

**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería en Electrónica**

**Cibertec Int. S. A.**

**“Diseño de la interfaz PCI para la tarjeta HDLC-4M”**

**Informe de Proyecto de Graduación para optar por el Grado de Bachiller en  
Ingeniería Electrónica**

**Simón Roberto Sánchez Segura**

**Cartago, Junio 2001**

## DEDICATORIA

*Al Señor Jesucristo, por ser mi salvador y guía.*

*A mi madre, por ser inspiración y soporte en la adversidad.*

*A mi familia y amigos, por creer en mí siempre.*

## **AGRADECIMIENTO**

Quiero extender un agradecimiento especial al Ingeniero Alejandro Tenorio, por darme la oportunidad de trabajar en este proyecto. La experiencia adquirida durante los últimos meses ha sido de mucho provecho para mi carrera.

También deseo agradecer a la Junta Directiva de Cibertec Int., por abrirme las puertas de esta destacada compañía; y a los compañeros de trabajo que han sido una motivación para hacer las cosas mejor cada día.

Finalmente, les doy las gracias a mis profesores y compañeros del Instituto Tecnológico de Costa Rica, que durante estos años han hecho un gran aporte a mi formación como persona de bien.

## RESUMEN

El siguiente trabajo se desarrolló con el fin de cubrir una necesidad de la empresa Cibertec Int.: diseñar una tarjeta PCI capaz de recibir los datos de señalización SS7 proveniente de sus equipos de muestreo, y filtrar esta información para que el computador supervisor reciba únicamente los mensajes más importantes. Este módulo se desarrolló pensando en la atención de múltiples enlaces E1 en paralelo, de forma que la información se concentre en un punto de mayor capacidad de procesamiento.

El primer paso fue una investigación teórica, seguida por la búsqueda de componentes disponibles en el mercado y el correspondiente proceso de selección. Luego se procedió a estudiar a fondo la manera de conectar los elementos para finalizar con una versión definitiva del circuito en Orcad. Además, se investigó a fondo sobre la forma de programar los chips y específicamente, las rutinas que debe llevar el código para una correcta interacción de la tarjeta con el bus PCI.

Los objetivos del proyecto finalizan con la versión definitiva del hardware y las rutinas de arranque de la tarjeta. La continuación de este proyecto hacia niveles más avanzados queda a cargo del departamento de Investigación y Desarrollo de Cibertec Int.

Palabras clave: HDLC, 80960RM, PM7366, Freedm8, ST-Bus, PCI.

## ABSTRACT

The present work was designed to fulfill an existing need for the company Cibertec International: designing a PCI card capable of receiving signaling data SS7, coming from its sampling equipment and also filter that information so the supervisor computer receives only the most important messages. This module was developed thinking in attention of multiple E1 parallel links, in order for the information to get concentrated on a bigger processing capacity spot.

The first step was the theoretical research, followed by the search of available components in the market and the corresponding selection process. Afterwards there was a study about how to connect the elements to obtain the final circuit version on ORCAD. Besides, there was a study on how to program the chips, specifically, which routines must the code follows to have a correct interaction between the card and the PCI bus.

The project goals end with the final hardware version and the card start routines. The responsibility of this project's advanced levels remains in charge of the Investigation and Development Department at Cibertec International.

Keywords: HDLC, 80960RM, PM7366, Freedm8, ST-Bus, PCI.

## ÍNDICE GENERAL

<b>Capítulo 1 Introducción</b>	<b>14</b>
<b>1.1 Descripción de la empresa</b>	<b>15</b>
<b>1.2 Definición del problema y su importancia</b>	<b>19</b>
<b>1.3 Objetivos</b>	<b>22</b>
<b>Capítulo 2 Antecedentes</b>	<b>24</b>
<b>2.1 Estudio del problema a resolver</b>	<b>25</b>
<b>2.1.1 El ITSS (International Traffic Security System)</b>	<b>25</b>
<b>2.1.1.1 Métodos usados para el fraude de tráfico telefónico</b>	<b>26</b>
<b>2.1.1.2 Procedimientos de llamadas alternas ilegales</b>	<b>28</b>
<b>2.1.1.3 Arquitectura del ITSS</b>	<b>37</b>
<b>2.1.2 Funcionamiento de la HDLC-256 dentro del ITSS</b>	<b>40</b>
<b>2.1.2.1 Formato de comunicación serie ST-Bus</b>	<b>41</b>
<b>2.1.2.2 Protocolo de comunicación HDLC</b>	<b>44</b>
<b>2.1.2.3 Descripción por bloques de la HDLC-256</b>	<b>47</b>
<b>2.1.2.4 Descripción del chip MT8952B</b>	<b>49</b>
<b>2.1.2.5 Características básicas del microprocesador TMS320C26</b>	<b>55</b>
<b>2.1.2.6 Conexión y comunicación con el bus ISA</b>	<b>58</b>
<b>2.2 Requerimientos de la empresa</b>	<b>67</b>
<b>2.2.1 Requerimientos en la entrada del bus serial</b>	<b>67</b>
<b>2.2.2 Requerimientos de capacidad de procesamiento</b>	<b>69</b>
<b>2.2.3 Requerimientos en el bus de la computadora</b>	<b>70</b>

2.2.4	Requerimientos de instalación, configuración y PnP	71
2.2.5	Requerimientos de vigencia en el mercado	71
2.2.6	Requerimientos de montaje de los componentes	71
2.3	Solución propuesta	72
<b>Capítulo 3</b>	<b>Procedimiento metodológico</b>	<b>78</b>
3.1	Investigación preliminar	81
3.2	Preparación del diagrama funcional	82
3.3	Análisis del estándar PCI	83
3.4	Conexión del circuito puente	85
3.5	Diseño de la estructura del bus secundario	85
3.6	Uso del bus secundario	86
3.7	Programación de encabezados	87
3.8	Documentación	88
3.9	Adquisición de licencias	89
3.10	Contratación de empresas para el PCB y el ensamble	90
3.11	Transferencia tecnológica	91
<b>Capítulo 4</b>	<b>Descripción del hardware utilizado</b>	<b>92</b>
4.1	Introducción al bus PCI	93
4.2	Aspectos básicos de la arquitectura PCI	96

<b>4.2.1 Rendimiento del bus PCI</b>	<b>98</b>
<b>4.2.2 Ranuras de expansión PCI</b>	<b>99</b>
<b>4.2.3 Descripción de señales</b>	<b>101</b>
<b>4.2.4 Ciclos de Bus</b>	<b>104</b>
<b>4.2.4.1 Ciclo de reconocimiento de interrupción (0000)</b>	<b>105</b>
<b>4.2.4.2 Ciclo especial (0001)</b>	<b>106</b>
<b>4.2.4.3 Lectura de I/O (0010) y escritura de I/O (0011)</b>	<b>107</b>
<b>4.2.4.4 Lectura de memoria (0110) y escritura de memoria (0111)</b>	<b>108</b>
<b>4.2.4.5 Lectura (1010) y escritura (1011) de configuración</b>	<b>108</b>
<b>4.2.4.6 Ciclo de dirección dual (1101)</b>	<b>109</b>
<b>4.2.4.7 Memory-Read Line (1110)</b>	<b>109</b>
<b>4.2.4.8 Memory Write and Invalidate (1111)</b>	<b>109</b>
<b>4.3 Arquitectura genérica de una plataforma PCI</b>	<b>109</b>
<b>4.3.1 Buses fuente y destino</b>	<b>111</b>
<b>4.3.2 Propiedad del bus PCI</b>	<b>113</b>
<b>4.3.3 Recursos PCI, dispositivos y carga del bus</b>	<b>113</b>
<b>4.3.4 Central Resource</b>	<b>114</b>
<b>4.3.5 Árbitro Central</b>	<b>115</b>
<b>4.3.6 Señales de interrupción</b>	<b>116</b>
<b>4.3.7 Líneas de señal tipo “Sideband”</b>	<b>117</b>
<b>4.3.8 Señales J-TAG</b>	<b>117</b>
<b>4.4 Requerimientos eléctricos del bus PCI</b>	<b>119</b>
<b>4.4.1 Distinción entre tarjetas de expansión según el voltaje del bus</b>	<b>119</b>
<b>4.4.2 Señales de reloj</b>	<b>121</b>
<b>4.4.3 Resistencias de Pull-Up</b>	<b>123</b>

4.4.4 Distribución de los pines en la ranura y los dispositivos	124
4.4.5 Diseño de impresos para tarjetas de expansión	130
4.4.6 Condensadores de desacople	131
4.5 Plug and Play	131
4.5.1 Requerimientos para Plug and Play	132
4.5.2 Operación del Plug and Play	134
4.5.3 Datos de configuración extendida del sistema (ESCD)	135
4.5.4 PnP y dispositivos no PnP	136
4.6 Descripción del microprocesador 80960-RM	137
4.6.1 Puente PCI a PCI	143
4.6.1.1 Interfaz PCI primaria	147
4.6.1.2 Interfaz PCI secundaria	148
4.6.1.3 Colas ascendentes y descendentes	149
4.6.1.4 Registros de configuración	151
4.6.2 Accesos de configuración	152
4.6.3 Unidad de traslado de direcciones	154
4.7 Descripción del PM-7366	154
4.8 Descripción del clock-buffer	159
Capítulo 5 Descripción del software del sistema	160
5.1 Tipos de dato y direccionamiento	162
5.2 Elementos principales del ambiente	163
5.3 Registros de control ubicados en espacio de memoria (MMRs)	165

5.3.1 Programación de los registros del CORE	169
5.3.2 Programación de los registros del BIU	174
5.3.3 Programación de los registros del puente PCI-PCI	177
5.4 Procedimiento de inicio del 80960-RM	186
<b>Capítulo 6 Análisis de resultados</b>	<b>189</b>
6.1 Explicación del diseño	190
6.2 Alcance y limitaciones	197
<b>Capítulo 7 Conclusiones y recomendaciones</b>	<b>199</b>
7.1 Conclusiones	200
7.2 Recomendaciones	202
<b>Capítulo 8 Referencias consultadas</b>	<b>203</b>
8.1 Bibliografía	204
8.2 Hojas de datos	204
8.3 Documentos en formato PDF	204
8.4 Direcciones de Internet	205
8.5 CD-ROM	205
<b>Capítulo 9 Apéndices</b>	<b>206</b>
9.1 Características mecánicas de los componentes	207
9.2 Último avance recibido con relación al diseño del impreso	214

## ÍNDICE DE FIGURAS

<b>Figura 1.1</b>	Organigrama de Cibertec Int.	18
<b>Figura 1.2</b>	Diagrama funcional por bloques de la HDLC-256.	20
<b>Figura 2.1</b>	Interconexión de la HDLC-256 con su equipo periférico.	41
<b>Figura 2.2</b>	Alineación del ST-Bus usando F0\ y C4\.	43
<b>Figura 2.3</b>	Estructura de la HDLC-256, con sus principales componentes.	48
<b>Figura 2.4</b>	Descripción funcional por bloques del MT8952B.	51
<b>Figura 2.5</b>	Ingreso de los datos a la tarjeta a través de los ST-Bus.	54
<b>Figura 2.6</b>	Estructura lógica de los ST-Bus en la entrada de la HDLC-256.	54
<b>Figura 2.7</b>	Diagrama de bloques simplificado del TMS320C26.	57
<b>Figura 2.8</b>	Temporización de la escritura en la memoria CY7C419.	60
<b>Figura 2.9</b>	Arreglo de circuitos que permiten usar el registro de estado.	63
<b>Figura 2.10</b>	Modificaciones en la distribución de los datos en la entrada.	68
<b>Figura 2.11</b>	Interconexión de la HDLC-4M con su equipo periférico y con el PC.	73
<b>Figura 2.12</b>	Estructura aprobada en Cibertec para el módulo HDLC-4M.	75
<b>Figura 4.1</b>	Arquitectura de Plataforma Genérica con un solo bus PCI.	112
<b>Figura 4.2</b>	Tipos de ranuras y tarjetas PCI, según su llave específica.	121
<b>Figura 4.3</b>	Desfase en la señal de reloj PCI de dos dispositivos.	122
<b>Figura 4.4</b>	Formas de onda esperadas para los relojes PCI.	123
<b>Figura 4.5</b>	Distribución de pines sugerida para un componente PCI - PQFP.	129
<b>Figura 4.6</b>	Diagrama de bloques del 80960-RM.	140
<b>Figura 4.7</b>	Diagrama de bloques del módulo puente, dentro del 80960RM.	145
<b>Figura 4.8</b>	Operación del circuito puente: selección de transacciones.	148
<b>Figura 4.9</b>	Diagrama de bloques del PM-1766.	155
<b>Figura 4.10</b>	Diagrama de bloques del clock-buffer.	159
<b>Figura 5.1</b>	Entorno de programación del procesador 80960-RM.	164
<b>Figura 5.2</b>	Mapa de memoria básico en el bus local del 80960RM.	168
<b>Figura 5.3</b>	Estructura de los registros PMCON.	171
<b>Figura 5.4</b>	Estructura del registro BCON.	171

<b>Figura 5.5</b>	Estructura de los registros que forman el LMCON.	173
<b>Figura 5.6</b>	Estructura de los registros que forman el LMCON.	173
<b>Figura 5.7</b>	Ubicación del bus local dentro de la estructura del 80960-RM.	175
<b>Figura 5.8</b>	Estructura de los MMRs usados por el BIU.	176
<b>Figura 5.9</b>	Estructura del registro PCR.	183
<b>Figura 5.10</b>	Estructura del registro PSR.	184
<b>Figura 5.11</b>	Estructura del registro BCR.	185
<b>Figura 5.12</b>	Diagrama de flujo para la iniciación del procesador.	188
<b>Figura Ap1</b>	Presentación mecánica del controlador PM7366.	207
<b>Figura Ap2</b>	Presentación mecánica del procesador 80960-RM.	208
<b>Figura Ap3</b>	Presentación mecánica de los componentes discretos.	208
<b>Figura Ap4</b>	Presentación mecánica del comparador LMV331.	209
<b>Figura Ap5</b>	Presentación mecánica del clock-buffer CY2505.	209
<b>Figura Ap6</b>	Presentación mecánica del socket para la SDRAM.	210
<b>Figura Ap7</b>	Presentación mecánica del módulo DIMM MT4LSDT1664A.	211
<b>Figura Ap8</b>	Presentación mecánica del Latch 74LCX841WM.	211
<b>Figura Ap9</b>	Presentación mecánica de la EPROM M27C2001-55XL1X.	212
<b>Figura Ap10</b>	Presentación mecánica de la base para EPROM.	213
<b>Figura Ap11</b>	Avance enviado por Dataline PCB: HDLC-4M, cara superior.	214
<b>Figura Ap12</b>	Avance enviado por Dataline PCB: HDLC-4M, layout superior.	215
<b>Figura Ap14</b>	Avance enviado por Dataline PCB: HDLC-4M, cara inferior.	216
<b>Figura Ap15</b>	Avance enviado por Dataline PCB: HDLC-4M, layout inferior.	217

## ÍNDICE DE TABLAS

<b>Tabla 2.1</b>	Formato de un cuadro HDLC.	45
<b>Tabla 2.2</b>	Función de los terminales en IC9.	64
<b>Tabla 2.3</b>	Función de los terminales en IC14.	65
<b>Tabla 2.4</b>	Función de los terminales en IC15.	66
<b>Tabla 2.5</b>	Función de los terminales en IC17.	67
<b>Tabla 4.1</b>	Descripción de las señales usadas en PCI.	102
<b>Tabla 4.2</b>	Codificación de los comandos en el estándar PCI.	104
<b>Tabla 4.3</b>	Ciclos especiales en el bus PCI.	107
<b>Tabla 4.4</b>	Condiciones de medición usadas para determinar T <sub>skew</sub> .	122
<b>Tabla 4.5</b>	Requerimientos temporales básicos de los relojes PCI, @ 33 MHz.	123
<b>Tabla 4.6</b>	Resistencias de Pull-Up para señales del bus PCI.	124
<b>Tabla 4.7</b>	Distribución de pines en una ranura PCI.	125
<b>Tabla 4.8</b>	Distribución de pines en una ranura PCI (continuación).	126
<b>Tabla 4.9</b>	Distribución de pines en una ranura PCI (continuación).	127
<b>Tabla 4.10</b>	Distribución de pines en una ranura PCI (continuación).	128
<b>Tabla 4.11</b>	Identificación de la potencia consumida por una tarjeta PCI.	129
<b>Tabla 4.12</b>	Estructura del comando de configuración.	153
<b>Tabla 5.1</b>	Diferencias de nomenclatura entre PCI y 80960.	162
<b>Tabla 5.2</b>	Regiones controladas por los registros PMCON.	170
<b>Tabla 5.3</b>	Formato del encabezado de configuración del puente.	180
<b>Tabla 5.4</b>	Conjunto de MMRs usados por la Unidad Puente PCI-PCI.	181
<b>Tabla 5.5</b>	Conjunto de MMRs usados por la Unidad Puente PCI-PCI (cont.).	182
<b>Tabla 5.6</b>	Configuración del código de clase para el 80960-RM	185
<b>Tabla 5.7</b>	Modos de iniciación del 80960-RM.	186
<b>Tabla 6.1</b>	Factibilidad: Uso de un bridge disponible en el mercado.	190
<b>Tabla 6.2</b>	Factibilidad: Uso de un bridge disponible en el mercado (cont.).	191
<b>Tabla 6.3</b>	Factibilidad: Uso de circuitería no especializada.	192
<b>Tabla Ap1</b>	Dimensiones del controlador PM7366.	207

<b>Tabla Ap2</b>	Dimensiones de la EPROM M27C2001-55XL1X.	212
<b>Tabla Ap3</b>	Dimensiones límite permitidas para una tarjeta PCI tipo estándar.	213

# **CAPÍTULO 1**

## **INTRODUCCIÓN**

## **1.1 Descripción de la empresa**

Cibertec Internacional S.A. es una pequeña empresa multinacional fundada en 1979. Su planta de ensamblaje está ubicada en el Parque Industrial Zona Franca de Alajuela (SARET), Costa Rica. Su oficina de diseño y desarrollo de programación está ubicada en San José, en la vecindad de la Universidad de Costa Rica. La oficina de ventas y finanzas está ubicada en Ciudad de Panamá.

Sus veinte años los ha dedicado principalmente al desarrollo de alta tecnología electrónica para la industria de las telecomunicaciones.

Hoy día el producto principal es el "ITSS" (International Traffic Security System) el cual se ha desarrollado para prevenir el fraude telefónico en el servicio de larga distancia internacional. El sistema, que trabaja bajo plataforma de redes, fue desarrollado en su totalidad por esta compañía.

La maquinaria y equipo especializado consiste en herramientas menores, mesas de trabajo, equipo de ensamble de tarjetas electrónicas, equipo normal de oficina, además de instrumentos y equipos para aplicaciones electrónicas y de cómputo de alta tecnología, empleado en laboratorio de control de calidad.

Cibertec Int. Costa Rica no genera ventas, ya que es una empresa ensambladora de equipo de telecomunicaciones, que transfiere su producción al costo a Cibertec Int. Panamá, que se encarga de las ventas. La planta en Costa Rica incluye en su presupuesto de operaciones los siguientes tópicos: costos de conversión, gastos de mantenimiento e instalación, materiales, partes locales y gastos extraordinarios para los proyectos programados a producirse durante el año.

La principal instancia de Cibertec es su Junta Directiva, formada por los Ingenieros Juan Carlos Halphen, Bernal Thalman, Hernán Jiménez y el Licenciado Orlando Jiménez.

Bajo este nivel de jerarquía se encuentra la Gerencia General, a cargo del Ingeniero Luis Jiménez, y es desde aquí donde se dirigen los otros departamentos.

En Costa Rica la empresa cuenta con un aproximado de 50 empleados, tomando en cuenta ingenieros, secretarias, contadores, operarios, inspectores de calidad, dibujantes, etc.

En el local de San José laboran 20 personas que se dedican principalmente a la programación, contaduría, administración y desarrollo de hardware. Debido al constante traslado de personas al extranjero para realizar tareas específicas, este número es muy variable.

Por su parte, en Alajuela se encuentra la sección de operarios, que corresponde al grueso del personal. Este grupo se encarga de las labores técnicas: ensamblaje y preparación de los equipos.

Una de las características más importantes de la compañía Cibertec, que le ha permitido tener un considerable crecimiento y una gran proyección, ha sido la inversión que dedica a la búsqueda e implementación de proyectos. El Departamento de Investigación y Desarrollo es el que se encarga de generar la tecnología necesaria para satisfacer las necesidades de los clientes.

Por medio de un proceso de investigación, cotización de partes, diseño y pruebas, se persigue mantener un alto nivel tecnológico en todas las aplicaciones. Además, por medio de los ciclos de desarrollo, se consigue mejorar el desempeño de los sistemas y agregar funciones más elaboradas.

En Investigación y Desarrollo trabajan once personas, cuatro de las cuales son actualmente practicantes de universidades estatales, mientras que los siete restantes son Ingenieros.

Los encargados de hardware se dedican casi exclusivamente a la investigación de nuevos circuitos de aplicación, mientras que los de software se ocupan de controladores, bases de datos y plataformas para las aplicaciones.

El Coordinador General de Investigación y Desarrollo es el Ingeniero Alejandro Tenorio, que trabaja en Cibertec desde hace 8 años.

Este departamento se localiza casi en su totalidad en el edificio de San José, sin embargo en ciertos procesos de prueba es necesario utilizar el laboratorio de Alajuela, por lo cual se requiere la movilización hasta la planta de SARET.

Como puede verse en la Figura 1.1, la empresa en Costa Rica es dirigida por el Ing. Luis Jiménez, que ocupa el cargo de Gerente General. Además está presente la Junta Directiva, conformada por los dueños de la compañía. Esta Directiva es quien toma las decisiones de mayor importancia, y se encargan de verificar que todo el trabajo esté bien coordinado.

Como ya se mencionó, la producción de Cibertec Int. Costa Rica se exporta, como equipo vendido a Cibertec Panamá. Esta última entidad es la encargada de las ventas, contabilidad y finanzas de la empresa en general. Por esta razón, los encargados de estos departamentos en Costa Rica funcionan hasta cierto punto como asistentes del personal en Panamá, tomando en cuenta que su responsabilidad local es ante el Gerente General.

Por lo amplio del mercado, existe un grupo de ingenieros “viajeros” encargados de la instalación y mantenimiento de los equipos.



## **1.2 Definición del problema y su importancia**

El principal producto de Cibertec es el ITSS, tal y como se menciona en la sección anterior. Una de las etapas de este sistema se encarga de comunicar la información recopilada por los equipos periféricos a una computadora, para su posterior análisis y almacenaje.

Esta etapa se denomina HDLC-256 y fue diseñada por los Ingenieros de Cibertec hace varios años con el fin de permitir la decodificación de información relacionada con señalización SS7 en las versiones de UIT y ANSI.

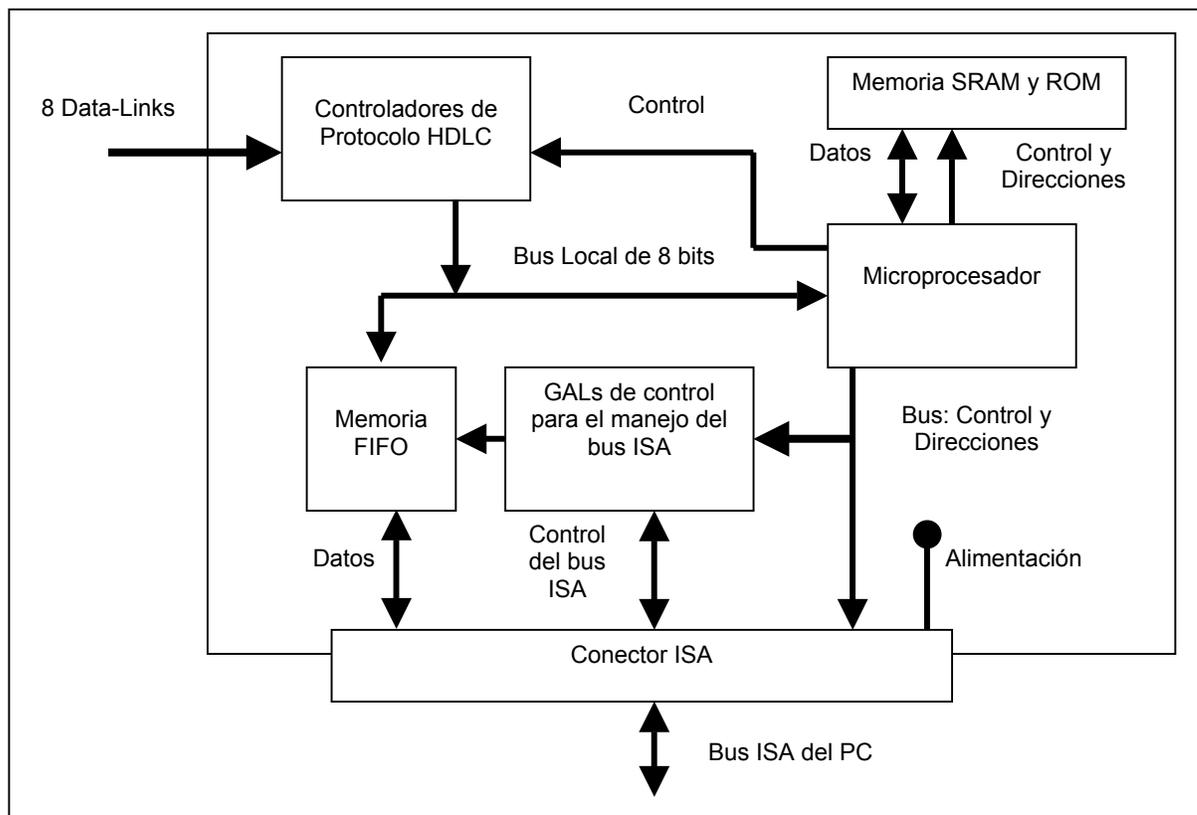
La HDLC-256 se conecta al bus ISA, y por lo tanto trabaja basada en los estándares propios de este bus. La tecnología de fabricación se basa en montajes DIP y QFP, utilizando dos o más capas en el impreso, y con soldadura convencional por el método de perforaciones en la tarjeta.

La información que viene de los equipos externos corresponde a 8 data links en formato ST-Bus, que contienen información concerniente a los protocolos de comunicación de las llamadas telefónicas. Analizando esta información se puede determinar si una llamada específica está siendo procesada usando métodos fraudulentos o ilegales.

Esta tarjeta de expansión trabaja en conjunto con el resto del ITSS, y no se utiliza en forma independiente para realizar otras aplicaciones. Por lo tanto, se produce únicamente para trabajar como etapa integral del Sistema de Seguridad de Tráfico Internacional y no para la venta individual.

En la Figura 1.2 se muestra el diagrama de bloques de la HDLC-256. Aquí se muestran los componentes principales y el flujo de los datos.

La tarjeta utiliza un procesador para gobernar todos los movimientos de información, 8 circuitos decodificadores de protocolo HDLC, bancos de memoria básicos, una cola (memoria FIFO) para comunicación con la computadora y un conjunto de GALs que sirven para sincronizar el bus ISA con el bus local de la tarjeta.



**Figura 1.2** Diagrama funcional por bloques de la HDLC-256.

El inconveniente que afronta Cibertec, en relación con la tarjeta de expansión actual, va orientado principalmente a la desaparición progresiva del bus ISA en los equipos de cómputo modernos, y la necesidad de aumentar la capacidad de procesamiento.

Se decidió que es necesario rediseñar el módulo completo, utilizando componentes vigentes en el mercado y adaptándolo para poder comunicarse con la computadora por medio de bus PCI, que presenta características muy superiores a los estándares anteriores.

La problemática que se ataca en este proyecto es el diseño de la interfaz PCI de la tarjeta nueva, para lo cual se requiere la evaluación del protocolo y una investigación que determine los componentes adecuados para el circuito. Además, es importante realizar todas las adaptaciones necesarias, de tal forma que el módulo cumpla los requerimientos de “Plug and Play”, como se ha definido para todos los recursos PCI existentes en el mercado.

Para Cibertec es indispensable modernizar su decodificador de señalización SS7, de tal manera que sea compatible con los equipos de cómputo nuevos. Además, esto significa una ventaja en cuanto a adquisición de los componentes se refiere, puesto que muchos de los chips usados en la HDLC-256 han empezado a escasear en el mercado.

Las ventajas que ofrece el bus PCI en cuanto a velocidad de transmisión permite además aumentar el volumen de procesamiento de la tarjeta para que atienda una mayor cantidad de data links. En este sentido, la proyección es cuadruplicar la capacidad de la HDLC-256, lo que implica decodificar 32 data links en paralelo.

Es importante destacar que este es el primer paso en una serie de modificaciones que Cibertec planea realizar en sus equipos, en pos de la modernización. La obtención de resultados positivos a corto o mediano plazo son esenciales para impulsar los procesos posteriores, y toda la experiencia que se acumule en este proyecto permitirá a la empresa orientarse en cuanto la ruta que debe tomar para actualizar los sistemas.

### **1.3 Objetivos**

- 1.3.1 Revisar detalladamente el material que tiene la empresa en relación con los antecedentes del proyecto y la solución planteada para el problema.
- 1.3.2 Describir por completo el funcionamiento del sistema global y luego en particular, del módulo en el cual se centra el desarrollo del proyecto.
- 1.3.3 Determinar cuales serán los componentes principales de la tarjeta y su aporte al trabajo integral del módulo.
- 1.3.4 Ubicar los manuales y hojas de datos de los componentes, para el estudio detallado de sus características.
- 1.3.5 Identificar los diversos buses y conjuntos de señales que se va a necesitar para lograr un correcto desempeño del sistema.
- 1.3.6 Preparar un diagrama funcional (de bloques) donde se presente los componentes de la tarjeta y las principales señales que van a comunicarlos.
- 1.3.7 Describir las características fundamentales del estándar PCI, sus señales, ventajas respecto a buses antiguos y forma de aplicación al proyecto.
- 1.3.8 Explicar los diferentes ciclos de bus que se permiten en PCI y la forma en que se controla el tráfico de la información.
- 1.3.9 Determinar la metodología que sigue el estándar PCI para cumplir con los requerimientos "Plug and Play".
- 1.3.10 Identificar los diferentes componentes internos del microprocesador, y la función que cada uno de ellos realiza dentro del comportamiento del sistema.

- 1.3.11 Identificar las diferentes etapas que componen la estructura interna del circuito puente PCI/PCI.
- 1.3.12 Elaborar el diagrama de conexiones para las principales señales de los buses primario y secundario, con la debida justificación.
- 1.3.13 Organizar la estructura lógica del bus secundario para especificar las conexiones que no se hayan realizado hasta el momento.
- 1.3.14 Reconocer cuales son los organismos internacionales encargados de regular el uso del bus local PCI.
- 1.3.15 Obtener y presentar a la empresa los formularios requeridos para su inscripción legal como fabricante, y para la identificación del módulo, según la reglamentación correspondiente al uso de PCI.
- 1.3.16 Planificar la distribución de los espacios de memoria y puertos de I/O utilizables por los dispositivos del bus secundario.
- 1.3.17 Señalar los registros de configuración que se relacionan con "Plug and Play", dentro de los circuitos integrados que se va a utilizar.
- 1.3.18 Identificar dentro del puente, los registros que serán utilizados en el proyecto y la función relacionada a cada uno de ellos.
- 1.3.19 Preparar un informe final para el I.T.C.R. donde se muestre todo lo relacionado con el avance del proyecto y los logros alcanzados.
- 1.3.20 Hacer una serie de documentos informativos para los encargados de la empresa, que describan el progreso del trabajo y los aspectos básicos de la nueva teoría utilizada.
- 1.3.21 Planificar una serie de reuniones con los dueños de la empresa para estudiar los logros del proyecto y la posibilidad de extenderlo a niveles más avanzados.

## **CAPÍTULO 2**

### **ANTECEDENTES**

## **2.1 Estudio del problema a resolver**

En el análisis y estudio de la problemática que se resuelve con este proyecto, se incluye una descripción completa del sistema global, para luego centrar la atención en el trabajo del módulo específico dentro del todo. De esta forma se presentan los resultados de la investigación hecha en cuanto al funcionamiento de la HDLC-256, sus componentes individuales y la metodología de diseño utilizada por sus creadores en Cibertec.

### **2.1.1 El ITSS (International Traffic Security System)**

Este sistema consiste de una plataforma de arquitectura modular que detecta y previene patrones de tráfico telefónico irregular, para controlar a tiempo real, diferentes tipos de fraude, tales como aquellos desarrollados para traspasar a los Operadores de Larga Distancia.

El ITSS recolecta, analiza y archiva datos del tráfico telefónico, obteniendo estadísticas para mejorar la eficiencia en: Control de Fraude, Operación, Mantenimiento y Planeamiento.

Funciona como monitor centralizado de tráfico telefónico a tiempo real que proporciona un rango amplio de aplicaciones que pueden administrarse independientemente por los diferentes departamentos de la Administración Telefónica.

Los reportes que genera el sistema presentan estadísticas de tráfico a tiempo real o históricamente, asistiendo a las conciliaciones con los conectantes para mejorar la precisión y pérdidas de tiempo.

Por lo general, los clientes de Cibertec utilizan estos reportes como herramientas legales cuando se requiere un procedimiento jurídico contra los vendedores de servicio telefónico anormal.

### **2.1.1.1 Métodos usados para el fraude de tráfico telefónico**

#### **El "Call Back" o llamada revertida**

El avance tecnológico ha permitido que una gran cantidad de empresas radicadas en Estados Unidos, utilicen sin costo la infraestructura de las empresas de telecomunicaciones establecidas, para capturar el tráfico telefónico internacional saliente de un país, violando la soberanía de las naciones y el principio fundamental de las telecomunicaciones internacionales, el cual se da a la luz de convenios bilaterales.

El Callback es un servicio de llamada telefónica revertida utilizado por la telefonía internacional, generalmente debido a la presencia de tarifas telefónicas altas en nuestros países.

El mecanismo utilizado es el conocido como "intento de llamada no completada". Con tecnologías modernas en el país que recibe este tipo de llamadas, identifican al abonado que está llamando y le devuelven la llamada dándole tono de marcar como si estuviera en el otro país donde la tarifa es mucho más baja. Desde este momento las llamadas que haga el abonado local se registran como llamadas originadas en el exterior, generalmente en Estados Unidos, y por lo tanto son facturadas con tarifas y moneda de ese país.

Las Empresas Telefónicas Regionales (ROAs) pierden económicamente porque prestan sus instalaciones para conectar la llamada que no concluye y porque los usuarios de este servicio no pagan los impuestos que normalmente se cobran por el uso de la telefonía local al exterior. Las pérdidas por este concepto se estiman a nivel mundial en más de 1.200 millones de dólares anuales con una tendencia lógica de crecimiento que llevaría a las empresas de Callback a alcanzar el 100% del tráfico telefónico internacional, lo cual provocaría la quiebra de las empresas telefónicas establecidas.

Los esfuerzos de algunos países por solucionar el problema del Callback mediante la reducción de tarifas, sólo han producido una merma en sus propios ingresos, ya que el Callback sigue utilizando sin costo la infraestructura, por lo que siempre podrá ofrecer tarifas más atractivas.

Los intentos por controlar el Callback se han circunscrito a la esfera legal con muy poca efectividad, dado que las empresas de Callback, amparadas en la tecnología, se mantienen al margen de toda regulación. Un ejemplo de este esfuerzo en el campo legal son las resoluciones de la OEA y de la UIT, adjuntas.

El nuevo desarrollo tecnológico de Cibertec cubre las deficiencias técnicas de los equipos de las centrales telefónicas internacionales, permitiendo así el ordenamiento y la reglamentación de los servicios, hasta ahora clandestinos, del Callback.

### **ISR o desvío no autorizado del tráfico internacional**

El fraude de tráfico telefónico internacional (ISR) es una metodología sistemática y organizada usada por “Carriers” y Operadores de tráfico internacional para eliminar o reducir el pago acordado de tasas contables con los operadores ROAs de países externos a EEUU. Este tipo de fraude, también denominado “reventa de minutos” no utiliza las centrales internacionales supervisadas.

Consiste en aquellas llamadas internacionales entrantes o salientes efectuadas a través de enlaces satelitales y que no utilizan la infraestructura de las empresas operadoras.

Para facilitar este servicio, personas naturales o jurídicas locales deben contar con líneas telefónicas facilitadas por la Administradora de la RED. Usan una antena parabólica equipada con todos los sistemas de transmisión, frecuencias de transmisión y satélites donde arriendan segmentos a compañías extranjeras.

Algunas de estas antenas tienen permisos de operación para el transporte de datos y de Internet, otras son totalmente clandestinas, ninguna está autorizada a transportar voz.

Control de Fraude: El sistema ITSS de Cibertec sirve como un instrumento efectivo para controlar y auditar las pérdidas de ingresos que las Compañías Telefónicas sufren de su actividad de larga distancia.

Cibertec ha desarrollado el ITSS para resolver el fraude internacional bajo una estrategia orientada a dos conjuntos diferentes de tráfico fraudulento: llamadas que cruzan la central Internacional (controladas por el sistema ITS-I), y tráfico internacional cuyas llamadas no utilizan la Central Internacional (controlado por el sistema ITS-L).

El aseguramiento de ingresos no puede ser explicado totalmente sin la presentación de todos los escenarios que perjudican a los Operadores de Telecomunicaciones. Seguidamente se presentan dichos escenarios y la solución de Cibertec para cada uno de los problemas que se indican.

#### **2.1.1.2 Procedimientos de llamadas alternas ilegales**

Es importante conocer cual es la metodología utilizada por las empresas dedicadas al fraude telefónico internacional para realizar las llamadas anormales, ya sea usando llamada revertida o con desvío del tráfico.

##### **Operación del "CallBack" tradicional**

La llamada revertida tradicional es el método más fácil de convertir tráfico saliente de cualquier país a tráfico entrante a ese país. La transacción no involucra algún Conectante explícitamente y es difícil detectarlo por el operador afectado. Lo económico detrás de esta actividad se basa en el costo efectivo de las tasas contables a los países de destino.

Por ejemplo, La tasa de una llamada originada en los Estados Unidos USA contra la tasa contable existente y la tarifa de la administración afectada.

El "CallBack" evita el cobro de la renta ganada justamente de las llamadas internacionales por las Administraciones Telefónicas en sus propio países reduciendo así su renta neta.

Múltiples empresas pequeñas y subsidiarias de Conectantes Internacionales y otras compañías multinacionales explotan el "CallBack". Dichas compañías no tienen ningún acuerdo ni presencia en el país afectado.

Hay tres tipos principales de "CallBack". El primero da al usuario un tono directo de marcación de los Estados Unidos por lo que ellos pueden marcar desde su propio teléfono para poner una llamada a cualquier parte del mundo. Este método es llamado el "CallBack" tradicional. El segundo no requiere de la llamada de disparo, es llamado "supresión de respuesta", pues la marcación se recoge antes de enviar la señal de contestación. El tercer método, "otras formas de disparo" agrupan todos los otros métodos que no contemplan actividad DTMF en el enlace (sin señalización secundaria).

Forma de operar:

- El operador de "CallBack" contacta al usuario local y le ofrece llamadas m[as baratas a cualquier parte del mundo. El usuario acepta la propuesta del operador de las llamadas revertidas y este le asigna un número telefónico dedicado, código de acceso y/o un número de identificación personal o contraseña (PIN).
- El usuario local llama a su número asignado deja timbrar dos veces y cuelga el teléfono.

- Automáticamente el operador de "CallBack" llama de vuelta al número del abonado local.
- Después de la invitación a marcar el usuario digita su número de destino.
- La llamada es conectada.

Soluciones:

Hay varias maneras de atacar este problema. La primera solución es levantar estadística de los destinos de las llamadas salientes sin respuesta. Entonces cuando el usuario intenta hacer la llamada de disparo esta se bloquearía y subsecuentemente también la llamada revertida. Esta solución no se recomienda porque el operador de "CallBack" cambia constantemente el número designado para el disparo. También en ocasiones esta técnica atenta contra llamadas inocentes.

Otra manera utilizada por las Administraciones Telefónicas para eliminar este problema es manteniendo un registro de los números locales que generan cualquier llamada salientes cortas sin ser atendida, luego se bloquea la llamada entrante siguiente a ese número. Los métodos mencionados son semejantes, levantar este tipo de listas requiere de un tiempo mayor que el que requiere el operador de "CallBack" para cambiar el número de disparo.

La solución para el "CallBack" de Cibertec es directa y opera en tiempo real. Esta aplicación permite la detección y bloqueo de diferentes tipos de llamadas "CallBack".

El ITSS registra toda llamada revertida pues es detectada por la vigilancia de marcación DTMF irregular.

El ITSS detecta la señalización secundaria saliente DTMF (Dual Tone Multi-Frequency) en las llamadas internacionales entrantes.

No hay razón otra que el "CallBack" para que una llamada entrante muestre actividad DTMF en su trayectoria saliente (los servicios que utilizan marcación DTMF lo hacen en el otro sentido, pues por ejemplo, en ningún caso un banco llama al usuario para ofrecerle consultar su cuenta corriente).

Además dispone de aptitudes de avanzada, que le permite la identificación de los tonos con su subsiguiente registro y bloqueo, además de registrar el número de usuario de "CallBack" y la duración y destino de las llamadas en el caso que estas se deseen facturar.

### **Supresión de respuesta "CallBack"**

En algunas centrales el audio se conecta inmediatamente después de recibida la notificación de que la llamada ha sido contestada por la central de origen. Esto permite al Operador de "CallBack" utilizar la misma llamada de disparo para registrar el número de destino de manera que cuando la llamada de retorno regresa ya viene conectada y no hay algún tono DTMF en la llamada entrante, desarrollo posterior de los operadores con el fin de evitar control sobre este tipo de llamadas ilegales.

Puesto que las centrales permiten el timbrado por un tiempo entre 45-90 segundos, en una llamada internacional hay suficiente tiempo para transmitir la información de destino para ser utilizada por la llamada revertida subsiguiente. Algunas veces este proceso se ejecuta mediante cajas negras instaladas donde los usuarios para este propósito.

Forma de operar:

- En la supresión de respuesta el Operador de "CallBack" solicita al usuario marcar en la llamada saliente el número de destino, puesto que la llamada no es contestada, no hay cargos pero la información del destino de la llamada ha sido enviada al operador de "CallBack".

- El Operador de "CallBack" establece las llamadas de destino y la originaria.
- Luego la llamada es conectada a ambos puntos: Origen y Destino.

Solución:

Existen dos soluciones principales para este tipo de fraude. Es posible programar algunas centrales para que el audio se conecte solamente hasta haber recibido el mensaje de respuesta. Pero no todas las centrales tienen la capacidad de realizar esta operación, bajo estas circunstancias el ITSS de Cibertec es igualmente efectivo. El sistema detecta la actividad DTMF antes de que se haya recibido la señal de respuesta por lo que desconecta el audio o modifica aleatoriamente los tonos DTMF anteriores a que la llamada haya sido contestada. En el caso de marcación accidental la conexión se restablece tan pronto la señal de contestación es recibida, en cualquiera de los casos, los dígitos colectados se reportarán al ordenador supervisor.

### **Otros métodos de disparo**

Existen otros métodos de activar las llamadas "CallBack" que no se realizan por una llamada saliente, sino que a través de redes de datos públicas o privadas, enlaces directos, centrales locales, etc. Este método transfiere toda la información al Operador de "CallBack" de manera que cuando la llamada revertida se hace, lo único que queda es que el usuario levante el aparato telefónico y esperar que la parte llamada conteste.

El procedimiento es el siguiente:

- El operador de "CallBack" le da al usuario un método alternativo de enviar la información de destino de la llamada. Este método puede ser reconocimiento de voz, correo electrónico, enlace de datos X-25, llamada itineraria, para disparar la llamada revertida.

- El Operador establece las llamadas de destino y de origen.
- La llamada es conectada.

Solución:

Este método de llamada revertida es el menos amigable para el usuario. El usuario de "CallBack" debe tener un medio diferente a la telefonía estándar, para poder comunicarse mediante este método con el operador de "CallBack".

Estos otros métodos no son muy populares pues contienen un alto riesgo para el Operador de CallBack puesto que la parte llamada puede contestar antes de que el usuario local conteste la llamada, con las consecuentes pérdidas.

El ITSS es capaz de detectar el progreso de la llamada secundaria y por lo tanto tomar la acción de bloqueo seleccionada.

La solución esta incorporada en los circuitos internacionales. El ITSS tiene la capacidad de detectar aquellas llamadas que han sido establecidas después del primer paso para que la llamada sea completada (al usuario del "CallBack").

Utilizando en mismo método de detección las llamadas revertidas asistidas por operador son también controladas.

El procedimiento completo es el siguiente:

- La llamada entrante es detectada, registrándose el número llamado.
- Se detecta el progreso de establecimiento de la llamada.
- La llamada es bloqueada por alguno de los métodos previamente seleccionado.

- El usuario no puede comunicarse vía "CallBack" y eventualmente lo hace por los medios normales.

Específicamente, el sistema de Cibertec es capaz de detectar y bloquear las formas de "CallBack" basadas en los siguientes principios:

- a. Asistidas por operadora
- b. Reconocimiento de voz
- c. Conexión X 25
- d. Conexión por MODEM
- e. Conexión por Internet
- f. Marcación abreviada
- g. Líneas dedicadas

El sistema es capaz de reconocer aquellas llamadas con marcación secundaria y actuar efectivamente contra ellas inhibiendo su completación.

La acción se ejecuta en tiempo real y en una forma transparente a la señalización normal entre los extremos de los enlaces internacionales. Esta acción no afecta la marcación secundaria llevada a cabo sobre llamadas entrantes o salientes utilizadas para acceder sistemas interactivos de respuesta de voz (IVR). Tampoco afecta la marcación sobre centrales telefónicas privadas (PABX) ni a uso normal de tarjetas de llamada, etc.

## **"Bypass" (Tercer país)**

El término genérico "Bypass" que en ocasiones se utiliza para definir el fraude "ISR" aquí será utilizado para señalar los métodos ilegales que tienen que ver con llamadas puestas a través de un determinado país con destino final a un tercer país.

Generalmente hay acuerdos con las Administraciones Conectantes para permitir a los viajeros llamar a su casa mediante algunas tarjetas de llamada (Country Direct Service). Estos acuerdos no incluyen el derecho de llamar a terceros países pero esto se hace frecuentemente.

Mediante el uso de tarjetas de llamada los usuarios utilizando números especiales tienen acceso a otros destinos, reportando a la Administración Telefónica Local como llamadas al destino autorizado, generalmente los Estados Unidos de América.

Esta técnica de fraude se ha especializado de tal manera que los proveedores instalan "cajas negras" en los usuarios para hacer el proceso totalmente automático.

### **Solución:**

El mecanismo de llamada manual o automático, le pide al usuario marcar el número de destino final una vez completada la llamada original. El ITSS, instalado en el Puerto Internacional, monitorea las llamadas salientes por su actividad DTMF.

La marcación DTMF se registra y es analizada. Si la marcación DTMF indica claramente un destino final a un tercer país (por ejemplo marcación que inicie con los dígitos de acceso internacional de los Estados Unidos 011, Marcación Internacional sobre una llamada contestada), entonces la llamada podrá ser bloqueada.

Adicional a esto podrán tomarse acciones legales contra el Operador de tarjetas de llamada (registradas o no), puesto que el número de destino está debidamente registrado por el ITSS de Cibertec.

El equipo se instala in los enlaces internacionales. El sistema es capaz de detectar y registrar la actividad DTMF en las llamadas salientes completadas. La marcación DTMF es analizada por el sistema.

Si el análisis muestra que se trata de consulta de tarjetas de crédito, "telebanking", máquinas de respuesta automática remota y un sin número de servicios que se brindan por medio de las redes internacionales, el sistema sólo actuará si se trata de marcación a terceros países.

Una vez realizado el análisis por el sistema dependiendo de la política de la Administración Telefónica Local, se bloqueará o no este tipo de llamadas o bien se tomará las acciones pertinentes.

Se dispone de análisis fuera de línea y en la misma red, para determinar el uso irregular de las tarjetas de llamada y su comportamiento, con el fin de realimentar periódicamente el sistema.

### **Fraude ISR (International Simple Resale)**

El ITSS de Cibertec identifica el fraude ISR en tres grupos:

- a. ISR-I Detección del aterrizaje de tráfico ilegal y conectado dentro de la red de la Administración Local Mayor.
- b. El método "Retargeting" o "Call Hijacking" que involucra a competidores locales con interconexión a la red local Mayor. El competidor le ofrece mejores tarifas de interconexión a los "carriers" internacionales y recoge el tráfico internacional directamente en sus centrales. Entonces el competidor cambia la señalización de la llamada a su destino final, usando un enlace de datos marginal, y la envía a la red pública mayor como si fuera una llamada local.

- c. ISR-O Detección de tráfico internacional saliente que no pasa por los circuitos y puertos reglamentarios. Hay tres rutas alternativas básicas: Redes de datos (incluyendo Internet), Tele-puertos y Líneas Directas.

### **2.1.1.3 Arquitectura del ITSS**

El Sistema de Cibertec se constituye de dos puntos importantes: El RTG (Generador Remoto de Tráfico) y los puntos TMS.

Todas las llamadas generadas por el RTG se identifican en el punto de captura local. En este punto de detección las llamadas se registran con la señalización local correspondiente. Si la llamada que llega al TMS no es identificada como internacional en su señalización, la llamada deberá tener un número local de origen. Este número local pertenecerá a una compañía o individuo que está haciendo aterrizaje de tráfico ilegal.

Dado que el sistema esta totalmente sincronizado en fecha y hora con una precisión de más o menos 50 milésimos de segundo y que todos los puntos del sistema, tanto el RTG como los TMS envían sus reportes de al Centro de Procesamiento y Control (CPS). Usualmente esta es evidencia suficiente para que el Operador Local pueda tomar acciones legales contra ellos.

Por otro lado, los TMS que se colocan en las centrales internacionales se encargan de revisar la señalización de las llamadas que cruzan por las centrales internacionales, sin intervenir en la información de voz. A partir del estudio de esta señalización (la cual está basada en estándares y acuerdos internacionales) se puede determinar cuando se está tramitando una llamada fraudulenta a través de la infraestructura de la compañía afectada. Además, se puede desviar o cortar las llamadas que se detecten con destino a un tercer país. Así se cubre tanto el fraude a través de las centrales, como el aterrizaje no legal de llamadas.

## **El Generador Remoto de Tráfico o RTG**

El "RTG" es una red de equipos instalados remotamente en diferentes países, preparado para recibir órdenes las 24 horas, por medio de correos a través de Internet, para generar llamadas. También hay un software de comunicación entre los "TMS" y la red "RTG" en forma continua con sincronización de fecha y hora en el tiempo.

El Generador de Tráfico Remoto consiste de una serie de aplicaciones mediante las cuales las unidades de hardware proveen cobertura total a través de todos los conectantes a los diferentes países. Una parte intrincada del RTG/TMSS permite al equipo de Cibertec monitorear el tráfico fraudulento que es generado a la red telefónica del cliente.

Las principales partes del sistema son:

- Configuración remota automática y levantamiento de blancos del sistema.
- Generador de Tráfico.
- Acceso a Archivos con el Tráfico Generado.
- Perfiles del cliente y derechos de acceso.

La función principal del RTG es efectuar llamadas al país de destino seleccionado por medio de los diferentes conectantes y a diferentes específicos números telefónicos. Siguiendo un itinerario, el RTG responde con la cantidad de llamadas por hora y para cada número definido. Puede configurarse vía correo electrónico en los siguientes aspectos: la definición de los números blanco, un itinerario para cada uno de ellos y el conectante o tarjeta de llamada a ser utilizado. Para incluir nuevos conectantes al sistema, la información deberá ser enviada vía "e-mail" al operador del RTG.

El RTG lleva consigo una importante misión para el correcto funcionamiento del sistema ITSS. Por medio del RTG que genera las llamadas al país de destino donde los TMS rastrean cada llamada, se determina fácilmente y en forma automática las rutas y portadoras utilizadas.

### **Puntos TMS**

Los puntos locales se encargan de atender las llamadas generadas desde el RTG y verificar la ruta seguida. Además revisan constantemente el tráfico en las centrales telefónicas internacionales para localizar procedimientos irregulares en la señalización.

Los TMS están constituidos por una serie de equipos que trabajan en conjunto para capturar, analizar, ordenar y almacenar la información descriptiva de las llamadas telefónicas que pasan por ellos.

Los elementos de un TMS son básicamente:

- Un juego de tarjetas CPU encargadas de tomar los datos de señalización de las líneas sin interrumpir la comunicación normal;
- Un Supervisor, que es una computadora en la cual se procesa la información recolectada en las tarjetas;
- Un Servidor de Tráfico, que toma los datos del supervisor y los registra en una base de datos.

El TMS tiene capacidades de comunicación con el resto del equipo de Cibertec, y es capaz de transmitir sus datos a un sitio web, desde donde los usuarios pueden realizar consultas. Además, existe un Servidor de Reportes que constantemente está siendo actualizado desde las bases de datos.

### **2.1.2 Funcionamiento de la HDLC-256 dentro del ITSS**

La tarjeta HDLC es una pequeña parte dentro de toda la arquitectura que compone el ITSS. Se localiza específicamente en las unidades TMS, para comunicar la información capturada por las tarjetas CPU a la computadora Supervisor.

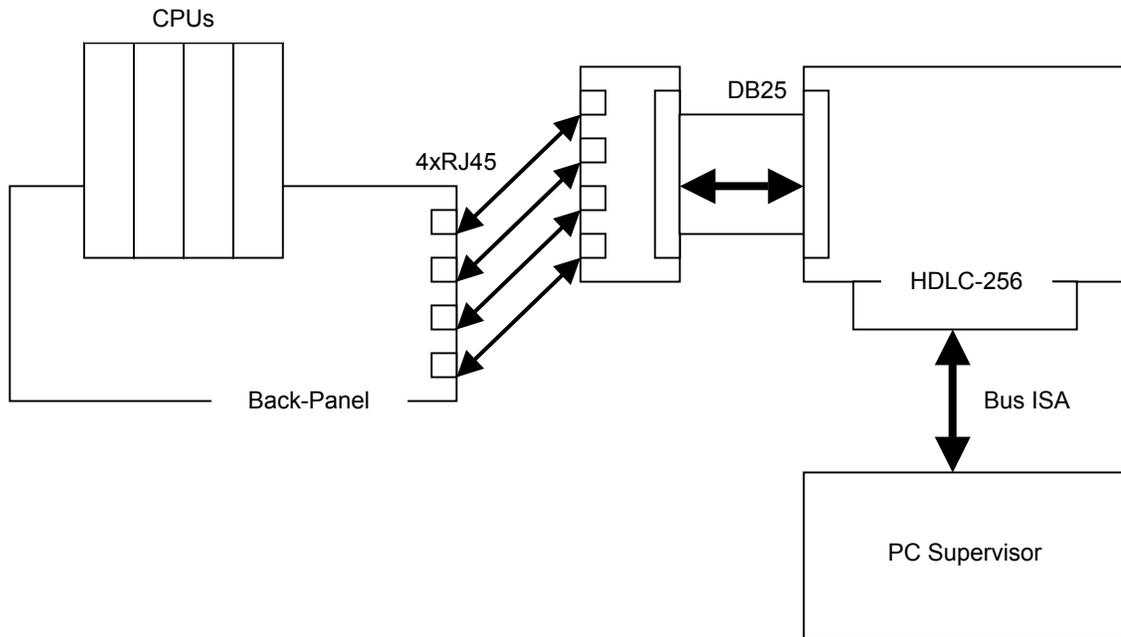
Este módulo es una tarjeta de expansión diseñada para trabajar en el estándar de PC. Su misión es interactuar con el software instalado en el Supervisor para que la señalización que se está vigilando pueda ser interpretada adecuadamente, y según el contenido o estructura que sigue, se presente al usuario el estado de las líneas.

Con la información que se recolecte en el Supervisor es que se forman las bases de datos en el Servidor de Tráfico, y se producen los reportes donde aparecen los procesos anormales detectados. Sin la HDLC no es posible recopilar los datos que se generan en las tarjetas CPU, y por lo tanto el sistema no poseería capacidad de almacenamiento, presentación de los datos por pantalla, ni generación de reportes completos. Este módulo es esencial ya que es el primer paso en la comunicación del equipo diseñado por Cibertec con el sistema de cómputo.

En la Figura 2.1 se muestra la interconexión de los equipos de monitoreo de protocolo con la HDLC-256 y a través de ella con el computador Supervisor. La información que recibe a través del cable DB25 está en formato ST-Bus, y contiene datos codificados usando el protocolo HDLC, estandarizado por la CCITT.

Como puede notarse, de las tarjetas CPU sale un sistema de cableado basado en RJ45. Cada tarjeta envía su información a través de estas líneas. Existe una pequeña interfaz entre los CPUs del Back-panel y la HDLC, en la cual se unifica el cableado de las cuatro tarjetas en un DB25, que se conecta en el puerto correspondiente de la HDLC-256.

El módulo tiene además un grupo de dip-switch que el usuario puede utilizar para hacer ligeros cambios en la dirección de I/O que se va a usar, mientras que el IRQ requerido por la tarjeta se controla usando jumpers. Estas opciones deben ser configuradas correctamente por el encargado de la instalación, para que no exista conflicto con otras tarjetas o recursos del sistema.



**Figura 2.1** Interconexión de la HDLC-256 con su equipo periférico.

### 2.1.2.1 Formato de comunicación serie ST-Bus

En el lado del puerto DB25, la HDLC tiene que manejar transferencias de información por medio de un bus serial. Este bus ha sido diseñado especialmente para equipos de telecomunicaciones por la compañía Mitel, y se denomina "Serial Telecom Bus" o ST-Bus.

La arquitectura de este bus puede ser usada tanto para digitalización de voz controlada por software, como para conmutación de datos y comunicación entre procesadores. Los niveles lógicos son cero si la tensión está por debajo de 0.4 V, y uno si está sobre 2.4 V.

Las líneas de información (*streams*) del ST-Bus operan a una velocidad de 2048 Kbit/s que transportan una línea de cuadros (*frames*) de 125  $\mu$ s, cada uno de los cuales contienen 32 canales de 8 bits. Un canal ocupa lo que se denomina una ventana temporal (*time slot*).

Aquí se presenta el concepto denominado TDM (Multiplexación por división en el tiempo) en el cual varios canales se transmiten utilizando un espacio determinado en el lapso de tiempo asignado a un cuadro. Para esto es necesario, además de la señal de reloj, un pulso de sincronización que indique el momento en que inicia cada cuadro.

Los codificadores y decodificadores deben estar en capacidad de sincronizar correctamente su funcionamiento, para que los datos correspondientes a un canal dado viajen en el lapso temporal que les es asignado.

El ST-Bus sirve para transferir cierta cantidad de canales de información lógica a través de un único cable. Así se logra minimizar el espacio físico de los circuitos, el costo de la red de comunicación, y la cantidad de pines requeridos en los chips para las transferencias.

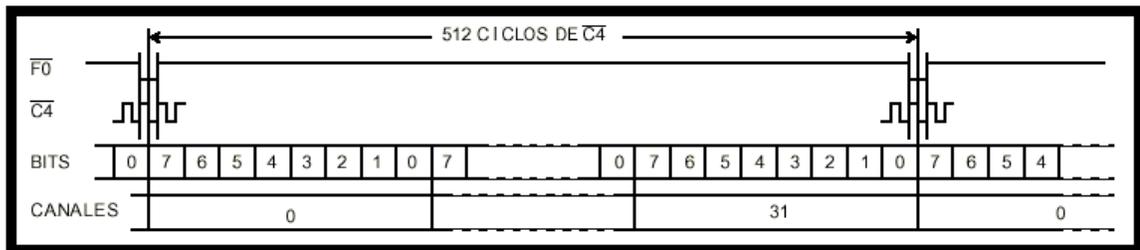
La interfaz ST-Bus entre dos circuitos consiste en dos señales de control y tantas líneas de entrada o salida (*streams*) como sea necesario.

Dada la velocidad de transferencia estándar (más de 2Mbit/s) es posible transmitir aproximadamente 8000 cuadros, cada uno con un contenido de 256 bits. Dado que los canales usan 8 bits por cuadro, esto implica que trabajan a una velocidad de transferencia de 64Kbits/s. Los canales están referenciados al inicio del cuadro, y están numerados del 0 al 31.

Por lo general, la información de un sistema determinado es asignada a un canal del bus, mientras que los otros canales quedan disponibles para el uso.

Si la sincronización se da correctamente, no existe traslape entre los datos enviados por un canal y otro, y es posible que varios dispositivos se comuniquen usando el mismo ST-Bus pero diferentes canales para el trasiego de información.

La Figura 2.2 muestra la forma en que se disponen los datos de acuerdo con las líneas de sincronización: C4\ y F0\, donde la primera corresponde al reloj, y la segunda al pulso de inicio de cuadro.



**Figura 2.2** Alineación del ST-Bus usando F0\ y C4\.

Internamente, los dispositivos que utilizan ST-Bus deben poseer un arreglo de contadores que indiquen en cada momento el canal que se está transmitiendo, y el bit específico dentro de este canal. La función de F0\ es retornar estos contadores a cero, para que se inicie un nuevo cuadro.

En algunos sistemas, la señal de sincronización se extiende durante todo el primer time slot, lo que conlleva a que en estos casos no es posible utilizar el canal cero. Por esta razón, se acostumbra dejar este espacio libre de datos, con el objetivo de mantener lo más que se pueda un estándar en cuanto al uso de este bus.

Queda abierto al usuario del bus, el tipo de información que será transportada por él. El único elemento limitador en esta interfaz es su velocidad de transmisión, ya que como está diseñado para telecomunicaciones, el ancho de banda que posee no es adecuado para aplicaciones como envío de sonido de alta calidad, o la comunicación de datos desde y hacia dispositivos de memoria en un sistema de cómputo.

### 2.1.2.2 Protocolo de comunicación HDLC

Una vez que se examinó las características del bus serial por el cual se comunican datos entre las tarjetas CPUs y la HDLC-256, es necesario profundizar un poco el tipo de información que viaja por los canales del ST-Bus.

Desde hace algunos años, en todo el mundo se ha dado un intento para estandarizar los formatos de comunicación usados en telecomunicaciones digitales. Específicamente en el área de la codificación correspondiente a señalización telefónica, se utiliza mucho el Protocolo HDLC, definido en las recomendaciones X.25 (Nivel 2) de la CCITT. Precisamente de aquí toma su nombre el módulo en cuestión.

HDLC significa “High Level Data Link” o sea, “Enlace de Datos de Alto Nivel. Este protocolo es una estructura orientada al envío de paquetes de bits, utilizado en los sistemas telefónicos para comunicar la señalización de las llamadas.

El formato utilizado permite un chequeo de los mensajes para localizar errores. Cada bloque es separado por un código especial llamado bandera (*flag*). De esta manera se le dice al receptor cuando empieza un bloque de datos y cuando finaliza.

Una transmisión se realiza por medio de cuadros (*frames*). Cada cuadro inicia con una bandera, continúa con un mínimo de dos bytes de información, dos bytes para detección de errores. Por último, se envía una bandera de cierre.

Es importante destacar que el término cuadro o frame se utiliza para definir dos cosas distintas, dependiendo si se está hablando sobre protocolo HDLC, o ST-Bus. En el primero corresponde al paquete completo de datos recibidos, el cual está delimitado por sus banderas. En ST-Bus, es una trama de 32 canales de 8 bits, que tiene una duración de 125  $\mu$ s, como se examinó anteriormente.

En la Tabla 2.1 se muestra la estructura que debe llevar el bloque de información completo en protocolo HDLC.

**Tabla 2.1** Formato de un cuadro HDLC.

FLAG	DATA FIELD	FCS	FLAG
1 Byte	n Bytes (n≥2)	2 Bytes	1 Byte

### **Banderas**

La bandera es un patrón único de 8 bits (01111110) que define los límites de un cuadro.

La sección transmisora genera las banderas y se las inserta automáticamente a la información que se transmite. La sección receptora está constantemente buscando este patrón, por medio de un proceso de chequeo bit por bit, con el fin de establecer la sincronización de los datos.

### **Datos**

El campo de datos está referido a la Dirección, Control o Información que se desea comunicar. Para que sea válido debe contener un mínimo de 16 bits.

### **Frame Check Sequence (FCS)**

Los 16 bits que siguen al campo de datos se denominan Secuencia de Chequeo de Cuadro. Esta palabra se forma por medio de un cálculo numérico que se realiza conforme los datos van ingresando en el cuadro. Para ello se utiliza una norma estandarizada de cálculo, definida por el polinomio:  $G(x)=x^{16} +x^{12} +x^5 +1$ .

El transmisor calcula este valor y lo agrega al final del cuadro. Por su parte, el receptor debe realizar el cálculo también y comparar su resultado con lo que reciba.

En caso de que los valores sean distintos, el circuito receptor debe tener un mecanismo para comunicarle al transmisor que hubo un error y hacer una petición para el reenvío del mensaje.

### **Inserción de ceros**

Durante el proceso de envío de datos, es posible que entre los caracteres exista una emulación del valor de la bandera. Para evitar que esta situación conlleve a una interpretación errónea del mensaje, el transmisor se debe encargar de colocar bits de valor cero luego de que se detecte el envío consecutivo de cinco unos lógicos.

Este bit agregado no forma parte del mensaje, pero es necesario para evitar que en el receptor aparezca el símbolo de la bandera. Por su parte, el receptor debe estar capacitado para eliminar estos ceros y recuperar el mensaje correcto.

### **Aborte de mensajes**

Si el emisor decide por alguna circunstancia abortar el mensaje, se aplica a la línea un grupo de ocho unos consecutivos como mínimo. El receptor, al captar esta secuencia procede a vaciar sus registros de almacenamiento y regresa al estado de espera.

Esta convención se utiliza de forma tal que en caso de un error en la línea, un estado permanente de circuito abierto en la entrada conlleve a aborte del mensaje.

### **Llenado del tiempo libre**

Cuando no se está transmitiendo mensajes por la línea, el transmisor tiene varias opciones.

La primera de estas opciones se denomina estado Idle, y se logra al colocar una cadena permanente de unos en la línea. El receptor debe tener la capacidad de detectar esta secuencia y ubicarse en estado de espera. Para salir de este estado, basta con escribir un dato en el registro Tx del transmisor, o bien recibir una bandera en la entrada del receptor.

El segundo método para llenar el espacio libre es un envío constante de banderas, que no coloca a los circuitos en espera, sino que únicamente los mantiene alertas para el inicio de la comunicación. Esta opción se denomina Interframe.

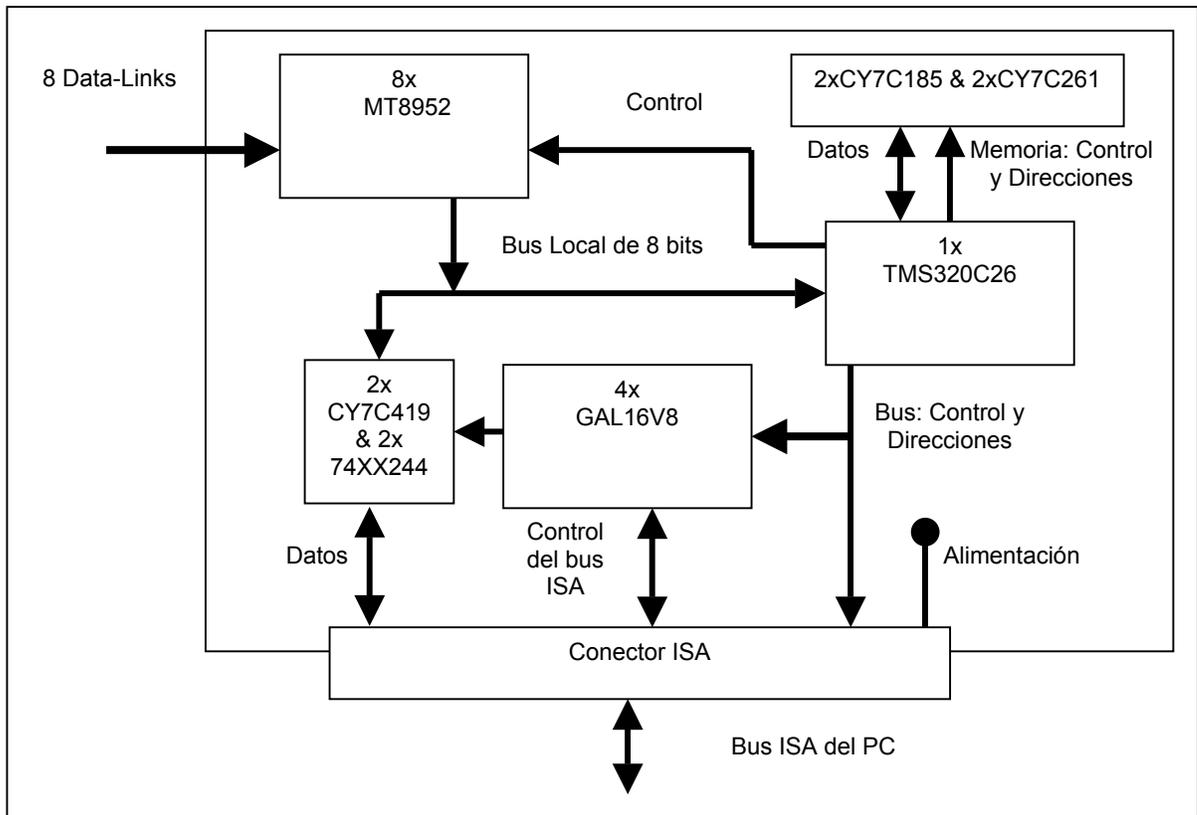
La última forma se llama “Go-Ahead”, y se logra enviando un dato 7F<sub>HEX</sub> repetidas veces. Esto coloca los módulos en un estado de semi-espera que se interrumpe en el momento de aparición de una bandera.

### **2.1.2.3 Descripción por bloques de la HDLC-256**

El módulo HDLC-256 consta de varias secciones bien definidas. En la Figura 2.3 se muestran las distintas etapas que constituyen el circuito de la tarjeta, y los diferentes enlaces de información entre ellas.

Los controladores de protocolo HDLC corresponden a un conjunto de 8 chips MT8952B, los cuales se encargan de recibir la información desde los ST-Bus y establecer la comunicación con el transmisor, siguiendo las pautas dadas para HDLC. Estos chips capturan la información desde los ST-Bus, realizan las rutinas relacionadas con la identificación de los datos empaquetados en HDLC y presentan la información recuperada en un bus básico de 8 bits.

En teoría, también se puede establecer el proceso inverso, de forma que un dato del bus se captura y se arregla para ser enviado por los ST-Bus siguiendo las convenciones de HDLC, sin embargo esto no está implementado aún para el módulo.



**Figura 2.3** Estructura de la HDLC-256, con sus principales componentes.

El microprocesador trabaja como gobernador del sistema, haciendo un trabajo de sincronización entre los ocho controladores, revisando los mensajes que no son importantes, y colocando los que sí interesan en la cola principal. Este procesador utiliza un banco de memoria ROM para guardar su programa y un banco de SRAM para las variables.

En la memoria FIFO se almacenan los datos que ya están listos para que el computador los reciba. Esto se da por medio de una interrupción (IRQ) cuando se detecta la señal "Cola Medio Llena". Esto permite al CPU tomar lectura de los datos y continuar hasta que se active la señal de Cola Vacía.

Al igual que los 8952, este arreglo de FIFO está diseñado para permitir el envío de mensajes y datos desde el PC hacia el procesador local, función que no es utilizada en las aplicaciones hasta el momento.

El bloque de GALs permite ejercer un control auxiliar o sincronización entre las señales del bus local y el bus ISA de la computadora.

Todos las etapas de la tarjeta trabajan en conjunto, pero el procesador es quien se encarga de coordinar todos los procesos. De esta forma, mientras los 8952 están recibiendo información de los canales HDLC, el microprocesador puede estar dando la orden de vaciar la FIFO al PC. Cuando existe un evento importante, se activa una interrupción del TMS320, y este se encarga de realizar la rutina correspondiente.

Este trabajo no afecta en nada los procesos primarios en la computadora, y así se logra reducir la carga del CPU, entregándole únicamente los datos más importantes de la señalización SS7, sin los bloques de información auxiliares que se usan en el protocolo HDLC.

#### **2.1.2.4 Descripción del chip MT8952B**

La función de este chip es tomar estructuras orientadas a protocolos de bit y la formatea en protocolos de paquetes conmutados, como se define en las recomendaciones X.25 (Nivel 2) de la CCITT.

La tarjeta HDLC-256 utiliza 8 chips 8952B de Mitel. Este se identifica como un controlador de protocolo HDLC compatible con un puerto para ST-Bus y con un puerto paralelo de 8 bits.

Los datos en el puerto de ST-Bus viajan codificados en HDLC, mientras que en el puerto de 8 bits se tiene la información sin paquetes auxiliares.

Esto puede darse en ambas direcciones: los datos entran en 8 bits paralelos y son empaquetados en protocolo HDLC y enviados por un canal del ST-Bus; o pueden provenir del ST-Bus para ser extraídos del protocolo y puestos en el bus.

Quiere decir, que el mismo chip trabaja como receptor o como transmisor, según la configuración programada en sus registros internos.

El 8952 está capacitado para ofrecer transparencia en la transmisión de datos en el puerto serial, por medio de la inserción o borrado de ceros, el manejo de las banderas y el código de error FCS. Provee información referente al estado de los canales y las secuencias de aborto.

Puede generar y detectar secuencias Go-Ahead y reconocer direcciones de bloqueo en el primer byte recibido. Además existe la opción de desactivar las funciones de protocolo, y proveer acceso inmediato y transparente al bus serie a través del puerto paralelo.

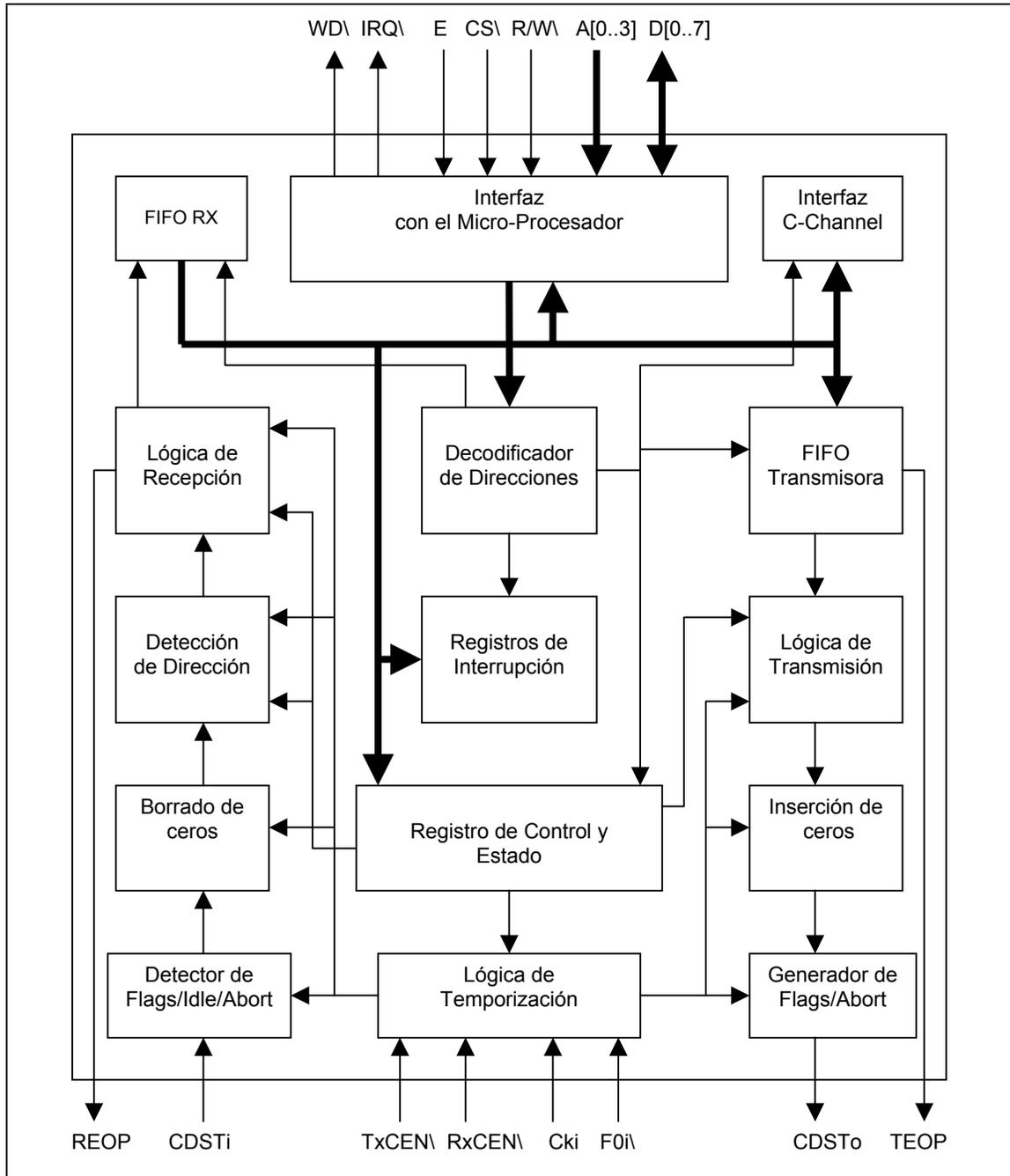
La estructura de este chip se muestra en la Figura 2.4. Allí se presentan las diferentes etapas encargadas de procesar la información.

El controlador de HDLC tiene dos puertos. El puerto serie transmite y recibe paquetes de información formateada y el puerto paralelo provee una interfaz con microprocesador, para acceder los registros desde fuera.

El puerto serie se puede configurar para operar en dos modos, según el valor del bit IC en el Timing Control Register. Puede transmitir / recibir paquetes en el canal seleccionado del ST-Bus, o puede, usando TxCEN\ y RxCEN\, transmitir / recibir paquetes a una velocidad igual a CKi.

Las salidas TEOP y REOP son señales que indican el fin de los paquetes transmitidos y recibidos, en cada caso.

Ambos se identifican como un pulso pequeño en alto que se da durante el último bit de la bandera de cierre o la secuencia de abortar.



**Figura 2.4** Descripción funcional por bloques del MT8952B.

Existen dos modos de temporización, que afectan la forma en que trabaja el puerto serie cuando está configurado para ST-Bus. Estos modos se conocen como interno y externo, y no modifican en nada al puerto paralelo.

Cuando el ST-Bus funciona en modo interno los paquetes de datos se envían y reciben usando F0i\ y C2i/C4i. Esta es la forma directa de acoplar el chip a ST-Bus. El canal 1 está reservado para llevar información de control (C-Channel).

La frecuencia del oscilador aplicado a la entrada CKI puede ser ajustada con el bit BRCK del Timing Control Register para mantener el ST-Bus a 2048 Kbps. Si BRCK es cero, el CKI debe ser de 4,096 MHz, y si es uno el CKI debe ser de 2,048 MHz.

En modo interno las entradas TxCEN\ y RxCEN\ son ignoradas.

La otra forma de trabajo del bus serial como ST-Bus se denomina externa, y es controlada usando TxCEN\ y RxCEN\. En este caso las transmisiones y recepciones son independientes, y los paquetes son movidos serialmente a una velocidad igual a CKI. La salida se envía en el flanco positivo del reloj, mientras que las entradas se leen en el flanco negativo. Los controles son independientes y asíncronos y tienen efecto sólo hasta que el paquete de bits actual termine.

Al activar el bit IFTF, el chip se configura para trabajar en Transferencia de Datos Transparente (TDT).

Los modos interno y externo difieren un poco en la forma que se temporizan sus señales, sin embargo en ambos casos se mantiene activas las funciones de protocolo. Si se coloca en uno el bit IFTF del Registro de Control, todas las funciones de protocolo se desactivan.

Esto provee un acceso bidireccional al puerto serie a través de la interfaz del microprocesador, con una memoria FIFO de 19 bytes en cada dirección.

En todas las transmisiones se envía primero el bit menos significativo de la palabra, o sea D0.

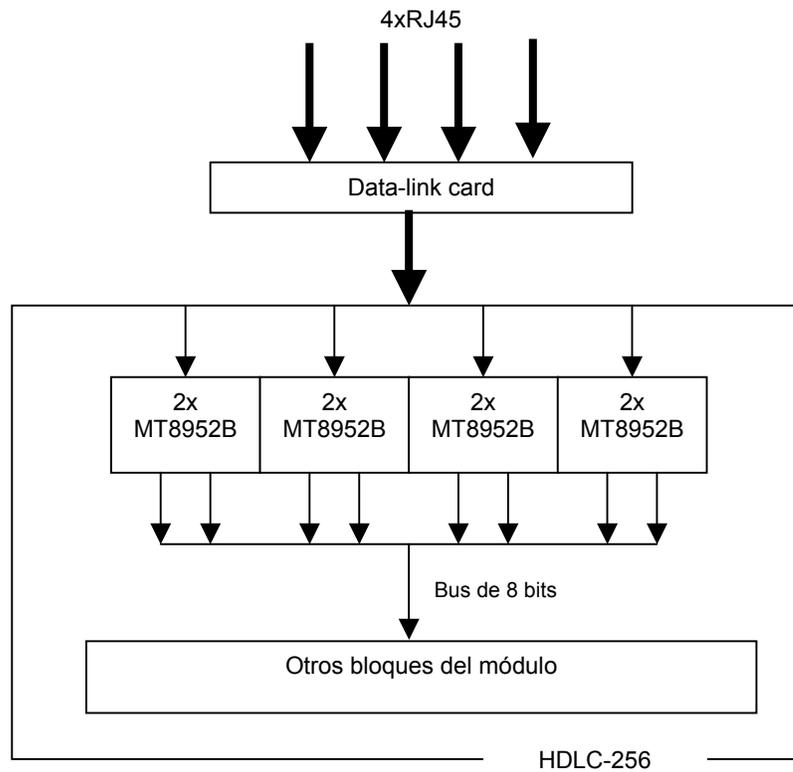
Este chip tiene la posibilidad de bloquearse para todas las recepciones que no inicien con un byte llave o dirección. Si está configurado para rechazar los mensajes que no se identifican con el byte de entrada correcto, el valor de la dirección se almacena en el Receive Address Register. Hay un byte de entrada que es reconocido como llave universal: FF<sub>HEX</sub>, y que identifica mensajes urgentes, los cuales deben ser aceptados por todos los circuitos del arreglo, sin importar la dirección que tengan programada.

Una característica importante del MT8952B es la cantidad de canales del ST-Bus que puede aprovechar. Hay varias formas de configurar el circuito en modo interno, para que acepte información en uno, dos o tres canales.

El canal uno se reserva para C-Channel, y dependiendo de la configuración, se da uso a los canales del dos al cuatro. Particularmente, en el caso de la HDLC-256, los chips trabajan en modo interno por parejas. Cada pareja de integrados recibe una línea ST-Bus que contiene dos data-links (uno en canal 2 y otro en canal 3) los cuales son procesados por separado.

En la Figura 2.5 se muestra un diagrama de cómo la información ingresa al módulo, desde los cables de RJ45 hasta que los MT8952B los colocan en el bus local.

La distribución de los datos en cuanto al espacio temporal se refiere (time slots) al ingresar a la tarjeta se muestra en la Figura 2.6. Cada MT8952 recibe un canal y se encarga de procesarlo.



**Figura 2.5** Ingreso de los datos a la tarjeta a través de los ST-Bus.

Composición de cada ST-Bus en la entrada de la tarjeta

Ch[4..31]	Ch3	Ch2	Ch1	Ch0
X	DL2	DL1	X	X

Composición de cada ST-Bus de los MT8952B pares

Ch[4..31]	Ch3	Ch2	Ch1	Ch0
X	X	DL1	X	X

Composición de cada ST-Bus de los MT8952B impares

Ch[4..31]	Ch3	Ch2	Ch1	Ch0
X	DL2	X	X	X

**Figura 2.6** Estructura lógica de los ST-Bus en la entrada de la HDLC-256.

Una vez que se filtra la información de protocolo HDLC, los datos se almacenan en una pequeña cola que al llenarse activa la interrupción del micro para que éste tome los datos por el bus de 8 bits. En la HDLC-256 no se utiliza C-Channel.

Dado el uso limitado del espacio temporal en el ST-Bus, con este componente es imposible realizar ampliaciones del número de enlaces de datos que pueden ser manejados por la HDLC sin aumentar la cantidad de chips y el cableado.

Para una descripción más profunda del funcionamiento y los registros internos del MT8952B, se hace referencia a las hojas de datos citadas en la sección 8.2, para este chip en particular.

#### **2.1.2.5 Características básicas del microprocesador TMS320C26**

El control de las funciones que se realizan en el bus interno de la tarjeta HDLC-256 está a cargo del microprocesador TMS320C26. Este chip está optimizado para realizar tareas relacionadas con el Procesamiento Digital de Señales, sin embargo en esta aplicación no se explotan sus características particulares DSP sino que se utiliza por su flexibilidad en la conexión con otros recursos.

Las principales características de este procesador son:

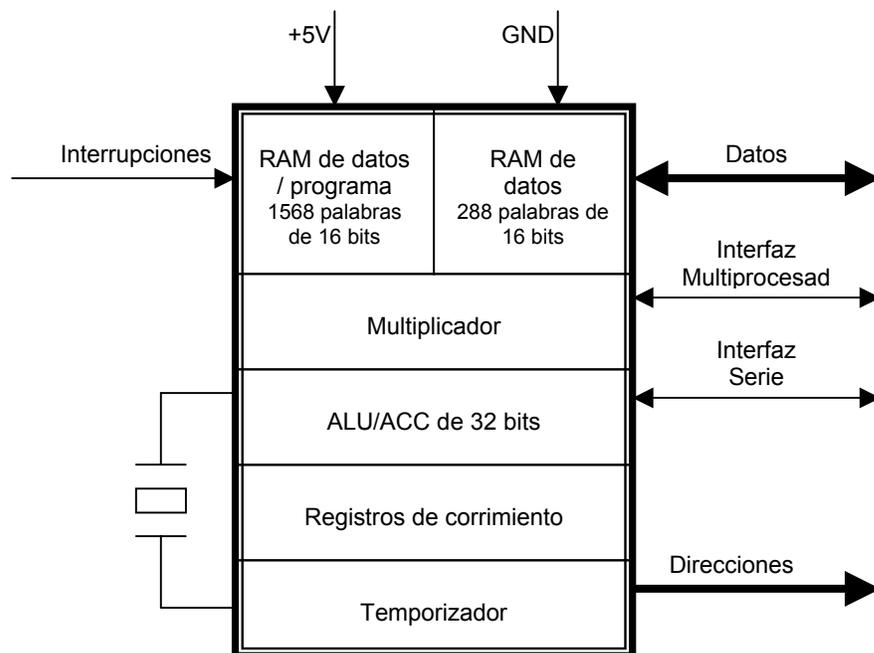
- a. Tiempo que toma realizar un ciclo de instrucción: 100 ns.
- b. 544 palabras de 16 bits en espacio de RAM interna.
- c. 1568 palabras de 16 bits programables como RAM de programa o datos.
- d. 128 Kbytes direccionables en total, para memoria de datos o de programa.
- e. ALU / Acumulador de 32 bits.
- f. Multiplicador paralelo de 16 x 16 bits con resultado de 32 bits.

- g. Instrucción de multiplicación y acumulación en un solo ciclo.
- h. Repetición de instrucciones para uso eficiente del espacio de programa y velocidad de ejecución.
- i. Movimiento de datos y programa en bloques.
- j. Temporizador interno para operaciones de control.
- k. Ocho registros auxiliares con unidad aritmética dedicada.
- l. Pila de ocho niveles implementada en hardware.
- m. Dieciséis canales de entrada y dieciséis de salida.
- n. Registro de corrimiento de 16 bits.
- o. Estados de espera para comunicación con dispositivos externos más lentos.
- p. Puerto serie para interfaz directa codec.
- q. Entrada de sincronización para configuraciones sincrónicas multiprocesador.
- r. Interfaz para memoria de datos globales.
- s. Compatibilidad con código de procesadores anteriores.
- t. Capacidad para hacer DMA.
- u. Instrucciones para filtrado adaptivo, FFT, y precisión aritmética extendida.
- v. Generador de reloj interno.
- w. Tensión de alimentación de 5V.
- x. Disponible en encapsulado PLCC de 68 pines.

La arquitectura del TMS320C26 está diseñada para maximizar la velocidad del sistema y el uso de los recursos que posee. Esto manteniendo cierto nivel de flexibilidad que permita una amplia gama de aplicaciones.

Por medio de un grupo de señales de control, es capaz de establecer transacciones de información en bloques, comunicarse con componentes lentos y realizar operaciones multiprocesador. Las características relacionadas con la aritmética de alta velocidad ayudan a mejorar el rendimiento en procedimientos relacionados con DSP.

En la Figura 2.7 se muestra el diagrama de bloques básico de este chip, en el cual se puede destacar el multiplicador como unidad independiente o auxiliar de la ALU.



**Figura 2.7** Diagrama de bloques simplificado del TMS320C26.

El firmware que controla este chip se encuentra en una memoria EPROM externa, y para los propósitos de la aplicación HDLC-256 también se agregó un banco de SRAM con el fin de ampliar la capacidad de almacenamiento temporal.

Dentro de las funciones que se encarga de implementar están:

- Inicializar los MT8952B y configurarlos en el modo de operación correcto.
- Atender las interrupciones provenientes de los MT8952B y descargar los datos a memoria RAM.
- Interpretar los mensajes recibidos por los ST-Bus y descartar los que no tienen importancia para el programa del supervisor.
- Ubicar los datos que van al CPU dentro de la FIFO, para que la computadora se encargue de recibirlos.
- Reconocer la información que sea enviada desde la computadora, y de ser el caso, ubicarla en uno de los canales ST-Bus de los controladores de protocolo HDLC.
- Mantener el control sobre un circuito Watch-Dog constantemente, antes que se dé la orden para reiniciar el módulo.

#### **2.1.2.6 Conexión y comunicación con el bus ISA**

La interfaz de la HDLC-256 con el bus ISA está compuesta por un arreglo de dos memorias FIFO, dos buffer y una serie de GALs programadas para sincronizar las señales del bus en la computadora, con el bus interno que controla el TMS320C26.

## **Características de la memoria FIFO**

Las memorias FIFO utilizadas corresponden al modelo CY7C419 fabricado por Cypress. Estos circuitos funcionan como colas de 256 palabras, cada una con un ancho de 9 bits.

El uso del MSB es opcional, de forma que si en la configuración de la memoria se especifica, la memoria trabajará únicamente a ocho bits, descartando cualquier dato que se encuentre en el MSB. Este es el caso de la FIFO tal como está el diseño de la HDLC-256.

La organización del CY7C419 permite realizar lecturas en el puntero de salida mientras se realiza escrituras en el puntero de entrada, todo en el mismo pulso de reloj. Posee banderas que indican estados lleno o vacío, con el fin de evitar operaciones de overrun y underun. Posee tres pines para proveer la expansión tanto en capacidad como en ancho de palabra, por medio de arreglos de varios chips.

La lectura puede ser sincrónica y asincrónica a una velocidad de 50 MHz máximo.

Presenta una entrada RT, que cuando se pone en bajo ordena a la memoria realizar retransmisión del último dato. Tanto R como W (habilitadores de lectura y escritura) deben estar en alto durante la retransmisión.

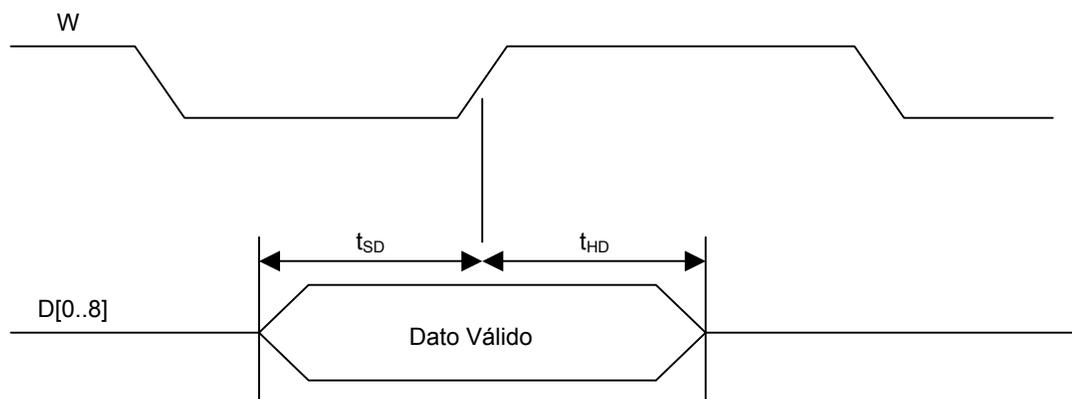
El ciclo de lectura requiere entre 20 y 35 ns., mientras que el tiempo de acceso está entre 10 y 25 ns. La FIFO funciona como una RAM de doble puerto, donde es posible utilizar las habilitaciones R y W en forma independiente para activar las celdas respectivas, para acceder en forma asíncrona las entradas y las salidas.

El tiempo requerido para incrementar los punteros de lectura o escritura es mucho menor que el tiempo que puede requerir el dato para su propagación en la memoria, esto en el caso en que la memoria se utilice como una arquitectura convencional de registros.

El estado inicial de la FIFO luego de un reset presenta todas las condiciones de una memoria vacía. Todas las salidas se encuentran en alta impedancia, y los datos en las celdas no son válidos. La condición de reset se da luego de conectar la alimentación, y después de aplicar un pulso de re-iniciación en la terminal Master Reset.

El proceso de escritura coloca un dato en la dirección del puntero de entrada, y mueve este último hacia arriba para que el siguiente dato pueda ser ubicado. El acceso a la última localidad vacía se indica con un alto en la terminal pin FF, y debe ser valorado por el controlador de memoria como una advertencia para evitar el overrun.

El flanco de bajada de W inicia el ciclo de escritura. El dato que aparece en la entrada D[0..8] después del tiempo  $t_{SD}$ , y debe esperarse un tiempo  $t_{HD}$  (Hold Data), luego del flanco de subida de la señal W. En la Figura 2.8 se puede apreciar la ubicación correcta del dato válido en el bus, de forma que sea correctamente escrito en la celda de memoria.



**Figura 2.8** Temporización de la escritura en la memoria CY7C419.

La bandera HF (media llena) se activa cuando se escribe en la siguiente a la mitad de la memoria. La bandera vacía se activa después de la transición de bajo a alto de W cuando la memoria se encuentra vacía. Por su parte, la bandera FF se pone en bajo luego de que se escribe en la última posición de la memoria.

Si se trata de escribir en la FIFO cuando se encuentra llena, la acción es ignorada y el puntero no se incrementa. La bandera FF se pone de nuevo en alto luego de que se ha leído un dato de la memoria, pues ya la condición de llena no es verdadera.

Para el procedimiento de lectura, es el flanco de bajada en R inicia el ciclo, siempre y cuando la bandera de memoria vacía (EF) no se encuentre en bajo. Las salidas Q[0..8] se encuentran en tercer estado entre dos ciclos de lectura, es decir cuando no se está dando lectura de un dato, las salidas están en tercer estado.

La capacidad de retransmisión es beneficiosa en la transferencia de paquetes de datos. Se habilita la recepción del dato para ser reconocido por el receptor y se retransmite si es necesario.

La entrada de retransmisión (RT) se activa en el modo normal y las expansiones de ancho. La retransmisión se intenta cuando existe un número de escritura igual o menor a la capacidad mientras se da el último ciclo del master reset. Un pulso en bajo en la entrada RT devuelve el puntero interno de lectura a la primera localización física de la memoria FIFO. Las señales de R y W deben estar en alto durante el  $t_{RTR}$  (retransmit recovery time) pero después de la retransmisión están en bajo. Después de una retransmisión el puntero de lectura apuntará hacia la ubicación del puntero de escritura.

Cuando se trabaja a la máxima frecuencia las banderas son validadas al inicio del siguiente ciclo, para prever escrituras falsas en memorias llenas, o una lectura falsa si la memoria está vacía.

## **Registro de estado**

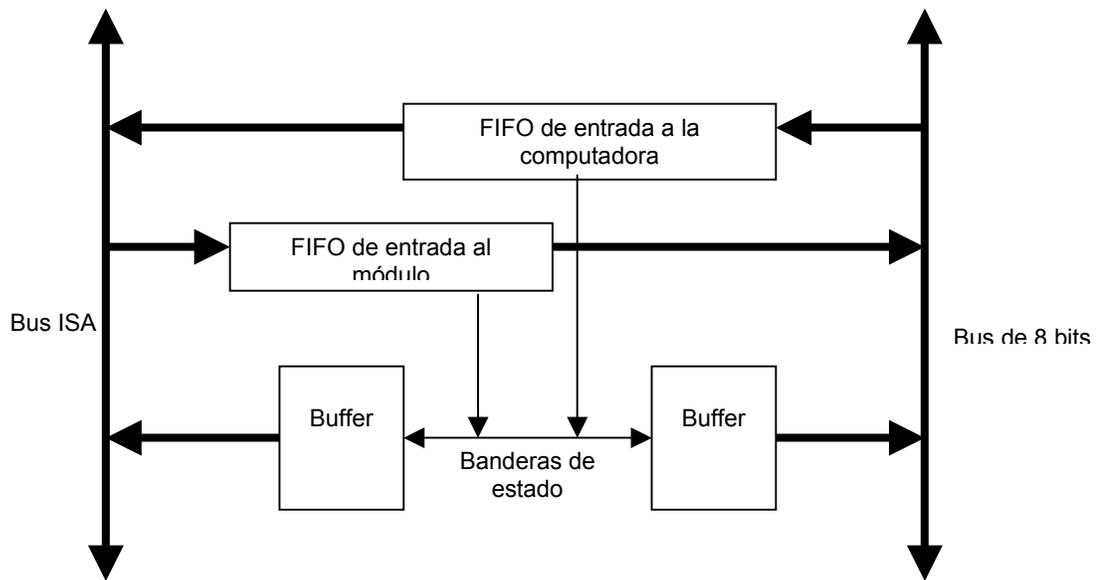
En la HDLC-256 hay dos colas de datos, que corresponden a dos chips CY7C419. La primera se encarga de guardar información que va desde el módulo hacia la computadora, y la otra en dirección inversa. Las banderas de estos chips se reconocen como el estado de comunicaciones de la tarjeta, y se conectan a las entradas de los buffer.

Cuando el microprocesador interno quiere leer el estado de comunicaciones, se encarga de habilitar el buffer de entrada al bus local. Si la computadora requiere leer el estado, entonces se habilita el otro buffer para que los datos salgan al bus ISA.

En la Figura 2.9 se tiene el esquema seguido por este arreglo de elementos. De esta manera, existe una dirección de I/O para obtener información sobre las banderas de las FIFO tanto desde el procesador local, como por parte de la computadora. Esto permite actuar según lo que refleje esta información y variar controlar el flujo de los datos.

Desde ambos buses esta dirección es accesible únicamente para lectura. Los buffer utilizados son de la familia TTL, específicamente el 74ABT244.

La sincronización de este arreglo con el bus ISA se realiza por medio de un sistema de control programado en GALs.



**Figura 2.9** Arreglo de circuitos que permiten usar el registro de estado.

### Lógica de sincronización de los buses

Para lograr la compatibilidad entre la señalización del bus interno y la del bus ISA, fue necesario planificar una estructura lógica capaz de realizar las tareas de traducción de comandos, además de la asignación y habilitación de las direcciones.

Como toda tarjeta de expansión, la HDLC-256 debe tener asignado un espacio de I/O y una IRQ entre los recursos disponibles del sistema. Una de las GALs se encarga de convertir estos parámetros para que las órdenes de la computadora causen el movimiento correcto de información en el bus local y la interrupción del CPU se active con un evento específico dentro de la tarjeta.

Hay cuatro GALs, modelo 16V8 de Lattice. La numeración de estos chips dentro del circuito es IC9, IC14, IC15 e IC17. Cada una desempeña un cierto papel en el acople de los dispositivos.

El circuito GAL IC9 se encarga de seleccionar uno de los controladores de protocolo HDLC, partiendo de ciertas líneas de dirección del TMS320C26, de forma tal que cuando se hace un ciclo a la localidad en la cual está ubicado el MT8952B correspondiente, la línea respectiva /CS se activa y habilita únicamente al chip seleccionado.

Como los controladores de HDLC están mapeados en espacio de memoria, IC9 debe chequear la salida DS del microprocesador. Además, dado que todos los accesos implican un ciclo de bus, los chips son habilitados hasta que aparece la señal STRB, la cual indica cuando el procesador está listo para recibir datos (lectura) o cuando ya ha colocado el dato correcto en el bus (escritura).

En la Tabla 2.2 se presenta la disposición de los pines en IC9.

**Tabla 2.2** Función de los terminales en IC9.

Entrada	Pin	Salida	Pin
A13	2	/CS0	19
A14	3	/CS1	18
A6	4	/CS2	17
A5	5	/CS3	16
A4	6	/CS4	15
/DS	7	/CS5	14
/STRB	8	/CS6	13
X	9	/CS7	12

Siguiendo con IC14, éste se encarga de sincronizar el habilitador (Enable) de los 8952 con el reloj del procesador. Está encargado de activar la señal Ready del procesador, la cual da se mantiene en bajo cuando se desea insertar tiempos de espera. Además, dependiendo de las direcciones que le llegan a la entrada, puede permitir lectura del estado de las FIFOs o bien de los datos en la cola de entrada.

En las entradas se recibe R/W, /DS y /STRB, que son indicadores de la orden que está ejecutando el procesador y el bloque en el cual se debe dar el proceso. La Tabla 2.3 muestra la disposición de pines en este chip.

**Tabla 2.3** Función de los terminales en IC14.

Entrada	Pin	Salida	Pin
R/W	2	E-Clock	19
/DS	3	READY	18
A13	4	Q1	17
A12	5	BUSY	16
A11	6	BUSY2	15
A0	7	/FIFO	13
/STRB	8	/DSP244	12

Con respecto a IC15, este es el circuito que más se relaciona con la señalización de control del bus ISA. Una de las entradas se denomina /SEL3XX, y se activa cuando la computadora desea realizar un acceso al puerto designado para la tarjeta. El valor de este puerto se puede modificar por medio de un dip-switch, según lo establezca el encargado de la instalación.

Una vez que se activa el rango de direcciones correspondiente al módulo, por medio de las entradas A[0..2] se determina el dispositivo que va a ser seleccionado desde el bus ISA. Además, las entradas /IOW e /IOR determinan si el ciclo será de escritura o de lectura respectivamente.

En la salida de este chip se tiene un conjunto de líneas de habilitación y selección de los dispositivos en el bus local. Lo anterior se muestra en la Tabla 2.4.

Si desde el bus ISA se direcciona la posición 100, hay una señal de escritura a I/O, y el dispositivo se encuentra activo, la señal WDP se activa produciendo un reset del Watch-Dog. Por su parte, WDE se genera direccionando 001 y corresponde al WD clock.

**Tabla 2.4** Función de los terminales en IC15.

Entrada	Pin	Salida	Pin
/SEL3XX	2	/WDP	19
IOR	3	/WDE	18
IOW	4	/WPC	17
A2	5	/PC244	16
A1	6	/RPC	15
A0	7	/RST	14
X	8	/OWS	13
X	9	/IOCS16	12

Por último se tiene el IC17. Una sección de este chip está encargada de controlar las señales de la SRAM para su correcto uso. Además controla el ingreso de datos desde la cola que viene de la computadora y activa la interrupción asignada a la tarjeta.

Desde el lado del microprocesador, se decodifica un bloque de direcciones que corresponde a la ventada de RAM asignada. Una vez realizada la conexión electrónica, el programador sólo debe preocuparse por no exceder la cantidad de memoria disponible. La asignación de los espacios de memoria se realiza en los encabezados, si la programación es en ensamblador; en el caso de compiladores para lenguaje C, esta asignación se hace automáticamente en la compilación. Algunas salidas de este chip están destinadas a activar la memoria RAM cuando el procesador lo indique. Se utiliza una GAL para disminuir la cantidad de chips necesarios. Las funciones de los pines para IC17 se muestran en la Tabla 2.5.

**Tabla 2.5** Función de los terminales en IC17.

Entrada	Pin	Salida	Pin
/FIFO	2	/RDSP	19
/STRB	3	INT	18
RW	4	/WDSP	17
/DS	5	/OERAM	16
TINT	6	/CE1	15
/HF	7	/WERAM	14
A13	8	X	13
X	9	X	12

## **2.2 Requerimientos de la empresa**

El diseño de la nueva tarjeta debe cumplir con una serie de requerimientos, de forma tal que sea compatible con los equipos periféricos actuales y a la vez pueda adecuarse a los equipos de cómputo modernos.

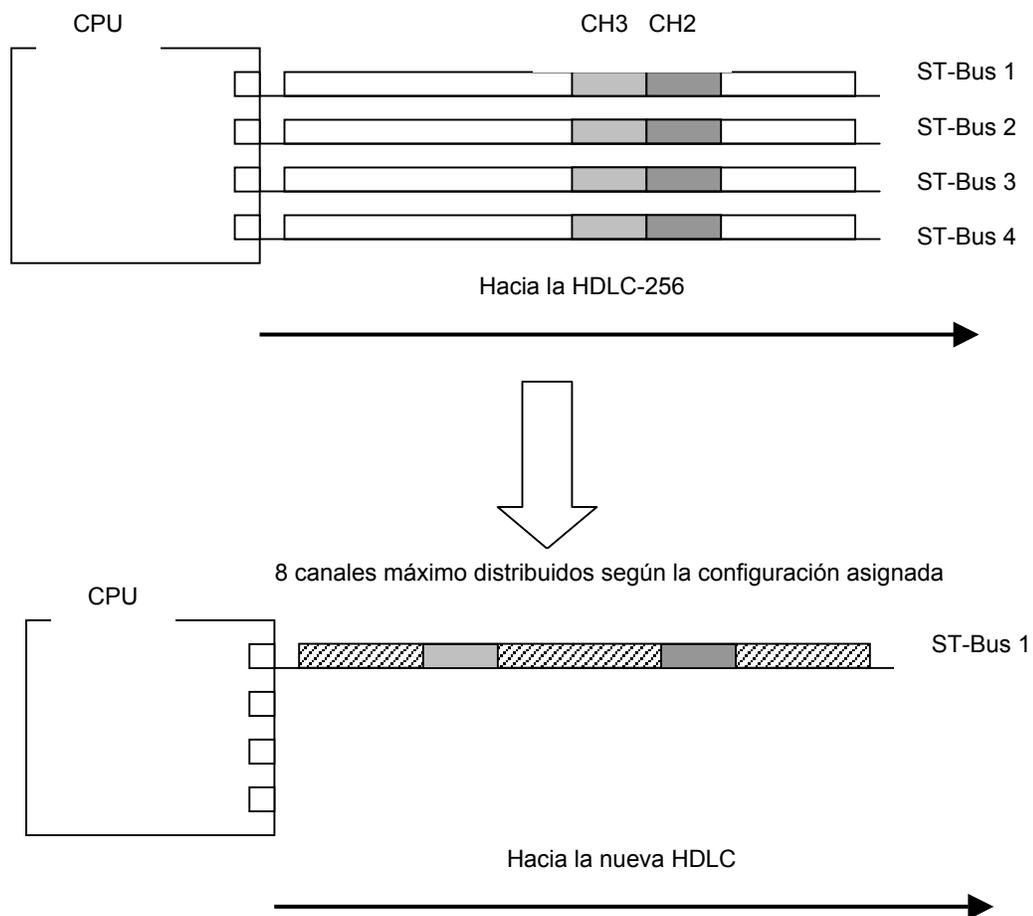
Además, se ha decidido asignar una mayor capacidad de procesamiento, para disminuir la cantidad de módulos necesarios para atender la información proveniente de un Back-Panel.

### **2.2.1 Requerimientos en la entrada del bus serial**

La tarjeta HDLC-256 está sujeta a las características particulares del MT8952B. Precisamente por esto, los ST-Bus que recibe utilizan únicamente los canales 2 y 3 para la transmisión de los datos desde los CPU. Esto es inflexible, y debe ser así en todos los sistemas. El requerimiento de la empresa en este aspecto consiste en utilizar los ST-Bus de una forma más eficiente, aprovechando más sus capacidades para el traslado de información y logrando un sistema flexible, que pueda configurar por software el número de canales que va a atender en cada ST-Bus.

La convención adoptada por Cibertec implica un máximo de 32 canales para cada nuevo módulo HDLC, con una distribución de 8 canales máximo en cuatro ST-Buses. Esto no conlleva a problemas de compatibilidad con los equipos antiguos, ya que la distribución de los data-links en las tarjetas CPU es programable.

La modificación que se deberá hacer es meramente una cuestión de software, cambiando la distribución de la información de 2 canales por cuatro ST-Bus a 8 canales en un ST-Bus, tal como se muestra la Figura 2.10.



**Figura 2.10** Modificaciones en la distribución de los datos en la entrada.

Esta configuración debe quedar abierta, ofreciendo al diseñador del software libertad para elegir la posición de cada canal y la cantidad de ellos, siempre que no se exceda el límite de 8 asignado como criterio de diseño. Se recalca que este no es el tope de canales permitido por ST-Bus.

Ahora, dada esta distribución de datos, se debe respetar las características propias del estándar ST-Bus dadas por Mitel, en cuanto a las tensiones que se manejan, las señales de sincronización, la frecuencia de trabajo, etc.

De esta forma permanece abierta la opción de realizar cambios futuros en la arquitectura de las tarjetas CPU, siempre que el formato de comunicación se mantenga invariante: códigos de protocolo HDLC enviados a través de ST-Bus.

### **2.2.2 Requerimientos de capacidad de procesamiento**

Las tarjetas HDLC-256 están capacitadas para manejar la información proveniente de ocho enlaces de datos. Cada uno de estos módulos está conectado a un CPU y procesa el protocolo HDLC para que la computadora obtenga los datos de señalización SS7 recopilados por ese CPU.

Para el nuevo diseño se pretende aumentar la capacidad, de forma que una sola tarjeta HDLC pueda recibir y procesar sin problema los datos provenientes de cuatro CPUs. Esto corresponde a la información obtenida en un back-panel de los equipos ITSS.

La máxima capacidad está prefijada en 32 data-links. El módulo debe ser programable para aceptar variaciones a esta regla tales como distribuciones no simétricas en la carga de los ST-Buses.

Este aspecto permitirá a los equipos de Cibertec aprovechar de una forma más eficiente el espacio físico y la distribución de los cableados, además de permitir una mayor flexibilidad en la configuración de los sistemas.

### **2.2.3 Requerimientos en el bus de la computadora**

Este punto es precisamente sobre el cual se profundiza más en el proyecto. El módulo HDLC-256 es una tarjeta de expansión basada en bus ISA, y por lo tanto cumple con los estándares utilizados en computadores viejos.

Los equipos de cómputo modernos tienden a abandonar las versiones antiguas de buses, incluyendo ISA. Esto se da básicamente porque el incremento en la velocidad de los procesadores y el tamaño de memoria requiere estándares de bus más eficientes, veloces y robustos.

Cibertec pretende obtener una versión tarjeta HDLC compatible con bus PCI. Este requerimiento cambia totalmente la orientación a seguir en cuanto la estructura lógica y electrónica que debe plantearse en la interfaz con la computadora, porque PCI posee una gran cantidad de especificaciones de diseño.

Para que el módulo sea compatible con PCI se debe seguir un estricto procedimiento de diseño eléctrico, paralelo con la arquitectura relacionada con el flujo de información.

Este factor es fundamental en las aspiraciones de Cibertec por obtener una tarjeta de expansión acorde con las necesidades del mercado actual, que sea utilizable en cualquier parte del mundo por equipos de diversos fabricantes.

Seguir los requerimientos de PCI implica aumentar el tamaño del bus de datos, cambiar los controladores y la forma como se está comunicando la tarjeta actual con las plataformas de aplicación en el ITSS.

#### **2.2.4 Requerimientos de instalación, configuración y PnP**

Uno de los objetivos que se desea alcanzar en Cibertec con el nuevo módulo es que ofrezca todas las características de compatibilidad “Plug and Play” que (de por sí) se incluyen en el estándar PCI. Esto facilitará la instalación y configuración de la tarjeta en equipos compatibles con esta filosofía.

Además, el comportamiento de la tarjeta debe ser configurable en el procedimiento de iniciación, donde además se debe asignar sus recursos y ubicación dentro del mapa de memoria y/o periféricos de la computadora.

#### **2.2.5 Requerimientos de vigencia en el mercado**

Es importante para la empresa, fabricar una tarjeta con componentes vigentes en el mercado, de fácil adquisición y que utilicen la tecnología más actual para periféricos de computadoras.

Esto con el fin de que el producto tenga una larga vida útil, y que el diseño pueda ser adaptado fácilmente a los avances que se den en los años venideros. Además, el precio de los componentes es más barato si estos se encuentran en proceso de producción por parte de sus fabricantes; una vez que se vuelven obsoletos y es más difícil adquirirlos, se vuelven más caros.

Se pretende utilizar elementos que sean identificables en el proceso de iniciación, para flexibilizar la configuración de la tarjeta, según los recursos que tenga presentes. Así las modificaciones de software serán mínimas si en versiones mejoradas de la tarjeta se debe cambiar algún chip.

#### **2.2.6 Requerimientos de montaje de los componentes**

Con el fin de utilizar la tecnología más reciente en el diseño de esta tarjeta, es necesario utilizar un ensamble en montaje superficial.

Esto implica lidiar con chips en presentaciones tales como BGA o SOIC. Es necesario considerar este factor en el proceso de elección de partes, para que se mantenga un nivel de estandarización en la presentación de todos los componentes.

Además es importante especificar perfectamente detalles relacionados con las dimensiones físicas de cada parte, para que no exista disconformidades cuando se encargue la fabricación del impreso y el ensamble.

### **2.3 Solución propuesta**

La solución propuesta para satisfacer la necesidad de Cibertec Int. de un nuevo módulo HDLC se denomina HDLC-4M, que hace referencia a la cantidad de memoria local mínima requerida para el procesamiento de 32 data links en forma simultánea: 4 Mega bytes.

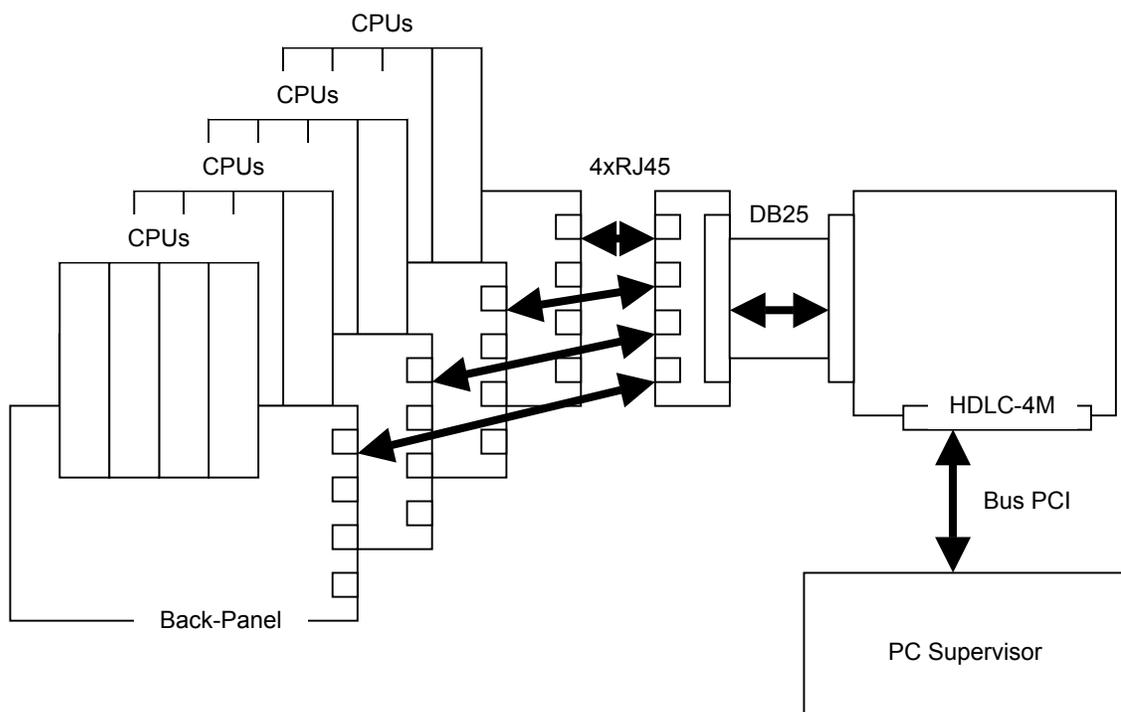
La HDLC-4M está basada en el uso de un nuevo chip disponible en el mercado: el PM7366 de PMC Sierra Inc., que también es conocido como Freedm-8. Gracias a las características de este nuevo componente, es posible recibir información por muchos canales del ST-Bus, contrario al caso del MT8952-B que únicamente aprovecha los espacios temporales 2 y 3.

Así, la tarjeta HDLC-4M dispondrá de cuatro entradas ST-Bus, cada una de las cuales podrán comunicar múltiples canales de información. El número máximo de data-links procesados será de 32, que van a estar distribuidos entre las cuatro entradas ST-Bus.

En el proceso de iniciación se debe realizar una distribución determinada del número de canales asignados por ST-Bus según sea el requerimiento del sistema. El patrón más sencillo es ubicar 8 canales por bus, o sea, una distribución balanceada.

La capacidad del módulo HDLC-4M permite atender simultáneamente los datos provenientes de cuatro back-panel, usando distribución balanceada, y realizando una ligera modificación al firmware de las tarjetas CPU.

Esto implica que la nueva versión de la HDLC estará capacitada para ubicarse en el sistema de la forma que se muestra en la Figura 2.11. Aquí se observa cómo el aumento en la capacidad de procesamiento ayudará a disminuir el espacio físico y la cantidad de tarjetas requeridas para que el supervisor reciba la información de protocolo SS7, provenientes de un número mayor de CPUs.



**Figura 2.11** Interconexión de la HDLC-4M con su equipo periférico y con el PC.

Si se hace una comparación de la Figura 2.11 con la Figura 2.1 se puede observar una serie de detalles en los cuales difieren estas tarjetas:

- a. La cantidad de back-panel que pueden ser atendidos en forma simultánea aumenta de uno en la 256 a cuatro en la 4M.
- b. El bus de comunicaciones de la computadora pasa de ser ISA en la 256, a PCI en la 4M.

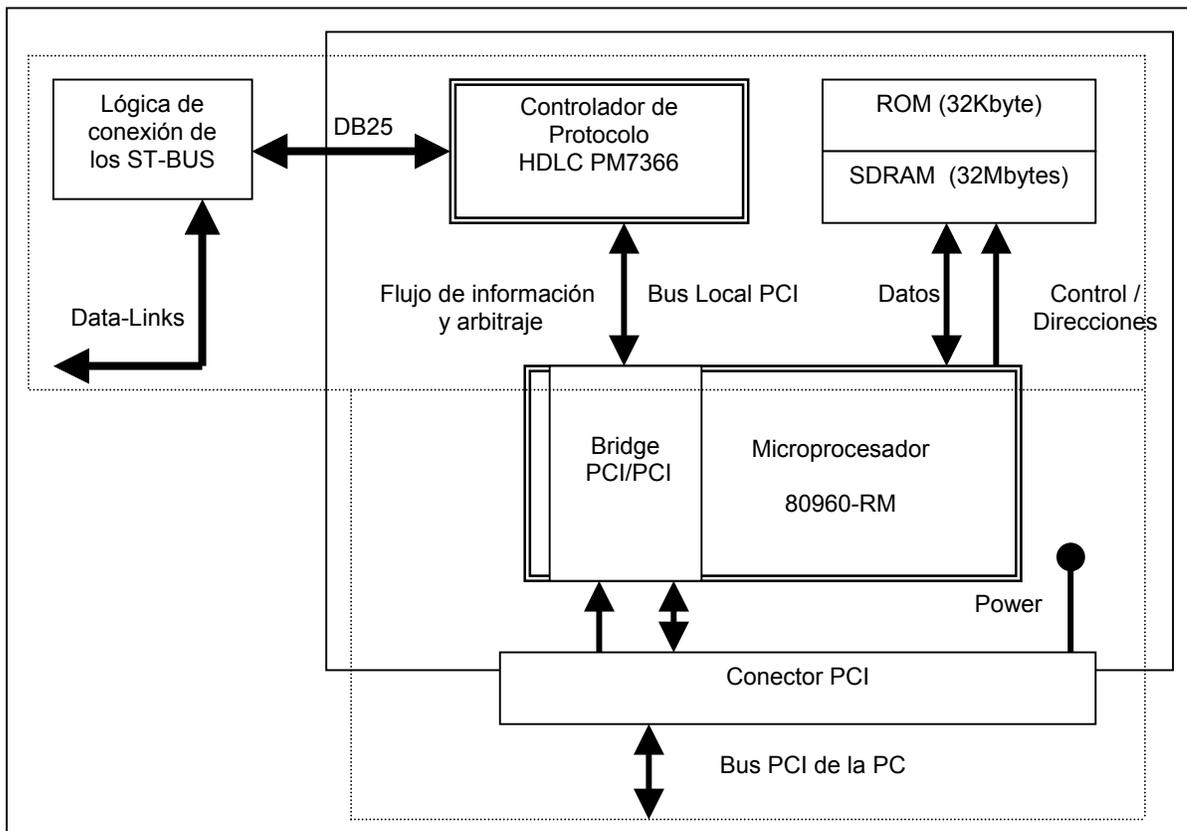
Además es necesario aclarar que, aunque en las figuras citadas no se observa este detalle, los componentes en los cuales se basan estas tarjetas son totalmente diferentes.

En el caso de la HDLC-4M, estos nuevos componentes se han elegido de forma que se pueden adaptar al estándar ST-Bus en la entrada, son capaces de procesar mensajes en protocolo HDLC, y tienen un sistema de comunicación con el bus PCI de la computadora. En la Figura 2.12 se presenta el diagrama de bloques correspondiente a la arquitectura aprobada para la HDLC-4M.

Con el fin de satisfacer los requerimientos del sistema en cuanto a velocidad de procesamiento e interfaz con PCI, se utiliza un microprocesador nuevo, que sustituye al TMS320C26 en las labores de control y filtrado de mensajes.

Este sistema mínimo posee memoria ROM y RAM, además de una serie de circuitos adicionales encargados de manejar los relojes, sincronizar los dispositivos y manejar los buses que usen multiplexación en sus funciones.

La presentación de los componentes se ajusta a los estándares para montaje superficial, y en algunos se utiliza bases. La memoria RAM corresponde a un módulo de SDRAM en montaje tipo DIMM, que se conecta por medio de un socket, mientras que la ROM de programa es un chip convencional con bus de 8 bits, también montado sobre base para facilitar la rutina de actualización del firmware.



**Figura 2.12** Estructura aprobada en Cibertec para el módulo HDLC-4M.

El bloque denominado “Lógica de conexión de los ST-Bus” está fuera de la tarjeta, y corresponde a la adaptación que debe hacerse en el dispositivo que acopla los RJ45 con el cable DB25.

Luego está el controlador de protocolo HDLC, que funciona de forma similar a varios chips MT8952B integrados. Este chip es fabricado por PCM Sierra Inc. y posee una gran flexibilidad en cuanto a su configuración de trabajo. Puede procesar hasta 256 data-links, sin embargo se ha fijado un máximo de 32 para el proyecto 4M.

Su finalidad es recibir la información correspondiente a la señalización SS7, eliminar los rastros del Protocolo HDLC y colocar estos datos en el bus PCI local para que puedan ser almacenados y procesados por el microprocesador.

Este procesador es una versión especializada de la familia 960 de Intel. Corresponde a un chip especial para funciones de I/O, con interfaz PCI hacia dos buses y puente incorporado. Está capacitado para controlar memoria SDRAM, y lee su programa desde una ROM o memoria Flash de 8 bits.

Dentro de este chip se analiza los mensajes obtenidos del controlador HDLC para enviar al computador únicamente aquellos que tienen importancia para el programa del supervisor.

Además, el micro debe realizar todas las tareas de control y distribución de la memoria interna del PM7366, ya que éste último es dependiente de un "host" en cuanto su iniciación y manejo.

La memoria SDRAM se usa para guardar los datos correspondientes a las tablas de distribución de los datos en el Freedm, las variables locales y temporales del programa, y los bloques de información lista para ser enviada al computador.

Todo el código de programa es leído por el procesador desde una memoria PROM o EPROM, en la cual se guarda el archivo de extensión HEX, generado por el compilador adecuado.

Los aspectos relacionados con la iniciación de la tarjeta, la identificación de los dispositivos por PnP, la asignación de recursos y la distribución de las tablas en memoria, deben ser cuidadosamente programados en los encabezados del código.

Una vez que todos los componentes se encuentren configurados correctamente, y la comunicación en el bus del PC sea establecida, el proceso de manejo de los datos se reduce a un trasiego de información desde espacios de memoria local hacia el CPU y la atención de interrupciones.

Finalmente, se debe resaltar el hecho de que los equipos periféricos CPUs no requieren modificaciones de hardware, y que el formato de comunicaciones serie sigue siendo ST-Bus, con lo que se mantiene un alto nivel de compatibilidad con los equipos periféricos.

La descripción detallada del hardware utilizado en este módulo se presenta en el capítulo 4, donde se incluyen detalles específicos de cada elemento, la forma en que se eligieron y su conexión con el resto de la tarjeta.

Por otro lado, una vez que se logra un alto nivel de avance en el área electrónica se procede a la programación de los dispositivos, de acuerdo con sus propiedades específicas.

La sección de software se presenta en el capítulo 5, donde se describe el ambiente en el cual se llevará a cabo la programación. Los objetivos llegan hasta la correcta iniciación del módulo, así que los alcances logrados no profundizan en cuanto a las rutinas y procedimientos que la tarjeta realizará.

# **CAPÍTULO 3**

## **PROCEDIMIENTO METODOLÓGICO**

Con el fin de cumplir los objetivos planteados en este proyecto, se ha seguido una serie de pasos bien definidos, de forma que el trabajo global pudiera ser enfrentado como un conjunto de problemas más pequeños.

Las actividades que han sido completadas hasta el momento son las siguientes:

- a. Investigación preliminar,
- b. Preparación del diagrama funcional,
- c. Análisis del estándar PCI,
- d. Conexión del circuito puente,
- e. Diseño de la estructura del bus secundario,
- f. Uso del bus secundario,
- g. Programación de encabezados,
- h. Documentación.

Por su parte, existe una serie de actividades que ha sido desarrollado únicamente en forma parcial, como plataforma para el trabajo que se puede realizar en la empresa a partir de los resultados obtenidos:

- a. Adquisición de licencias,
- b. Contratación de empresas para el PCB y el ensamble,
- c. Transferencia tecnológica.

Estos pasos a su vez se dividen en tareas más pequeñas, las cuales están sujetas a un cronograma de cumplimiento.

Basándose en este conjunto de tareas, se llevó un documento paralelo denominado Bitácora, en el cual se ha tomado nota de las actividades diarias, para verificar el cumplimiento del cronograma.

Casi todas tareas cumplen con un cierto orden cronológico, pues en varios casos fue necesario avanzar en un paso antes de poder dar el siguiente. Sin embargo hubo procedimientos que se llevaron a cabo en paralelo, dedicando momentos para tareas pequeñas de áreas separadas.

Por ejemplo, a la vez que se iba profundizando en el análisis del estándar PCI, se hizo más fácil realizar las conexiones de la interfaz con el bus de la computadora.

En cuanto a los procedimientos que no han sido completados, son un inicio para lo que la empresa pueda desarrollar a partir de este momento, dadas las condiciones teóricas planteadas en los objetivos. Esto se justifica en el hecho de que los factores de tiempo en pedidos de componentes y obtención de licencias no permiten llegar a un resultado antes de que se cumpla el límite para la presentación del proyecto al I.T.C.R.

La metodología utilizada permitió llegar a un desarrollo completo del hardware, y un avance en la programación por medio del análisis y estructuración de las rutinas de arranque. A partir de aquí se debe seguir una nueva metodología para la compra de partes, la fabricación del impreso, el ensamble, la programación completa y las pruebas de laboratorio.

### **3.1 Investigación preliminar**

Este fue el primer paso del proyecto. Se llevó a cabo durante las primeras semanas de trabajo. Comprende un conjunto de temas que se investigaron, relacionados con el ambiente en el cual se va ubicar el proyecto y los aspectos fundamentales sobre los cuales debe basarse.

Dentro de esta investigación se incluye la información recopilada sobre la empresa Cibertec y su equipo. Se hizo un documento en el cual se resume las características del ITSS y los tipos de fraude telefónico que ataca.

Una vez que se logró un conocimiento amplio sobre el equipo dentro del cual se ubica la tarjeta nueva, se puntualizó el estudio en la versión actual: la HDLC-256. Para poder conocer cuales características debe cumplir el diseño nuevo, primero se analizó la estructura y la lógica de funcionamiento del circuito creado por los ingenieros de Cibertec.

Así se pudo determinar aquellos requerimientos básicos que deben permanecer invariantes y cuales deben ser modificados. En esta etapa se profundizó un poco en el funcionamiento del ST-Bus, del protocolo de comunicaciones HDLC y del chip MT8952B.

Se obtuvieron las hojas de datos de los componentes principales de la HDLC-256 para comprender a fondo su funcionamiento individual y la forma como se comunican unos con otros en el módulo completo.

A esto se agrega un proceso de búsqueda de información sobre el bus ISA. Así se logró un primer perfil de la forma en que la tarjeta de expansión se comunica con la computadora.

Debe agregarse que en este tiempo se llevó a cabo el proceso de adaptación del estudiante al ambiente laboral de la empresa.

### **3.2 Preparación del diagrama funcional**

A partir de la investigación hecha, se determinó cuales eran los requerimientos que la nueva tarjeta debía que cumplir. Se llevó a cabo un estudio de las posibles alternativas que podrían seguirse para solucionar el problema planteado, incluyendo varios diagramas de bloques basados en diferentes chips disponibles en el mercado.

Se realizó un estudio comparativo para averiguar si es más factible usar un circuito puente disponible en el mercado o realizar la interfaz con el bus PCI a partir de un arreglo de GALs y FIFOs.

Este proceso consistió en revisar una serie de documentos web, y contactar empresas fabricantes o distribuidoras para solicitar precios y características distintas de sus productos.

Los pasos siguientes en la elección de los componentes estuvieron orientados hacia el tipo de ROM, la presentación de controlador de protocolo HDLC (que ya había sido escogido por Cibertec) y los chips auxiliares necesarios en la interconexión de los elementos.

Con una lista de las principales partes que formarían la tarjeta, se realizó la búsqueda de los manuales y las hojas de datos en formato PDF, los cuales fueron ubicados en los sitios web de sus respectivos fabricantes, o en su defecto, de distribuidores autorizados.

El último paso que se dio para cumplir con esta sección del proyecto fue la conexión del flujo de información, de acuerdo con las características particulares de los componentes elegidos.

Al finalizar este proceso, la tarjeta ya tuvo una composición definitiva. El siguiente paso en el diseño del circuito implicaba la creación del esquemático en Orcad, basado en el diagrama de bloques logrado.

### **3.3 Análisis del estándar PCI**

Paralelamente al avance en cuanto a los elementos que la tarjeta debería llevar, fue necesario adentrarse en el estudio del estándar PCI.

Para poder entender el funcionamiento y características de este bus, se utilizó cierta bibliografía adquirida por Cibertec. A partir de este conjunto de libros, se adquirió un perfil más concreto sobre todos los aspectos que rodean al diseño y uso de este bus. Además, se investigó un poco en la Internet sobre el tipo de tarjetas de expansión que usan PCI y sus fabricantes, para tener una idea de la clase de aplicaciones que se puede encontrar en el mercado.

Los primeros pasos en el estudio de la bibliografía estuvieron enfocados hacia la ranura PCI, las diferentes señales que se encuentran y las características físicas. Esto a un nivel muy descriptivo, hasta poder lograr cierto grado de familiaridad con el bus.

Luego de revisar los aspectos básicos, se hizo claro que PCI no es sólo cuestión de un bus eléctrico, sino que tiene implícita una gran cantidad de factores tanto de software como de hardware, y que su impacto en el paradigma de las computadoras es considerable.

En este punto, se inició en proceso de descripción detallada de los términos relacionados con la estructura del bus: master, target, arbitro, recursos, dispositivos, etc. Además, profundizó más en aspectos como las funciones de cada línea de señal, y el procedimiento establecido para realizar los ciclos de bus.

A partir de aquí se debió realizar una separación de los bloques de señales por su función específica dentro del trabajo del bus. Con ello se inició el proceso de conexión de los componentes, teniendo claro el porqué de estas relaciones eléctricas en las primeras versiones del esquemático.

La forma en que interactúan los recursos de un bus PCI es mucho más compleja que en sus antecesores. Esto mejora su rendimiento pero requiere un protocolo de control de mayor nivel.

En cuanto a las características eléctricas se refiere, por la velocidad de operación que usa, el bus PCI requiere cuidados especiales en la distribución de sus pistas, la carga que cada componente implica para el bus, las señales de reloj y el desacople, entre otros.

Todo esto debió ser analizado y aplicado en el proyecto. El uso de componentes diseñados para trabajar con PCI facilita las conexiones, sin embargo es necesario seguir al pie de la letra las recomendaciones dadas por la industria para asegurar el correcto funcionamiento de la tarjeta.

Aunque la obtención del circuito impreso no forma parte del proyecto, es importante citar las normas básicas que se debe acatar al diseñar un PCB que cumpla con las recomendaciones de PCI.

Con un panorama más claro de lo que implica trabajar con PCI, si debió unir todos los puntos estudiados: conexión eléctrica al nivel de esquemático, interacción entre los componentes, transferencia de información y consideraciones eléctricas en la fabricación del circuito.

Es importante señalar que el estándar PCI completo es muy amplio, y para documentación más detallada, profunda y revisada sobre este particular, se remite a los volúmenes citados en las Referencias Consultadas.

### **3.4 Conexión del circuito puente**

Gracias a los procesos de identificación con el bus PCI y sus señales, e identificación de los componentes, se pudo iniciar la creación de los modelos en Orcad para la interconexión de las señales.

El primer paso fue la descripción completa de la ranura PCI y el diseño de su modelo para Orcad. Este proceso se llevó a cabo disponiendo las señales de tal manera que coincidieran con el modelo del bridge. En algunos casos hubo que agregar circuitos auxiliares, condensadores de desacople y resistencias de pull-up, tal como se recomienda en las especificaciones para PCI. Así se procedió a la conexión del primario.

Cuando esto estuvo listo, se inició la conexión del secundario con el circuito controlador de protocolo HDLC, para el cual también se hizo un modelo.

Al finalizar esta etapa se obtuvo el esquemático básico con los dos buses PCI conectados. A este circuito se deben agregar las conexiones de memoria y elementos auxiliares.

### **3.5 Diseño de la estructura del bus secundario**

La estructura del bus secundario se refiere tanto al cableado eléctrico como a la distribución lógica de los dispositivos en este bus. Particularmente, en la solución planteada, sólo el chip controlador de protocolo HDLC está conectado al secundario del bridge, y dadas las características del microprocesador, el manejo de sus recursos se hace por software.

Por lo tanto, se procedió a revisar que las señales del controlador HDLC estuvieran bien conectadas, según los propósitos para los cuales se va a usar.

Se realizó un análisis sobre la forma de conectar la memoria local del procesador, que es accesible desde los buses por medio de una interfaz interna del 80960RM. Además, se desarrolló la estructura del mapa de memoria en el bus local (desde el punto de vista del procesador) ubicando tanto los bloques de memoria, como las ventanas hacia los buses y los dispositivos periféricos que posee el chip.

Esta estructuración debe permitir hacer variaciones de nivel intermedio en el funcionamiento de la tarjeta por medio de cambios en la configuración, sin alterar el hardware. El objetivo de esta forma de diseño es lograr un alto grado de flexibilidad de la tarjeta sin tener que hacer cambios en su arquitectura.

Para ello se debió buscar componentes que permitan una amplia gama de aplicaciones a partir de variaciones en su programación. Gracias al carácter genérico del bus PCI, y la flexibilidad del ST-Bus, es posible realizar modificaciones en el enfoque que se le dé al módulo por medio de cambios en los registros de control.

Al finalizar esta etapa se contó con el modelo lógico completo para la distribución de las áreas de memoria, que el programa deberá cargar al iniciar su operación.

### **3.6 Uso del bus secundario**

En cuanto al uso del bus secundario, se trabajó en aspectos relacionados con el tráfico de la información y los tipos de rutinas que el procesador deberá manejar para el traslado de los datos según la situación.

También se investigó un poco sobre el modo de funcionamiento de las unidades para DMA que poseen los chips, y los métodos de traslado de direcciones que utiliza en el acceso del bus interno desde las interfaces PCI.

Se ha definido la forma de trabajar en las dos direcciones de flujo de datos: desde los puertos hacia el PC, y desde el PC dirigidos hacia los puertos ST-Bus.

### **3.7 Programación de encabezados**

Dentro de la estructura diseñada para la tarjeta se incluyó varios componentes de alto nivel de complejidad, que deben ser programadas adecuadamente para su correcta interacción con el medio circundante y entre sí. La base de la tarjeta es el microprocesador, que se debe encargar de comunicar a los otros dispositivos la forma como deben trabajar para que exista compatibilidad y sincronización entre todas las etapas.

El programa del módulo se guardará dentro de una memoria ROM y debe contener las instrucciones adecuadas para que el procesador habilite los recursos y se reporte correctamente a la computadora.

Desde el punto de vista del hardware, la programación de los encabezados se refiere a colocar el valor apropiado en cada uno de los registros internos encargados de controlar las diversas funciones de los componentes.

El primer paso consiste en investigar cuales son los registros del procesador, y cuales deben ser sus valores para que éste trabaje de acuerdo con lo esperado para la aplicación. Una vez que el micro ha sido configurado en todas sus unidades, se procede a escribir en los registros propios de los otros módulos de la tarjeta: espacio de configuración de la RAM y la Flash (en caso de que se utilicen) y control del Freedm.

Aquí no sólo se trata la función de cada registro importante y el valor que se debe asignar, sino que también debe existir un diagrama de flujo que indique, por bloques, los diversos pasos de esta rutina de arranque.

Además, es necesario tener claro el lenguaje sobre el cual se va a programar, el ambiente propio del procesador (código de ensamblador y arquitectura interna) y los requerimientos de programación que se debe respetar.

### **3.8 Documentación**

Paralelamente a todos los pasos del procedimiento, se ha llevado una serie de documentos que integran los resultados obtenidos y la teoría más importante extraída de hojas de datos y otra literatura.

Primero se realizó un documento que resume los aspectos básicos de la empresa y el sistema ITSS, manteniendo cierto nivel de confidencialidad requerido por Cibertec.

Luego, existe una etapa dedicada a los datos principales del estándar PCI, que deben ser dominados y aplicados correctamente al diseño de la tarjeta.

Con respecto a los componentes relacionados con el proyecto, tanto en la tarjeta antigua como en el nuevo diseño, se recopiló las características principales a la vez que se dio el estudio respectivo.

A partir de la información detallada que se encontró en los manuales se hizo una selección de lo más importante para incluirlo en la documentación de la tarjeta. Las características más importantes de los chips se extrajeron de sus hojas de datos, en donde se puede hallar una cantidad mucho mayor de información.

También se documentó los detalles principales sobre la forma de conectar los dispositivos y los cuidados que debe tenerse al diseñar la versión real de la tarjeta; esto paralelamente al avance del modelo en Orcad y los mapas de direcciones.

Para la documentación pedida por el I.T.C.R. se preparó un informe parcial con los resultados obtenidos a la mitad del proceso, a partir del cual se elaboró este informe final, en el cual se conjunta la información de todos los documentos preparados en este proyecto.

### **3.9 Adquisición de licencias**

En las primeras semanas del proyecto se localizó la entidad encargada de la regulación sobre el diseño y uso del bus PCI.

Se procedió entonces a obtener los documentos informativos necesarios, así como una copia de la reglamentación seguida por esta entidad y los formularios que deben ser llenados para la inscripción de una empresa y su equipo PCI.

Esta información fue entregada al coordinador del proyecto en la empresa, mientras que se hacía un análisis para determinar si es necesario pertenecer a esta organización reguladora si se desea diseñar en PCI. En este momento, Cibertec ha decidido incluir la afiliación y adquisición de las licencias dentro del presupuesto del proyecto.

Según los contactos que se establecieron con Intel, no es permitido el uso de los identificadores de Intel que ya posee el procesador, para colocarlos en los encabezados de identificación PnP. Esto está aún sin definir, pues para poder realizar pruebas más específicas es necesario tener un prototipo del módulo.

A pesar de los avances logrados en este punto, los papeles correspondientes a los trámites de inscripción aún no han sido enviados, lo que implica un retraso en el cronograma.

Este procedimiento quedó en manos de la empresa Cibertec, pues se desea esperar hasta verificar en forma práctica si la adquisición del Vendor ID y licencias de la tarjeta son estrictamente necesarias, o pueden ser pasadas por alto.

La metodología de trabajo contempla el proceso de programación que incluye estos valores como parte del código relacionado con PnP, tomando en cuenta que si en algún momento se llega a obtener estos permisos, lo único que debe hacerse es incluir sus valores específicos dentro del código.

### **3.10 Contratación de empresas para el PCB y el ensamble**

Este es un aspecto que no se había contemplado en el anteproyecto, sin embargo la situación se ha prestado para lograr un buen avance en este sentido.

El primer paso fue contactar una empresa en los Estados Unidos, que se especializara en el diseño de circuitos impresos, y que tuviera experiencia en PCI.

Se localizó una compañía llamada Dataline PCB y se entabló un proceso de comunicación con uno de sus diseñadores. Esta empresa finalmente fue contratada por Cibertec para realizar el arte de la tarjeta, a partir del esquemático de Orcad que se había desarrollado.

Por sugerencia de unos compañeros, se contactó con la empresa REMEC, ubicada en la Zona Franca Metropolitana, en Heredia. La posibilidad de contratar una empresa local para este trabajo fue vista con buenos ojos, ya que permite un mayor control sobre lo que se está haciendo.

Se hizo una visita donde se pudo ver la tecnología usada por REMEC para realizar montaje superficial, y se encontró que esta era una excelente alternativa para complementar el trabajo que pudiera hacerse con el impreso en Estados Unidos. Sin embargo en REMEC revocaron su ofrecimiento pues se consideró que el volumen de producción anual no era suficiente para cubrir los gastos.

Entonces se hizo contacto y se visitó la empresa AETEC, también ubicada en la Zona Franca Metropolitana. Esta compañía sí mostró disponibilidad, a pesar del bajo volumen de producción, y se ofreció a realizar el trabajo completo, desde la adquisición de componentes, el diseño del impreso y el montaje de las partes.

Para la cotización final en AETEC se está esperando los archivos gerber que la empresa Dataline genere. Además, fue necesario preparar una serie de documentos que son requisito para la cotización de ensamble.

Una vez finalizado el proceso de cotización, dependerá de Cibertec aceptar o no el contrato para encargar la fabricación del módulo, a partir del diseño completo logrado al finalizar este proyecto.

Existe otra opción para el ensamble, que es contactar con una de las compañías recomendadas por el Ingeniero de Dataline PCB, en lo Estados Unidos.

Estas alternativas para la producción de los prototipos han sido expuestas a los encargados en Cibertec, y próximamente se deberá tomar decisiones ejecutivas para definir hacia dónde se orientará la inversión, considerando los aspectos de calidad, precio y tiempo de entrega.

### **3.11 Transferencia tecnológica**

Durante todo el proceso que se ha llevado a cabo, se ha tenido contacto directo con el Asesor en la Empresa, encargado de vigilar el avance y dirigir los esfuerzos. Los resultados han sido comunicados a dicho asesor para que él a su vez mantenga informados a los superiores. En cuanto a la transferencia hacia los niveles aledaños del proceso productivo, se ha tenido varias reuniones en las cuales se ha discutido la naturaleza del proyecto y su avance.

Es importante que otras personas estén identificadas con el módulo, pues posiblemente será necesaria su intervención, principalmente en el diseño de los controladores, las librerías dinámicas, los programas de aplicación y los probadores.

Una vez finalizada la documentación, se realizará una exposición final con representación de la empresa y el I.T.C.R. en la cual se presentará los resultados conseguidos, y se discutirá los pasos a seguir para llegar a los prototipos, a las pruebas piloto y finalmente, a la primera versión completa y aprobada del módulo.

## **CAPÍTULO 4**

### **DESCRIPCIÓN DEL HARDWARE UTILIZADO**

## 4.1 Introducción al bus PCI

El bus local PCI (iniciales de “Peripheral Component Interconnect”) es el sucesor directo del bus local VESA, y fue creado con la finalidad de disminuir los problemas de los sistemas computacionales, al hacer más eficiente el traslado de la información por medio de un patrón eficiente y duradero, que fuera fácilmente adaptado a las necesidades de las tecnologías nacientes.

La arquitectura del bus PCI fue desarrollada por Intel, y expuesta al mercado en 1993. El bus está orientado para sistemas de quinta y sexta generación, por lo que los últimos diseños de tarjeta madre para procesadores 486 incluyeron características de compatibilidad con PCI, y a partir de entonces prácticamente todos los sistemas de cómputo lo utilizan.

El estándar del bus PCI presenta características de compatibilidad con una alta diversidad de componentes existentes en el mercado, así como una arquitectura abierta para el uso de sus recursos más avanzados en el futuro.

El modelo PCI básico utiliza un bus de 32 bits para el manejo multiplexado de datos y direcciones, a una velocidad de 33 MHz. La arquitectura está pensada para expandir el bus a 64 bits y la frecuencia de operación hasta 66 MHz, con lo que se logra cuadruplicar su capacidad básica en las transacciones de datos.

PCI es una solución para el problema de tráfico de información hacia y desde los puertos. Gracias al incremento en el ancho de banda que proporciona el PCI, se puede enviar hasta 264 MB de datos por segundo. En estos momentos la versión de 64 bits a 66 MHz no está disponible en el mercado.

Un aspecto que hace al bus PCI mucho más flexible que sus antecesores es la separación que existe entre él y el microprocesador del equipo o “Host”.

Existe un circuito controlador de PCI que separa los dispositivos que trabajan en el bus del ambiente local del Host. De esta forma no se restringe la velocidad de los procesadores, ya que el controlador se encarga de sincronizar las transferencias a ambos lados.

Además, gracias a esta característica se puede mantener la compatibilidad, para utilizar dispositivos PCI en equipos de diferentes fabricantes. Por ejemplo, una tarjeta de expansión PCI puede ser usada en una PC de arquitectura IBM o en una PowerPC que sigue los esquemas de Macintosh. El requisito para el diseñador es ubicar un circuito controlador apropiado entre el bus local del microprocesador y el bus PCI.

En estos momentos los fabricantes de procesadores de todo el mundo se han preocupado por suministrar los módulos apropiados para lograr la compatibilidad hacia PCI y por lo tanto no es muy costoso realizar estas conexiones.

La arquitectura convencional para computadores coloca en el bus del Host la mayor parte de sus recursos de memoria: RAM, ROM y caché, así como los controladores principales: PIC, DMA, o otros. En los buses inferiores se ubican los dispositivos periféricos y los elementos de almacenamiento masivo ya sean PCI o no, para los cuales se requiere circuitos interpretes o puentes.

En el diseño de PCI se ha logrado integrar las tarjetas de ampliación en relación con sus direcciones de puertos, canales de DMA e interrupciones. Esto está automatizado de tal manera que el usuario no deberá preocuparse en configurar sus dispositivos.

El bus PCI emplea un conector estilo MicroChannel de 124 pines (188 en el caso de la implementación para 64 bits) pero únicamente 47 de estas conexiones se emplean en una tarjeta de expansión (49 en el caso de un adaptador bus-master).

La gran cantidad de terminales en la ranura se debe a la incorporación de una línea de alimentación y otra de tierra en muchos puntos de la conexión física. Cada una de las señales activas del bus PCI está cerca, junto o frente a una señal de alimentación o tierra, con el fin de minimizar la radiación.

Para generar la señal de reloj, el controlador PCI no utiliza un oscilador propio, sino que toma el reloj del bus local y lo convierte a un valor permitido por el estándar PCI, generalmente entre 20 y 33 MHz. Por lo tanto, aunque la frecuencia de trabajo no está atada directamente al procesador, los datos se mueven sincronizados con él.

Las características del bus PCI se extienden también al nivel de software. Los dispositivos PCI incorporan registros que contienen información relevante a ellos, para que el BIOS y el Sistema Operativo puedan configurar automáticamente los componentes y tarjetas. Esto elimina la necesidad de jumpers y/o software de configuración particular. Además reduce la posibilidad de conflictos por recursos del sistema.

Lo anterior puede considerarse como el principio básico de la metodología denominada “Plug and Play” que ha sido impulsada con el fin de proveer al usuario de computadoras personales un medio fácil, rápido y eficiente de poner a funcional los recursos adicionales que adquiera para su sistema.

Hoy en día, PCI es el modelo que utilizan prácticamente todos los diseños de computadoras como bus principal en el trasiego de información hacia y desde los periféricos. Debe quedar claro que el microprocesador central utiliza su bus local diseñado específicamente para él y las operaciones que requieren alta velocidad (por lo general accesos a memoria RAM y caché), mientras que el bus PCI es la “autopista” de datos cuyo destino u origen es un recurso periférico del sistema.

Los elementos que se diseñen para estándares anteriores a PCI o con un formato derivado como SCSI, deben utilizar un intérprete que permita la compatibilidad. Así, para efectos del CPU la información se estará generando desde o colocando en un recurso PCI, que a su vez se encarga de establecer la comunicación con los otros formatos.

Es importante resaltar que existe una organización mundial encargada de regular el uso del estándar PCI. Este ente regulador se denomina “PCI Special Interest Group” o PCI SIG y está localizada en la ciudad de Portland en Estados Unidos. Cualquier empresa que quiera diseñar siguiendo el estándar PCI deberá inscribirse como un miembro de SIG.

Cada dispositivo que sea diseñado para trabajar en PCI debe estar evaluado y aprobado por la PCI SIG, y se asignará una numeración de identificación especial. Así, no importa la computadora en que el dispositivo sea colocado, el BIOS y el Sistema Operativo deberán ser capaz de identificar de que se trata, su procedencia y los recursos que necesita.

Este trabajo adicional al que se ven obligadas las empresas se hace con el objetivo de facilitar la instalación al usuario y asegurarse que el diseño será funcional en cualquier parte del mundo

## **4.2 Aspectos básicos de la arquitectura PCI**

La característica que convirtió al PC de IBM y compatible en un éxito a escala mundial fue la posibilidad de expandir las capacidades incluyendo nuevos componentes. Esta arquitectura de sistema abierto vino a causar una revolución en el paradigma de las computadoras, su uso y su distribución.

Sin embargo hay dos omisiones en el concepto original:

- No se dieron especificaciones formales para el diseño del hardware y tarjetas de expansión ISA.
- No existió una arquitectura para coordinar la integración de hardware y software en el sistema: asignación de DMA, IRQ, memoria e I/O ports en los dispositivos.

Así, en este punto no existía una interfaz consistente que permitiera al software controlador y sistema operativo determinar los dispositivos instalados y los recursos usados. Con el tiempo esto se convirtió en un factor a tomar en consideración, especialmente en sistemas donde se utilizan componentes móviles como los PCMCIA.

Los componentes que se ubican en la clase de no ser completamente identificable y configurable sin intervención del usuario se denominan dispositivos *Legacy*. Estos componentes en el bus ISA se denominan Tarjetas Legacy.

Para el diseño de un nuevo bus (PCI) fue necesario sobrepasar el método Legacy y desarrollar una nueva metodología de software en el BIOS, el controlador y el Sistema Operativo. Esta metodología se conoce como Plug and Play (PnP).

Un dispositivo PnP incluye un grupo de registros denominados “espacio de configuración” dentro de los cuales se almacena información que el sistema puede utilizar para conocer las características y las necesidades del módulo.

Los requerimientos de PnP no se relegan únicamente a las tarjetas de expansión sino que deben ser seguidos<sup>1</sup> por todos los elementos del sistema con que dispone el usuario: chips, memoria, discos, e incluso periféricos más externos como monitores y teclados.

---

<sup>1</sup> Este “deben ser seguidos” no es tan estricto aún, pero a corto plazo será una realidad en todos los sistemas de cómputo.

En el caso de las tarjetas de expansión PCI, es necesario cubrir todas las necesidades de información que tendrá el BIOS y el Sistema Operativo. El módulo debe estar en capacidad de enviar los datos que lo identifiquen y comunicar al sistema cual es su función y los recursos que necesita para realizarla correctamente, de forma que se acople en forma dinámica al proceso de interacción normal con los otros componentes de la computadora.

Por lo anterior, al estudiar PCI no únicamente se debe tratar los aspectos físicos y eléctricos, sino también una lógica de configuración que juega un papel fundamental en el funcionamiento del nuevo del estándar.

#### **4.2.1 Rendimiento del bus PCI**

El bus PCI provee un rendimiento superior que el bus local VESA. De hecho, PCI es el bus I/O de propósito general con el mejor rendimiento que se utiliza actualmente en computadoras personales.

Esto último se logra gracias a una serie de factores:

- a. Modo de transmisión “Burst” o ráfaga: El bus PCI puede transferir información en modo ráfaga, donde luego de la dirección inicial se realiza la transmisión de un conjunto de datos en fila, sin tener que actualizar el valor de dirección. En estos casos, el modo que internamente se manejará la posición del dato a acceder es configurable, y por lo general se utiliza ascendente, con incrementos de 4 bytes (1 Dword).
- b. Dominio del bus o “Bus Mastering”: El control del bus es cedido a los componentes que tienen la capacidad de dominio, y esto es administrado por una lógica de arbitración, lo que permite utilizar el bus en los procesos de mayor prioridad y mantener los otros procesos en planos inferiores. Así se logra distribuir mejor el tiempo de uso del bus.

- c. Opción de extender el ancho de banda: La especificación del bus PCI, versión 2.1, permite la expansión del bus a 64 bits y de la velocidad a 66 MHz. A pesar de que hasta el momento aún no se explotan estas capacidades, más que todo por cuestiones de compatibilidad, ya se están dando avances que utilizan como base la estructura abierta de PCI, por ejemplo, el uso del nuevo Puerto de Gráficos Acelerados (AGP).

La velocidad del bus PCI puede ser colocada en modo sincronizado o asíncrono, dependiendo del chipset y la tarjeta madre. En el caso sincronizado (que se usa en la mayoría de las computadoras), el bus corre una fracción de la velocidad que lleva el bus de memoria.

En sistemas viejos, donde el bus de memoria es de 50, 60 ó 66 MHz, la velocidad del PCI es de 25, 30 ó 33 MHz respectivamente.

En sistemas asíncronos, la velocidad del PCI se controla independiente del bus de memoria, y su valor se ajusta usando jumper en la tarjeta, o por medio del BIOS.

Si se fuerza la velocidad del procesador (overclocking) en sistemas sincrónicos, de forma que la frecuencia en el bus PCI sobrepasa los 33 MHz, se verá afectada la confiabilidad de los datos proveniente de dispositivos periféricos, y por lo general va a haber problemas de pérdida de información.

#### **4.2.2 Ranuras de expansión PCI**

El bus PCI ofrece más ranuras de expansión que la mayoría de las implementaciones de bus VLB, sin los problemas eléctricos que presenta el bus VESA. La mayoría de sistemas PCI soportan 3 ó 4 slots, mientras que algunos usan más.

Es importante resaltar que algunos sistemas viejos tienen ranuras donde explícitamente se ha acordado no permitir recursos de tipo master, sin embargo esto va en vías de desaparecer.

Queda como responsabilidad de los fabricantes, que en el manual de la tarjeta madre se haga referencia a esta característica.

Además, en el mercado existe una gran cantidad de tarjetas de expansión para PCI, en comparación con las que se diseñan para VLB. Los módulos más comunes son tarjetas de video, adaptadores SCSI, y tarjetas de red de alta velocidad. Los discos duros y unidades de disco se conectan al bus PCI por medio de un adaptador.

Existen algunos elementos como los puertos serie y paralelo que definitivamente no pueden integrarse al bus PCI directamente. Estos componentes del PC pueden permanecer en formato de bus ISA, tomando en cuenta que éste aún está en capacidad de manejar sin problemas la información que se genera hacia y desde ellos.

Hay tres presentaciones de tarjetas PCI: las de 5V, las de 3.3V y las de alimentación dual. Esta característica especial permite a los fabricantes que utilizan tecnologías relativamente viejas trabajar con PCI en formato de alimentación de 5V a la vez que otras empresas se ajustan a tecnologías recientes de 3.3V. Así no es necesario hacer variaciones en el estándar PCI, sino en cada tarjeta de expansión por separado, según la decisión de cada fabricante.

Las tarjetas duales utilizan una característica de la plataforma PCI, en la cual ciertas terminales de alimentación para I/O se usan en componentes que poseen la capacidad de trabajar con ambas tensiones, siempre y cuando se aplique la referencia correcta a una patilla especial de control.

Las ranuras PCI poseen una especie de “llave”, de forma que tarjetas diseñadas para 3.3V no pueden entrar en una ranura que trabaja a 5V y viceversa. Las tarjetas duales presentan ambas llaves, por lo que entran tanto en los slots de 3.3V como en los de 5V.

Para las ranuras de 3.3V, los pines que se ubican en la posición donde se coloca la llave de 5V son GND. De forma análoga, en las tarjetas de 5V, los pines ubicados en la posición de la llave de 3.3V son GND.

Los fabricantes deben seguir estas consideraciones en el momento de diseñar sus tarjetas, según las características de los chips que están usando y los equipos donde se van a conectar. El estándar PCI ha logrado prevenir los problemas de compatibilidad con el diseño de la tarjeta dual, sin embargo la tendencia general es emigrar hacia tecnologías de 3.3V por su bajo consumo energético (menor potencia disipada) y menores problemas por las altas frecuencias.

### **4.2.3 Descripción de señales**

El protocolo PCI define un conjunto de señales que incluyen cuatro clases de señales: datos, direcciones, control y prueba.

Dependiendo si el ancho del bus es de 32 ó 64 bits, la cantidad de señales varía, pero siempre siguen la nomenclatura dada. Además, la función de cada línea puede variar según el tipo de ciclo o la etapa de éste en que se encuentra.

Casi todas las señales trabajan sincronizadas con el reloj, referenciadas a su flanco positivo. En este instante es que, teóricamente, se leen o escriben los datos. Por ello el sistema debe asegurarse de estabilizar todas las señales antes de dicho flanco, para que los dispositivos no reciban datos falsos.

En las siguiente tabla se da una lista de las principales señales en el bus local PCI:

**Tabla 4.1** Descripción de las señales usadas en PCI.

Nombre	Función Principal
AD(x)	Líneas de Direcciones y Datos (Address/Data Lines).
CLK	Reloj (Clock). 33/66 MHz máximo dependiendo de la especificación usada.
C/BE(x)	Comando y Byte Activo (Command, Byte Enable).
FRAME	Usado para indicar si el ciclo está en fase de dirección o en fase de datos.
DEVSEL	Selector de dispositivo (Device Select).
IDSEL	Selector de iniciación de dispositivo (Initialization Device Select).
INT(x)	Interrupción (Interrupt).
IRDY	Iniciador listo (Initiator Ready).
LOCK	Usado para manejar el bloqueo de recursos en el bus PCI.
REQ	Petición (Request). Se usa para pedir el uso del bus PCI.
GNT	Garantía (Grant). Indica que el permiso para usar el bus PCI está garantizado.
PAR	Paridad (Parity). Usado como chequeo de paridad en AD[0..31] y C/BE#[0..3].
PERR	Error de paridad (Parity Error).
RST	Reset.
SBO	Snoop Backoff.
SDONE	Snoop Done.
SERR	Error del sistema (System Error). Indica un error de paridad en la dirección de un ciclo especial, o un error de sistema.
TOP	Petición al Master para detener el ciclo actual. Es manejado por el Target.
TCK	Reloj de prueba (Test Clock).
TDI	Entrada de datos de prueba (Test Data Input).
TDO	Salida de datos de prueba (Test Data Output).
TMS	Selector de modo de prueba (Test Mode Select).
TRDY	Objetivo listo (Target Ready).
TRST	Reset de la lógica de prueba (Test Logic Reset).

El bus PCI trata todas las transferencias como una operación en ráfaga. Cada ciclo comienza con una fase de dirección seguida por una o más fases de datos.

Las fases de datos puede repetirse indefinidamente, pero hay tiempos límites que son llevados por una serie de temporizadores, para que todos los dispositivos tengan un uso racionado del bus.

Estos tiempos los define el CPU dentro del espacio de configuración de cada dispositivo, que tiene su propio temporizador y se basa en el valor del registro Latency Timer en el espacio de configuración.

Las mismas líneas son utilizadas para las direcciones y los datos. Las líneas de comando también son usadas como “byte enables”; es decir, para elegir los bytes válidos del Dword disponible (32 bits). El tipo de transacción que se va a hacer se indica en las líneas de comando durante la fase de dirección.

En el estándar PCI se ha desechado las líneas de control de lectura / escritura y selección entre memoria y espacio de I/O. Por medio de la información enviada a través de las líneas de comando, queda completamente determinado el destino del ciclo y la acción a realizar en este dispositivo.

En la Tabla 4.2 se muestra los tipos de comandos definidos para PCI.

Aquí se muestra cómo este estándar ha ido más allá de simples lecturas y escrituras a los recursos, ofreciendo ciclos de acceso a espacio de configuración, ciclos especiales y comandos reservados para futuras implementaciones en versiones mejoradas del bus.

**Tabla 4.2** Codificación de los comandos en el estándar PCI.

C/BE#	Tipo de comando
0000	Reconocimiento de interrupción
0001	Ciclo especial
0010	Lectura de I/O
0011	Escritura de I/O
0100	Reservado
0101	Reservado
0110	Lectura de memoria
0111	Escritura de memoria
1000	Reservado
1001	Reservado
1010	Lectura de configuración
1011	Escritura de configuración
1100	Lectura múltiple de memoria
1101	Ciclo de dirección dual
1110	Lectura de memoria en línea
1111	Escritura e invalidación de memoria

#### **4.2.4 Ciclos de Bus**

La especificación de bus PCI define la interacción entre dos recursos PCI: el amo (master) y el objetivo (target). El master es un procesador o un bridge actuando por orden de un procesador. El target es un recurso de memoria o de I/O, o un puente que transporta datos a memoria o I/O hacia otro bus. De forma similar a otros buses, en PCI se define un protocolo claro en cuanto al proceso de lectura y escritura de los recursos, además de otras funciones específicas de PCI. En toda plataforma PCI es necesario definir claramente los espacios de memoria y de I/O.

Los ciclos en PCI son mucho más sofisticados que en otros estándares de bus. Se definen ciclos que no se tenían en versiones anteriores, tales como el acceso a espacio de configuración y los ciclos especiales.

Todos los protocolos soportan como mínimo los ciclos sencillos (single) de acceso a datos, sin embargo en PCI se agrega a esto lo que se conoce como acceso en ráfaga o Burst, que implica un bloque de información comunicada a partir un solo comando. En el protocolo PCI es posible adelantar a los recursos de memoria el tipo de acceso que se dará, para que se prepare internamente. Esto permite realizar traslados de datos utilizando líneas de caché múltiples o grandes bloques de bytes, en módulos de memoria diseñados para ello.

Otros de los atributos importantes en la especificación PCI son: uso de caché permitido en los targets de memoria, bloqueo de los targets de memoria por parte de un master, y terminación del ciclo y reintento (retry).

La especificación de PCI soporta 32 bits de datos y 32 de direcciones para ciclos de acceso a memoria, I/O y configuración, ciclos de reconocimiento de interrupción y ciclos especiales. También soporta 64 bits de direcciones para ciclos de acceso a memoria e I/O. Por último, soporta 64 bits de datos para accesos a memoria y (teóricamente) a I/O.

#### **4.2.4.1 Ciclo de reconocimiento de interrupción (0000)**

El protocolo de bus PCI define una versión de lectura denominada *interrupt acknowledge cycle*. Este ciclo opera con el mismo protocolo que una lectura (single o burst). El único recurso PCI que puede responder a este ciclo es aquel que contiene el controlador de interrupciones, que debe residir en un dispositivo del bus 0 o en el bus Legacy en cuyo caso el bridge PCI/Legacy debe encargarse de manejar este ciclo.

Cuando ningún target responde al ciclo, el master debe ejecutar un procedimiento de aborto. La culminación de un reconocimiento de interrupción con un retry, desconexión o aborte desde el target requiere que el master desactive su señal REQ#. El protocolo exige que esta señal permanezca desactivada un mínimo de dos periodos de CLK relativos al final del ciclo que no se pudo finalizar.

Durante este ciclo, el controlador de interrupciones reconoce automáticamente y reacciona al comando INTA. En la fase de datos, transfiere el vector de interrupción a las líneas AD. Este dato es tomado luego por el CPU para buscar la locación donde se almacena la rutina de atención.

#### **4.2.4.2 Ciclo especial (0001)**

En PCI existe una versión del ciclo de escritura llamado ciclo especial. Opera con el mismo protocolo que una escritura común (single o burst) excepto porque ninguno de los targets clama por el ciclo. Durante esta operación, se permite la escritura de un mensaje a todos los recursos del bus a la vez. La fuente de este mensaje es un master en el bus o un bridge que trasmite el ciclo desde otro bus.

Para la traslación de ciclos de especiales a través de un bridge, se utiliza una variación de ciclo de configuración denominado tipo 1.

Cuando un ciclo especial se está ejecutando, ningún recurso puede reclamar el bus. Consecuentemente las señales DEVSEL#, TRDY#, STOP# y AD# no son manejadas por los targets, sino que se mantienen en estado alto por medio de resistencias de pull-up.

El tipo de ciclo especial está dado por el valor las líneas AD[0..15] en la fase de datos, tal como se muestra en la Tabla 4.3.

**Tabla 4.3** Ciclos especiales en el bus PCI.

AD[0..15]	Tipo de ciclo especial
0x0000	Processor Shutdown
0x0001	Processor Halt
0x0002	x86 Specific Code
0x0003 to 0xFFFF	Reservados

#### **4.2.4.3 Lectura de I/O (0010) y escritura de I/O (0011)**

Este comando permite lectura o escritura de datos en dispositivos de I/O. Las líneas AD contienen la dirección de un byte (AD0 y AD1 deben ser decodificadas), de forma que los puertos I/O para PCI son de 8 ó 16 bits. El espacio permitido para alojar puertos de I/O es de 32 bits (4GB).

En las máquinas compatibles IBM, los CPU de Intel están limitados a 16 bits de espacio I/O (64KB). Tomando en cuenta que varias tarjetas ISA pueden estar usando mucho de este campo, los puertos de I/O están muy restringidos en este modelo de computadora.

PCI ha debido ajustarse a estos requerimientos y existen componentes especiales para manejar puertos de I/O en buses Legacy. De todas formas, cualquier bus PCI está en capacidad teórica de acceder a 32 bits de direcciones I/O.

El espacio de configuración PCI puede ser accesado también como puertos I/O en 0x0CF8 (Address) y 0x0CFC (Data). El puerto de dirección debe ser escrito primero.

#### **4.2.4.4 Lectura de memoria (0110) y escritura de memoria (0111)**

Son utilizadas por un master para realizar lecturas y escrituras al espacio de memoria. Las líneas AD contienen una dirección de Dword (AD0 y AD1 no necesitan ser decodificadas). Las líneas de habilitación de byte (C/BE#) indican cual bytes son válidos.

En PCI el espacio de memoria está claramente separado del espacio de I/O, lo que permite direccionar puertos y locaciones de memoria con la misma posición. Por lo general las lecturas a memoria son más veloces, y se pueden ejecutar en modo de ráfaga.

Es posible sin embargo, realizar accesos a espacio de I/O mapeado en memoria. En estos casos los registros de entrada de los puertos se leen o escriben con comandos para memoria, pero no se puede efectuar transmisión en ráfaga. Por ello en muchos dispositivos se divide el espacio de memoria en bloques, algunos de los cuales se marcan como contenedores de registros I/O y otros no, para utilizar ráfaga en estos últimos.

#### **4.2.4.5 Lectura (1010) y escritura (1011) de configuración**

Estos comandos permiten realizar lecturas y escrituras al espacio de configuración de los dispositivos, que tiene un largo de 256 bytes. Este es accesado en Dwords, de tal forma que AD0 y AD1 son cero, mientras AD[2..7] contienen la dirección y AD[8..10] son usadas para seleccionar la función en dispositivos multi-función. Las otras líneas no se utilizan.

Estos ciclos por lo general se realizan únicamente en el proceso de arranque y distribución de los recursos. Por medio de estos comandos es que se puede configurar los dispositivos automáticamente según los requerimientos de Plug and Play.

En cuanto al uso de las líneas de control se refiere, estos ciclos trabajan de forma similar a los de acceso a memoria o a puertos, sin embargo el espacio de configuración es pequeño, y algunas direcciones son sólo de lectura.

#### **4.2.4.6 Ciclo de dirección dual (1101)**

Un ciclo de dirección dual es necesario si se va a direccionar en 64 bits usando un bus físico de 32. La parte menos significativa de la dirección se envía primero en las líneas AD, y en la fase siguiente se coloca la parte más significativa. El segundo ciclo de dirección además contiene el tipo de comando para la transferencia. El bus PCI soporta un espacio de dirección I/O de 64 bits, aún cuando esto no es factible cuando se utiliza arquitectura de PC basada en Intel, dadas las limitaciones del CPU.

#### **4.2.4.7 Memory-Read Line (1110)**

Este ciclo es usado para leer más de dos bloques de 32 bits, típicamente sobre el final de una línea de caché. Esto es más eficiente que la lectura de memoria normal en ráfaga para una serie de accesos secuenciales a memoria.

#### **4.2.4.8 Memory Write and Invalidate (1111)**

Esto indica que un mínimo de una línea de caché será transferida. Esto permite a la memoria principal ser actualizada, salvando un ciclo de sobre-escritura de caché.

### **4.3 Arquitectura genérica de una plataforma PCI**

En la Figura 4.1 se muestra la plataforma completa PCI más sencilla que se puede encontrar, con tres buses. El bus Host es el de mayor eficiencia y enlaza los recursos primarios de la plataforma. Para una PC, el bus Host es del tipo x86 de Intel. El bus Legacy es un bus de eficiencia mediana o baja que permite adaptar funciones diversas al sistema. Entro estos buses Legacy se encuentran ISA, EISA, y MicroChannel. El bus PCI proporciona una eficiencia media.

Cada bus tiene su propósito definido dentro de la arquitectura:

- El Host bus provee un lazo de alta velocidad entre el CPU, la memoria (SDRAM en los sistemas actuales), y el caché (SRAM de alta velocidad). Típicamente estos buses no son cargados fuertemente, y el largo de sus líneas es muy corto, para minimizar las pérdidas por la alta frecuencia de operación.
- El bus Legacy provee recursos complementarios a los que se hayan en el Host bus. Puede contener otros CPUs como bus masters, memoria adicional y puertos de I/O. Se conectan por medio de pistas más largas y contienen soporte para ranuras de expansión. En una arquitectura de PC, el espacio de I/O también contiene los controladores para DMA, interrupciones, teclado y mouse. Además, se puede incluir el reloj de tiempo real, la interfaz de video y en el caso de EISA y MicroChannel, los registros de configuración.
- El bus PCI provee recursos adicionales complementarios al Host. Similar a los buses Legacy, contiene otros procesadores (masters), memoria y puertos I/O. En la nomenclatura de PCI los dispositivos de memoria e I/O se denominan targets.

Muchos de los recursos que aparecen en este modelo dentro de los buses de Host y Legacy están emigrando rápidamente hacia PCI. Este es el caso del controlador de video, que inicialmente se colocaba en los buses Legacy pero ahora aprovecha las capacidades de PCI y las ha utilizado en una extensión especial: el bus AGP. La migración de dispositivos de un bus a otro requiere de un sistema de locación y trasladación de direcciones para que se mantenga la compatibilidad y los programas no tengan problema al ejecutarse en máquinas con diferentes esquemas de distribución.

La circuitería utilizada para comunicar estos buses diferentes es llamada Puentes o “Bridges”. Su principal función es intercomunicar un protocolo de bus con otro distinto. Este protocolo incluye la definición de las líneas de control, de datos y de direcciones. Cuando un puente sirve de canal entre dos buses idénticos, su función principal es limitar la carga de cada bus.

El punto focal de toda la estructura es el bus de Host, por lo cual la forma en que se enlazan los otros buses a éste es uno de los aspectos más importante para el diseñador de computadoras. El CPU tiene por lo general la mayor eficiencia y es el amo (master) de mayor rango, quien supervisa las interrupciones y los errores.

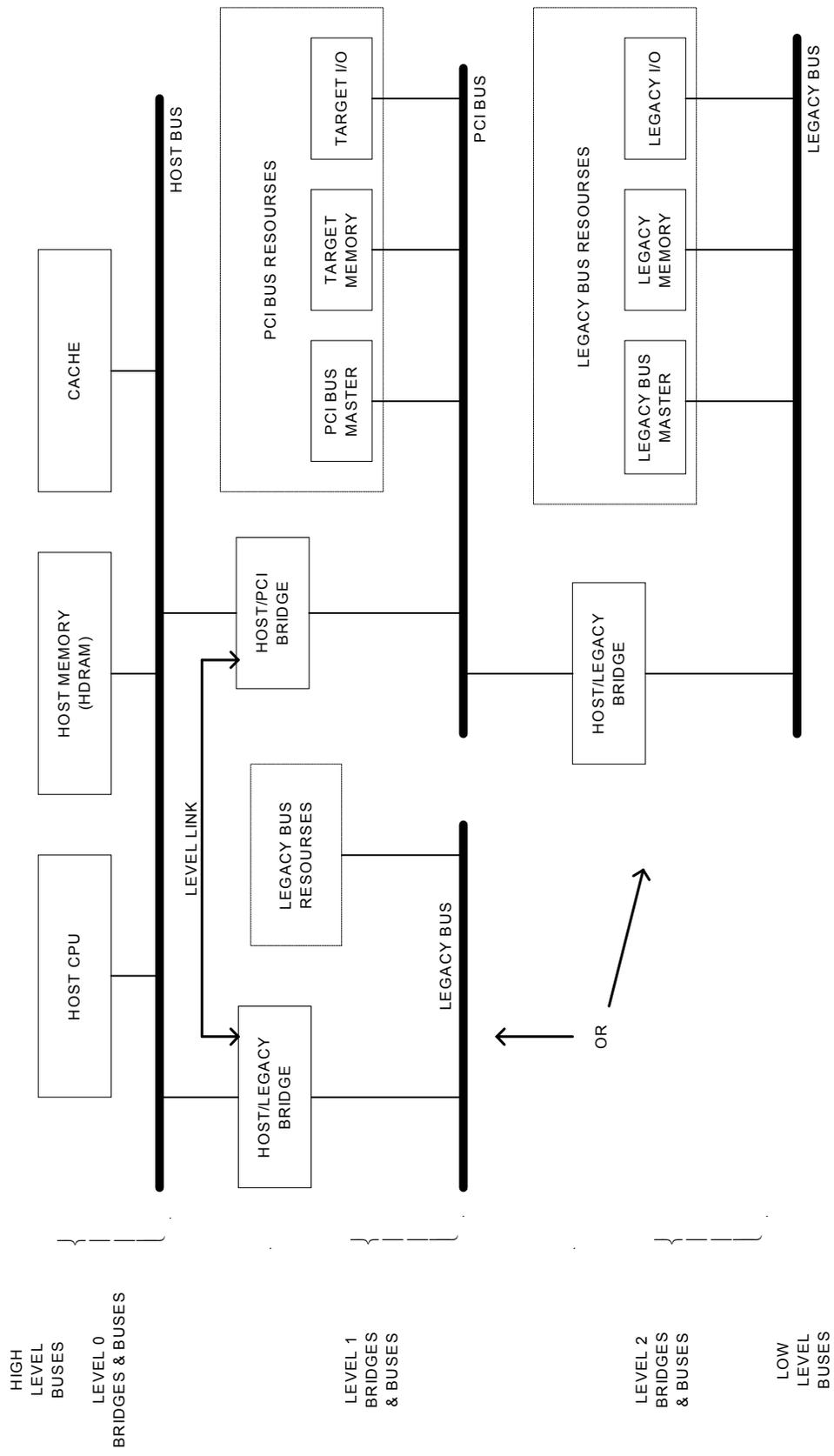
A partir de esta estructura básica, se puede agregar niveles inferiores de buses PCI, por medio de puentes de acople PCI/PCI.

Cada bus PCI dentro de la computadora lleva asignado un número que lo identifica. Este valor es asignado en los procedimientos de arranque según una convención especial. Como convenio se utiliza un algoritmo que recorre los ramales de jerarquía de un bus antes de asignar valores a los otros buses que están al mismo nivel.

#### **4.3.1 Buses fuente y destino**

Para describir el protocolo del bus a través de un bridge se usa la convención de nombres “source” y “destination” o fuente y destino.

La fuente es el bus que contiene el master mientras que el destino por su parte contiene el objetivo o target. Este acuerdo ayuda en la descripción del flujo de señales cuando hay trasiego de datos.



**Figura 4.1** Arquitectura de Plataforma Genérica con un solo bus PCI.

En el caso de un ciclo iniciado desde un nivel superior hacia un recurso de bus ubicado más abajo en la jerarquía, el bus fuente se encuentra arriba del bus destino. También se puede dar el caso contrario, por ejemplo cuando algún dispositivo periférico desea obtener o escribir un dato en la memoria del Host, la fuente del ciclo se encuentra en los niveles inferiores, mientras que el destino será la memoria principal de la computadora.

#### **4.3.2 Propiedad del bus PCI**

Cuando se va a utilizar el bus PCI por algún dispositivo, este debe tomar el control de las señales. Sólo los circuitos capacitados para funciones de bus master pueden realizar esta tarea. Cuando un master toma propiedad del bus (una vez que el árbitro lo autorice) es el único recurso utilizando el bus y la única entidad con permiso para ejecutar ciclos de bus.

Un bus master debe ser un circuito inteligente, que realice acciones coherentes en el bus, o un bridge actuando en representación de un master que reside en otro bus. Es cualquiera de estas dos situaciones la operación del bus es la misma.

En la arquitectura típica, el Host funciona como un master especial. Es el primer master en controlar el bus cuando el sistema se activa, es quien se encarga de inicializar toda la plataforma y es el único autorizado para realizar rutinas de atención de interrupciones y errores del PC.

#### **4.3.3 Recursos PCI, dispositivos y carga del bus**

Cada bus PCI puede soportar 10 cargas, que se definen como un solo CI (chip) que sigue los requerimientos eléctricos de PCI. Siempre que un CI se conecte vía ranura de expansión (slot), se debe considerar una carga por él y otra por el conector.

En forma colectiva, los chips y las tarjetas de expansión son vistas por el software como “devices” o dispositivos. Cada dispositivo contiene de 1 a 8 espacios de configuración. Cada espacio de configuración representa una función determinada en el trabajo del chip o del módulo. Es posible encontrar circuitos integrados con varias funciones, cada una de las cuales puede trabajar de forma independiente, alternándose el control del bus.

Desde el punto de vista del hardware, todos los elementos en el sistema PCI son “resources” o recursos. Los tres tipos son: bus master, bridge y target.

El master es quien posee el bus y ejecuta el ciclo de bus. El puente es una circuitería que interconecta dos estructuras de bus. Cuando un ciclo implica un bus destino, el puente se convierte en master de éste. Los targets son los recursos de memoria o I/O disponibles.

Existen 21 líneas de señal (IDSEL) usadas en cada bus PCI para seleccionar un dispositivo durante los ciclos de configuración. Cada línea corresponde a un único dispositivo y sólo esta se mantiene en alto cuando se va a leer o escribir el espacio de configuración de este elemento.

Un circuito “single-function” es un chip o tarjeta que requiere un IDSEL y tiene un espacio de configuración. Si es “multi-function”, tiene un IDSEL pero contiene dos áreas de configuración, de forma que la función particular en este caso se determina por el valor de ciertos bits en el comando utilizado.

#### **4.3.4 Central Resource**

Este circuito contiene soporte no asignado a otros recursos de plataforma. Contiene la circuitería que controla el Reset, el reloj y los estados de error que se producen. Aquí se generan señales auxiliares como CLK, RST#, PERR#, SERR#, y otras.

Además se encarga de controlar el acceso a memoria cache, cuando se utiliza esta opción por medio de las líneas SBO# y SDONE, dado en caso que no exista un controlador de cache en el puente Host/PCI. Como componente del bus PCI no desempeña ninguna función relacionada con el tráfico de datos, sino más bien con respecto al estado del bus y su funcionamiento normal.

#### **4.3.5 Árbitro Central**

Consiste en una circuitería que maneja buses PCI y Legacy en ciertas funciones. En PCI es quien controla REQ#, que es la petición del bus y GNT# que es la garantía de propiedad para los master. Aquí se manejan algoritmos de prioridad y tiempo para ceder el control del bus.

La etapa Legacy forma parte del bridge PCI/Legacy y depende del tipo de bus. Aquí se produce la señalización para manejar estos buses.

El control del bus no puede ser tomado por un master hasta que el árbitro le dé permiso por medio de la activación de la señal GNT# correspondiente. Por otro lado, cada master debe activar el REQ# siempre que desee usar el bus, para que el árbitro decida si puede o no.

Por medio de un algoritmo de tiempo y primacía, el circuito árbitro se encarga de hacer más fluido el tráfico de información de alta prioridad, y deja pequeños intervalos para los datos relacionados con dispositivos de baja importancia. Es necesario que el control del bus no permanezca mucho tiempo en poder de un solo master, en caso que haya otros requiriendo atención.

La otra función del árbitro es actuar como Park Master, o sea, control de las señales que deben estar asociadas a un valor lógico aún cuando ningún otro elemento las está usando.

En el protocolo PCI se persigue que el tiempo de una señal está flotante sea mínimo, así que en ciertos casos se hace necesario que un circuito quede encargado de llevarlas a un valor determinado si no son utilizadas. En algunas arquitecturas esta es función del árbitro, aunque también se acostumbra colocar resistencias de Pull-Up.

#### **4.3.6 Señales de interrupción**

La estructura de las interrupciones de PCI debe utilizarse con el controlador PCI 8259 para optimizar la compatibilidad.

El concepto de interrupciones sensibles al nivel se aplica en PCI. Cuatro líneas de interrupción son definidas: INTA#, INTB#, INTC# e INTD#, que en su conjunto se denominan INTx# y son conectadas a una circuitería compatible con el 8259. Las interrupciones más comunes dentro del PC son las del bus: 9, 10, 11 y 15.

Existen algunos requerimientos de protocolo para implementar las líneas de señal INTx#:

- Algunos dispositivos (cada uno con configuración individual) pueden ser conectados a una sola INTx#. Esto se denomina “interrupt sharing”.
- Un dispositivo es “cable-ORed” a una línea INTx#. Esto se logra con salidas son open-colector en todos los candidatos a interrupción, y quiere decir que la interrupción se activa con la petición de uno o más elementos.
- Una vez que un dispositivo a activado una señal INTx#, debe permanecer encendida hasta que la rutina de atención específica ordene su desactivación.

- Un circuito integrado de tipo “single-device” en la plataforma, o una tarjeta de expansión que requiere sólo una interrupción, debe usar únicamente la señal INTA#. Cada chip o tarjeta de expansión con al menos un dispositivo que requiera una rutina de atención de interrupción debe usar la línea INTA# antes de cualquier otra. Los dispositivos adicionales en el chip o la tarjeta pueden usar también INTA# o recurrir a una de las otras señales INTx#.
- Cada dispositivo individual puede pedir servicio en una de las líneas INTx#. No le está permitido solicitar atención en dos o más líneas a la vez.

#### **4.3.7 Líneas de señal tipo “Sideband”**

La especificación PCI permite la definición de líneas entre componentes que son propietarios de ellas. Estas líneas se definen como “Sideband” y no son parte del protocolo PCI. Consecuentemente sólo pueden estar definidas para los recursos PCI ubicados en el mismo plano físico; o sea, no pueden atravesar un conector PCI o un circuito bridge.

El uso de estas señales queda determinado por las características de los dispositivos que las van a manejar, y son responsabilidad de los mismos. Un ejemplo de estas señales son las líneas de control de memoria local, que puede tener un microprocesador auxiliar dentro de un bus PCI.

#### **4.3.8 Señales J-TAG**

Estas líneas no están definidas como parte del bus PCI y son opcionales para los recursos compatibles con PCI. El objetivo de estas señales es proveer detección de presencia de acuerdo con la norma IEEE 1149.1. Si un dispositivo se acoge a este estándar, entonces debe estar diseñado para soportar las señales TCK, TDI, TDO, TMS y opcionalmente TRST#.

Las líneas J-TAG no son activas cuando el recurso PCI se encuentra en un ciclo de bus, y deben corresponder a los requerimientos eléctricos, ya sea en una plataforma de 5V o en una del 3.3V. Forman parte del conector PCI, pero no necesariamente poseen las mismas características de carga que las otras líneas.

TCK, TMS y TRST# se conectan en modo común a todas las cargas que las acepten dentro del bus. Por su parte, la línea TDI de un recurso es conectada al TDO del siguiente, sucesivamente hasta formar una cadena cerrada dentro de la plataforma.

Adicionalmente a la normativa dada por IEEE, en la Guía de Diseño para Sistemas PCI y la Especificación 2.0 del Bus Local PCI se sugiere los siguientes usos para las líneas J-TAG:

- Usar el anillo de chequeo (test ring) en la tarjeta de expansión únicamente durante las pruebas de manufactura.
- Crear un anillo de chequeo en la plataforma principal y enlazarlo con los de las tarjetas de expansión.
- Los circuitos integrados pueden soportar direccionamiento hereditario “multi-drop”, como se especifica en la norma IEEE 1149.1.

Para los circuitos de aplicación estas señales no son importantes, sin embargo cuando no se utilizan el diseñador debe asegurarse que se encuentren correctamente dirigidas a tensiones de referencia por medio de resistencias pull-up o pull-down.

El uso de este pequeño bus permite chequear la integridad del componente por medio de rutinas de “test”.

#### **4.4 Requerimientos eléctricos del bus PCI**

PCI tiene sus especificaciones eléctricas bien definidas. Aún si algunas de las determinaciones pueden parecer un obstáculo más para los diseñadores, esto es necesario para promover la interoperabilidad de los componentes y las tarjetas PCI. Todas las especificaciones eléctricas para PCI han sido escritas tomando en cuenta la operación conjunta del sistema.

Los requerimientos eléctricos de PCI intentan satisfacer una alta variedad de posibles aplicaciones de mercado, sin colocar tantas normas que pueda intimidar la creatividad de los diseñadores. Esto se demuestra con la extensión natural del estándar a 64 bits, o el uso de tecnología de 3.3V. Casi todos los buses utilizan hoy día lógica compatible TTL de 5V, mientras que PCI ha anticipado (y promovido) el uso de lógica a 3.3V. Desdichadamente estas dos familias no se mezclan bien juntas, lo que conlleva a una serie de especificaciones distintas para PCI en 5V la versión de 3.3V.

##### **4.4.1 Distinción entre tarjetas de expansión según el voltaje del bus**

El bus PCI tiene dos niveles distintos de señalización: existe la versión de 3.3V y la de 5V. Estas familias basan su lógica en valores de tensión signados al 1 binario, según su nombre lo indica; o sea, con una referencia diferente en ambos casos.

Es importante aclarar que es posible encontrar dispositivos con un determinado valor de tensión en su alimentación, y otra referencia distinta para el manejo del bus PCI. Por lo tanto no se debe confundir la alimentación del dispositivo con la referencia que utiliza el ambiente del bus PCI, pues aunque pueden coincidir, también pueden ser diferentes.

El ambiente de señalización entre una tarjeta y su plataforma debe coincidir, ya que los componentes no soportan la mezcla entre las tensiones de referencia.

Esto quiere decir que si el ambiente de un bus PCI (todas las señales que forman parte del estándar del bus) está diseñado para trabajar a 5V, los dispositivos conectados a él deben ajustarse a este valor de referencia en sus terminales. De otra forma se producirá una incompatibilidad que puede dañar los elementos conectados al bus. Lo mismo se aplica para un ambiente de 3.3V al cual se conecta un recurso de 5V.

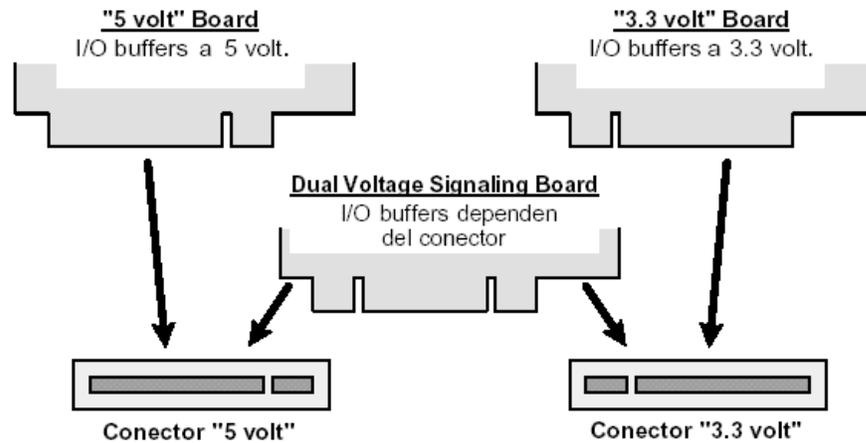
Para disminuir un poco el impacto de esta situación, algunos circuitos integrados utilizan terminales separados para la alimentación principal y para la referencia (VIAS) del bus PCI. Así el diseñador tiene la libertad de elegir el tipo de ambiente que va a utilizar, o dejar esto abierto para que la plataforma domine este factor.

Los conectores PCI están diseñados con una llave mecánica, para que una tarjeta de 5V no pueda ser introducida en una ranura de 3.3V y viceversa. Las tarjetas diseñadas para adaptarse al ambiente de la plataforma, por medio de la alimentación separada (VIAS), se denominan duales, y poseen tanto la llave de 5V como la de 3.3V.

En la ranura PCI existe un conjunto de líneas de alimentación especial denominadas +I/O V. La tensión en estas líneas depende exclusivamente del tipo de ambiente del bus, por lo que pueden ser utilizadas para alimentar los dispositivos que soportan dualidad en su funcionamiento. Al conectar +I/O V en la terminal VIAS, el diseñador se asegura que el ambiente del bus PCI local va a mantenerse con la misma referencia que el de la plataforma.

Aunque el uso de tarjetas duales permite un mayor rango de aplicación, los componentes que poseen este control son más caros, por lo que en muchos casos se siguen fabricando módulos compatibles únicamente con un nivel específico de tensión. La tendencia general en la industria es emigrar hacia el bus de 3.3V, porque permite un menor consumo de energía y menores problemas por la radiación y la reflexión.

En la Figura 4.2 se muestra la apariencia física, tanto de las ranuras como de las tarjetas PCI, según la característica de alimentación particular del bus.



**Figura 4.2** Tipos de ranuras y tarjetas PCI, según su llave específica.

#### 4.4.2 Señales de reloj

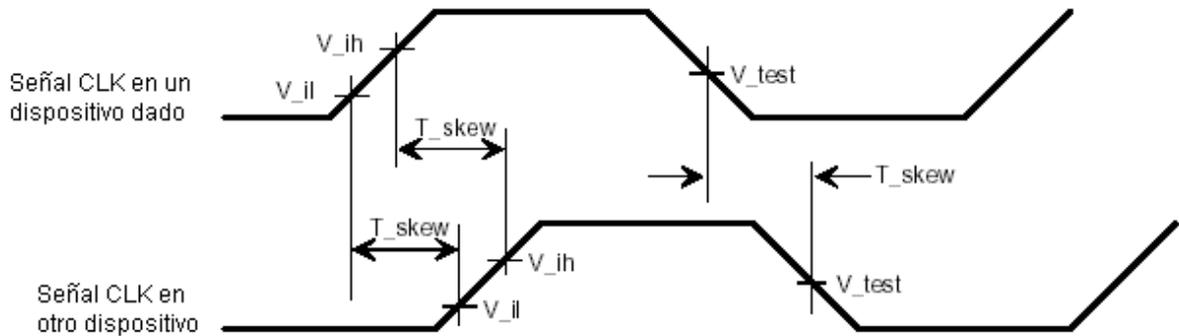
Todas las señales del bus PCI están sincronizadas con el flanco positivo del reloj principal (CLK) excepto las líneas RST#, SERR# e INTx#.

Un sistema PCI de 33 MHz posee un periodo de reloj de 30 ns. La especificación PCI requiere que sus componentes estén capacitados para trabajar correctamente a 33 MHz.

El ciclo de trabajo de la señal de reloj está definido de tal forma que el tiempo que la señal puede permanecer en bajo ( $T_{low}$ ) y el tiempo que puede permanecer en alto ( $T_{high}$ ) debe ser superior a 11 ns. Si el sistema trabaja a menos de 33 MHz, estos tiempos pueden ampliarse.

El reloj PCI debe ser generado y distribuido cuidadosamente sobre la plataforma y las tarjetas de expansión para mantener el desfase ( $t_{skew}$ ) lo más pequeño posible.

Como se muestra en la Figura 4.3 todos los componentes PCI deben obtener una copia del reloj con una diferencia de tiempo menor a 2 ns. El tiempo de desfase se mide entre los pines de los componentes.



**Figura 4.3** Desfase en la señal de reloj PCI de dos dispositivos.

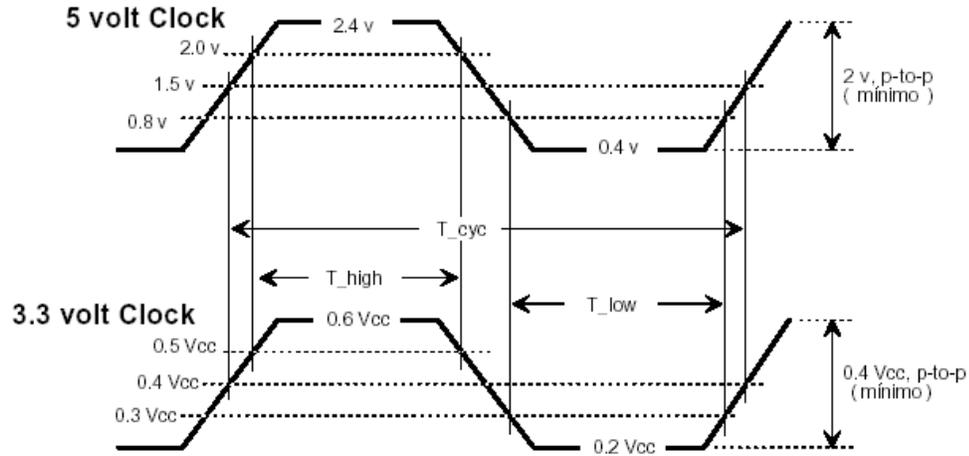
En la Tabla 4.4. se muestra las condiciones sobre las cuales se basa la recomendación para este límite de desfase.

**Tabla 4.4** Condiciones de medición usadas para determinar  $T_{skew}$ .

Símbolo	Señales a 5V	Señales a 3.3V	Unidades
$V_{TEST}$	1,5	0,4 $V_{CC}$	V
$T_{SKEW}$	2 (max)	2 (max)	ns

Ahora, es importante diferenciar los niveles que debe alcanzar una señal de reloj para ambiente de 5V y para ambiente de 3.3V. En la Figura 4.4 se compara las señales de reloj para estos dos casos.

Por su parte, en la Tabla 4.5 se tabulan las especificaciones temporales básicas que dichas señales deben satisfacer.



**Figura 4.4** Formas de onda esperadas para los relojes PCI.

**Tabla 4.5** Requerimientos temporales básicos de los relojes PCI, @ 33 MHz.

Símbolo	Parámetro	Mínimo	Máximo	Unidades
$T_{CYC}$	Periodo de CLK	30	$\infty$	ns
$T_{HIGH}$	Tiempo en alto de CLK	11		ns
$T_{LOW}$	Tiempo en bajo de CLK	11		ns
-	Taza de desfase de CLK	1	4	V / ns
-	Taza de desfase de RST#	50	-	mV / ns

#### 4.4.3 Resistencias de Pull-Up

Algunas señales del bus PCI requieren resistencia de pull-up para mantener un nivel lógico cuando se colocan en estado flotante. El valor de estas resistencias se resume en la Tabla 4.6

Para el cálculo del valor máximo se utiliza  $I_{ih}$  que no debe ser mayor a 70  $\mu$ A. El número de cargas no debe sobrepasar las 16.

**Tabla 4.6** Resistencias de Pull-Up para señales del bus PCI.

Nivel de tensión del ambiente	MIN ( $\Omega$ )	MAX ( $\Omega$ )
5V	963	$\frac{[V_{CC\_MIN} - 2.7]}{[\#cargas \times I_{ih}]}$
3.3V	2.42 K	$\frac{[V_{CC\_MIN} - 0.7V_{CC}]}{[\#cargas \times I_{ih}]}$

Las señales del bus que requieren pull-ups son: FRAME#, TRDY#, IRDY#, DEVSEL#, STOP#, SERR#, PERR#, LOCK#, INTx# y PME#. Si el bus es de 64 bits, también debe incluirse en esta lista las siguientes líneas: AD[32..63], C/BE[4..7]#, PAR64#, REQ64# y ACK64.

La unidad central de recursos de la plataforma puede requerir pull-ups en las señales REQ#, en cuyo caso las especificaciones para el valor asignado son independientes.

#### 4.4.4 Distribución de los pines en la ranura y los dispositivos

Una de las características eléctricas del bus PCI es la gran cantidad de pines asignados a señales de alimentación. Estas líneas se han distribuido de forma que se intercalan con las señales del bus, para disminuir el ruido y las pérdidas por radiación.

Como ya se señaló, hay diferencias entre las tarjetas de extensión de 3.3V y de 5V. Estas diferencias no sólo se reflejan en la llave mecánica, sino que también afectan las líneas de alimentación que llega a las tarjetas.

En las siguientes Tablas se muestra la distribución de señales en las dos clases de ranura PCI que se puede encontrar.

**Tabla 4.7** Distribución de pines en una ranura PCI.

Pin	Sistema de 5V		Sistema de 3.3V	
	Cara B	Cara A	Cara B	Cara A
Inicio de la sección de 32 bits				
1	-12V	TRST#	-12V	TRST#
2	TCK	+12V	TCK	+12V
3	GND	TMS	GND	TMS
4	TDO	TDI	TDO	TDI
5	+5V	+5V	+5V	+5V
6	+5V	INTA#	+5V	INTA#
7	INTB#	INTC#	INTB#	INTC#
8	INTD#	+5V	INTD#	+5V
9	PRSNT1#	Reserved	PRSNT1#	Reserved
10	Reserved	+5V (I/O)	Reserved	+3.3V (I/O)
11	PRSNT2#	Reserved	PRSNT2#	Reserved
12	GND	GND	3.3V KEY	
13	GND	GND		
14	Reserved	3.3Vaux	Reserved	3.3Vaux
15	GND	RST#	GND	RST#
16	CLK	+5V (I/O)	CLK	+3.3V (I/O)
17	GND	GNT#	GND	GNT#
18	REQ#	GND	REQ#	GND
19	+5V (I/O)	PME#	+3.3V (I/O)	PME#
20	AD[31]	AD[30]	AD[31]	AD[30]
21	AD[29]	+3.3V	AD[29]	+3.3V
22	GND	AD[28]	GND	AD[28]
23	AD[27]	AD[26]	AD[27]	AD[26]
24	AD[25]	GND	AD[25]	GND
25	+3.3V	AD[24]	+3.3V	AD[24]
26	C/BE[3]#	IDSEL	C/BE[3]#	IDSEL
27	AD[23]	+3.3V	AD[23]	+3.3V
28	GND	AD[22]	GND	AD[22]
29	AD[21]	AD[20]	AD[21]	AD[20]
30	AD[19]	GND	AD[19]	GND

**Tabla 4.8** Distribución de pines en una ranura PCI (continuación).

Pin	Sistema de 5V		Sistema de 3.3V	
	Cara B	Cara A	Cara B	Cara A
31	+3.3V	AD[18]	+3.3V	AD[18]
32	AD[17]	AD[16]	AD[17]	AD[16]
33	C/BE[2]#	+3.3V	C/BE[2]#	+3.3V
34	GND	FRAME#	GND	FRAME#
35	IRDY#	GND	IRDY#	GND
36	+3.3V	TRDY#	+3.3V	TRDY#
37	DEVSEL#	GND	DEVSEL#	GND
38	GND	STOP#	GND	STOP#
39	LOCK#	+3.3V	LOCK#	+3.3V
40	PERR#	SDONE	PERR#	SDONE
41	+3.3V	SBO#	+3.3V	SBO#
42	SERR#	GND	SERR#	GND
43	+3.3V	PAR	+3.3V	PAR
44	C/BE[1]#	AD[15]	C/BE[1]#	AD[15]
45	AD[14]	+3.3V	AD[14]	+3.3V
46	GND	AD[13]	GND	AD[13]
47	AD[12]	AD[11]	AD[12]	AD[11]
48	AD[10]	GND	AD[10]	GND
49	GND	AD[09]	M66EN	AD[09]
50	5V KEY		GND	GND
51			GND	GND
52	AD[08]	C/BE[0]#	AD[08]	C/BE[0]#
53	AD[07]	+3.3V	AD[07]	+3.3V
54	+3.3V	AD[06]	+3.3V	AD[06]
55	AD[05]	AD[04]	AD[05]	AD[04]
56	AD[03]	GND	AD[03]	GND
57	GND	AD[02]	GND	AD[02]
58	AD[01]	AD[00]	AD[01]	AD[00]
59	+5V (I/O)	+5V (I/O)	+3.3V (I/O)	+3.3V (I/O)
60	ACK64#	REQ64#	ACK64#	REQ64#

**Tabla 4.9** Distribución de pines en una ranura PCI (continuación).

Pin	Sistema de 5V		Sistema de 3.3V	
	Cara B	Cara A	Cara B	Cara A
61	+5V	+5V	+5V	+5V
62	+5V	+5V	+5V	+5V
Final de la sección de 32 bits (64 bits KEY)				
Inicio de la sección de 64 bits (64 bits KEY)				
63	Reserved	GND	Reserved	GND
64	GND	C/BE[7]#	GND	C/BE[7]#
65	C/BE[6]#	C/BE[5]#	C/BE[6]#	C/BE[5]#
66	C/BE[4]#	+5V (I/O)	C/BE[4]#	+3.3V (I/O)
67	GND	PAR64	GND	PAR64
68	AD[63]	AD[62]	AD[63]	AD[62]
69	AD[61]	GND	AD[61]	GND
70	+5V (I/O)	AD[60]	+3.3V (I/O)	AD[60]
71	AD[59]	AD[58]	AD[59]	AD[58]
72	AD[57]	GND	AD[57]	GND
73	GND	AD[56]	GND	AD[56]
74	AD[55]	AD[54]	AD[55]	AD[54]
75	AD[53]	+5V (I/O)	AD[53]	+3.3V (I/O)
76	GND	AD[52]	GND	AD[52]
77	AD[51]	AD[50]	AD[51]	AD[50]
78	AD[49]	GND	AD[49]	GND
79	+5V (I/O)	AD[48]	+3.3V (I/O)	AD[48]
80	AD[47]	AD[46]	AD[47]	AD[46]
81	AD[45]	GND	AD[45]	GND
82	GND	AD[44]	GND	AD[44]
83	AD[43]	AD[42]	AD[43]	AD[42]
84	AD[41]	+5V (I/O)	AD[41]	+3.3V (I/O)
85	GND	AD[40]	GND	AD[40]

**Tabla 4.10** Distribución de pines en una ranura PCI (continuación).

Pin	Sistema de 5V		Sistema de 3.3V	
	Cara B	Cara A	Cara B	Cara A
86	AD[39]	AD[38]	AD[39]	AD[38]
87	AD[37]	GND	AD[37]	GND
88	+5V (I/O)	AD[36]	+3.3V (I/O)	AD[36]
89	AD[35]	AD[34]	AD[35]	AD[34]
90	AD[33]	GND	AD[33]	GND
91	GND	AD[32]	GND	AD[32]
92	Reserved	Reserved	Reserved	Reserved
93	Reserved	GND	Reserved	GND
94	GND	Reserved	GND	Reserved
Final de la sección de 64 bits.				

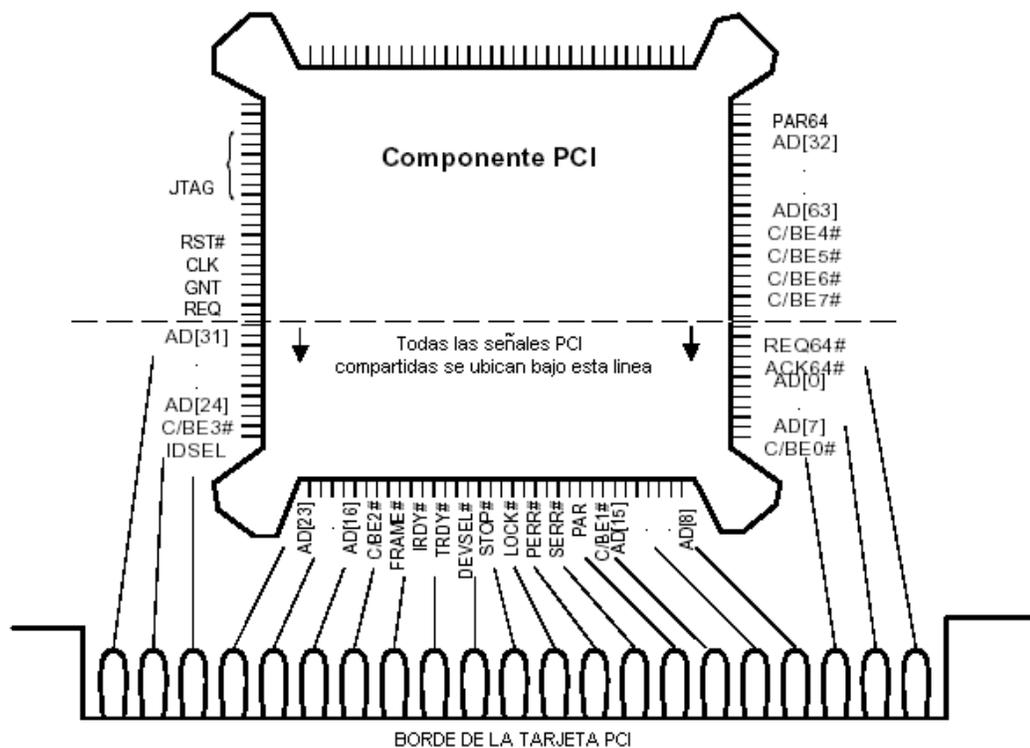
Hay algunos puntos importantes a señalar:

- Sobre la plataforma, las patillas identificadas como +5V(I/O) y +3.3V(I/O) son los pines diseñados para alimentación de I/O. Según la tensión para la cual está diseñada la plataforma, estos terminales se conectan al plano de alimentación de 3.3V ó de 5V. De esta forma una tarjeta dual puede reconocer el tipo de sistema sobre el cual está montada.
- Los pines REQ64# y ACK#64 están conectados a resistencias de pull-up sobre la plataforma. Por lo tanto no se permite el uso de pull-ups para estas líneas, en las tarjetas de expansión. Ambos pines son definidas como parte de la ranura de 32 bits.
- Las terminales PRSNT1# y PRSNT2# identifican la potencia de consumo esperada para la tarjeta de expansión, según como se muestra en la Tabla 16.

**Tabla 4.11** Identificación de la potencia consumida por una tarjeta PCI.

PRSNT2#	PRSNT1#	Potencia máxima permitida (Watts)
GND	GND	7.5
GND	OPEN	15
OPEN	GND	25
OPEN	OPEN	No hay tarjeta

- Los componentes integrados utilizan un orden especial en la distribución de sus pines para lograr un enrutamiento más sencilla hacia la ranura PCI. Por ejemplo, los dispositivos que utilizan montajes PQFP poseen una distribución de su interfaz al bus como se muestra en la Figura 4.5.



**Figura 4.5** Distribución de pines sugerida para un componente PCI - PQFP.

#### 4.4.5 Diseño de impresos para tarjetas de expansión

La distribución de pistas en una tarjeta de expansión debe cumplir una serie de requerimientos:

- La longitud máxima de una pista desde el borde de la tarjeta hasta un componente es de 1.5 pulgadas. Hay tres excepciones:
  - a. Para las señales AD[32..63], C/BE[4..7]#, PAR64, REQ64# y ACK64# se acepta una longitud máxima de 2 pulgadas.
  - b. La señal de reloj CLK debe tener una longitud de  $2.5 \pm 0.1$  pulgadas y se conecta sólo a una carga.
  - c. Los pines de sistema, de interrupción y de JTAG no poseen restricciones de longitud.
  - d. La máxima longitud desde el borde de la tarjeta hasta los planos de alimentación y tierra es de 0.25 pulgadas, para un espesor de 20 milí pulgadas en las pistas.
- Todas las líneas de señal PCI que pueden ser compartidas por otros recursos en la plataforma, deben tener únicamente una carga compatible PCI por cada tarjeta de expansión. Ninguna resistencia de pull-up, otros dispositivos discretos o componentes no compatibles con PCI deben ser conectados. Un puente PCI/PCI debe ser utilizado como buffer si la tarjeta de expansión contiene circuitería que va a violar estas restricciones. La única excepción a este requerimiento es el uso de una resistencia entre las líneas de señal AD y IDSEL.
- La impedancia característica sin carga de las líneas PCI compartidas ( $Z_0$ ) debe mantenerse dentro de los 60 y los 100 Ohmios.

- Idealmente, las tarjetas de expansión deben ser de seis capas: una para el plano de 3.3V, una para el de 5V, una para el de GND, y tres para señales. Por consideraciones de costo, un impreso de cuatro capas puede ser más práctico; en este caso, dos capas se utilizan para las señales (generalmente las exteriores), una para tierra, y otra para las líneas de alimentación.

#### **4.4.6 Condensadores de desacople**

Existe un conjunto de condensadores que se usan para desacoplar los planos de alimentación. Los requerimientos básicos para colocar estos componentes son los siguientes:

- Para las tarjetas duales, los terminales +V I/O deben estar desacoplados individualmente por medio de un condensador de 0.047  $\mu$ F hacia la capa de tierra. Es posible utilizar un condensador más grande si se desea desacoplar varias líneas a la vez.
- Todos los terminales de 3.3V y 5V que no sean utilizados deben ser conectados a un condensador de alta velocidad de 0.01  $\mu$ F con una pista de al menos 0.02 pulgadas de ancho y como máximo 0.25 pulgadas de largo. De nuevo, si el trazado de las pistas lo requiere, es posible abarcar varias terminales con un condensador mayor.

#### **4.5 Plug and Play**

Una gran variedad de diferentes tarjetas que pueden ser agregadas a una PC para expandir sus capacidades es a la vez una ventaja y un problema. El proceso de configurar el sistema y distribuir adecuadamente los recursos es parte del precio a pagar cuando se tiene muchos elementos diferentes, más aún cuando no se ajustan a un estándar común. Este proceso es una de las partes más difíciles para el usuario que adquiere una PC nueva o que realiza una actualización de su equipo.

En un intento para resolver esta situación, la especificación Plug and Play (también llamada PnP) fue desarrollada por Microsoft en cooperación con Intel y muchos otros productores de hardware. El objetivo de Plug and Play es crear una computadora en la cual el hardware y el software trabajen juntos para automatizar la configuración de los dispositivos y la asignación de recursos. Así se permite los cambios de hardware y la adición de componentes sin la necesidad de un proceso de re-asignación en forma manual.

Como el nombre sugiere, PnP está hecho para que los usuarios únicamente deben conectar sus aditamentos y el sistema se encargue de reconocerlos, ubicarlos dentro del sistema global y ponerlos a funcionar correctamente, todo en forma automática.

Los primeros intentos para poner a funcionar el paradigma de PnP se hicieron con los buses EISA y MCA, sin embargo estos no lograron impactar el mercado. El lanzamiento fuerte de PnP se dio en 1995 cuando Microsoft presentó su Windows 95, y el hardware diseñado específicamente para él.

#### **4.5.1 Requerimientos para Plug and Play**

La detección y configuración automática del hardware y el software no es una tarea sencilla. Para realizar este trabajo se requiere cooperación entre las dos áreas implicadas. Los cuatro “compañeros” que deben ser compatibles con Plug and Play para que el sistema trabaje bien son:

- El hardware de sistema: La estructura eléctrica, desde el chipset y los controladores de bus, debe ser capaz de manejar dispositivos PnP. Para los sistemas modernos basados en PCI, esto está incorporado ya que PCI fue diseñado tomando en cuenta los requerimientos PnP. Además, en ellos se permite PnP en el bus ISA, con una circuitería especial para enlazar los dos estándares. Las computadoras viejas, que trabajan únicamente con ISA o VL no son compatibles con PnP.

- El hardware periférico: El módulo que se está agregando al sistema debe ser compatible con PnP, lo cual no es muy difícil actualmente. Desde modems hasta tarjetas de red, los circuitos de expansión que se fabrican hoy día cumplen con los requisitos para trabajar con PnP. Esto permite que se dé un intercambio de información entre el módulo y el sistema, para el reconocimiento individual y la asignación de los recursos.
- El BIOS del sistema: Este componente juega un papel fundamental en el adecuado funcionamiento de PnP dentro de un sistema. En el BIOS se guardan las rutinas que permiten la identificación de los elementos y la asignación de los recursos. El BIOS además es quien comunica esta información al sistema operativo, para que se pueda configurar los manejadores y otro software necesario para el trabajo de los dispositivos. En algunos casos, sistemas que no usan PnP por la versión de su BIOS pueden ser actualizados y funcionar adecuadamente con un BIOS nuevo, compatible con PnP (este es el caso de los primeros sistemas PCI para Pentium producidos entre 1993 y 1995).
- El sistema operativo: Finalmente, el cuarto elemento esencial para el trabajo con PnP es el Sistema Operativo, que trabaja en conjunto con el BIOS. Este programa se encarga de activar cualquier software de bajo nivel, como drivers y librerías, necesarios para que las aplicaciones grandes actúen sobre el hardware. Aquí es donde el sistema se comunica con el usuario y le permite ver la configuración, hacerle cambios si es necesario. Dado a que Microsoft fue quien impulsó en primera instancia este paradigma, sus sistemas operativos a partir de Windows 95 son los que están garantizados como 100% compatibles con PnP.

Como puede notarse, es necesario tener muchos elementos de tecnología actual para trabajar con PnP. Esta es la razón por la cual los elementos o sistemas previos a 1996 por lo general no pueden adaptarse a este estándar.

#### **4.5.2 Operación del Plug and Play**

La mayoría del trabajo que involucra realizar las funciones PnP queda a cargo del BIOS del sistema durante el proceso de arranque o “boot”. En el paso apropiado del arranque, el BIOS debe seguir un procedimiento especial para determinar y configurar los dispositivos PnP en el sistema.

Existe un algoritmo estricto en cuanto a los pasos que el BIOS debe seguir en el tiempo de arranque, cuando está manejando un sistema basado en PCI:

- a. Se crea una tabla de recursos con los IRQs, canales de DMA y direcciones I/O disponibles, excluyendo aquellas que están reservadas por dispositivos del sistema.
- b. Se busca e identifica los dispositivos PnP y no PnP en los buses ISA y PCI.
- c. Se lee la última configuración de sistema guardada en el área ESCD, de la memoria no volátil.
- d. Se compara la configuración actual con la última guardada. Si no hay diferencias, se continúa con el proceso de arranque; esta parte del proceso finaliza y el resto del levantamiento avanza desde este punto.
- e. Si la configuración es nueva (hay diferencias en la comparación), inicia la reconfiguración del sistema. Se inicia con la tabla de recursos, eliminando cualquiera que esté siendo usado por elementos no PnP.

- f. Se chequea los datos en el BIOS para saber si algún recurso adicional del sistema ha sido reservado para usar con dispositivos no PnP, y así eliminarlo de la tabla.
- g. Se asigna los recursos a las tarjetas PnP desde los datos que aparecen en la tabla, y se les informa sobre sus nuevas asignaciones.
- h. Se actualiza el área de ESCD salvando la nueva configuración. La mayoría de los BIOS presentan un mensaje como “Updating ESCD... Successful” (“Actualización de ESCD... Exitoso”).
- i. Se continúa con el proceso de iniciación o arranque.

En la sección PCI/PnP, dentro del BIOS de una computadora basada en bus PCI, se puede encontrar una serie de opciones que afectan la forma en que se trabaja PnP.

#### **4.5.3 Datos de configuración extendida del sistema (ESCD)**

Si el BIOS tuviera que asignar recursos a cada dispositivo PnP siempre que arranca, se darían dos problemas básicos. Primero, se gastaría tiempo sin ningún propósito en una tarea que ya se hizo anteriormente. Tomando en cuenta que la mayoría de los usuarios cambian su configuración de hardware muy pocas veces, rehacer esto siempre no es necesario. Como segundo punto, y más importante: es posible que el BIOS no siempre haga la misma asignación de los recursos, lo que puede conllevar a encontrar los recursos cambiados entre una sesión y otra sin haber modificado en hardware.

El ESCD está diseñado para evitar estos problemas. Esta es un área especial de la memoria CMOS donde se mantiene el BIOS. Se utiliza para guardar los datos correspondientes a la configuración del hardware instalado.

Cuando la máquina arranca, el BIOS revisa esta área para verificar que no haya cambios en el hardware con respecto a la última vez que el sistema levantó. Si no hay cambios, se omite el proceso de configuración y simplemente se asigna los mismos valores que ya se tenían guardados.

El ESCD se utiliza también como un lazo de comunicación entre el BIOS y el sistema operativo. Ambos usan el área de ESCD para leer el estado actual del hardware y grabar cambios. Windows 95 lee el ESCD para determinar si hay cambios de configuración y actuar apropiadamente ante ellos. Windows 95 también permite a los usuarios realizar la distribución de recursos manualmente por medio del Manejador de Dispositivos. Esta información se graba en el ESCD para que sea tomada en cuenta por el BIOS en el siguiente proceso de encendido, y no la cambie a partir de entonces.

Los datos del ESCD se guardan en una memoria no volátil de tipo CMOS, de la misma forma que se guardan las opciones básicas del BIOS.

#### **4.5.4 PnP y dispositivos no PnP**

Los módulos o circuitos que no soportan el estándar PnP pueden ser usados en estos sistemas, pero van a presentar problemas especiales. Estas unidades se denominan de tipo Legacy y hacen la asignación de recursos algo más complicada porque no pueden ser configurados automáticamente por el BIOS.

Generalmente el BIOS no trabaja con ellos más que para determinar cuáles recursos están utilizando y así descartarlos de la tabla, de forma que no se dé choque cuando se haga la asignación a los dispositivos PnP.

#### **4.6 Descripción del microprocesador 80960-RM**

Este procesador es una versión mejorada del i960-VH que, además de las características que incorporaba anteriormente, agrega una serie de funciones especiales para trabajar con puertos de entrada / salida en equipos de cómputo modernos.

Dentro de sus principales características están:

- Procesador Intel 80960JT de alta eficiencia.
  - Ejecución de una instrucción por ciclo de reloj.
  - 16 Kbyte de caché para instrucciones.
  - 4 Kbyte de caché para datos mapeados directamente.
  - Dieciséis registros globales de 32 bits.
  - Dieciséis registros locales de 32 bits.
  - 1 Kbyte de RAM interna.
  - Caché de registros locales.
  - Dos temporizadores de 32 bits integrados.
- Unidad puente PCI/PCI.
  - Ocho buffers para transacciones retardadas de lectura / escritura.
  - Interfaces primaria y secundaria PCI de 32 bits.
  - Dos buffers permiten colocar hasta 12 transacciones encargadas en cada lado

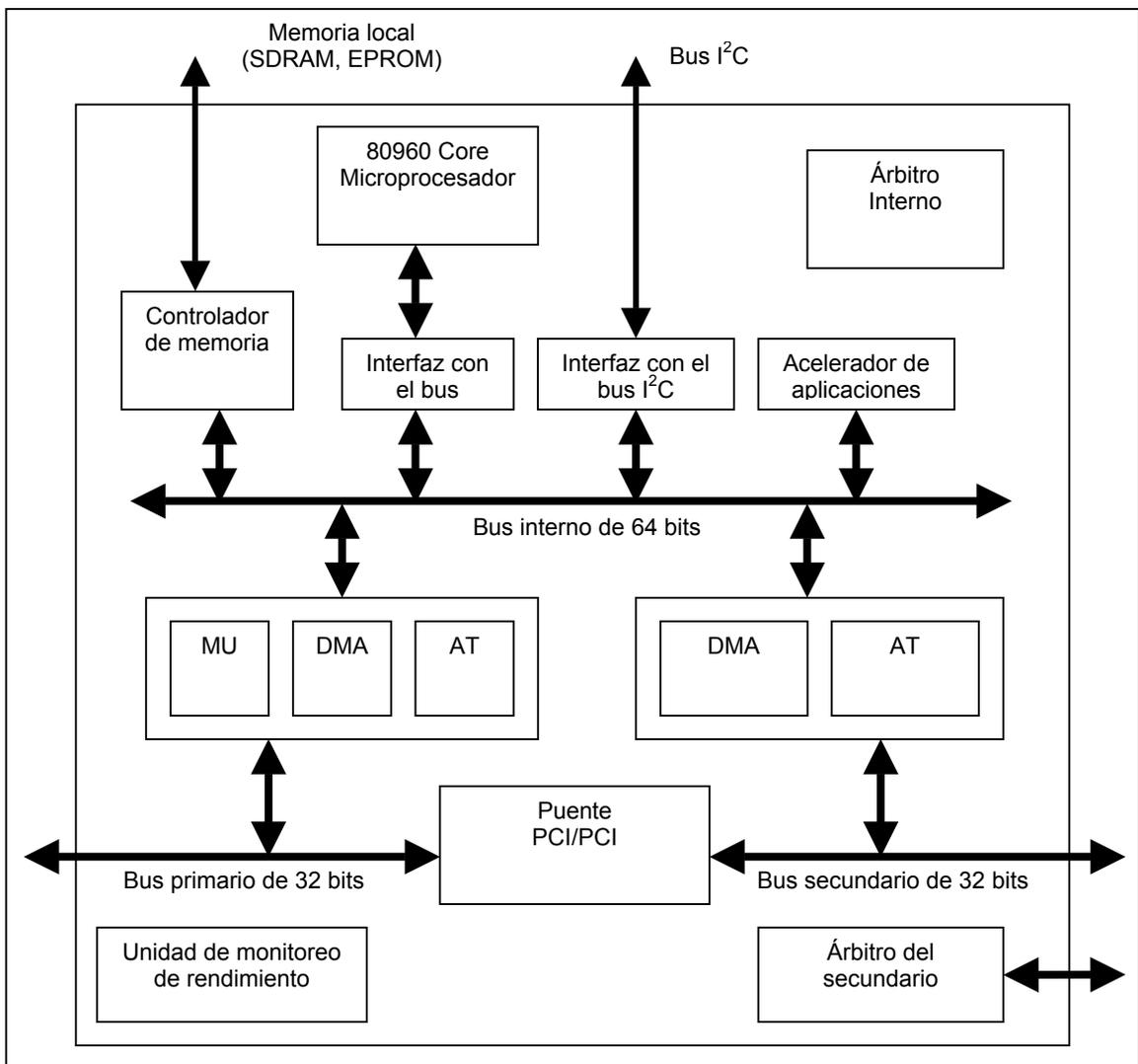
- Transporta comandos de memoria, I/O y configuración de un bus a otro.
- Unidad de mensajes I<sup>2</sup>O.
  - Cuatro registros de mensaje.
  - Dos registros "Doorbell".
  - Cuatro colas circulares.
  - 1004 registros índice.
- Controlador de memoria.
  - 128 Mbytes de SDRAM a 64 bits, o 64 Mbytes de SDRAM a 32 bits.
  - Corrección de error Single-Bit y Double-Bit, ECC.
  - Dos bancos independientes para SRAM/ROM/Flash (8 Mbytes por banco, 8 bits).
- Dos unidades de trasladación de dirección.
  - Conexión del bus interno a los buses PCI.
  - Soporta trasladación de dirección Inbound/Outbound.
  - Soporta direccionamiento directo Outbound.
- Controlador DMA.
  - Tres canales independientes.
  - Interfaz entre el controlador de memoria y el PCI.
  - Direccionamiento interno y PCI de 64 bits.

- Interfaz independiente a los buses PCI primario y secundario.
- Transferencias en ráfaga de hasta 132 Mbytes por segundo a los buses PCI y locales.
- Direccionamiento directo de y desde los buses PCI.
- El hardware soporta transferencias no alineadas.
- Dos canales dedicados al bus PCI primario, y uno al secundario.
- Unidad de interfaz con I<sup>2</sup>C.
  - Bus serial.
  - Capacidades de Master/Slave.
  - Funciones para el manejo del sistema.
- Unidad de arbitración para el bus secundario.
  - Soporta hasta seis dispositivos PCI en el secundario.
  - Algoritmo de arbitración multi-prioridad.

Todas estas características lo hacen idóneo para aplicaciones donde la carga de un bus PCI debe ser compartida por un grupo de dispositivos que a su vez, puede utilizar los recursos de memoria del procesador y pueden encargarse de realizar tareas específicas a través de las líneas de interrupción.

Este chip está capacitado para controlar el flujo de información entre los buses de forma que el programador utiliza el bridge como si fuera un periférico, configurando sus registros y revisando sus banderas de estado.

El diagrama descriptivo de este chip se muestra en la Figura 4.6.



**Figura 4.6** Diagrama de bloques del 80960-RM.

El módulo central de procesamiento corresponde a un procesador 80960, versión escalada de la arquitectura i960 desarrollada por Intel.

Algunos factores que contribuyen a la eficiencia del procesador son:

- Ejecución de la mayoría de instrucciones en un ciclo de reloj.

- Unidad independiente para multiplicar y dividir.
- Una cola de instrucciones eficiente disminuye el tiempo muerto por rompimiento de la cadena precargada.
- La arquitectura permite ejecución de instrucciones solapadas.
- Bus interno de 128 bits para los registros de caché.
- 16 Kbyte de caché integrada para instrucciones.
- 4 Kbyte de caché integrada para datos.
- 1 Kbyte de RAM integrada permite la ejecución sin ciclos de espera.

La ubicación del espacio de memoria local es independiente de los buses PCI, y dependiendo de la aplicación puede configurarse como: visible para los buses PCI, privada o propiedad del procesador, o bien una combinación de ambas .

El módulo de interfaz con el bus interno (BIU) permite realizar transacciones de diferentes modos, manteniendo separado el bus de procesador (32 bits, 100MHz) con el bus interno del circuito (64 bits, 66MHz) que maneja el espacio de memoria externa y los periféricos. El BIU posee algunos buffers para datos y direcciones:

- Un buffer de escritura: guarda 1 dirección y hasta 16 bytes de datos.
- Un buffer de lectura: guarda hasta 16 bytes de datos.
- Un buffer de precarga (pre-fetch): guarda hasta 16 bytes de instrucciones.

La interfaz con el bus I<sup>2</sup>C es una opción para el traslado de información a una gran cantidad de dispositivos periféricos que hoy día son compatibles con este bus. Utiliza una mínima cantidad de terminales y es muy eficiente.

Por su parte, el controlador de memoria permite controlar directamente sistemas de memoria externos, incluyendo bancos de SDRAM, ROM y Flash. Provee una conexión directa que por lo general no requiere lógica adicional de control.

Presenta señales de selección programables, generador de estados de espera y una lógica de corrección de errores ECC. Además, permite que la memoria externa sea direccionable desde los buses PCI.

A continuación se encuentra la Unidad de Aceleración de Aplicaciones (AAU) que provee un sistema de transferencia de alta velocidad hacia la memoria local del procesador. Ejecuta transferencia de datos desde y hacia la memoria, y tiene una interfaz programable. Sus funciones auxiliares permiten una ejecución más eficiente de los programas.

El Árbitro Interno por su parte, debe encargarse de dar los permisos a todos los otros dispositivos para utilizar el bus interno. De esta forma el sistema se asegura que no exista choque de información, y que el bus está siendo utilizado eficientemente por todos los recursos.

El bloque marcado como MU corresponde a la Unidad de Mensajes, que se encarga de transferir datos entre el sistema PCI primario y el procesador local. Funciona como traductor entre el chip y el bus PCI, y de esta forma es posible comunicar mensajes a la plataforma del primario (computadora) o recibirlos desde ella.

Luego se tiene las unidades para DMA. Estas etapas se encargan de sincronizar las transferencias de datos en bloques, entre la memoria local y los buses PCI. De esta forma se puede transmitir secciones completas de información sin intervención del procesador.

En cuanto al control de direcciones se refiere, se tienen las ATU o Unidades de Traslación de Direcciones. Estas se encargan de coordinar que las direcciones manejadas internamente sean compatibles con las que se ven desde los buses PCI, manteniendo estos valores independientes. Quiere decir que si un bloque del mapa de memoria del procesador es visto por uno de los buses PCI, la ubicación en el mapa del bus no tiene porque ser la misma.

La unidad periférica de mayor importancia para el manejo del bus y la interfaz con él es el Puente PCI/PCI interno. Este dispositivo se encarga de conectar dos buses PCI independientes, y se puede coordinar con el ATU para accesos transparentes al espacio de memoria local desde cualquiera de los dos buses.

#### **4.6.1 Puente PCI a PCI**

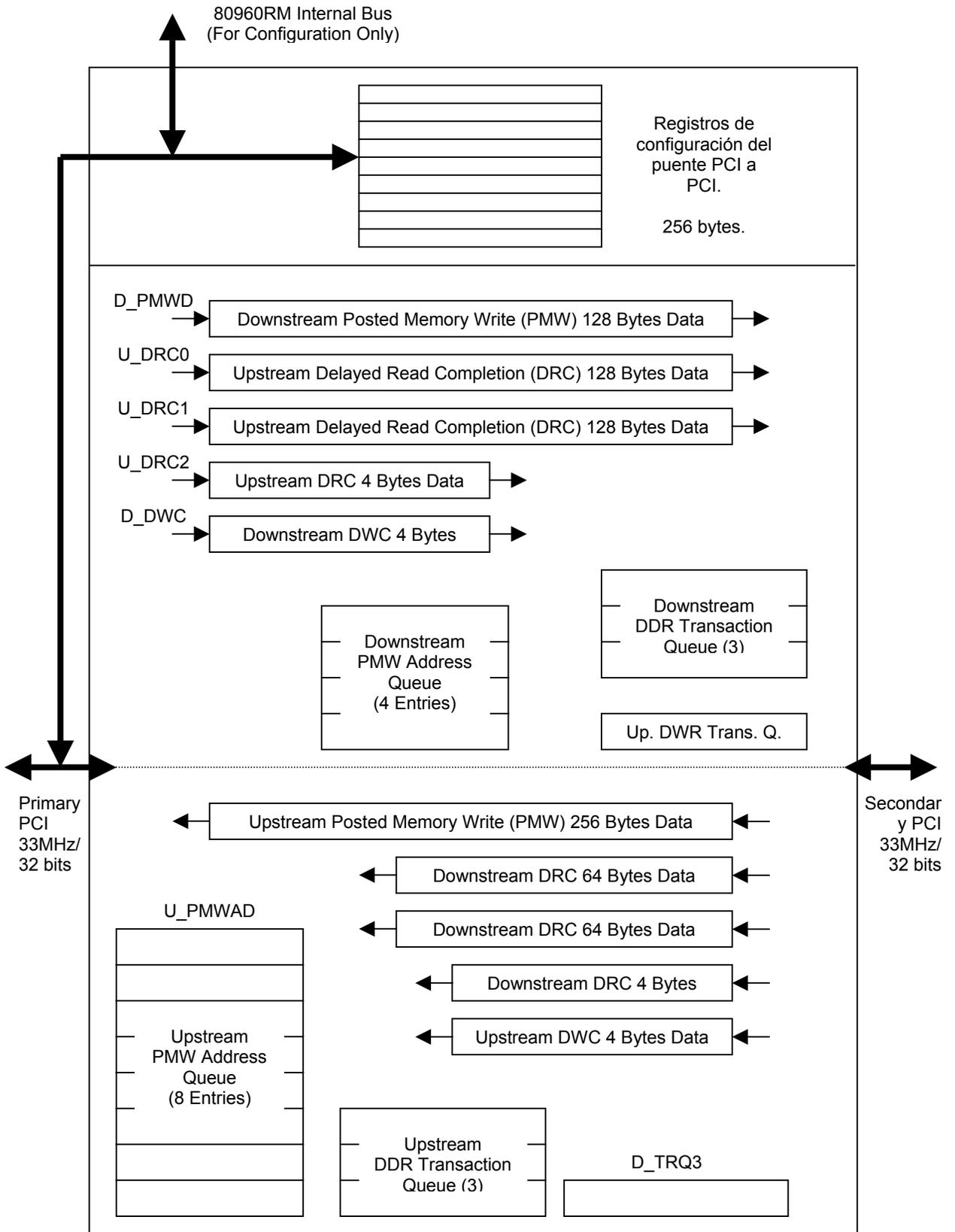
La unidad puente PCI a PCI extiende las capacidades de un bus primario más allá de sus limitaciones eléctricas de 10 cargas PCI a 33 MHz. Esta unidad utiliza el concepto de buses hereditario; cada bus en la jerarquía se encuentra eléctricamente separado de los otros, pero todos los buses constituyen un único bus lógico. El puente no incrementa el ancho de banda de un bus, sino simplemente permite que se extienda su capacidad de manejo de dispositivos en aplicaciones que requieren una mayor cantidad de componentes I/O de las permitidas por la especificación PCI.

Las características principales de la unidad puente PCI-PCI son:

- Compatible con la Revisión 2.1 de la Especificación para el Bus Local PCI.
- Compatible con la Revisión 1.0 de la Especificación para Puentes PCI-PCI.
- Capacidad de hasta 264 Mbytes por segundo tanto en el bus primario como en el secundario en la versión de 64 bits.
- Operación sincrónica entre el primario y el secundario.

- Soporte para masters y targets de 32 bits en ambos buses.
  - Soporte adicional para configuraciones independientes de bus (32 bits).
- Operación independiente de los buses permite realizar tareas concurrentes en ambas direcciones.
- Múltiples operaciones *Memory Write* y *Memory Write and Invalidate* en colas hacia ambas direcciones concurrentemente.
  - Hasta 4 transacciones PMW con un total de 128 Bytes de datos para escritura en dirección descendente.
  - Hasta 8 transacciones PMW con un total de 256 Bytes de datos para escritura en dirección ascendente.
- Soporta hasta tres ciclos de lectura retardados iniciados desde el bus primario y tres similares desde el secundario.
  - 260 bytes dedicados para completar las lecturas retardadas en dirección ascendente.
  - 132 bytes dedicados para completar las lecturas retardadas en dirección descendente.
- Espacios de dirección separados para memoria e I/O en el bus secundario.
- Direccionamiento dual permitido en ciclos iniciados desde el secundario.
- Espacio de dirección y configuración privado, para dispositivos marcados como tales en el bus PCI secundario.

En la Figura 4.7 se muestra el diagrama de bloques de esta sección del chip.



**Figura 4.7** Diagrama de bloques del módulo puente, dentro del 80960RM.

La unidad puente opera como un filtro de direcciones entre el bus PCI primario y el secundario. PCI soporta tres tipos de espacio de dirección:

- Direccionamiento básico, que puede ser de 32 bits (SAC) o de 64 bits (DAC).
- 64 Kbytes de espacio de puertos I/O (direccionamiento de 16 bits).
- Espacio separado de configuración.

Un puente PCI a PCI está programado con un rango continuo de direcciones dentro del espacio de memoria e I/O, que se convierte en el espacio de direccionamiento del secundario. Cualquier dirección presente en el lado del primario ubicada en el rango programado es llevada hacia el secundario para ejecutar el ciclo allí, mientras que las direcciones fuera de este límite son ignoradas por el circuito.

El secundario trabaja en la forma inversa, trasladando los ciclos con dirección fuera del rango al bus primario e ignorando los que caen dentro del mismo.

Las interfaces primaria y secundaria del puente PCI trabajan como master y targets compatibles con la Revisión 2.1 de la Especificación para Bus Local PCI. Una transacción iniciada en un lado de puente requiere la acción del mismo como target en el bus fuente, y como master en el destino.

El puente es transparente para el software de los dispositivos en ambos lados.

Todos los ciclos en el secundario que se aplican a direcciones dentro del rango del puente son ignorados.

La unidad puente PCI a PCI puede ser separada en cuatro unidades lógicas principales:

- Interfaz PCI primaria,
- Interfaz PCI secundaria,
- Colas Ascendentes / Descendentes,
- Registros de Configuración.

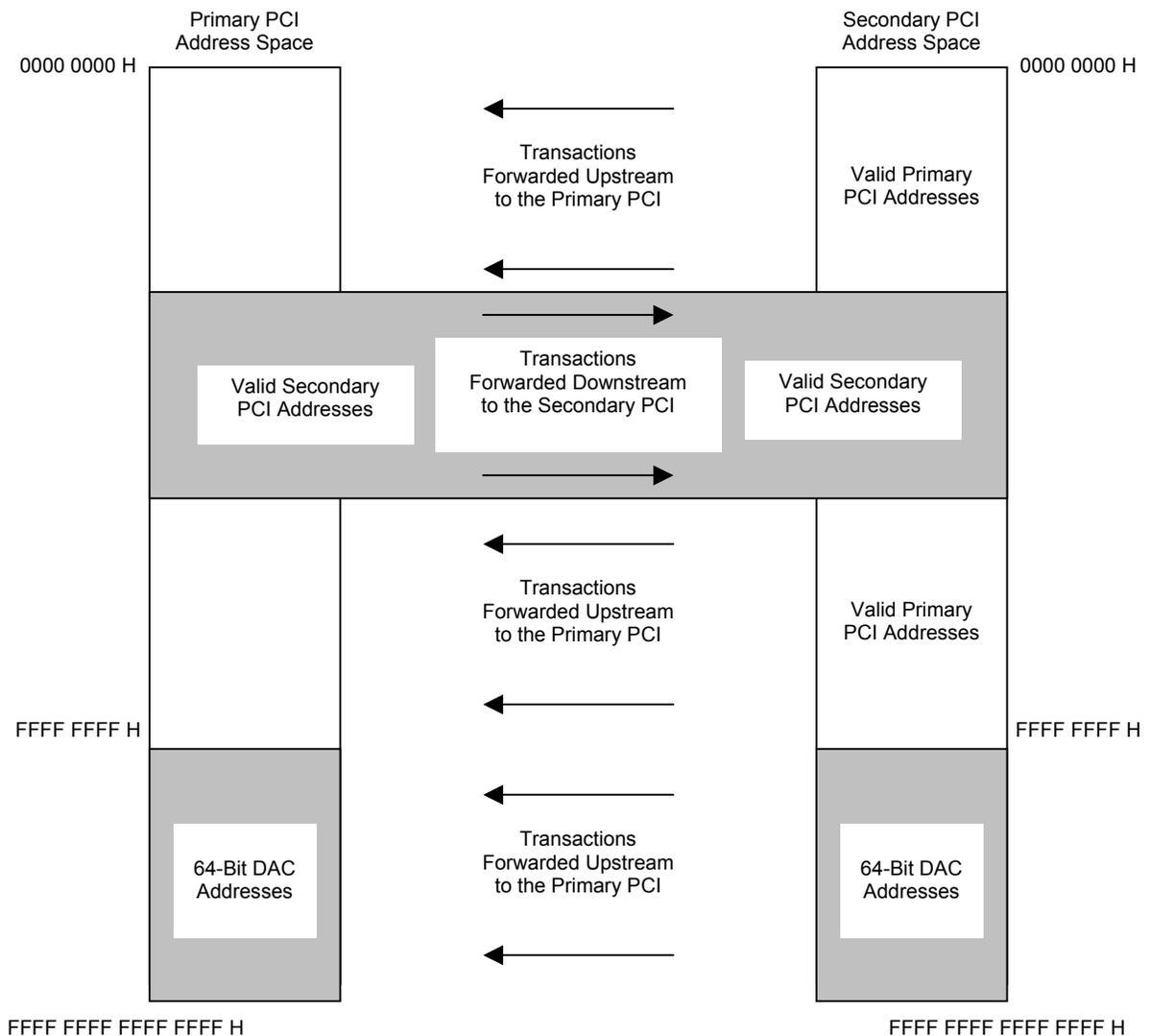
#### **4.6.1.1 Interfaz PCI primaria**

La interfaz PCI primaria puede actuar como master o target en el bus Primario. En la mayoría de los sistemas, esta es la Interfaz que se conecta en el bus más cercano al Host, es decir, que posee un número de asignación menor. Consiste de 50 señales definidos por la especificación para puentes PCI/PCI y cuatro pines opcionales para interrupciones.

El primario implementa las funciones de iniciador y target PCI dependiendo del tipo de transacción que se está realizando. Cuando un ciclo ascendente es iniciado en el bus secundario, la Interfaz primaria es la encargada de completar este comando, procediendo como master. Cuando un ciclo descendente es iniciado en el primario, la Interfaz PCI primaria cumple el papel de target, como si este fuera el dispositivo requerido por el ciclo.

La interfaz PCI primaria es responsable por toda la interpretación de comandos, la decodificación de direcciones y el manejo de errores, para transacciones iniciadas en el bus primario.

En la Figura 4.8 se muestra la lógica de filtrado de la información que realiza el puente, según la dirección de los ciclos iniciados en sus buses.



**Figura 4.8** Operación del circuito puente: selección de transacciones.

#### 4.6.1.2 Interfaz PCI secundaria

Esta etapa funciona casi de la misma manera que su similar en el bus primario. Consiste en una implementación de las funciones de master y target en el ambiente del bus secundario, con un grupo extendido de posibles cargas disponibles para el sistema. Está compuesto por las 49 patillas requeridas por la especificación para arquitectura de buses PCI/PCI. Además se agregan cuatro pines para interrupción adicionales, que pueden ser usados por los dispositivos del secundario.

Como target, la Interfaz de secundario es responsable de atender todos las transacciones que no caen dentro de la sección de memoria o I/O asignada al secundario, y proceder de forma adecuada en el primario para completar los ciclos.

La Interfaz del secundario también implementa un espacio de direccionamiento separado para dispositivos privados en el bus, que no puede ser visto desde el primario. Esta Interfaz permite el uso de ciclos de direccionamiento dual únicamente para transacciones de memoria iniciadas desde el bus secundario, ya que no es posible ubicar la región válida de secundario por sobre la dirección FFFF FFFF<sub>H</sub>.

#### **4.6.1.3 Colas ascendentes y descendentes**

El procesador 80960-RM implementa una arquitectura de colas para mejorar el ancho de banda de todos los ciclos de escritura y reducir la duración de las lecturas tramitadas entre ambos buses. Tal y como lo exige las especificaciones para puentes PCI a PCI, esta unidad permite transacciones encargadas (posted) y retardadas (delayed).

En una transacción retardada, la información requerida para completar el ciclo se almacena en colas, mientras que al master se le envía una orden de reintentar. Entonces se procede a buscar los datos en el otro extremo del circuito. Si el ciclo se repite y aún no se tiene los datos listos, se continúa ordenando una terminación retry. Cuando los datos ya están listos, se entregan al primario en el correspondiente ciclo de repetición.

En una transacción encargada, los datos se colocan en una cola dentro del circuito, y el ciclo en el bus iniciador finaliza. Cuando la información traspase la estructura de las colas, se dará un ciclo idéntico al original, pero en el bus destino.

Este puente posee una arquitectura no simétrica, soportando flujo de datos según los requerimientos de aplicaciones inteligentes para I/O.

Para transacciones hacia abajo (iniciadas en el primario), el puente suporta la siguiente cantidad y tipos de colas:

- Hasta cuatro transacciones con 128 bytes de datos para escrituras encargadas en espacio de memoria.
  - Implementación FIFO, que soporta escrituras de longitud variable dentro de la misma cola. Cualquier combinación de tamaños para ráfagas, desde una hasta cuatro transacciones.
  - Soporta ciclos *Memory Write* y *Memory Write and Invalidate*.
- 132 bytes para completar lecturas retardadas (DRC), con tres Colas de Dirección de Transacción separadas.
  - Dos colas DRC de 64 bytes.
  - Una cola DRC de 4 bytes.
  - Gracias a las Colas de Transacción se puede retener las direcciones durante las lecturas retardadas.
  - Soporta lecturas del tipo: *Memory Read*, *Memory Read Line*, *Memory Read Multiple*, *Configuration Read*, e *I/O Read*.
- Presenta una cola separada de 4 bytes, para escritura de I/O y configuración.
  - Maximizados como ciclos de escritura retardados.

Para transacciones hacia arriba (iniciadas en el secundario), el puente soporta un conjunto mayor de colas, para adaptarse a la alta cantidad de tráfico que se tiene en el primario.

- Hasta 8 transacciones con datos de 256 bytes, para escrituras encargadas.
  - Implementación FIFO, que soporta escrituras de longitud variable dentro de la misma cola. Cualquier combinación de tamaños para ráfagas, desde una hasta ocho transacciones.
  - Soporta ciclos *Memory Write* y *Memory Write and Invalidate*.
- 260 bytes para completar lecturas retardadas (DRC), con tres Colas de Dirección de Transacción separadas.
  - Dos colas DRC de 128 bytes.
  - Una cola DRC de 4 bytes.
  - Gracias a las Colas de Transacción se puede retener las direcciones durante las lecturas retardadas.
  - Soporta lecturas del tipo: *Memory Read*, *Memory Read Line*, *Memory Read Multiple*, *Configuration Read*, e *I/O Read*.
- Presenta una cola separada de 4 bytes, para ciclos de escritura retardados.
- Escritura de I/O y configuración.

#### **4.6.1.4 Registros de configuración**

Cada dispositivo PCI implementa un espacio de configuración separado y registros para este fin. Las Especificaciones para el Bus Local PCI, Rev. 2.1 requieren que el espacio de configuración sea de 256 bytes, con los primeros 64 ajustados a un formato predeterminado como encabezado. El puente interno al 80960-RM contiene los 64 bytes iniciales más registros de configuración adicionales para controlar la operación del dispositivo.

Los primeros 16 bytes de este espacio implementan los registros comunes que deben tener todos los dispositivos PCI. El valor en el Registro de Tipos de Encabezado (sólo de lectura) define el formato para los restantes 48 bytes del encabezado, y para el caso de un bridge PCI-PCI su valor es 01H.

Los dispositivos desde el primario pueden acceder a la información de estos registros por medio de ciclos de configuración Tipo 0. Los dispositivos en el secundario no pueden leer o escribir sobre este espacio de ninguna manera. Los registros de configuración guardan toda la decodificación de direcciones, condiciones de error e información de estado necesaria para ambos lados del puente.

#### **4.6.2 Accesos de configuración**

Hay dos clases de targets para comandos de configuración PCI:

- Dispositivos que residen en el bus primario PCI.
- Dispositivos que residen en buses PCI subordinados (por lo general en el nivel secundario) que son vistos a través de un puente PCI-PCI.

La codificación de direcciones durante un ciclo de configuración determina el tipo de target al cual va dirigido el comando. A continuación se muestra las diferentes codificaciones de las direcciones asociadas con cada tipo de comando de configuración PCI.

Los comandos de tipo 0 y de tipo 1 se distinguen por los bits de dirección AD[0..1]

Un comando de configuración tipo 0 en el primario es aceptado o no dependiendo del valor de la entrada P\_IDSEL. Un comando tipo 1 en el primario puede ser ignorado, llevado sin cambios al bus inferior, convertido a un comando tipo 0 en el secundario, o convertido en un ciclo especial en el secundario.

En la siguiente tabla se muestra la función de cada bit que recibe el bus de datos en la etapa de comando de un ciclo de configuración, dependiendo del tipo de comando.

**Tabla 4.12** Estructura del comando de configuración.

Función del bit	Comandos tipo 0 Posición de los bits (cantidad)	Comandos tipo 1 Posición de los bits (cantidad)
Tipo de comando	1:0 (2)	1:0 (2)
Número del Registro	7:2 (6)	7:2 (6)
Número de Función	10:8 (3)	10:8 (3)
Número de Dispositivo	N/A	15:11 (5)
Número de Bus	N/A	23:16 (8)
Reservado	31:11 (20)	31:24 (8)

Una escritura de configuración tipo 1 en el secundario puede ser ignorada, llevada hacia el bus superior bajo ciertas condiciones, o convertida en un ciclo especial en el primario. El puente no puede convertir comandos de tipo 1 en el secundario a tipo 0 en el primario, e ignora todos los comandos de lectura y escritura tipo 0 iniciados en el secundario.

Los comandos de configuración sólo son aceptados en el primario si el bit “Configuration Cycle Retry” dentro del Registro Extendido de Comandos del Puente (EBCR) está borrado. Si este bit está en alto, la Interfaz primaria responde Retry a todos los ciclos de configuración, ya sea de tipo 0 ó 1.

Todos los comandos de configuración son de 32 bits y por lo tanto no utilizan extensiones de 64 bits ni en el primario ni en el secundario. Además, el 80960RM no soporta transacciones en ráfaga durante ciclos de configuración. Las escrituras a estos registros son desconectadas luego de la fase de los primeros 32 bits de datos. Las lecturas de tipo 1 pueden leer hasta un máximo de 1Dword (manejada como transacciones retardadas) dependiendo de los bytes activos con las señales C/BE# durante la fase de datos.

### **4.6.3 Unidad de traslado de direcciones**

Una importante características del chip 80960RM es su capacidad de compartir los espacios de direccionamiento de los buses PCI con su mapa local interno. Esto permite realizar operaciones de intercambio de datos desde los buses hacia la memoria local y viceversa.

La importancia de esta característica se hace más evidente en la etapa de programación, donde el diseñador tiene la libertad de colocar una ventana de direcciones en su mapa de memoria, la cual va a corresponder con un espacio específico de los buses PCI externos.

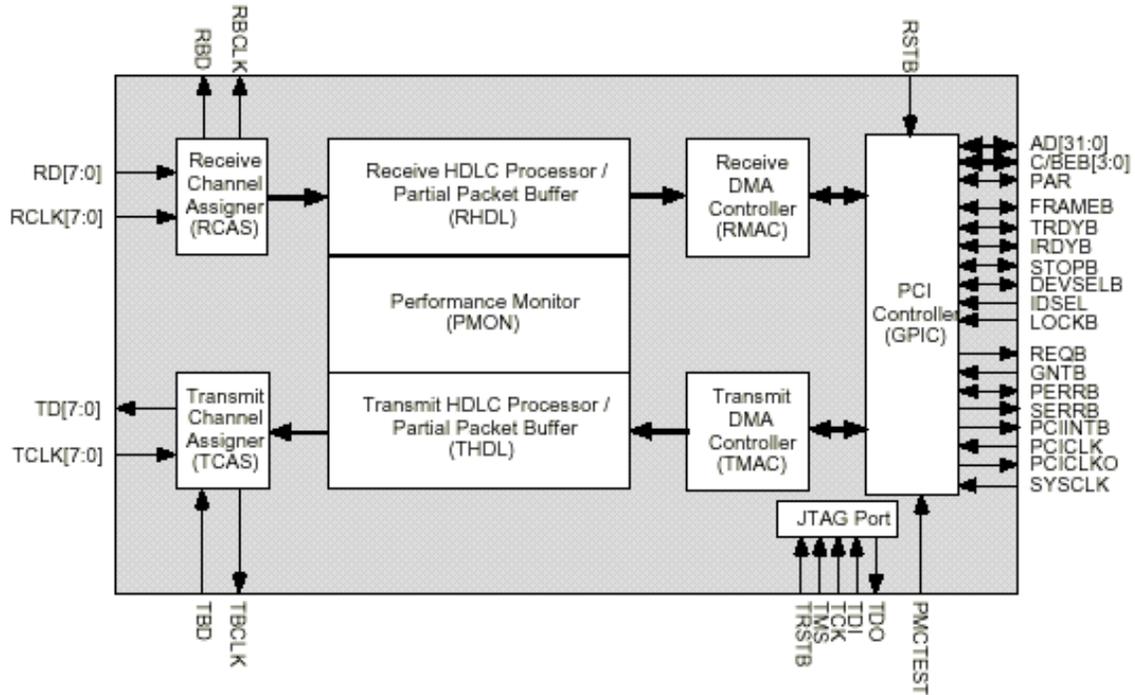
Gracias al uso de la unidad de traslado de direcciones (ATU), para el programador estos accesos serán transparentes, y se puede utilizar los mismos códigos de operación que para accesos a memoria.

Para lograr compatibilidad en los valores de dirección que se usan en los ciclos de bus interno y externo, la ATU posee un conjunto de registros de control y estado. Estos registros se estudian con mayor detalle en la descripción de software, especialmente aquellos que determinan las localidades de inicio y fin de las ventanas.

## **4.7 Descripción del PM-7366**

El circuito controlador de protocolo HDLC se encarga de recibir información y agregarle o quitarle, según la dirección de paso, los elementos propios del protocolo de comunicaciones HDLC, diseñado por la CCITT.

Este chip cuenta con ocho puertos seriales de entrada/salida capaces de manejar hasta 128 canales de información en total. En el otro extremo del sistema posee una interfaz compatible PCI. En la Figura 4.9 se muestra la estructura de este chip.



**Figura 4.9** Diagrama de bloques del PM-1766.

Este circuito no procesa canales ST-Bus, sino variaciones de estos sincronizados de forma diferente según sean E1 ó T1.

Para el caso del proyecto, se debe trabajar en modo para E1, con un pequeño circuito de adaptación.

La primera sección corresponde al Asignador de Canal de Recepción (RCAS) que procesa hasta ocho enlaces seriales, dentro de los cuales se puede codificar 32 canales por enlace. Cada bus serie es independiente, por lo que posee su propio reloj. Cuando se dan eventos en que varios streams han sido acumulados, existe una jerarquía de prioridad en los enlaces, donde el bus cero posee la mayor prioridad y el 7 tiene la mínima.

En enlaces E1 y T1 cada time slot puede ser asignado a un canal diferente. El byte o bit de alineamiento se puede identificar observando que el reloj no cambia durante un intervalo de tiempo (no hay transacciones en el reloj).

El block Asignador de Canal determina el número de canal del dato que se está procesando actualmente. Posee 256 palabras de RAM. La dirección en la RAM se construye concatenando el número de canal y el número de Time slot.

La sección siguiente es el Procesador de Recepción de HDLC/ Buffer Parcial, capaz de procesa hasta 128 canales. Cada canal puede ser configurado individualmente para la detección de la secuencia de banderas, el bit de prevención de emulación de bandera y la verificación de CRC. El paquete de datos se escribe en el buffer parcial. Al final de cada frame, el status del paquete incluye banderas de error en el CRC, de error en el alineamiento del octeto y de violación a la longitud máxima.

Luego se tiene el Procesador HDLC, que es una máquina de estados capaz de procesar hasta 128 canales independientes. El vector de estado y la información proporcionada es almacenada en la memoria RAM interna del dispositivo.

La configuración del procesador de HDLC es accesada utilizando ciclos indirectos al canal. Cuando una operación indirecta es efectuada, la información es leída o escrita en la RAM durante un ciclo nulo de reloj generado por el Asignador de Canal de Recepción. Escribiendo un nuevo dato en el canal reinicia todo el vector de estado del canal.

A continuación, hay una sección denominada Procesador del Buffer Parcial de Paquete (Partial Packet Buffer Processor) que controla los 8 KBytes de la RAM interna del FREEDM-8, la cual está dividida en bloques de 16 bytes.

Un puntero a bloques de RAM se utiliza para concatenar los bloques de memoria en una FIFO circular. No es necesario que los bloques para formar la FIFO se encuentren dispuestos en forma continua.

El software del sistema es el responsable por la alineación de los bloques a un canal de la FIFO específico. Este procesador se encuentra dividido en tres secciones:

- a. Escritor
- b. Lector
- c. Roamer

El escritor es la sección de este procesador que se encarga de escribir en la FIFO la información y el vector de status. El lector transfiere la información que se encuentra en la FIFO al canal de transferencia del controlador de DMA de recepción. El roamer es una máquina de estados que provee las señales de lectura para un canal específico. Si se da un desbordamiento del buffer el escritor termina el servicio dando una bandera de desbordamiento e ignora el resto de los datos.

El algoritmo de la FIFO se basa en una transferencia por canal programable. En vez de una cantidad dada de bloques llenos en la FIFO del canal, el procesador da el número de transacciones. Cuando el escritor realiza una transacción de un número dado de bloques o de escrituras de banderas de final de paquete en la FIFO del canal, se crea una transacción.

Si el lector realiza un número dado de transacciones del bloque, o de fin de paquete al bloque RMAC, una transacción es borrada. De esta manera se pueden transferir pequeños bloques de datos sin tener que tener una cantidad precisa de bloques llenos dentro de la FIFO.

El roamer realiza la el proceso de cuenta para todos los canales. Para esto incrementa el contador de transacción cuando el escritor da una nueva señal de transacción y busca en una cola las banderas para decidir cual canal necesita atención. El lector transfiere el dato al RMAC hasta que el tamaño de la transferencia en el canal se cumple o se encuentra una bandera de final de paquete. El lector comunica al roamer que la transacción ha sido ejecutada. Así se actualiza el contador de transacción y se limpia la bandera de non-zero transaction count si se requiere.

Finalmente, el roamer queda listo para atender al próximo canal que posea la bandera de transacción en alto.

El escritor y lector determina si las FIFOS están vacías o llenas utilizando banderas. Cada bloque tiene asociada una bandera. El escritor pone esta bandera después que el bloque ha sido escrito y el lector limpia la bandera cuando el último bloque ha sido leído.

La bandera se inicializa cuando el puntero del block escrito realizando un acceso indirecto. El escritor declara que la FIFO de un canal tiene sobre-flujo cuando el escritor trata de escribir un dato en un bloque que tiene la bandera en alto.

La etapa del chip que se encuentra más cerca de la interfaz con el bus PCI es el Controlador DMA, encargado de coordinar la transacción de los paquetes de datos de la Interfaz de Recepción de Paquetes y el almacenamiento subsecuente de los datos en la memoria del Host.

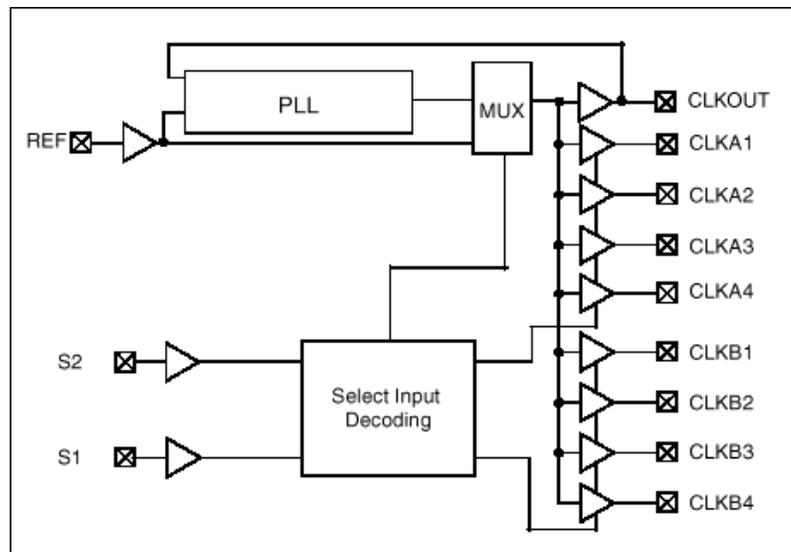
Un paquete puede ser recibido mediante un número separado de transacciones, intervenidas por transacciones provenientes de otros canales de DMA. Una vez que se ha recibido el dato enviado a la memoria del host, el DMA de transacción inicia las transacciones con el objetivo de dar mantenimiento a sus propias estructuras de datos o descriptores.

## 4.8 Descripción del clock-buffer

Con el fin de aumentar la capacidad de carga de las señales de reloj, se va a utilizar un dispositivo encargado de generar copias de la entrada en su salida, con un tiempo de propagación muy corto.

Esto permite tomar un pulso de reloj, asignarlo a la entrada, y alimentar una mayor cantidad de cargas a partir de las réplicas obtenidas en las salidas.

El diagrama de bloques de este circuito se muestra en la Figura 4.10 que corresponde al modelo CY2309 de Cypress. También existe una versión más pequeña, con únicamente 5 salidas, denominado CY2305.



**Figura 4.10** Diagrama de bloques del clock-buffer.

Este buffer permite cumplir con el requerimiento de la señal de reloj de PCI, el cual estipula que CLK no debe estar conectado a más de una carga PCI. Por medio de este chip se puede obtener gran cantidad de réplicas con un tiempo de propagación muy pequeño, para aplicarlas a los diferentes dispositivos que lo requieran dentro de la tarjeta.

# **CAPÍTULO 5**

## **DESCRIPCIÓN DEL SOFTWARE DEL SISTEMA**

Una vez que es diseño del hardware ha sido terminado, es necesario iniciar una etapa estudio y análisis del software que el módulo deberá utilizar para su correcto funcionamiento.

Debe quedar claro que el circuito por si solo no es capaz de realizar ninguna labor coordinada o lógica, sino que requiere de un programa en el cual se coloca los pasos que el diseñador espera realizar al iniciar el funcionamiento del sistema al responder a las diversas situaciones que se puedan dar.

El programa de un circuito puede consistir de varios bloques de código para diferentes procesadores o arreglos lógico-secuenciales, dependiendo de la arquitectura que se tenga presente. En la aplicación que se tiene únicamente se requiere un código, pues la única entidad capaz de procesar programas es el microprocesador.

Por lo tanto, todo el desarrollo del software interno de la tarjeta (firmware) está ligado a la programación del chip 80960-RM. Este procesador posee gran cantidad de unidades periférica integradas, que lo hacen similar a un microcontrolador, sin embargo no cuenta con memoria de programa interna.

Gracias al controlador de memoria que incorpora, se puede colocar un chip de ROM, EPROM, EEPROM o Flash para guardar el programa. Las variables y estructuras de datos que se vaya a utilizar deben colocarse dentro de la memoria RAM interna (que es muy poca) o en un bloque de RAM externa.

Para iniciar el estudio de la programación del este chip, es necesario revisar a fondo el ambiente que el diseñador va a encontrar, según las definiciones que Intel ha dado a su producto.

Se recuerda que los objetivos del proyecto se limitan a los encabezados de arranque y no profundiza en el desarrollo de procedimientos o funciones.

## 5.1 Tipos de dato y direccionamiento

Dentro del ambiente de programación permitido por el código ensamblador del Core i960 se encuentra la definición de tipos de datos, que son los siguientes:

- Integer (con signo; 8, 16 y 32 bits),
- Ordinal (entero sin signo; 8, 16 y 32 bits),
- Long Word (64 bits),
- Triple Word (96 bits),
- Quad Word (128 bits),
- Bit Field,
- Bit.

Debe aclararse que, según la nomenclatura utilizada por la literatura de PCI, la palabra Word se refiere a un bloque de 16 bits (2 bytes), mientras que dentro de las convenciones que se utilizan en el 80960 se estipula que Word significa 32 bits (4 bytes). De esta manera, en la Tabla 5.1, se resume las relaciones entre el nombre dado al bloque de datos y su longitud, de acuerdo con la literatura específica:

**Tabla 5.1** Diferencias de nomenclatura entre PCI y 80960.

Cantidad de Bits	PCI	80960
16	Word	Short-Word
32	Double-Word	Word

En cuanto al direccionamiento, dependiendo de la instrucción particular, se utiliza modos Absoluto, Indirecto por Registro, Indexado con Desplazamiento y de Puntero de Instrucción con Desplazamiento. Estos modos se explican detalladamente en la guía de diseño del chip.

## 5.2 Elementos principales del ambiente

La arquitectura i960 define un ambiente de programación que permite la ejecución de programas, y el almacenamiento y manipulación de datos. Esto incluye los siguientes elementos:

- Un espacio de dirección de 32 bits (hasta 4 Gbyte en total),
- Cache de instrucciones, registros y datos,
- Registros globales y de propósito general,
- Un conjunto de literales,
- Registros de control,
- Un conjunto de registros de estado del procesador.

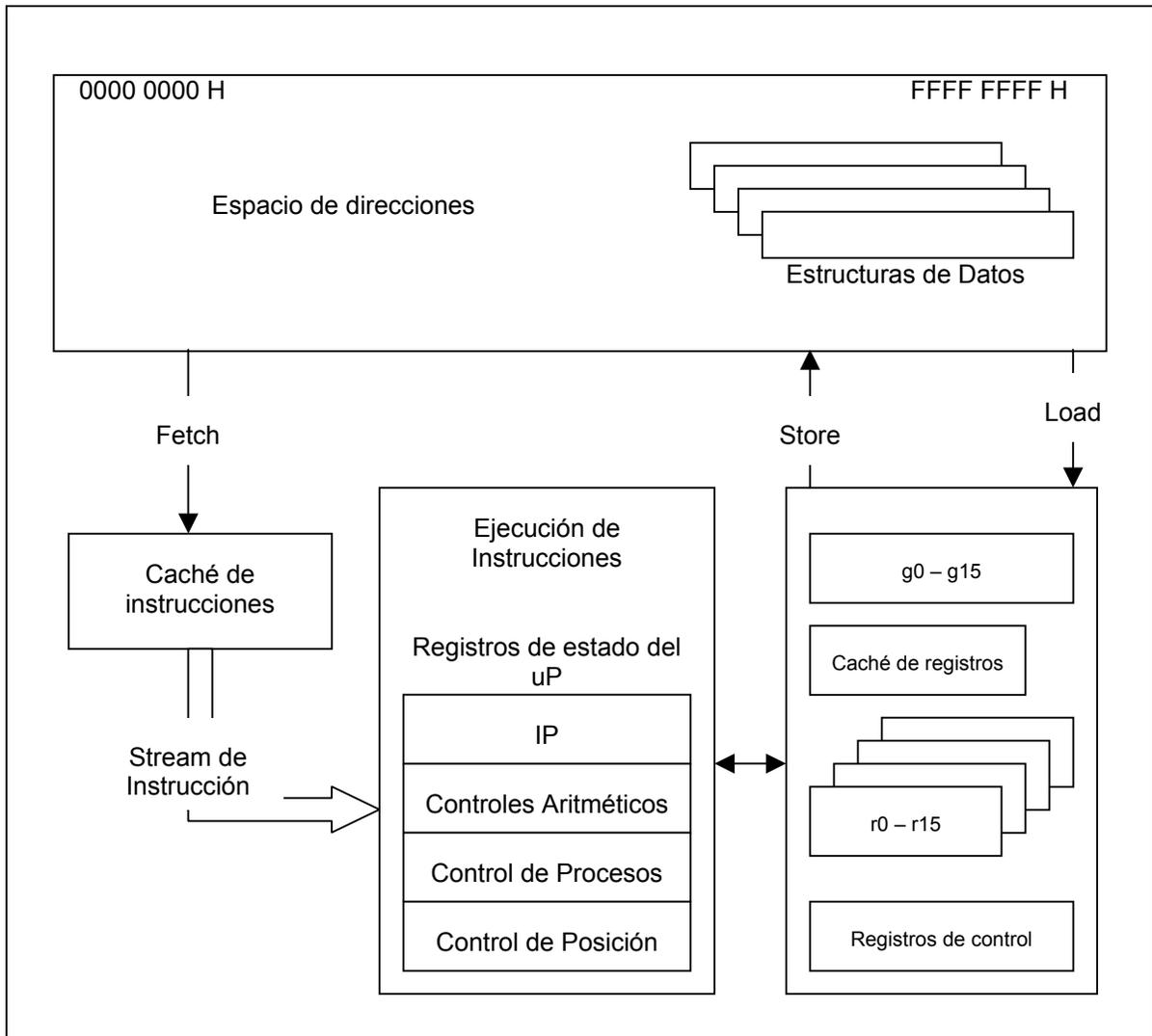
Dentro del espacio de memoria volátil, se incluye el uso de varias estructuras de datos para el manejo de las llamadas a interrupción, los errores y la inicialización.

Estas estructuras son:

- La pila de interrupciones,
- La pila local,
- La pila del supervisor,
- La tabla de control,
- La tabla de fallas,
- La tabla de interrupciones,

- Las tablas de procedimientos del sistema,
- El bloque de control de procesos y
- El registro de iniciación al encendido.

En la Figura 5.1 se muestra una descripción por diagrama de bloques, del ambiente de programación propio del 80960-RM.



**Figura 5.1** Entorno de programación del procesador 80960-RM.

Los registros globales g[0..15] se utilizan como variables temporales para realizar cálculos. Por su parte, los registros locales r[0..15] se usan para almacenar variables locales dentro de un procedimiento. Todos ellos son de 32 bits.

Dependiendo de la instrucción que se está ejecutando, algunos de estos literales son utilizados para almacenar información requerida.

Los direccionamientos indirectos por registro utilizan los literales como base para el cálculo de la celda de memoria que se va a leer o escribir. Esto también se usa cuando se declaran estructuras de datos, por ejemplo en arreglos. Así el programador tiene una mayor libertad para diseñar su código, respaldado por una arquitectura flexible.

### **5.3 Registros de control ubicados en espacio de memoria (MMRs)**

Existe un grupo de registros que se encargan de controlar las funciones y verificar el estado de los periféricos, del núcleo procesador y del bus local. Estos registros no se ubican en un espacio de direccionamiento separado sino que se agregan en una posición específica del bus local dentro del chip.

De esta forma se facilita su lectura y escritura, pues son tratados por el programador como si fueran espacios de memoria. Cuando se da un acceso a estos registros, la arquitectura se encarga de no generar un ciclo de bus externo.

El espacio de los MMRs para los periféricos empieza en 0000 1000 H y finaliza en 0000 18FF H. Las funciones de control del Core o centro de procesamiento se ubican en el rango que va desde FF00 0000 H hasta FFFF FFFF H.

Los registros encargados de controlar los periféricos permiten modificar las configuraciones directamente utilizando las mismas instrucciones escritura que están definidas para la memoria. De forma similar, por medio de lecturas en registros de estado, se puede verificar las condiciones actuales de las unidades.

En el espacio de control de registros se da soporte a las siguientes unidades del chip:

- Controlador DMA,
- Controlador de memoria,
- Controlador de interrupciones externas y de PCI,
- Unidad de mensajería,
- Árbitro interno,
- Puente PCI-PCI,
- Unidad de traslación de direcciones PCI,
- Interfaz con el bus I<sup>2</sup>C,
- Unidad de monitoreo de rendimiento,
- Unidad aceleradora de aplicaciones.

El total de bytes destinados para estas funciones es de 2304, que se encuentran distribuidos en 576 palabras de 32 bits.

Una vez que el procesador ha sido configurado en el proceso de arranque, los MMRs pueden ser modificados tanto desde el Core como desde los buses PCI (a través de las ATU).

Los MMRs no soportan accesos por ráfaga (burst). Dado que únicamente se permite ciclos de bus de 32 bits, la región de memoria donde se encuentran los MMRs debe ser de este ancho. Los registros pueden ser accesibles por bytes desde el programa, gracias al uso de máscaras de validación de bytes.

Esto quiere decir que el ciclo de bus se realiza a 32 bits, pero según las máscaras que estén activas, sólo se utilizará un determinado segmento del bloque. En algunos registros se tiene restricciones a la norma, y es un requerimiento el uso de los 32 bits.

Desde este espacio de memoria se puede realizar lecturas de información que por lo general está restringida a los ciclos de configuración:

- Vendor ID,
- Device ID,
- Revisión ID,
- Class Code,
- Header Type
- Bridge subsystem ID,
- Bridge subsystem vendor ID.

La arquitectura no define un espacio dedicado de I/O, sino que espacios del mapa de memoria donde se pueden ubicar estas secciones, siempre y cuando se le indique al procesador, para que los ciclos tengan un tiempo de espera mayor.

El mapa del bus local está compuesto por:

- a. Un encabezado (NMI-Vectores de interrupción opcionales-RAM interna).
- b. Espacio reservado.
- c. Registros de periféricos.



### 5.3.1 Programación de los registros del CORE

Los MMRs reservados para el Core del procesador sirven para configurar el bus local, incluyendo la habilitación del caché para un cloque dado de la memoria, el ancho del bus y la interfaz BIU.

El primer aspecto que se debe definir corresponde a los atributos de la memoria que pueden ser lógicos y físicos.

Los atributos físicos se refieren a los aspectos relacionados con las señales de control que el procesador debe utilizar y el ancho del bus. Los atributos lógicos se refieren a la forma de interpretar los datos, el formato y el control de caché. Para controlar los atributos físicos se usa los registros PMCON y BCON.

PMCOM significa configuración física de memoria, y comprende los MMRs PMCON0\_1 al PMCON14\_15.

Dentro de la arquitectura, el BIU espera todos los accesos salientes del Core destinados a una región de 32 bits (4 GBytes). Los ocho registros PMCON son leídos automáticamente durante la inicialización y sus valores se guardan en el IBR.

Inmediatamente después de un reset de hardware, el contenido de los registros PCOM es marcado como inválido en el registro de control del bus BCON. Cuando esta situación se mantiene, el valor de PMCON14\_15 se toma como base para llenar los demás campos.

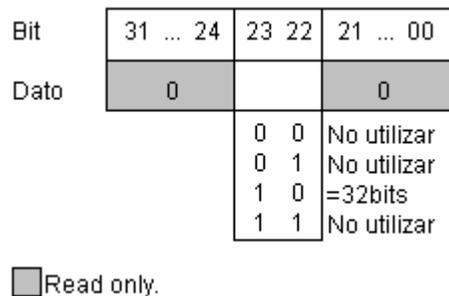
PMCON14\_15 es borrado cuando ocurre el reset de hardware, con lo cual el bus local queda configurado en un ancho de 8 bits. El procesador debe leer los PMCON para copiar sus valores en la tabla de control, o en el caso de que el bit 2 de BCON sea cero, se saca una copia de PMCON14\_15 y se escribe en todos los campos de la tabla de control.

La distribución del espacio de memoria que cada PMCON se encarga de controlar se muestra en la Tabla 5.2.

**Tabla 5.2** Regiones controladas por los registros PMCON.

Registro	Región controlada	Ancho del bus
PMCON0_1	0000 0000 H a 0FFF FFFF H & 1000 0000 a 1FFF FFFF H	32 bits / periféricos
PMCON2_3	2000 0000 H - 2FFF FFFF H & 3000 0000 H - 3FFF FFFF H	32 bits
PMCON4_5	4000 0000H - 4FFF FFFFH & 5000 0000H - 5FFF FFFFH	32 bits
PMCON6_7	6000 0000H - 6FFF FFFFH & 7000 0000H - 7FFF FFFFH	32 bits
PMCON8_9	8000 0000H - 8FFF FFFFH & 9000 0000H - 9FFF FFFFH	32 bits / outbound / ATU window
PMCON10_11	A000 0000H - AFFF FFFFH & B000 0000H - BFFF FFFFH	32 bits
PMCON12_13	C000 0000H - CFFF FFFFH & D000 0000H - DFFF FFFFH	32 bits
PMCON14_15	E000 0000H - EFFF FFFFH & F000 0000H - FFFF FFFFH	32 bits

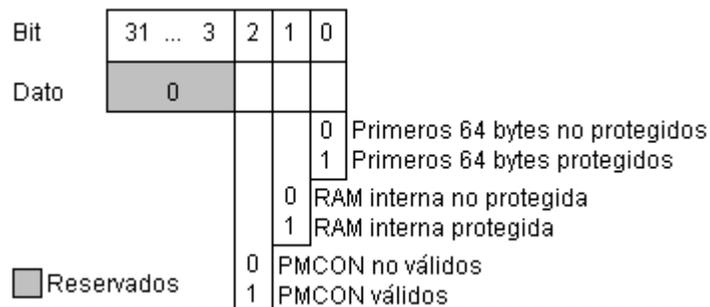
La estructura de estos registros se muestra en la Figura 5.3. Este procesador en particular debe ser colocado en modo 32 bits, para su correcto funcionamiento. Durante las primeras rutinas de iniciación, el bus local se mantiene en 8 o 16 bits, sin embargo esto no forma parte del ambiente de programación. La opción de colocar el bus en otros anchos se debe a cuestiones de compatibilidad.



**Figura 5.3** Estructura de los registros PMCON.

El valor recomendado para estos registros es 0080 0000 H, que configura el bus para trabajar con bloques de memoria organizados en palabras de 32 bits.

Con respecto al registro BCON, su estructura se muestra en la Figura 5.4.



**Figura 5.4** Estructura del registro BCON.

El i960 puede trabajar en dos modos: usuario y supervisor. El supervisor está en capacidad de bloquear los cambios que el usuario pueda hacer en la configuración de la RAM y los primeros bytes del mapa. Para esto se usa BCOM[0-1].

Continuando con los atributos lógicos de la memoria, se provee un mecanismo para definir dos tipos de memoria (LMTs) que básicamente determinan si un bloque de memoria puede ser cacheable o no.

Típicamente hay diversos LMTs definidos dentro de un solo subsistema de memoria. Por ejemplo, datos dentro de un área de la SDRAM pueden utilizar la caché, mientras que a otras partes de la misma no se les permite.

Con el objetivo de crear la delimitación de las regiones lógicas se utiliza los Registros de Configuración Lógica (LMCON).

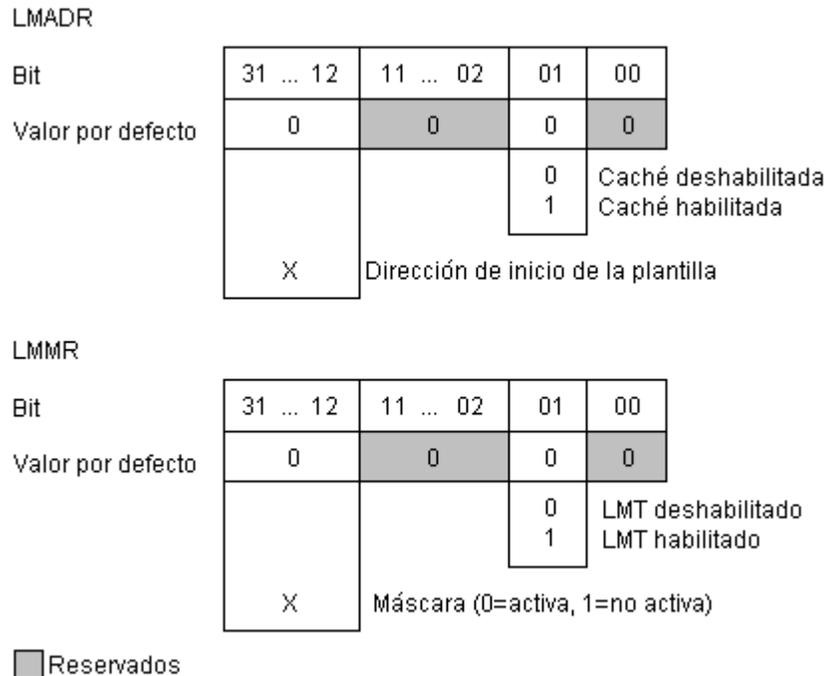
Un par LMCON define un modelo de memoria para áreas determinadas por sus valores. El 80960RM posee dos pares de registros LMCON. La extensión de una plantilla (área de memoria con formato lógico dado) se basa en una posición de inicio y una máscara de dirección.

Fuera de las plantillas que se definen por los LMCON, se utiliza la lógica por defecto, que se configura en DLMCON.

Cada LMCON está formado por los registros LMADR (base) y LMMR (máscara). La estructura de estos campos se muestra en la Figura 5.5.

El bit 1 de los LMMR define si el LMT está o no funcionando. Si no está habilitado, el bloque de memoria correspondiente se rige según la configuración guardada en DLMCON. El uso de caché para las instrucciones no se altera por medio de estos campos de control.

Para obtener la región que va a ser cubierta por el LMT, se toma la dirección presente en la instrucción y se enmascaran los bits del 12 al 31 según exista máscara activa (cero) o no, en la posición correspondiente del LMMR. El resultado se compara con la base LMADR y si concuerdan, la localidad de memoria se considera parte de la plantilla.



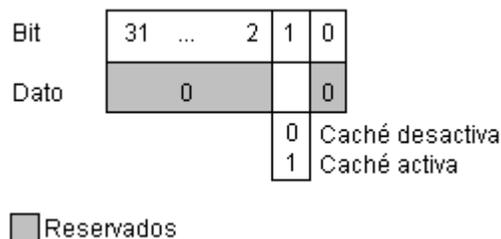
**Figura 5.5** Estructura de los registros que forman el LMCON.

La operación lógica que describe este proceso es la siguiente:

$$[(EFA < 12..31 >) \text{XNOR}(\text{LMADR}_x < 12..31 >)] \text{OR}(\text{not LMMR}_x < 12..31 >)$$

donde EFA corresponde a la dirección de la celda que se está utilizando en la instrucción.

Por su parte, el registro de control DLMCON se muestra en la Figura 5.6.



**Figura 5.6** Estructura de los registros que forman el LMCON.

Como ejemplo, para comprender mejor el funcionamiento de las máscaras, supóngase que se desea una región de 64 Kbytes a partir e 0010 0000 H. La posición final sería 0010 FFFF H, por lo tanto todas las direcciones de memoria que coinciden en sus 16 bits más significativos con el registro base, deben formar parte de la plantilla.

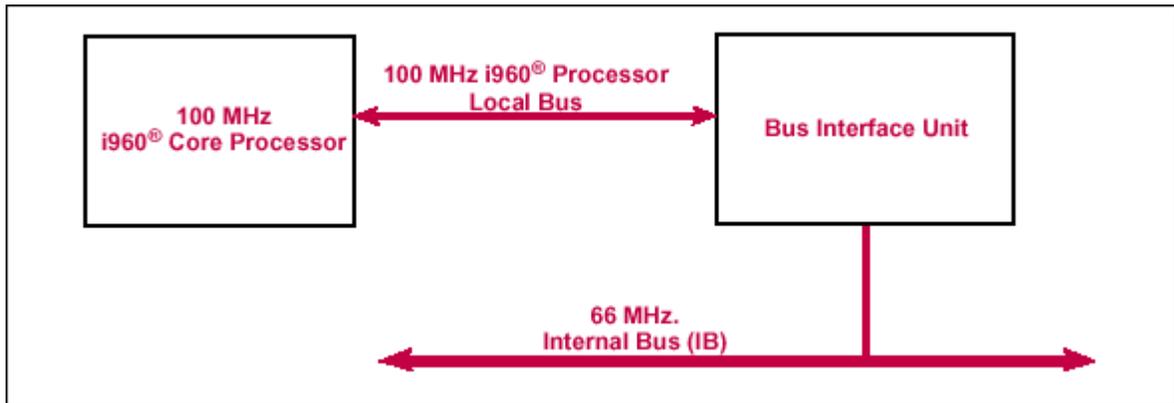
De esta forma, LMADR[12..31] = 0010 0xxx H y LMMR[12..31] = FFFF 0xxx H.

Si el programador utiliza hábilmente las características de la máscara puede definir espacios de plantillas complejos, que no necesariamente usan celdas de memoria adyacentes.

### **5.3.2 Programación de los registros del BIU**

Para programar el BIU es importante entender dónde se ubica el bus local dentro de la arquitectura, para diferenciarlo del bus interno (conexión entre las unidades) y los buses externos (PCI y memoria). En la Figura 5.7 se muestra la forma en que se interconectan los buses local e interno a través del BIU.

El BIU es el encargado de establecer un lazo entre los componentes de un sistema 80960 y el Core, permitiendo al procesador leer las instrucciones, manipular datos e interactuar con los recursos disponibles en su ambiente de periféricos. Para incrementar el rendimiento, el procesador permite accesos en ráfaga, que sirven para establecer un flujo continuo de paquetes de 32 bits en forma automática.



**Figura 5.7** Ubicación del bus local dentro de la estructura del 80960-RM.

Como ya se mencionó en la programación de los atributos de memoria, el 80960-RM no trabaja con un bus local diferente a 32 bits de ancho, excepto al momento de arrancar el sistema (cuando se lee el IBR).

Se tiene entonces un bus interno de 64 bits, a 66 MHz, y un bus local de 32 bits que puede trabajar hasta 100MHz. Ambos buses se encuentran dentro del chip, pero están aislados entre sí por el BIU.

Para realizar la sincronización entre los ciclos del bus interno y el bus local, se cuenta con un conjunto de buffers, que brindan un almacenamiento temporal de la información. Gracias al uso del BIU, se permite realizar precarga de instrucciones del lado del bus interno, mientras el bus local se encuentra realizando otras operaciones.

No se realiza ninguna modificación en el valor de las direcciones, sino únicamente la interfaz entre 64 bits y 32 bits. El bus interno no soporta más de 4 Gbyte de espacio de direccionamiento, así que no hay pérdida de bits en este traslado. En cuanto al intercambio de datos, cuando el bus interno es de 64 bits los datos pasan en dos bloques, y el BIU se encarga de coordinar esto con el Core.

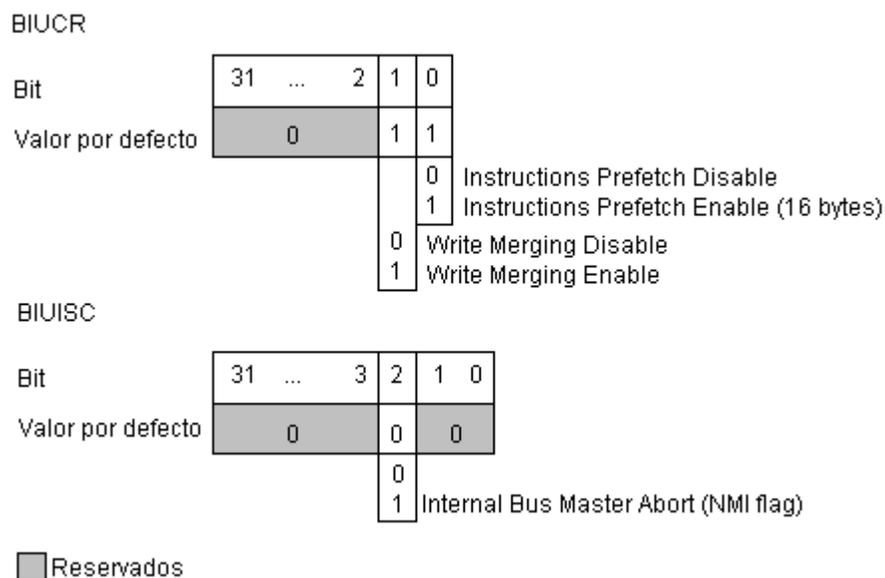
Otras tareas importantes del BIU son:

- Write Buffering,
- Instruction Fetch Bypass,
- Instruction Prefetch,
- Write Merging.

Estas se encargan de mejorar la eficiencia en el flujo de la información.

Existe un dispositivo opcional denominado MTT que, dependiendo de su programación puede utilizar el BIU por unos cuantos ciclos de reloj.

Los MMRs propiedad del BIU son: el Registro de control de BIU (BIUCR) y el Registro de estado de interrupciones BIU (BIUISR). En la Figura 5.8 se muestra su estructura.



**Figura 5.8** Estructura de los MMRs usados por el BIU.

El BIUCR inicia por defecto en la configuración que permite una mayor eficiencia. Por su parte, el BIUISR es afectado cuando se da un Master Abort desde el bus interno, y activa la NMI del procesador.

### **5.3.3 Programación de los registros del puente PCI-PCI**

El puente PCI a PCI tiene dos estados de reset independientes, una para cada interfaz de bus (primaria y secundaria). La señal S\_RST# se obtiene como la operación OR entre el P\_RST# y el bit “Secondary Bus Reset” en el registro BCR. La señal S\_RST# es asíncrona con el pulso de reloj S\_CLK.

Cuando el bit SBR es colocado en alto y llegado cierto momento se baja (reset por software), el 80960-RM puede ser programado para activar una interrupción en el Core. Esto se logra manejando adecuadamente el valor del “Secondary Decode Enable Register” (SDER).

Durante la secuencia de reset (no más de tres ciclos de reloj desde el encendido de la señal P\_RST#), la Unidad de Arbitración del Secundario debe parquear el bus secundario.

Para que la Unidad Puente opere correctamente en un sistema dado, algunos puntos básicos deben darse como proceso de inicio.

El procedimiento que se señala a continuación es requerido para todos los puentes PCI-PCI:

- a. El Número del Bus Primario, el Número del Bus Secundario y el Número del Bus Subordinado deben ser programados, con valores válidos. Esto debe ser hecho para permitir al software de configuración probar el espacio en los buses descendentes.

- b. Si se va a trasladar accesos de I/O descendentes, los registros IOBR e IOLR deben ser programados con los valores adecuados, y luego el bit de habilitación de espacio I/O debe ser activado en PCR. Si los puertos I/O incluyen espacio ISA, el bit de habilitación de bus ISA se debe activar en el registro BCR.
- c. Si se va a trasladar accesos de memoria descendentes, los registros correspondientes a rangos de I/O Mapeados en Memoria y Memoria Prefetchable (MBR/MLR y PMBR/PMLR respectivamente) pueden ser programados con los valores adecuados para abrir sus ventanas de acceso. Si únicamente se requiere un tipo de rango, entonces se puede programar los mismos valores en ambos pares. Una vez que ya se ha establecido las ventanas, se enciende el bit habilitador de memoria en el registro PCR.
- d. Si los buses inferiores poseen masters, se debe activar el habilitador de Bus Master en el PCR. Una vez que este bit es activado, todos los accesos de memoria e I/O en el secundario que no se ubican dentro de las ventanas definidas, son trasladados al primario. Si los bits de habilitación de los rangos no han sido activados, todos los ciclos son trasladados hacia el primario.
- e. Los registros CLSR, PLTR y SLTR deben ser programados con el valor adecuado antes de que el puente sea totalmente funcional. La mayoría los sistemas encienden el bit habilitador de SERR# tanto en PCR como en BCR.
- f. El bit "Configuration Cycle Retry" en el registro EBCR debe ser borrado para permitir que el host configure el bridge.

Esta lista constituye la base mínima requerida para iniciar el puente. Queda bajo la responsabilidad del software de configuración activar o desactivar la base adicional y los recursos específicos del procesador.

Es importante señalar que si el puente está utilizando dispositivos privados en el secundario, y sus entradas IDSEL se encuentran en S\_AD[16..25], entonces el registro de selección IDSEL debe ser programado antes que el software de configuración del sistema pruebe el bus secundario.

Para describir los registros relacionados con el puente es necesario aclarar que hay dos espacios encargados de guardar la información concerniente a esta unidad.

Primeramente se tiene el espacio de configuración, que cumple con los requisitos establecidos para puentes PCI-PCI por las especificaciones para este bus. Los registros ubicados en este espacio son leídos o escritos desde los buses por medio de ciclos de configuración.

El segundo espacio de registros corresponde al rango definido para el puente dentro del mapa de bus interno (MMRs). Esto permite al Core definir características de trabajo del puente y chequear su estado. Los MMRs del puente también pueden ser utilizados por los buses si la configuración del ATU lo permite.

En el caso del 80960-RM, todos los registros del espacio de configuración excepto el “Centralized Discard Timer” se han ubicado también como MMRs, para que el Core tenga acceso a ellos desde el bus interno.

Si se utiliza un enfoque desde el espacio de configuración, se dispone de registros de 8, 16 y 32 bits, en un formato predefinido. Todos los puentes PCI-PCI disponibles en el mercado deben satisfacer un estándar en el diseño de este espacio, con el fin de mantener la compatibilidad.

En la Tabla 5.3 se muestra cual es el formato que lleva el espacio de configuración. Lo que se encuentra bajo el offset 3EH corresponde al molde dado por la revisión 1.0 de la arquitectura para puentes PCI-PCI mientras que los campos sobre el por encima de 3EH son controles específicos del 80960RM.

**Tabla 5.3** Formato del encabezado de configuración del puente.

Disposición de los registros				Offset de Direcciones PCI	
ID de dispositivo		ID de fabricante		00H	Estándar para los puentes PCI-PCI
Estado del primario		Comando del primario		04H	
Código de clase			ID de Revisión	08H	
Reservado	Tipo de encabezado	Tiempo de latencia del primario	Tamaño de línea caché	0CH	
Reservado				10H	
Reservado				14H	
Tiempo de latencia del secundario	Número del bus subordinado	Número del bus secundario	Número del bus primario	18H	
Estado del secundario		Límite I/O	Base I/O	1CH	
Límite de memoria		Base de memoria		20H	
Límite de memoria prefetchable		Base de memoria prefetchable		24H	
Reservado				28H	
Reservado				2CH	
Reservado				30H	
ID de subsistema		ID de fabricante de subsistema		34H	
Reservado				38H	
Control del puente		Reservado		3CH	
Control de IDSEL de secundario		Control extendido del puente		40H	
Estado de interrupciones del primario				44H	Específicos del 80960-RM
Estado de interrupciones del secundario				48H	
Control de arbitración del secundario				4CH	
Control de rutas de interrupciones PCI				50H	
Reservado		Límite de I/O en el secundario	Base de I/O en el secundario	54H	
Límite de memoria en el secundario		Base de memoria en el secundario		58H	
Control de colas		Habilitador de decodificador de secundario		5CH	
Temporizador centralizado para descartar				60H	

Cuando sucede un reset en el primario la mayoría de los registros en esta zona vuelven a su valor por defecto. Por otro lado, el único bit que se afecta con un reset interno (por software) es EBCR[5].

Desde el punto de vista de los registros ubicados en el mapa de direcciones del bus interno, es posible realizar un manejo de la configuración del puente por medio de lecturas y escrituras simples.

De esta forma, a menos que la ATU lo permita, el espacio de configuración será accesible a los buses PCI sólo por ciclos de configuración, mientras que para el Core le es permitido utilizarlo por medio de accesos a los MMRs.

Según el tipo de ciclo que se utiliza, los atributos de un registro particular cambian. Por ejemplo, el campo “Device ID Register” está marcado como read/write para el bus interno, sin embargo es de tipo read only si se utiliza un ciclo de configuración desde los buses PCI.

En las siguientes tablas se expone los MMRs propiedad del puente PCI-PCI.

**Tabla 5.4** Conjunto de MMRs usados por la Unidad Puente PCI-PCI.

Dirección en el bus interno	Nombre	Acrónimo
1000 H	Registro de identificación de fabricante	VIDR
1002 H	Registro de identificación de dispositivo	DIDR
1004 H	Registro de comando del primario	PCR
1006 H	Registro de estado del primario	PSR
1008 H	Registro de identificación de revisión	RID
1009 H	Registro de código de clase	CCR
100C H	Registro de tamaño de línea caché	CLSR
100D H	Registro temporizador de latencia en el primario	PLTR
100E H	Registro de tipo de encabezado	HTR
1018 H	Registro de número del bus primario	PBNR

**Tabla 5.5** Conjunto de MMRs usados por la Unidad Puente PCI-PCI (cont.).

Dirección en el bus interno	Nombre	Acrónimo
1019 H	Registro de número del bus secundario	SBNR
101A H	Registro de número del bus subordinado	SubBNR
101B H	Registro temporizador de latencia en el secundario	SLTR
101C H	Registro base de I/O	IOBR
101D H	Registro límite de I/O	IOLR
101E H	Registro de estado del bus secundario	SSR
1020 H	Registro base de memoria	MBR
1022 H	Registro límite de memoria	MLR
1024 H	Registro base de memoria prefetchable	PMBR
1026 H	Registro límite de memoria prefetchable	PMLR
1034 H	Registro de ID del fabricante del subsistema	BSVIR
1036 H	Registro de ID del subsistema	BSIR
103E H	Registro de control del puente	BCR
1040 H	Registro extendido de control del puente	BECR
1042 H	Registro de selección del IDSEL en el secundario	SISR
1044 H	Registro de estado de interrupciones en el primario	PBISR
1048 H	Registro de estado de interrupciones en el secundario	SBISR
104C H	Registro de control del árbitro en el secundario	SACR
1050 H	Registro de selección de rutas para interrupciones PCI	PIRSR
1054 H	Registro base de I/O en el secundario	SIOBR
1055 H	Registro límite de I/O en el secundario	SIOLR
1058 H	Registro base de memoria en el secundario	SMBR
105A H	Registro límite de memoria en el secundario	SMLR
105C H	Registro habilitador de la decodificación en el secundario	SDER
105E H	Registro de control de cola	QCR

Con el fin de describir mejor el uso de estos registros, se ha tomado los más importante para una revisión detallada de su función. Para detalles completos se hace referencia al Manual de Desarrollador del 80960-RM dado por Intel.



## Registro de estado del primario (PSR)

Contiene las banderas básicas que indican el estado de la interfaz con el primario. La Figura 5.10 muestra su composición por bits.

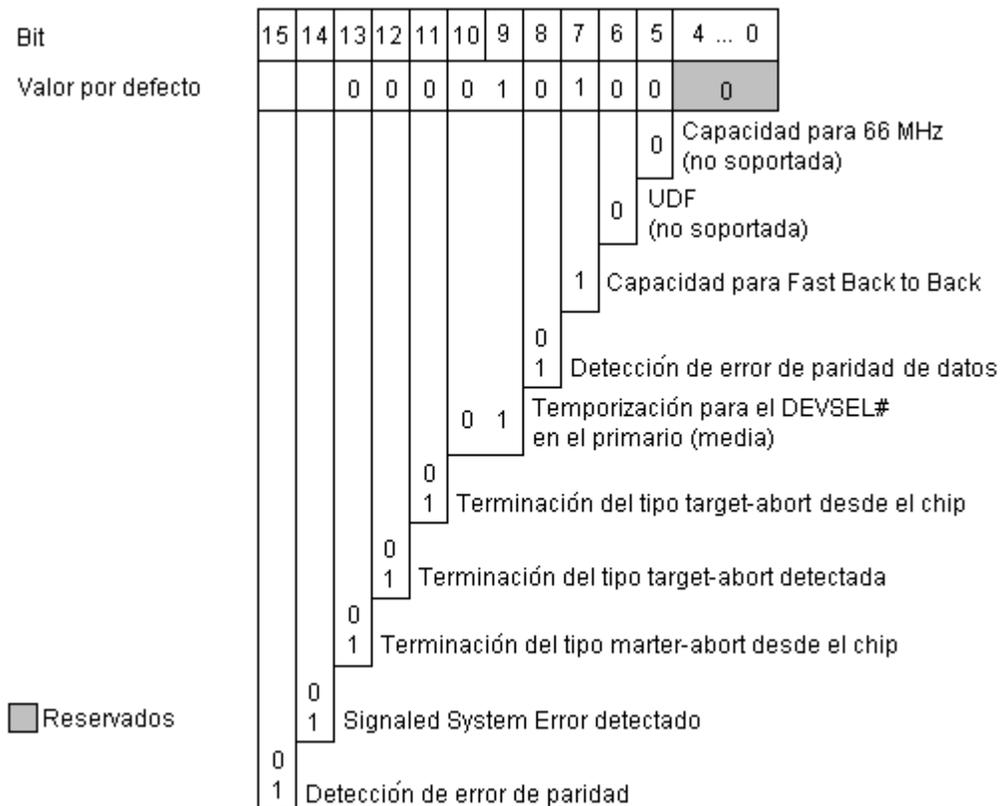


Figura 5.10 Estructura del registro PSR.

## Registro de Código de Clase (CCR)

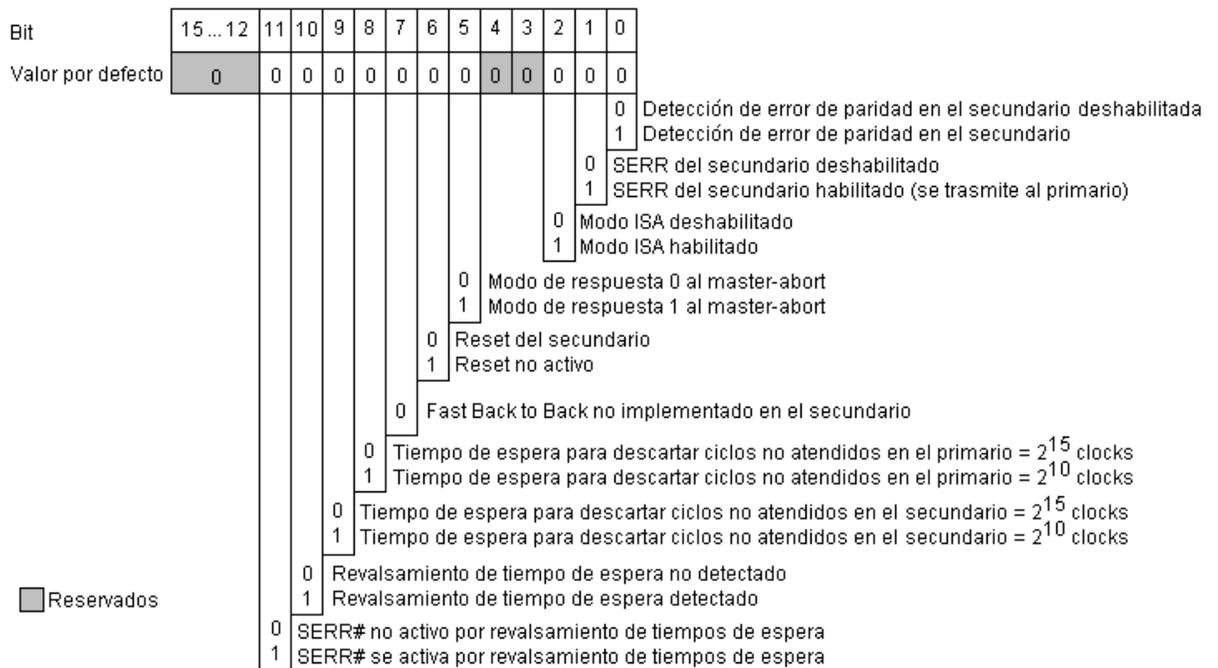
Este registro se encarga de decirle al software de configuración el tipo de función presente en el dispositivo PCI. En la Tabla 5.6 se muestra cual es el código de clase propio del circuito 80960-RM.

**Tabla 5.6** Configuración del código de clase para el 80960-RM

Bits	Valor por defecto	Descripción
16..23	06 H	Clase base: Puente
8..15	04 H	Subclase: Puente PCI-PCI
0..7	00 H	Interfaz de programación

### Registro de control del puente (BCR)

Este registro provee una extensión al PCR, con controles para funciones de ambos buses. En la Figura 5.11 se muestra el desglose de sus bits.



**Figura 5.11** Estructura del registro BCR.

Aquí se ha abarcado las funciones más básicas del espacio de configuración, sin embargo el programado tiene gran cantidad de opciones adicionales que puede utilizar para realizar rutinas específicas.

## 5.4 Procedimiento de inicio del 80960-RM

El arranque del procesador se puede dividir básicamente en dos secciones: inicio del Core, e inicio de las demás unidades. Existe cuatro modos de inicio disponibles, que se seleccionan por medio de los valores presentes en RST\_MODE# y RETRY cuando la señal P\_RST# está en bajo. Estos modos definen cuando se inicia el Core y cuando el primario inicia el traspaso de ciclos.

Cuando el Core arranca, el procesador utiliza una Imagen Inicial de Memoria (IMI) para establecer su estado. La IMI incluye:

- El registro de inicio (IBR), que contiene la dirección de la primera instrucción en el código del usuario y el PRCB.
- El bloque de control de procesos (PRCB), que contiene los punteros a las estructuras de datos del sistema y otra información usada por el procesador para iniciar su trabajo.
- Las estructuras de datos del sistema, que son cargadas en el inicio para un uso más rápido de ellas durante el programa.

El momento justo en que empieza el proceso de iniciación es cuando la señal P\_RST# pasa de un nivel de tensión bajo a alto. Aquí se elige el modo de arranque según lo que se establece en la Tabla 5.7.

**Tabla 5.7** Modos de iniciación del 80960-RM.

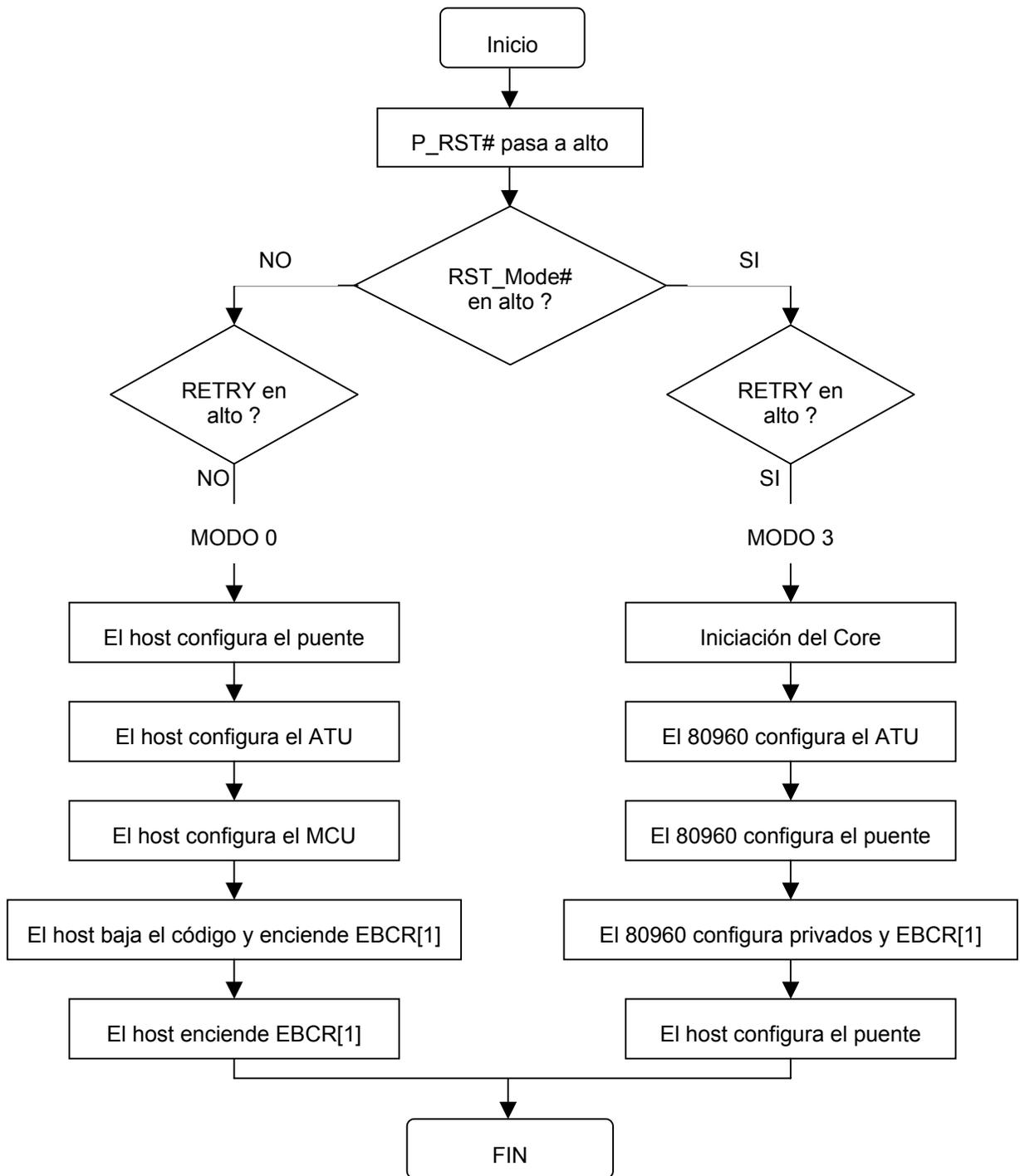
RST_MODE#	RETRY	Modo de inicio	Interfaz PCI Primaria	I960 CORE
0	0	Modo 0	Acepta transacciones	Reset
0	1	Modo 1	Reintenta todas los ciclos de configuración	Reset
1	0	Modo 2	Acepta transacciones	Inicia
1	1	Modo 3 (por defecto)	Reintenta todas los ciclos de configuración	Inicia

En el 80960-RM no se recomienda el uso de los modos 1 y 2.

El modo por defecto es el 3, y se adapta muy bien para la mayoría de las aplicaciones. Este modo permite que el procesador programe sus periféricos antes de aceptar cualquier ciclo de configuración desde el primario. De esta forma se puede controlar la forma en que el procesador se presenta ante el software de configuración del host.

En el caso del modo 0, se permite la entrada a ciclos de configuración desde el primario antes de iniciar al Core. Así es el software del host quien obtiene el control sobre el procesador y se encarga de programar sus registros básicos, asignarle sus recursos dentro del sistema y ubicarlo dentro de los dispositivos detectados.

En la Figura 5.12 se muestra el procedimiento de iniciación.



**Figura 5.12** Diagrama de flujo para la iniciación del procesador.

# **CAPÍTULO 6**

## **ANÁLISIS DE RESULTADOS**

## 6.1 Explicación del diseño

En la Figura 2.12 se presenta el modelo del sistema en diagrama de bloques. Este diseño se ha basado en el chip controlador de protocolo HDLC, el PM-7366, que ya había sido elegido por Cibertec antes de iniciar el proyecto. Tomando como base las características periféricas de este circuito, se inició la búsqueda de la estructura más recomendada para la interfaz con el bus PCI.

Fue necesario realizar un análisis de factibilidad para determinar la mejor opción para conectarse al bus. Como resultado de tal estudio se obtuvo la siguiente información:

**Tabla 6.1** Factibilidad: Uso de un bridge disponible en el mercado.

Costo	Circuito integrado Puente PCI/PCI de Intel, 21152: <u>\$57.91 por unidad</u> (Arrow Electronics). Microprocesador de Intel con Puente PCI/PCI incorporado, 80960-RM: <u>\$86.32</u> (Avnet-Marshall).
Características Principales	<ul style="list-style-type: none"> <li>• Cumple completamente con la Revisión 2.1 de la Especificación para el Bus Local PCI.</li> <li>• Cumple completamente con la Revisión 1.1 de la Especificación para Arquitectura de Puentes PCI-a-PCI.</li> <li>• Cumple completamente con la Especificación para “Advanced Configuration Power Interface” (ACPI).</li> <li>• Cumple completamente con la Revisión 1.1 de la Especificación para Manejo de Potencia PCI. • Soporta ambientes de señales tanto en 5 V como en 3.3 V.</li> </ul>
Funciones	<ul style="list-style-type: none"> <li>• Implementación de transacciones “retardadas” para todos los tipos de lectura: configuración, memoria e I/O; hasta tres transacciones simultáneas en cada dirección.</li> <li>• Provee una lógica para arbitrar el bus secundario.</li> <li>• Permite todo tipo de transacciones:             <ul style="list-style-type: none"> <li>- Todos los comandos de I/O y memoria.</li> <li>- Comando de configuración de Tipo 1 a Tipo 1.</li> <li>- Comandos de configuración de Tipo 1 a Tipo (únicamente hacia el bus inferior).</li> <li>- Todos los comandos configuración de Type 1 ciclo especial.</li> </ul> </li> </ul>

**Tabla 6.2** Factibilidad: Uso de un bridge disponible en el mercado (cont.).

Requisitos	<ul style="list-style-type: none"> <li>• Es necesario seguir en cada aspecto del diseño las recomendaciones hechas para trabajar con PCI sin que se den problemas de reflexión y pérdidas en las pistas. Para este caso esto se limita a la tarjeta (PCB), dentro de la cual aplica el largo y ancho de las pistas, los condensadores de desacople, las resistencias para disminuir los reflejos y las capas específicas para la alimentación.</li> <li>• El tipo de ensamblaje deberá ser de montaje superficial, o por medio de una base especial.</li> <li>• Las señales de reloj deben estar adecuadamente sincronizadas para lo cual se exige un largo de pistas similar para cada señal de CLK.</li> </ul>
Potencia	Aproximadamente 1,2 Watts, con alimentación de 3.3V.
Dimensiones	<p>El 21152 vienen en un chip de 160 pines tipo PQFP. Longitudes máxima en ancho y largo: 31,20 mm BSC.<sup>2</sup></p> <p>El 80960-RM es un circuito integrado de 540 pines tipo H-PBGA. Longitudes máxima en ancho y largo: 42.50 mm BSC.</p>
Velocidad	<p>El circuito 2152 está desarrollado para trabajar a 33 MHz, con una restricción: la pista del reloj debe tener una longitud de 2,5 (<math>\pm 0,1</math>) pulgadas desde el conector de la tarjeta.</p> <p>Por su parte el 80960-RM se basa en el reloj del PCI (33 MHz) y por medio de PLLs internos puede multiplicar esta frecuencia por 2 ó 3.</p>
Actividades relacionadas con el desarrollo de la aplicación	<p>Investigación.</p> <p>Adquisición de licencias.</p> <p>Conexión de pines.</p> <p>Programación.</p> <p>Diseño del impreso.</p> <p>Compra.</p> <p>Montaje y pruebas.</p>
Personal requerido	<p>Diseñador.</p> <p>Dibujante.</p> <p>Proveedor.</p> <p>Fabricante de impresos.</p> <p>Personal para el montaje.</p> <p>Personal para pruebas.</p>

<sup>2</sup> En el Estándar ANSI para Dimensionamiento y Tolerancia, Sección 1.3.2, se define Dimensión Básica (BSC) como: Un valor numérico usado para describir el tamaño exacto teórico, perfil, orientación o ubicación de la característica o dato de un elemento. Esta es la base desde la cual se permiten variaciones establecidas por la tolerancia en otras dimensiones, en notas o en cuadros de control de características.

**Tabla 6.3** Factibilidad: Uso de circuitería no especializada.

Costo	<p>Un paquete de 100 GALs 22V10 en Digikey cuesta \$429.00, de forma que el costo por unidad queda en \$4.29.</p> <p>Los buffers tri-state y los latches de 8 bits cuestan \$60 en bloques de 100 unidades, ambos en Digikey.</p> <p>Un aproximado de componentes para realizar el bloque de intercomunicación es: 6 latches, 6 buffers, y 4 GALs, aproximadamente. Esto hace un total de \$24.36, sólo en los componentes básicos.</p>
Funciones	Básicamente debe estar en capacidad de realizar las operaciones de configuración y los comandos de lectura y escritura en un pequeño espacio de I/O.
Requerimientos	Es necesario realizar un diseño cuidadoso y detallado, para que los tiempos de propagación en los chips no vayan a alterar la sincronización de las señales. A esto se agrega la distribución y el largo del cableado, que no debe salirse de los requerimientos eléctricos de PCI.
Potencia	Depende de la cantidad de componentes que se utilicen. Estos elementos trabajarán posiblemente con tecnología de 5V.
Dimensiones	Cada GAL22V10 mide 30x10 mm en DIP de 24 pines o bien 15x15 mm en PLCC de 28 pines, mientras que los buffer y los registros tienen una dimensión de 26x10 mm.
Velocidad	Para evitar problemas de pérdidas se debe trabajar a una frecuencia máxima de 33 MHz, teniendo en cuenta todas las consideraciones necesarias en la colocación y distribución de las pistas.
Actividades relacionadas con el desarrollo de la aplicación	<p>Investigación.</p> <p>Diseño del diagrama funcional.</p> <p>Cumplimiento de las especificaciones de PCI.</p> <p>Búsqueda de los componentes.</p> <p>Conexión interna.</p> <p>Diseño y distribución de las pistas entre componentes.</p> <p>Diseño de las rutinas de control.</p> <p>Programación.</p> <p>Conexión con su entorno.</p> <p>Diseño del impreso.</p> <p>Compra de partes.</p> <p>Montaje y pruebas.</p>
Personal requerido	<p>Diseñador.</p> <p>Programador.</p> <p>Dibujante.</p> <p>Proveedor.</p> <p>Fabricante de impresos.</p> <p>Personal para el montaje.</p> <p>Personal para pruebas.</p>

De las tres opciones analizadas, fue el uso del microprocesador con puente incorporado la que surgió como la más apropiada para el proyecto.

Cuando este proceso de comparación estaba casi completo apareció un factor que no estaba contemplado hasta el momento: el tipo de RAM que debería tener el módulo.

La primera opción que se había explorado era utilizar SRAM (estática) para el trabajo que debe realizar el CPU local de la tarjeta. Esta memoria no necesita refrescamiento y es la más sencilla de conectar, sin embargo es la más cara.

Hoy día, la aplicación más explotada de la SRAM es para memoria caché. Los módulos de 4 MB (1024K x 32) en presentación SIMM, marca Cypress, valen \$463.50 @15ns y \$408.13 @35ns. Estos precios son muy altos, y es muy probable que en los meses próximos este tipo de memoria sea descontinuado. Dado lo anterior, se descartó el uso de SRAM.

La memoria DRAM (dinámica) es mucho más barata, pero necesita refrescamiento. Esto se realiza mediante una lógica de control especial se incorpora en la mayoría de procesadores actuales.

Existen dos tipos de DRAM: la sincrónica y la asíncrona. La asíncrona se clasifica básicamente en memoria EDO (Extended Data Out) y FPM (Fast-Page-Mode), mientras que la sincrónica se conoce como SDRAM.

La memoria EDO y FPM se encuentra en presentaciones DIMM de 4 MB en adelante, sin embargo su producción está prácticamente descontinuada.

La SDRAM es el tipo de memoria vigente en el mercado. Básicamente se diferencia de los otros tipos de RAM dinámica, en que utiliza un reloj para sincronizar todas sus señales. Esta es la memoria que se vende con los equipos de cómputo en la actualidad.

Se encontró en el mercado módulos DIMM de 4 MB mínimo, con buses de 32 y 64 bits, y velocidades desde 33 MHz hasta más de 100 MHz. Según los precios localizados en Arrow Electronics Inc, 4 MB de SDRAM (1024K x 32) cuesta \$25.29 por unidad. En el caso de los 16 MB (4096K x 32), el valor es de \$76.97.

La tendencia general es aumentar la capacidad de los módulos y discontinuar las versiones más pequeñas, gracias a la disminución de los precios en el mercado. En definitiva, se decidió utilizar un DIMM de SDRAM, con capacidad mínima de 32 MB (8192K x 32), que pueda trabajar a 33, 66 ó 100 MHz.

Tal decisión vino a reafirmar la elección del procesador, ya que la versión que se había propuesto para trabajar en conjunto con el puente no posee controlador de SDRAM. En el caso del chip 80960-RM sí se cuenta con esta función, además de tener el bridge incorporado en su sistema.

En cuanto al tipo de ROM, se decidió usar EPROM de 8 bits, pues es la más sencilla de conectar y manejar. Para reducir el espacio físico, se buscó una memoria en presentación QFP de 32 pines, que iría montada sobre una base especial.

Además fue necesario incluir circuitos especiales denominados Clock-buffer. Estos chips se encargan de proporcionar cuatro señales de reloj idénticas a la entrada, gracias a un arreglo interno de PLLs, con un tiempo de desfasamiento muy pequeño (1 ns) y capacitadas para distribuir la señal de sincronización a las cargas que lo requieran.

El primero de estos chips se utiliza para distribuir el reloj del primario, entregado por la plataforma PCI (33 MHz). Las señales de salida se aplican al microprocesador y al controlador de protocolo HDLC, para que los ciclos de bus estén bien sincronizados. Luego, es necesario usar otro en la interfaz con la SDRAM, pues esta lleva internamente varios bancos separados y requieren de relojes individuales.

Por otro lado, fue necesario definir las etapas de la tarjeta que trabajan con 5V, cuales lo hacen con 3.3V y cuales deben ser diseñadas para alimentación dual. Esto se hace de esta forma para que el módulo sea lo más flexible posible, y que pueda trabajar en cualquier sistema de cómputo que tenga ranura PCI, sin importar el voltaje de alimentación sobre el cual está basada la plataforma.

Por las características particulares de los chips 80960-RM y Freedm-8, su alimentación principal es de 3.3V. Sin embargo el ambiente de sus buses PCI es controlado por una patilla especial de referencia. En el caso del controlador de protocolo HDLC es necesario utilizar un comparador para aplicar el valor lógico en la entrada de control de referencia.

Las señales de los dispositivos periféricos en los chips se alimentan con la tensión +VIO suministrada por la ranura de expansión.

Por otra parte, se debe movilizar una línea de alimentación hacia la tarjeta Data-Link para suministrar 5V a los dispositivos presentes. Esta línea sale a través del conector DB25 y debe estar desacoplada correctamente y filtrada para que no introduzca ruido en el ambiente del PC.

El resultado de este proceso de diseño es un esquemático en el cual se han realizado todas las conexiones entre dispositivos. Luego de un proceso de análisis y evaluación que se hizo en conjunto con personas identificadas con los requerimientos eléctricos del bus PCI se llegó a una versión definitiva que fue enviada a los Estados Unidos para desarrollar el circuito impreso.

Basándose en el estándar PCI y los manuales de diseño para los chips, se conectó todas las señales auxiliares necesarias y se determinó cuales no serían utilizadas del todo en la aplicación.

Para la versión piloto, se colocó una serie de Jumpers con el fin de poder realizar modificaciones mínimas en el hardware. De esta forma se espera tener mayor flexibilidad al momento de trabajar con los prototipos. Además, se agregó puntos de prueba (Test Points) para poder observar el comportamiento de algunas señales importantes.

En cuanto al diseño del software, se inició con el estudio del ambiente que implica trabajar con el 80960-RM. Esto implica registros internos, tipo de datos, ancho del bus, uso de caché, mapas de direcciones y configuración de los recursos y periféricos.

La iniciación del circuito puente está muy relacionada con el software Plug and Play de la computadora donde se instalará el módulo. En el proceso de arranque, por medio de ciclos de configuración, el PC asigna los recursos requeridos por el puente. Sin embargo existe lo que se denomina Modo 3 para iniciar el procesador, que permite tener un mayor control sobre lo que se pedirá al computador durante la configuración del puente y la asignación de recursos para la tarjeta.

Usando el Modo 3 el usuario se asegura de programar todas las unidades del 80960-RM antes de permitir accesos en el primario.

Por último se hace notar la capacidad que tiene el diseño para adaptarse a diferentes elementos de hardware que se desee usar:

- Gracias al uso de memoria SDRAM en formato DIMM de 168 pines, es posible usar módulos de diferentes tamaños, que se identifican en el proceso de configuración del MCU por medio de un bus I<sup>2</sup>C
- Gracias a un grupo de jumpers y el montaje de la ROM en una base compatible pin a pin con memoria Flash, se facilita el proceso de emigrar la tecnología de almacenaje del programa.

## 6.2 Alcance y limitaciones

A partir de la información recopilada y el estudio detallado del problema y su solución, se tiene un circuito completo que satisface con las necesidades de Cibertec. El esquemático listo, con todos los detalles, se muestra en el Apéndice.

Este circuito ya ha pasado por una etapa de revisión y la programación está en etapa de encabezados. Se ha logrado un diseño capaz de procesar los mensajes provenientes de las unidades CPU y eliminar los elementos del protocolo HDLC. Además, los buses PCI están conectados y el procesador tiene su sistema de memoria listo. La interfaz eléctrica con el bus PCI de la computadora ya está terminada y el sistema debe permitir una identificación correcta con el software de configuración.

Con respecto a la situación práctica del proyecto, ya se contrató el diseño y fabricación del impreso a la compañía PCB Dataline, y al momento de presentar este informe al I.T.C.R. en Cibertec se está esperando el resultado final de esta sub-contratación, que corresponde a 10 tabletas prototipo con las máscaras blancas, los estenciles y los archivos binario requeridos para el ensamble.

Además, la empresa ya adquirió el módulo de desarrollo y está pendiente por su llegada al país en las próximas semanas.

El siguiente paso es realizar las rutinas en lenguajes de más alto nivel para establecer un correcto diálogo con la computadora (drivers y librerías dinámicas), además de los procedimientos que se requieren en el firmware de la tarjeta para el manejo y procesamiento de los datos localmente.

El diseño está pensado para alimentación dual, y utiliza recursos presentes en el mercado. Se puede adaptar fácilmente a las tecnologías que puedan surgir en los próximos años por lo que promete ser un diseño duradero.

Las limitaciones que se encuentran son básicamente de carácter administrativo, pues el avance del proyecto a partir de ahora depende mucho de la flexibilidad que Cibertec presente en la contratación del ensamblaje y la afiliación a PCI SIG, si se decide cumplir con este particular.

En el desarrollo teórico de la programación se puede avanzar sin problema, pues ya se cuenta con las hojas de datos y manuales de desarrollador necesarios para determinar los registros principales con sus valores de iniciación. Luego será cuestión de diseñar los procedimientos y las subrutinas de interrupción, basándose en el software que viene con el kit de desarrollo.

En cuanto a limitaciones electrónicas que pueda presentar la tarjeta terminada, se presentan básicamente dos:

a. El ancho del bus PCI usado es de 32 bits. Posiblemente en los años venideros surja la versión de 64 bits de PCI, lo que puede provocar cierto nivel de obsolescencia en la tarjeta, respecto a los equipos de cómputo que se pueda tener en unos 5 a 10 años.

b. Se ha establecido un límite de 32 data-links para la tarjeta, que posiblemente sean suficientes para abarcar una buena parte de la capacidad del procesador. En caso de que se desee agregar mayor volumen de información, se presentará el inconveniente de la saturación y se deberá tener mucho cuidado con la posibilidad de pérdida de datos por exceso de flujo en la entrada.

En resumen, se ha concluido satisfactoriamente el desarrollo teórico del módulo y se están llevando a cabo ciertos procesos para arrancar el proceso de fabricación. Sin embargo aún no se tiene los prototipos ensamblados, y la programación se encuentra aún en etapas iniciales. Para este punto, los objetivos planteados al nivel de Proyecto de Graduación han sido alcanzados. El rumbo que tome el desarrollo del proyecto a partir de ahora es responsabilidad de Cibertec Int.

# **CAPÍTULO 7**

## **CONCLUSIONES Y RECOMENDACIONES**

## **7.1 Conclusiones**

7.1.1 El equipo ITSS en el cual trabajará el módulo es un sistema para la detección de fraude telefónico en llamadas internacionales.

7.1.2 La tarjeta HDLC-256 está capacitada para recibir 8 datalinks del tipo E1 a través de un conjunto de puertos ST-Bus, procesarlos y comunicarlos a la computadora por medio de un bus ISA de 4,33 MHz.

7.1.3 La capacidad de procesamiento de la tarjeta HDLC-256 le permite recibir la información proveniente de cuatro CPUs del equipo ITSS.

7.1.4 Los componentes principales de la tarjeta HDLC-4M son un procesador 80960-RM de Intel y un controlador de protocolo HDLC de PMC-Sierra denominado PM7366.

7.1.5 El sistema HDLC-4M posee un bus PCI primario, un bus PCI secundario, un bus para memoria SDRAM de 64 bits, un bus para ROM de 8 bits y un conjunto de 8 puertos ST-Bus.

7.1.6 El bus PCI es un estándar de alto nivel, que no sólo comprende aspectos de arquitectura en el diseño, sino que enlaza el hardware con una nueva filosofía de software denominada "Plug and Play".

7.1.7 El bus PCI básico posee un ancho de 32 bits y trabaja a 33 MHz, sin embargo tiene las previstas para aumentar hasta 64 bits y 66 MHz.

7.1.8 PCI es compatible con tecnologías TTL de 5V, sin embargo utiliza y promueve la migración hacia el uso de dispositivos de 3.3V.

7.1.9 En PCI existen tres espacios de direccionamiento básicos: memoria, I/O y configuración, de los cuales únicamente el de configuración es requisito.

7.1.10 El espacio de configuración es el que permite al software de PnP identificar y asignar recursos a un dispositivo PCI.

7.1.11 Además de la unidad principal de procesamiento, el 80960-RM posee un grupo de unidades internas que se utilizan para funciones específicas, de las cuales las principales son el puente PCI-PCI, la unidad de traslación de direcciones y el controlador de memoria.

7.1.12 Los principales componentes del puente PCI-PCI son las Interfaces de Bus, las Colas Ascendentes y Descendentes y el Espacio de Configuración.

7.1.13 La organización del espacio del bus se controla usando registros base y límite para definir espacios de I/O, de memoria o de memoria prefetchable, según sea la necesidad de la aplicación.

7.1.14 Es posible abrir ventanas de los buses PCI para ser accesibles desde el bus interno, por medio de la configuración adecuada de la ATU. Esto permite establecer un flujo de datos directamente entre la memoria local y los buses PCI.

7.1.15 Los diseñadores y productores de tarjetas PCI deben afiliarse a una organización mundial denominada PCI-SIG, que por \$3000 anuales asigna un número de identificador al módulo y a otro al fabricante, para ser usados en el espacio de configuración.

7.1.16 La configuración del procesador se realiza por medio de accesos a los denominados MMRs, que son registros usados por las unidades del chip para determinar sus características de trabajo.

7.1.17 Para diseñar un módulo PCI es necesario considerar aspectos eléctricos relacionados con las características no ideales de las pistas en el impreso, que no pueden ser ignoradas dada la alta velocidad de operación.

## **7.2 Recomendaciones**

7.2.1 El siguiente paso en cuanto la obtención de los prototipos terminados corresponde, es utilizar los archivos Gerber que se reciban de Dataline-PCB para solicitar una cotización en la empresa AETEC. Además, es importante considerar la opción de ensamble recomendada por Ronald Siewko (contacto en Dataline).

7.2.2 Para el montaje superficial que se va a realizar es importante buscar una empresa que ofrezca múltiples opciones de prueba, para asegurarse que cada componente queda en su lugar. Especial cuidado se requiere para los chips BGA.

7.2.3 Para continuar con el avance de la programación se hace indispensable contar con el módulo de desarrollo. Una vez que se tiene esto, se debe iniciar realizando programas sencillos donde se establezca comunicación con la computadora y a partir de aquí se empieza a aumentar el nivel de complejidad.

7.2.4 Paralelamente al uso del módulo de desarrollo, es necesario realizar pruebas similares con el prototipo, para corroborar su funcionalidad.

7.2.5 A partir de la obtención de los prototipos, es necesario incluir personal del área de computación, para que participen en el desarrollo de los controladores, las librerías y los nuevos programas de aplicación.

7.2.6 Para las pruebas con información proveniente de ST-Bus, se recomienda iniciar con pocos canales a la vez, enviando mensajes muy básicos. Una vez que se obtenga éxito en estas pruebas entonces se aumenta el grado de complejidad.

7.2.7 Una vez que se haya aprobado el hardware, se debe hacer un pedido de tarjetas mayor, en el cual se eliminen los jumpers y los puntos de prueba.

7.2.8 El hardware desarrollado posee características muy flexibles. Se recomienda a la empresa analizar el posible uso de este módulo en otras aplicaciones.

**CAPÍTULO 8**  
**REFERENCIAS CONSULTADAS**

## 8.1 Bibliografía

- Solari, Edward & Willse, George. PCI Hardware and Software. 4th Ed. Annabooks: San Diego, CA. USA. 1998.
- PCI Special Interest Group. PCI Local Bus Specification. Rev. 2.1. PCI-SIG: Portland, OR. USA. 1995.

## 8.2 Hojas de datos

- Texas Instrument. TMS320C2x User's Guide. Rev. C. TI Inc.: USA. 1993.
- Cypress. CY2305-CY2309, Low-Cost 3.3V Sero Delay Buffer. Cypress Semiconductor Corporation: San Jose, CA. USA. 1999.
- Cypress. CY7C419/21/25/29/33, 256/512/1K/2K/4K x9 Asynchronous FIFO. Cypress Semiconductor Corporation: San Jose, CA. USA. 1997.
- Micron. MT4LSDT464A, MT4LSDT864A & MT4LSDT1664A. Synchronous DRAM Modules. Micron Technology Inc. 2000.
- ST. M27C2001, 2 Mbits UV ROM. STMicroelectronics Group. 2000.

## 8.3 Documentos en formato PDF

- Intel ® 80960RM I/O Processor. Order Number: 273156-005. May, 2000.
- i960 ® RM/RN I/O Processor, Developer's Manual. Order Number: 273158-001. July 1998.
- MSAN-126. ST-BUS Generic Device Specification (Rev. B), Application Note. June 1995.

#### **8.4 Direcciones de Internet**

- <http://developer.intel.com/design/iio/>
- <http://www.pcm-sierra.com>
- <http://www.mitel.com>
- <http://www.cypress.com>
- <http://users.supernet.com/sokos/pci.txt>
- <http://www.pcguide.com/ref/mbsys/buses/>
- <http://www.pcisig.com>
- <http://www.st.com>
- <http://www.mouser.com>
- <http://www.arrow.com>
- <http://www.digikey.com>
- <http://www.detalinepcb.com>
- <http://www.aetec.com>

#### **8.5 CD-ROM**

- PMC-Sierra Inc. “Internetworking Silicon Solutions”, Product Information Catalog. Volume 6 – Issue 2. Canada, 2000.

# **CAPÍTULO 9**

## **APÉNDICES**

## 9.1 Características mecánicas de los componentes

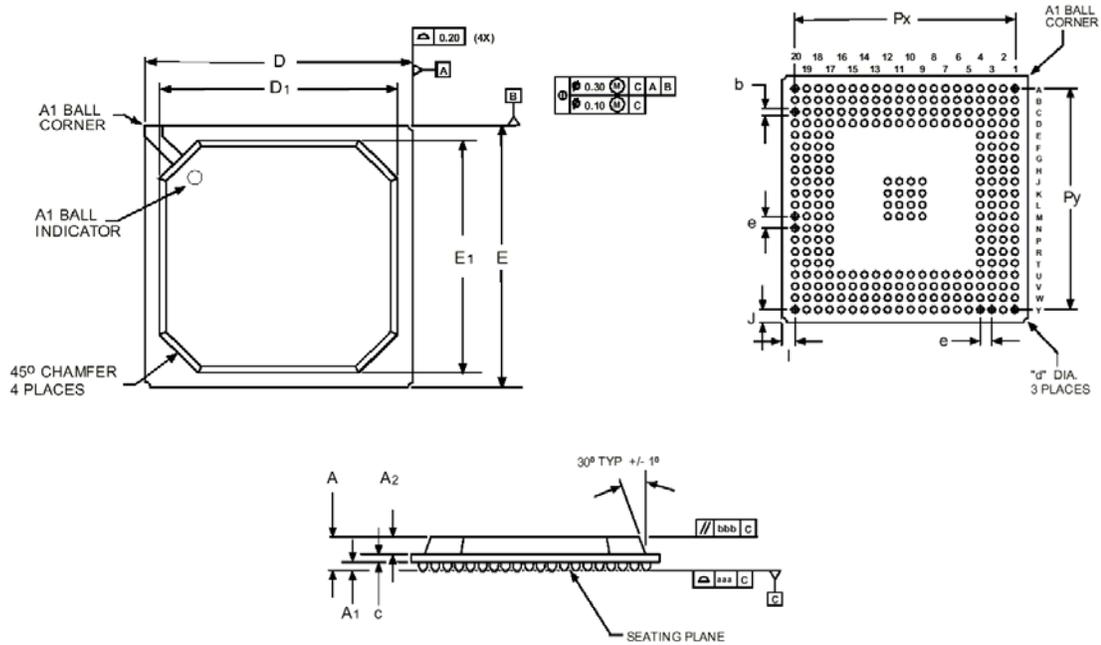


Figura Ap1 Presentación mecánica del controlador PM7366.<sup>3</sup>

Tabla Ap1 Dimensiones del controlador PM7366.

PACKAGE TYPE : 272 PLASTIC BALL GRID ARRAY - PBGA																			
BODY SIZE : 27 x 27 x 2.33 MM (4 layer)																			
D	A	A	A1	A2	D	D1	E	E1	I	J	b	c	c	d	e	Px	Py	aaa	bbb
in.	(2 layer)	(4 layer)										(2 layer)	(4 layer)						
Min.	1.92	2.12	0.50	1.12	26.80	23.50	26.80	23.50	-	-	0.60	-	-	-	-	24.03	24.03	-	-
Nom.	2.13	2.33	0.60	1.17	27.00	24.00	27.00	24.00	1.435	1.435	0.76	0.36	0.56	1.0	1.27	24.13	24.13	-	-
Max.	2.32	2.54	0.70	1.22	27.20	24.70	27.20	24.70	-	-	0.90	-	-	-	-	24.23	24.23	0.15	0.35

<sup>3</sup> NOTES:  
 All dimensions in millimeter.  
 Dimension aaa denotes coplanarity.  
 Dimension bbb denotes parallel.

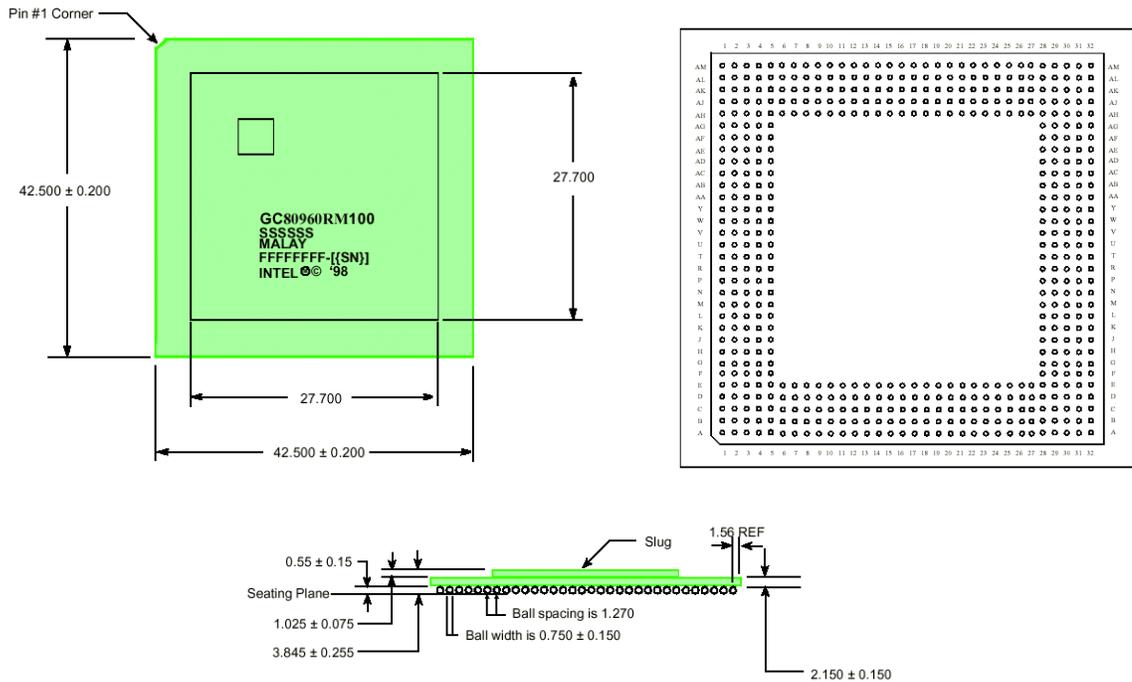


Figura Ap2 Presentación mecánica del procesador 80960-RM.<sup>4</sup>

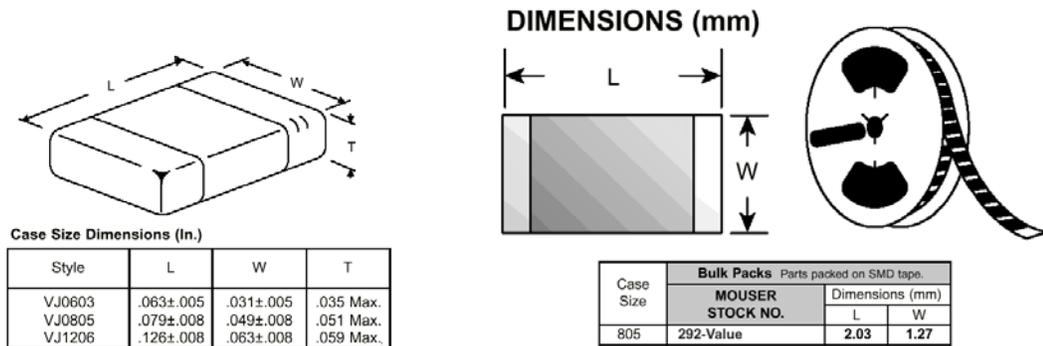


Figura Ap3 Presentación mecánica de los componentes discretos.

<sup>4</sup> NOTES:

All dimensions and tolerances conform to ANSI Y14.5M 1982.  
 Dimensions are measured at the maximum solder ball diameter parallel to primary datum.  
 Primary datum and seating plane are defined by the spherical crowns of the solder balls.  
 All dimensions are in millimeters.  
 S spec numbers are only printed on the C-X steppings.

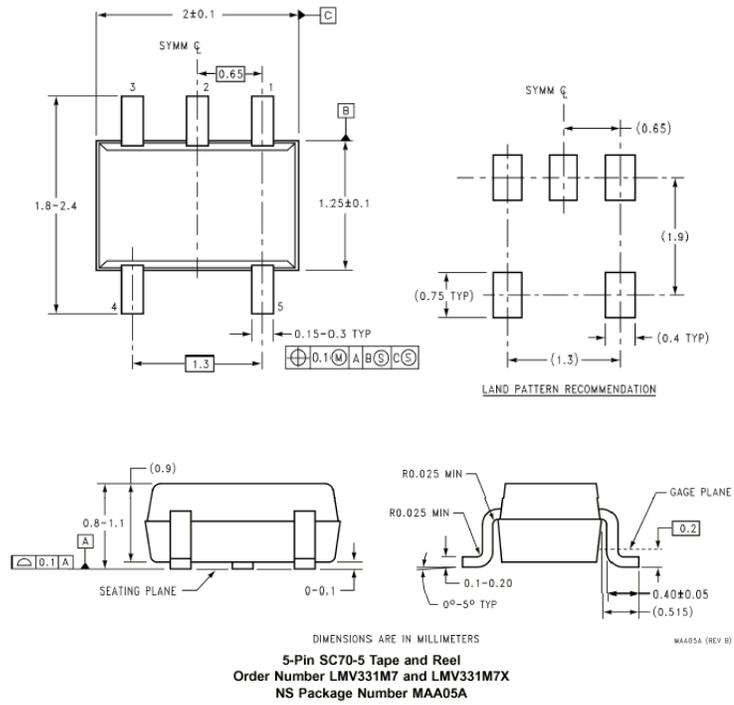


Figura Ap4 Presentación mecánica del comparador LMV331.

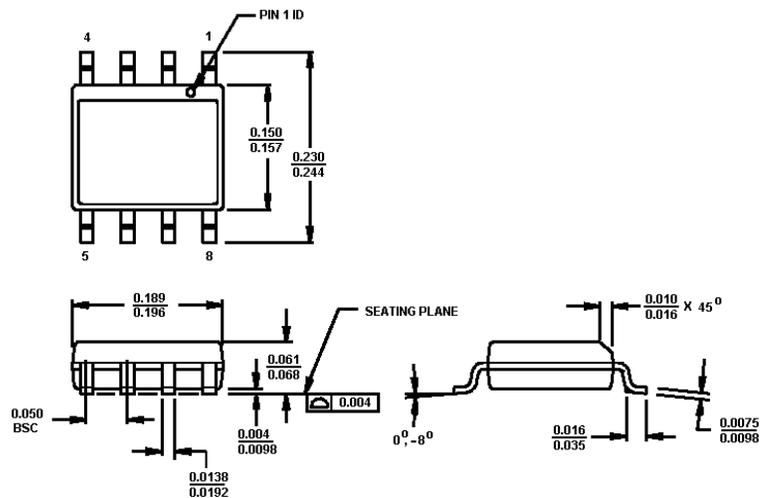
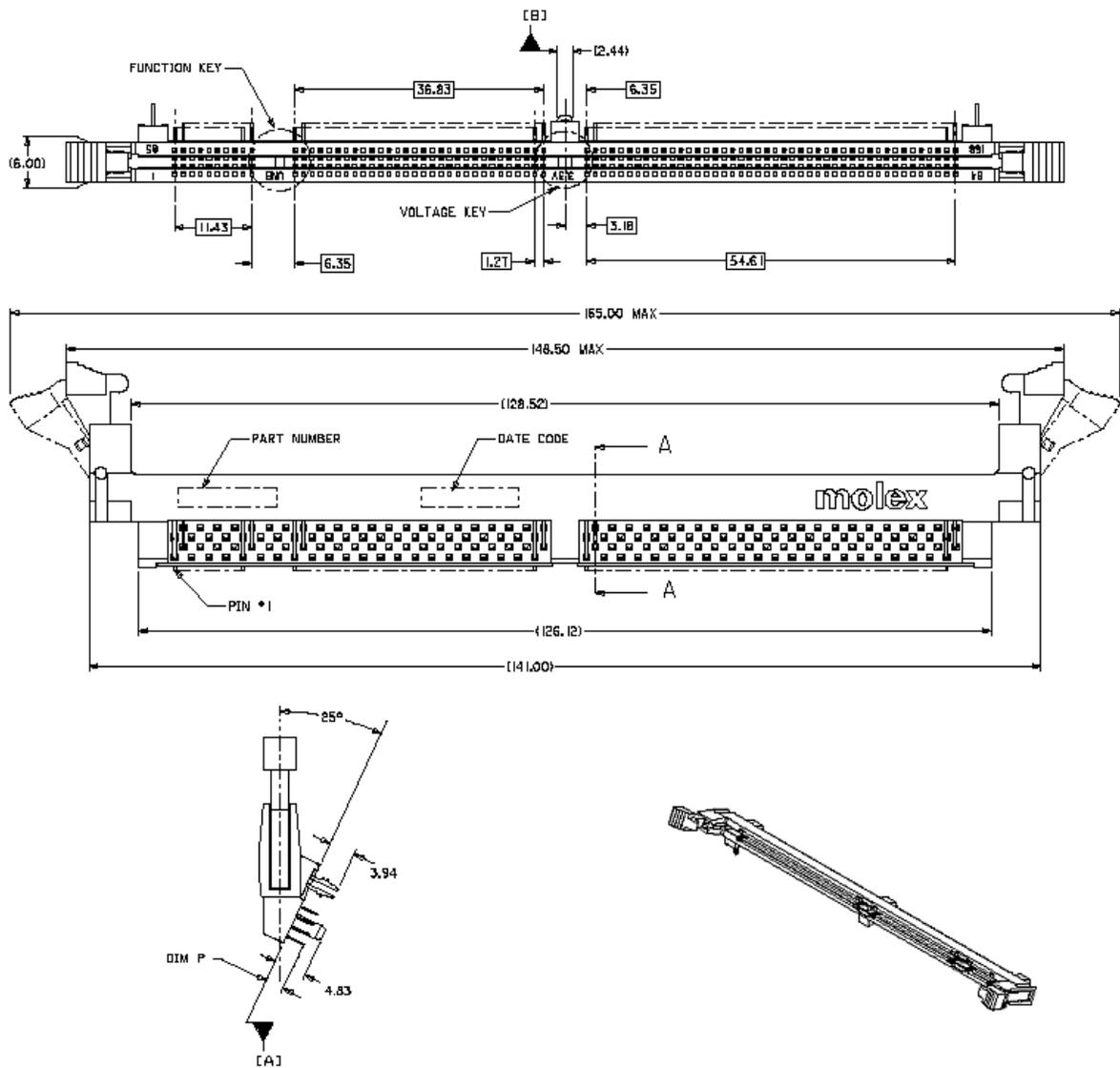


Figura Ap5 Presentación mecánica del clock-buffer CY2505.<sup>5</sup>

<sup>5</sup> NOTES:

Dimensions in inch (MIN/MAX)

PIN 1 ID is optional, round on single leadframe, rectangular on matrix leadframe.



**Figura Ap6** Presentación mecánica del socket para la SDRAM.<sup>6</sup>

<sup>6</sup> FEATURES AND BENEFITS:

Accepts Standards JEDEC MO-161 modules.

Dual ejector latches align modules for insertion and provide an audible click when in the closed position.

Forklocks provide PCB retention during and after soldering.

Base-plate ensures PC tail true position.

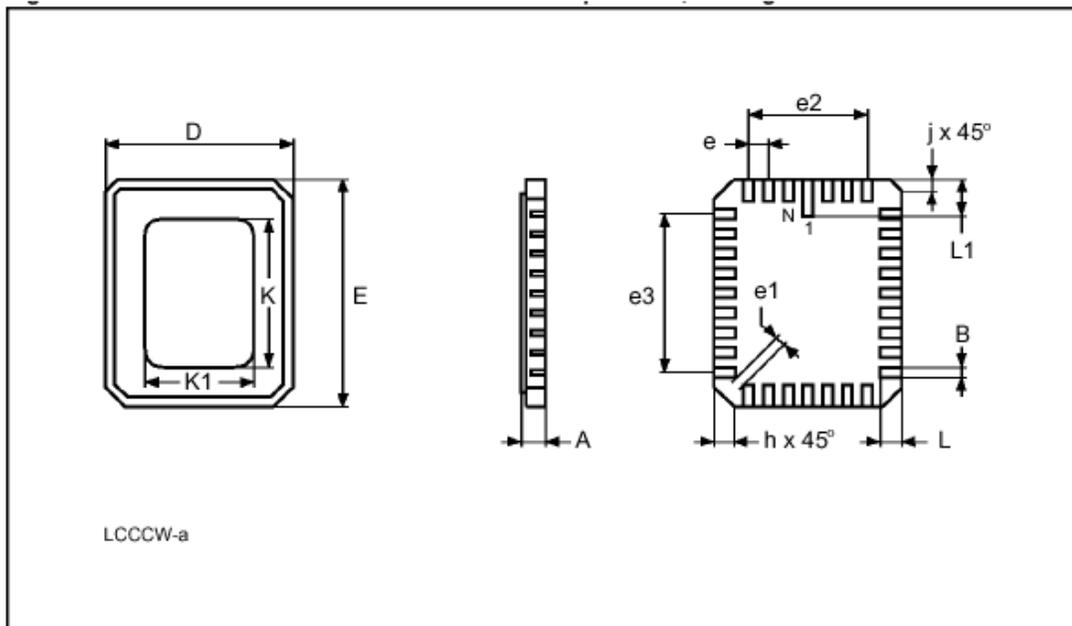
Tight stacking distances: 22.86 mm (.900") for typical SOJ packaging, 13.34 mm (.525") for TSOP packaging.

Several options for keying and solder-tail length.

Circuits: 168.

Entry: 25 degrees Angle.





Drawing is not to scale.

**Figura Ap9** Presentación mecánica de la EPROM M27C2001-55XL1X.

**Tabla Ap2** Dimensiones de la EPROM M27C2001-55XL1X.

LCCC32W - 32 lead Leadless Ceramic Chip Carrier, Package Mechanical Data

Symbol	mm			inches		
	Typ	Min	Max	Typ	Min	Max
A			2.28			0.090
B		0.51	0.71		0.020	0.028
D		11.23	11.63		0.442	0.458
E		13.72	14.22		0.540	0.560
e	1.27	-	-	0.050	-	-
e1		0.39	-		0.015	-
e2	7.62	-	-	0.300	-	-
e3	10.16	-	-	0.400	-	-
h	1.02	-	-	0.040	-	-
j	0.51	-	-	0.020	-	-
L		1.14	1.40		0.045	0.055
L1		1.96	2.36		0.077	0.093
K		10.50	10.80		0.413	0.425
K1		8.03	8.23		0.316	0.324
N		32			32	



## 9.2 Último avance recibido con relación al diseño del impreso

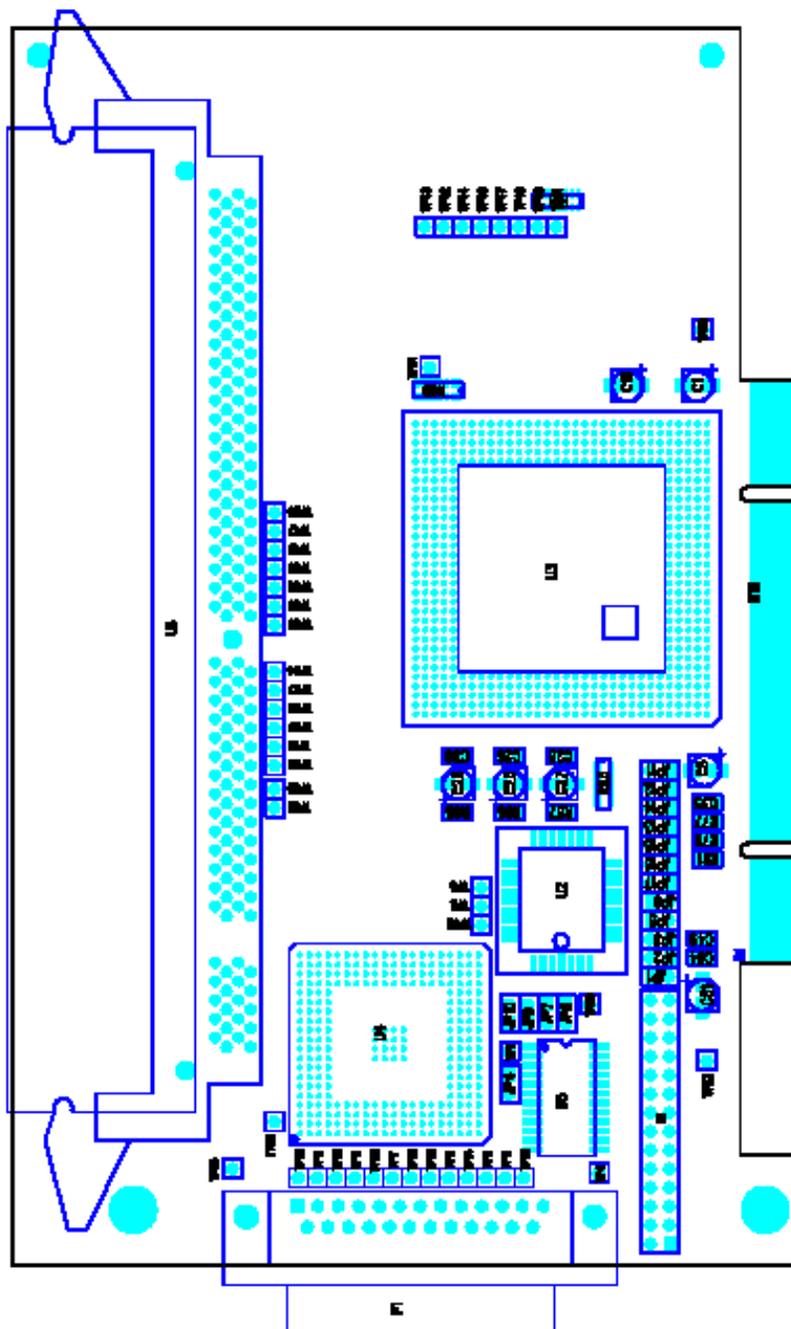
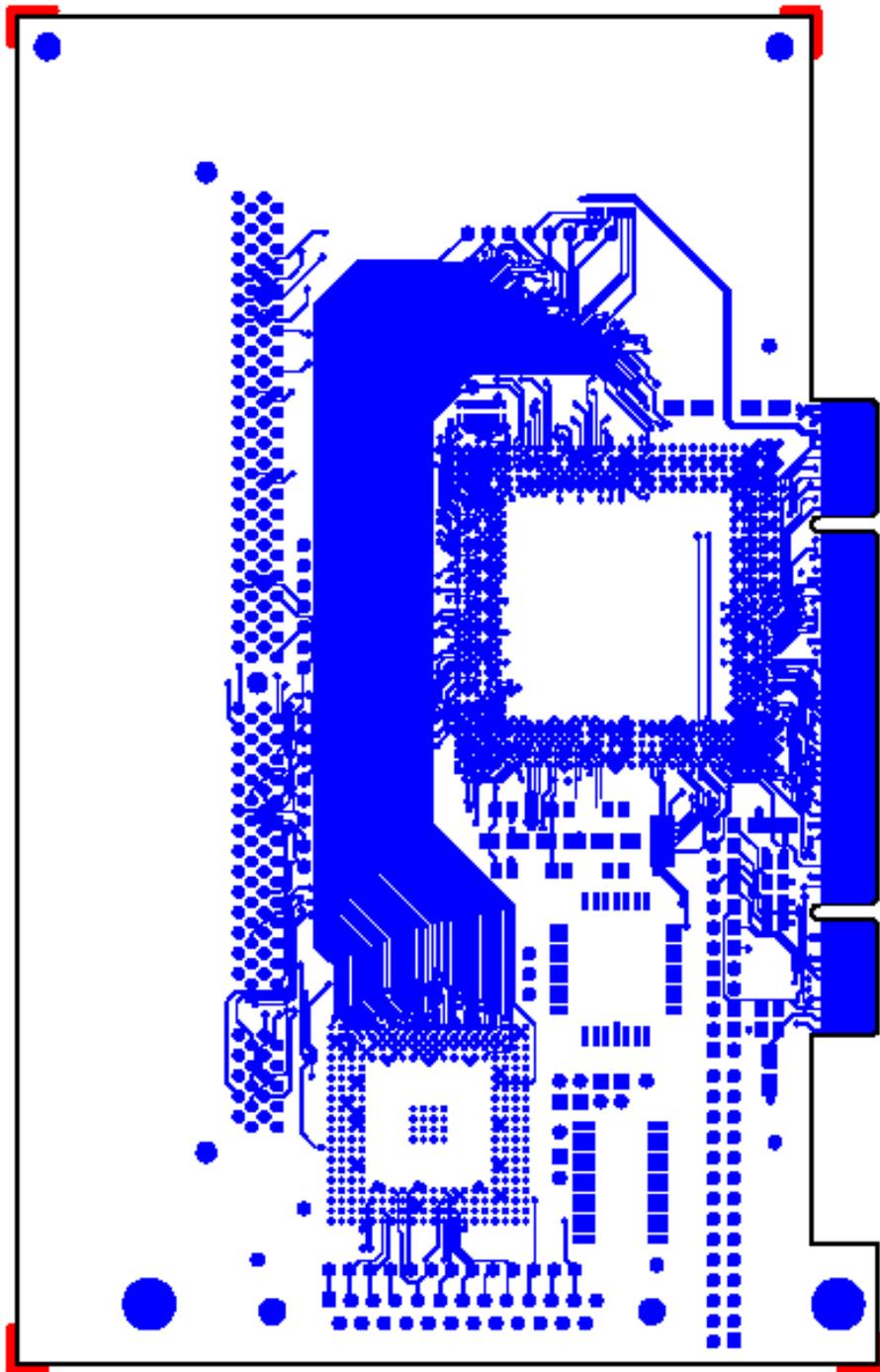


Figura Ap11 Avance enviado por Dataline PCB: HDLC-4M, cara superior.



**Figura Ap12** Avance enviado por Dataline PCB: HDLC-4M, layout superior.

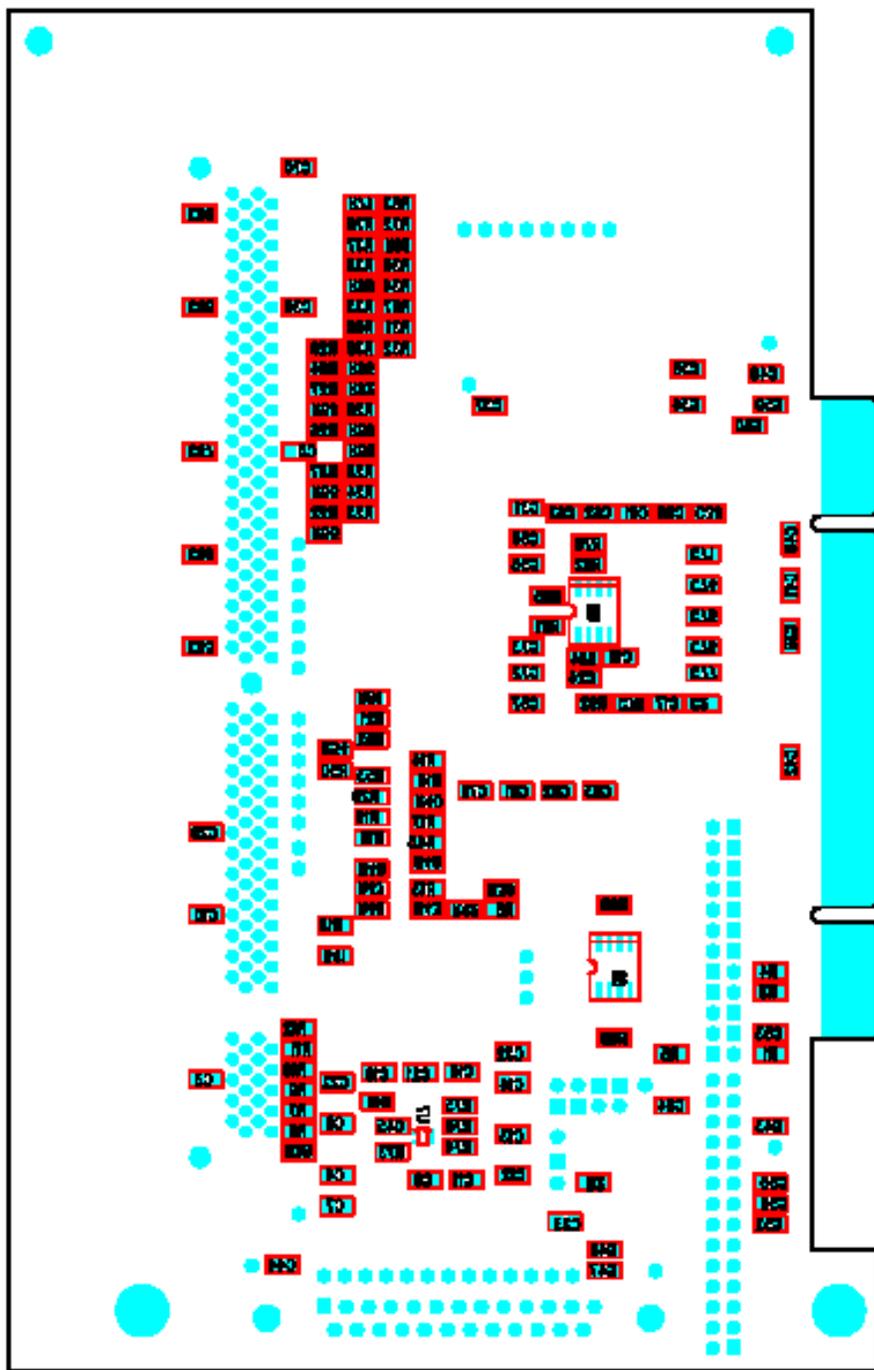
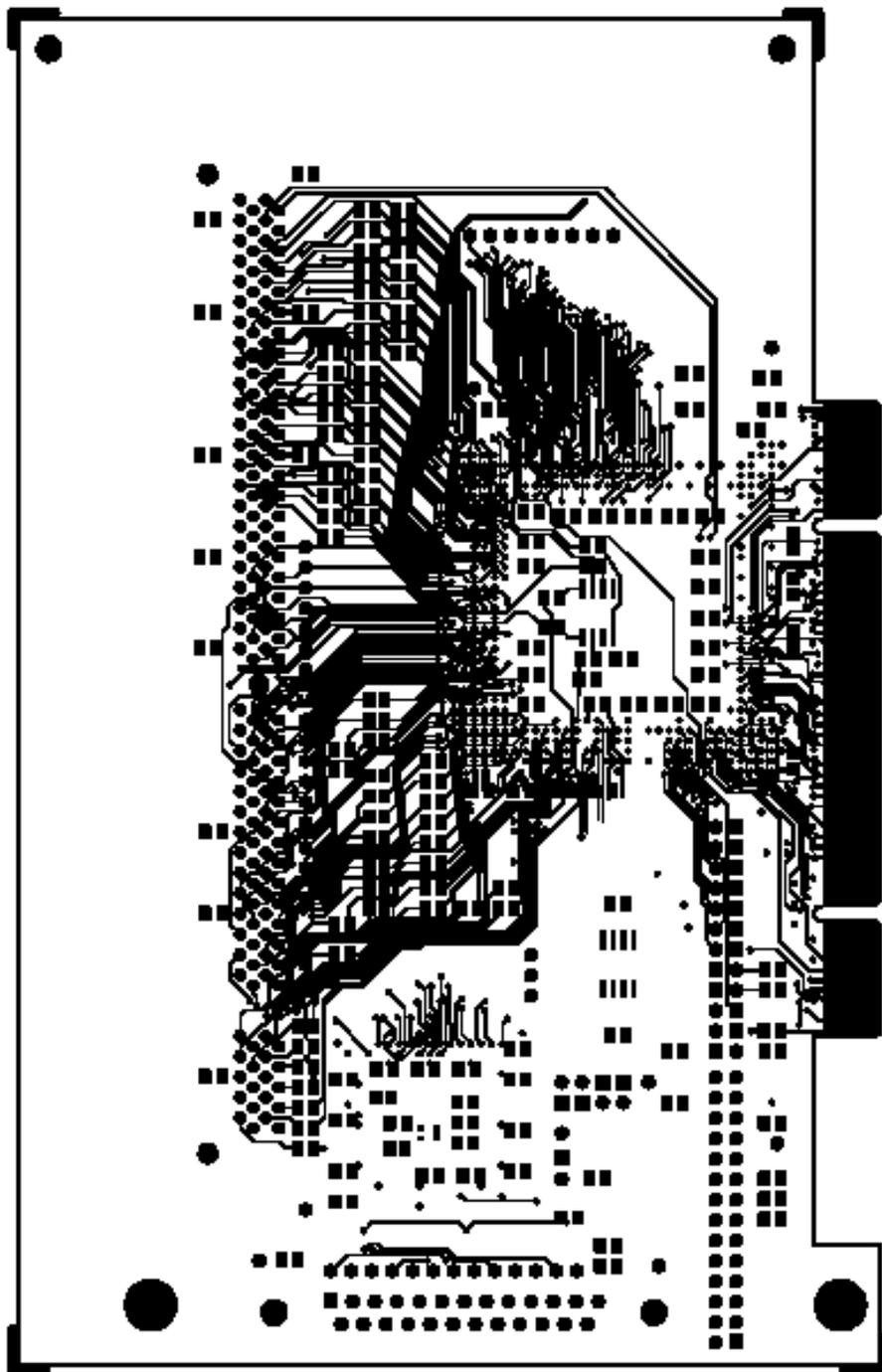


Figura Ap14 Avance enviado por Dataline PCB: HDLC-4M, cara inferior.



**Figura Ap15** Avance enviado por Dataline PCB: HDLC-4M, layout inferior.