Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica

Cibertec Internacional

"Rediseño de la etapa de recepción de los buses ST-Bus en la tarjeta HDLC- 256 con salida a bus PCI"

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

Meizel Leiva Rojas

Cartago, 2001

Resumen

La búsqueda de un formato de comunicación eficiente y fácilmente aplicable a los sistemas ya existentes, así como un protocolo de comunicación flexible para transportar información de audio o del tipo digital, ha hecho del formato ST-Bus y el protocolo HDLC herramientas importantes en el área de las telecomunicaciones. La utilización de buses cada vez más rápidos y de mayor eficiencia en la transferencia de datos ha llevado al estándar PCI a ocupar el primer lugar en los buses de expansión en las computadoras actuales desplazando casi por completo a su antecesor el bus ISA.

Estos puntos son primordiales en el desarrollo de la tarjeta HDLC-4M que Cibertec Int. ha querido crear, donde se ha buscado aumentar la cantidad de líneas ST-Bus que se puedan procesar en comparación con su antecesora, la tarjeta HDLC-256. Esto se logró utilizando el PM7366, para el que se realizaron las rutinas de inicialización así como el mapa de memoria necesario para su efectivo funcionamiento. La utilización de buses PCI, tanto internamente en la tarjeta (utilización de buses secundarios) como en el bus que comunica a la tarjeta con la computadora huésped, logra el objetivo de utilizar tecnología de punta en cuanto a transferencia de datos se refiere, para lo cual se aprovecharon las características que presenta el microprocesador 80960RM de Intel. Así mismo, se buscó utilizar en esta tarjeta componentes con permanencia garantizada en el mercado para evitar problemas de obsolescencia a corto plazo. El uso de componentes de montaje superficial asegura el uso de tecnología de punta en la confección total de la tarjeta, así como la disminución de problemas en alta frecuencia.

Además se logró realizar el diagrama del circuito interfaz entre las líneas ST-Bus y el controlador de protocolo HDLC, así como el diagrama de conexión entre el mismo controlador y el microprocesador, y entre el 80960RM y las memorias, tanto de programa como de datos.

Palabras claves: 80960RM, PM7366, FREEDM-8, HDLC, ST-Bus, PCI, ISA

Abstract

The search for an efficient communication format easily applicable to the existing system, as well as flexible communication protocol that's able to transport audio or digital information has made of the ST-Bus format and the HDLC protocol an important tool on the telecommunication area. The use of faster and more efficient buses for data transfer, has taken the standard PCI to be the number one bus on the actual computer expansion, displacing almost completely its predecessor the ISA Bus.

There are fundamental issues on the HDLC-4M card development created by Cibertec International, wanting to increase the amount of ST-Bus lines processed compared to the previous one, the HDLC-256 card. This was accomplished by using the PM7366 from which the starting routines and the memory map headed for its effective operation were taken. The use pf PCI buses, both internally in the card (use of secondary buses) and on the bus that connects the card with the host computer achieves the goal of using of top technology in data transfer.

Besides high production materials and superficial assembly were used. This was to avoid the future obsolescence of the card because of the lack of components and to take advantage of the use of high quality technology, even on the components assembly. Also, it was made a diagram of the interface circuit between the ST-Bus lines and the HDLC protocol controller, as well as the connection diagram between the same controller and de microprocessor, and between the 80960RM and the program and data memories.

Key words: PM7366, FREEDM-8, HDLC, ST-Bus, PCI, ISA

DEDICATORIA

A mi madre, a quien debo la existencia y todo lo que hasta hoy soy. A Belforth, por su amor, comprensión y espera. A Sehiry, Marcos, Ivannia y Fraciny por su apoyo incondicional y por su confianza en la empresa en la que me he encaminado. Los amo.

AGRADECIMIENTO

"Porque Dios da la sabiduría, y de su boca viene el conocimiento y la inteligencia. Él provee de sana sabiduría a los rectos; es escudo a los que caminan rectamente. Es el que guarda las veredas del juicio, y preserva el camino de sus santos. Entonces entenderás justicia, juicio y equidad, y todo buen camino. Cuando la sabiduría entrare en tu corazón, y la ciencia fuere grata a tu alma, la discreción te guardará; te preservará la inteligencia..."

Proverbios 2, 6-11

Gracias a Dios y a nuestra madre María, que en el cielo han sido mi fuerza y mi inteligencia. Agradezco profundamente a los Ingenieros Arnoldo Rojas Coto y Néstor Hernández Hostaller por el apoyo que me brindaron durante todos mis años de estudio. En especial, agradezco a mi madre por creer en mi.

INDICE GENERAL

Índice de Figuras

Índice de tablas

Capítulo 1
Introducción

1.1 Descripción de la empresa

Cibertec Internacional es una empresa multinacional fundada en 1979. Sus actividades se encuentran enfocadas principalmente en el desarrollo de alta tecnología electrónica para la industria de las telecomunicaciones. Su principal producto es el TMSS (Telecommunication Traffic Management & Surveillance System) el cual se ha desarrollado para prevenir el fraude telefónico en el servicio de larga distancia internacional.

Esta empresa posee su planta de ensamblaje ubicada en la Zona Franca de Alajuela (SARET), donde se realizan los montajes de las piezas en las tarjetas, y además el diseño de circuitos impresos. Las oficinas de diseño y desarrollo de programación se encuentran ubicadas en San José, en el cantón de Montes de Oca. El departamento de ventas se encuentra localizado en la ciudad de Panamá.

Actualmente laboran en esta empresa 10 ingenieros eléctricos y electrónicos encargados de diseño y montaje de equipo, 4 ingenieros en informática dedicados al desarrollo del software necesario para las aplicaciones de la empresa. Cuenta además con dos contadores, todos estos ubicados en las oficinas de Sabanilla.

En la planta en Alajuela trabajan 15 operarias, 5 ingenieros en el desarrollo de impresos, y 9 funcionarios más en el departamento de finanzas.

La junta directiva de la empresa se encuentra conformada por el Ing. Juan Carlos Halphen, el Lic. Orlando Jiménez N., el Ing. Bernal Thalman C., y el Ing. Hernán Jiménez N.. El Gerente General de la empresa es el Ing. Luis Jiménez N.

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

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

El organigrama de la empresa se muestra en la figura 1.1.

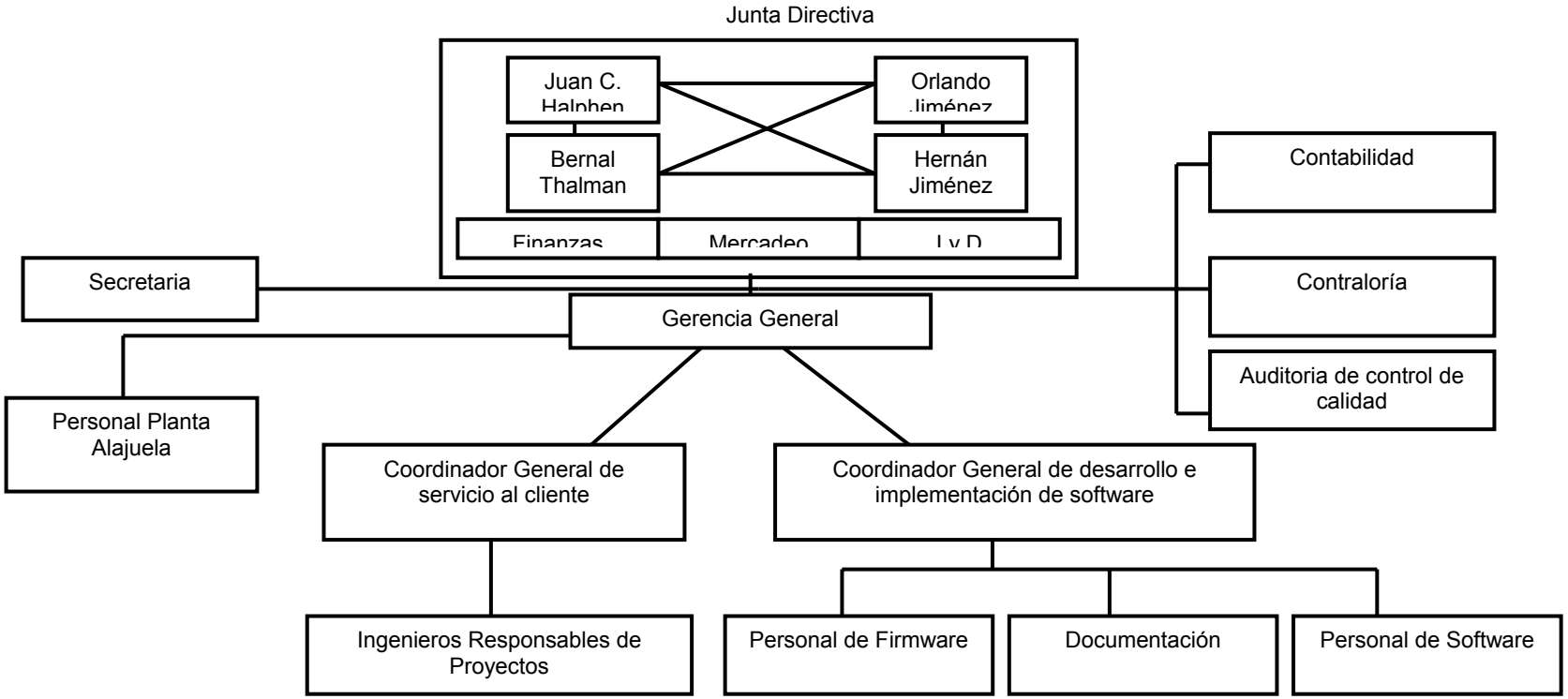


Figura 1.1 Organigrama de Cibertec Internacional

1.2 Definición del problema y su importancia

El sistema I.T.S.S., utilizado para la detección de fraude telefónico en el servicio de larga distancia, posee una etapa de interfaz con la computadora mediante una tarjeta llamada HDLC-256, la que procesa la información proveniente de una etapa E1, cuyos datos vienen en formato ST-BUS y con protocolo HDLC. Estos datos son filtrados de acuerdo a la señalización buscada, y mediante rutinas que se ejecutan en el procesador de la tarjeta. Así la información se encuentra lista para ser enviada a la computadora, utilizando un bus ISA.

La figura 1.2 muestra el diagrama de bloques de la tarjeta HDLC-256.

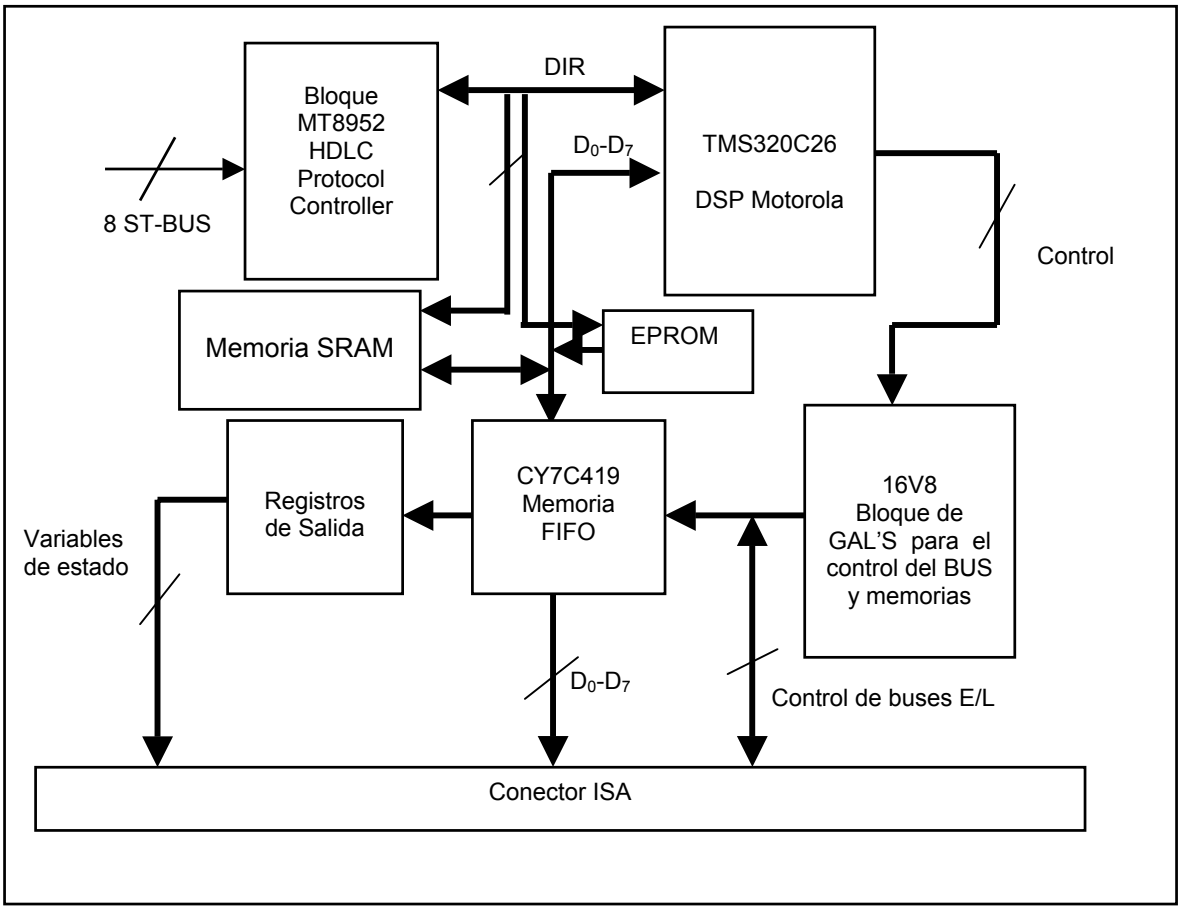


Figura 1.2 Diagrama de bloques de la tarjeta HDLC –256.

El problema que presentó la empresa al iniciar el proyecto consistió en la obsolescencia que presentan los componentes que conforman la tarjeta HDLC-256, así como la necesidad de aumentar la capacidad de procesamiento de la misma.

Cada tarjeta tiene la capacidad de procesar ocho canales de información del bus ST-bus en recepción y transmisión. En un punto de muestreo se colocan 4 tarjetas E1, de donde proviene la información en formato ST-Bus, de esta manera se deben procesar 32 canales simultáneamente de 16 enlaces ST-Bus. Dado este requerimiento es necesario conectar cuatro tarjetas HDLC-256, que ocuparían cuatro ranuras ISA de 16 bits en el computador, lo cual es difícil de encontrar en modelos actuales de computadoras.

Sin embargo la situación se complica aún más cuando el tráfico es denso, y es necesario colocar más de cuatro unidades E1, y por ende un número mayor de tarjetas HDLC-256 para el procesamiento de toda la información generada.

Los dispositivos cuya función sobre la tarjeta es la de tomar la información en formato ST-Bus y procesar el formato HDLC eran los MT8952, que utilizan técnica de multiplexación en el tiempo (TDM), lo cual permite al formato ST-Bus el procesamiento de 32 canales temporales de información. Este sistema presentó la limitante de poder transportar información únicamente en los canales del 0 al 3, sin poderse utilizar para el manejo de información los 28 canales restantes. Además, al poder procesarse tan sólo un canal por dispositivo al mismo tiempo, hace que se tengan que colocar 8 circuitos integrados de este tipo, haciendo que se ocupe un espacio significativo en la tarjeta, tan sólo en la etapa de recepción o transmisión de la señal.

Esto acarreaba también el problema de espacio físico en la conexión del sistema, ya que se necesitan 16 cables para transportar los 16 ST-Bus de información. Cuando se colocan los sistemas, sería más cómodo si se pudiera procesar más de 2 canales por enlace y con esto se reduciría los costos de instalación.

El bus de comunicación de la tarjeta HDLC-256 con la computadora es un bus ISA. Actualmente algunas computadoras ya no cuentan con este tipo de ranura, o bien, no poseen los suficientes, lo que conduce a pensar que pronto serán eliminados. Por esto que nace la necesidad imperiosa de poder comunicarse con un bus PCI, en vez de utilizar un bus ISA.

En cuanto al tipo de montaje, hasta el momento Cibertec ha usado el tipo “through hole”, sin embargo se quiere migrar hacia el montaje superficial. La nueva versión de la tarjeta HDLC será el comienzo de este cambio. El problema para el proyecto de graduación se enfoca en buscar un componente capaz de aprovechar más canales del ST-Bus, y de esta manera procesar más canales simultáneamente. Además el dispositivo debe de poder comunicarse utilizando un bus PCI, y debe de poder recibir información y transmitir información.

La solución de este problema es de vital importancia para Cibertec Internacional, ya que la tarjeta HDLC es parte de su producto principal, por lo que la obsolescencia del mismo, atenta con el futuro de la empresa. Además el hecho de la utilización de montaje superficial en esta tarjeta, y el uso de bus PCI para la comunicación con la computadora representa el inicio de la migración buscada por la empresa a un sistema de soldadura y de comunicación que no se encuentran a las puertas la obsolescencia.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar en forma teórica el circuito de la etapa de recepción de datos en formato ST-Bus de la tarjeta HDLC-4M utilizando el dispositivo FREEDM-8, y establecer los parámetros de programación de dicho dispositivo, así como la conexión para la efectiva comunicación del chip con el microprocesador local.

1.3.2 Objetivos específicos

1.2.1.1 Conocer el formato de transmisión de datos ST-Bus.

1.2.1.2 Identificar los componentes que conforman la tarjeta HDLC-256.

1.2.1.3 Buscar información acerca de los componentes que conforman la tarjeta HDLC-256.

1.2.1.4 Conocer el funcionamiento de los componentes que se encuentran en la tarjeta HDLC.

1.2.1.5 Conocer el funcionamiento del Bus ISA.

1.2.1.6 Conocer el funcionamiento de la tarjeta HDLC-256.

1.2.1.7 Conocer el funcionamiento bus PCI revisión 2.2.

1.2.1.8 Analizar el funcionamiento del circuito integrado FREEDM-8.

1.2.1.9 Determinar la lógica necesaria para interfazar los data links provenientes de la tarjeta E1, al circuito controlador de protocolo HDLC.

1.2.1.10 Analizar el funcionamiento del microprocesador i960-RM.

1.2.1.11 Seleccionar los módulos de memoria a utilizar en el sistema.

1.2.1.12 Determinar la estructura de conexión de la memoria con el circuito controlador de protocolo HDLC.

1.2.1.13 Determinar la estructura de conexión del microprocesador local con el dispositivo controlador del protocolo HDLC.

1.2.1.14 Realizar la lógica necesaria para interfazar los relojes de cada ST-Bus con el FREEDM.

1.2.1.15 Realizar el mapa de memoria necesario para el efectivo funcionamiento del FREEDM-8.

1.2.1.16 Realizar las rutinas de inicialización de los componentes en su respectivo compilador.

1.2.1.17 Realizar el circuito esquemático de conexión entre el FREEDM-8, el microprocesador i960-RM y el módulo de memoria seleccionado en ORCAD 9.1.

Capítulo 2
Antecedentes

2.1 Estudio del problema a resolver

Como el problema que se presentó en la empresa fue la obsolescencia de la tarjeta HDLC-256, y ésta es tan sólo un módulo dentro del sistema principal de venta de Cibertec Internacional, I.T.S.S, es importante explicar a que se refiere el I.T.S.S. y el papel que juega la tarjeta HDLC-256 en este sistema, para así, determinar las características que deberá cumplir la nueva versión de ésta y asegurar la compatibilidad con el sistema anterior.

2.1.1 Análisis formal del sistema I.T.S.S.

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

Este sistema 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 a tiempo real o históricamente estadísticas de tráfico, asistiendo a las conciliaciones con los conectantes para mejorar la precisión y pérdidas de tiempo.

Existen varias formas detectadas en las que se puede realizar fraude telefónico, como lo son:

1. El Call Back o llamada revertida.
2. ISR (Reventa de minutos)- Desvío no autorizado del tráfico internacional
3. Bypass (Tercer país)

El Call Back

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

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

Hay tres tipos principales de " Call Back". 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 call back 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).

Cibertec Internacional logra combatir este tipo de fraude por medio de la detección de DTFM (Dual Tone Multiple Frecuency), ya que la única razón para la que exista DTMF en una llamada entrante es el Call Back, ya que las centrales de los bancos utilizan DTMF, un banco nunca llama a un cliente para ofrecerle que consulte su cuenta bancaria.

ISR (Reventa de minutos) - 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 no utiliza las centrales internacionales supervisadas.

Son 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, 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.

"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 país con destino final otro 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.

La solución para este problema es: el mecanismo de llamada manual o automático, le pide al usuario marcar el número de destino final una vez completada la llamada original. El I.T.S.S., 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 I.T.S.S. de Cibertec.

El equipo se instala en los enlaces internacionales. El sistema es capaz de detectar y registrar la actividad DTMF en las llamadas salientes que se completen. 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 no actuará, pues lo hará sólo si se trata de marcación a terceros países. Una vez hecho 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 tomarán 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 retroalimentar periódicamente el sistema.

Fraude ISR (reventa de minutos) "International Simple Resale"

El I.T.S.S. de Cibertec identifica el fraude ISR en tres grupos:

1. ISR-I Detección del aterrizaje de tráfico ilegal y conectado dentro de la red de la Administración Local Mayor.

2. 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.

3. 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.

2.1.1.1 Arquitectura del I.T.S.S.

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 está totalmente sincronizado en fecha y hora con una precisión aproximadamente de 50 milésimas de segundo y que todos los puntos del sistema, tanto el RTG como los TMS envían sus reportes al Centro de Procesamiento y Control (CPS). Usualmente ésta 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 intricada 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 I.T.S.S.. 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 tarjeta HDLC-256 dentro del I.T.S.S.

La tarjeta HDLC es una pequeña parte dentro de toda la arquitectura que compone el I.T.S.S.. Se localiza específicamente en las unidades TMS, para comunicar la información capturada por las tarjetas CPU E1 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 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 con el sistema de cómputo.

Es necesario en este punto, ya que se ha aclarado la aplicación que tiene el sistema I.T.S.S, y la función que cumple la tarjeta HDLC-256 dentro de la infraestructura del mismo, ahondar en los componentes que la conforman, y de esta manera aclarar el funcionamiento de la misma, y los problemas que presentan cada uno de los dispositivos que se están utilizando.

Sin embargo es importante también, antes de explicar el funcionamiento de cada uno de los componentes, explicar el formato de transmisión y el protocolo de señalización utilizado por la CPU E1, para comunicarse con la tarjeta HDLC.

2.1.2.1 Formato de transmisión ST-Bus

El ST-Bus fue creado por Mitel Semiconductors para poder interfazar fácilmente información de tipo audio, video, control y datos cuando ésta se encuentra digitalizada.

Este es un bus serial que requiere de una señal de framing para el alineamiento del frame, una señal de reloj para el temporizado del bus, y las líneas de información serial. El uso de "streams" o "hileras de información" minimiza el área que se necesita en la tarjeta para la transferencia de datos.

La tasa de transferencia por "stream" es de 2.048, 4,096 o 8,196 Mbit/s. Este stream se divide en frames o cuadros cuyo periodo de 125μs. El inicio de un "frame" o trama de ST-Bus se indica por una señal de "framing" o bien de inicio de trama. Un canal ocupa lo que se denomina una ventana temporal (*time slot*). La tasa de transferencia, excepto para el caso de 2.048 Mbit/s es de la mitad del reloj de entrada. Se puede realizar una partición en el tiempo de la trama en los canales que sean necesarios, y de esta forma se podrá transportar información para diferentes dispositivos en un solo ST-Bus. Esta técnica se conoce como TDM (Multiplexación por división de tiempo).

El TDM consiste en combinar varias señales de información, las cuales han sido muestreadas por algún método dado, en el tiempo según una secuencia dada. De esta manera, se tendrá en periodos de tiempos exactos la información de una señal específica.

Para el alineamiento de las señales se utilizan dos métodos. Se produce un pulso al inicio de un frame, o bien se utiliza una señal en estilo de habilitador, la cual se expande habilitada mientras se transmite. Por lo que se puede hablar de la existencia de dos líneas de control y una de información

El diagrama en el tiempo de este tipo de comunicación que se utiliza en la entrada de la tarjeta HDLC-256 de Cibertec se muestra en la figura 2.1.

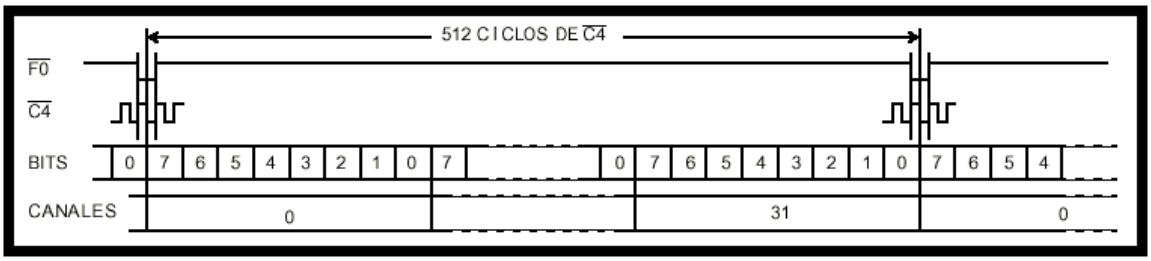


Figura 2.1 Diagrama de ubicación de los canales de información en el tiempo para el caso de 32 canales.

Los niveles lógicos de este bus usados en la tarjeta HDLC-256 son de 0V y 5V, con un nivel bajo máximo de 0.8V y un nivel alto mínimo de 2,8V. Además se utilizó la división del frame en bytes, obteniéndose 32 canales de información.

2.1.2.2 Protocolo HDLC (High Level Data Link Control)

El protocolo HDLC es definido en el X.25 (nivel 2) con las recomendaciones de la CCITT.

En este protocolo se divide la información en cuadros o “frames”. El formato de cada “frame” o cuadro se muestra en la tabla 2.1.

Tabla 2.1 Formato del cuadro de transmisión para el protocolo HDLC

Bandera de inicio	Campo para datos	Código para detección de error (FCS)	Bandera de Fin
Un byte	N bytes (n>=2)	Dos bytes	Un byte

La *bandera de inicio* y la *bandera de fin* es un patrón de 8 bits (01111110) las cuales definen las barreras del cuadro de transmisión. La sección de transmisión se encarga de adjuntar estas dos banderas en el cuadro que será transmitido. La sección de recepción se encarga de hacer un análisis bit a bit para localizar los bytes de bandera. Estas banderas se usan únicamente para sincronizar los cuadros.

El *campo de datos* corresponde a las direcciones de control y a los datos, como se define en las recomendaciones de la CCITT. Este campo puede tener un mínimo de dos bytes de longitud.

El chequeo de secuencia del cuadro (FCS) responde a un polinomio generador. Se compara los cálculos de la sección de transmisión con los de la sección de recepción y si se genera un resultado distinto, existe un error en la transmisión.

2.1.2.3 El controlador de protocolo HDLC de Mitel (MT8952)

El diagrama funcional del controlador de protocolo MT8952 se puede observar en la figura 2.2. Éste posee dos puertos: un puerto serie y un puerto paralelo. El puerto serie transmite y recibe paquetes de datos con formato ST-Bus y protocolo DIC. El puerto paralelo realiza la interfaz con el microprocesador para accesar los registros internos del controlador.

El puerto serie puede ser configurado para operar en dos modos dependiendo de la programación del bit IC en el registro de control. El chip puede recibir o transmitir información en el canal seleccionado si se está trabajando con formato ST-Bus; o bien, puede utilizar las señales de habilitación #RxCEN y #TxCEN para recibir o transmitir información a una tasa igual al reloj que se esté utilizando.

El puerto paralelo permite la transferencia de datos entre el controlador de protocolo y el sistema de bus del microprocesador. Esta interfaz consiste de un bus de datos de 8 bits, un bus de direcciones de cuatro bits, un reloj, un seleccionador de chip y una señal para el control de lectura/escritura. Posee una línea de interrupción, por lo que para identificar la fuente que genera la interrupción se debe generar un polling en los registros internos del controlador.

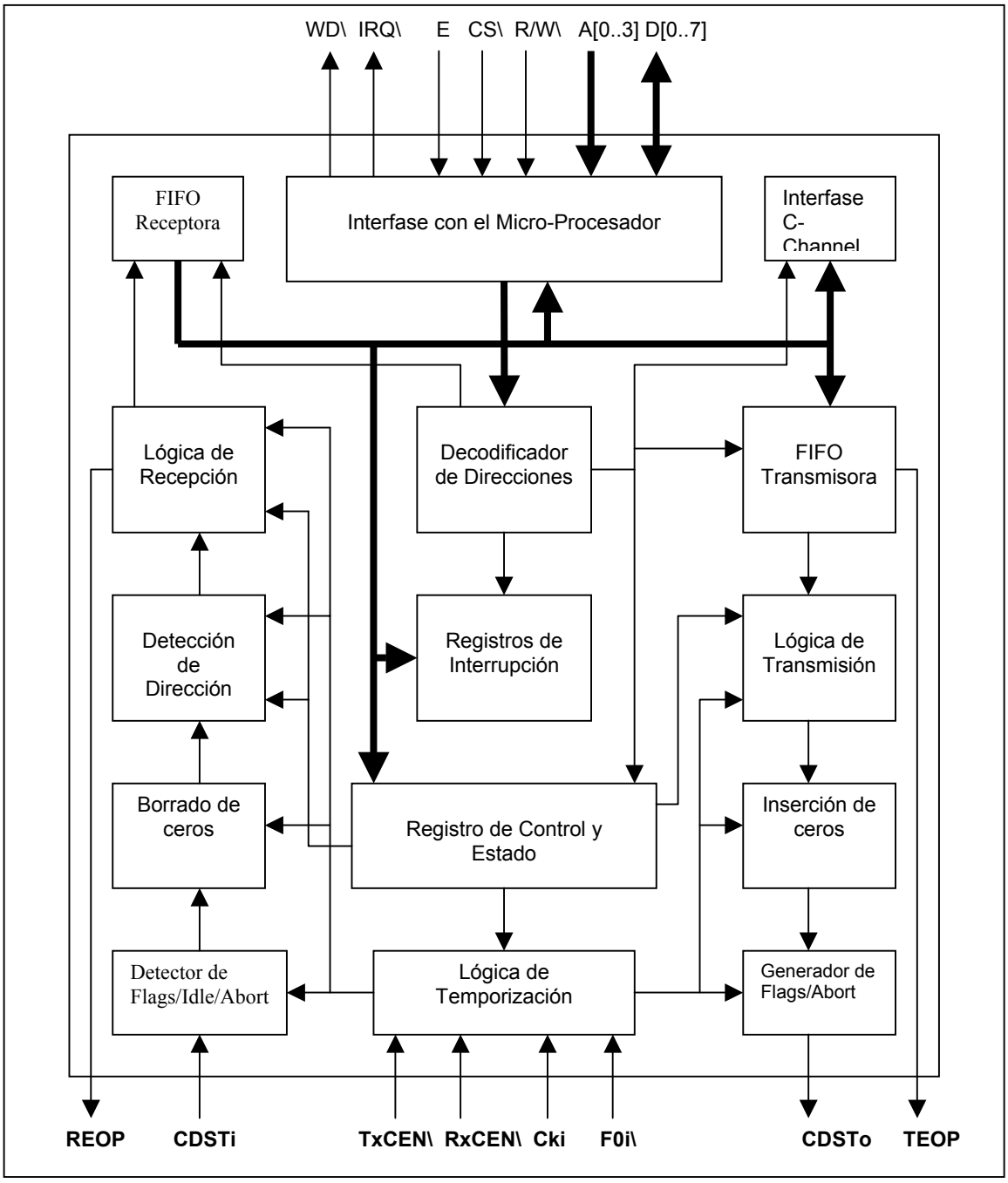


Figura 2.2 Diagrama de bloques del controlador de protocolo HDLC MT8952.

Las señales de TEOP y REOP indican el fin de un paquete transmitido o recibido respectivamente. TEOP es un pulso que se produce durante el último bit de la bandera de fin, o bien, en la secuencia de aborto. REOP se produce cuando un periodo de recepción se ha completado o bien se ha detectado una secuencia de aborto.

Utiliza dos temporizaciones, las cuales únicamente afectan al bus serial, y no al paralelo.

La temporización interna se utiliza para interfazar varios dispositivos utilizando el ST-bus y se utilizan las señales F0i\ y C2i/C4i para alineamiento y temporización. El circuito integrado utiliza las señales de ST-Bus #F0i y C2i/#C4i, y habilita las secciones de transmisión y recepción en el "time slot" o canal ST-Bus que se determinan por los bits TC0-TC3 en el registro de control de temporización. Es en este punto donde nace una de las debilidades de la tarjeta HDLC-256, ya que este registro sólo permite el manejo de cuatro canales ST-Bus, con algunas limitaciones, como se muestra en la explicación de las funciones de este registro. Esto acarrea que sea necesario colocar ocho controladores de protocolo HDLC en la tarjeta HDLC-256, y por tanto que se deban conectar 16 cables para las tareas de recepción y transmisión.

La temporización externa, la recepción y transmisión son controladas por las señales #RxCEN y #TxCEN, y la temporización la toman del pin CKI.

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).

Utilizando el bit IFTF del registro de control es posible hacer una transferencia transparente de los datos, esto es, se pasarán los datos sin eliminar el protocolo. Esta función está disponible en ambos sentidos (transmisión y recepción), y cada sentido cuenta con 19 bytes de memoria interna para almacenar datos. El tamaño reducido de la memoria interna, provoca que no se puedan procesar más datos,

ya que esta se desborda rápidamente.

Posee un watchdog el cual permanecerá en alto, hasta el pulso 2^{10} de f0b. Si al llegar a este punto el registro encargado del control de este módulo no es limpiado, se activará esta bandera, escribiendo un cero en el pin asignado para esta labor.

En el modo de transmisión el LSB es el dato D0, y es el primero que se envía. En el modo de recepción el primer bit que se recibe es considerado como el LSB.

Registros internos del MT8952

Registro de estado de la FIFO:

Indica el estado de la FIFO de transmisión y recepción. Su estructura se define en la tabla 2.2.

Tabla 2.2 Estructura del registro de estado de la FIFO

D7	D6	D5	D4	D3	D2	D1	D0
Estado del byte de recepción		Estado de la FIFO de recepción		Estado de la FIFO de transmisión		0	0

Este registro permite conocer, si el byte leído es el último de la FIFO, si la FIFO encuentra llena o vacía, o bien que le faltan tres bytes para llenarse. La combinación de bits para estas descripciones pueden hallarse en las hojas de datos del chip proporcionadas por el fabricante.

Registro de dato recibido:

Es el dato que se lee de la FIFO.

Registro de dato transmitido:

Es el dato que va a ser transmitido del bus de datos a la FIFO.

Registro de control:

En este registro se realiza el control general de las funciones del dispositivo. La estructura de este registro se muestra en la tabla 2.3.

Tabla 2.3 Estructura del registro de control

D7	D6	D5	D4	D3	D2	D1	D0
Habilitación de transmisión	Habilitación de recepción	Detector de dirección llave	Limite para los seis bits más altos de dirección	Controla el tercer estado del bus, o bien la transmisión transparente		Es la bandera de que luego del próximo byte se dará una secuencia de aborto	Indica que es el ultimo byte que se escribirá en la FIFO de transmisión

Registro de dirección de recepción.

Es la dirección llave. Permite que sólo los datos que posean esta dirección sean aceptados por el controlador, los demás serán ignorados.

Registro de control del C-Channel

Es el dato que se transmite por el canal 1.

Registro de control de temporización:

Controla los modos de temporización del dispositivo entre otros ítemes relacionados con este tema. En este registro se localiza el problema que presenta el MT8952 en la tarjeta.

La estructura de este registro se presenta en la tabla 2.4.

Tabla 2.4 Estructura del registro de temporización

D7	D6	D5	D4	D3	D2	D1	D0
RST	IC	C1EN	BRCK	TC3	TC2	TC1	TC0

RST: Es el reset por software del sistema.

IC: Se utiliza para seleccionar el temporizado interno.

C1EN: Habilita al dispositivo para transmitir por el canal 1 el C-Canal.

BRCK: Se utiliza en el temporizado interno para seleccionar la tasa de transferencia.

TC0-TC3: En el modo de temporización interna las secciones de transmisión y recepción se habilitan durante los tiempos definidos por estos bits. Esto aplica únicamente en los canales de ST-Bus 0,2,3 y 4 que traen paquetes de información o en transferencia transparente. La tabla 2.5 muestra la forma de habilitación de los canales dados.

Tabla 2.5 Combinaciones de los bits de control para la habilitación de los canales del ST-Bus.

Bits de control de temporizado				Número de canal de ST-Bus	Bits por frame
TC3	TC2	TC1	TC0		
X	0	0	0	0	1
X	0	0	1	0	2
0	0	1	0	0	6
1	0	1	0	0	7
X	0	1	1	2	8
X	1	0	0	3	8
X	1	0	1	4	8
X	1	1	0	2 y 3	16
X	1	1	1	2,3 y 4	24

2.1.2.4 El 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 como DSP sino que se utiliza por su flexibilidad en la conexión con otros recursos.

Las principales características de este procesador son:

- Tiempo que toma realizar un ciclo de instrucción: 100 ns.
- 544 palabras de 16 bits en espacio de RAM interna.
- 1568 palabras de 16 bits programables como RAM de programa o datos.
- 128K palabras de 16 bits direccionables en total, para memoria de datos o de programa.
- ALU / Acumulador de 32 bits
- Multiplicador paralelo de 16 x 16 bits con resultado de 32 bits.
- Instrucción de multiplicación y acumulación en un solo ciclo.
- Repetición de instrucciones para uso eficiente del espacio de programa y velocidad de ejecución.
- Movimiento de datos y programa en bloques.
- Temporizador interno para operaciones de control.
- Ocho registros auxiliares con unidad aritmética dedicada.
- Pila de ocho niveles implementada en hardware.
- Dieciséis canales de entrada y dieciséis de salida.
- Registro de corrimiento de 16 bits.
- Estados de espera para comunicación con dispositivos externos más lentos.

- Puerto serie para interfaz directa con el codec.
- Entrada de sincronización para configuraciones sincrónicas multiprocesador.
- Interfaz para memoria de datos globales.
- Compatibilidad con código de procesadores anteriores.
- Capacidad para hacer DMA.
- Instrucciones para filtrado adaptivo, FFT, y precisión aritmética extendida.
- Generador de reloj interno.
- Tensión de alimentación: 5V.
- 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 2.3 se muestra el diagrama de bloques básico de este chip, en el que se puede destacar el multiplicador como unidad independiente o auxiliar de la ALU.

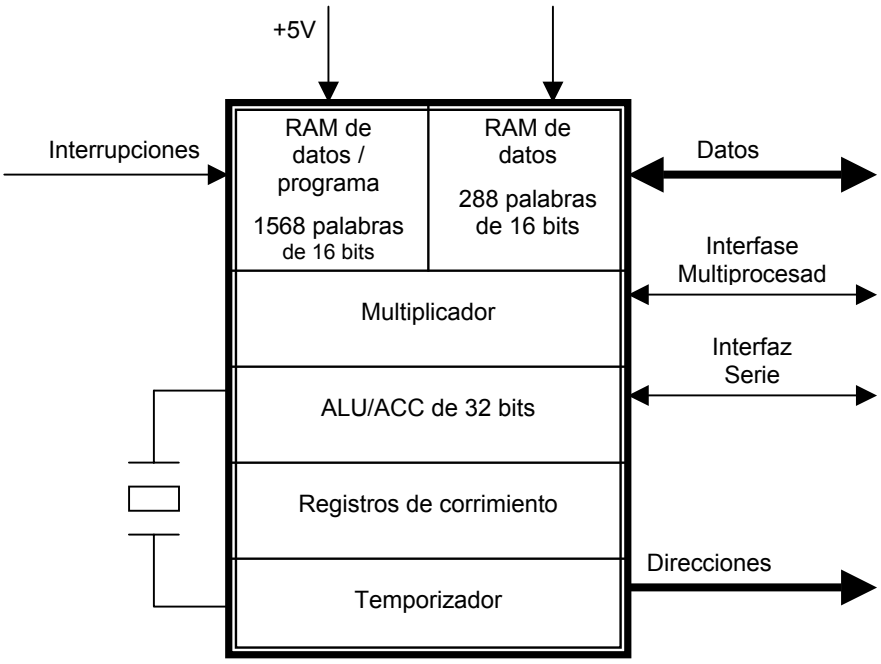


Figura 2.3 Diagrama de bloques del microprocesador TM320C26

El problema principal que se encontró con este microprocesador es que no tiene manejo del bus PCI, factor importante en el desarrollo del nuevo proyecto. Además, se encuentra descontinuado por su casa matriz, Texas Instruments, por lo que es de suma urgencia la búsqueda de un nuevo microprocesador, que cumpla con todas las funciones del DSP TM320C26 en la HDLC-256 y con las especificaciones necesarias para el nuevo modelo.

2.1.2.5 Las memorias FIFO CY7C419

Son memorias de 256 palabras de 9 bits. Se organizan de tal manera que el dato es leído en la misma secuencia en que fue escrito. Posee banderas de llena o vacía para evitar desbordamientos. Posee tres pines para prever la expansión tanto en capacidad como en ancho de palabra.

La lectura puede ser sincrónica y asincrónica en una tasa de 50 MHz.

El ciclo de lectura se encuentra entre 20 y 35 ns, además el tiempo de acceso

se encuentra entre 10 y 25 ns.

Posee una arquitectura de RAM de doble puerto, esto es, la celda activa solo la habilitación de lectura o escritura siendo independiente de otras, esto es necesario para accesar en forma asincrónica las entradas y las salidas. Un segundo beneficio es que 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 acceso a la última localidad vacía se indica con un alto en el pin FF.

Para escribir un dato en la FIFO se sigue el siguiente protocolo: El flanco de bajada de W inicia el ciclo de escritura. El dato debe aparecer en la entrada (D_0-D_8) después del tiempo t_{SD}, (Data Set – Up Date) y t_{HD} (Hold Data), antes del flanco de subida de la señal W, el dato se encontrará almacenado secuencialmente en la memoria.

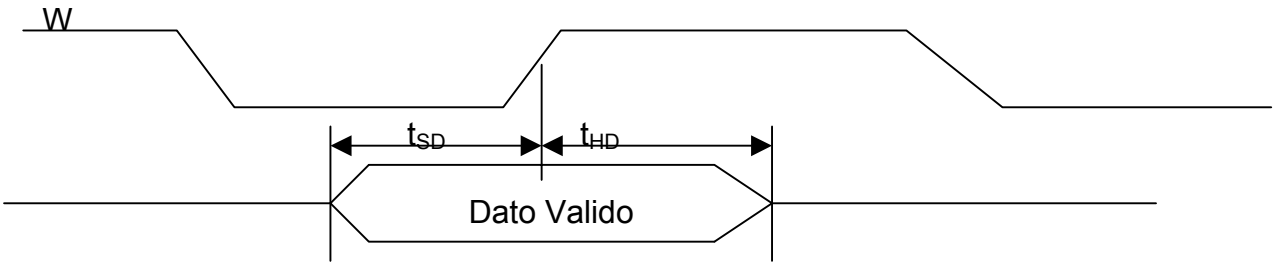


Figura 2.4 Diagrama temporal de escritura de un dato en la memoria FIFO

La bandera HF (media llena) se activa cuando se escribe en la posición media más uno. La bandera vacía se activa después de la transición de bajo a alto de W cuando la memoria se encuentra vacía. La bandera #FF se activa luego de que se da la escritura en la última posición de la memoria. Si se trata de escribir en una FIFO que se encuentra llena, la acción es ignorada, y el puntero no se incrementa.

Esta bandera (#FF) se desactiva luego de que se ha leído un dato de una memoria que se encontraba llena.

El protocolo de lectura en estas memorias es el siguiente: el flanco de bajada en R inicia el ciclo de lectura, siempre y cuando la bandera de memoria vacía (EF) no se encuentre en bajo. Las salidas (Q_0 – Q_8) se encuentran en tercer estado entre dos ciclos de lectura, es decir si no se está dando lectura de un dato. Se repite esta condición si la memoria se encuentra vacía, o cuando la memoria no está activa en una expansión de capacidad.

2.1.2.6 Memoria RAM CY7C185

Es una memoria de 8192 palabras de 8 bits. Es de fácil expansión y posee dos chip enable uno en bajo y otro en alto además de drivers de tercer estado.

Tiene un apagado automático que reduce su consumo hasta en un 70% cuando no se encuentra habilitada.

Un bajo en WE habilita la escritura/lectura de la memoria. Cuando la habilitación en bajo y en alto se encuentran activas y WE tiene un cero se está escribiendo en la memoria en la dirección que indiquen los pines de dirección. Cuando se tiene la habilitación en bajo y la habilitación en alto activadas según corresponda y WE con un uno se está leyendo la memoria, la posición que indique las líneas de dirección. Cuando no se tienen las habilitaciones, la memoria se encuentra en tercer estado.

Los voltajes de bajo son de máximo 0.4 V, y los voltajes mínimos de alto 2.4 V. Los voltajes máximos de alto es Vcc+0.3 V. El Vcc es de –0.5 a 7V.

2.1.2.7 Lógica de las GAL's en la tarjeta HDLC-256

La GAL localizada como IC9 se encarga de seleccionar uno de los controladores de protocolo MT8952, de forma tal que cuando el microprocesador direcciona uno de ellos el #CSn respectivo se activa en bajo y permite que el controlador correspondiente se active. Obsérvese que la lógica es dependiente de #STRB y de #DS debido a que los dispositivos se encuentran mapeados en memoria, y por lo tanto se deben de activar estas señales. La tabla 2.6 muestra la disposición de las señales en el dispositivo programable.

Tabla 2.6 Disposición física de las señales en la GAL IC9

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

La GAL IC14 habilita a los MT8259 y si estos se encuentran direccionados por el microprocesador se da la señal de que están o serán ocupados. En el caso de que estén direccionados pero no activados, ya que no se ha completado aún el ciclo de habilitación, o bien, se encuentran direccionados y ocupados se dará la señal de espera al microprocesador para que espere que los MT8259 terminen su trabajo. La tabla 2.7 muestra la disposición de los pines de esta GAL.

Tabla 2.7 Disposición de pines en la GAL IC14

Nombre de las señales de entrada	Pin
R/W	2
/DS	3
A13	4
A12	5
A11	6
A0	7
/STRB	8
Nombre de las señales de salida	
E-Clock,Q1	19,17
READY,BUSY	18,16
BUSY2	15
/FIFO	13
/DSP244	12

La GAL IC15 se encuentra involucrada con el manejo del bus ISA. Las ecuaciones de ésta se explican a continuación:

Reloj: Entrada del bus ISA (SYSCLK) si el dispositivo está seleccionado, (#Sel3xx en bajo).

WDP: Reset Watch Dog. Si se direcciona 100, hay una señal de escritura a puerto del PC y el dispositivo se encuentra seleccionado, se resetea el Watchdog del sistema.

WDE: es el reloj del watchdog y se genera de la siguiente manera se direcciona 001, se da una escritura en el I/O del PC.

WPC: La PC escribe en FIFO Rx, debe estar seleccionada la GAL, dada la señal IOW y se direcciona 00X.

PC244: Habilita Buffer de la FIFO Tx. Debe estar seleccionada la GAL de lectura a puerto direccionando 010.

RPC: PC lee memoria FIFO Tx. La GAL seleccionada da la señal de lectura a puerto. Se direcciona 11x. El OE de esta señal se controla con Sel3xx, si está la salida habilitada.

La tabla 2.8 muestra la disposición física del patillaje de este circuito integrado.

Tabla 2.8 Disposición física de las señales en la GAL IC15

Nombre de la señal de entrada	Pin
#sel3xx	2
IOR	3
IOW	4
A2	5
A1	6
A0	7
Nombre de la señal de salida	
#WDP	19
#WDE	18
#WPC	17
#PC244	16
#RPC	15
#RST	14
#OWS	13
#IOCS16	12

Finalmente 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. La disposición física de las señales en el dispositivo se muestra a continuación en la tabla 2.9.

Tabla 2.9 Disposición física de las señales en el integrado IC17.

Nombre de la señal de entrada	Pin
/FIFO	2
/STRB	3
RW	4
/DS	5
TINT	6
/HF	7
A13	8
X	9
Nombre de las señales de salida	
/RDSP	19
INT	18
/WDSP	17
/OERAM	16
/CE1	15
/WERAM	14
X	13
X	12

2.1.2.8 El bus ISA (Industry Standard Architecture)

Es un bus de 8 o 16 bits de datos, 24 bits de dirección por lo que puede direccionar 16 Megabytes. Su reloj es de 4-8.33MHz, asíncrono.

El conector del bus ISA provee fuentes de 5,12,-5,-12V. La tarjeta ISA provee acceso fácil y provee una forma fácil de detectar fallas. Sus características eléctricas se detallan en la tabla 2.10.

Las interrupciones son disparadas por flanco TTL- no mascarables. Es ideal para de bajo a mediano ancho de banda.

Tabla 2.10 Características de alimentación del bus ISA

+12V a 1.5A
-12V a 0.3A
+5V a 4.5A
-5V a 0.2A

La figura 2.5 muestra el diagrama de transferencia entrada/salida a memoria.

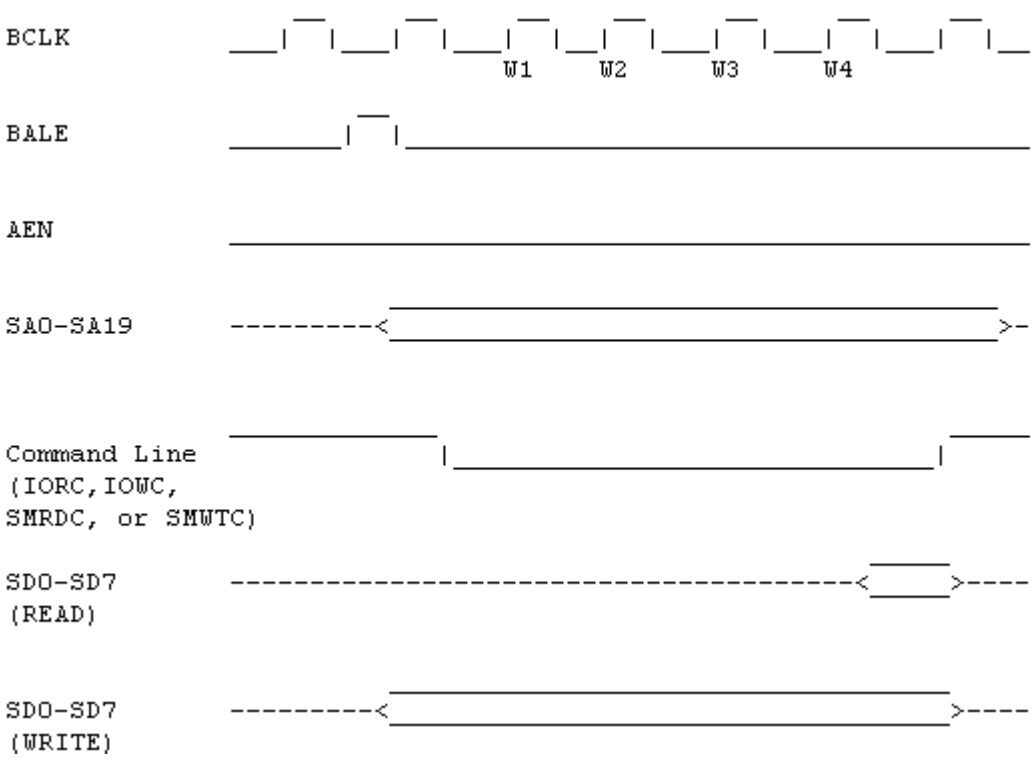


Figura 2.5 Diagrama de transferencia entrada/salida a memoria (se muestran 4 ciclos de espera). BALE es puesto en uno, y la dirección es puesta en el bus SA.

El dispositivo esclavo puede muestrear la dirección en SA durante el ciclo de bajada de BALE, ésta es válida hasta el final del último ciclo de transferencia. Puede notarse que AEN se mantiene en bajo durante todo el ciclo de transferencia debido a que no hay acceso al DMA.

La línea de comando es puesta en bajo (ya sea IOWC, IORC para comandos de entrada/salida o bien SMRDSC o SMWTC para comandos de memoria, read y write respectivamente). Para operaciones de escritura el dato debe estar en el bus de datos durante todo el ciclo de transferencia. Para operaciones de lectura el dato será válido en el flanco de bajada del último ciclo de la transferencia.

La línea NOWS (utilizada para acortar el número de tiempos de espera) se revisa en la mitad de cada ciclo de espera.

CHRDY (que indica que el canal se encuentra listo para dispositivos esclavos) se revisa durante la primera mitad de ciclo de reloj. Si está en bajo, se insertarán tiempos de espera. Por defecto, para transferencias de 8 bits, se tendrán 4 tiempos de espera, pero estos pueden ser cambiados.

La figura 2.6 muestra el diagrama temporal de transferencia para el caso de 16 bits de datos.

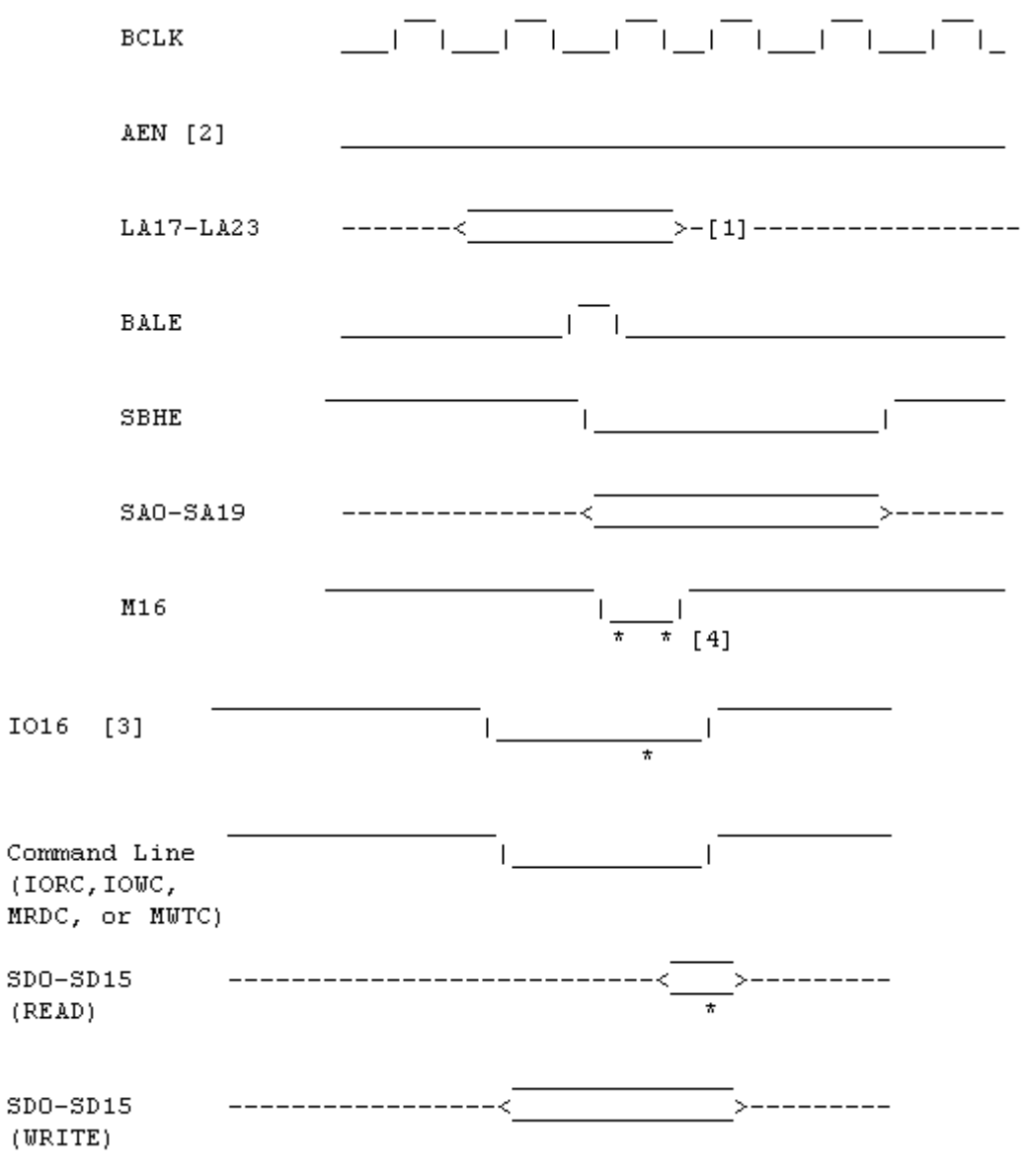


Figura 2.6 Diagrama de transferencia para 16 bits

[1] La porción de la dirección en LA para el siguiente ciclo puede ahora ser puesta en el bus. Esto se usa para que la tarjeta comience a decodificar la dirección con anticipación. El predireccionamiento debe estar activo.

[2] AEN en bajo pues no hay uso de DMA

[3]IO16 (IO tamaño 16) Algunos controladores muestrean esta señal durante el mismo ciclo de reloj que M16, dentro del primer tiempo de espera. En este caso IO16 necesita estar en bajo antes de que la dirección sea decodificada, lo cual es después que la línea de comando I/O es activa.

[4] M16 (acceso a memoria de 16 bits) es muestreada en el segundo tiempo en caso de que la tarjeta no active la señal en el primer muestreo.

La transferencia en 16 bits muestra el mismo esquema de temporización que en 8 bits. La dirección válida debe aparecer en el bus LA en el inicio del ciclo de transferencia. A diferencia del bus SA, el LA no posee Latch, por lo que la dirección no es válida en todo el ciclo de transferencia. El LA bus puede ser almacenado con el ciclo de bajada del BALE. La dirección en LA aparecerá al mismo tiempo que la dirección en SA. En cualquier sistema la dirección válida aparecerá con el flanco de bajada de BALE.

Las tarjetas de I/O no necesitan monitorear el bus LA o BALE, porque las direcciones de I/O están siempre dentro del espacio abarcado por el bus SA.

SBHE puede ser puesto en bajo por el sistema, y el adaptador de la tarjeta puede responder con IO16 o M16 en el tiempo apropiado, o la transferencia puede ejecutarse en 8 bits en dos intentos. Si el sistema necesita IO16 o M16 después de que las líneas de comando son válidas, esto requiere que IO16 o M16 se pongan en bajo luego que la dirección es decodificada. Si el sistema inicia un ciclo de memoria ignora IO16, y viceversa para ciclos I/O y M16.

Para operaciones de lectura, el dato es muestreado en el flanco de bajada del último ciclo de reloj. Para operaciones de escritura, el dato válido aparecerá en el bus después del final del ciclo, como se muestra en el diagrama. Mientras en el diagrama de tiempo se indica que el dato necesita muestrearse en el flanco de subida del reloj, la mayoría de los sistemas tendrán el valor del dato por todo el tiempo de reloj.

El problema en la utilización de este tipo de bus en la tarjeta es el grado de obsolescencia del mismo, pues actualmente las computadoras no poseen suficientes, o bien, no poseen, ranuras de este tipo, por lo que es necesario buscar un bus de comunicación con la computadora que ofrezca un uso más estandarizado, que pueda asegurar una vida útil de la tarjeta de al menos 5 años y sea capaz de transportar los datos a una mayor velocidad de lo que establece el estándar ISA, esto para poseer la mayor compatibilidad posible con los equipos computacionales.

2.1.2.9 Funcionamiento de la tarjeta HDLC-256

La tarjeta HDLC-256 posee un banco de 8 controladores de protocolo HDLC MT8952. En la etapa de recepción, los controladores toman la información proveniente de la tarjeta E1, en ST-Bus con protocolo HDLC, y realizan el trabajo de filtrado, esto es, eliminan la información de protocolo y almacenan el dato. Una vez realizado este filtrado, la información es enviada al bus de datos de 8 bits del microprocesador, donde se realiza el control de mensajes válidos y no válidos.

El microprocesador cuenta con un banco de memoria EPROM para almacenar el programa y con un banco de RAM. Una vez que se ha establecido en el microprocesador que el dato es válido, este es colocado en memorias FIFO, para luego enviarlo a la computadora y ésta lo procese. Una vez que la FIFO se encuentra llena hasta la mitad de su capacidad, existe un registro de estado en la tarjeta, el cual corresponde a las banderas de las FIFO, al detectarse que la bandera de “Half Full” se encuentra activada, la computadora recoge todos los datos que se encuentra en la memoria.

El bloque de GAL's tienen como función sincronizar el banco de controladores, así como algunas de las señales del bus ISA.

Esta versión de tarjeta posee las conexiones en los circuitos previstas para la transmisión, sin embargo, esta etapa no ha sido implementada.

Los problemas que esta tarjeta presenta es la baja cantidad de canales de ST-Bus que se pueden utilizar para poder manejar información con los controladores de protocolo actuales. La obsolescencia del microprocesador de la tarjeta, del bus ISA, y la baja velocidad de transmisión entre la tarjeta y la PC. El detalle técnico de los problemas en esta tarjeta se pueden localizar en la descripción de cada componente que la conforman y que se ve envuelto en estas debilidades.

2.2 Requerimientos de la empresa

La empresa requería que se cambiara el controlador de protocolo HDLC, de forma tal que se utilizara un dispositivo más flexible, que permitiera el uso de más de 2 canales por ST-Bus, así se podrá ampliar la cantidad de canales que se pueden procesar simultáneamente.

Además requería que se cambiara el bus ISA por un bus PCI, que es el más reciente en estándares de buses, y por lo tanto es de uso más común en las computadoras, lo que solucionaría el problema de obsolescencia en el bus de comunicación con la computadora.

Sin embargo el uso del bus PCI, trae como problema principal, buscar un nuevo microprocesador que sea capaz de realizar el arbitraje entre el bus principal y el bus secundario del dispositivo controlador de protocolo HDLC. Además se debe de escoger un microprocesador que asegure su existencia en el mercado, para evitar los problemas de descontinuidad, que es justamente lo que se está tratando de solucionar.

Se deseaba también incursionar en el área de montaje superficial, por lo tanto la selección de los componentes de la tarjeta debían de cumplir con este requisito.

2.3 Propuesta de solución

En vista de todos los problemas que presentaba la tarjeta HDLC-256 se propuso la utilización de un nuevo componente capaz de procesar hasta 128 canales HDLC, con salida a un bus PCI interno, de manera tal que se podrán utilizar más de dos canales del formato de transmisión ST-Bus, y al ser su salida a un bus PCI la velocidad de procesamiento de los datos incrementará, y se solucionaría en parte el problema cuando existen puntos muy concurridos, pues se podrán procesar más datos en menor tiempo, y al ser un sólo integrado el que procese los 32 canales de cuatro tarjetas E1, y pudiéndose aprovechar más de dos canales por cada data link se solucionaría el problema de espacio físico.

Por tanto, se propuso utilizar el PM7366 (Freedm-8), que es un circuito capaz de realizar la tarea anteriormente descrita, y cuya salida es a un bus PCI, el cual tiene comunicación con un microprocesador local, que posee interfaz PCI. Se están recibiendo 4 enlaces ST-Bus, en los cuales se utilizan 8 canales. El Freedm-8 se encarga de eliminar el protocolo HDLC, y dejar únicamente la información del encabezado de las llamadas para ser procesadas por el microprocesador local, esto en la etapa de recepción. En la etapa de transmisión, el dato enviado por el microprocesador local es empacado en formato ST-Bus con protocolo HDLC. En este microprocesador, para aprovechar el incremento de velocidad del bus, se realizarán, además de las tareas de filtrado que se realizan actualmente, parte de las tareas que actualmente ejecuta la computadora, y esos datos se trasladarán luego a la computadora huésped, agilizando el procesamiento de datos, y evitando interrumpir a la computadora en periodos muy cortos de tiempo. El microprocesador utilizado, 80960-RM, incluye el controlador de memoria para dos bancos de SDRAM y dos bancos de ROM, lo cual es ventajoso, tanto en el sentido de manejo de memoria por parte del microprocesador así como en el manejo de la memoria por parte del FREEDM-8. El microprocesador seleccionado posee la ventaja de poseer un puente PCI-PCI interno, por lo cual la efectiva comunicación del microprocesador

con la computadora huésped y el dispositivo PM7366 se basa en la programación del procesador.

El formato de los datos cambiará, en el sentido de que ya no se procesarán más en 8 bits, sino en 32 aprovechando el ancho de bus que presenta el formato PCI revisión 2.2.

La figura 2.7 muestra el diagrama de bloques general de esta solución:

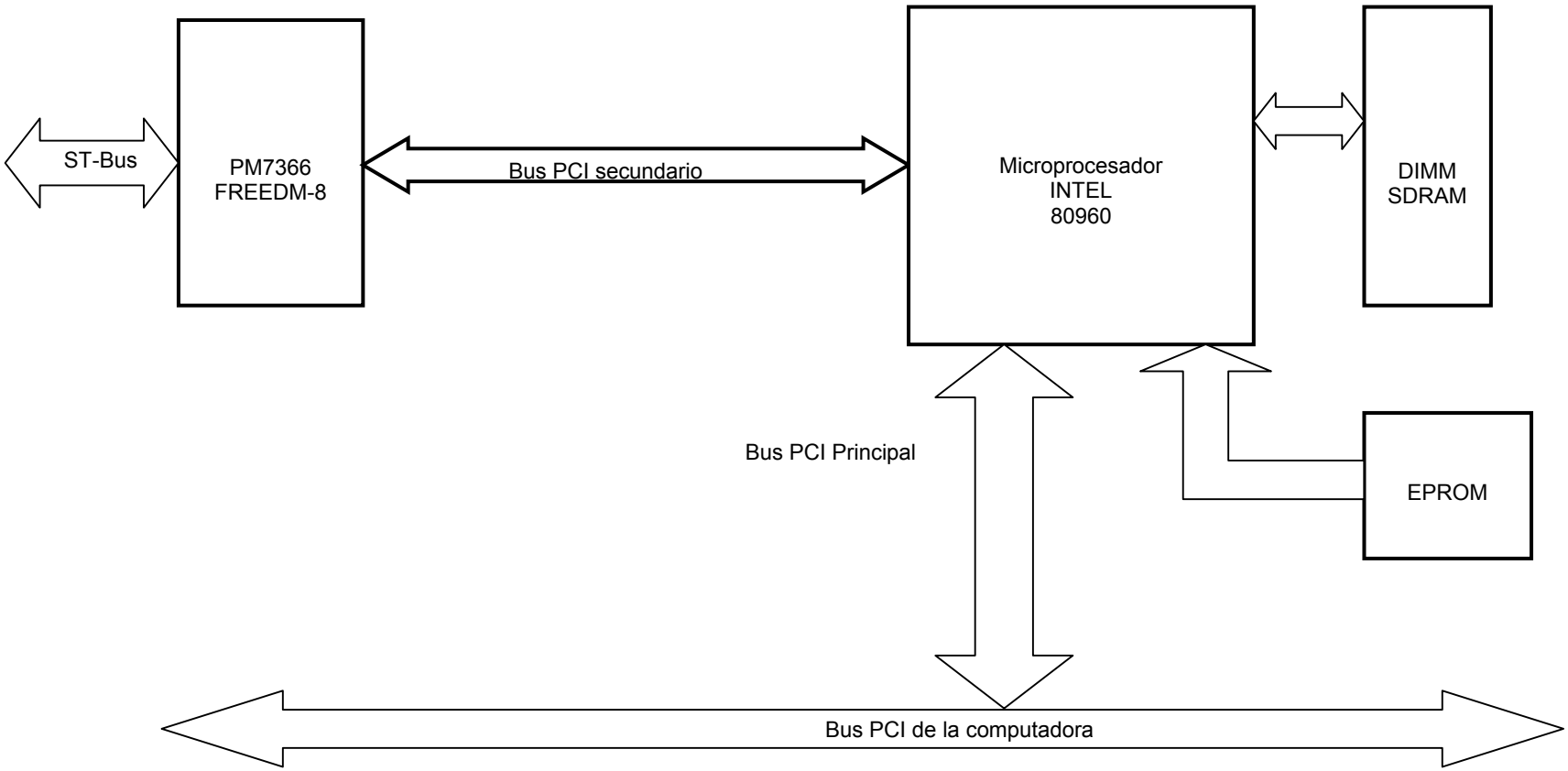


Figura 2.7 Diagrama de bloques general de la propuesta de solución desarrollada para la tarjeta HDLC-256.

Este sistema recibe en cada línea de ST-Bus, 8 canales de información, proveniente de una etapa llamada Data Link, la cual consta de una etapa de cuatro conectores RJ-45, donde viaja la información en formato HDLC.

En esta etapa de Data Link se realiza la lógica necesaria para poder realizar el alineamiento de las líneas ST-Bus. El FREEDM-8 utiliza un byte de alineamiento cuando trabaja con señalización E1, y no posee una señal de entrada para el pulso f0b utilizado para el alineamiento en ST-Bus. En cambio, busca que no exista transición en el reloj durante la transferencia del canal cero, como señal de inicio de una nueva trama. La figura 2.8 muestra la diferencia entre el alineamiento del ST-Bus con una tercera señal y la técnica de alineamiento aplicada por el PM7366.

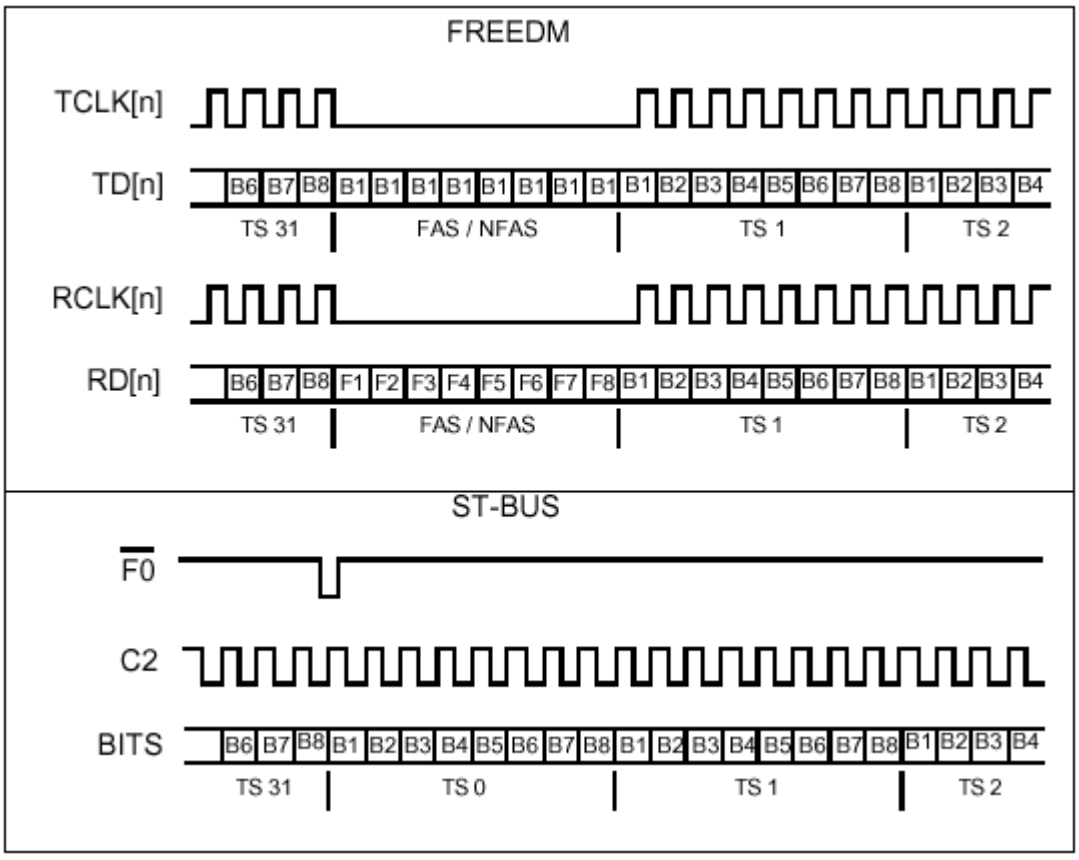


Figura 2.8 Alineamiento de las señales de ST-Bus. (Arriba) Alineamiento de la señalización en el FREEDM-8. (Abajo) Alineamiento con una tercera señal según el formato ST-Bus

Para realizar este tipo de alineamiento se debe realizar una lógica que antecede la conexión de las líneas de datos con el controlador de protocolo HDLC.

Existe un registro de desplazamiento, 74AC323, cuya función es realizar un corrimiento de la señal de alineamiento del ST-Bus, para asegurar que se podrá cancelar el reloj durante los 8 bits siguientes a que se produzca la señal de alineamiento del ST-Bus. A la vez, se realiza una multiplicación de todos los bits del registro con el reloj del ST-Bus negado, obteniéndose, mientras se encuentra el cero de alineamiento dentro del registro, un cero a la salida de la GAL, y al salir éste, se obtendrá el reloj negado, cumpliéndose con los requerimientos de interfaz al FREEDM-8.

Este circuito será montado en una tarjeta aparte, esto con el sentido de no ubicar lógica ajena al procesamiento de los datos de señalización, sobre la nueva versión de la tarjeta.

Una vez realizada la lógica de alineamiento de las señales, estas se encuentran listas para ser procesadas por el PM7366. Este recibe cuatro líneas de ST-bus. De cada enlace se toman ocho canales de información válida. Estos datos, luego de que el protocolo HDLC ha sido retirado, serán colocados en una FIFO interna cuyo tamaño es programable, y cuando ésta se encuentre llena, se dará una transferencia utilizando DMA. El funcionamiento de este dispositivo se encuentra en la descripción de hardware, en el capítulo 4. Utilizando el controlador de memoria que incorpora el microprocesador, el FREEDM-8 almacenará los datos recolectados en la memoria SDRAM. Una vez que los datos se encuentren en la SDRAM estos serán procesados por el microprocesador, y una vez procesados trasladados a la computadora huésped.

Capítulo 3
Procedimiento Metodológico

3.1 Conocer el formato de transmisión de datos ST-Bus.

Utilizando el material existente en la empresa de la compañía MITEL, se estudió el formato que conlleva el ST-Bus en cuanto a tramas, la forma de sincronizar y las velocidades características de transmisión. Además se observó experimentalmente la comunicación utilizando este formato, tomando las salidas en ST-Bus de una tarjeta E1. Se realizó documentación de lo estudiado, la cual se adjunta en el capítulo 2 de este documento.

3.2 Identificar los componentes que conforman la tarjeta HDLC-256.

Haciendo uso del circuito esquemático y del circuito en tarjeta, y luego de una descripción detallada del sistema I.T.S.S. por parte de la empresa, se identificaron los chips que se encuentran funcionando y su posición lógica (recepción, procesamiento y transmisión a la PC) en el funcionamiento de la tarjeta. Esta documentación se agrega en el capítulo 2 de este documento.

3.3 Buscar información acerca de los componentes que conforman la tarjeta HDLC-256.

Habiendo ubicado ya los componentes que trabajan en el sistema y su posición se buscó toda la información necesaria para cada uno de los componentes, con esto se logró establecer las bases para tener un entendimiento total de la tarjeta HDLC-256. Para estos se utilizaron manuales de MITEL para los controladores de protocolo HDLC, manuales del DSP, funcionamiento del bus ISA, de las memorias y los programas de las GAL's que la conforman, así se cubre la totalidad de los componentes de la tarjeta. Los datos de toda esta información se encuentra en la bibliografía de este documento, y la documentación se localiza en el capítulo de antecedentes.

3.4 Conocer el funcionamiento de la tarjeta HDLC-256.

Haciendo uso del circuito esquemático, la tarjeta, el material del programa de las GAL's y las hojas de datos de todos los componentes, se realizó un estudio de la tarjeta, desde la etapa de recepción de datos en formato ST-Bus, el proceso de filtrado de protocolo en los controladores de protocolo HDLC, la comunicación con el DSP, y el manejo de memoria RAM, FIFO y ROM de la tarjeta, así como la comunicación con la computadora utilizando un bus ISA y el manejo de la lógica respectiva para este bus.

De esta forma se logra una comprensión del problema a resolver, y de los requerimientos principales del sistema a desarrollar. Se documentó lo estudiado en el presente documento.

3.5 Conocer el principio de operación del bus PCI revisión 2.1.

Por medio del manual de especificación del Bus PCI 2.1, y utilizando la red de Internet, se estudió el principio de funcionamiento de este bus, como realiza la interacción con los periféricos, memoria, con los procesadores, el arbitraje entre buses secundarios, el manejo de dispositivos target, las especificaciones en cuanto a pistas en las tarjetas y velocidades de operación.

3.6 Analizar el funcionamiento del circuito integrado FREEDM-8.

Haciendo uso de todo el material dado por la empresa acerca del PM7364 y del PM7366, se analizaron sus opciones, y su funcionamiento, su programación y los alcances que tiene el mismo. Se decidió acerca del modelo que le convenía a la empresa para el diseño y se realizó documentación de las secciones más importantes de estos documentos, la cual se incorpora a este informe.

3.7 Determinar la lógica necesaria para interfazar los data links provenientes de la tarjeta E1, al circuito controlador de protocolo HDLC.

Se estudió en las hojas de datos de componentes los tipos de enlaces a los cuales puede manejar, y la forma en la que realiza el alineamiento de señalización

E1, además se estudió el documento "Interfazando el FREEM con ST-Bus", el cual brinda una opción para realizar esta interfaz, y poder comunicar el FREEDM con datos del tipo requerido. Sin embargo la opción requerida conllevaba la utilización de espacio físico significativo, por lo que se analizó un opción que conllevara menos espacio. Se estudió la forma de realizar toda esta lógica con GAL's, pero significaba también mucho espacio. Por lo que se analizó la utilización de registros universales, y PAL's. De esta manera se analizó el estado de fabricación de los componentes seleccionados, enviando correos electrónicos a las respectivas empresas fabricantes, los cuales respondieron que la producción estaba activa. Por lo tanto se procedió a analizar cada uno de los componentes, su conexión y programación.

3.8 Analizar el funcionamiento del microprocesador i960-RM

Se estudió las posibilidades que ofrecía el microprocesador 80960-VH en cuanto al manejo del bus PCI, y la forma de incorporar al mismo al sistema deseado. Se realizó el análisis del controlador de memoria que incorpora este microprocesador, los tipos de memoria que este podía soportar. Apartir de este estudio se determinó que este procesador no era la opción más apropiada para la tarjeta que se deseaba diseñar. Las razones que sustentan esta afirmación se pueden encontrar en el análisis de resultados.

Luego de determinar que el 80960-VH no era la mejor opción, se buscó en la red de Internet, en la página www.intel.com un nuevo procesador que cumpla en mayor forma con las características buscadas. Se eligió para el diseño el procesador 80960RM, que es de la misma familia del 80960-VH. Utilizando las hojas de datos y demás material sobre el tema que ofrece Intel se estudiaron las características principales de este procesador, el funcionamiento del controlador de memoria, y el manejo del bus secundario. De esta información se preparó un documento que se incluyen en la descripción del hardware utilizado.

3.8 Seleccionar los módulos de memoria a utilizar en el sistema.

Utilizando páginas de Internet de casas como Micrón, Microchip, Hitachi, se buscaron los módulos de memoria volátil y de programa. Se estudiaron sus ventajas de hardware, tiempos de acceso, costos y estados de producción. En vista de que los módulos de SDRAM es lo más utilizado en las computadoras actuales, y cumplen con estándares de construcción, lo que permite independizarse de fabricantes y de los tamaños del módulo, se decidió utilizar de este tipo.

La memoria EPROM seleccionada es del tipo PLCC, para que ocupe el menor espacio posible sobre la tarjeta.

3.8 Determinar la estructura de conexión de la memoria con el circuito controlador de protocolo HDLC.

Utilizando las hojas de datos del PM7366 y las hojas de datos del microprocesador, se estudió la forma en que maneja el microcontrolador 80960RM la memoria (controlador de memoria) y de esta manera se determinó como se le debe conectar la memoria al circuito controlador de protocolo HDLC.

3.9 Determinar la estructura de conexión del microprocesador local con el dispositivo controlador del protocolo HDLC.

Con las hojas de datos del controlador de memoria fue posible determinar las conexiones necesarias que tienen que existir entre el controlador de protocolo HDLC y el microprocesador, además de las demás señales de control, para que este posea un funcionamiento adecuado

3.10 Realizar la lógica necesaria para interfazar los relojes de cada ST-Bus con el FREEDM.

Utilizando el programador ABEL para las PAL que forman parte del control de la interfaz de las líneas ST-Bus con el PM7366, se realizó el programa de las PAL's, y utilizando el programa Orcad 9.1 se realizó el circuito que será la tarjeta Data Link, que permitirá el efectivo alineamiento de los ST-Bus, con el FREEDM-8.

3.11 Realizar el circuito esquemático de conexión entre el FREEDM-8, el microcontrolador 80960-RM y el módulo de memoria seleccionado en ORCAD 9.1.

Utilizando las hojas de datos de los componentes principales, siguiendo las especificaciones PCI, en cuanto a capacitancias de desacople, y resistencia de “pull up”, se realiza las conexiones de forma que la tarjeta pueda funcionar en dualidad de voltajes, es decir en 3.3V y 5V. Dadas estas características, se buscaron los componentes necesarios para realizar este tipo de tarjeta, así como los componentes necesarios para realizar las conexiones con la memoria, y los relojes del controlador de protocolo HDLC y microcontrolador. Esto se detalla en la descripción del hardware del proyecto. Las conexiones de los circuitos se realizaron en el programa Orcad 9.1. Además se buscó información acerca de empresas que posean experiencias en circuitos impresos de tarjetas PCI y de montaje superficial, tanto en el país como en el extranjero.

3.12 Realizar el mapa de memoria necesario para el efectivo funcionamiento del FREEDM-8.

Utilizando las hojas de datos del 80960RM se logró realizar el mapa de memoria del procesador. Con este mapa y las hojas de descripción del software para el FREEDM-8, se ubicaron las posiciones en memoria de las estructuras necesarias para el efectivo funcionamiento del controlador de protocolo HDLC, la localización de los registros normales para manejo y configuración del PM7366 y la ubicación de los registros de configuración del bus PCI para el mismo dispositivo.

3.13 Realizar las rutinas de inicialización de los componentes en su respectivo compilador.

Estas rutinas se realizaron en Turbo C en espera del software CTOOLS que es el compilador del sistema. Para entrega del proyecto se incluyen los diagramas de flujo que describe estas inicializaciones.

Capítulo 4
Descripción del Hardware Utilizado

4.1 El FREEDM-8 (PM7366)

4.1.1 Descripción general

Este dispositivo implementa un procesador HDLC y un manejo de memoria vía bus PCI para un máximo de 128 canales bidireccionales.

Para enlaces en las que la información viene dividida en canales (enlaces canalizados) de información, es posible manejar hasta 128 canales bidireccionales HDLC para ser asignados a time slots individuales, con un máximo de 32 o 24 según sea E1 o T1. En el caso de E1 es posible trabajar únicamente con 31 canales, ya que el canal cero se tomará para la alineación.

También es capaz de procesar datos no canalizados, es decir, se considera que cada enlace es un solo canal, lo que significa, que existirán 8 canales individuales.

Este dispositivo es capaz de procesar hasta 8 enlaces canalizados o no canalizados. La tasa de transferencia en los ocho enlaces es de hasta 4 MHz.

En la dirección de recepción, el FREEDM-8 permite el alineamiento del canal y la extracción y validación del dato. En cada canal HDLC se da la alineación usando la detección de la secuencia de la bandera, y realizando la inserción de ceros para la prevención de la emulación de las banderas. El dato que resulta del paquete se almacena en los 8 KB de RAM, llamado buffer parcial interno. Este buffer actúa como una FIFO, para cada uno de los canales asignados. Estos paquetes de datos son enviados a la memoria del host a través del bus PCI utilizando un controlador de DMA. El FREEDM-8 valida la secuencia de chequeado para cada paquete, y verifica que cada paquete sea un número entero de bytes dentro de un mínimo y máximo programable. El estado del paquete se actualiza antes de enlazar el paquete a una cola “ready” de recepción. El FREEDM-8 comunica al host PCI que hay paquetes en la cola “ready” de recepción , y opcionalmente se puede activar una de las interrupciones del bus.

Alternativamente, en la dirección de recepción, cuenta también con un modo de operación transparente. Cada canal configurado en este modo, pasa a la memoria del host todos los octetos, sin eliminar el protocolo.

En la dirección de transmisión, el bus PCI del host provee los paquetes a transmitir usando una cola “ready” de transmisión. Para cada canal HDLC, el FREEDM-8 agrega todo el protocolo necesario para su transmisión a los datos.

El FREEDM-8 es configurado, controlado y monitoreado utilizando la interfaz del bus PCI.

En la figura 4.1 se muestra el diagrama de bloques del FREEDM-8 (PM-7366).

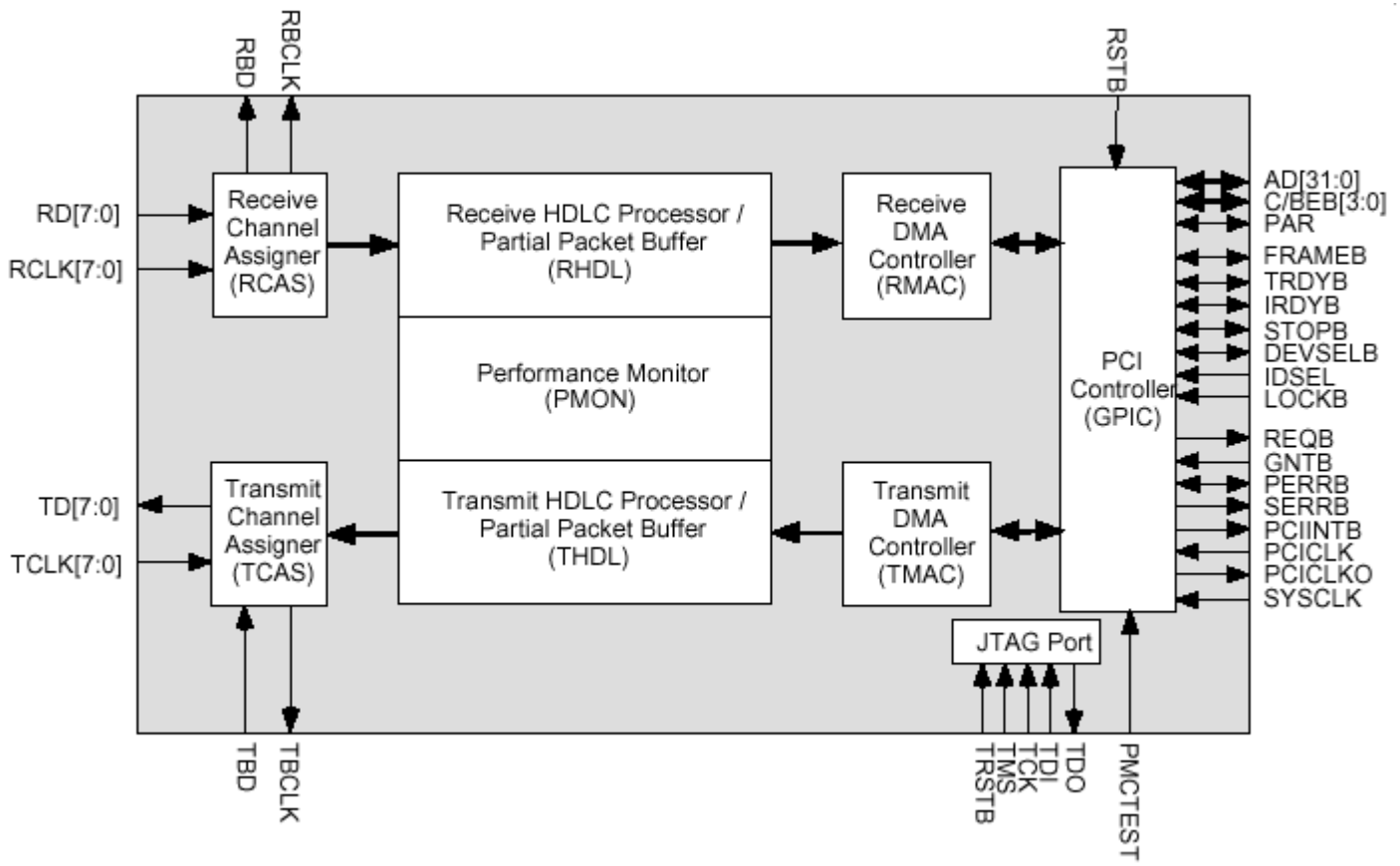


Figura 4.1 Diagrama de bloques del controlador de protocolo HDLC FREEDM-8 (PM-7366)

Descripción de pines

RCLK[n]:

Las líneas de señal de clock del receptor contienen las líneas de reloj para la temporización de los 8 enlaces independientes. El procesamiento de las señales se dan en orden descendente desde RCLK[0] a RCLK[7]. Las líneas RD[n] (dato recibido) son muestreadas en el flanco de subida del RCLK correspondiente.

Cuando se utiliza señalización E1, el reloj es suspendido durante todo el canal cero, con el objetivo de alinear la trama. Este reloj nominalmente es de 50% de ciclo de trabajo a 2.048 MHz para E1.

Para enlaces no canalizados se pueden alinear durante las tramas que no pertenecen al formato de transmisión HDLC.

RCLK[2:0] está nominalmente en un rango de trabajo entre 0 y 52 MHz. RCLK [31:3] esta nominalmente a 50% de ciclo de trabajo, entre 0 y 10 MHz.

RD[n]:

Las líneas de recepción de datos contienen los datos de los 8 posibles enlaces independientes en modo normal (PMCTEST está en bajo). La información es procesada con una prioridad descendente desde 0 hasta 7. Por esta razón la mayor tasa de conexión se debe encontrar en el RD[0].

Cuando se trata de enlaces E1 se debe utilizar el canal cero de información para el alineamiento.

Para enlaces con información no canalizada RD[n] contiene el paquete de datos HDLC. Para una comunicación correcta, los bits deben tener una posición definida dentro de la trama.

RBD:

Contiene el bit de error de la tasa de transferencia de datos en la recepción. RBD reporta acerca de un dato en una de las señales RD[7:0] seleccionada y se actualizada en el flanco de bajada de RBCLK. RBD se puede poner en tercer estado poniendo el RBEN bit en el FREEDM-8 master BERT control en bajo.

RBCLK:

Es la señal de reloj del BERT de recepción. RBCLK es la versión buferizada de la señal de RCLK[0:8] seleccionada. Puede ser puesto en tercer estado.

TCLK[n]:

Las líneas de señal de clock del transmisor contienen las líneas de reloj para la temporización de los 8 enlaces independientes. El procesamiento de las señales se dan en orden descendente desde TCLK[0] a TCLK[7]. Las líneas TD[n](dato recibido) son muestreadas en el flanco de bajada del TCLK correspondiente.

Cuando se utiliza señalización E1, el reloj es suspendido durante todo el canal cero, con el objetivo de alinear la trama. Este reloj nominalmente es de 50% de ciclo de trabajo a 2.048 MHz para E1.

Para enlaces no canalizados se pueden alinear durante las tramas que no pertenecen al formato de transmisión HDLC.

TCLK[2:0] está nominalmente en un rango de trabajo entre 0 y 52 MHz. TCLK [8:3] esta nominalmente a 50% de ciclo de trabajo, entre 0 y 10 MHz.

TD[n]:

Las líneas de transmisión de datos contienen los datos de los 8 posibles enlaces independientes en modo normal (PMCTEST está en bajo). La información es procesada con una prioridad descendente desde 0 hasta 7. Por esta razón la mayor tasa de conexión se debe encontrar en el TD[0].

Cuando se trata de enlaces E1 se debe utilizar el canal cero de información para el alineamiento.

Para enlaces con información no canalizada TD[n] contiene el paquete de datos HDLC. Para una comunicación correcta, los bits deben tener una posición definida dentro de la trama.

TBD:

Contiene el bit de error de la tasa de transferencia de datos en la transmisión. TBD reporta acerca de un dato en una de las señales TD[7:0] seleccionada y se actualizada en el flanco de bajada de TBCLK.

TBCLK:

Las líneas de señal de clock del transmisor contienen las líneas de reloj para la temporización de los 8 enlaces independientes. El procesamiento de las señales se dan en orden descendente desde TCLK[0] a TCLK[7]. Las líneas TD[n] (dato recibido) es muestreado en el flanco de bajada del TCLK correspondiente.

PCICLK:

Provee la temporización para el bus PCI. Nominalmente tiene un ciclo de trabajo del 50% y un rango de frecuencia de 0 a 33 MHz.

PCICLKO:

La salida del reloj del bus PCI. Es una versión buferizada del reloj PCI, y esta puede ser utilizada para manejar el reloj del sistema.

AD[n]:

Son las direcciones y datos del bus PCI. Durante el primer ciclo de reloj de una transacción contiene un byte de dirección física, y en los ciclos subsecuentes de la transacción se ubicarán los datos.

Una transacción se define como una fase de direcciones seguida de una o más fases de datos. La ubicación del byte más significativo, y del menos significativo es programable.

Cuando el FREEDM es el iniciador, AD[0:31] es una salida durante la primera fase de la transacción (direcciones). Para transacciones de escritura, AD[0:31] sigue siendo una salida. Para operaciones de lectura es una entrada.

Cuando el FREEDM es un target, AD[0:31] es una entrada de bus durante la fase de direcciones. Durante la escritura, las líneas continúan funcionando como entrada, y durante la lectura estas funcionan como salida.

Cuando este no se encuentra implicado en una transacción, estas salidas se encuentran en tercer estado.

C/BEB[0:3]:

El bus de comandos PCI o el byte de habilitación. Durante la primera transacción del bus, contiene el código de comando del bus. Para los siguientes ciclos informa acerca de cuales bytes traen datos válidos. Cuando alguno de estos bits se encuentra en alto, significa que el byte asociado no es válido.

PAR:

Indica la paridad del bus AD[31:0] y el bus C/BEB[3:0]. La paridad par es calculada sobre todas las 36 señales en el bus. Siempre reporta la paridad del dato del ciclo de reloj anterior.

Cuando el FREEDM-8 es el iniciador, PAR en una salida para escrituras y entrada para lecturas.

Cuando el FREEDM-8 es el target, PAR es una entrada para escritura y salida para lectura.

FRAMEB:

Identifica un ciclo de transacción. Cuando FRAMEB está en bajo, el inicio de una transacción se está produciendo. Cuando pasa a alto, se esta indicando que la última fase de datos está ocurriendo. Cuando el FREEDM-8 es el iniciador, FRAMEB es una salida. Cuando el FREEDM-8 es el target, FRAMEB es una entrada. En otras ocasiones se encuentra en tercer estado.

TRDYB:

Indica que la tarjeta ya esta lista para empezar o continuar una transacción. Esta señal actúa junto con IRDYB para completar una transacción. Cuando una transacción esta en proceso TRDYB se encuentra en alto para indicar que el target no ha completado la transacción, e inserta tiempos de espera. Se encuentra en bajo para indicar que ya la transacción ha sido finalizada.

Es una entrada cuando el FREEDM-8 es el iniciador, y una salida cuando es el target. Cuando se está accesando los registros del FREEDM-8 se mantiene en alto para extender los ciclos de datos en múltiplos del SYSCLK. Cuando el PCI no está envuelto en la transacción, se encuentra en tercer estado.

IRDYB:

Indica cuando el iniciador está listo para empezar o continuar una transacción. IRDYB trabaja en conjunto con TRDYB para completar la transacción de datos. Trabaja de la misma manera que en el bus PCI.

STOPB:

Pide al iniciador detener la transacción en el bus. Cuando el FREEDM es el iniciador es una entrada, cuando es un target es una salida.

IDSEL:

Habilita los accesos de lectura y escritura a los registros de configuración.

DEVSELB:

Indica que el target requiere de una transacción del bus.

LOCKB:

Usado para manejar el bloqueo de recursos en el bus PCI.

REQB:

Hace la petición a un árbitro externo para el control del bus PCI.

GNTB:

Indica la concesión del control sobre el bus PCI en respuesta a la petición del bus por medio de la vía REQB.

PCIINTB:

Señal de interrupción del bus PCI. Indica que hay petición de DMA.

PERRB:

Indica un error de paridad sobre los buses AD[31:0] y C/BEB[3:0]. Ocurre cuando la paridad calculada no corresponde con la paridad de la señal PAR.

SERRB:

Indica un error de paridad en la dirección.

SYSCLK:

Es el reloj del sistema. Posee un ciclo de trabajo del 50% y un rango de frecuencia de 25 MHz a 33 MHz.

RSTB:

Provee la señal de reset asincrónico.

PMCTEST:

Habilita al FREEDM-8 para el modo de test. Cuando PMCTEST se encuentra en alto, se dará una producción de vectores que podrán serán ejecutados para

verificar el funcionamiento del mismo.

TCK:

Provee el temporizado para las operaciones de test.

TMS:

Controla las operaciones de test que pueden ser ejecutadas utilizando el puerto IEEE P1149.1.

TDI:

Son los datos de Test. Necesita pull up.

TDO:

Salidas de los datos de test.

TRSTB:

Es el reset en el modo test. Necesita resistencia de pull up.

VBIAS [3:0]:

Provee vías de 5V para el manejo de dispositivos de 5V.

EN5V:

Habilita el bus PCI para trabajar en ambiente de 5V.

TA[0:10]:

Son las direcciones en el modo de test.

TRS:

Se utiliza para seleccionar entre acceso a registro normal o en modo test. Se encuentra en alto para seleccionar modo de test y en bajo para el modo normal.

TRDB:

Es la señal de habilitación de lectura, se encuentra en bajo durante un acceso de lectura en la producción de los vectores de test. En la operación normal se debe

encontrar en alto.

TDAT[0:15]:

Es el bus de datos bidireccional del modo de prueba.

Descripción funcional del dispositivo FREEDM-8

En la figura 4.2 se muestra el diagrama del protocolo HDLC que maneja el FREEDM-8.

	Bandera	Información	FSC	Bandera	Bandera	

Figura 4.2 Trama HDLC que es capaz de procesar el PM7366.

Los datos de entrada son examinados por los bytes de banderas (01111110) los cuales alinean el inicio y final de un paquete HDLC.

Utiliza la inserción de un cero después de la detección de cinco unos seguidos para evitar la emulación de la bandera. La longitud del paquete puede ser en múltiplos de bytes.

El mínimo tamaño del paquete de información es de dos bytes (dirección y control).

El CRC es calculado utilizando el polinomio de la CCITT o el CRC-32. En el primer caso el polinomio utilizado es: $g(X)=1+g_1 X+g_2X^2+...+g_{n-1}X^{n-1}+X^n$, y en el caso de CRC-32 es $g(X)=1+X+X^2+X^4+X^5+X^7+X^8+X^{10}+X^{11}+X^{16}+X^{22}+X^{23}+X^{26}+X^{32}$.

Asignador de Canal de Recepción

Procesa hasta ocho enlaces seriales. Cada canal es independiente, por lo que posee su propio reloj. Cuando se dan eventos en que varios streams han sido acumulados, existe una priorización de los enlaces, donde el enlace 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).

Asignador de canal

El block asignador de canal determina el numero de canal del dato que se esta procesando actualmente. Posee 256 palabras de RAM. La dirección en la RAM se construye concatenando el numero de canal y el numero de Time slot.

Procesador de Recepción de HDLC/ Buffer Parcial

Procesa hasta 128 canales. Cada canal puede ser configurado para la detección de la secuencia de banderas, el bit de prevención de emulación de bandera y verificación de CRC. El paquete de datos se escribe en el buffer parcial. Al final del frame, el status del paquete incluye el error del CRC, el error de alineamiento del octeto y la violación a la longitud máxima.

Procesador HDLC

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 se accesa utilizando accesos indirectos al canal. Cuando una operación indirecta es efectuada, la información es accesada de la RAM durante un ciclo nulo de reloj generado por el asignador de canal de recepción. Escribiendo un nuevo dato en el canal resetea todo el vector de estado del canal.

Procesador del Buffer Parcial de Paquete (Partial Packet Buffer Processor).

Controla los 8KB de la RAM interna del FREEDM-8, el cual está dividido 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 que forman la FIFO se encuentren continuos.

El software del sistema es el responsable por el alineamiento de los bloques a

un canal de la FIFO específico.

Este procesador se encuentra dividido en tres secciones:

1. Escritor
2. Lector
3. 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 programable por canal. 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. Cuando el lector realiza un número dado de transacciones del block o de fin de paquete al block 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 transacción contando para todos los canales. El roamer incrementa el contador de transacción cuando el escritor da una nueva señal de transacción. El roamer 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. El roamer

actualiza el contador de transacción y limpia la bandera de non-zero transaction count si se requiere. El roamer, después de esto está listo para atender al próximo canal que posea la bandera de transacción en alto.

El escritor y lector determina si las FIFO's 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 sobreflujo cuando el escritor trata de escribir un dato en un bloque que tiene la bandera en alto.

Controlador DMA de recepción (RMAC)

Almacena el paquete de datos recibidos en la memoria del host. Los accesos a memoria se realizan por medio de un block controlador de PCI. El RMAC y el host realizan su comunicación utilizando una serie de descriptores.

Estructuras de datos

La RMAC copia los buffer de datos en la memoria del host utilizando las RPD, RPDR, RPDDRR y las RPDRF.

Los Receive Packet Descriptor (RPD) guardan el tamaño del buffer de datos, la dirección del buffer y la información de status del paquete.

Los Receive Packet Descriptor Reference (RPDR) es un puntero que es usado como índice en la tabla de RPD.

Los Receive Packet Descriptor Reference Ready y Receive Packet Descriptor Reference Free permiten a la RMAC y al host pasar RPDR’s en las dos direcciones.

Receive Packet Descriptor (RPD)

El host escribe los campos del RPD los cuales describen el tamaño y dirección del buffer en la memoria del host. El RMAC escribe los campos del RPD que provee el número de bytes usados en cada buffer de datos, la información del enlace y el status del paquete recibido. Los RPD son almacenados en la memoria del host y en una tabla de descriptores de recepción.

La estructura que presentan estos descriptores se muestra en la tabla 4.1.

Tabla 4.1 Estructura de un descriptor del PM7366.

31 0

<table>
<tr><td colspan="5">Dirección inicial del buffer de datos [31:0]</td></tr>
<tr><td>Bytes en Buffer [0:15]</td><td>Status [0:5]</td><td>Offset [0:1]</td><td>CE</td><td>RCC [6:0]</td></tr>
<tr><td colspan="2">Reservado (18)</td><td colspan="3">Next RPD Pointer [13:0]</td></tr>
<tr><td colspan="2">Reservado (16)</td><td colspan="3">Tamaño del Buffer de Recepción [18:0]</td></tr>
</table>

Dirección de inicio del buffer de datos: Apunta al buffer de datos en el host. No se encuentra durante la inicialización. Es válida para todos los RPD's.

Receive Channel Code (RCC): Lo usa la RMAC para indicar cual canal está asociado con esta RPD.

Para una lista enlazada de RPD's todos los campos de RCC son válidos y todos contienen el mismo número de canal.

Fin de cadena (CE): Indica el fin de una lista enlazada de RPD's. Cuando el CE es un uno lógico, el RPD actual es el último RPD de una lista enlaza.

Status: Indica el estado del paquete recibido según la tabla 4.2.

Tabla 4.2 Lógica de bits en el espacio de status

Status[0]	Desbordamiento del buffer de recepción.
Status[1]	El paquete excede el máximo tamaño permitido.
Status[2]	Error en el CRC
Status[3]	La longitud del paquete no contiene un número exacto de bytes.
Status[4]	Se detectó que se abortó la secuencia de HDLC.
Status[5]	No se usa.

Bytes en Buffer:

Indica el número de bytes que actualmente se usan en el buffer de datos del RPD actual.

Próximo Puntero de RPD:

Almacena el RPDR que habilita al RMAC a soportar listas enlazadas de RPD's. Este campo es válido cuando CE es cero lógico, y contiene el RPDR del siguiente RPD en la lista enlazada. Se utiliza cuando se necesita más de un buffer.

Tamaño del buffer de recepción:

Da el tamaño en bytes del buffer de datos del RPD actual. Se configura durante la inicialización. Este espacio no puede ser cero, debe ser múltiplo de 16 y no debe ser mayor a 32752.

El tamaño del buffer de recepción de datos y la dirección de inicio son escritos únicamente por el host. El RMAC lee estos campos para determinar donde almacenar el dato. Los campos restantes son escritos por el RMAC.

Tabla de descriptores de recepción

Reside en la memoria del host y almacena todos los RPD's. Puede almacenar un máximo de 16384 RPD's. La base de esta tabla es programable utilizando el registro Rx Packet Descriptor Table Base (RPDTB). Esta tabla es indexada por un Receive Packet Descriptor Reference (RPDR), el cual es un puntero de 14 bits definido por el offset por un RPD de la base de la tabla.

De esta manera la dirección de un RPD se obtiene de la siguiente forma:

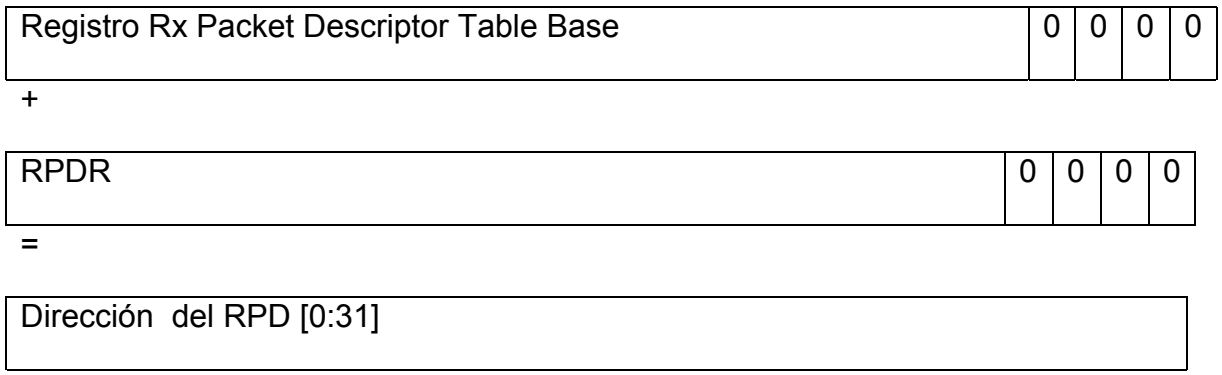


Figura 4.3 Formación de la dirección de un RPD

El registro de base de la tabla de descriptores de recepción está en el RMAC. Lo inicializa el host y es accesado por las colas de recepción de paquetes.

Colas de recepción de paquetes

Son utilizados para transmitir RPDR entre el host y el RMAC. Son tres colas:

- RPDR Large Buffer Free Queue (RPDRLFQ)
- RPDR Large Buffer Free Queue (RPDRSFQ)
- RPDR Ready Queue (RPDRRQ)

Las tres colas contienen RPDR que hacen referencia a RPD's que definen buffer listos para el procesamiento del host. El RMAC corre o saca RPDR's de las colas

“ free” cuando se necesitan buffer de datos libres. El RMAC coloca un RPDR en la cola de listos cuando se ha llenado el buffer con los datos de un paquete completo.

El host remueve RPDR’s de la cola lista para procesar los buffer de datos. El host coloca el RPDR de nuevo en la cola “free” después que termina de leer los datos en el buffer.

Cuando se comienza a procesar un paquete el RMAC utiliza un small buffer RPD para almacenar los paquetes de datos. Si el paquete requiere más de un buffer de datos, el RMAC utiliza un buffer de mayor tamaño (large buffer) de RPD’s, para almacenar el resto del paquete. El RMAC enlaza todos los RPD’s requeridos para almacenar los paquetes y devuelve el RPDR asociado con el primer RPD en la cola ready.

Todas las colas de paquetes recibidos residen en la memoria del host y están definidos por la Rx Queue Base (RQB) register y registros índices los cuales residen en la RMAC. La Rx Base Queue es la dirección base para la cola de recepción de datos. Cada cola de paquetes tiene cuatro registros índices los cuales definen el inicio, el fin y las localizaciones de lectura y escritura en la cola. Cada registro índice es de 16 bits y define un offset de la base de recepción. La dirección en el host de un RPDR se calcula sumando el registro índice al registro Rx Queue Base. El host inicializa el registro de base de colas y todos los registros índices. Cuando el RMAC o el host remueve elementos de la cola, el que los mueve actualiza el puntero de lectura para esa cola. Cuando una entidad pone uno se debe actualizar el puntero de escritura.

El índice de lectura para cada cola apunta al último RPDR válido mientras el índice de escritura apunta a la primera posición válida dentro de la cola; en esta posición es posible escribir un dato.

El índice de final apunta a la localización de memoria donde termina la cola. En esta localidad no pueden ser escritos RPDR. Nótese que el índice de inicio de una cola puede ser el final de otra.

Una cola está vacía cuando el índice de lectura de es uno menor que el de escritura. Una cola está llena cuando el índice de lectura es igual al índice de escritura.

Las colas ready poseen un campo de status. La RMAC llena estos campos para marcar si el paquete ha sido recibido efectivamente o no. El host puede forzar a la RMAC a almacenar los datos recibidos en bufferes de un solo tamaño.

El RMAC enlaza un buffer large de RPD a un small buffer de RPD si se necesita más de un buffer por paquete.

La estructura de una cola RPDRR se muestra en la tabla 4.3:

Tabla 4.3 Elementos de una cola RPDRR

15 0

Status [1:0]	RPDR[13:0]

Status[1:0]:

La decodificación para estos espacios se muestra en la tabla 4.4:

Tabla 4.4 Decodificación para los bits de status de RPDRR

00	Recepción correcta del paquete de datos
01	Recepción incorrecta del paquete de datos
10	No se ha dado el paquete de datos
11	Reservado

Aunque STATUS+RPDR totalizan tan sólo 16 bits, cada entrada de una cola son 32 bits. Cuando el RMAC escribe un STATUS+RPDR en una cola de listos, este coloca el tercer byte es ceros y el cuarto no lo modifica (el más significativo).

Tabla de Receive Channel Descriptor Reference

En una base por canal, el RMAC coloca en caché la información al igual que la información actual del DMA se coloca en una tabla de Receive Channel Descriptor Reference (RCDR). El RMAC puede procesar hasta 128 canales y almacenar tres palabras dobles por canal. Esta información se almacena en un caché interno en orden decreciente al número del acceso del bus del host requerido para procesar cada dato. La estructura de esta tabla se muestra en la figura 4.4.

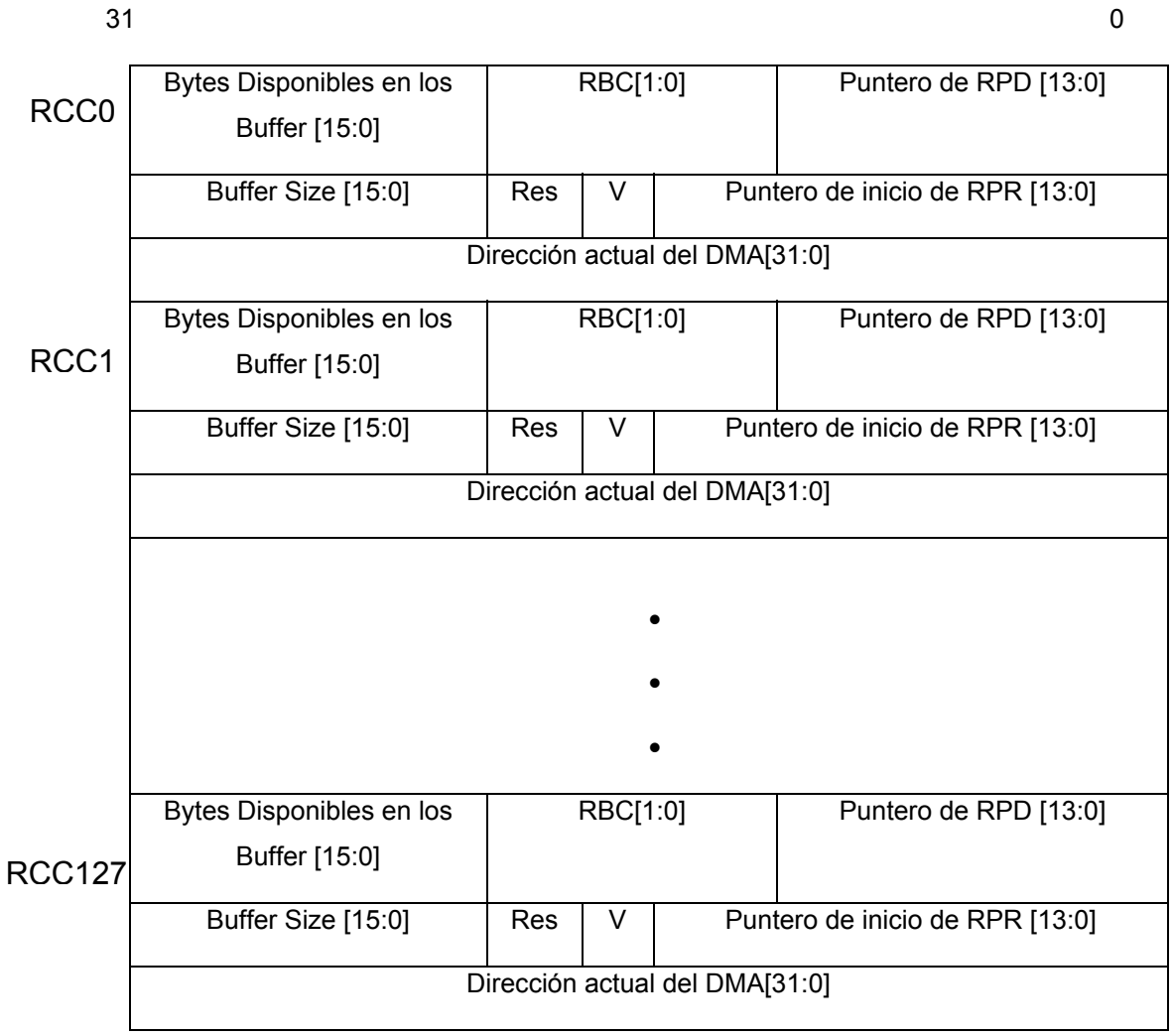


Figura 4.4 Tabla de descriptores de referencia para la recepción

Bytes disponible en el buffer[15:0]:

Este campo es utilizado para mantener el número de bytes disponibles en el buffer de datos actual. El RMAC inicializa este campo como el tamaño del buffer menos el offset del buffer. Este campo es decrementado cada vez que un byte es escrito en el buffer.

RBC[1:0]:

Se utiliza para mantener el número de buffer usados cuando se almacenan paquetes que no poseen limites. El RMAC inicializa el RBC con el valor de RAWMAX[1:0] en el registro de control del RMAC. Cada vez que un buffer se llena este campo es decrementado. Cuando este campo es cero la cadena de RPDs es puesto en la cadena de ready y otra nueva cadena es iniciada.

Puntero RPD[13:0]:

Contiene el puntero del RPD actual.

Tamaño del buffer:

Tiene el tamaño en bytes del buffer que actualmente se está escribiendo.

V:

Indica que el paquete actual ha sido recibido bien en el DMA. Cuando el bit V está en uno, el otro campo en la tabla de RCDR que entra en el canal de DMA tiene información válida.

Start RPD Pointer[13:0]:

Contiene el puntero del primer RPD que ha sido recibido por el paquete.

Dirección actual del DMA:

Guarda la dirección del host de la próxima palabra del buffer actual. El RMAC incrementa este campo este campo en cada acceso en el buffer.

Controlador de DMA

Coordina la recepción 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 inicia las transacciones con el objetivo de dar mantenimiento a sus propias estructuras de datos (descriptores).

Write Data Pipeline/Mux

Encausa los datos recibidos entre el bloque de RHDL y el bloque de GPIC, insertando los retardos necesarios para habilitar las transacciones del controlador de DMA para generar apropiadamente las señales de control a la interfaz del GPIC.

Además provee un multiplexor a las líneas de salida en la interfaz de GPIC, permitiendo al controlador de DMA la salida de datos de transacciones que él mismo ha iniciado.

Caché de información de descriptores

Provee un almacén para la tabla Receive Channel Descriptor Reference (RCDR) que ha sido descrita anteriormente.

Caché de colas libres

Implementa seis elementos de caché para las RPDR small Buffer Free Queue y seis elementos para las RPDR Large Buffer Free Queue. Este caché es utilizado para almacenar “free small buffer “ y “large buffer RPDR”. Guardando en caché estas estructuras reduce el número de accesos al bus del host que el RMAC debe hacer.

Cada caché se maneja independientemente. Los elementos de caché se

consumen de uno en uno como estos sean necesitados por la RMAC.

Se utilizó la versión de 8 enlaces, en lugar de la de 32 enlaces, debido a que el nivel de procesamiento de los datos que se tienen es hasta para 32 canales, y como ambas versiones, presentan 128 canales HDLC, y el PM7366 es la versión que requiere menos inversión económica se decidió hacer el uso de este dispositivo. La información completa de este dispositivo la puede hallar en las hojas de datos que provee PMC-Sierra.

4.2 El microprocesador 80960-RM

Este microprocesador posee dos buses PCI los cuales son compatibles con las especificaciones 2.1 de PCI. Integra un puente PCI-PCI, el cual conecta a los dos buses PCI independientes, ya que poseen diferentes relojes.

Incorpora también un controlador de memoria, capaz de manejar el control de memoria del tipo SDRAM, ROM y Flash. Provee en forma programable los tiempos de espera de la selección de chip. El diagrama de bloques general de este microprocesador se muestra en la figura 4.5.

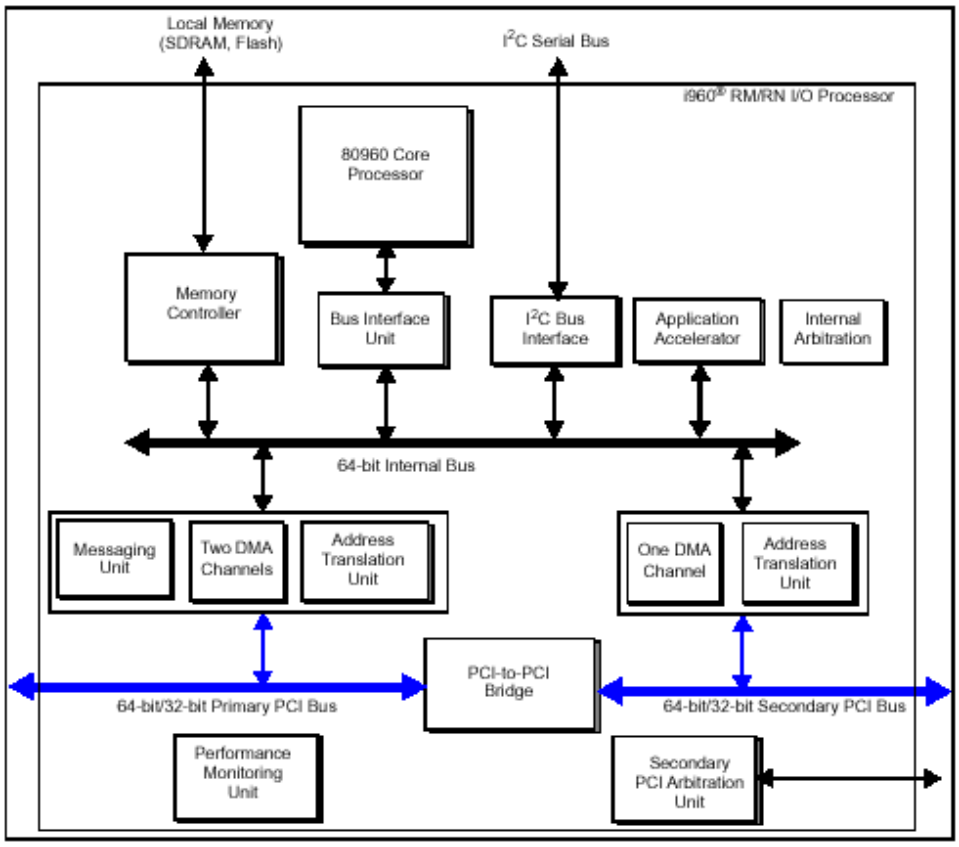


Figura 4.5 Diagrama de bloques del microprocesador 80960RM

Puede manejar datos de los siguientes tipos:

• Bit

• Campos de bits

• Enteros (8-, 16-, 32-, 64-bit)

• Ordinales (8-, 16-, 32-, 64-bit enteros)

• Triple palabra (96 bits)

• Palabra cuádruple (128 bits)

Tiene nueve tipos diferentes de direccionamiento:

- Absoluto
- Absoluto con desplazamiento
- Indirecto por registro
- Indirecto por registro con offset.
- Indirecto por registro con desplazamiento
- Indirecto por registro con índice
- Indirecto por registro con índice y desplazamiento
- Índice con desplazamiento
- Puntero de instrucción con desplazamiento

Para efectos de este informe el componente esencial de este microprocesador se basa en el controlador de memoria, por lo que se presenta una descripción del mismo.

4.2.1 Controlador de memoria

Puede controlar hasta 16 MB de memoria Flash de 8 bits y entre 8 y 128 MB de SDRAM de 64 bits o bien entre 4 y 64 MB de SDRAM de 32 bits.

4.2.1.1 Manejo de memoria Flash

Puede controlar los accesos a uno o dos dispositivos de 8 bits en memoria Flash. Puede realizar lectura en ráfaga hasta de 8 bytes.

Las señales de control para la memoria Flash se muestran en la tabla 4.5.

Tabla 4.5 Señales de interfaz con memoria Flash.

Nombre del pin	Descripción
RCE[1:0]#	Chip Enable. Debe de activarse en todas las transacciones que se realicen sobre el dispositivo Flash.
RWE#	Write Enable. Controla los buffer de entrada de los datos.
ROE	Output Enable. Se activa durante las lecturas, y se desactivan durante las escrituras. Controla las salidas del dispositivo Flash durante las transacciones de escritura.
RAD[16:0]	Bus de direcciones/datos. Puede manejar hasta 16 Mbit de memoria Flash. El bus de datos se multiplexa sobre estas líneas.
RALE	Address Latch Enable. Indica la transferencia de una dirección física. Esta línea se activa durante un ciclo de direcciones para la Flash, y se desactiva durante el inicio de un ciclo de datos.

Un diagrama de bloques de la conexión del microprocesador con una memoria tipo Flash se muestra en la figura 4.6.

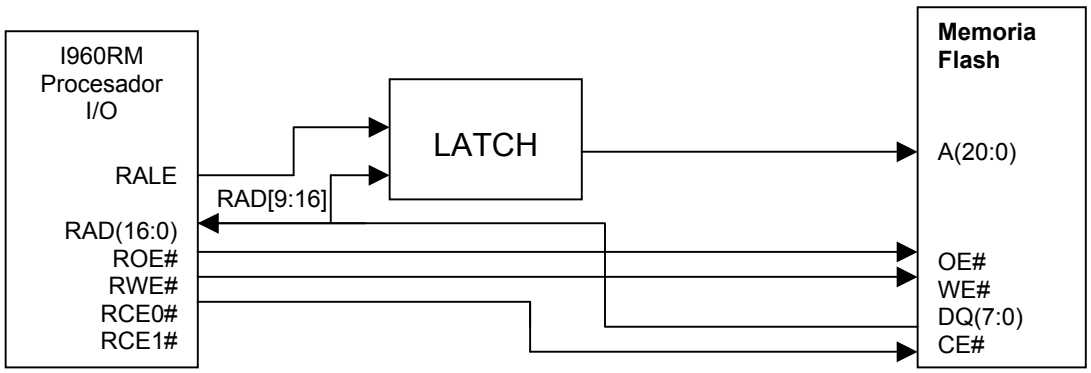


Figura 4.6 Diagrama de bloques de conexión de la memoria Flash

Direccionamiento de la memoria Flash.

Como el bus interno datos es de 64 bits es posible que el bus haga lecturas en ráfaga de hasta 8 bytes. La interfaz de Flash posee el bus de datos y el bus de direcciones multiplexados. El bus de direcciones es de 23 bits, por lo que existen dos ciclos de dirección. Durante la primera fase de direcciones los bits del [22:9] de direcciones se encuentran en las líneas RAD[16:3], por lo que es necesario guardar esta información de dirección utilizando un LATCH de 14 bits y la línea RALE. Durante la segunda fase de dirección, los bits [8:0] se presentan en RAD[8:0]. El bus de datos se encuentra multiplexado en RAD[16:9]. El MCU incrementa los 3 bits de direcciones menos significativos para los siguientes accesos.

Un rango válido para RCE[0]# se define por la dirección de inicio programada en el registro base con la dirección final determinada por el tamaño del módulo programado en el registro dado.

Ciclo de lectura de la Flash

Un ciclo de lectura implica el manejo de direcciones, la habilitación de salida y la habilitación del chip.

La figura 4.7 muestra un ciclo de lectura a un dispositivo Flash.

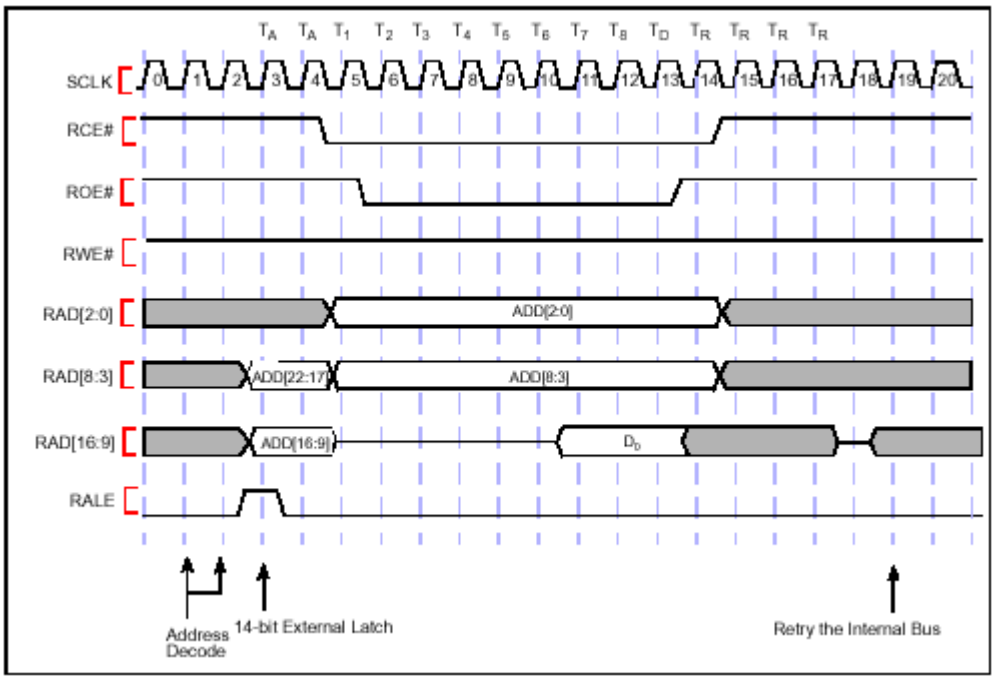


Figura 4.7 Ciclo de lectura a una memoria Flash por el procesador 80960RM.

Cuando se requieren datos de la memoria Flash, el MCU decodifica los byte enables internos para las bits RAD[2:0]. La petición de lectura puede resultar en una lectura múltiple sobre la interfaz de Flash dependiendo de C/BE[7:0]. La máquina de estados de la Flash incrementa las líneas RAD[2:0] para cada lectura. El MCU es el responsable de almacenar adecuadamente los bytes en las líneas correspondientes.

Escritura a Flash

El MCU realiza la petición y la acepta con cero tiempos de espera. No realiza escrituras en ráfaga.

La figura 4.8 muestra un ciclo de escritura a la memoria Flash.

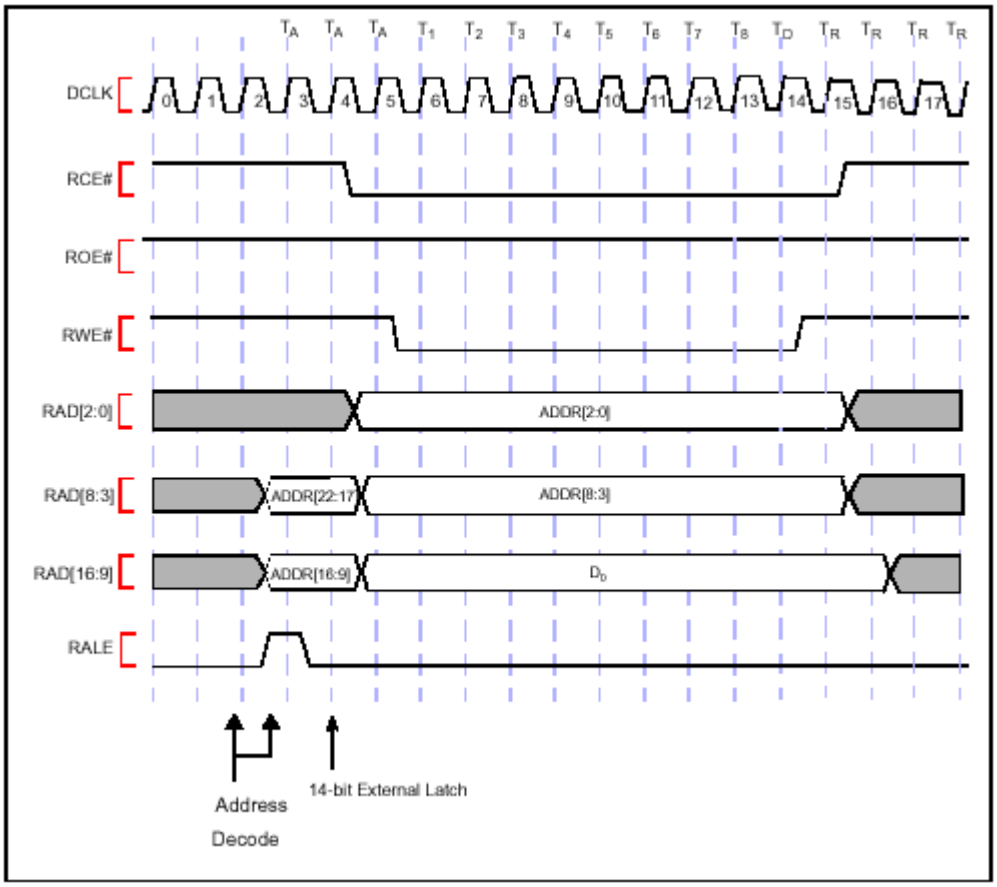


Figura 4.8 Ciclo de escritura a la memoria Flash por el microprocesador 80960RM.

4.2.1.2 Manejo de SDRAM

El 80960RM puede manejar hasta dos bancos de SDRAM.

Este puede realizar transferencia por ráfaga de hasta cuatro, esto siempre y cuando no sobrepase el tamaño de la página. El tamaño de la página para dispositivos de 16Mbit es de 1KB, y el tamaño de la página para un dispositivo de 64 Mbit es de 2KB.

Los pines de interfaz con la SDRAM se muestran en la tabla 4.6.

Tabla 4.6 Pines del microprocesador 80960RM correspondientes al controlador de memoria

Nombre del Pin	Descripción
DCLKOUT	SDRAM Clock Out: Es el reloj de salida para el buffer de reloj para la SDRAM.
DCLKIN	SDRAM Clock In. Es el retorno del reloj del buffer.
SCKE[1:0]	Clock Enables. Se desactiva un ciclo después de SCKE[1:0].
SDQM[7:0]	Data Mask: En una escritura, estas señales deshabilitan el dato para evitar falsas escrituras.
SCE[1:0]#	Chip Select: Se utiliza una selección de chip por banco.
SWE#	Habilitación de escritura: Controla los buffer de entrada de datos.
SBA[1:0]	Selección del banco de SDRAM: Controla cual banco interno es leído o escrito. Para dispositivos de 16 Mbit se utiliza únicamente SBA[0] y para los dispositivos de 64Mbit se utilizan ambas líneas.
SA[10]	Si se encuentra en alto durante una lectura o escritura, una precarga ocurre después del comando. Durante la activación de un renglón forma parte de la dirección.
SA[11:0]	Bits de dirección.
SRAS#	Indica que la dirección presente en estas líneas es un renglón.
SCAS#	Indica que la dirección presente en estas líneas es una columna.
DQ[63:0]	Bus de datos.
SCB[7:0]	Es el bus de datos de ECC.

Utilizando los SDRAM chip enables y las selecciones de banco internos se pueden mantener un máximo de 8 páginas abiertas simultáneamente en dispositivos de 64Mbits.

Tamaño y configuraciones de la SDRAM

Los dispositivos de 16 Mbit son aquellos en los que se comprenden internamente dos bancos. El MCU controla la selección de banco cambiando el bit SBA[0]. Los dispositivos de 64Mbit comprenden aquellos que posee cuatro bancos internos. El MCU controla el cambio de banco realizando los cambios en las líneas SBA[0:1].

Los dos SDRAM chip enable (SCE[1:0]#) se utilizan para el manejo de dos bancos separados de SDRAM.

Modo de 32 bits

Este microprocesador acepta el manejo de memorias con 32 bits en el bus de datos, sin embargo la cantidad de memoria que puede procesar en este modo se reduce a la mitad. Este modo se activa analizando durante el RESET el bit 32BITMEM_EN#.

Determinación de páginas

Para el modo de 64 bits, la MCU asume un tamaño de página de 2Kbyte. En el modo de 32 bits, la MCU asume 1Kbyte en el tamaño de la página.

Para dispositivos de 16Mbit, sólo se puede tener abierta una página por cada banco, y por cada banco interno, lo que implica que puede tener abiertas hasta un máximo de 4 páginas simultáneamente.

Para dispositivos de 64Mbit, se deduce de una explicación similar a la anterior que se pueden tener abiertas hasta 8 páginas simultáneamente.

Para determinar si la página se encuentra abierta o cerrada para la lectura o escritura, el MCU compara la dirección de la transacción actual con la dirección almacenada en el registro de dirección de la página apropiada. Los habilitadores de chip y los seleccionadores de banco determinan cual es la página con la que la dirección debe ser comparada.

Si la transacción presente no encuentra abierta la página seleccionada, cierra la que se encuentra apuntada por SCE[1:0]# y SBA[1:0] utilizando un comando de precarga. La MCU abre la página actual con la activación del renglón y la transacción culmina con la lectura o la escritura. Cuando la página es abierta la dirección se guarda en el registro apuntado por SCE[1:0] y SBA[1:0] para que este pueda ser comparado en futuras transacciones.

Si encuentra la página abierta, entonces, sin la necesidad de realizar una activación de renglón la lectura o escritura es ejecutada. Si el timer de refrescamiento expira y el MCU realiza un autorefrescamiento, todas las páginas son cerradas.

Ciclo de lectura de la SDRAM

Lectura de en una página abierta

Esto ocurre cuando una dirección cae dentro de un renglón que se encuentra actualmente activo. Si esto sucede, el MCU no necesita activar la señal de SRAS#. Este caso se muestra en la figura 4.9.

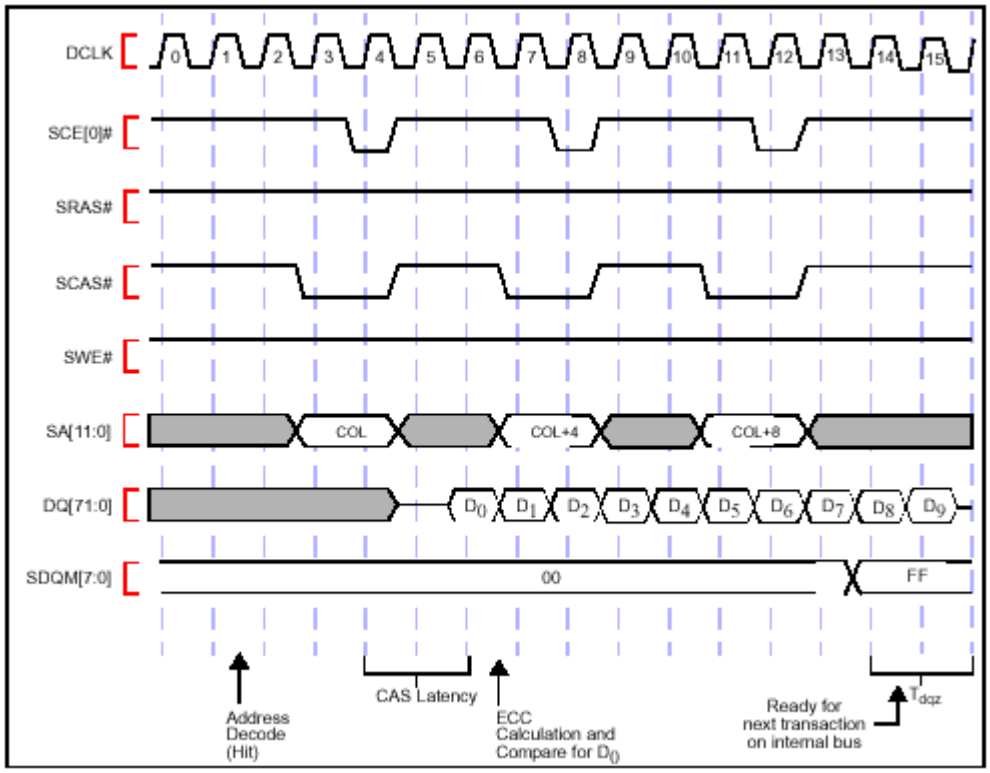


Figura 4.9 Diagrama de lectura de la SDRAM cuando encuentra la página abierta

Lectura de una página que se encuentra cerrada.

Esto acarrea un retrazo en tiempo, ya que se debe de cerrar la página que actualmente se encuentra abierta y abrir la que se necesita, esto es activando la señal SRAS#. La figura 4.10 muestra este caso.

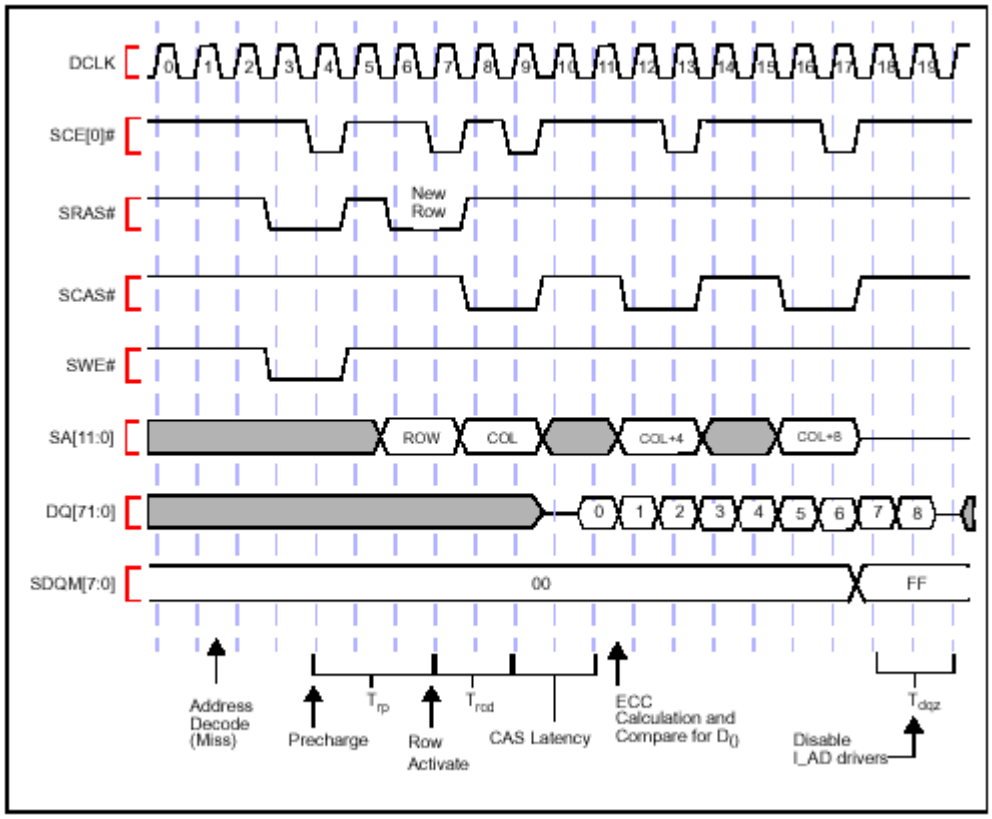


Figura 4.10 Lectura de la SDRAM cuando encuentra la página cerrada.

Los ciclos de escritura son muy similares.

Ciclo de refrescamiento de la SDRAM

Como la SDRAM es una memoria dinámica necesita que se refresque periódicamente. El intervalo de este ciclo de refrescamiento es programable en el registro RFR. La SDRAM genera el ciclo de refrescamiento internamente. El MCU inicia dos ciclos de refrescamiento secuenciales (uno por banco) después de que el temporizador de refrescamiento expira y no se encuentra realizando ninguna transacción. La figura 4.11 muestra el diagrama de tiempo en el caso en que el temporizador de refrescamiento expire en medio de una transacción incompleta.

Una vez que el temporizador expira, el MCU sabe que es necesario un ciclo de refrescamiento. El temporizador de refrescamiento continúa la cuenta para el próximo ciclo de refrescamiento.

El MCU espera a que la última transacción termine. Mientras esto ocurre el ciclo de refrescamiento se guarda en una cola mientras se termina la transacción.

La MCU cierra todas las páginas que se encuentren abiertas mediante un comando en todos los bancos que se encuentren activos.

Se realiza el refrescamiento, y la memoria queda lista para la próxima transacción o ciclo de refrescamiento.

El registro RFR se programa según con la frecuencia con que esté trabajando internamente el microprocesador. Si el bus primario es de 33MHz, como es el caso de la tarjeta HDLC-4M este se debe programar con un 400H.

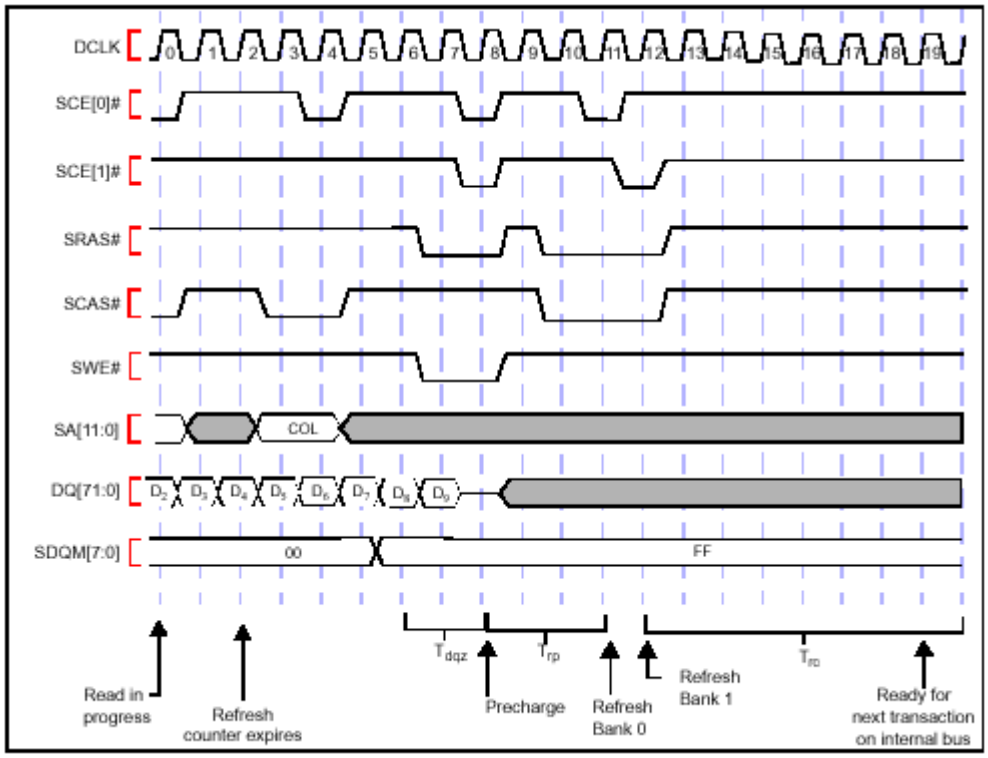


Figura 4.11 Ciclo de refrescamiento de la memoria SDRAM

4.3 Unidad puente PCI-PCI del procesador 80960-RM

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 la Arquitectura de 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) en ambos buses.
- Operación independiente de los buses permite realizar tareas concurrentes en ambas direcciones.
- Múltiples operaciones *Memory Write* y *Memory Write and Invelidate* 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 secundario del puente.
- Modo de direccionamiento dual (64 bits) permitido en ciclos iniciados desde el secundario.
- Espacio de dirección y configuración privado, para dispositivos marcados como tales en el bus PCI secundario.

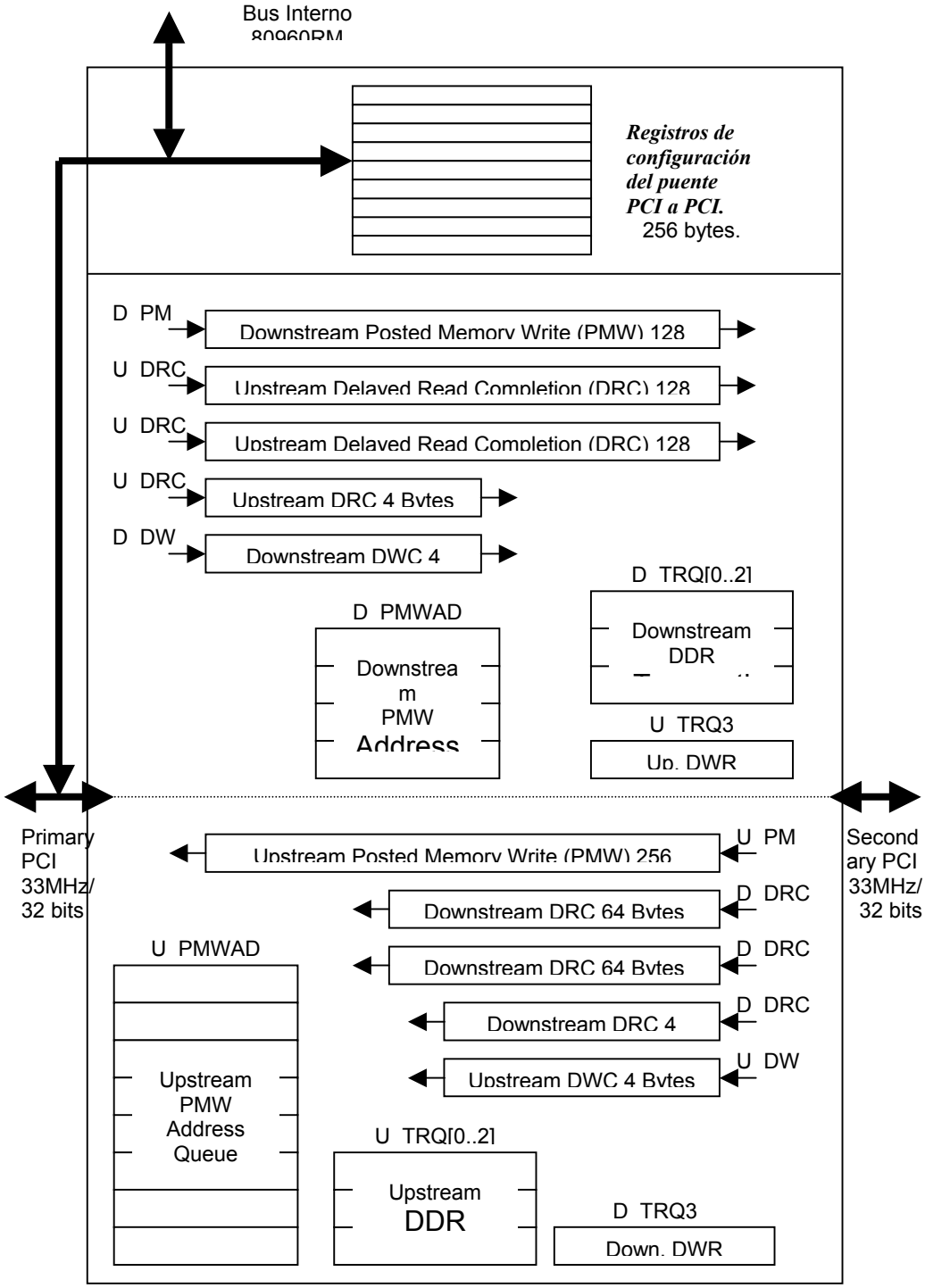


Figura 4.12 Diagrama de bloques del circuito puente integrado

Teoría de operación

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.

Descripción de la Arquitectura

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.

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.

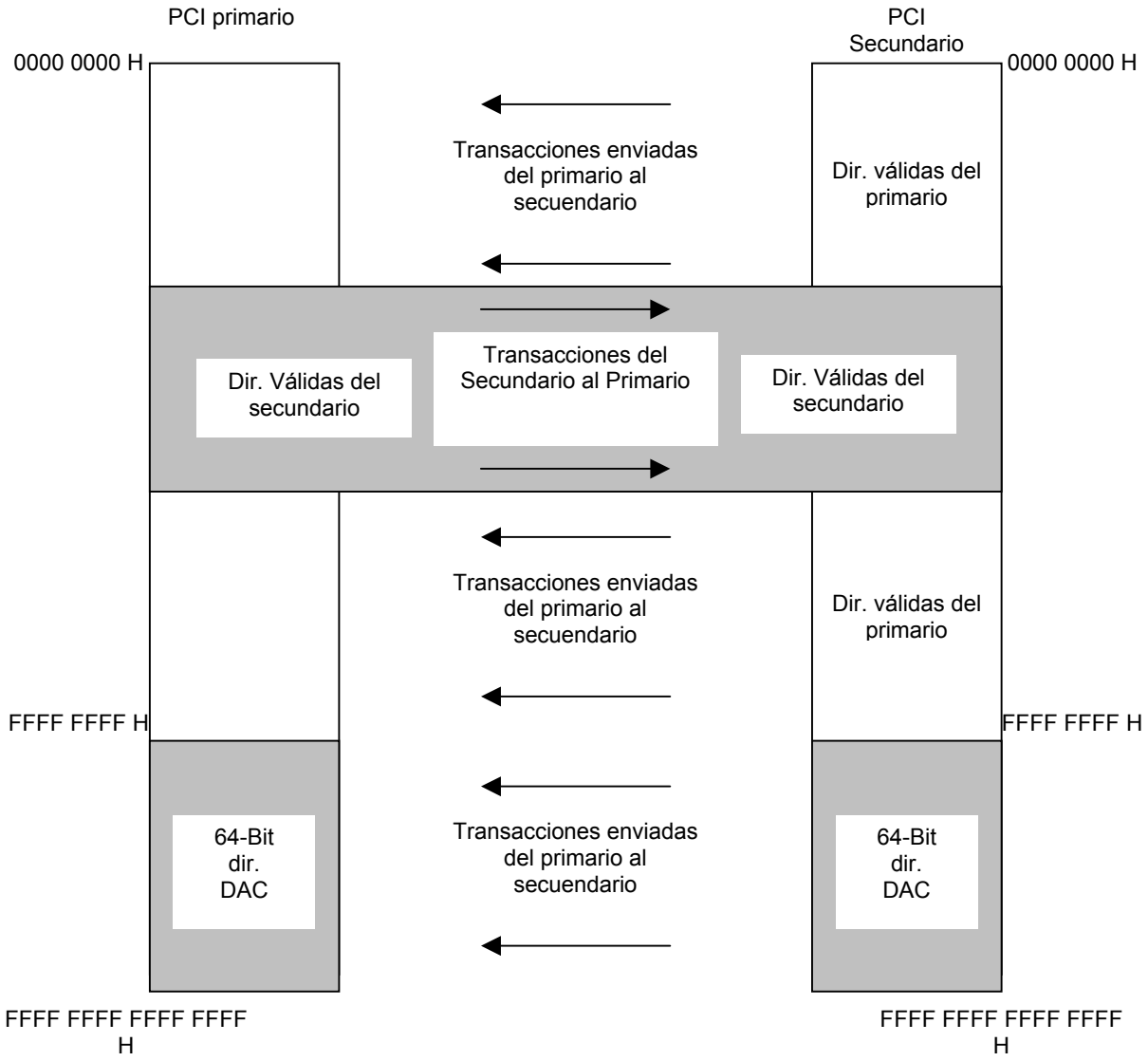


Figura 4.13 Transacciones de datos

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 $_{H}$.

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.

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.

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]

Tabla 4.7 Comandos a los registros de los buses PCI

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)

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.

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.

4.4 El módulo de memoria SDRAM

Se utilizó el módulo MT4LSDT464AG-10EC6. Es un dispositivo CMOS de alta velocidad, con acceso aleatorio dinámico, y con una capacidad de 4M de palabras de 64 bits. Posee internamente cuatro módulos configurados en cuatro bancos de SDRAM con una interfaz sincrónica. Permite realizar operaciones de ráfaga. La lectura y escritura en este módulo está orientada a ráfaga, accesando la posición de inicio y continuando en una secuencia determinada. Son diseñados para trabajar con 3.3 V, y provee un auto refrescamiento. El diagrama de bloques de este sistema se muestra en la figura 4.12.

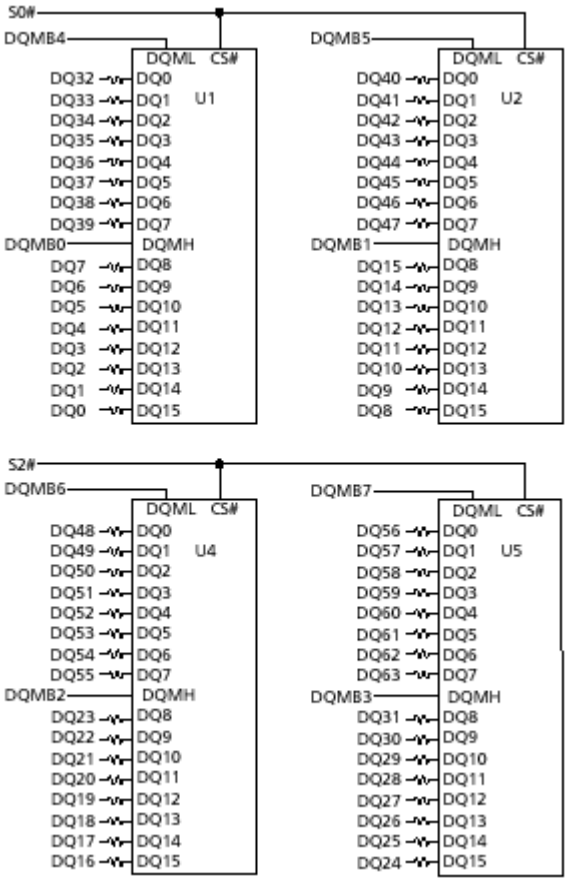


Figura 4.12 Diagrama de bloques del módulo de memoria a utilizar.

4.5 El modulo de memoria EPROM

Se utilizará el modulo M27C2001 de microchip, el cual posee un voltaje fuente de 3.3 V, un consumo de 8 mA, y un tiempo de acceso de 200ns. Se utilizará el empaquetado de PLCC para disminuir el espacio físico, por lo que para la programación se necesitará un adaptador de PLCC a DIP.

4.6 El comparador LMV311

Es un circuito comparador de bajo consumo de potencia, que puede posee un voltaje de entrada en el rango de 3-5V. Posee un retraso de 1000ns, sin embargo, como la comparación que realiza es solo al inicio de la conexión el ancho de banda del mismo no afecta en el circuito. Para evitar que este se sature se seleccionó

una alimentación de 5V.

4.7 El buffer de reloj CY2305

Es el buffer de reloj que necesitan los componentes de la tarjeta para poder funcionar. Se toma el reloj proveniente del bus PCI de la tarjeta, se pasa por este buffer, el cual reparte esta señal en cinco salidas. Es un dispositivo de bajo consumo de potencia, que se maneja a 3.3 V.

Posee un delay entre la entrada y las salidas Para asegurar que el desfase entre las señales de salida sea cero, se debe de asegurar que las capacitancias de carga en todos los relojes debe de ser la misma. El diagrama de bloques de este dispositivo se muestra en la figura 4.13.

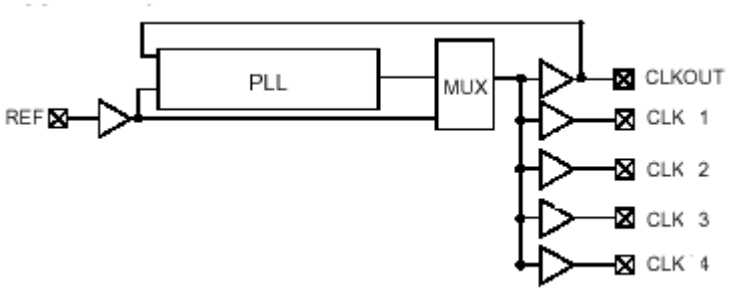


Figura 4.13 Diagrama de bloques del buffer de reloj

4.8 El latch de direcciones 74LCX841

Se trata de un buffer cuya función es la de poder manejar altas cargas capacitivas o bien bajas impedancias, o para manejarse como driver de buses bidireccionales.

4.9 Los registros universales 74AL323

Es un registro el cual puede ser cargado en forma serie o paralelo, según se seleccione en sus líneas de control. El desplazamiento puede ser hacia la derecha o a la izquierda según se establezca, y se pueden leer continuamente las posiciones de los 8 bits que se encuentran en el registro. Para más información leer las hojas

de datos del dispositivo en el TTL Handbook.

4.10 **Las PAL's** TIBPAL22V10-10C

Es un dispositivo lógico programable el cual se puede programar únicamente una vez. Posee 21 entradas y 10 salidas. Las salidas pueden ser registros o combinacionales. Se maneja a 5V, y puede ser programada con el software ABEL.

4.11 Bus PCI

El bus PCI es diseñado para manejar más aplicaciones que los buses existentes. Posee un campo datos de 32 bits, con un reloj de 33 MHz y una tasa máxima de transferencia de datos de 132 MB/sec. Existe un modelo a 33 MHz con un bus de datos de 64 bits. El estándar de 64 bits de datos a 66 MHz existe, pero no se encuentra disponible.

Como trabaja PCI

PCI no es un bus local real, sin embargo se encuentra entre el nivel de bus local (procesador/memoria/ y la caché del sistema)y la tarjeta estándar de expansión ISA.. El bus PCI se encuentra aislado del CPU por medio de un puente/controlador PCI. El CPU puede escribir datos en los periféricos del PCI, como el disco duro, y el controlador de PCI puede almacenar inmediatamente los datos en el buffer. Esto permite al CPU realizar la siguiente operación sin tener que esperar que se efectúe la transferencia completa. El buffer realizará la transferencia de los datos a los periféricos en una tasa de transferencia lo más eficientemente posible.

PCI posee busmasters, dispositivos inteligentes que cuando se anexan al sistema de bus, pueden tomar el control del bus y realizar las tareas independientemente del CPU. El CPU puede seguir corriendo en concurrencia con los periféricos del busmaster. El puede seguir operando independientemente cuando un periférico del PCI es activo por su diseño buferizado.

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

fases de datos pueden ser repetidas indefinidamente, pero están limitadas por la base de tiempo que define el tiempo máximo que el dispositivo PCI puede tomar el control del bus. Este tiempo es definido por el CPU en la fase de configuración. Las mismas líneas que se utilizan para datos se utilizan para direcciones (multiplexados)

Cada dispositivo tiene su tiempo asignado. Las líneas de comando son sólo utilizadas por las líneas de byte habilitador. Esto reduce el número de pines en el conector de PCI. Las líneas de comando (C/BE3 a C/BE0) indican el tipo de transferencia a realizarse durante la fase de dirección. La tabla 4.7 muestra la combinación de estos bits para lograr los comandos dados.

Tabla 4.8 Combinación de los bits C/BE [3:0] con el comando respectivo en el bus PCI.

C/BE [3:0]	Comando
0000	Reconocimiento de Interrupción
0001	Ciclo Especial
0010	Lectura de E/S
0011	Escritura de E/S
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 Dual de Dirección
1110	Lectura de una línea de memoria
1111	Escritura a memoria e invalidación

Los tres tipos básicos de transferencias son E/S, memoria y configuración. El ciclo de transferencia comienza cuando FRAME# es activado. Durante la fase de dirección AD[31:0] contienen una dirección válida y C/BE#[3:0] contienen un

comando válido de bus. Al iniciar la fase de datos, C/BE#[3:0] indican cuales bytes están envueltos en la fase de datos actual. Las líneas de IRDY# indica que el iniciador está listo para la transacción, mientras la línea TRDY# indica que el target está listo para la transacción. Un ciclo de transferencia se muestra en la figura 4.14

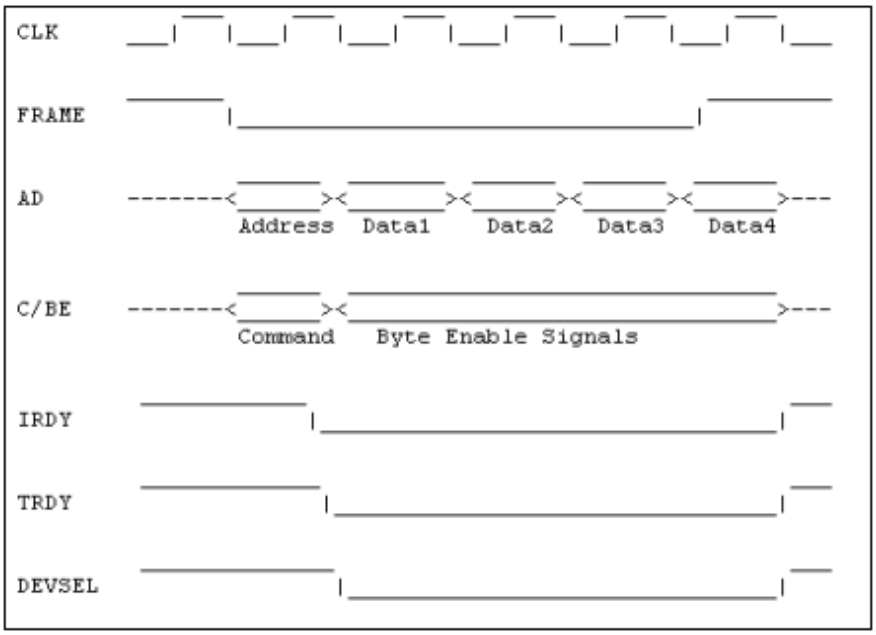


Figura 4.14 Ciclo de transferencia en un bus PCI.

Ciclos de bus

Reconocimiento de interrupción (0000):

El controlador de interrupción automáticamente reconoce y reacciona con un comando de INTA (interrupt acknowledge). En la fase de datos se transfiere el vector de interrupción en las líneas AD.

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 trasladación 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 como se muestra en la tabla 4.8.

Tabla 4.9 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

I/O Read (0010) e I/O Write (0011):

Escritura o lectura en un periférico:

Las líneas AD contienen un byte de dirección (AD0 y AD1 deben estar decodificadas) . Los puertos PCI I/O deben ser de 8 o 16 bits. PCI contiene un espacio de 32 bits.

Lectura de memoria (0110) y escritura a memoria (0111):

Una lectura o escritura a memoria. Las líneas AD contienen una doble palabra de dirección. AD0 y AD1 no necesitan estar decodificadas. Las líneas de Byte Enable (C/BE) indican cuales bytes son validos.

Lectura de configuración (1010) y escritura de configuración (1011) :

Una lectura o escritura al espacio de configuración del dispositivo PCI, el cual

posee una capacidad de 256 bytes. Es accesado en dobles palabras. AD0 y AD1 contienen 0, AD2-7 contiene las dobles palabras de dirección. AD8-10 son utilizadas para seleccionar la unidad de direccionamiento de las unidades de mal funcionamiento y las otras líneas no son utilizadas.

Lectura de múltiples memorias (1100):

Esta es una extensión del ciclo de bus de lectura de memoria. Es utilizado para leer bloques de memoria largos sin caché, lo cual beneficia para memorias grandes de acceso secuencial.

Ciclo de dirección dual (1101):

Dos ciclos de dirección son necesarios cuando se usa una dirección de 64 bits , pero sólo existen 32 bits de dirección física. La porción menos significativa de la dirección es puesta en las líneas AD, seguidos de los 32 bits mas significativos. En el segundo ciclo de dirección contiene el comando para el tipo de transferencia (I/O, memoria, etc.). El bus PCI provee un espacio de 64 bits de I/O

Memory read line (1110):

Este ciclo es utilizado para en bloque de más de 2 bloques de datos de 32 bits. Típicamente arriba del final de la línea de caché.

Escritura a memoria e invalidación (1111):

Esta indica que el mínimo de una línea de caché está por ser transferida.

Capítulo 5
Descripción del Software del Sistema

5.1 Software de la HDLC-4M

En este caso se realizó la inicialización del componente PM7366 y de la interfaz a memoria por parte del microprocesador. La figura 5.1 muestra el mapa de memoria del microprocesador 80960RM para la aplicación que se está desarrollando. Este mapa muestra la localización tanto de la memoria de programa como la de datos cuya posición es fundamental para la ubicación de las estructuras que conforman el PM7366.

Los registros del PM7366, así como sus estructuras, ocupan lugar dentro de la memoria SDRAM, sin embargo como las direcciones que maneja el FREEDM-8 se le debe sumar un OFFSET, es necesario ubicar primero este OFFSET en el registro correspondiente del FREEDM-8.

Para ubicar la cantidad de memoria necesaria se escribe todos unos en el registro de base, y luego se lee de él. Analizando desde el bit menos significativo en adelante se puede determinar la cantidad de espacio de dirección que se necesita para trabajar.

Sin embargo, es posible realizar un mapeo de los espacios necesarios para el funcionamiento del dispositivo basándose en la ubicación de la memoria SDRAM del microprocesador.

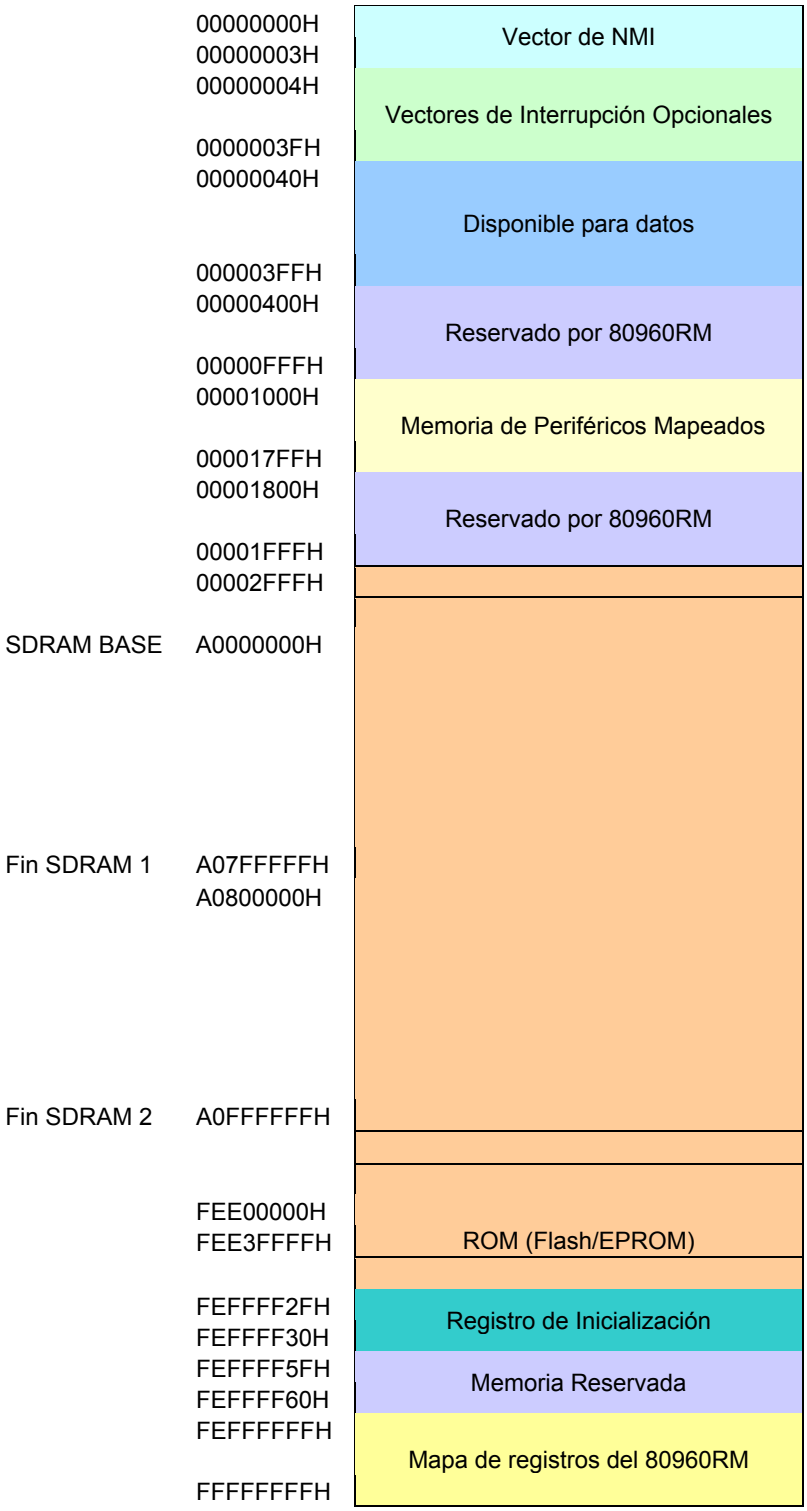


Figura 5.1 Mapa de memoria del microprocesador utilizado

Este mapa se logra programando los registros de la sección del controlador de memoria tal y como se muestran en la tabla 5.1.

Tabla 5.1 Valores programados en los registros del controlador de memoria para realizar el mapa de memoria de la figura 5.1 con la memoria SDRAM y EPROM seleccionadas.

Nombre del Registro	Default	Programado	Abreviatura
SDRAM Initialization Register	0x07H	0x06H	SDIR
SDRAM Control Register	0x00H	0x00H?	SDCR
SDRAM Base Register	0x00H	A0000000H	SDBR
SDRAM Boundary Register 0	0x00H	10x04	SBR0
SDRAM Boundary Register 1	0x00H	10x08	SBR1
ECC Control Register	0x00H	0x00H	ECCR
ECC Log Registers	0x00H		ELOG0,1
ECC Address Register	0x00H		ECAR0,1
ECC Test Register	0x00H		ECTST
Flash Base Register 0	FE80x0H	FEE00000H	FEBR0,
Flash Base Register 1	0x00H		
Flash Bank Size Register 0	0x08H	0x03H	FBSR0
Flash Bank Size Register 1	0x00H	0x00H	FBSR1
Flash Wait States Registers	0x77H	0x77H	FWSR0,1
Memory Controller Int. St.Reg	0x00H		MCISR
Refresh Frecuency Reg.	0x300H	0x400H	RFR

Las estructuras que ocupan mapearse son: los registros accesibles por la computadora huésped, los registros de configuración de PCI, las tablas de descriptores tanto de recepción como de transmisión y las colas utilizadas para una eficiente transferencia de datos con la memoria.

Para definir el tamaño de las tablas de descriptores tanto de recepción como de transmisión es necesario saber el número de referencias a utilizarse, recordando que estas referencias son utilizadas para realizar las listas enlazadas de los bloques que conforman las FIFO's internas de cada canal. Se ha decidió asignarle un total de 4 bloques de 16 bytes a cada canal, lo que quiere decir que se tendrá una FIFO 64 bytes interna para cada canal utilizado y como por cada bloque asignado se

sobreentienden 3 referencias, estas se deben ir multiplicando en proporción al número de bloques agregado. Siguiendo la forma de calculo de a tabla de descriptores que se muestra en la notas de desarrollo de software estas deben tener un tamaño de 6KB, sin embargo se data un margen de 2KB y se agregarán 8KB.

El buffer de datos es donde se almacenan los datos recibidos en de los streams de recepción, o bien donde se toman los datos para ser transmitidos. Se ha decido tomar este espacio de memoria del máximo tamaño que este puede tener para aprovechar la capacidad de almacenado en la mayor forma posible.

Al reunir toda esta información se obtiene el siguiente mapa de memoria:

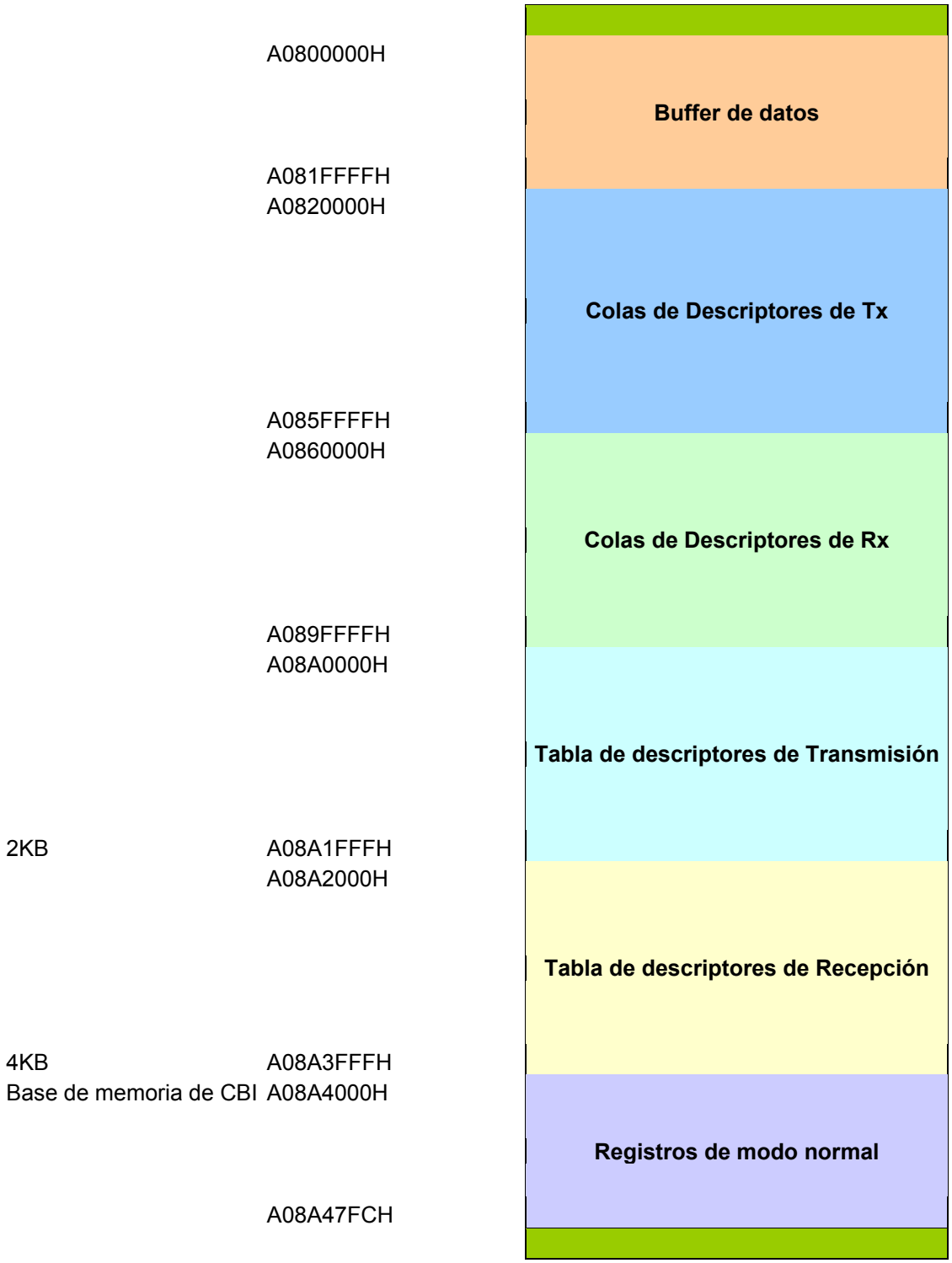


Figura 5.2 Mapa de memoria para el FREEDM-8

Para la realización de este mapa únicamente resta programar los registros del PM7366 necesarios, tal y como se muestra en la siguiente tabla.

Tabla 5.2 Registros del FREEDM-8 programados para definir el mapa de memoria anterior.

Dirección (OFFSET PCI)	Nombre del Registro	Valor luego del Reset	Valor programado
0x288	RMAC Packet Descriptor Table Base LSW	0	2000
0x28C	RMAC Packet Descriptor Table Base MSW	0	A08A
0x308	TMAC Transmit Descriptor Table Base LSW	0	0000
0x30C	TMAC Transmit Descriptor Table Base MSW	0	A08A
0x290	RMAC Receive Queue Base LSW	0	0000
0x294	RMAC Receive Queue Base MSW	0	A086
0x310	TMAC Transmit Queue Base LSW	0	0000
0x314	TMAC Transmit Queue Base MSW	0	A082
0x298	RMAC Packet Descriptor Reference Large Buffer Free Queue Start	0	A0860000H
0x29C	RMAC Packet Descriptor Reference Large Buffer Free Queue Write	0	A860004H
0x2A0	RMAC Packet Descriptor Reference Large Buffer Free Queue Read	0	A0860008H
0x2A4	RMAC Packet Descriptor Reference Large Buffer Free Queue End	0	A0860208H

Continuación tabla 5.2.

0x2A8	RMAC Packet Descriptor Reference Small Buffer Free Queue Start	0	A086020CH
0X2AC	RMAC Packet Descriptor Reference Small Buffer Free Queue Write	0	A0860210H
0x2B0	RMAC Packet Descriptor Reference Small Buffer Free Queue Read	0	A0860214H
0x2B4	RMAC Packet Descriptor Reference Small Buffer Free Queue End	0	A0860418H
0x2B8	RMAC Packet Descriptor Reference Ready Queue Start	0	A086041CH
0x2BC	RMAC Packet Descriptor Reference Ready Queue Write	0	A0860420H
0x2C0	RMAC Packet Descriptor Reference Ready Queue Read	0	A0860420H
0x2C4	RMAC Packet Descriptor Reference Ready Queue End	0	A0860628H
0x318	TMAC Transmit Descriptor Refence Free Queue Start	0	A0820000H
0x31C	TMAC Transmit Descriptor Refence Free Queue Write	0	A0820004H
0x320	TMAC Transmit Descriptor Refence Free Queue Read	0	A0820008H
0x324	TMAC Transmit Descriptor Refence Free Queue End	0	A0820204H
0x328	TMAC Transmit Descriptor Refence Ready Queue Start	0	A0820208H
0x32C	TMAC Transmit Descriptor Refence Ready Queue Write	0	A082020CH
0x330	TMAC Transmit Descriptor Refence Ready Queue Read	0	A0820210H

Continuación tabla 5.2.

0x334	TMAC Transmit Descriptor Refence Ready Queue End	0	A08206B4H

Se debe observar que el FREEDM-8 maneja cinco colas, dos para los descriptores de recepción y tres para los descriptores de transmisión. Las funciones de cada una de las colas se encuentran descritas en el capítulo de descripción del Hardware utilizado.

El tamaño de cada una de las colas debe de responder a un número de palabras dobles iguales a la cantidad de referencias existentes más una palabra doble, esto quiere decir que se utiliza el número máximo de referencias igual a 128, entonces se deberán utilizar 129 palabras dobles para cada una de las colas. El tamaño de cada una de las colas se define por medio de los punteros de inicio y de fin de cola. En la estructura de la tarjeta HDLC-4M estas se definen como se muestra en la figura 5.3.

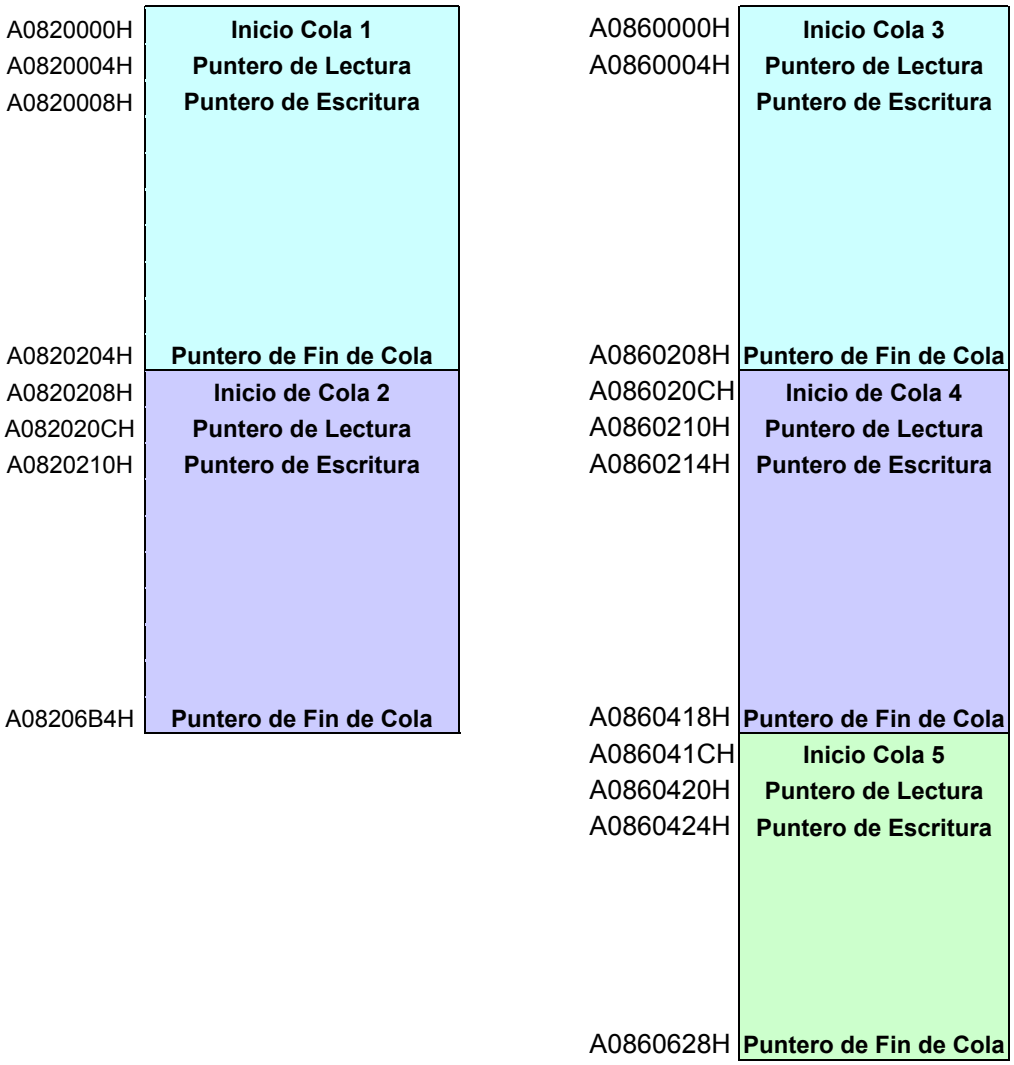


Figura 5.3 Localización de las cinco colas del FREEDM-8 en el mapa de memoria.

En este punto se encuentran definidas e inicializadas las estructuras necesarias para la eficiente comunicación entre el FREEDM-8 y la memoria utilizando el controlador de memoria del microprocesador 80960 RM.

Es necesario realizar la inicialización de todos los enlaces del controlador de protocolo HDLC para que estos puedan trabajar con señalización del tipo E1. Para esto es necesario realizarla programación de los registros creados en memoria RAM para el FREEDM-8. Estos registros son los que se muestran a continuación el valor en hexadecimal necesario para el funcionamiento deseado.

Tabla 5.3 Valores programados para la inicialización de los enlaces el PM7366 para un efectivo funcionamiento con señalización E1.

Dirección	Nombre del Registro	Nombre del espacio de bits	Valor programado
0x180-0x19C	RCAS Link #n Configuration	CEN	1
0x180-0x19C	RCAS Link #n Configuration	E1	1
0x180-0x19C	RCAS Link #n Configuration	BSYNC	X
0x108	RCAS Framing Bit Threshold	FTHRES[6:0]	0x1F
0x480-0x49C	TCAS Link #n Configuration	CEN	1
0x480-0x49C	TCAS Link #n Configuration	E1	1
0x480-0x49C	TCAS Link #n Configuration	BSYNC	X
0x408	TCAS Framing Bit Threshold	FTHRES[6:0]	0x1F
0x40C	TCAS Idle Time-slot Fill Data	FDATA[7:0]	0xFF

Para escribir los datos recibidos en la memoria se utiliza una transacción por DMA y luego esto se transforma en una transacción a través del bus PCI. Para un efectivo funcionamiento del bloque de DMA es necesario programar el registro RMAC Control ubicado en la dirección 0x280. El valor programado afectará a todos los canales de recepción. Por defecto el bloque de DMA para la recepción se encuentra deshabilitado, por tanto se debe de habilitar si se quiere que se

produzcan transacciones entre los bloques de memoria internar y el host. Por tanto, como se muestra en la tabla 5.4 los espacios de bits deben ser programados de la siguiente manera:

Tabla 5.4 Espacios de bits en el registro RMAC Control

Nombre del bit	Valor a programarse
ENABLE	1
LCACHE	1
SCACHE	1
RAWMAX[1:0]	00
RPQ_RDY n[2:0]	100
RPQ_LFN	10
RPQ_SFN	10

En cuanto a la programación del bloque de transmisión este por defecto se encuentra deshabilitado, por tanto se debe habilitar y preparar para el uso efectivo de las colas y estructuras que han sido creadas para la comunicación. El registro TMAC Control ubicado en la dirección 0x300 se debe de programar de la siguiente manera:

Tabla 5.5 Programación del registro TMAC Control

Nombre del Bit	Valor a Programarse
ENABLE	1
CACHE	1
TDQ_FRN[1:0]	10

Es necesario también configurar la unidad de control del bus PCI. Para este propósito debe de programarse el registro GPIC Control ubicado en la posición 0x040H, y cuyos espacios de bits deben ser programados como se muestra en la tabla 5.6.

Tabla 5.6 Programación del registro GPIC Control

Nombre del espacio de bits	Valor a programar
Reservado	0
LENDIAN	0
SOE_E	1
PONS_E	1
RPWTH[4:0]	1

La configuración de los canales de HDLC y de las FIFO se realizan en los registros de control de los bloques de RHDL y THDL. En el caso del bloque de RHDL se programa tal y como muestra la tabla 5.7. Este registro corresponde al RHDL Configuration que se localiza en la dirección 0x224.

Tabla 5.7 Configuración del bloque de Recepción

Nombre del espacio de bits	Valor programado
MAX[15:0]	0x200H
LENABRT	1
TSTD	1
Reservado	0x7

Para configurar la sección de transmisión se debe de programar el registro THDL Configuration localizado en la dirección 0x3B0. Los valores programados afectan a todos los canales de transmisión. Estos valores se pueden encontrar en la tabla 5.8.

Tabla 5.8 Configuración del bloque de transmisión THDL.

Nombre del espacio de bits	Valor Programado
BURST[2:0]	0X03
BURSTEN	1
TSTD	1
BIT8	0

FIFO de canales de canales de recepción

Para programar una FIFO de recepción se debe de repetir el siguiente procedimiento para cada uno de los bloques que conforman la lista circular enlazada.

Realizar un polling del bit BUSY en el registro RHDL Indirect Block Select (0x210) hasta que este sea cero. Esto previene que un acceso previo a la RAM no se complete.

1. Una vez cumplida la condición anterior escriba en el registro RHDL Indirect Block Data (0x214) el siguiente block en la lista enlazada, o termine el proceso si ya todos los bloques han sido programados.
2. Especifique el bloque y actualice el puntero interno de block escribiendo en el registro RHDL Indirect Block Select (0x210).

FIFO de canales de transmisión

Un canal de transmisión es programado repitiendo este procedimiento para cada bloque que conforma una lista enlazada.

1. Realice un polling del bit BUSY en el registro THDL Indirect Block Select (0x3A0) hasta que este se coloque en cero.
2. Escriba en el registro THDL Indirect Block Data (0x3A4) el próximo bloque en la lista enlazada.
3. Especifique el número de block y actualice el puntero de RAM interna escribiendo en el registro THDL Indirect Block Select.

Configuración de los canales RHDL

Los canales pueden ser configurados de diferentes maneras utilizando los registros RHDL Indirect Channel Data Register #1 y #2. En este caso estos se programaron de la siguiente manera:

Tabla 5.9 Configuración de los canales RHDL para la tarjeta HDLC-4M

Nombre del espacio de bits	Nombre del Registro	Dirección del registro	Valor programado
DELIN	RHDL Indirect Channel Data Register #1	0x204	1
STRIP	RHDL Indirect Channel Data Register #1	0x204	1
CRC[1:0]	RHDL Indirect Channel Data Register #1	0x204	01B
XFER[2:0]	RHDL Indirect Channel Data Register #2	0x208	0x07H
OFFSET1:0]	RHDL Indirect Channel Data Register #2	0x208	00
INVERT	RHDL Indirect Channel Data Register #2	0x208	1
PRIORITY	RHDL Indirect Channel Data Register #2	0x208	0
7BIT	RHDL Indirect Channel Data Register #2	0x208	0

Configuración de los canales de transferencia de datos

Al igual que los canales de recepción, los canales de transmisión son configurables en la forma de tratar los streams de entrada utilizando los siguientes registros:

Tabla 5.10 Registros a programar para configurar los canales de transmisión

Nombre del espacio de bits	Nombre del registro	Dirección del Registro	Valor Programado
DELIN	THDL Indirect Channel Data Register #1	0x384	1
IDLE	THDL Indirect Channel Data Register #1	0x384	0
CRC[1:0]	THDL Indirect Channel Data Register #1	0x384	01B
FLEN[8:0]	THDL Indirect Channel Data Register #2	0x388	0x04
DFCS	THDL Indirect Channel Data Register #2	0x388	0
INVERT	THDL Indirect Channel Data Register #2	0x388	1
PRIORITYB	THDL Indirect Channel Data Register #2	0x388	0
7BIT	THDL Indirect Channel Data Register #2	0x388	1
XFER[2:0]	THDL Indirect Channel Data Register #3	0x38C	0x07
FLAG[2:0]	THDL Indirect Channel Data Register #3	0x38C	001
LEVEL[3:0]	THDL Indirect Channel Data Register #3	0x38C	0000
TRANS	THDL Indirect Channel Data Register #3	0x38C	1

Los diagramas de flujo de estas inicializaciones se pueden observar en las siguientes figuras.

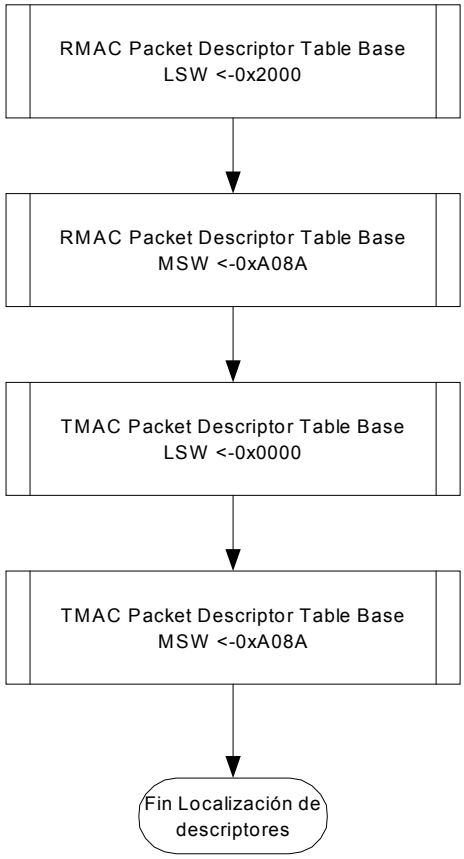


Figura 5.4 Establecimiento en memoria de las direcciones bases para los descriptores

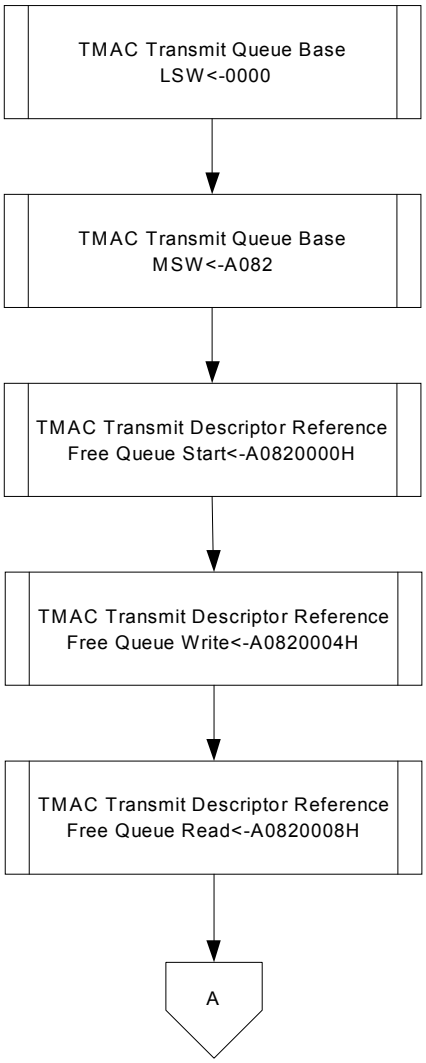
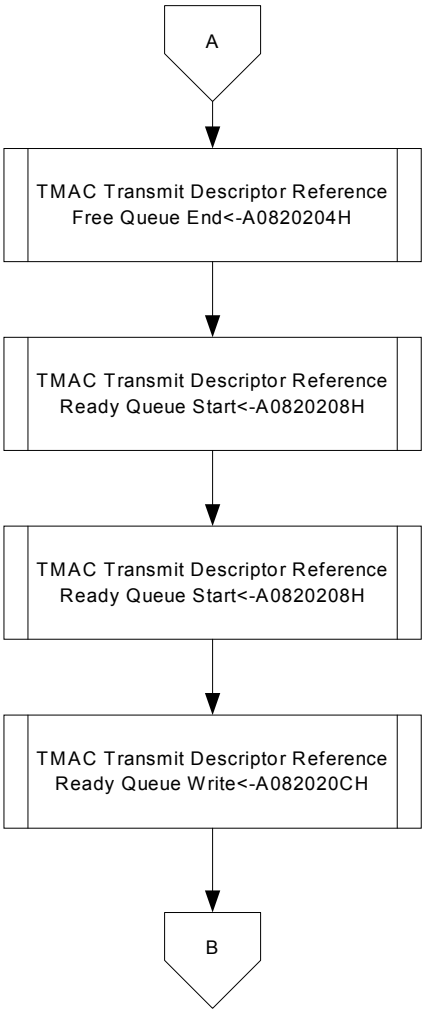
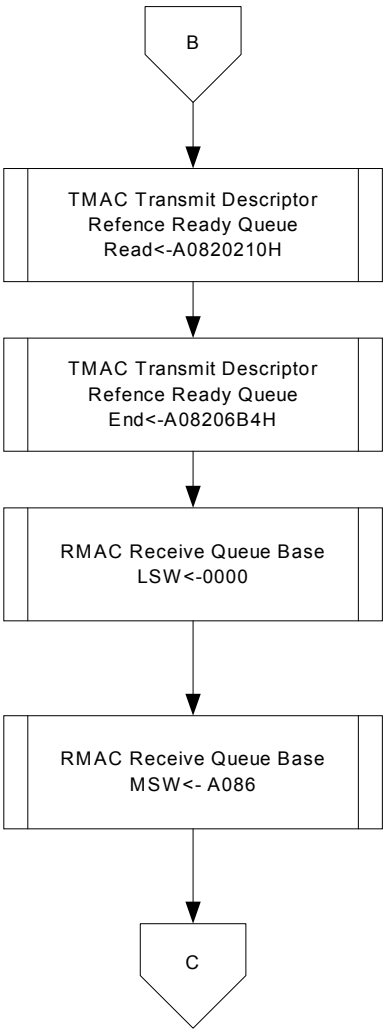


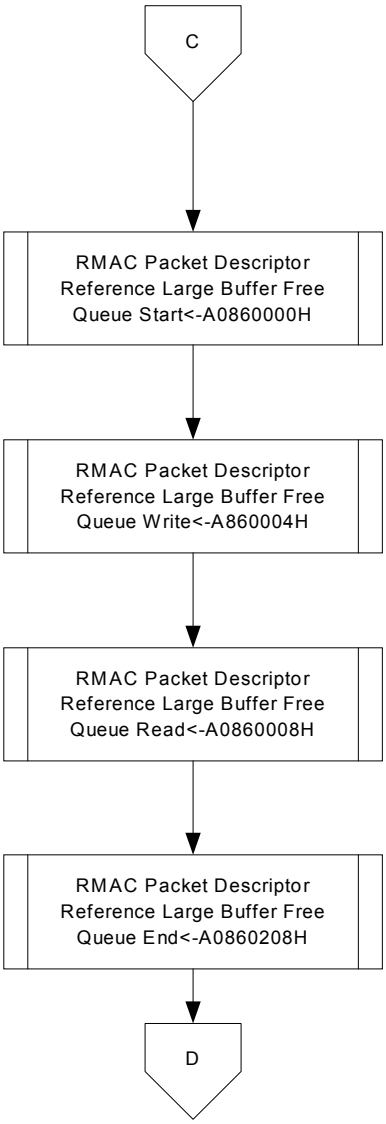
Figura 5.5 Diagrama de flujo de la rutina de inicialización de las colas



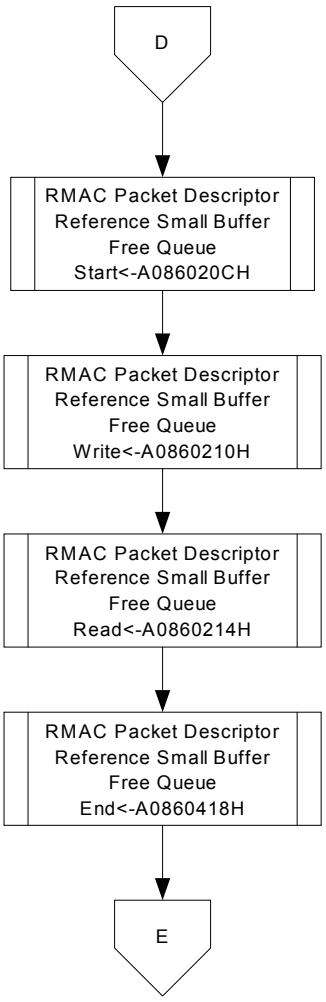
Continuación Figura 5.5 Diagrama de flujo de la rutina de inicialización de las colas



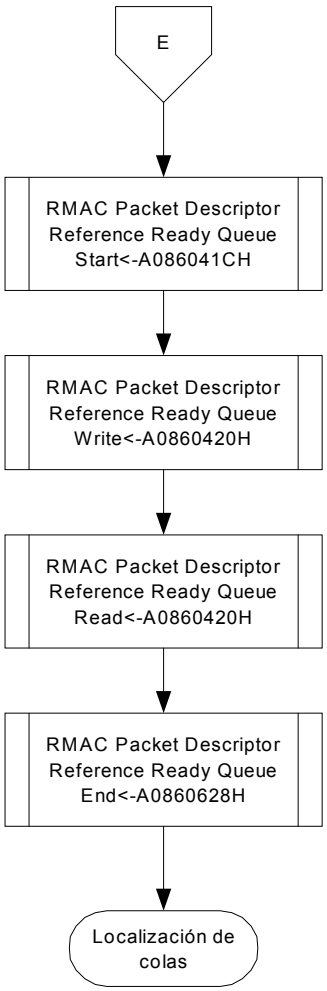
Continuación Figura 5.5 Diagrama de flujo de la rutina de inicialización de las colas



Continuación Figura 5.5 Diagrama de flujo de la rutina de inicialización de las colas



Continuación Figura 5.5 Diagrama de flujo de la rutina de inicialización de las colas



Continuación Figura 5.5 Diagrama de flujo de la rutina de inicialización de las colas

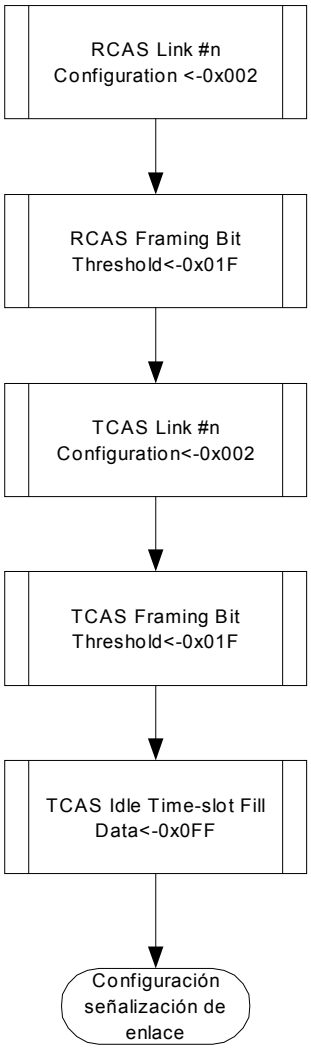


Figura 5.6 Diagrama de flujo de la configuración de los enlaces.

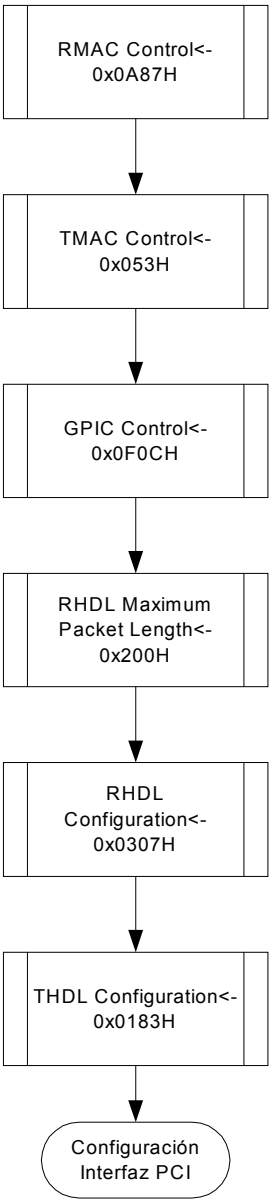


Figura 5.7 Diagrama de flujo de la inicialización del bus PCI

5.2 Software del datalink

Para el caso del data link el software fue desarrollado en ABEL, y consta de las ecuaciones combinacionales, que realizan la combinación de las señales provenientes del registro de desplazamiento y del reloj, con lo cual se obtiene un reloj invertido, el cual no varía durante el canal cero del ST-Bus.

Estas ecuaciones son las siguientes:

```
relojA=bit0A&bit1A&bit2A&bit3A&bit4A&bit5A&bit6A&bit7A&(!CLKA);

relojB=bit0B&bit1B&bit2B&bit3B&bit4B&bit5B&bit6B&bit7B&(!CLKB);
```

El diagrama de bloques que describen el proceso realizado en las GAL's se muestra en la figura 5.8

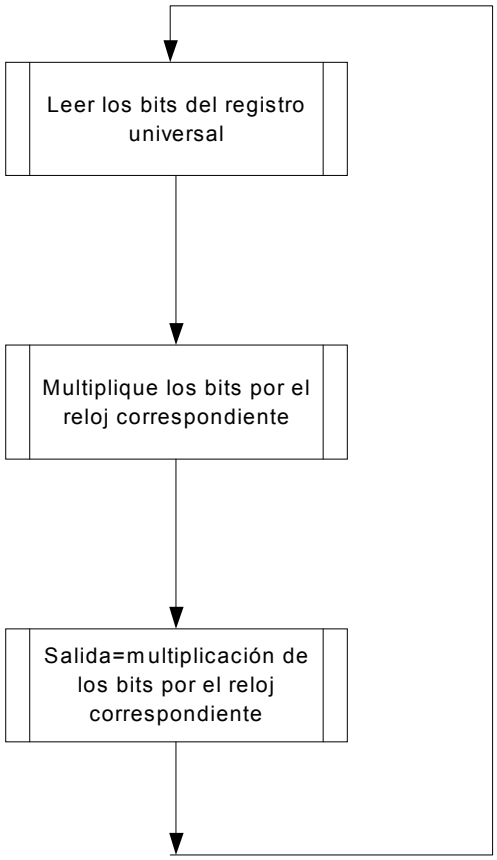


Figura 5.8 Diagrama de flujo del programa que posee la GAL en la tarjeta DATALINK PCI.

Capítulo 6
Análisis y resultados

6.1 Explicación del diseño

6.1.1 Data Link:

Consta de la etapa que permite el alineamiento de los buses ST-Bus. Los datos provenientes de la tarjeta E1 ingresan, en formato ST-Bus, a cuatro conectores RJ-45. En cada RJ-45 se ubican dos ST-Bus, uno para transmisión y otro para recepción con la señal de alineamiento de f0b y el reloj del stream. Como el PM-7366 requiere de que en el byte que conforma el canal 0 del ST-Bus el clock no realice cambio, es decir que permanezca en 0, se utiliza un registro de desplazamiento, donde se va corriendo el bit de alineamiento f0b durante los 8 bits que lo conforma. Los 8 bits de este registro y el inverso del reloj del stream son multiplicados por una AND en todo momento.

La señal de alineamiento fob, que es la entrada serial al registro universal, permanecerá en alto hasta que ocurra el primer ciclo de reloj del canal cero, donde realizará una transición a cero durante un ciclo de reloj. Este cero se desplazará en el registro mientras transcurren los ocho bits del canal cero y luego saldrá, dejando de una carga de unos en todos los bits del registro. Con esto se tiene que mientras el cero de la transición de la señal de alineamiento se encuentre dentro del registro el reloj se mantendrá en cero y el resto del tiempo éste será una versión negada del reloj original, esta forma de alineamiento es similar a la f1. Esta información, lista para ser procesada por el controlador de protocolo HDLC, PM-7366 se transporta a la tarjeta HDLC-4M a través de un conector DB25 y un cable plano.

Las señales en el conector plano se colocaron de manera tal que existiera una señal en medio de dos señales de tierra, esto para lograr un blindaje a la señal y reducir la cantidad de ruido que se pueda introducir en la comunicación.

Ésta, posterior a la entrega de este proyecto, será montada en un circuito impreso, cuyos componentes serán de montaje tipo DIP. El montaje del mismo se realizará en la planta de Cibertec Internacional en Alajuela, y el PCB de la tarjeta se enviará a fabricar a Estados Unidos a la empresa Paragón.

6.1.1 Tarjeta HDLC-4M

El proyecto consistía en rediseñar la etapa de recepción de datos para la nueva versión de la tarjeta HDLC-256.

En la etapa de recepción, ingresan 4 ST-Bus provenientes de la tarjeta Data Link los cuales vienen en el formato que requiere el PM-7366 para el alineamiento de la trama del bus, o sea con alineamiento del tipo f1. El FREEDM-8 toma estos datos y les quita toda la información dedicada a protocolo HDLC. Los 8 canales de información que vienen en los ST-Bus, son colocados en la pila interna de 8KB. Se programó que para cada canal existan 4 bloques de 16 bytes. Una vez que la información recolectada llene la FIFO, o bien, se requiera de la información de algún o algunos canales, la información será enviada a la memoria SDRAM del microprocesador huésped utilizando el DMA que se encuentra internamente en el controlador de protocolo HDLC.

Como el FREEDM-8 requiere de memoria RAM para guardar las posiciones de las colas y descriptores necesarios para la efectiva comunicación con el microprocesador local, se escogió un procesador que tuviera el controlador de memoria interno, y que permitiera la transferencia transparente de información entre la memoria y el dispositivo. La programación del microcontrolador para la efectiva comunicación entre la memoria y el PM-7366 se realizó de manera tal que diera soporte a los tipos de memoria seleccionados. Por lo tanto, la conexión que se realizó en el esquemático entre el PM-7366 y el microprocesador por medio del bus PCI es suficiente para la efectiva comunicación con la memoria.

Una vez que los datos son recibidos, el PM-7366 se encuentra programado para que elimine toda la información de protocolo HDLC. Posee los índices necesarios, para ubicar dentro de la memoria interna, arreglada como memorias FIFO para cada uno de los canales, la información según la dirección que le corresponda.

Una vez que la memoria FIFO de uno o varios canales específicos llegue a su capacidad, o bien, que sea necesario leer la información que se tenga recolectada, el FREEDM-8 realizará un pedido de interrupción en la línea de interrupción externa del bus PCI, que se encuentra conectada a la interrupción no mascarable, NMI, del procesador 80960RM. Al producirse la interrupción, será necesario el análisis del registro de interrupciones para verificar la fuente de interrupción, y una vez reconocida la interrupción, se realizará la transferencia de datos a través del DMA. Esta transferencia por DMA es transparente al usuario, ya que se verá toda la transacción como comandos del PCI.

Estos comandos de PCI, en el caso de recepción de datos hacia la PC, serán de escritura a memoria. Para estas escrituras, al ser un comando y no una línea como se encuentra fabricada la memoria, es necesario introducir un controlador. El procesador 80960RM incorpora un controlador de memoria para diferentes tipos, y actúa transparente a la unidad de procesamiento, ya que las unidades de traslado de datos y direcciones son las encargadas del funcionamiento de este bloque. Es por este motivo se programó la sección de registros del controlador de memoria. Con esto la interfaz de memoria con el FREEDM-8 se da por concluida con las conexiones realizadas al secundario y esta programación.

Las estructuras necesarias para las transacciones como lo son los descriptores y las colas permiten organizar las FIFO's internas y pasar los datos de la memoria del PM-7366 al host.

En la sección de transmisión, los datos provenientes de la computadora se trasladarán directamente del bus primario al secundario, para ser procesados por el FREEDM-8. Estos datos serán recibidos y escritos en la FIFO interna. Cuando sea necesario, estos datos se les adicionará el protocolo HDLC y se empaquetaran en formato ST-Bus y serán enviados a la etapa de Data Link. Esta sección de traslado del primario al secundario es realizada por el árbitro y corresponde al control del puente PCI-PCI.

Debido a que el puente PCI-PCI que se encuentra integrado al microprocesador no posee buffer de reloj, es necesario colocar un buffer de reloj el cual pueda levantar la corriente proveniente del bus PCI de la computadora y pueda proporcionar el reloj al dispositivo controlador de protocolo HDLC. La forma de conexión que se presenta, se da para reducir los desfases a la salida de este buffer. Se puede ver mayor información de esto en las notas de aplicación de este componente.

Además como la tarjeta debe de proveer un bus PCI de 5V o de 3.3V según sea el slot de conexión de la computadora, fue necesario la conexión de una configuración de comparador, de forma tal que si las líneas +I/OV son de 3.3V la salida del comparador será un cero lógico, y como este se conecta a la línea EN5V, que es la que controla el voltaje del bus PCI, el cero configura al bus a trabajar en 3.3V. Si el voltaje en las líneas +I/OV es de 5V, la salida del comparador es un 1 lógico y por tanto se configura el bus del FREEDM-8 a trabajar a 5V.

Los 5V que se conectan a las líneas de VBIAS[1:3] provee al controlador la capacidad poder manejar dispositivos de 5V, como lo son las líneas ST-Bus y la salida del comparador.

La red capacitiva que presenta la alimentación del FREEDM-8 funciona como acople, para evitar ruidos y reflexiones en las líneas del bus.

6.2 Alcances y limitaciones

Actualmente la tarjeta se encuentra en la fase de producción del PCB y próxima a la fabricación de las tarjetas prototipo para su prueba. Se tienen, como un punto extra en cuanto a lo que se había prometido con respecto al proyecto de graduación, los cuatro últimos avances que la empresa DataLinePCB ha podido enviar, estos se adjuntan en los anexos.

Durante el diseño de esta tarjeta se han dado circunstancias que han obligado en muchos casos a cambiar la forma de realizar alguna tarea conforme a lo que se tenía establecido en el inicio del proyecto, y algunas veces llevo al cambio del dispositivo encargado de tal función.

Cuando se inició el estudio del dispositivo controlador de protocolo HDLC, se había decidido, por parte de la empresa, la utilización del dispositivo PM-7364, sin embargo, luego de comparar su funcionamiento con un dispositivo análogo, el PM7366, se pudo comprobar que su funcionamiento era exactamente el mismo, y la única diferencia radicaba en el que PM7364 es capaz de manejar hasta 32 conexiones de ST-Bus en cada uno de los sentidos, mientras que el PM-7366 puede manejar hasta 8 conexiones de ST-Bus en cada sentido. Al analizar cual sería un número conveniente de conexiones, sin afectar el buen funcionamiento de la tarjeta, en lo que a velocidad de procesamiento de datos en el microprocesador se refiere, así como en la máxima utilización de Time-Slots de cada uno de los ST-Bus, provenientes de la tarjeta CPU E1, se logró determinar que, si es posible que se logre transportar toda la información de una CPU E1 en dos canales de la HDLC-256, al aumentar la cantidad de canales en los que se transmiten los datos necesarios, y al poder expandirse aún más, pues permite que se utilicen de cada stream de ST-Bus hasta 31 canales, entonces se puede llegar a la conclusión de que el PM7366 es la mejor opción, además que es de costo más reducido que el PM7364. Luego de esta decisión, se vio la opción que presentaba el FREEDM-8, en la que se presentaba en dos versiones, una de 256 y otra de 272 pines. Como la

versión de 256 pines es pin compatible con el PM7364 se decidió utilizar este modelo, y fue con este con el que se realizaron los primeros modelos de los esquemáticos, sin embargo, debido a que la diferencia de precios entre los dos modelos, era de más de $60, se decidió utilizar el otro modelo, dejando la ventaja de pin compatible atrás, y realizando los cambios respectivos sobre el circuito esquemático.

En cuanto al uso de este dispositivo, se encuentra que la decisión de su uso es la más acertada por varios motivos. Por un lado el hecho de que permita utilizar hasta 31 canales en señalización del tipo E1, o bien los 24 canales en señalización del tipo T1, hace que se aproveche al máximo la característica de TDM, que es el poder llevar información de varias fuentes por un mismo medio, en un intervalo de tiempo definido, únicamente asignando posiciones en el tiempo para cada segmento de información. Esto era uno de los problemas que presenta la tarjeta HDLC-256, cuyo controladores de protocolo HDLC MT8952 únicamente podían aprovechar dos de los 32 canales o segmentos de tiempo que le brinda ST-Bus para transportar la información, perdiéndose segmentos que podrían ser utilizados para transportar más información. Además la ventaja de que este dispositivo, únicamente por medio de software puede configurarse para trabajar con información no canalizada, es decir, con información que no es de TDM, hace que se pueda ver como que cada stream es un bus serial y transportar información serialmente de un dispositivo cualquiera al microprocesador local de la tarjeta, y a partir de éste a la PC, y hacer esto también en el sentido contrario, es decir, sacar datos del PC en forma paralelo a través del bus PCI y enviarlos a otro sistema en forma serial. Además el hecho de que las transferencias de datos desde los bloques de transmisión o de recepción se puedan dar en forma transparente, esto es, sin agregar o quitar el protocolo HDLC, hace aún más flexible el uso de la tarjeta ya que no se limita a la utilización de este protocolo para realizar la transferencia de datos. Es por esto, que la tarjeta HDLC-4M no es tan sólo una interfaz que logra tomar los datos extraídos de la línea telefónica por medio de la tarjeta CPU E1, empaquetados en formato ST-Bus y transmitidos con

protocolo HDLC, sino que esta se transforma en una interfaz serial para casi cualquier tipo de comunicación serial que algún sistema requiera para comunicarse con un PC, o bien, que la PC necesite comunicarse utilizando un bus serial, como se ha visto en el caso de la Quatech o de la tarjeta Digiboard.

Puede soportar interfaces de +5V, únicamente conectando a +5V las entradas de VBIAS, lo cual facilitó el uso de dispositivo TTL en la tarjeta Data Link que antecede a la conexión de la tarjeta HDLC-4M con la tarjeta CPU-E1 que también funciona con lógica a 5V, por tanto se puede hablar de una compatibilidad hacia atrás.

Aunque el diseño se había planeado que fuera para utilizar cuatro de los enlaces ST-Bus de los ocho que ofrece el integrado PM-7366, se realizó la conexión de un header que da opción de salida utilizando un cable plano a los enlaces restante de ST-Bus que no fueron utilizados, esto para prever una expansión en el uso de enlaces de la tarjeta por parte de la compañía si la tarjeta fuese utilizada en un sistema con características diferentes de las que tenía el sistema para el cual esta fue creada. Se realizó la programación para la utilización de los cuatro enlaces que inicialmente se habían escogido, y pensándose en utilizar ocho canales por cada enlace, con lo que resulta un uso de 32 canales, de los 128 posibles. Sin embargo, la opción de ampliación tanto de la cantidad de enlaces como de canales es totalmente posible, la primera mediante el header que ha sido agregado, y lo segundo mediante programación.

El montaje de la tarjeta Data Link en un módulo separado posee dos objetivos, el primero y de mayor peso, es que al ser la tarjeta diseñada del tipo PCI debe de cumplirse con estándares de tamaño, y por tanto se debe de prever aprovechar al máximo posible el espacio de posicionamiento que ofrece la tarjeta para dispositivos que es necesario que manejen la velocidad de PCI, al ser la velocidad en los data link de 2.048 MHz no es necesario que se encuentre sobre la tarjeta PCI, además que se ocuparía espacio de la tarjeta en lógica ajena al procesamiento de datos en formato HDLC, así que la colocación de esta tarjeta en un módulo aparte simplifica la lógica presente en la tarjeta HDLC-4M, brinda más espacio para realizar las

conexiones sobre la tarjeta PCI, y es aquí de donde nace el segundo objetivo, que es independizar a la tarjeta HDLC-4M del formato HDLC E1 y que pueda ser utilizado para otro tipo de señalización como lo es la señalización del tipo T1, o bien, como ya se habló la transferencia de datos entre la PC y otros dispositivos que utilizan otras formas de comunicación serie utilizando la tarjeta, variando únicamente el software.

La lógica de la tarjeta Data Link se decidió realizarla diferente al método que proponía la compañía PMC-Sierra, que era utilizando compuertas lógicas. Esto llevaba al uso de un gran espacio sobre la tarjeta DATALINK y a utilizar dispositivos que podrían descontinuarse próximamente, por lo que se pensó en el uso de dispositivos CPLD. Por tanto, se utilizó un registro universal para poder realizar los corrimientos necesarios, así como la lectura simultánea de los 8 bits que conforman el registro, de esta manera, utilizando un dispositivo CPLD ser realiza la multiplicación de todos los bits del registro con el reloj, y se logra obtener la forma de reloj que buscaba para el alineamiento de los buses de ST-Bus para que el PM7366. El tipo de CPLD utilizado es del tipo PAL, así se podrá asegurar que no serán borrados con el tiempo o con algún pico de voltaje que se pudiese producir y borrara la información que ha sido programada.

Dentro de los problemas que se han encontrado al diseñar esta tarjeta es la búsqueda de componentes que no se encuentren cerca de dejar de producirse, esto es, buscar que la obsolescencia de los componentes se encuentre tan lejana, como el ambiente de la electrónica así lo permitan. El microprocesador 80960-VH, presentaba algunas deficiencias que hizo que el cambio fuese necesario. Este microprocesador podía manejar memoria del tipo SRAM, lo cual era una de las propuestas que la empresa tenía, ya que quería una memoria de datos lo más rápida como fuera posible, sin embargo la memoria SRAM es de precio muy elevado (arriba de $400 en módulos de 4MB), por tanto se sale de las expectativas de costo de memoria que la empresa tenía previsto. Al analizar el tipo de memoria DRAM que este mismo microprocesador es capaz de utilizar, se pudo observar que era del tipo FPM/EDO, la cual, es un tipo de memoria que se está dejando de usar, ya que

estas presentan el dato por periodos pequeños de tiempo, y aunque el segundo tipo logra que el dato permanezca durante más tiempo sobre el bus de datos, la utilización de un reloj externo para sincronizar los comandos de entrada, los datos y las direcciones con el ciclo del base del reloj, hace mas eficiente el funcionamiento de la DRAM sincrónica. También la SDRAM presenta señales para manejo y registros para programar la latencia del #CAS y los tipos y largos de ráfagas. Para mayor información refiérase al documento "Migrando de FPM/EDO a SDRAM" Buscando dentro de las memorias DRAM cual era el modelo que es más conveniente y más utilizado se halló que el tipo que reunía estas características era la memoria del SDRAM, la cual es la que se utiliza corrientemente en los bancos de memoria de las computadoras actuales, sin embargo, faltaba por determinar si se usaba la memoria SDRAM en chip o si se optaba por utilizar un módulo SIMM o DIMM. Debido a las tendencias actuales del mercado, lo más estándar en cuanto a memorias se refiere, son los módulos DIMM de SDRAM. Con toda esta nueva investigación se encontró uno de los problemas que presentaba el microprocesador 80960RM, este no poseía dentro de las opciones del controlador de memoria este tipo de memoria.

Por otro lado, se había propuesto a la empresa la utilización de un circuito puente para realizar el arbitraje de los buses primario (que va a la PC) y del bus secundario (que va al dispositivo controlador de protocolo HDLC). Esta propuesta fue rechazada por la empresa, ya que significaba la compra de otro componente de precio significativo, además que este puente se iba a ver subutilizado. El procesador 80960-VH posee únicamente una interfaz PCI y no posee arbitro interno, por tanto era necesaria una lógica extra, externa para realizar la labor de arbitraje. La propuesta por parte de la empresa era la utilización de dispositivos CPLD's para efectuar toda la lógica necesaria, sin embargo el cumplir con todas las disposiciones de tiempo, alambrado, cargas, etc que PCI exige era una labor larga y riesgosa, ya que no se puede asegurar que lo programado cumpla al 100% con la revisión 2.1 de PCI.

Estas dos razones se consideraron de peso como para realizar una nueva búsqueda de microprocesador, y de esta manera justificar a la empresa que la forma de realizar el arbitraje era mejor hacerla con un circuito puente ya implementado, así como el cambio de tipo de memoria a utilizar.

Se analizó la familia de procesadores 80960 y es logró localizar la versión 80960 RM donde se presentaban las ventajas deseadas. Este contaba con un controlador de memoria capaz de manejar memoria del tipo SDRAM, y que dentro de sus hojas de datos daban las especificaciones para la utilización de módulos DIMM, tanto de dispositivos de 64 Mbit así como de dispositivos de 16 Mbit, lo cual se ajusta a las características que se estaba buscando. La integración del circuito puente en el microprocesador, evita la compra de un tercer dispositivo, y se logra realizar efectivamente la labor de arbitraje. Este microprocesador cuenta con dos interfaces PCI, en las cuales, se conecta en el secundario el controlador de protocolo HDLC y en el primario la computadora. El manejo de la sección de comunicación PCI queda a cargo del estudiante Simón Sánchez.

Debido a que se debía utilizar el menor espacio físico sobre la tarjeta, y buscar que los componentes fueran lo mas estándar posible en cuanto al tipo de montaje, se buscó que la memoria EPROM fuera del tipo PLCC. Sin embargo, memoria del tipo EPROM de 3.3V se encontraron únicamente del tipo OTP, el cual no es el apropiado para prototipos como lo es la tarjeta diseñada, esto por motivo de que siempre existirán errores en la programación que será necesario corregir, y con este tipo de memorias, cada vez que se localice un error será necesario reemplazar la memoria sin tener la posibilidad de borrarla y volverla a utilizar, sino que se deberá adquirir otra, lo cual no es viable económicamente para la empresa. Por esta razón, en la etapa de prototipo se decidió utilizar memoria EPROM de 5V, con montaje LCCC. El tipo seleccionada permite que se pueda borrar y volver a programar. Además, el microprocesador seleccionado, permite el manejo de memoria Flash, por lo que en la tarjeta del prototipo existe la prevista para el uso de este tipo de memoria, ya que cuando se termine la programación y depuración del sistema, se realizará el

cambio a este tipo de memoria que es mucho más rápida que la EPROM tradicional. Esta prevista a memoria Flash nos lleva a que en cuanto a memorias, el sistema cuente con la utilización de dispositivos de tecnología de punta, que son los que actualmente se están usando.

Por esta razón fue necesario de ocuparse en la búsqueda de sockets tanto para localizar la memoria sobre la tarjeta así como para realizar la programación, verificando, antes de hacer la selección de memorias, que los programadores presentes en la empresa sean capaz de programar la serie de memoria seleccionada.

Se pueden observar en la tarjeta realizada algunas resistencias de pull up, pull down y capacitores. Las funciones de estas resistencias son las de cumplir con especificaciones de PCI. Este estándar exige el uso de resistencias de pull up en líneas como FRAME, IRDY, TRDY, entre otras. Según este estándar, el valor de estas resistencias debe ser entre 2.8KΩ a 8.2KΩ  r lo tanto, para localizar estas dentro del rango predeterminado, se decidió ponerlas de 3.3 V.

Hacia el lado del microprocesador, se ubican otras resistencias de pull up/down que se dejaron seleccionables por jumpers, esto se decidió realizar de esta manera ya que dependen de configuraciones propias de la operación del microprocesador, como lo es si se está trabajando en el modo de reset normal, si se está trabajando con un bus de datos de memoria de 64 o 32 bits.

Los capacitores funcionan como filtros en las alimentaciones, debido a la alta velocidad que se trabaja en PCI es normal encontrar picos de voltaje por factores como reflexión de onda, o bien por las múltiples alimentaciones que se utilizan en los dispositivos, se puede dar en el encendido de la tarjeta que exista una sobre corriente, estos capacitores funcionan como filtros de CD, y recortadores de picos en altas frecuencias.

Además, se debe de recalcar que sobre la tarjeta existe una lógica encargada de ubicar a los buses internos a trabajar en un ambiente seleccionable de 3.3V o de

5V, esto utilizando las líneas de I/OV que proporciona el conector PCI. Con estas líneas, si se tienen aquí, 3.3V el ambiente seleccionado es de 3.3V, y si se encuentran 5V, pues entonces se trabajará con ambiente de 5V. Por lo tanto, el microprocesador se configura siguiendo las mismas regulaciones que el bus, mas el FREEDM-8 necesita de una lógica adicional que sea capaz de colocar un 1 lógico cuando se esté en ambiente de 5V, y un cero lógico (que para el FREEDM es de 0V). Esta lógica se logró con la utilización del comparador LM311. Es por esta razón que como se muestra en el apéndice 1 la tarjeta cuenta con dos ranuras, o dos llaves, la de 3.3V y la de 5V, así esta tarjeta es lo que se conoce como una tarjeta dual, pues puede manejar los dos ambientes.

El buffer del reloj empleado es llamado de cero retraso, esto significa que utilizando una configuración tal que todas las salidas posean la misma carga el desfase de las señales de salida tenderá a cero, esto es deseable en esta tarjeta pues la sincronización de las señales juega un papel fundamental en el buen funcionamiento de la tarjeta.

El latch que se usa para demultiplexar el bus de datos y direcciones en la memoria EPROM tiene como característica especial, que aunque este se conecta a una fuente de 3.3V de alimentación puede soportar entradas y salidas de 5V. Esta ventaja es aprovechada para la prevista a memoria Flash, ya que la memoria Flash utilizada es de 3.3V, y la EPROM es de 5V, por tanto su utilización facilita la prevista a cambio de memoria.

Capítulo 7
Conclusiones y Recomendaciones

7.1 Conclusiones

7.1.1 El formato ST-Bus permite transportar información a diferentes puntos utilizando la técnica de multiplexado en el tiempo.

7.1.2 Los controladores de protocolo HDLC de la tarjeta HDLC-256 sólo permitían transportar información en 2 de los 32 canales que ofrece el formato ST-Bus seleccionado.

7.1.3 La memoria FIFO que se utiliza en la tarjeta HDLC-256 permite conocer mediante pines de salida si ésta se encuentra llena, vacía o a la mitad, característica aprovechada en la tarjeta para realizar un registro de estado de la tarjeta.

7.1.4 El procesador utilizado en la tarjeta HDLC-256 es un TMS320C26, el cual realiza un trabajo de filtrado en los datos entrantes para determinar si se trata de la señalización buscada.

7.1.5 El bus ISA es un bus de 4/8MHz, el cual posee un bus de datos de 8/16 bits, con una capacidad de direccionamiento de 24 líneas, es decir 16 Mbytes.

7.1.6 La tarjeta HDLC-256 es capaz de comunicarse con 4 tarjetas CPU E1, de las que toma dos streams, uno en cada dirección. Toma esta información filtra buscando la señalización deseada y la envía hacia la computadora.

7.1.7 El bus PCI es un bus de 33/66 MHz, con un bus de datos de 32/64 bits y un bus de direcciones de 32/64 líneas. Las transacciones las realiza por comando, así como las configuraciones que son realizadas en ciclos especiales.

7.1.8 El integrado PM-7366 es capaz de procesar hasta 8 enlaces seriales.

7.1.9 El FREEDM puede trabajar con datos en señalización tipo E1, T1, y no canalizada.

7.1.10 El PM-7366 posee una memoria interna de 8KB, distribuida en bloques a

manera de FIFO para guardar datos recolectados de los enlaces.

7.1.11 En el lado de recepción los datos pueden ser pasados a la FIFO interna con y sin el protocolo HDLC según sea programado.

7.1.12 Los time-slots que poseen información pueden ser asignados a un canal de los 128 posibles por medio de programación.

7.1.13 La salida al bus PCI del dispositivo controlador de protocolo HDLC se realiza por una petición de DMA dada por un bloque interno en el PM-7366.

7.1.14 Luego de que se ha realizado la petición de DMA la etapa de control del bus PCI interna en el dispositivo controlador de protocolo se encarga de realizar todas los cambios en las líneas del bus para realizar la transacción de los datos.

7.1.15 En la sección de transmisión los datos son tomados del buffer de datos localizados en la memoria RAM, y pasados a la FIFO, de donde se trasladan a la etapa de transmisión, en la que se agrega el protocolo para la transmisión.

7.1.16 La interfaz para el procesamiento de datos tipo E1 en la tarjeta HDLC-4M se diseño como un módulo aparte de la tarjeta para independizar por medio de hardware el funcionamiento de la tarjeta al tipo de señalización a utilizarse.

7.1.17 El alineamiento necesario para utilizar señalización E1 con el PM-7366 es del tipo f1.

7.1.18 Se utilizaron PAL's en el diseño de la tarjeta Data Link para evitar el peligro de que se borrara la información.

7.1.19 El procesador 80960RM posee un bus PCI primario y un bus PCI secundario con su respectivo módulo de arbitraje.

7.1.20 El procesador 80960RM posee un controlador de memoria que soporta memoria del tipo SDRAM , EPROM y Flash.

7.1.21 Las transacciones a memoria por los buses PCI puede ser transparente a la

unidad de procesamiento.

7.1.22 La unidad que se encarga de realizar la transferencia de datos transparente es llamada Memory Unit (MU).

7.1.23 La unidad encargada de realizar el traslado de direcciones es la Address Translate Unit (ATU).

7.1.24 El tipo de memoria seleccionada es la SDRAM debido a que es la más utilizada actualmente y de costo menos elevado en memorias volátiles.

7.1.25 El manejo de la línea de habilitación de ambiente de 5V para el bus PCI en el dispositivo PM-7366 se realiza con un comparador, para permitir el uso de la tarjeta en ambientes de 5V o de 3.3V.

7.1.26 El FREEDM-8 se comunica con la memoria utilizando el controlador interno del procesador 80960-RM, por medio del bus secundario, por lo que la conexión al secundario es suficiente para la comunicación a memoria.

7.1.27 El mapa de memoria realizado para el microprocesador se realizó siguiendo las indicaciones de Intel, procurando no hacer choques en los espacios de memoria.

7.1.28 La ubicación de las estructuras del FREEDM en la memoria se realizaron dando las máximas longitudes necesarias para el control de 32 canales.

7.1.29 El circuito esquemático realizado no es el Gerber del sistema, por tanto este circuito no cumple con las especificaciones PCI, pero si posee todas las conexiones necesarias para su funcionamiento.

7.1.30 Los Gerber Data son realizados por la empresa DataLine PCB en U.S.A, la cual posee la experiencia necesaria para su realización.

7.1.31 El montaje de los componentes en la tarjeta son del tipo SMT, y se realizará en la empresa AETEC en la Zona Franca Metropolitana, en Heredia.

7.2 Recomendaciones

1. Realizar las pruebas al circuito utilizando inicialmente el Kit de desarrollo que ha sido comprado.
2. Utilizar el sistema de CPU-E1 para el envío y recepción de Steams ST-Bus.
3. Realizar las pruebas determinando una posición fija para los jumpers para que estos sean eliminados.
4. Realizar la programación por segmentos para probar por etapas el circuito.
5. Realizar las pruebas de la tarjeta Data Link prototipo en GAL's y no en Pals ya que en caso de errores estas se pueden borrar y reutilizar.
6. Documentar todos los cambios que se deban realizar sobre el prototipo con la justificación necesaria, para futuros desarrollos.

Capítulo 8
Bibliografía

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.

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 Asyncrhronous FIFO. Cypress Semiconductor Corporation: San José, CA. USA. 1997.

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.

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

CD-ROM

PMC-Sierra Inc. “Internetworking Silicon Solutions”, Product Information Catalog. Volume 6 – Issue 2. Canada, 2000.

Capítulo 9

Apéndices y anexos

Apéndice 1: Características Mecánicas de los componentes utilizados

Para realizar las cotizaciones tanto en la empresa DataLine PCB como en AETEC, para la realización de los Gerber Data, PCB y montaje de componentes, fue preciso realizar un documento con las características mecánicas de todos los componentes utilizados, y estos se adjuntan a continuación:

Ap 1.1 PM7366-PI: Información mecánica

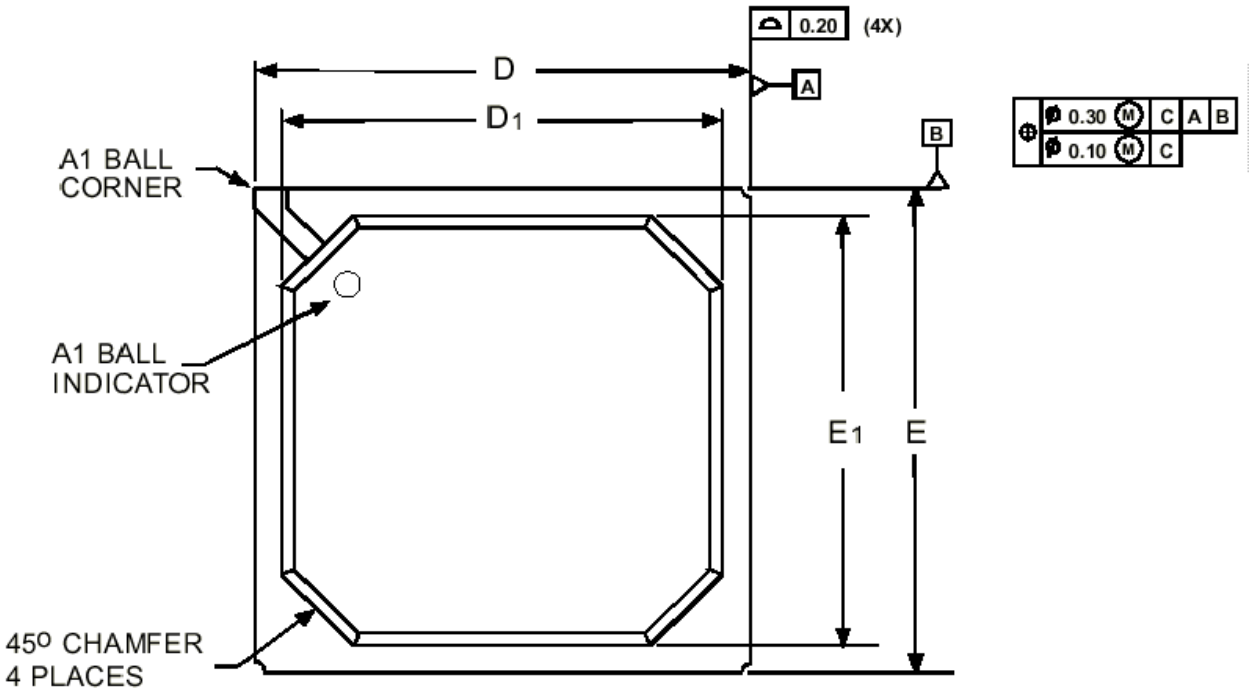


Figura Ap 1.1 Dimensiones físicas del chip PM-7366

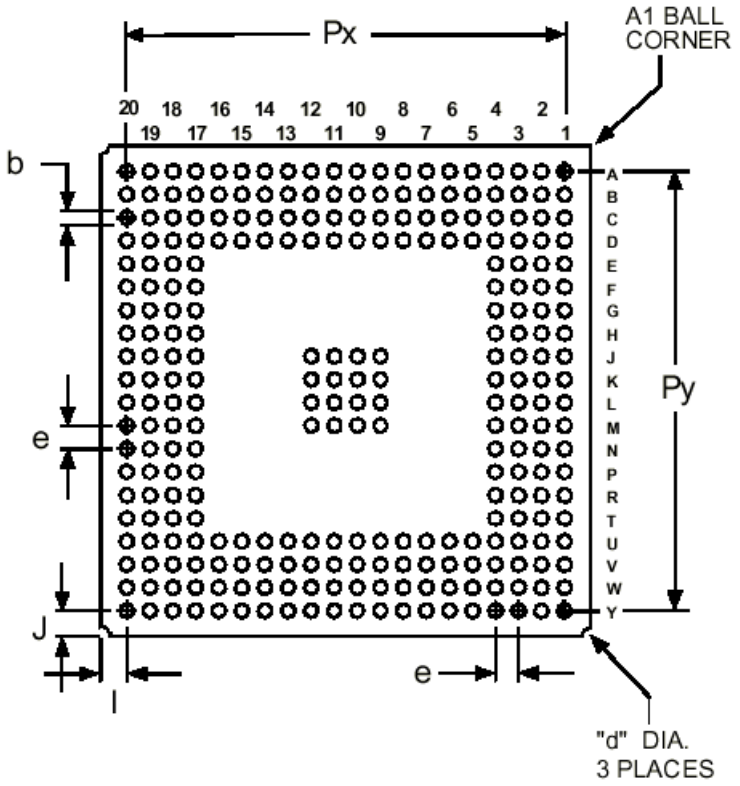


Figura Ap 1.2 Vista posterior del chip PM-7366 y dimensiones importantes

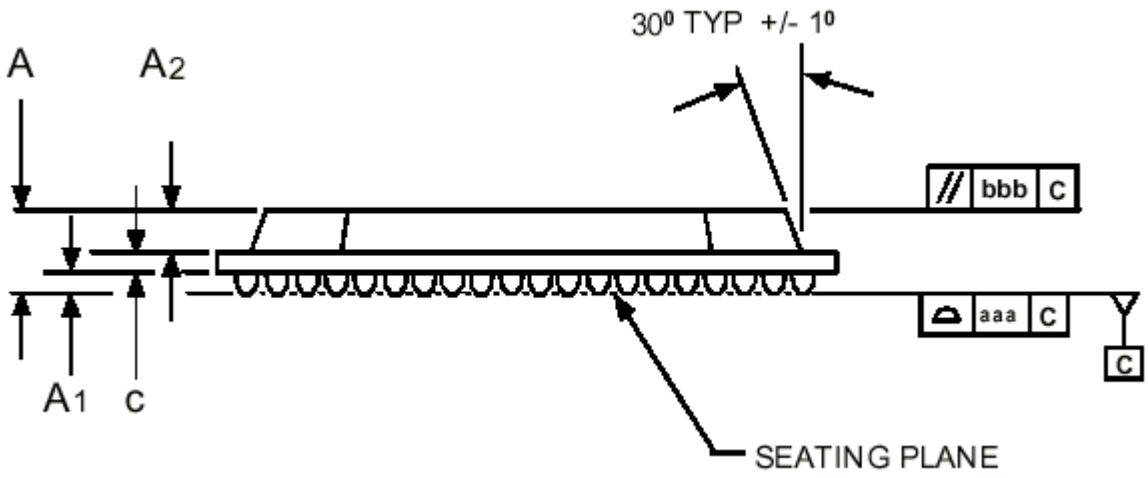


Figura Ap 1.3 Dimensiones de perfil del FREEDM-8

Tabla Ap 1.1 Cantidades de las dimensiones del PM-7366

PACKAGE TYPE : 272 PLASTIC BALL GRID ARRAY - PBGA																		
BODY SIZE : 27 x 27 x 2.33 MM (4 layer)																		
D in.	A (2 layer)	A (4 layer)	A1	A2	D	D1	E	E1	I	J	b	C (2 layer)	C (4 layer)	d	e	Px	Py	aaa	bbb
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

NOTAS:

1. Todas las dimensiones están en milímetros
2. Dimensión aaa denota que es coplanar.
3. Dimensión bbb denota que es paralelo.

Ap 1.2 80960-RM: Información mecánica

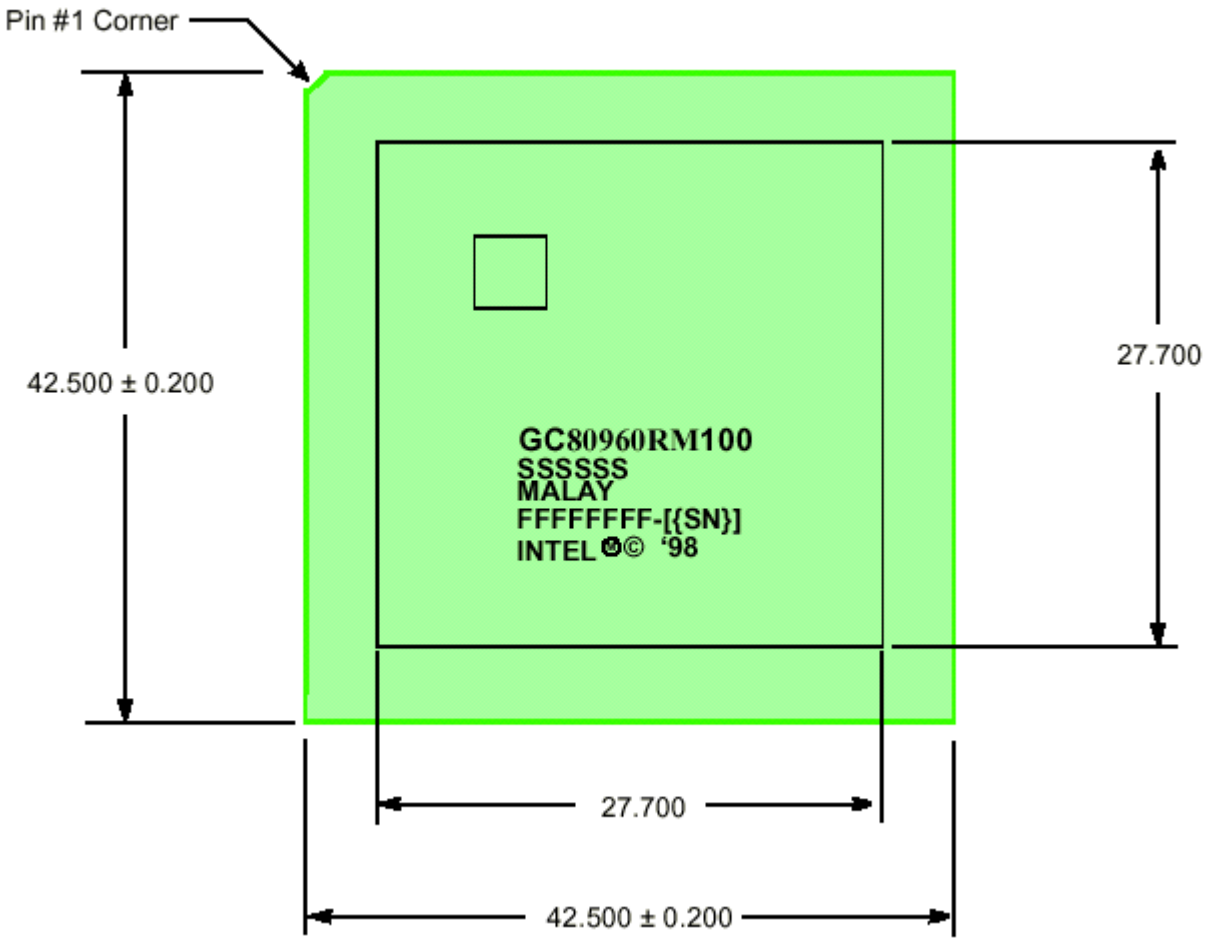


Figura Ap 1.4 Dimensiones de la parte superior del procesador 80960-RM

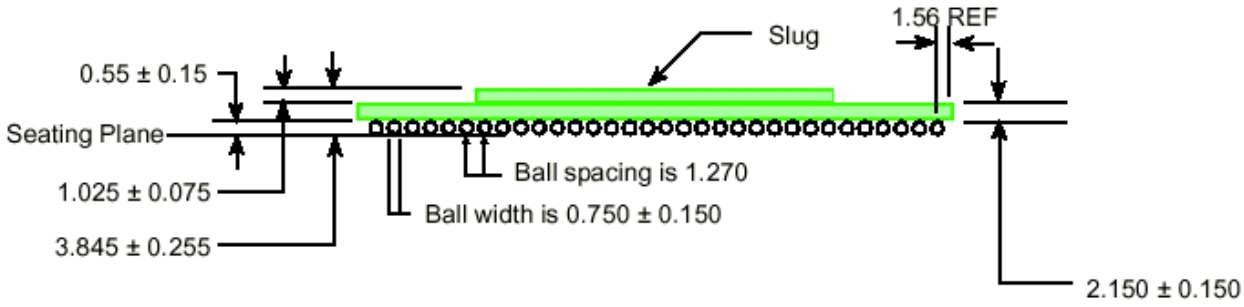


Figura Ap 1.5 Dimensiones del perfil del procesador 80960-RM

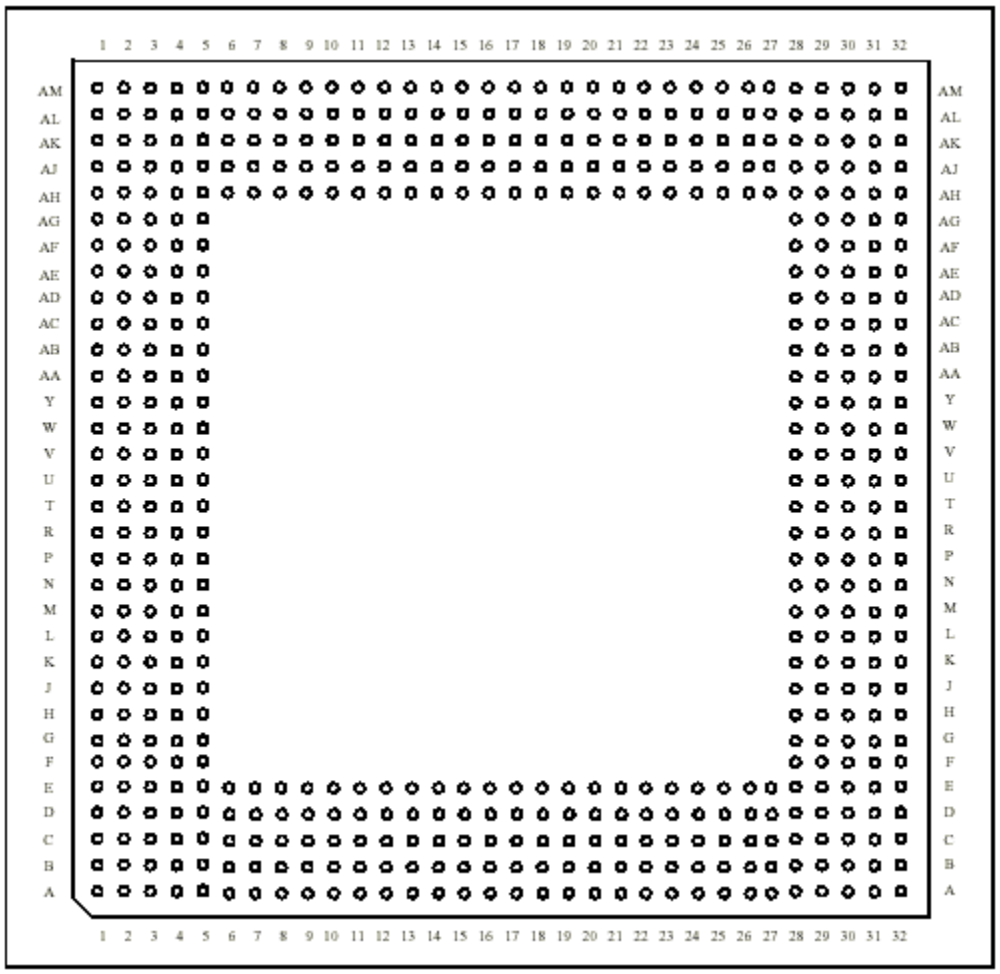


Figura Ap 1.6 Representación gráfica de la parte inferior del procesador 80960-RM

NOTAS:

1. Todas las dimensiones y tolerancias están conforme a ANSI Y14.5M 1982
2. Las dimensiones se han medido al diámetro máximo del punto de soldadura.
3. Todas las dimensiones son dadas en milímetros

Ap 1.3 Socket DIMM 875870159: Información mecánica

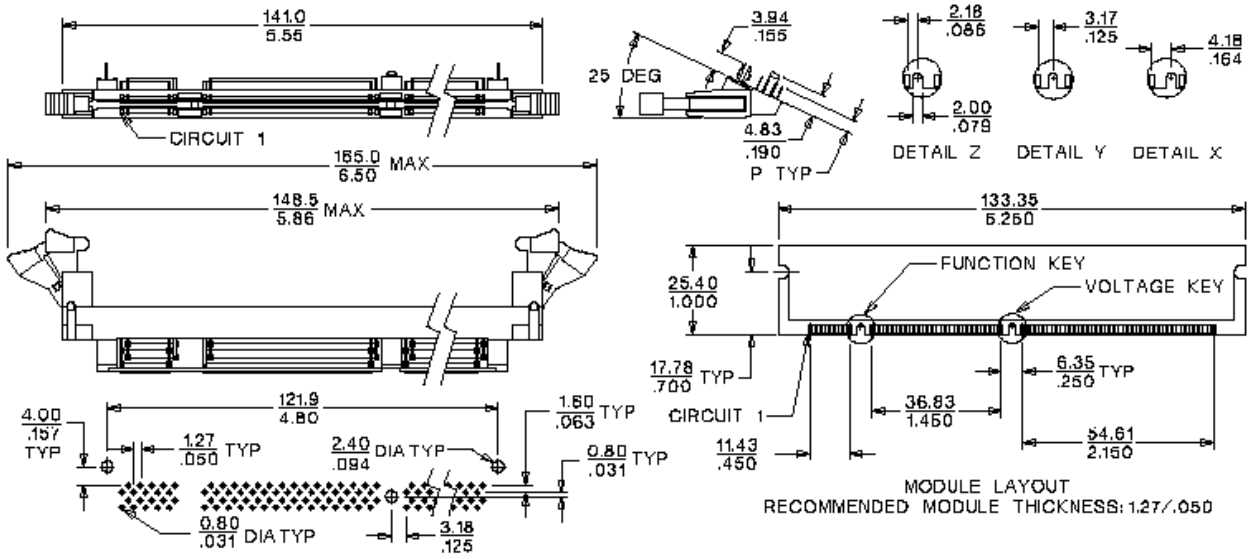


Figura Ap 1.7 Dimensiones del socket del DIMM de memoria.

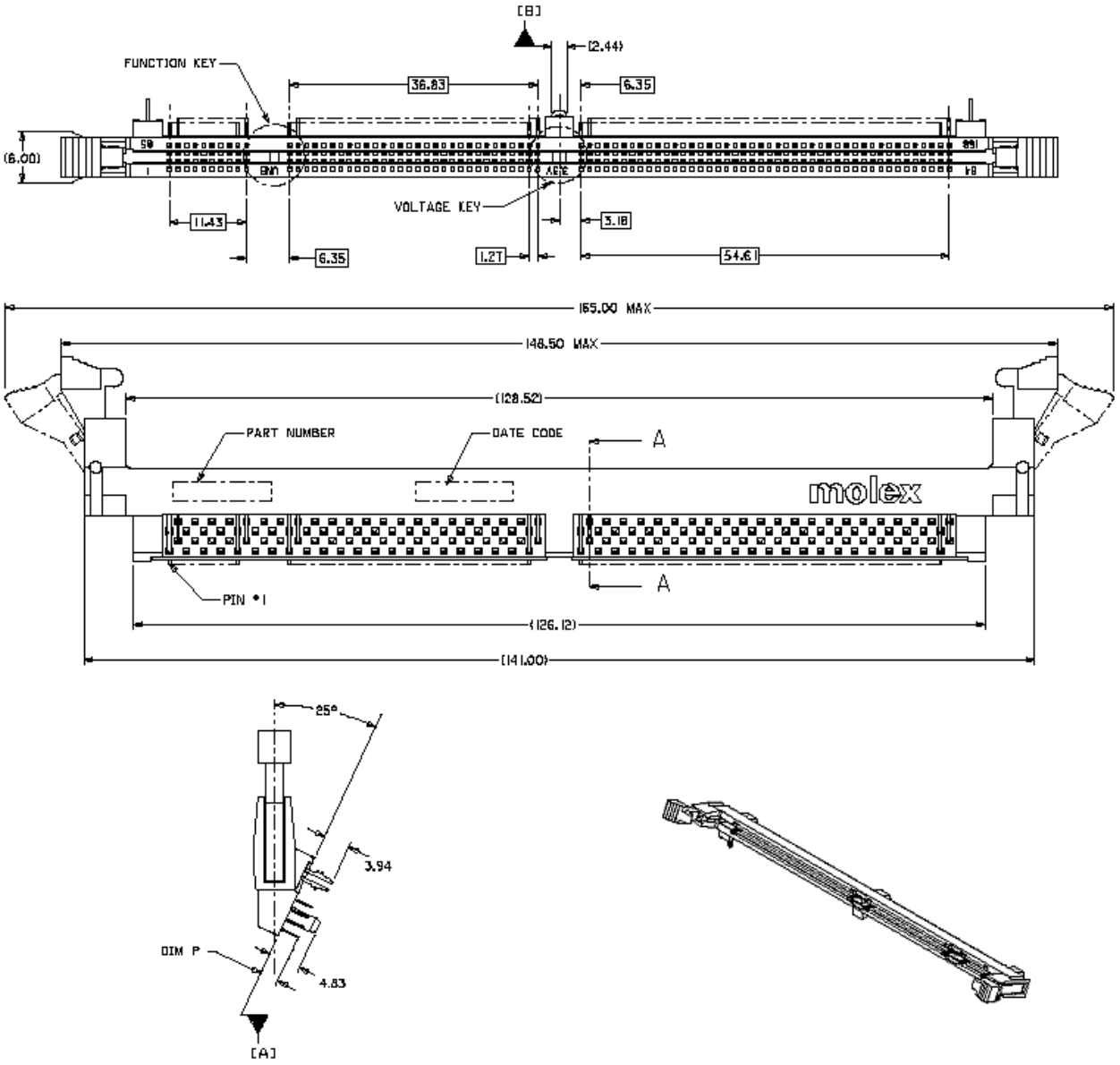


Figura Ap 1.8 Dimensiones físicas del Socket del DIMM de memoria.

Ap 1.4 Clock Buffer CY2505: Información mecánica

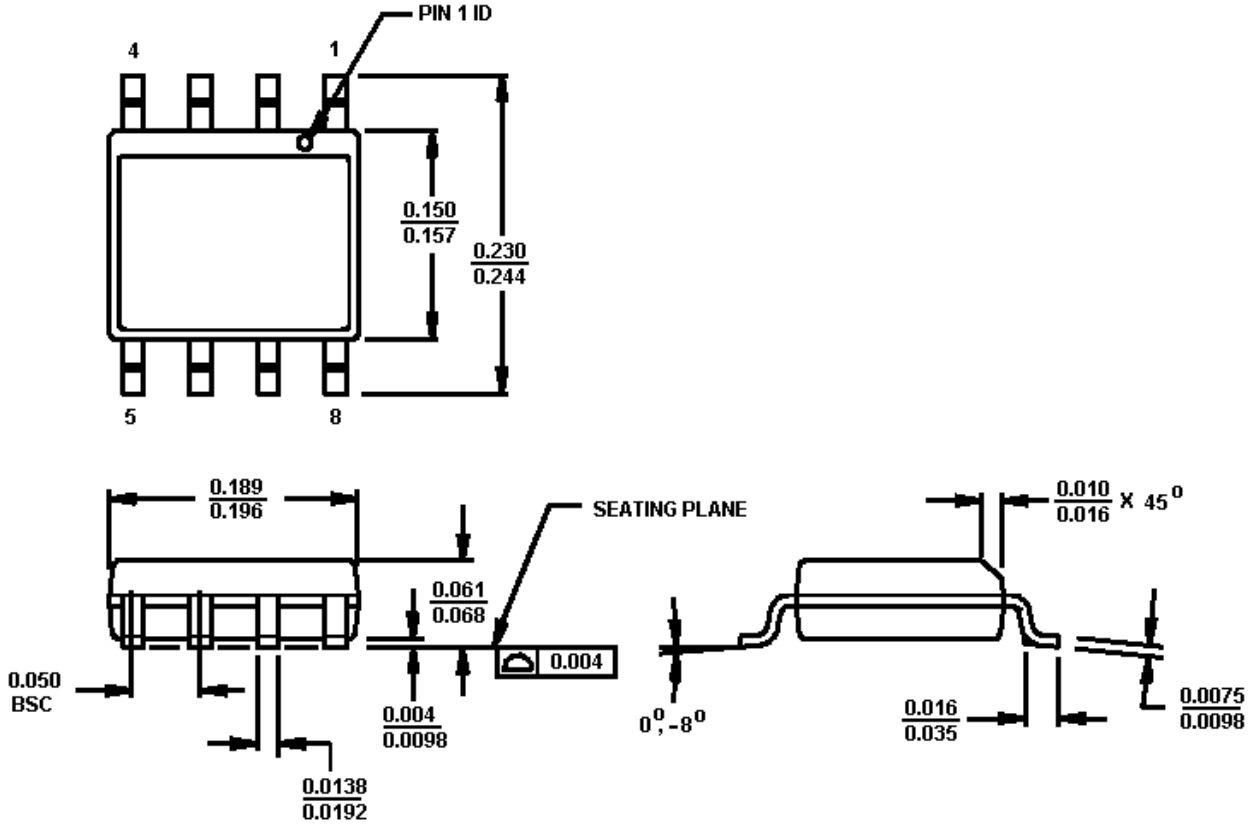
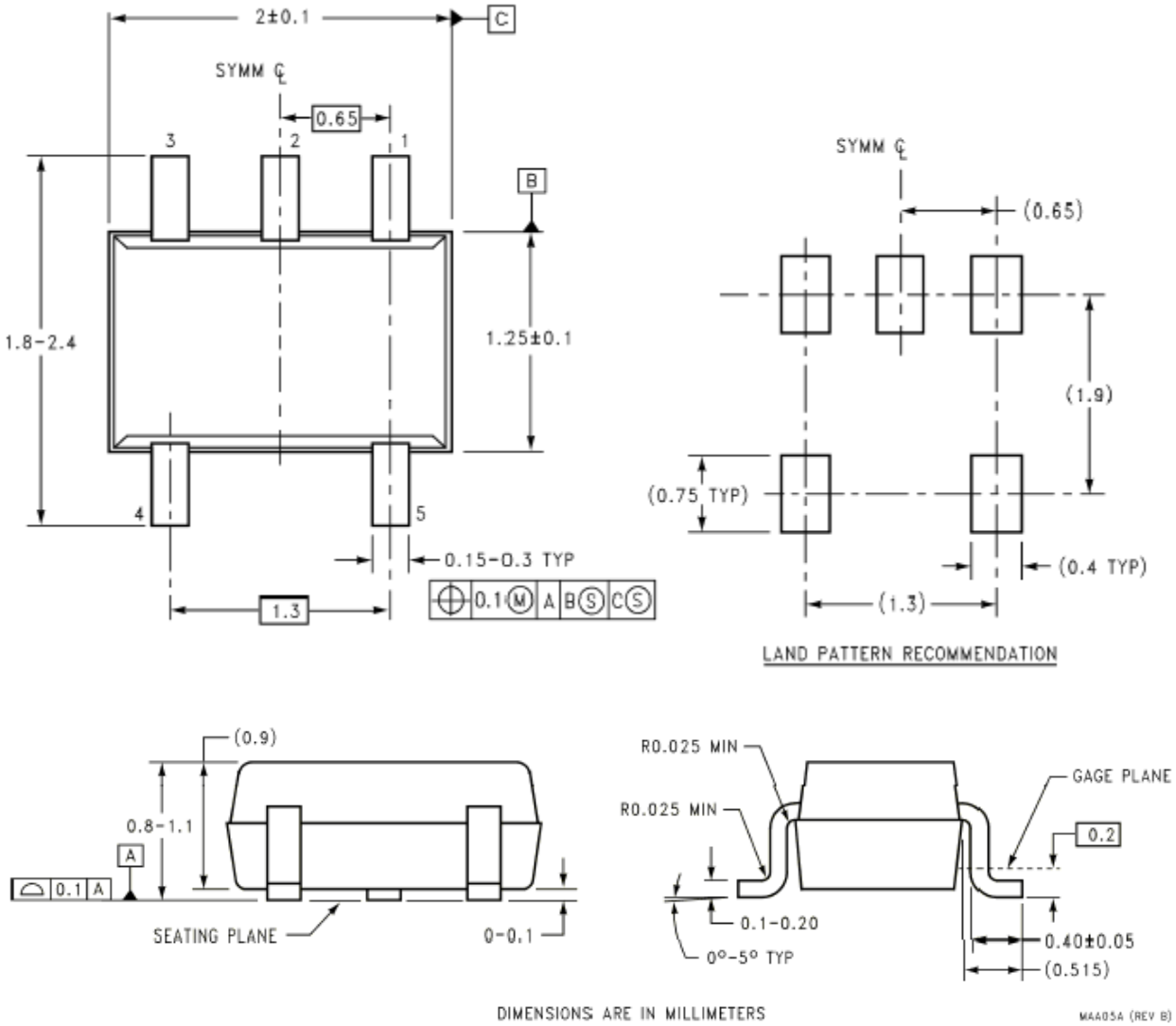


Figura Ap 1.9 Dimensiones físicas del buffer de reloj CY2505

NOTAS:

1. Dimensiones dadas en pulgadas (MIN/MAX)
2. El PIN 1 ID es opcional

Ap 1.5 Comparador LMV331: Información mecánica



5-Pin SC70-5 Tape and Reel
Order Number LMV331M7 and LMV331M7X
NS Package Number MAA05A

Figura Ap 1.10 Dimensiones físicas del comparador LMV331

Ap1.6 SDRAM module MT4LSDT1664A: Información mecánica

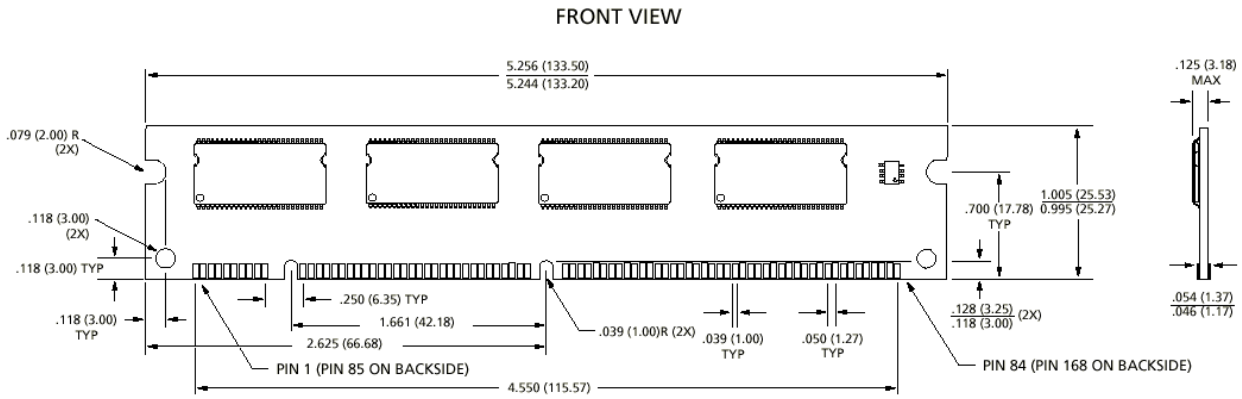
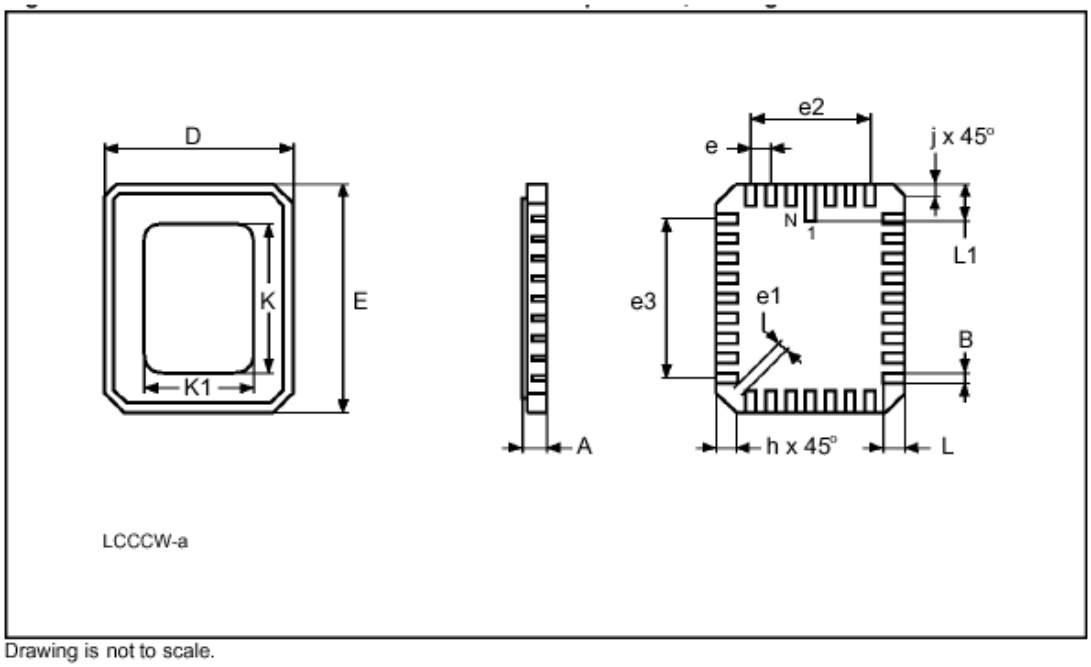


Figura Ap 1.11 Dimensiones del DIMM de memoria.

NOTES:

1. Todas las dimensiones se dan en pulgadas.

Ap1.7 EPROM M27C2001-55XL1X: Información mecánica



Drawing is not to scale.

Figura Ap 1.12 Dimensiones la memoria EPROM M27C001.

Tabla Ap 1.2 Cantidades de las dimensiones de la memoria EPROM

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		

Ap 1.8 EPROM SOCKET: Información mecánica

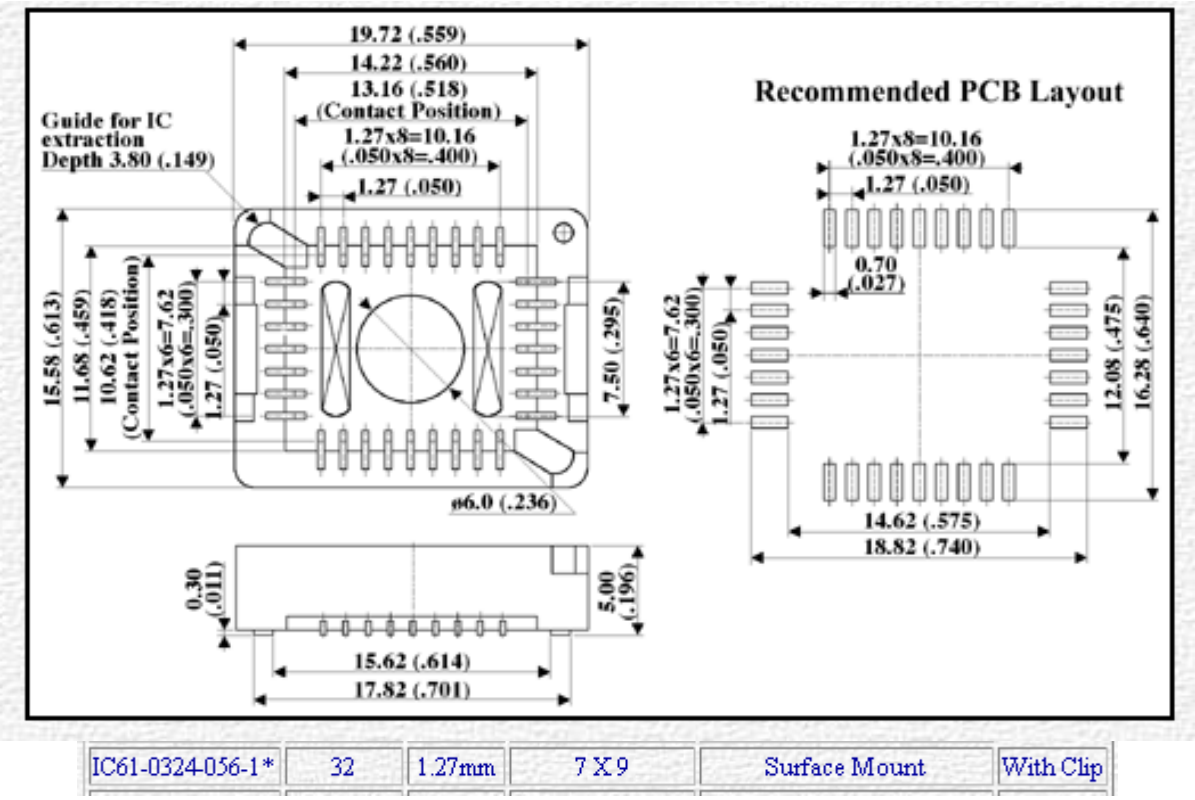
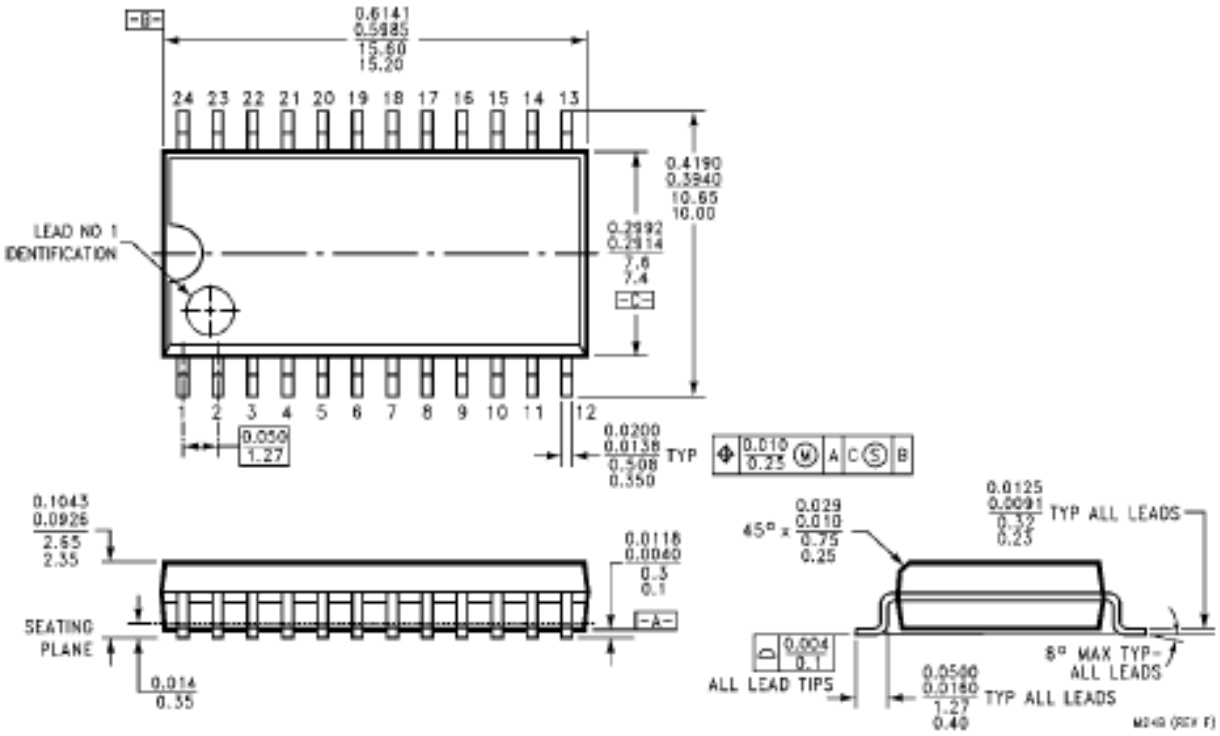


Figura Ap 1.13 Dimensiones del socket IC61-0324-056-1 para la memoria EPROM .

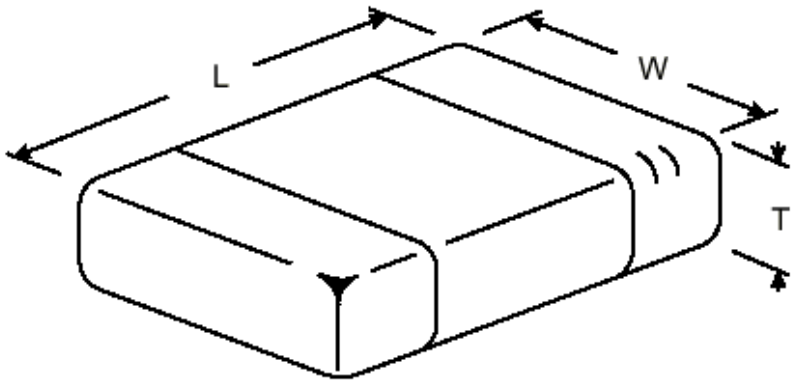
Ap 1.9 Latch 74LCX841WM: Información mecánica



24-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300 Wide
Package Number M24B

Figura Ap 1.14 Dimensiones físicas del latch 74LCX841WM .

Ap 1.10 10nF Capacitor: Información mecánica

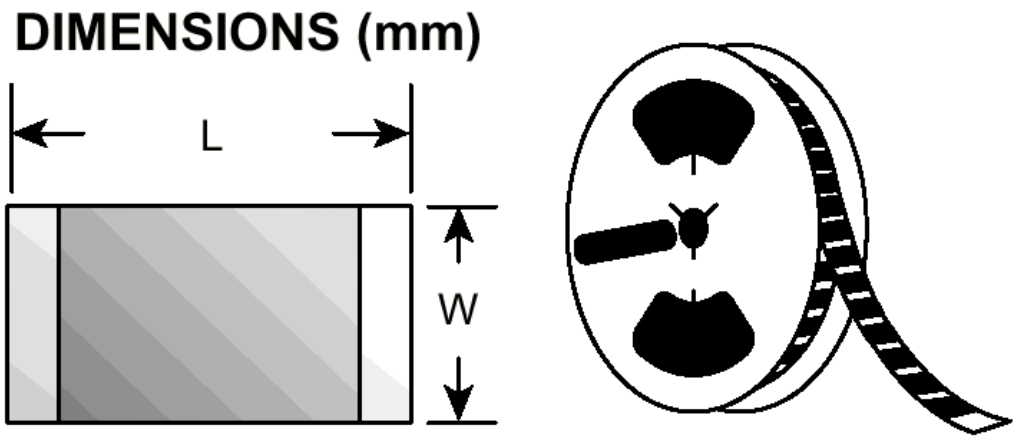


Case Size Dimensions (In.)

Style	L	W	T
VJ0603	.063±.005	.031±.005	.035 Max.
VJ0805	.079±.008	.049±.008	.051 Max.
VJ1206	.126±.008	.063±.008	.059 Max.

Figura Ap 1.15 Dimensiones físicas de los capacitores usados en la tarjeta.

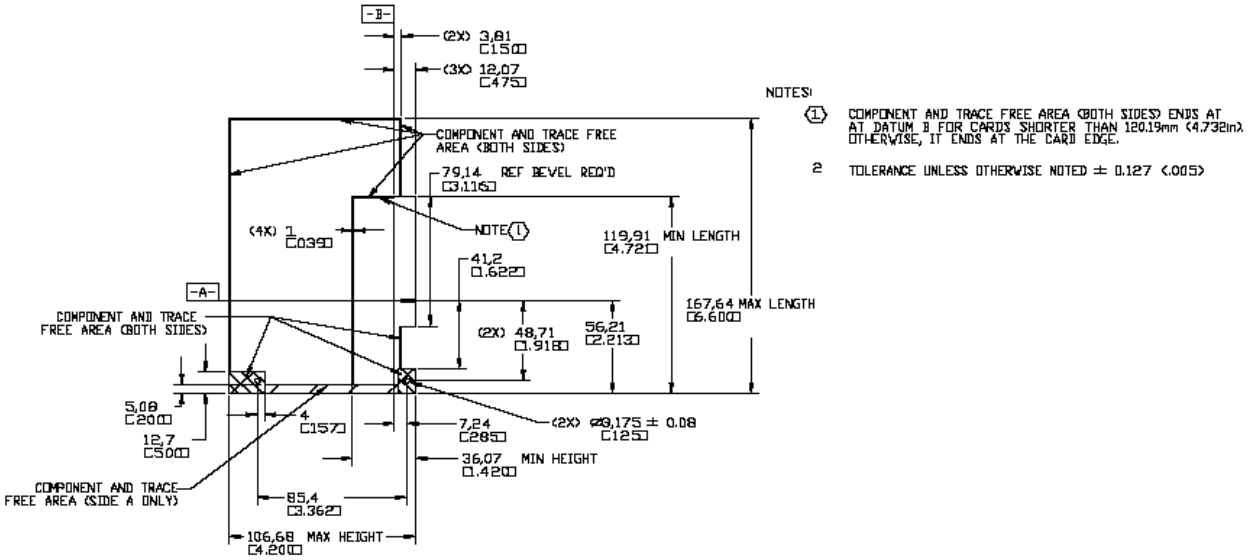
Ap 1.11 Resistencias SMT: Información mecánica



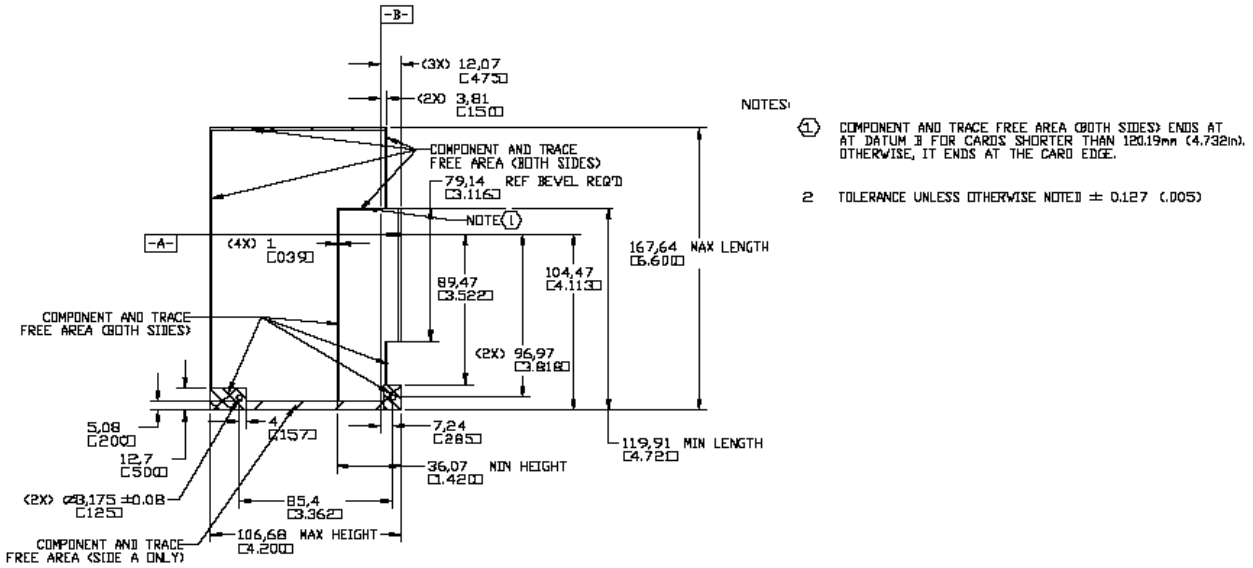
Case Size	Bulk Packs Parts packed on SMD tape.		
	MOUSER STOCK NO.	Dimensions (mm)	
		L	W
805	292-Value	2.03	1.27

Figura Ap 1.16 Dimensiones físicas de las resistencias usadas en la tarjeta.

Ap 1.12 Tarjeta PCI : Información mecánica



PCI Raw Variable Height Short Card (3.3V, 32-bit)



PCI Raw Variable Height Short Card (5V, 32-bit)

Figura Ap 1.17 Dimensiones del estándar de PCI seleccionada para la tarjeta HDLC-4M.

NOTES:

1. Las dimensiones se aplican a la tarjeta de alimentación dual
2. Se usan las dimensiones máximas del estándar.

Tabla Ap 1.3 Dimensiones del estándar PCI seleccionado

	Min.	(*)Max.
L	4.721’ [119.91mm]	6.6’ [167.64mm]
H	1.42’ [36.07mm]	4.2’ [106.68mm]

Apéndice 2 Avances en el diseño del Gerber Data

Ap 2.1 Primer avance presentado por DataLinePCB

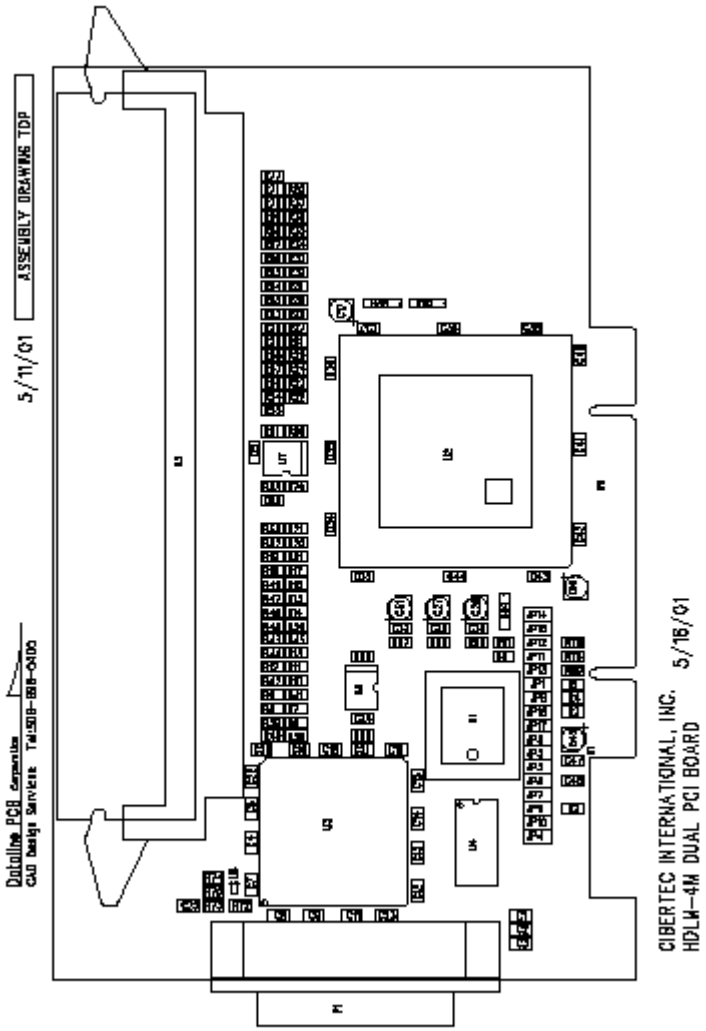


Figura Ap 2.1 Cara frontal de la tarjeta HDLC-4M en el primer avance enviado.

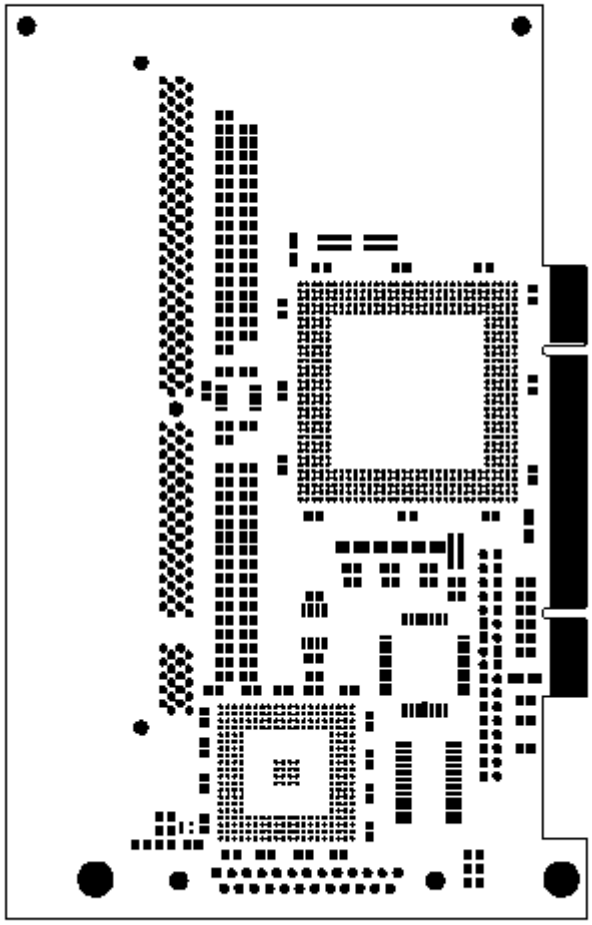


Figura Ap 2.2 Cara posterior de la tarjeta HDLC-4M en el primer avance enviado.

Ap 2.2 Segundo avance presentado por la empresa DataLinePCB

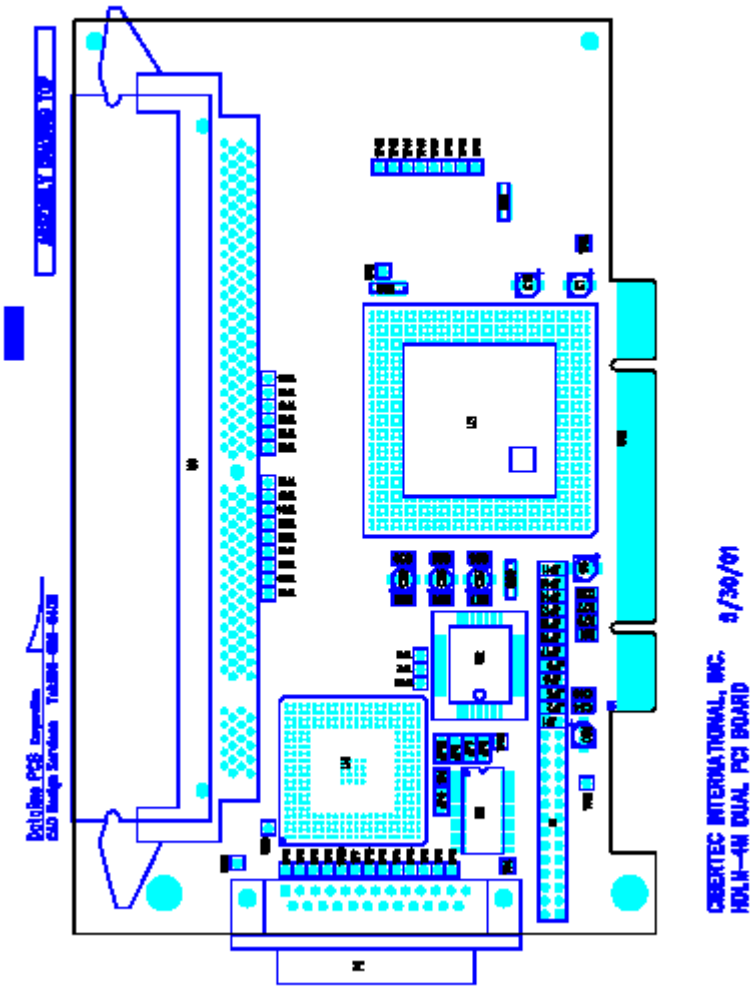


Figura Ap 2.3 Cara frontal de la tarjeta HDLC-4M en el segundo avance enviado.

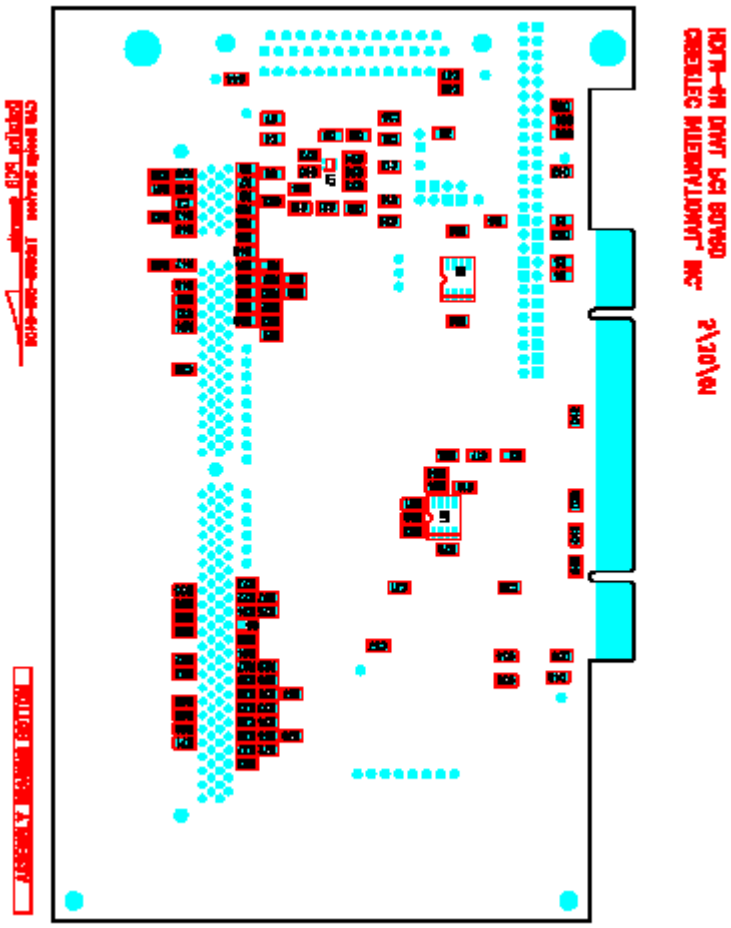


Figura Ap 2.4 Cara posterior de la tarjeta HDLC-4M en el segundo avance enviado.

Ap 2.3 Tercer avance presentado por la empresa DataLinePCB

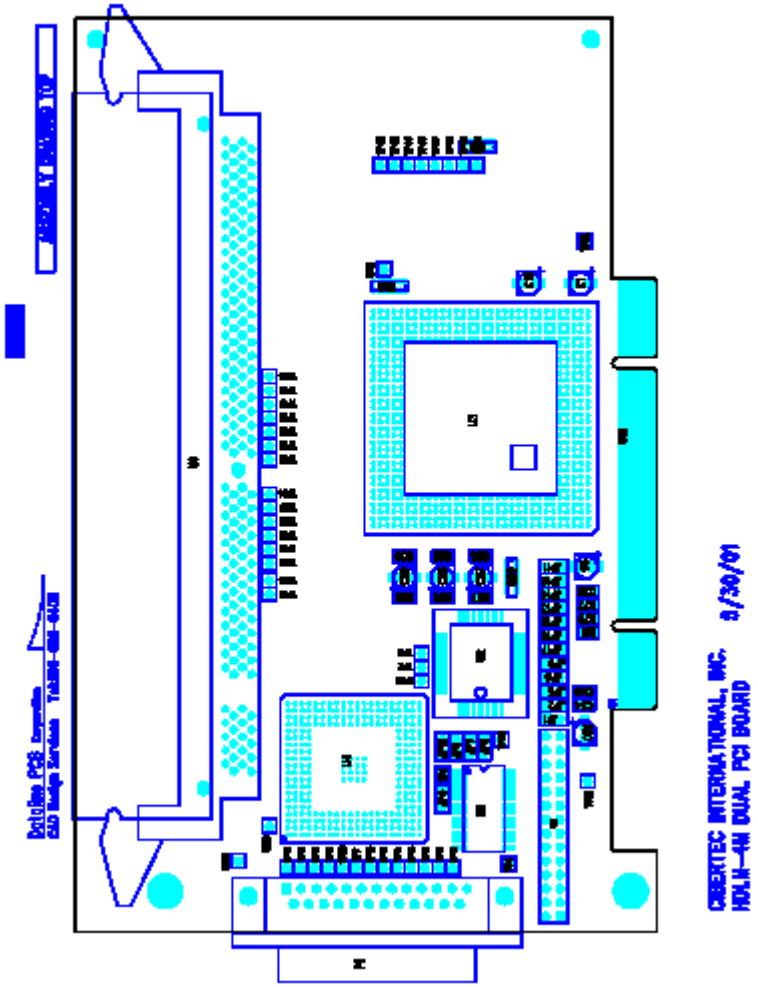


Figura Ap 2.5 Cara frontal de la tarjeta HDLC-4M en el tercer avance enviado.

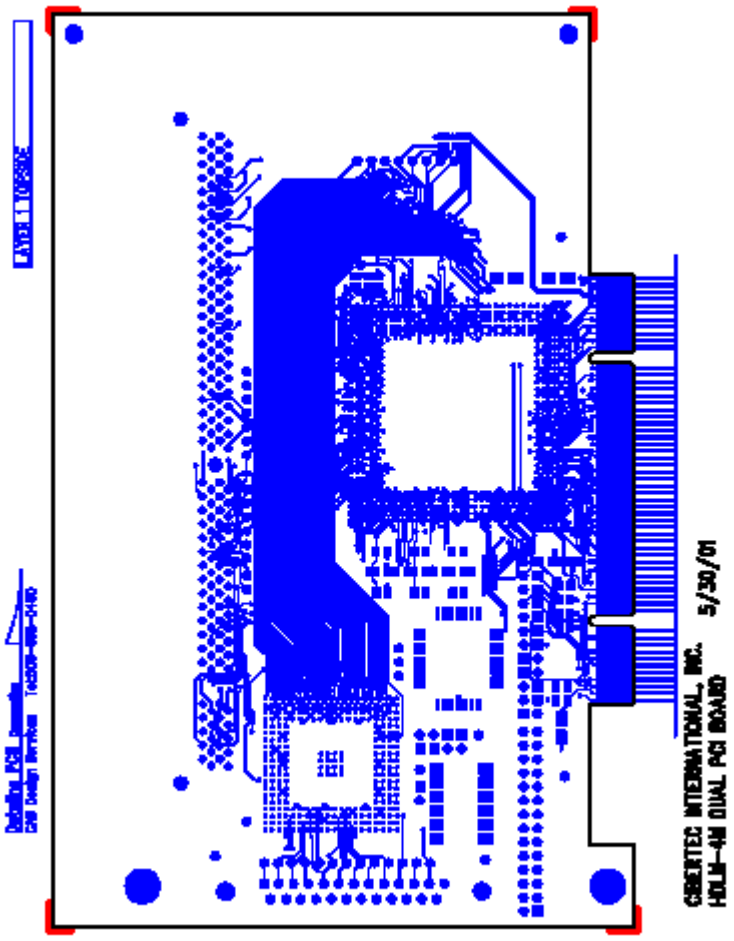


Figura Ap 2.6 Algunas pistas en el plano de señales de la tarjeta HDLC-4M en el tercer avance enviado.

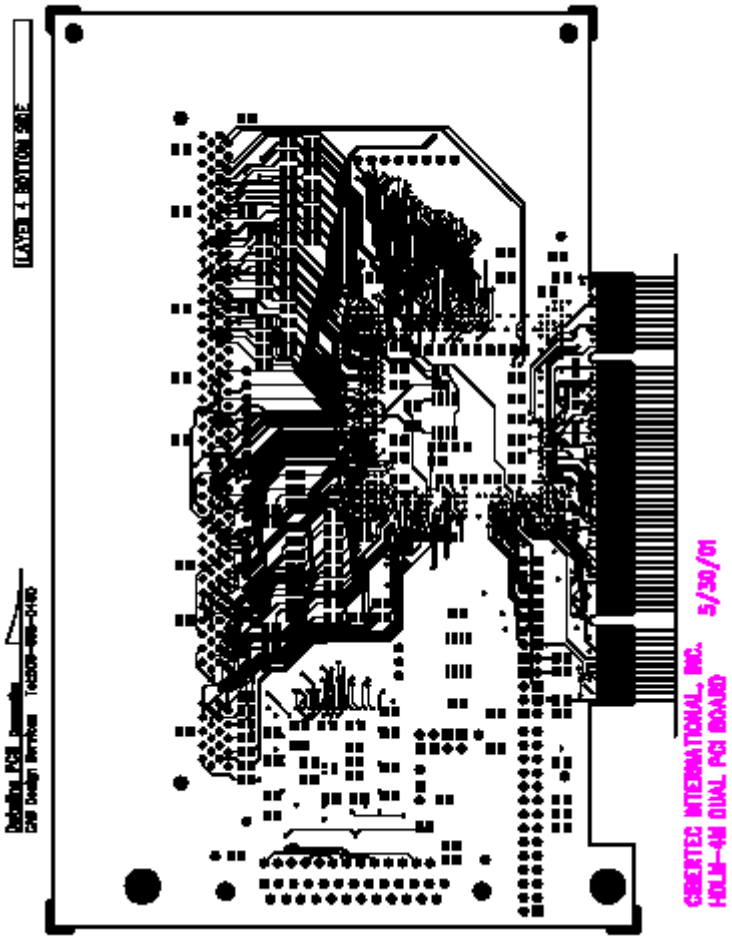


Figura 2.7 Algunas pistas en la tarjeta HDLC-4M enviadas en el tercer avance

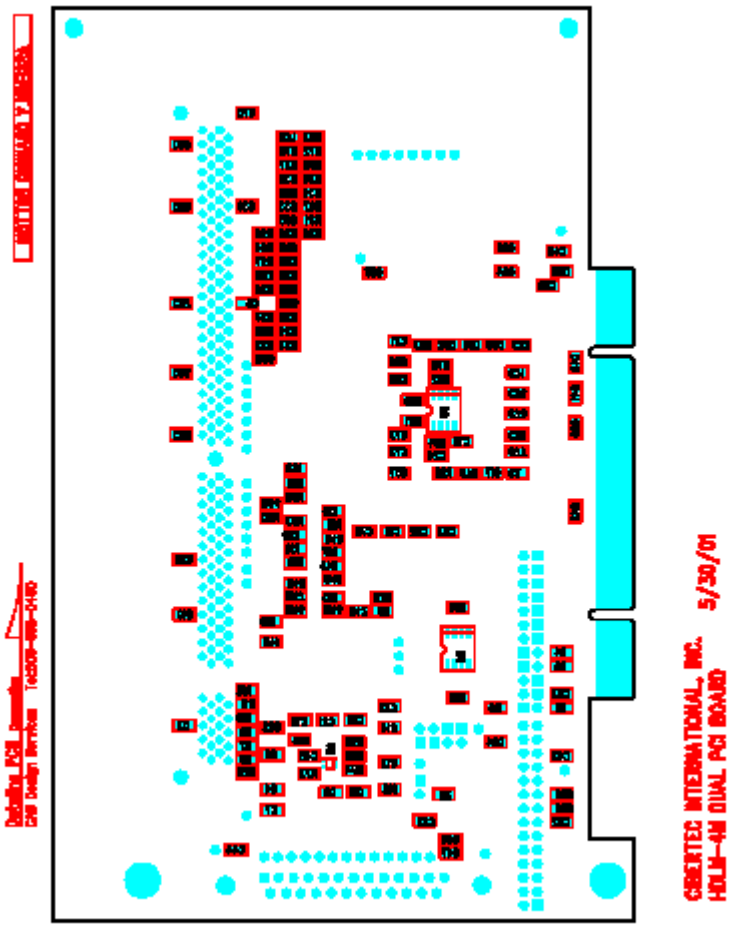


Figura 2.8 Cara posterior la tarjeta HDLC-4M enviadas en el tercer avance

Ap 2.4 Cuarto y final avance presentado por la empresa DataLinePCB

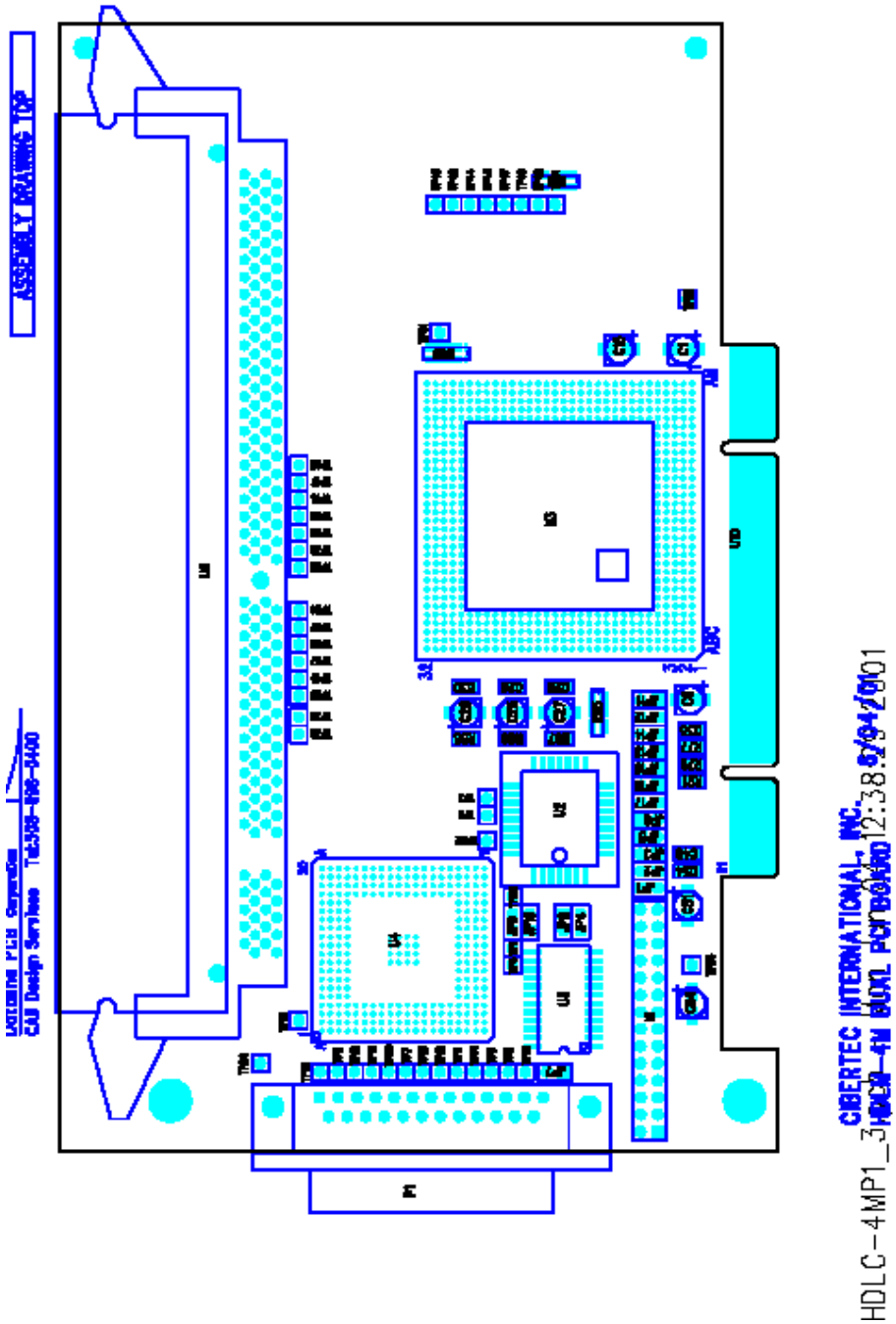


Figura 2.9 Capa de componentes de la tarjeta HDLC-4M enviadas en el cuarto avance

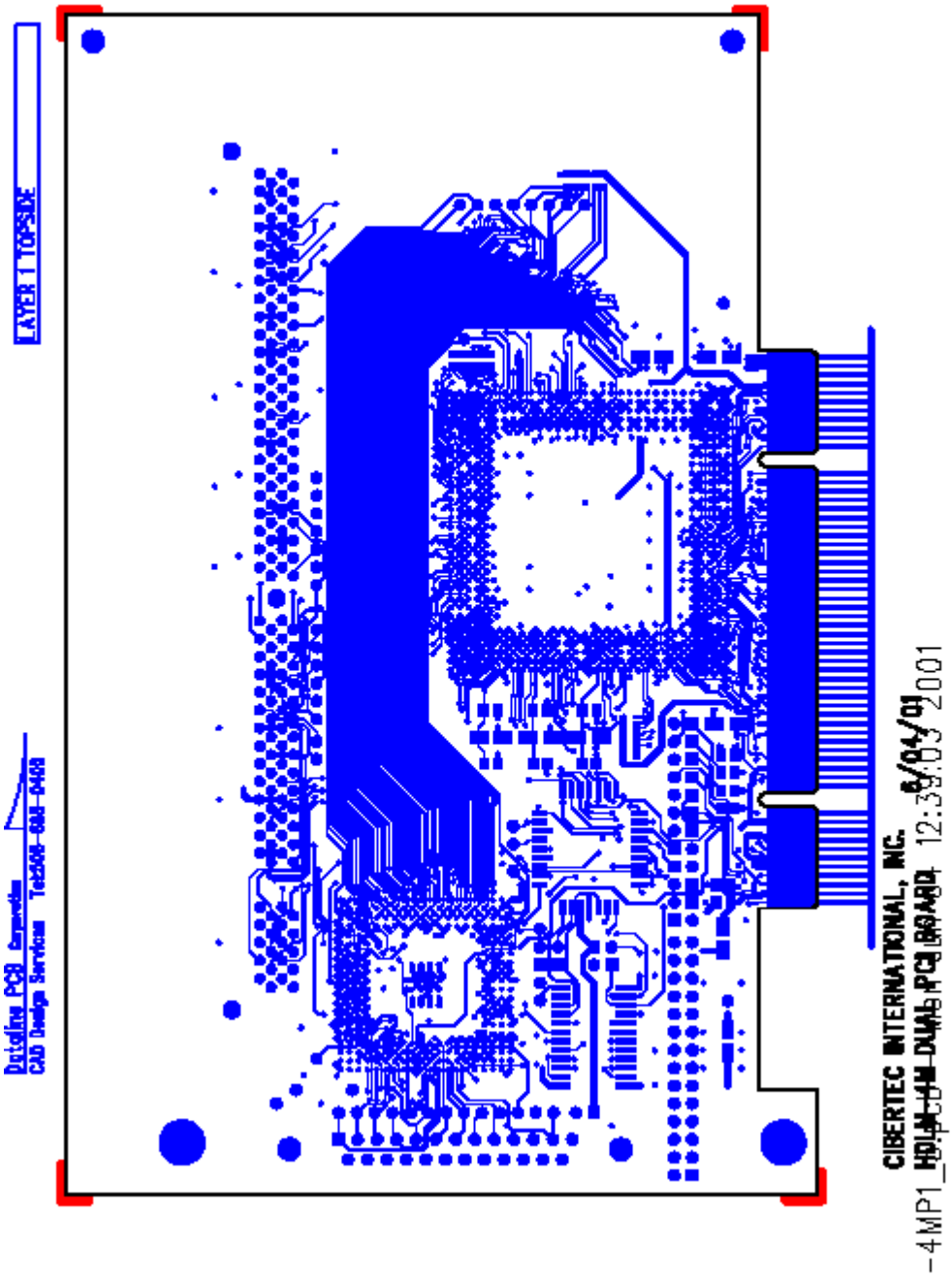


Figura 2.10 Capa de pistas de la tarjeta HDLC-4M enviadas en el cuarto avance

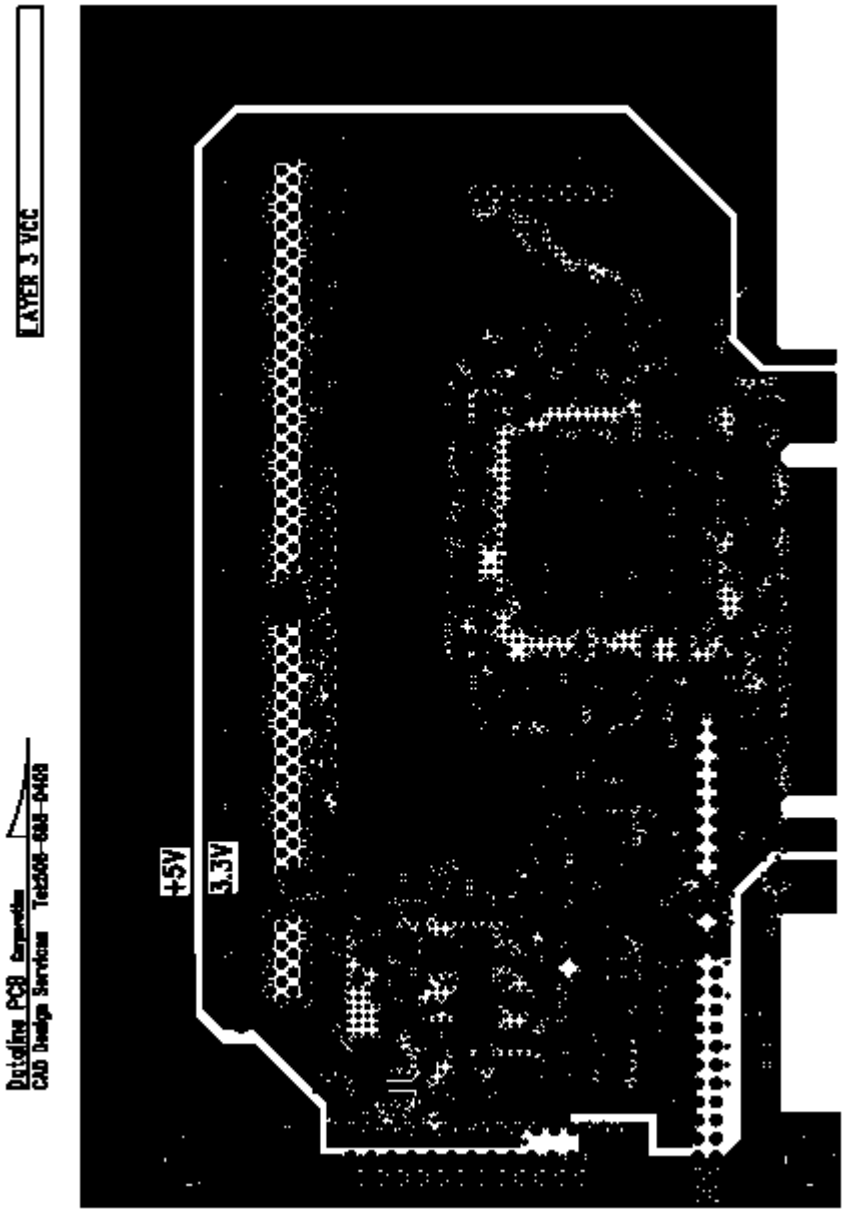


Figura 2.11 Capa de alimentación de la tarjeta HDLC-4M enviadas en el cuarto avance

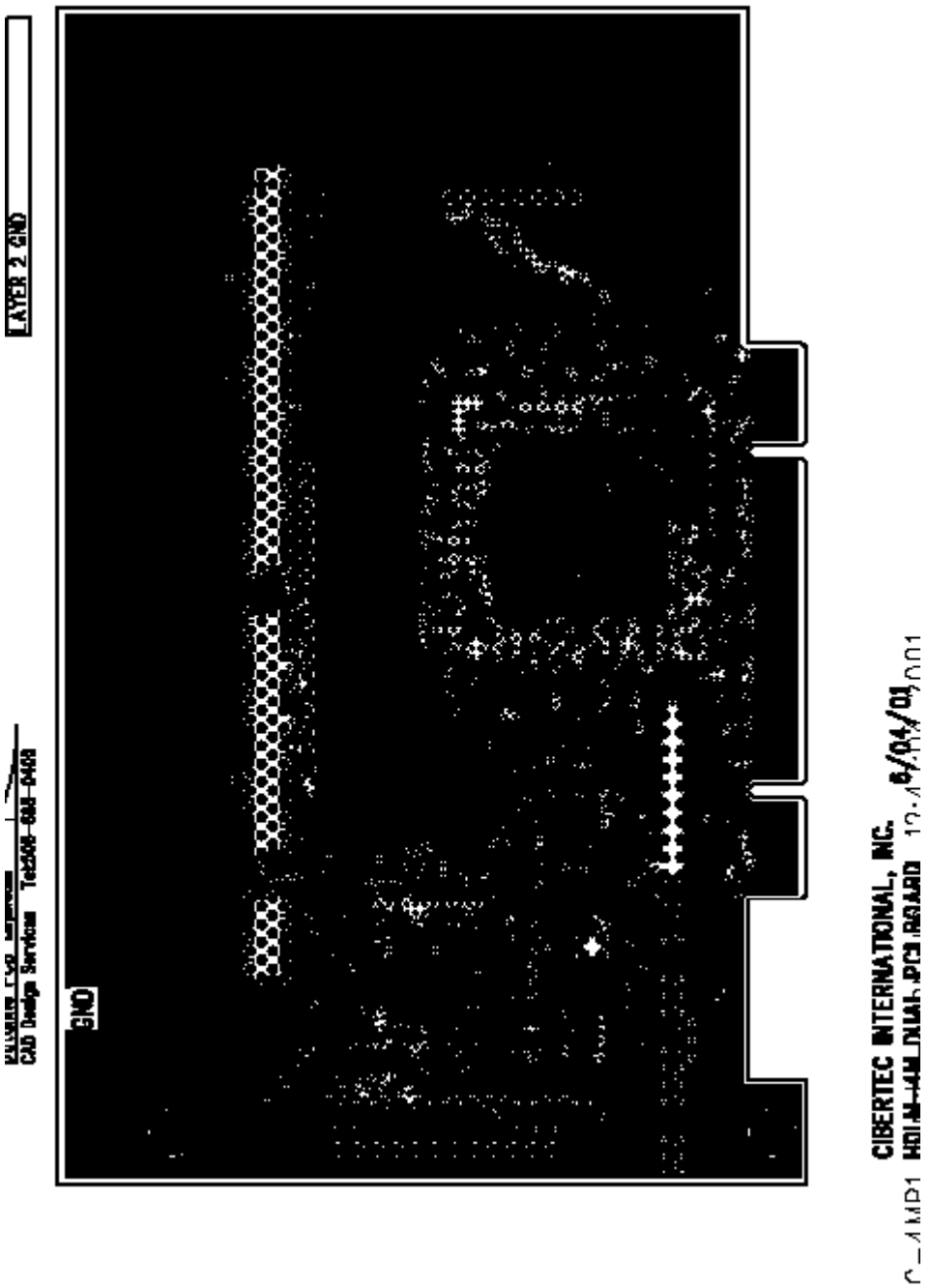


Figura 2.12 Capa de tierra de la tarjeta HDLC-4M enviadas en el cuarto avance

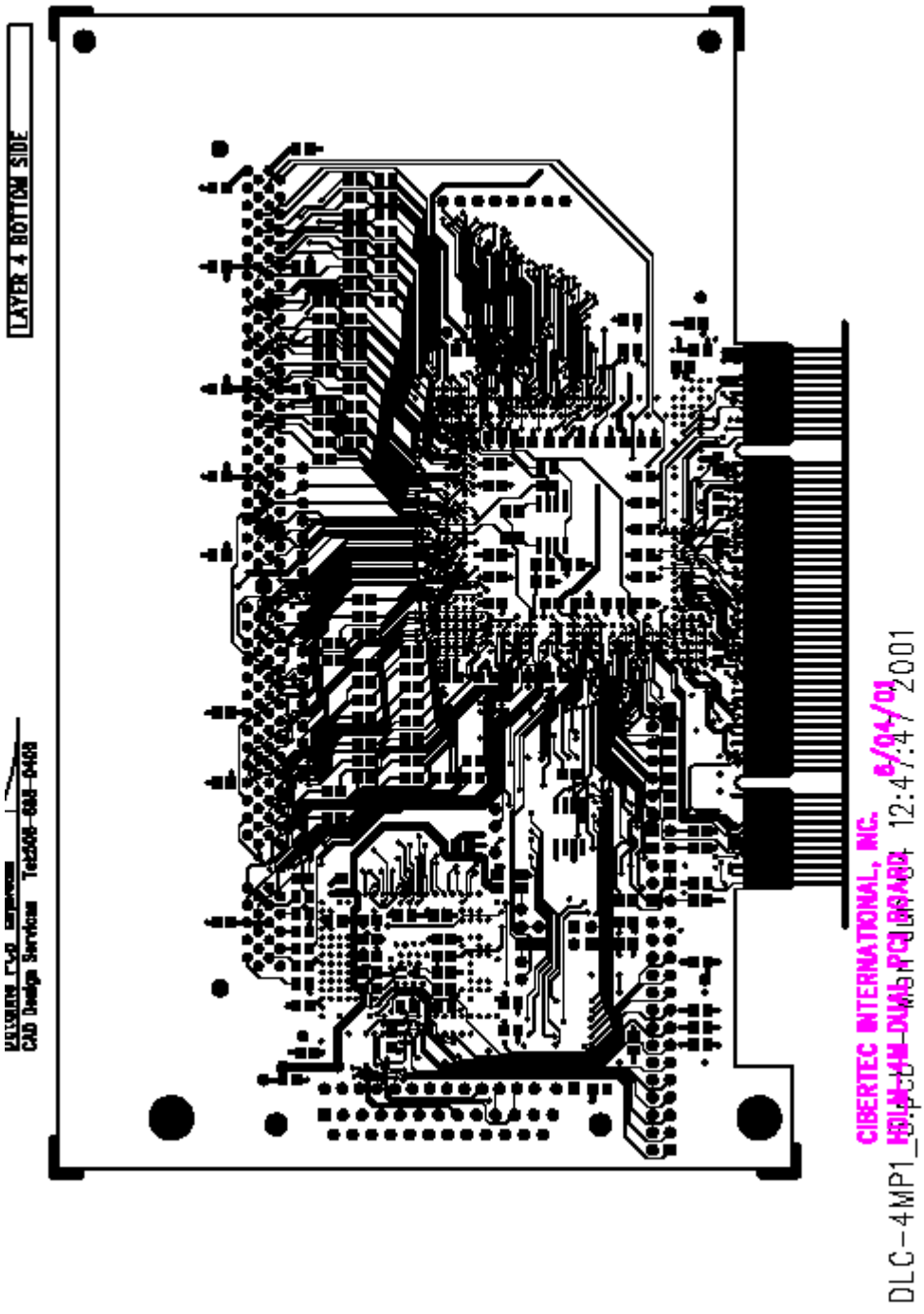


Figura 2.13 Capa de pistas en la tarjeta HDLC-4M enviadas en el cuarto avance

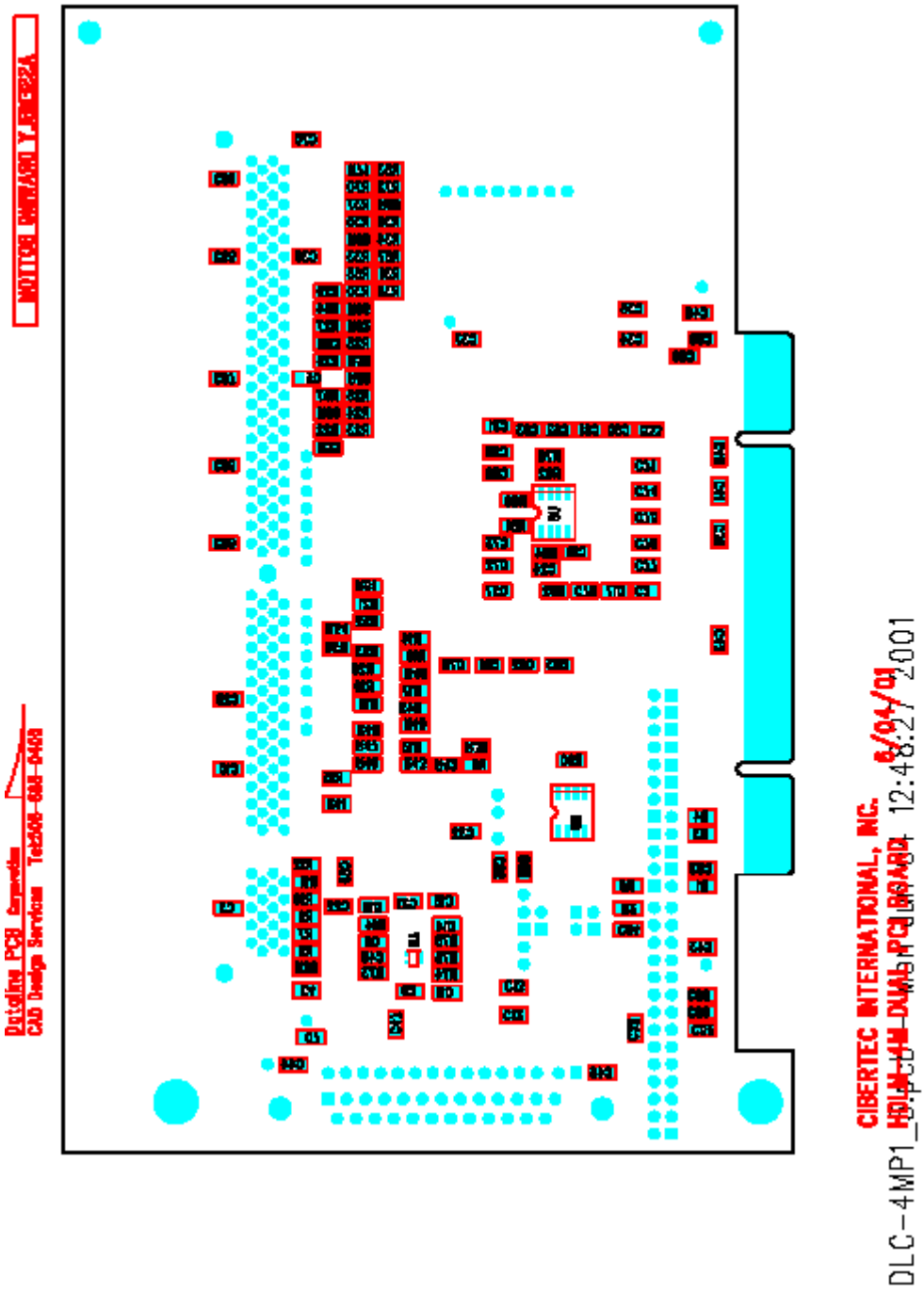


Figura 2.14 Cara posterior de la tarjeta HDLC-4M enviadas en el cuarto avance