GENERALIDADES E INFORMACION BASICA DE SEÑALIZACIÓN #7

#### 1.1 INTRODUCCION A LAS REDES DE SEÑALIZACIÓN

Sistema de señalización #7 es una arquitectura para desarrollar señalización fuera de banda (del termino en ingles: *Out of band*) en soporte del establecimiento de una llamada, construcción enrutamiento, e intercambio de información de las funciones de una red publica telefónica (PSTN). Esto identifica las funciones a ser desarrolladas por una red de sistema de señalización y un protocolo para habilitar su desarrollo.

##### 1.1.1 Concepto de señalización

La señalización se refiere al intercambio de información entre los componentes requeridos para proporcionar y mantener una llamada.

Como usuarios de la red pública telefónica, intercambiamos señalización con los elementos de la red todo el tiempo, Ejemplos de señalización entre el teléfono de un usuario y la red telefónica incluye: Marcación de dígitos, envío de tono de marcaje, Acceso a buzón de vos, envío de un tono de llamada en espera, etc.

C7 es un medio por el cual los elementos de una red telefónica intercambian información. La información es transportada en forma de mensajes.

El C7 es caracterizado por su alta velocidad, datos empaquetados y señalización fuera de banda.

##### 1.1.2 Tipos de señalización

La señalización se puede realizar, mas comúnmente, mediante dos tipos o Sistemas de Señalización: *Canal Asociado (ó CAS)* y *Canal Común (CCS # 7)*.

**Por Canal Asociado**, la información se transmite por el mismo canal por el que se está llevando a cabo la comunicación. Cada canal de vos tiene asociado un canal de vos exclusivo.

**Por Canal Común**, se refiere a cuando se establece un circuito independiente por el que se transmite toda la información relativa a varios grupos de abonado. La señalización viaja por un canal independiente del canal de vos ó datos, el mismo canal de señalización lleva la señalización de varios enlaces.



### **1.1.3 Señalización fuera de banda**

Señalización fuera de banda quiere decir que los datos de señalización no son colocados sobre el mismo camino de los datos de voz.

Estamos acostumbrados a pensar en señalización como señalización en banda. Escuchamos el tono de dial, marcamos dígitos, y escuchamos sonar sobre el mismo canal del mismo par de alambres. Cuando la llamada se completa, hablamos sobre al mismo trayecto que fue usado para la señalización. La Telefonía tradicional es usada para trabajar también en esta manera. La señalización colocada sobre una llamada entre un conmutador y otro, siempre se coloca sobre la misma línea que puede eventualmente llevar la llamada. La señalización toma la forma de una serie de tonos multi-frecuencia (MF). Muchos como pulsos de tonos marcados entre conmutadores.

La señalización fuera de banda establece un canal digital separado para el intercambio de información de señalización. Este canal es llamado enlace de señalización. Los enlaces de señalización son usados para llevar todos los mensajes de señalización necesarios entre nodos. Así, cuando una llamada es colocada, los dígitos marcados, la selección del trayecto, y otra información pertinente es enviada entre conmutadores usando sus enlaces de señalización, en lugar del camino principal el cual puede ultimadamente llevar la voz. Hoy en día, los enlaces de señalización llevan información a una velocidad de 56 o 64 kbps. Es interesante notar que mientras C7 es usada únicamente para señalización entre elementos de la red, el ISDN canal D extiende el concepto de señalización fuera de banda a la interfase entre el subscriber y el conmutador. Con el servicio ISDN, señalización que debe ser transportada entre el usuario y el conmutador local es llevado en un canal digital separado, llamado canal D. La voz o datos que comprenden la llamada son llevados en uno o mas canales B.

### **1.1.4 Objetivos de la señalización fuera de banda**

La señalización fuera de banda tiene varias ventajas que hacen a esta más deseable que la señalización tradicional en banda.

- Permite el transporte de mas datos a mayor velocidad (64 ó 56 kbps puede llevar datos mucho mas rápido que MF)
- Permite señalización en cualquier momento de la duración de la llamada, no únicamente en el comienzo.
- Habilita señalización para elementos de la red sin tener una línea de conexión directa.

## **1.2 LA INTERFAZ E1 2048 kbps**

La jerarquía plesiocrona digital (PDH) tiene dos sistemas primarios de comunicaciones así como su respectiva fundación, que se diferencian por la cantidad de “time slots” y por la velocidad de transmisión.

Están los sistemas basados en T1 sobre los 1544Kbit/s que esta recomendada por la ANSI y el sistema E1 basado sobre 2048Kbits/s que es recomendado por la ITU-TS. El sistema T1 es usado principalmente en USA, Canadá y Japón, los europeos y ciertos países no europeos utilizan el sistema E1, a continuación se detallan las características más importantes.

	<b>Características comunes</b>	<b>E1 y T1</b>
A	Frecuencia de muestreo	8kHz
B	Numero de muestras por señal telefónica	8000 por segundo
C	Largo de la trama PCM	$1/b = 1/8000/s = 125\mu s$
D	Numero de bits en cada palabra de código	8
E	Razón de bits del canal telefónico	$b \times d = 8000/s \times 8 \text{ bit} = 64kb/s$

Tabla 1.1 *Características comunes de los sistemas primarios.*

	<b>Característica diferentes</b>	<b>E1</b>	<b>T1</b>
F	Codificación/decodificación	Ley A	Ley $\mu$
	Numero de segmentos en características	13	15
G	Numero de timeslots por trama PCM	32	24
H	Numero de bits por trama PCM	$D \times g = 8 \times 32 = 256$ bits	$D \times g + 1^* = 8 \times 24 + 1^* = 193$ bits
I	Tamaño de un timeslot de 8 bits	$(C \times d)/h = (125\mu s \times 8)/256 = \text{aprox. } 3.9\mu s$	$(C \times d)/h = (125\mu s \times 8)/193 = \text{aprox. } 5.2\mu s$
K	Razón de bit de la señal multiplexado por división de tiempo	$B \times h = 8000/s \times 256 \text{ bits} = 2048kb/s$	$B \times h = 8000/s \times 193 \text{ bits} = 1544kb/s$

Tabla 1.2 *Características del el sistemas de comunicaciones E1 y T1.*

### 1.3 SEÑAL PCM

La señal analógica (palabra) de un teléfono primero es convertida a una señal de modulación por amplitud de pulso usando un proceso llamado muestreo.

Luego usando cuantificación y codificación esta muestra de la señal analógica (PAM) es convertida a una señal digital PCM.

### 1.3.1 Muestreo

La muestra es una medida periódica del valor de la señal análoga. Una señal muestreada contiene toda la información si la frecuencia de muestreo es al menos el doble de la frecuencia más alta de la señal a ser muestreada, el “teorema de muestreo de Shannon”. Como la señal analógica en telefonía es de banda limitada de 300 a 3400Hz, una frecuencia de muestreo de 8000Hz, una muestra cada 125μs, es suficiente.

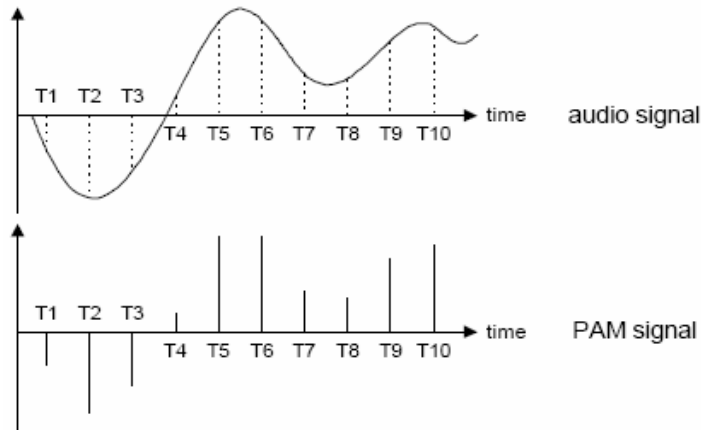


Figura 1.1 *Muestreo de la señal de voz*

### 1.3.2 Cuantificación y codificación

En el sistema E1, las señales PAM son cuantificadas usando una compresión característica de 13 segmentos conocida como la ley A, la cual es gobernada por la siguiente expresión

$$y = \frac{1 + \ln Ax}{1 + \ln A}$$

X = compresor de entrada de voltaje normalizado

Y = compresor de salida de voltaje normalizado

Esta característica está hecha de 7 segmentos de tamaño diferentes para las mitades positivas y negativas, con dos segmentos alrededor del punto cero formando unos segmentos simples de línea recta, cada segmento está dividido en pasos lineales con segmentos cerca del punto cero teniendo 32 pasos y el resto teniendo 16 pasos. Esto resulta en una cuantificación no lineal de la señal muestreada, la cual tiene un efecto útil sobre la razón ruido de cuantificación de la señal (S/Q).

Mucha de la información de la voz humana está a bajas amplitudes y los segmentos cercanos al punto cero cubren una sexta parte del rango de amplitud y es dividido en 32 pasos. Considerando que la mitad superior del rango dinámico de la señal de entrada es cubierta por los últimos 16 pasos del segmento de la característica.

Con los 128 pasos para las amplitudes positivas y negativas de la señal, hacen un total de 256 pasos que requieren 8 bits ( $2^8$ )

El bit mas significativo es el bit de signo y se pone en 1 para amplitudes positivas. Los siguientes 3 bits son utilizados para los segmentos con los últimos 4 para los 16 pasos en cada segmento en cada segmento.

Se puede ver que la razón S/Q aumenta la linealidad sobre los primero segmentos lineales de la característica de compresión, y luego aplanan el valor de 37 a 38dB el cual permanece prácticamente constante sobre el rango dinámico, en la practica esto significa que hay una razón S/Q constante sobre el rango de amplitudes para señales de voz de alrededor de 33dBm a -5dBm, resultando en una buena inteligibilidad de las señales de voz transmitidas de esta forma.

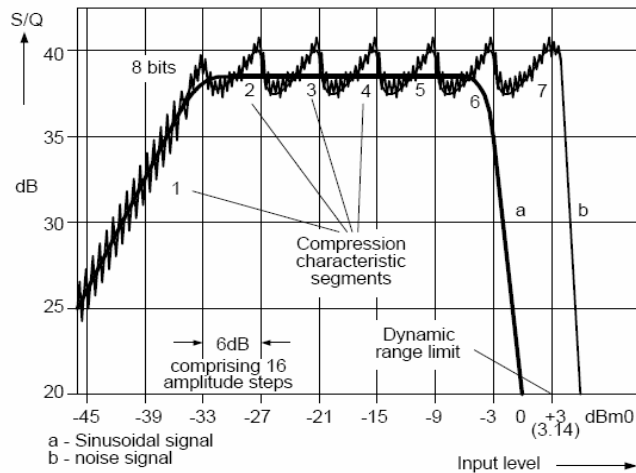
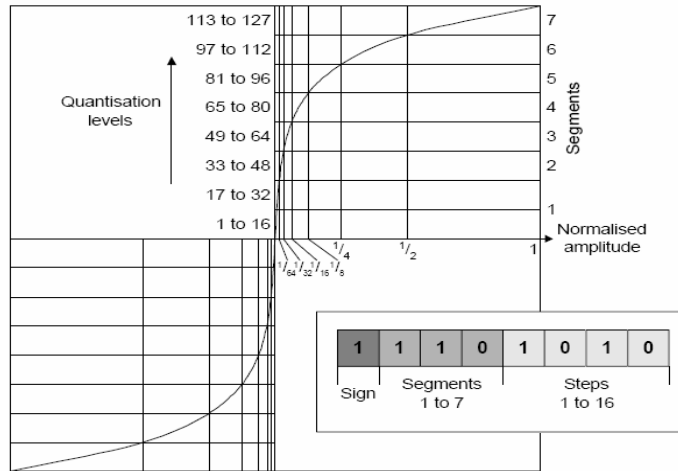
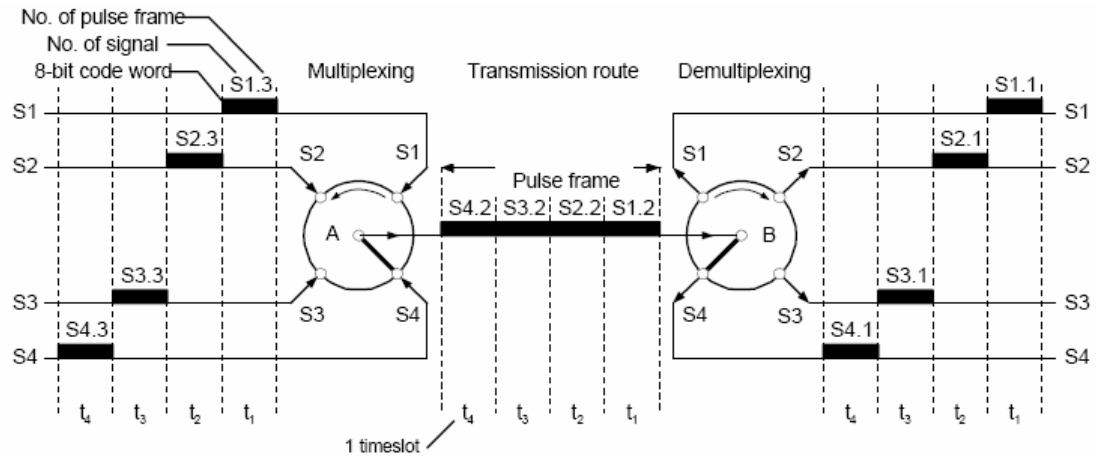


Figura 1.2 Niveles de cuantificación (Ley A) y característica ruido por cuantificación

### 1.3.3 PCM FRAMING

#### 1.3.3.1 Principios de multiplexación

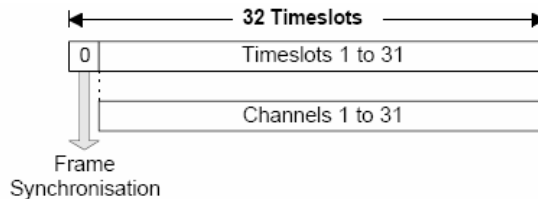


**Figura 1.3** *Multiplexación.*

La fuente de codificación produce palabras código de 8 bits a una razón de 8kHz por cada canal de voz, dando 64kbit/s la cual puede ser transmitida, entonces, para perfeccionar la utilización del medio de transmisión, las señales son transmitidas por multiplexación por división de tiempo, donde las palabras código son introducidas y contenidas en una trama de modulación de código por pulso (PCM), la figura anterior muestra el principio de la multiplexación por división de tiempo ilustrada por la transmisión de cuatro señales digitales.

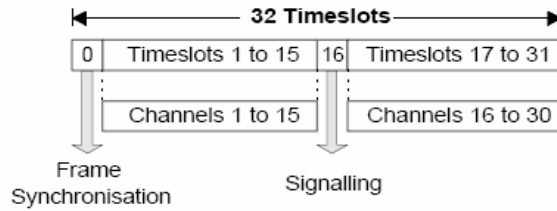
#### 1.3.3.2 La trama primaria

Una trama primaria consiste de 32 palabras código llamadas "Time Slots" y son numeradas del 0 al 31. Una trama PCM31 incluye 31 timeslots utilizados para tráfico y un time slot utilizado para sincronización



**Figura 1.4** *Trama primaria.*

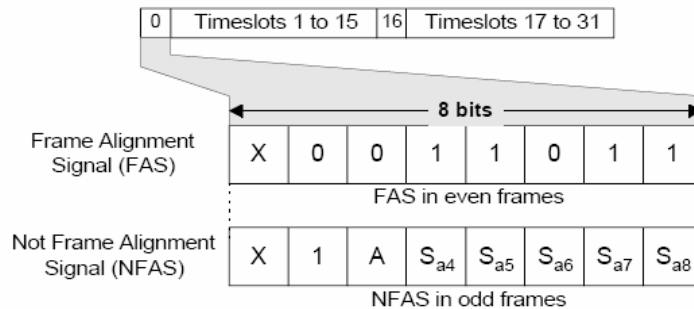
En un sistema PCM30 la trama comprende 30 timeslots utilizados para tráfico y dos palabras código que son utilizadas para sincronización y propósitos de señalización.



**Figura 1.5** Trama PCM30.

### 1.3.3.3 Alineación de trama

Los lados de transmisión y recepción son sincronizados a la trama PCM con la ayuda de la señal de alineación de trama (FAS) la cual es transmitida en el timeslot 0 de cada segunda trama, la señal de no alineación de trama (NFAS) es transmitida en el timeslot 0 de las tramas alternas.



**Figura 1.6** Señal de alineación de trama.

### 1.3.3.4 Señal de alineación de trama

Numero de bit	1	2	3	4	5	6	7	8
Valor binario	$S_i(C)$	0	0	1	1	0	1	1

Tabla 1.3 Bits de alineación de trama.

Bit 1:  $S_i$  esta reservado para uso internacional, en PCM30 o PCM31 o es utilizado para la transmisión del residuo de la división CRC en PCM30 o PCM31.

Bits 2 al 8: FAS

El lado receptor del sistema PCM determina los timeslots de la trama PCM sobre la base de las señales de alineación de trama, tanto que los bits recibidos pueden ser asignados a los varios canales en la secuencia correcta.

La FAS es transmitida en el timeslot 0 de cada trama PCM par, ejemplo las tramas número 0, 2, 4, 6 y demás, es siempre una palabra de 7 bits con la secuencia binaria 0011011 comenzado en el bit 2.

El bit 1 en el timeslot 0 es conocido como el bit  $S_i$  y esta reservado para uso internacional, es normalmente puesto a 1 excepto en los sistemas que usan CRC, aquí el residuo de la división la cual resulta de la comparación transmitida al lado receptor de la transmisión usando este bit.

Señal de no alineamiento de trama, La NFAS es usada para transportar información acerca del estatus del enlace y proveer señales de control para la velocidad de los multiplexores primarios

Numero de bit	1	2	3	4	5	6	8	9
Valor binario	$S_i$ (M)	1	A	$S_{a4}$	$S_{a5}$	$S_6$	$S_{a7}$	$S_{a8}$

Tabla 1.4 Bits de NFAS.

**Bit 1:**  $S_i$  esta reservado para uso internacional, en PCM30 o PCM31, o M es utilizado para transmisión de la señal CRC de alineación de multitrama en PCM30C o PCM31C

**Bit 2:** es puesta a uno y previene la simulación de la FAS.

**Bit 3:** es una muestra de la indicación de la alarma remota.

**Bits 4 a 8:**  $S_{a4}$  a  $S_{a8}$  son bits libres los cuales pueden se utilizados de la siguiente forma: Las recomendaciones de la ITU-T permiten que los bits  $S_{a4}$  a  $S_{a8}$  sean utilizados en aplicaciones específicas punto a punto (ejemplo para equipos transcodificadores) dentro de las fronteras nacionales. Cuando estos bits no son utilizados y los enlaces cruzan a una frontera internacional ellos deberían ponerse en 1.

El Bit  $S_{a4}$  puede ser utilizado como un enlace de datos basado en mensajes para monitoreo de operaciones, mantenimiento y rendimiento. Este canal se origina en el punto donde la trama es generada y termina donde la trama es dividida.

### 1.3.3.5 Sincronización de trama

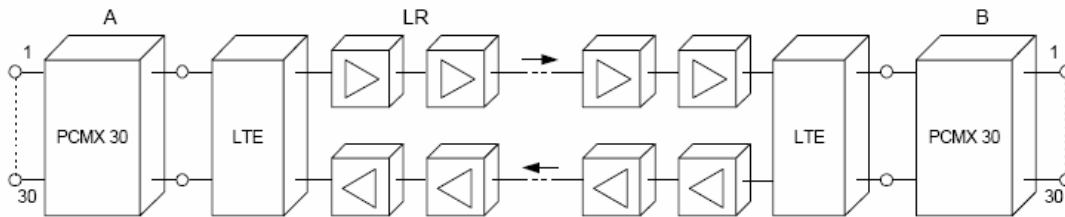


Figura 1.7 Diagrama de Sincronización de trama.

Considerando los multiplexores del diagrama anterior. El multiplexor PCM B sincronizara sobre el torrente entrante de bits del multiplexor A bajo las siguientes condiciones:

1. FAS correcto,  $S_1 0 0 1 1 0 1 1$ , es recibido en el time slot 0 de una trama.
2. El bit 2 en el time slot 0 (NFAS) de la siguiente trama recibida debe ser 1.
3. FAS,  $S_1 0 0 1 1 0 1 1$ , es recibido en el time slot 0 de la trama subsiguiente

El multiplexor es sincronizado en las tramas entrantes solamente si las tres condiciones son cumplidas.

### 1.3.4 Señalización R2

En los sistemas PCM30 y PCM30C el time slot 16 es utilizado por señalización de canal asociado (CAS). La información necesaria para la conmutación y transmisión de los 30 canales telefónicos (códigos de señalización y estatus) son introducidos y transmitidos es este timeslot.

El intercambio de señalización entre los multiplexores en los canales de directa e inversa toma lugar utilizando la compresión de 4 bits (a, b, c, d) de las señales de pulso las cuales están formadas por el equipo de señalización múltiplex de la señal origen en la central telefónica. Un ejemplo de un método de la señalización de la central telefónica y múltiplex (E & M).

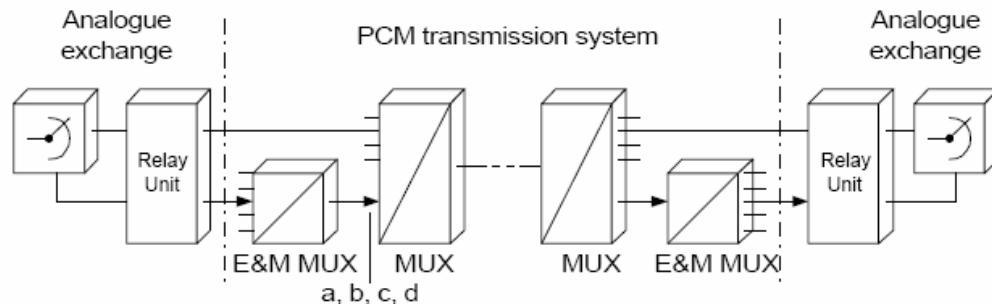


Figura 1.8 Señalización utilizando multiplexores E&M.

#### 1.3.4.1 Chequeo de redundancia cíclica (CRC)

Con la introducción del ISDN (red de servicios digitales integrados), los suscriptores son provistos con canales transparentes de 64kbps para locución o transmisión de datos, transparencia en este sentido significa que la señal binaria transmitida por el suscriptor es transmitida sobre el trayecto de la señal entera sin que sea alterada en ninguna forma por la conversión analógica/digital u otro medio, con el bit de integridad de secuencia preservado.



Hay un peligro con este tipo de comunicación de datos ya que el suscriptor puede voluntariamente o involuntariamente transmitir el patrón de bit 10011011 el cual corresponde a la FAS, esto puede indicarle al multiplexor PCM una re-sincronización por esta aparente FAS, con el resultado de que todos los canales PCM serán incorrectamente asignados.

Para evitar esta desastrosa anomalía del sistema, la recomendación G.704 especifica el uso del chequeo de redundancia cíclica CRC-4 para sistemas de 2048kbit/s, estos son conocidos en los sistemas PCM30C y PCM31C.

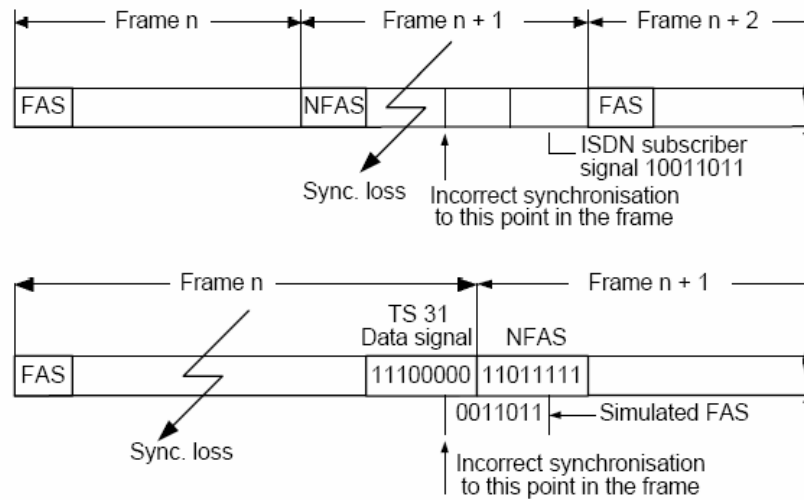


Figura 1.9 ejemplos de sincronización incorrecta.

### 1.3.4.2 Método CRC-4

Al lado transmisor del multiplexor PCM forma un bloque de chequeo CRC (bloque n) de 8 tramas PCM consecutivas, este bloque contiene 2048 bits (8 x 256 bits), el bloque de chequeo es multiplicado por  $x^4$  y luego dividido por el generador polinomial

$$x^4 + x + 1.$$

Ejemplo de cálculo del CRC-4

Data block n	Multiplier $x^4$	Generator polynomial $x^4 + x + 1$
10100000	:	10011
<u>10011</u>		
11100		
<u>10011</u>		
11110		
<u>10011</u>		
1101		
1101 Remainder (signature)		

El residuo del proceso de división, es llamado también la firma del sistema, y comprende 4 bits. Estos son escritos dentro del bit 1 en la señal de alineación de trama del siguiente bloque de datos ( $n + 1$ ) como los bits designados  $C_1, C_2, C_3, C_4$

Después de esto, el bloque de datos “n” es transmitido al lado receptor y de nuevo es sujeto a los procesos de multiplicación y división de la cual se obtiene un residuo de 4 bits.

Cuando el bloque  $n+1$  es transmitido, el residuo de la división de bloque de datos sobre el lado transmisor e también transmitido al lado receptor, donde es comparado con el residuo de la división del bloque de dato  $n$  del lado receptor, si los dos residuos son idénticos, no han ocurrido errores de bit durante el envío. Si hay una diferencia entre los dos residuos, solo puede significar que el bloque recibido ha sido degradado y uno o más bits fallaron durante la transmisión.

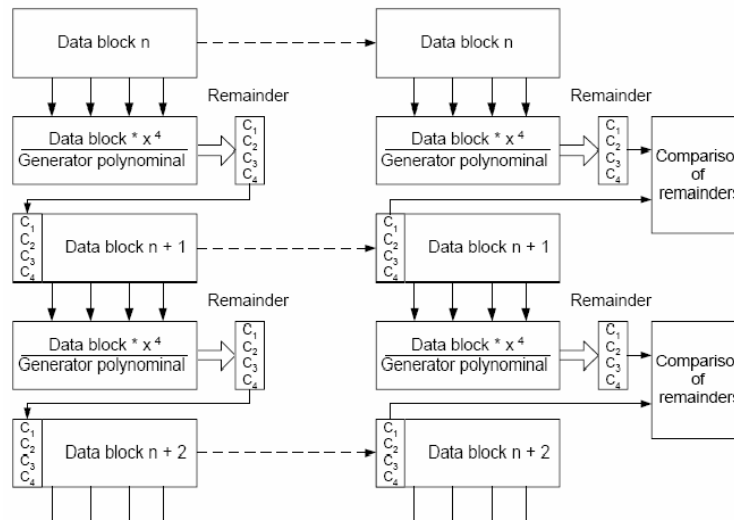


Figura 1.10 Diagrama de la transmisión de residuos en el método CRC-4

La transmisión de los cocientes requiere una capacidad que es obtenida haciendo uso del redundante bit 1 en la FAS de otra forma de cada trama impar, localizar los cuatro bits de chequeo  $C_1, C_2, C_3, C_4$  haciendo el cociente, una trama CRC es formada.

La multitrama CRC-4 consiste de 16 tramas PCM justo como la multitrama de señalización y por lo tanto tiene una duración de 2ms, esta multitrama es dividida en dos sub-multitramas de 8 tramas enmarcadas, I y II, una multitrama de señal de alineación CRC (CRC MFAS) es utilizada para sincronizar el lado receptor de esta multitrama.

La CRC MFAS es una señal de 6 bits de 001011 y es insertada bit a bit dentro del primer bit del NFAS en las tramas 1, 3, 5, 7, 9 y 11. Los primeros bits de la tramas 13 y 15 son llamados los bits E y son usados para indicar los bloques de datos con errores de bit que retornan al lado transmisor, si el bit E en la trama 13 = 0 indica que hay un error CRC en los datos contenidos en la sub-multitrama I y si el bit E en la trama 15 es 0 indica la

misma situación para la sub-multitrama II, estos bits son también llamados errores CRC remotos o distantes.

Como cada sub-multitrama CRC comprende 8 tramas PCM estándar es por lo tanto de  $8 \times 125\mu\text{s}$  de longitud, tanto que el sistema transporta 1000 CRC comparaciones por segundo. Cuando es comparado con el monitoreo de la FAS con los transportados en los sistemas sin CRC, el sistema CRC tiene la ventaja de un alto grado de certeza en la detección de errores posibles porque todo lo transmitido es monitoreado.

El sistema sin CRC monitorea solamente una pequeña parte de la señal, particularmente 7 bits por cada 505 bits. El método CRC, sin embargo, no detecta todos los posibles errores, un error múltiple en un bloque CRC puede significar la formación de una firma correcta, incluso si el bloque contiene errores, ya que el residuo CRC es una palabra de 4 bit, es decir que  $1/16$  o 6.25% de los bloques pueden contener errores, a pesar de una firma correcta, en otras palabras, la certeza con la cual un error puede ser detectado es de 97.75% del número total de errores.

El método CRC no puede determinar exactamente los errores simples así como no es posible decir cuantos errores fueron la causa de un residuo incorrecto en el “check sum” el resultado es por lo tanto un “mas grande que”, lo cual es suficiente, ya que el monitoreo continuo hace posible mantener una constante observación de la calidad de la transmisión

#### **Sincronización de trama (con CRC-4)**

Un sistema de transmisión utilizando un CRC-4 transporta 1000 comparaciones CRC cada segundo. Si el número de comparaciones negativas (incorrectas) excede el umbral de 914 en 1000 (91.4%) el sistema pierde sincronización, la re-sincronización toma lugar de la siguiente manera

1. Sincronización normal de el sistema PCM
  - a) señal de alineación correctamente recibida
  - b) el bit segundo en el NFAS debe ser 1
  - c) El siguiente FAS también debe ser recibida correctamente
  
2. Sincronización de una multitrama CRC  
El bit de posición 1 del NFAS contenido en las tramas de la multitrama CRC es chequeada para la señal de alineación multitrama de 00001011

La sincronización de la multitrama es alcanzada cuando al menos 2 CRC MFAS han sido correctamente recibidas dentro de un periodo de 8ms, (4 CRC multitramas). Entre estos dos CRC MFAS correctos debe haber 2ms o unos múltiplos de ello.

Solamente cuando las dos condiciones anteriores son cumplidas se puede decir que el sistema esta sincronizado y los cálculos CRC comiencen

## 1.3.5 ALARMAS

### 1.3.5.1 Alarmas remotas

Los multiplexores están conectados juntos a la transmisión PCM tomando lugar en ambas direcciones, y también se dice que los mensajes de alarma son transmitidos bi-direccionalmente

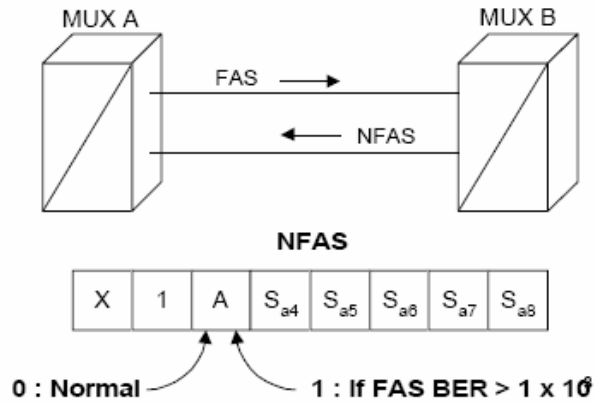


Figura 1.11 Indicación de alarma remota.

### 1.3.5.2 Indicación alarma remota

El NFAS es utilizado para transmitir información de servicio, el bit 3 del NFAS indica una alarma remota (o distante)

Si el bit 3:

Es 0 – significa operación ecuánime; no alarma.

Es 1 – significa que una de las siguientes situaciones de alarma ha ocurrido:

1. falla de fuente de alimentación
2. falla de código
3. falla de la señal entrante de 2048kbps
4. error de alineamiento de trama
5. razón de error de bit de la señal de alineamiento de trama  $> 1 \times 10^{-3}$

El multiplexor localizado en el punto B continuamente monitorea los FAS entrante por errores de bits. El FAS es recibido en el timeslot 0 o tramas alternas las cuales son cada 250 $\mu$ s o 4000 veces por segundo, si el resultado de la medida del error de bit del FAS es  $\leq 1 \times 10^{-3}$ , la transmisión es ecuánime. El NFAS transmitido de regreso al punto A será S<sub>i</sub> 1 0 1 1 1 1 1.

Cuando la razón de error de bit alcanza un valor más grande que  $1 \times 10^{-3}$ , la correcta operación del enlace de transmisión no es posible y el multiplexor de recepción se des-

sincroniza. Esto es indicado por puesta en 1 del bit A del NFAS lo cual resulta en una alarma, llamada indicación de alarma remota (RAI) o alarma distante. En este caso el NFAS transmitido de regreso al punto A es  $S_i$  1 1 1 1 1 1 1.

### **1.3.5.3 Señal de indicación de alarma (AIS)**

El multiplexor en el punto A registra esta alarma y luego detiene la transmisión normal de las señales de voz y datos y transmite una secuencia continua de 1s. Esto causa que el multiplexor en el punto B muestre una alarma AIS. Esta señal de 1s mantiene el mecanismo de recuperación de reloj en los regeneradores para que la re-sincronización sea intentada tan pronto como la razón de error de bit del FAS sea menor o igual a  $1 \times 10^{-3}$ .

La ITU-T define AIS como una secuencia de 509 1s en un bloque de 509 bit el cual es una señal conteniendo menos que 3 ceros en un periodo de 2 tramos. Una señal con todos los bits es estado 1 excepto por el FAS (00101 = 3 ceros) no es un AID valido y debería ser declarado como una perdida de sincronización de la trama.

### **Perdida de sincronización**

La pérdida de sincronización es declarada en los sistemas PCM30 (PCM31) si son recibidas tres palabras FAS incorrectas o en los sistemas PCM30C (PCM31C) si hay más que 914 errores CRC en un segundo

Perdida de sincronización en multitramas. Si la señalización MFAS es perdida luego la alarma de perdida de sincronización es declarada.

### **1.3.6 Códigos de línea**

Hay dos principales funciones de codificado de línea:

- para asegurar que hay suficiente información de cronometraje de la señal recibida
- para prevenir “caídas”

El repetidor o de-multiplexor deriva el reloj de muestreo de la señal entrante y la calidad de este reloj derivado depende del numero de transiciones de la señal entrante.

La caída es un efecto que puede ocurrir cuando la transportación de datos sobre el circuito tiene respuesta DC igual a cero.

Los niveles de voltaje nominal constante pueden elevarse o caerse debido a las cargas o descargas de la capacitancia de la línea de transmisión. Las caídas resultan en una ambigüedad de los niveles de señal y es conocido como ambigüedad de línea base

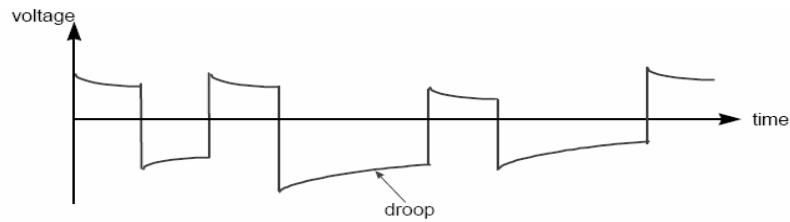


Figura 1.12 caídas en los niveles de voltaje de la señal.

Alguna onda bipolar “zero-mean” que tenga largas hileras de 1s o 0s puede resultar en caídas en los niveles de voltaje como se mostró anteriormente.

### 1.3.6.1 Código AMI

La inversión alterna de marca (AMI) es una codificación de línea que previene la ambigüedad de línea base, aquí la polaridad de la señal es cambiada cada 1 binario del mensaje transmitido

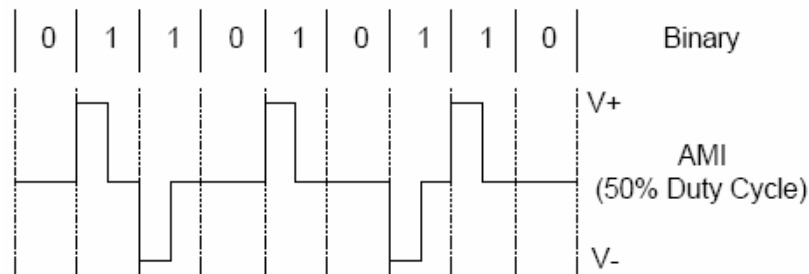


Figura 1.13 Tren con codificación AMI.

Así como la señal binaria tenga un suficiente número de 1s entonces el reloj de la señal que se deriva de ella será adecuada para el receptor para enganchar la velocidad de datos. Si hay cadenas largas de ceros entonces el receptor puede perder la sincronización de la señal de entrada.

### 1.3.6.2 Código HDB3

La plataforma HDB3 para código de alta densidad bipolar en el cual un máximo de tres ceros puede ocurrir en secuencia. Las siguientes reglas son utilizadas para convertir una señal binaria a una señal codificada en HDB3:

- Regla 1: si cuatro bits ocurren consecutivamente, el cuarto cero es reemplazado por un bit de violación o V-bit. El V-bit tiene la misma polaridad del anterior bit

1, el cual resulta en una violación de la regla AMI. La sustitución es por lo tanto: 0000 se vuelve 000V

- Regla 2: si hay un numero impar de bits 1 entre el bit de violación a ser insertado y violación previa, el primer cero de la secuencia de cuatro ceros es reemplazada por el llamado bit B. La sustitución es por lo tanto: 0000 se vuelve B00V

Esto tiende a un número no par de bits 1, el cual es necesario para asegurar que el bit de violación insertado tenga la polaridad opuesta del bit de violación, tanto que el código permanece libre de algún componente DC. Los bits de violación y bits B deben siempre ser insertados con polaridad alterna (AMI) para preservar la naturaleza libre de DC del código.

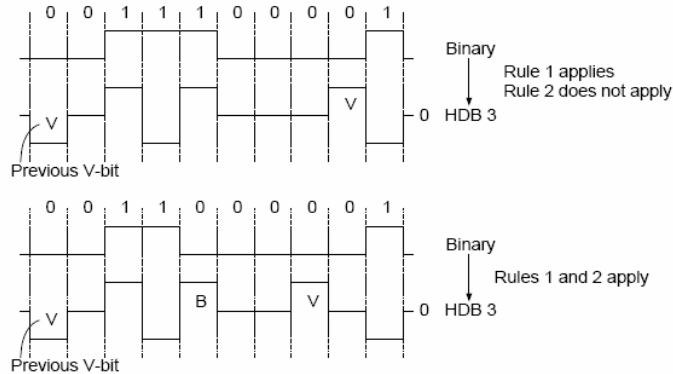


Figura 1.14 *Tren codificación HDB3.*

### 1.3.7 Recomendaciones ITU-T G.703 (PCM 2048Kbps)

Forma del pulso (nominalmente rectangular)	Todas las marcas de una señal valida deben estar conforme con la mascara respectiva de la señal. El valor V corresponde al valor pico nominal	
Par(es) en cada dirección	Un par coaxial	Un par simétrico
Prueba de impedancia de carga	75 ohms resistivos	120 ohms resistivos
Voltaje pico nominal de una marca (pulso)	2.37V	3V
Voltaje pico de un espacio (no pulso)	$0 \pm 0.273V$	$0 \pm 0.3V$
Ancho nominal de pulso	244ns	
Relación de las amplitudes negativas y positivas al centro del intervalo del pulso	0.95 z 1.05	
Relación de los anchos del pulso positivo y negativo nominal de la mitad de amplitud	0.95 a 1.05	

Tabla 1.5 *parámetros correspondientes a la mascara del pulso.*

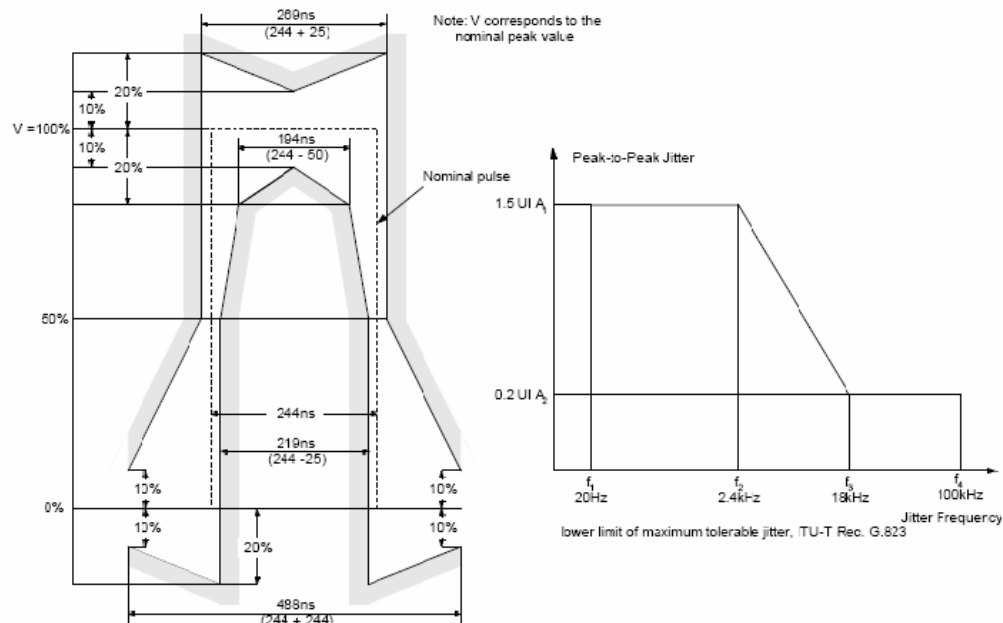


Figura 1.15 mascara del pulso del tren PCM.

## 1.4 CONCLUSIONES DEL CAPITULO I

En este capítulo se menciona el concepto de señalización y los tipos de señalización, las velocidades de transmisión en que se basan los sistemas de comunicación (T1 1.544Mbps en USA, Canadá y Japón y 2.048Mbps en Europa o otros no Europeos). La forma en que las señales de voz analógicas son muestreadas y codificadas en PCM. Las Muestras codificadas son enviadas a otras centrales telefónicas a través de Tramas primarias que a su vez se dividen en timeslots. Las tramas se dividen en 32 timeslots que se les llama también canales.

En señalización R2 se tienen 30 canales para voz y datos, un canal para alineación de trama y uno para señalización, en señalización #7 un solo canal puede llevar la señalización de varios enlaces por lo que se pueden tener 31 canales para voz y datos, y uno para alineación de trama.

Señalización # 7 proporciona muchas capacidades, usa una longitud variable de señales unitarias (con un máximo de longitud), proporcionando versatilidad y flexibilidad. La señalización es fuera de banda lo que significa que los datos de señalización no son transmitidos en el mismo timeslot de voz o datos.



Existen dos formas de re-sincronización de trama, la primera es recibir tres FAS consecutivas correctas y la otra es recibir FAS , NFAS y FAS para que el sistema se considere sincronizado.

El método CRC-4 es bastante efectivo para chequear la integridad de los mensajes, ya que los bits pueden sufrir degradación o caída de voltaje en el medio de transmisión o puede existir por parte del usuario la generación de la palabra de alineación de trama..

## **1.5 REFERENCIA BIBLIOGRÁFICA**

[1] Conde Jacobo, Alexander Vladimir, Rivas Preza David Alberto. “Diseño e implementación de un instrumento virtual de monitoreo de señalización CAS digital”. Universidad de El Salvador. 2004.

[2] CCITT. Introducción al sistema de señalización N.º 7 del CCITT, Rec. Q.700; Marzo/93.

[3] Tibbs, John. “E1 Pocket Guide. The World of E1”. Vo. 5. Wavetek Wandel Goltermann.

[4] UIT-T. “Características físicas y eléctricas de los interfaces digitales jerárquicos”. Rec. G.703. 1991.

[5] Guy Redimí. “SS7 Product Manager” Broocktrout Technology. 2001

[6] Calderón Osorio, Daniel Alberto. “Diseño de instrumentos virtuales para medir señalización siete por canal común de la Unión Internacional de Telecomunicaciones (CCS7UIT) en tiempo real”. Universidad de El Salvador. 2003.

[7] IEC. “Fundamentals of Telecommunications”. Web ProForum Tutorials. <http://www.iec.org>.

[8] ITU-T “Sistemas de comunicaciones PCM” sistema E1 basado sobre 2048Kbits/s.

[9] UIT-T. “Procedimientos de alineación de trama y de verificación por redundancia cíclica (VRC) relativos a las estructuras de trama básica definidas en la recomendación G.704”. Rec. G.706. 1991.

## CAPITULO II

### SISTEMA DE SEÑALIZACIÓN # 7

#### 2.1 ESTRUCTURA DE LA RED DE SEÑALIZACION

La red de señalización # 7 esta separada de la red de voz, y es usada para propósitos de conmutación de mensajes de datos los cuales permiten el establecimiento de conexiones telefónicas, llamadas y mantenimiento de la misma red de señalización. El método usado para transmitir mensajes a través de la red es el de conmutación de paquetes.

Los conmutadores telefónicos usados en muchas centrales telefónicas desarrollan una doble función. Además de conectar los circuitos de voz también deben realizar funciones de señalización. Esto es logrado a través de computadoras, que dando soporte a los conmutadores telefónicos, son conectadas a través de enlaces digitales a otras computadoras en la red.

Estas computadoras son llamadas *puntos de señalización (signal points)*. Estos puntos de señalización realizan tres funciones. Son generadores y receptores de todos los mensajes en la red y están localizados en las centrales telefónicas. Todos los mensajes son conmutados a través de la red usando puntos de transferencia (*transfer points*). Estos puntos de transferencia no originan mensajes y no siempre son el destino final de los mensajes; sin embargo son usados para realizar la conmutación de mensajes entre las centrales telefónicas. Otra función es la de proporcionar acceso a bases de datos.

Todos los nodos en la red de señalización # 7 son llamados *puntos de señalización*. Un punto de señalización tiene la habilidad de realizar discriminación de mensajes (lee la dirección y determina si el mensaje es para ese nodo), así como de encaminar mensajes hacia otro punto de señalización.

Los puntos de señalización son de tres tipos diferentes:

- Punto de Servicio Conmutación (Service Switching Point: SSP)
- Punto de Transferencia de Señalización (Signal Transfer Point: STP)
- Punto de Servicio de Control (Service Control Point: SCP)

Los puntos de señalización proporcionan acceso a la red de señalización # 7, proveen acceso a bases de datos ubicadas dentro y fuera de la red, y transfieren mensajes a otros puntos de señalización dentro de la red. Los puntos de señalización son instalados en pares dentro de la red para obtener redundancia y diversidad. El objetivo de esta redundancia es para asegurar que la red siempre opere, proporcionando caminos alternos

en el caso de fallas, asegurando que los mensajes pueden siempre alcanzar su destino final (ver *figura 2.1*).

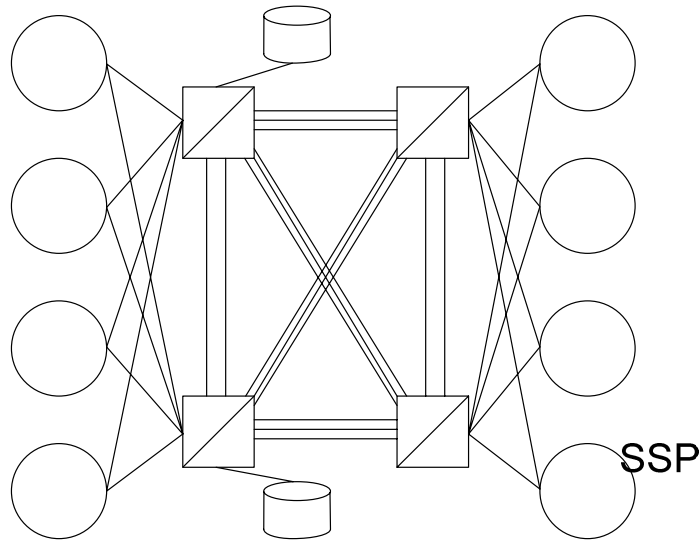


Figura 2.1 Red típica C7, con múltiples enlaces.

STP

Los enlaces que unen los puntos de señalización entre si son también colocados en pares. Estos enlaces de datos son típicamente DS0s (E1, T1, J1).

SSP

La red es desarrollada en diferentes ámbitos, o planos distintos. Hay un ámbito internacional, usando el estándar ITU-TS del protocolo de señalización # 7, y un ámbito nacional existente dentro de países en los cuales es desarrollado. Por ejemplo, en EE.UU. se desarrolla el estándar ANSI para el ámbito nacional. Todos los países son capaces de comunicarse entre si a través de *puertas de enlace* o *gateways* (por su traducción del inglés), los cuales convierten la versión nacional del protocolo de señalización # 7 a la versión internacional dicho protocolo. La conversión toma lugar en un SSP.

SSP

### 2.1.1 Punto de Servicio de Conmutación (SSP)

El punto de servicio de conmutación (SSP) es el medio de intercambio de señalización en las centrales telefónicas locales de la red de telefonía. Un SSP puede ser una combinación de un conmutador de circuitos de voz y un conmutador de mensajes de señalización # 7, o una computadora conectada al conmutador local de intercambio de circuitos de voz. Los SSP proveen la funcionalidad de comunicación con los conmutadores de voz y la creación de paquetes, o unidades de señalización, necesarias para transmisión en la red de señalización # 7 (ver *figura 2.2*).

STP

El SSP debe convertir la señalización de los conmutadores de voz en mensajes de señalización # 7, para poder ser enviados a otros puntos de intercambio a través de la red de señalización.

SSP

En el caso de acceso a una base de datos, el SSP puede enviar peticiones a la base de datos a través de la red C7 para encontrar sistemas localizados regionalmente en la red.

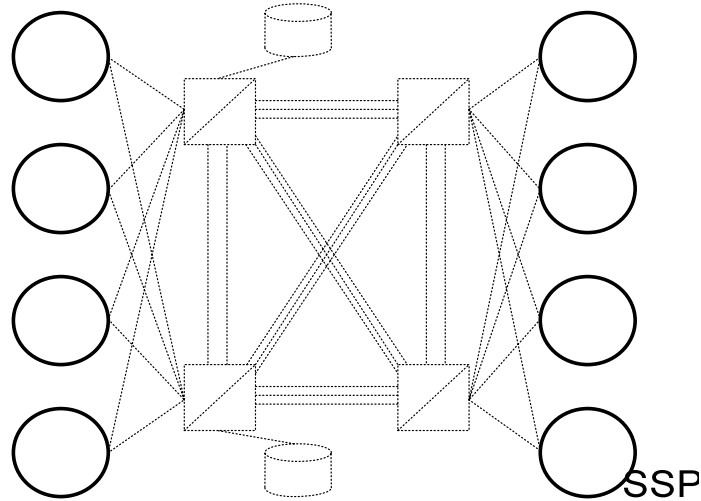


Figura 2.2 Relaciones del SSP en la red señalización #7.

La función del SSP es usar la información proporcionada por la parte llamante (tal como los números marcados) y determinar como conectar la llamada. Una tabla de asignación de rutas identificara cual circuito usar para conectar la llamada, y cual punto será su destino. Un mensaje de señalización # 7 debe ser enviado a un SSP de intercambio adyacente para pedir la conexión de un circuito en una ruta específico.

El nodo de intercambio adyacente concede los permisos para conectar este ruta, enviando un mensaje al nodo de intercambio adyacente que origino la petición. Usando la información de la parte llamante, proporcionada en el mensaje, el nodo de intercambio adyacente puede determinar como conectar la llamada a su destino final. Esto puede requerir varias conexiones entre nodos de intercambio adyacentes. La función del SSP es administrar estas conexiones hasta que el destino final es alcanzado.

### 2.1.2 Punto de Transferencia de Señalización (STP)

Todos los paquetes en señalización # 7 viajan de un SSP a otro, a través de los servicios de un *punto de transferencia de señalización* (STP). El STP sirve como un encaminador en las redes de señalización # 7. Usualmente los mensajes no son generados por un STP. El STP conmuta los mensajes recibidos de varios SSPs a través de la red, a su destino final (ver figura 2.3).

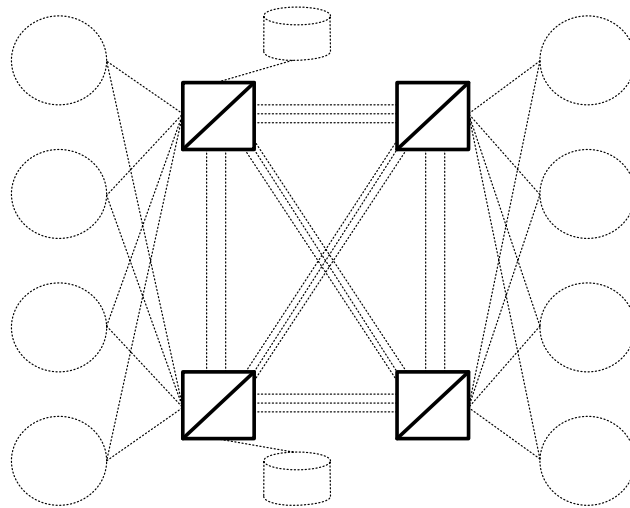


Figura 2.3 Relaciones del STPs en las redes de señalización # 7.

STP

Hay tres niveles de STPs:

- Punto de transferencia de señalización nacional.
- Punto de transferencia de señalización internacional.
- Puerta de Enlace de transferencia de señalización.

El STP nacional existe dentro de una red nacional y es capaz de transferir mensajes usando el mismo estándar o protocolo nacional. Los mensajes pueden ser pasados a otro nivel de STP, el STP internacional, pero el STP nacional no tiene la capacidad de convertir mensajes a otra versión o formato. La conversión entre protocolos es frecuentemente usada para interconectar el STP nacional con el STP internacional, proporcionando la conversión de un estándar nacional (tal como ITU-TS) a un estándar internacional (ANSI).

STP

Las funciones del STP internacional son las mismas que el STP nacional, pero es usada en las redes internacionales. Estas proporcionan interconexión entre todos los países que usan un protocolo nacional al estándar ITU-TS. Esto asegura la inter-conectividad mundial de redes. Todos los nodos conectados al STP internacional, deben usar el protocolo estándar ITU-TS.

La puerta de enlace STP (*gateway STP*) proporciona la conversión de un estándar nacional al estándar ITU-TS, o algún otro estándar nacional. La puerta de enlace STP se utiliza a generalmente como acceso a la red internacional, proporcionando acceso y conversión del mensaje al protocolo estándar ITU-TS.

Otra característica de los STP es la capacidad de efectuar mediciones. Las mediciones pueden ser divididas en dos funciones básicas: *mediciones de tráfico* y *mediciones de uso*. Las mediciones de tráfico proporcionan un conteo fijo e información estadística sin importar el tipo de mensaje que entra y sale de la red. Para propósitos de mantenimiento,

los eventos en la red son registrados (tal como el tiempo en que un enlace permanece fuera de servicio, el tiempo en que un procesador permanece ocupado, etc.).

Por medio del monitoreo de mensajes de señalización #7 enviados a través de la red, los computadores pueden determinar cuando una llamada esta siendo conectada a un número específico de un suscriptor, y determinar la duración de la llamada. Usando la red de señalización # 7 en lugar de las centrales telefónicas, como fuente de los registros de facturación, las actividades de facturación pueden ser centralizadas dentro de la red, proporcionando mayor eficiencia y ahorro de costos a las compañías telefónicas.

Otra función importante del STP es la de proporcionar acceso a las bases de datos por medio de los SCP. Estos últimos prestan servicios tales como tarjetas de llamada, asignación de rutas a números 800, y servicios de identificación de llamada.

El STP es la entidad de señalización # 7 más versátil, proporcionando un amplio conjunto de servicios a los usuarios de la red. No importa si estos sean servicios de conversión de protocolo o funciones de enrutamiento, el STP es el componente de mayor uso en la red, y el vehículo para entregar nuevos servicios a los suscriptores.

### 2.1.3 Punto de control de servicio (SCP)

El punto de control de servicio (SCP) sirve como una interfase para las bases de datos de las compañías telefónicas. Estas bases de datos son usadas para guardar información acerca de los servicios proporcionados a los suscriptores, encaminamiento de números de servicio especial (tal como números 800 y 900), validación de llamadas de tarjeta y protección de fraude.

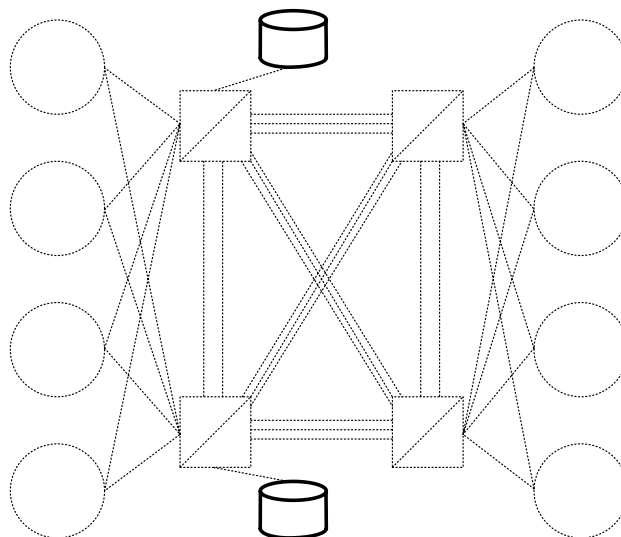


Figura 2.4 Relaciones del SCP en la red de señalización #7.

El SCP es usualmente una computadora usada como una interfase al sistema de base de datos. Algunas aplicaciones nuevas de los SCP son implementados en los STPs, proporcionando una solución integrada. En todos los casos, la dirección de un SCP es un punto de código (*point code*), mientras la dirección de una base de datos es un número de subsistema.

El SCP puede desarrollar conversión de protocolos de señalización #7 a X.25, o puede proveer la capacidad de comunicación directa con la base de datos de la computadora, a través del uso de *primitivas*. Una primitiva es una interfase que provee acceso de un nivel del protocolo a otro nivel.

El tipo de base de datos depende de la red. Cada proveedor de servicio tiene diferentes requerimientos y sus bases de datos pueden diferir. Las bases de datos mas comúnmente usadas son listadas a continuación:

- Base de datos de servicio de administración de llamadas (CMSDB)
- Probabilidad del numero local (LNP)
- Base de datos de información de línea (LIDB)
- Parte llamante (CNAM)
- Base de datos de servicios de negocios (BSDB)
- Registro de localización de Residencia (HLR)
- Registro de localización de visitante (VLR)

Cada base de datos contiene información para una aplicación específica, estas también dan una dirección llamada Número de subsistema (*Subsystem Number*), para uso de solicitudes de encaminamiento de los (SSP) a través de la red de señalización #7 hacia la base de datos actual.

#### **2.1.4 Enlaces de datos de señalización**

Todos los puntos de señalización #7 son interconectados vía enlaces de señalización de datos. Estos enlaces de datos poseen tasas de transferencia de datos de 56/64Kbps, 10/100Mbps y 1.536Mbps (la excepción a esta regla es Japón, donde los enlaces usados son a 4.8 Kbps).

Un *Enlace de Datos de Señalización* es un trayecto de transmisión bi-direccional para la señalización, compuesto de dos *canales de datos* que funcionan conjuntamente en sentidos opuestos de transmisión a la misma velocidad de datos. Constituye el nivel funcional más bajo (nivel 1) de la jerarquía funcional del sistema de señalización #7.

De los 32 canales del sistema a 2.048 Mb/s, uno se usa como siempre para sincronismo y otro se constituye en el *Signaling Data Link* (SDL). El time slot usado para el SDL por costumbre suele ser el # 16; pero en general puede usarse cualquier otro (salvo el time slot # 0 el cual es usado para sincronismo).

Existen tres modos de señalización que pueden ser usados:

**Señalización asociada**, este enlace esta directamente en paralelo con la conexión de voz por la cual se proporciona señalización.

**La señalización no asociada**, utiliza una trayectoria lógica separada de la voz actual. Hay múltiples nodos involucrados para alcanzar el destino final, mientras la voz puede estar en una trayectoria directa hacia el destino.

**La señalización cuasi-asociada**, utiliza un número mínimo de nodos para alcanzar el destino final. Este es el mejor método de señalización, porque cada nodo introduce retardo adicional en la entrega de la señalización.

Este último es el método más favorable para la señalización, y es por eso que las redes de señalización #7 utilizan dicho método.

Los enlaces de datos de señalización son etiquetados de acuerdo a su función. No existe una diferencia física entre los enlaces, la diferencia esta únicamente en la manera en que son utilizados durante la transferencia de mensajes y como interactúa la administración de la red con los enlaces (ver *figura 2.5*).

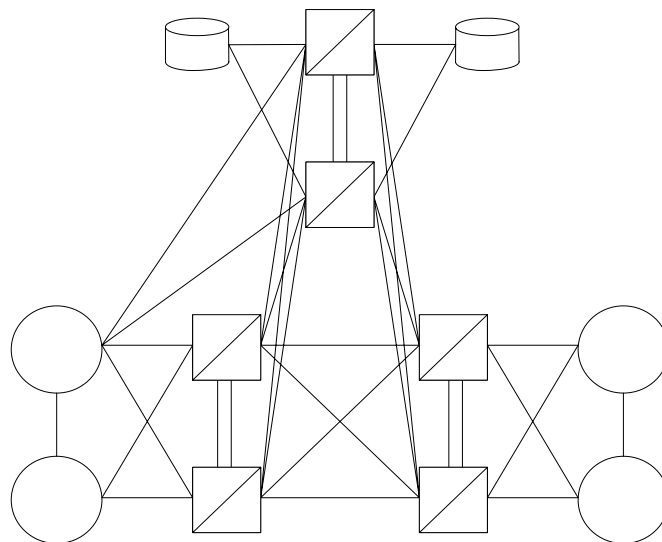


Figura 2.5 *Diferentes tipos de enlaces de acuerdo a su disposición en la red.*

Los enlaces son colocados en grupos llamados *linkset*. Todos los enlaces dentro de un *linkset* deben de pertenecer al mismo nodo adyacente. El equipo de conmutación alternara la transmisión a través de todos los enlaces en un *linkset* para asegurarse de balancear la carga. Pueden ser asignados a un *linkset* hasta 16 enlaces.



Hay seis diferentes tipos de enlaces utilizados en señalización #7:

**Enlaces de acceso (A):**

Son usados entre el SSP y el STP, o SCP y STP. Estos enlaces proveen acceso a la red y a las bases de datos a través del STP. Hay siempre por lo menos dos enlaces A, uno para cada par de STP locales.

**Enlaces de puente (B):**

Son usados para conectar parejas de STPs a otras parejas de STPs al mismo nivel jerárquico. Los enlaces de puente son desplegados en grupos de cuatro.

**Enlaces cruzados (C):**

Conecta un STP a su pareja STP. Estos siempre son desplegados en pares, para mantener redundancia en la red. El tráfico normal en señalización #7 no es asignado a estos enlaces, excepto en condiciones de congestión. Los únicos mensajes que viajan entre los pares de STP durante condiciones normales son los mensajes de administración de red.

**Enlaces diagonales (D):**

Son usados para conectar pares de STPs de un nivel primario a otro par de STPs de nivel secundario en la jerarquía de puntos de señalización.

**Enlaces extendidos (E):**

Son usados para conectar pares de STPs remotos desde un SSP. Los enlaces extendidos alternan la ruta de los mensajes de señalización #7 en el caso que ocurra una congestión dentro de los pares de STP.

**Enlaces totalmente asociados (F):**

Son usados cuando pueda existir una gran cantidad de tráfico entre los STPs, o cuando un SSP no puede ser conectado directamente a un STP. Estos enlaces permiten a los SSPs usar el protocolo y acceder a las bases de datos de señalización #7 incluso cuando no es económico proporcionar una conversión directa a un par de STP.

### **2.1.5 Rutas**

Un punto de señalización debe definir *rutas*. Una ruta es un conjunto de enlaces (*linkset*) utilizados para alcanzar un destino en particular. Un conjunto de enlaces puede pertenecer a más de una ruta. Una colección de rutas es conocida como conjunto de rutas (*routeset*).

Un conjunto de rutas es asignado a un destino. Los conjuntos de rutas son necesarios porque, si existiese una sola ruta y esta no estuviera disponible, no se puede tomar una ruta alterna y la señalización no podría ser enviada a su destino. Un conjunto de rutas proporciona rutas alternas al mismo destino en el caso de que cualquier ruta no este disponible.

Un destino es una dirección introducida en la tabla de encaminamiento de un punto de señalización remoto

### **2.1.6 Point Codes.**

A fin de trasladar la información de señalización de un punto a otro dentro de la red, la ITU-T ha asignado una dirección única o *Point code* a cada nodo dentro de la red Internacional. Dos puntos dentro de la red Internacional NO pueden tener la misma dirección (lo mismo aplica en redes nacionales).

Cada mensaje de señalización incluye explícitamente el *Point Code de Origen* y el *Point Code Destino*, mas comúnmente conocidos por sus acrónimos en inglés:

- OPC ó Originating Point Code ,
- DPC ó Destination Point Code.

Un Point Code que es adyacente a otro Point Code dentro de la red, regularmente se conoce como APC. A veces resulta que un APC y un DPC son el mismo punto; pero en general no es así.

### **2.1.7 Rendimiento del enlace**

Los enlaces deben permanecer disponibles para el tráfico de señalización #7 todo el tiempo. Cuando un enlace falla, los otros enlaces dentro del grupo deben tomar el trafico, así mismo, si una entidad de señalización (tal como un STP) falla, su pareja debe asumir la carga. Esto significa que los enlaces pueden de repente ser cargados con más tráfico del que pueden manejar. Por esta razón las entidades de señalización #7 son diseñadas para enviar menos del 40% del tráfico en cualquier enlace.

### **2.1.8 Interfase de enlace físico**

Los enlaces de datos de señalización son conectados al equipo de red usando interfaces eléctricas. Estas interfaces están estandarizadas, y definidas por grupos tales como ITU-TS y EIA. El tipo de interfase puede depender del tipo de equipo usado con el enlace. Las interfaces operan al nivel uno del protocolo de señalización #7, y proporcionan el medio eléctrico óptico para transmitir paquetes de datos dentro de la red. A continuación se describen las interfaces mas comúnmente usadas en C7.

## V.35

Esta interfase es comúnmente usada desde la unidad de servicio de datos hacia el punto de señalización #7. La interfase V.35 puede también ser usada por un panel de conectores digitales (DSX). V.35 provee velocidades de transferencia de 56 o 64 kbps. Velocidades de datos más lentas son también soportadas.

### ***DS0A (señal digital 0)***

Este es un canal a 56/64 kbps localizado en un DS1 (o mayor). El enlace puede ser transportado con un circuito existente DS1/DS3 entre centrales telefónicas, con la condición de que uno de los DS0As este dedicado a señalización #7.

La interfase DS0A proporciona transferencia segura de datos digitales. No hay opciones de configuración que sean de preocupación para los circuitos DS0, excepto por los esquemas de codificación y la velocidad de los datos.

Cuando se utilizan enlaces DS0A, el problema que se encuentra con más frecuencia es la pérdida de sincronización de reloj, causando que el enlace se coloque fuera de servicio.

El V.35 y del DSA0 son las interfaces mas utilizadas para enlaces de conexión a los nodos de red, junto con el TCP/IP que esta ganando popularidad. Hay otras interfaces utilizadas para interconexión de terminales y equipo de comunicación tales como: RS-232, V.24 y el RS-449.

## 2.2 MODELO EQUIVALENTE DEL C7

El protocolo SS7 difiere un poco del modelo OSI (*figura 2.6*). Mientras que el modelo OSI consiste de siete capas diferentes, el estándar SS7 utiliza solamente 4 niveles. El termino “nivel” es utilizado en el mismo contexto que “capa”.

Las funciones ejecutadas por estos cuatro niveles corresponden con las siete capas del modelo OSI. Algunas de estas funciones utilizadas por el modelo OSI no tienen propósito en las redes SS7 y por lo tanto no están definidas.

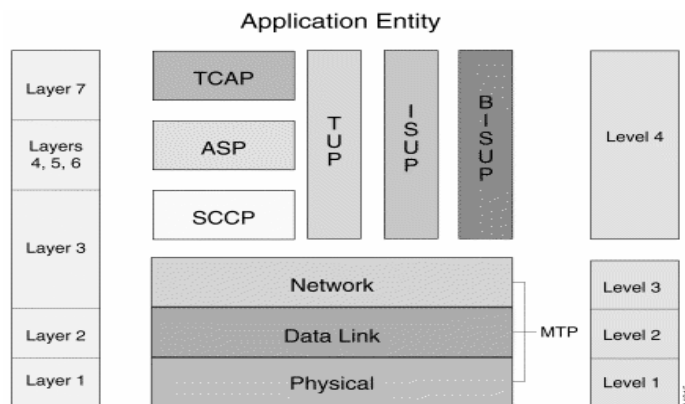


Figura 2.6 Modelo de pila SS7

### 2.2.1 Nivel uno – capa física

La capa física en sistema de señalización #7 es virtualmente la misma que la del modelo OSI, el modelo OSI no especifica una interfase a utilizar, ya que esta diferirá de red a red. El sistema de señalización #7, puede utilizar una interfase específica.

El estándar de señalización #7 no especifica alguna interfase a ser utilizada, de hecho, el estándar permite al protocolo utilizar cualquier interfase a cualquier velocidad. La teoría es que el protocolo debería estar habilitado para utilizar algún tipo de interfase y algún tipo de medio, manteniendo verdadera transparencia a través de las capas. Los otros factores, son distancia y velocidad de transmisión necesitadas para soportar el tráfico mezclado en cada red.

### 2.2.2 Nivel dos – nivel de enlace de datos

El nivel de enlace de datos de la pila del protocolo de señalización #7 proporciona a la red funciones de detección/corrección de error y entrega secuenciada para todos los paquetes de mensaje de señalización #7. Como en el modelo OSI, este nivel está dedicado únicamente a la transmisión de datos de un nodo al siguiente nodo en la red, no está dedicado a que los datos alcancen el destino final del mensaje. Como el mensaje viaja de nodo a nodo, cada nodo examina el dígito marcado (contenido en el nivel 4) y utiliza esa información para determinar la siguiente ruta para el mensaje.

La secuencia numérica es utilizada por esta capa para determinar si algún mensaje se ha perdido durante la transmisión. Un mensaje perdido indica un error, el cual es contado por un contador de errores mantenido por el nivel tres. Después de una cantidad significativa de errores, el enlace es tomado y llevado fuera de servicio y la red comienza con los procedimientos de diagnóstico y recuperación.

Otra función de chequeo de error mantenida en el nivel dos es la *secuencia de chequeo de trama*, el SS7 utiliza CRC-16 para el chequeo de error de los datos de usuario. El propósito de este mecanismo es mantener la integridad de los datos. El flujo de bits está sujeto a la ecuación CRC-16, y el residuo es colocado dentro del campo FCS (*Frame Check Sequence*). Cuando el mensaje es recibido por el nodo distante, la misma ecuación es evaluada de nuevo, solo en este único momento el valor es comparado con el valor en el campo FCS del mensaje recibido.

Si existe error en el mensaje, o un mensaje se ha perdido, el nivel dos es responsable para pedir su retransmisión.

Un indicador de longitud es proporcionado para permitir que el nivel dos determine que tipo de paquetes (unidades de señalización) está recibiendo. El nivel dos debe conocer el tipo de unidad de señalización que está siendo recibido así como saber cómo procesar el

mensaje. Si determina que hay información deseada por las capas superiores, entonces este nivel pasara el contenido del mensaje al nivel de red, o nivel tres.

### 2.2.3 Nivel tres – nivel de red

Este nivel de red proporciona tres funciones: *discriminación de mensaje*, *encaminamiento* y *distribución de mensajes*. Todas las funciones dependen de los servicios del nivel dos. Cuando un mensaje es recibido, es pasado por el nivel dos hacia el nivel tres para discriminación de mensajes.

**La discriminación de mensajes**, determina la dirección del mensaje. Si el mensaje contiene la dirección local (del nodo receptor), entonces el mensaje es pasado a la distribución de mensaje. Si el mensaje no esta enviado al nodo local, entonces es pasado a la función de encaminamiento de mensaje.

**La función de encaminamiento de mensaje**, lee las direcciones de la llamada y la parte llamante para determinar cual dirección física va a encaminar. Las direcciones de la parte llamada y la parte llamante son consideradas direcciones lógicas, y la dirección física la dirección del nodo. La dirección física en redes de señalización #7 se conoce como un *point code*. Cada nodo en la red debe tener un único *point code*. La función de enrutamiento determina a cual *point code* encaminar el mensaje basado en la información guardada en sus tablas de encaminamiento. El *point code* en muchos casos no es el destino final para un mensaje, pero si el *point code* adyacente para este nodo. Esto permite a los mensajes ser encaminados a través de la red y re-encaminados en el evento de una falla de red a otro nodo. El esquema de encaminamiento es determinado por los proveedores de red y puede variar dependiendo de la filosofía.

**La distribución de mensajes**, es utilizada cuando una discriminación de mensaje determina que la dirección es local. La distribución de mensajes es responsable por identificar a cual parte de mensaje usuario es enviado (basado en el octeto de la información del servicio del campo del mensaje) y encamina el mensaje a su usuario interno (un protocolo del conjunto de protocolos de señalización #7).

Hay tres funciones de administración de red en este nivel: *Administración de enlace*, *administración de encaminamiento*, y *administración de tráfico*.

**La administración de enlace**, utiliza la unidad de señalización de enlace de datos (LSSU) para notificar a los nodos adyacentes de los problemas de enlace. Un problema de enlace no necesariamente significa que el enlace no puede transmitir mensaje. Los errores de software o problemas de procesador sobre las tarjetas de interfase de enlace pueden causar que el enlace se vuelva inútil.

**La administración de encaminamiento**, proporciona el mecanismo para re-encaminamiento de tráfico alrededor de los nodos que han fallado o se han

congestionados. Esta es una función de nivel tres y trabaja con la función de administración de enlace.

***La administración de tráfico***, mecanismo que permite a la red controlar el flujo de ciertos mensajes basados en el protocolo, sin impedir otro tráfico que no debería de ser afectado.

#### **2.2.4 Nivel cuatro – parte usuaria**

El nivel 4 en las redes de señalización #7 consiste de varios protocolos diferentes, todos llamados partes de usuario y la parte de aplicación. Para una conexión y desconexión de llamada básica telefónica, son utilizados la parte usuaria telefónica (TUP) o protocolos ISUP.

Para acceder a las bases de datos de red, es utilizado el protocolo TCAP. El TCAP soporta las funciones requeridas para conectar a una base de datos externa, desarrollar una consulta a la base de datos, y recuperar información. La información o datos recuperados son enviados de regreso en forma de un mensaje TCAP al punto de señalización que lo solicito. El TCAP soporta control remoto de otras entidades en la red. Una conmutador de red puede invocar una característica o una función en otro conmutador enviando un mensaje TCAP de una entidad a otra

### **2.3 UNIDADES DE SEÑALIZACION**

El sistema de señalización # 7 es una red basada en la conmutación de paquetes por lo que los datos transmitidos son agrupados en paquetes. Un paquete contiene toda la información necesaria para encaminar datos a través de la red, sin la necesidad de establecer una conexión con el punto de destino.

#### **2.3.1 Generalidades de conmutación en la red.**

Hay tres métodos básicos para la conmutación en una red: *conmutación de circuitos*, *conmutación de mensajes* y *conmutación de paquetes*:

##### ***Conmutación de circuitos:***

Utiliza una conexión física entre dos puntos para la transmisión de los datos. El circuito permanece conectado hasta que ambos puntos han terminado la transmisión. Un buen ejemplo de redes conmutación de circuitos es la *red publica telefónica conmutada* (PSTN por sus siglas en ingles).

### ***Conmutación de mensajes:***

Usa una estructura de mensajes para encaminar los datos a través de la red. Los datos son acompañados por una dirección y un mensaje. Los datos son enviados en su totalidad y no incluye ningún sistema de detección de errores o control del flujo de datos.

### ***Conmutación de paquetes:***

Ordena todos los datos en un paquete o un grupo de paquetes y los transmite en la forma de un paquete completo, proporcionando toda la información necesaria para encaminar y procesar los datos recibidos. Se incluye en la redes de conmutación de paquetes procesos de administración de red, detección y corrección de errores.

La conmutación de paquetes es la forma más eficiente para la transmisión de datos en una red. Usualmente estas redes usan diferentes tipos de paquetes dependiendo de la función a realizar.

Las aplicaciones de las redes de señalización # 7 pueden ser de dos tipos: *aplicaciones relacionadas con circuitos* y *aplicaciones no relacionadas a circuitos*.

#### ***Aplicaciones relacionadas con circuitos.***

Están directamente relacionadas con la conexión y desconexión de circuitos telefónicos usados para la comunicación entre abonados. Estos circuitos están localizados en una red distinta de la red de señalización y son usados únicamente para conectar dos suscriptores. La red de señalización identifica, en estos circuitos, el tipo de transmisión entre voz y datos además de diferentes parámetros para dicha transmisión (por ejemplo, velocidades de transferencia de datos, métodos de descodificación usados en las interfaces de voz).

#### ***Aplicaciones no relacionadas a circuitos.***

Estas constituyen todo el otro tipo de tráfico en las redes de señalización # 7. Para dar soporte a las aplicaciones relacionadas con circuitos de la PSTN, los conmutadores telefónicos deben de ser capaces de comunicarse entre ellos. Este tráfico debe ser soportado por la red de señalización # 7, la cual presta dicho servicio.

### **2.3.2 Estructura de paquetes.**

El sistema de señalización # 7 usa tres diferentes estructuras para los paquetes, llamadas *unidades de señalización* (SU por sus siglas en ingles), proporcionando tres diferentes niveles de servicio. El protocolo de señalización # 7 usa estas tres unidades de señalización para la transmisión de información sobre administración de red. La información es enviada usando solo un tipo de estas tres unidades.

Las tres unidades de señalización son:

- Unidad de señalización de relleno (fill-in signal units, FISU).
- Unidad de señalización del estado del enlace (link status signal units, LSSU).
- Unidad de señalización de mensajes (message signal units, MSU).

Estas son continuamente transmitidas en ambas direcciones en un enlace que esta en servicio entre dos puntos de señalización. Toda transmisión en el enlace de señalización esta separada en octetos. Las unidades de señalización son reconstruidas, tomando uno a uno los bits del octeto que transporta la señalización, en los puntos de señalización por medio de software.

### 2.3.2.1 Unidades de señalización de relleno (FISUs):



Figura 2.7 Estructura de la unidad de señalización de relleno.

Unidad de señalización que provee el nivel de servicio mas bajo. La unidad de señalización de relleno actúa como una bandera en sistemas de señalización # 7 basada en multiplexacion por división de tiempo (TDM). Cuando no hay información para ser transmitida y la red esta ociosa, esta unidad de señalización es enviada.

Existe una gran diferencia en otras redes, donde las banderas son usadas para este propósito. Una bandera es un patrón usado para mantener sincronización de reloj en redes asíncronas. Cuando el enlace se degrada, no existe indicación de que se pueda establecer la transmisión por lo cual fallara y el enlace tendrá que ser puesto en observación o diagnostico.

En las redes de señalización #7, para lograr mantener un alto grado de fiabilidad se utilizan las FISUs. Esta unidad de señalización no proporciona ninguna información de mensaje, pero contiene una mínima cantidad de información necesaria para mantener el enlace. Los números de secuencia, por ejemplo, son usados para la confirmación de una unidad de señalización previa.

El campo mas significativo en las FISUs es el campo de chequeo de trama (FCS). Este campo es usado por el nivel tres para determinar si existe cualquier error. Este campo se encuentra en todas las unidades de señalización y es usado para transportar el residuo de

Numero de bits

8

2

6

FCS

LI

1  
FI



la aplicación del algoritmo CRC-16 realizado por el punto de señalización. De esta forma la capa superiores pueden constantemente evaluar el estado de un enlace, aun durante periodos de tráfico ocioso.

Las FISUs también pueden ser usadas para la confirmación de unidades de señalización previamente recibidas. Esto se hace enviando un FISU con un número *de secuencia hacia atrás* (BSN) igual al *número de secuencia hacia adelante* (FSN) de la unidad de señalización a ser confirmada. En otras palabras, el número de secuencia hacia atrás identifica el número de secuencia de la última unidad de señalización recibida con éxito.

Para realizar un aviso de un mensaje mal recibido (confirmación negativa) se requiere el uso del *bit de indicación hacia atrás* (BIB) y el *bit de indicación hacia adelante* (FIB). Usualmente, el BIB y el FIB son del mismo valor. Sin embargo, cuando existe una confirmación negativa de un paquete, el BIB cambia de estado, asumiendo una posición negada del FIB. Esto significa que se requiere una retransmisión.

El receptor del FISU u otra unidad de señalización con bits de indicación opuestos examinan el BSN del paquete enviado para determinar cual unidad de señalización se requiere su retransmisión. Es más eficiente que el uso de tramas de supervisión, tal como lo hacen otros protocolos.

Cuando no hay errores, los bits de indicación mantienen el mismo valor; es decir, el FIB y BIB son exactamente el mismo. Cuando una retransmisión ocurre, la unidad de señalización retransmitida es enviada con un FIB de igual estado que el BIB indicador de la falla y se mantiene ese valor hasta que ocurre otra petición de retransmisión. Esta alternancia se efectúa en el desarrollo de las transmisiones.

El FISU nunca es retransmitido si se encuentra error en el ya que no hay razón para retransmitir esta unidad de señalización por que no proporciona ninguna información. Son usadas solo para monitorear la integridad del enlace de señalización.

*El indicador de longitud* (LI), identifica el tipo de unidad de señalización a ser recibida. Este campo indica la longitud del *campo de información del servicio* (SF), la cual no existe para la FISU, y por lo tanto siempre es cero para este caso. La longitud de todo el FISU es estática todo el tiempo y tiene 48 bits de larga.

Una bandera (FLAG) es la encargada de delimitar el comienzo de cualquier unidad de señalización. Hay solamente una bandera de inicio y no existe bandera de final. La bandera de indicación del inicio de una unidad de señalización es la bandera de final de la unidad de señalización previa.

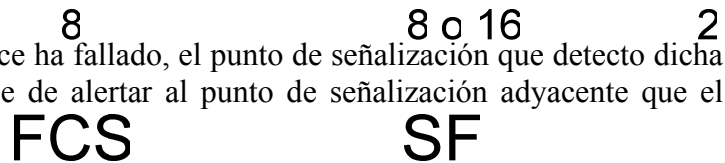
### 2.3.2.2 Unidades de señalización del estado del enlace (LSSUs):



Figura 2.8 Estructura de la unidad de señalización del estado del enlace.

Esta unidad de señalización es enviada entre dos puntos de señalización para indicar el estado del enlace que transmite la señalización. De esta forma, el LSSU solo tiene funcionalidad entre dos puntos de señalización adyacentes ya que no transmite información más allá de dicho entorno.

Cuando se determina que un enlace ha fallado, el punto de señalización que detecto dicha condición de error es responsable de alertar al punto de señalización adyacente que el enlace no esta disponible.



La primera acción a realizar en un caso de falla del enlace es eliminar todo el tráfico de dicho enlace. El LSSU es enviado al punto de señalización adyacente para informarle que todo el tráfico debe ser removido del enlace y que este último esta siendo realineado (el re-alineamiento es el proceso usado por el nivel dos y tres para corregir problemas en el enlace). Para ello no se requiere ningún mensaje de confirmación. Esta unidad de señalización es simplemente una unidad de señalización informativa.

El LSSU contiene los mismos campos que el FISU, con la diferencia de un nuevo campo: *el campo de estado (status field, SF)*. El campo de estado maneja la información del estado del enlace en el cual se transmite esta unidad de señalización y de ninguna manera reflejara el estado de otros enlaces. Cuando ocurre un error en la estructura de esta unidad de señalización, no será re-transmitida. Un LSSU detectado como erróneo (por medio del CRC) será descartado por el punto de señalización adyacente y dicho error es tomado como error del enlace.

Cuando el nivel tres determina que hay un problema, se transmitirá un LSSU identificando la causa del problema. El campo de estado (*status field*) en la estructura del mensaje LSSU (ver *figura 2.8*) describe el estado específico del enlace; por ejemplo: si el enlace esta desalineado o si el procesador esta sobrecargado en el punto de señalización que origina el mensaje.

Dependiendo del estado del enlace, el punto de señalización que recibe el LSSU puede enviar un mensaje de administración de la red en como respuesta (hacia otro punto de señalización adyacente), indicando la indisponibilidad de alcanzar un punto de señalización particular.

El valor del indicador de longitud LI en una LSSU, puede ser 1 o 2. Actualmente, la longitud del campo de estado en una LSSU es siempre de un octeto.

### 2.3.2.3 Unidades de señalización de mensajes (MSUs):

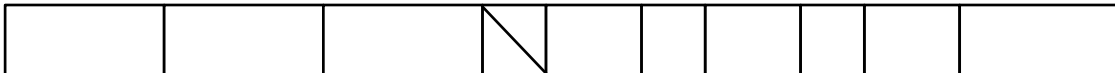


Figura 2.9 Estructura de la unidad de señalización de mensajes.

El MSU tal como se muestra en la *figura 2.9* proporciona la estructura para transmitir mensajes a los protocolos de capas superiores en el modelo de señalización # 7. La diferencia entre esta unidad de señalización y las dos unidades de señalización vistas anteriormente, es la aparición de dos nuevos campos: *el octeto indicador del servicio y el campo de información de servicio*.

**El octeto indicador del servicio (SIO)** (por sus siglas en inglés), es usado por la discriminación de mensajes en el nivel tres, para la determinación del tipo de protocolo que esta presente en esta MSU. Si el SIO indica que el protocolo a ser usado será ISUP, este último será el usuario de nivel cuatro. También el SIO identifica la versión del protocolo: si esta será nacional o internacional (ANSI, ITU-TS, etc.).

**El campo de información del servicio (SIF)** (por sus siglas en inglés), es un campo con capacidad de hasta 272 octetos para datos de usuario de nivel superior (nivel cuatro). Pero el SIF no necesariamente tiene que ser usado para información de nivel cuatro. La administración de red también es un usuario de este campo de información de señalización y dicho servicio es una función de nivel tres.

El campo indicador de longitud (LI), para esta unidad de señalización, puede tener cualquier valor arriba de 2; sin embargo, ya que es un campo de solamente 6 bits su máximo valor se limita a 63. Esta limitante también lo es para el significado literal de este campo. El objetivo real de este campo, es para determinar el tipo de unidad de señalización. Cualquier valor en el LI que sea mayor que 2 es siempre una MSU, y cualquier valor superior a 2 es realmente insignificante. Cuando el SIF excede los 64 octetos en longitud, el LI de dicho MSU permanece con el valor de 63.

### 2.3.3 Primitivas

Para establecer comunicación entre los niveles del modelo de señalización #7, debe ser implementado un método estandarizado para lograr dicho objetivo.

La comunicación entre el nivel dos y tres y entre nivel tres y cuatro es controlado por software. Las primitivas son el método usado por software para pasar información de un nivel bajo a un nivel alto y viceversa. Una primitiva es un elemento cien por ciento constituyente en software y son discutidas para entender el entorno que juegan en el sistema de señalización #7.

X	Nombre Genérico	Nombre Específico	Parametro
---	-----------------	-------------------	-----------

X = MTP o N (SCCP)

**Figura 2.10** Estructura de las primitivas.

Una primitiva proporciona cuatro campos. El primer campo, marcado como “X” indica el originador de la primitiva. Si el MTP esta pasando información hacia el ISUP, este campo debe indicarlo como “MTP”. El siguiente campo es el *nombre genérico*, el cual identifica el tipo de información a ser proporcionada. El siguiente campo es el *nombre específico* que describe la acción que se esta desarrollando, y puede ser:

- *Petición*: usada para invocar algún tipo de servicio de otro nivel.
- *Indicación*: usada para informar el nivel de requerimiento del servicio requerido.
- *Respuesta*: es enviada para completar una transacción particular entre un elemento de servicio y un usuario.
- *Confirmación*: es enviada para informar a la parte de usuario que una conexión ha sido establecida o un servicio requerido ha sido invocado.

El propósito de las primitivas, una vez mas, es el de proveer un medio de comunicación entre varios protocolos en el modelo de señalización #7 dentro del punto de señalización.

## 2.4 INTRODUCCIÓN A LOS PROTOCOLOS DEL SISTEMA DE SEÑALIZACIÓN #7

La red de señalización #7 proporciona algunos servicios básicos a la PSTN. El impulso detrás del desarrollo de esta red fue el de separar toda información de señalización de la red de voz.

A continuación se da una breve descripción de los protocolos usados en el sistema de señalización #7:

### 2.4.1 Parte de Transferencia de mensajes (MTP)

El MTP es el protocolo para el transporte usado por todos los otros protocolos en la red de señalización #7. Posee tres niveles o capas

- **La capa física o MTP de nivel uno.**  
Es la que permite el uso de cualquier tipo de interfase digital para soportar la tasa de transferencia de datos requerida por la red.
- **El MTP de nivel dos.**  
Es el que provee las funciones necesarias para proveer detección y corrección de errores básicos para todas las unidades de señalización. A este protocolo le concierne la entrega de unidades de señalización entre dos puntos de señalización adyacentes. Una vez que la información ha alcanzado el punto de señalización adyacente, es función del nivel tres de MTP el determinar como encaminar los mensajes.
- **El MTP de nivel tres.**  
Proporciona cuatro funciones: *discriminación de mensajes, distribución de mensajes, encaminamiento de mensajes y administración de la red.*

*Discriminación de mensajes*, usa el campo etiqueta de encaminamiento de los MSU para determinar, primeramente, hacia donde se esta enviando dicho mensaje.

*Distribución de mensajes*, usa el octeto indicador de servicio (SIO) para determinar quien será el usuario del mensaje.

*Encaminamiento de mensajes*, coloca una nueva etiqueta de encaminamiento a un mensaje saliente y determina cual enlace de señalización será usado para encaminar la llamada.

*Administración de la red*, hay varias funciones que se realizan para la administración de la red, las cuales son:

- *Administración del enlace*: concierne con la integridad de un enlace individual. Es capaz de bloquear mensajes de un enlace particular y notificar al punto de señalización adyacente que haga lo mismo.
- *Administración de tráfico*: provee el mecanismo para encaminar tráfico de enlaces fuera de servicio dentro de un conjunto de enlaces o linkset. Usa las MSU para enviar mensajes de cambio de enlace y de retorno a enlaces, dirigidos a puntos de señalización adyacentes.
- *Administración de encaminamiento*: usado para advertir a otros puntos de señalización en la red, acerca de la no disponibilidad de uno de ellos para alcanzar un determinado punto de señalización.

#### **2.4.2 Capa de Adaptación del usuario a MTP2 (M2UA).**

Este protocolo es usado para proveer los servicios del nivel dos de MTP en redes TCP/IP usadas para el transporte de señalización #7.

#### **2.4.3 Capa de Adaptación del usuario a MTP3 (M3UA).**

Este protocolo provee las funciones del nivel tres del MTP para redes TCP/IP usadas para el transporte de señalización #7. Este protocolo es usado para la conversión de códigos de puntos de señalización (*point codes*) a direcciones IPs.

#### **2.4.4 Protocolo de Transmisión de Señalización de Control (SCTP).**

Este protocolo es usado como un medio de transporte en las redes IP en conjunto con M2UA y M3UA. Es usado en lugar de TCP o UDP y ha sido desarrollado específicamente para uso de redes de señalización #7 enviado por IPs.

#### **2.4.5 Parte de Control de Conexión de la Señalización (SCCP).**

El propósito de SCCP es el de proveer el encaminamiento de usuario a usuario. El MTP únicamente es capaz de proveer el encaminamiento de punto a punto. Esto significa que un mensaje puede ser encaminado basado únicamente en el enlace físico disponible desde un punto de señalización.

SCCP provee el direccionamiento para encaminar un mensaje a través de la red entera. Esta información es usada en cada punto de señalización por el encaminamiento del nivel tres del MTP para determinar que conjunto de enlaces usar.

#### **2.4.6 Parte de Usuario del ISDN (ISUP).**

El ISUP es un protocolo de las aplicaciones relacionadas con circuitos, usado para el establecimiento de conexiones de circuitos y el mantenimiento de las mismas a través del desarrollo de una llamada. ISUP esta asociado únicamente con la transmisión de voz y datos y no brinda soporte a tecnologías de banda ancha tal como Frame Relay y ATM.

Soporta circuitos analógicos y digitales de voz.

Después que la conexión ha sido establecida, ISUP brinda comunicación entre lo dos conmutadores telefónicos de los abonados finales. Esta característica puede ser necesaria para brindar servicios de peticiones de usuarios, tales como llamada en conferencia o contestador automático.

#### **2.4.7 Parte de Usuario del ISDN de Banda Ancha (BISUP).**

Para brindar servicio al ISDN de banda ancha (BISDN) y arquitecturas ATM, el protocolo ISUP has sido modificado. Esta nueva versión del ISUP proporciona nuevos tipos de mensajes y parámetros, los cuales proporcionan el soporte necesario para redes de banda ancha y ATM.

#### **2.4.8 Parte de Usuario para Telefonía (TUP).**

El TUP es usado en redes internacionales. Este protocolo es compatible con ISUP, con la diferencia principal entre los tipos de mensajes y parámetros. Aun con dichas diferencias, los dos protocolos pueden ser compatibles perfectamente, aun si la relación de los parámetros de uno con respecto al otro no existe. El TUP esta siendo reemplazado por el ISUP en el ámbito internacional.

#### **2.4.9 Parte de Aplicación de Capacidad de Transacción (TCAP).**

El TCAP es probablemente es mas versátil de todos los protocolos del sistema de señalización #7. Es usado para dos propósitos: *acceso remoto a bases de datos y peticiones de servicios a puntos en redes remotas.*

Hoy en día, TCAP esta limitado al acceso a bases de datos, sin embargo mas y mas redes están prestando nuevos servicios avanzados los cuales requieren el uso de TCAP para que sean usados tales servicios por redes remotas.

El TCAP ha sido diseñado para proporcionar control remoto de entidades de la red hacia otras redes, las cuales pueden prestas muchas oportunidades. Por ejemplo, un subscritor desea cambiar un servicio telefónico. Normalmente, esto requeriría de una llamada a la compañía telefónica, la cual remotamente acensando a la base de datos del subscritor agregaría el servicio deseado por el subscritor a un registro de dicha base de datos. Una

orden de servicio sería generada y el nuevo servicio sería programado en el conmutador local correspondiente al suscriptor. Con las capacidades del TCAP, los suscriptores pueden tener la capacidad de configurar ellos mismos el servicio deseado sin la necesidad de la espera en la lista de requerimientos de la compañía.

## **2.5 PROTOCOLO MTP**

### **2.5.1 Generalidades**

La parte transferencia de mensajes (MTP, *message transfer part.*) proporciona las funciones que permiten que la información significativa de la parte usuario transmitida a la MTP sea transferida a través de la red del sistema de señalización # 7 hacia el destino requerido. Además, en la MTP se incluyen funciones que permiten remediar los fallos de la red y del sistema que podrían afectar la transferencia de la información de señalización. Esto constituye para el usuario MTP, un servicio secuenciado sin conexión.

### **2.5.2 Objetivos**

Los objetivos generales de la parte transferencia de mensajes consisten en proporcionar los medios necesarios para:

- a) El transporte y entrega fiables de la información de señalización de la parte de usuario a través de la red del sistema de señalización # 7.
- b) Reaccionar ante los fallos del sistema y de la red que puedan afectar a las funciones indicadas en a), y tomar las medidas necesarias para que se realicen dichas funciones.

### **2.5.3 Características generales**

#### **2.5.3.1 Método de descripción**

- Funciones proporcionadas por cada nivel dentro de la MTP;
- Servicios proporcionados por la MTP;
- Interacción con la red de señalización;
- Interacción con el usuario de la MTP;
- Capacidad de transferencia de mensajes de la MTP.

Las funciones de cada nivel de la MPT se llevan a cabo por medio del protocolo del nivel entre dos sistemas que proporciona un *servicio de nivel* a los niveles superiores (es decir,



nivel 1: enlace de datos de señalización, nivel 2: enlace de señalización, nivel 3: red de señalización).

#### **2.5.4 Descripción del MTP (nivel dos)**

El MTP realiza todas las funciones de las capas uno, dos y tres del modelo OSI. El nivel dos del MTP realiza detección y corrección de error, el control de error lo realiza por medio de un bit de chequeo. Otras funciones importantes que se realizan en la capa dos MTP son las siguientes:

- limitación de las unidades de señalización.
- Alineación de las unidades de señalización.
- Detección de error en las unidades de señalización
- Corrección de error en las unidades de señalización
- Alineación inicial de enlace de señalización
- Monitoreo de error de enlace de señalización.
- Control de flujo.

##### ***Limitación de las unidades de señalización.***

Todas las unidades de señalización son precedidas de una bandera. La bandera es un patrón de 8 bits, comienzan con un cero, luego siguen con 6 unos consecutivos, y terminan en un cero (01111110). La bandera es usada para indicar el inicio de una unidad de señalización y el final de otra. La delimitación de unidades de señalización es de gran importancia para las capas superiores.

##### ***Alineación de unidades de señalización.***

Un enlace se considera alineado cuando las unidades de señalización son recibidas en orden, sin violaciones de densidad de unos, y con el número adecuado de octetos (basados en el tipo de mensaje). Las unidades de señalización deben tener una longitud total que sea múltiplo de 8 bits. Si la unidad de señalización no tiene una longitud que sea múltiplo de 8 bits, o si la información en el campo de señalización (SIF) de la unidad de señalización (MSU) excede los 272 octetos de capacidad, la unidad de señalización recibida es considerada errónea. El enlace es considerado en error cuando ocurren demasiados errores. El número de errores se monitorea por medio de un contador de errores en las unidades de señalización. Este es determinado por un contador que monitorea los errores de las unidades de señalización (SUREM). Este cuenta el número de errores en un enlace de señalización. Cada enlace mantiene un único contador. El propósito de este contador es determinar cuando ocurren demasiados errores y así poder colocar el enlace en estado de fuera de servicio. Los errores están limitados a errores de alineación. La causa mas frecuente de los errores de alineación es que las señales de reloj no están sincronizadas en los dos puntos del enlace. La función de administración de la red en el nivel tres es la responsable por realinear el enlace (cuando es tomando fuera de

servicio y se encarga de re-sincronizarlo). El nivel dos es responsable por reportar cualquier error al nivel tres (administración de enlace). Cuando el enlace esta fuera de servicio, antes de ser habilitado para enviar MSUs, este debe ser probado. El procedimiento de prueba es conocido como “procedimiento de alineación”. Hay dos tipos de procedimientos de alineación que se utilizan: procedimiento de alineación normal y procedimiento de alineación emergente.

### ***Detección de error de las unidades de señalización.***

Los errores se detectan usando el bit de chequeo y el numero de secuencia de la unidad de señalización, si el bit de chequeo indica error, la unidad de señalización es descartada y se envía un reconocimiento negativo al punto de señalización que origino la unidad de señalización. *Los errores también son contados por la unidad de señalización de monitoreo de error (SUREM).*

### ***Corrección de error de unidades de señalización.***

Cuando un error es detectado en una unidad de señalización, esta es descartada. El nivel dos cuenta los errores (SUREM o intervalo de monitoreo de error) y pide que la unidad de señalización que fue descartada sea retransmitida.

Cuando se detectan demasiados errores en cualquier enlace, el enlace es colocado fuera de servicio, y se comienzan los procedimientos de alineación para probar el enlace y posteriormente ponerlo automáticamente en servicio. El enlace no debe ser puesto en servicio hasta que ha terminado el “*periodo de prueba*” del procedimiento de alineación.

### ***Monitoreo de error del enlace de señalización.***

Tres tipos de monitoreo de error son usados. Dos son usados cuando el enlace esta en servicio y el otro es usado cuando el enlace esta en procedimiento de alineación. El SUREM y el intervalo de monitoreo de error se usan mientras el enlace esta en servicio.

Con el SUREM, con cada unidad de señalización recibida erróneamente se incrementa un contador. El contador se disminuye con cada 256 unidades de señalización recibidas sin error. Cuando el contador alcanza el valor de 64, el enlace es sacado de servicio y comienzan los procedimientos de alineación. La velocidad de monitoreo de error en el procedimiento de alineación (AERM) es un contador incremental. Cada vez que un error es encontrado durante la alineación, el contador es incrementado en uno. Cuando el contador determina que hay demasiados errores, el enlace es colocado fuera de servicio, y procedimientos de alineación comienzan nuevamente.

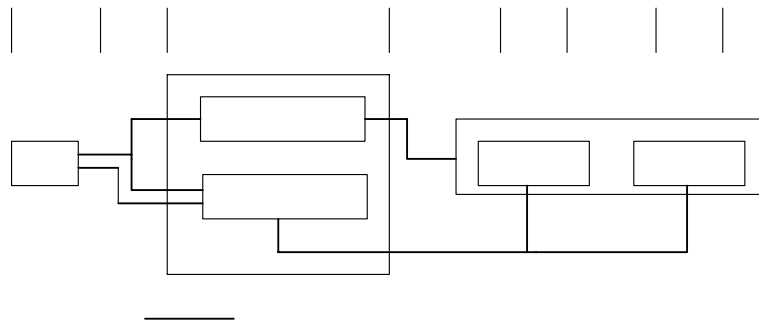
### ***Control de flujo.***

El control de flujo de unidades de señalización permite reducir el trafico cuando un punto de señalización esta congestionado. El LSSU se usa para enviar indicaciones de congestión a otros nodos. Cuando un LSSU de congestión es recibido, el punto de señalización receptor detiene el envío de MSUs hasta que la condición de congestión se

reduce. El control de flujo también utiliza prioridad para los tipos de unidades de señalización, para asegurar que las unidades de señalización más importantes tal como MSUs sean transmitidas, sobre todo durante una condición de congestión. El control de flujo no solo indica condiciones de congestión, también indica fallas en los procesos. El control de flujo en el nivel dos no se debe confundir con la administración de tráfico del nivel cuatro. El control de flujo del nivel dos es incorporado para enlaces individuales y no indica el estado del punto de señalización. En el nivel dos se realizan consideraciones de prioridad para unidades de señalización, estas consideraciones no se toman en cuenta en el nivel cuatro que es la parte de usuario. En el nivel tres la administración de red es diferente, este hace consideraciones al usuario de una unidad de señalización. En el nivel tres la administración controla el flujo de mensajes hacia el nivel cuatro, mientras que en el nivel dos controla el flujo de mensajes hacia un enlace. Si la condición de congestión continua, el enlace puede ser colocado fuera de servicio y realineado, usando los procedimientos de alineación. Esto previene que un enlace se bloquee cuando esta congestionado.

#### 2.5.4.1 Estructura del MTP nivel dos (Unidades de Señalización).

Los componentes utilizados por el nivel dos se observan en la *figura 2.11*, estos son encontrados en los tres tipos de unidades de señalización. La secuencia de números hacia atrás y la secuencia de números hacia delante son utilizados por los paquetes y también son utilizados por el nivel dos para asegurar que todos los paquetes transmitidos sean recibidos, además se utilizan para reconocimientos positivos y negativos. Los bits indicadores son usados para solicitar una retransmisión. El indicador de longitud (*length indicador*) permite al nivel dos determinar que tipo de unidad de señalización será enviada, y el campo de chequeo de redundancia cíclica (CRC) se utiliza para detectar errores en los datos de las unidades de señalización.



**Figura 2.11** Componentes utilizados en el Nivel dos del MTP.

#### ***Banderas.***

Las banderas son usadas para indicar el comienzo de una unidad de señalización y el final de otra unidad de señalización. El bit patrón de la bandera podría ser duplicado dentro de

la información de un MSU (violación de densidad de unos), causando que ocurra un error. Para prevenir la duplicación del bit patrón de bandera, el punto de señalización transmisor utiliza bits de relleno que son insertados delante de la transmisión después de los cinco unos consecutivos. El valor de este bit es siempre 0. Al insertar un cero después de cada patrón de cinco unos consecutivos antes de cada transmisión de una unidad de señalización, el punto de señalización transmisor asegura que nunca se enviaran seis unos consecutivos, por lo cual el 0 es insertado justamente después de la transmisión de la unidad de señalización. El punto de señalización que recibe una señal con cinco unos consecutivos, remueve el cero insertado en la unidad de señalización. Por lo que esta es una regla absoluta, un cero siempre será insertado después de cinco 1s consecutivos, y siempre es removido después de cinco unos consecutivos. Cuando ocurre una violación de densidad de unos, la unidad de señalización se considerada fuera de alineación y el nivel tres es notificado de una falla en el enlace.

### ***Numero de secuencia.***

El protocolo de la red de señalización #7 utiliza una numeración secuencial. En una numeración secuencial que trabaja dentro del nivel dos del MTP, el FSN indica el número de unidad de señalización que será enviada. Esta numeración secuencial es incrementada en uno después de que una señal es transmitida, excepto en el caso de FISU o LSSU. La FISU y la LSSU toman la FSN de la última MSU o LSSU enviada, y nunca la incrementan. El BSN se utiliza para el reconocimiento de las unidades de señalización recibidas. Por ejemplo, si se envía la secuencia de números del 1 al 7 y son recibidos por un punto de señalización distante, la señal que enviara el punto de señalización receptor debe tener un BSN de 7, lo cual indica que se reconocieron todas las unidades de señalización sin error.

Las unidades de señalización son chequeadas por su integridad (campo de chequeo de bits), y por que tengan una longitud apropiada. Una unidad de señalización debe ser por lo menos de seis octetos de longitud, si la unidad de señalización tiene menos de seis octetos, esta será descartada y la tasa de monitoreo es incrementada, y se envía un reconocimiento negativo para solicitar una retransmisión de la unidad de señalización que se recibió con error. Una unidad de señalización debe tener una longitud de múltiplos de 8 bits, o la unidad de señalización se detectara como errónea.

La tasa de monitoreo de errores de las unidades de señalización (SUREM) es un contador incremental que se incrementa en uno cuando ocurre un error. Los errores incluyen: unidades de señalización recibidas fuera de secuencia, con un mal CCR o con una longitud incorrecta. Después de 256 unidades de señalización recibidas sin error (unidades de señalización consecutivas), el SUREM se disminuye en uno.

Cuando el SUREM alcanza el valor de 64 errores, el enlace se reportada al nivel tres como fallido y es tomado fuera de servicio y colocado a través de procedimientos de alineación. El nivel tres controla las funciones de administración y dirige al nivel dos durante los procedimientos de alineación. El nivel dos no inicia los procedimientos de alineación: la función de este es simplemente reportar los errores y tomar la dirección de administración del enlace del nivel tres.

### ***Bits Indicadores.***

Los bits indicadores son usados para solicitar una retransmisión. Hay dos tipos de bits indicadores: un bit indicador hacia delante (*forward indicator bit FIB*) y un bit indicador hacia atrás (*backward indicator bit BIB*). En condiciones normales, los dos indicadores deben tener el mismo valor (0 o 1). Cuando una unidad de señalización solicita una retransmisión el BIB debe ser invertido, y el FIB mantiene el valor inicial. Esto indica a un punto de señalización distante que un error ocurrió y una retransmisión debe tomar lugar. Cuando la retransmisión alcanza el punto de señalización que origino la petición de retransmisión, una confirmación de recepción debe ser enviado. El FIB en la confirmación de recepción puede ser fijado al mismo valor del BIB y debe permanecer a este valor hasta que se solicite otra petición de retransmisión. Este proceso es llamado *detección y corrección básica de error*.

### ***Indicador de longitud.***

El indicador de longitud es usado por el nivel dos para determinar que tipo de unidad de señalización será enviada. Los valores del indicador de longitud pueden ser:

- 0 (Fill-in unidad de señalización)
- 1 o 2 (unidad de señalización del estado del enlace)
- 3 o mas (mensaje de unidad de señalización)

El indicador de longitud debe mostrar la longitud del siguiente campo, este indicador no existe en el FISU. En el LSSU el siguiente campo es el del estado del enlace (SF), el cual puede ser de dos o tres octetos de longitud. Si la unidad de señalización es una MSU, hay dos campos: el servicio de indicador de octeto (SIO) y el campo de información de señalización (SIF). El SIO es un campo de ocho bits, mientras el SIF es un campo de longitud variable usado por el nivel cuatro.

La máxima longitud del SIF es de 272 octetos, obviamente el indicador de longitud (LI) no puede acomodar un numero tan largo. Por lo tanto, cuando la longitud del SIF es de 64 octetos o mayor, el LI es puesto a 63. Únicamente el nivel dos usa el indicador de longitud, y su propósito es indicar que tipo de unidad de señalización será recibida. Los LIs pueden ser encontrados en paquetes a lo largo del nivel cuatro para indicar la longitud de los campos variables dentro del nivel cuatro.

### ***Chequeo de redundancia cíclica (CCR).***

El CCR es el último campo en la unidad de señalización. Este campo es calculado usando los campos que siguen inmediatamente después de la bandera, después del bit de chequeo. El proceso es bastante simple, el punto de señalización que transmite desarrolla el chequeo antes de transmitir la unidad de señalización. El residuo es transmitido en el campo de secuencia de chequeo. El punto de señalización receptor desarrolla un cálculo similar y compara el residuo con el del campo de secuencia de chequeo de la unidad de

señalización recibida. Si los resultados son distintos, la unidad de señalización es descartada y el contador de error SUREM es incrementado.

#### 2.5.4.2 Método de control básico de error.

Cuando un enlace es colocado en servicio, las funciones de detección y corrección de error se utilizan para mantener una transmisión apropiada de mensajes en el nivel dos de la red de señalización #7. Hay dos métodos de detección de error: *básico* y *preventiva redundancia cíclica (PCR)*.

La detección y corrección básica de error es usado cuando un enlace de señalización usa instalaciones terrestres.

El control de error básico usa un bit indicador dentro de una porción de la unidad de señalización, con el cual solicita la retransmisión de unidades de señalización que se recibieron con error (*figura 2.12*). Cuando una MSU es transmitida, el punto de señalización transmisor coloca el FIB y el BIB al mismo valor, en este estado permanecen los bits indicadores cuando no hay errores, no importando si son puestos a 0 o 1.

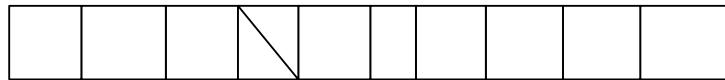


Figura 2.12 *Petición de retransmisión.*

Cuando un punto de señalización detecta un error, la unidad de señalización errónea es descartada y se envía un reconocimiento negativo al punto de señalización transmisor. El reconocimiento negativo puede ser enviado en un FISU, un LSSU o un MSU. El BIB se invierte para indicar una petición de retransmisión.

El BSN debe contener el valor del ultimo MSU bueno, el BSN es utilizado por el nivel dos en el punto de señalización transmisor para determinar cual unidad de señalización en el buffer de transmisión debe ser retransmitida. Todas las unidades de señalización enviadas deben ser reconocidas y removidas del buffer, y quedar solamente las que serán retransmitidas.

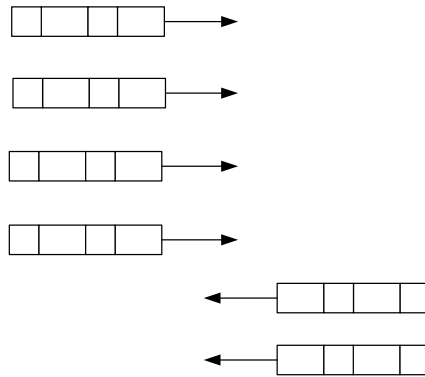
Cuando comienza una retransmisión, el valor del FIB es invertido para igualar el valor del BIB recibido en la petición de retransmisión. Los dos indicadores deben de nuevo tener el mismo valor. Estos retienen este valor hasta que una retransmisión es solicitada nuevamente, y el BIB es invertido a un valor diferente del FIB.

La secuencia de numeración en la red de señalización #7 puede ser un valor de 0 a 127. Este es conocido como modulo 128, significa números en secuencia que pueden ser enviados antes que un reconocimiento pueda ser recibido, este es llamado como *longitud de ventana*.

Cuando un enlace tiene demasiados errores, las retransmisiones pueden congestionar un punto de señalización. Cuando hay un número grande de unidades de señalización a retransmitir puede que existan problemas.

El tamaño de la ventana de cualquier punto de señalización debe ser determinado basado en el tipo de transmisión, la capacidad de el enlace, y el numero promedio de transmisiones que serán enviadas.

En la *figura 2.13* se observa una buena secuencia de transmisiones, con un reconocimiento positivo. El reconocimiento positivo es enviado por medio de una unidad de señalización en dirección opuesta con un reconocimiento (el BSN) de la unidad de señalización previamente transmitida. Cuando el reconocimiento es recibido, la unidad de señalización puede ser eliminada del buffer transmisor.



*Figura 5.3*

Figura 2.13 *Secuencia de transmisión exitosa.*

Debe ser mencionado que un reconocimiento no tiene que ser recibido después de cada unidad de señalización.

Un reconocimiento puede ser recibido solamente de algunas unidades de señalización transmitidas. Esto no indica un error. Como se observa en la figura 5.4, el reconocimiento puede venir un poco después. Algunas veces, los procesos están ocupados y no pueden reconocer todo a la vez.

**Unidades de señalización del estado del enlace (LSSU)**

El LSSU se muestra en la figura 2.14, este también es utilizado por el nivel dos para notificar a los nodos adyacentes del estado del nivel dos en el punto de señalización transmisor. El estado del enlace es llevado sobre el mismo enlace, y no sobre otros enlaces (asumiendo por su puesto, que el enlace funciona en el nivel dos).

Si el nivel dos falla completamente, entonces nada es recibido sobre este enlace, incluyendo los FISUs. En este caso el nivel tres es notificado que los reconocimientos no

Punto de señaliz

FSN BIB BSN

1 24 1 32

25 1 32

1 26 1 32

1 27 1 32

fueron recibidos, y el nivel tres empieza una recuperación del enlace fallido. La recuperación del enlace es acompañado de procedimientos de alineación.

El LSSU es transmitido por el software y hardware asociado con un enlace. El enlace en puede o no tener imperfecciones. Los problemas que se encuentran más frecuentemente en los puntos de señalización del enlace son las interfaces. Puesto que en estas interfaces es donde reside el software del nivel dos. Una falla en el hardware de la interfase del enlace puede causar que el software en el nivel dos inicialice el envío de LSSUs. El LSSU provee tres tipos de información del estado del enlace:

- Nivel dos esta congestionado.
- El nivel dos no puede acceder a los niveles tres y cuatro (procesador detenido).
- El procedimiento de alineación se ha implementado.

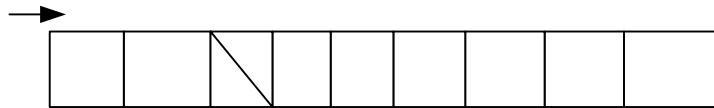


Figura 2.14 Componentes del LSSU.

El *campo de estado* (SF) es usado para proporcionar información del estado del enlace al punto de señalización adyacente. Esta información no incluye el número del enlace, lo cual significa que esta unidad de señalización no puede ser usada para notificar a los nodos acerca del estado del enlace sobre otros enlaces de señalización. Esto pertenece únicamente al enlace en el cual este fue recibido.

Otro importante concepto con el LSSU es que el estado del enlace es realmente el estado del nivel dos y tres en el punto de señalización transmisor, y no el estado del enlace como medio de transmisión.

El campo de estado (SF) es de 8 o 16 bits, aunque únicamente solo tres bits son se usan actualmente. Los tres bits mas significativos proveen la información del estatus actual, mientras los bits restantes permanecen a cero.

- **El indicador de estado ocupado (SIB)**, indica que el nivel dos esta congestionado en el punto de señalización que transmite el mensaje. Cuando un punto de señalización recibe un SIB, para antes de transmitir los MSUs comienza a enviar FISUs. Si la condición persiste de tres a seis segundos, el nivel tres es informado de una falla en el enlace. Entonces el nivel tres inicializa los procedimientos de alineación para que comiencen a ejecutarse en el enlace afectado.

El punto de señalización congestionado puede continuar enviando SIBs a intervalos regulares de tiempo, hasta que la condición de congestión se reduce. Cualquier MSUs recibido por el punto de señalización transmisor puede ser reconocido, pero no puede enviar nuevos MSUs.

Para indicar que la congestión a disminuido y los procesos normales pueden comenzar, el punto de señalización afectado debe comenzar nuevamente a transmitir MSUs y detener el envío de SIBs. El punto de señalización receptor reconoce la ausencia de SIBs y comienza a recibir MSUs normales.

Numero de bits 8 8 o 16  
FCS SF



- ***El indicador del estado fuera de proceso (SIPO)***, indica que el punto de señalización que esta transmitiendo mensajes de señalización no puede comunicarse con los niveles tres y cuatro. Esto puede ser a causa de una falla en el procesador central (CPU) o una falla nodal completa.  
El nivel dos frecuentemente esta funcionando, debido a que este reside dentro del hardware de la interfase del enlace. El punto de señalización envía un SIPO para notificar a un punto de señalización distante que pare de enviar MSUs.  
Cuando un punto de señalización recibe un SIPO, detiene todas las transmisiones de MSUs, se envían y FISUs al punto de señalización afectado.  
En el nivel tres, aunque el procesador este fuera de servicio, este todavía puede ser capaz de controlar las funciones del nivel dos. Cuando esta condición ocurre, el nivel tres puede pedir al nivel dos que vacíe su buffer (el buffer receptor y el buffer transmisor), y todos los MSUs en el buffer son descartados. Cuando un MSU es recibido desde un punto de señalización remoto, el nivel dos del punto de señalización afectado envía un FISU con el FSN y el FIB puesto al mismo valor del BSN y el BIB de la última MSU recibida del punto de señalización remoto. Los procesos normales de todos los mensajes continúan.
  
- ***El indicador de estado de fuera de servicio (SIOS)***. Esta condición ocurre cuando una unidad de señalización es recibida y tiene una violación de densidad de unos (el campo de datos simula una bandera) o el campo de información de señalización excede la capacidad máxima de 272 octetos. El SIOS es enviado cuando el enlace es fallido y los procedimientos de alineación son inicializados.  
Un LSSU fuera de servicio indica que el punto de señalización que transmite los mensajes no puede recibir o transmitir ningún MSUs dado que el procesador esta parado. Cuando un punto de señalización recibe un SIOS, este detiene la transmisión de MSUs y comienza a transmitir FISUs. El SIOS es también enviado al comienzo del procedimiento de alineación.
  
- ***El estado de enlace normal (SIN) o emergencia (SIE)***, indica que el punto de señalización transmisor ha comenzado procedimientos de alineación. El enlace es deshabilitado para MSUs y únicamente FISUs son transmitidas al punto de señalización afectado hasta que un periodo de prueba ha transcurrido. Después de un exitoso periodo de pruebas, los MSUs pueden ser transmitidos sobre el enlace al punto de señalización afectado.  
En el caso de que un LSSU sea recibido con errores, no se solicitan retransmisiones, simplemente se descarta la unidad de señalización. El FSN de un LSSU no incrementa, asume el valor del último MSU. El BSN incrementa cuando un reconocimiento es enviado al punto de señalización distante.

El LSSU es procesado dentro del nivel dos y no pasa a nivel tres. Sin embargo, el nivel dos puede pasar información de control al nivel tres, dependiendo de el estatus de la

LSSU, por ejemplo si una LSSU es recibida con el estatus de “ocupado”, el nivel dos notifica al nivel tres que detenga las transmisiones de MSUs.

Es importante recordar que las LSSU trabajan independientemente de la administración de red, la cual es una función del nivel tres. De hecho, la administración de red usa los MSUs para enviar información de administración a otros nodos y así poder usar cualquier enlace o ruta para enviar mensajes a los nodos adyacentes. El nivel tres se utiliza cuando los enlaces fallan completamente y las LSSUs no pueden ser transmitidas sobre los enlaces fallidos.

### ***Unidad de señalización de procedimiento de alineación***

El proceso de alineación consiste en enviar unidades de señalización con una correcta longitud y sin violaciones de densidad de unos. Cuando ocurre un error debido a uno de estos eventos, el punto de señalización afectado puede comenzar procedimientos de alineación.

El propósito de los procedimientos de alineación es reestablecer el tiempo y la alineación de los puntos de señalización, para que el punto de señalización afectado pueda determinar donde comienzan y terminan las unidades de señalización. Como se menciono previamente, la condición de fuera de alineación ocurre cuando la bandera es simulada dentro de los datos (violación de densidad de unos) o la información del campo de señalización es demasiado larga (mayor que 272 octetos), que indicaría que la bandera fue errónea.

Los procedimientos para reestablecer la transmisión y la recepción en el nivel dos de los nodos y no afectar los otros enlaces en cada punto de señalización, provee pruebas en un periodo de tiempo para asegurar que la transmisión sobre el enlace es fiable, previniendo que ocurran errores en el futuro.

Hay dos procedimientos de alineación utilizados: *alineación normal* y *alineación emergente*.

***La alineación normal***, es usada cuando hay otros enlaces asociados con el enlace que falla (tal como un conjunto de enlaces). Los otros enlaces deben tener conexión al mismo destino.

***La alineación de emergencia***, va a través del mismo procedimiento. Pero dentro de un pequeño periodo de tiempo. El nivel tres es responsable por determinar cual procedimiento de alineación usar.

Hay cuatro estados introducidos durante el procedimiento de alineación. Los temporizadores asociados con cada estado aseguran que el punto de señalización no se quede atascado en ningún estado. Cuando cualquiera de los temporizadores termina su tiempo, la alineación comienza nuevamente. La siguiente explicación describe cada estado y los eventos que ocurren durante estos estados.

### ***Estado 00-inactivo***

Este estado indica que el procedimiento es suspendido, y es el primer estado que se introduce. El estado 00 se introduce cuando el procedimiento de alineación es abortado (debido a la existencia de errores) durante el tiempo en que el enlace esta siendo probado. Cuando esto ocurre, el nivel tres (que es el de administración de red) re-direcciona las unidades de señalización a otros enlaces. Si el enlace falla en el periodo de prueba, el nivel tres coloca nuevamente el enlace en estado 00 por un periodo de tiempo.

### ***Estado 01- no alineado***

Este se introduce cuando el nivel tres es inicializado. El LSSU es usado para enviar un estatus de fuera de alineación (SIO). Cuando este estatus es introducido, el temporizador T2 del nivel dos es inicializado. El temporizador T2 es colocado para una alineación normal de 11.3 o 23 segundos. Si un nodo de señalización en cualquier parte de la red usa un mecanismo para asignar automáticamente los enlaces de señalización, el temporizador T2 debe ser inicializado en cada nodo.

Los nodos tienen la opción de asignar enlaces exclusivamente para señalización, pero estos enlaces pueden en cualquier momento estar habilitados para otras aplicaciones, tal como transmisión de voz. Los puntos de señalización tienen la capacidad por medio del nivel tres de interrumpir las aplicaciones para las cuales los enlaces están siendo ocupados, y reasignar el enlace para llevar tráfico de señalización en base a una necesidad (tal como en el caso de que un enlace de señalización falle dentro de un conjunto de enlaces). El estado 01 permanece hasta que el nivel tres inicia el siguiente estado de transmisión. Durante este estado, el punto de señalización transmite LSSUs con una indicación de estado SIOS.

### ***Estado 02- alineado***

El estado 02, indica que el enlace esta alineado y es capaz de detectar banderas y unidades de señalización sin error.

Durante el tiempo que el enlace se encuentra en el estado 02, el temporizador del nivel dos T3 es inicializado. Cuando el enlace sale del estado dos 02 y entra al estado tres 03, T3 es detenido. Si termina el tiempo de T3, y el enlace no ha salido del estado 02, el enlace es retornado al estado 00, y comienza todo el proceso nuevamente.

### ***Estado 03- Periodo de prueba***

El periodo de prueba se utiliza para probar la integridad del enlace y el nivel dos en el punto de señalización. Durante este periodo, LSSUs con el valor de SIN o SIE son enviados, y se cuentan los errores. Hay dos periodos de prueba: normal y emergencia. El periodo de prueba normal dura 2.3 segundos, durante este tiempo, no pueden ocurrir más de cuatro errores mientras se encuentra en el estado 03. La velocidad de monitoreo del error de alineación mantiene un conteo de todos los errores recibidos durante el periodo de prueba. El AERM es un contador incremental, el cual cuenta todos los errores de

transmisión, incluyendo errores de chequeo de redundancia cíclica y los errores de violaciones de densidad de unos.

Durante el periodo de prueba, FISUs son enviadas en el enlace, LSSU normales o de emergencia (SIN o SIE) son también enviadas para indicar que el enlace esta siendo probado.

El periodo de prueba emergente tiene una duración aproximada de 0.6segundos, tiempo durante el cual, no puede ocurrir mas de un error. Este es también monitoreado por el AERM. Cuando ocurren demasiados errores según el procedimiento, el enlace es retornado al estado 00, y comienza todo el proceso nuevamente.

Si un enlace no puede ser restaurado, es retornado al estado de inactividad y los procedimientos de alineación son repetidos, hasta que el enlace es restaurado o el personal de mantenimiento detecta la falla y toma acciones correctivas. Cuando un enlace esta fallando continuamente o en alineación constante, indica que un equipo esta fallando, el cual puede ser corregido por intervención manual (reemplazar la interfase del enlace frecuentemente soluciona el problema).

La medición de tráfico proporcionada por muchos puntos de señalización puede ser una herramienta muy útil para determinar el número de fallas en determinado periodo de tiempo.

### 2.5.5 MTP nivel tres

Hay dos categorías de funcionalidad en este nivel: manejo *de mensajes de señalización* y *administración de señalización de redes*.

***El manejo de mensajes de señalización***, es utilizado para el encaminado de mensajes al enlace apropiado y determina si los mensajes son enviados al nodo receptor o son reenviados.

***La administración de señalización de red***, es utilizada para re-encaminar tráfico cuando los nodos se vuelven no disponibles.

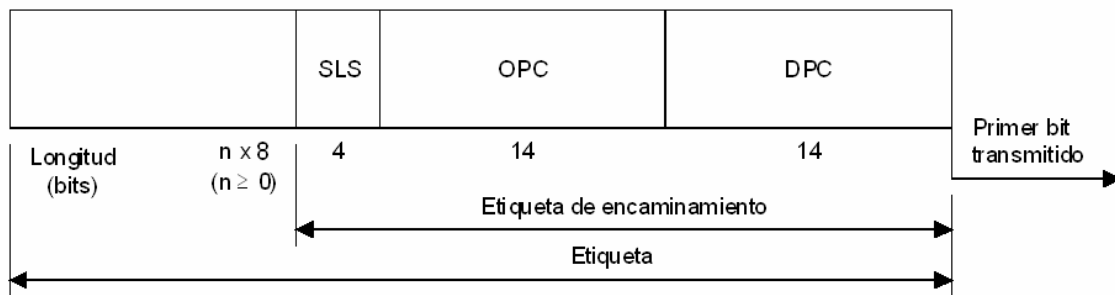
La parte de transferencia de mensaje (MTP) nivel tres se apoya sobre los servicios del nivel dos para la entrega de los mensajes. La interfase entre los dos niveles consiste de un conjunto de primitivas. Las primitivas permiten ser enviados por medio de parámetros, al nivel dos para ser encaminados sobre la red en forma de mensajes SS7. Al mismo tiempo. Las primitivas permiten al nivel dos enviar parámetros al nivel tres. Para procesamiento de mensajes.

El procesamiento de mensajes comienza en el nivel tres. El nivel tres debe determinar el destino del mensaje y el usuario del mensaje, y mantener el estatus de la red. El nivel tres utiliza primitivos para comunicarse con los usuarios del nivel cuatro.

### 2.5.5.1 Estructura de los mensajes.

#### 2.5.5.1.1 Discriminación de mensajes.

La discriminación aplica a todos los mensajes recibidos. La discriminación de mensajes utiliza la *etiqueta de encaminado* (ver figura 2.15) de la unidad de señalización de mensaje (MSU) para determinar el destino de un mensaje. La etiqueta de encaminamiento proporciona la dirección de origen (punto de código de origen [OPC]), la dirección de destino (punto de código destino [DPC]), y el código de señalización de enlace (SLC).



DPC Código de punto de destino  
OPC Código de punto de origen  
SLS Selección de enlaces de señalización

Figura 2.15 Estructura de la etiqueta de encaminamiento.

La discriminación de mensaje lee la etiqueta de enrutamiento, específicamente el punto de código destino, para determinar si el destino es el nodo receptor o si el nodo debe ser transferido a otro nodo. Si el mensaje es deseado por el nodo receptor, entonces el nodo es dado a la función de distribución de mensaje, el cual debe determinar cual parte usuaria recibirá el mensaje.

El usuario es determinado por la lectura del octeto *indicador de servicio (SIO)*. Este campo está dividido en dos partes: el *indicador de servicio* y el *campo de sub-servicio*.

**El indicador de servicio**, identifica la parte usuaria que recibe el mensaje. Una parte usuaria no tiene que ser una función de nivel cuatro. La administración de redes y la administración de pruebas de red son funciones de nivel tres que utilizan el campo de información de una MSU. El indicador de servicio puede indicar si mensajes de prueba especial o regular son enviados por la red de administración.

**El octeto de sub-servicio**, identifica si este mensaje es de una red internacional o una red nacional. Esta parte del mensaje es utilizada para identificar el tipo de punto de código (*point code*) utilizado, así la función de discriminación puede determinar como leer la etiqueta de encaminamiento. Si el valor del indicador de red en el campo de sub-servicio

indica una red internacional, la etiqueta de encaminamiento utiliza diferente estructura como lo haría para un mensaje nacional.

Si el octeto indicador de servicio indica que esto es un mensaje nacional, el SIO puede ser utilizado para indicar dos diferentes versiones de la estructura nacional. La “reserva” o código reservado en el campo de sub-servicio puede ser utilizado para indicar una diferente versión de la estructura nacional.

Si el mensaje no es deseado por el nodo receptor, luego es dado a la función de encaminamiento de mensaje. La función de encaminamiento de mensaje debe determinar cual enlace escoger para alcanzar el nodo destino.

Si el mensaje es para otro nodo, la función de encaminamiento de mensaje determina que no puede alcanzar, la administración de la red debe responder. La administración de red es una función de nivel tres que es invocada cuando una falla de punto de señalización es detectada o una falla de enlace es detectada.

#### 2.5.5.1.2 Distribución de mensajes

La distribución de mensaje debe determinar para quien es el mensaje dado. Esto es determinado examinando el octeto indicador de servicio del campo indicador de servicio.

En el evento que un usuario no esta disponible, un mensaje *parte usuaria no disponible* (UPU) es enviado por la administración de red como una indicación al punto de origen que el mensaje fue desechado por la función de le parte usuaria no estaba equipada en el destino o estaba no disponible. El mensaje UPU también proporciona un código causa, aunque este código causa no es muy específico. Hay tres causas proporcionadas: *la función de parte usuaria no estaba equipada en el destino*, *la función de parte usuaria no estaba accesible*, o *la función de parte usuaria no debió ser completada por una razón desconocida*. El mensaje UPU es mostrado en la figura 2.16.

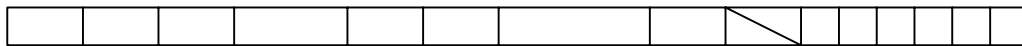


Figura 2.16 Componentes del mensaje UPU

#### 2.5.5.1.3 Encaminamiento de mensajes

Si un mensaje es recibido de otro destino, la discriminación de mensajes pasa el mensaje al *encaminamiento de mensaje* para transmisión sobre la red. Esta es una función común del STP. Un punto de señalización Terminal, como un *punto de conmutación de servicios* (SSP), podría probablemente recibir mensajes que necesitaran ser transmitidos.

La función de encaminamiento debe primero determinar la dirección de destino por la lectura de la etiqueta de encaminamiento y observación por el punto código destino. La siguiente figura ilustra la estructura de un PC de ámbito internacional.



Figura 2.17 Estructura del PC internacional.

Todo código de punto de señalización internacional (ISPC) debe constar de tres sub-campos de identificación, como se indica en la Figura 2.17. El sub-campo de 3 bits (NML) debe identificar una región geográfica del mundo. El sub-campo de 8 bits (K-D) debe identificar una zona geográfica o red en una zona específica. El sub-campo de 3 bits (CBA) debe identificar un punto de señalización en una zona geográfica o red específica. La combinación del primero y segundo sub-campos podría considerarse como un código de zona/red de señalización.

Hay dos formas de encaminamiento que pueden ser desplegadas dentro de un punto de señalización: *encaminamiento completo de punto de código* y *encaminamiento parcial de punto de código*.

**En el caso de encaminamiento completo**, el punto de señalización mira en el punto de código entero para determinar como será encaminado el mensaje. Esto significa que la tabla de encaminamiento para ese punto de señalización debe incluir todos los puntos código para todos los puntos de señalización de conexión y los puntos terminales en la red.

**En el caso de encaminamiento parcial**, este usa el método de encaminamiento por clusters pero se lee solamente la dirección de red. Es muy útil en los puntos de señalización que conectan a otras redes. Las puertas de enlace utilizan esta forma de encaminamiento.

#### **2.5.5.1.4 Selección de enlace**

La señalización de enlace utilizada para encaminar la llamada se determina por el valor del campo de *selección de enlace de señalización* (SLS ver figura 2.15) localizado el campo de *código de enlace de señalización* (SLC) en la tabla de encaminamiento. El código SLC es un código preasignado representando el enlace físico.

Todos los enlaces le son asignados un SLC y un *número de Terminal*. El número de Terminal es una conexión lógica a un enlace físico. El rango de SLCs es de 0 a 127. Esto significa que cada enlace tendrá más que un SLC.

Antes de la transmisión, el encaminado de nivel tres determina cual enlace es el enlace de salida. Esto es basado en varios factores, primero, todos los enlaces dentro de un conjunto de enlaces debe transportar mas que los otros, el tamaño del MSU debe también estar figurado en capacidad de trafico en cada enlace, tanto que un enlace no se vuelca cargado con grandes mensajes mientras otro esta enviando solamente mensajes pequeños.

Después un enlace es seleccionado, el código de enlace de señalización del enlace físico es colocado en el campo SLC de la etiqueta de encaminamiento y enviado el nivel dos para transmisión. Cuando el mensaje es recibido en el punto de señalización remoto, los bits en el campo SLC son rotados una posición a la derecha por el encaminamiento de nivel tres, actualmente solamente los últimos cinco bits significativos son rotados.

La división de carga debe ser utilizado cada vez que hay mas de un enlace o conjunto de enlace. En el evento que haya más de un conjunto de enlace, el bit menos significativo (LSB) es utilizado para seleccionar el conjunto de enlaces a utilizar.

Los cuatro bits restantes identifican el enlace dentro del conjunto. En el evento donde solamente exista un solo conjunto de enlaces, el LSB es utilizado como parte del identificador del enlace, pero el bit más significativo (bit 5) es dejado sin uso.

#### **2.5.5.2 Administración de red**

La administración de red a nivel tres proporciona los procedimientos y la funcionalidad para re-encaminar trafico a través de enlaces alternos y conjuntos de enlaces o para controlar le flujo de trafico a un punto código destino especifico. Hay tres funciones separadas de administración de red utilizadas en SS7

- Administración de trafico
- Administración de enlace
- Administración de ruta



***La administración de tráfico***, es utilizada entre dos puntos de señalización para desviar tráfico lejos de los enlaces fallidos. Los mensajes de administración de tráfico son originados por el punto de señalización, el cual detecta un problema en un enlace y es evaluado sobre un enlace alternativo para informar a su punto de señalización adyacente de no encaminar mensajes sobre el enlace afectado. La administración de tráfico también activa funciones de administración de rutas del punto de señalización receptor a sus nodos adyacentes.

***La administración de enlace***, consiste de procedimientos de activación y desactivación, también como restauración de enlace. Estas funciones son utilizadas con el nivel dos para restaurar enlaces fallidos dentro del servicio. La función de administración de enlace es el que dispara los procedimientos de nivel dos para realineación de trama.

***La administración de rutas***, es utilizada para desviar tráfico de un punto de señalización específico. Esta es una función proporcionada por los SPTs, sumando un importante elemento a la red. La administración de rutas es activada por los mensajes de administración de tráfico y es enviada al STPs adyacentes cuando un mensaje de administración de tráfico es recibido.

### **2.5.5.3 Procedimientos de administración de señalización de red**

La administración de red también proporciona aproximación por capas para administrar problemas en la red. Los procedimientos son proporcionados para que traten con el congestionamiento y fuera de servicio, del nivel de enlace en todas las formas hasta el nivel de ruta. Este nivel dos proporciona el medio para detectar errores sobre los enlaces individuales. Cuando es encontrado un error, el nivel dos reporta el error al nivel tres, el cual entonces debe determinar cual procedimiento invocar. Los procedimientos comienzan en el nivel mas bajo, el nivel de enlace, y trabaja su forma para el nivel de ruta.

La administración de enlace no tiene procedimientos directos en el nivel de enlace. Estos procedimientos no tienen un impacto directo sobre el enrutamiento o el status del punto de señalización, ellos hacen activar otros eventos de administración de red a nivel tres.

La administración de tráfico no desvía no desvía tráfico lejos de un punto de señalización. El propósito de la administración es redirigir tráfico a un punto de señalización a otro enlace. Sin embargo, la administración de tráfico impacta rutas y conjunto de rutas a destinos específicos, si una ruta particular es utilizada por otro punto de señalización para alcanzar su destino, y la administración de tráfico ha desviado tráfico lejos de esa ruta, los puntos de señalización adyacentes tienen que invocar procedimientos de administración de ruta.

La administración de ruta desvía tráfico lejos de un punto de señalización que se ha vuelto no disponible o congestionado. Esta razón puede variar, pero sin importar la razón, la administración de tráfico y enlace estarán envueltas en el punto de señalización afectado.

Los mensajes de administración de red utilizan *la unidad de señal mensaje* (MSU) para encaminamiento de mensajes al punto de señalización apropiado. Dentro del *campo de información de señalización* (SIF) esta la información vinculada al *point code* que experimenta las fallas o el enlace que ha fallado, en adición, los códigos de estatus, códigos de prioridad, y otros códigos de mantenimiento pueden ser incluidos.

#### 2.5.5.4 Procedimientos de administración de enlace

La administración de enlace proporciona los procedimientos necesarios para administrar los enlaces individuales dentro de un punto de señalización. La administración de enlace no ve enlaces de un conjunto de enlaces o incluso una perspectiva de ruta, pero si desde una perspectiva individual.

El mensaje de administración (mostrada en la *figura 2.18*) utiliza un código de cabecera H0 y H1 para identificar el tipo de mensaje de administración que esta siendo enviado. El código cabecera H0 identifica el tipo de mensaje, considerando que el código cabecera H1 identifica el tipo de mensaje para el procedimiento que esta siendo declarado. Los siguientes son algunos de los valores para el código cabecera H0 (para mayor información ver recomendación UIT-T Q706):

H0	Descripción
0000	Reserva
0001	Mensajes de cambio y retorno de cambio
0010	Mensaje de emergencia de cambio

Tabla 2.1 Algunos valores que toma H0.

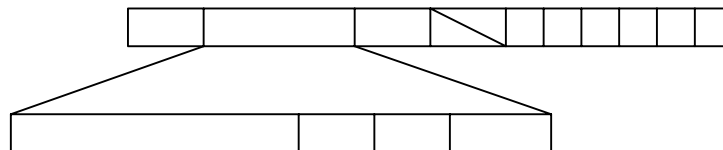


Figura 2.18 Mensaje de mantenimiento de red

La administración de enlace requiere el uso de las funciones de la parte de transferencia de mensaje nivel dos o SAAL para reportar fallas y status de enlace a un punto de señalización adyacente. Estos mensajes no son propagados a través de la red. La administración del enlace esta solamente vinculada a las terminales locales de un enlace.

Hay tres funciones proporcionadas por la administración del enlace

- Activación del enlace
- Restauración del enlace
- Desactivación del enlace

**Activación del enlace**, cuando un enlace es activado primero, el nivel tres dirige al nivel dos para comenzar los procedimientos de realineación y colocar el enlace en servicio. Antes de que los mensajes puedan ser transmitidos sobre el enlace, la administración enviara mensajes de prueba sobre el enlace, para asegurar la integridad del enlace. El nivel dos es utilizado para informar a los puntos de señalización adyacentes de las actividades en el nivel tres. Esto es completado utilizando la unidad de señal de estatus de enlace.

**La restauración del enlace**, involucra la unidad de señalización del estado del enlace (LSSU) para informar a los puntos de señalización adyacentes de los eventos que están tomando lugar. Los puntos de señalización que detectaron los errores son responsables por la invocación de los procedimientos de alineación y la notificación acerca del status a los puntos de señalización adyacentes.

**La desactivación del enlace**, es utilizado cuando un enlace es encontrado erróneo y necesita ser puesto en alineación. La desactivación del enlace debe primero detener todo el trafico en el enlace, el cual invocara procedimiento de administración de trafico (desviación del trafico del enlace fallido). La administración del enlace entonces desconecta el enlace de su conexión lógica (código de enlace de señalización). Para acomodar la administración de enlace y la asignación automática, un mensaje de señalización-datos-enlace-conexión (*figura 2.19*) debe ser generado y enviado al punto de señalización adyacente para informar de la nueva asignación. El punto de señalización adyacente debe hacer la misma asignación por el enlace para trabajar.

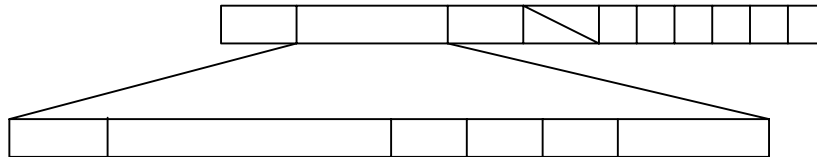


Figura 2.19 Mensaje señalización-datos-enlace-conexión

### 2.5.5.5 Procedimientos de administración de tráfico

La administración de tráfico es utilizada para desviar tráfico de un enlace fallido. Utiliza varios mensajes los cuáles son enviados valiéndose de la unidad de señalización mensaje (MSU) sobre los enlaces adyacentes o conjunto de enlaces de enlaces adyacentes a un punto de señalización adyacente.

La administración de tráfico también trata con la fuente del congestionamiento y proporciona procedimiento de control de flujo también. El objetivo de la administración de tráfico es tratar con la fuente de un problema siempre que sea posible.

La administración de tráfico también maneja problemas sobre un nivel más directo que otros procedimientos de administración de red.

La administración de tráfico proporciona los mecanismos para la desviación de tráfico debido a:

- No disponibilidad de enlace de señalización
- Disponibilidad de enlace de señalización
- No disponibilidad de encaminamiento de señalización
- Disponibilidad de encaminamiento de señalización
- Punto de señalización restringido
- Punto de señalización no disponible

### 2.5.5.6 Estructura del mensaje de administración de tráfico.

El mensaje de administración de tráfico proporciona el código de señalización de un enlace fallido y, en algunos casos, la secuencia numérica directa del último MSU bueno recibido en el enlace fallido. Esta información es enviada al punto de señalización adyacente así que ellos pueden asumir el tráfico por el enlace fallido y asegurar que no se pierden mensajes

#### ***Cambio:***

El mensaje de cambio es utilizado para desviar tráfico lejos de un enlace fallido. LSSUs son enviados por el nivel dos para indicar el estado del enlace a través de este procedimiento. Esto permite a los dos puntos de señalización mantener un estado actualizado mientras el enlace esta siendo realineado. Cuando no están siendo enviados LSSUs. Unidades de señalización de relleno (FISU) son enviados.

La *figura 2.20* muestra el contenido del mensaje de cambio

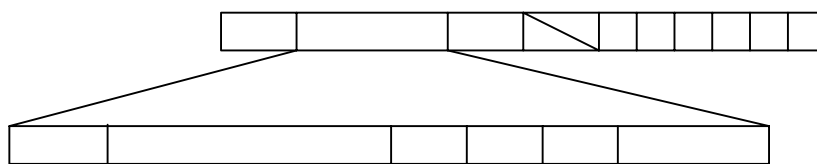


Figura 2.20 *Mensaje de orden de intercambio (COO).*

#### ***Retorno de cambio:***

El mensaje de retorno de cambio es utilizado cuando un enlace fallido ha sido restaurado, y el tráfico puede ahora continuar sobre ese enlace, la estructura del enlace, como se ve en la *figura 2.21*, consiste del código enlace de señalización del ahora restaurado enlace de señalización y el código del retorno de cambio. Los siguientes son los códigos cabecera H1 utilizados en los mensajes retorno de cambio.

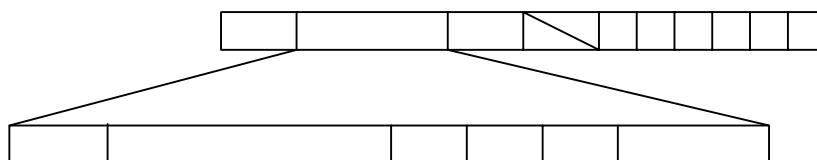


Figura 2.21 *Declaración de retorno de intercambio*

El *reconocimiento de retorno de cambio* (CBA) puede ser enviado sobre algún enlace disponible mientras el mensaje es encaminado al punto de origen del mensaje retornado. Una vez el reconocimiento es recibido, los MSUs que fueron guardados en el buffer de retorno de cambio son enviados sobre el enlace de señalización ahora disponible.

***Intercambio de emergencia:***

0000 Código de retorno de cambio

En el caso que un procedimiento de intercambio es iniciado pero el buffer de transmisión no puede ser leído, un procedimiento de intercambio de emergencia es utilizado, el intercambio de emergencia no proporciona la secuencia numérica directa (MTP L2) o la secuencia numérica (SAAL) de el MSU bueno recibido, porque el buffer ha sido limpiado y la información no está disponible.

***Re-encaminamiento forzoso:***

El procedimiento de encaminamiento forzoso es iniciado el caso que una ruta a un destino específico se vuelve no disponible. El propósito es re-encaminar el tráfico alrededor del correspondiente punto al destino. Sin perder mensajes u causar el congestionamiento de otra ruta.

***Re-encaminamiento controlado:***

El objetivo del procedimiento de re-encaminamiento controlado es restaurar el tráfico en la ruta más favorable, en casos donde una ruta particular fue previamente restringida.

***Administración de Inhibición:***

La administración de inhibición es utilizada por la administración de enlace para bloquear un enlace de señalización de nivel cuatro. El estatus del enlace no cambia a nivel dos. El propósito de este procedimiento es permitir al personal enviar mensajes de prueba sobre el enlace inhibido, o permitir a la administración del enlace enviar mensajes de prueba sobre el enlace sin ninguna interferencia de alguna otra parte usuaria.

### ***Control de flujo:***

El control de flujo en el nivel tres es utilizado para controlar el flujo de los mensajes de la parte usuaria de la fuente, mucho de este control de flujo es implementación dependiente. El procedimiento implementado para congestión o no disponibilidad de partes usuarias son dependientes de los fabricantes de los equipos SS7. El estándar solo define la necesidad de tales procedimientos y hace sugerencias de cómo ellos pueden ser enviados.

El objetivo de estos procedimientos es tratar con la congestión en la fuente, donde los mensajes son generados. Esto, de acuerdo, es en el nivel cuatro. El protocolo no tiene interacción, realmente, a este nivel, porque estas son funciones internas. Sin embargo, el protocolo dispara estas funciones internas.

Si un mensaje de transferencia-prohibida es recibido por un destino particular, el nivel tres interactuara a través de la administración de la red con el protocolo y dirigirá a nivel cuatro también. La comunicación del protocolo con estas funciones internas es completada mediante el uso de primitivas.

### **2.5.5.7 Procedimientos de administración de rutas**

La administración de rutas es utilizada por un punto de señalización para notificar a sus puntos de señalización adyacentes de problemas de encaminamiento, el problema de encaminamiento es usualmente atribuido a la perdida de un enlace de señalización o conjunto de enlaces, los cuales juntos comprenden una ruta.

El propósito de la administración de rutas es redirigir el tráfico alrededor de la ruta afectada. Esto es completado a través del uso de mensajes de transferencia. Los cuales identifican el destino fallido por código de punto e instruyen al punto de señalización sobre como reaccionar (*figura 2.22*)

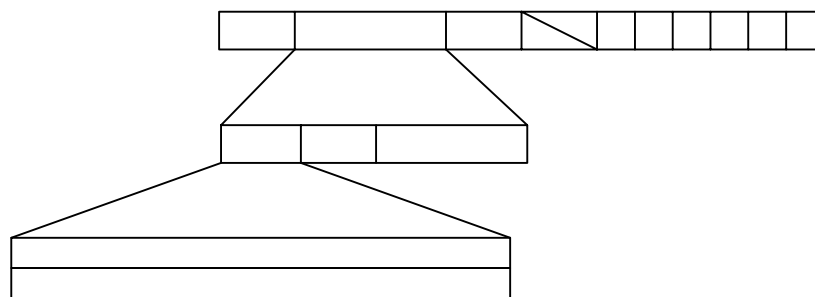


Figura 2.22 *Transferencia-prohibida y transferencia-cluster-prohibida*

Cuando una ruta falla, un mensaje de administración de ruta es enviado al punto de señalización adyacente para avisarle que el punto de señalización origen no puede alcanzar su destino a través de esa ruta y una alternativa debe ser seleccionada. Para administración de rutas normal, los siguientes mensajes son utilizados:

- Transferencia-prohibida (TFP)
- Transferencia-permitida (TFA)
- Transferencia-restringida (TFR)
- Transferencia-controlada (TFC)
- Señalización-ruta-ajuste-prueba (RST)
- Señalización-ruta-ajuste-congestión-prueba (RCT)

Durante la administración de rutas por cluster

- Transferencia-cluster-prohibida (TCP)
- Transferencia-cluster-permitida (TCA)
- Transferencia-cluster-restringida (TCR)
- Cluster-ruta-ajuste-prueba (CRST)

## **2.6 PROTOCOLO ISUP.**

### **2.6.1 Consideraciones generales**

Los mensajes de la parte usuario de RDSI se transportan en los enlaces de señalización mediante las unidades de señalización estudiadas en la *sección 2.3* de este capítulo.

El octeto indicador de servicio (SIO) para la parte usuario de RDSI se codifica 0101.

El campo de información de señalización (SIF) de cada unidad de señalización de mensaje que contiene un mensaje de la parte usuario de RDSI está constituido por un número entero de octetos y tiene los siguientes componentes:

- Etiqueta de encaminamiento
- Código de identificación de circuito
- Código de tipo de mensaje
- Parte obligatoria de longitud fija
- Parte obligatoria de longitud variable
- Parte facultativa, que puede contener campos de parámetros de longitud fija y de longitud variable.

La siguiente figura ilustra la estructura de los mensajes de la parte de usuario:

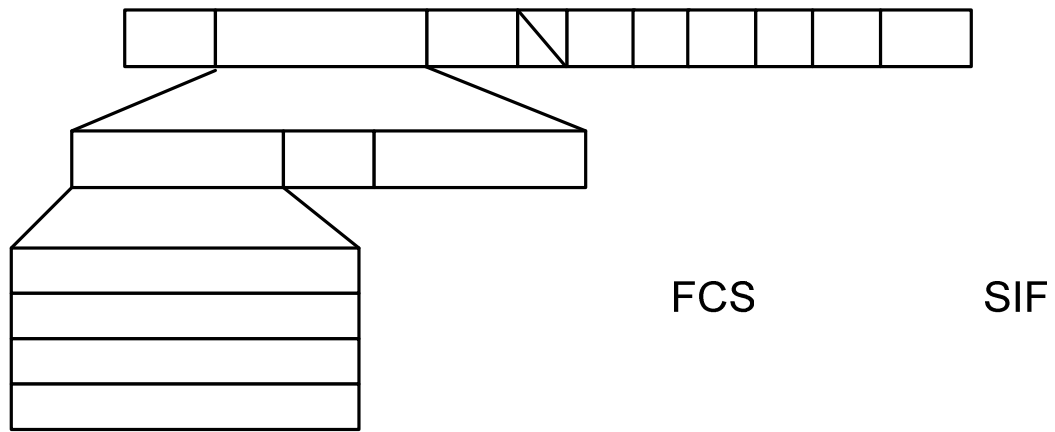


Figura 2.23 Estructura del mensaje ISUP dentro de las MSUs.

A continuación son descritos, cada uno de los elementos anteriormente listados:

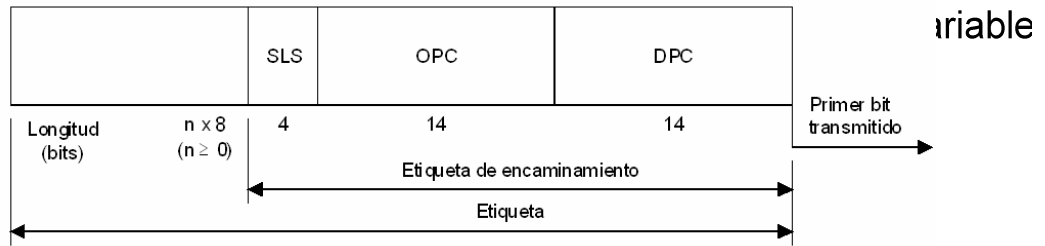
Información de Señalización  
 CIC

### 2.6.1.1 Etiqueta de encaminamiento

La etiqueta contenida en un mensaje de señalización y utilizada por la parte de usuario, para identificar la tarea específica a la que se refiere el mensaje (por ejemplo, circuito telefónico) es utilizada también por la parte de transferencia de mensajes para encaminar el mensaje hacia su punto de destino. Etiqueta usada por la discriminación de mensajes (sección 2.5.5.1.1) cuyos elementos son detallados a continuación, y para efectos de ilustración se ha vuelto a copiar la figura 2.15 en esta sección.

Código de Tipo de Mensaje  
 Parte Obligatoria de Longitud Fija

Parte Obligatoria de Longitud Variable



DPC Código de punto de destino  
 OPC Código de punto de origen  
 SLS Selección de enlaces de señalización

Figura 2.24 Etiqueta de encaminamiento.

**El código de punto de destino** (DPC, *destination point code*) indica el punto de destino del mensaje y el **código de punto de origen** (OPC, *originating point code*) señala el punto de origen del mensaje. Dentro de cada campo, el bit menos significativo ocupa la primera posición y se transmite en primer lugar.



**El campo de selección de enlaces de señalización** (SLS, *signalling link selection*) se utiliza, para efectuar la repartición de la carga. Este campo existe en todos los tipos de mensajes y ocupa la misma posición. La única excepción a esta regla es la de algunos mensajes del nivel 3 de la parte de transferencia de mensajes (por ejemplo, la orden de paso a enlace de reserva). En el caso de los mensajes de la parte de transferencia de mensajes, nivel 3, el campo de selección del enlace de señalización corresponde exactamente al código del enlace de señalización (SLC, *signalling link code*) que indica el enlace de señalización entre el punto de destino y el punto de origen al que se refiere el mensaje.

En cada conexión de circuito ha de utilizarse la misma etiqueta de encaminamiento en cada mensaje transmitido para esa conexión.

### 2.6.1.2 Código de identificación de circuito

El formato del código de identificación de circuito (CIC) se muestra en la *figura 2.25*.

8	7	6	5	4	3	2	1
Código de identificación de circuitos (bits menos significativos)							
Reserva				CIC (bits más significativos)			

Figura 2.25 *Campo de identificación de circuito.*

Para atribuir códigos de circuitos individuales, primero se deben adoptar acuerdos bilaterales, o reglas predeterminadas para aplicar en los intercambios de MSUs.

Para aplicaciones internacionales, los cuatro bits de reserva se utilizan para ampliar el CIC, siempre que se obtenga un acuerdo bilateral antes de que se aumente el tamaño. Para aplicaciones nacionales, los cuatro bits de reserva pueden utilizarse cuando se necesiten.

Para circuitos derivados de un trayecto digital a 2048 kbit/s, el código de identificación de circuito contiene, en los cinco bits menos significativos, una representación binaria del time slot asignado al trayecto de comunicación.

Los bits restantes del código de identificación de circuito se utilizan, cuando es necesario, para distinguir únicamente el circuito de todos los circuitos de otros sistemas que interconectan el punto de origen y el de destino.

### 2.6.1.3 Código de tipo de mensaje

El código de tipo de mensaje consta de un octeto y es obligatorio para todos los mensajes. El código de tipo de mensaje define la función del mensaje y el formato de cada mensaje de la parte usuario de RDSI.

Los tipos de mensajes se indican en la siguiente tabla (*Tabla 2.2*).

Tipo de mensaje	Código
Dirección completa	00000110
Respuesta	00001001
Bloqueo	00010011
Acuse de bloqueo	00010101
Progresión de la llamada	00101100
Bloqueo de grupo de circuitos	00011000
Acuse de bloqueo de grupo de circuitos	00011010
Indagación sobre grupo de circuitos @	00101010
Respuesta a indagación sobre grupo de circuitos @	00101011
Re-inicialización de grupo de circuitos	00010111
Acuse de re-inicialización de grupo de circuitos	00101001
Desbloqueo de grupo de circuitos	00011001
Acuse de desbloqueo de grupo de circuitos	00011011
Información sobre tasación @	00110001
Confusión	00101111
Conexión	00000111
Continuidad	00000101
Petición de prueba de continuidad	00010001
Facilidad @	00110011
Facilidad aceptada	00100000
Rechazo de facilidad	00100001
Petición de facilidad	00011111
Transferencia hacia delante	00001000
Petición de identificación	00110110
Respuesta de identificación	00110111
Información @	00000100
Petición de información @	00000011
Dirección inicial	00000001
Acuse de establecimiento de bucle @	00100100
Gestión de recursos de red	00110010
Sobrecarga @	00110000
Paso de largo @	00101000
Liberación	00001100
Liberación completada	00010000
Re-inicialización de circuito	00010010
Reanudación	00001110
Segmentación	00111000
Dirección subsiguiente	00000010
Suspensión	00001101
Desbloqueo	00010100
Acuse de desbloqueo	00010110
CIC no equipado @	00101110
Parte usuario disponible	00110101
Prueba de parte usuario	00110100
Información de usuario a usuario	00101101
Códigos reservados (utilizados en la versión 1984)	00001010 00001011 00001111 00100010 00100011 00100101 00100110
Códigos reservados (utilizados en la versión 1988)	00011101 00011100 00011110 00100111

Tabla 2.2 *Tipos de mensaje ISUP.*

#### **2.6.1.4 Parte obligatoria de longitud fija**

La parte obligatoria de longitud fija, contiene los parámetros que son obligatorios y que tienen una longitud fija para cada tipo de mensaje. La posición, longitud y orden de estos parámetros está definido por el tipo de mensaje. Por lo tanto, los nombres de los parámetros y los indicadores de longitud no se incluyen en el formato general del mensaje.

#### **2.6.1.5 Parte obligatoria de longitud variable**

En los parámetros obligatorios de longitud variable, se utilizan punteros para indicar el principio de cada parámetro. Cada puntero se codifica con un byte. El nombre de cada parámetro, el orden en que se envían los punteros y el número de punteros, está definido por el tipo de mensaje.

En esta parte, también se incluye un puntero que tiene la función indicar el principio de la parte facultativa, *si un tipo de mensaje no tiene parte facultativa, este puntero no aparecerá*. En el caso de un mensaje que sí contiene parte facultativa pero no es considerada, entonces el puntero parecerá con todos sus bits puestos a cero.

*Todos los punteros se envían consecutivamente al principio de la parte obligatoria de longitud variable.*

Cada parámetro de esta parte obligatoria de longitud variable contiene: el indicador de longitud del parámetro y el contenido del parámetro.

En el caso de que no existan parámetros variables obligatorios, pueden existir parámetros facultativos, por lo que en los parámetros variables obligatorios se incluye el puntero de parámetros facultativos (codificado 00000000 si no hay parámetros facultativos, y codificado 00000001 si existe alguno).

#### **2.6.1.6 Parte facultativa**

La parte facultativa está constituida por parámetros que pueden o no estar presentes dependiendo del tipo de mensaje. Esta parte puede contener parámetros de longitud fija y parámetros de longitud variable. Un parámetro facultativo no puede aparecer en múltiples ocasiones adentro de un mensaje. Los parámetros facultativos no tienen un orden de transmisión. Cada uno de estos parámetros estará constituido por:

- El nombre de parámetro (un octeto)
- El indicador de longitud (un octeto)
- El contenido del parámetro.

Si en un mensaje, existen parámetros facultativos, al terminar la transmisión de todos, se transmite el octeto de fin de parámetros facultativos, con todos los bits puestos a cero. Si en el mensaje no hay parámetros facultativos, entonces este octeto no se transmite.

## 2.6.2 Principios de formación de los mensajes

Cada mensaje está constituido por un número de PARÁMETROS. Cada parámetro tiene un NOMBRE que se codifica como un solo octeto. La longitud de un parámetro puede ser fija o variable; el mensaje puede tener también un INDICADOR DE LONGITUD, de un octeto, para cada parámetro.

### 2.6.2.1 Formatos y códigos de los parámetros

#### *Códigos de tipo de parámetros*

La codificación del tipo de parámetros se indica en la siguiente tabla (*tabla 2.3*).

Tabla 2.3 *Parámetros en los mensajes ISUP*

Nombre del parámetro	Código
Información de distribución de acceso	00101110
Transporte de acceso	00000011
Nivel automático de congestión	00100111
Indicadores de llamada hacia atrás	00010001
Información de desvío de llamada	00110110
Información de historial de llamada	00101101
Referencia de llamada @	00000001
Número de la parte llamada	00000100
Número de la parte llamante	00001010
Categoría de la parte llamante	00001001
Indicador de estado de circuito @	00100110
Indicador de causa	00010010
Indicador de tipo de mensaje de supervisión de grupo de circuitos	00010101
Código de enclavamiento de grupo cerrado de usuarios	00011010
Número conectado	00100001
Petición de conexión	00001101
Indicadores de continuidad	00010000
Información de protección contra el eco	00110111
Fin de parámetros facultativos	00000000
Información de suceso	00100100
Indicador de facilidad	00011000
Indicadores de llamada hacia delante	00000111
Indicadores de cobro revertido automático (reservado)	01000001
Dígitos genéricos @	11000001
Notificación genérica	00101100
Número genérico	11000000
Referencia genérica (reservado)	01000010
Contador de saltos (reservado)	00111101
Indicadores de información @	00001111
Indicadores de petición de información @	00001110
Número de localización	00111111
Indicador de petición MCID	00111011
Indicador de respuesta MCID	00111100
Información de compatibilidad de mensaje	00111000
Precedencia con apropiación multi-nivel (MLPP)	00111010
Indicadores de la naturaleza de la conexión	00000110
Facilidades específicas de la red @	00101111
Indicadores de llamada hacia atrás facultativos	00101001
Indicadores de llamada hacia adelante facultativos	00001000
Número llamado original	00101000

Código de punto ISC de origen	00101011
Información de compatibilidad de parámetros	00111001
Contador de retardo de propagación	00110001
Gama y estado	00010110
Número re-direccionante	00001011
Información de re-direccionamiento	00010011
Número de re-direccionamiento	00001100
Restricción de número de re-direccionamiento	01000000
Operaciones a distancia @	00110010
Activación del servicio @	00110011
Código de punto de señalización @	00011110
Número subsiguiente	00000101
Indicadores de suspensión/reanudación	00100010
Selección de red de tránsito @	00100011
Requisitos del medio de transmisión	00000010
Requisitos del medio de transmisión prima	00111110
Medio de transmisión utilizado	00110101
Información de servicio de usuario	00011101
Información de servicio de usuario primo	00110000
Información de tele servicio de usuario	00110100
Indicadores de usuario a usuario	00101010
Información de usuario a usuario	00100000
Códigos reservados (utilizados en la versión de 1984, Libro Rojo)	00010100 00011001 00011011 00011100 00011111 00100101
Reservado para identificador de intervalos múltiples	
Códigos reservados (utilizados en la versión de 1988, Libro Azul)	00010111
NOTA – Por el momento no se proporciona el formato.	

Tabla 2.4 *Parámetros en los mensajes ISUP (Continuación).*

### ***Codificación de los punteros***

Este puntero indica el número de octetos que hay entre el propio puntero y el primer octeto de los parámetros asociados a este puntero. Un puntero que contenga todos ceros, indica que no hay parámetros presentes.

### ***Formato de mensajes***

Cada tipo de mensaje tiene un formato detallado, En la *Figura 2.26* se muestra un diagrama del formato general. Entre parámetros no deben existir octetos sin utilizar.

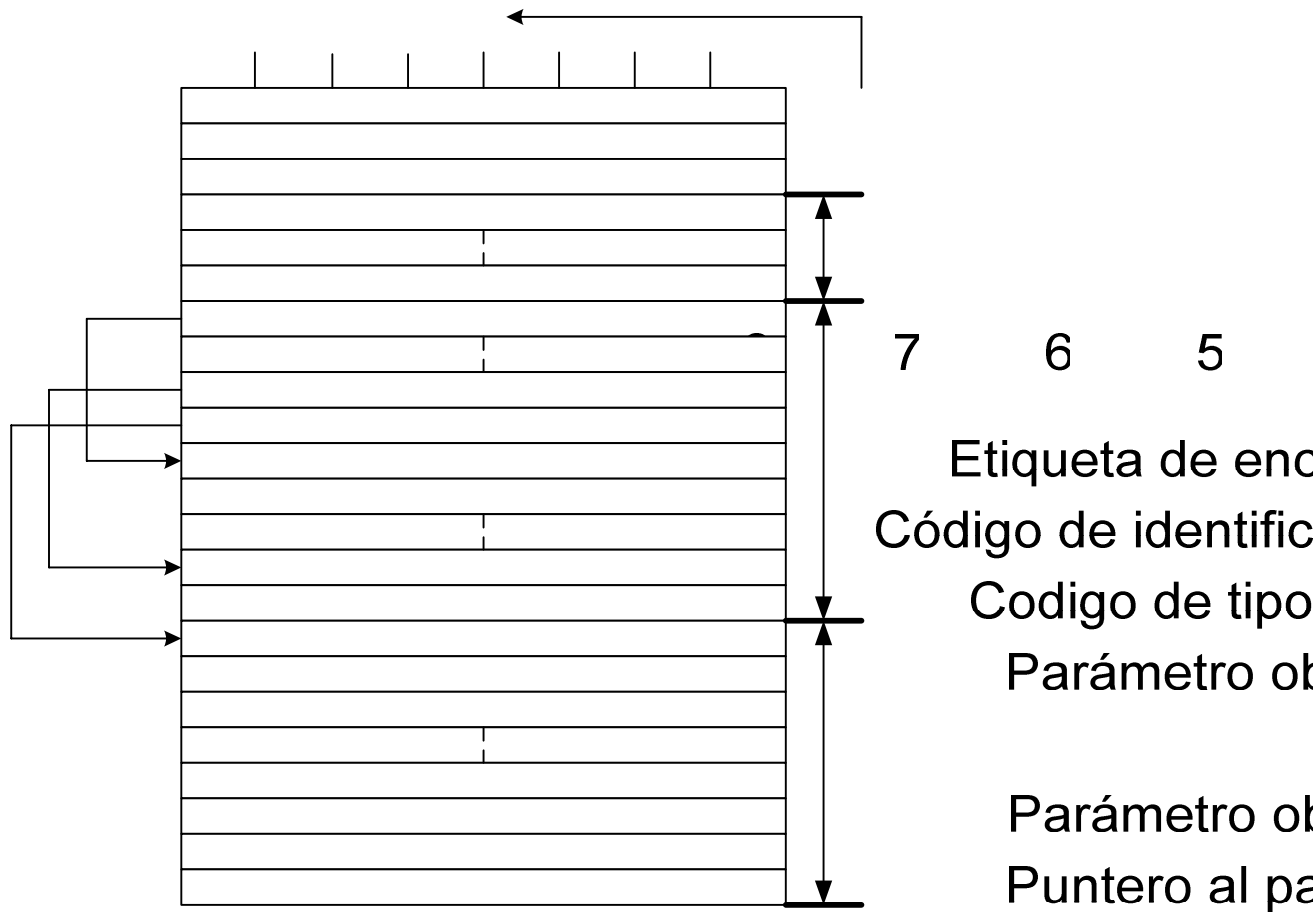


Figura 2.26 *Formato General de mensaje ISUP.*

### *Orden de transmisión*

Dado que todos los campos están constituidos por un número entero de octetos, los formatos pueden representarse como de octetos colocados en forma de pila. El primer octeto transmitido es el de la parte superior de la pila, y el último transmitido es el de la parte inferior (ver *Figura 2.26*). A menos que se indique otra cosa, dentro de cada octeto y sub-campo se transmite primero el bit menos significativo.

## **2.6.3 Principales mensajes usados para el establecimiento de una llamada de voz**

### **2.6.3.1 IAM**

El IAM proporciona la información del circuito el cual incluye la identificación del portador (*carrier*), y algún requerimiento especial considerado para una llamada. Este mensaje se usa para ejemplificar la estructura de los mensajes en ISUP por la cantidad de parámetros que contiene. En la *figura 2.27* se muestra la estructura de un mensaje IAM

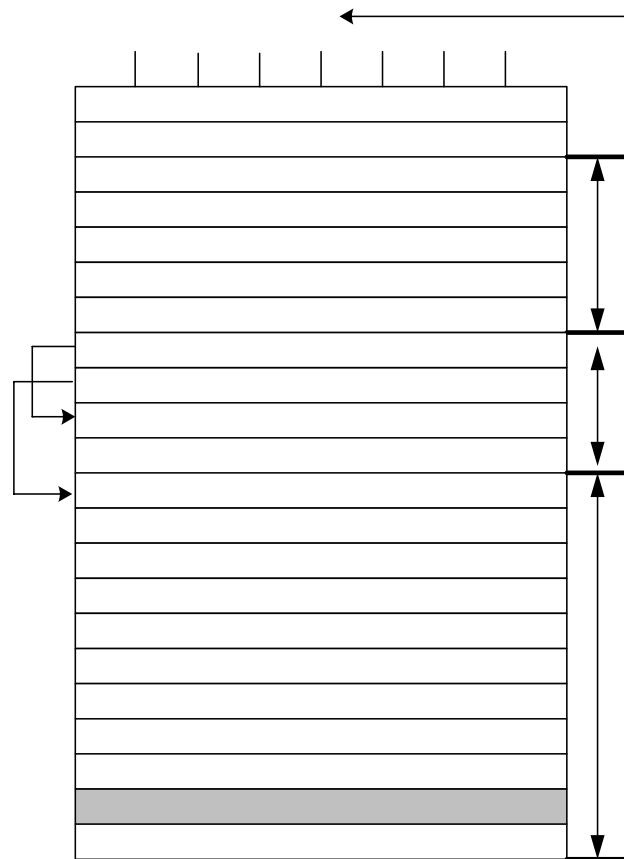


Figura 2.27 Formato del IAM.

### 2.6.3.2 ACM

El ACM es enviado a una central telefónica distante después de recibir las señales de direccionamiento (IAM y alguna otra información) necesarias para establecer la conexión entre dos centrales telefónicas.

El ACM indica que la llamada está siendo procesada, y la central telefónica distante está chequeando la disponibilidad de la parte llamada. Esto podría significar que el teléfono de la parte llamada está timbrando. La figura 2.28 muestra la estructura del mensaje

8 7 6 5 4

Etiqueta de encaminamiento

Código de identificación de la parte llamada

Mensaje Inicial de Dirección

Indicador de la naturaleza de la llamada

Indicadores de llamada horizontal

Categoría de la parte llamada

Requisitos del medio de transmisión

Puntero al número de la parte llamada

Puntero al principio de la parte llamada

Indicador de longitud del número de la parte llamada

Nombre de parámetro = Selección de red de destino

Indicador de longitud de la selección de red de destino

Nombre de parámetro = Número de la parte llamada

Indicador de longitud de la parte llamada

Parte llamada

Nombre de parámetro = Petición de conexión

Indicador de longitud de la petición de conexión

Petición de conexión

Otros parámetros facultativos

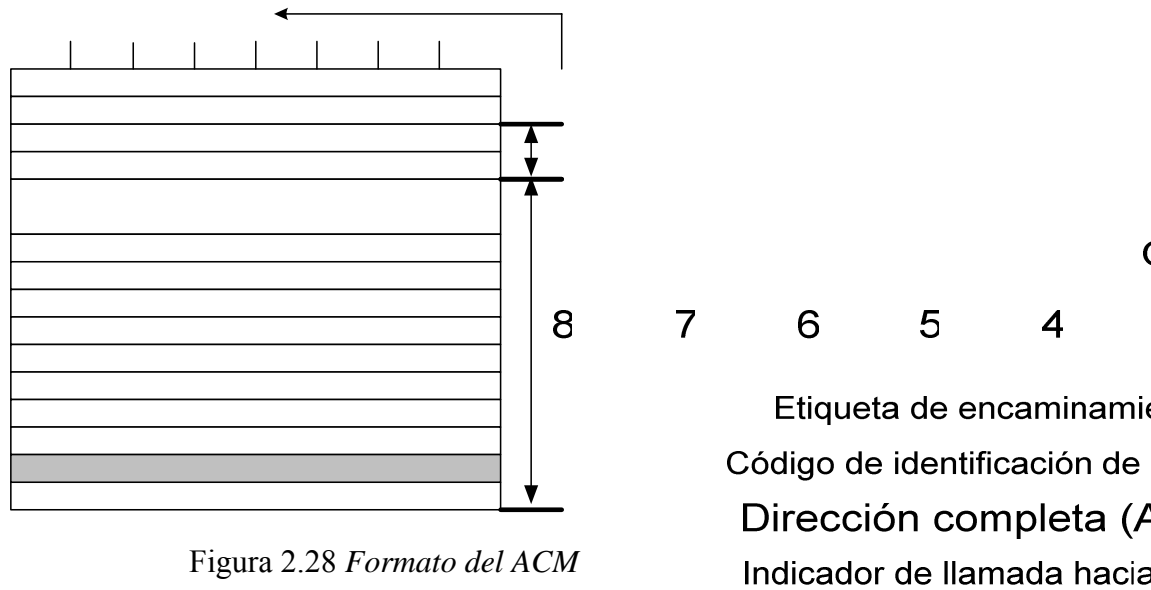


Figura 2.28 Formato del ACM

### 2.6.3.3 ANM

El ANM es enviado en la dirección de hacia atrás para indicar a la parte llamada que ha respondido. En redes semi-automáticas, este parámetro es utilizado para supervisión de llamadas. En redes automáticas el ANM es utilizado para medir las llamadas con el propósito de facturar. La figura 2.29 muestra la estructura básica de un ANM

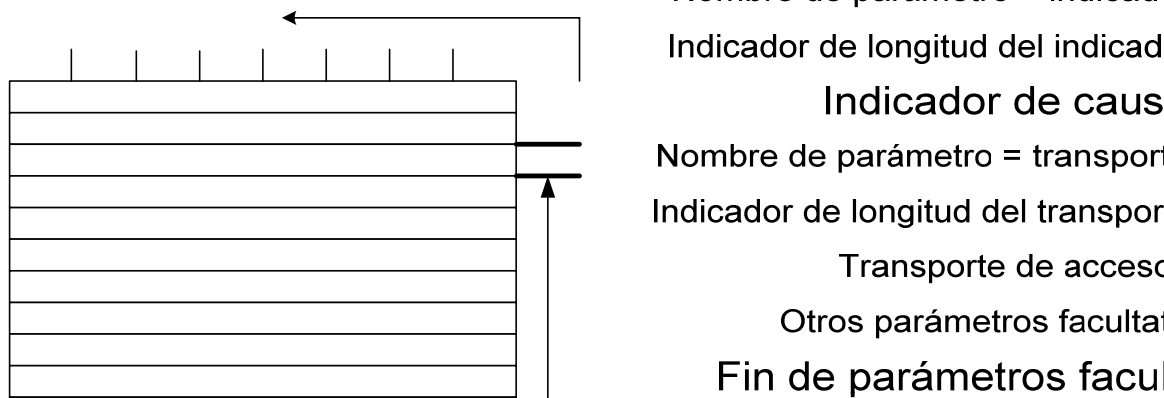


Figura 2.29 Formato de la estructura de un ANM.

### 2.6.3.4 REL

Este es enviado en cualquier dirección indicando que cualquiera de las dos partes (llamado o llamante) ha colgado el teléfono. El REL no retorna el circuito a su estado



desocupado, sin embargo, una liberación completada (RLC) debe ser recibido antes de que el circuito retorne a su estado ocioso. En la *figura 2.30* se ilustra el formato de un REL.

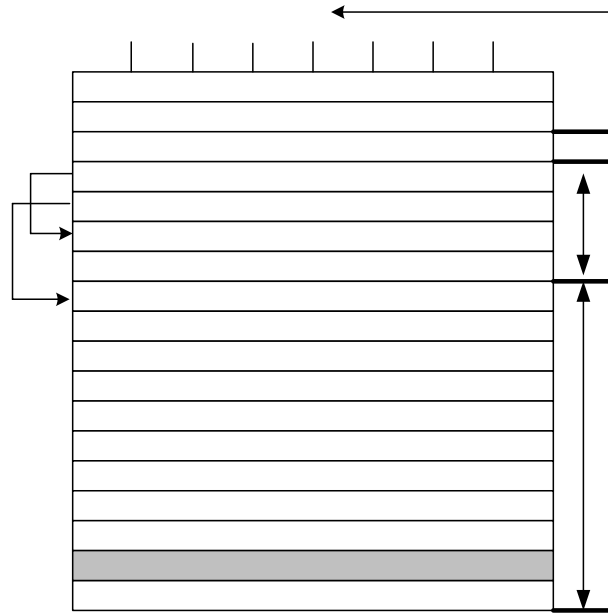


Figura 2.30 Formato del mensaje REL.

### 2.6.3.5 RLC

Este es utilizado para indicar que se recibió un REL y sirve como un reconocimiento de la liberación del circuito. Una vez se recibe un RLC el circuito utilizado es liberado y pasa a un estado de inactividad. La *figura 2.31* muestra su estructura.

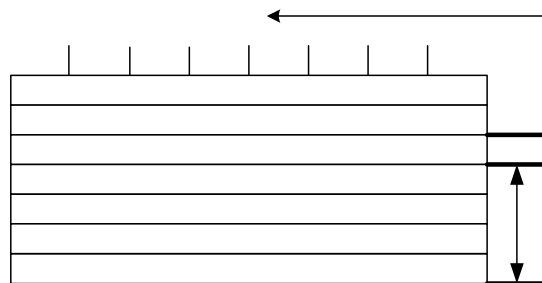


Figura 2.31 Estructura del RLC.

### 2.6.4 Ejemplo del establecimiento de una llamada Básica

El siguiente ejemplo demuestra como el sistema de señalización #7, y específicamente la parte de usuario ISUP, realiza el establecimiento de una llamada básica, y en la cual se aplicaran los tipos de mensajes discutidos con anterioridad

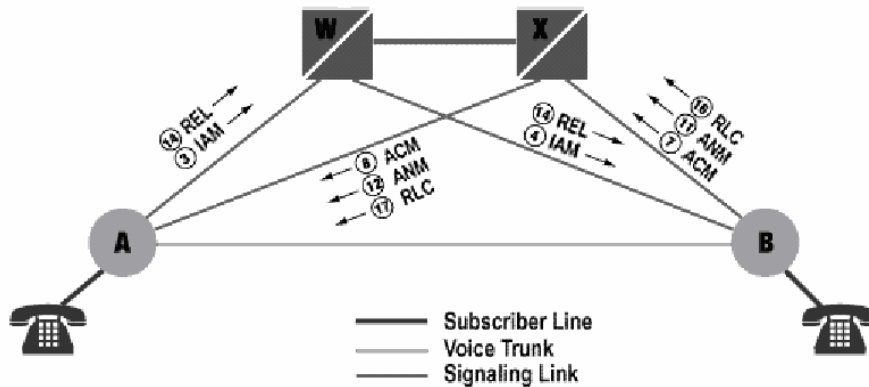


Figura 2.32 Ejemplo del establecimiento de una llamada

En este ejemplo, un suscriptor en el conmutador A, hace una llamada a un suscriptor en el conmutador B.

1. El conmutador A analiza los dígitos marcados y determina que necesita para enviar la llamada al conmutador B.
2. El conmutador A selecciona un camino inactivo entre él y el conmutador B y formula una dirección de mensaje inicial (IAM), que es el mensaje básico para iniciar una llamada. EL IAM es enviado al conmutador B, este identifica el conmutador origen (conmutador A) y el conmutador destino (conmutador B), el camino seleccionado, el numero del que llama y los números marcados, así como otra información mas allá de el alcance de este ejemplo.
3. El conmutador A toma uno de estos enlaces A y transmite el mensaje sobre el enlace para enviarlo al conmutador B
4. STP W recibe un mensaje, inspecciona la etiqueta de dirección y determina que este debe ser enviado al conmutador B. Este transmite el mensaje en el enlace BW.
5. El conmutador B recibe el mensaje. Analiza el mensaje, determina que este sirve a la llamada y que el numero llamado esta desocupado.
6. El conmutador B formula un mensaje de direccionamiento completo (ACM). El cual indica que el AIM alcanzo su destino. El mensaje identifica el conmutador destinatario (A), el conmutador origen (B), y el camino seleccionado.
7. El conmutador B toma uno de estos enlaces A, y transmite el ACM sobre el enlace para enviarlo al conmutador A, al mismo tiempo este completa el camino de la

llamada en la dirección contraria (hacia el conmutador B), enviando un tono de timbrado sobre el mismo camino hacia el conmutador A y tonos de timbrado a la línea del subscriptor que esta llamando.

8. El STP X recibe el mensaje, inspecciona la etiqueta de dirección, y determina que este debe ser enviado al conmutador A. Este transmite el mensaje en el enlace AX.
9. Recibiendo el ACM, el conmutador A conecta al subscriptor que esta llamando a la línea en el camino seleccionado, en dirección del conmutador B (Para que el que llama pueda escuchar el tono de timbrado enviado por el conmutador B).
10. Cuando el subscriptor llamado levanta el teléfono, el conmutador B formula un mensaje respuesta (ANM), identificando la intención de recibir en el conmutador A, el conmutador que envía el mensaje (B), y el camino seleccionado.
11. El conmutador B selecciona el mismo enlace usado por el conmutador A para transmitir el ACM (Enlace BX), y envía el ANM. Para este tiempo, el enlace debe estar conectado a la línea de la llamada en ambas direcciones (para permitir la conversación).
12. El STP X reconoce que el ANM se ha enviado al conmutador A y envía esto sobre el enlace AX.
13. El conmutador A asegura que el subscriptor que llama esta conectado en el camino saliente (en ambas direcciones) y la conversación puede tomar lugar.
14. Si el subscriptor que llama cuelga primero (siguiendo la conversación), el conmutador A debe generar el mensaje de liberar (REL) enviándolo al conmutador B, identificando el camino asociado con la llamada. Este envía el mensaje en el enlace AW.
15. El STP W recibe el REL, determina que éste esta diseccionado al conmutador B y sigue usando el enlace WB.
16. EL conmutador B recibe el REL, desconecta el camino seleccionado de la línea del subscriptor, retorna el camino a su estado de inactividad, genera un mensaje de liberación completa (RLC) enviado al conmutador A, y lo transmite en el enlace BX. El RLC identifica el camino usado para llevar la llamada.
17. STP X recibe el RLC, determina que este se ha enviado al conmutador A y lo envía sobre el enlace AX.
18. Recibiendo el RLC, el conmutador libera el camino identificado.

## 2.7 CONCLUSIONES DEL CAPITULO II

La red de señalización #7 se basa en la conmutación de mensajes, por los cuales se hace posible el establecimiento de llamadas, los puntos de señalización realizan las funciones de generación y recepción de mensajes en la red y están localizados en las centrales telefónicas. Todos los puntos de señalización #7 son interconectados vía enlaces de señalización de datos.

El protocolo SS7 difiere del modelo OSI. El modelo OSI consiste de siete capas diferentes, C7 utiliza solamente 4 capas. Las funciones ejecutadas por estos cuatro niveles corresponden con las siete capas del modelo OSI. Algunas funciones utilizadas por el modelo OSI no tienen propósito en las redes SS7 y por lo tanto no se definen.

La capa física en sistema de señalización #7 es virtualmente la misma que la del modelo OSI, el nivel dos esta dedicado únicamente a la transmisión de datos de un nodo al siguiente nodo dentro de una red, el nivel tres proporciona las funciones de discriminación, encaminamiento y distribución de mensajes, El nivel 4 consiste de varios protocolos diferentes, todos llamados partes de usuario y parte de aplicación. Para establecer una llamada telefónica, se utilizan la parte usuaria telefónica (TUP) o el protocolo ISUP.

El sistema de señalización # 7 es una red basada en la conmutación de paquetes por lo que los datos transmitidos son agrupados en paquetes, la conmutación de paquetes ordena todos los datos en un paquete y los transmite en la forma de un paquete completo, proporcionando toda la información necesaria para procesar los datos. El sistema de señalización # 7 usa tres diferentes estructuras para los paquetes, llamadas unidades de señalización (FISU, LSSU, MSU), la FISU es la Unidad de señalización que provee el nivel de servicio mas bajo, no proporciona ninguna información de mensaje, pero contiene una mínima cantidad de información necesaria para mantener el enlace, La FSSU es la unidad de señalización que es enviada entre dos puntos de señalización para indicar el estado del enlace que transmite la señalización y la MSU proporciona la estructura para transmitir mensajes a los protocolos de capas superiores.

## **2.8 REFERENCIA BIBLIOGRÁFICA**

[1] Travis L Russell. “Sistem Signaling #7” McGraw Hill

[2] Calderón Osorio, Daniel Alberto. “Diseño de instrumentos virtuales para medir señalización siete por canal común de la Unión Internacional de Telecomunicaciones (CCS7UIT) en tiempo real”. Universidad de El Salvador. 2003.

[3] UIT-T. Descripción funcional de la parte usuario de RDSI del sistema de señalización n.º 7; Rec. Q.761; Marzo/93.

[4] UIT-T. Descripción funcional de la parte transferencia de mensajes del sistema de señalización n.º 7, Rec. Q.701; marzo/93.

[5] UIT-T. Sistema de señalización n.º 7 – funciones y mensajes en la red de señalización; Rec. Q.704; marzo/93.

[6] UIT-T . Formatos y códigos de la parte usuario de red digital de servicios integrados del sistema de señalización n.º 7; Rec. Q.763; marzo/93.

## CAPITULO III

### DISEÑO DEL CIRCUITO DE INTERFAZ

#### 3.1 INTRODUCCION

El siguiente capítulo trata del diseño de un circuito electrónico capaz de funcionar como interfase entre un canal de señalización (enlace E1) y una computadora, de tal forma que los datos que se transporten por el canal, producto de la comunicación entre dos nodos de señalización, sean monitoreados por un programa para su posterior análisis.

Este dispositivo cumple con las normas UIT-T anteriormente presentadas, y debe ser tal que no provoque ningún tipo de interferencia en el canal de señalización de forma que su uso sea transparente para el sistema de señalización.

Se presentara un diagrama general del dispositivo así como también la explicación detallada de cada uno de los bloques que lo constituyen, diagramas esquemáticos y de tiempo de las señales manejadas por estos. Al final se presentan las especificaciones técnicas del dispositivo.

#### 3.2 DIAGRAMA DE BLOQUES GENERAL

El circuito de interfaz tiene como propósito capturar la información de comunicación entre dos centrales telefónicas o nodos que se transporta una línea E1 y enviar esta información hacia un PC. Para cumplir con esta función, el circuito debe realizar distintos procesos, entre los cuales están acoplamiento de impedancias, recuperación de reloj, decodificación de HDB3 a TTL, sincronización con la trama E1 y enviar los datos a un puerto específico de la computadora.

El trabajo de diseño del circuito se realizó como lo explica el diagrama de bloques que se muestra en la Figura 3.1 *Diagrama de bloques de circuito interfaz.*

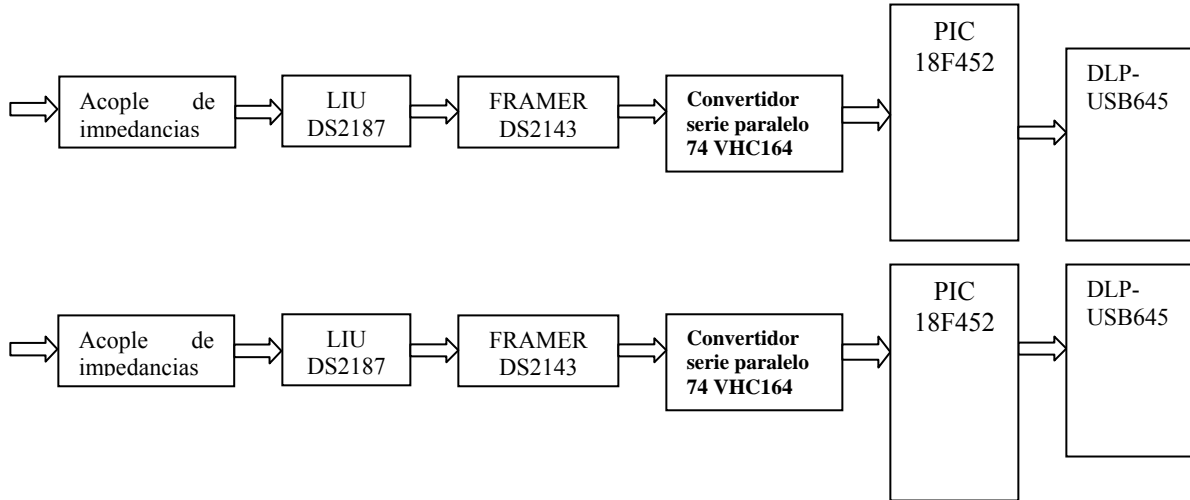


Figura 3.1 *Diagrama de bloques de circuito interfaz.*

El circuito consta de dos canales idénticos tomados de la comunicación de dos nodos, en función de monitorear la transmisión y recepción de datos en uno de los nodos.

La explicación de cada uno de los bloques del circuito se presenta a continuación en orden según la etapa de diseño.

### 3.3 CIRCUITO DE ACOUPLE

La etapa de acoplamiento, tal como se aprecia en el esquema de bloques, está en contacto directo con la línea E1. Su importancia radica no solo en el hecho que es precisamente aquí donde los datos son adquiridos, sino también en que es la parte que garantiza que el equipo cumple la característica de ser transparente, es decir, que no perturbe la información entre el transmisor y receptor.

Las señales de entrada y salida en esta etapa están codificadas en HDB3 y deben cumplir con la máscara presentada en la norma G.703. El circuito de acoplamiento se ha diseñado para trabajar con una línea de  $75\Omega$  desbalanceada, común en los sistemas de transmisión.

El circuito de acople de línea (figura 3.2) fue tomado de la hoja de datos del circuito integrado DS2187 para diseño de interfaces E1/T1. Está compuesto por un transformador y un arreglo de resistencias. El transformador provee de aislamiento magnético, es decir, elimina cualquier componente de corriente continua que pueda haber en la línea y filtra componentes de alta frecuencia. Los valores de las resistencias son tales que la impedancia de entrada de la etapa de acople de la interfaz de línea sea vista por la línea como una impedancia de  $75\Omega$ . El transformador utilizado en la implementación de este circuito es el Schott 22166, especialmente diseñado para líneas E1/T1.

El transformador Schott 22166 tiene una relación de 1:2, con tab-central, los componentes envueltos en el circuito de acople son las resistencias  $R_1$  y  $R_2$  (como se muestra en la figura 3.2). Para aplicaciones de CEPT  $R_1$  y  $R_2$  deben ser de  $150\Omega$  para una impedancia de  $75\Omega$  y  $240\Omega$  para impedancias de  $120\Omega$ .

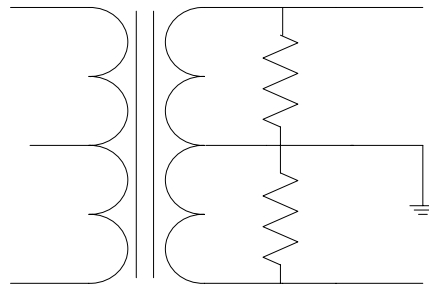


Figura 3.2 Circuito de acoplamiento.

### 3.4 INTERFAZ DE LINEA E1 (DS2187)

### Entrada Positiva

Esta etapa se encarga del proceso más complejo dentro del aparato, que incluye recuperación de reloj y decodificación de los datos serie HDB3 a niveles TTL/CMOS y formato NRZ. Estas funciones son implementadas por el circuito integrado DS2187 (del fabricante: DALLAS SEMICONDUCTOR). La figura 3.3 muestra el diagrama de bloques interno de este circuito integrado.

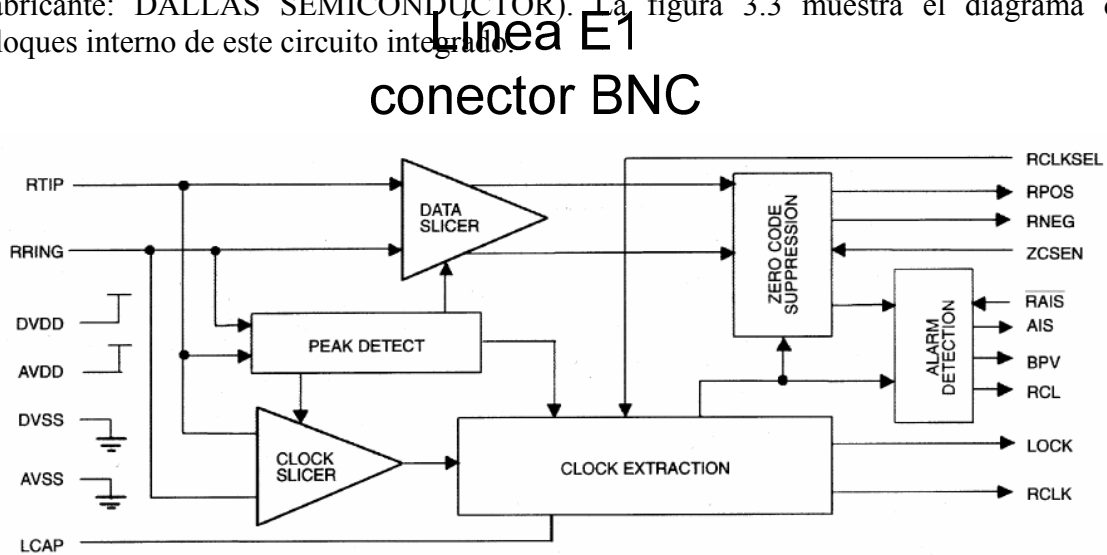


Figura 3.3 Diagrama de bloques interno del DS2187.



El DS2187 es un integrado para ser usado como línea interfase en redes de comunicaciones, ya sea para T1 (1.544MHz) o E1 (2.048MHz). El dispositivo extrae reloj y datos de un par trenzado o un cable coaxial y elimina la necesidad de componentes caros ó cambios manuales cuando existen terminaciones T1 y E1.

El DS2187 recibe la señal HDB3 en las terminales RTIP (positiva) y RRing (negativa). A estos pines se conecta la salida del circuito de acoplamiento. El integrado puede también trabajar con señales en código AMI, por lo que es necesario configurarlo para trabajar en código HDB3 poniendo en alto (= 1) el pin RCLKSEL.

Como se detallo en el circuito de etapa de acople, las señales de entrada al DS2187 son acopladas por medio de un transformador de 1:2 con derivación central. R1 y R2 son puestos a 150Ω para aplicaciones en E1 a 75Ω. Una circuitería especial en las entradas RTIP y RRIN permite señales negativas por debajo de Vss.

### **3.4.1 Detector de picos y deslizamientos:**

Los pulsos presentes en RTIP y RRING son muestreados por un circuito detector de picos. El voltaje de umbral para las señales de reloj y los datos capturados esta definido a un 50% de el voltaje de pico muestreado.

Los niveles de pico en la entrada de RTIP y RRING deben exceder 0.6volts para establecer un umbral mínimo, señales debajo de este nivel causaran RCL a las transiciones en alto después de 192 bits.

### **3.4.2 Extracción de reloj:**

El DS2187 utiliza lazos de frecuencia cerrada (FLL) y de Fase digital cerrada (DPLL) para recuperar datos y reloj de la señal AMI entrante. En aplicaciones T1 se utiliza un reloj de 18.528 MHz dividido por 11, 12 ó 13 para marcar la fase de las fluctuaciones de la señal entrante. En aplicaciones E1, un reloj de 24.576MHz dividido por 11, 12 o 13 proporciona detección de fluctuaciones. La salida DPLL es capturada en una etapa de buffer y presentada en RCLK. El FLL es un circuito el cual rastrea la frecuencia promedio de la señal entrante, minimizando el efecto de las fluctuaciones en la salida DPLL.

Durante el tiempo de adquisición o si el pin RLC pasa a alto, el pin LOCK podría ir a bajo para indicar perdida de sincronización de la señal entrante. Una vez este pin pasa a alto, el FLL activa la frecuencia cerrada y los datos validos son presentados en las salidas RPOS y RNEG.

### 3.4.3 Supresión de código Cero

El dispositivo puede decodificar palabras de código B8ZS (RCLKSEL=0) o HDB3 (RCLKSEL=1) y reemplazar con todos ceros cuando ZCSEN=1. Cuando ZCSEN=0 las palabras de código pueden pasar a través del integrado sin ser alteradas.

### 3.4.4 Detección de alarma

Los datos extraídos son monitoreados por la red por condiciones de alarma y error. RLC es puesto a uno cuando ocurren 192 ceros consecutivos; este es limpiado en el siguiente uno. AIS es puesto a uno cuando menos de tres ceros aparecen en RPOS y RNEG durante los últimos dos periodos de la señal RAIS, una vez puesto a uno, AIS puede permanecer en 1 para los dos siguientes periodos de RAIS. AIS retornara a cero cuando mas de dos ceros aparezcan. BPV reporta violaciones bipolares cuando estas ocurren en RPOS y RNEG; Código B8ZS puede no ser reportado en BPV cuando ZCSEN=1.

El diagrama de la etapa de interfaz de línea se presenta en la figura 3.4 (se han omitido los pines de configuración).

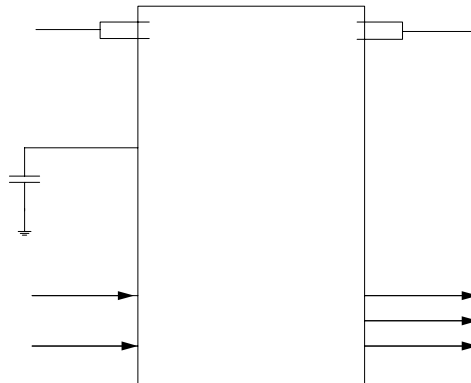


Figura 3.4 Diagrama del circuito de Interfaz de línea.

La figura 3.5 muestra el diagrama de tiempo de la salida de datos en el pin RPOS, RNEG y el reloj recuperado en RCLK. Estas son las señales que recibe la siguiente etapa de la interfase.

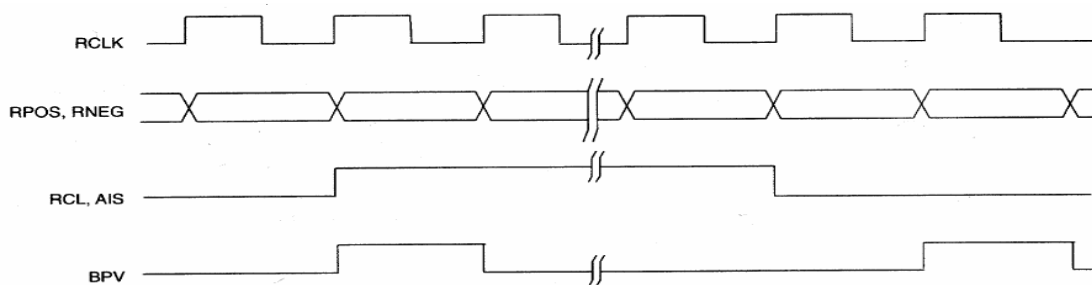


Figura 3.5 Diagrama de tiempo de las señales de salida del DS2187

5V

### 3.5 ETAPA DE FRAMER

Para esta etapa se ha utilizado el framer DS2143, El DS2143 es un framer E1 manejado por software, este puede actuar como un esclavo o como coprocesador con un microcontrolador o con un microprocesador. El DS2143 es bastante flexible y puede ser configurado en muchas orientaciones vía software. El controlador contiene un conjunto de 69 registros internos, cada uno de 8 bits, a los cuales se puede tener acceso. Estos registros internos son usados para configurar el dispositivo y obtener información de un enlace E1. El dispositivo conoce las últimas especificaciones para E1, incluyendo las recomendaciones CCITT G.704, G.706 Y G.732.

El DS2143 es un controlador E1 que tiene 4 secciones principales: el lado de recepción, el lado de transmisión, el controlador de la línea interfase y el puerto paralelo de control. Ver figura 3.6 donde se muestra el diagrama de bloques. En el lado de recepción, el dispositivo puede sincronizar el tren serial E1 vía los pines RPOS y RNEG. La sincronización puede localizar los patrones de trama y multitrama y establecer sus respectivas posiciones. Esta información es usada por el resto del circuito interfase.

El DS2143 es un framer “fuera de línea”, lo cual significa que todo el tren E1 que entra al dispositivo puede salir de este sin ningún cambio. Una vez el tren de datos E1 esta en el framer, los datos de señalización pueden ser extraídos. Un almacenamiento de dos tramas puede ser habilitado o ignorado.

#### **Características del DS2143:**

- Puerto paralelo de control
- Circuito de Almacenamiento elástico de dos tramas
- Extracción e inserción de señalización CAS
- Secciones de transmisión y recepción totalmente independientes
- Completa detección de alarmas
- Acceso a los bits Si y Sa
- Detección de pérdidas en reloj transmisor
- Codificación y decodificación HDB3
- Transmisión totalmente transparente.
- Contadores de error.
- Circuito de soporte bit a bit para los bits Sa
- Salida de reloj programable.
- Generación de trama de sincronización.
- Realimentación local.
- Soporte automático de CRC-4.
- Detección de pérdidas en el reloj de recepción.
- Soporte para E1 y T1 según G.802

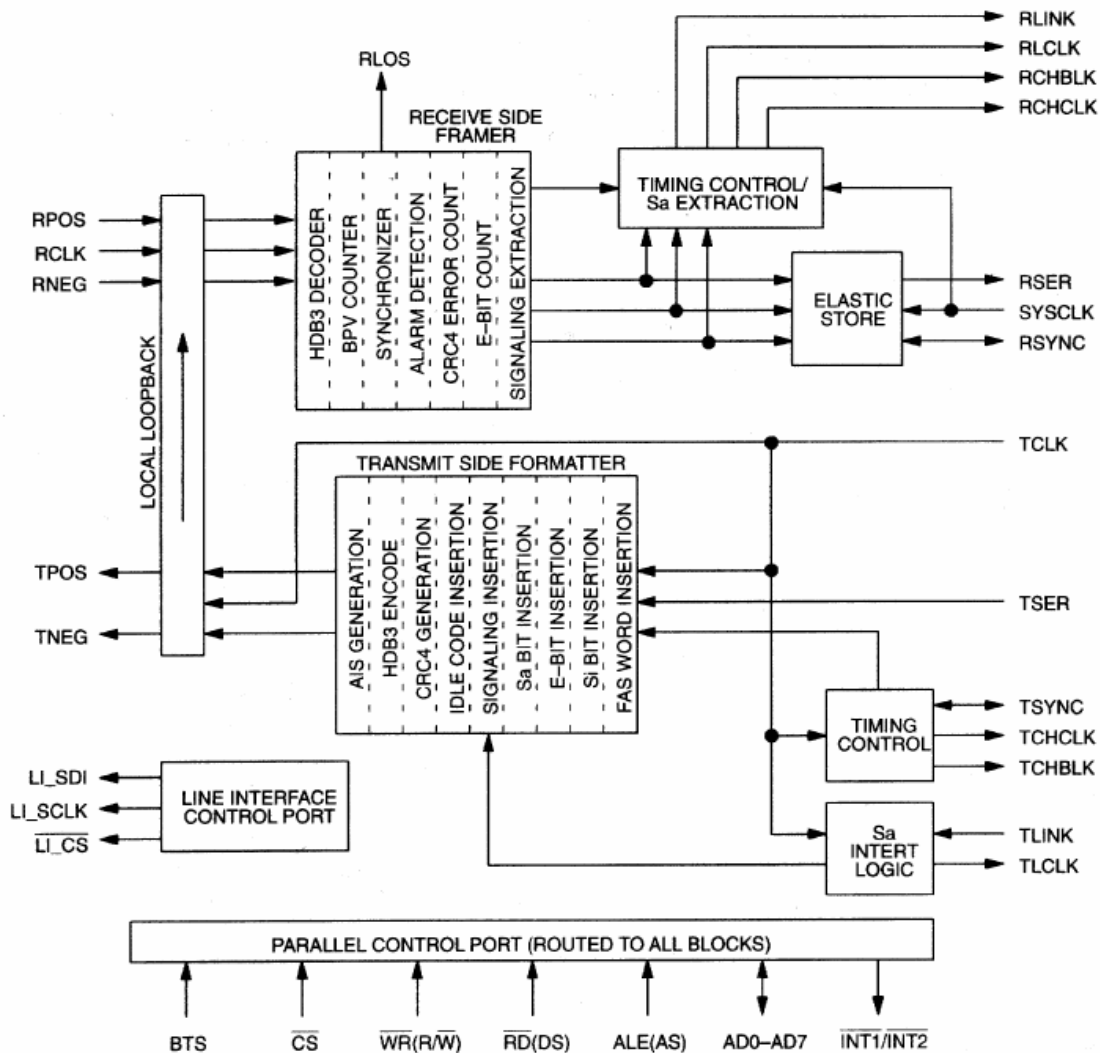


Figura 3.6 Diagrama de bloques del DS2143.

### 3.5.1 PUERTO PARALELO

En DS2143 el puerto paralelo es controlado por medio de un bus bi-direccional multiplexado en direcciones/datos que puede ser controlado por un microcontrolador o microprocesador externo. El DS2143 puede operar con configuración de tiempos *Intel* o *Motorola*. Si el pin BTS es puesto en bajo, la configuración *Intel* es seleccionada; si es puesto en alto, la configuración *Motorola* es seleccionada. Ver en la figura 3.7 y 3.8 los diagramas de tiempo para configuración *Intel*. El bus multiplexado tiene sus propios pines porque las señales de direcciones y de datos comparten el mismo camino. Las direcciones son presentadas a los pines en la primera porción del ciclo de bus y los datos pueden ser presentados a los pines en la segunda porción del ciclo de bus. Las direcciones deben ser validas previo el borde de bajada de ALE(AS), en el tiempo de los latches de direcciones del DS2143 en los pines AD0 a AD7. La escritura valida debe ser presentada

y sostenida estable durante la última porción del pulso DS o WR. En un ciclo de lectura, el DS2143 coloca el byte de datos en la última porción del pulso DS o RD. El ciclo de lectura es terminado y el bus retorna a un estado de alta impedancia como RD en las transiciones en alto en la configuración de tiempos Intel o como DS las transiciones a bajo en la configuración de *Motorola*.

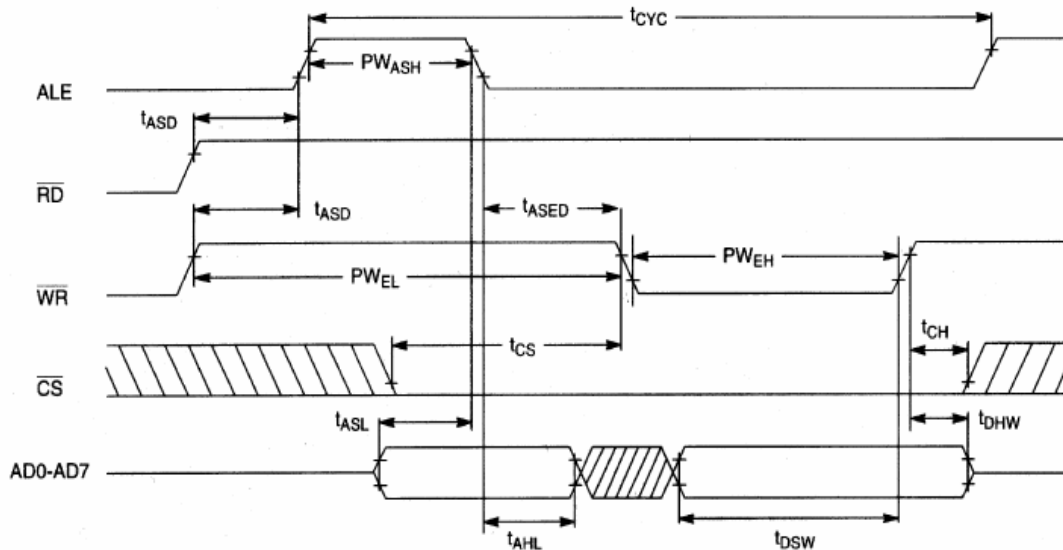


Figura 3.7 Tiempo de escritura del DS2143

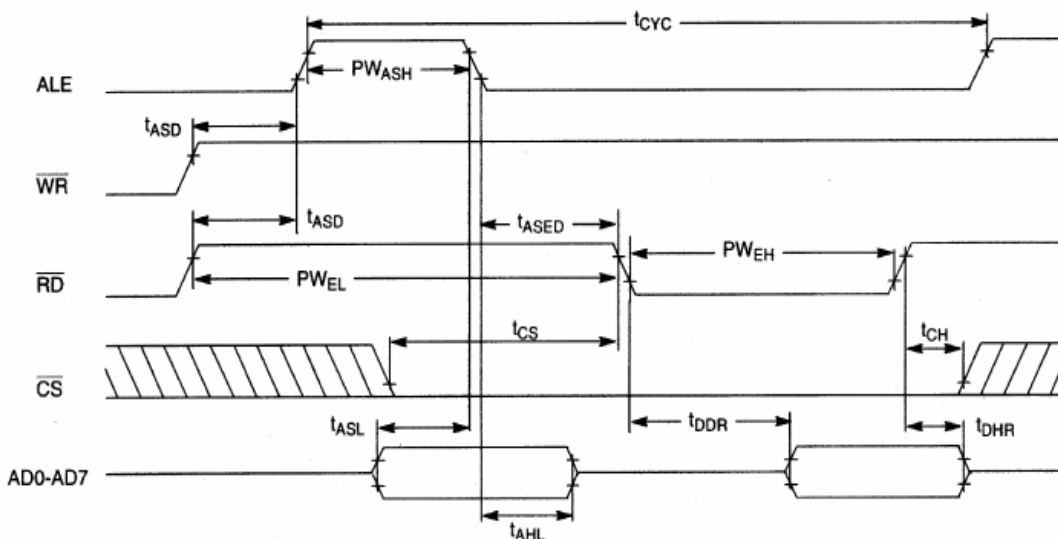


Figura 3.8 Tiempo de lectura del DS2143

### 3.5.2 CONTROL Y PRUEBA DE REGISTROS

La operación del DS2143 se configura por medio de cinco registros. Los registros de control son acezados únicamente cuando el sistema es encendido. Una vez el DS2143 ha sido inicializado, se podría necesitar acceder a los registros de control únicamente si hay cambios en la configuración del sistema. Hay dos registros de control de recepción (RCR1 y RCR2), dos registros de control de transmisión (TCR1 y TCR2) y un registro de control común (CCR). A continuación se describen los registros de control de recepción y el registro de control común.

<b>(MSB)</b>				<b>(LSB)</b>			
RSMF	RSM	RSIO	-	-	FRC	SYNCE	RESYNC

Tabla 3.1 *Registro 1 de control de recepción* (dirección=10 Hex)

SIMBOLO	POSICION	NOMBRE Y DESCRIPCION
RSMF	RCR1.7	<b>RSYNC Función de sincronización de multitrama.</b> Únicamente usado si el pin RSYNC es programado en el modo de multitrama (RCR1.6=1). 0 = RSYNC salidas CAS en los limites de multitrama 1 = RSYNC salidas CRC4 en los limites de multitrama
RSM	RCR1.6	<b>RSYNC Selección de modo.</b> 0 = Modo de trama (ver en la hoja de datos los diagramas de tiempo) 1 = Modo multitrama (ver en la hoja de datos los diagramas de tiempo)
RSIO	RCR1.5	<b>RSYNC selección de I/O.</b> 0 = RSYNC es una salida (depende de RCR1.6) 1 = RSYNC es una entrada (únicamente valido si el almacenamiento elástico de dos tramas es habilitado) (nota: este bit debe ser puesto a cero cuando RCR2.1=0)
-	RCR1.4	<b>No asignado.</b> Debe ser puesto cero cuando es escrito.
-	RCR1.3	<b>No asignado.</b> Debe ser puesto cero cuando es escrito.
FRC	RCR1.2	<b>Criterio de re-sincronización de trama.</b> 0 = Re-sincroniza si la FAS es recibida en error 3 veces consecutivas 1 = Re-sincroniza si la FAS o el bit 2 de no-FAS es recibida en error 3 veces consecutivas.

SYNCE	RCR1.1	<b>habilitación de sincronización.</b> 0 = auto re-sincronización habilitada 1 = auto re-sincronización deshabilitada
RESYNC	RCR1.0	<b>Re-sincronización.</b> Cuando pasa de bajo a alto, una Re-sincronización es inicializada. Debe ser limpiado y puesto de nuevo para subsecuentes re-sincronizaciones.

FRAME OR MULTIFRAME LEVEL	CRITERIO DE SINCRONIZACION	CRITERIO DE RESINCRONIZACION	ITU SPEC.
FAS	FAS presente en tramas N y N+ 2, y NFAS presentes en tramas en tramas N + 1.	Tres FAS incorrectas consecutivas recibidas.  Alternando (RCR1.2=1) al criterio anterior, o tres bit2 de la NFAS recibidas incorrectamente.	G.706 4.1.1 4.1.2
CRC4	Dos palabras validas de Alineación de MF encontradas en 8ms	915 o mas CRC4 palabras de código fuera de 1000 recibidas en error.	G.706 4.2 4.3.2
CAS	Encontrada una palabra de alineación de MF y previo al time slot 16 contiene código de todos ceros.	Dos palabras de alineación de MF consecutivas recibidas en error.	G.732 5.2

Tabla 3.2 *Criterio de Sincronización/ re sincronización.*

(MSB)						(LSB)	
Sa8S	Sa7S	Sa6S	Sa5S	Sa4S	SCLKM	ESE	-

Tabla 3.3 *Registro 2 de control de recepción (dirección=11 Hex)*

SIMBOLO	POSICION	NOMBRE Y DESCRIPCION
Sa8S	RCR2.7	<b>Selección de bit Sa8.</b> Puesto a 1 reporta el bit Sa8 en el pin RLINK; Puesto a 0 no reporta el bit Sa8.

Sa7S	RCR2.6	<b>Selección de bit Sa7.</b> Puesto a 1 reporta el bit Sa7 en el pin RLINK; Puesto a 0 no reporta el bit Sa7.
Sa6S	RCR2.5	<b>Selección de bit Sa6.</b> Puesto a 1 reporta el bit Sa6 en el pin RLINK; Puesto a 0 no reporta el bit Sa6.
Sa5S	RCR2.4	<b>Selección de bit Sa5.</b> Puesto a 1 reporta el bit Sa5 en el pin RLINK; Puesto a 0 no reporta el bit Sa5.
Sa4S	RCR2.3	<b>Selección de bit Sa4.</b> Puesto a 1 reporta el bit Sa4 en el pin RLINK; Puesto a 0 no reporta el bit Sa4.
SCLKM	RCR2.2	<b>Selección del modo SYSCLK.</b> 0 = si SYSCLK es 1.544 MHz. 1 = si SYSCLK es 2.048 MHz.
ESE	RCR2.1	<b>Habilitar el almacenamiento Elástico.</b> 0 = almacenamiento Elástico es deshabilitado. 1 = almacenamiento Elástico es habilitado.
-	RCR2.0	<b>No Asignado.</b> Debe ser puesto a cero cuando se escribe.

(MSB)							(LSB)
LLB	THDB3	TG802	TCRC4	RSM	RHDB3	RG802	RCRC4

Tabla 3.4 *Registro común de control (dirección =14 hex)*

<b>SIMBOLO</b>	<b>POSICION</b>	<b>NOMBRE Y DESCRIPCION</b>
LLB	CCR.7	<b>Realimentación local.</b> 0 = Realimentación deshabilitada. 1 = Realimentación habilitada.
THDB3	CCR.6	<b>Transmisión de HDB3 habilitada.</b> 0 = HDB3 deshabilitada. 1 = HDB3 habilitada.



TG802	CCR.5	<b>Transmisión G.802 habilitada.</b> (ver en el datasheet los diagramas de tiempo) 0 = no fuerza TCHBLK en alto durante el bit 1 del time slot 26. 1 = fuerza TCHBLK en alto durante el 1 del time slot 26.
TCRC4	CCR.4	<b>Transmisión CRC4 habilitada.</b> 0 = CRC4 deshabilitada. 1 = CRC4 habilitada.
RSM	CCR.3	<b>Selección de modo de señalización.</b> 0 = modo de señalización CAS. 1 = modo de señalización CCS.
RHDB3	CCR.2	<b>Recepción de HDB3 habilitada.</b> 0 = HDB3 deshabilitada. 1 = HDB3 habilitada.
RG802	CCR.1	<b>Recepción G.802 habilitada.</b> 0 = no fuerza RCHBLK en alto durante el bit 1 del Time slot 26. 1 = fuerza RCHBLK en alto durante el 1 del time slot 26.
RCRC4	CCR.0	<b>Recepción de CRC4 habilitada.</b> 0 = CRC4 deshabilitada. 1 = CRC4 habilitada.

Además del modo de operación normal (transmisión/recepción), el DS2143 posee un modo de prueba que es la realimentación local, (CCR.7 = 1). Esta realimentación local es ampliamente usada en pruebas y aplicaciones de depuración. En realimentación, el DS2143 toma los datos del lado de transmisión al lado de recepción. Esta realimentación es sinónimo de reemplazar la señal de entrada RCLK con TCLK, y las entradas RPOS/RNEG con las salidas TPOS/TNEG.

Cuando la realimentación es habilitada ocurre lo siguiente:

1. los datos en RPOS y RENG son ignorados.
2. todos los datos en el lado de recepción tomaran tiempo de sincronización con TCLK en lugar de RCLK.
3. todas las funciones son habilitadas.

### 3.5.3 REGISTROS DE ESTADO Y DE INFORMACION

Hay un conjunto de cuatro registros que contienen información en tiempo real del estado del DS2143: registro status 1(SR1), registro status 2(SR2), registro de información de recepción (RIR), y registro del estado de sincronización (SSR). Cuando un evento particular ocurre o esta ocurriendo, el bit apropiado en uno de estos tres registros es

puesto a uno, todos estos registros operan en forma de latched (excepto el SSR). Esto significa que si un evento ocurre y un bit es puesto a uno en cualquiera de estos registros, este permanecerá a uno hasta que el usuario lea este bit. El bit será limpiado cuando este sea leído y no será puesto a uno hasta que un evento ocurra nuevamente o si la condición de alarma todavía persiste.

El DS2143 desarrolla un monitoreo del enlace en las posiciones de los bits Sa, además proporciona acceso a cada uno de estos bits Sa por medio de dos registros internos (RNAF Y TNAF) o por medio de dos pines externos (RLINK y TLINK).

En recepción, los bits Sa son siempre reportados en el registro interno RNAF (ver sección 13 en la hoja de datos del DS2143 para más detalles). Los bits Sa son siempre reportados en el pin RLINK. Ver sección 13 para detalles del tiempo. Por medio de RCR2, el usuario puede controlar el pulso en el pin RLCLK, para cualquier combinación de los bits Sa. Esto permite al usuario crear un reloj que pueda ser usado para capturar los bits Sa que se necesiten.

### 3.5.4 RELOJ DE CANAL Y RELOJ DE BLOQUE DE REGISTROS

El DS2143 posee un pin de salida el cual indica cada canal con un reloj de 256KHz el cual se pone en alto durante el bit menos significativo de cada canal, este pin es ampliamente usado para la conversión serie paralelo de los canales de datos. También posee un reloj de bloque de registros que en recepción son los registros de bloque de canal (RCBR1/ RCBR2/ RCBR3/ RCBR4) y en transmisión los registros de bloque de canal (TCBR1/ TCBR2/ TCBR3/ TCBR4) son los que controlan los pines RCHBLK y TCHBLK respectivamente. Los pines de salida RCHBLK y TCHBLK son programables, pueden ser forzadas en alto o en bajo durante canales individuales. Estas salidas pueden ser usadas como reloj de canal para controlar o monitorear CCS. Cuando los bits apropiados son puestos a uno, los pines RCHBLK y TCHBLK pueden ser mantenidos en alto durante el tiempo correspondiente a un canal. Ver el diagrama de tiempo de la figura 3.9.

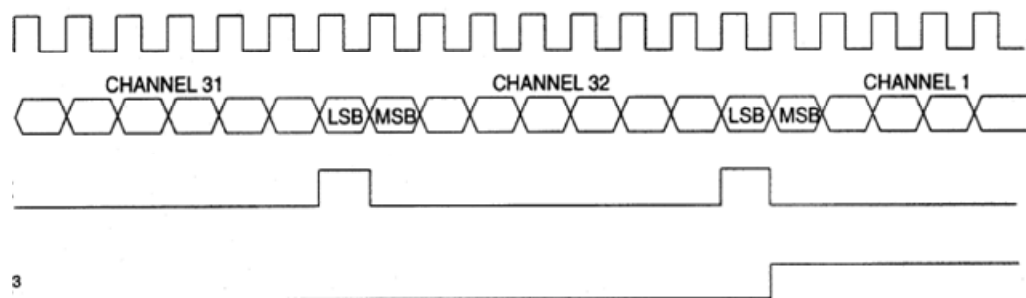


Figura 3.9 Diagrama de tiempo de RCHCLK y RCHBLK

El diagrama de la etapa de framer se presenta en la figura 3.10 (se han omitido los pines de configuración).

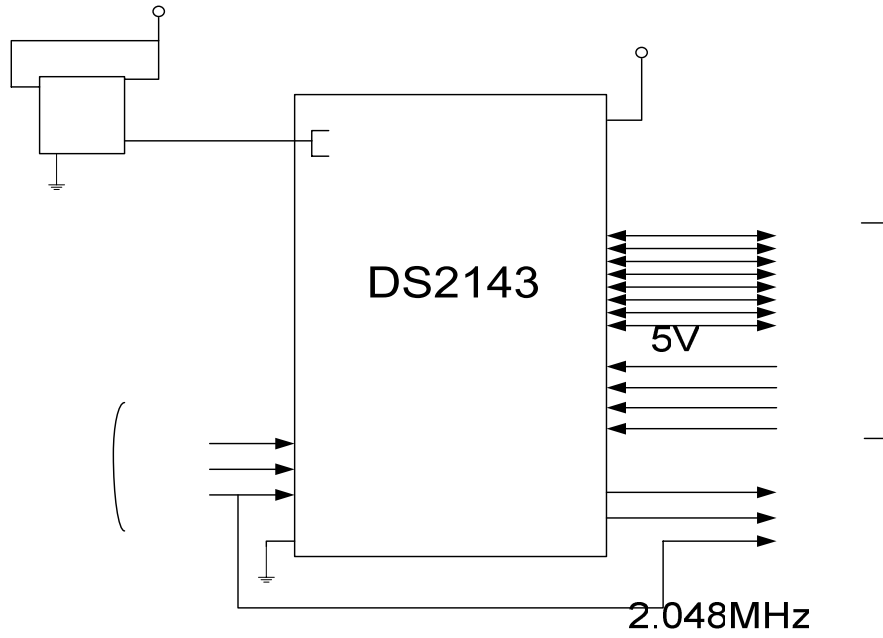


Figura 3.10 Diagrama del circuito del framer.

TCLK  
SYSCLK

### 3.6 ETAPA DE CONVERSIÓN SERIE PARALELO

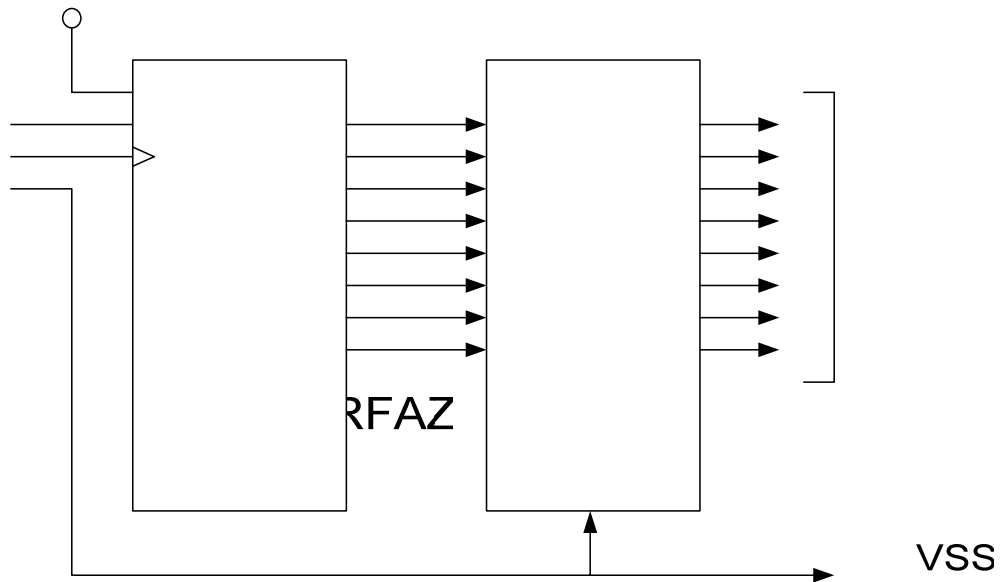


Figura 3.11 Circuito de la etapa de conversión serie paralelo

Esta etapa se encarga de convertir los datos serie, obtenidos del framer, en paralelos de 8 bits, para lo cual se utiliza el IC 74LS164. Dicha conversión se realiza en función de trabajar con datos paralelos por su facilidad de manejo, y para liberar al microcontrolador de dicha conversión y tener un margen de 39 instrucciones para poder realizar la captura de los datos, además de la existencia de interfaces paralelas a otros protocolos, como USB.

Esta etapa recibe el reloj recuperado y los datos serie provenientes del framer, convierte los datos a paralelo y los sincroniza con el RCHBLK para la posterior captura del canal deseado.

El 74VHC164 posee dos entradas de datos, A y B. Cualquiera de estas puede servir como entrada para el flujo de datos, únicamente debe ponerse la otra en 1 para que funcione. La entrada de reloj es activa en flanco de subida, y presenta el primer bit recibido en la entrada Q0 para luego irse desplazando a Q1, Q2, hasta Q7 a medida que se introducen nuevos bits. La entrada de reloj es, por supuesto, el reloj recuperado, el cual se identifica en el diagrama como RCLK.

Las salidas Q0-Q7 se conectan al latch de salida. En la interfaz E1, los bits de los octetos se envían siempre del más significativo al menos significativo, por tanto el bit más significativo (MSD) es Q7 y el menos significativo (LSD) es Q0.

### **3.6.1 Latch de salida**

Esta etapa consiste en la captura de los ocho bits en paralelo del convertidor serie-paralelo, en el momento adecuado en el cual se alinean los ocho bits del time slot deseado, esta captura se realiza cuando el pin RCHBLK pasa a alto (5V).

Para este objetivo se usa el IC 74LS573 cuya función es mantener en su salida el octeto durante el mayor tiempo posible, evitando que se presenten los valores intermedios. El 74LS573 posee la entrada LE (Latch Enable), que captura el dato presente en las entradas en el flanco de bajada, y lo mantiene en las salidas mientras  $LE = 0$ . Si  $LE = 1$ , entonces las salidas siguen a las entradas. LE se conecta al pin RCHBLK de el framer, el cual nos indica cuando esta presente el time slot deseado. Con esta configuración, el latch hará la captura cuando RCHBLK pase de alto a bajo, lo que nos indica la presencia del time slot deseado. Cabe mencionar que la señal de RCHBLK que alimenta a LE es lo suficientemente larga para permitir la copia de las entradas  $D_{0-7}$  hacia las salidas  $O_{0-7}$ , y ya que dicha señal solo dura poco tiempo en alto, la mayoría del tiempo se presentara a la salida del latch el octeto del ultimo time slot, actualizándose en cada disparo de RCHBLK.

## **3.7 ETAPA DEL PIC18F452**

### **3.7.1 Descripción General del microcontrolador PIC18F452**

#### **CARACTERISTICAS DEL PIC18F452**

##### **CPU tipo RISC de alto desempeño:**

- Conjunto de instrucciones compatibles con los PIC16 y PIC17.
- 32 kbytes de programa de memoria.
- 1.5 kbytes de memoria de datos.
- Operación a 10MPIS
  - Velocidad de operación: DC-40Mhz de entrada de reloj.
  - De 4MHz a 10MHz de entrada de reloj con PLL activo.
- 16 bit de ancho de instrucción, 8 bit de ancho de datos.
- Instrucciones de un solo ciclo, con excepción de los saltos (dos ciclos).
- Ciclo de instrucción: DC-100 ns.

##### **PERIFERICOS**

- Tres interrupciones externas.
- Temporizador 0: contador/temporizador de 8-bit/16-bit con pre-escalador de 8 bits.
- Temporizador 1: contador/temporizador de 16 bits.
- Temporizador 2: contador/temporizador de 8 bits con registro de periodo de 8 bits (tiempo basado por PWM).
- Temporizador 3: contador/temporizador de 16 bits.
- Opción de reloj oscilador secundario- Temporizador 1/ Temporizador 3.
- Dos módulos capturadotes/comparadores/PWM.
- Puerto serie síncrono con SPI y I<sup>2</sup>C.
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI)
- Modulo de conversión análogo-digital de 10 bits

##### **Características especiales:**

- 100,000 de ciclos de lectura/escritura de memoria de programa tipo FLASH ENHANCED.
- 1,000,000 de ciclos de lectura/escritura de memoria interna tipo EEPROM
- Retención de datos en EEPROM mayor de 40 años.

## **Configuración del Oscilador**

El PIC18F452 puede operar en ocho diferentes modos de oscilación, que se pueden programar mediante los bits FOSC2, FOSC1 Y FOSC0.

- 1) LP
- 2) XT
- 3) HS
- 4) HS + PLL
- 5) RC
- 6) RCIO
- 7) EC
- 8) ECIO

### **HS PLL**

Un circuito de lazo de fase cerrada es proporcionado como una opción programable, para poder multiplicar la frecuencia del cristal por cuatro. Para una entrada de 10MHz, la frecuencia interna podría ser de 40MHz. El PLL es únicamente habilitado cuando la configuración de bits son programados en modo HS.

### **3.7.2 Arquitectura Interna**

El PIC18F452A al igual que los demás miembros de su familia, se caracterizan por:

- Flujo de instrucciones PIPELINE
- Su procesador es tipo RISC
- Tiene una arquitectura HARVARD
- El formato de las instrucciones es ortogonal
- Todas las instrucciones tienen la misma longitud (16 bits)
- La arquitectura está basada en bancos de registros

En las siguientes sub-secciones aclararemos todos los conceptos anteriores

#### **3.7.2.1 Flujo de instrucciones PIPELINE**

Aplica la técnica de segmentación que permite al procesador realizar simultáneamente la ejecución de una instrucción y la búsqueda de código de la siguiente. Un ciclo de instrucción consiste de cuatro ciclos Q (Q1, Q2, Q3 y Q4), De esta manera, se puede ejecutar una instrucción en un ciclo. (Cada ciclo de instrucción son cuatro ciclos de reloj).

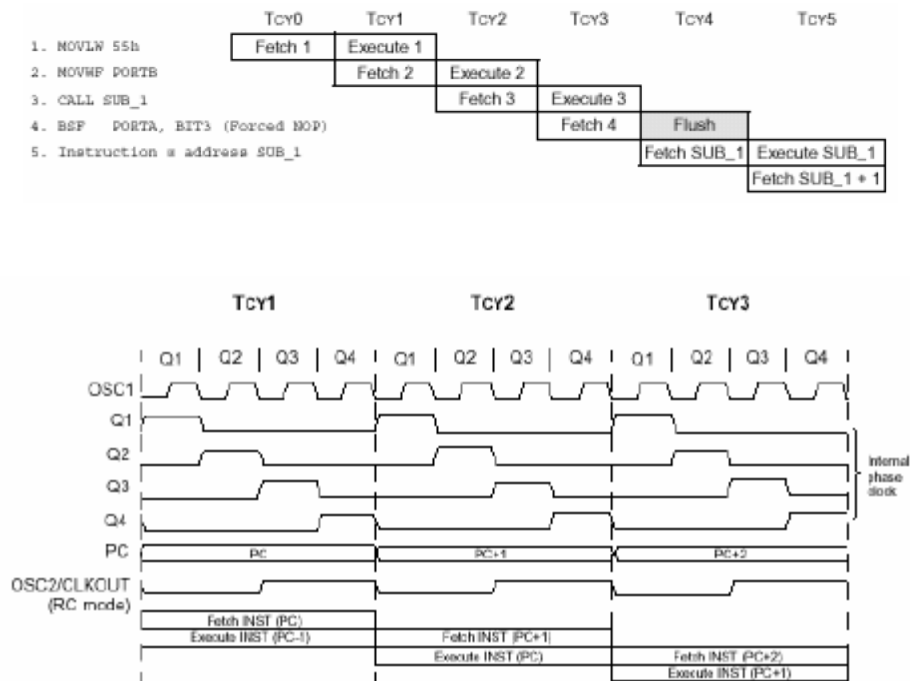


Figura 3.12 Diagrama de ciclos de Instrucción del PIC18F452

### 3.7.2.2 Procesador tipo RISC

Las CPU's atendiendo al tipo de instrucciones que utilizan pueden clasificarse en:

- **CISC:** (*Complex Instruction Set Computer*) Computadores de juego de instrucciones complejo, que disponen de un repertorio de instrucciones elevado (unas 80), algunas de ellas muy sofisticadas y potentes, pero que como contrapartida requieren muchos ciclos de máquina para ejecutar las instrucciones complejas.
- **RISC:** (*Reduced Instruction Set Computer*) Computadores de juego de instrucciones reducido, en los que el repertorio de instrucciones es muy reducido (en nuestro caso 75), las instrucciones son muy simples y suelen ejecutarse en un ciclo máquina. Además los RISC deben tener una estructura *pipeline* y ejecutar todas las instrucciones a la misma velocidad.
- **SISC.** (*Specific Instruction Set Computer*) Computadores de juego de instrucciones específico.

### 3.7.2.3 Arquitectura Harvard

Tradicionalmente los microprocesadores se basan en la estructura de Von Neumann, como la de la figura 3.12, que se caracteriza por disponer de una única memoria principal en la que se almacenan los datos y las instrucciones. A esta memoria se accede a través de un sistema de buses único:

- Bus de datos
- Bus de direcciones
- Bus de control

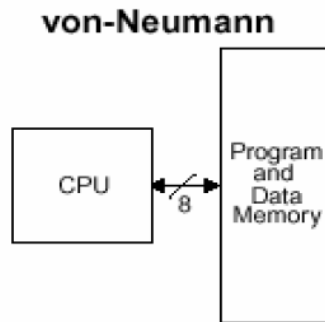


Figura 3.13 Estructura von Neumann

El modelo Harvard, representado en la figura siguiente, dispone de dos memorias:

- Memoria de datos
- Memoria de Programa

Además cada memoria dispone de su respectivo bus, lo que permite, que la CPU pueda acceder de forma independiente y simultánea a la memoria de datos y a la de instrucciones. Como los buses son independientes éstos pueden tener distintos contenidos en la misma dirección.

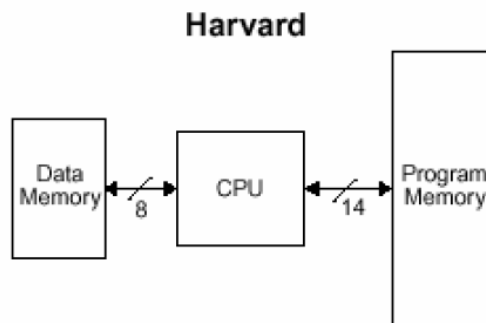


Figura 3.14 Estructura harvard



### 3.7.2.4 Arquitectura Ortogonal

Cualquier instrucción puede utilizar cualquier elemento de la arquitectura como fuente o destino.

### 3.7.2.5 Arquitectura basada en bancos de registros

Implica que todos los elementos del sistema, es decir, temporizadores, puertos de entrada/salida, posiciones de memoria, etc., están implementados físicamente como registros.

En los PIC el manejo del banco de registros, que participan activamente en la ejecución de las instrucciones, es muy interesante al ser ortogonales. La ALU (Unidad Aritmético-Lógica) efectúa sus operaciones con dos operándos, uno que proviene del registro W (Work, por su sigla en ingles), que en otras CPUs recibe el nombre de Acumulador, y el otro que se encuentra en cualquier otro registro o del propio código de instrucción.

En el PIC18f452 el mapa de memoria de datos esta dividida en 16 bancos que contienen 256 bytes cada uno. Los datos de memoria contienen registros de funciones especiales (SFR) y registros de propósito general (GPR). Los SFRs son usados para control y estado del controlador, y los GPRs son usados para almacenar datos y operaciones. El mapa de memoria se muestra a continuación:

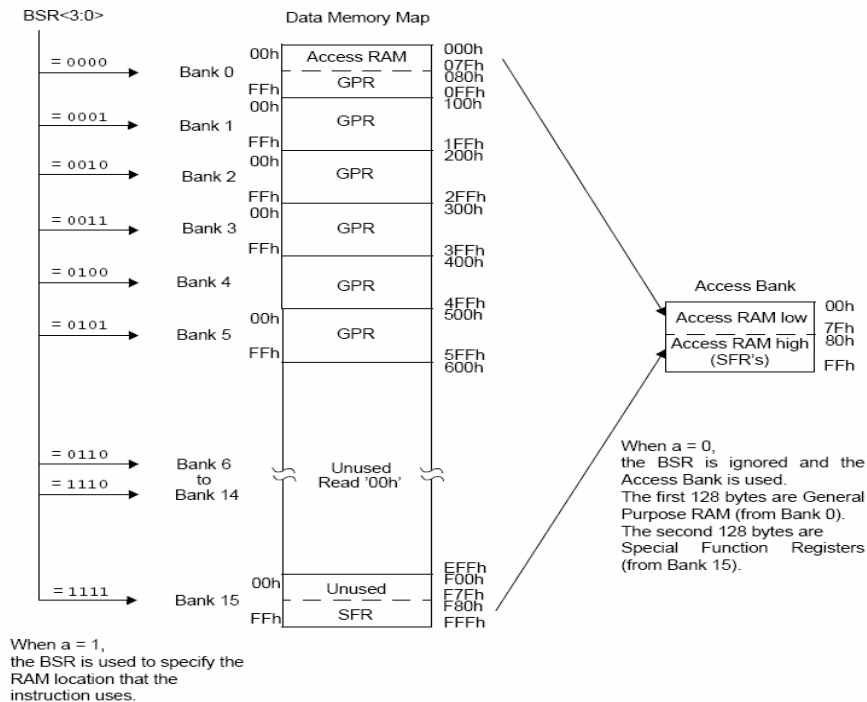


Figura 3.15 Mapa de memoria del PIC18F452

Address	Name	Address	Name	Address	Name	Address	Name
FFh	TOSU	FDh	INDF2 <sup>(3)</sup>	FBh	CCPR1H	F9h	IPR1
FEh	TOSH	FDEh	POSTINC2 <sup>(3)</sup>	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2 <sup>(3)</sup>	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2 <sup>(3)</sup>	FBCh	CCPR2H	F9Ch	—
FFBh	PCLATU	FDBh	PLUSW2 <sup>(3)</sup>	FBBh	CCPR2L	F9Bh	—
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	—
FF9h	PCL	FD9h	FSR2L	FB9h	—	F99h	—
FF8h	TBLPTRU	FD8h	STATUS	FB8h	—	F98h	—
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	—	F97h	—
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	—	F96h	TRISE <sup>(2)</sup>
FF5h	TABLAT	FD5h	T0CON	FB5h	—	F95h	TRISD <sup>(2)</sup>
FF4h	PRODH	FD4h	—	FB4h	—	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	LVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	—
FF0h	INTCON3	FD0h	RCON	FB0h	—	F90h	—
FEFh	INDF0 <sup>(3)</sup>	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—
FEEh	POSTINC0 <sup>(3)</sup>	FCEh	TMR1L	FAEh	RCREG	F8Eh	—
FEDh	POSTDEC0 <sup>(3)</sup>	FCDh	T1CON	FADh	TXREG	F8Dh	LATE <sup>(2)</sup>
FECh	PREINC0 <sup>(3)</sup>	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD <sup>(2)</sup>
FEBh	PLUSW0 <sup>(3)</sup>	FCBh	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	CAh	T2CON	FAAh	—	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	—
FE7h	INDF1 <sup>(3)</sup>	FC7h	SSPSTAT	FA7h	EECON2	F87h	—
FE6h	POSTINC1 <sup>(3)</sup>	FC6h	SSPCON1	FA6h	EECON1	F86h	—
FE5h	POSTDEC1 <sup>(3)</sup>	FC5h	SSPCON2	FA5h	—	F85h	—
FE4h	PREINC1 <sup>(3)</sup>	FC4h	ADRESH	FA4h	—	F84h	PORTE <sup>(2)</sup>
FE3h	PLUSW1 <sup>(3)</sup>	FC3h	ADRESL	FA3h	—	F83h	PORTD <sup>(2)</sup>
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	—	FA0h	PIE2	F80h	PORTA

- Note 1:** Unimplemented registers are read as '0'.  
**2:** This register is not available on PIC18F2X2 devices.  
**3:** This is not a physical register.

Figura 3.16 Mapa registros de funciones especiales.

### 3.7.3 Lógica de Interrupciones

Las interrupciones constituyen quizá el mecanismo más importante para la conexión del microcontrolador con el mundo exterior, sincronizando la ejecución de programas con acontecimientos externos.

El funcionamiento de las interrupciones es similar al de las subrutinas de las cuales se diferencian principalmente en los procedimientos que las ponen en marcha. Así como las subrutinas se ejecutan cada vez que en el programa aparece una instrucción CALL, las interrupciones se ponen en marcha al aparecer en cualquier instante un evento externo al programa, es decir por un mecanismo hardware.

El PIC18f452 tiene múltiples fuentes de interrupciones, y una característica de prioridad que permite a cada interrupción asignar una alta o baja prioridad, Para la interrupción de alta prioridad es el vector 000018h. Hay 10 registros los cuales pueden ser usados para control de la operación de interrupciones:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

Entre las fuentes de interrupción más importantes que posee el PIC18F452 tenemos:

1. externas interrupciones RB0/INT0, RB1/INT1, RB2/INT2
2. Desbordamiento del temporizador TMR0

Cuando se produce cualquiera de los sucesos indicados anteriormente, se origina una petición de interrupción, que si se acepta, guarda el valor del PC actual en la Pila, pone a cero el bit GIE (*Global Interrupt Enable*), lo que prohíbe cualquier otra interrupción y se carga el PC con el valor 00008H o 000018H, que son las posiciones de los vectores de interrupción, y comienza a ejecutarse el programa de atención a la interrupción que se encuentra a partir de esta dirección. Cualquiera de las interrupciones también puede sacar al procesador del modo de reposo.

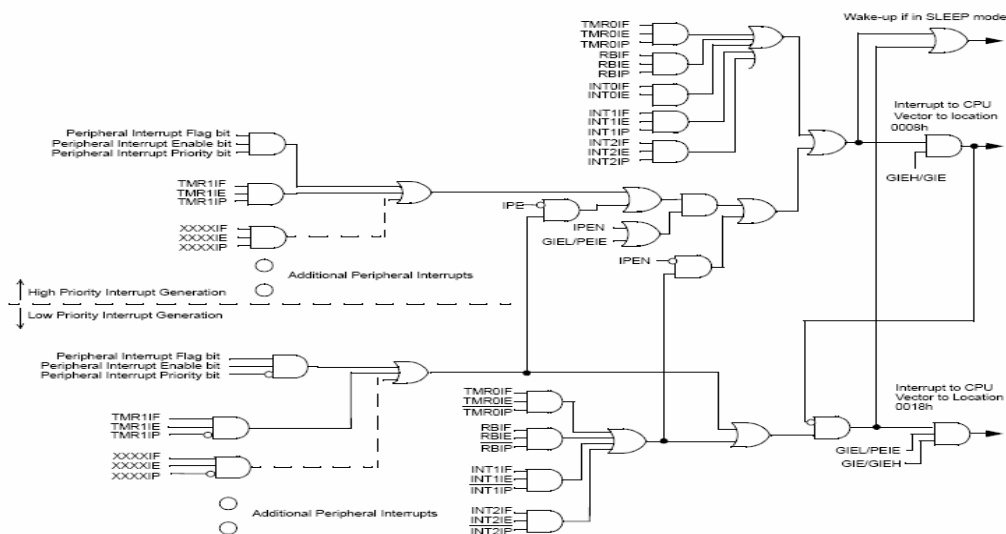


Figura 3.17 Lógica de Interrupciones

El bit GIE (*Global Interrupt Enable*) es el de activación global del permiso de interrupción, y se borra automáticamente cuando se reconoce una interrupción para evitar que se produzca ninguna otra mientras se está atendiendo a la primera. Al retornar de la interrupción con una instrucción RETFIE, el bit GIE se vuelve a activar poniéndose a 1. Para el resto de los bit de indicación de interrupción (es decir, el resto de las banderas) no se ha previsto mecanismo de puesta a cero, por lo que es el programa de atención a la interrupción el que debe realizar el tratamiento de la correspondiente interrupción y además, el que debe poner el o las banderas de indicación de interrupción a 0. De no ser así, no se podrá salir de la rutina de atención a la interrupción.

El microcontrolador dispone de dos vectores de interrupción en la dirección 000008h para interrupciones de alta prioridad y 000018h para interrupciones de baja prioridad; esto quiere decir que, sea cual sea la fuente de la interrupción, el PC se carga con 000008h o 000018h. Por lo tanto, el programa de atención a la interrupción debe encargarse de comprobar el estado de cada uno de las banderas para saber cual es el dispositivo que produce la interrupción y actuar según el caso.

Como ya hemos dicho el único registro que se salva en la PILA es PC, luego si se necesita preservar algún otro registro debe ser el propio programa de atención a la interrupción el que se encargue de salvar su estado al inicio de la rutina y de devolverlos al final del mismo, de igual modo que se hacía en las subrutinas.

Resumiendo, las acciones que se realizan automáticamente el microcontrolador y las que el programador debe tener en cuenta en sus programas son las siguientes:

1. Cuando se activa una posible causa de interrupción, la bandera correspondiente se activa. Si el bit de permiso correspondiente está a 1 y el bit de habilitación de todas las interrupciones (GIE) está a 1, se produce la interrupción.
2. Para evitar que se produzca otra interrupción mientras se está atendiendo a otra anterior, el bit GIE se pone a 0.
3. El valor del PC se guarda en la PILA
4. El PC se carga con los valores 000008h o 000018h, que son los vectores de interrupción.
5. El programador, debe comenzar la rutina de atención a la interrupción con un salto a la posición de memoria donde se encuentra el programa, seguidamente se guardan todos los registros que puedan ser modificados por esta, seguidamente si están habilitadas varias vías de interrupción, se debe explorar el valor de las banderas para determinar la causa de la interrupción.
6. Dependiendo de la causa de la interrupción, la rutina de interrupción se bifurca a la subrutina correspondiente.
7. Se deben devolver los valores que tenían los registros antes de producirse la interrupción y se deben borrar por software las banderas que indican las fuentes de las interrupciones, antes del retorno al programa principal.
8. Cuando se llega a la última instrucción de la rutina de interrupción, RETFIE, se carga el PC con el valor que se guardó inicialmente en la PILA y el bit GIE se pone automáticamente a 1.

### **3.7.4 Módulos de interés**

A continuación se muestra una descripción de los módulos a ser usados por el diseño de la interfaz.

#### **3.7.4.1 Puerto A**

El puerto A es un puerto bi-direccional de 7 bits, El registro de dirección de datos es TRISA. Este puerto es configurado para entradas y salidas digitales, con este puerto se controla la lectura y escritura de la etapa del framer (DS2143).

#### **3.7.4.2 Puerto B**

Este puerto es bi-direccional de 8 bits, El registro de dirección de datos es TRISB. En este puerto los bits 7-1 son configurados para entradas y salidas digitales, los bits 7-4 son utilizados para controlar lecturas y escrituras a la USB DLP452, los bits 3-2 se utilizan para captura de los datos presentes en el latch 74CH573 y mantenerlos para la posterior captura y el bit 0 es configurado para atender la interrupción de bloque de canal.

#### **3.7.4.3 Puerto C**

Este puerto es bi-direccional de 8 bits, El registro de dirección de datos es TRISC. Los 8 bits del puerto son configurados como entradas y salidas digitales. Los 8 bits se utilizan como salidas para envío de datos al Framer y como entradas para la captura de datos presentes en el latch 74HC573, cuando se presenta la interrupción de bloque de canal.

#### **3.7.4.4 Puerto D**

El puerto es bi-direccional de 8 bits, El registro de dirección de datos es TRISD. Este puerto es configurado como estradas y salidas digitales, se utiliza para lectura y escritura de datos al USB DLP252.

### **3.7.5 Pin de interrupción externa (RB0/INT)**

La fuente de interrupciones INT es sumamente importante para atender eventos externos en tiempo real. Cuando en la línea RB0/INT se hace una petición de interrupción, entonces, de forma automática, el bit INTxIF del registro INTCON se pone a 1 y si el bit GIE=1, se pone en marcha el mecanismo que ya hemos comentado de la interrupción. Mediante el bit INTEDG0 del registro INTCON2, se puede seleccionar el flanco activo de RBO/INT, ya que con este puesto 1 el flanco activo es el de subida y cuando está a 0 el flanco activo es el de bajada.

El programa de atención a la interrupción antes de regresar al programa principal debe borrar la bandera INTxIF, puesto que en caso contrario al ejecutar la instrucción de retorno de interrupción RETFIE se volverá a desarrollar el mismo proceso de interrupción.

### 3.7.6 Flujograma del controlador principal en el PIC18F452

La siguiente figura ilustra el proceso principal que ejecuta el microprocesador PIC18F452 para coordinar la lógica de detección en la interfaz E1 y así poder capturar cada uno de los bytes de los paquetes de señalización. El proceso arranca con la energización del integrado, y continua por medio de la escritura de una palabra de configuración que se recibe por medio de la interfaz USB, también conectada al microcontrolador..

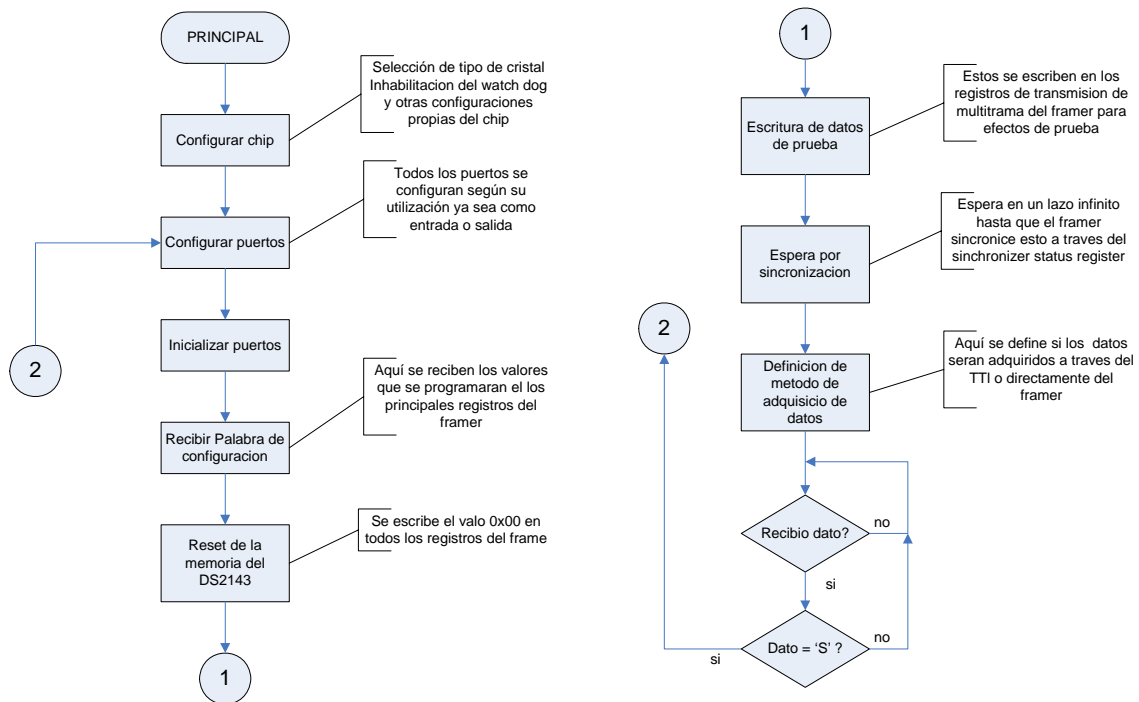


Figura 3.18 Algoritmos de programa principal.

### 3.7.7 Lógica de Interrupciones en el PIC18F452

Como se menciona en la sección 3.7.3, el microcontrolador trabaja con interrupciones las cuales se activan con el cambio de estado lógico de un pin externo del microcontrolador. Al activarse, se pone en marcha el código diseñado para atender dicha interrupción. La siguiente figura ilustra los dos procesos de atención a las dos fuentes de interrupciones: escritura de registros por medio de la interfaz USB y llegada de un bloqueo de canal.

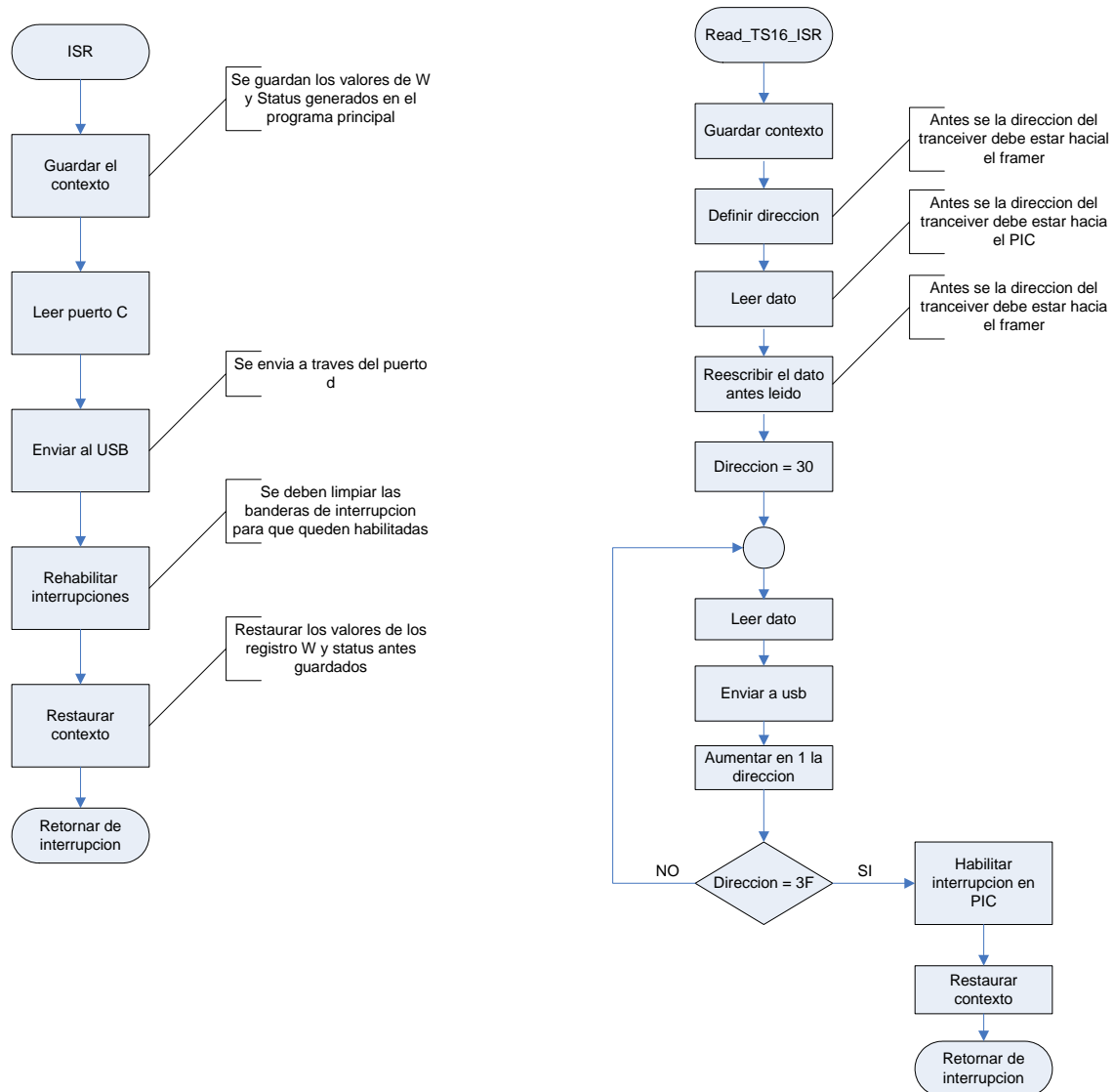


Figura 3.19 Algoritmos de rutinas de atención a las dos fuentes de interrupción.

### 3.7.8 Sub-rutinas para escritura en chip Framer DS2143

Para poder configura, escribir y leer el circuito dedicado, tipo Framer, debe ser necesario respetar el protocolo de comunicación que define el fabricante. Este puede ser de tipo *Motorola* o *Intel* (según estado lógico de un pin del integrado). La siguiente figura, describe los procesos necesarios para la implementación del protocolo de comunicación con el chip dedicado, el cual se basa en el protocolo *Intel*.

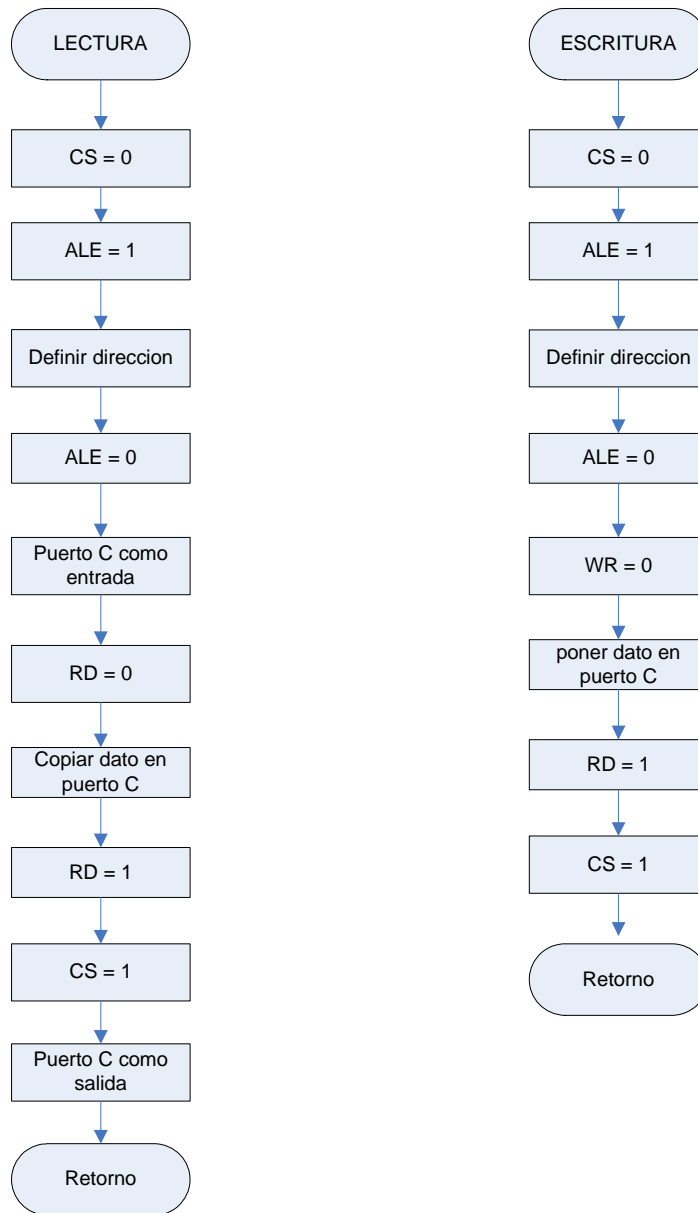


Figura 3.20 Subrutinas de escritura para el DS2143.



## 3.8 INTERFAZ USB

### 3.8.1 DLP-USB245M

En la actualidad los fabricantes de computadoras, especialmente las computadoras portátiles ya no incorporan el puerto RS-232, en su lugar utilizan los puertos USB. Cada computadora posee varios puertos USB para la conexión de diferentes periféricos, lo cual obliga a la creación de hardware orientado al puerto USB.

El puerto RS232 ofrece razones de transferencia de 230 Kbits por segundo [[www.dlpdesign.com/usb-prev/TamingUSB.pdf](http://www.dlpdesign.com/usb-prev/TamingUSB.pdf)] y el puerto paralelo ofrece velocidades superiores pero es un puerto generalmente ocupado por la impresora. Para la transferencia de datos a bajas velocidades, estos puertos sin duda representan una buena alternativa, pero para aplicaciones que requieran sobrepasar dichas velocidades existe la alternativa de usar el puerto USB, pues este brinda razones de transferencia arriba de 1 MByte por segundo.

El manejo del puerto USB es más complejo que los otros puertos, por eso es necesario que exista un intermediario entre los controladores de Windows para el puerto USB y el entorno de programación de dicho puerto. Una buena alternativa lo brinda *FTDI Chip* [[www.ftdichip.com](http://www.ftdichip.com)] quienes proporcionan un pequeño chip (FT245BM) y software para una rápida conexión a la computadora. Este chip es el componente principal de la tarjeta DLP USB245M

El DLP USB245M brinda un mecanismo sencillo para la transferencia de información sin la necesidad de utilizar un microcontrolador. Esta tarjeta posee dos modos de operación que a continuación se describen.

#### **Virtual COM Port (VCP)** [[www.dlpdesign.com/usb/dlp-usb245m12.pdf](http://www.dlpdesign.com/usb/dlp-usb245m12.pdf)]

En este modo el periférico es visto como un puerto COM estándar para la aplicación del software. En este modo los comandos para establecer el *baud rate* (la velocidad de transferencia de datos) son ignorados. El dispositivo siempre transfiere datos a su alta velocidad sin importar la configuración del *baud rate* en la aplicación. Además es importante saber que el dispositivo USB transfiere los datos en paquetes y no caracteres individuales como generalmente se hace por los otros puertos, esto no debe alarmar pues luego se verá que se puede mantener un control preciso de los datos recibidos y lograr mejores velocidades.

Para trabajar en este modo la empresa *FTDI Chip* provee los controladores totalmente gratis y existen versiones para Windows 98, Windows ME y Windows 2000/XP que pueden coexistir en el mismo directorio sin crear conflictos.

El diseño de la interfaz no usará este modo, por lo cual no presentamos un análisis más profundo, sin embargo, su uso no es complicado y no requiere más que la instalación del software y la puesta en marcha.

## D2XX Direct Drivers [[www.dlpdesign.com/usb/dlp-usb245m12.pdf](http://www.dlpdesign.com/usb/dlp-usb245m12.pdf)]

*FTDI Chip* brinda una solución alternativa al driver VCP que le permite al software de la aplicación, interactuar con el dispositivo FT245BM usando un archivo DLL en lugar del puerto virtual COM (VCP). Este paquete consiste de un driver WDM de Windows que se comunica con el dispositivo FT245BM por medio del *Windows USB Stack* y un archivo DLL que interactúa con la aplicación del software escrito en VC++, C++ Builder, Delphi, VB, Python, etc.

En resumen, este modo nos provee un medio de comunicación con el puerto USB y el archivo DLL sirve como intermediario entre nuestra aplicación y el dispositivo externo (aunque en medio hay muchos más intermediarios).

Este modo nos es mucho más atractivo pues brinda un lenguaje de programación desde el cual el puerto USB se programa, la facilidad es sorprendente y se demostrará en el código de la presente aplicación.

### 3.8.2 Configuración de Pines

Ahora es necesario profundizar en el funcionamiento del DLP USB245M, primero que nada se debe conocer su pin out [[www.dlpdesign.com/usb/dlp-usb245m12.pdf](http://www.dlpdesign.com/usb/dlp-usb245m12.pdf)] el cual consta de 24 pines que se explicarán brevemente a continuación, luego se retomarán los pines de mayor importancia para la transferencia de datos. La configuración de pines se puede observar en la siguiente figura.

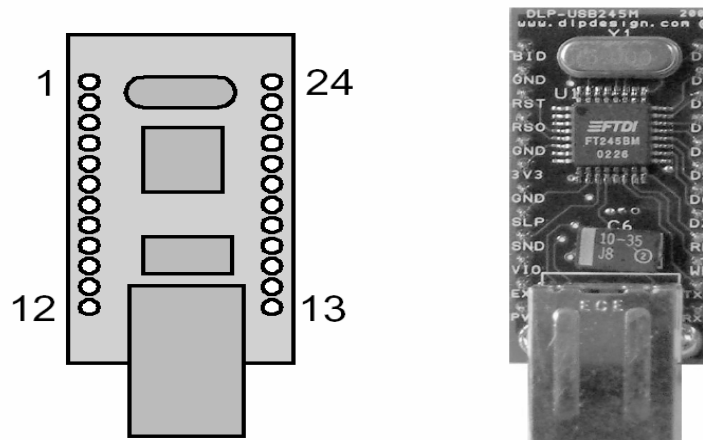


Figura 3.21 Instalación del controlador.

Tabla 3.5 Descripción de los pines del USB245M

<b>PIN</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
1	BOARD ID (Salida)	Este pin identifica la tarjeta que se está utilizando. Nivel lógico alto para el DLP USB232M Nivel lógico bajo para el DLP USB245M. No interviene en la transferencia de datos y si no se utiliza deberá dejarse sin conexión (NC).
2	GROUND	Este pin debe conectarse a la tierra digital del circuito.
3	RESET# (Entrada)	Es una entrada para reiniciar la DLP USB245M a partir de un dispositivo externo. Puede ser de ayuda cuando se requiera limpiar la memoria del USB e iniciar un nuevo proceso sin tener que apagar todo el equipo. Si este pin no se emplea, deberá conectarse a VCC.
4	RESETO# (Salida)	Es la salida del generador de reset interno. Se mantiene en alta impedancia aproximadamente 2ms después de que $VCC > 3.5 V$ y el reloj interno inicie, entonces se fija esta salida a los 3.3 V del regulador interno. Tomando RESET# bajo también se fuerza a que RESETO# baya a alta impedancia. RESETO# no es afectado por reset del bus USB.
5	GROUND	Este pin debe estar conectado a la tierra digital del circuito.
6	3V3OUT (Out)	Esencialmente sirve para alimentación interna, sin embargo ofrece la posibilidad de ser utilizarlo para polarizar circuitos externos siempre y cuando la corriente exigida de éste pin sea menor o igual a 5mA. Si no se conoce la corriente exigida por el circuito externo se recomienda evitar el uso de éste pin.
7	GROUND	Este pin debe estar conectado a la tierra digital del circuito.
8	SLEEP (Out)	Este pin indica cuando el USB245M entra en estado de suspensión. Permanece en bajo después de ser configurado vía USB y al entrar en suspensión mantiene la salida en alto. Cuando entra en suspensión también la circuitería lógica externa debe hacerlo, esto se puede controlar por medio de un transistor MOSFET canal tipo “p” utilizado como interruptor.
9	SND/WUP (In)	Si el USB245M esta en modo de suspensión, un flanco positivo en este pin inicia una secuencia de encendido. Si el dispositivo no esta en modo suspensión, un flanco positivo en este pin causa que los datos en el buffer de escritura, sean enviados a la PC en la próxima petición de entrada de datos del USB, sin importar que tantos bits estén en el buffer. Este pin puede usarse para controlar la transferencia de datos desde el hardware para aplicaciones de gran velocidad en donde la memoria del USB245 se llena rápidamente. De lo contrario se puede controlar el flujo de datos desde la PC.
10	VCC-IO (In)	Cuando existe lógica externa usando 3.3V (CMOS level) conectar este pin a dicha alimentación, sin embargo si se desea

		alimentación desde el bus se debe conectar al pin 10, 11 y 12. (ver configuración de polarización en la siguiente sección)
11	EXTVCC (In)	Usado para alimentación externa (4.4 - 5.25V), de lo contrario se conecta a PORTVCC para alimentación desde el puerto USB.
12	PORTVCC (Out)	Sirve para brindar alimentación desde el puerto USB. Conectar a EXTVCC si el modulo será alimentado por el puerto USB (configuración típica). La corriente máxima permitida por el puerto USB es de 500mA, la cual debe distribuirse en la tarjeta USB245M y la electrónica alimentada por dicho puerto. Si la corriente sobrepasa los 500mA se recomienda alimentar el circuito y tarjeta desde una fuente externa.
13	RXF# (Out)	Cuando tiene un nivel bajo por lo menos 1 byte esta disponible en el buffer de recepción FIFO (128 bytes) y esta listo para ser leído por RD#. RXF cambia a nivel alto cuando el buffer de recepción esta vacío.
14	TXE# (Out)	Cuando esta en un nivel alto el buffer de transmisión FIFO (385 bytes) esta lleno, u ocupado almacenando el ultimo byte escrito. No se debe intentar escribir datos al buffer transmisor cuando este pin esta en nivel alto.
15	WR (In)	Cuando detecta un flanco de bajada WR lee las 8 líneas de datos y escribe el byte dentro del buffer transmisor FIFO. Los datos escritos en el buffer transmisor son enviados a la PC dentro valor del <i>timeout</i> del buffer transmisor TX (por defecto 16ms) y colocado en buffer RS-232 abierto por el programa de la aplicación. El USB245M permite reprogramar el valor del <i>timeout</i> del buffer transmisor a un valor entre 1 y 255ms dependiendo de los requerimientos de la aplicación, también el pin SND puede ser usado para enviar cualquier dato restante en el buffer transmisor sin importar el <i>timeout</i> .
16	RD# (In)	Cuando cambia de nivel alto a bajo, RD# toma un byte del buffer de transmisión FIFO y lo coloca en las 8 líneas de datos. Cuando el nivel de RD# cambia a alto, las ocho líneas de datos retornan a un estado de alta impedancia y prepara el próximo byte (si esta disponible) en el FIFO a ser leído.
17	D7	Bit #7 del bus de datos I/O bi-direccional
18	D6	Bit #6 del bus de datos I/O bi-direccional
19	D5	Bit #5 del bus de datos I/O bi-direccional
20	D4	Bit #4 del bus de datos I/O bi-direccional
21	D3	Bit #3 del bus de datos I/O bi-direccional
22	D2	Bit #2 del bus de datos I/O bi-direccional
23	D1	Bit #1 del bus de datos I/O bi-direccional
24	D0	Bit #0 del bus de datos I/O bi-direccional

Tabla 3.5 Descripción de los pines del USB245M (Continuación)

Para la determinar la conexión de los pines de la tarjeta, primero hay que determinar la polarización a usar (ver *Diferentes tipo de polarización* en la siguiente sección), si se utilizara una gran cantidad de integrados en la lógica externa se recomienda el uso del modo de *polarización externa a 5 V*.

Si se desea enviar datos a la computadora se usaran los pines TXE# y WR, si se desea recibir datos desde la computadora se usara RXF# y RD#. Si se desea un flujo doble se usaran los 4 pines antes mencionados.

Una vez establecida la conexión de los pines anteriores se puede establecer un flujo de transferencia de datos (respetando los diagramas de tiempo para el ciclo de lectura y escritura).

### 3.8.3 Diferentes tipos de polarización

Una vez comprendido el pin out de la tarjeta se procede a la selección de la polarización [[www.dlpdesign.com/usb/dlp-usb245m12.pdf](http://www.dlpdesign.com/usb/dlp-usb245m12.pdf)]. Cada una posee requerimientos diferentes de uso y deben ser tomados en cuenta al momento de elegir el tipo de alimentación requerido.

Alimentación básica desde el bus USB

La figura siguiente ilustra la configuración típica de alimentación desde el puerto USB.

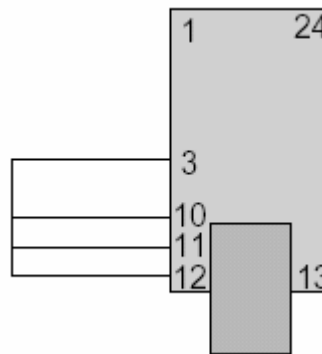


Figura 3.22 Alimentación básica desde el bus USB.

Un dispositivo alimentado desde el USB obtiene su alimentación desde el bus. Las reglas básicas para dispositivos alimentados desde el bus USB son las siguientes:

Al conectarse, el dispositivo no debe consumir más de **100 mA**.

En modo suspendido, el dispositivo no debe consumir más de **500 uA**.

Un dispositivo de alta potencia alimentado desde el bus (Uno que consuma mas de 100 mA) debe usar el pin SLEEP# para mantener la corriente debajo de 100mA al conectarse y 500 uA en modo suspendido.

Ningún dispositivo puede consumir más de 500 mA desde el bus USB.

Si el control sobre la corriente no puede manejarse sobre los límites establecidos se recomienda el uso de otro tipo de alimentación. A continuación se ilustra el tipo de polarización usada en el proyecto.

### 3.8.3.1 Alimentación externa (sistema a 5V)

Puesto que el hardware de la interfaz a desarrollar requiere del uso de una gran variedad de componentes electrónicos, la corriente consumida puede sobrepasar los límites establecidos para la *alimentación básica desde el USB* explicada anteriormente, por lo tanto se utilizará una alimentación externa. La siguiente figura ilustra la configuración típica de alimentación externa. Un dispositivo con esta configuración obtiene la alimentación desde su propia fuente de voltaje y no consume corriente desde el bus USB.

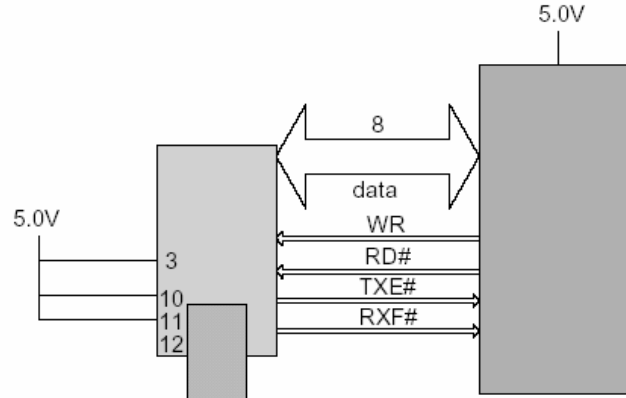


Figura 3.23 Alimentación externa (sistema a 5V).

Las reglas básicas para dispositivos alimentados externamente (5 V) son las siguientes:

- Un dispositivo con alimentación externa no debería forzar corriente sobre el bus USB cuando el controlador *USB Host* o *Hub* este apagado.
- Un dispositivo con alimentación externa puede tomar tanta corriente como necesite en operación normal y modo suspendido ya que tiene su propia alimentación.

Este sistema de polarización es más estable que el primero puesto que depende de una fuente externa y no hay que preocuparse por sobrepasar la corriente límite del bus USB. En el presente proyecto optamos por usar esta polarización.

### 3.8.3.2 Alimentación externa (sistema con interfase lógica a 3.3V).

Otro tipo de polarización es el mostrado en la siguiente figura que ilustra como configurar el DLP USB245M a una interfaz con dispositivos lógicos de 3.3V. También en esta figura se muestra la alimentación de 3.3 V hacia el pin 10 (VCCIO) desde la electrónica externa, que causara que los pines de la interfaz de IO FIFO manejen salidas a niveles de 3.3 V.

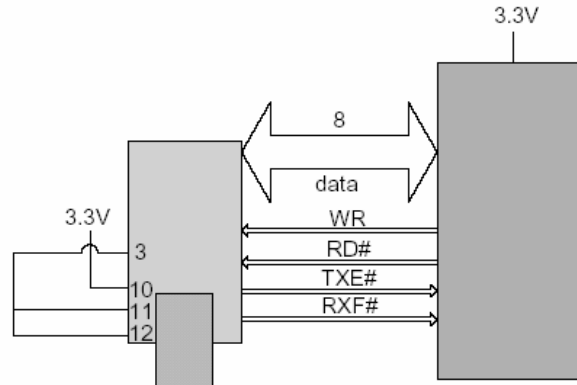


Figura 3.24 Alimentación externa (sistema con interfase lógica a 3.3V).

### 3.8.4 DIAGRAMAS DE TIEMPO

Una vez seleccionada la polarización se debe comprender el funcionamiento de los diagramas de tiempo para lectura y escritura [[www.dlpdesign.com/usb/dlp-usb245m12.pdf](http://www.dlpdesign.com/usb/dlp-usb245m12.pdf)], en nuestro caso solo ocupamos el diagrama para el ciclo de escritura. Esto nos permite el uso adecuado de las señales de control WR, RD#, TXE# y RXF# para un flujo de datos controlado.

#### 3.8.4.1 Diagrama de temporización para el ciclo de lectura

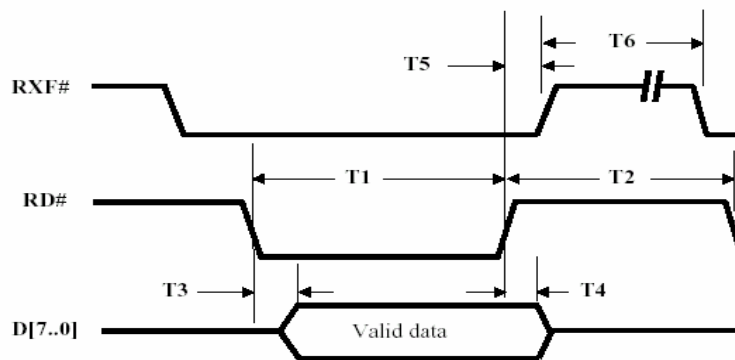


Figura 3.25 Diagrama de tiempo para el ciclo de lectura.

La figura anterior muestra un ciclo de lectura de la memoria de recepción FIFO, estos diagramas son importantes para comprender la lógica de funcionamiento de la lectura de datos, para lo cual hay que tener presente los tiempos involucrados en el ciclo de lectura.

Time	Description	Min	Max	Unit
T1	RD Active Pulse Width	50		ns
T2	RD to RD Pre-Charge Time	50		ns
T3	RD Active to Valid Data		30	ns
T4	Valid Data Hold Time from RD Inactive	10		ns
T5	RD Inactive to RXF#	5	25	ns
T6	RXF inactive after RD cycle	80		ns

Tabla 3.6 *Tiempos máximos y mínimos para el ciclo de lectura.*

### 3.8.4.2 Diagrama de temporización para el ciclo de escritura

Al igual que en la lectura de datos, el ciclo de escritura establece tiempos mínimos y máximos de las señales involucradas. Se muestra el diagrama de tiempos para la escritura de datos y los tiempos involucrados.

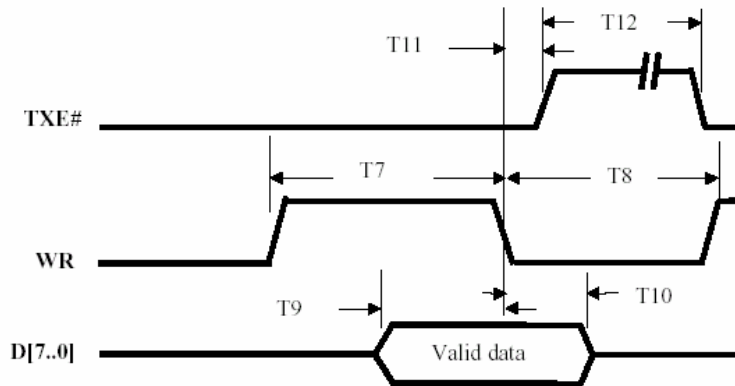


Figura 3.26 *Diagrama de tiempo para el ciclo de escritura.*

Time	Description	Min	Max	Unit
T7	WR Active Pulse Width	50		ns
T8	WR to WR Pre-Charge Time	50		ns
T9	Data Setup Time before WR inactive		20	ns
T10	Data Hold Time from WR inactive	10		ns
T11	WR Inactive to TXE#	5	25	ns
T12	TXE# inactive after RD cycle	80		Ns

Tabla 3.7 *Tiempos máximos y mínimos para el ciclo de escritura.*



### **3.9 CONCLUSIONES DEL CAPITULO III**

El circuito electrónico es capaz de funcionar como interfase entre un enlace de señalización E1 a 2.048Mbps y una computadora, Este dispositivo cumple con las normas UIT-T. Por la forma en que esta diseñado el circuito, este puede servir no solo para decodificar el protocolo ISUP sino también: BISUP, TUP, SCCP, y todo protocolo que se transporte en un canal en un enlace E1.

El circuito puede capturar mas de un canal en un mismo enlace de señalización, pero tiene la desventaja de que no pueden ser canales adyacentes, por lo que no se podría capturar un protocolo que ocupase dos canales adyacentes. La propuesta de mejora para la captura de dos canales adyacentes se presenta en el capitulo 4.

### **3.10 REFERENCIA BIBLIOGRÁFICA**

[1] Conde Jacobo, Alexander Vladimir, Rivas Preza David Alberto. “Diseño e implementación de un instrumento virtual de monitoreo de señalización CAS digital”. Universidad de El Salvador. 2004.

[2] UIT-T. “Procedimientos de alineación de trama y de verificación por redundancia cíclica (VRC) relativos a las estructuras de trama básica definidas en la recomendación G.704”. Rec. G.706. 1991.

[3] Calderón Osorio, Daniel Alberto. “Diseño de instrumentos virtuales para medir señalización siete por canal común de la Unión Internacional de Telecomunicaciones (CCS7UIT) en tiempo real”. Universidad de El Salvador. 2003.

[4] UIT-T. “Características físicas y eléctricas de los interfaces digitales jerárquicos”. Rec. G.703. 1991.

## CAPITULO IV

### IMPLEMENTACION Y FALLAS

#### 4.1 INTRODUCCION

Las pruebas realizadas en equipos de comunicación en operación en centrales de COCESNA y Digicel, tenían como objetivo garantizar la confiabilidad de la captura de los datos provenientes de la línea E1 y asegurar la capacidad de trabajo del instrumento en paralelo y transparente en el sistema donde se realizaran las mediciones.

Dichas pruebas consistieron en la conexión del circuito a al par de líneas E1 correspondientes a la transmisión y recepción del lado de un multiplexor de primer orden. Este multiplexor posee 30 canales de voz/datos de asignación fija, y es capa de manejar diferentes tipos de señalización, entre ellas R2 digital y C7.

Sin embargo, se tuvieron problemas en la captura de los datos, ya que estos no mostraban patrón alguno correspondiente a los definidos por el protocolo de señalización implementado. Dicho problemas nos condujeron a realizar diferentes acciones para determinar la causa de la falla, e implementar acciones que corrigieran estos defectos de diseño.

En el siguiente capitulo se presenta un compendio de los problemas obtenidos para la implementación del sistema, así como una descripción de las acciones realizadas en función de lograr el correcto funcionamiento del sistema y que sirvan como marco de referencia para futuras implementaciones.

#### 4.2 PROBLEMAS PRESENTES EN LA IMPLEMENTACION.

La fase inicial de diseño de la interfaz toma como base el diseño en hardware planteado por la tesis “*Diseño e implementación de un instrumento virtual de monitoreo de señalización CAS R2 digital*” de Alexander Vladimir Conde Jacobo y David Alberto Rivas Preza, con ciertas modificaciones de forma. Se implemento en laboratorio, haciéndole pruebas con un circuito TTL (mismo planteado por la tesis anterior) en configuración de contador a una frecuencia de 2.048 Mbps, obteniéndose lecturas congruentes de la interfaz USB por medio de la programación de un pequeño controlador diseñado para este propósito (programado en python 2.5), y leyendo un solo timeslot.

Se agrego a este diseño un conjunto transformador y LIU para poder conectar el circuito directamente a la línea E1 de las centrales anteriormente mencionadas, obteniéndose lecturas en la central de COSESNA. Sin embargo la implementación en dicha central de la línea E1 hacia uso de un protocolo diferente al inicialmente planteado y fue necesario

asistir a la central de Digicel para la obtención de datos con el mismo protocolo a decodificar. Esta prueba nos arrojó resultados no alentadores, ya que nuestro circuito en esta ocasión no generaba ningún tipo de captura.

El problema se descubre analizando el siguiente esquema de conexión:

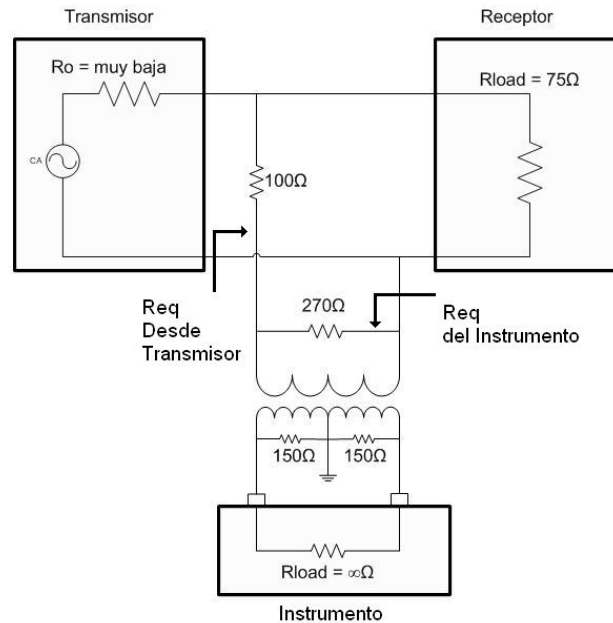


Figura 4.1 Esquema inicial de conexión.

Como se puede observar en la *figura 4.1*, la impedancia de entrada a nuestro circuito en conjunto con la resistencia de  $100\Omega$ , forman un divisor de voltaje, lo que hace reducir el mismo en una cantidad proporcional a cada impedancia. En este sentido, el nivel de señal que se obtiene en las terminales de nuestro instrumento se ve reducido y no cumple con los niveles de umbral necesarios para que los circuitos integrados distingan entre los niveles lógicos. Además el circuito planteado en el transformador no fue el adecuado ya que, bajo este esquema, nuestro instrumento representaría una impedancia de carga terminal en paralelo con la carga del receptor lo que provoca una impedancia equivalente, vista desde el transmisor, diferente del valor de impedancia de la línea y por lo tanto el circuito provocaría un efecto de carga. La resistencia de  $100\Omega$  fue introducida al arreglo por medio de un *ULink* o conector en “Y”, el cual es utilizado para la conexión de instrumentos de monitoreo y posee ese valor de resistencia.

En base al efecto de carga del esquema anterior, se propuso la siguiente modificación al esquema:

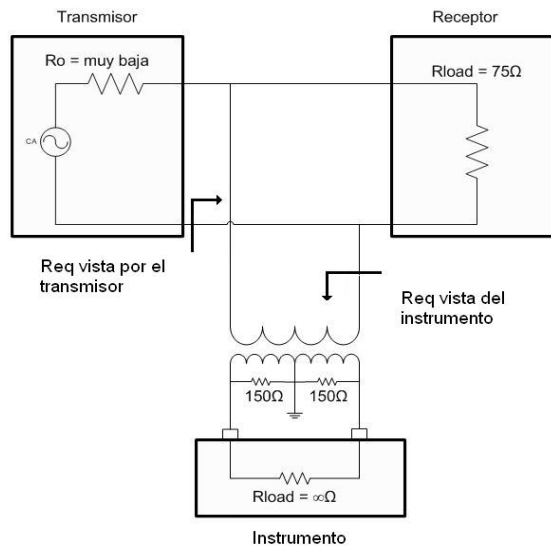


Figura 4.2 Esquema de conexión final.

En la figura anterior, se puede observar que bajo este nuevo esquema, la impedancia equivalente de nuestro instrumento es “infinita” pues no existe corriente fluyendo hacia los pines de entrada del circuito integrado LIU, por lo tanto no existirá corriente en el primario de nuestro transformador y la impedancia de entrada equivalente será infinita. Al hacer el análisis, desde el lado del transmisor se ve una impedancia equivalente a la impedancia de entrada del receptor, lo que cumple con las condiciones de impedancia de línea.

Una vez superados estos obstáculos, se comenzó a la captura de datos, con un porcentaje de fiabilidad incierto, ya que no mostraban patrones congruentes con los definidos por el protocolo.

En las siguientes secciones, se mostrarán las acciones tomadas para la correcta implementación del hardware, y se discuten las posibles causas de fallas.

## 4.3 CRITERIOS DE DISEÑO DEL CIRCUITO IMPRESO

### 4.3.1 Como evitar Interferencias

Los capacitores son elementos encargados de “almacenar” energía. Si a una línea llegara un pico o baja de tensión generada por una interferencia, estos tratarán de contrarrestar ese efecto oponiéndose al cambio: si es un pico lo absorberán y si es una baja de tensión, otorgarán a la línea parte de la energía almacenada por ellos, tratando siempre que la línea se encuentre a la misma tensión.

Los capacitores más usados para el filtrado de interferencias (transitorios) son los de cerámica, de poliéster y los multicapa, sus apariencias son las siguientes:



Figura 4.3 *Condensadores no polarizados*



Figura 4.4 *Condensadores polarizados*

Los condensadores de cerámica se usan normalmente en una capacidad de 100nF (nano-faradio) = 0.1uF (micro-faradio), los de tantalio en valores entre 1uF y 33 uF.

En sistemas digitales es necesario asegurar que la alimentación de los distintos circuitos integrados no produzca interferencia ya que podría influir al normal funcionamiento de los mismos. En el caso de los sistemas con microcontrolador el ruido en fuentes de alimentación puede lograr que el integrado falle por causa de pérdidas de datos, al inducir corrientes parasitas en las líneas de pistas de sus pines.

Los capacitores se colocan para filtrar la alimentación del IC, y este debe estar lo más próximo posible de los pines de alimentación del IC. De nada serviría colocarlo lejos de los mismos, cada IC debe poseer 1 condensador propio.

#### **4.3.2 Desacople de circuitos dentro de una misma tarjeta**

Hay ocasiones que dentro de un mismo circuito impreso coexisten varios tipos de circuitos que son alimentados por la misma fuente, por tal motivo es necesario desacoplar la alimentación para evitar que las posibles interferencias generadas por uno de ellos hagan funcionar de mala manera al otro de los circuitos.

La corriente proveniente de la fuente de alimentación debe ser filtrada por capacitores que alimentan la parte lógica del circuito, y luego debe ser filtrada nuevamente para evitar que las interferencias provocadas por otro IC retornen a la parte lógica y la interfieran.

### **4.3.3 Consideraciones para el diseño de la tableta**

#### **4.3.3.1 Masas independientes:**

Las pistas de masas en el diseño de una tableta son muy importantes, ya que la mala disposición de las mismas puede conllevar a consecuencias negativas en el funcionamiento correcto del circuito en general.

Como todo conductor, las pistas poseen una resistencia eléctrica y sobre ella se produce una caída de tensión que es proporcional a la intensidad que los recorre e inversamente proporcional a la resistencia que presentan, por lo tanto si una misma pista de la tableta es la encargada de conectar el tierra de la fuente con dos circuitos integrados, en esta pista se ha de producir una caída de tensión según sea el consumo de corriente instantáneo de los IC. En la medida que el consumo de corriente de los IC varía con respecto a su funcionamiento, esta tensión también lo hará, por lo tanto podemos decir que tendremos una señal de variante entre la tierra de la fuente y la de los IC, como ambos IC están unidos por sus tierras podemos decir que hay la misma variación para ambos, por lo tanto esta variación que puede ser producida solo por uno de los integrados puede influenciar al otro de manera no deseada, entonces cada IC o sector que posea diferenciación con respecto a otro es conveniente que posea tierras independientes hasta el último condensador de la fuente de alimentación.

#### **4.3.3.2 Chasis del equipo:**

En caso que el equipo electrónico deba funcionar en un ambiente muy ruidoso (en términos eléctricos) o que parte del circuito posea una alta sensibilidad, es conveniente que todo el conjunto (fuente, tableta, etc.) este alojado en una única caja metálica, y que la caja se encuentre conectada tanto a neutro del circuito como a la puesta a tierra. De esta manera la caja actuará de blindaje para las interferencias.

#### **4.3.3.3 Tamaño de las pistas**

En un circuito donde la sensibilidad es alta, las pistas deben ser lo más cortas y anchas posibles, de esta forma se evitara en gran medida las interferencias por inducción, las mismas consideraciones hay que tener en cuenta si las frecuencias que pasan por tal pista son elevadas.

En la medida de lo posible, el tierra debe tratar de ocupar la mayor cantidad de espacio dentro de una tableta, de esta forma se completa el llamado “plano de tierra” el cual atenúa en gran medida cualquier efecto de inducción que ocurra entre pistas colindantes.

La sección de una pista debe poder soportar adecuadamente la corriente que por ella ha de circular, de lo contrario se producirán caídas de tensión que generarán transitorios.

Las curvas de las pistas debe tratarse de que no superen un ángulo máximo de 45°, de lo contrario si la señal que por ella circula es de la suficiente frecuencia, podría producirse una auto inducción sobre la misma y actuar como una bobina desformando la señal y hasta produciendo auto-oscilaciones.

#### 4.3.3.4 Capacitor de Bulk

El capacitor de Bulk es el que se coloca en la alimentación de todo el circuito, tanto de la alimentación digital como analógica, el valor de este capacitor debe ser de 10 veces la suma de todos los capacitores colocados a cada IC dentro del circuito impreso.

### 4.4 POSIBLES CAUSAS DE FALLAS

#### 4.4.1 Alta concentración de Ruido

En los ambientes donde normalmente se realizan las pruebas de los instrumentos para analizar protocolos, presentan una alta concentración de ruido, por lo que a la hora de diseño se deben tomar en cuenta todas las consideraciones presentadas en “El criterio de diseño de los circuitos impresos”

#### 4.4.2 Desviación de frecuencia de fase (Jitter).

Jitter es la variación de la señal digital de su posición ideal en el tiempo, el jitter es fundamentalmente una expresión del ruido en la fase. En la siguiente figura se observa un reloj ideal y un reloj con jitter.

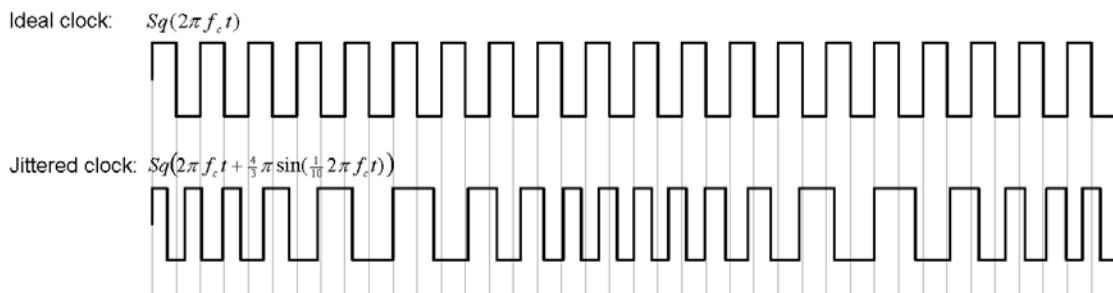


Figura 4.5 *Reloj ideal y reloj con Jitter.*

La generación de jitter ó jitter intrínseco es propio de cada componente eléctrico, El jitter se presenta en dos cantidades: Pico a pico  $J_{pp}$  y rms  $J_{rms}$ . La generación del Jitter es una combinación de jitter aleatorio y jitter determinístico.

## 4.5 MODIFICACIONES PROPUESTAS

### 4.5.1 Circuito de acople

Con base al análisis de impedancia de carga mostrado en 4.2, este mismo instrumento puede ser modificado para poder comportarse como un equipo terminal y adecuarse a la impedancia de línea.

$$Z_1 = N^2 Z_2$$

$$N = \frac{V_1}{V_2} = \frac{1}{2}$$

$$Z_1 = \left(\frac{1}{2}\right)^2 (R + R)$$

$$Z_1 = \left(\frac{1}{4}\right) 2R = \frac{R}{2}$$

Figura 4.6 *Análisis de resistencias equivalentes.*

Según la figura anterior, la impedancia equivalente vista desde fuera del instrumento es el equivalente a la mitad de cada una de las resistencias individuales que aparecen en el secundario. Por lo tanto para un sistema de  $75 \Omega$  se necesitan dos resistencias de  $150 \Omega$ , y para una de

Por lo tanto auxiliándonos de puentes o *jumpers* podemos lograr generalizar nuestro diseño para que se acople a todas las posibles impedancias de línea, por medio de configuración manual en hardware.

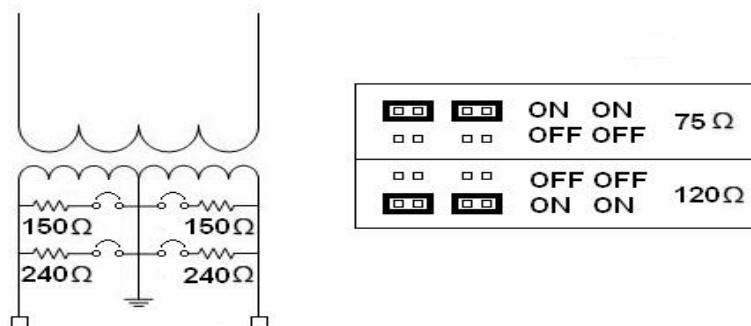


Figura 4.7 *Esquema de conexión de puentes, para modificar la impedancia equivalente de línea.*



### 4.5.2 Ruido interno generado por oscilador

El oscilador utilizado no es el adecuado para ser usado en circuitos impresos, ya que genera altos niveles de ruido en los circuitos integrados, además de que consume demasiada potencia.

El nivel de ruido generado por un oscilador se mide con un osciloscopio, colocando este en AC, a escala de 100mV y si los niveles de ruido observados sobrepasan los 100mV, indica que el oscilador no es el adecuado para circuitos impresos.

El CI DS2187 extrae reloj y datos, por lo que mejor forma de eliminar ruido proveniente del oscilador es utilizando el reloj recuperado del DS2187 usando la configuración "LOCAL LOOP BACK TIME DEL DS2143" y eliminar por completo el uso de un oscilador externo.

### 4.5.3 Lectura de dos canales adyacentes

El circuito propuesto para monitoreo, tiene la desventaja de que cuando se configura para leer dos canales adyacentes, a este le es imposible detectar el primer canal, debido a los pulsos de bloqueo de canal se unen formando un solo pulso, lo cual hace que tanto el Latch y el microcontrolador detectan solamente el ultimo canal, por lo que se propone la siguiente solución para poder adquisición de dos canales adyacentes:

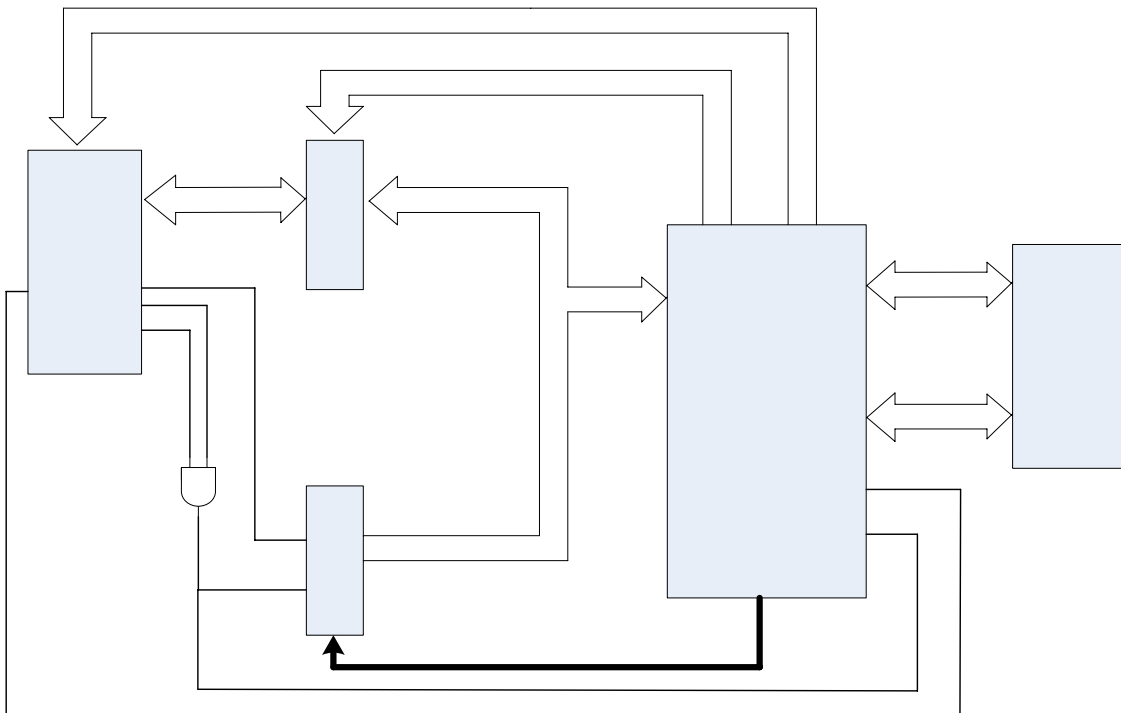


Figura 4.8 *Diagrama de la Solución.*

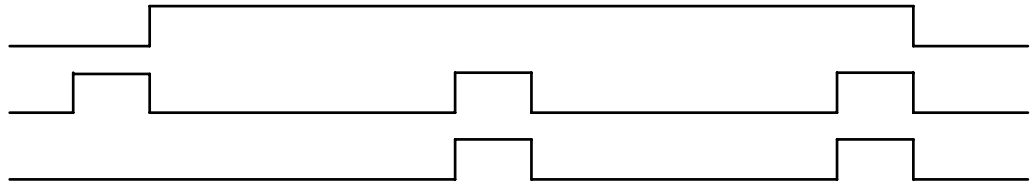


Figura 4.9 *Diagrama de Tiempo de la solución.*

#### 4.5.4 Sincronización de dos canales

RCHBLK

Para la sincronización de los dos canales de monitoreo (Tx, Rx) deberá eliminarse por lo menos medio segundo de adquisición de datos, esto para evitar disparidades de datos.

RCHCLK

#### 4.5.5 PROPUESTA DE MEJORA DEL CIRCUITO

NRCHCLK

Este es el circuito propuesto para mejora en el hardware, reduciendo los costos de hardware en la utilización de un solo microcontrolador y una sola tarjeta USB, Esto se hace utilizando la ventaja de los múltiples pines de interrupción externa que posee el microcontrolador permitiendo la captura de los datos dos canales (Tx y Rx)

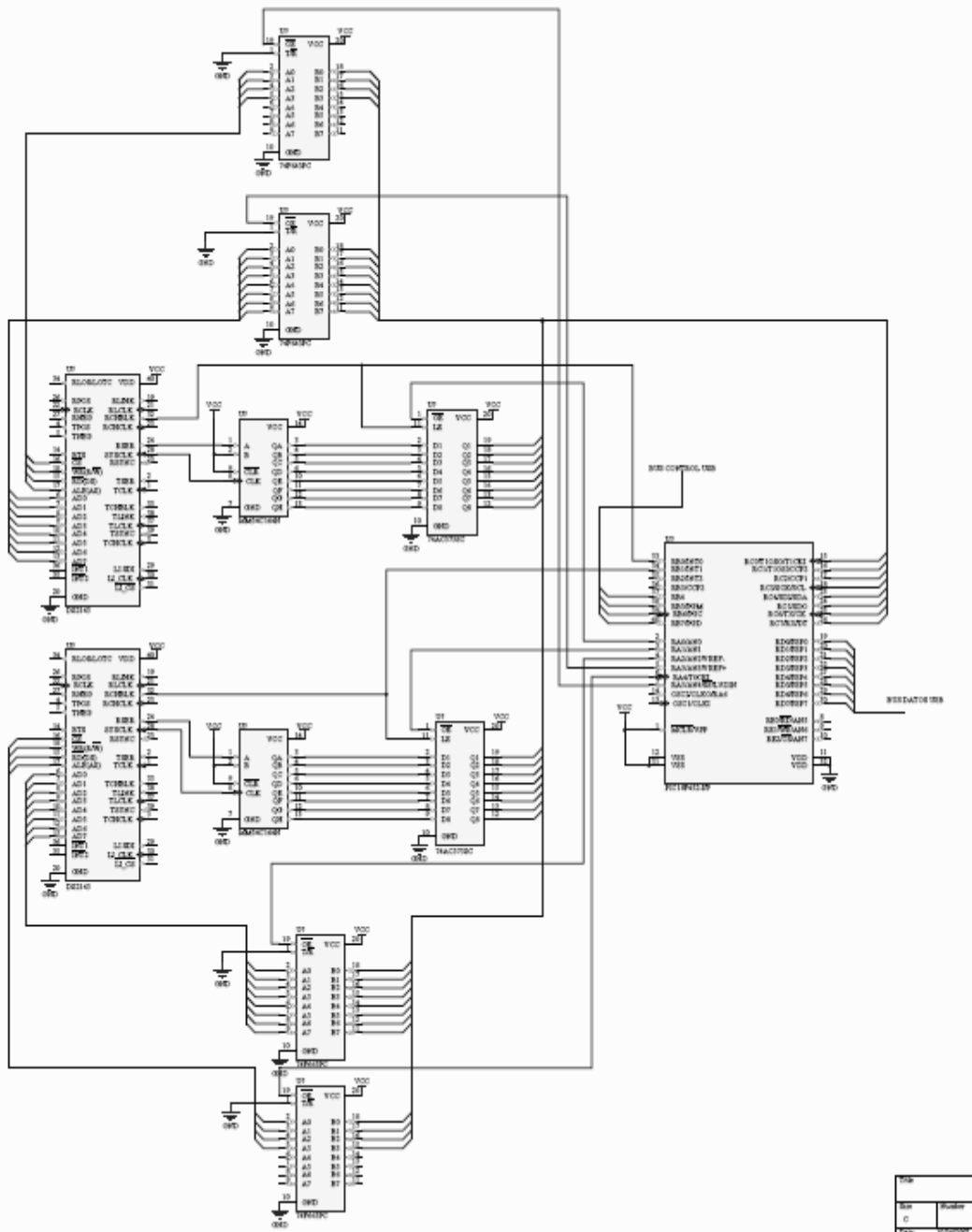


Figura 4.10 Diagrama eléctrico de la propuesta de mejora.

#### **4.6 CONCLUSIONES DEL CAPITULO IV**

Uno de los problemas que se tuvo en la captura de los datos fue el acople de impedancias, que se soluciono con el CI DS2187 que proporciona alta impedancia en la entrada de datos. En la elaboración de un circuito impreso para evitar interferencias se deben tener en cuenta las consideraciones de diseño, tales como: capacitares de desacoplo, Desacople de los CI dentro de la tarjeta, plano tierra, tamaño de las pistas y capacitor de BULK.

Los ambientes donde se desarrollaron las pruebas, presentaban alta concentración de ruido, por lo que es muy importante tomar en cuenta las consideraciones de diseño de los circuitos impresos. Otra consideración a tomar en cuenta la desviación de frecuencia de fase (JITTER) que puede perturbar los datos que se esperan obtener.

Para mejorar la robustez y versatilidad del circuito se pueden realizar modificaciones tales como: El circuito de acople para alternar entre un enlace a  $75\Omega$  y uno a  $120\Omega$ , eliminar el oscilador externo para evitar ruido en la tableta, lectura de canales adyacentes y la utilización de un solo USB.

#### **4.7 REFERENCIA BIBLIOGRÁFICA**

[1] Carlos Posada. “Como evitar interferencias” del foro todopic.  
[www.todopic.com.ar](http://www.todopic.com.ar)

[2] Bonnie C. Baker, “Técnicas de reducción de Ruido” Microchip Technology, Inc.

[3] Application Note 1432. “Jitter Analysis Techniques for High Data Rates”.  
Agilent Technologies

## CAPITULO 5

### DISEÑO DE SOFTWARE

#### 5.1 INTRODUCCIÓN

En este capítulo se presenta el desarrollo del programa de descodificación por software, como parte del instrumento virtual de monitoreo. El programa tiene como objetivo el procesamiento de los datos recolectados por la interfaz en hardware y transmitidos a través del puerto USB hacia una interfaz gráfica de usuario donde son mostrados. Dichos datos se obtienen de un enlace de comunicación entre dos centrales telefónicas a través de interfaces E1. El programa gestiona la comunicación con el circuito de interfaz y ejecuta la descodificación en tiempo real de la información de señalización que transporta el enlace E1. Dicha información es mostrada a través de la interfaz gráfica de usuario, la cual posee opciones de almacenamiento y análisis post captura.

El programa ha sido desarrollado en *Python 2.5*, que es un lenguaje de programación de alto nivel con enormes potencialidades en desarrollo de software y un excelente tiempo de aprendizaje. Para la realización de la interfaz gráfica de usuario se ha auxiliado de la librería *wxPython*, la cual es una extensión para *python* de las librerías *wxWidgets*. La principal característica de *wxWidgets* es que se enlaza con las librerías nativas del sistema, con el fin de abstraer los objetos concretos de la librería en cada plataforma a alto nivel. El código fuente del programa completo de decodificación se incluye en los anexos.

#### 5.2 EL DRIVER FTD2XX

En el capítulo 3 fue explicada la manera en que, gracias al circuito de interfaz en hardware, los datos provenientes de la línea E1 son convertidos de un tren de pulsos serie a octetos sincronizados con los timeslots de la trama E1, para luego ser transferidos hacia la PC a través de la interfaz paralela / USB.

USB es un protocolo bastante complejo. Afortunadamente la interfaz DLP-USB245M implementa este protocolo a través del chip controlador FT245BM y el driver FTD2XX. A partir de este momento el interés se dirige hacia la estructura y el manejo de las librerías del controlador FTD2XX.

### 5.3 INTERFASES DE PROGRAMACION DEL CONTROLADOR FTD2XX

Como ya se ha mencionado, el controlador proporciona una librería de funciones necesarias para la programación y adquisición de datos. Estas funciones están clasificadas en cuatro grupos de acuerdo a sus características, para tener mayor claridad.

- *Funciones de la interfaz clásica:* Este grupo clasifica aquellas funciones que ejercen la parte básica del funcionamiento del controlador, como la identificación del mismo, lectura y escritura simple, revisión de estatus, inicialización, reinicialización, etc.
- *Funciones de la interfaz de programación de la EEPROM:* El módulo cuenta con una memoria EEPROM, la cual contiene parámetros y características del chip controlador. Este grupo de funciones permiten leer y modificar estos parámetros, tales como la viñeta del controlador, tipo de transferencia, alimentación, etc.
- *Funciones extendidas de la API:* Son un complemento a las funciones básicas, con ellas es posible modificar otros parámetros que pertenecen al protocolo USB en sí, como son el tamaño de transferencia, latency timer, etc.
- *Funciones de la FT-Win32 API:* Es una alternativa un poco más sofisticada a las funciones de la interfaz clásica, basada en las llamadas a la Win32 API, pero básicamente posee la misma utilidad. Es de hacer notar que no deben mezclarse ambos grupos de funciones dentro de un mismo programa.

### 5.4 FLUJO DE DATOS A TRAVES DEL PUERTO USB

La interfaz recibe un flujo de datos constante, sobre el que se tiene poco control. Surge entonces la necesidad de realizar todos los ajustes necesarios para conseguir una transferencia de forma eficiente, y de esta manera evitar pérdidas de datos. Para ello es necesario primero entender la manera en que los datos son transferidos desde el dispositivo USB hasta la aplicación.

En el protocolo USB, el intercambio de los datos se da entre dos IC controladores, de los cuales hay uno que maneja todo el proceso, al que se denomina *Host*, y que se encuentra normalmente del lado de la PC.

Cuando se desea enviar datos en una u otra dirección, ambos controladores intercambian primero información de control de la transferencia (handshaking) y luego envían los datos de usuario. A este proceso se le denomina transacción. Estas transacciones pueden ser de 2 tipos: de entrada y salida, desde el punto de vista del *host*. La información de control y de usuario se envía organizada en forma de tramas, pues USB es un protocolo serial. A la

cantidad máxima de información de usuario que puede intercambiarse en cada transacción se le denomina *Maximum Transfer Size*.

Existen distintos modos de transferencia de información, que poseen distinto formato de tramas, manejo del ancho de banda, información de control, control de errores y tiempo de entrega de la información. Entre estos modos pueden mencionarse el *Bulk*, *Isócrono*, *Interrupt* y *Control*.

El controlador FT245BM puede trabajar en modo *Bulk* y en modo *Isócrono*. Las librerías del controlador D2XX soportan, en su versión actual, únicamente el modo *Bulk*.

En este modo, los datos de usuario están organizados en paquetes de 64 bytes, pudiendo soportar un máximo de 64 kbytes por transacción. Si la información enviada no es un múltiplo de 64 bytes, y por consiguiente alguno de estos paquetes no posee 64 bytes, la transacción se toma como terminada, y para enviar más datos, debe iniciarse otra transacción. Si se utiliza en FT245BM, la carga efectiva de usuario por paquete es de 62 bytes, ya que este utiliza 2 bytes para enviar cierta información de control.

Hablando específicamente de la interfaz paralelo / USB, cuando se manda un flanco de bajada a través del pin WR, se escribe el byte presente en las entradas en un buffer FIFO interno, que posee un tamaño de 384 bytes.

El envío de los datos presentes en el buffer interno puede propiciarse de dos formas:

- La expiración de un temporizador del controlador llamado latency timer. Su valor puede ser ajustado entre 1 y 255 milisegundos, con una resolución de 1 ms, siendo por defecto de 16 milisegundos.
- Un flanco de subida en la entrada SND del controlador del USB.

Una vez enviados, los datos son recibidos en otro buffer dentro de la PC, llamado buffer de transferencia, que posee un tamaño igual al valor del *Maximum Transfer Size*. La aplicación no posee acceso a este buffer. Una vez se completa la transacción, ya sea por que se llenó el buffer o porque se ha recibido un paquete menor a 64 bytes, los datos se transfieren otro buffer, conocido como buffer de lectura. Sólo cuando los datos llegan a este buffer están disponibles para la aplicación, que puede leerlos en la cantidad que estime conveniente.

Hay que aclarar que el número de datos disponibles para la aplicación es menor al número de datos transferidos, ya que, como se había mencionado, solo 62 bytes son datos efectivos en cada paquete.

## 5.5 LENGUAJE DE PROGRAMACION "PYTHON"

*Python* es un lenguaje de programación dinámico el cual dispone de potentes estructuras de datos de alto nivel y una solución de programación orientada a objetos simple pero eficaz, la cual permite que pueda ser usado para muchos propósitos en el desarrollo de software.

Ofrece mucho soporte para la integración con otros lenguajes y herramientas, y tiene incluido un extenso conjunto de librerías estándar. *Python* es un lenguaje de programación fácil de aprender y potente.

La elegante sintaxis de *Python*, su gestión de tipos dinámica y su naturaleza interpretada hacen de él el lenguaje ideal para guiones (de su vocablo en inglés: *scripts*) y desarrollo rápido de aplicaciones en muchas áreas y en la mayoría de las plataformas.

*Python* corre en Windows, Linux/Unix, Mac OSX, y muchos otros sistemas operativos. También puede ser portado a las máquinas virtuales de Java y .NET. Es distribuido bajo la licencia *open source license* la cual lo hace libre de uso, aun para desarrollo de software comercial.

### 5.5.1 PROGRAMACIÓN ORIENTADA A OBJETOS

Ya que la utilización de la programación orientada a objetos es uno de los pilares más importantes en el desarrollo del software de nuestra aplicación, vale la pena realizar una introducción a la implementación de tal herramienta para este lenguaje de programación.

Una de las más poderosas herramientas que posee un lenguaje de programación es la posibilidad de implementar la abstracción, siendo las clases y los objetos los encargados de realizar dicha tarea.

El soporte que posee *Python* para la programación orientada a objetos es muy potente y fácil de usar. Se pueden crear clases y usarlas para crear instancias de objetos los cuales puede tener cualquier número de atributos y métodos. *Python* también ofrece la herencia.

Las definiciones de clases, como las definiciones de funciones deben ejecutarse para tener efecto. Las sentencias de dentro de una definición de clase serán definiciones de funciones, sin embargo se permite otro tipo de sentencias, lo que resulta útil en algunos casos. Las definiciones de funciones interiores a la clase suelen tener una lista de argumentos un poco especial, dictada por las convenciones de llamada a método.

Cuando se entra en una definición de clase, se genera un nuevo espacio nominal, que se utiliza como ámbito local; así que todas las asignaciones a variables locales caen dentro de este nuevo espacio nominal. En particular, las definiciones de funciones enlazan aquí el nombre de la nueva función.



Cuando se abandona una definición de clase de manera normal (se ejecuta la última línea de su código), se crea un *objeto de clase*. Es, sencillamente, un envoltorio del contenido del espacio nominal creado por la definición de la clase. El ámbito local original (el que estaba activo cuando se entró en la definición de clase) se reinstancia y el objeto clase se enlaza con el nombre de clase dado en la cabecera de la función.

Las siguientes líneas de código escritas en *Python*, ejemplifican la definición de una clase para este lenguaje por medio de la definición de la clase *DataColector*, la cual se encarga de recolectar los datos del buffer de lectura del *host*:

```
class DataColector(threading.Thread):
    def __init__(self, shared,USBHandler):
        threading.Thread.__init__(self)
        self.h=USBHandler
        self.sharedData = shared
        self.bandera = True
        self.lista=[]
        self.buffer=[]
        self.inicio = True

    def stop(self):
        self.bandera = False

    def run(self):
        while self.bandera:
```

Figura 5.1 Implementación de la clase *DataColector*..

Existe en *Python* una nomenclatura peculiar para este lenguaje respecto a los lenguajes tradicionales (C++ y Java). Esta consiste en el uso de una indentación para separar bloques estructurales del lenguaje, ayudando así a la comprensión en la lectura del código. Sin embargo esto no es opcional, como en otros lenguajes, y constituye un requisito para la ejecución de código.

La línea *def \_\_init\_\_(self, shared, USBHandler):* hace referencia a la definición de la función o método constructor del objeto, caracterizado por el nombre *\_\_init\_\_*, el cual se ejecuta cuando el objeto es instanciado. Toma como argumentos de inicialización: el mismo objeto, un objeto *shared* que será un buffer intermedio para transferencia de paquetes, y un *USBHandler* que representa el objeto de lectura a la USB.

## 5.5.2 PROGRAMACIÓN MULTI-THREADING EN PYTHON

El multihilado es una técnica para independizar tareas que no son secuencialmente dependientes. Se pueden utilizar los hilos para mejorar la respuesta de las aplicaciones que aceptan entrada del usuario, tales como las que implementan interfaces gráficas,

mientras otras tareas se ejecutan en segundo plano. Un caso parecido es la ejecución de E/S en paralelo con los cálculos de otro hilo.

Los hilos son usados en muchas situaciones donde el programa necesita realizar muchas tareas que no son necesariamente interdependientes. La programación de interfaces gráficas de usuario, por ejemplo, ofrecen el uso de dos hilos: uno para manejar las tareas de la interfaz tales como: re-dibujar la ventana, cambiar los botones, etc. y otros para la ejecución de tareas específicas tales como: conectarse a una base de datos, ejecutar la acción sobre los datos, etc.

*Python* ofrece dos librerías para la realización de hilos: *thread* y *threading*, de las cuales se hablara únicamente de *threading* por ser la utilizada en el desarrollo del software.

El modulo *threading* define una clase de nombre *thread* la cual sirve para el manejo de hilos. Para crear un nuevo hilo, se crea un nuevo objeto de esta clase y se llama a su método *Start()*.

También se puede crear una clase hija que herede de la clase *thread*. De esta forma se debe de sobre escribir el método *run()* perteneciente a su clase base, en donde se escribirá el código que efectuará este hilo. En ese contexto, dicho código debe de contener un lazo para la ejecución de tareas repetitivas. El hilo terminara de ejecutarse cuando se haya ejecutado la ultima línea de código del método *run()* sobre-escrito.

Ya que esta nueva clase, se esta heredando de la clase base *thread*, debe tomarse en cuenta que en el método constructor *\_\_init\_\_* de la nueva clase se debe de incluir como primer instrucción una llamada al método constructor *\_\_init\_\_* de la clase base. Esto para efectos de pasar argumentos al constructor de la clase base:

```
class nuevoHilo(threading.Thread):  
  
    def __init__(self):  
        threading.Thread.__init__(self)  
        #posibles banderas para detencion de hilo  
    def run(self):  
        #Codigo del hilo  
    def stop(self):  
        #Codigo para detener hilo
```

Figura 5.2 Implementación de un objeto tipo *threading*, derivado de su clase base.

Para la ejecución del hilo anterior simplemente se realiza una instanciación de la clase anterior y se llama al método del objeto instanciado:

```
Obj = nuevoHilo()  
Obj.start()
```

Figura 5.3 Creación de un objeto hilo.

### 5.5.2.1 SINCRONIZMO DE THREADINGS

El mayor reto de las aplicaciones multi-hilo es coordinar los hilos que comparten datos u otros recursos. Para este fin, el módulo de multi-hilo proporciona varias primitivas de sincronización que incluyen bloqueos, eventos, variables de condición y semáforos.

#### ***Bloqueo:***

La primitiva usada para nuestra aplicación. El método consiste en crear un objeto de tipo *Lock* el cual es visto como un “*token*”. El hilo que posea en determinado momento este *token* será el único hilo disponible para actuar sobre los datos, mientras que los demás tendrán que esperar su turno para poseer el *token*. Este acceso se realiza secuencialmente, es decir, el primer hilo que pida acceso al objeto *Lock* estará al principio de la cola de espera sobre los demás.

Para la petición y para ceder el objeto *Lock*, los hilos tendrán que llamar a los métodos *acquire()* y *release()*, respectivamente del mismo objeto *Lock*.

Aunque estas herramientas son potentes, los errores ligeros de diseño pueden causar problemas difíciles de reproducir.

### 5.5.3 LIBRERÍA DE OBJETOS GRÁFICOS: WXPYTHON

*wxPython* es una de las librerías mas fáciles y mas poderosas para desarrollar potentes interfaces graficas de usuario. Muchas herramientas de desarrollo de interfaces graficas de usuario que permiten la generación automática de código (como por ejemplo *Visual Basic*) no poseen la implementación de un lenguaje con la claridad, flexibilidad, y potencialidad de la que *Python* posee. Aun mas, *wxPython* es un *open source project*, con el código fuente y el archivo de instalación distribuido bajo una licencia que permite ser usado en proyectos de distribución libre y en proyectos de desarrollo de software comercial.

A continuación se desarrolla un pequeño ejemplo, el cual sirve para ejemplificar el uso de la librería, así como para mostrar la funcionalidad de la misma:

```

import wx
class App(wx.App):
    def OnInit(self):
        frame = wx.Frame(parent=None, title='Bare')
        frame.Show()
        return True
app = App()
app.MainLoop()

```

Figura 5.4 Código ejemplificando uso de la librería para crear objetos gráficos.

El programa anterior no hace mayor cosa que crear y desplegar una ventana vacía (sin embargo lo hace en pocas líneas de código comparado a otras librerías). Ilustra los cinco pasos básicos que se deben completar para el desarrollo de una interfaz usando *wxPython*:

1. Importar al guión el paquete de desarrollo de *wxPython*: `import wx`.
2. Heredar nuestra clase base de la clase de aplicación de *wxPython*.
3. Definir el método de inicialización de la clase base, en donde se definirá nuestra interfaz.
4. Crear la instancia de nuestra clase base.
5. Entrar en el ciclo principal de nuestro objeto base.

El resultado de la ejecución de este ejemplo se muestra en la siguiente figura:

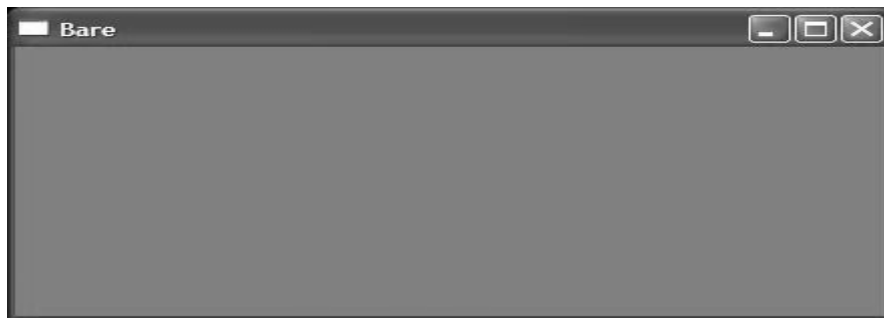


Figura 5.5 Creación de un objeto tipo ventana.

La creación de botones, etiquetas de texto, menús, conlleva una estructura similar a la que el objeto *frame* representa en el ejemplo anterior. Es decir, todos los elementos gráficos que están presentes en una interfaz gráfica de usuario, son objetos de tipo ventana o *widgets* que se crean o instancian de la clase que los define, por medio del método constructor de dicha clase. Dicho método constructor posee parámetros obligatorios que son pasados a la hora de crear el nuevo objeto, como por ejemplo el parámetro *parent* el cual le indica al método constructor, cual será el objeto contenedor de este. Muchas librerías de interfaces gráficas, utilizan un método llamado “*pack*” para empacar o indicar cual objeto contenedor era el encargado de contener al nuevo objeto creado. Dicho método era llamado por el objeto hijo, indicándole cual era el nombre de su padre como

argumento. Sin embargo *wxPython* hace obligatorio el uso de este parámetro a la hora de la instanciación de la clase hija.

Un ejemplo de la instanciación o creación de un nuevo objeto tipo *widget* es el que se muestra a continuación. Creamos la instanciación a la clase *wx.Button* para la creación de un botón y lo incluiremos en el ejemplo anterior, como la siguiente línea:

```
helloButton = wx.Button(frame,-1,"Hola")
```

La línea anterior debe ser colocada antes de llamar al método *Show()* del objeto *frame*, y el resultado de dicho cambio se refleja en la siguiente figura:



Figura 5.6 Creación de un objeto tipo ventana y su botón.

El marco de referencia anterior, pretende ser una breve introducción a la programación de la librería *wxPython* para efectos de comprensión del código de la interfaz. El desarrollo de programación de la librería queda fuera del alcance de este documento.

#### 5.5.4 Patrón MVP

Un programa secuencial es un conjunto de instrucciones ejecutadas una a la vez que permite controlar la secuencia lógica o rumbo de control en nuestras aplicaciones de software. Cuando se programa bajo esta lógica, las definiciones de funciones y objetos deberán aparecer antes de la implementación de las mismas, dificultando bajo este contexto el uso de librerías para la creación de objetos gráficos. La secuencia lógica de nuestro algoritmo principal, requerirá, en determinado momento, el uso de un botón y otro elemento grafico que no ha sido declarado aun y, en el otro sentido, la lógica de los elementos gráficos requerirán una función especial (a ejecutarse cuando se presione un botón) que no se ha definido todavía. El uso de librerías graficas en conjunto con la lógica principal de nuestra aplicación requiere de un mecanismo especial el cual permita conjugar las definiciones de nuestro programa con las de los objetos gráficos.

Un *framework* es una infraestructura en software organizada con el fin de ayudar al diseño e implementación de programas, en este caso interfaces gráficas. Esta infraestructura normalmente consiste en librerías de programación y un conjunto de

patrones de diseño: convenios o modelos de desarrollo que dan una estructura unificada a las interacciones gráficas en el entorno.

El modelo MVP (*Model-View-Presenter*) consiste en el diseño de software bajo el esquema de tres clases mutuamente cooperantes: el modelo, la vista y el presentador.

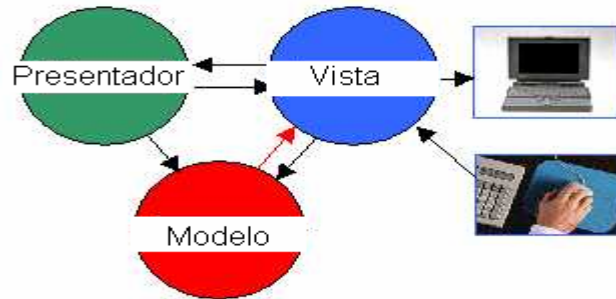


Figura 5.7 Esquema de framework usado para la implementación del software.

### **El modelo**

Esta clase es un contenedor de los datos de la aplicación y proporciona métodos para acceder consistentemente a ellos. Esta clase debe ser completamente independiente de la interfaz de usuario, para permitir ser re-utilizada en nuevas escenas de programación que contengan diferentes tipos de interfaz gráfica de usuario, por lo tanto el modelo no debe tener conocimiento directo de la existencia de la clase vista y presentador.

### **La vista**

Es una venta que se encarga de desplegar los datos del usuario, típicamente es un conjunto de controles tipo ventana. En la figura 5.7 la flecha roja indica un enlace indirecto entre el modelo y la vista tal que la vista es realmente un observador dentro del modelo. El modelo no mantiene una referencia directa a la vista, sin embargo para informar al observador cuando un dato ha cambiado, se genera una interrupción de notificación de evento. La vista ha tenido que configurar previamente estos eventos de notificación, y cuando se disparan es la vista la que se encarga de manejar el método adecuado que desplegara en ella los datos del modelo.

### **El presentador**

Mientras que la vista es la responsable de desplegar los datos de usuario, la responsabilidad del presentador es manejar las entradas de usuario y usarlas para manipular los datos dentro del modelo. La gestión del usuario (selecciones en listas, movimientos del ratón, funciones en teclado) son recibidas primeramente por la vista y luego pasadas al presentador para su interpretación. El presentador debe tener conocimiento de la definición de la lista y el modelo, así como el acceso a estas clases.

## 5.6 Estructura del Programa de Descodificación

A continuación se procederá a explicar, de forma esquemática, la implementación de la programación del instrumento. En ella, se implementa el *framework MVP*, como núcleo principal en el desarrollo de software.

### 5.6.1 Flujogramas del software

El esquema de programación completo, puede ser explicado en varios segmentos para ayudar a su comprensión

- Descripción general del software
- Clase colectora de datos
- Descodificación Capa 2
- Codificación ISUP
- Captura en modo Traceo de Llamadas
- Interfaz Grafica de Usuario

#### 5.6.1.1 Descripción general del software

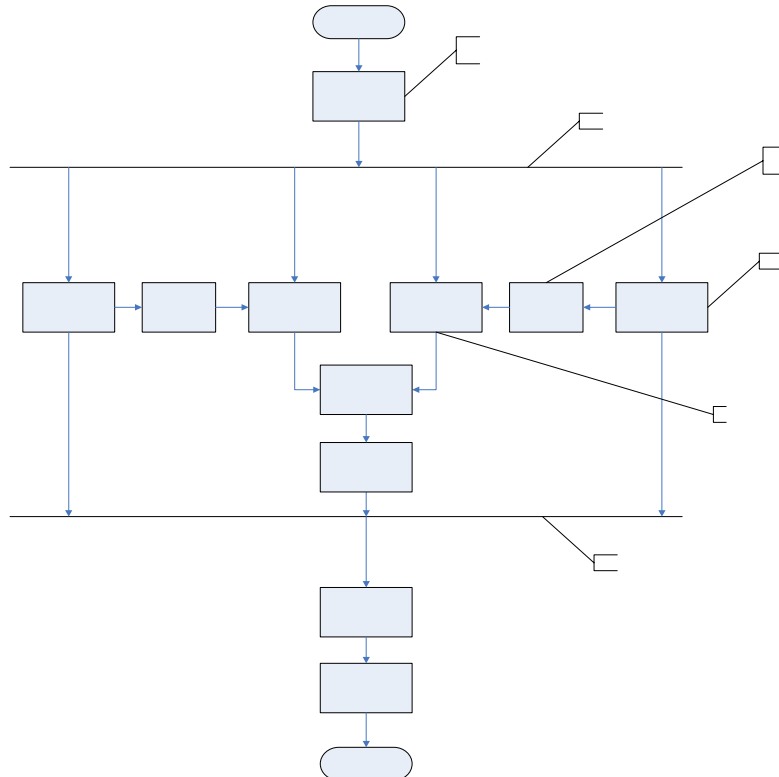


Figura 5.8 Esquema general de funcionamiento en la etapa de adquisición.

El software en general trabaja con cuatro hilos que se ejecutan paralelamente en pares, dos hilos para el lado de transmisión y dos para el lado de recepción. Para cada par de hilos, uno de los hilos se encarga de recolectar los datos serializados del USB determinando el límite de cada mensaje, para luego almacenarlos mensaje por mensaje en un búfer compartido, el otro hilo se encarga de realizar la lectura de cada mensaje en el búfer compartido y realizar la decodificación para luego almacenar en un archivo los mensajes con su respectivo número de identificación: ID. Los mensajes son mostrados en pantalla a través de la interfaz grafica de usuario en tiempo real por el mismo hilo que se encarga de decodificar cada mensaje. Una vez los hilos son detenidos por el usuario (activando un botón o función de la interfaz grafica de usuario), se activa la función de decodificación de la capa 4 (ISUP) para presentar esta información en pantalla.

### 5.6.1.2 Descripción de la clase colectora de datos:

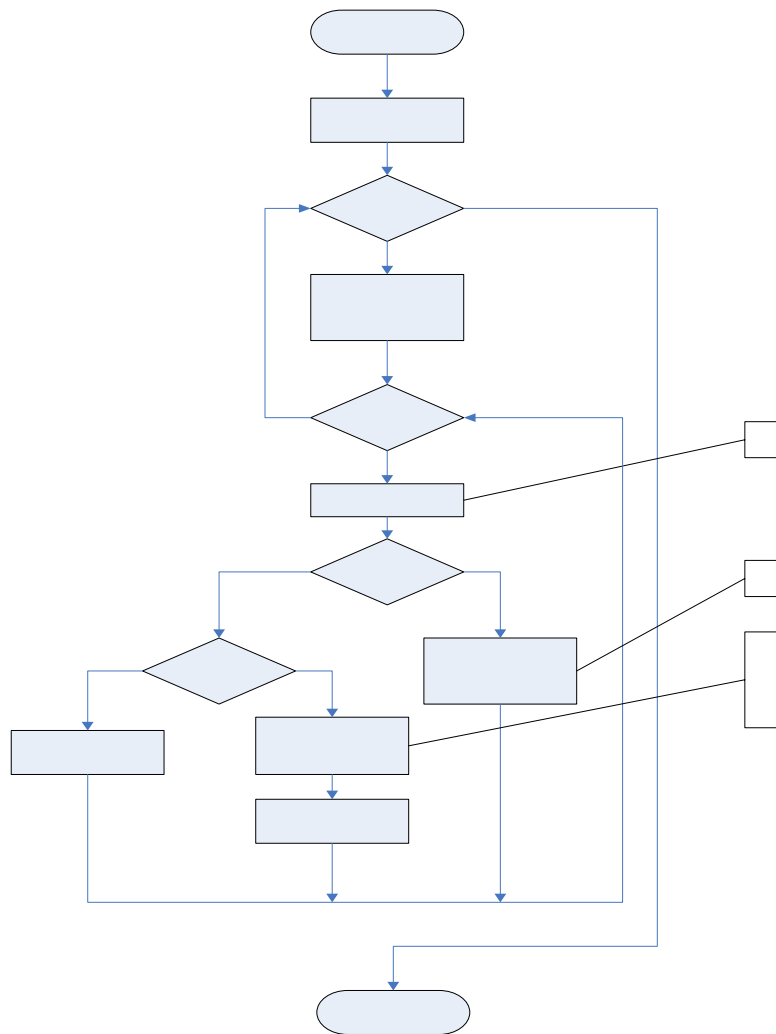


Figura 5.9 Lógica de la recolección de datos y determinación de mensajes.

Data colector

Inicializar atributo de clase

Detener ?



Se realiza una captura de 1000 datos del buffer de lectura de la USB de la PC y se almacenan en Lista, donde la lógica de programación realiza la comprobación siguiente: si la longitud de Lista es mayor que cero, se procede a extraer los bytes uno a uno de Lista. Si la longitud de Lista es cero significa que se ha agotado los bytes en Lista y por lo tanto será necesario efectuar una nueva captura de datos del buffer de lectura del *host*. Cada byte extraído de Lista es comparado con el valor 7E para determinar el límite de cada mensaje. Una vez determinado el límite de mensaje, por medio de esta comparación, los bytes que no son iguales a 7E son concatenados y almacenados en un buffer compartido con el otro hilo que realiza la decodificación de la capa 2, para posteriormente presentar en pantalla los mensajes decodificados. El hilo realiza el proceso anterior en forma repetitiva hasta que el usuario detiene el proceso por medio de un comando a través de la interfaz gráfica de usuario, deteniendo y finalizando la clase recolectora de datos.

<b>Mapa de memoria:</b>		
<b>Nombre de la clase</b>	<b>Atributos</b>	<b>Descripción</b>
Datacolector	USBHandler	Objeto referencia del buffer de la USB en la PC
	Lista	Contiene la lectura de datos leídos del buffer de la USB de la PC, correspondientes a una iteración del ciclo de lectura del hilo.
	Dato	Atributo en el cual se almacenan un byte de la lectura del atributo Lista
	Buffer	Lista en la que se concatenan los byte de un mismo mensaje
	Mensaje	Buffer que contiene los mensaje.

Tabla 5.1 *Atributos de la Lógica de la recolección de los datos*

### **5.6.1.3 Descodificación Capa Dos.**

En la decodificación de la capa 2, primero se verifica si el usuario ha elegido detener por medio del software, si no se ha elegido detener, se extrae un mensaje del buffer compartido y se convierte en una lista o arreglo. La lista debe tener una longitud mayor a 5 bytes para que sea un mensaje válido, si la longitud del mensaje es menor a 5 bytes, se desprecia el mensaje y se extrae el siguiente mensaje. Si el mensaje es válido se extrae el primer byte y se filtra el BSN y BIB. Seguido se extrae el segundo byte y se filtra el FSN y FIB, después se extrae el tercer byte y se filtra en LI (Indicador de longitud del mensaje). Si el LI=0 significa que el mensaje es FISU por lo que se guardan los datos extraídos, se procede a la sincronización con el otro canal y se muestran en pantalla los resultados. Si el LI es 1 ó 2, significa que el mensaje es FSSU por lo que procede a

guardar en un vector, los datos extraídos y el resto del mensaje, se realiza la sincronización con el otro canal y se muestran en pantalla los resultados. Si LI es mayor que 2 significa que el mensaje es MSU, se extrae otro byte y se verifica que el usuario sea ISUP, si el usuario no es ISUP se procede a la sincronización con el otro canal y se muestran los resultados en pantalla, pero si el usuario es ISUP, se extraen los bytes necesarios para obtener el DPC , OPC, SLS, y CIC, todos los datos obtenidos se almacenan en un vector, y posteriormente se muestra esta decodificación en pantalla por medio de la interfaz grafica de usuario.

Datalevel2decoder	Temp	Variable de almacenamiento de la longitud de los mensajes.
	msj_d_tx	Lista de almacenamiento del mensaje en el lado de transmisión.
	byte_d	Variable de almacenamiento del byte extraído del mensaje, para la posterior decodificación.
	LI	Variable de almacenamiento de la longitud del mensaje.
	Vector	Lista de almacenamiento del mensaje decodificado.
	SI	Variable de identificación del tipo de usuario (0101 usuario ISUP).
	DPC	Variable de almacenamiento del punto de código destino.
	OPC	Variable de almacenamiento del punto de código origen.
	SLS	Variable de almacenamiento de selección de enlace
	CIC	Variable de almacenamiento de identificador de circuito.
	tipo	Variable de almacenamiento del tipo de mensaje
	ISUPRAW	Contenido del protocolo ISUP
	Date	Variable de tiempo local
	Item	Variable de almacenamiento de bytes de un mensaje.
	Cover	Variable de almacenamiento de los mensajes pasados a string
	Coder	Variable de almacenamiento de los mensajes convertidos a ASCII.

Tabla 5.2 Atributos de la Lógica de decodificación de nivel dos.

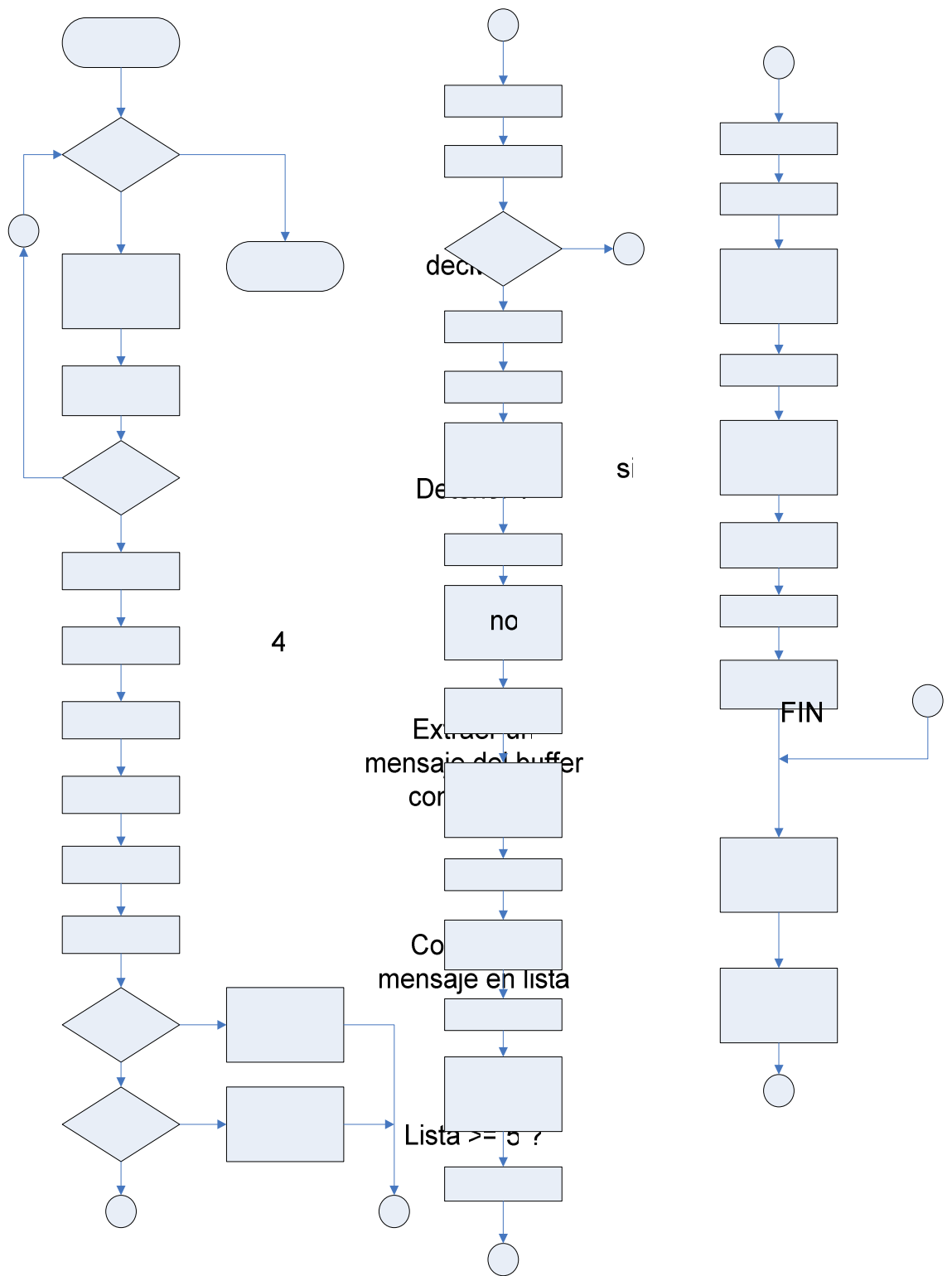


Figura 5.10 Lógica de la descodificación de nivel dos.  
 Extraer un byte

### 5.6.1.4 Codificación ISUP.

Como se mostraba en las secciones anteriores, ya se había decodificado una parte del paquete C7, sin embargo falta la decodificación de la estructura propia del protocolo ISUP la cual consiste en la traducción parámetro por parámetro tal como aparece en el datagrama de la figura 2.26.

Ya que cada parámetro posee un nombre, un código en binario correspondiente a dicho nombre y un valor en números enteros de octetos, podemos utilizar nuevamente la programación orientada a objetos para crear un objeto que traduzca dado el nombre, código y valor del parámetro, cada uno de los bit o arreglos de ellos, del parámetro dado. De esta manera, un bloque de código principal manejara este objeto enviando tantas peticiones de traducción como parámetros posea el mensaje.

La traducción de los parámetros consta de la comparación de los valores binarios de bloques de bit de los octetos, con una base de datos definida por la norma Q763.

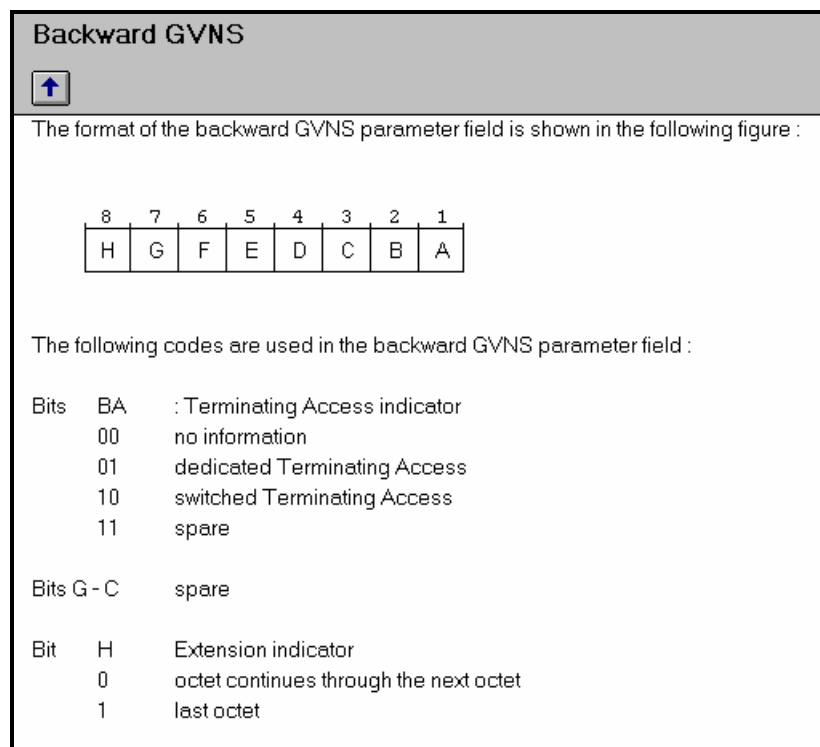


Figura 5.11 Ejemplo de tablas de traducción implementadas para la decodificación ISUP.

La figura 5.13 nos ilustra el proceso de comparaciones que hay que seguir para extraer la información que se encuentra implícita en los valores binarios de los segmentos de octetos en los parámetros.

Como puede observarse, el byte del parámetro en cuestión es dividido en segmentos para su codificación, y estas divisiones son propias de cada parámetro, por lo tanto los procesos de decodificación serán explícitos para determinado parámetro.

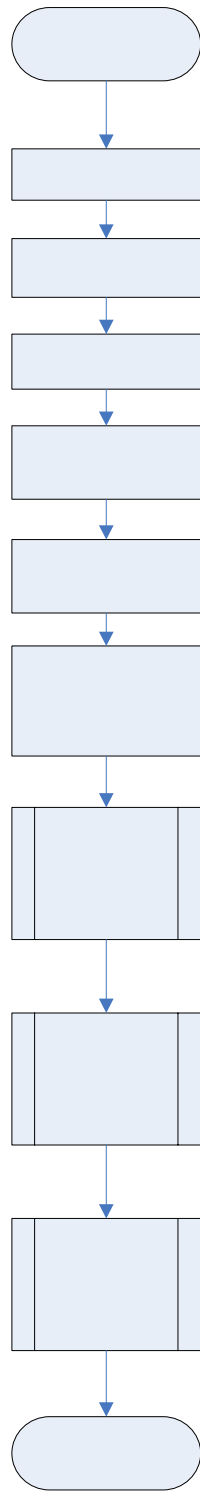
También será necesaria que dicha información de decodificación pueda ser retornada en un formato que permita a un bloque de código externo al objeto, imprimirla en pantalla de forma automática y como un proceso estándar para cada decodificación de cualquier parámetro.

La idea será producir una salida de decodificación para cada uno de los mensajes, tal y como se presenta en la siguiente figura:

Octeto	Hex	Dec	Bin	Tipo	Valor	Descripción
5	6b	107	01101011	MF	01101011	DPC=2155
6	88	136	10001000	MF	--001000	
12	16	22	00010110	MF	-----10	Backward Call Indicators Charge Indicator=Charge
				MF	----01--	Called Party's Status Indicator=Subscriber free
				MF	--01----	Called Party's Category Indicator=Ordinary subscriber
				MF	00-----	End to End Method Indicator=No end-end method available
13	34	52	00110100	MF	-----0	Interworking Indicator=No interworking encountered
				MF	-----0-	End to End Information Indicator=No end-end information available
				MF	----1--	ISDN User Part Indicator=ISUP used all the way

Figura 5.12 Salida en pantalla esperada de la decodificación ISUP.

En ella se puede observar que octeto tras octeto de la trama del mensaje, es convertido en su valor hexagesimal, decimal y binario, para luego dar paso a la decodificación de los segmentos de bits correspondientes al mismo, en un proceso que puede caracterizarse como repetitivo. Las siguientes figuras muestran los algoritmos implementados para la decodificación de los parámetros ISUP.



Inici

E

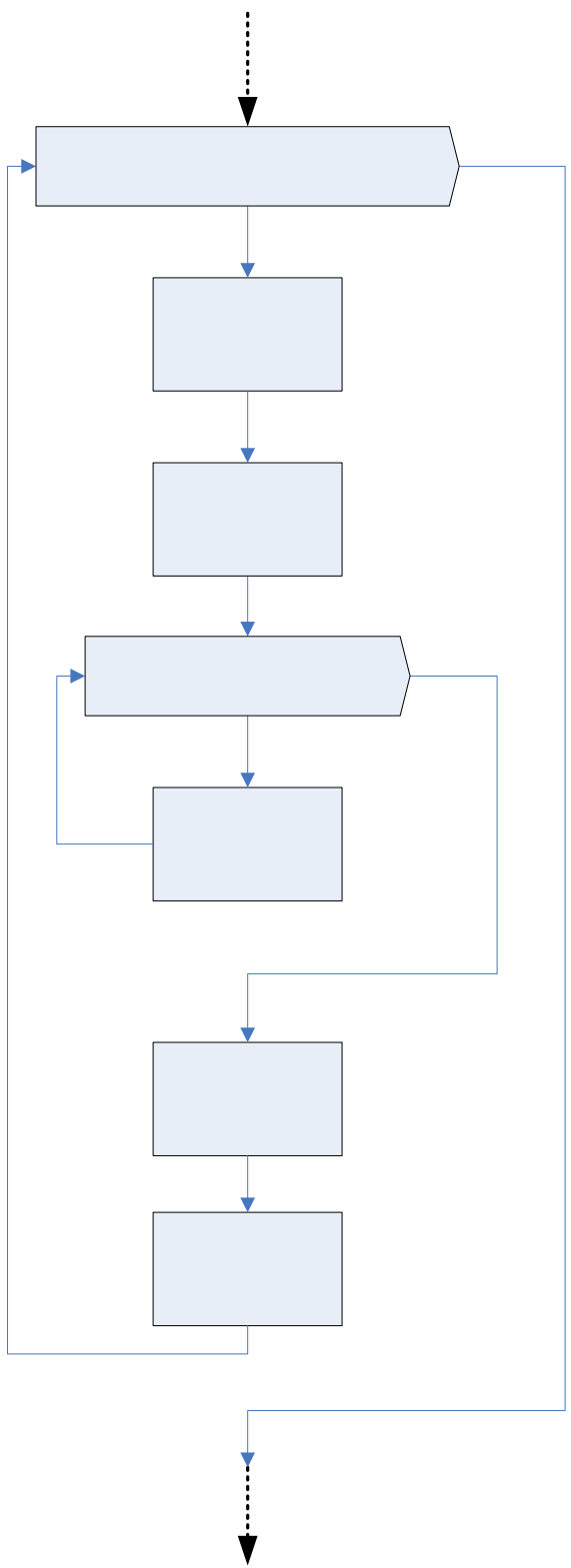
Deco

Extr

E

alm

Figura 5.13 *Lógica resumida de la decodificación ISUP.*



Por cao

Extr

M

Por

Figura 5.14. *Lógica de la extracción de los parámetros obligatorios fijos*

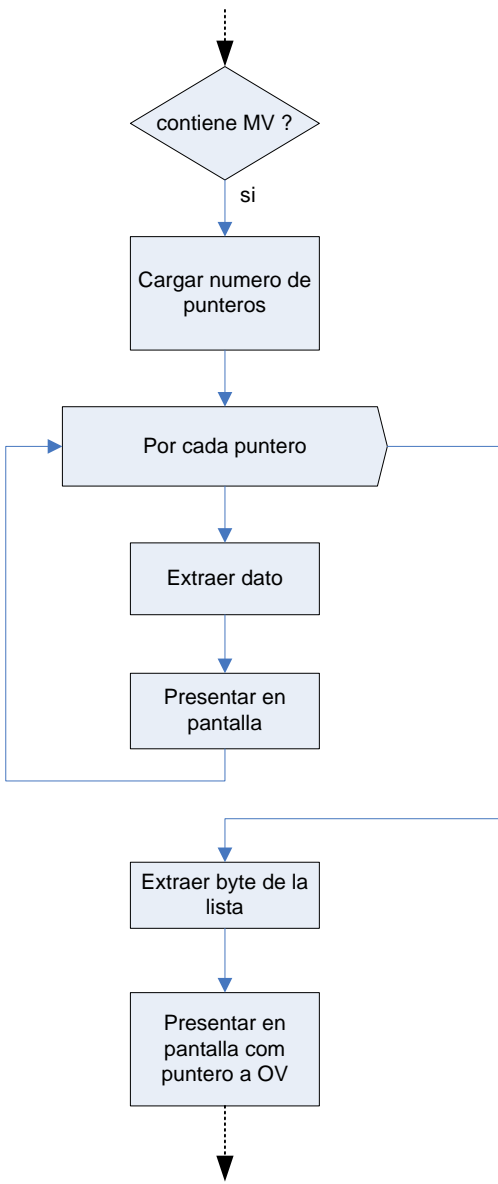


Figura 5.15 Lógica para la extracción de punteros.



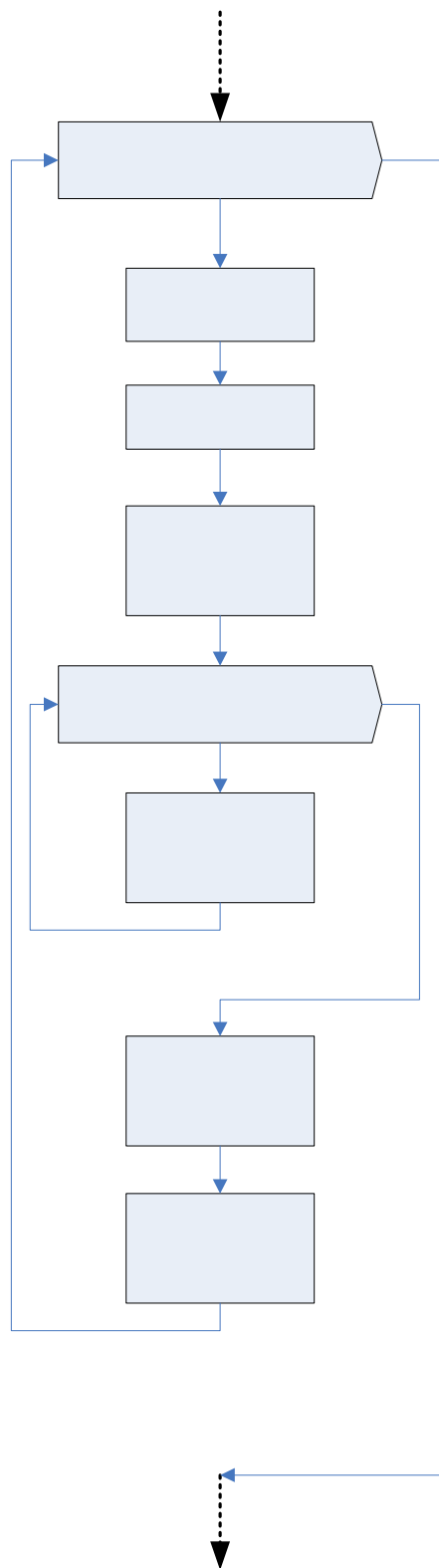
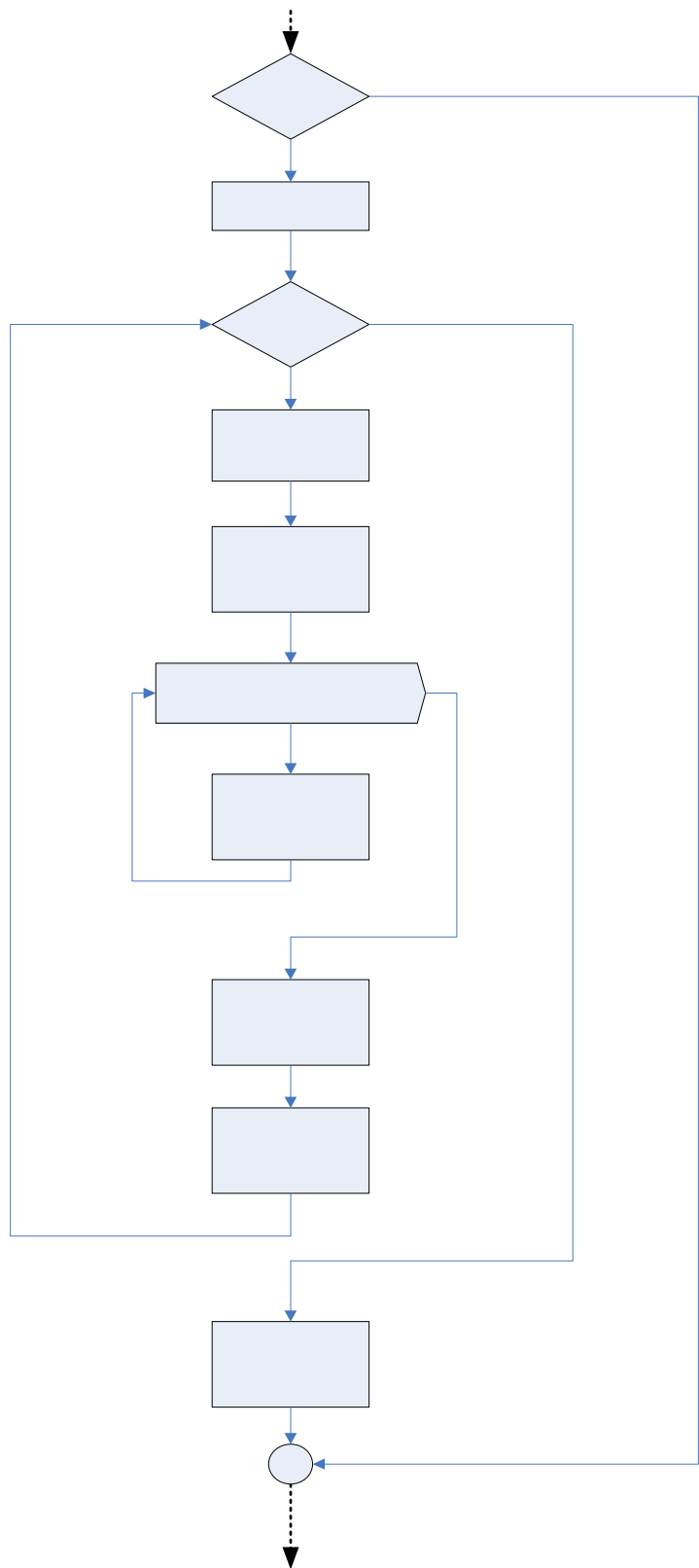


Figura 5.16 *Lógica para la extracción de parámetros obligatorios variables.*



Contiene O

si

Extraer codig  
parametr

Codigo != 0

si

Extraer nombr  
codigo de  
parametr

Extraer nume  
bytes de  
parametr

Por el # de byt  
parametr

Figura 5.17 Lógica para la extracción de parámetros facultativos.

En la decodificación ISUP, a partir de los datos almacenados en archivos y a través de la selección de un mensaje decodificado por Capa 2 en la lista mostrada por la interfaz grafica de usuario, se extrae la parte que contiene los parámetros ISUP (ISUPRAW). De la parte de ISUP se extrae el primer byte que indica el tipo de mensaje, después dependiendo del tipo de mensaje se extraen y decodifican los parámetros obligatorios fijos, se extraen y decodifican los parámetros obligatorios variables, se extraen y decodifican los parámetros optativos si el mensaje los posee.

ISUPdecoder	tools	Atributo para la descodificación del tipo de mensaje
	isup	Atributo de almacenamiento de los parámetros de usuario.
	MsgName	Atributo de almacenamiento del tipo de mensaje.
	value	Valor de cada parámetro
	MessageDecoded	Lista de almacenamiento mensaje decodificado MTP3
	name	Nombre del mensaje
	parametro	el nombre del parámetro ó subparametro decodificado
	meaning	Significado del parámetro decodificado.
	counter	Atributo para desplegar los parámetros de usuario.
	pointers	Punteros de los parámetros obligatorios.
	LVector	Longitud del vector de los parámetros.

Tabla 5.3 *Atributos de la Lógica de descodificación de parámetros de usuario ISUP.*

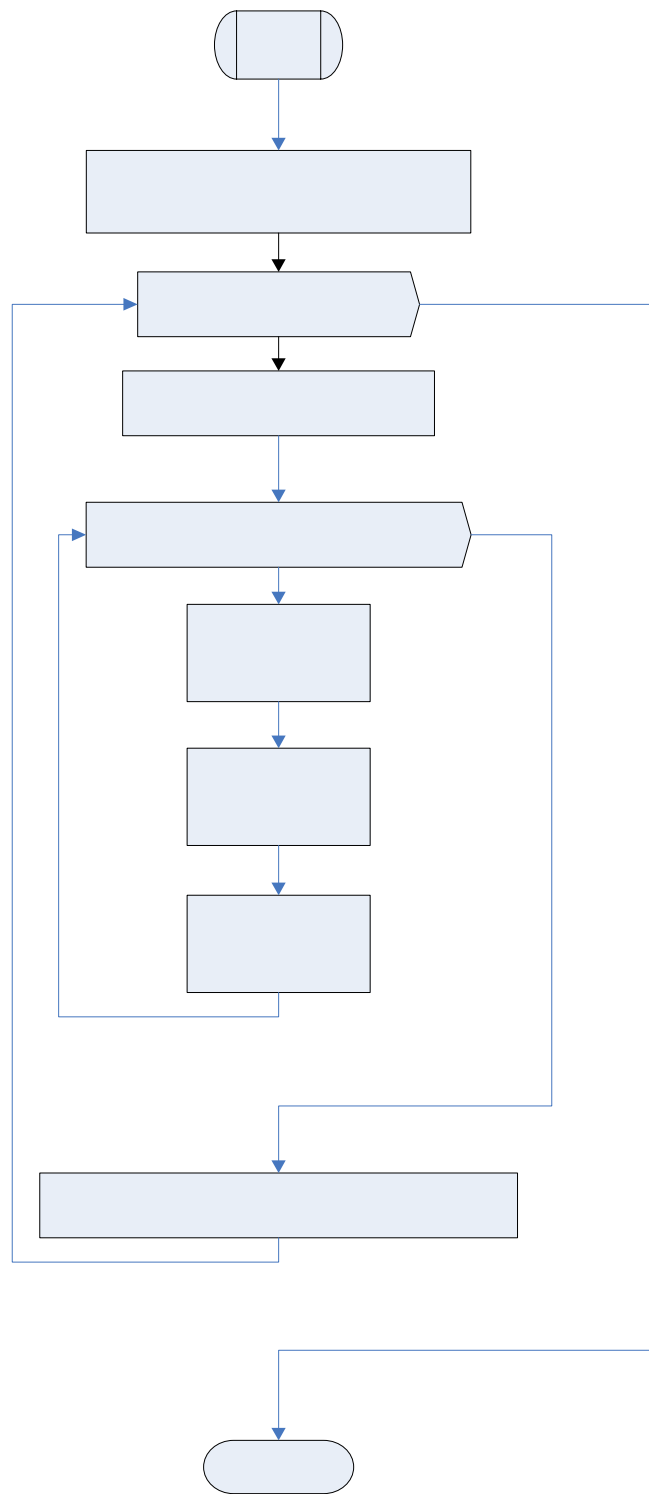


Figura 5.18 Lógica general aproximada para la decodificación de los parámetros de un mensaje

INI

Inicializar a cero  
decimales, signo  
binarios

Para cada  
Para

Codificar en  
guardarlo en su

Para cada **Segn**  
Actual (segun co  
paran

Deco  
signifi  
guarda  
'mea

Deco  
decimal  
en list

### 5.6.1.5 Captura en modo Traceo de Llamadas.

Este modo de puesta en marcha del sistema consiste en la identificación, codificación y muestra en pantalla, en tiempo real de grupos de mensajes asociados por el identificador de circuito de canal (CIC), el cual es asignado directamente por el nodo de señalización. Estos grupos de mensajes, son representados en la interfaz grafica de usuario por una única línea que contiene información de cada uno de los mensajes, información presentada secuencialmente a medida que se obtiene la información de cada uno de los mensajes que son capturados por el circuito en hardware.

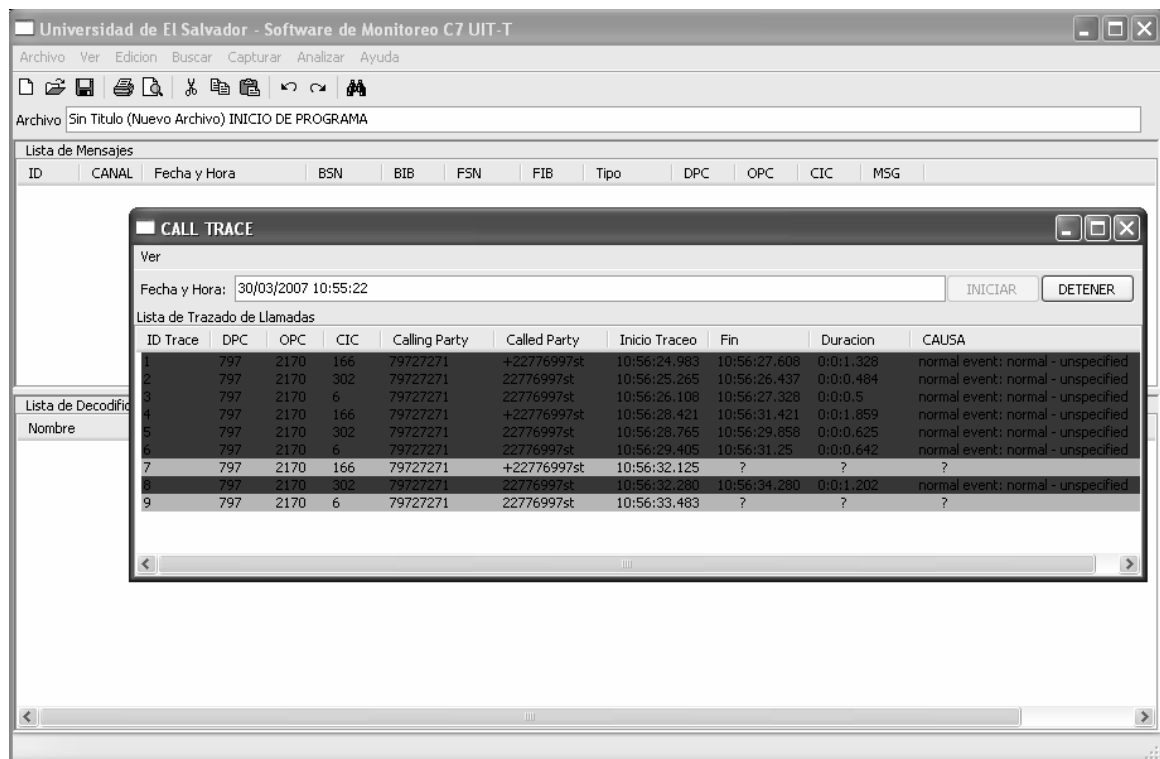


Figura 5.19 Ventana de capturas de llamadas.

Como se muestra en la figura 5.21, cada una de las líneas representa una sola llamada y esta asociada con un grupo de mensajes propios de la llamada. Cada llamada es un proceso independiente, por lo que los eventos de mensajes correspondientes a una llamada pueden ser paralelos y la información se muestra en tiempo real para cada una de las llamadas.

La lógica de programación y funcionamiento del sistema para captura de llamada es similar a la usada para la decodificación de capa 2 mostrada en el algoritmo de la figura 5.12. Sin embargo posee un mecanismo diferente en la forma en que los datos compartidos para el procedimiento de decodificación mensaje por mensaje.

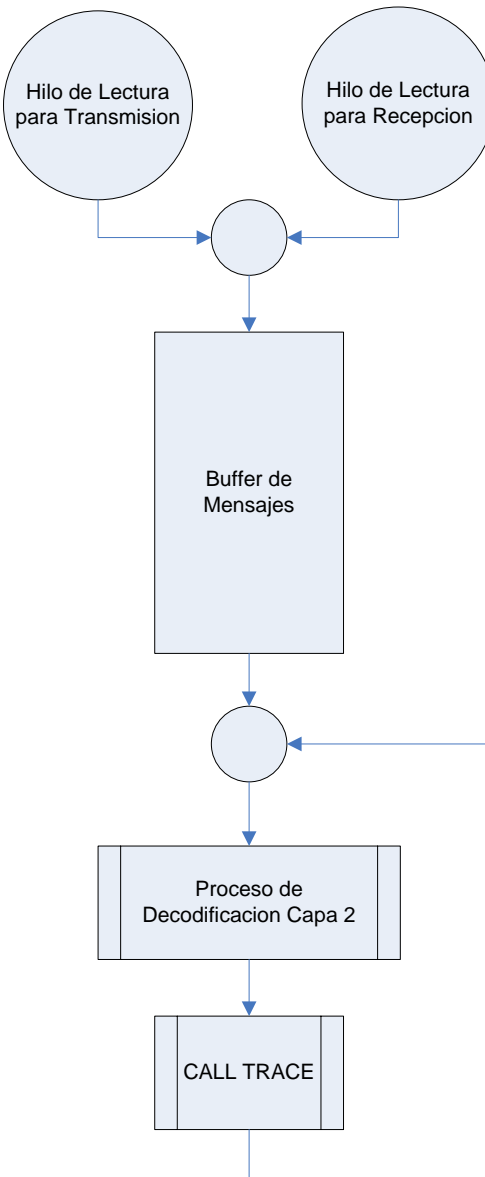
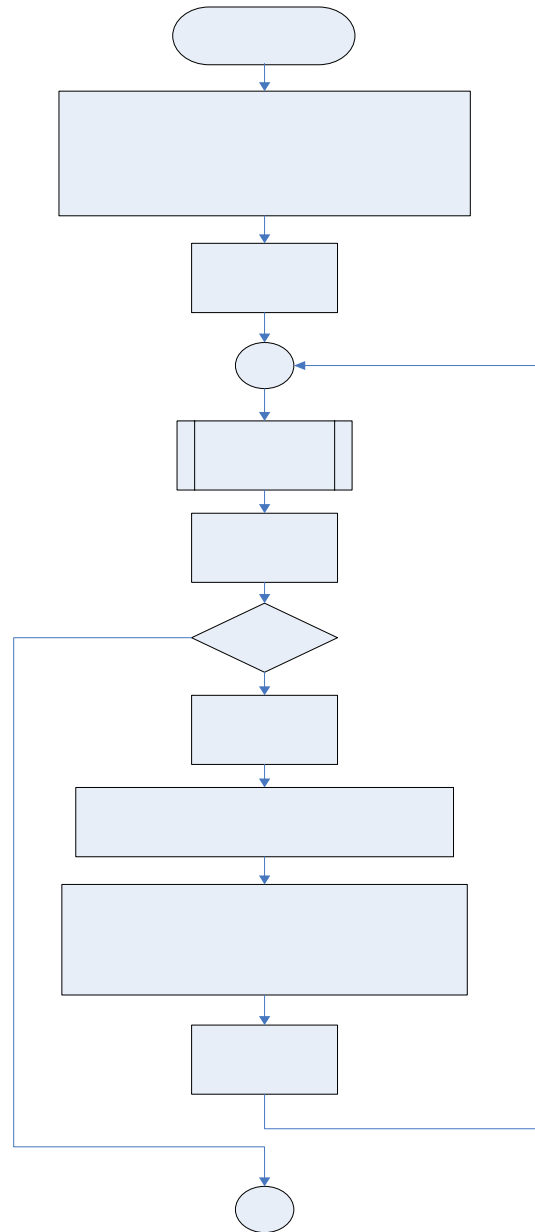


Figura 5.20 *Lógica de descodificación para captura de llamadas.*

Como se muestra en la figura 5.22, el proceso de descodificación utiliza únicamente tres hilos para su funcionamiento. Dos de ellos (los colectores de datos) idénticos a los mostrados en la figura 5.11, la diferencia radica en el procedimiento para la decodificación. En este nuevo esquema, la decodificación es efectuada únicamente por un solo hilo, el cual decodifica todos los mensajes que se comparten en el búfer. Los hilos colectores se encargan de alimentar a este único búfer. Una bandera indicando la dirección del mensaje (Tx y Rx) distingue entre los mensajes.

El proceso siguiente a la decodificación, es la lógica de rastreo de llamadas efectuada en el mismo ciclo de decodificación para cada mensaje, la cual consiste en verificar el CIC de un nuevo mensaje IAM y ligar todos aquellos mensajes que contienen el mismo CIC, agrupándolos en sub-conjunto de mensajes representados por una única línea en la interfaz grafica de usuario. Este proceso es auxiliado por vectores paralelos los cuales almacenan la información direccionada por el CIC de los mensajes IAM capturados secuencialmente. Los siguientes algoritmos muestran el funcionamiento de la lógica de rastreo de llamada:



Decod  
 Inicialización  
 Lista de ID  
 Lista de CICs  
 Lista de Listas de Me  
 Lista de Tiempos Inic  
 Lista de Tiempos Fin  
 Lista de Estado

Inici  
 co

Figura 5.21 Algoritmo para Rastreo de llamadas (Primera Parte).

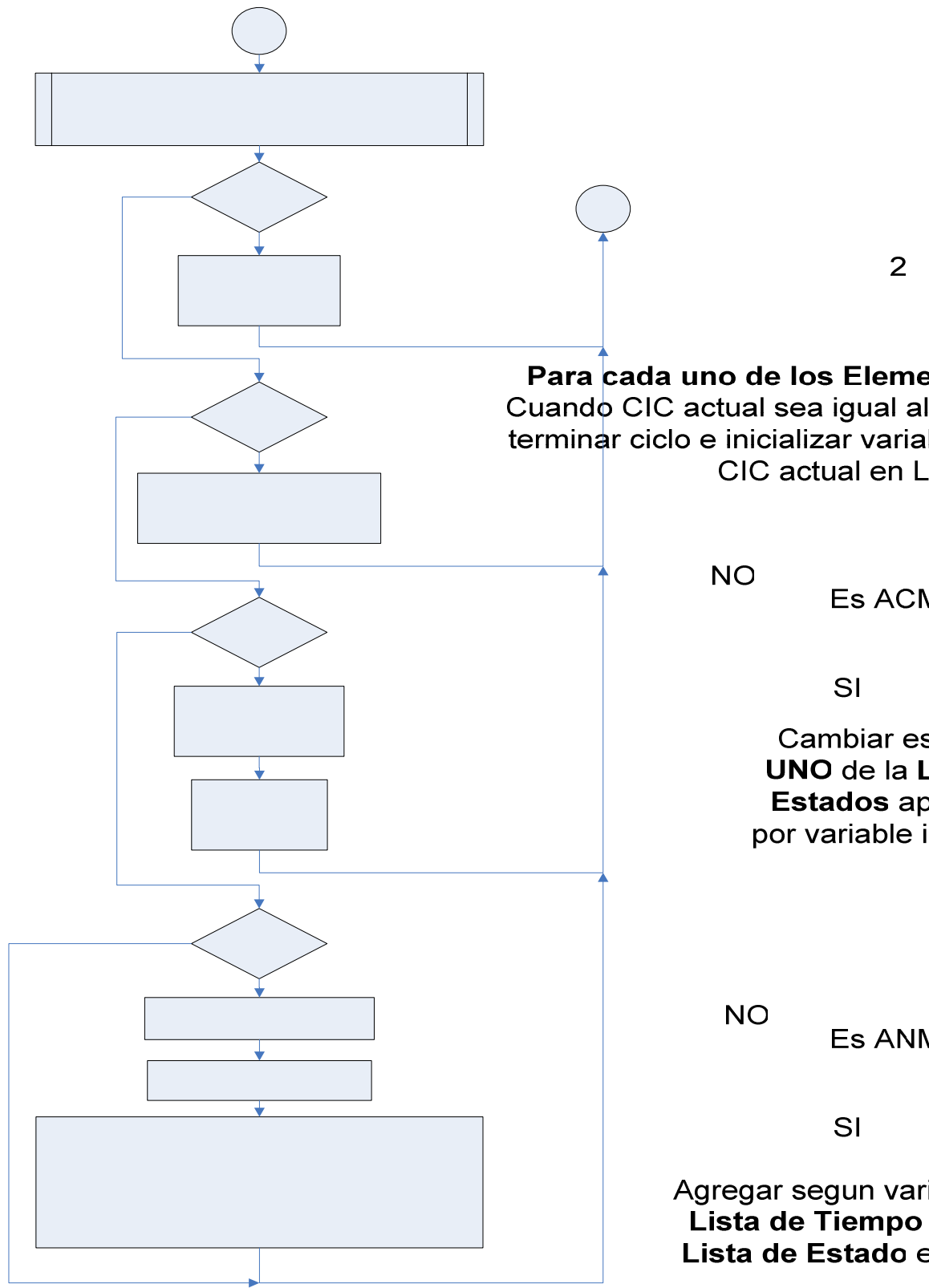


Figura 5.22 Algoritmo para Rastreo de llamadas (Segunda Parte).



hilosTraceMod DataColector	Lista de ID	Variable de PC de tipo vector que contiene el identificador numérico de una llamada. Trabaja en paralelo con las otras variables con Lista de CIC como vector principal.
	Lista de CICs	Variable de PC de tipo vector que contiene el CIC de cada llamada. vector principal.
	Lista de Listas de Mensajes Capa2	Variable de PC de tipo vector que contiene por elementos una lista que a su vez contiene todos los identificadores de mensajes relacionados con la llamada. Trabaja en paralelo con las otras variables con Lista de CIC como vector principal.
	Lista de Tiempos Iniciales	Variable de PC de tipo vector que contiene el tiempo inicial de una llamada. Trabaja en paralelo con las otras variables con Lista de CIC como vector principal.
	Lista de Tiempos Finales	Variable de PC de tipo vector que contiene el tiempo final de una llamada. Trabaja en paralelo con las otras variables con Lista de CIC como vector principal.
	Lista de Estados	Variable de PC de tipo vector que contiene el estado del proceso de decodificación de una llamada. Trabaja en paralelo con las otras variables con Lista de CIC como vector principal.

Tabla 5.4 *Atributos de la Lógica de decodificación en la captura de llamadas.*

### 5.6.1.6 Interfaz Grafica de Usuario.

La interfaz grafica de usuario es una herramienta de todo el sistema que permite la presentación grafica de los datos capturados por el circuito de interfaz en hardware y controla el funcionamiento de todo el sistema.

Representa el mecanismo por medio del cual, el usuario del sistema enviara ordenes de paro, activación, inicialización y configuración de las funciones que permiten la captura, almacenamiento y análisis post-captura de los datos obtenidos de la comunicación entre dos nodos de señalización.

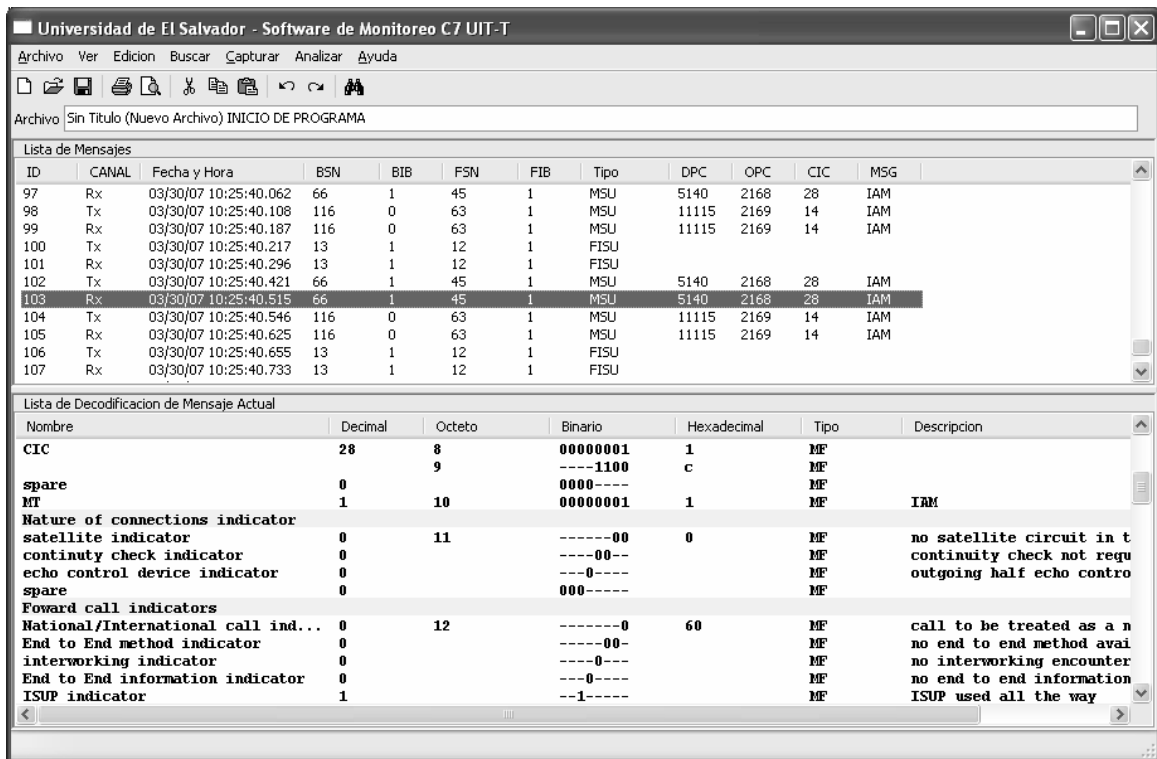


Figura 5.23 Captura de pantalla de Interfaz Grafica de Usuario.

La interfaz grafica de usuario esta constituida por objetos gráficos los cuales son definidos en un proceso de inicialización, antes de la puesta en marcha para su uso final. Estos elementos se pueden clasificar en:

- *Barra de menús*: contiene todos y cada uno de los comandos implementados para la manipulación del hardware y de todos los algoritmos implementados del sistema.
- *Barra de herramientas*: contiene la representación grafica de las funciones mayormente usadas de la barra de menús, en formas de botones.
- *Barra de archivo*: muestra información importante sobre el estado y nombre de los archivos de captura de datos.
- *Lista superior de decodificación*: objeto grafico del tipo de lista que muestra en forma secuencial, la lista de mensajes de capa 2 decodificados en tiempo real.
- *Lista inferior de decodificación*: objeto grafico del tipo de lista que muestra todos los parámetros de decodificación capa 4 (ISUP) de uno de los elementos de la lista superior. Esta función solo es activada cuando se selecciona un elemento de la lista superior y el sistema ha sido detenido después de una captura.

- *Barra de estado*: Muestra ayuda sobre las funciones de la barra de menús.

El procedimiento o lógica que se sigue para la inicialización de todos los elementos que contiene la interfaz grafica de usuario depende del paquete de desarrollo grafico usado, para nuestro caso de *wxPython* en el cual se definen los procedimientos y secuencias para generar los objetos y enlazarlos a las acciones de la lógica de decodificación. La siguiente figura muestra el algoritmo usado para tal fin.

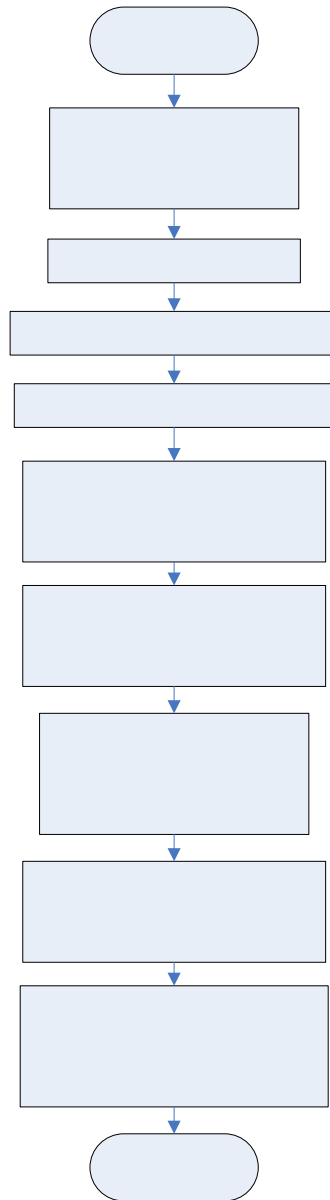


Figura 5.24 Algoritmo para configuración de Interfaz Grafica de Usuario.

C  
Crea  
Creacio  
Creacion  
Creacion de  
Creacion

## 5.7 CONCLUSIONES DEL CAPITULO V

El software desarrollado tiene la capacidad de decodificar señalización por canal común C7 y el protocolo ISUP contenida en cualquiera de los 31 canales transportados en una línea E1 (programable según mutuo acuerdo entre nodos de señalización), desplegando en pantalla información de los paquetes del protocolo capturados durante la ejecución del programa y permitiendo ver el detalle octeto por octeto de cada uno de estos.

El programa puede ser puesto en marcha en dos modos de funcionamiento: en *modo de monitoreo*, cuando los paquetes son mostrados uno a uno (a través de un control gráfico tipo lista) conforme sean capturados del circuito de interfaz y en *modo de traceo de llamada*, cuando los paquetes son agrupados de acuerdo a la información relacionada con una misma llamada. En este último modo los paquetes son filtrados de acuerdo al número de circuito y es mostrado únicamente (en un control gráfico tipo lista) un elemento representativo conteniendo la información más importante de esa llamada.

La estructura de programación utilizada (programación orientada a objetos y abstracción mediante clases) y la plataforma de programación utilizada (*python 2.5*), permite la implementación de nuevas funciones sin modificar el núcleo de código de funcionamiento, heredando de las clases ya definidas y sobre-escribiendo los métodos de estas.

Entre las mejoras que pueden ser añadidas se encuentra: las funciones de búsquedas de parámetros dentro de la lista de mensajes desplegados, análisis estadísticos de los parámetros e implementación de filtros complejos que permitan mostrar únicamente ciertos tipos de parámetros en los mensajes.

## 5.8 REFERENCIA BIBLIOGRÁFICA

- [1] Deitel and Deitel. "Python How to Program". Diciembre 2001. Segunda Edición
- [2] Dave Brueck and Stephen Tanner, "Python 2.1 Bible" 2001. Hungry Minds, Inc.
- [3] Noel Rappin and Robin Dunn. "wxPython in Action". 2006 Manning Publications Co.