

Fiber Distributed Data Interface (FDDI) Databook

FDDI Databook

FDDI DATABOOK

1991 Edition

FDDI Overview

DP83200 FDDI Chip Set

Development Support

Application Notes and System Briefs

Appendix/Physical Dimensions

1

2

3

4

5

TRADEMARKS

Following is the most current list of National Semiconductor Corporation's trademarks and registered trademarks.

ABICTM	Embedded System	Microtalker™	SABRTM
Abuseable™	Processor™	MICROWIRE™	Script/Chek™
Anadig™	E-Z-LINK™	MICROWIRE/PLUSTM	SCXTM
ANS-R-TRAN™	FACT™	MOLE™	SERIES/800™
APPSTM	FACT Quiet Series™	MPATM	Series 900™
ASPECT™	FAIRCAD™	MSTM	Series 3000™
Auto-Chem Deflasher™	Fairtech™	Naked-8™	Series 32000®
BCPTM	FAST®	National®	Shelf/Chek™
BI-FET™	5-Star Service™	National Semiconductor®	Simple Switcher™
BI-FET II™	Flash™	National Semiconductor	SofChek™
BI-LINETM	GENIX™	Corp.®	SONICTM
BIPLANTM	GNXTM	NAX 800™	SPIRE™
BLCTM	GOTM	Nitride Plus™	Staggered Refresh™
BLXTM	HAMRTM	Nitride Plus Oxide™	STARTM
BMACTM	HandiScan™	NML™	Starlink™
Brite-Lite™	HEX 3000™	NOBUSTM	STARPLEXTM
BSITM	HPCTM	NSC800™	Super-Block™
BTL™	I ³ L®	NSCISE™	SuperChip™
CDD™	ICM™	NSX-16™	SuperScript™
CheckTrack™	INFOCHEX™	NS-XC-16™	SYS32™
CIM™	Integral ISETM	NTERCOM™	TapePak®
CIMBUSTM	Intellisplay™	NURAM™	TDSTM
CLASICTM	ISE™	OXISSTM	TeleGate™
Clock/Chek™	ISE/06™	P2CMOSTM	The National Anthem®
COMBO®	ISE/08™	PC Master™	Time/Chek™
COMBO I™	ISE/16™	Perfect Watch™	TINATM
COMBO II™	ISE32™	Pharma/Chek™	TLCTM
COPSTM microcontrollers	ISOPLANARTM	PLAN™	Trapezoidal™
CRD™	ISOPLANAR-Z™	PLANARTM	TRI-CODE™
DA4™	KeyScan™	PLAYERTM	TRI-POLY™
Datachecker®	LMCMOSTM	Plus-2™	TRI-SAFETM
DENSPAK™	M ² CMOSTM	Polycraft™	TRI-STATE®
DIB™	Macrobust™	POSilink™	TURBOTRANSCEIVER™
Digitaltalker®	Macrocomponent™	POSItalker™	VIPTM
DISCERN™	MAXI-ROM®	Power + Control™	VR32™
DISTILL™	Meat/Chek™	POWERplanar™	WATCHDOG™
DNR®	MenuMaster™	QUAD3000™	XMOSTM
DPVMTM	Microbus™ data bus	QUIKLOOK™	XPUTM
E ² CMOSTM	MICRO-DACTM	RAT™	Z STARTM
ELSTAR™	μtalker™	RTX16™	883B/RETSTM
			883S/RETSTM

ABEL™ is a trademark of Data I/O Corporation.

GAL® is a registered trademark of Lattice Semiconductor.

IBM®, PC® and PC-AT® are registered trademarks of International Business Machines Corporation.

PAL® is a registered trademark of and used under license from Advanced Micro Devices, Inc.

UNIX® is a registered trademark of AT&T Bell Laboratories.

XLNT Manager™ is a trademark of XLNT Designs Incorporated.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation 2900 Semiconductor Drive, P.O. Box 58090, Santa Clara, California 95052-8090 1-800-272-9959 TWX (910) 339-9240

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied, and National reserves the right, at any time without notice, to change said circuitry or specifications.

Introduction to FDDI (Fiber Distributed Data Interface)

The role of fiber optics in Local Area Network (LAN) technology is as a high performance supplement to the 10 Mbit/sec IEEE 802.3 Ethernet network. The high performance fiber network of choice will be FDDI as specified by ANSI X3T9.5 Standard.

As computer applications become more complex, they drive the need for high-bandwidth interfaces such as FDDI.

In addition to offering the highest bus speed performance of any standard LAN, FDDI offers a low-cost means for bridging from a conventional speed Ethernet to a high-speed fiber optic link. It works such that a backbone FDDI ring connects local islands of Ethernet workgroups to similar islands located in another part of an interoffice LAN (see *Figure 1*). Data flow across any of the Ethernet configurations—10 Base 2, 10 Base T, twisted-pair—can become bogged down considerably by heavy data transfer activity over an Ethernet backbone network. An FDDI backbone operating at about a 10-times-faster data rate than Ethernet can eliminate this bottleneck by providing a secondary high-speed data highway between these lower-speed workgroups.

FDDI LANs offer higher performance and are overcoming one of their earlier barriers to commercial use: cost-effect-

ive performance. Work has begun to standardize the use of twisted pair over shorter distances.

Because of its favorable cost/performance benefits, high-level technology and backbone capability, FDDI and Ethernet are projected as the network combination of the future.

What FDDI Needs

In order for FDDI to grow into a successful primary and backbone network, computer manufacturers need a standard chip set to make it practical and cost-effective. National Semiconductor, already the leading supplier of fully compliant IEEE 802.3 network integrated circuits for Ethernet, has developed a set of chips to support FDDI, the DP83200 FDDI chip set.

The key to building a family of industry-standard FDDI devices is to constantly work toward higher levels of integration. To this end, National has created semiconductor technologies that are migratable from today's multi-chip designs to tomorrow's single-chip solutions. Further integration to a 2-chip and single chip solution will provide even greater cost and performance benefits to network builders.

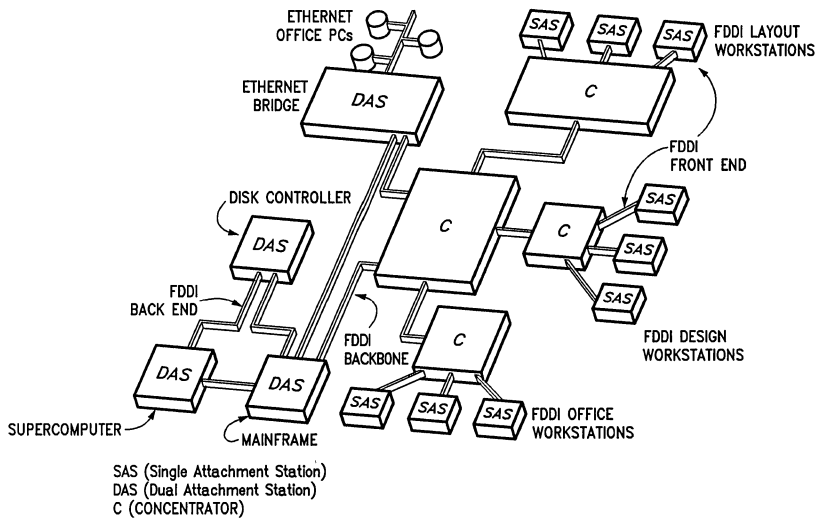


FIGURE 1. FDDI can serve as a high-speed backbone for Ethernet or as a fast network to speed data between high-performance workstations, supercomputers and mainframes.

TL/F/10822-1

Proven By The Process

CMOS is the technology of choice for FDDI ICs. But CMOS varies considerably, according to manufacturer. National's version, called M²CMOSTM, is one of the few types that can accommodate high-performance logic, memory and analog circuitry on the same substrate. National has also developed one of the leading BiCMOS processes in the industry.

Since FDDI data rates are among the highest of any standard data communications application, a BiCMOS process with high-speed bipolar capability and low power consumption will be required in certain parts of the fiber-optic interface circuitry. Packaging will also become a crucial issue in the development of a chip set since it impacts both cost and performance.

Lab Work

As FDDI integrated network products enter the real world of the office or engineering workgroups, system integrators must ensure that all hardware and software components operate correctly with each other. Because of the complexity of the networks, workstations and their software, system integrators must perform interoperability tests. These are designed to verify the proper operation of the complete system.

Unfortunately, assembling all of the hardware and software needed to conduct interoperability testing outside the actual system is extremely difficult. But such testing is so important to chip and system designers that National has set up a dedicated FDDI LAN laboratory that allows its engineers and customers to accelerate their development of ad-

vanced networking products. Also available to customers on request is a document describing the interoperability testing performed by National.

The FDDI LAN lab is equipped with software, computers, printers and cable equipment from a variety of manufacturers. National engineers assist customers by testing the performance of individual components, including National's own networking chip sets, as well as entire systems.

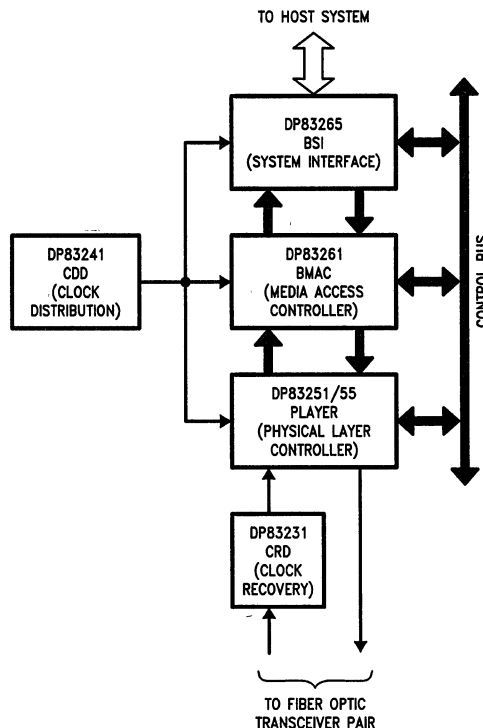
The lab is a test bed for the FDDI chip set. Using the lab facilities, engineers can exercise the devices in all modes of operation to discover any problems before a system user does. If a user does discover a problem—at the chip or system level—the lab provides a resource in which to find a solution.

The Next Generation Solution

Designing and building the integrated circuit portion of FDDI is a great challenge to any semiconductor manufacturer. Not only are processing and packaging expertise essential, the successful chip maker must be well-versed in high-level behavioral modeling, top-down design methodologies, advanced software and CMOS/BiCMOS circuit design. The need for a high-performance fiber optic network is unquestioned. Evaluating hardware implementations is the first step in providing you with the performance and network management capabilities offered by the FDDI protocol.

National Semiconductor, with its long resume of Ethernet leadership and IC excellence, application support materials and experience, is positioned to take the hardware lead in these next-generation FDDI networking solutions.

National DP83200 FDDI Chip Set Block Diagram



TL/F/10822-2



Product Status Definitions

Definition of Terms

Data Sheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This data sheet contains preliminary data, and supplementary data will be published at a later date. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification Noted	Full Production	This data sheet contains final specifications. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

National Semiconductor Corporation reserves the right to make changes without further notice to any products herein to improve reliability, function or design. National does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

Table of Contents

Alphanumeric Index	ix
Section 1 FDDI Overview	
An Overview of FDDI: The Fiber Distributed Data Interface	1-3
Section 2 DP83200 FDDI Chip Set	
DP83231 CRD Device (FDDI Clock Recovery Device)	2-3
DP83241 CDD Device (FDDI Clock Distribution Device)	2-18
DP83251/DP83255 PLAYER Device (FDDI Physical Layer Controller)	2-36
DP83261 BMAC Device (FDDI Media Access Controller)	2-130
DP83265 BSI Device (FDDI System Interface)	2-260
Section 3 Development Support	
DP83200EB FDDI AT Evaluation Kit	3-3
DP83200SMT XLNT Manager FDDI Station Management (SMT) Software Support Package	3-15
Section 4 Application Notes and System Briefs	
AN-675 Designing FDDI Concentrators	4-3
AN-741 Differentiating FDDI Concentrators	4-11
AN-740 Incorporating FDDI MAC Level Bridging	4-28
AN-726 Station Management Simplified	4-36
AN-727 A Guide to the Implementation of Physical Connection Management	4-46
AN-728 FDDI Station Management with the National Chip Set	4-59
AN-736 Interfacing the HPC46064 to the DP83200 FDDI Chip Set	4-66
AN-678 BMAC Device Software Design Guide	4-72
AN-689 BMAC Device Hardware Design Guide	4-78
AN-730 BSI Device Software Design Guide	4-99
AN-674 Layout Recommendations for a System Using National's FDDI Chip Set	4-109
AN-679 Point-to-Point Fiber Optic Links	4-150
SB-107 High-Speed, Point-to-Point, Fiber, Data Communication	4-158
SB-112 FDDI Concentrator	4-160
SB-115 An FDDI-Ethernet Router	4-163
SB-116 FDDI-Adapter Card	4-165
Section 5 Appendix/Physical Dimensions	
FDDI and Networking Acronyms	5-3
Glossary of Terms for FDDI	5-4
Physical Dimensions	5-7
Bookshelf	
Distributors	

Alpha-Numeric Index

An Overview of FDDI: The Fiber Distributed Data Interface	1-3
AN-674 Layout Recommendations for a System Using National's FDDI Chip Set	4-109
AN-675 Designing FDDI Concentrators	4-3
AN-678 BMAC Device Software Design Guide	4-72
AN-679 Point-to-Point Fiber Optic Links	4-150
AN-689 BMAC Device Hardware Design Guide	4-78
AN-726 Station Management Simplified	4-36
AN-727 A Guide to the Implementation of Physical Connection Management	4-46
AN-728 FDDI Station Management with the National Chip Set	4-59
AN-730 BSI Device Software Design Guide	4-99
AN-736 Interfacing the HPC46064 to the DP83200 FDDI Chip Set	4-66
AN-740 Incorporating FDDI MAC Level Bridging	4-28
AN-741 Differentiating FDDI Concentrators	4-11
DP83200EB FDDI AT Evaluation Kit	3-3
DP83200SMT XLNT Manager FDDI Station Management (SMT) Software Support Package	3-15
DP83231 CRD Device (FDDI Clock Recovery Device)	2-3
DP83241 CDD Device (FDDI Clock Distribution Device)	2-18
DP83251 PLAYER Device (FDDI Physical Layer Controller)	2-36
DP83255 PLAYER Device (FDDI Physical Layer Controller)	2-36
DP83261 BMAC Device (FDDI Media Access Controller)	2-130
DP83265 BSI Device (FDDI System Interface)	2-260
FDDI and Networking Acronyms	5-3
Glossary of Terms for FDDI	5-4
SB-107 High-Speed, Point-to-Point, Fiber, Data Communication	4-158
SB-112 FDDI Concentrator	4-160
SB-115 An FDDI-Ethernet Router	4-163
SB-116 FDDI-Adapter Card	4-165



Section 1
FDDI Overview



Section 1 Contents

An Overview of FDDI: The Fiber Distributed Data Interface	1-3
---	-----



An Overview of FDDI: The Fiber Distributed Data Interface

FLOYD E. ROSS

Abstract—Fiber Distributed Data Interface (FDDI), which employs an optical fiber medium, is a 100 Mbit/s local area network (LAN) based on a token ring protocol. Updating earlier efforts on this subject by the author and others, this paper presents an overview of this technology and explores the basis for its success. Particular emphasis is placed on the technical specifications for an upwards-compatible version of FDDI, FDDI-II, which adds the capability for circuit-switched services to the packet services of the basic FDDI, thus creating an integrated services LAN.

I. Introduction

The Fiber Distributed Data Interface (FDDI) is a 100 Mbit/s local area network (LAN). Using optical fiber as the medium, the FDDI protocol is based on a token ring access method. FDDI is being developed in Accredited Standards Committee (ASC) X3T9—chartered to develop computer input/output (I/O) interface standards. In the late 1970's, ASC X3T9 recognized the need for a new I/O channel standard as an alternative to Federal Information Processing Standards (FIPS) 60–63. By late 1982, work had been started on FDDI in the ASC X3T9.5 technical committee. In mid-1983, two proposals encompassing the Physical layer and the Media Access Control for FDDI were submitted by Sperry.

Through this same time period, the Institute of Electrical and Electronics Engineers (IEEE) P802 standards project was developing local area network (LAN) standards with data rates up to 20 Mbits/s. FDDI followed the packet data architectural concepts of IEEE P802 and chose the emerging 4 Mbits/s token ring protocol of IEEE P802.5 as the starting point for the FDDI protocol. These choices placed FDDI in an ideal position to be both the backbone network and the follow-on network to the IEEE P802 LAN's.

Meanwhile, in another standards arena, the Open Systems Interconnection (OSI) model had been put in place. This layered the design of computer interconnections, allowing the development of separate standards for the different layers, and provided the proper framework for the development of a set of FDDI standards.

Another factor of significance in the development of FDDI was an increased use of high-performance video workstations for a variety of applications. These brought an emphasis on facilitating low-cost implementations.

Driven by these forces, FDDI grew from the original Sperry proposals to satisfy the needs of many applications, including the back-end (I/O channel) interface, LAN backbone, and front-end high-performance LAN applications. In the course of its development, each new wave of supporters

brought new demands which, in turn, were accommodated by the emerging FDDI standards. As a result, the set of services offered by FDDI is broad enough to allow individual optimization of FDDI networks to satisfy the needs of diverse environments.

One enhancement, FDDI-II, will offer significantly increased services by integrating circuit-switched data traffic capabilities into what had originally been strictly a packet LAN. The impetus for FDDI-II came from the new generation of digital PBX's of the early 1980's. Their needs were similar to those of many real-time applications including digital voice and video networks as well as sensor and control data streams. All of these disciplines contributed to the emerging FDDI-II design definition which began in late 1984.

A. OUTLINE OF PAPER

Section II of this paper addresses several issues of overall significance to FDDI. Section III describes general FDDI characteristics, with the Physical layer, MAC layer, and SMT operation more fully described in Sections IV, V, and VI, respectively. Section VII supplies an overview of FDDI-II concepts, with Section VIII providing details on FDDI-II operation. The status of the FDDI standardization effort is covered in Section IX.

II. FDDI as the Solution

The widespread acceptance of FDDI has been due to a number of factors. The IEEE P802 effort provided several medium-speed (1–20 Mbit/s) LAN's. In effect, IEEE P802 popularized the LAN, and by doing so, created a market for a higher speed LAN to perform the backbone function for lower speed IEEE 802 LAN's and to satisfy applications that require a higher performance LAN.

Contributing also to the acceptance of FDDI is the dramatic improvement in the price and performance characteristics of optical fiber and related components such as optical transmitters and receivers. With the many other advantages that the use of optical fiber offers—high data bandwidth, security, safety, immunity to electromagnetic interference, and reduced weight and size—the concept of an all-fiber LAN was most attractive. Because the FDDI design has been optimized to the use of optical fiber since its inception, FDDI is now in a leadership role in the development of optical fiber LAN technology.

FDDI has maximized the value of standardization. The true value of standards is being recognized in all marketplaces—from the individual customer to multi-national corporations and the federal government. With emphasis firmly on standards, FDDI has conformed to the ISO Model. The rich functionality that has been integrated into FDDI to meet the needs of a number of market segments means that it can be the one standard satisfying the requirements of the broad, high-speed LAN marketplace. The value of one high-speed optical fiber LAN standard is immense.

Manuscript received May 15, 1988; revised March 1, 1989.
The author is with UNISYS, Malvern, PA 19355.
IEEE Log Number 8929561.

© 1990 IEEE. Reprinted, with permission, from IEEE Journal on Selected Areas in Communications, Volume 7, Issue 7, pps 1043-1051, Sept 1989.

A. ADVANTAGES OF A RING DESIGN

No paper on FDDI would seem complete without at least a cursory summary of the advantages that a ring design offers. A ring can be shown to offer superior reliability, availability, and serviceability, even in the face of physical damage to the network. A ring topology can be designed to be capable of continued operation despite any projected failure.

Other advantages include the interconnect simplicity of the physical hardware at the interface level. The point-to-point connections around the ring not only provide an easy focus of standardization, but also allow different ring links to have different characteristics and optimization points. Optical fiber, which does not adapt well to bus configurations, can be easily accommodated. Optical fiber has sufficient bandwidth that bit-serial transmission may be used, thus significantly reducing the size, cost, and complexity of the hardware required by a network.

Ring topologies offer advantages in the ease of initial configuration and reconfiguration as the network requirements change. Failing stations or fiber links can be isolated through the use of appropriate protocols. These protocols also provide for the logical addition and deletion of stations without detrimental effects on existing ring traffic. Actual physical addition or removal of stations from the network is also facilitated because ring initialization, failure isolation, recovery, and reconfiguration mechanisms can provide for continued operation even while the cables are being rearranged.

Ring topologies inherently impose no restrictive logical limit on the length of ring links, the number of stations, or the total extent of the network that can be accommodated.

Ring topologies, and the protocols supported by them, offer significant performance advantages. These include insensitivity to load distribution, the ease of fairly allocating the available bandwidth, low arbitration times, bounded access delay, and no requirement for long preambles.

In today's technology, a ring topology appears to best satisfy the requirements of high-performance networks operating from 20 to 500 Mbits/s where high connectivity and large extents are required. Some of the references, and in particular [10], are recommended for an in-depth review of the advantages that a ring design offers.

III. FDDI Characteristics

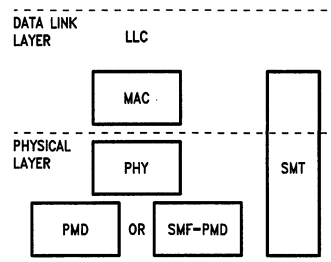
The initial version of FDDI uses optical fiber with light-emitting diodes (LED's) transmitting at a nominal wavelength of 1325 nm over multimode fiber. Connections between stations are made with a dual fiber cable employing a polarized duplex connector. A single-mode fiber (SMF) version of Physical Layer Medium Dependent (PMD) uses laser diode transmitters, with two power levels categories specified, the lower of which retains the same receivers as the basic PMD. SMF-PMD will allow individual links to be extended up to 60 or possibly even 100 km.

The data transmission rate is 100 Mbits/s. The effective sustained data rate at the data link layer can be well over 95 percent of this peak rate. The four out of five code used on the optical fiber medium requires a 125 Mbaud transmission rate. The nature of the clocking, which adjusts for accumulated jitter between frames, limits frames to 4500 octets maximum. Multiple frames may, however, be transmitted on the same access opportunity.

A total of 1000 physical connections and a fiber path of 200 km have been used as the basis for calculation of the default values of the recovery timers. Considering reconfiguration requirements, these choices allow a maximum configuration of 500 stations (each station represents two physical connections) linked by 100 km duplex cable. The choice of longer times than the default values for the recovery timers will allow larger networks to be configured. For smaller networks, performance can be optimized by choosing shorter times for the recovery timers. There is no minimum configuration requirement.

A. STATION ORGANIZATION

Figure 1 shows the component entities necessary for an FDDI station. Identified components, conforming to both the IEEE 802 structure and the OSI concept of layering, are: Station Management (SMT), which specifies the local portion of the network management application process, including the control required for the proper internal configuration and operation of a station in an FDDI ring; Media Access Control (MAC), which specifies the lower sublayer of the data link layer, including the access to the medium, addressing, data checking, and data framing; Physical Layer Protocol (PHY), which specifies the upper sublayer of the physical layer, including the encode/decode, clocking and framing for transmission; and Physical Layer Medium Dependent (PMD), which specifies the lower sublayer of the physical layer, including power levels and characteristics of the optical transmitter and receiver, interface optical signal requirements, the connector receptacle footprint, the requirements of confirming optical fiber cable plants, and the permissible bit error rates. Two alternative versions of PMD are shown; the basic PMD and SMF-PMD which allows the use of single-mode optical fiber.



TL/F/10827-1

FIGURE 1. FDDI Relationship to OSI Model

FDDI MAC provides a superset of the services required by the Logical Link Control (LLC) protocol developed by IEEE P802.2. Figure 1 assumes the use of the IEEE 802.2 LLC as the upper sublayer of the data link layer. Any other appropriate LLC may be used.

These basic component entities allow a variety of station types as shown in the FDDI topology example of Figure 2. As shown, the FDDI trunk ring consists of a pair of counter-rotating rings. Two main classes of stations, depending on whether or not they are allowed to attach directly to the trunk ring, as specified. A dual attachment station has two PHY entities. It may attach directly to the trunk ring or indirectly via a concentrator. It may have one or more MAC entities. In the case of two MAC entities, one may be in each of the counterrotating rings or both may be in the same ring. A dual attachment station may have optical bypass switches to remove it from both rings, at the same time healing them, if the station is powered down or disabled by SMT.

A single attachment station has one PHY and one MAC, and therefore cannot be attached directly into the main FDDI ring. Instead, it must be attached to the ring via a concentrator. A concentrator is a special station that has extra PHY's used to attach other stations that are to be inserted into the main FDDI ring. Varying levels of functionality, including multiple MAC's, are permitted in concentrators.

In Figure 2 stations 1, 2, and 4 are dual attachment stations. Station 3 is a concentrator connecting stations 4, 5, and 6 into the FDDI ring. Stations 5 and 6 are single attachment stations. Station 4, even though it is a dual attachment station, must perform as a single attachment station insofar as its attachment to the ring via the concentrator is concerned. Operation of the second PHY of station 4 on some other ring is allowed, but there must not be any interconnection of the two rings within this station at a level visible to MAC or PHY.

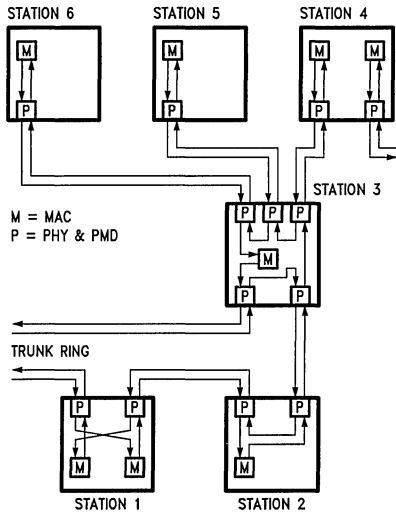


FIGURE 2. FDDI Topology Example

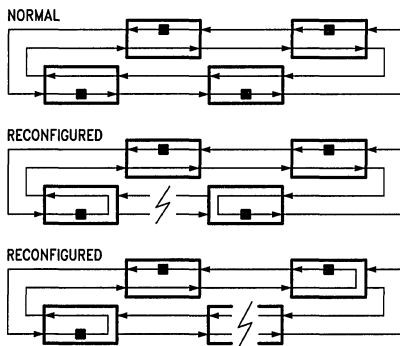


FIGURE 3. Reconfiguration of Counterrotating Rings

B. RELIABILITY PROVISIONS

Ring topologies allow for the isolation of failing attachments through several mechanisms. Counterrotating rings, as shown in Figure 3, are basic to the FDDI structure. The counterrotating ring concept uses two rings connected to

each station or concentrator—one clockwise and the other counterclockwise. When a failure in a link occurs, the stations on either side reconfigure internally as shown in the middle diagram. The functional stations adjacent to the break make use of the connection in the reverse direction to close the ring, thus eliminating the bad link. In this figure, the dark squares represent the logical (MAC) attachment within the stations. Should a station itself fail, as shown in the bottom diagram of this figure, the stations on either side reconfigure to eliminate the failing station and both of the links to it.

Stations may offer a bypass capability, whereby an optical switch is used to bypass a station's receiver and transmitter connections so that the signal from the previous station is passed directly to the next station. Bypassing may be activated by a station itself, at the instigation of its neighbor, by a human operator, automatically at the removal of power, or by some overall network-controlling function.

Yet another approach is the use of concentrators, as shown in Figure 2, that are attached directly to the trunk ring and, in turn, provide drop connections for a number of stations—in this case, stations 4, 5, and 6. A concentrator may then monitor all of its slave stations and isolate any faulty station. It may also provide for the graceful logical insertion and removal of stations from the ring.

TABLE I. Symbol Coding

Decimal	Code Group	Symbol	Name	Assignment
00	00000	Q	QUIET	Line State Symbol
31	11111	I	IDLE	Line State Symbol
04	00100	H	HALT	Line State Symbol
24	11000	J		Starting Delimiter
17	10001	K		Starting Delimiter
05	00101	L		Starting Delimiter
13	01101	T		Ending Delimiter
07	00111	R	RESET	Control Indicator
25	11001	S	SET	Control Indicator
30	11110	0		Data Symbol 0000
09	01001	1		Data Symbol 0001
20	10100	2		Data Symbol 0010
21	10101	3		Data Symbol 0011
10	01010	4		Data Symbol 0100
11	01011	5		Data Symbol 0101
14	01110	6		Data Symbol 0110
15	01111	7		Data Symbol 0111
18	10010	8		Data Symbol 1000
19	10011	9		Data Symbol 1001
22	10110	A		Data Symbol 1010
23	10111	B		Data Symbol 1011
26	11010	C		Data Symbol 1100
27	11011	D		Data Symbol 1101
28	11100	E		Data Symbol 1110
29	11101	F		Data Symbol 1111
01	00001	V	Violation	Not Transmitted
02	00010	V	Violation	Not Transmitted
03	00011	V	Violation	Not Transmitted
06	00110	V	Violation	Not Transmitted
08	01000	V	Violation	Not Transmitted
12	01100	V	Violation	Not Transmitted
16	10000	V	Violation	Not Transmitted

The use of all three techniques allows FDDI networks to be configured to tolerate a variety of station or link failures and physical network configurations without catastrophic consequences. When failures occur, the network automatically reconfigures, eliminating any failing element and maintaining ring operation. Continuous monitoring of the failed link or station allows the network to automatically reconfigure and restore normal operation when repair is effected. Any of these reconfigurations may result in the loss of individual frames, which then need to be retransmitted.

C. DATA ENCODING

Information on the medium uses a four out of five group code, with each code group called a symbol. As shown in Table I, in the 32-member symbol set, 16 symbols are data symbols, each representing four bits of ordered binary data. Three symbols are used for line-state signaling which is recognized by the physical layer hardware, two are used as control indicators, and four are used for starting and ending delimiters. One of the starting delimiters (L) is not used in the basic FDDI, but is reserved for FDDI-II. The remaining seven symbols of the symbol set are not to be transmitted since they violate code run length and dc balance requirements. The QUIET symbol is a necessary member of the line-state symbol set since it is used to indicate the absence of any functional signal.

D. FRAME AND TOKEN FORMATS

Information is transmitted on the FDDI ring in frames which are variable in length. Tokens are special short fixed-length "frames" that are used to signify the right to transmit data. Figure 4 shows the frame and token formats.

The Preamble (PA) field, consisting nominally of 16 IDLE symbols (a maximum frequency signal that is used for establishing and maintaining clock synchronization), precedes every transmission. The Starting Delimiter (SD) field consists of a two-symbol sequence (JK) that is uniquely recognizable independent of previously established symbol boundaries. The SD establishes the symbol boundaries for the content that follows.

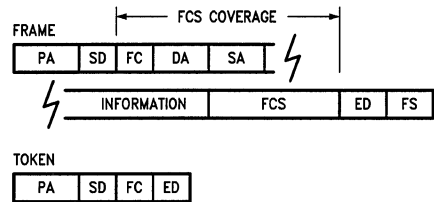
The Frame Control (FC) is a two-symbol field that defines the type of frame and its characteristics. It distinguishes between synchronous and asynchronous frames, the length of the address field (16 or 48 bits), and the kinds of frame (e.g., LLC or SMT). One set of FC values is reserved for implementer frames that have no defined format and are to be repeated unchanged by all conforming FDDI stations. The FC field also provides for two kinds of tokens, restricted and unrestricted. The latter is used in a special class of service which provides for extended dialogs among a limited set of cooperating stations. Two Ending Delimiter (ED) symbols (TT) complete a token.

The Destination Address (DA) and Source Address (SA) fields may be either 16 or 48 bits long, depending on the FC value. DA may be either an individual or a group address, the latter of which has the potential to be recognized by more than one station.

The 32-bit Frame Check Sequence (FCS) field is a cyclic redundancy check using the standard polynomial used in the IEEE 802 protocols. The information field of a frame, like the other fields covered by the FCS check, consists only of data symbols. Data symbols are not used in fields not covered by the FCS check.

The ED field of a frame is one delimiter symbol (T). It is followed by the Frame Status (FS) field that has a minimum of three control indicator symbols that are modified by the

station as it repeats the frame. These indicate, when set, that an error has been detected in the frame by the station, that the addressed station has recognized its address, and that the frame has been copied by the station.



TL/F/10827-4

FIGURE 4. Frame and Token Formats

IV. Physical Layer (PHY and PMD) Operation

PHY provides the protocols and PMD the optical fiber hardware components that support a link from one FDDI station to another. PHY simultaneously receives and transmits. The transmitter accepts symbols from MAC, converts these to five-bit code groups, and transmits the encoded serial data stream on the medium.

The receiver recovers the encoded serial data stream from the medium, establishes symbol boundaries based on the recognition of a start delimiter, and forwards decoded symbols to MAC. Additional symbols (QUIET, IDLE, and HALT) are interpreted by PHY and used to support SMT functions.

PHY also provides the bit clocks for each station. The total ring, including all stations and links, must remain the same apparent bit length (i.e., no bits may be created or deleted) during the transmission of a frame around the ring. Otherwise, an error would be generated in the frame as it is repeated around the ring. In the face of jitter, voltage, temperature, and component aging effects, such stability can only be realized through special provision. PHY provides an elasticity buffer which is always inserted between the receiver and the transmitter. The receiver employs a variable frequency clock, using standard techniques such as a phase-locked loop oscillator, to recover the clock of the previous transmitting station from the received data. The transmitter, in contrast, uses a local fixed-frequency clock. The elasticity buffer in each station compensates for the difference in frequency between the local clock and that of the upstream station by adjusting the bit delay through the station. The elasticity buffer in each station is reinitialized to its center position during the preamble (PA) that precedes each frame or token. This has the effect of increasing or decreasing the length of the PA, which is initially transmitted as 16 or more symbols, as it proceeds around the ring.

The transmitter clock has been chosen with 0.005 percent stability. With a minimum elasticity buffer of 10 bits, frames of up to 4500 octets in length can be transmitted without exceeding the limits of the elasticity buffer.

Simulations of early FDDI designs, presented to the technical committee, showed that rather than maintaining a nominal 16 symbol PA length, there was a tendency for the PA lengths in long rings to move toward a flat unbounded distribution. Longer PA's created no problem by themselves, but came at the expense of shorter, or even negative, length PA's. A negative PA indicates that the PA has been so severely shortened that it has completely disappeared and that symbols have consequently been lost off the end of a frame, resulting in the loss of the entire frame. This problem

was solved by means of a smoothing buffer function incorporated into PHY. This examines the PA length between frames and either inserts or deletes, as the case may be, preamble symbols (or bytes) in order to maintain the PA near the nominal 16 symbol length, thus ensuring that the preamble never decreases below 4 or 5 bytes. Simulation work presented to the technical committee showed that the algorithm chosen, even under worst case conditions, reduced the probability of frame loss to less than 10^{-12} . Later work presented to the technical committee has reaffirmed this result with extended testing of a large physical ring configuration.

V. Token MAC Functional Operation

A major function of any station is deciding which station has control of the medium. MAC schedules and performs all data transfers on the ring.

The basic concept of a ring is that each station repeats the frame that it has received from its upstream neighbor to its downstream neighbor. If the destination address (DA) of the frame matches that MAC's address and there is no error indicated, then the frame is copied into a local buffer with MAC notifying LLC (or SMT) of the frame's arrival. MAC modifies the indicator symbols in the FS field of the frame as it repeats it to indicate the detection of an error in the frame, the recognition of its own address, and the copying of the frame. The frame propagates around the ring to the station that originally placed it on the ring. The transmitting station may examine the indicator symbols in the FS field to determine the success of the transmission. The MAC of this transmitting station is responsible for removing from the ring all of the frames that it has placed on the ring (a process termed stripping). MAC recognizes these frames for stripping by the fact that the SA contained in them is its own address. IDLE symbols are placed on the medium during stripping.

If MAC has a frame from LLC (or SMT) to transmit, it may do so only after a token has been captured. A token is a special frame that indicates that the medium is available for use. Priority requirements, necessary to assure the proper handling of frames, are implemented in the rules of token capture. Under these rules, if a given station is not allowed to capture the token, then it must repeat it (or in certain cases reissue a token) to the next station in the ring. Only after having captured a token and stripping it from the ring is MAC allowed to transmit a frame or frames. When finished, MAC issues a new token to signify that the medium is available for use by another station.

The FDDI MAC uses a Timed Token Rotation (TTR) protocol to control access to the medium. Under this protocol, each station measures the time that has elapsed since a token was last received. The initialization procedures establish the Target Token Rotation Time (TTRT) equal to the lowest value that is bid by any of the stations. Two classes of service are defined. Synchronous service allows use of a token whenever MAC has synchronous frames queued for

transmission. Asynchronous service allows use of a token only when the time since a token last was received has not exceeded the established TTRT. Multiple levels of priority for asynchronous frames may be provided within a station by specifying additional (more restrictive) time thresholds for token rotation.

The use of the TTR protocol imparts some useful operational characteristics. It allows stations to request and establish (via SMT procedures) guaranteed bandwidth and response time for synchronous frames. It establishes a guaranteed minimum response time for the ring because, in the worst case, the time between the arrival of two successive tokens will never exceed twice the value of TTRT. It also provides a guaranteed level of ring utilizations equal to $(TTRT-RL)/TTRT$ where RL is the physical ring latency—essentially the time for a token to propagate around the ring under no load conditions. Reference^[15] and others by the same author have dealt extensively with these aspects of FDDI operation.

Low values of TTRT (e.g., 4 ms) may be used to establish an average token rotation time of 4 ms and a guaranteed response time not exceeding 8 ms. This would be useful in a time-critical application (e.g., packetized voice). Larger value of TTRT may be used to establish very high ring utilizations under heavy loads. For instance, using TTRT of 50 ms and a ring latency (RL) of 0.25 ms (reasonable for a ring consisting of 75 stations and 30 km of fiber), the above formula shows that a utilization of 99.5 percent can be achieved.

VI. Station Management (SMT) Operation

SMT is the local portion of the network management application process, including the control required for proper operation of an FDDI station in an FDDI ring. SMT monitors activity and exercises overall control of station activity. These functions include control and management within a station for such purposes as initialization, activation, performance monitoring, maintenance, and error control. Additionally, SMT communicates with other SMT entities on the network for the purpose of controlling network operation. Examples of these SMT functions include the administration of addressing, allocation of network bandwidth, and network control and configuration.

The SMT Connection Management (CMT) function establishes the physical connections between adjacent stations. For this function, CMT uses stream composed of QUIET, HALT, and IDLE symbols in low-level signaling protocols. Once a physical connection is established, CMT creates a logical configuration within the station by activating the appropriate paths between the PHY and MAC entities within that station. A large degree of flexibility is provided in the logical configuration, which may be established consistent with station functionality and desired station personality. This flexibility allows FDDI to support a wide range of topologies and applications.

VII. FDDI-II Concepts

FDDI-II is an upward-compatible enhancement of the basic FDDI that adds a circuit-switched service to the existing packet capability.

A packet service is a service where the elements of data to be transferred are placed in frames. Packets may vary in length and are self-defining in that each contains delimiters that mark its beginning and end and an address that specifies the target station. FDDI packets are called frames.

In contrast, a circuit-switched service provides a continuous connection between two or more stations. Instead of using addresses, the connection is established based upon some prior agreement, which may have been negotiated using packet messages or established by some other suitable convention known to the stations involved. This prior agreement typically takes the form of knowing the location of a time slot, or slots, that occur regularly relative to a readily recognizable timing marker.

A common timing marker used in North America is the Basic System Reference Frequency (BSRF), a 125 μ s clock used by the public networks. Use of this clock is assumed for FDDI-II. In local FDDI usage, this is referred to as the cycle clock and is signaled by the JK starting delimiter of the FDDI-II cycle format.

In FDDI-II, a circuit-switched connection is described as N bits beginning at byte M after the cycle clock marker in Wideband Channel (WBC) number X. The last descriptor is necessary because FDDI-II has 16 WBC's that may be independently assigned to either packet-switched or circuit-switched data. This definition allows connections at data rates of all multiples of 8 kbits/s (i.e., $N = 1$) up to the 6.144 Mbit/s data rate of a WBC. If need be, multiple WBC's may be used to accommodate higher data rates.

The data transferred in a circuit-switched mode is best described as a stream of data. The data rate is appropriate to the service being provided—with, for example, 64 kbits/s being used for a digital voice data stream. Other data stream rates, even up to many Mbits/s in the case of video, are used for other applications. Once a connection is established, the data rate remains constant.

The contrasting nature of packet-switched and circuit-switched data is of interest. Most packet data traffic occurs in random quantities and at random times. This is referred to as asynchronous traffic. Other packet traffic, more regular in nature and occurring in relatively predictable quantities on a regular time basis, is referred to as synchronous (packet) traffic. In contrast, isochronous data occur in precise amounts on a precise time basis. They typically represent a sequence of digital samples from a sensor (e.g., voice or video). More importantly, isochronous data must be synchronized with clock information to ensure the accurate regeneration of the sampling clock (as distinct from the bit clock) to minimize distortion in data reconstruction. Isochronous data are more easily transferred in a circuit-switched network.

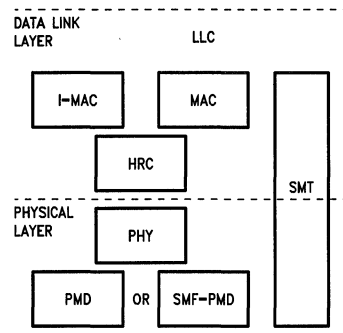
Networks that carry isochronous data must maintain precise synchronism with the cycle clock. For the FDDI ring, this means that one station (called the cycle master) must insert a delay, for all isochronous data so that the ring appears to be an exact multiple of 125 μ s in length. FDDI incorporates this delay in the cycle master in such a way that it does not

cause any delay in the packet traffic. This is essential in providing an integrated services network with acceptable packet service.

VIII. FDDI-II Operation

Figure 5, much like Figure 1, shows how FDDI-II is implemented using one additional standard—Hybrid Ring Control (HRC). HRC becomes the new lowest sublayer of the data link layer, taking its place between MAC and PHY. HRC multiplexes data between the (packet) MAC and the isochronous MAC (I-MAC). This requires that the (packet) MAC be able to transmit and accept data on a noncontinuous basis because packet data are interleaved with isochronous data.

FDDI-II is a network with 100 Mbits/s of bandwidth available. This bandwidth may be devoted totally to operation as a packet network. Alternatively, portions of this bandwidth, in units of WBC's, may be dynamically separated for use with circuit-switched data. Up to 16 WBC's may be assigned. Each WBC is 6.144 Mbits/s, which is four times the North American and three times the European basic access rate to the telephone network. WBC's are full duplex and are independently allocatable and deallocatable. In effect, a broadband circuit capability has been provided with 16 available channels.



TL/F/10827-5

FIGURE 5. FDDI-II Relationship to OSI Model

The WBC's provide a bandwidth division mechanism between the packet and isochronous traffic with a granularity of 6.144 Mbits/s. The allocation of virtual services within the isochronous traffic is allowed with an 8 kbit/s granularity. Once a station has been assigned a WBC or a number of WBC's, that station may suballocate the combined bandwidth of these WBC's as required. This suballocation may be in terms of any multiples of 8 kbit/s subchannels, including the commonly used 16, 32, 64 (B channel), 384, 1536, 1920, and 2048 kbit/s subchannels. Mixtures of these data rates in the same WBC are allowed. If preferred, the aggregate of any or all of the allocated WBC's may be used as one virtual service, satisfying the needs of such applications as high-resolution video. Thus, a multiplicity of virtual circuits may be provided within the same FDDI-II ring.

Assignment of all 16 WBC's, each at 6.144 Mbits/s, yields a total bandwidth of 98.304 Mbits/s. After allowance for the preamble and the cycle header, if all WBC's were allocated, a residual 768 kbit/s channel would then be left for packet traffic. This bandwidth, consisting of 12 bytes every cycle (125 ms), known as the Packet Data Group (PDG), is inter-

leaved with the 16 WBC's as shown in *Figure 6*. The order of transmission is left to right by row starting with the top row. Data steering logic in HRC augments the PDG with the bandwidth of any WBC's that are not assigned. Each WBC is one of the columns in *Figure 6* and represents a bandwidth of 6.144 Mbits/s or 96 bytes per cycle. This is an efficient system that allows the bandwidth of all unallocated WBC's to be used by the packet channel. Thus, for example, with eight (i.e., one-half) of the WBC's assigned to isochronous service, 49.92 Mbits/s of bandwidth would be available for packet traffic.

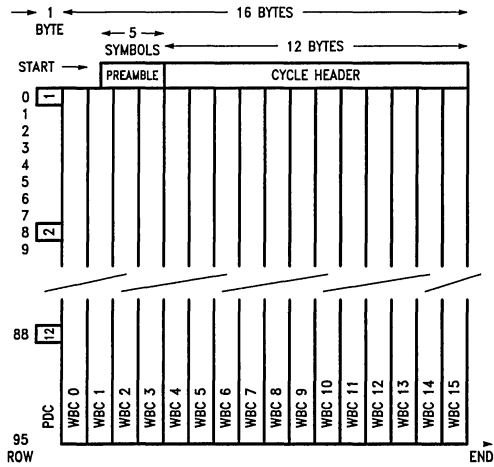


FIGURE 6. FDDI-II Cycle Format

A ring is initialized in basic (token) mode and switched to a hybrid mode of operation, combining both packet-switched and circuit-switched data capabilities, only after a station has negotiated for and won the right to be cycle master and has the synchronous bandwidth allocation required to support it. This cycle master then generates cycles at an 8 kHz rate (every 125 μ s) and inserts the latency required to maintain an integral number of cycles synchronously on the ring. Alternative designs may allow the ring to be initialized directly in hybrid mode.

The cycle header format is shown in *Figure 7*. It follows the Preamble (PA) which is nominally five symbols long. The Starting Delimiter (SD) is the same symbol pair (JK) used as the SD for frames when FDDI is operating in basic mode. In hybrid mode, frames use an IL symbol pair, readily recognized in the rigidly formatted context of hybrid mode, as their starting delimiter. The Cycle Sequence (CS) byte provides a modulo 192 cycle sequence count. The Maintenance Voice Channel (MVC) byte provides a 64 kbit/s voice channel for maintenance purposes.

The 16 symbols of programming information (P0-P15) determine whether the corresponding WBC (0-15) is allocated to packet or isochronous traffic. Thus, each of these (Pn) controls the multiplexing of one of the columns (WBCn) shown in *Figure 6*. The C1 and C2 symbols are used for synchronization control in the transfer of this programming information to the other stations by the cycle master.

A. FDDI-II PRIORITY LEVELS

Four kinds of traffic may coexist in an FDDI-II ring. Once WBC's have been allocated, the isochronous traffic within them has the highest priority.

Second highest priority is given to synchronous packet traffic where predictable units of data are to be delivered at regular intervals. Delivery is guaranteed with a delay not exceeding twice TTRT. These data may be transmitted following the capture of either a restricted or nonrestricted token.

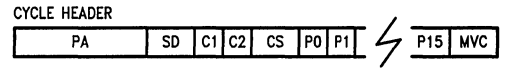


FIGURE 7. Cycle Header Format

The bandwidth required for both isochronous and synchronous traffic is allocated from the available FDDI bandwidth. The allocation algorithm must ensure that the total allocation does not exceed 100 percent. Unallocated bandwidth is used on an as-available basis for asynchronous packet traffic.

Third highest priority is given to asynchronous traffic operating in restricted token mode. Such traffic may be transmitted upon the capture of either a restricted or a nonrestricted token. Cooperating stations may enter a restricted token mode of operation, which allows them to issue and use restricted tokens, only after negotiating an agreement using nonrestricted tokens. Restricted token mode operation allows stations to vie for available asynchronous bandwidth on a dialog basis.

Lowest priority is given to asynchronous traffic that may be transmitted only by capturing a nonrestricted token. This mode of operation allows stations to vie for the available asynchronous bandwidth on a single frame basis.

B. FDDI-II APPLICATIONS

FDDI-II considerably expands the range of applications that may be addressed by FDDI rings. An FDDI-II ring may connect high-performance processors, mass storage systems, high-performance workstations, and perform the backbone function to a number of lower performance LAN's. The same ring may have some of its bandwidth allocated to isochronous services provided by the WBC's. This isochronous bandwidth may in turn be suballocated into a variety of virtual circuit services such as video, voice, and quite possibly, control or sensor data streams. The division of bandwidth between these two kinds of services may be adjusted based on the time of day or other requirements. In practice, no single instance of FDDI connects to all these types of equipment, and certainly not to all the equipment of a large site. Instead, multiple instances of FDDI-II rings will coexist with high traffic units, such as processors, attaching to multiple FDDI networks.

In a packet-switched mode, FDDI can be a backbone for bridges to a variety of other lower-speed LAN's, for example, the various IEEE 802 media access methods. It may also provide a backbone for gateways to the public data networks. In both cases, connections to processors and mass storage subsystems can be provided. Connections to high-performance workstations are likewise provided.

It is anticipated that FDDI-II implementations will follow those of the basic FDDI by one to two years. Committee X3T9.5 has specified the rules of coexistence of these two FDDI implementations to ensure interconnectability and interoperability of FDDI-II implementations operating in basic mode with the basic FDDI implementations. This also allows the use of FDDI-II chips for all applications once they become available.

IX. FDDI Standards Effort and Status

The FDDI standardization effort is taking place in the ASC X3T9.5 committee which meets bimonthly with an attendance of well over 100. The official voting membership, limited to one per corporation, is approximately 80. Ad hoc working meetings, scheduled as required, are presently focusing on SMT, FDDI-II, and SMF-PMD detail design definitions as well as issues concerning MAC-level bridging.

After completion by ASC X3T9.5, a draft proposed American National Standard (dpANS) is forwarded to ASC X3T9 for a technical letter ballot which approves forwarding to X3. X3 conducts a four-month public review, followed by an X3 letter ballot. Finally, the dpANS is forwarded to the Board of Standards Review (BSR) for formal approval. Upon approval, the standard is given a final editing to conform to ANSI requirements and published.

MAC (Rev. 10) is an American National Standard (ANSI X3.138-1987) and was published in July 1987. Current activities are focused on potential enhancements to MAC, including formalization of the 48-bit addressing structure used in IEEE 802 LAN's, reliability enhancement in the criteria for frame recognition, and incorporation of the FDDI-II requirements.

PHY (Rev. 15) is also an American National Standard (ANSI X3.148-1988) and was published in December 1988.

PMD (Rev. 8), X3.166-198x, completed the X3 four-month public review in late 1987. This initial forwarding had been intended to allow a full set of comments on the optical parameters documented in it. Five sets of comments received resulted in a number of refinements to the optical specifications of PMD. There were also a number of comments on the connector choice and its documentation. In April 1988, X3T9.5 finalized the optical specifications and reaffirmed its connector choice. PMD, having completed a second (two-month) public review in December 1988, is now being processed for final approval as a standard.

Work has progressed smoothly on the SMF-PMD project which was started in mid-1987. An X3T9 technical letter ballot approved the SMF-PMD document (Revision 4) and it has been forwarded to X3 for the four-month public review and final approval as a standard.

The technical definition of FDDI-II, long maintained in a working paper, has now been completed and is contained in Revision 5 of the Hybrid Ring Control (HRC) document. This document also passed an X3T9 technical letter ballot and has also been forwarded to X3 for public review and approval.

Participation in the working meetings on SMT has been high through much of 1987 and into 1989, with well over 100 attendees at the most recent meetings. Representing the meeting ground between the widely differing FDDI implementation and usage philosophies, SMT has proved a difficult task. Contentious items have included the use of concentrators without a MAC, link confidence test duration, physical layer link error monitoring, and physical layer signaling. But the committee's determination to produce an SMT standard assuring interoperability has prevailed and SMT is progressing toward a technical letter ballot in the latter part of 1989.

In a similar process designed to produce technically equivalent ISO standards for FDDI, the FDDI documents are being processed by ISO/IEC JTC1/SC 13. This process has resulted in the approval of MAC (IS 9314-2) and PHY (IS 9314-1) as ISO standards. PMD (DIS 9314-3) has passed the letter ballot for approval as an International Standard with publishing expected in late 1989. SMF-PMD and HRC are currently being processed for submission to ISO.

Led by the ASC X3T9.5, this FDDI standards effort has continued to reflect the FDDI product implementations of its many supporters, including a number of semiconductor manufacturers working towards FDDI chip sets. Several FDDI chip sets available either now or in the near future promise cost-effective implementations of these FDDI standards in the near future.

X. Conclusion

FDDI has proved a very successful standardization effort, far more successful than those who conceived it could ever have imagined. Users, system houses, chip designers, and component vendors have come together to produce a set of standards of lasting significance. With the widespread use of the basic FDDI will come a broad based market for FDDI-II which satisfies the much wider range of requirements typical of voice, video, and sensor/control based data streams.

References

- [1] ANSI/IEEE Standard, 802.5-1985, "Token Ring Access Method and Physical Layer Specifications."
- [2] American National Standard, "FDDI Token Ring Media Access Control (MAC)," ANSI X3.139-1987.
- [3] American National Standard, "FDDI Token Ring Physical Layer Protocol (PHY)," ANSI X3.148-1988.
- [4] Draft Proposed American National Standard, "FDDI Token Ring Physical Layer Medium Dependent (PMD)," ACS X3T9.5 Rev. 9, Mar. 1989 (ANSI X3.166-198x).
- [5] Draft Proposed American National Standard, "FDDI Hybrid Ring Control (HRC)," ASC X3T9.5 Rev. 5, Apr. 1989.
- [6] Draft Proposed American National Standard, "FDDI Token Ring Station Management (SMT)," ASC X3T9.5, Rev. 5, May 1989.
- [7] Draft Proposed American National Standard, "FDDI Token Ring Single-Mode Fiber Physical Layer Medium Dependent (SMF-PMD)," ASC X3T9.5 Rev. 4, Apr. 1989.

References (Continued)

- [8] F. Ross, "FDDI—A Tutorial," *IEEE Commun. Mag.*, May 1986.
- [9] V. Iyer and S. Joshi, "FDDI's 100-Mbps Protocol Improves on 802.5 spec's 4-Mbps Limit," *EDN*, May 2, 1985.
- [10] F. E. Ross, "Rings are 'Round for Good,'" *IEEE Network Mag.*, Jan. 1987.
- [11] M. J. Johnson, "Reliability Mechanisms of the FDDI High Bandwidth Token Ring Protocol," *Comput. Networks ISDN Syst.*, Vol. II, No. 2, pp. 121–131, 1986.
- [12] H. Salwen, "In Praise of Ring Architecture for Local Area Networks," *Comput. Design*, pp. 183–192, Mar. 1983.
- [13] R. M. Grow, "A Timed Token Protocol for Local Area Networks," Presented at Electro '82, Token Access Protocol (17/3), May 1982.
- [14] M. J. Johnson, "Fairness of Channel Access for Non-Time-Critical Traffic Using the FDDI Token Ring Protocol," in *Proc. Seminar Real-Time Local Area Networks*, Bandol, France, Apr. 1986, pp. 145–157.
- [15] —, "Proof That Timing Requirements of the FDDI Token Ring PROTOcol are Satisfied," *IEEE Trans. Commun.*, vol. COM-35, pp. 620–625, 1987.
- [16] F. E. Ross, "FDDI—A Perspective," in *1988 Fiber Optics Source-Book*, 3rd Annu. Ed., pp. 193–210.
- [17] M. J. Johnson, "Analysis of FDDI Synchronous Traffic Delays," in *Proc. Syst. Design Networks*, Apr. 1988, pp. 65–72.
- [18] —, "Performance Analysis of FDDI," Presented at EFOC/LAN '88, June 1988.
- [19] J. Hamstra, "FDDI Design Tradeoffs," Presented at the 13th Conf. Local Comput. Networks, Oct. 1988.
- [20] W. E. Burr, "The FDDI Optical Data Link," *IEEE Commun. Mag.*, May 1986.

Floyd E. Ross received the B.A. degree in mathematics and physics from Mount Allison University in 1955 and did postgraduate work at McMaster University, Hamilton, Ont., Canada.

He is a Staff Engineer with UNISYS, Devon, PA, with responsibilities in communications networks and in LAN definition and implementation. He is an active contributor to accredited standards committee X3T9 on I/O Interfaces and to IEEE 802. He is Vice Chairman of X3T9.5 and has been the leader of the FDDI standards development effort since its inception in that committee. He has chaired many of the FDDI working meetings including those on SMT and FDDI-II and is technical editor of the four basic FDDI standards. He has also presented and authored a number of papers on both local area networks and FDDI.



Section 2
DP83200 FDDI Chip Set



Section 2 Contents

DP83231 CRD Device (FDDI Clock Recovery Device)	2-3
DP83241 CDD Device (FDDI Clock Distribution Device)	2-18
DP83251/DP83255 PLAYER Device (FDDI Physical Layer Controller)	2-36
DP83261 BMAC Device (FDDI Media Access Controller)	2-130
DP83265 BSI Device (FDDI System Interface)	2-260

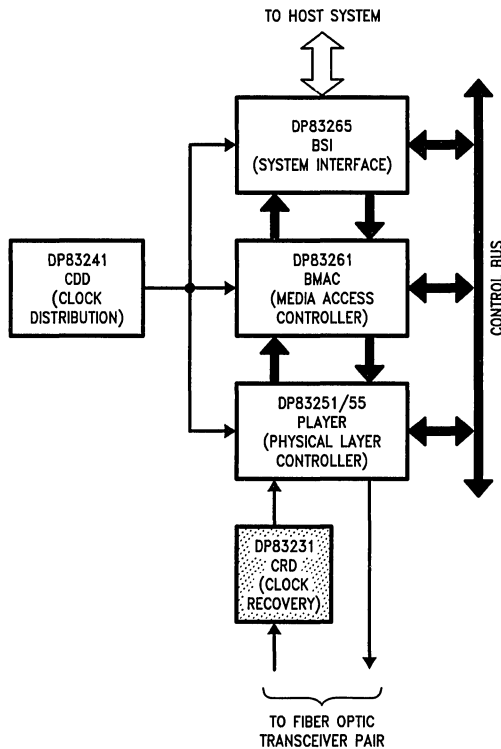
DP83231 CRD™ Device (FDDI Clock Recovery Device)

General Description

The DP83231 CRD device is a clock recovery device that has been designed for use in 100 Mbps FDDI (Fiber Distributed Data Interface) networks. The device receives serial data from a Fiber Optic Receiver in differential ECL NRZI 4B/5B group code format and outputs resynchronized NRZI received data and a 125 MHz received clock in differential ECL format for use by the DP83251/55 PLAYER™ device.

Features

- Clock recovery at 100 Mbps data rate
- Internal 250 MHz VCO
 - 0.1% VCO operating range
 - Crystal controlled
- Precision window centering delay line
- Single +5V supply
- 28-pin PLCC package
- BiCMOS processing



TL/F/10384-1

FIGURE 1-1. FDDI Chip Set Block Diagram

Table of Contents

1.0 FDDI CHIP SET OVERVIEW

2.0 FUNCTIONAL DESCRIPTION

3.0 PIN DESCRIPTIONS

4.0 ELECTRICAL CHARACTERISTICS

- 4.1 Absolute Maximum Ratings
- 4.2 Recommended Operating Conditions
- 4.3 DC Electrical Characteristics
- 4.3 AC Electrical Characteristics

5.0 DETAILED INFORMATION

- 5.1 Special External Components
- 5.2 Layout Recommendations
- 5.3 Input and Output Schematics
- 5.4 Debug Procedure
- 5.5 AC Test Circuits

1.0 FDDI Chip Set Overview

National Semiconductor's FDDI chip set consists of five components as shown in *Figure 1-1*. For more information about the other devices in the chip set, consult the appropriate data sheets and application notes.

DP83231 CRD™ Device Clock Recovery Device

The Clock Recovery Device extracts a 125 MHz clock from the incoming bit stream.

Features

- PHY Layer loopback test
- Crystal controlled
- Clock locks in less than 85 μ s

DP83241 CDD™ Device Clock Distribution Device

From a 12.5 MHz reference, the Clock Distribution Device synthesizes the 125 MHz, 25 MHz and 12.5 MHz clocks required by the BSI, BMAC, and PLAYER devices.

DP83251/55 PLAYER™ Device Physical Layer Controller

The PLAYER device implements the Physical Layer (PHY) protocol as defined by the ANSI FDDI PHY X3T9.5 Standard.

Features

- 4B/5B encoders and decoders
- Framing logic
- Elasticity Buffer, Repeat Filter, and Smoother
- Line state detector/generator
- Link error detector
- Configuration switch
- Full duplex operation
- Separate management port that is used to configure and control operation.

In addition, the DP83255 contains an additional PHY_Data.request and PHY_Data.indicate port required for concentrators and dual attach stations.

DP83261 BMAC™ Device Media Access Controller

The BMAC device implements the Timed Token Media Access Control protocol defined by the ANSI FDDI X3T9.5 MAC Standard.

Features

- All of the standard defined ring service options
- Full duplex operation with through parity
- Supports all FDDI Ring Scheduling Classes (Synchronous, Asynchronous, etc.)
- Supports Individual, Group, Short, Long, and External Addressing
- Generates Beacon, Claim, and Void frames internally
- Extensive ring and station statistics gathering
- Extensions for MAC level bridging
- Separate management port that is used to configure and control operation
- Multi-frame streaming interface

DP83265 BSIT™ Device System Interface

The BSI Device implements an interface between the National FDDI BMAC device and a host system.

Features

- 32-bit wide Address/Data path with byte parity
- Programmable transfer burst sizes of 4 or 8 32-bit words
- Interfaces to low-cost DRAMs or directly to system bus
- Provides 2 Output and 3 Input Channels
- Supports Header/Info splitting
- Efficient data structures
- Programmable Big or Little Endian alignment
- Full Duplex data path allows transmission to self
- Confirmation status batching services
- Receive frame filtering services
- Operates from 12.5 MHz to 25 MHz synchronously with host system

2.0 Functional Description

The DP83231 uses two phase locked loops (PLL's) to perform the clock recovery function. The function of the first PLL is to establish a 250 MHz Voltage Controlled Oscillator (VCO) with a narrow frequency range which can be pulled by the second PLL. The function of the second PLL is to force this same VCO to track the incoming data so that a Receive Clock output and a data synchronizing flip-flop can be driven from it. Operation of the VCO at 250 MHz ensures that the received clock output operating at half of the VCO frequency has a 50% duty cycle waveform independent of any VCO waveform dissymmetry.

The first PLL uses a 10.41666 MHz crystal as a pullable frequency reference to generate the 250 MHz VCO. The limited frequency pulling range of the crystal ensures that the capture range of the 250 MHz VCO is limited to less than 0.1% of the specified data transition rate, thus eliminating the possibility of fractional or harmonic lock up modes. The output of the VCO is divided by twenty four and applied to the feedback input of the phase detector in the first PLL. The phase detector compares the phase of the VCO divided by twenty four signal against the phase of the crystal to maintain VCO lock at 250 MHz. If the phase transition of the signal derived from the VCO arrives at the phase detector before that of the crystal, the charge pump circuitry will apply a negative current pulse to the VCO FLTR node whose width is proportional to the phase error. The charge pulled out of the filter capacitors will drive the voltage applied to the VCO downward. This reduction in the VCO's control voltage will slow down the frequency of the VCO and will appear during successive cycles to reduce the VCO's phase and frequency error. As the frequency of the crystal varies, in response to the second PLL, the frequency of the 250 MHz VCO will change in an attempt to remain 24 times the crystal's frequency.

The second PLL delays the phase transitions of the selected incoming stream of data (DATA \pm or LBD \pm), and then compares them against the phase transitions of a gated 125 MHz signal derived from the 250 MHz VCO. The delayed incoming data is applied to the reference input of a phase detector and the gated VCO signal is applied to its feedback input. If the positive and negative phase transitions of the incoming data do not line up with the phase transitions of the gate VCO signal, the charge pump circuitry associated with that phase detector will apply current pulses to the OSC FLTR \pm nodes which are proportional to the phase error. The change in the charge on the filter capacitors will modify the reverse bias on the varactors in the crystal's tank circuit thus causing the frequency of the 10.41666 MHz crystal (and consequently the VCO) to shift

in the direction which will reduce their phase error. When the phase of the VCO and the incoming data are aligned, a VCO divided by two signal can be used as the Receive Clock output. Because the two PLL's share a common VCO feedback path, the cutoff frequency of the loop filters associated with the second PLL are specified to be approximately 10 times lower than the cutoff frequency of the first PLL to prevent instability between the two loops.

The delay line associated with the second PLL precisely centers the data transitions within the data window. The delay line remains accurate independent of temperature, power supply, IC process variation or external components. The design also ensures that the charge pump up and down circuits both produce an active pulse at each zero phase crossing when in lock to guarantee a linear phase detector gain characteristic.

The CRD continually monitors the data frequency at the selected data inputs. If this input frequency drops below $\frac{1}{2}$ the minimum allowed frequency (about 3 MHz) the CRD resets itself by internally deasserting CRD-EN. This centers the crystal frequency, and restarts the internal VCO.

The CRD EN pin is provided to initialize the CLK DET circuitry and enable the crystal to track incoming data. The part is enabled when this pin is active High. Deassertion of this pin will cause the CLK DET circuitry and the OSC FLTR \pm pins to be disabled in a manner similar to when legitimate data is not being received. Deassertion of the CRD EN pin also momentarily causes (1 μ s) the VCO FLTR pin to be pulled to ground and stops the VCO and RXC \pm outputs. After this time, the VCO will be restarted and its output frequency will climb quickly to approximately 250 MHz.

The device is capable of locking on to a stream of Halt or Master line states in less than 100 μ s when using a 10.41666 MHz crystal to govern the 250 MHz VCO. Lock on time for a stream of Idle line states is less than 10 μ s once Halt or Master line status is obtained. During quiet line conditions the chip will output a continual stream of Received Clock whose frequency will be within less than 0.1% of the upstream station's data rate. The Received Data outputs are always active. Prior to the CLK DET output transitioning active High, the Received Data outputs may issue invalid data (see Typical Waveforms). When the device is locked, Received Data is presented on the falling edge of the Receive Clock output insuring sufficient setup and hold margin for the receiving device.

An ECL to TTL translator is provided on the chip to convert the FORX's ECL signal detect output level to TTL for use by the PLAYER device.

2.0 Functional Description (Continued)

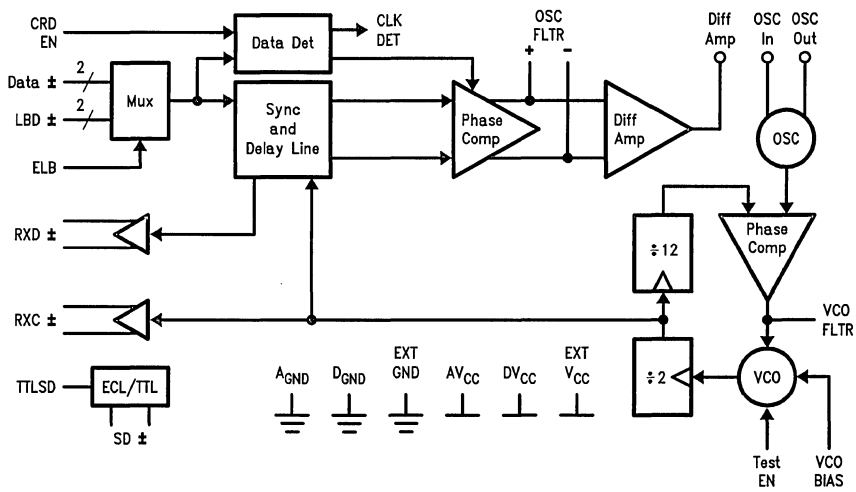
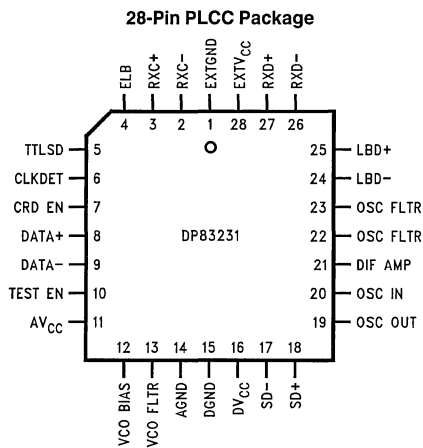


FIGURE 2-1. DP83231 Block Diagram

TL/F/10384-3



Order Number DP83231AV
See NS Package Number V28A

FIGURE 2-2. DP83231 Pinout

TL/F/10384-2

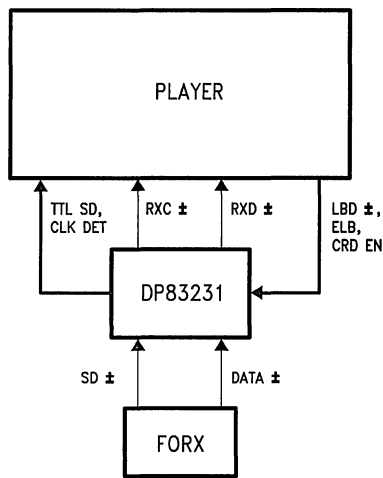


FIGURE 2-3. System Connection Diagram

TL/F/10384-4

3.0 Pin Descriptions

Symbol	Pin No.	I/O	Description
DATA ⁺ , DATA ⁻	8, 9	I	DATA ±: 4B/5B serial NRZI data inputs originating from a fiber optic receiver and presented for the purpose of resynchronization and clock recovery. These differential 100k ECL compatible inputs are selected when the ELB input is at a logic Low level.
LBD ⁺ , LBD ⁻	25, 24	I	Loopback Data ±: 4B/5B serial NRZI data inputs originating from a local PLAYER device and presented for the purpose of station diagnostics. These differential 100k ECL compatible inputs are selected when the ELB input is at a logic High level.
ELB	4	I	Enable Loopback: TTL compatible input which selects between the DATA ± inputs or the LBD ± inputs. The LBD inputs are selected when the ELB pin is at a logic High level and the DATA inputs when at a logic Low level.
CLK DET	6	O	Clock Detect: CMOS output used to indicate that the chip has detected the presence of a continuous data frequency greater than 3.0 MHz. A logic High level on the output will indicate that valid input data has been detected.
CRD EN	7	I	CRD Enable: TTL compatible input which directs the differential charge pump outputs to either operate the crystal oscillator at the center of its operating range or to track out the VCO phase errors in the second PLL. The CRD EN input will reset the CLK DET function and will force the oscillator to the center of its operating range when at a logic LOW level and will allow normal PLL tracking operation when at a logic High level. Deassertion of the CRD EN input will momentarily stop the VCO.
OSC FLTR ⁺ , OSC FLTR ⁻	23, 22	O	Oscillator Filter ±: The differential charge pump up and down outputs which are part of the second PLL. A three element filter should be connected to each of these pins which consists of one capacitor in parallel with a resistor and another capacitor to ground. These outputs are driven to their maximum upper operating level when the CRD EN pin is at a logic LOW level or when valid data frequencies are not recognized at the data inputs.
DIF AMP OUT	21	O	Differential Amplifier Output: The differential amplifier output associated with the second PLL which is used to adjust the frequency of the external crystal.
OSC_IN, OSC_OUT	20, 19	I	Oscillator Input and Output: The terminals for the crystal oscillator which require connection of the crystal tank circuit, varactors, and capacitors.
RXC ⁺ , RXC ⁻	3, 2	O	Receive Clock: Differential 100K ECL receive clock outputs which operate at 125 MHz synchronized to the selected inputs (NRZI DATA ± or LBD ±) when valid line state data is present. When valid line state data is not present these outputs continue to operate at a nominal frequency of 125 MHz ± 12 kHz. These outputs should be terminated externally with a conventional ECL 50Ω equivalent load.
RXD ⁺ , RXD ⁻	27, 26	O	Receive Data: Differential 100K ECL received data outputs which provide a resynchronized equivalent of the selected NRZI DATA or LBD inputs. The received data output transitions occur concurrent with the falling edge of the RXC ± output. These outputs should be terminated externally with a conventional ECL 50Ω equivalent load.
VCO FLTR	13	O	VCO Filter: Low pass filter associated with the first PLL. A three element filter should be connected to this pin which consists of one capacitor in parallel with a resistor and another capacitor to ground.
SD ⁺ , SD ⁻	18, 17	I	Signal Detect: Differential inputs to a 100K ECL to TTL translator intended for conversion of the fiber optic receiver's ECL signal detect to TTL for a player device. The inputs are used in the test modes as inputs for single stepping and gating the VCO.
TTLSD	5	O	TTL Signal Detect: Intended to be a signal detect output in TTL format for use by the PLAYER chip.
TEST EN	10	I	Test Enable: CMOS input which enables the test functions. This input must be at a logic low level in normal operation.
DV _{CC}	16		Digital V_{CC}: Positive power supply for most of the internal logic circuitry intended for + 5V operation ± 5% relative to ground. Bypass capacitors should be placed as close as possible across the DV _{CC} and DGND pins. DV _{CC} , AV _{CC} and EXTV _{CC} should be tied together through chokes.

3.0 Pin Descriptions (Continued)

Symbol	Pin No.	I/O	Description
EXTV _{CC}	28		External V_{CC}: Positive power supply for all the input and output buffers intended for +5V operation $\pm 10\%$ relative to ground. Bypass capacitors should be placed as close as possible across the EXTV _{CC} and EXTGND pins. DV _{CC} , AV _{CC} and EXTV _{CC} should be tied together at the device pins through chokes.
DGND	15		Digital Ground: Power supply return for the internal circuitry. DGND, AGND and EXT GND pins should be tied together.
EXTGND	1		External Ground: Power supply return for the input and output buffers. DGND, AGND and EXT GND pins should be tied together.
AV _{CC}	11		Analog V_{CC}: Positive power supply for the critical analog circuitry intended for +5V operation $\pm 5\%$ relative to ground. Bypass capacitors should be placed as close as possible across the AV _{CC} and AGND pins. DV _{CC} , EXTV _{CC} and AV _{CC} should be tied together through chokes.
AGND	14		Analog Ground: Power supply return for the critical analog circuitry. DGND, EXTGND and AGND pins should be tied together.
VCO BIAS	12	I	VCO Bias: TTL compatible input that sets the nominal frequency for the VCO by the selection of the resistor value between this input and AV _{CC} . A 30 k Ω value for this resistor will provide nominally 125 MHz on the RXC outputs.

4.0 Electrical Characteristics

4.1 ABSOLUTE MAXIMUM RATINGS

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature -65°C to $+150^{\circ}\text{C}$

TTL Signals

Inputs 5.5V

Outputs 5.5V

ECL Signals

Output Current -20 mA

Supplies

EXTV_{CC} to EXTGND -0.5V to $+7\text{V}$

DV_{CC} to DGND -0.5V to $+7\text{V}$

AV_{CC} to AGND -0.5V to $+7\text{V}$

ESD Susceptibility 2000V

4.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter		Min	Typ	Max	Units
V _{CC} to GND	Power Supply		4.75	5	5.25	V
V _{IH}	High Level Input Voltage	TTL	2.0			V
		ECL	V _{CC} - 1.165		V _{CC} - 0.880	
V _{IL}	Low Level Input Voltage	TTL			0.8	V
		ECL	V _{CC} - 1.810		V _{CC} - 1.475	
I _{OH}	High Level Output Current	TTL Outputs (Note 1)			-0.4	mA
I _{OL}	Low Level Output Current	TTL Outputs (Note 1)			4.0	mA
FVCO	VCO Frequency			250		MHz
FXTL	Crystal Frequency			10.416667		MHz
TA	Operating Temperature		0	25	70	$^{\circ}\text{C}$

Note 1: TTL outputs include CLK DET and TTLSLD.
ECL outputs include RXC \pm and RXD \pm .

4.0 Electrical Characteristics (Continued)

4.3 DC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Max	Units
V_{IC}	Input Clamp Voltage	$I_{IN} = 18 \text{ mA}$		-1.5	V
V_{OH}	High Level Output Voltage	TTL Outputs: $I_{OH} = -400 \mu\text{A}$		$V_{CC} - 2$	V
		ECL Outputs: 50Ω Load to $V_{CC} - 2\text{V}$	$V_{CC} - 1025$	$V_{CC} - 880$	mV
V_{OL}	Low Level Output Voltage	TTL Outputs: $I_{OL} = 4 \text{ mA}$		0.5	V
		ECL Outputs: 50Ω Load to $V_{CC} - 2\text{V}$	$V_{CC} - 1810$	$V_{CC} - 1620$	mV
I_I	Max High Level Input Current	TTL Inputs: $V_{IN} = 7\text{V}$		100	μA
I_{IH}	High Level Input Current	TTL Inputs: $V_{IN} = 2.7\text{V}$	-20	20	μA
I_{IL}	Low Level Input Current	TTL Inputs: $V_{IN} = 0.4\text{V}$	-20	20	μA
I_{filter}	Charge Pump Current	Source	-0.3	-0.7	mA
		Sink	0.3	0.7	mA
		TRI-STATE®	-500	500	nA
I_{CC}	Supply Current			180*	mA

*Includes 60 mA due to external ECL termination of two differential signals.

For 100k ECL output buffers, output levels are specified as:

$$V_{OH_max} = V_{CC} - 0.88\text{V}$$

$$V_{OL_max} = V_{CC} - 1.62\text{V}$$

Since the outputs are differential, the average output level is $V_{CC} - 1.25\text{V}$. The test load per output is 50Ω at $V_{CC} - 2\text{V}$. The external load current through this 50Ω resistor is thus:

$$\begin{aligned} I_{load} &= [(V_{CC} - 1.25) - (V_{CC} - 2)]/50\text{A} \\ &= 0.015\text{A} \\ &= 15 \text{ mA} \end{aligned}$$

There are 2 pairs of differential ECL signals, so the total ECL current is 60 mA.

4.4 AC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Max	Units
T1	Phase Difference		-2	2	ns
T2	RXC Pos. Pulse Width	(Note 1)	3	5	ns
T3	CLK DET Time	CRD EN Neg. Pulse Width = 1 μs (Valid DATA \pm Present)		100	μs
T4	Valid Data Time	CRD EN = High		100	μs

Note 1: These parameters are not tested, but are assured by correlation with characterization data.

4.0 Electrical Characteristics (Continued)

TYPICAL WAVEFORMS

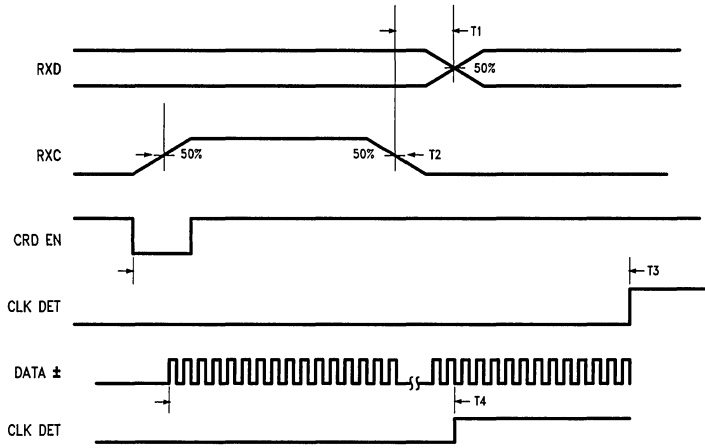


FIGURE 4-1. DP83231 Timing Waveforms

TL/F/10384-5

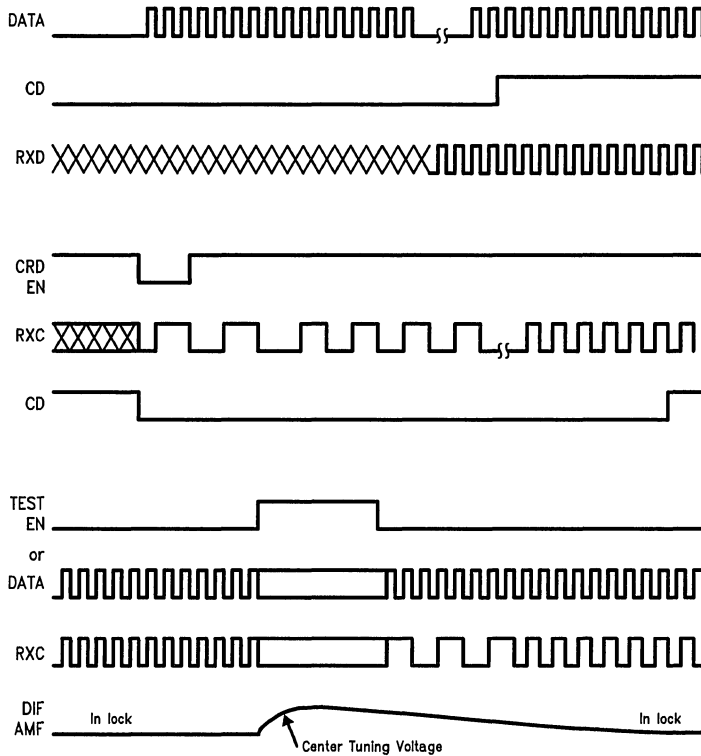
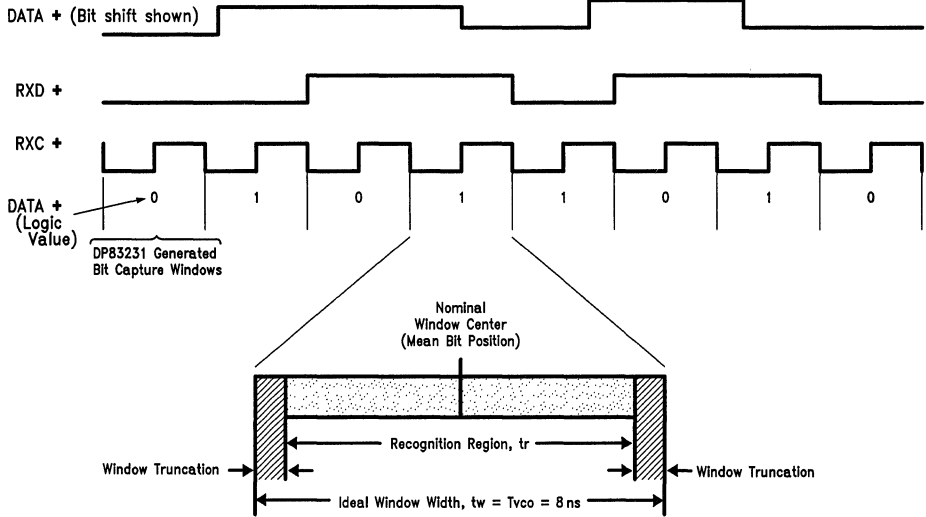


FIGURE 4-2. Typical Waveforms

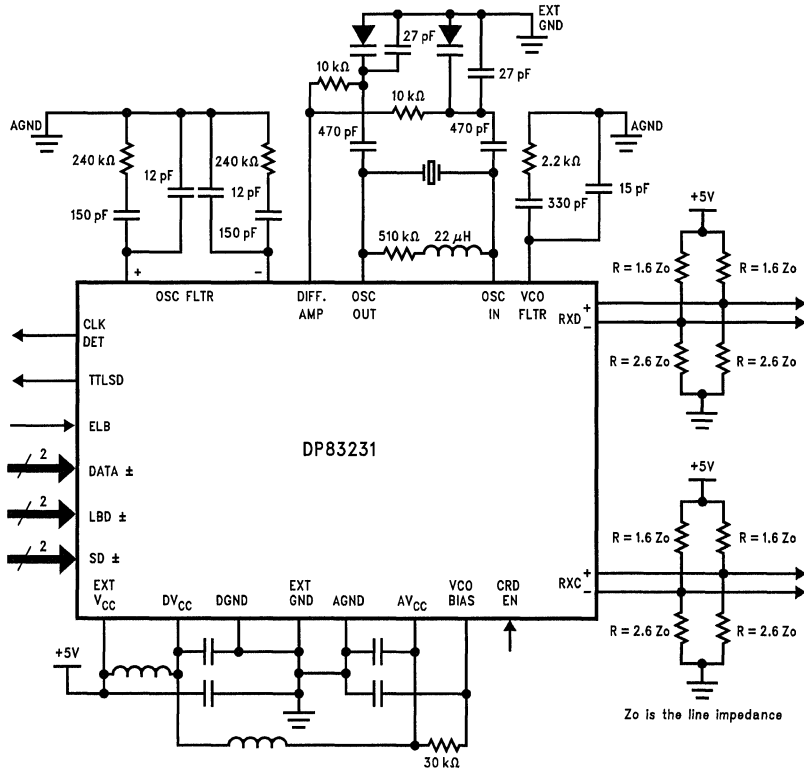
TL/F/10384-6

4.0 Electrical Characteristics (Continued)



TL/F/10384-7

FIGURE 4-3. Synchronization Window



All component values $\pm 5\%$.

TL/F/10384-8

FIGURE 4-4. General Wiring Diagram

5.0 Detailed Information

5.1 SPECIAL EXTERNAL COMPONENTS

Crystals

- Manufacturer: Nel Frequency Controls (414) 763-3591
Part #: C5400N
- Manufacturer: Standard Crystal Corporation (818) 443-2121
Part #: 800R-A-10.41667-32

Key Specifications:

- Center Frequency: 10.41667 MHz
- Load Capacitance, C_L : 32 pF
- Frequency Calibration: ± 20 PPM
- Frequency Stability (0°C–70°C): ± 20 PPM
- Aging: $< \pm 10$ PPM
- Pullability: either a motional capacitance of ≥ 0.021

or

a change of at least 100 PPM when the C_L is changed from 32 pF to 18 pF and a change of -100 PPM when the C_L is changed from 32 pF to 50 pF.

Varactors

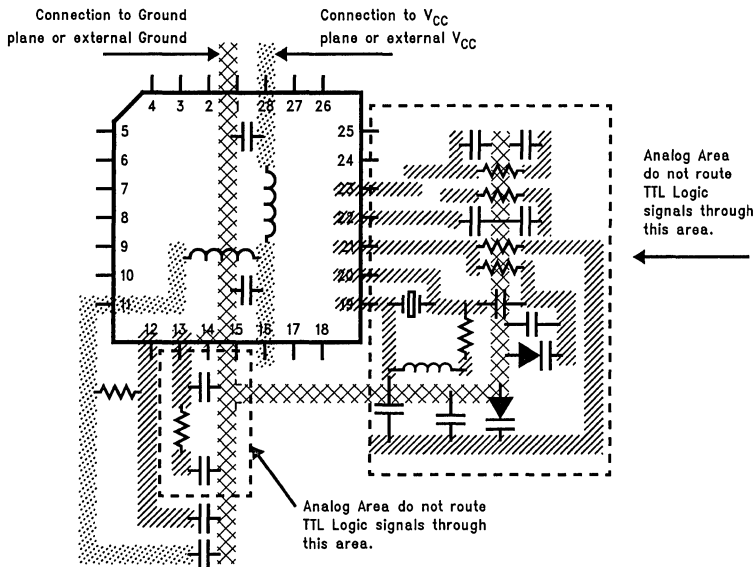
- Manufacturer: Alpha Industries (617) 935-5150
Part #: DKV6510-71

Key Specifications:

- Capacitance: @ $V_r = 1V$: $C > 85$ pF
- @ $V_r = 4V$: 15 pF $< C < 30$ pF

5.2 LAYOUT RECOMMENDATIONS

- The part should be bypassed between the $EXTV_{CC}$ and $EXTGND$ as close to the chip as possible (preferably under the chip using chip caps). The part should also be bypassed between the DV_{CC} and $DGND$ and the AV_{CC} and $AGND$ as close to the chip as possible.
- No TTL logic lines should pass through the crystal OSC FLTRs or VCO FLTR circuitry areas to avoid the possibility of noise due to crosstalk.
- The crystal, OSC FLTRs and the VCO FLTR circuitries should be connected to Ground on isolated branches off of the $DGND$ pin. If using a multilayered board with dedicated V_{CC} and Ground planes ensure that for the ground plane that the ceramic resonator, OSC FLTRs and the VCO FLTR circuitries have their own small isolated islands that are connected to the $DGND$ and $AGND$ pins as described above.
- The DV_{CC} and AV_{CC} pins should be connected to V_{CC} on an isolated branches off of the $EXTV_{CC}$ pin, preferably being connected through a ferrite bead or a small inductor.
- The $DGND$ and $AGND$ pins should be connected to GND on an isolated branches off of the $EXTGND$ pin. Connection to the ground plane should be made only at the $EXTGND$ pin.



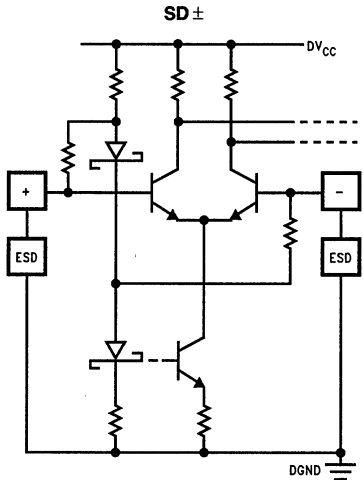
This drawing was done with convenience in mind.

FIGURE 5-1. Recommended Layout

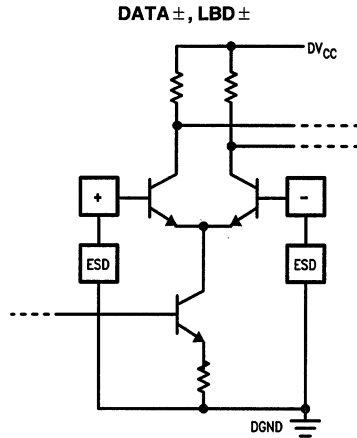
TL/F/10384-10

5.0 Detailed Information (Continued)

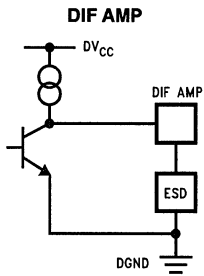
5.3 INPUT AND OUTPUT SCHEMATICS



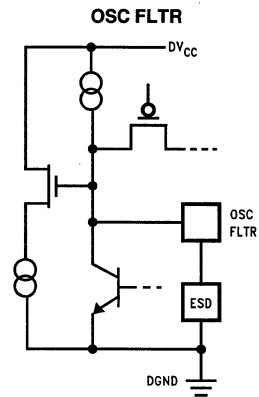
TL/F/10384-11



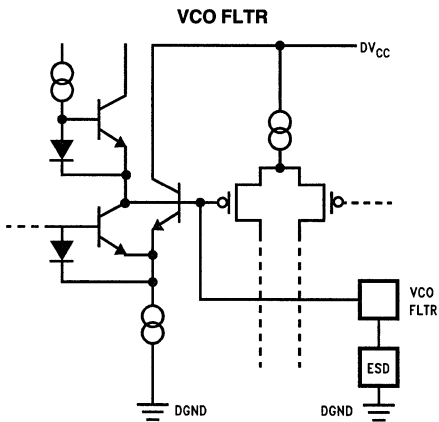
TL/F/10384-12



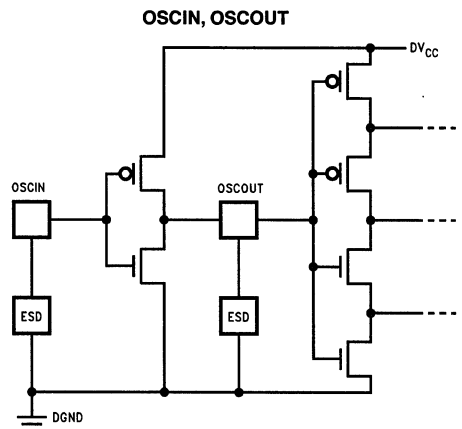
TL/F/10384-13



TL/F/10384-14



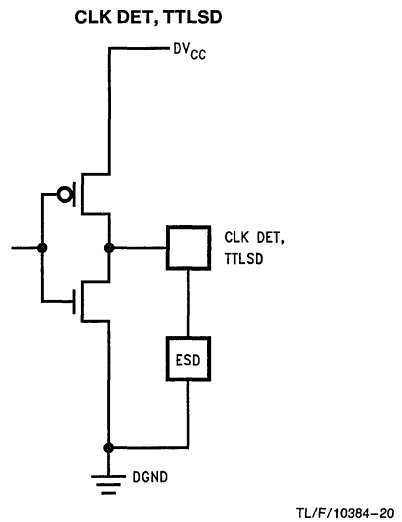
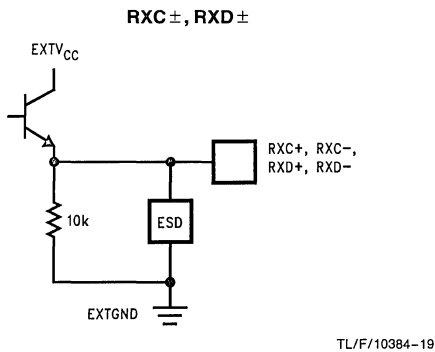
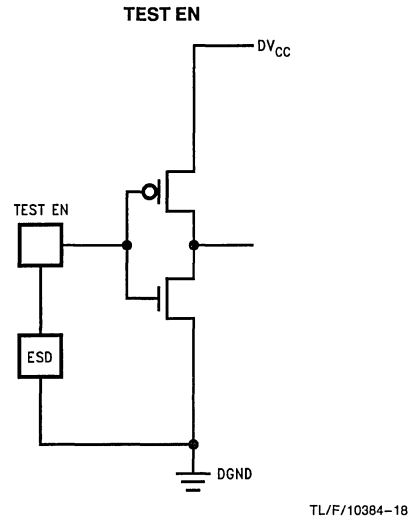
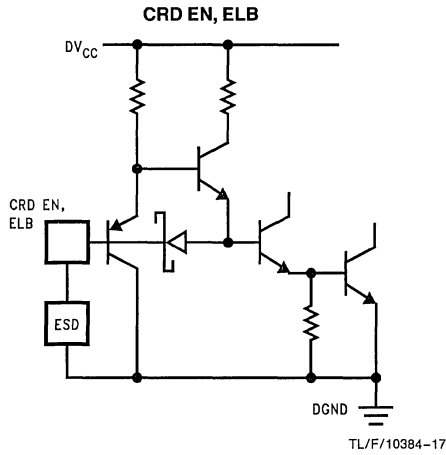
TL/F/10384-15



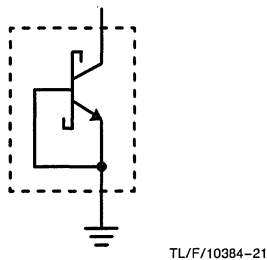
TL/F/10384-16

5.0 Detailed Information (Continued)

5.3 INPUT AND OUTPUT SCHEMATICS (Continued)



Typical ESD Structure



5.0 Detailed Information (Continued)

5.4 DEBUG PROCEDURE

Evaluation of the DP83231 should begin by tying the CRD EN and TEST EN pins low and confirming that the SD± pins are above 2V. This will disable the differential phase comparator allowing the crystal resonator to run at its center frequency and will keep the part out of a test mode. The first PLL (see Figure 5-2) should be evaluated. The variable capacitor in the crystal resonator circuitry should be tuned so that the crystal resonator oscillates at 10.41666 MHz. If the oscillator circuit fails to oscillate the voltage levels of the OSC IN and OSC OUT pins should be examined. The DC voltage on these pins should be equal to approximately $V_{CC} \div 2$ (with or without the crystal present). The capacitors which form the oscillator tank circuit should be returned to the isolated ground branch in close proximity. After checking the crystal frequency, examine the RXC± output and verify that this frequency is twelve times the crystal frequency. If this is not true then the VCO FLTR output should be examined for possible PC board shorts, opens or filter instability. The VCO FLTR pin should be stable at approximately a 1.5V DC level in operation.

If the VCO FLTR pin is oscillating then the loop filter components for this pin were either chosen inappropriately or were placed in the incorrect position.

Once it is known that the first PLL is working, force CRD EN high and input a constant 62.5 MHz ± 50 ppm (1T pattern) data stream to the DATA± inputs (see Figure 5-3). To see how well the second loop is working examine the DIF AMP pin. If the incoming data rate is exactly 62.5 MHz and the crystal resonator was accurately adjusted as described above, then the DIF AMP pin voltage should be stable at approximately 2.25V. The voltage at this pin will vary

from the nominal value dependent on temperature and data rate frequency error. If this pin is oscillating then the OSC FLTR pins are unstable and the filters should be examined for possible PC board shorts, opens or instability. If the DIF AMP pin is near ground then check to see if the ELB input is selecting the correct data input. If the DIF AMP pin continues to be near ground or V_{CC} , then the accuracy of the 62.5 MHz source should be examined to verify it is within the ± 3 KHz (50 PPM) FDDI system data rate specification.

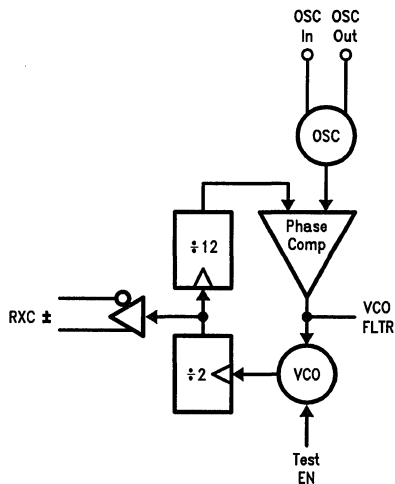


FIGURE 5-2. 1st PLL

TL/F/10384-22

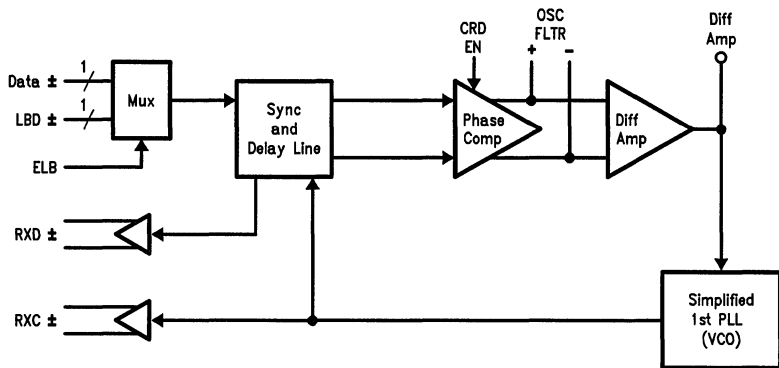
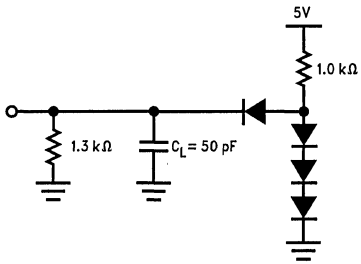


FIGURE 5-3. 2nd PLL

TL/F/10384-23

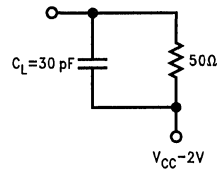
5.0 Detailed Information (Continued)

5.5 AC TEST CIRCUITS



TL/F/10384-24

FIGURE 5-4. Switching Test Circuit for All TTL Output Signals



TL/F/10384-25

FIGURE 5-5. Switching Test Circuit for All ECL Input and Output Signals



DP83241 CDD™ Device (FDDI Clock Distribution Device)

General Description

The CDD device is a clock generation and distribution device intended for use in FDDI (Fiber Distributed Data Interface) networks. The device provides the complete set of clocks required to convert byte wide data to serial format for fiber medium transmission and to move byte wide data between the PLAYER™ and BMACT™ devices in various station configurations. 12.5 MHz and 125 MHz differential ECL clocks are generated for the conversion of data to serial format and 12.5 MHz and 25 MHz TTL clocks are generated for the byte wide data transfers.

Features

- Provides 12.5 MHz and 25 MHz TTL clocks
- 12.5 MHz and 125 MHz ECL clocks
- 5 phase TTL local byte clocks eliminate clock skew problems in concentrators
- Internal VCO requires no varactors, coils or adjustments
- Option for use of High Q external VCO
- 125 MHz clock generated from a 12.5 MHz crystal
- External PLL synchronizing reference for concentrator configurations
- 28-pin PLCC package
- BiCMOS processing

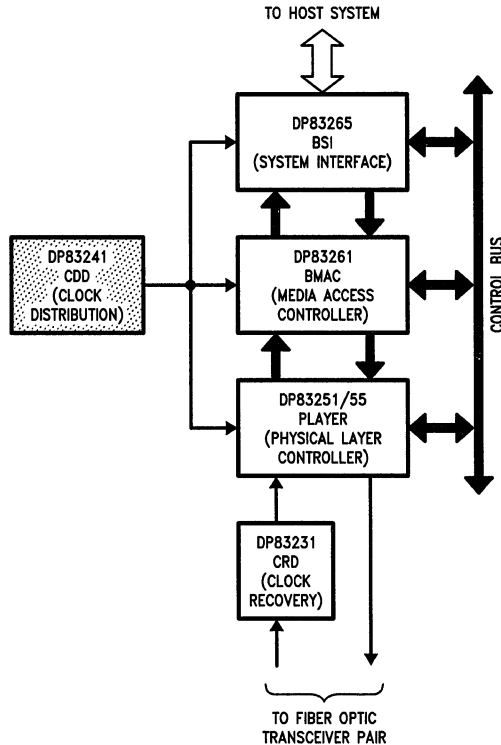


FIGURE 1-1. FDDI Chip Set Block Diagram

TL/F/10385-1

Table of Contents

1.0 FDDI CHIP SET OVERVIEW

2.0 FUNCTIONAL DESCRIPTION

3.0 PIN DESCRIPTIONS

4.0 ELECTRICAL CHARACTERISTICS

- 4.1 Absolute Maximum Ratings
- 4.2 Recommended Operating Conditions
- 4.3 DC Electrical Characteristics
- 4.4 AC Electrical Characteristics

5.0 DETAILED INFORMATION

- 5.1 External Components
- 5.2 Concentrator and Dual Attach Station Configurations
- 5.3 Layout Recommendations
- 5.4 Input and Output Schematics
- 5.5 System Debugging Flowchart
- 5.6 AC Test Circuits

1.0 FDDI Chip Set Overview

National Semiconductor's FDDI chip set consists of five components as shown in *Figure 1-1*. For more information about the other devices in the chip set, consult the appropriate data sheets and application notes.

DP83231 CRD™ Device Clock Recovery Device

The Clock Recovery Device extracts a 125 MHz clock from the incoming bit stream.

Features

- PHY Layer loopback test
- Crystal controlled
- Clock locks in less than 85 μ s

DP83241 CDD™ Device Clock Distribution Device

From a 12.5 MHz reference, the Clock Distribution Device synthesizes the 125 MHz, 25 MHz and 12.5 MHz clocks required by the BSI, BMAC, and PLAYER devices.

DP83251/55 PLAYERTM Device Physical Layer Controller

The PLAYER device implements the Physical Layer (PHY) protocol as defined by the ANSI FDDI PHY X3T9.5 Standard.

Features

- 4B/5B encoders and decoders
- Framing logic
- Elasticity Buffer, Repeat Filter, and Smoother
- Line state detector/generator
- Link error detector
- Configuration switch
- Full duplex operation
- Separate management port that is used to configure and control operation.

In addition, the DP83255 contains an additional PHY__Data.request and PHY__Data.indicate port required for concentration and dual attach stations.

DP83261 BMACTM Device Media Access Controller

The BMAC device implements the Timed Token Media Access Control protocol defined by the ANSI FDDI X3T9.5 MAC Standard.

Features

- All of the standard defined ring service options
- Full duplex operation with through parity
- Supports all FDDI Ring Scheduling Classes (Synchronous, Asynchronous, etc.)
- Supports Individual, Group, Short, Long, and External Addressing
- Generates Beacon, Claim, and Void frames internally
- Extensive ring and station statistics gathering
- Extensions for MAC level bridging
- Separate management port that is used to configure and control operation
- Multi-frame streaming interface

DP83265 BSITM Device System Interface

The BSI Device implements an interface between the National FDDI BMAC device and a host system.

Features

- 32-bit wide Address/Data path with byte parity
- Programmable transfer burst sizes of 4 or 8 32-bit words
- Interfaces to low-cost DRAMs or directly to system bus
- Provides 2 Output and 3 Input Channels
- Supports Header/Info splitting
- Efficient data structures
- Programmable Big or Little Endian alignment
- Full Duplex data path allows transmission to self
- Confirmation status batching services
- Receive frame filtering services
- Operates from 12.5 MHz to 25 MHz synchronously with host system

2.0 Functional Description

The CDD device clocks are all generated from and phase aligned to either a 12.5 MHz crystal oscillator or a TTL input reference source using digital phase locked loop techniques. The architecture of the Clock Distribution Device ensures that the output clocks which are generated have frequency tolerances identical to the 50 PPM crystal reference. When the reference input signal is a backplane signal, the matching of the phase comparator input path delays guarantees phase alignment within 3 ns.

The phase locked loop generates the desired clocks as shown in the device Block Diagram. One of the Local Byte Clock (LBC) phases is connected to the FEEDBK IN input of the phase comparator where its phase and frequency are compared against that of the selected input reference signal. Any phase error between these signals results in a correction of the voltage into the Voltage Controlled Oscillator (VCO) which is proportional to the amount of phase error. The correction voltage tends to drive the frequency of the VCO in the direction which, when divided down, minimizes the LBC to reference signal phase difference. When the phase transition of the LBC occurs before that of the reference input the VCO frequency is sensed as being too fast and produces a negative going correction to the VCO input. This in turn slows down the VCO's frequency and delays the subsequent LBC phase transitions.

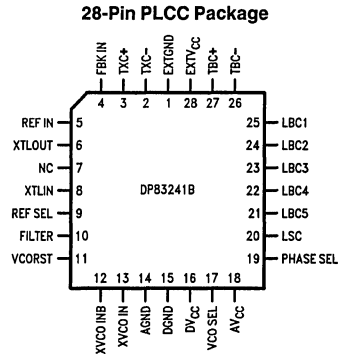
The device's differential 125 MHz ECL transmit clock and differential 12.5 MHz ECL load strobe are used by the PLAYER device to convert data from byte wide NRZ format to serial NRZI format for fiber medium transmission. A 12.5 MHz TTL local byte clock is provided for use by the PLAYER and the BMAC devices. Five phases of the local byte clock are provided for use in large multi-board concentrator configurations to aid in cancelling out backplane delays. A 25 MHz Local Symbol Clock (LSC) is provided which is in phase with the local byte clocks and has a 40% HIGH and 60% LOW duty cycle.

The device provides three user-selectable features. The REF SEL input provides the option to lock the device's outputs to a crystal oscillator or to an external TTL signal (REF IN). The REF IN signal is particularly useful in concentrators where multiple boards need to be phase locked to a com-

mon reference signal. The VCO SEL input provides the option to use the internally provided VCO or an external LC voltage controlled oscillator. Although the stability of the internal VCO should be adequate for most applications the external VCO option provides the means of obtaining the maximum possible oscillator Q. The PHASE SEL input pin provides the option of selecting whether the five phase LBC outputs are phase offset 36 degrees or 72 degrees (8 ns or 16 ns).

The phase locked loop (PLL) elements, with the exception of the loop filter which consist of two capacitors and a resistor, are fully contained within the device. The internal VCO associated with the PLL has been implemented totally within the device and requires no external LC oscillator tank coils, capacitors, or varactors. The external VCO option does provide a means of using these conventional LC oscillator techniques if desired.

Connection Diagram



TL/F/10385-25

Order Number DP83241BV
See NS Package Number V28A

FIGURE 2-1. DP83241 Pinout

Block Diagram

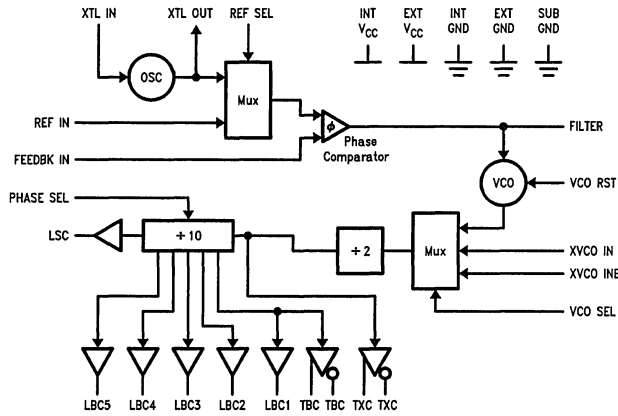


FIGURE 2-2. DP83241 Block Diagram

TL/F/10385-3

3.0 Pin Descriptions

Symbol	Pin No.	I/O	Description
DV _{CC}	16		Digital V_{CC}: Positive power supply for all the internal circuitry intended for operation at 5V ±5% relative to GND. A bypass capacitor should be placed as close as possible across the DV _{CC} and DGND pins.
EXTV _{CC}	28		External V_{CC}: Positive power supply for all the output buffers intended for operation at 5V ±5% relative to GND. A bypass capacitor should be placed as close as possible across the EXTV _{CC} and EXTGND pins.
DGND	15		Digital Ground: Internal circuit power supply return.
EXTGND	1		External Ground: Output buffer power supply return.
AGND	14		Analog Ground: Substrate ground used to ensure proper device biasing and isolation.
AV _{CC}	18		Analog V_{CC}: Positive power supply for the critical analog circuitry, intended for +5V operation ±5% relative to Ground. A bypass cap should be placed as close as possible between AV _{CC} and AGND.
XTL IN	8	I	External Crystal Oscillator Input: XTL IN can also be used as a CMOS compatible reference frequency input for the PLL. This input is selected when REF SEL is at a logical LOW level. The component connections required for oscillator operation are shown in the application diagrams.
XTL OUT	6		External Crystal Oscillator Output: XTL OUT is not intended for use as a logic drive output pin.
REF IN	5	I	Reference Input: TTL compatible input for use as the PLL's phase comparator reference frequency input when the REF SEL is at a logic HI level. This input is for use in concentrator configurations where there are multiple CDD devices at a given site requiring synchronization.
FEEDBK IN	4	I	Feedback Input: TTL compatible input for use as the PLL's phase comparator feedback input to close the loop. This input is intended to be driven from one of the LBCs (Local Byte Clocks). This input is designed to provide the same frequency and within 2 ns of the same phase as REF IN when REF IN is in active operation.
REF SEL	9	I	Reference Select: TTL compatible input which selects either the crystal oscillator inputs XTL IN and XTL OUT or the REF IN inputs as the reference frequency inputs for the PLL. The crystal oscillator inputs are selected when REF SEL is at a logic LOW level and the REF IN input is selected as the reference frequency when REF SEL is at a logic HI level.
FILTER	10	O	Filter: Low pass PLL loop filter pin. A three element filter, consisting of one capacitor in parallel with a resistor and another capacitor, should be connected between this pin and ground.
VCO SEL	17	I	VCO Select: TTL compatible input used to select either the internal VCO or an external VCO through the XVCO IN and XVCO INB pins. The internal VCO is selected when the VCO SEL pin is at a logic HIGH level and the external VCO is selected when at a logic LOW level.
XVCO IN, XVCO INB	13, 12	I	External VCO Inputs: Differential inputs for use with an external VCO. These inputs are D.C. biased to approximately one half V _{CC} , and can be connected to either a full differential VCO, or a single-ended VCO. To use a single-ended VCO, couple the signal into one of the inputs through a series low value capacitor and bypass the other input to GND through a 0.01 μF capacitor. When not in use, ground one input, and let the other float.

3.0 Pin Descriptions (Continued)

Symbol	Pin No.	I/O	Description
VCO RST	11	I	VCO Reset: TTL compatible input used to reset the internal VCO on system power up. This input stops the VCO from oscillating when at a logic HI level thereby reinitializing each of the gates in the ring oscillator.
TXC ⁺ , TXC ⁻	3, 2	O	Transmit Clock: 100K ECL compatible differential outputs for use at 125 MHz as the fiber medium Transmit Clock (TXC) source for the PLAYER device.
TBC ⁺ , TBC ⁻	27, 26	O	Transmit Byte Clock: 100K ECL compatible differential outputs for use at 12.5 MHz as a load strobe or transmit byte clock by the PLAYER device to convert byte wide data to serial format for fiber medium transmission. These outputs are positioned to transition on the falling edge of the TXC + clock output to provide the maximum setup and hold margin. They are also phase coherent with the TTL LBC1 output, but the phase transition occurs approximately 10 ns earlier.
LBC1 thru 5	25, 24, 23, 22, 21	O	Local Byte Clocks: TTL compatible local byte clock outputs which are phase locked to crystal oscillator reference signals. These outputs have a 50% duty cycle waveform at 12.5 MHz. The PHASE SEL input determines whether the five phase outputs are phase offset by 8 ns or 16 ns.
LSC	20	O	Local Symbol Clock: TTL compatible 25 MHz output for driving the BMAC device. This output's negative phase transition is aligned with the LBC1 output transitions and has a 40% HI and 60% LOW duty cycle.
PHASE SEL	19	I	Phase Select: TTL compatible input used to select either a 8 ns or 16 ns phase offset between the 5 local byte clocks. The LBC's are phase offset 8 ns apart when PHASE SEL is at a logic LOW level and 16 ns apart when at a logic HI level.

4.0 Electrical Characteristics

4.1 ABSOLUTE MAXIMUM RATINGS

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature -65°C to +150°C

TTL Signals

Inputs -0.5V to +5.5V
Outputs -0.5V to +5.5V

ECL Signals

Output Current -50 mA

Supplies

EXTV_{CC} to EXTGND -0.5V to +7V

DV_{CC} to DGND -0.5V to +7V

AV_{CC} to AGND -0.5V to +7V

ESD Protection 1500V

4.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter		Min	Typ	Max	Units
V _{CC} to GND	Power Supply		4.75	5.0	5.25	V
V _{IH}	High Level Input Voltage	TTL	2.0			V
		ECL	V _{CC} - 1.165		V _{CC} - 0.880	
V _{IL}	Low Level Input Voltage	TTL			0.8	V
		ECL	V _{CC} - 1.810		V _{CC} - 1.475	
I _{OH}	High Level Output Current	TTL Outputs (Note 1)			-0.4	mA
I _{OL}	Low Level Output Current	TTL Outputs (Note 1)			8.0	mA
F _{VCO}	VCO Frequency (INT or EXT)			250		MHz
F _{REF}	Reference Input Frequency			12.5		MHz
T _A	Operating Temperature		0	25	70	°C

Note 1: TTL outputs include LBC1, LBC2, LBC3, LBC4, LBC5 and LSC.

4.3 DC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Max	Units
V _{IC}	Input Clamp Voltage	I _{IN} = 18 mA		-1.5	V
V _{OH}	High Level Output Voltage	TTL Outputs: I _{OH} = -400 μA	V _{CC} - 2		V
		ECL Outputs: 50Ω Load to V _{CC} - 2V	V _{CC} - 1025	V _{CC} - 880	mV
V _{OL}	Low Level Output Voltage	TTL Outputs: I _{OL} = 8 mA		0.5	V
		ECL Outputs: 50Ω Load to V _{CC} - 2V	V _{CC} - 1810	V _{CC} - 1620	mV
I _I	Max High Level Input Current	TTL Inputs: V _{IN} = 7V		100	μA
I _{IH}	High Level Input Current	TTL Inputs: V _{IN} = 2.7V	-20	20	μA
I _{IL}	Low Level Input Current	TTL Inputs: V _{IN} = 0.4V	-20	20	μA
I _{Filter}	Charge Pump Current	Source	-0.7	+0.7	mA
		Sink	0.2	0.7	mA
		TRI-STATE®	-250	250	nA
I _{CC}	Supply Current			170*	mA

*Includes 60 mA due to external ECL termination of two differential signals.

For 100k ECL output buffers, output levels are specified as:

$$V_{OH_Max} = V_{CC} - 0.88V$$

$$V_{OL_Max} = V_{CC} - 1.62V$$

Since the outputs are differential, the average output level is V_{CC} - 1.25V. The test load per output is 50Ω at V_{CC} - 2V. The external load current through this 50Ω resistor is thus:

$$\begin{aligned} I_{Load} &= [(V_{CC} - 1.25) - (V_{CC} - 2)] / 50 \text{ Amps} \\ &= 0.015 \text{ Amps} \\ &= 15 \text{ mA} \end{aligned}$$

There are 2 pairs of differential ECL signals, so the total ECL current is 60 mA.

4.0 Electrical Characteristics (Continued)

4.4 AC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Max	Units
T1	TBC to TXC	(Note 1)	-1.5	1.5	ns
T2	TBC to LBC1		10	20	ns
T _{Phase1}	LBC1 to LBC2	PHASE SEL = Low	3	13	ns
		PHASE SEL = High	43	53	ns
T _{Phase2}	LBC1 to LBC3	PHASE SEL = Low	11	21	ns
		PHASE SEL = High	11	21	ns
T _{Phase3}	LBC1 to LBC4	PHASE SEL = Low	19	29	ns
		PHASE SEL = High	59	69	ns
T _{Phase4}	LBC1 to LBC5	PHASE SEL = Low	27	37	ns
		PHASE SEL = High	27	37	ns
T3	LSC to LBC1		-4	6	ns
T4	LSC Positive Pulse Width		12	19	ns
T5	REF IN to FEEDBK IN	In Lock (Note 1)	-3	3	ns
T6	TXC Positive Pulse Width	(Note 1)	3.5	4.5	ns
T7	LBC Positive Pulse Width		35	45	ns

Note 1: These parameters are not tested, but are assured by correlation with characterization data.

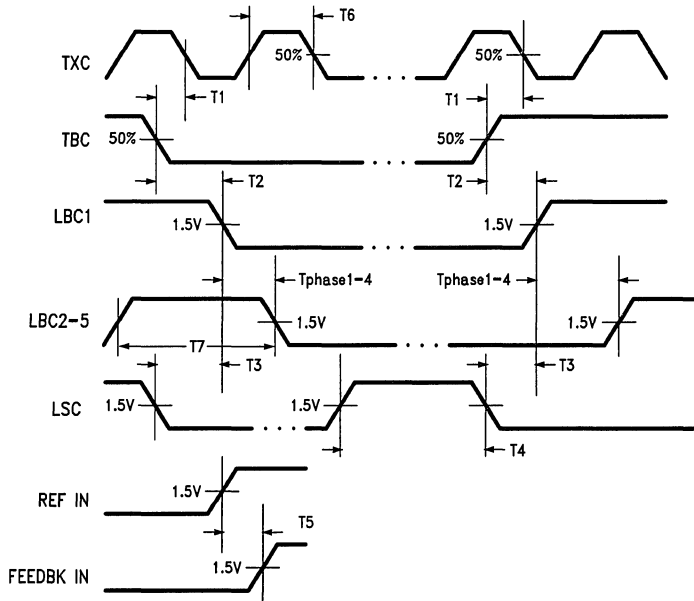


FIGURE 4-1. AC Timing Waveforms

TL/F/10385-4

4.0 Electrical Characteristics (Continued)

4.4 AC ELECTRICAL CHARACTERISTICS (Continued)

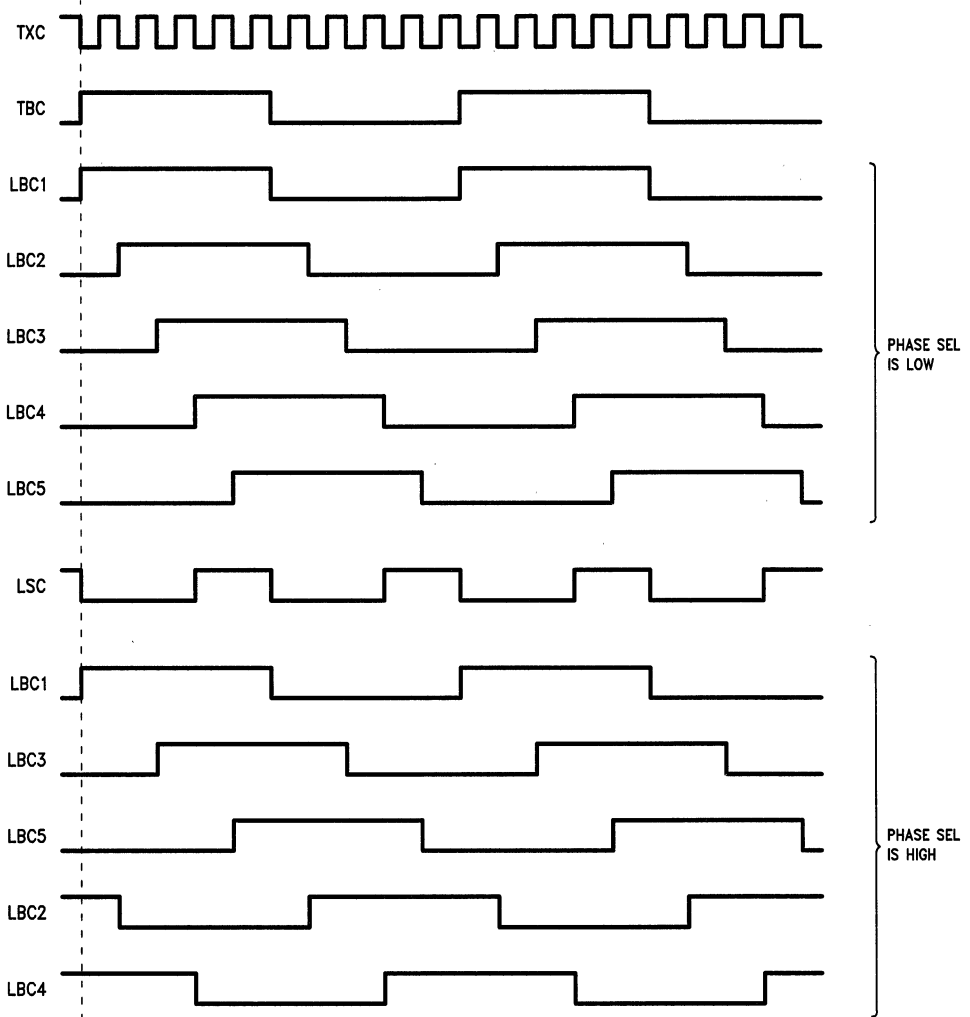


FIGURE 4-2. Typical Clock Relationships

TL/F/10385-5

Loop Filter Calculations

Several constants need to be known in order to determine the loop filter components. They are the loop divide ratio, N , the phase detector gain, K_p , the VCO gain, K_o , the loop bandwidth, W_o , and the phase margin, ϕ . The constants K_p and K_o for the DP83241 are fixed at $80 \mu\text{A}/\text{rad}$ and $0.8 \text{ Grad}/\text{V}$ respectively. N is equal to the VCO frequency divided by the reference frequency. w_o is recommended to be less than $1/20$ th of the reference frequency (times 2π rads). Having found all these constants, the following equations are used to find the component values:

For $\phi = 57^\circ$ phase margin:

$$R_1 = (1.1 N w_o) / (K_p K_o)$$

$$C_1 = (3 K_p K_o) / (N w_o^2)$$

$$C_2 = (0.3 K_p K_o) / (N w_o^2)$$

For a phase margin other than 57° :

$$R_1 = (\text{Cosec } \phi + 1) ((N w_o / 2 K_p K_o))$$

$$C_1 = (\text{Tan } \phi) ((2 K_p K_o) / (N w_o^2))$$

$$C_2 = (\text{Sec } \phi - \text{Tan } \phi) ((K_p K_o) / (N w_o^2))$$

The component equations above are not meant to provide optimal solutions for all implementations.

Let us now design an example system with the following characteristics:

- 12.5 MHz Crystal reference.
- 250 MHz VCO.

Since the VCO is twenty times the frequency of the reference frequency, we get $N = 20$. We will set w_o to be $1/30$ th of the reference frequency or $2.62 \times 10^6 \text{ Rad}$.

From these values we get:

$$C_1 = 1400 \text{ pF}, C_2 = 140 \text{ pF}, \text{ and } R_1 = 900 \Omega$$

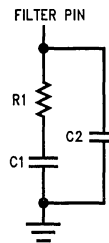
Let us now design an example system with the following characteristics:

- 12.5 MHz Crystal reference.
- 250 MHz External VCO with a gain of $40 \text{ MRad}/\text{V}$.

We will set w_o to be $1/78$ th of the reference frequency or $1.0 \times 10^6 \text{ Rad}$.

From these values we get:

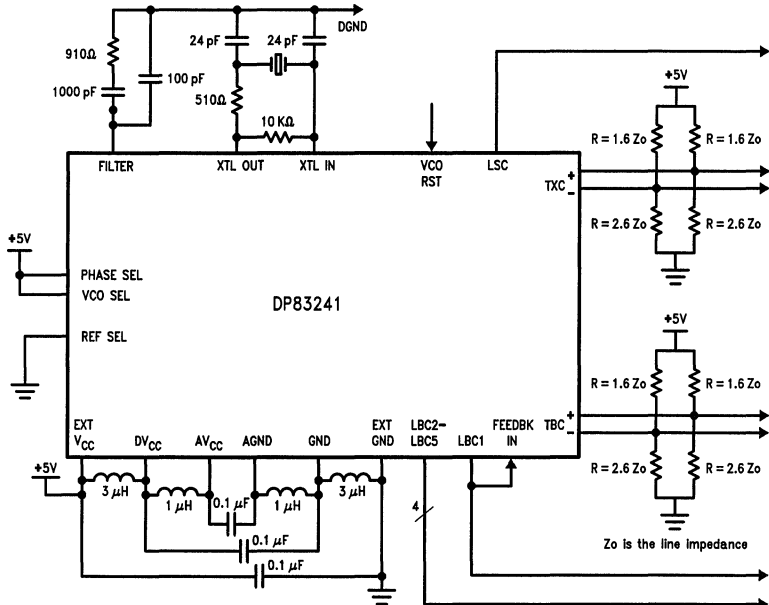
$$C_1 = 470 \text{ pF}, C_2 = 47 \text{ pF}, \text{ and } R_1 = 6.8 \text{ k}\Omega$$



TL/F/10385-27

5.0 Detailed Information

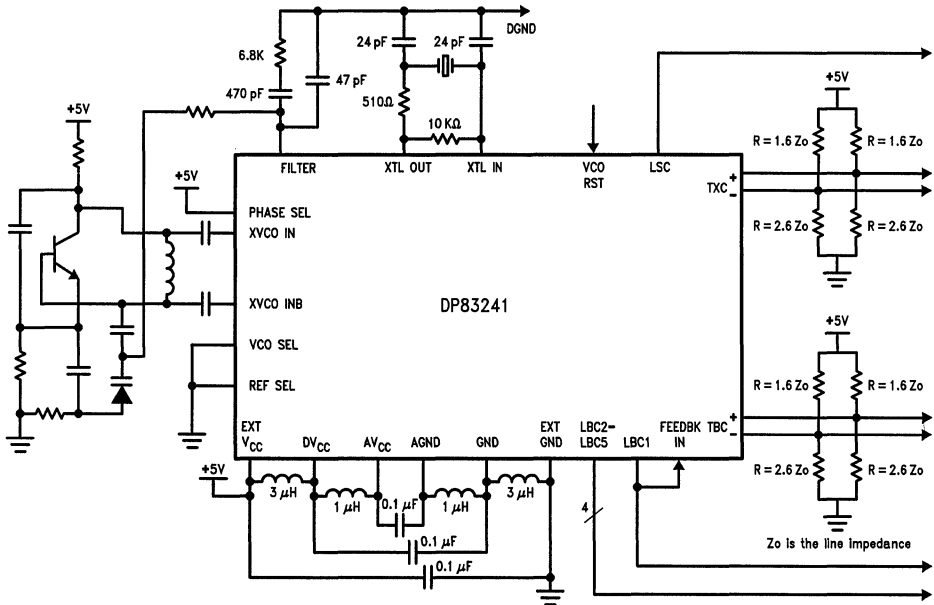
5.1 EXTERNAL COMPONENTS



The Filter components are based on a 12.5 MHz Crystal and a 250 MHz VCO.
All component values $\pm 10\%$.

TL/F/10385-6

FIGURE 5-1. General Wiring Diagram



The Filter components are based on a 12.5 MHz Crystal and an external 250 MHz VCO with a gain of 40 MRad/V.
All component values $\pm 10\%$.

TL/F/10385-7

FIGURE 5-2. General Wiring Diagram with an External VCO

5.0 Detailed Information (Continued)

TABLE 5-1. Special External Components

Crystal Resonator:	Part #: Manufacturer: Key Specifications:	C5410N NEL 12.5000 MHz Center Frequency, 20 PPM Accuracy, 0°C to +70°C 15 pF Load Capacitance
Varactor Diode:	Part #: Manufacturer: Key Specifications:	MV2105 Motorola Cap Tolerance ± 10%
VHF NPN Transistor:	Part #: Manufacturer: Key Specifications:	PN3563 National Semiconductor
Inductor:	Part #: Manufacturer: Key Specifications:	1½ Turns

5.2 CONCENTRATOR AND DUAL ATTACH STATION CONFIGURATIONS

5.2.1 Concentrator Applications

An application where many of the features of the CDD device are used is a FDDI concentrator. A concentrator is used to connect several workstations and peripherals to a single node in the network. A concentrator provides the ability to easily bypass or insert multiple stations into the network. The CDD device in each station is driven from a common oscillator instead of each CDD device being driven by its own crystal. In a small concentrator, the same LBC phase can be used in each station since the data flight time from one board to another is small compared to the LBC

period and the skew between the CDD devices on the two boards is minimal. In a larger concentrator configuration where this skew becomes too large, the data setup time of the downstream station will be directly impacted. One way to avoid this problem is to latch data into the next station. The strobe for the latch will be supplied by one of the LBC outputs from the upstream station's CDD device. An LBC output phase is chosen that occurs after the physical layer data is stable. Assuming that the data and LBC flight times are equal, the LBC output will latch data for the next station. The LBC output phase should be selected to give the optimum setup and hold time for the receiving station's physical layer function.

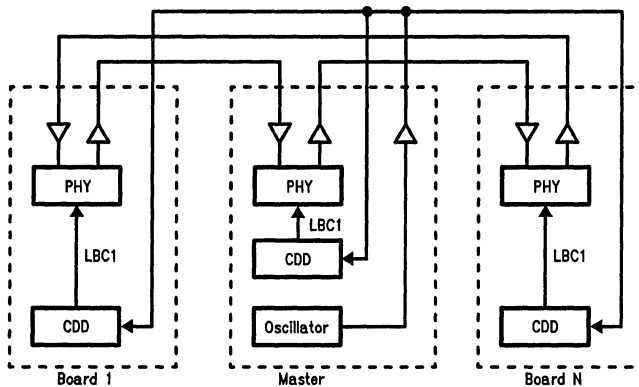
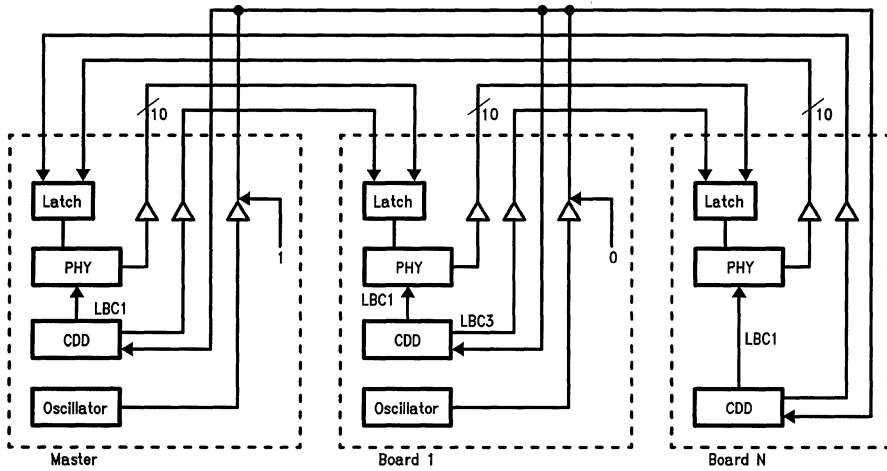


FIGURE 5-3. Small Concentrator Application

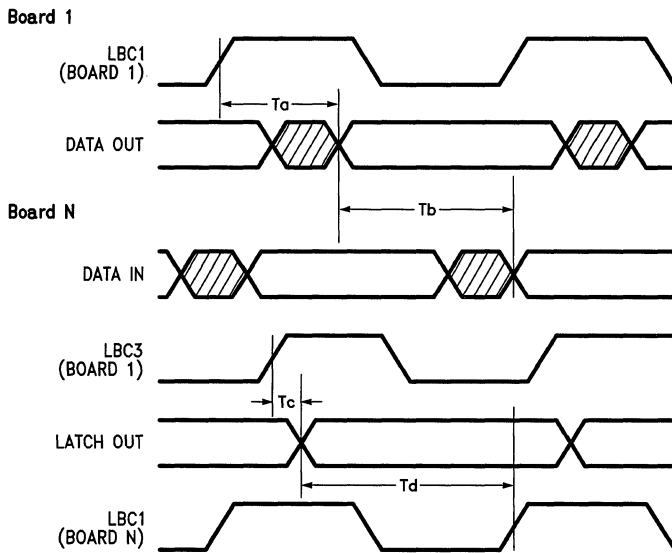
TL/F/10385-8

5.0 Detailed Information (Continued)



TL/F/10385-9

FIGURE 5-4. Large Concentrator Application



TL/F/10385-10

- T_a = Time to latch data out of the Physical Layer (Board 1)
- T_b = Data flight time
- T_c = Latch delay
- T_d = Ideal setup time for incoming data
- $T_d = T_{d1} + T_{d2} + T_{d3}$
- T_{d1} = Reference error between CDD devices
- T_{d2} = Minimum phase resolution of CDD device = 8 ns
- T_{d3} = Setup time

FIGURE 5-5. Large Concentrator Timing

5.0 Detailed Information (Continued)

5.2.2 CDD Device Driving Multiple PLAYER Devices

In a FDDI concentrator or dual attach station, it may be necessary for a single DP83241 Clock Distribution Device (CDD device) to drive multiple DP83251/55 PLAYER devices. Since these PLAYER devices will be running synchronously to each other they must have the same clocks. The easiest way to accomplish this is to have one CDD device drive multiple PLAYER devices.

We are only concerned with the ECL outputs being able to drive multiple loads. The conventional way of directly wiring the one output to many inputs will not work. If the ECL signals are split into multiple traces then reflections will result which may ruin the signal's integrity. An appropriate method, where individual traces with a series resistor connected to each load, is used instead. The series resistor should match the line impedance and be placed as close to the CDD device as possible. The resistor will act as a voltage divider and cut the voltage level of the signal in half. When this modified signal reaches the input of the unterminated gate, reflections will cause the signal to double and the receiving input will see the full voltage swing. The reflection will then travel back towards the CDD device, but the series resistance will stop this action. An emitter pulldown resistor is

needed at the CDD device for this method of routing the ECL signals. The value of this resistor can be calculated from the following equation:

$$R_e (\text{Max}) = \frac{10 Z_o - R_S}{n}$$

Where:

- $R_e (\text{Max})$ — Largest emitter pulldown resistor that can be used
- n — Number of parallel lines being driven
- Z_o — Trace impedance
- R_S — Series damping resistor

Another method for sending the ECL signal to multiple players is to route the ECL signal as a bus line and have each load connected to the bus. The ECL bus line must be terminated only at the very end with a matching impedance (e.g.: a 50Ω line will be terminated with a 50Ω load to $V_{CC} - 2V$). It is preferred that the input pin be directly connected to the bus and not have a signal tap connected to the bus. However, if a tap off the bus is necessary, the shortest possible tap is recommended.

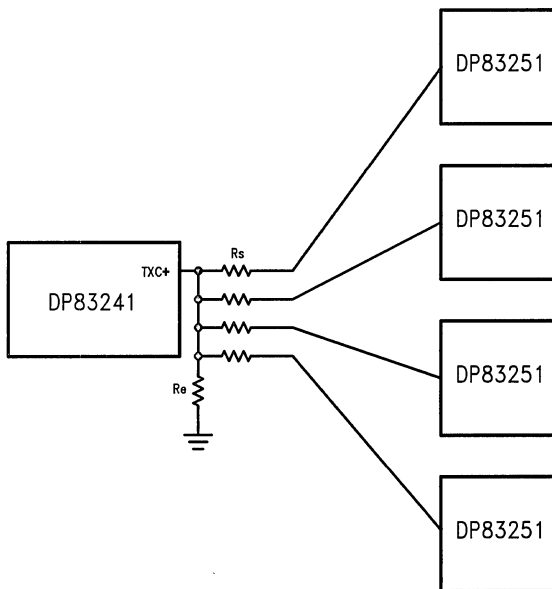


FIGURE 5-7. CDD Device Driving Four PLAYER Devices in Parallel

TL/F/10385-12

5.0 Detailed Information (Continued)

The most conservative method for routing the ECL signals to multiple loads is to use the F100115 Quad Low Skew Driver. This device takes a differential ECL signal and outputs four of the same differential signals with a skew between them of less than 75 ps. Two of these devices will allow the CDD device to drive four PLAYER devices. This setup allows the ECL signals to be routed in a point to point configuration to each PLAYER device.

As with any high speed signal, the routing of the signal must be carefully done. Sharp corners and other changes in trace impedance should be avoided to reduce reflections in high speed signal traces. Traces longer than one inch should have a series or parallel termination scheme. Further system considerations can be found in National's F100K Design Guide. If these methods are followed the DP83241 signals will be able to drive multiple DP83251/55 PLAYER devices without any problems.

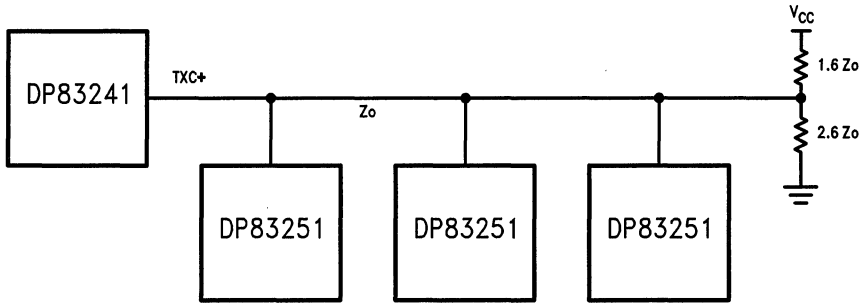
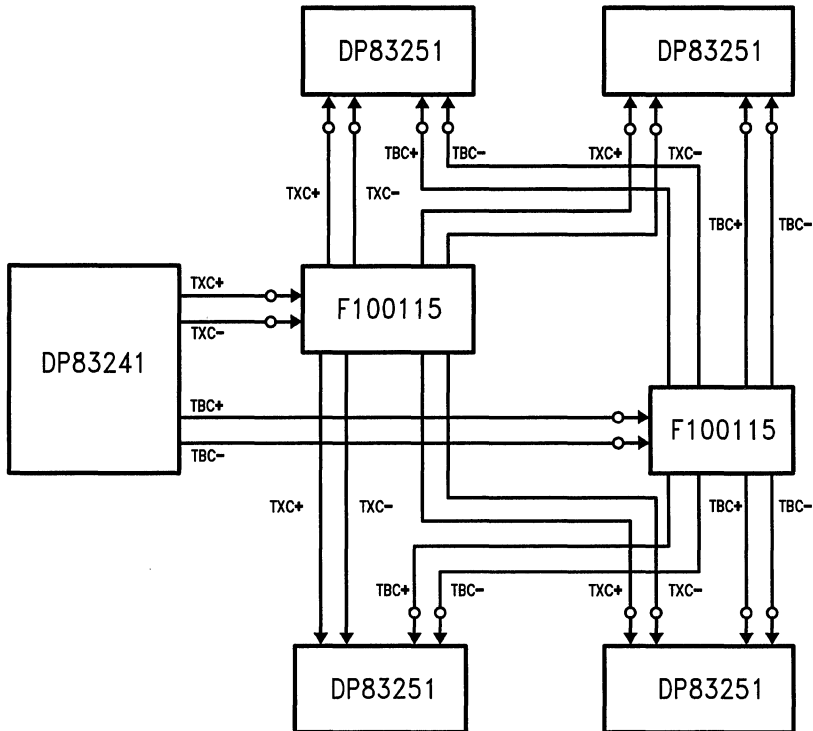


FIGURE 5-8. Proper Bus Line Termination and Connections

TL/F/10385-13



° Parallel Termination

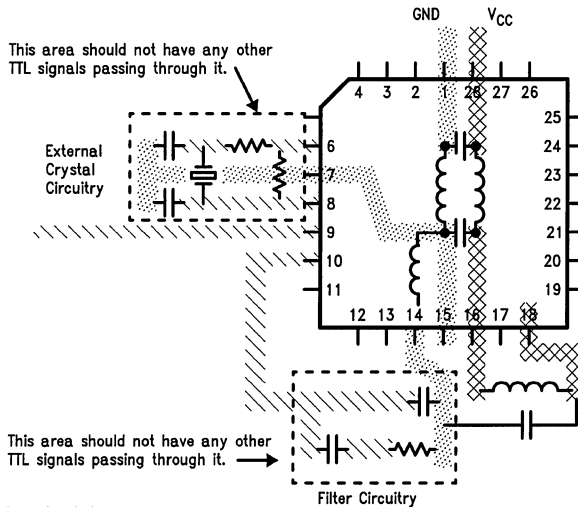
FIGURE 5-9. CDD Device Driving Four PLAYER Devices Using Two F100115 Quad Low Skew Drivers

TL/F/10385-14

5.0 Detailed Information (Continued)

5.3 LAYOUT RECOMMENDATIONS

- The part should be bypassed between the $EXTV_{CC}$ and $EXTGND$ as close to the chip as possible (preferably under the chip using chip caps). The part should also be bypassed between the DV_{CC} and $DGND$ as close to the chip as possible.
- The part should be bypassed between AV_{CC} and $AGND$ as close to the chip as possible.
- No TTL logic lines should pass through the external crystal or filter circuitry areas to avoid the possibility of noise due to crosstalk.
- The filter circuitry should be connected to Ground on an isolated branch off of the $AGND$ pin.
- The DV_{CC} pin should be connected to V_{CC} on an isolated branch off of the $EXTV_{CC}$ pin, preferably being connected through a ferrite bead or small inductor.
- The AV_{CC} pin should be connected to V_{CC} on an isolated branch off of the DV_{CC} pin, preferably being connected through a ferrite bead or small inductor.
- The external crystal circuitry should be connected to Ground on an isolated branch off of the $DGND$ pin.
- The $DGND$ pin should be connected to Ground off of an isolated branch of the $EXTGND$ pin, connected through a ferrite bead or small inductor.
- The $AGND$ pin should be connected to Ground off of an isolated branch of the $DGND$ pin, connected through a ferrite bead or small inductor.
- If the part is being driven by an external reference, the $XTL\ IN$ pin should be tied to either GND or V_{CC} .
- If using a multilayered board with dedicated V_{CC} and Ground planes, ensure that the external crystal circuitry has its own small isolated ground island that is connected to the $AGND$, $DGND$ and $EXTGND$ pins as described above.
- See *Figure 5.1* for component values.
- For best performance tie the $VCORST$ pin to $AGND$.



This drawing was done with convenience in mind.

Note: Pin 7 need not be hooked up.

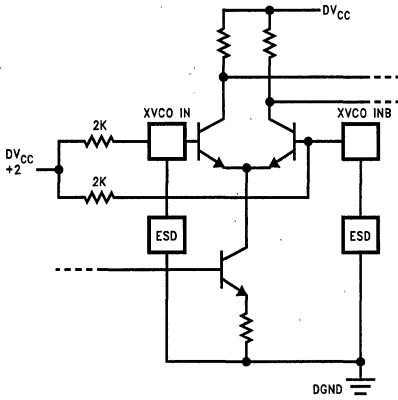
TL/F/10385-15

FIGURE 5-10. Recommended Layout

5.0 Detailed Information (Continued)

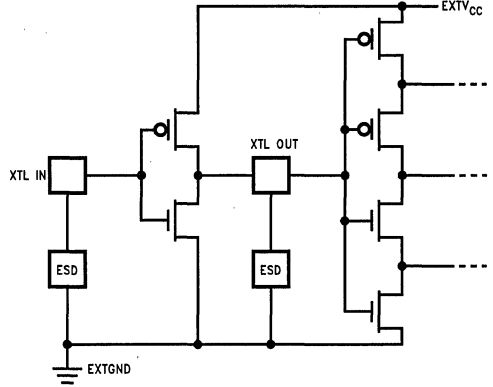
5.4 INPUT AND OUTPUT SCHEMATICS

XVCO IN, XVCO INB



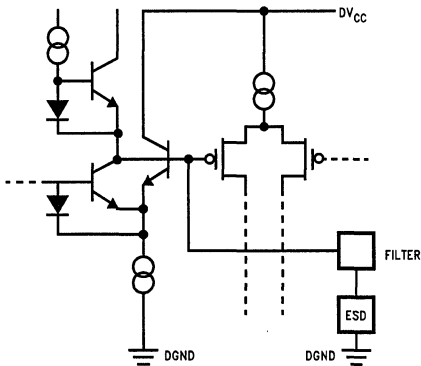
TL/F/10385-16

XTL IN, XTL OUT



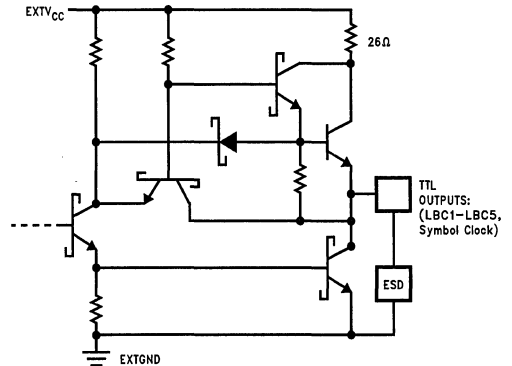
TL/F/10385-17

Filter



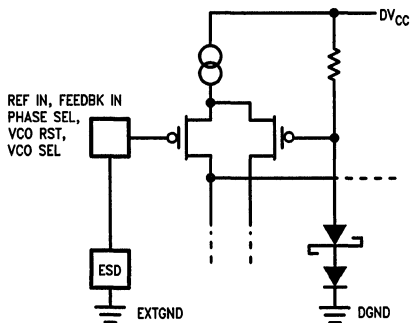
TL/F/10385-18

TTL Outputs: LBC1-LBC5, Symbol Clock



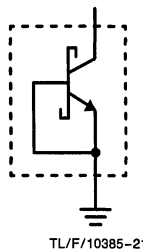
TL/F/10385-19

CMOS Inputs: REF IN, FEEDBK IN, PHASE SEL, VCO RST, VCO SEL



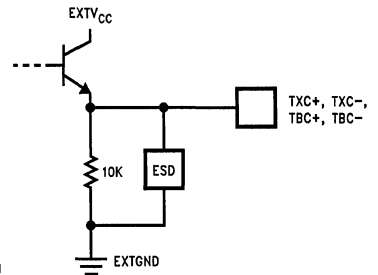
TL/F/10385-20

Typical ESD Structure



TL/F/10385-21

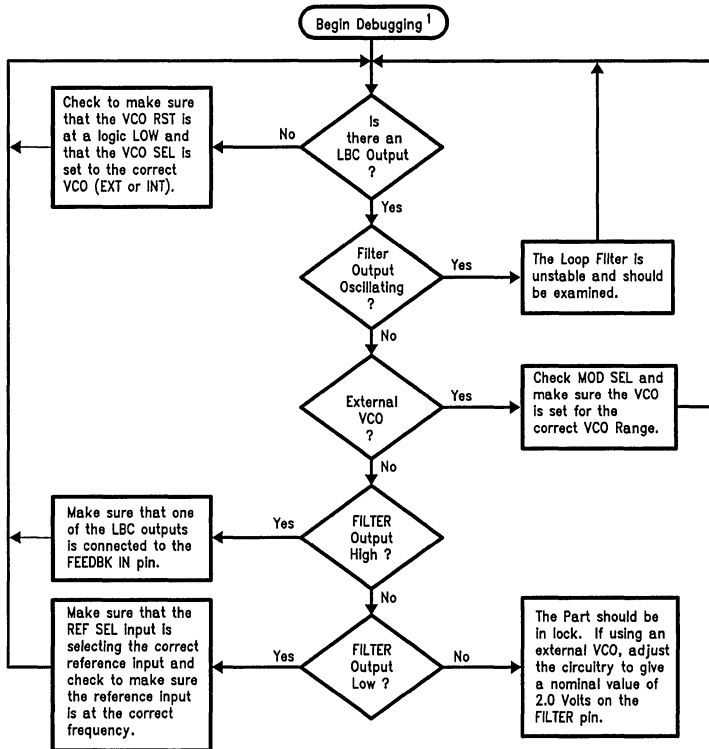
TXC+, TXC-, TBC+, TBC-



TL/F/10385-22

5.0 Detailed Information (Continued)

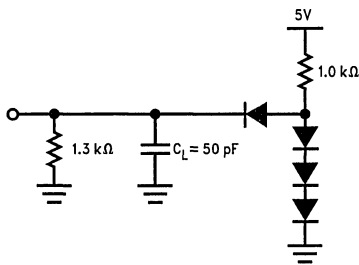
5.5 SYSTEM DEBUGGING FLOWCHART



TL/F/10385-23

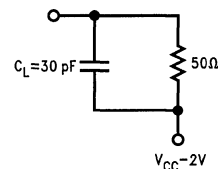
Note 1: If the crystal oscillator is chosen as the input reference source then the XTAL OUT pin should be checked for the correct frequency of oscillation. If the oscillator fails to oscillate then the DC voltage on these pins should be checked and be equal to approximately $V_{CC} \div 2$ (with or without the crystal oscillator present).

5.6 AC TEST CIRCUITS



TL/F/10385-24

FIGURE 5-11. Switching Test Circuit for All TTL Output Signals



TL/F/10385-26

FIGURE 5-12. Switching Test Circuit for All ECL Output Signals



DP83251/55 PLAYER™ Device (FDDI Physical Layer Controller)

General Description

The DP83251/DP83255 PLAYER device implements one Physical Layer (PHY) entity as defined by the Fiber Distributed Data Interface (FDDI) ANSI X3T9.5 Standard. The PLAYER device contains NRZ/NRZI and 4B/5B encoders and decoders, serializer/deserializer, framing logic, elasticity buffer, line state detector/generator, link error detector, repeat filter, smoother, and configuration switch.

Features

- Low power CMOS-BIPOLAR process
- Single 5V supply
- Full duplex operation
- Separate management interface (Control Bus)
- Parity on PHY-MAC Interface and Control Bus Interface
- On-chip configuration switch
- Internal and external loopback
- DP83251 for single attach stations
- DP83255 for dual attach stations

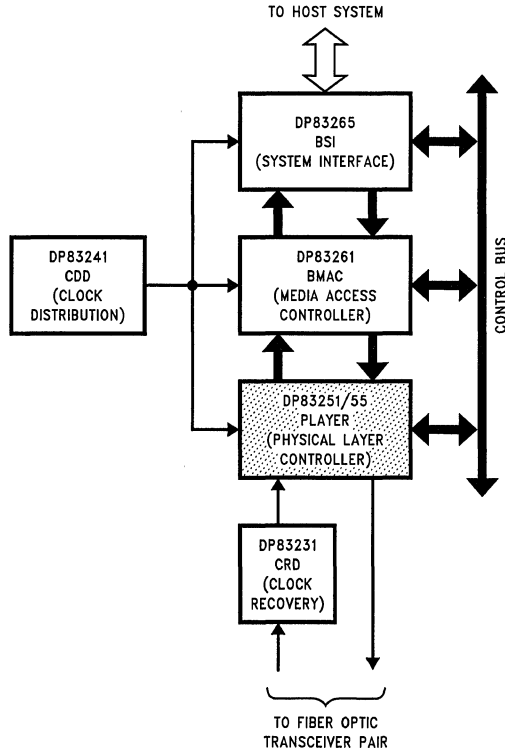


FIGURE 1-1. FDDI Chip Set Block Diagram

TL/F/10386-1

Table of Contents

- 1.0 FDDI CHIP SET OVERVIEW**
- 2.0 ARCHITECTURE DESCRIPTION**
 - 2.1 Overview
 - 2.2 Interfaces
- 3.0 FUNCTIONAL DESCRIPTION**
 - 3.1 Receiver Block
 - 3.2 Transmitter Block
 - 3.3 Configuration Switch
- 4.0 MODES OF OPERATION**
 - 4.1 Run Mode
 - 4.2 Stop Mode
 - 4.3 Loopback Mode
 - 4.4 Cascade Mode
- 5.0 REGISTERS**
- 6.0 PIN DESCRIPTIONS**
 - 6.1 DP83251
 - 6.2 DP83255
- 7.0 ELECTRICAL CHARACTERISTICS**
 - 7.1 Absolute Maximum Ratings
 - 7.2 Recommended Operating Conditions
 - 7.3 DC Electrical Characteristics
 - 7.4 AC Electrical Characteristics
 - 7.5 Test Circuits
- 8.0 DETAILED DESCRIPTIONS**
 - 8.1 Framing Hold Rules
 - 8.2 Noise Events
 - 8.3 Link Errors
 - 8.4 Repeat Filter
 - 8.5 Smoother
 - 8.6 National Byte-wide Code for PHY-MAC Interface

1.0 FDDI Chip Set Overview

National Semiconductor's FDDI chip set consists of five components as shown in *Figure 1-1*. For more information on the other devices of the chip set, consult the appropriate datasheets and application notes.

DP83231 CRD™ Device Clock Recovery Device

The Clock Recovery Device extracts a 125 MHz clock from the incoming bit stream.

Features

- PHY Layer loopback test
- Crystal controlled
- Clock locks in less than 85 μ s

DP83241 CDD™ Device Clock Distribution Device

From a 12.5 MHz reference, the Clock Distribution Device synthesizes the 125 MHz, 25 MHz, and 12.5 MHz clocks required by the BSI, BMAC and PLAYER devices.

DP83251/55 PLAYER™ Device Physical Layer Controller

The PLAYER device implements the Physical Layer (PHY) protocol as defined by the ANSI FDDI PHY X3T9.5 Standard.

Features

- 4B/5B encoders and decoders
- Framing logic
- Elasticity Buffer, Repeat Filter and Smoother
- Line state detector/generator
- Link error detector
- Configuration switch
- Full duplex operation
- Separate management port that is used to configure and control their operation

In addition, the DP83255 contains an additional PHY_Data.request and PHY_Data.indicate port required for concentrators and dual attach stations.

DP83261 BMAC™ Device Media Access Controller

The BMAC device implements the Timed Token Media Access Control protocol defined by the ANSI FDDI X3T9.5 MAC Standard.

Features

- All of the standard defined ring service options
- Full duplex operation with through parity
Supports all FDDI Ring Scheduling Classes (Synchronous, Asynchronous, etc.)
- Supports Individual, Group, Short, Long, and External Addressing
- Generates Beacon, Claim, and Void frames internally
- Extensive ring and station statistic gathering
- Extensions for MAC level bridging
- Separate management port that is used to configure and control their operation
- Multi-frame streaming interface

DP83265 BSI™ Device System Interface

The BSI device implements the interface between the BMAC device and a host system.

Features

- 32-bit wide Address/Data path with byte parity
- Programmable transfer burst sizes of 4 or 8 32-bit words
- Interfaces to low cost DRAMs or directly to system bus
- Provides 2 Output and 3 Input Channels
- Supports Header/Info splitting
- Efficient data structures
- Programmable Big or Little Endian alignment
- Full duplex data path allows transmission to self
- Confirmation status batching services
- Receive frame filtering services
- Operates from 12.5 MHz to 25 MHz synchronously with the host system

2.0 Architecture Description

2.1 OVERVIEW

The PLAYER device is comprised of four blocks: Receiver, Transmitter, Configuration Switch and Control Bus Interface as shown in *Figure 2-1*.

Receiver

During normal operation, the Receiver Block accepts serial data as inputs at the rate of 125 Mbps from the Clock Recovery Device (DP83231). During the Internal Loopback mode of operation, the Receiver Block accepts data from the Transmitter Block as inputs.

The Receiver Block performs the following operations:

- Converts the incoming data stream from NRZI to NRZ, if necessary
- Decodes the data from 5B to 4B coding
- Converts the serial bit stream into 10-bit bytes
- Compensates for the differences between the upstream and local clocks
- Decodes Line States
- Detects link errors

Finally, the Receiver Block presents data symbol pairs (bytes) to the Configuration Switch Block

Configuration Switch

An FDDI station may be in one of three configurations: Isolate, Wrap or Thru. The Configuration Switch supports these configurations by switching the transmitted and received data paths between the PLAYER and BMAC devices.

The configuration switching is performed internally, therefore no external logic is required for this function.

Transmitter

The Transmitter Block accepts 10-bit bytes from the Configuration Switch.

The Transmitter Block performs the following operations:

- Encodes the data from 4B to 5B coding.
- Filters out code violations from the data stream.

- Generates Idle, Master, Halt, Quiet or other user defined symbol pairs upon request.
- Converts the data stream from NRZ to NRZI format ready for transmission, if necessary.
- Provides smoothing function when necessary.

During normal operation, the Transmitter Block presents serial data to the fiber optic transmitter. While in the External Loopback mode, the Transmitter Block presents serial data to the Clock Recovery Device.

Control Bus Interface

The Control Bus Interface allows a user to:

- Program the Configuration Switch.
- Enable/disable functions within the Transmitter and Receiver Blocks (i.e., NRZ/NRZI Encoder, Smoother, PHY Request Data Parity, Line State Generation, Symbol Pair Injection, NRZ/NRZI Decoder, Cascade Mode, etc.).

The Control Bus Interface also performs the following functions:

- Monitors Line States received
- Monitors link errors detected by the Receiver Block
- Monitors other error conditions

2.2 INTERFACES

The PLAYER device connects to external components via 5 functional interfaces: Serial Interface, PHY Port Interface, Control Bus Interface, Clock Interface, and the Miscellaneous Interface.

Serial Interface

The Serial Interface connects the PLAYER device to a fiber optic transmitter (FOTX) and the Clock Recovery Device (DP83231).

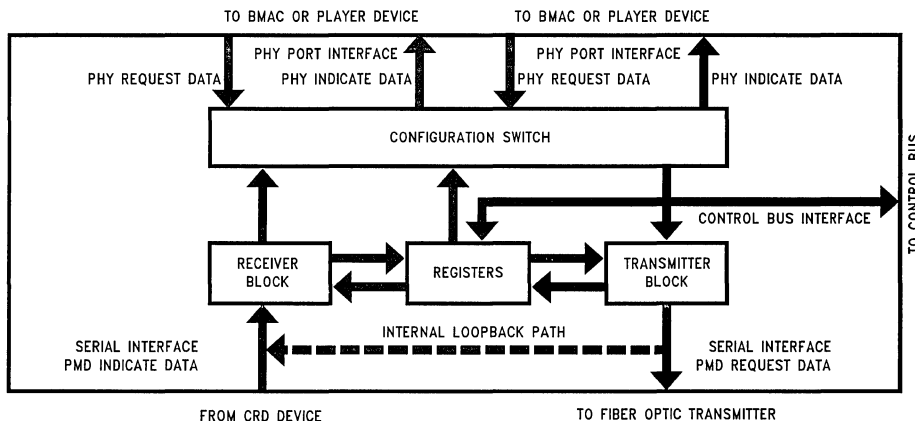


FIGURE 2-1. PLAYER Device Block Diagram

TL/F/10386-2

2.0 Architecture Description

(Continued)

PHY Port Interface

The PHY Port Interface connects the PLAYER device to one or more BMAC devices and/or PLAYER devices. Each PHY Port Interface consists of two byte-wide-interfaces, one for PHY Request data input to the PLAYER device and one for the PHY Indicate data output of the PLAYER device. Each byte-wide interface consists of a parity bit (odd parity), a control bit, and two 4-bit symbols.

The DP8355 PLAYER device has two PHY Port Interfaces and the DP83251 has only one PHY Port Interface.

Control Bus Interface

The Control Bus Interface connects the PLAYER device to a wide variety of microprocessors and microcontrollers. The Control Bus is an asynchronous interface which provides access to 32 8-bit registers.

Clock Interface

The Clock Interface consists of 12.5 MHz and 125 MHz clocks used by the PLAYER device.

The clocks are generated by either the Clock Distribution Device (CDD device) or the Clock Recovery Device (CRD device).

Miscellaneous Interface

The Miscellaneous Interface consists of:

- A reset signal
- User definable sense signals
- User definable enable signals
- Synchronization for cascaded PLAYER devices (a high-performance non-FDDI mode)
- CMOS power and ground, and ECL ground and power

3.0 Functional Description

The PLAYER Device is comprised of four blocks: Receiver, Transmitter, Configuration Switch and Control Bus Interface.

3.1 RECEIVER BLOCK

During normal operation, the Receiver Block accepts serial data as inputs at the rate of 125 Mbps from the Clock Recovery Device (DP83231). During the Internal Loopback mode of operation, the Receiver Block accepts data from the Transmitter Block as input.

The Receiver Block performs the following operations:

- Converts the incoming data stream from NRZI to NRZ, if necessary
- Decodes the data from 5B to 4B coding
- Converts the serial bit stream into National byte-wide code
- Compensates for the differences between the upstream and local clocks
- Decodes Line States
- Detects link errors

Finally, the Receiver Block presents data symbol pairs to the Configuration Switch Block.

The Receiver Block consists of the following functional blocks:

- NRZI to NRZ Decoder
- Shift Register
- Framing Logic
- Symbol Decoder
- Line State Detector
- Elasticity Buffer
- Link Error Detector

See Figure 3-1.

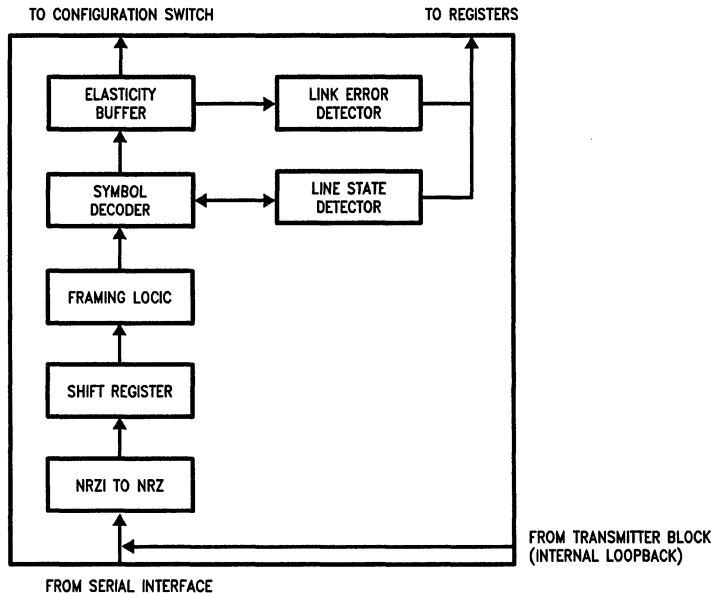


FIGURE 3-1. Receiver Block Diagram

TL/F/10386-3

3.0 Functional Description (Continued)

NRZI TO NRZ DECODER

The NRZI to NRZ Decoder converts Non-Return-To-Zero-Invert-On-Ones data to Non-Return-To-Zero data.

This function can be enabled and disabled through bit 7 (RNRZ) of the Mode Register (MR). When the bit is cleared, it converts the incoming bit stream from NRZI to NRZ. When the bit is set the incoming NRZ bit stream is passed unchanged.

SHIFT REGISTER

The Shift Register converts the serial bit stream into symbol-wide data for the 5B/4B Decoder.

The Shift Register also provides byte-wide data for the Framing Logic.

FRAMING LOGIC

The Framing Logic performs the Framing function by detecting the beginning of a frame or the Halt-Halt or Halt-Quiet symbol pair.

The J-K symbol pair (11000 10001) indicates the beginning of a frame during normal operation. The Halt-Halt (00100 00100) and Halt-Quiet (00100 00000) symbol pairs are detected during Connection Management (CMT).

Framing can be temporarily suspended (i.e. framing hold), in order to maintain data integrity. The Framing Hold rules are explained in Section 8.1.

SYMBOL DECODER

The Symbol Decoder is a two level system. The first level is a 5-bit to 4-bit converter, and the second level is a 4-bit symbol pair to the NSC byte-wide code converter.

The first level latches the received 5-bit symbols and decodes them into 4-bit symbols. Symbols are decoded into two types: data and control. The 4-bit symbols are sent to the Line State Detector and the second level of the Symbol Decoder. See Table 3-1 for the 5B/4B Symbol Decoding list.

The second level translates two 4-bit symbols from the 5B/4B converter and the line state information from the Line State Detector into the National byte-wide code. More details on the National byte-wide code can be found in Section 8.6.

LINE STATE DETECTOR

The FDDI Physical Layer (PHY) standard specifies eight Line States that the Physical Layer can transmit. These Line States are used in the Connection Management process. They are also used to indicate data within a frame during the normal operation.

The Line State Detector detects nine Line States, one more than the required Line States specified in the standard.

The Line States are reported through the Current Receive State Register (CRSR), Receive Condition Register A (RCRA), and Receive Condition Register B (RCRB).

Line States Description

Active Line State

The Line State Detector recognizes the incoming data to be in the Active Line State upon the reception of the Starting Delimiter (JK symbol pair).

The Line State Detector continues to indicate Active Line State while receiving data symbols, Ending Delimiter (T symbols), and Frame Status symbols (R and S) after the JK symbol pair.

Idle Line State

The Line State Detector recognizes the incoming data to be in the Idle Line State upon the reception of 2 Idle symbol pairs nominally (plus up to 9 bits of 1 in start up cases).

Idle Line State indicates the preamble of a frame or the lack for frame transmission during normal operation. Idle Line State is also used in the handshake sequence of the PHY Connection Management process.

TABLE 3-1. Symbol Decoding

Symbol	Incoming 5B	Decoded 4B
0	11110	0000
1	01001	0001
2	10100	0010
3	10101	0011
4	01010	0100
5	01011	0101
6	01110	0110
7	01111	0111
8	10010	1000
9	10011	1001
A	10110	1010
B	10111	1011
C	11010	1100
D	11011	1101
E	11100	1110
F	11101	1111
I (Idle)	11111	1010
H (Halt)	00100	0001
JK (Starting Delimiter)	11000 & 10001	1101
T (Ending Delimiter)	01101	0101
R (Reset)	00111	0110
S (Set)	11001	0111
Q (Quiet)	00000	0010
V (Violation)	00001	0010
V	00010	0010
V	00011	0010
V	00101	0010
V	00110	0010
V	01000	0010
V	01100	0010
V	10000	0010
V'		0011
I'		1011

Notes:

V' denotes PHY Invalid or an Elasticity Buffer stuff byte.

I' denotes Idle symbol in ILS or an Elasticity Buffer stuff byte.

Super Idle Line State

The Line State Detector recognizes the incoming data to be in the Super Idle Line State upon the reception of eight consecutive Idle symbol pairs nominally (plus 1 symbol pair).

The Super Idle Line State is used to insure synchronization.

3.0 Functional Description (Continued)

No Signal Detect

The Line State Detector recognizes the incoming data to be in the No Signal Detect state upon the deassertion of the Signal Detect signal. No Signal Detect indicates that the incoming link is inactive.

Master Line State

The Line State Detector recognizes the incoming data to be in the Master Line State upon the reception of eight consecutive Halt-Quiet symbol pairs nominally (plus up to 2 symbol pairs in start up cases).

The Master Line State is used in the handshake sequence of the PHY Connection Management process.

Halt Line State

The Line State Detector recognizes the incoming data to be in the Halt Line State upon the reception of eight consecutive Halt symbol pairs nominally (plus up to 2 symbol pairs in start up cases).

The Halt Line State is used in the handshake sequence of the PHY Connection Management process.

Quiet Line State

The Line State Detector recognizes the incoming data to be in the Quiet Line State upon the reception of eight consecutive Quiet symbol pairs nominally (plus up to 9 bits of 0 in start up cases).

The Quiet Line State is used in the handshake sequence of the PHY Connection Management process.

Noise Line State

The Line State Detector recognizes the incoming data to be in the Noise Line State upon the reception of 16 noise symbol pairs.

The Noise Line State indicates that data is not received correctly. A detailed description of a noise event can be found in Section 8.2.

Line State Unknown

The Line State Detector recognizes the incoming data to be in the Line State Unknown state upon the reception of one inconsistent symbol pair (i.e. data that is not expected). This may be the beginning of a new line state.

Line State Unknown indicates that data is not received correctly. If the condition persists the noise line state may be entered.

ELASTICITY BUFFER

The Elasticity Buffer performs the function of a "variable depth" FIFO to compensate for clock skews between the Receive Clock (RXC \pm) and the Local Byte Clock (LBC).

Bit 5 (EBOU) of the Receive Condition Register B (RCRB) is set to 1 to indicate an error condition when the Elasticity Buffer cannot compensate for the clock skews.

The Elasticity Buffer will support maximum clock skews of ± 50 ppm with a maximum packet length of 4500 bytes.

To make up for the accumulation of frequency disparity between the two clocks, the Elasticity Buffer will insert or delete Idle symbol pairs in the preamble. Data is written into the byte-wide registers of the Elasticity Buffer with the Re-

ceive Clock, while data is read from the registers with the Local Byte Clock.

The Elasticity Buffer will recenter (i.e. set the read and write pointers to a predetermined distance from each other) upon the detection of a JK or every four byte times during PHY Invalid (i.e. MLS, HLS, QLS, NLS, NSD) and Idle Line State. To resolve metastability problems, the Elasticity Buffer is designed such that a given register cannot be written and read simultaneously under normal operating conditions. In a symbol-wide station, a 5-bit off boundary JK following after a maximum size frame situation may be produced which may result in a small increase in the probability of an error caused by a metastability condition.

LINK ERROR DETECTOR

The Link Error Detector provides continuous monitoring of an active link (i.e. during Active and Idle Line States) to insure that it meets the minimum Bit Error Rate requirement as set by the standard or user to remain on the ring.

Upon detecting a link error, the internal 8-bit Link Error Monitor Counter is decremented. The start value for the Link Error Monitor Counter is programmed through the Link Error Threshold Register (LETR). When the Link Error Monitor Counter reaches zero, bit 4 (LEMT) of the Interrupt Condition Register (ICR) is set to 1. The current value of the Link Error Monitor Counter can be read through the Current Link Error Count Register (CLECR). For higher error rates the current value is an approximate count because the counter rolls over.

There are two ways to determine Link Error Rate: polling and interrupt.

Polling

The Link Error Monitor Counter is set to the value of FF. This start value is programmed through the Link Error Threshold Register (LETR).

Upon detecting a link error, the Current Link Error Counter is decremented.

The Host System reads the current value of the Link Error Monitor Counter via the Current Link Error Count Register (CLECR). The Counter is then reset to FF.

Interrupt

The Link Error Monitor Counter is set to the value of FF. This start value is programmed through the Link Error Threshold Register (LETR).

Upon detecting a link error, the Line Error Monitor Counter is decremented. When the counter reaches zero, bit 4 (LEMT) of the Interrupt Condition Register (ICR) is set to 1, and the interrupt signal goes low.

The Host System is interrupted when the Link Error Monitor Counter reaches 0.

A state table describing Link Errors in more detail can be found in Section 8.3.

Miscellaneous Items

When bit 0 (RUN) of the Mode Register (MR) is set to zero, or when the PLAYER device is reset through the Reset pin (\overline{RST}), the Signal Detect line (TTLSD) is internally forced to zero and the Line State Detector is set to Line State Unknown.

3.0 Functional Description (Continued)

3.2 TRANSMITTER BLOCK

The Transmitter Block accepts 10-bit bytes from the Configuration Switch.

The Transmitter Block performs the following operations:

- Encodes the data from 4B to 5B coding
- Filters out code violations from the data stream
- Is capable of generating Idle, Master, Halt, Quiet, or other user defined symbol pairs
- Converts the data stream from NRZ to NRZI ready for transmission
- Serializes data

During normal operation, the Transmitter Block presents serial data to the fiber optic transmitter.

While in the External Loopback mode, the Transmitter Block presents serial data to the Clock Recovery Device.

The Transmitter Block consists of the following functional blocks:

- Data Registers
- Parity Checker
- 4B/5B Encoder
- Repeat Filter
- Smoother
- Line State Generator
- Injection Control Logic
- Shift Register
- NRZ to NRZI Encoder

See Figure 3-2.

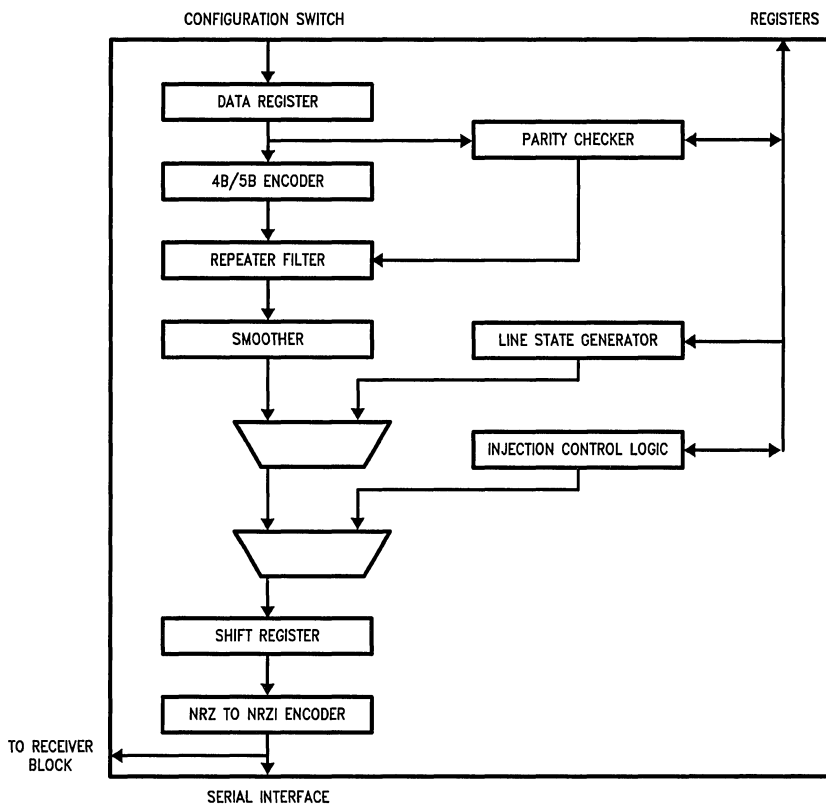


FIGURE 3-2. Transmitter Block Diagram

TL/F/10386-4

3.0 Functional Description (Continued)

DATA REGISTERS

Data from the Configuration Switch is stored in the Data Registers. The 10-bit byte-wide data consists of a parity bit, a control bit, and two 4-bit symbols as shown in *Figure 3-3*.

b9	b8	b7	b0
Parity Bit	Control Bit	Data Bits	

FIGURE 3-3. Byte-Wide Data

PARITY CHECKER

The Parity Checker verifies that the parity bit in the Data Register represents odd parity (i.e. odd number of 1s).

The parity checking is enabled and disabled through bit 6 (PRDPE) of the Current Transmit State Register (CTSR).

If a parity error occurs, the Parity Checker will set bit 0 (DPE) in the Interrupt Condition Register (ICR) and report the error to the Repeat Filter.

4B/5B ENCODER

The 4B/5B Encoder converts the two 4-bit symbols from the Configuration Switch into their respective 5-bit codes.

See Table 3-2 for the Symbol Encoding list.

REPEAT FILTER

The Repeat Filter is used to prevent the propagation of code violations in data frames, to the downstream station.

Upon receiving violations in data frames, the Repeat Filter replaces them with two Halt symbol pairs followed by Idle symbols. Thus the code violations are isolated and recovered at each link and will not be propagated throughout the entire ring.

Details on Repeat Filter operation are described in Section 8.4.

SMOOTHER

The Smoother is used to keep the preamble length of a frame to a minimum of 6 Idle symbol pairs.

Idle symbols in the preamble of a frame may have been added or deleted by each station to compensate for the difference between the Receive Clock and its Local Clock. The preamble needs to be maintained at a minimum length to allow stations enough time to complete processing of one frame and prepare to receive another. Without the Smoother function, the minimum preamble length (6 Idle symbol pairs) may not be maintained as several stations may consecutively delete Idle symbols.

The Smoother attempts to keep the number of Idle symbol pairs in the preamble at 7 by:

- Deleting an Idle symbol pair in preambles which have more than 7 Idle symbol pairs

and/or

- Inserting an Idle symbol pair in preambles which have less than 7 idle symbol pairs (i.e. Extend State).

The Smoother Counter starts counting upon detecting an Idle symbol pair. It stops counting upon detecting a JK symbol pair.

More details on the operation of the Smoother can be found in Section 8.5.

LINE STATE GENERATOR

The Line State Generator allows the transmission of the PHY Request data and can also generate and transmit Idle, Master, Halt, or Quiet symbol pairs which can be used to implement the Connection Management procedures as specified in the FDDI Station Management (SMT) document.

The Line State Generator is programmed through Transmit bits 0 to 2 (TM<2:0>) of the Current Transmit State Register (CTSR).

Based on the setting of these bits, the Transmitter Block operates in the Transmit Modes where the Line State Generator overwrites the Repeat Filter and Smoother outputs.

See Table 3-3 for the listing of the Transmit Modes.

TABLE 3-2. 4B/5B Symbol Encoding

Symbol	4B Code	Outgoing 5B
0	0000	11110
1	0001	01001
2	0010	10100
3	0011	10101
4	0100	01010
5	0101	01011
6	0110	01110
7	0111	01111
8	1000	10010
9	1001	10011
A	1010	10110
B	1011	10111
C	1100	11010
D	1101	11011
E	1110	11100
F	1111	11101
N	0000	11110 or 11111
JK (Starting Delimiter)	1101	11000 and 10001
T (Ending Delimiter)	0100 or 0101	01101
R (Reset)	0110	00111
S (Set)	0111	11001

TABLE 3-3. Transmit Modes

Active Transmit Mode	Normal Transmission Mode
Off Transmit Mode	Transmit Quiet symbol pairs and disable the Fiber Optic Transmitter
Idle Transmit Mode	Transmit Idle symbol pairs
Master Transmit Mode	Transmit Halt-Quiet symbol pairs
Quiet Transmit Mode	Transmit Quiet symbol pairs
Reserved Transmit Mode	Reserved for future use. If selected, Quiet symbol pairs will be transmitted.
Halt Transmit Mode	Transmit Halt symbol pairs

3.0 Functional Description (Continued)

INJECTION CONTROL LOGIC

The Injection Control Logic replaces the data stream with a programmable symbol pair. This function is used to transmit data other than the normal data frame or Line States.

The Injection Symbols overwrite the Line State Generator (Transmit Modes) and the Repeat Filter and Smoother outputs.

These programmable symbol pairs are stored in the Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB). The Injection Threshold Register (IJTR) determines where the Injection Symbol pair will replace the data symbols.

The Injection Control Logic is programmed through the bits 0 and 1 (IC<1:0>) of the Current Transmit State Register (CTSR) to one of the following Injection Modes (see Figure 3-4):

1. No Injection (i.e. normal operation)
2. One Shot
3. Periodic
4. Continuous

In the No Injection mode, the data stream is transmitted unchanged.

In the One Shot mode, ISRA and ISRB are injected once on the nth byte after a JK, where n is the programmed value specified in the Injection Threshold Register.

In the Periodic mode, ISRA and ISRB are injected every nth symbol.

In the Continuous mode, all data symbols are replaced with the contents of ISRA and ISRB. This is the same as periodic mode with IJTR = 0.

SHIFT REGISTER

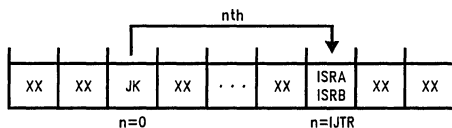
The Shift Register converts encoded parallel data to serial data. The parallel data is clocked into the Shift Register by the Transmit Byte Clock (TBC±), and clocked out by the Transmit Bit Clock (TxC±).

NRZ TO NRZI ENCODER

The NRZ to NRZI Encoder converts the serial Non-Return-To-Zero data to Non-Return-To-Zero-Invert-On-One data.

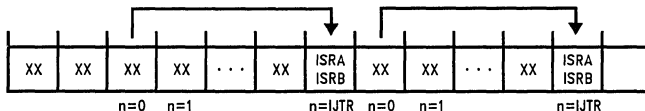
This function can be enabled and disabled through bit 6 (TNRZ) of the Mode Register (MR). When programmed to "0", it converts the bit stream from NRZ to NRZI. When programmed to "1", the bit stream is transmitted NRZ.

One Shot (Notes 1, 3)



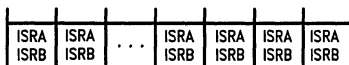
TL/F/10386-5

Periodic (Notes 2, 3)



TL/F/10386-6

Continuous (Note 3)



TL/F/10386-33

Where

ISRA: Injection Symbol Register A

ISRB: Injection Symbol Register B

IJTR: Injection Threshold Register

Note 1: In one shot when $n + 0$ the JK is replaced.

Note 2: In periodic when $n = 0$ all symbols are replaced.

Note 3: Max value on $n = 255$.

FIGURE 3-4. Injection Modes

3.0 Functional Description (Continued)

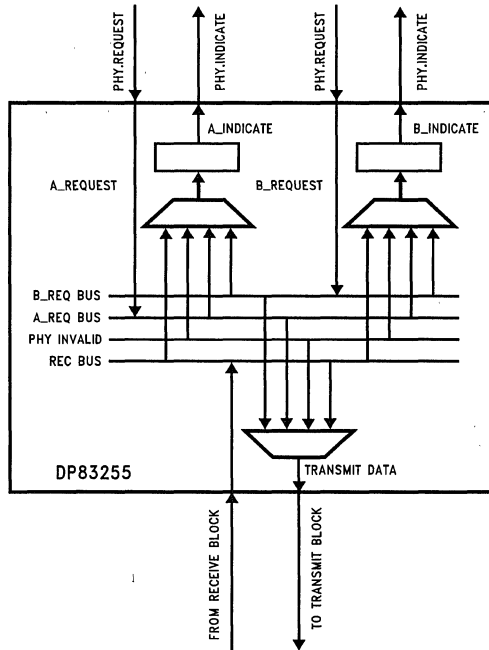
3.3 CONFIGURATION SWITCH

The Configuration Switch consists of a set of multiplexors and latches which allow the PLAYER device to configure the data paths without the need of external logic. The Configuration Switch is controlled through the Configuration Register (CR).

The Configuration Switch has four internal buses, the A_Request bus, the B_Request bus, the Receive bus, and the PHY_Invalid bus. The two Request buses can be driven by external input data connected to the external PHY Port Interface. The Receive bus is internally connected to the Receive Block of the PLAYER device, while the PHY_Invalid bus has a fixed 10-bit LSU pattern, useful during the connection process. The configuration switch also has three internal multiplexors, each can select any of the four buses to connect to its respective data path. The first two are

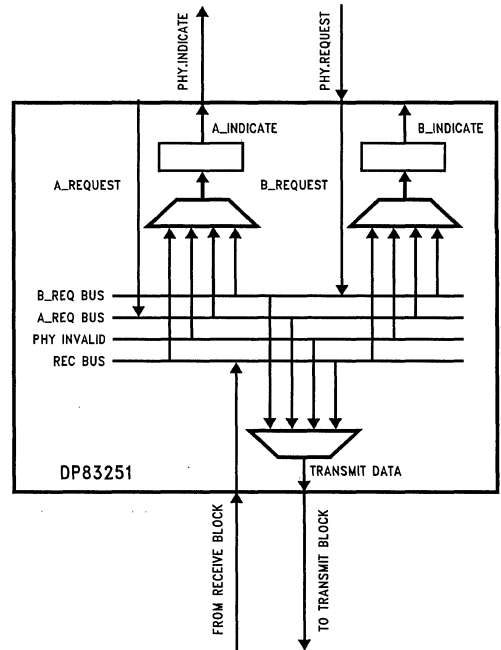
PHY Port Interface output data paths, A_Indicate and B_Indicate, that can drive output data paths of the external PHY Port Interface. The third output data path is connected internally to the Transmit Block.

The Configuration Switch is the same on both the DP83251 device and the DP83255 device. However, the DP83255 has two PHY Port interfaces connected to the Configuration Switch, whereas the DP83251 has one PHY Port Interface. The DP83255 uses the A_Request and A_Indicate paths as one PHY Port Interface and the B_Request and B_Indicate paths as the other PHY Port interface (see *Figure 3-5a*). The DP83251, having only one port interface, uses the B_Request and A_Indicate paths as its external port. The A_Request and B_Indicate paths of the DP83251 are null connections and are not used by this device (see *Figure 3-5b*).



TL/F/10386-7

FIGURE 3-5a. Configuration Switch Block Diagram for DP83255



TL/F/10386-34

FIGURE 3-5b. Configuration Switch Block Diagram for DP83251

3.0 Functional Description (Continued)

STATION CONFIGURATIONS

Single Attach Station (SAS)

The Single Attach Station can be connected to either the Primary or Secondary ring via a Concentrator. Only 1 MAC is needed in a SAS.

Both the DP83251 and DP83255 can be used in a Single Attach Station. The DP83251 can be connected to the MAC via its only PHY Port Interface. The DP83255 can be connected to the MAC via either of the 2 PHY Port Interfaces. See *Figures 3-6 and 3-7*.

Dual Attach Station (DAS)

A Dual Attach Station can be connected directly to the dual ring. There are two types of Dual Attach Stations: DAS with a Single MAC and DAS with Dual MACs. See *Figures 3-8 and 3-9*.

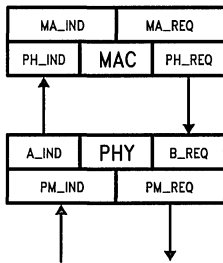
Although two DP83251s can be connected together to build a Dual Attach Station, it is recommended that the DP83255 is to be used for this type of station configuration.

A DAS with Single MAC can be configured as follows:

- B_Indicate data of PHY_A is connected to A_Request input of PHY_B. B_Request input of PHY_A is connected to A_Indicate output of PHY_B.
- The MAC can be connected to either the A_Request input and the A_Indicate output of PHY_A or the B_Request input and the B_Indicate output of PHY_B.

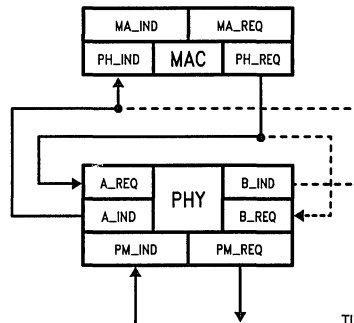
The DAS with Dual MACs can be configured as follows:

- B_Indicate data of PHY_A is connected to A_Request input of PHY_B. B_Request input of PHY_A is connected to A_Indicate output of PHY_B.
- The MAC_1 is connected to the B_Indicate output and the B_Request Input of PHY_B.
- The MAC_2 is connected to the A_Indicate output and the A_Request Input of PHY_A.



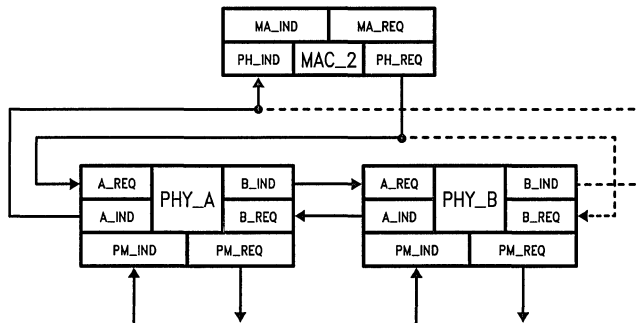
TL/F/10386-8

FIGURE 3-6. Single Attach Station Using the DP83251



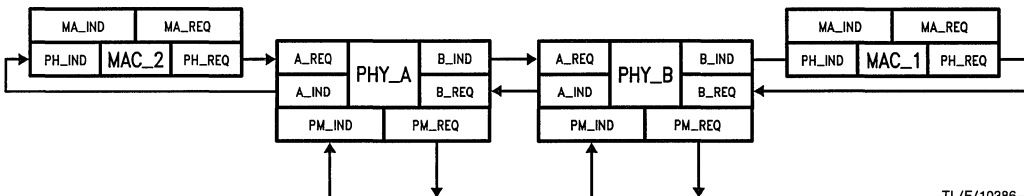
TL/F/10386-9

FIGURE 3-7. Single Attach Station (SAS) Using the DP83255



TL/F/10386-10

FIGURE 3-8. Dual Attachment Station (DAS), Single MAC



TL/F/10386-11

FIGURE 3-9. Dual Attachment Station (DAS), Dual MACs

3.0 Functional Description (Continued)

CONCENTRATOR CONFIGURATIONS

There are 2 types of Concentrators: Single Attach and Dual Attach. These Concentrators can be designed with or without the MAC(s). Its configuration is determined based upon its type and the number of active MACs in the Concentrator.

Using the PLAYER devices, a Concentrator can be built with many different configurations without the need of any external logic.

Both the DP83251 and DP83255 can be used to build a Single Attach Concentrator. Only the DP83255 is recommended for the Dual Attach Concentrator design.

See Application Note 675, Designing FDDI Concentrators and Application Note 741, Differentiating FDDI Concentrators for further information.

Concepts

A Concentrator is comprised of 2 parts: the Dual Ring Connect portion and the Master Ports.

The Dual Ring Connection portion connects the Concentrator to the dual ring directly or to another Concentrator. If the Concentrator is connected directly to the dual ring, it is a part of the "Dual Ring Tree". If the Concentrator is connected to another Concentrator, it is a "Branch" of the "Dual Ring Tree".

The Master Ports connect the Concentrator to its "Slaves". A Slave could be a Single Attach Station or another Concentrator (thus forming another Branch of the Dual Ring Tree).

When a MAC in a concentrator is connected to the Primary or Secondary Ring, it is required to be situated at the exit port of that concentrator (i.e. its PH__IND is connected to the IND Interface of the last Master Port in the Concentrator (PHY__M n) that is connected to that ring).

A Concentrator can have two MACs connected to both the Primary and Secondary rings. In addition, a Roving MAC can be included in the Concentrator configuration. A Roving MAC can be used to test the stations connected to the Concentrator before allowing them to join the Dual Ring. This may require external multiplexers.

Single Attach Concentrator

A Single Attach Concentrator is a Concentrator that has only one PHY at the Dual Ring Connect side. It cannot, therefore, be connected directly to the Dual Ring. A Single Attach Concentrator is a Branch to the Dual Ring Tree. It is connected to the ring as a Slave of another Concentrator.

Multiple Single Attach Concentrators can be connected together hierarchically to build multiple levels of branches in a Dual Ring.

The Single Attach Concentrator can be connected to either the Primary or Secondary ring depending on the connection with its Concentrator (the Concentrator that it is connected to a slave).

Figure 3-10 shows a Single Attach Concentrator with a Single MAC.

Dual Attach Concentrator

A Dual Attach Concentrator is a Concentrator that has two PHYs at the Dual Ring Connect side. It is connected directly to the dual ring and is a part of the Dual Ring Tree.

The Dual Attach Concentrator is connected to both the Primary and Secondary rings.

Dual Attach Concentrator with Single MAC

Figure 3-11 shows a Dual Attach Concentrator with a Single MAC.

Because the Concentrator has one MAC, it can only transmit and receive frames on the ring where the MAC is connected. The Concentrator can only repeat frames on the other ring.

Dual Attach Concentrator with Dual MACs

Figure 3-12 shows Dual Attach Concentrator with Dual MACs.

Because the Concentrator has two MACs, it can transmit and receive frames on both the Primary and Secondary rings.

3.0 Functional Description (Continued)

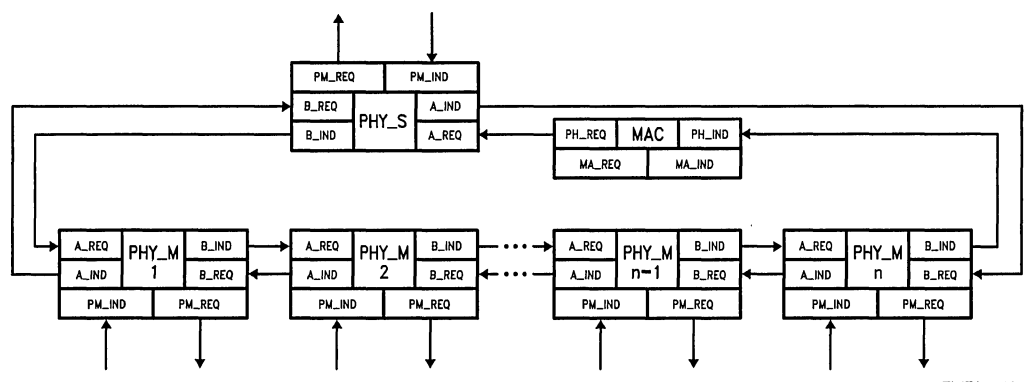


FIGURE 3-10. Single Attach Concentrator (SAC), Single MAC

TL/F/10386-12

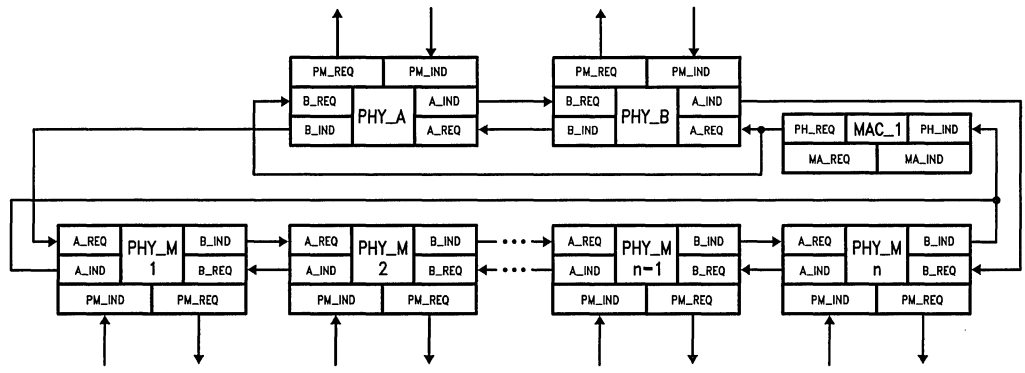


FIGURE 3-11. Dual Attach Concentrator (DAC), Single MAC

TL/F/10386-13

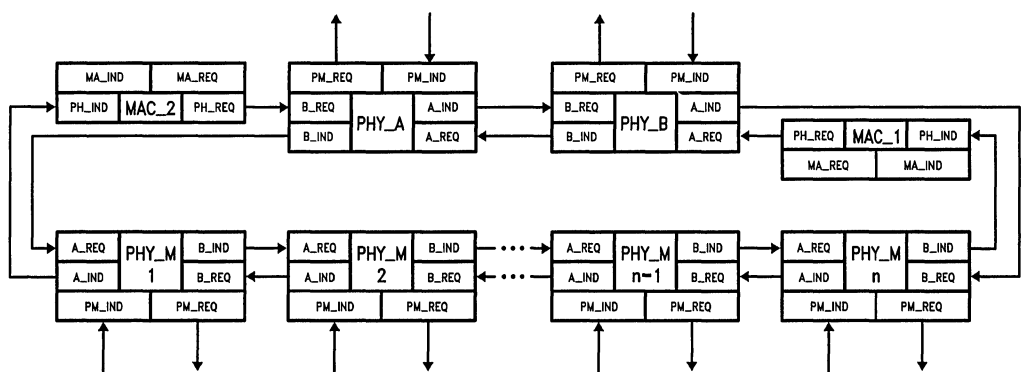


FIGURE 3-12. Dual Attach Concentrator (DAC), Dual MACs

TL/F/10386-14

4.0 Modes of Operation

The PLAYER device can operate in 4 basic modes: RUN, STOP, LOOPBACK, and CASCADE.

4.1 RUN MODE

RUN is the normal mode of operation.

In this mode, the PLAYER device is configured to be connected to the media via the Fiber Optic Transmitter and Receiver at the Serial Interface. It is also connected to other PLAYER device(s) and/or BMAC device(s) via the Port A and Port B Interfaces.

While operating in the Run mode, the PLAYER device receives and transmits Line States (Quiet, Halt, Master, Idle) and frames (Active Line State).

4.2 STOP MODE

The PLAYER device operates in the STOP mode while it is being initialized or configured.

The PLAYER device is also reset to the STOP mode automatically when the RST pin (pin 71 on the DP83251 and pin 111 on the DP83255) is set to ground.

When in STOP mode, the PLAYER device performs the following functions:

- Resets the Repeat Filter.
- Resets the Smoother.
- Resets the Receiver Block Line State Counters.
- Flushes the Elasticity Buffer.
- Forces Line State Unknown in the Receiver Block.
- Outputs LSU symbol pairs (0 1 0011 1010) through the PHY Data Indicate pins (AIP, AIC, AID <7:0>, BIP, BIC, BID <7:0>).
- Outputs Quiet symbol pairs through the PMD Data Request pins (TXD±).
- Resets all Control Bus register contents to zero or default values.

4.3 LOOPBACK MODE

The PLAYER device provides three types of loopback tests: Configuration Switch Loopback, Internal Loopback, and External Loopback. These Loopback modes can be used to test different portions of the device.

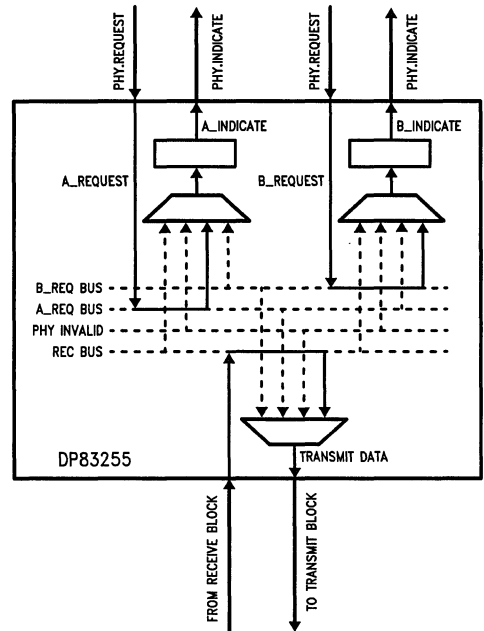
4.3.1 Configuration Switch Loopback

The Configuration Switch Loopback can be used to test the data paths of the BMAC device(s) that are connected to the PLAYER device before transmitting and receiving data through the network.

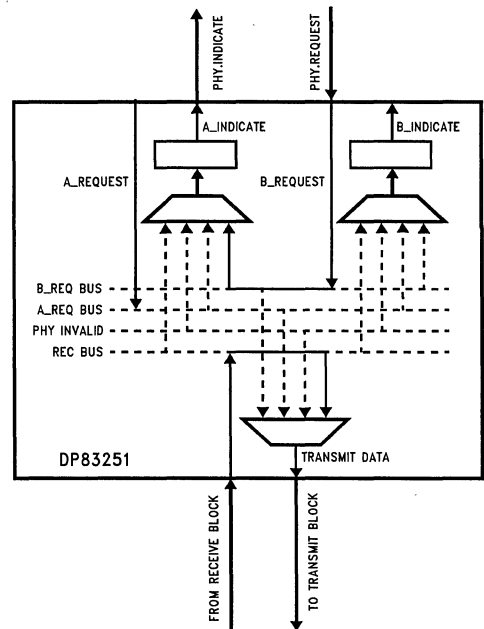
In the Configuration Switch Loopback mode, the PLAYER device performs the following functions:

- Selects Port A PHY Request Data, Port B PHY Request Data, or PHY Invalid to connect to Port A PHY Indicate Data via the A_IND Mux.
- Selects Port A PHY Request Data, Port B PHY Request Data, or PHY Invalid to connect to Port B PHY Indicate Data via the B_IND Mux.
- Connects data from the Receiver Block to the Transmitter Block via the Transmitter_Mux. (The PLAYER device is repeating incoming data from the media in the Configuration Switch Loopback mode.)

See Figure 4-1a and 4-1b for block diagrams.



TL/F/10386-42
FIGURE 4-1a. Configuration Switch Loopback for DP83255



TL/F/10386-43
FIGURE 4-1b. Configuration Switch Loopback for DP83251
FIGURE 4-1.

4.0 Modes of Operation (Continued)

4.3.2 Internal Loopback

The Internal Loopback mode can be used to test the functionality of the PLAYER device and to test the data paths between the PLAYER and BMAC devices before ring insertion.

When in the Internal Loopback mode, the PLAYER device performs the following functions:

- Directs the output data of the Transmitter Block to the input of the Receiver Block through internal paths (see *Figure 2-1* PLAYER Device Block Diagram).
- Ignores the PMD Data Indicate pins ($RXD\pm$ and $RXC\pm$),
- Outputs Quiet symbols through the PMD Data Request pins ($TXD\pm$), and

- Outputs Quiet symbols through the External Loopback Data pins ($LBD\pm$).

The level of the Quiet symbols transmitted through the $TXC\pm$ pins is programmable through the Transmit Quiet Level bit of the Mode Register.

The level of the Quiet symbols transmitted through the $LBD\pm$ pins is always high, regardless of the Transmit Quiet Level bit of the Mode Register.

If both Internal Loopback and External Loopback modes are selected, Internal Loopback mode will have priority over External Loopback mode.

See *Figure 4-2* for a block diagram.

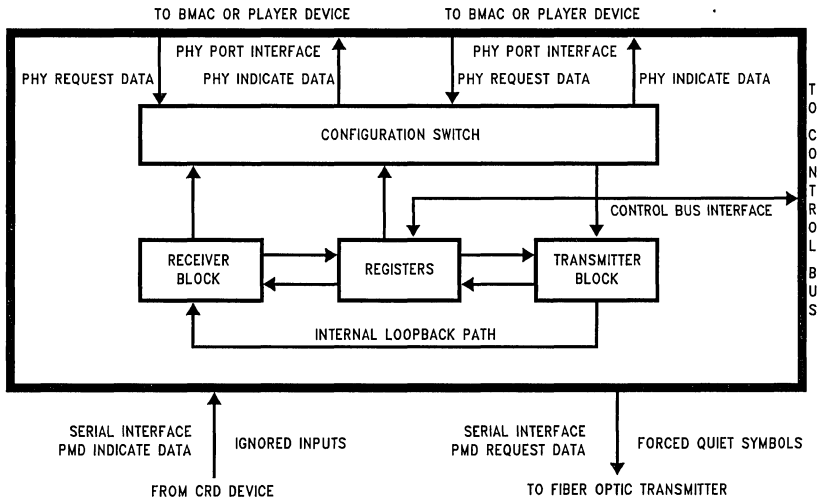


FIGURE 4-2. Internal Loopback

TL/F/10386-16

4.0 Modes of Operation (Continued)

4.3.3 External Loopback

The External Loopback mode can be used to test the functionality of the PLAYER device and to test the data paths between the PLAYER, CRD, and BMAC devices before transmitting and receiving data through the network.

When in the External Loopback mode, the PLAYER device performs the following functions:

- Directs the output data of the Transmitter Block to the external Loopback Data pins (LBD±), which are normally connected to the Clock Recovery Device (see *Figure 2. PLAYER Device Block Diagram*).

- Outputs Quiet symbols through the PMD Data Request pins (TXD±).

The level of the Quiet symbols transmitted through the TXC± pins is programmable through the Transmit Quiet Level bit of the Mode Register.

If both Internal Loopback and External Loopback modes are selected, Internal Loopback mode will have priority over External Loopback mode.

See *Figure 4-3* for a block diagram.

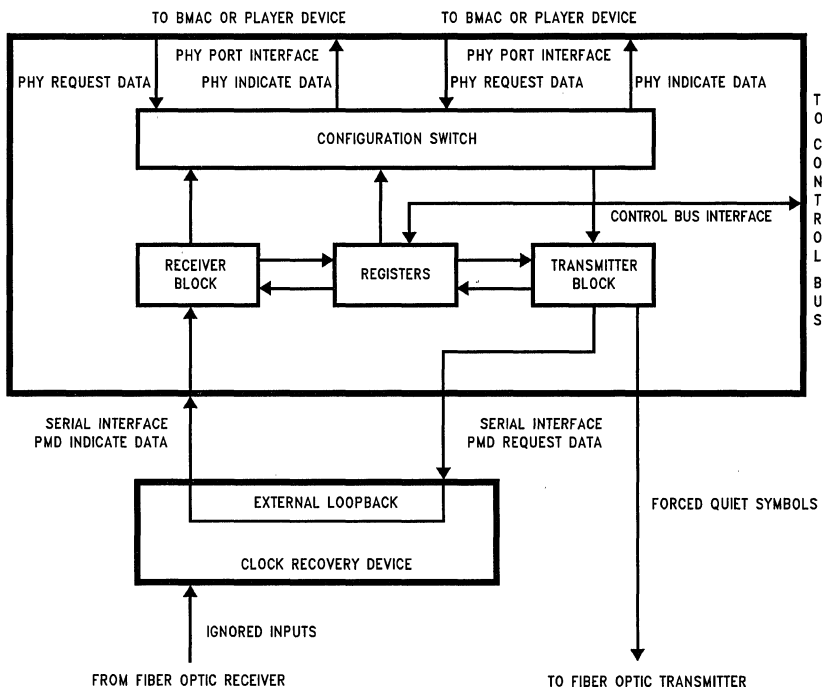


FIGURE 4-3. External Loopback

TL/F/10386-17

4.0 Modes of Operation (Continued)

4.4 CASCADE MODE

The PLAYER device can operate in the Cascade (parallel) mode—*Figure 4-4*—which is used in high bandwidth, point-to-point data transfer applications. This is a non-FDDI mode of operation.

CONCEPTS

In the Cascade mode, multiple PLAYER devices are connected together to provide data transfer at multiples of the FDDI data rate. Two cascaded PLAYER devices provide a data rate twice the FDDI data rate; three cascaded PLAYER devices provide a data rate three times the FDDI data rate, etc.

Multiple data streams are transmitted in parallel over each pair of cascaded PLAYER devices. All data streams start simultaneously and begin with the JK symbol pair on each PLAYER device.

Data is synchronized at the receiver of each PLAYER device by the JK symbol pair. Upon receiving a JK symbol pair, a PLAYER device asserts the Cascade Ready signal to indicate the beginning of data reception.

The Cascade Ready signals of all PLAYER devices are open drain ANDed together to create the Cascade Start signal. The Cascade Start signal is used as the input to indicate that all PLAYER devices have received the JK symbol pair. Data is now being received at every PLAYER device and can be transferred from the cascaded PLAYER devices to the host system.

See *Figure 4-5* for more information.

OPERATING RULES

When the PLAYER device is operating in Cascade mode, the following rules apply:

1. Data integrity can be guaranteed if the worst case fiber optic transmission skew between parallel fiber cables is less than 40 ns. This amounts to about 785 meters of fiber, assuming a 1% worst case variance.
2. Even though this is a non-FDDI application, the general rules for FDDI frames must be obeyed.

- Data frames must be a minimum of three bytes long (including the JK symbol pair). Smaller frames will cause Elasticity Buffer errors.
 - Data frames must have a maximum size of 4500 bytes, with a JK starting delimiter and a (T or R or S)x or x(T or R or S) ending delimiter byte.
3. Due to the different clock rates, the JK symbol pair may arrive at different times at each PLAYER device. The total skew between the fastest and slowest cascaded PLAYER devices receiving the JK starting delimiter must not exceed 80 ns.
 4. The first PLAYER device to receive a JK symbol pair will present it to the host system and assert the Cascade Ready signal. The PLAYER device will present one more JK as it waits for the other PLAYER devices to recognize their JK. The maximum number of consecutive JKs that can be presented to the host is 2.
 5. The Cascade Start signal is set to 1 when all the cascaded PLAYER devices release their Cascade Ready signals.
 6. Bit 4 (CSE) of the Receive Condition Register B (RCRB) is set to 1 if the Cascade Start signal (CS) is *not* set before the second falling edge of clock signal LBC from when Cascade Ready (CR) was released. CS has to be set within approximately 80 ns of CR release. This condition signifies that not all cascaded PLAYERS have received their respective JK symbol pair within the allowed skew range.
 7. If the JK symbols are corrupted in the point-to-point links, some PLAYER devices may not report a Cascaded Synchronization Error.
 8. To guarantee integrity of the interframe information, the user must put at least 8 Idle symbol pairs between frames. The PLAYER device will function properly with only 4 Idle symbol pairs, however the interframe symbols may be corrupted with random non-JK symbols.

The BMAC device could be used to provide required framing and optical FCS support.

4.0 Modes of Operation (Continued)

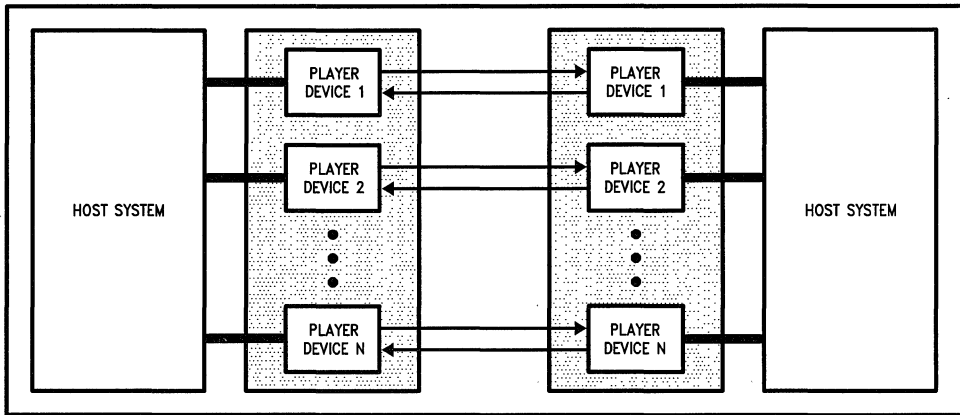


FIGURE 4-4. Parallel Transmission

TL/F/10386-18

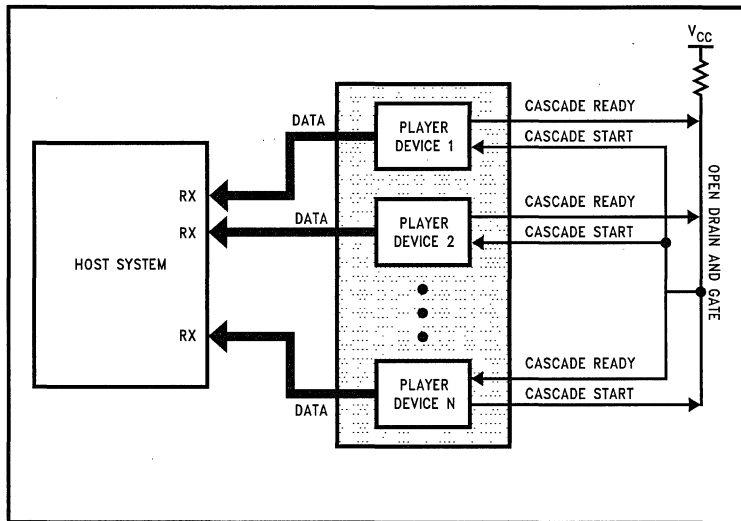


FIGURE 4-5. Cascade Mode of Operation

TL/F/10386-19

Note: N is recommended to be less than 3 for this mode. See Application Note 679 for larger values of N.

5.0 Registers

The PLAYER device is initialized, configured, and monitored via 32 8-bit registers. These registers are accessible through the Control Bus Interface.

Table 5-1 is a Register Summary List. Table 5-2 shows the contents of each register.

TABLE 5-1. Register Summary

Register Address	Register Symbol	Register Name	Access Rules	
			Read	Write
00h	MR	Mode Register	Always	Always
01h	CR	Configuration Register	Always	Always
02h	ICR	Interrupt Condition Register	Always	Conditional
03h	ICMR	Interrupt Condition Mask Register	Always	Always
04h	CTSR	Current Transmit State Register	Always	Conditional
05h	IJTR	Injection Threshold Register	Always	Always
06h	ISRA	Injection Symbol Register A	Always	Always
07h	ISRB	Injection Symbol Register B	Always	Always
08h	CRSR	Current Receive State Register	Always	Write Reject
09h	RCRA	Receive Condition Register A	Always	Conditional
0Ah	RCRB	Receive Condition Register B	Always	Conditional
0Bh	RCMRA	Receive Condition Mask Register A	Always	Always
0Ch	RCMRB	Receive Condition Mask Register B	Always	Always
0Dh	NTR	Noise Threshold Register	Always	Always
0Eh	NPTR	Noise Prescale Threshold Register	Always	Always
0Fh	CNCR	Current Noise Count Register	Always	Write Reject
10h	CNPCR	Current Noise Prescale Count Register	Always	Write Reject
11h	STR	State Threshold Register	Always	Always
12h	SPTR	State Prescale Threshold Register	Always	Always
13h	CSCR	Current State Count Register	Always	Write Reject
14h	CSPCR	Current State Prescale Count Register	Always	Write Reject
15h	LETR	Link Error Threshold Register	Always	Always
16h	CLECR	Current Link Error Count Register	Always	Write Reject
17h	UDR	User Definable Register	Always	Always
18h	IDR	Device ID Register	Always	Write Reject
19h	CIJCR	Current Injection Count Register	Always	Write Reject
1Ah	ICCR	Interrupt Condition Comparison Register	Always	Always
1Bh	CTSCR	Current Transmit State Comparison Register	Always	Always
1Ch	RCCRA	Receive Condition Comparison Register A	Always	Always
1Dh	RCCRB	Receive Condition Comparison Register B	Always	Always
1Eh	RR0	Reserved Register 0	Always	Write Reject
1Fh	RR1	Reserved Register 1	Always	Write Reject

5.0 Registers (Continued)

TABLE 5-2. Register Content Summary

Register Address	Register Symbol	Bit Symbols							
		D7	D6	D5	D4	D3	D2	D1	D0
00h	MR	RNRZ	TNRZ	TE	TQL	CM	EXLB	ILB	RUN
01h	CR	BIE	AIE	TRS1	TRS0	BIS1	BIS0	AIS1	AIS0
02h	ICR	UDI	RCB	RCA	LEMT	CWI	CCR	CPE	DPE
03h	ICMR	UDIM	RCBM	RCAM	LEMTM	CWIM	CCRM	CPEM	DPEM
04h	CTSR	RES	PRDPE	SE	IC1	IC0	TM2	TM1	TM0
05h	IJTR	IJT7	IJT6	IJ5	IJT4	IJT3	IJT2	IJT1	IJT0
06h	ISRA	RES	RES	RES	IJS4	IJS3	IJS2	IJS1	IJS0
07h	ISRB	RES	RES	RES	IJS9	IJS8	IJS7	IJS6	IJS5
08h	CRSR	RES	RES	RES	RES	LSU	LS2	LS1	LS0
09h	RCRA	LSUPI	LSC	NT	NLS	MLS	HLS	QLS	NSD
0Ah	RCRB	RES	SILS	EBOU	CSE	LSUPV	ALS	ST	ILS
0Bh	RCMRA	LSUPIM	LSCM	NTM	NLSM	NLSM	HLSM	QLSM	NSDM
0Ch	RCMRB	RES	SILSM	EBOUM	CSEM	LSUPVM	ALSM	STM	ILSM
0Dh	NTR	RES	NT6	NT5	NT4	NT3	NT2	NT1	NT0
0Eh	NPTR	NPT7	NPT6	NPT5	NPT4	NPT3	NPT2	NPT1	NPT0
0Fh	CNCR	NCLSCD	CNC6	CNC5	CNC4	CNC3	CNC2	CNC1	CNC0
10h	CNPCR	CNPC7	CNPC6	CNPC5	CNPC4	CNPC3	CNPC2	CNPC1	CNPC0
11h	STR	RES	ST6	ST5	ST4	ST3	ST2	ST1	ST0
12h	SPTR	SPT7	SPT6	SPT5	SPT4	SPT3	SPT2	SPT1	SPT0
13h	CSCR	SCLSCD	CSC6	CSC5	CSC4	CSC3	CSC2	CSC1	CSC0
14h	CSPCR	CSPC7	CSPC6	CSPC5	CSPC4	CSPC3	CSPC2	CSPC1	CSPC0
15h	LETR	LET7	LET6	LET5	LET4	LET3	LET2	LET1	LET0
16h	CLECR	LEC7	LEC6	LEC5	LEC4	LEC3	LEC2	LEC1	LEC0
17h	UDR	RES	RES	RES	RES	EB1	EB0	SB1	SB0
18h	IDR	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0
19h	CIJCR	IJC7	IJC6	IJC5	IJC4	IJC3	IJC2	IJC1	IJC0
1Ah	ICCR	UDIC	RCBC	RCAC	LEMTC	CWIC	CCRC	CPEC	DPEC
1Bh	CTSCR	RESC	PRDPEC	SEC	IC1C	IC0C	TM2C	TM1C	TM0C
1Ch	RCCRA	LSUPIC	LSCC	NTC	NLSC	MLSC	HLSC	QLSC	NSDC
1Dh	RCCRB	RESC	SILSC	EBOUC	CSEC	LSUPVC	ALSC	STC	ILSC
1Eh	RR1	RES	RES	RES	RES	RES	RES	RES	RES
1Fh	RR2	RES	RES	RES	RES	RES	RES	RES	RES

5.0 Registers (Continued)

MODE REGISTER (MR)

The Mode Register is used to initialize and configure the PLAYER device.

In order to minimize interruptions on the network, it is recommended that the PLAYER device first be put in STOP mode (i.e. set the RUN bit to zero) before programming the Mode Register, the Configuration Register, or the Current Transmit State Register.

ACCESS RULES

ADDRESS	READ	WRITE
00h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
RNRZ	TNRZ	TE	TQL	CM	EXLB	ILB	RUN

Bit	Symbol	Description
D0	RUN	<p>RUN /STOP</p> <p>0: Enables the STOP mode. Refer to Section 4.2, STOP Mode of Operation, for more information.</p> <p>1: Normal Operation (i.e. RUN mode).</p> <p>Note: The RUN bit is automatically set to 0 when the $\overline{\text{RST}}$ pin is asserted (i.e. set to ground).</p>
D1	ILB	<p>INTERNAL LOOPBACK:</p> <p>0: Disables Internal Loopback mode (i.e. normal operation).</p> <p>1: Enables Internal Loopback mode.</p> <p>Refer to Section 4.3, Loopback Mode of Operation, for more information.</p>
D2	EXLB	<p>EXTERNAL LOOPBACK</p> <p>0: Disables External Loopback mode (i.e. normal operation).</p> <p>1: Enables External Loopback mode.</p> <p>Refer to Section 4.3, Loopback Mode of Operation, for more information.</p>
D3	CM	<p>CASCADE MODE:</p> <p>0: Disables synchronization of cascaded PLAYER devices.</p> <p>1: Enables the synchronization of cascaded PLAYER devices.</p> <p>Refer to Section 4.4, Cascade Mode of Operation, for more information.</p>
D4	TQL	<p>TRANSMIT QUIET LEVEL: This bit is used to program the transmission level of the Quiet symbols.</p> <p>0: Low level Quiet symbols are transmitted through the PMD Data Request pins (i.e. TXD+ = low, TXD- = high).</p> <p>1: High level Quiet symbols are transmitted through the PMD Data Request pins (i.e. TXD+ = high, TXD- = low).</p>
D5	TE	<p>TRANSMIT ENABLE: The TE bit controls the action of FOTX Enable (TXE) pin independent of the current transmit mode. When TE is 0, the TXE output disables the optical transmitter; when TE is 1, the optical transmitter is disabled during the Off Transmit Mode (OTM) and enabled otherwise. The On and Off level of the TXE is dependent on the FOTX Enable Level (TEL) pin to the PLAYER device. The following rules summarizes the output of TXE:</p> <ol style="list-style-type: none"> (1) If TE = 0, then TXE = Off (2) If TE = 1 and OTM, then TXE = Off (3) If TE = 1 and not OTM, then TXE = On.
D6	TNRZ	<p>TRANSMIT NRZ DATA:</p> <p>0: Transmits data in Non-Return-To-Zero-Invert-On-Ones format.</p> <p>1: Transmits data in Non-Return-To-Zero format.</p>
D7	RNRZ	<p>RECEIVE NRZ DATA:</p> <p>0: Receives data in Non-Return-To-Zero-Invert-On-Ones format.</p> <p>1: Receives data in Non-Return-To-Zero format.</p>

5.0 Registers (Continued)

CONFIGURATION REGISTER (CR)

The Configuration Register controls the Configuration Switch Block and enables/disables both the A and B Indicate output ports.

Note that the B_Indicate output port is offered only on the DP83255 (for Dual Attach Stations), and not in the DP83251 (for Single Attach Stations).

For further information, refer to Section 3.3, Configuration Switch.

ACCESS RULES

ADDRESS	READ	WRITE
01h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
BIE	AIE	TRS1	TRS0	BIS1	BIS0	AIS1	AIS0

Bit	Symbol	Description															
D0, D1	AIS0, AIS1	<p>A__INDICATE SELECTOR <0, 1>: The A__Indicate Selector <0, 1> bits select one of the four Configuration Switch data buses for the A__Indicate output port (AIP, AIC, AID <7:0>).</p> <table border="0"> <thead> <tr> <th>AIS1</th> <th>AIS0</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>PHY Invalid Bus</td> </tr> <tr> <td>0</td> <td>1</td> <td>Receiver Bus</td> </tr> <tr> <td>1</td> <td>0</td> <td>A__Request Bus</td> </tr> <tr> <td>1</td> <td>1</td> <td>B__Request Bus</td> </tr> </tbody> </table>	AIS1	AIS0		0	0	PHY Invalid Bus	0	1	Receiver Bus	1	0	A__Request Bus	1	1	B__Request Bus
AIS1	AIS0																
0	0	PHY Invalid Bus															
0	1	Receiver Bus															
1	0	A__Request Bus															
1	1	B__Request Bus															
D2, D3	BIS0, BIS1	<p>B__INDICATE SELECTOR <0, 1>: The B__Indicate Selector <0, 1> bits select one of the four Configuration Switch data buses for the B__Indicate output port (BIP, BIC, BID <7:0>).</p> <table border="0"> <thead> <tr> <th>BIS1</th> <th>BIS0</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>PHY Invalid Bus</td> </tr> <tr> <td>0</td> <td>1</td> <td>Receiver Bus</td> </tr> <tr> <td>1</td> <td>0</td> <td>A__Request Bus</td> </tr> <tr> <td>1</td> <td>1</td> <td>B__Request Bus</td> </tr> </tbody> </table> <p>Note: Even though this bit can be set and/or cleared in the DP83251 (for Single Attach Stations), it will not affect any I/Os since the DP83251 does not offer a B__Indicate port.</p>	BIS1	BIS0		0	0	PHY Invalid Bus	0	1	Receiver Bus	1	0	A__Request Bus	1	1	B__Request Bus
BIS1	BIS0																
0	0	PHY Invalid Bus															
0	1	Receiver Bus															
1	0	A__Request Bus															
1	1	B__Request Bus															
D4, D5	TRS0, TRS1	<p>TRANSMIT REQUEST SELECTOR <0, 1>: The Transmit Request Selector <0, 1> bits select one of the four Configuration Switch data buses for the input to the Transmitter Block.</p> <table border="0"> <thead> <tr> <th>TRS1</th> <th>TRS0</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>PHY Invalid Bus</td> </tr> <tr> <td>0</td> <td>1</td> <td>Receiver Bus</td> </tr> <tr> <td>1</td> <td>0</td> <td>A__Request Bus</td> </tr> <tr> <td>1</td> <td>1</td> <td>B__Request Bus</td> </tr> </tbody> </table> <p>Note: If the PLAYER device is in Active Transmit Mode (i.e. the Transmit Mode bits (TM<2:0>) of the Current Transmit State Register (CTSR) are set to 000) and the PHY Invalid Bus is selected, then the PLAYER device will transmit a maximum of four Halt symbol pairs and then continuous Idle symbols due to the Repeat Filter.</p>	TRS1	TRS0		0	0	PHY Invalid Bus	0	1	Receiver Bus	1	0	A__Request Bus	1	1	B__Request Bus
TRS1	TRS0																
0	0	PHY Invalid Bus															
0	1	Receiver Bus															
1	0	A__Request Bus															
1	1	B__Request Bus															
D6	AIE	<p>A__INDICATE ENABLE:</p> <p>0: Disables the A__Indicate output port. The A__Indicate port pins will be at TRI-STATE when the port is disabled.</p> <p>1: Enables the A__Indicate output port (AIP, AIC, AID<7:0>).</p>															
D7	BIE	<p>B__INDICATE ENABLE:</p> <p>0: Disable the B__Indicate output port. The B__Indicate port pins will be at TRI-STATE when the port is disabled.</p> <p>1: Enables the B__Indicate output port (BIP, BIC, BID<7:0>).</p> <p>Note: Even though this bit can be set and/or cleared in the DP83251 (for Single Attach Stations), it will not affect any I/Os since the DP83251 does not offer a B__Indicate port.</p>															

5.0 Registers (Continued)

INTERRUPT CONDITION REGISTER (ICR)

The Interrupt Condition Register records the occurrence of an internal error event, the detection of Line State, an unsuccessful write by the Control Bus Interface, the expiration of an internal counter, or the assertion of one or more of the User Definable Sense pins.

The Interrupt Condition Register will assert the Interrupt pin (\overline{INT}) when one or more bits within the register are set to 1 and the corresponding mask bits in the Interrupt Condition Mask Register (ICMR) are also set to 1.

ACCESS RULES

ADDRESS	READ	WRITE
02h	Always	Conditional

D7	D6	D5	D4	D3	D2	D1	D0
UDI	RCB	RCA	LEMT	CWI	CCR	CPE	DPE

Bit	Symbol	Description
D0	DPE	<p>PHY_REQUEST_DATA PARITY ERROR: This bit will be set to 1 when:</p> <p>(1) The PHY Request Data Parity Enable bit (PRDPE) of the Current Transmit State Register (CTSR) is set to 1 and</p> <p>(2) The Transmitter Block detects a parity error in the incoming PHY Request Data.</p> <p>The source of the data can be from the PHY Invalid Bus, the Receiver Bus, the A__Bus, or the B__Bus of the Configuration Switch.</p>
D1	CPE	<p>CONTROL BUS DATA PARITY ERROR: This bit will be set to 1 when:</p> <p>(1) The Control Bus Parity Enable pin is asserted (CBPE = V_{CC}) and</p> <p>(2) The Control Bus Interface detects a parity error in the incoming Control Bus Data (CBD<7:0>) during a write cycle.</p>
D2	CCR	<p>CONTROL BUS WRITE COMMAND REJECT: This bit will be set to 1 when an attempt to write into one of the following read-only registers is made:</p> <ul style="list-style-type: none"> Current Receive State Register (Register 08, CRSR) Current Noise Count Register (Register 0F, CNCR) Current Noise Prescale Count Register (Register 10, CNPCR) Current State Count Register (Register 13, CSCR) Current State Prescale Count Register (Register 14, CSPCR) Current Link Error Count Register (Register 16, CLECR) Device ID Register (Register 18, IDR) Current Injection Count Register (Register 19, CIJCR) Reserved Register 0 (Register 1E, RR0) Reserved Register 1 (Register 1F, RR1)
D3	CWI	<p>CONDITIONAL WRITE INHIBIT: Set to 1 when bits within mentioned registers do not match bits in compare register. This bit ensures that new (i.e. unread) data is not inadvertently cleared while old data is being cleared through the Control Bus Interface.</p> <p>This bit is set to 1 to prevent the setting or clearing of any bit within the following registers:</p> <ul style="list-style-type: none"> Interrupt Condition Register (Register 02, ICR) Current Transmit State Register (Register 04, CTSR) Receive Condition Register A (Register 09, RCRA) Receive Condition Register B (Register 0A, RCRB) <p>when they differ from the value of the corresponding bit in the following registers respectively:</p> <ul style="list-style-type: none"> Interrupt Condition Compare Register (Register 1A, ICCR) Current Transmit State Compare Register (Register 1B, CTSCR) Receive Condition Compare Register A (Register 1C, RCCRA) Receive Condition Compare Register B (Register 1D, RCCRB) <p>This bit must be cleared by software. Note that this differs from the BMAC device bit of the same name.</p>

5.0 Registers (Continued)

INTERRUPT CONDITION REGISTER (ICR) (Continued)

Bit	Symbol	Description
D4	LEMT	<p>LINK ERROR MONITOR THRESHOLD: This bit is set to 1 when the internal 8-bit Link Error Monitor Counter reaches zero. It will remain set until cleared by software.</p> <p>During the reset process (i.e. $\overline{RST} = GND$), the Link Error Monitor Threshold bit is set to 1 because the Link Error Monitor Counter is initialized to zero.</p>
D5	RCA	<p>RECEIVE CONDITION A: This bit is set to 1 when:</p> <ol style="list-style-type: none"> (1) One or more bits in the Receive Condition Register A (RCRA) is set to 1 and (2) The corresponding mask bits in the Receive Condition Mask Register A (RCMRA) are also set to 1. <p>In order to clear (i.e. set to 0) the Receive Condition A bit, the bits within the Receive Condition Register A that are set to 1 must first be either cleared or masked.</p>
D6	RCB	<p>RECEIVE CONDITION B: This bit is set to 1 when:</p> <ol style="list-style-type: none"> (1) One or more bits in the Receive Condition Register B (RCRB) is set to 1 and (2) The corresponding mask bits in the Receive Condition Mask Register B (RCMRB) are also set to 1. <p>In order to clear (i.e. set to 0) the Receive Condition B bit, the bits within the Receive Condition Register B that are set to 1 must first be either cleared or masked.</p>
D7	UDI	<p>USER DEFINABLE INTERRUPT: This bit is set to 1 when one or both of the Sense Bits (SB0 or SB1) in the User Definable Register (UDR) is set to 1.</p> <p>In order to clear (i.e. set to 0) the User Definable Interrupt Bit, both Sense Bits must be set to 0.</p>

5.0 Registers (Continued)

INTERRUPT CONDITION MASK REGISTER (ICMR)

The Interrupt Condition Mask Register allows the user to dynamically select which events will generate an interrupt.

The Interrupt pin will be asserted (i.e. $\overline{INT} = \text{GND}$) when one or more bits within the Interrupt Condition Register (ICR) are set to 1 and the corresponding mask bits in this register are also set to 1.

This register is cleared (i.e. set to 0) and all interrupts are initially masked during the reset process.

ACCESS RULES

ADDRESS	READ	WRITE
03h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
UDIM	RCBM	RCAM	LEMTM	CWIM	CCRM	CPEM	DPEM

Bit	Symbol	Description
D0	DPEM	PHY_REQUEST_DATA PARITY ERROR MASK: The mask bit for the PHY_Request Data Parity Error bit (DPE) of Interrupt Condition Register (ICR).
D1	CPEM	CONTROL BUS DATA PARITY ERROR MASK: The mask bit for the Control Bus Data Parity Error bit (CPE) of the Interrupt Condition Register (ICR).
D2	CCRM	CONTROL BUS WRITE COMMAND REJECT MASK: The mask bit for the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR).
D3	CWIM	CONDITIONAL WRITE INHIBIT MASK: The mask bit for the Conditional Write Inhibit bit (CWI) of the Interrupt Condition Register (ICR).
D4	LEMTM	LINK ERROR MONITOR THRESHOLD MASK: The mask bit for the Link Error Monitor Threshold bit (LEMT) of the Interrupt Condition Register (ICR).
D5	RCAM	RECEIVE CONDITION A MASK: The mask bit for the Receive Condition A bit (RCA) of the Interrupt Condition Register (ICR).
D6	RCBM	RECEIVE CONDITION B MASK: The mask bit for the Receive Condition B bit (RCB) of the Interrupt Condition Register (ICR).
D7	UDIM	USER DEFINABLE INTERRUPT MASK: The mask bit for the User Definable Interrupt bit (UDI) of the Interrupt Condition Register (ICR).

5.0 Registers (Continued)

CURRENT TRANSMIT STATE REGISTER (CTSR)

The Current Transmit State Register can program the Transmitter Block to internally generate and transmit Idle, Master, Halt, Quiet, or user programmable symbol pairs, in addition to the normal transmission of incoming PHY Request data. The Smoother and PHY Request Data Parity may also be enabled and disabled through this register.

The Transmit Modes overwrite the Repeat Filter and Smoother outputs, while the Injection Symbols overwrite the Transmit Modes.

During the reset process (i.e. $\overline{RST} = GND$) the Transmit Mode is set to Off ($TM<2:0> = 010$), the Smoother is enabled (i.e. SE is set to 1), and the Reserved bit (b7) is set to 1. All other bits of this register are cleared (i.e. set to 0) during the reset process.

ACCESS RULES

ADDRESS	READ	WRITE
04h	Always	Conditional

D7	D6	D5	D4	D3	D2	D1	D0
RES	PRDPE	SE	IC1	IC0	TM2	TM1	TM0

Bit	Symbol	Description																																				
D0, D1, D2	TM0, TM1, TM2	<p>Transmit Mode <0, 1, 2>: These bits select one of the 6 transmission modes for the PMD Request Data output port (TXD±).</p> <table border="1"> <thead> <tr> <th>TM2</th> <th>TM1</th> <th>TM0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Active Transmit Mode (ATM): Normal transmission of incoming PHY Request data.</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Idle Transmit Mode (ITM): Transmission of Idle symbol pairs (11111 11111).</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Off Transmit Mode (OTM): Transmission of Quiet symbol pairs (00000 00000) and deassertion of the FOTX Enable pin (TXE).</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Reserved: Reserved for future use. Users are discouraged from using this transmit mode. If selected, however, the transmitter will generate Quiet symbol pairs (00000 00000).</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Master Transmit Mode (MTM): Transmission of Halt and Quiet symbol pairs (00100 00000).</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Halt Transmit Mode (HTM): Transmission of Halt symbol pairs (00100 00100).</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Quiet Transmit Mode (QTM): Transmission of Quiet symbol pairs (00000 00000).</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Reserved: Reserved for future use. Users are discouraged from using this transmit mode. If selected, however, the transmitter will generate Quiet symbol pairs (00000 00000).</td> </tr> </tbody> </table>	TM2	TM1	TM0	Description	0	0	0	Active Transmit Mode (ATM) : Normal transmission of incoming PHY Request data.	0	0	1	Idle Transmit Mode (ITM) : Transmission of Idle symbol pairs (11111 11111).	0	1	0	Off Transmit Mode (OTM) : Transmission of Quiet symbol pairs (00000 00000) and deassertion of the FOTX Enable pin (TXE).	0	1	1	Reserved : Reserved for future use. Users are discouraged from using this transmit mode. If selected, however, the transmitter will generate Quiet symbol pairs (00000 00000).	1	0	0	Master Transmit Mode (MTM) : Transmission of Halt and Quiet symbol pairs (00100 00000).	1	0	1	Halt Transmit Mode (HTM) : Transmission of Halt symbol pairs (00100 00100).	1	1	0	Quiet Transmit Mode (QTM) : Transmission of Quiet symbol pairs (00000 00000).	1	1	1	Reserved : Reserved for future use. Users are discouraged from using this transmit mode. If selected, however, the transmitter will generate Quiet symbol pairs (00000 00000).
TM2	TM1	TM0	Description																																			
0	0	0	Active Transmit Mode (ATM) : Normal transmission of incoming PHY Request data.																																			
0	0	1	Idle Transmit Mode (ITM) : Transmission of Idle symbol pairs (11111 11111).																																			
0	1	0	Off Transmit Mode (OTM) : Transmission of Quiet symbol pairs (00000 00000) and deassertion of the FOTX Enable pin (TXE).																																			
0	1	1	Reserved : Reserved for future use. Users are discouraged from using this transmit mode. If selected, however, the transmitter will generate Quiet symbol pairs (00000 00000).																																			
1	0	0	Master Transmit Mode (MTM) : Transmission of Halt and Quiet symbol pairs (00100 00000).																																			
1	0	1	Halt Transmit Mode (HTM) : Transmission of Halt symbol pairs (00100 00100).																																			
1	1	0	Quiet Transmit Mode (QTM) : Transmission of Quiet symbol pairs (00000 00000).																																			
1	1	1	Reserved : Reserved for future use. Users are discouraged from using this transmit mode. If selected, however, the transmitter will generate Quiet symbol pairs (00000 00000).																																			

5.0 Registers (Continued)

CURRENT TRANSMIT STATE REGISTER (CTSR) (Continued)

Bit	Symbol	Description															
D3, D4	IC0, IC1	<p>Injection Control <0, 1>: These bits select one of the 4 injection modes. The injection modes overwrite data from the Smoother, Repeat Filter, Encoder, and Transmit Modes.</p> <p>IC0 is the only bit of the register that is automatically cleared by the PLAYER device after the One Shot Injection is executed. The automatic clear of IC0 during the One Shot mode can be interpreted as an acknowledgment that the One Shot has been completed.</p> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; width: 10%;">IC1</td> <td style="text-align: center; width: 10%;">IC0</td> <td></td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>No Injection: The normal transmission of incoming PHY Request data (i.e. symbols are not injected).</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>One Shot: In one shot mode, Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB) are injected n symbol pairs after a JK, where n is the programmed value of the Injection Count Register (IJCR). If IJCR is set to 0, the JK symbol pair is replaced by ISRA and ISRB. Once the One Shot is executed, the PLAYER device automatically sets IC0 to 0, thereby returning to normal transmission of data.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Periodic: In Periodic mode, Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB) are injected every (n + 1)th symbol pair, where n is the programmed value of the Injection Count Register (IJCR). If IJCR is set to 0, all data symbols are replaced with ISRA and ISRB.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Continuous: In Continuous mode, all data symbols are replaced with Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB).</td> </tr> </table>	IC1	IC0		0	0	No Injection: The normal transmission of incoming PHY Request data (i.e. symbols are not injected).	0	1	One Shot: In one shot mode, Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB) are injected n symbol pairs after a JK, where n is the programmed value of the Injection Count Register (IJCR). If IJCR is set to 0, the JK symbol pair is replaced by ISRA and ISRB. Once the One Shot is executed, the PLAYER device automatically sets IC0 to 0, thereby returning to normal transmission of data.	1	0	Periodic: In Periodic mode, Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB) are injected every (n + 1)th symbol pair, where n is the programmed value of the Injection Count Register (IJCR). If IJCR is set to 0, all data symbols are replaced with ISRA and ISRB.	1	1	Continuous: In Continuous mode, all data symbols are replaced with Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB).
IC1	IC0																
0	0	No Injection: The normal transmission of incoming PHY Request data (i.e. symbols are not injected).															
0	1	One Shot: In one shot mode, Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB) are injected n symbol pairs after a JK, where n is the programmed value of the Injection Count Register (IJCR). If IJCR is set to 0, the JK symbol pair is replaced by ISRA and ISRB. Once the One Shot is executed, the PLAYER device automatically sets IC0 to 0, thereby returning to normal transmission of data.															
1	0	Periodic: In Periodic mode, Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB) are injected every (n + 1)th symbol pair, where n is the programmed value of the Injection Count Register (IJCR). If IJCR is set to 0, all data symbols are replaced with ISRA and ISRB.															
1	1	Continuous: In Continuous mode, all data symbols are replaced with Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB).															
D5	SE	<p>SMOOTHER ENABLE:</p> <p>0: Disables the Smoother. 1: Enables the Smoother.</p> <p>When enabled, the Smoother can redistribute Idle symbol pairs which were added or deleted by the local or upstream receivers.</p> <p>Note: Once the counter has started, it will continue to count irrespective of the incoming symbols with the exception of a JK symbol pair. This bit should be enabled for interoperable operation.</p>															
D6	PRDPE	<p>PHY__REQUEST DATA PARITY ENABLE:</p> <p>0: Disables PHY__Request Data parity. 1: Enables PHY__Request Data parity.</p>															
D7	RES	<p>RESERVED: Reserved for future use.</p> <p>Note: Users are discouraged from using this bit. The reserved bit is set to 1 during the reset process. It may be set or cleared without any effects to the functionality of the PLAYER device.</p>															

5.0 Registers (Continued)

INJECTION THRESHOLD REGISTER (IJTR)

The Injection Threshold Register, in conjunction with the Injection Control bits (IC<1:0>) in the Current Transmit State Register (CTSR), set the frequency at which the Injection Symbol Register A (ISRA) and Injection Symbol Register B (ISRB) are inserted into the data stream. It contains the start value for the Injection Counter.

The Injection Threshold Register value is loaded into the Injection Counter when the counter reaches zero or during every Control Bus Interface write-cycle of this register.

The Injection Counter is an 8-bit down-counter which decrements every 80 ns. Its current value is read for CIJCR.

The counter is active only during One Shot or Periodic Injection Modes (i.e. Injection Control<1:0> bits (IC<1:0>) of the Current Transmit State Register (CTSR) are set to either 01 or 10). The Transmitter Block will replace a data symbol pair with ISRA and ISRB when the counter reaches 0 and the Injection Mode is either One Shot or Periodic.

If the Injection Threshold Register is set to 0 during the One Shot mode, the JK will be replaced with ISRA and ISRB. If the Injection Threshold Register is set to 0 during the Periodic mode, all data symbols are replaced with ISRA and ISRB.

The counter is initialized to 0 during the reset process (i.e. RST = GND).

For further information, see the Injection Control Logic subsection within Section 3.2.

ACCESS RULES

ADDRESS	READ	WRITE
05h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
IJT7	IJT6	IJT5	IJT4	IJT3	IJT2	IJT1	IJT0

Bit	Symbol	Description
D0	IJT0	INJECTION THRESHOLD BIT <0> : Least significant bit (LSB) of the start value for the Injection Counter.
D1-6	IJT1-6	INJECTION THRESHOLD BIT <1-6> : Intermediate bits of start value for the Injection Counter.
D7	IJT7	INJECTION THRESHOLD BIT <7> : Most significant bit (MSB) of the start value for the Injection Counter.

5.0 Registers (Continued)

INJECTION SYMBOL REGISTER A (ISRA)

The Injection Symbol Register A, along with Injection Symbol Register B, contains the programmable value (already in 5B code) that will replace the data symbol pairs.

The One Shot mode, ISRA and ISRB are injected n bytes after the next JK, where n is the programmed value of the Injection Threshold Register. In the Periodic mode, ISRA and ISRB are injected every nth symbol pair. In the Continuous mode, all data symbols are replaced with ISRA and ISRB.

ACCESS RULES

ADDRESS	READ	WRITE
06h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	IJS4	IJS3	IJS2	IJS1	IJS0

Bit	Symbol	Description
D0	IJS0	INJECTION THRESHOLD BIT <0> : Least significant bit (LSB) of Injection Symbol Register A.
D1-3	IJS1-3	INJECTION THRESHOLD BIT <1-3> : Intermediate bits of Injection Symbol Register A.
D4	IJS4	INJECTION THRESHOLD BIT <4> : Most significant bit (MSB) of Injection Symbol Register A.
D5	RES	RESERVED : Reserved for future use. Note: Users are discouraged from using this bit. The reserved bit is set to 1 during the reset process. It may be set or cleared without any effects to the functionality of the PLAYER device.
D6	RES	RESERVED : Reserved for future use. Note: Users are discouraged from using this bit. The reserved bit is set to 1 during the reset process. It may be set or cleared without any effects to the functionality of the PLAYER device.
D7	RES	RESERVED : Reserved for future use. Note: Users are discouraged from using this bit. The reserved bit is set to 1 during the reset process. It may be set or cleared without any effects to the functionality of the PLAYER device.

5.0 Registers (Continued)

INJECTION SYMBOL REGISTER B (ISRB)

The Injection Symbol Register B, along with Injection Symbol Register A, contains the programmable value (already in 5B code) that will replace the data symbol pairs.

The One Shot mode, ISRA and ISRB are injected n bytes after the next JK, where n is the programmed value of the Injection Threshold Register. In the Periodic mode, ISRA and ISRB are injected every nth symbol pair. In the Continuous mode, all data symbols are replaced with ISRA and ISRB.

ACCESS RULES

ADDRESS	READ	WRITE
07h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	IJS9	IJS8	IJS7	IJS6	IJS5

Bit	Symbol	Description
D0	IJS5	INJECTION THRESHOLD BIT <5> : Least significant bit (LSB) of Injection Symbol Register B.
D1–3	IJS6–8	INJECTION THRESHOLD BIT <6–8> : Intermediate bits of Injection Symbol Register B.
D4	IJS9	INJECTION THRESHOLD BIT <9> : Most significant bit (MSB) of Injection Symbol Register B.
D5	RES	RESERVED: Reserved for future use. Note: Users are discouraged from using this bit. The reserved bit is set to 1 during the reset process. It may be set or cleared without any effects to the functionality of the PLAYER device.
D6	RES	RESERVED: Reserved for future use. Note: Users are discouraged from using this bit. The reserved bit is set to 1 during the reset process. It may be set or cleared without any effects to the functionality of the PLAYER device.
D7	RES	RESERVED: Reserved for future use. Note: Users are discouraged from using this bit. The reserved bit is set to 1 during the reset process. It may be set or cleared without any effects to the functionality of the PLAYER device.

5.0 Registers (Continued)

CURRENT RECEIVE STATE REGISTER (CRSR)

The Current Receive State Register represents the current line state being detected by the Receiver Block. Once the Receiver Block recognizes a new Line State, the bits corresponding to the previous line state are cleared, and the bits corresponding to the new line state are set.

During the reset process ($\overline{RST} = \text{GND}$), the Receiver Block is forced to Line State Unknown (i.e. the Line State Unknown bit (LSU) is set to 1).

ACCESS RULES

ADDRESS	READ	WRITE
08h	Always	Write Reject

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	LSU	LS2	LS1	LS0

Bit	Symbol	Description																																				
D0, D1 D2	LS0, LS1, LS2	<p>LINE STATE <0, 1, 2>: These bits represent the current Line State being detected by the Receiver Block. Once the Receiver Block recognizes a new line state, the bits corresponding to the previous line state are cleared, and the bits corresponding to the new line state are set.</p> <table border="1"> <thead> <tr> <th>LS2</th> <th>LS1</th> <th>LS0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Active Line State (ALS): Received a JK symbol pair (11000 10001), and possibly followed by data symbols.</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Idle Line State (ILS): Received a minimum of two consecutive Idle symbol pairs (11111 11111).</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>No Signal Detect (NSD): The Signal Detect pin (TTLSD) has been deasserted, indicating that the Clock Recovery Device is not receiving data from the Fiber Optic Receiver.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Reserved: Reserved for future use.</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Master Line State (MLS): Received a minimum of 8 consecutive Halt-Quiet symbol pairs (00100 00000).</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Halt Line State (HLS): Received a minimum of 8 consecutive Halt symbol pairs (00100 00100).</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Quiet Line State (QLS): Received a minimum of 8 consecutive Quiet symbol pairs (00000 00000).</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Noise Line State (NLS): Detected a minimum of 16 noise events. Refer to the Receiver Block for further information on noise events.</td> </tr> </tbody> </table>	LS2	LS1	LS0	Description	0	0	0	Active Line State (ALS) : Received a JK symbol pair (11000 10001), and possibly followed by data symbols.	0	0	1	Idle Line State (ILS) : Received a minimum of two consecutive Idle symbol pairs (11111 11111).	0	1	0	No Signal Detect (NSD) : The Signal Detect pin (TTLSD) has been deasserted, indicating that the Clock Recovery Device is not receiving data from the Fiber Optic Receiver.	0	1	1	Reserved : Reserved for future use.	1	0	0	Master Line State (MLS) : Received a minimum of 8 consecutive Halt-Quiet symbol pairs (00100 00000).	1	0	1	Halt Line State (HLS) : Received a minimum of 8 consecutive Halt symbol pairs (00100 00100).	1	1	0	Quiet Line State (QLS) : Received a minimum of 8 consecutive Quiet symbol pairs (00000 00000).	1	1	1	Noise Line State (NLS) : Detected a minimum of 16 noise events. Refer to the Receiver Block for further information on noise events.
LS2	LS1	LS0	Description																																			
0	0	0	Active Line State (ALS) : Received a JK symbol pair (11000 10001), and possibly followed by data symbols.																																			
0	0	1	Idle Line State (ILS) : Received a minimum of two consecutive Idle symbol pairs (11111 11111).																																			
0	1	0	No Signal Detect (NSD) : The Signal Detect pin (TTLSD) has been deasserted, indicating that the Clock Recovery Device is not receiving data from the Fiber Optic Receiver.																																			
0	1	1	Reserved : Reserved for future use.																																			
1	0	0	Master Line State (MLS) : Received a minimum of 8 consecutive Halt-Quiet symbol pairs (00100 00000).																																			
1	0	1	Halt Line State (HLS) : Received a minimum of 8 consecutive Halt symbol pairs (00100 00100).																																			
1	1	0	Quiet Line State (QLS) : Received a minimum of 8 consecutive Quiet symbol pairs (00000 00000).																																			
1	1	1	Noise Line State (NLS) : Detected a minimum of 16 noise events. Refer to the Receiver Block for further information on noise events.																																			
D3	LSU	LINE STATE UNKNOWN : The Receiver Block has not detected the minimum conditions to enter a known line state. When the Line State Unknown bit is set, LS<2:0> represent the most recently known line state.																																				
D4	RES	<p>RESERVED: Reserved for future use. The reserved bit is set to 0.</p> <p>Note: Users are discouraged from using this bit. An attempt to write into this bit will cause the PLAYER device to ignore the Control Bus write cycle and set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1.</p>																																				

5.0 Registers (Continued)**CURRENT RECEIVE STATE REGISTER (CRSR)** (Continued)

Bit	Symbol	Description
D5	RES	RESERVED: Reserved for future use. The reserved bit is set to 0. Note: Users are discouraged from using this bit. An attempt to write into this bit will cause the PLAYER device to ignore the CBUS write cycle and set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1.
D6	RES	RESERVED: Reserved for future use. The reserved bit is set to 0. Note: Users are discouraged from using this bit. An attempt to write into this bit will cause the PLAYER device to ignore the CBUS write cycle and set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1.
D7	RES	RESERVED: Reserved for future use. The reserved bit is set to 0. Note: Users are discouraged from using this bit. An attempt to write into this bit will cause the PLAYER device to ignore the CBUS write cycle and set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1.

5.0 Registers (Continued)

RECEIVE CONDITION REGISTER A (RCRA)

The Receive Condition Register A maintains a historical record of the Line States recognized by the Receiver Block.

When a new Line State is entered, the bit corresponding to that line state is set to 1. The bits corresponding to the previous Line States are not cleared by the PLAYER device, thereby maintaining a record of the Line States detected.

The Receive Condition A bit (RCA) of the Interrupt Condition Register (ICR) will be set to 1 when one or more bits within the Receive Condition Register A is set to 1 and the corresponding mask bit(s) in Receive Condition Mask Register A (RCMRA) is also set to 1.

ACCESS RULES

ADDRESS	READ	WRITE
09h	Always	Conditional

D7	D6	D5	D4	D3	D2	D1	D0
LSUPI	LSC	NT	NLS	MLS	HLS	QLS	NSD

Bit	Symbol	Description
D0	NSD	NO SIGNAL DETECT: Indicates that the Signal Detect pin (TTLSD) has been deasserted and that the Clock Recovery Device is not receiving data from the Fiber Optic Receiver.
D1	QLS	QUIET LINE STATE: Received a minimum of eight consecutive Quiet symbol pairs (00000 00000).
D2	HLS	HALT LINE STATE: Received a minimum of eight consecutive Halt symbol pairs (00100 00100).
D3	MLS	MASTER LINE STATE: Received a minimum of eight consecutive Halt-Quiet symbol pairs (00100 00000).
D4	NLS	NOISE LINE STATE: Detected a minimum of sixteen noise events.
D5	NT	NOISE THRESHOLD: This bit is set to 1 when the internal Noise Counter reaches 0. It will remain set until a value equal to or greater than one is loaded into the Noise Threshold Register or Noise Prescale Threshold Register. During the reset process (i.e. $\overline{RST} = \text{GND}$), since the Noise Counter is initialized to 0, the Noise Threshold bit will be set to 1.
D6	LSC	LINE STATE CHANGE: A line state change has been detected.
D7	LSUPI	LINE STATE UNKNOWN & PHY INVALID: The Receiver Block has not detected the minimum conditions to enter a known line state. In addition, the most recently known line state was one of the following line states: No Signal Detect, Quiet Line State, Halt Line State, Master Line State, or Noise Line State.

5.0 Registers (Continued)

RECEIVE CONDITION REGISTER B (RCRB)

The Receive Condition Register B maintains a historical record of the Line States recognized by the Receiver Block.

When a new Line State is entered, the bit corresponding to that line state is set to 1. The bits corresponding to the previous Line States are not clear by the PLAYER device, thereby maintaining a record of the Line States detected.

The Receive Condition B bit (RCB) of the Interrupt Condition Register (ICR) will be set to 1 when one or more bits within the Receive Condition Register B is set to 1 and the corresponding mask bits in Receive Condition Mask Register B (RCMRB) is also set to 1.

ACCESS RULES

ADDRESS	READ	WRITE
0Ah	Always	Conditional

D7	D6	D5	D4	D3	D2	D1	D0
RES	SILS	EBOU	CSE	LSUPV	ALS	ST	ILS

Bit	Symbol	Description
D0	ILS	IDLE LINE STATE: Received a minimum of two consecutive Idle symbol pairs (11111 11111).
D1	ST	STATE THRESHOLD: This bit will be set to 1 by the PLAYER device when the internal State Counter reaches zero. It will remain set until a value equal to or greater than one is loaded into the State Threshold Register or State Prescale Threshold Register, and this register is cleared. During the reset process (i.e. $\overline{RST} = GND$), since the State Counter is initialized to 0, the State Threshold bit is set to 1.
D2	ALS	ACTIVE LINE STATE: Received a JK symbol pair (11000 10001), and possibly data symbols following.
D3	LSUPV	LINE STATE UNKNOWN & PHY VALID: Receiver Block has not detected the minimum conditions to enter a know line state when the most recently known line state was one of the following line states: Active Line State or Idle Line State
D4	CSE	CASCADE SYNCHRONIZATION ERROR: When a synchronization error occurs, the Cascade Synchronization Error bit is set to 1. A synchronization error occurs if the Cascade Start signal (CS) is not asserted within approximately 80 ns of Cascade Ready (CR) release.
D5	EBOU	ELASTICITY BUFFER UNDERFLOW/OVERFLOW: The Elasticity Buffer has either overflowed or underflowed. The Elasticity Buffer will automatically recover if the condition which caused the error is only transient.
D6	SILS	SUPER IDLE LINE STATE: Received a minimum of eight Idle symbol pairs (11111 11111).
D7	RES	RESERVED: Reserved for future use. The reserved bit is set to 0 during the reset process. Note: Users are discouraged from using this bit. It may be set or cleared without any effects to the functionality of the PLAYER device.

5.0 Registers (Continued)

RECEIVE CONDITION MASK REGISTER A (RCMRA)

The Receive Condition Mask Register A allows the user to dynamically select which events will generate an interrupt.

The Receive Condition A bit (RCA) of the Interrupt Condition Register (ICR) will be set to 1 when one or more bits within the Receive Condition Register A (RCRA) is set to 1 and the corresponding mask bit(s) in this register is also set to 1.

Since this register is cleared (i.e. set to 0) during the reset process, all interrupts are initially masked.

ACCESS RULES

ADDRESS	READ	WRITE
0Bh	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
LSUPIM	LSCM	NTM	NLSM	MLSM	HLSM	QLSM	NSDM

Bit	Symbol	Description
D0	NSDM	NO SIGNAL DETECT MASK: The mask bit for the No Signal Detect bit (NSD) of the Receive Condition Register A (RCRA).
D1	QLSM	QUIET LINE STATE MASK: The mask bit for the Quiet Line State bit (QLS) of the Receive Condition Register A (RCRA).
D2	HLSM	HALT LINE STATE MASK: The mask bit for the Halt Line State bit (HLS) of the Receive Condition Register A (RCRA).
D3	MLSM	MASTER LINE STATE MASK: The mask bit for the Master Line State bit (MLS) of the Receive Condition Register A (RCRA).
D4	NLSM	NOISE LINE STATE MASK: The mask bit for the Noise Line State bit (NLS) of the Receive Condition Register A (RCRA).
D5	NTM	NOISE THRESHOLD MASK: The mask bit for the Noise Threshold bit (NT) of the Receive Condition Register A (RCRA).
D6	LSCM	LINE STATE CHANGE MASK: The mask bit for the Line State Change bit (LSC) of the Receive Condition Register A (RCRA).
D7	LSUPIM	LINE STATE UNKNOWN & PHY INVALID MASK: The mask bit for the line State Unknown & PHY Invalid bit (LSUPI) of the Receive Condition Register A (RCRA).

5.0 Registers (Continued)

RECEIVE CONDITION MASK REGISTER B (RCMRB)

The Receive Condition Mask Register B allows the user to dynamically select which events will generate an interrupt.

The Receiver Condition B bit (RCB) of the Interrupt Condition Register (ICR) will be set to 1 when one or more bits within the Receive Condition B (RCRA) is set to 1 and the corresponding mask bits in this register is also set to 1.

Since this register is cleared (i.e. set to 0) during the reset process, all interrupts are initially masked.

ACCESS RULES

ADDRESS	READ	WRITE
0Ch	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
RES	SILSM	EBOUM	CSEM	LSUPVM	ALSM	STM	ILSM

Bit	Symbol	Description
D0	ILSM	IDLE LINE STATE MASK: The mask bit for the Idle Line State bit (ILS) of the Receive Condition Register B (RCRB).
D1	STM	STATE THRESHOLD MASK: The mask bit of the State Threshold bit (ST) of the Receive Condition Register B (RCRB).
D2	ALSM	ACTIVE LINE STATE MASK: The mask bit for the Active Line State bit (ALS) of the Receive Condition Register B (RCRB).
D3	LSUPVM	LINE STATE UNKNOWN & PHY VALID MASK: The mask bit for the Line State Unknown & PHY Valid bit (LSUPV) of the Receive Condition Register B (RCRB).
D4	CSEM	CASCADE SYNCHRONIZATION ERROR MASK: The mask bit for the Cascade Synchronization Error bit (CSE) of the Receive Condition Register B (RCRB).
D5	EBOUM	ELASTICITY BUFFER OVERFLOW/UNDERFLOW MASK: The mask bit for the Elasticity Buffer Overflow/Underflow bit (EBOU) of the Receive Condition Register B (RCRB).
D6	SILSM	SUPER IDLE LINE STATE MASK: The mask bit for the Super Idle Line State bit (SILS) of the Receive Condition Register B (RCRB).
D7	RESM	RESERVED MASK: The mask bit for the Reserved bit (RES) of the Receive Condition Register B (RCRB).

5.0 Registers (Continued)

NOISE THRESHOLD REGISTER (NTR)

The Noise Threshold Register contains the start value for the Noise Counter. This counter may be used in conjunction with the Noise Prescale Counter for counting the Noise events. Definition of Noise event is explained in detail in Section 8.2. The Noise Counter decrements once every 80 ns if the noise Prescale counter is zero and there is a noise event. As a result, the internal noise counter takes

$$((NPTR + 1) \times (NTR + 1)) \times 80 \text{ ns}$$

to reach zero in the event of continuous Noise event.

The threshold values for the Noise Counter and Noise Prescale Counter are simultaneously loaded into both counters if one of the following conditions is true:

(1) Both the Noise Counter and Noise Prescale Counter reach zero and the current Line State is either Noise Line State, Active Line State, or Line State Unknown.

or

(2) The current Line State is either Halt Line State, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect or

(3) The Noise Threshold Register or Noise Prescale Threshold Register goes through a Control Bus Interface write cycle.

In addition, the value of the Noise Prescale Threshold register is loaded into the Noise Prescale Counter if the Noise Prescale Counter reaches zero.

The Noise Counter and Noise Prescale Counter will continue to count, without resetting or reloading the threshold values, if a Line State change occurs and the new line state is either Noise Line State, Active Line State or Line State Unknown.

When both the Noise Threshold Counter and Noise Counter both reach zero, the Noise Threshold bit of the Receive Condition Register A will be set.

ACCESS RULES

ADDRESS	READ	WRITE
0Dh	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
NT7	NT6	NT5	NT4	NT3	NT2	NT1	NT0

Bit	Symbol	Description
D0	NT0	NOISE THRESHOLD BIT <0> : Least significant bit (LSB) of the start value for the Noise Counter.
D1–5	NT1–5	NOISE THRESHOLD BIT <1–5> : Intermediate bits of start value for the Noise Counter.
D6	NT6	NOISE THRESHOLD BIT <6> : Most significant bit (MSB) of the start value for the Noise Counter.
D7	RES	RESERVED : Reserved for future use. Note: Users are discouraged from using this bit. Write data is ignored since the reserved bit is permanently set to 0.

5.0 Registers (Continued)

NOISE PRESCALE THRESHOLD REGISTER (NPTR)

The Noise Prescale Threshold Register contains the start value for the Noise Prescale Counter. The Noise Prescale Counter is a count-down counter and it is used in conjunction with the Noise Counter for counting the Noise events. The Noise Prescale Counter decrements once every 80 ns while there is a noise event. When the Noise Prescale Counter reaches zero, it reloads the count with the content of the Noise Prescale Threshold Register and also causes the Noise Counter to decrement.

The threshold values for the Noise Counter and Noise Prescale Counter are simultaneously loaded into both counters if one of the following conditions is true:

- (1) Both the Noise Counter and Noise Prescale Counter reach zero and the current Line State is either Noise Line State, Active Line State, or Line State Unknown.

or

- (2) The current Line State is either Halt Line State, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect

or

- (3) The Noise Threshold Register or Noise Prescale Threshold Register goes through a Control Bus Interface write cycle.

In addition, the value of the Noise Prescale Threshold register is loaded into the Noise Prescale Counter if the Noise Prescale Counter reaches zero.

The Noise Counter and Noise Prescale Counter will continue to count, without resetting or reloading the threshold values, if a Line State change occurs and the new line state is either Noise Line State, Active Line State, or Line State Unknown.

When both the Noise Threshold Counter and Noise Counter both reach zero, the Noise Threshold bit of the Receive Condition Register A will be set.

ACCESS RULES

ADDRESS	READ	WRITE
0Eh	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
NPT7	NPT6	NPT5	NPT4	NPT3	NPT2	NPT1	NPT0

Bit	Symbol	Description
D0	NPT0	NOISE PRESCALE THRESHOLD BIT <0> : Least significant bit (LSB) of the start value of the Noise Prescale Counter.
D1-6	NPT1-6	NOISE PRESCALE THRESHOLD BIT <1-6> : Intermediate bits of start value for the Noise Prescale Counter.
D7	NPT7	NOISE PRESCALE THRESHOLD BIT <7> : Most significant bit (MSB) of the start value for the Noise Prescale Counter.

5.0 Registers (Continued)

CURRENT NOISE COUNT REGISTER (CNCR)

The Current Noise Count Register takes a snap-shot of the Noise Counter during every Control Bus Interface read-cycle of this register.

During a Control Bus Interface write-cycle to the Current Noise Count Register, the PLAYER device will set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1 and will ignore a write-cycle.

ACCESS RULES

ADDRESS	READ	WRITE
0Fh	Always	Write Reject

D7	D6	D5	D4	D3	D2	D1	D0
NCLSCD	CNC6	CNC5	CNC4	CNC3	CNC2	CNC1	CNC0

Bit	Symbol	Description
D0-6	CNC0-6	CURRENT NOISE COUNT BIT <0-6>
D7	NCLSCD	NOISE COUNTER LINE STATE CHANGE DETECTION

5.0 Registers (Continued)

CURRENT NOISE PRESCALE COUNT REGISTER (CNPCR)

The Current Noise Prescale Count Register takes a snap-shot of the Noise Prescale Counter during every Control Bus Interface read-cycle of this register.

During a Control Bus Interface write-cycle to the Current Noise Prescale Count Register, the PLAYER device will set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1 and will ignore a write-cycle.

ACCESS RULES

ADDRESS	READ	WRITE
10h	Always	Write Reject

D7	D6	D5	D4	D3	D2	D1	D0
CNPC7	CNPC6	CNPC5	CNPC4	CNPC3	CNPC2	CNPC1	CNPC0

Bit	Symbol	Description
D0-7	CNPC0-7	CURRENT NOISE PRESCALE COUNT BY <0-7>

5.0 Registers (Continued)

STATE THRESHOLD REGISTER (STR)

The State Threshold Register contains the start value of the State Counter. This counter is used in conjunction with the State Prescale Counter to count the Line State duration. The State Counter will decrement every 80 ns if the State Prescale Counter is zero and the current Line State is Halt Line, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect. State The State Counter takes

$$((SPTR + 1) \times (STR + 1)) \times 80 \text{ ns}$$

to reach zero during a continuous line state condition.

The threshold values for the State Counter and State Prescale Counter are simultaneously loaded into both counters if one of the following conditions is true:

(1) Both the State Counter and State Prescale Counter reach zero and the current Line State is Halt Line State, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect

or

(2) A line state change occurs and the new Line State is Halt Line State, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect

or

(3) The State Threshold Register or State Prescale Threshold Register goes through a Control Bus Interface write cycle.

In addition, the value of the State Prescale Threshold register is loaded into the State Prescale Counter if the State Prescale Counter reaches zero.

The State Counter and State Prescale Counter will reset by reloading the threshold values, if a Line State change occurs and the new Line State is Halt Line State, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect.

ACCESS RULES

ADDRESS	READ	WRITE
11h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0

Bit	Symbol	Description
D0	ST0	STATE THRESHOLD BIT <0> : Least significant bit (LSB) of the start value for the State Counter.
D1–5	ST1–5	STATE THRESHOLD BIT <1–5> : Intermediate bits of start value for the State Counter.
D6	ST6	STATE THRESHOLD BIT <6> : Most significant bit (MSB) of the start value for the State Counter.
D7	RES	RESERVED : Reserved for future use. Note: Users are discouraged from using this bit. Write data is ignored since the reserved bit is permanently set to 0.

5.0 Registers (Continued)

STATE PRESCALE THRESHOLD REGISTER (SPTR)

The State Prescale Threshold Register contains the start value for the State Prescale Counter. The State Prescale Counter is a down counter. The Register is used in conjunction with the State Counter to count the Line State duration.

The State Prescale Counter will decrement every 80 ns if the current Line State is Halt Line State, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect. As a result, the State Prescale Counter takes $SPTR \times 80$ ns to reach zero during a continuous line state condition. When the State Prescale Counter reaches zero, the State Prescale Threshold Register will be reloaded into the State Prescale Counter.

The threshold values for the State Counter and State Prescale Counter are simultaneously loaded into both counters if one of the following conditions is true:

(1) Both the State Counter and State Prescale Counter reach zero and the current Line State is Halt Line State, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect.

or

(2) A Line State change occurs and the new Line State is Halt Line State, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect

or

(3) The State Threshold Register or State Prescale Threshold Register goes through a Control Bus Interface write cycle.

The State Counter and State Prescale Counter will reset by reloading the threshold values, if a Line State change occurs and the new Line State is Halt Line State, Idle Line State, Master Line State, Quiet Line State, or No Signal Detect.

ACCESS RULES

ADDRESS	READ	WRITE
12h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
SPT7	SPT6	SPT5	SPT4	SPT3	SPT2	SPT1	SPT0

Bit	Symbol	Description
D0	SPT0	STATE PRESCALE THRESHOLD BIT <0> : Least significant bit (LSB) of the start value for the State Prescale Counter.
D1-6	SPT1-6	STATE PRESCALE THRESHOLD BIT <1-6> : Intermediate bits of start value for the State Prescale Counter.
D7	SPT7	STATE PRESCALE THRESHOLD BIT <7> : Most significant bit (MSB) of the start value for the State Prescale Counter.

5.0 Registers (Continued)

CURRENT STATE COUNT REGISTER (CSCR)

The Current State Count Register takes a snap-shot of the State Counter during every Control Bus Interface read-cycle of this register.

During a Control Bus Interface write-cycle to the Current State Count Register, the PLAYER device will set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1 and will ignore a write-cycle.

ACCESS RULES

ADDRESS	READ	WRITE
13h	Always	Write Reject

D7	D6	D5	D4	D3	D2	D1	D0
SCLSCD	CSC6	CSC5	CSC4	CSC3	CSC2	CSC1	CSC0

Bit	Symbol	Description
D0-6	CSC0-6	CURRENT STATE COUNT BIT <0-6>
D7	SCLSCD	STATE COUNTER LINE STATE CHANGE DETECTION

5.0 Registers (Continued)

CURRENT STATE PRESCALE COUNT REGISTER (CSPCR)

The Current State Prescale Count Register takes a snap-shot of the State Prescale Counter during every Control Bus interface read-cycle of this register.

During a Control Bus Interface write-cycle to the Current State Prescale Count Register, the PLAYER device will set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1 and will ignore a write-cycle.

ACCESS RULES

ADDRESS	READ	WRITE
14h	Always	Write Reject

D7	D6	D5	D4	D3	D2	D1	D0
CSPC7	CSPC6	CSPC5	CSPC4	CSPC3	CSPC2	CSPC1	CSPC0

Bit	Symbol	Description
D0-7	CSPC0-7	CURRENT STATE PRESCALE COUNT <0-7>

5.0 Registers (Continued)

LINK ERROR THRESHOLD REGISTER (LETR)

The Link Error Threshold Register contains the start value for the Link Error Monitor Counter, which is an 8-bit down-counter that decrements if link errors are detected.

When the Counter reaches 0, the Link Error Monitor Threshold Register value is loaded into the Link Error Monitor Counter and the Link Error Monitor Threshold bit (LEMT) of the Interrupt Condition Register (ICR) is set to 1.

The Link Error Monitor Threshold Register value is also loaded into the Link Error Monitor Counter during every Control Bus Interface write-cycle of LETR.

The Counter is initialized to 0 during the reset process (i.e. $\overline{RST} = \text{GND}$).

ACCESS RULES

ADDRESS	READ	WRITE
15h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
LET7	LET6	LET5	LET4	LET3	LET2	LET1	LET0

Bit	Symbol	Description
D0	LET0	LINK ERROR THRESHOLD BIT <0> : Least significant bit of the start value for the Link Error Monitor Counter.
D1–6	LET1–6	LINK ERROR THRESHOLD BIT <1–6> : Intermediate bits of start value for the Link Error Monitor Counter.
D7	LET7	LINK ERROR THRESHOLD BIT <7> : Most significant bit of the start value for the Link Error Monitor Counter.

5.0 Registers (Continued)

CURRENT LINK ERROR COUNT REGISTER (CLECR)

The Current Link Error Count Register takes a snap-shot of the Link Error Monitor Counter during every Control Bus Interface read-cycle of this register.

During a Control Bus Interface write-cycle, the PLAYER device will set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1 and will ignore a write-cycle.

ACCESS RULES

ADDRESS	READ	WRITE
16h	Always	Write Reject

D7	D6	D5	D4	D3	D2	D1	D0
LEC7	LEC6	LEC5	LEC4	LEC3	LEC2	LEC1	LEC0

Bit	Symbol	Description
D0-7	LEC0-7	LINK ERROR COUNT BIT <0-7>

5.0 Registers (Continued)

USER DEFINABLE REGISTER (UDR)

The User Definable Register is used to monitor and control events which are external to the PLAYER device.

The value of the Sense Bits reflects the asserted/deasserted state of their corresponding Sense pins. On the other hand, the Enable bits assert/deassert the Enable pins.

ACCESS RULES

ADDRESS	READ	WRITE
17h	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	EB1	EB0	SB1	SB0

Bit	Symbol	Description
D0	SB0	SENSE BIT 0: This bit is set to 1 if the Sense Pin 0 (SP0) is asserted (i.e. $SP0 = V_{CC}$) for a minimum of 160 ns. Once the asserted signal is latched, Sense Bit 0 can only be cleared through the Control Bus Interface, even if the signal is deasserted. This ensures that the Control Bus Interface will record the source of events which can cause interrupts in a traceable manner.
D1	SB1	SENSE BIT 1: This bit is set to 1 if the Sense Pin 1 (SP1) is asserted (i.e. $SP1 = V_{CC}$) for a minimum of 160 ns. Once the asserted signal is latched, Sense Bit 1 can only be cleared through the Control Bus Interface, even if the signal is deasserted. This ensures that the Control Bus Interface will record the source of events which can cause interrupts in a traceable manner.
D2	EB0	ENABLE BIT 0: The Enable Bit 0 allows control of external logic through the Control Bus Interface. The User Definable Enable Pin 0 (EP0) is asserted/deasserted by this bit. 0: EP0 is deasserted (i.e. $EP0 = GND$). 1: EP0 is asserted (i.e. $EP0 = V_{CC}$).
D3	EB1	ENABLE BIT 1: This bit allows control of external logic through the Control Bus Interface. The User Definable Enable Pin 1 (EP1) is asserted/deasserted by this bit. 0: EP1 is deasserted (i.e. $EP1 = GND$). 1: EP1 is asserted (i.e. $EP1 = V_{CC}$).
D4–7	RES	RESERVED: Reserved for future use. The reserved bit is set to 0 during the initialization process (i.e. $\overline{RST} = GND$). Note: Users are discouraged from using this bit. It may be set or cleared without any effects to the functionality of the PLAYER device.

5.0 Registers (Continued)

DEVICE ID REGISTER (IDR)

The Device ID Register contains the binary equivalent of the revision number for this device. It can be used to ensure proper software and hardware versions are matched.

During the Control Bus Interface write-cycle, the PLAYER device will set the Control Bus Write Command Register bit (CCR) of the Interrupt Condition Register (ICR) to 1, and will ignore write-cycle.

ACCESS RULES

ADDRESS	READ	WRITE
18h	Always	Write Reject

D7	D6	D5	D4	D3	D2	D1	D0
DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0

Bit	Symbol	Description
D0	DID0	DEVICE ID BIT <0> : Least significant bit (LSB) of the revision number.
D1-6	DID1-6	DEVICE ID BIT <1-0-6> : Intermediate bits of the revision number.
D7	DID7	DEVICE ID BIT <7> : Most significant bit (MSB) of the revision number.

5.0 Registers (Continued)

CURRENT INJECTION COUNT REGISTER (CIJCR)

The Current Injection Count Register takes a snap-shot of the Injection Counter during every Control Bus Interface read-cycle of this register.

During a Control Bus Interface write-cycle, the PLAYER device will set the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR) to 1 and will ignore a write-cycle.

The Injection Counter is an 8-bit down-counter which decrements every 80 ns.

The counter is active only during One Shot or Periodic Injection Modes (i.e. Injection Control <1:0> bits (IC<1:0>) of the Current Transmit State Register (CTSR) are set to either 01 or 10).

The Injection Threshold Register (IJTR) value is loaded into the Injection Counter when the counter reaches zero and during every Control Bus Interface write-cycle of IJTR.

The counter is initialized to 0 during the reset process (i.e. $\overline{RST} = GND$).

ACCESS RULES

ADDRESS	READ	WRITE
19h	Always	Write Reject

D7	D6	D5	D4	D3	D2	D1	D0
IJC7	IJC6	IJC5	IJC4	IJC3	IJC2	IJC1	IJC0

Bit	Symbol	Description
D0	IJC0	INJECTION COUNT BIT <0> : Least significant bit (LSB) of the current value of the Injection Counter.
D1–6	IJC1–6	INJECTION COUNT BIT <1–6> : Intermediate bits representing the current value of the Injection Counter.
D7	IJC7	INJECTION COUNT BIT <7> : Most significant bit (MSB) of the current value of the Injection Counter.

5.0 Registers (Continued)

INTERRUPT CONDITION COMPARISON REGISTER (ICCR)

The Interrupt Condition Comparison Register ensures that the Control Bus must first read a bit modified by the PLAYER device before it can be written to by the Control Bus Interface.

The current state of the Interrupt Condition Register (ICR) is automatically written into the Interrupt Condition Comparison Register (i.e. ICCR = ICR) during a Control Bus Interface read-cycle of ICR.

During a Control Bus Interface write-cycle, the PLAYER device will set the Conditional Write Inhibit bit (CWI) of the Interrupt Condition Register (ICR) to 1 and disallow the setting or clearing of a bit within ICR when the value of a bit in ICR differs from the value of the corresponding bit in the Interrupt Condition Comparison Register.

ACCESS RULES

ADDRESS	READ	WRITE
1Ah	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
UDIC	RCBC	RCAC	LEMTC	CWIC	CCRC	CPEC	DPEC

Bit	Symbol	Description
D0	DPEC	PHY__REQUEST DATA PARITY ERROR COMPARISON: The comparison bit for the PHY__Request Data Parity Error bit (DPE) of the Interrupt Condition Register (ICR).
D1	CPEC	CONTROL BUS DATA PARITY ERROR COMPARISON: The comparison bit for the Control Bus Data Parity Error bit (CPE) of the Interrupt Condition Register (ICR).
D2	CCRC	CONTROL BUS WRITE COMMAND REJECT COMPARISON: The comparison bit for the Control Bus Write Command Reject bit (CCR) of the Interrupt Condition Register (ICR).
D3	CWIC	CONDITIONAL WRITE INHIBIT COMPARISON: The comparison bit for the Conditional Write Inhibit bit (CWI) of the Interrupt Condition Register (ICR).
D4	LEMTC	LINK ERROR MONITOR THRESHOLD COMPARISON: The comparison bit for the Link Error Monitor Threshold bit (LEMT) of the Interrupt Condition Register (ICR).
D5	RCAC	RECEIVE CONDITION A COMPARISON: The comparison bit for the Receive Condition A bit (RCA) of the Interrupt Condition Register (ICR).
D6	RCBC	RECEIVE CONDITION B COMPARISON: The comparison bit for the Receive Condition B bit (RCB) of the Interrupt Condition Register (ICR).
D7	UDIC	USER DEFINABLE INTERRUPT COMPARISON: The comparison bit for the User Definable Interrupt bit (UDIC) of the Interrupt Condition Register (ICR).

5.0 Registers (Continued)

CURRENT TRANSMIT STATE COMPARISON REGISTER (CTSCR)

The Current Transmit State Comparison Register ensures that the Control Bus must first read a bit modified by the PLAYER device before it can be written to by the Control Bus Interface.

The current state of the Current Transmit State Register (CTSR) is automatically written into the Current Transmit State Comparison Register A (i.e. CTSCR = CTSR) during a Control Bus Interface read-cycle of CTSR.

During a Control Bus Interface write-cycle, the PLAYER device will set the Conditional Write Inhibit bit (CWI) of the Interrupt Condition Register (ICR) to 1 and disallow the setting or clearing of a bit within the CTSR when the value of a bit in the CTSR differs from the value of the corresponding bit in the Current Transmit State Comparison Register.

ACCESS RULES

ADDRESS	READ	WRITE
1Bh	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
RESC	PRDPEC	SEC	IC1C	IC0C	TM2C	TM1C	TM0C

Bit	Symbol	Description
D0	TM0C	TRANSMIT MODE <0> COMPARISON: The comparison bit for the Transmit Mode <0> (TM0) of the Current Transmit State Register (CTSR).
D1	TM1C	TRANSMIT MODE <1> COMPARISON: The comparison bit for the Transmit Mode <1> bit (TM1) of the Current Transmit State Register (CTSR).
D2	TM2C	TRANSMIT MODE <2> COMPARISON: The comparison bit for the Transmit Mode <2> bit (TM2) of the Current Transmit State Register (CTSR).
D3	IC0C	INJECTION CONTROL <0> COMPARISON: The comparison bit for the Injection Control <0> bit (IC0) of the Current Transmit State Register (CTSR).
D4	IC1C	INJECTION CONTROL <1> COMPARISON: The comparison bit for the Injection Control <1> bit (IC1) of the Current Transmit Register (CTSR).
D5	SEC	SMOOTHER ENABLE COMPARISON: The comparison bit for the Smoother Enable bit (SE) to the Current Transmit State Register (CTSR).
D6	PRDPEC	PHY__REQUEST DATA PARITY ENABLE COMPARISON: The comparison bit for the PHY__Request Data Parity Enable bit (PRDPE) of the Current Transmit State Register (CTSR).
D7	RESC	RESERVED COMPARISON: The comparison bit for the Reserved bit (RES) of the Current Transmit State Register (CTSR).

5.0 Registers (Continued)

RECEIVE CONDITION COMPARISON REGISTER A (RCCRA)

The Receive Condition Comparison Register A ensures that the Control Bus must first read a bit modified by the PLAYER device before it can be written to by the Control Bus Interface.

The current state of RCRA is automatically written into the Receive Condition Comparison Register A (i.e. RCCRA = RCRA) during a Control Bus Interface read-cycle of RCRA.

During a Control Bus Interface write-cycle, the PLAYER device will set the Conditional Write Inhibit bit (CWI) of the Interrupt Condition Register (ICR) to 1 and prevent the setting or clearing of a bit within RCRA when the value of a bit in RCRA differs from the value of the corresponding bit in the Receive Condition Comparison Register A.

ACCESS RULES

ADDRESS	READ	WRITE
1Ch	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
LSUPIC	LSCC	NTC	NLSC	MLSC	HLSC	QLSC	NSDC

Bit	Symbol	Description
D0	NSDC	NO SIGNAL DETECT COMPARISON: The comparison bit for the No Signal Detect bit (NSD) of the Receive Condition Register A (RCRA).
D1	QLSC	QUIET LINE STATE COMPARISON: The comparison bit for the Quiet Line State bit (QLS) of the Receive Condition Register A (RCRA).
D2	HLSC	HALT LINE STATE COMPARISON: The comparison bit for the Halt Line State bit (HLS) of the Receive Condition Register A (RCRA).
D3	MLSC	MASTER LINE STATE COMPARISON: The comparison bit for the Master Line State bit (MLS) of the Receive Condition Register A (RCRA).
D4	NLSC	NOISE LINE STATE COMPARISON: The comparison bit for the Noise Line State bit (NLS) of the Receive Condition Register A (RCRA).
D5	NTC	NOISE THRESHOLD COMPARISON: The comparison bit for the Noise Threshold bit (NT) of the Receive Condition Register A (RCRA).
D6	LSCC	LINE STATE CHANGE COMPARISON: The comparison bit for the Line State Change bit (LSC) of the Receive Condition Register A (RCRA).
D7	LSUPIC	LINE STATE UNKNOWN & PHY INVALID COMPARISON: The comparison bit for the Line State Unknown & PHY Invalid bit (LSUPI) of the Receive Condition Register A (RCRA).

5.0 Registers (Continued)

RECEIVE CONDITION COMPARISON REGISTER B (RCRB)

The Receive Condition Comparison Register B ensures that the Control Bus must first read a bit modified by the PLAYER device before it can be written to by the Control Bus Interface.

The current state of RCRB is automatically written into the Receive Condition Comparison Register B (i.e. RCCR = RCRB) during a Control Bus Interface read-cycle RCRB.

During a Control Bus Interface write-cycle, the PLAYER device will set the Conditional Write Inhibit bit (CWI) of the Interrupt Condition Register (ICR) to 1 and prevent the setting or clearing of a bit within RCRB when the value of a bit in RCRB differs from the value of the corresponding bit in the Receive Condition Comparison Register B.

ACCESS RULES

ADDRESS	READ	WRITE
1Dh	Always	Always

D7	D6	D5	D4	D3	D2	D1	D0
RESC	SILSC	EBOUC	CSEC	LSUPVC	ALSC	STC	ILSC

Bit	Symbol	Description
D0	ILSC	IDLE LINE STATE COMPARISON: The comparison bit for the Idle State bit (ILS) of the Receive Condition Register B (RCRB).
D1	STC	STATE THRESHOLD COMPARISON: The comparison bit for the State Threshold bit (ST) of the Receive Condition Register B (RCRB).
D2	ALSC	ACTIVE LINE STATE COMPARISON: The comparison bit for the Active Line State bit (ALS) of the Receive Condition Register B (RCRB).
D3	LSUPVC	LINE STATE UNKNOWN & PHY VALID COMPARISON: The comparison bit for the Line State Unknown & PHY Valid bit (LSUPV) of the Receive Condition Register B (RCRB).
D4	CSEC	CASCADE SYNCHRONIZATION ERROR COMPARISON: The comparison bit for the Cascade Synchronization Error bit (CSE) of the Receive Condition Register B (RCRB).
D5	EBOUC	ELASTICITY BUFFER OVERFLOW/UNDERFLOW COMPARISON: The comparison bit for the Elasticity Buffer Overflow/Underflow bit (EBOU) of the Receive Condition Register B (RCRB).
D6	SILSC	SUPER IDLE LINE STATE COMPARISON: The comparison bit for the Super Idle Line State bit (SILS) of the Receive Condition Register B (RCRB).
D7	RESC	RESERVED COMPARISON: The comparison bit for the Reserved bit (RES) of the Receive Condition Register B (RCRB).

RESERVED REGISTER 0 (RR0) ADDRESS 1Eh—DO NOT USE

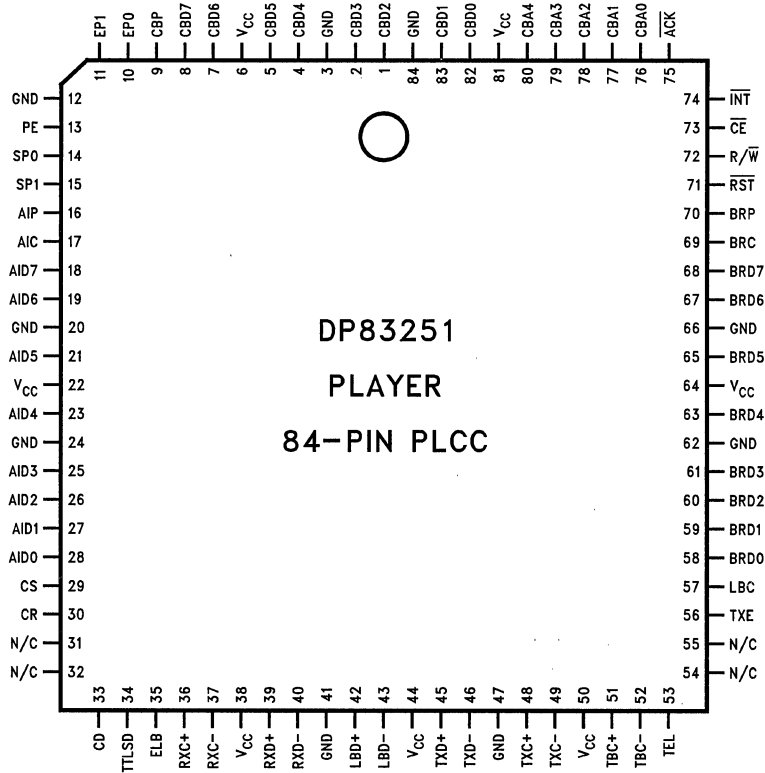
RESERVED REGISTER 1 (RR1) ADDRESS 1Fh—DO NOT USE

6.0 Pin Descriptions

6.1 DP83251

The pin descriptions for the DP83251 are divided into 5 functional interfaces: Serial Interface, PHY Port Interface, Control Bus Interface, Clock Interface, and Miscellaneous Interface.

For a Pinout Summary List, refer to Table 6-1.



Order Number DP83251V
See NS Package Number V84A

FIGURE 6-1. DP83251 84-Pin PLCC Pinout

TL/F/10386-20

6.0 Pin Descriptions (Continued)

TABLE 6-1. DP83251 Pinout Summary

Pin No.	Signal Name	Symbol	I/O	ECL/TTL/Open Drain/Power
1	Control Bus Data <2>	CBD2	I/O	TTL
2	Control Bus Data <3>	CBD3	I/O	TTL
3	CMOS I/O Ground	GND		+0V
4	Control Bus Data <4>	CBD4	I/O	TTL
5	Control Bus Data <5>	CBD5	I/O	TTL
6	CMOS I/O Power	V _{CC}		+5V
7	Control Bus Data <6>	CBD6	I/O	TTL
8	Control Bus Data <7>	CBD7	I/O	TTL
9	Control Bus Data Parity	CBP	I/O	TTL
10	Enable Pin 0	EP0	O	TTL
11	Enable Pin 1	EP1	O	TTL
12	CMOS Logic Ground	GND		+0V
13	Control Bus Data Parity Enable	CBPE	I	TTL
14	Sense Pin 0	SP0	I	TTL
15	Sense Pin 1	SP1	I	TTL
16	PHY Port A Indicate Parity	AIP	O	TTL
17	PHY Port A Indicate Control	AIC	O	TTL
18	PHY Port A Indicate Data <7>	AID7	O	TTL
19	PHY Port A Indicate Data <6>	AID6	O	TTL
20	CMOS I/O Ground	GND		+0V
21	PHY Port A Indicate Data <5>	AID5	O	TTL
22	CMOS I/O Power	V _{CC}		+5V
23	PHY Port A Indicate Data <4>	AID4	O	TTL
24	CMOS Logic Ground	GND		+0V
25	PHY Port A Indicate Data <3>	AID3	O	TTL
26	PHY Port A Indicate Data <2>	AID2	O	TTL
27	PHY Port A Indicate Data <1>	AID1	O	TTL
28	PHY Port A Indicate Data <0>	AID0	O	TTL
29	Cascade Start	CS	O	TTL
30	Cascade Ready	CR	I	Open Drain
31	No Connect	N/C		
32	No Connect	N/C		
33	Clock Detect	CD	I	TTL
34	Signal Detect	TTLSD	I	TTL
35	External Loopback Enable	ELB	O	TTL

6.0 Pin Descriptions (Continued)

TABLE 6-1. DP83251 Pinout Summary (Continued)

Pin No.	Signal Name	Symbol	I/O	ECL/TTL/Open Drain/Power
36	Receive Bit Clock +	RXC+	I	ECL
37	Receive Bit Clock -	RXC-	I	ECL
38	ECL Logic Power	V _{CC}		+5V
39	Receive Data +	RXD+	I	ECL
40	Receive Data -	RXD-	I	ECL
41	ECL Logic Ground	GND		+0V
42	External Loopback Data +	LBD+	O	ECL
43	External Loopback Data -	LBD-	O	ECL
44	ECL I/O Power	V _{CC}		+5V
45	Transmit Data +	TXD+	O	ECL
46	Transmit Data -	TXD-	O	ECL
47	ECL Logic Ground	GND		+0V
48	Transmit Bit Clock +	TXC+	I	ECL
49	Transmit Bit Clock -	TXC-	I	ECL
50	ECL Logic Power	V _{CC}		+5V
51	Transmit Byte Clock +	TBC+	I	ECL
52	Transmit Byte Clock -	TBC-	I	ECL
53	FOTX Enable Level	TEL	I	TTL
54	No Connect	N/C		
55	No Connect	N/C		
56	FOTX Enable	TXE	O	TTL
57	Local Byte Clock	LBC	I	TTL
58	PHY Port B Request Data <0>	BRD0	I	TTL
59	PHY Port B Request Data <1>	BRD1	I	TTL
60	PHY Port B Request Data <2>	BRD2	I	TTL
61	PHY Port B Request Data <3>	BRD3	I	TTL
62	CMOS Logic Ground	GND		+0V
63	PHY Port B Request Data <4>	BRD4	I	TTL
64	CMOS I/O Power	V _{CC}		+5V
65	PHY Port B Request Data <5>	BRD5	I	TTL
66	CMOS I/O Ground	GND		+0V
67	PHY Port B Request Data <6>	BRD6	I	TTL
68	PHY Port B Request Data <7>	BRD7	I	TTL
69	PHY Port B Request Control	BRC	I	TTL
70	PHY Port B Request Parity	BRP	O	TTL

6.0 Pin Descriptions (Continued)

TABLE 6-1. DP83251 Pinout Summary (Continued)

Pin No.	Signal Name	Symbol	I/O	ECL/TTL/Open Drain/Power
71	~ PLAYER Device Reset	$\overline{\text{RST}}$	I	TTL
72	Read/ ~ Write	$\text{R}/\overline{\text{W}}$	I	TTL
73	Chip Enable	$\overline{\text{CE}}$	I	TTL
74	~ Interrupt	$\overline{\text{INT}}$	O	Open Drain
75	~ Acknowledge	$\overline{\text{ACK}}$	O	Open Drain
76	Control Bus Address<0>	CBA0	I	TTL
77	Control Bus Address<1>	CBA1	I	TTL
78	Control Bus Address<2>	CBA2	I	TTL
79	Control Bus Address<3>	CBA3	I	TTL
80	Control Bus Address<4>	CBA4	I	TTL
81	CMOS Logic Power	V _{CC}		+5V
82	Control Bus Data<0>	CBD0	I/O	TTL
83	Control Bus Data<1>	CBD1	I/O	TTL
84	CMOS Logic Ground	GND		+0V

6.0 Pin Descriptions (Continued)

SERIAL INTERFACE

The Serial Interface consists of I/O signals used to connect the PLAYER device to the Physical Medium Dependent (PMD) sublayer.

The PLAYER device uses these signals to interface to a Fiber Optic Transmitter (FOTX), Fiber Optic Receiver (FOXR), Clock Recovery Device (CRD device), and Clock Distribution Device (CDD device).

Symbol	Pin No.	I/O	Description
CD	33	I	Clock Detect: A TTL input signal from the Clock Recovery Device indicating that the Receive Clock (RXC \pm) is properly synchronized with the Receive Data RXD \pm .
TTLSD	34	I	Signal Detect: A TTL signal from the clock Recovery Device indicating that a signal is being received by the Fiber Optic Receiver.
RXD+ RXD-	39 40	I	Receive Data: Differential 100K ECL, 125 Mbps serial data input signals from the Clock Recovery Device.
TXD+ TXD-	45 46	O	Transmit Data: Differential, 100K ECL, 125 Mbps serial data output signals to the Fiber Optic Transmitter.
ELB	35	O	External Loopback Enable: A TTL output signal to the Clock Recovery Device which enables/disables loopback data through the Clock Recovery Device. This signal is controlled by the Mode Register.
LBD+ LBD-	42 43	O	Loopback Data: Differential, 100K ECL, 125 Mbps, external serial loopback data output signals to the Clock Recovery Device. When the PLAYER device is not in external loopback mode, the LBD+ signal is kept high and the LBD- signal is kept low.
TEL	53	I	FOTX Enable Level: A TTL input signal to select the Fiber Optic Transmitter Enable (TXE) signal level.
TXE	56	O	FOTX Enable: A TTL output signal to enable/disable the Fiber Optic Transmitter. The output level of the TXE pin is determined by three parameters, the Transmit Enable (TE) bit in the Mode Register, the TM2-TM0 bits in the Current Transmit State Register, and also the input to the TEL pin. The following rules summarizes the output of the TXE pin: (1) If TE = 0 and TEL = GND, then TXE = V _{CC} (2) If TE = 0 and TEL = V _{CC} , then TXE = GND (3) If TE = 1 and OTM and TEL = GND, then TXE = V _{CC} (4) If TE = 1 and OTM and TEL = V _{CC} , then TXE = GND (5) If TE = 1 and not OTM and TEL = GND, then TXE = GND (6) If TE = 1 and not OTM and TEL = V _{CC} , then TXE = V _{CC}

6.0 Pin Descriptions (Continued)

PHY PORT INTERFACE

The PHY Port Interface consists of I/O signals used to connect the PLAYER Device to the Media Access Control (MAC) sublayer or other PLAYER Devices. The DP83251 Device has one PHY Port Interface which consists of the B__Request and the A__Indicate paths.

Each path consists of an odd parity bit, a control bit, and two 4-bit symbols.

Refer to Section 3.3, the Configuration Switch, for further information.

Symbol	Pin No.	I/O	Description
AIP	16	O	PHY Port A Indicate Parity: A TTL output signal representing odd parity for the 10-bit wide Port A Indicate signals (AIP, AIC, and AID<7:0>).
AIC	17	O	PHY Port A Indicate Control: A TTL output signal indicating that the two 4-bit symbols (AID<7:4> and AID<3:0>) are either control symbols (AIC = 1) or data symbols (AIC = 0).
AID7 AID6 AID5 AID4	18 19 21 23	O	PHY Port A Indicate Data: TTL output signals representing the first 4-bit data/control symbol. AID7 is the most significant bit and AID4 is the least significant bit of the first symbol.
AID3 AID2 AID1 AID0	25 26 27 28	O	PHY Port A Indicate Data: TTL output signals representing the second 4-bit data/control symbol. AID3 is the most significant bit and AID0 is the least significant bit of the second symbol.
BRP	70	I	PHY Port B Request Parity: A TTL input signal representing odd parity for the 10-bit wide Port A Request signals (BRP, BRC, and BRD<7:0>).
BRC	69	I	PHY Port B Request Control: A TTL input signal indicating that the two 4-bit symbols (BRD<7:4>) and BRD<3:0>) are either control symbols (BRC = 1) or data symbols (BRC = 0).
BRD7 BRD6 BRD5 BRD4	68 67 65 63	I	PHY Port B Request Data: TTL input signals representing the first 4 bit data/control symbol. BRD7 is the most significant bit and BRD4 is the least significant bit of the first symbol.
BRD3 BRD2 BRD1 BRD0	61 60 59 58	I	PHY Port B Request Data: TTL input signals representing the second 4-bit data/control symbol. BRD3 is the most significant bit and BRD0 is the least significant bit of the second symbol.

6.0 Pin Descriptions (Continued)

CONTROL BUS INTERFACE

The Control Bus Interface consists of I/O signals used to connect the PLAYER device to Station Management (SMT).

The Control Bus is an asynchronous interface between the PLAYER device and a general purpose microprocessor. It provides access to 32 8-bit internal registers.

Refer to *Figure 22, Control Bus Timing Diagram*, for more information.

Symbol	Pin No.	I/O	Description
\overline{CE}	73	I	Chip Enable: An active-low, TTL, input signal which enables the Control Bus port for a read or write cycle. R/\overline{W} , $CBA < 4:0 >$, CBP , and $CBD < 7:0 >$ must be valid at the time \overline{CE} is low.
R/\overline{W}	72	I	Read/ ~ Write: A TTL input signal which indicates a read Control Bus cycle ($R/\overline{W} = 1$), or a write Control Bus cycle ($R/\overline{W} = 0$). This signal must be valid when \overline{CE} is low and held valid until \overline{ACK} becomes low.
\overline{ACK}	75	O	~ Acknowledge: An active low, TTL, open drain output signal which indicates the completion of a read or write cycle. During a read cycle, $CBD < 7:0 >$ are valid as long as \overline{ACK} is low ($\overline{ACK} = 0$). During a write cycle, a microprocessor must hold $CBD < 7:0 >$ valid until \overline{ACK} becomes low. Once \overline{ACK} is low, it will remain low as long as \overline{CE} remains low ($\overline{CE} = 0$).
\overline{INT}	74	O	~ Interrupt: An active low, open drain, TTL, output signal indicating that an interrupt condition has occurred. The Interrupt Condition Register (ICR) should be read in order to find out the source of the interrupt. Interrupts can be masked through the use of the Interrupt Condition Mask Register (ICMR).
CBA4 CBA3 CBA2 CBA1 CBA0	80 79 78 77 76	I	Control Bus Address: TTL input signals used to select the address of the register to be read or written. CBA4 is the most significant bit (MSB), CBA0 is the least significant bit (LSB) of the address signals. These signals must be valid when \overline{CE} is low and held valid until \overline{ACK} becomes low.
CBPE	13	I	Control Bus Parity Enable: A TTL input signal which, during write cycles, will enable or disable the Control Bus parity checker. Note that the Control Bus will always generate parity during read cycles, regardless of the state of this signal.
CBP	9	I/O	Control Bus Parity: A bidirectional, TTL signal representing odd parity for the Control Bus data ($CBD < 7:0 >$). During a read cycle, the signal is held valid by the PLAYER device as long as \overline{ACK} is low. During a write cycle, the signal must be valid when \overline{CE} is low, and must be held valid until \overline{ACK} becomes low. If incorrect parity is used during a write cycle, the PLAYER device will inhibit the write cycle and set the Control Bus Data Parity Error (CPE) bit in the Interrupt Condition Register (ICR).
CBD7 CBD6 CBD5 CBD4 CBD3 CBD2 CBD1 CBD0	8 7 5 4 2 1 83 82	I/O	Control Bus Data: Bidirectional, TTL signals containing the data to be read from or written to a register. During a read cycle, the signal is held valid by the PLAYER device as long as \overline{ACK} is low. During a write cycle, the signal must be valid when \overline{CE} is low, and must be held valid until \overline{ACK} becomes low.

6.0 Pin Descriptions (Continued)

CLOCK INTERFACE

The Clock Interface consists of 12.5 MHz and 125 MHz clocks used by the PLAYER device. The clocks are generated by either the Clock Distribution Device or Clock Recovery Device.

Symbol	Pin No.	I/O	Description
LBC	57	I	Local Byte Clock: A TTL, 12.5 MHz, 50% duty cycle, input clock from the Clock Distribution Device. The Local Byte Clock is used by the PLAYER device's internal CMOS logic and to latch incoming/outgoing data of the Control Bus Interface, Port A Interface, Port B Interface, and other miscellaneous I/Os.
RXC+ RXC-	36 37	I	Receive Bit Clock: Differential 100k ECL, 125 MHz clock input signals from the Clock Recovery Device. The Receive Bit Clock is used by the Serial Interface to latch the Receive Data (RXD \pm).
TXC+ TXC-	48 49	I	Transmit Bit Clock: Differential 100k ECL, 125 MHz clock input signals from the Clock Distribution Device. The Transmit Bit Clock is used by the Serial Interface to latch the Transmit Data (TXD \pm).
TBC+ TBC-	51 52	I	Transmit Byte Clock: Differential 100k ECL, 12.5 MHz clock input signals from the Clock Distribution Device. The Transmit Byte Clock is used by the PLAYER device's internal Shift Register Block.

6.0 Pin Descriptions (Continued)

MISCELLANEOUS INTERFACE

The Miscellaneous Interface consists of a reset signal, user definable sense signals, user definable enable signals, Cascaded PLAYER devices synchronization signals, ground signals, and power signals.

Symbol	Pin No.	I/O	Description
$\overline{\text{RST}}$	71	I	<p>Reset: An active low, TTL, input signal which clears all registers. The signal must be kept asserted for a minimum of 160 ns.</p> <p>Once the $\overline{\text{RST}}$ signal is asserted, the PLAYER device should be allowed 960 ns to reset internal logic. Note that bit zero of the Mode Register will be set to zero (i.e. Stop Mode). See Section 4.2, Stop Mode of Operation for more information.</p>
SP0	14	I	<p>User Definable Sense Pin 0: A TTL input signal from a user defined source. Bit zero (Sense Bit 0) of the User Definable Register (UDR) will be set to one if the signal is asserted for a minimum of 160 ns.</p> <p>Once the asserted signal is latched, Sense Bit 0 can only be cleared through the Control Bus Interface, even if the signal is deasserted. This ensures that the Control Bus Interface will record the source of events which can cause interrupts.</p>
SP1	15	I	<p>User Definable Sense Pin 1: A TTL input signal from a user defined source. Bit one (Sense Bit 1) of the User Definable Register (UDR) will be set to one if the signal is asserted for a minimum of 160 ns.</p> <p>Once the asserted signal is latched, Sense Bit 1 can only be cleared through the Control Bus Interface, even if the signal is deasserted. This ensures that the Control Bus Interface will record the source of events which can cause interrupts.</p>
EP0	10	O	<p>User Definable Enable Pin 0: A TTL output signal allowing control of external logic through the CBUS Interface. EP0 is asserted/deasserted through bit two (Enable Bit 0) of the User Definable Register (UDR). When Enable Bit 0 is set to zero, EP0 is deasserted. When Enable Bit 0 is set to one, EP0 is asserted.</p>
EP1	11	O	<p>User Definable Enable Pin 1: A TTL output signal allowing control of external logic through the CBUS Interface. EP1 is asserted/deasserted through bit two (Enable Bit 1) of the User Definable Register (UDR). When Enable Bit 1 is set to zero, EP1 is deasserted. When Enable Bit 1 is set to one, EP1 is asserted.</p>
CS	29	I	<p>Cascade Start: A TTL input signal used to synchronize cascaded PLAYER devices in point-to-point applications.</p> <p>The signal is asserted when all of the cascaded PLAYER devices have the Cascade Mode (CM) bit of Mode Register (MR) set to one, and all of the Cascade Ready pins of the cascaded PLAYER devices have been released. For further information, refer to Section 4.4, Cascade Mode of Operation.</p>
CR	30	O	<p>Cascade Ready: An Open Drain output signal used to synchronize cascaded PLAYER devices in point-to-point applications.</p> <p>The signal is released (i.e. an Open Drain line is released) when all the cascaded PLAYER devices have the Cascade Mode (CM) bit of the Mode Register (MR) set to one and a JK symbol pair has been received. For further information, refer to Section 4.4, Cascade Mode of Operation.</p>

6.0 Pin Descriptions (Continued)

POWER AND GROUND

All power pins should be connected to a single 5V power supply. All ground pins should be connected to a common 0V supply.

Symbol	Pin No.	I/O	Description
GND	3		Ground: Power supply return for Control Bus Interface CMOS I/Os.
V _{CC}	6		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for Control Bus Interface CMOS I/Os.
GND	12		Ground: Power supply return for internal CMOS logic.
GND	20		Ground: Power supply return for Port A Interface CMOS I/Os.
V _{CC}	22		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for the Port A Interface CMOS I/Os.
GND	24		Ground: Power supply return to internal CMOS logic.
V _{CC}	38		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for internal ECL logic.
GND	41		Ground: Power supply return for internal ECL logic.
V _{CC}	44		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for the Serial Interface ECL I/Os.
GND	47		Ground: Power supply return for internal ECL logic.
V _{CC}	50		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for the Serial Interface ECL I/Os.
GND	62		Ground: Power supply return for internal CMOS logic.
V _{CC}	64		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for the Port A Interface CMOS I/Os.
GND	66		Ground: Power supply return for Port A Interface CMOS I/Os.
V _{CC}	81		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for internal CMOS logic.
GND	84		Ground: Power supply return for internal CMOS logic.

NO CONNECT PINS

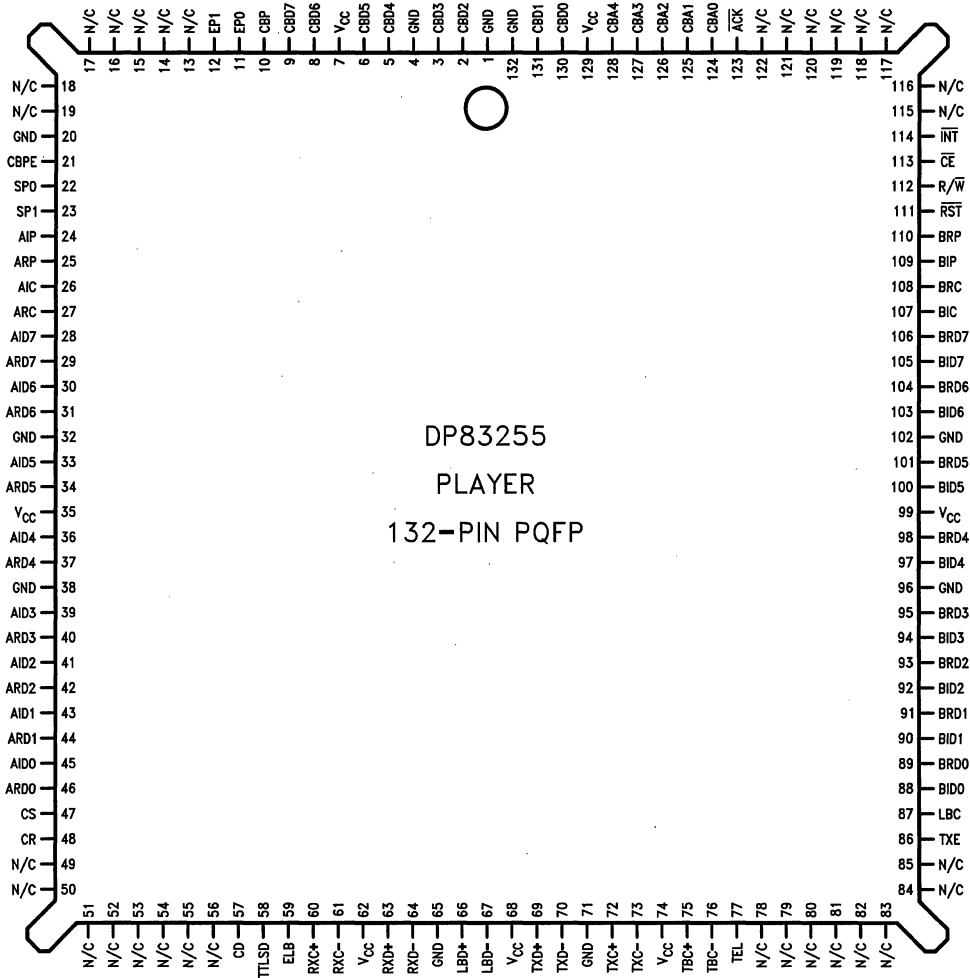
Symbol	Pin No.	I/O	Description
N/C	31		No Connect: Not used by the PLAYER device
N/C	32		No Connect: Not used by the PLAYER device
N/C	54		No Connect: Not used by the PLAYER device
N/C	55		No Connect: Not used by the PLAYER device

6.0 Pin Descriptions (Continued)

6.2 DP83255

The pin descriptions for the DP83255 are divided into six functional interfaces; Serial Interface, PHY Port Interface, Control Bus Interface, Clock Interface, and Miscellaneous Interface.

For a Pinout Summary List, refer to Table 6-2.



TL/F/10386-21

Order Number DP83255AVF
 See NS Package Number VF132A
FIGURE 6-2. DP83255 132-Pin PQFP Pinout

6.0 Pin Descriptions (Continued)

TABLE 6-2. DP83255 Pinout Summary

Pin No.	Signal Name	Symbol	I/O	ECL/TTL/Open Drain/Power
1	CMOS Logic Ground	GND		+0V
2	Control Bus Data <2>	CBD2	I/O	TTL
3	Control Bus Data <3>	CBD3	I/O	TTL
4	CMOS I/O Ground	GND		+0V
5	Control Bus Data <4>	CBD4	I/O	TTL
6	Control Bus Data <5>	CBD5	I/O	TTL
7	CMOS I/O Power	V _{CC}		+5V
8	Control Bus Data <6>	CBD6	I/O	TTL
9	Control Bus Data <7>	CBD7	I/O	TTL
10	Control Bus Data Parity	CBP	I/O	TTL
11	Enable Pin 0	EP0	O	TTL
12	Enable Pin 1	EP1	O	TTL
13	No Connect	N/C		
14	No Connect	N/C		
15	No Connect	N/C		
16	No Connect	N/C		
17	No Connect	N/C		
18	No Connect	N/C		
19	No Connect	N/C		
20	CMOS Logic Ground	GND		+0V
21	Control Bus Data Parity Enable	CBPE	I	TTL
22	Sense Pin 0	SP0	I	TTL
23	Sense Pin 1	SP1	I	TTL
24	PHY Port A Indicate Parity	AIP	O	TTL
25	PHY Port A Request Parity	ARP	I	TTL
26	PHY Port A Indicate Control	AIC	O	TTL
27	PHY Port A Request Control	ARC	I	TTL
28	PHY Port A Indicate Data <7>	AID7	O	TTL
29	PHY Port A Request Data <7>	ARD7	I	TTL
30	PHY Port A Indicate Data <6>	AID6	O	TTL
31	PHY Port A Request Data <6>	ARD6	I	TTL
32	CMOS I/O Ground	GND		+0V
33	PHY A Indicate Data <5>	AID5	O	TTL
34	PHY A Request Data <5>	ARD5	I	TTL
35	CMOS I/O Power	V _{CC}		+5V

6.0 Pin Descriptions (Continued)

TABLE 6-2. DP83255 Pinout Summary (Continued)

Pin No.	Signal Name	Symbol	I/O	ECL/TTL/Open Drain/Power
36	PHY A Indicate Data <4>	AID4	O	TTL
37	PHY A Request Data <4>	ARD4	I	TTL
38	CMOS Logic Ground	GND		+0V
39	PHY Port A Indicate Data <3>	AID3	O	TTL
40	PHY Port A Request Data <3>	ARD3	I	TTL
41	PHY Port A Indicate Data <2>	AID2	O	TTL
42	PHY Port A Request Data <2>	ARD2	I	TTL
43	PHY Port A Indicate Data <1>	AID1	O	TTL
44	PHY Port A Request Data <1>	ARD1	I	TTL
45	PHY Port A Indicate Data <0>	AID0	O	TTL
46	Port A Request Data <0>	ARD0	I	TTL
47	Cascade Start	CS	I	TTL
48	Cascade Ready	CR	O	Open Drain
49	No Connect	N/C		
50	No Connect	N/C		
51	No Connect	N/C		
52	No Connect	N/C		
53	No Connect	N/C		
54	No Connect	N/C		
55	No Connect	N/C		
56	No Connect	N/C		
57	Clock Detect	CD	I	TTL
58	Signal Detect	TTLSD	I	TTL
59	External Loopback Enable	ELB	O	TTL
60	Receive Bit Clock +	RXC+	I	ECL
61	Receive Bit Clock -	RXC-	I	ECL
62	ECL Logic Power	V _{CC}		+5V
63	Receive Data +	RXD+	I	ECL
64	Receive Data -	RXD-	I	ECL
65	ECL Logic Ground	GND		+0V
66	External Loopback Data +	LBD+	O	ECL
67	External Loopback Data -	LBD-	O	ECL
68	ECL I/O Power	V _{CC}		+5V
69	Transmit Data +	TXD+	O	ECL
70	Transmit Data -	TXD-	O	ECL

6.0 Pin Descriptions (Continued)

TABLE 6-2. DP83255 Pinout Summary (Continued)

Pin No.	Signal Name	Symbol	I/O	ECL/TTL/Open Drain/Power
71	ECL Logic Ground	GND		+0V
72	Transmit Bit Clock +	TXC+	I	ECL
73	Transmit Bit Clock –	TXC–	I	ECL
74	ECL Logic Power	V _{CC}		+5V
75	Transmit Byte Clock +	TBC+	I	ECL
76	Transmit Byte Clock –	TBC–	I	ECL
77	FOTX Enable Level	TEL	I	TTL
78	No Connect	N/C		
79	No Connect	N/C		
80	No Connect	N/C		
81	No Connect	N/C		
82	No Connect	N/C		
83	No Connect	N/C		
84	No Connect	N/C		
85	No Connect	N/C		
86	FOTX Enable	TXE	O	TTL
87	Local Byte Clock	LBC	I	TTL
88	PHY Port B Indicate Data <0>	BID0	O	TTL
89	PHY Port B Request Data <0>	BRD0	I	TTL
90	PHY Port B Indicate Data <1>	BID1	O	TTL
91	PHY Port B Request Data <1>	BRD1	I	TTL
92	PHY Port B Indicate Data <2>	BID2	O	TTL
93	PHY Port B Request Data <2>	BRD2	I	TTL
94	PHY Port B Indicate Data <3>	BID3	O	TTL
95	PHY Port B Request Data <3>	BRD3	I	TTL
96	CMOS Logic Ground	GND		+0V
97	PHY Port B Indicate Data <4>	BID4	O	TTL
98	PHY Port B Request Data <4>	BRD4	I	TTL
99	CMOS I/O Power	V _{CC}		+5V
100	PHY Port B Indicate Data <5>	BID5	O	TTL
101	PHY Port B Request Data <5>	BRD5	I	TTL
102	CMOS I/O Ground	GND		+0V
103	PHY Port B Indicate Data <6>	BID6	O	TTL
104	PHY Port B Request Data <6>	BRD6	I	TTL
105	PHY Port B Indicate Data <7>	BID7	O	TTL

6.0 Pin Descriptions (Continued)

TABLE 6-2. DP83255 Pinout Summary (Continued)

Pin No.	Signal Name	Symbol	I/O	ECL/TTL/Open Drain/Power
106	PHY Port B Request Data <7>	BRD7	I	TTL
107	PHY Port B Indicate Control	BIC	O	TTL
108	PHY Port B Request Control	BRC	I	TTL
109	PHY Port B Indicate Parity	BIP	O	TTL
110	PHY Port B Request Parity	BRP	I	TTL
111	~ PLAYER Device Reset	$\overline{\text{RST}}$	I	TTL
112	Read/ ~ Write	R/ $\overline{\text{W}}$	I	TTL
113	Chip Enable	$\overline{\text{CE}}$	I	TTL
114	~ Interrupt	$\overline{\text{INT}}$	O	Open Drain
115	No Connect	N/C		
116	No Connect	N/C		
117	No Connect	N/C		
118	No Connect	N/C		
119	No Connect	N/C		
120	No Connect	N/C		
121	No Connect	N/C		
122	No Connect	N/C		
123	~ Acknowledge	$\overline{\text{ACK}}$	O	Open Drain
124	Control Bus Address <0>	CBA0	I	TTL
125	Control Bus Address <1>	CBA1	I	TTL
126	Control Bus Address <2>	CBA2	I	TTL
127	Control Bus Address <3>	CBA3	I	TTL
128	Control Bus Address <4>	CBA4	I	TTL
129	CMOS Logic Power	V _{CC}		+ 5V
130	Control Bus Data <0>	CBD0	I/O	TTL
131	Control Bus Data <1>	CBD1	I/O	TTL
132	CMOS Logic Ground	GND		+ 0V

6.0 Pin Descriptions (Continued)

SERIAL INTERFACE

The Serial Interface consists of I/O signals used to connect the PLAYER device to the Physical Medium Dependent (PMD) sublayer.

The PLAYER device uses these signals to interface to a Fiber Optic Transmitter (FOTX), Fiber Optic Receiver (FORX), Clock Recovery Device (CRD device), and Clock Distribution Device (CDD device).

Symbol	Pin No.	I/O	Description
CD	57	I	Clock Detect: A TTL input signal from the Clock Recovery Device indicating that the Receive Clock (RXC \pm) is properly synchronized with the Receive Data (RXD \pm).
TTLSD	58	I	Signal Detect: A TTL input signal from the Clock Recovery Device indicating that a signal is being received by the Fiber Optic Receiver.
RXD+ RXD-	63 64	I	Receive Data: Differential 100K ECL, 125 Mbps serial data input signals from the Clock Recovery Device.
TXD+ TXD-	69 70	O	Transmit Data: Differential, 100K ECL, 125 Mbps serial data output signals to the Fiber Optic Transmitter.
ELB	59	O	External Loopback Enable: A TTL output signal to the Clock Recovery Device which enables/disables loopback data through the Clock Recovery Device. This signal is controlled by the Mode Register.
LBD+ LBD-	66 67	O	Loopback Data: Differential, 100K ECL, 125 Mbps, serial external loopback data output signals to the Clock Recovery Device. When the PLAYER device is not in external loopback mode, the LBD+ signal is kept high and the LBD- signal is kept low.
TEL	77	I	FOTX Enable Level: A TTL input signal to select the Fiber Optic Transmitter Enable (TXE) signal level.
TXE	86	O	FOTX Enable: A TTL output signal to enable/disable the Fiber Optic Transmitter. The output level of the TXE pin is determined by three parameters, the Transmit Enable (TE) bit in the Mode Register, the TM2-TM0 bits in the Current Transmit State Register, and also the input to the TEL pin. The following rules summarize the output of the TXE pin: (1) If TE = 0 and TEL = GND, then TXE = V _{CC} (2) If TE = 0 and TEL = V _{CC} , then TXE = GND (3) If TE = 1 and OTM and TEL = GND, then TXE = V _{CC} (4) If TE = 1 and OTM and TEL = V _{CC} , then TXE = GND (5) If TE = 1 and not OTM and TEL = GND, then TXE = GND (6) If TE = 1 and not OTM and TEL = V _{CC} , then TXE = V _{CC}

6.0 Pin Descriptions (Continued)

PHY PORT INTERFACE

The PHY Port Interface consists of I/O signals used to connect the PLAYER Device to the Media Access Control (MAC) sublayer or other PLAYER Devices. The DP83255 Device has two PHY Port Interfaces. The A__Request and A__Indicate paths form one PHY Port Interface and the B__Request and B__Indicate paths form the second PHY Port Interface. Each path consists of an odd parity bit, a control bit, and two 4-bit symbols.

Refer to Section 3.3, the Configuration Switch, for more information.

Symbol	Pin No.	I/O	Description
AIP	24	O	PHY Port A Indicate Parity: A TTL output signal representing odd parity for the 10-bit wide Port A Indicate signals (AIP, AIC, and AID<7:0>).
AIC	26	O	PHY Port A Indicate Control: A TTL output signal indicating that the two 4-bit symbols (AID<7:4> and AID<3:0>) are either control symbols (AIC = 1) or data symbols (AIC = 0).
AID7 AID6 AID5 AID4	28 30 33 36	O	PHY Port A Indicate Data: TTL output signals representing the first 4-bit data/control symbol. AID7 is the most significant bit and AID4 is the least significant bit of the first symbol.
AID3 AID2 AID1 AID0	39 41 43 45	O	PHY Port A Indicate Data: TTL output signals representing the second 4-bit data/control symbol. AID3 is the most significant bit and AID0 is the least significant bit of the second symbol.
ARP	25	I	PHY Port A Request Parity: A TTL input signal representing odd parity for the 10-bit wide Port A Request signals (ARP, ARC, and ARD<7:0>).
ARC	27	I	PHY Port A Request Control: A TTL input signal indicating that the two 4-bit symbols (ARD<7:4> and ARD<3:0>) are either control symbols (ARC = 1) or data symbols (ARC = 0).
ARD7 ARD6 ARD5 ARD4	29 31 34 37	I	PHY Port A Request Data: TTL input signals representing the first 4 bit data/control symbol. ARD7 is the most significant bit and ARD4 is the least significant bit of the first symbol.
ARD3 ARD2 ARD1 ARD0	40 42 44 46	I	PHY Port A Request Data: TTL input signals representing the second 4-bit data/control symbol. ARD3 is the most significant bit and ARD0 is the least significant bit of the second symbol.

6.0 Pin Descriptions (Continued)

PHY PORT INTERFACE (Continued)

Symbol	Pin No.	I/O	Description
BIP	109	O	PHY Port B Indicate Parity: A TTL output signal representing odd parity for the 10-bit wide Port B Indicate signals (BIP, BIC, and BID<7:0>).
BIC	107	O	PHY Port B Indicate Control: A TTL output signal indicating that the two 4-bit symbols (BID<7:4> and BID<3:0>) are either control symbols (BIC = 1) or data symbols (BIC = 0).
BID7 BID6 BID5 BID4	105 103 100 97	O	PHY Port B Indicate Data: TTL output signals representing the first 4-bit data/control symbol. BID7 is the most significant bit and BID4 is the least significant bit of the first symbol.
BID3 BID2 BID1 BID0	94 92 90 88	O	PHY Port B Indicate Data: TTL output signals representing the second 4-bit data/control symbol. BID3 is the most significant bit and BID0 is the least significant bit of the second symbol.
BRP	110	I	PHY Port B Request Parity: A TTL input signal representing odd parity for the 10-bit wide Port B Request signals (BRP, BRC, and BRD<7:0>).
BRC	108	I	PHY Port B Request Control: A TTL input signal indicating that the two 4-bit symbols (BRD<7:4> and BRD<3:0>) are either control symbols (BRC = 1) or data symbols (BRC = 0).
BRD7 BRD6 BRD5 BRD4	106 104 101 98	I	PHY Port B Request Data: TTL input signals representing the first 4-bit data/control symbol. BRD7 is the most significant bit and BRD4 is the least significant bit of the first symbol.
BRD3 BRD2 BRD1 BRD0	95 93 91 89	I	PHY Port B Request Data: TTL input signals representing the second 4-bit data/control symbol. BRD3 is the most significant bit and BRD0 is the least significant bit of the second symbol.

6.0 Pin Descriptions (Continued)

CONTROL BUS INTERFACE

The Control Bus Interface consists of I/O signals used to connect the PLAYER device to Station Management (SMT).

The Control Bus is an asynchronous interface between the PLAYER device and a general purpose microprocessor. It provides access to 32 8-bit internal registers.

Refer to *Figure 22*, Control Bus Timing Diagram, for further information.

Symbol	Pin No.	I/O	Description
\overline{CE}	113	I	Chip Enable: An active-low, TTL, input signal which enables the Control Bus port for a read or write cycle. R/\overline{W} , $CBA <4:0 >$, CBP , and $CBD <7:0 >$ must be valid at the time \overline{CE} is low.
R/\overline{W}	112	I	Read/ ~ Write: A TTL input signal which indicates a read Control Bus cycle ($R/\overline{W} = 1$), or a write Control Bus cycle ($R/\overline{W} = 0$). This signal must be valid when \overline{CE} is low and held valid until \overline{ACK} becomes low.
\overline{ACK}	123	O	~ Acknowledge: An active low, TTL, open drain output signal which indicates the completion of a read or write cycle. During a read cycle, $CBD <7:0 >$ are valid as long as \overline{ACK} is low ($\overline{ACK} = 0$). During a write cycle, a microprocessor must hold $CBD <7:0 >$ valid until \overline{ACK} becomes low. Once \overline{ACK} is low, it will remain low as long as \overline{CE} remains low ($\overline{CE} = 0$).
\overline{INT}	114	O	~ Interrupt: An active low, open drain, TTL, output signal indicating that an interrupt condition has occurred. The Interrupt Condition Register (ICR) should be read in order to determine the source of the interrupt. Interrupts can be masked through the use of the Interrupt Condition Mask Register (ICMR)
CBA4 CBA3 CBA2 CBA1 CBA0	128 127 126 125 124	I	Control Bus Address: TTL input signals used to select the address of the register to be read or written. CBA4 is the most significant bit and CBA0 is the least significant bit of the address signals. These signals must be valid when \overline{CE} is low and held valid until \overline{ACK} becomes low.
CBPE	21	I	Control Bus Parity Enable: A TTL input signal which, during write cycles, will enable or disable the Control Bus parity checker. Note that the Control Bus will always generate parity during read cycles, regardless of the state of this signal.
CBP	10	I/O	Control Bus Parity: A bidirectional, TTL signal representing odd parity for the Control Bus data ($CBD <7:0 >$). During a read cycle, the signal is held valid by the PLAYER device as long as \overline{ACK} is low. During a write cycle, the signal must be valid when \overline{CE} is low, and must be held valid until \overline{ACK} becomes low. If incorrect parity is used during a write cycle, the PLAYER device will inhibit the write cycle and set the Control Bus Data Parity Error (CPE) bit in the Interrupt Condition Register (ICR).
CBD7 CBD6 CBD5 CBD4 CBD3 CBD2 CBD1 CBD0	9 8 6 5 3 2 131 130	I/O	Control Bus Data: Bidirectional, TTL signals containing the data to be read from or written to a register. During a read cycle, the signal is held valid by the PLAYER device as long as \overline{ACK} is low. During a write cycle, the signal must be valid when \overline{CE} is low, and must be held valid until \overline{ACK} becomes low.

6.0 Pin Descriptions (Continued)

CLOCK INTERFACE

The Clock Interface consists of 12.5 MHz and 125 MHz clocks used by the PLAYER device. The clocks are generated by either the Clock Distribution Device or Clock Recovery Device.

Symbol	Pin No.	I/O	Description
LBC	87	I	Local Byte Clock: A TTL, 12.5 MHz, 50% duty cycle, input clock from the Clock Distribution Device. The Local Byte Clock is used by the PLAYER device's internal CMOS logic and to latch incoming/outgoing data of the Control Bus Interface, Port A Interface, Port B Interface, and other miscellaneous I/Os.
RXC+ RXC-	60 61	I	Receive Bit Clock: Differential, 100k ECL, 125 MHz clock input signals from the Clock Recovery Device. The Receive Bit Clock is used by the Serial Interface to latch the Receive Data (RXD \pm).
TXC+ TXC-	72 73	I	Transmit Bit Clock: Differential, 100k ECL, 125 MHz clock input signals from the Clock Distribution Device. The Transmit Bit Clock is used by the Serial Interface to latch the Transmit Data (TXD \pm).
TBC+ TBC-	75 76	I	Transmit Byte Clock: Differential, 100k ECL, 12.5 MHz clock input signals from the Clock Distribution Device. The Transmit Byte Clock is used by the PLAYER device's internal Shift Register Block.

6.0 Pin Descriptions (Continued)

MISCELLANEOUS INTERFACE

The Miscellaneous Interface consists of a reset signal, user definable sense signals, user definable enable signals, Cascaded PLAYER device's synchronization signals, ground signals, and power signals.

Symbol	Pin No.	I/O	Description
$\overline{\text{RST}}$	111	I	<p>Reset: An active low, TTL, input signal which clears all registers. The signal must be kept asserted for a minimum of 160 ns.</p> <p>Once the $\overline{\text{RST}}$ signal is asserted, the PLAYER device should be allowed 960 ns to reset internal logic. Note that bit zero of the Mode Register will be set to zero (i.e. Stop Mode). See Section 4.2, Stop Mode of Operation for more information.</p>
SP0	22	I	<p>User Definable Sense Pin 0: A TTL input signal from a user defined source. Bit zero (Sense Bit 0) of the User Definable Register (UDR) will be set to one if the signal is asserted for a minimum of 160 ns.</p> <p>Once the asserted signal is latched, Sense Bit 0 can only be cleared through the Control Bus Interface, even if the signal is deasserted. This ensures that the Control Bus Interface will record the source of events which can cause interrupts.</p>
SP1	23	I	<p>User Definable Sense Pin 1: A TTL input signal from a user defined source. Bit one (Sense Bit 1) of the User Definable Register (UDR) will be set to one if the signal is asserted for a minimum of 160 ns.</p> <p>Once the asserted signal is latched, Sense Bit 0 can only be cleared through the Control Bus Interface, even if the signal is deasserted. This ensures that the Control Bus Interface will record the source of events which can cause interrupts.</p>
EP0	11	O	<p>User Definable Enable Pin 0: A TTL output signal allowing control of external logic through the Control Bus Interface. EP0 is asserted/deasserted through bit two (Enable Bit 0) of the User Definable Register (UDR). When Enable Bit 0 is set to zero, EP0 is deasserted. When Enable Bit 0 is set to one, EP0 is asserted.</p>
EP1	12	O	<p>User Definable Enable Pin 1: A TTL output signal allowing control of external logic through the Control Bus Interface. EP1 is asserted/deasserted through bit two (Enable Bit 1) of the User Definable Register (UDR). When Enable Bit 1 is set to zero, EP1 is deasserted. When Enable Bit 1 is set to one, EP1 is asserted.</p>
CS	47	I	<p>Cascade Start: A TTL input signal used to synchronize cascaded PLAYER devices in point-to-point applications.</p> <p>The signal is asserted when all of the cascaded PLAYER devices have the Cascade Mode (CM) bit of the Mode Register (MR) set to one, and all of the Cascade Ready (CR) pins of the cascaded PLAYER devices have been released.</p> <p>For further information, refer to Section 4.4, Cascade Mode of Operation.</p>
CR	48	O	<p>Cascade Ready: An Open Drain output signal used to synchronize cascaded PLAYER devices in point-to-point applications.</p> <p>The signal is released when all the cascaded PLAYER devices have the Cascade Mode (CM) bit of the Mode Register (MR) set to one and a JK symbol pair has been received.</p> <p>For further information, refer to section 4.4, Cascade Mode of Operation.</p>

6.0 Pin Descriptions (Continued)

POWER AND GROUND

All power pins should be connected to a single 5V power supply. All ground pins should be connected to a common 0V ground supply.

Symbol	Pin No.	I/O	Description
GND	1		Ground: Power supply return for internal CMOS logic.
GND	4		Ground: Power supply return for Control Bus Interface CMOS I/Os.
V _{CC}	7		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for Control Bus Interface CMOS I/Os.
GND	20		Ground: Power supply return for internal CMOS logic.
GND	32		Ground: Power supply return for Port A Interface CMOS I/Os.
V _{CC}	35		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for the Port A Interface CMOS I/Os.
GND	38		Ground: Power supply return for internal CMOS logic.
V _{CC}	62		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for internal ECL logic.
GND	65		Ground: Power supply return for internal ECL logic.
V _{CC}	68		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for the Serial Interface ECL I/Os.
GND	71		Ground: Power supply return for internal ECL logic.
V _{CC}	74		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for internal ECL logic.
GND	96		Ground: Power supply return for internal CMOS logic.
V _{CC}	99		Power: Positive 5V power supply ($\pm 5\%$ relative to ground) for Port B Interface CMOS I/Os.
GND	102		Ground: Power supply return for Port B Interface CMOS I/Os.
V _{CC}	129		Power: 5V power supply ($\pm 5\%$ relative to ground) for internal CMOS logic.
GND	132		Ground: Power supply return for internal CMOS logic.

6.0 Pin Descriptions (Continued)

NO CONNECT PINS

Symbol	Pin No.	I/O	Description
N/C	13		No Connect: Not used by the PLAYER device
N/C	14		No Connect: Not used by the PLAYER device
N/C	15		No Connect: Not used by the PLAYER device
N/C	16		No Connect: Not used by the PLAYER device
N/C	17		No Connect: Not used by the PLAYER device
N/C	18		No Connect: Not used by the PLAYER device
N/C	19		No Connect: Not used by the PLAYER device
N/C	49		No Connect: Not used by the PLAYER device
N/C	50		No Connect: Not used by the PLAYER device
N/C	51		No Connect: Not used by the PLAYER device
N/C	52		No Connect: Not used by the PLAYER device
N/C	53		No Connect: Not used by the PLAYER device
N/C	54		No Connect: Not used by the PLAYER device
N/C	55		No Connect: Not used by the PLAYER device
N/C	56		No Connect: Not used by the PLAYER device
N/C	78		No Connect: Not used by the PLAYER device
N/C	79		No Connect: Not used by the PLAYER device
N/C	80		No Connect: Not used by the PLAYER device
N/C	81		No Connect: Not used by the PLAYER device
N/C	82		No Connect: Not used by the PLAYER device
N/C	83		No Connect: Not used by the PLAYER device
N/C	84		No Connect: Not used by the PLAYER device
N/C	85		No Connect: Not used by the PLAYER device
N/C	115		No Connect: Not used by the PLAYER device
N/C	116		No Connect: Not used by the PLAYER device
N/C	117		No Connect: Not used by the PLAYER device
N/C	118		No Connect: Not used by the PLAYER device
N/C	119		No Connect: Not used by the PLAYER device
N/C	120		No Connect: Not used by the PLAYER device
N/C	121		No Connect: Not used by the PLAYER device
N/C	122		No Connect: Not used by the PLAYER device

7.0 Electrical Characteristics

7.1 ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage		-0.5		7.0	V
DC_{IN}	Input Voltage		-0.5		$V_{CC} + 0.5$	V
DC_{OUT}	Output Voltage		-0.5		$V_{CC} + 0.5$	V
	Storage Temperature		-65		150	°C

7.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage		4.75		5.25	V
T_A	Operating Temperature		0		70	°C

7.3 DC ELECTRICAL CHARACTERISTICS

The DC characteristics are over the operating range, unless otherwise specified.

DC electrical characteristics for the TTL, TRI-STATE output signals of PHY, Port Interfaces, and CBUS Interface.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{OZ1}	TRI-STATE Leakage (CBP & CBD7-0)	$V_{OUT} = V_{CC}$			10	μA
I_{OZ2}	TRI-STATE Leakage (CBP & CBD7-0)	$V_{OUT} = V_{GND}$			-10	μA
I_{OZ3}	TRI-STATE Leakage (AID & BID)	$V_{OUT} = V_{CC}$ (Note 1)			60	μA
I_{OZ4}	TRI-STATE Leakage (AID & BID)	$V_{OUT} = GND$			-500	μA

Note 1: Output buffer has a p-channel pullup device.

DC electrical characteristics for all TTL input signals and the following TTL output signals: External Loopback (ELB), Fiber Optic Transmitter Enable (TXE), Enable Pin 0 (EP0), and Enable Pin 1 (EP1).

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OH}	Output High Voltage	$I_{OH} = -2 \text{ mA}$	$V_{CC} - 0.5$			V
V_{OL}	Output Low Voltage	$I_{OL} = 4 \text{ mA}$			0.5	V
V_{IH}	Input High Voltage		2.0			V
V_{IL}	Input Low Voltage				0.8	V
V_{IC}	Input Clamp Voltage	$I_{IN} = -18 \text{ mA}$			-1.5	V
I_{IL}	Input Low Current	$V_{IN} = GND$			-10	μA
I_{IH}	Input High Current	$V_{IN} = V_{CC}$			+10	μA

7.0 Electrical Characteristics (Continued)

DC electrical characteristics for all Open Drain output signals (\overline{INT} , \overline{ACK} and CR).

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OL}	Output Low Voltage	$I_{OL} = 8 \text{ mA}$			0.5	V
I_{OZ}	TRI-STATE Leakage	$V_{OUT} = V_{CC}$			10	μA

DC electrical characteristics for all 100k ECL input and output signals.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OH}	Output High Voltage	$V_{IN} = V_{IH} (\text{max})$	$V_{CC} - 1.025$		$V_{CC} - 0.880$	V
V_{OL}	Output Low Voltage	$V_{IN} = V_{IL} (\text{min})$	$V_{CC} - 1.810$		$V_{CC} - 1.620$	V
V_{IH}	Input High Voltage		$V_{CC} - 1.165$		$V_{CC} - 0.880$	V
V_{IL}	Input Low Voltage		$V_{CC} - 1.810$		$V_{CC} - 1.475$	V
I_L	Input Low Current	$V_{IN} = \text{GND}$			-10	μA
I_H	Input High Current	$V_{IN} = V_{CC}$			100	μA

Supply Current electrical characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{CC}	Total Supply Current	LBC = 12.5 MHz TXC = 125 MHz			440*	mA

***Note:** The PLAYER device has two pairs of differential ECL outputs, therefore 60 mA of the total supply current is actually consumed by external termination resistors and the maximum current consumed by the PLAYER device alone is only 380 mA. The ECL termination current is calculated as follows:

$$V_{OH_max} = V_{CC} - 0.88\text{V}$$

$$V_{OL_max} = V_{CC} - 1.62\text{V}$$

Since the outputs are differential, the average output level is $V_{CC} - 1.25\text{V}$. The test load per output is 50Ω at $V_{CC} - 2\text{V}$, therefore the external load current through the 50Ω resistor is:

$$\begin{aligned} I_{LOAD} &= [(V_{CC} - 1.25) - (V_{CC} - 2)]/50 \\ &= 0.015\text{A} \\ &= 15 \text{ mA} \end{aligned}$$

As result, two pairs of ECL outputs consume 60 mA.

7.0 Electrical Characteristics (Continued)

7.4 AC ELECTRICAL CHARACTERISTICS

The AC Electrical characteristics are over the operating range, unless otherwise specified.

AC Characteristics for the Control Bus Interface

Symbol	Parameter	Min	Max	Units
T1	\overline{CE} Setup to LBC	5		ns
T2	LBC Period	80		ns
T3	LBC to \overline{ACK} Low		45	ns
T4	\overline{CE} Low to \overline{ACK} Low	290	540	ns
T5	LBC Low to CBD(7-0) and CBP Valid		60	ns
T6	LBC to CBD(7-0) and CBP Active		60	ns
T7	\overline{CE} Low to CBD(7-0) and CBP Active	225	475	ns
T8	\overline{CE} Low to CBD(7-0) and CBP Valid	265	515	ns
T9	LBC Pulse Width High	35	45	ns
T10	LBC Pulse Width Low	35	45	ns
T11	\overline{CE} High to \overline{ACK} High		45	ns
T12	R/ \overline{W} , CBA(7-0), CBD(7-0) and CBP Set up to \overline{CE} Low	5		ns
T13	\overline{CE} High to R/ \overline{W} , CBA(7-0), CBD(7-0) and CBP Hold Time	0		ns
T14a	R/ \overline{W} to LBC Setup Time	0		ns
T14b	CBA to LBC Setup Time	10		ns
T14c	CBD and CBP to LBC Setup Time	0		ns
T15	\overline{ACK} Low to \overline{CE} High Lead Time	0		ns
T16	\overline{CE} Minimum Pulse Width High	20		ns
T17	\overline{CE} High to CBD(7-0) and CBP TRI-STATE		55	ns
T18	\overline{ACK} High to \overline{CE} Low	0		ns
T19	CBD(7-0) Valid to \overline{ACK} Low Setup	20		ns
T20a	LBC to R/ \overline{W} Hold Time	10		ns
T20b	LBC to CBA Hold Time	10		ns
T20c	LBC to CBD and CBP Hold Time	20		ns
T21	LBC to \overline{INT} Low		55	ns
T22	LBC to \overline{INT} High		60	ns

Asynchronous Definitions

T4 (min)	$T1 + (3 * T2) + T3$
T4 (max)	$T1 + (4 * T2) + T3$
T7 (min)	$T1 + (2 * T2) + T6$
T7 (max)	$T1 + (3 * T2) + T6$
T8 (min)	$T1 + (2 * T2) + T9 + T5$
T8 (max)	$T1 + (3 * T2) + T9 + T5$

Note: Min/Max numbers are based on T2 = 80 ns and T9 = T10 = 40ns.

7.0 Electrical Characteristics (Continued)

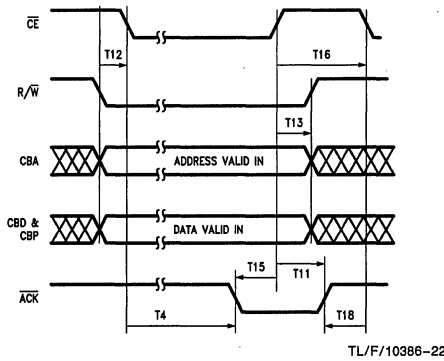


FIGURE 7-1. Control Bus Write Cycle Timing

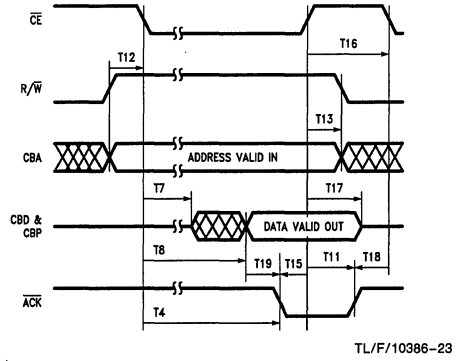


FIGURE 7-2. Control Bus Read Cycle Timing

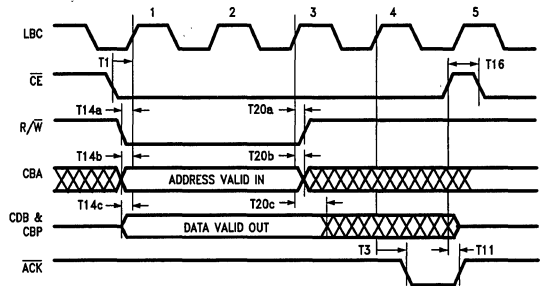


FIGURE 7-3. Control Bus Synchronous Write Cycle Timing

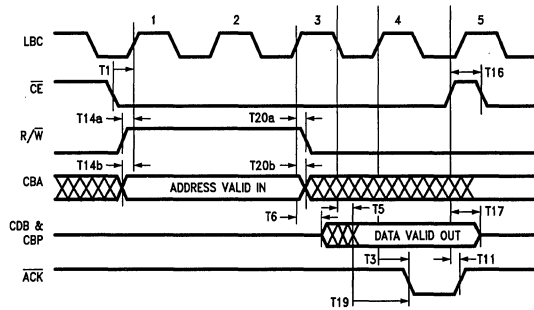


FIGURE 7-4. Control Bus Synchronous Read Cycle Timing

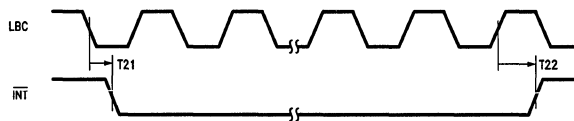


FIGURE 7-5. Control Bus Interrupt Timing

7.0 Electrical Characteristics (Continued)

AC Characteristics for the Clock Signals

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T23	TBC to TXC Hold Time	(Note 1)	2			ns
T24	TBC to TXC Setup Time	(Note 1)	2.5			ns
T25	TBC to LBC Skew		10		22	ns
T26	RXC Duty Cycle	(Note 1)	3.0		5.0	ns
T27	TXC Duty Cycle	(Note 1)	3.5		4.5	ns
T28	TBC Duty Cycle		37		43	ns
T29	LBC Duty Cycle		35		45	ns

Note 1: RXC duty cycle, TXC duty cycle, and TBC to TXC setup time are not tested, but are assured by correlation with characterization data.

Note 2: When PLAYER is used in FDDI applications, TBC and LBC periods will be 80 ns and RXC and TXC periods will be 8 ns.

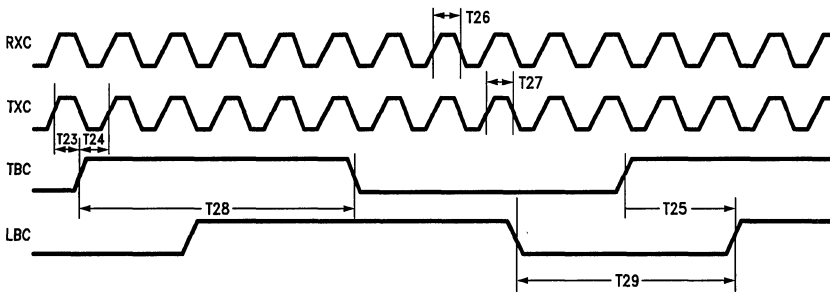


FIGURE 7-6. Clock Signals

TL/F/10386-36

7.0 Electrical Characteristics (Continued)

AC Characteristics for PHY Port Interfaces

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T30	LBC to Indicate Data Changes from TRI-STATE to Data Valid				70	ns
T31	LBC to Indicate Data Changes from Active to TRI-STATE				70	ns
T32	LBC to Indicate Data Sustain		7			ns
T33	LBC to Valid Indicate Data				45	ns
T34	Request Data to LBC Setup Time		15			ns
T35	Request Data to LBC Hold Time		5			ns

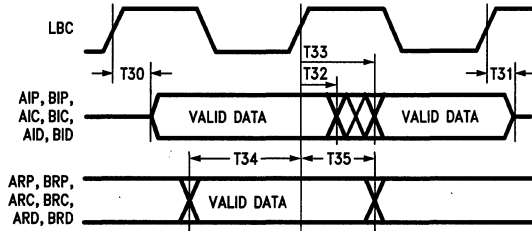


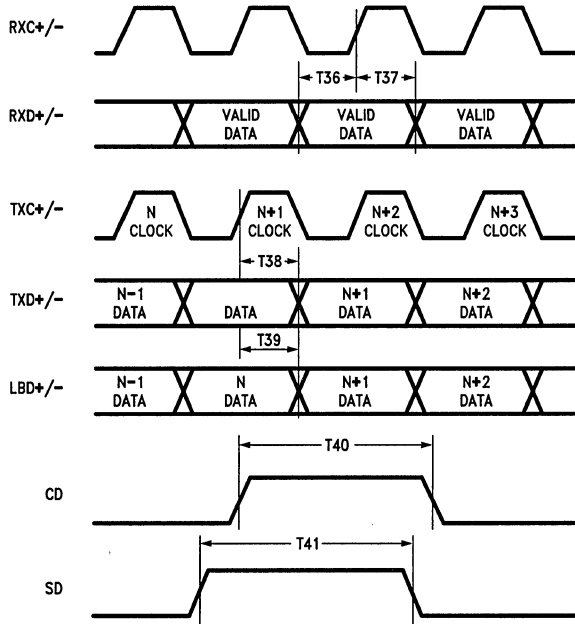
FIGURE 7-7. PHY Port Interface Timing

TL/F/10386-37

7.0 Electrical Characteristics (Continued)

AC Characteristics for the Serial Interface

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T36	RXD to RXC Setup Time		2			ns
T37	RXD to RXC Hold Time		2			ns
T38	TXC to TXD Change Time				8	ns
T39	TXC to LBD Change Time				8	ns
T40	CD Min Pulse Width		120			ns
T41	SD Min Pulse Width		120			ns

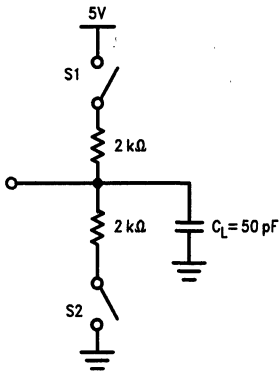


TL/F/10386-38

FIGURE 7-8. Serial Interface Timing

7.0 Electrical Characteristics (Continued)

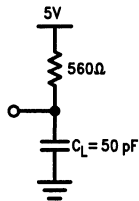
7.5 AC TEST CIRCUITS



TL/F/10386-26

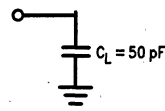
Note: S₁ is closed for T_{PZL} and T_{PLZ}
 S₂ is closed for T_{PZH} and T_{PHZ}
 S₁ and S₂ are open otherwise

FIGURE 7-9. Switching Test Circuit for All TRI-STATE Output Signals



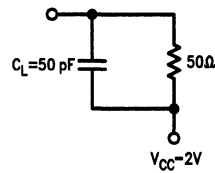
TL/F/10386-29

FIGURE 7-11. Switching Test Circuit for All Open Drain Output Signals (INT, ACK and CR)



TL/F/10386-28

FIGURE 7-10. Switching Test Circuit for All TTL Output Signals

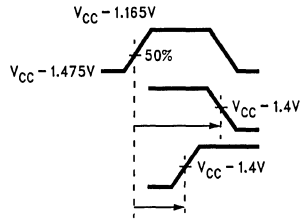


TL/F/10386-30

Note: C_L = 30 pF includes scope and all stray capacitance without device in test fixture

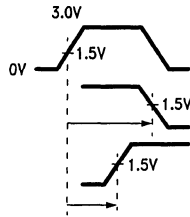
FIGURE 7-12. Switching Test Circuit for All ECL Input and Output Signals

Test Waveforms



TL/F/10386-39

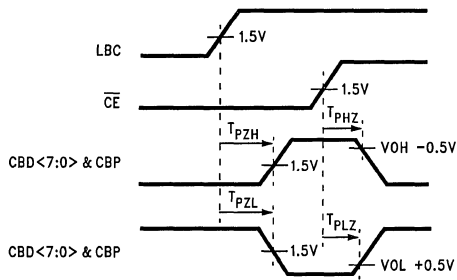
FIGURE 7-13. ECL Output Test Waveform



TL/F/10386-40

Note: All CMOS inputs and outputs are TTL compatible

FIGURE 7-14. TTL Output Test Waveform



TL/F/10386-41

FIGURE 7-15. TRI-STATE Output Test Waveform

8.0 Detailed Descriptions

This section describes in detail several functions that had been discussed previously in Section 3.0, Functional Descriptions.

8.1 FRAMING HOLD RULES

DETECTING JK

The JK symbol pair can be used to detect the beginning of a frame during Active Line State (ALS) and Idle Line State (ILS).

While the Line State Detector is in the Idle Line State the PLAYER device "reframes" upon detecting a JK symbol pair and enters the Active Line State.

During Active Line State, acceptance of a JK symbol (reframing) is allowed on any on-boundary JK which is detected at least 1.5 byte times after the previous JK.

During Active Line State, once reframed on a JK, the subsequent off-boundary JK is ignored, even if it is detected beyond 1.5 byte times after the previous JK.

During Active Line State, an Idle or Ending Delimiter (T) symbol will allow reframing on any subsequent JK, if a JK is detected at least 1.5 bytes times after the previous JK.

DETECTING HALT-HALT & HALT-QUIET

During Idle Line State, the detection of a Halt-Halt, or Halt-Quiet symbol pair will still allow the reframing of any subsequent on-boundary JK.

Once a JK is detected during Active Line State, off-boundary Halt-Halt, or Halt-Quiet symbol pairs are ignored until the Elasticity Buffer (EB) has an opportunity to recenter. They are treated as violations.

After recentering on a Halt-Halt, or Halt-Quiet symbol pair, all off-boundary Halt-Halt or Halt-Quiet symbol pairs are ignored until the EB has a chance to recenter during a line state other than Active Line State (which may be as long as 2.8 byte times).

8.2 NOISE EVENTS

A Noise Event is defined as follows:

A noise event is a noise byte, a byte of data which is not in line with the current line state, indicating error or corruption.

$$\text{Noise Event} = [\text{SD} \bullet \sim \text{CD}] + \\ [\text{SD} \bullet \text{CD} \bullet \text{PI} \bullet \sim (\text{II} + \text{JK} + \text{AB})] + \\ [\text{SD} \bullet \text{CD} \bullet \sim \text{PI} \bullet (\text{PB} = \text{II}) \bullet \text{AB}]$$

Where:

- = Logical AND
- + = Logical OR
- ~ = Logical NOT
- SD = Signal Detect
- CD = Clock Detect
- PB = Previous Byte
- PLS = Previous Line State
- PI = PHY Invalid = HLS + QLS + MLS + NLS + {ULS • [PLS = (ALS + ILS)]}
- ILS = Idle Line State
- ALS = Active Line State
- ULS = Unknown Line State
- HLS = Halt Line State
- QLS = Quiet Line State
- MLS = Master Line State
- NLS = Noise Line State
- ULS = Unknown Line State
- I = Idle symbol
- J = First symbol of start delimiter
- K = Second symbol of start delimiter
- R = Reset symbol
- S = Set symbol
- T = End delimiter
- A = n + R + S + T
- B = n + R + S + T + I
- n = Any data symbol

8.0 Detailed Descriptions (Continued)

8.3 LINK ERRORS

A Link Error is defined as follows:

$$\text{Link Error Event} = [\text{ALS} \bullet (\text{I} \sim \text{I} + \text{xV} + \text{Vx} + \text{H} \sim \text{H})] + [\text{ALS} \bullet \sim \text{SD}] + [\text{ILS} \bullet \sim (\text{II} + \text{JK})] + [\text{ILS} \bullet \sim \text{SD}] + [\text{ULS} \bullet (\text{PLS} = \text{ALS}) \bullet \text{Link_Error_Flag} \bullet \sim \text{SB} \bullet \sim (\text{HH} + \text{HI} + \text{II} + \text{JK})]$$

$$\text{Set Link_Error_Flag} = [\text{ALS} \bullet (\text{HH} + \text{NH} + \text{RH} + \text{SH} + \text{TH})]$$

$$\text{Clear Link_Error_Flag} = [\text{ALS} \bullet \text{JK}] + [\text{ILS} \bullet \text{JK}] + [\text{ULS} \bullet (\text{PLS} = \text{ALS}) \bullet \text{Link_Error_Flag} \bullet \sim \text{SB} \bullet \sim (\text{HH} + \text{HI} + \text{II} + \text{JK})]$$

Where:

- ~ = Logical NOT
- + = Logical OR
- = Logical AND

ILS = Idle Line State

ALS = Active Line State

ULS = Unknown Line State

x = Any symbol

I = Idle symbol

H = Halt symbol

J = First Symbol of start delimiter

K = Second symbol of start delimiter

V = Violation symbol

R = Reset symbol

S = Set symbol

T = End delimiter symbol

N = Data symbol converted to 0000 by the PLAYER device Receiver Block in symbol pairs that contain a data and a control symbol

PLS = Previous Line State

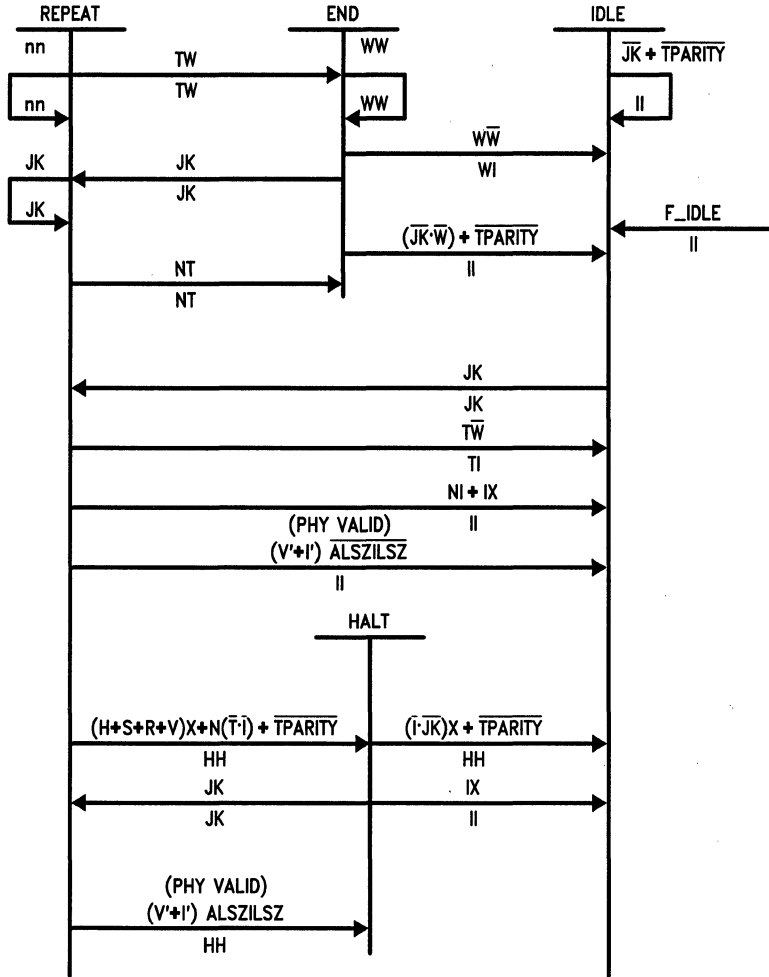
SD = Signal Detect

SB = Stuff Byte: Byte inserted by EB before a JK symbol pair for recentering or due to off-axis JK

8.0 Detailed Description (Continued)

8.4 REPEAT FILTER

The repeat filter prevents the propagation of code violations to the downstream station.



TL/F/10386-31

Note: Inputs to the Repeat Filter state machine are shown above the transition lines, while outputs from the state machine are shown below the transition lines.

Note: Abbreviations used in the Repeat Filter State Diagram are shown in Table VIII.

FIGURE 8-1. Repeat Filter State Diagram

8.0 Detailed Descriptions (Continued)

TABLE 8-1. Abbreviations used in the Repeat Filter State Diagram

F_IDLE:	Force Idle—True when not in Active Transmit Mode
W:	Represents the symbols R, or S, or T
~TPARITY:	Parity error
nn:	Data symbols (for C = 0 in the PHY-MAC Interface)
N:	Data portion of a control and data symbol mixture
X:	Any symbol (i.e. don't care)
V':	Violation symbols or symbols inserted by the Receiver Block
I':	Idle symbols or symbols inserted by the Receiver Block
ALSZILSZ:	Active Line State or Idle Line State (i.e. PHY Invalid)
~ALSZILSZ:	Not in Active Line State nor in Idle Line State (i.e. PHY Valid)
H:	Halt symbol
R:	Reset symbol
S:	Set symbol
T:	Frame ending delimiter
JK:	Frame start delimiter
I:	Idle symbol (Preamble)
V:	Code violations

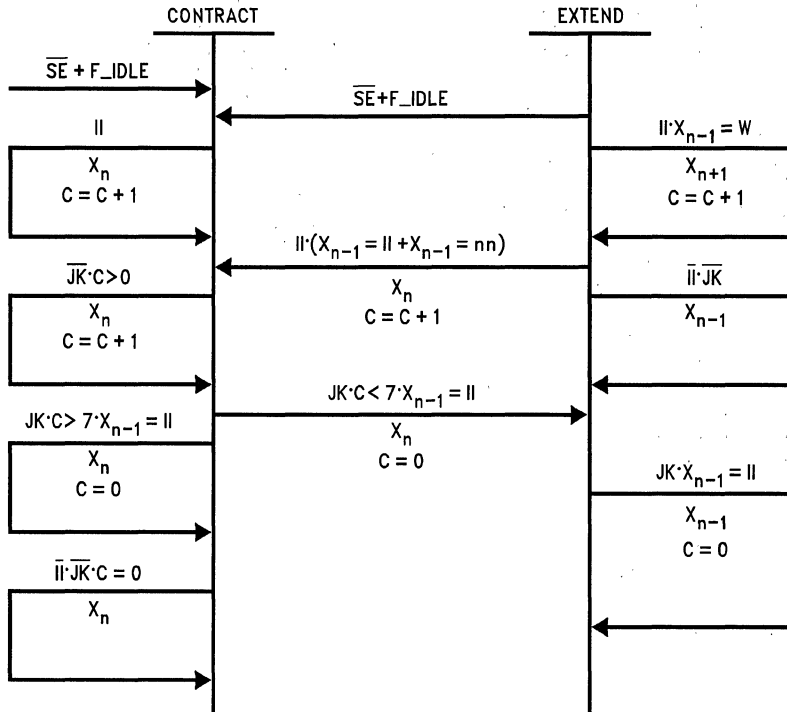
The Repeat Filter complies with the FDDI standard by observing the following:

1. In Repeat State, violations cause transitions to the Halt State and two Halt symbol pairs are transmitted (unless JK or Ix occurs) followed by transition to the Idle State.
2. When Ix is encountered, the Repeat Filter goes to the Idle State, during which Idle symbol pairs are transmitted until a JK is encountered.
3. The Repeat Filter goes to the Repeat State following a JK from any state.

The END State, which is not part of the FDDI standard, allows an R or S prior to a T within a frame to be recognized as a violation. It also allows NT to end a frame as opposed to being treated as a violation.

8.0 Detailed Descriptions (Continued)

8.5 SMOOTHER



Notes:
 SE: Smoother Enable
 C: Preamble Counter
 F_IDLE: Force_Idle (Stop or ATM)
 X_n: Current Byte
 X_{n-1}: Previous Byte
 W: RST

TL/F/10386-32

FIGURE 8-2. Smoother State Diagram

8.0 Detailed Descriptions (Continued)

8.6 NATIONAL BYTE-WIDE CODE FOR PHY-MAC INTERFACE

The PLAYER device outputs the National byte-wide code from its PHY Port Indicate Output to the MAC device. Each National byte-wide code may contain data or control codes or the line state information of the connection. Table 8-2 lists all the possible outputs.

During Active Line State all data and control symbols are being repeated to the PHY Port Indicate Output with the exception of data in data-control mixture bytes. That data symbol is replaced by zero. If only one symbol in a byte is a control symbol, the data symbol will be replaced by 0000 and the whole byte will be presented as control code. Note that the Line State Detector recognizes the incoming data to be in the Active Line State upon reception of the Starting Delimiter (JK symbol pair).

During Idle Line State any non Idle symbols will be reflected as the code l'uLS. If both symbols received during Idle

Line State are Idle symbols, then the Symbol Decoder generates l'kLS as its output. Note that in this case the coded byte is represented in the form Receive State (b7-4), Known/Unknown Bit (b3) and the Last Known Line State (b2-0). The Receive State is 4 bits long and it represents either the PHY Invalid (0011) or the Idle Line State (1011) condition. The Known/Unknown Bit shows if the symbols received match the line state information in the last 3 bits.

During any line state other than Idle Line State or Active Line State, the Symbol Decoder generates the code V'kLS if the incoming symbols match the current line state. The symbol decoder generates V'uLS if the incoming symbols do not match the current line state.

8.0 Detailed Descriptions (Continued)

Table 8-2.

Current Line State	Symbol 1		Symbol 2		National Code	
	Control Bit	Data	Control Bit	Data	Control Bit	Data
ALS	0,	n	0,	n	0,	n-n
ALS	0,	n	1,	C	1,	N-C
ALS	1,	C	0,	n	1,	C-N
ALS	1,	C	1,	C	1,	C-C
ILS	1,	l	1,	l	1,	l'-k-LS
ILS	1,	l	x,	Not l	1,	l'-u-LS
ILS	x,	Not l	1,	l	1,	l'-u-LS
ILS	x,	Not l	x,	Not l	1,	l'-u-LS
Stuff Byte during ILS	x,	x	x,	x	1,	l'-k-ILS
Not ALS and Not ILS	1,	M	1,	M	1,	V'-k-LS
Not ALS and Not ILS	1,	M	x,	Not M	1,	V'-u-LS
Not ALS and Not ILS	x,	Not M	1,	M	1,	V'-u-LS
Not ALS and Not ILS	x,	Not M	x,	Not M	1,	V'-u-LS
Stuff Byte during Not ILS	x,	x	x,	x	1,	V'-k-LS, V'-u-LS or l'-u-ILS
EB Overflow/Underflow					1,	0011 1011
SMT PI Connection (LSU)					1,	0011 1010

Where:

- n = Any data symbol in {0, 1, 2, ... F}
- C = Any control symbol in {V, R, S, T, l, H}
- N = 0000 = Code for data symbol in a data control mixture byte
- l = Idle Symbol
- M = Any symbol that matches the current line state
- l' = 1011 = First symbols of the byte in Idle Line State
- V' = 0011 = PHY Invalid
- LS = Line State
 - ALS = 000
 - ILS = 001
 - NSD = 010
 - MLS = 100
 - HLS = 101
 - QLS = 110
 - NLS = 111
- u = 1 = Indicates symbol received does not match current line state
- k = 0 = Indicates symbol received matches current line state
- x = Don't care

8.0 Detailed Descriptions (Continued)

Example:

Incoming 5B Code	Decoded 4B Code	National Byte-Wide Code (w/o parity)
98765 43210	C3210 C 3210	C 7653 3210
11111 11111 (II)	1 1010 1 1010 (II)	1 1011 0001 (I'-k-ILS)*
11111 11111 (II)	1 1010 1 1010 (II)	1 1011 0001 (I'-k-ILS)
11111 11111 (II)	1 1010 1 1010 (II)	1 1011 0001 (I'-k-ILS)
11000 10001 (JK)	1 1101 1 1101 (JK)	1 1101 1101 (JK Symbols)
----- (xx)	0 ---- 0 ---- (xx)	0 ---- ---- (Data Symbols)
----- (xx)	0 ---- 0 ---- (xx)	0 ---- ---- (Data Symbols)
----- (xx)	0 ---- 0 ---- (xx)	0 ---- ---- (Data Symbols)
(More data ...)		
----- (xx)	0 ---- 0 ---- (xx)	0 ---- ---- (Data Symbols)
----- (xx)	0 ---- 0 ---- (xx)	0 ---- ---- (Data Symbols)
----- (xx)	0 ---- 0 ---- (xx)	0 ---- ---- (Data Symbols)
01101 00111 (TR)	1 0101 1 0110 (TR)	1 0101 0110 (T and R Symbols)
00111 00111 (RR)	1 0110 1 0110 (RR)	1 0110 0110 (Two R Symbols)
11111 11111 (II)	1 1010 1 1010 (II)	1 1010 1010 (Idle Symbols)
11111 11111 (II)	1 1010 1 1010 (II)	1 1010 1010 (Idle Symbols)
11111 11111 (II)	1 1010 1 1010 (II)	1 1011 0001 (I'-k-ILS)
11111 11111 (II)	1 1010 1 1010 (II)	1 1011 0001 (I'-k-ILS)
11111 11111 (II)	1 1010 1 1010 (II)	1 1011 0001 (I'-k-ILS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 1011 1001 (I'-u-ILS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 1011 1001 (I'-u-ILS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 1011 1001 (I'-u-ILS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 1011 1001 (I'-u-ILS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 1011 1001 (I'-u-ILS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 1011 1001 (I'-u-ILS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 1011 1001 (I'-u-ILS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 0011 0101 (V'-k-HLS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 0011 0101 (V'-k-HLS)
00100 00100 (HH)	1 0001 1 0001 (HH)	1 0011 0101 (V'-k-HLS)
11111 11111 (II)	1 1010 1 1010 (II)	1 0011 1101 (V'-u-HLS)
11111 11111 (II)	1 1010 1 1010 (II)	1 1011 0001 (I'-k-ILS)
11111 11111 (II)	1 1010 1 1010 (II)	1 1011 0001 (I'-k-ILS)

*Assume the receiver is in the Idle Line State.



DP83261 BMAC™ Device (FDDI Media Access Controller)

General Description

The DP83261 BMAC device implements the Media Access Control (MAC) protocol for operation in an FDDI token ring. The BMAC device provides a flexible interface to the BSI™ device. The BMAC device offers the capabilities described in the ANSI X3T9.5 MAC Standard and several functional enhancements allowed by the Standard.

The BMAC device transmits, receives, repeats, and strips tokens and frames. It uses a full duplex architecture that allows diagnostic transmission and self testing for error isolation. The duplex architecture also allows full duplex data service on point-to-point connections. Management software is also aided by an array of on chip statistical counters, and the ability to internally generate Claim and Beacon frames without program intervention. A multi-frame streaming interface is provided to the system interface device.

Features

- Full duplex operation with through parity
- Supports all FDDI ring scheduling classes (asynchronous, synchronous, restricted asynchronous, and immediate)
- Supports individual, group, short, long and external addressing
- Generates Beacon, Claim and Void frames without intervention
- Provides extensive ring and station statistics
- Provides extensions for MAC level bridging
- Provides separate management interface
- Uses low power microCMOS

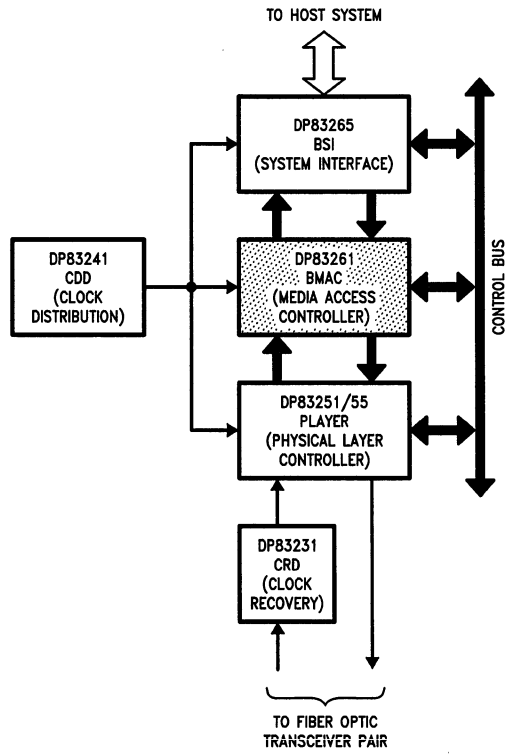


FIGURE 1-1. FDDI Chip Set Block Diagram

TL/F/10387-1

Table of Contents

1.0 FDDI CHIP SET OVERVIEW

2.0 ARCHITECTURAL DESCRIPTION

- 2.1 Ring Engine
- 2.2 Interfaces

3.0 FEATURE OVERVIEW

4.0 FDDI MAC FACILITIES

- 4.1 Symbol Set
- 4.2 Protocol Data Units
- 4.3 Frame Counts
- 4.4 Timers
- 4.5 Ring Scheduling

5.0 FUNCTIONAL DESCRIPTION

- 5.1 Token Handling
- 5.2 Servicing Transmission Requests
- 5.3 Request Service Parameters
- 5.4 Frame Validity Processing
- 5.5 Frame Status Processing
- 5.6 SMT Frame Processing
- 5.7 MAC Frame Processing
- 5.8 Receive Batching Support
- 5.9 Immediate Frame Transmission
- 5.10 Full Duplex Operation
- 5.11 Parity Processing

6.0 CONTROL INFORMATION

- 6.1 Conventions
- 6.2 Access Rules
- 6.3 Operation Registers
- 6.4 Event Registers
- 6.5 MAC Parameters
- 6.6 Timer Thresholds
- 6.7 Event Counters

7.0 SIGNAL DESCRIPTIONS

- 7.1 Control Interface
- 7.2 PHY Interface
- 7.3 MAC Indication Interface
- 7.4 MAC Request Interface
- 7.5 Electrical Interface
- 7.6 Pinout Summary
- 7.7 Pinout Diagram

8.0 ELECTRICAL CHARACTERISTICS

- 8.1 Absolute Maximum Ratings
- 8.2 Recommended Operating Conditions
- 8.3 DC Electrical Characteristics
- 8.4 AC Electrical Characteristics

APPENDIX A. RING ENGINE STATE MACHINES

- A.1 Receiver
- A.2 Transmitter

1.0 FDDI Chip Set Overview

National Semiconductor's DP83200 FDDI chip set consists of five components as shown in *Figure 1-1*. For more information on the other devices of the chip set, consult the appropriate datasheets and application notes.

DP83231 CRD™ Device Clock Recovery Device

The Clock Recovery Device extracts a 125 MHz clock from the incoming bit stream.

Features

- PHY Layer loopback test
- Crystal controlled
- Clock locks in less than 85 μ s

DP83241 CDD™ Device Clock Distribution Device

The Clock Distribution Device generates the clocks required by the FDDI Devices on a board.

Features

- Utilizes a 12.5 MHz crystal or reference
- Generates the 125 MHz, 25 MHz, and 12.5 MHz clock required by the BMAC, PLAYER, and BSI devices
- Generates 5 phases of the 12.5 MHz clock for use by external system logic

DP83251/55 PLAYER™ Device Physical Layer Controller

The PLAYER device implements the Physical Layer (PHY) protocol as defined by the ANSI FDDI PHY X3T9.5 Standard.

Features

- 4B/5B encoders and decoders
- Framing logic
- Elasticity Buffer, Repeat Filter and Smoother
- Line state detector/generator
- Link error detector
- Configuration switch
- Full duplex operation
- Separate management port that is used to configure and control operation

In addition, the DP83255 contains an additional PHY__Data.request and PHY__Data.indicate port required for concentrators and dual attach stations.

DP83261 BMACTM Device Media Access Controller

The BMAC device implements the Timed Token Media Access Control protocol defined by the ANSI X3T9.5 FDDI MAC Standard.

Features

- All of the standard defined ring service options
- Full duplex operation with through parity
- Supports all FDDI Ring Scheduling Classes (Synchronous, Asynchronous, etc.)
- Supports Individual, Group, Short, Long, and External Addressing
- Generates Beacon, Claim, and Void frames internally
- Extensive ring and station statistic gathering
- Extensions for MAC level bridging
- Separate management port that is used to configure and control operation
- Multi-frame streaming interface

DP83265 BSITM Device System Interface

The BSI device implements the interface between the BMAC device and a host system.

Features

- 32-bit wide Address/Data path with byte parity
- Programmable transfer burst sizes of 4 or 8 32-bit words
- Interfaces to low cost DRAMs or directly to system bus
- Provides 2 Output and 3 Input Channels
- Supports Header/Info splitting
- Operates from 12.5 MHz to 25 MHz synchronously with the host system
- Efficient data structures
- Programmable Big or Little Endian alignment
- Full duplex data path allows transmission to self
- Confirmation status batching services
- Receive frame filtering services

2.0 Architectural Description

The BMAC device receives, transmits, and strips or repeats Protocol Data Units (PDUs, i.e., Tokens and Frames) and handles the token management functions required by the timed token protocol in accordance with the FDDI MAC Standard.

The BMAC device is comprised of the Ring Engine (RE) and interfaces to the Control Bus (Control Interface), the PLAYER device (PHY Interface) and a System Interface such as the BSI device (MAC Interface) as shown in *Figure 2-1*.

On transmission, the system interface prepares one or more frames for transmission and requests a service opportunity. Based on the requested service class and requested token type, the Ring Engine waits for a token meeting the requested criteria. When a token is captured, the Ring Engine signals the interface and soon thereafter transmission begins. After traversing the ring, frames are stripped based on the

Source Address. Frames with a Source Address matching one of the station individual addresses are stripped by the Ring Engine. Status is available at the MAC interface for every transmitted frame.

For reception, the Ring Engine sequences through the incoming byte stream, comparing received destination addresses against the station's short or long address. The results of these comparisons are made available at the MAC interface. The System Interface then decides how to handle the frame. In the normal case, a frame with a Destination Address matching one of the station addresses is copied and passed to the system.

The BMAC device utilizes a full duplex, byte-wide (symbol pair) architecture. There are two bytes of delay in the Transmit path, three bytes of delay in Receive and Repeat paths, and two bytes of delay in the Loopback path.

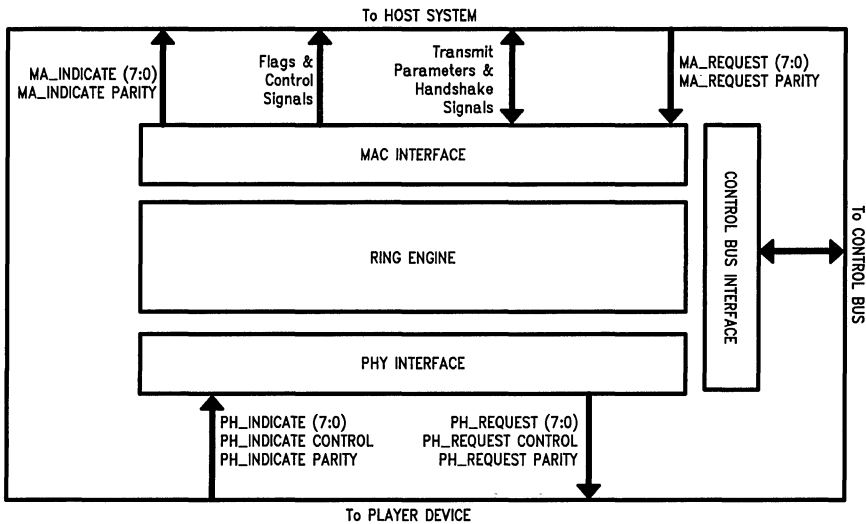


FIGURE 2-1. BMAC Device Interfaces

TL/F/10387-2

2.0 Architectural Description (Continued)

2.1 RING ENGINE

The BMAC device is operated by the Ring Engine which is comprised of four blocks: Receiver, Transmitter, MAC Parameter RAM, and Counters/Timers as shown in *Figure 2-2*.

2.1.1 Receiver

The Receiver Block accepts data from the PLAYER device in the byte stream format (PH_Indicate).

Upon receiving the data, the Receiver Block performs the following functions:

- Determines the beginning and ending of a Protocol Data Unit (PDU)
- Decodes the Frame Control field to determine the PDU type (frame or token)
- Compares the received Destination and Source Addresses with the internal addresses
- Processes data within the frame
- Calculates and checks the Frame Check Sequence at the end of the frame
- Checks the Frame Status field

And finally, the Receiver Block presents the data to the MAC Interface along with the appropriate control signals (MA_Indicate).

2.1.2 Transmitter

The Transmitter Block inserts frames from this station into the ring in accordance with the FDDI Timed Token MAC protocol. It also repeats frames from other stations in the ring. The Transmitter block multiplexes data from the MA_Request Interface and data from the Receiver Block. During Frame Transmission, data from the Request Interface is se-

lected. During Frame Repeating, data from the Receiver Block is selected.

During Frame Transmission, the Transmitter Block performs the following functions:

- Captures a token to gain the right to transmit
- Transmits one or more frames
- Generates the Frame Check Sequence during transmission and appends it at the end of the frame
- Generates the Frame Status field that is transmitted at the end of the frame
- Issues the token at the end of frame transmission

During Frame Repeating, the Transmitter Block performs the following functions:

- Repeats the received frame and modifies the Frame Status field at the end of the frame as specified by the standard

Whether transmitting or repeating frames, the Transmitter Block also performs the following functions:

- Strips the frame(s) that are transmitted by this station
- Generates Idle symbols between frames

Data is presented from the Transmitter Block to the PLAYER device in the byte stream format (PH_Request).

2.1.3 MAC Parameter RAM

The MAC Parameter RAM block is a dual port RAM that contains MAC parameters such as the station's short and long addresses. These parameters are initialized via the Control Interface. Both the Receiver and Transmitter Blocks may access the RAM.

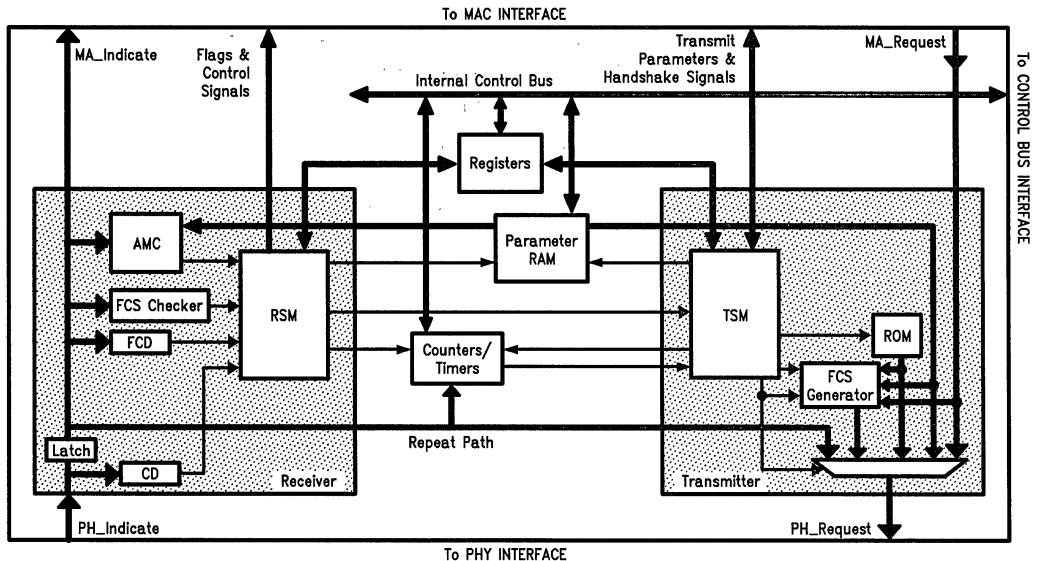


FIGURE 2-2. Ring Engine Overview Block Diagram

TL/F/10387-4

2.0 Architectural Description (Continued)

The Receiver uses these parameters to compare addresses in incoming frames with its addresses stored in the Parameter RAM.

The Transmitter uses the Parameter RAM for generating the Source Address for all frames (except when Source Address Transparency is enabled) and for the Destination Address and Information fields in Claim and Beacon frames.

The MAC Parameter RAM block is described in greater details in Section 6.5.

2.1.4 Counter/Timer

The Counter/Timer block maintains all of the Counters and Timers required by the Standard.

Events which occur too rapidly for software to count, such as the various Frame Counts, are included in the Event Counters. The size of the wrap around counters has been chosen to require minimal software intervention even under marginal operating conditions. Most of the Counters increment in response to events detected by the Receiver. The Counters are readable via the Control Interface.

The Token Rotation and Token Holding Timers which are used to implement the Timed Token Protocol are contained within the Timer Block.

The Counters and Timers are described in detail in Sections 6.6 and 6.7.

2.2 INTERFACES

2.2.1 PHY Interface

The PHY Interface is a synchronous interface that provides an encoded byte stream to the PLAYER device (the PHY Request byte stream), and receives an encoded byte stream from the PLAYER device (the PHY Indication byte stream).

The BMAC device connects to one or two PLAYER devices via the PH_Indicate and PH_Request Interfaces.

Data is transferred from the PLAYER device to the Ring Engine via the PH_Indicate Interface. Data is transferred from the Ring Engine to the PLAYER device via the PH_Request Interface.

The 10-bit byte transferred in both directions across the PH_Indicate and PH_Request interfaces consists of one parity bit (Odd parity), one Control bit, and 8 bits of data. The Control Bit determines if the 8 data bits are a data symbol pair or a control symbol pair.

2.2.2 MAC Interface

The MAC Interface provides the required information and handshakes to allow a system interface (such as the DP83265 BS) to exploit the capabilities of the Ring Engine.

The MAC Interface is synchronous and is divided into separate MAC Request and MAC Indication interfaces.

Data is transferred from the system interface to the Ring Engine via the MAC Request Interface. The MA_Request Interface consists of a parity bit (Odd parity) and byte-wide data along with the transmit parameters and handshake signals. The MAC Request Interface utilizes a handshake that separates token capture from data transmission. A captured token may be held until it is no longer usable. Void frames are automatically generated to allow data interface logic as much time as it needs to prepare a transmission.

Data is transferred from the Ring Engine to the system interface via the MAC Indication Interface. The MA_Indicate Interface consists of a parity bit (Odd parity) and byte-wide data along with Addressing Flags and Frame Sequencing signals. The Addressing Flags give the result of the address comparisons performed by the Ring Engine. These are used to decide whether to continue to copy or to reject frames. The MAC Indication Interface also accepts inputs to determine how to set the control indicators and increment the statistical counters based on external address comparison logic and frame copying logic. Frames may also be stripped based on external comparisons.

2.2.3 Control Bus Interface

The Control Interface implements the interface to the Control Bus by which to initialize, monitor and diagnose the operation of the BMAC device. The Control Interface is an 8-bit asynchronous interface in order to minimize pinout and layout. All information that must be synchronized with the data stream crosses the MAC Interface.

The Control bus is separated completely from the MAC and PHY Interfaces in order to allow independent operation of the processor on the Control Bus. The Control Interface provides the synchronization between the Control Bus and the Ring Engine.

3.0 Feature Overview

The BMAC device implements the standard FDDI MAC protocol. It also provides additional addressing, bridging, and service class functions to allow maximal flexibility in designing an FDDI station.

The BMAC device offers extensive diagnostic features including a number of diagnostic counters, a dedicated interface for control and configuration, and a capability to perform Self Testing. Furthermore, the BMAC device allows the tuning of certain parameters to increase the performance of the network.

3.0 Feature Overview (Continued)

3.1 FDDI MAC SUPPORT

The BMAC device implements the Standard ANSI X3T9.5 FDDI MAC protocol for transmitting, receiving, repeating and stripping frames. Many of the capabilities defined in MAC-2 are included in the BMAC device such as bridging end station support for setting the control indicators, and the statistic counters. The BMAC device provides all of the information necessary to implement the service primitives defined in the standard.

The BMAC device also implements many of the permitted extensions to the FDDI-MAC standard as captured in the FDDI MAC-2 document. These include the extensions for MAC level bridging, Group Addressing support that can be used for SMT, reporting of additional events to aid the ring management processes and enhanced versions of the state machines.

3.2 MAC ADDRESSING SUPPORT

Both long (48-bit) and short (16-bit) addressing are supported simultaneously, for both individual and Group addresses.

Up to 128 contiguous programmable group addresses and up to 15 Fixed Group Addresses plus the universal/broadcast address are recognized. Limited operation with null addresses is supported. An interface to external address matching logic is provided to augment the Ring Engine's addressing capabilities.

3.3 MAC BRIDGING SUPPORT

Several features are provided to aid in Bridging applications.

On the receive side, external address matching logic can be used to examine the PH_Indicate byte stream to decide whether to copy a frame, how to set the control indicators and how to increment the counters.

On the transmit side, transparency options are provided on the Source Address, the most significant bit of the Source Address, and the FCS.

In addition, support for an alternate Void stripping mechanism provides maximal flexibility in the generation of frames.

3.4 MAC SERVICE CLASS SUPPORT

All of the FDDI MAC service classes are supported by the BMAC device. These include the Synchronous, Asynchronous, Restricted Asynchronous, and Immediate service classes.

For Synchronous transmission, one or more frames are transmitted in accordance with the station's synchronous bandwidth allocation.

For Asynchronous transmission, one programmable asynchronous priority threshold is supported in addition to the threshold at the Negotiated Target Token Rotation time.

For Restricted Asynchronous transmission, support is provided to begin, continue and end restricted dialogues.

For Immediate transmissions, support is provided to send frames from either the Data, Beacon or Claim states and either ignore or respond to the received byte stream. After an immediate transmission a token may optionally be issued.

3.5 DIAGNOSTIC COUNTERS

The BMAC device includes a number of diagnostic counters that monitor ring and station performance.

These counters allow measurement of the following:

- Number of frames transmitted and received by the station
- Number of frames copied as well as frames not copied because of insufficient buffering
- Frame error rate of an incoming physical connection to the MAC
- Load on the ring based on the number of tokens received and the ring latency
- Ring latency
- Lost frames

The size of these counters has been selected to keep the frequency of overflow small, even under worst case operating conditions.

3.6 MANAGEMENT SERVICES

The BMAC device provides management services to the Host System via the Control Bus Interface. This interface allows access to internal registers to control and configure the BMAC device.

3.7 RING PARAMETER TUNING

The BMAC device includes settable parameters to allow tuning of the network to increase performance over a large range of network sizes.

The BMAC device supports systems of two stations with little cable between them to ring configurations much larger than the 1000 physical attachments and/or 200 km distance that are specified as the default values in the standard.

The BMAC device also handles frames larger than the 4500 byte default maximum frame size as specified in the Standard.

3.8 MULTI-FRAME STREAMING INTERFACE

The BMAC device provides an interface to support a multi-frame streaming interface. Multiple frames can be transmitted after a token is captured within the limits of the token timer thresholds.

3.9 GENERATES BEACON, CLAIM, AND VOID FRAMES INTERNALLY

For purposes of transient token and ring recovery, no processor intervention is required. The BMAC device automatically generates the appropriate MAC frames.

3.10 SELF TESTING

Because the BMAC device is full duplex, loopback testing is possible before entering the ring and during normal ring operation.

There are several possible loopback paths:

- internal to the BMAC device
- through the PLAYER device(s) using the PLAYER device configuration switch
- through the CRD device.

These paths allow error isolation down to the device level.

The BMAC device also supports through parity.

4.0 FDDI MAC Facilities

4.1 SYMBOL SET

The Ring Engine recognizes and generates a set of symbols. These symbols are used to convey Line States (such as the Idle Line State), Control Sequences (such as the Starting and Ending Delimiters) and Data.

Additional information regarding the symbol set can be found in the FDDI PHY Standard.

The Ring Engine expects that the Starting Delimiter will always be conveyed on an even symbol pair boundary. Following the starting delimiter, data symbols should always come in matched pairs. Similarly the Ending Delimiter should always come in one or more matched symbol pairs.

The symbol pairs conveyed at the PHY Interface are shown in Table 4-1.

4.2 PROTOCOL DATA UNITS

The Ring Engine recognizes and generates two types of Protocol Data Units (PDUs): Tokens and Frames.

The Token is used to control access to the ring. Only the station that has captured the token has the right to transmit new information. The format of a token is shown in Figure 4-1.

SFS		EFS	
PA	SD	FC	ED

FIGURE 4-1. Token Format

Frames are used to pass information between stations. The format of a frame is shown in Figure 4-2 with the field definitions in Table 4-2.

SFS		Protected by PCS					EFS	
PA	SD	FC	DA	SA	INFO	FCS	ED	FS

FIGURE 4-2. Frame Format

TABLE 4-1. Symbol Pair Set

Type	Symbols
Starting Delimiter	JK
Ending Delimiter	TT or TR or TS or nT
Frame Status	RR or RS or SR or SS
Idle	ll or nl
Data Pair	nn

Note : n represents any data symbol (0-F).
 Symbol pairs others than the defined symbols are treated as code violations.
 Section 7.2 has additional information on the symbol pairs generated and interpreted by the Ring Engine.

TABLE 4-2. PDU Fields

Name	Description	Size
SFS	Start of Frame Sequence	
PA	Preamble	8 or More Idle Symbol Pairs
SD	Starting Delimiter	JK Symbol Pair
FC	Frame Control Field	1 Data Symbol Pair
DA	Destination Address	2 or 6 Symbol Pairs
SA	Source Address	2 or 6 Symbol Pairs
INFO	Information Field	
FCS	Frame Check Sequence	4 Symbol Pairs
EFS	End of Frame Sequence	
ED	Ending Delimiter	At Least 1T Symbol for Frames; At Least 2T Symbols for Tokens
FS	Frame Status	3 or More R or S Symbols

4.0 FDDI MAC Facilities (Continued)

4.2.1 PDU Fields

Start of Frame Sequence

The Start of Frame Sequence (SFS) consists of the Preamble (PA) followed by the Starting Delimiter (SD).

The Preamble is a sequence of zero or more Idle symbols that is used to separate the PDUs. The Ring Engine Receiver can process and repeat a frame or token with no preamble. The Ring Engine Transmitter generates frames with at least 8 bytes of preamble. The Ring Engine Transmitter also guarantees that valid FDDI frames will never be transmitted with more than 40 bytes of preamble.

The Starting Delimiter is used to indicate the start of a new PDU. The Starting Delimiter is the JK symbol pair.

The Ring Engine expects the Starting Delimiter to be conveyed across the PH_Indication Interface as a single byte. Similarly, the Ring Engine only generates Starting Delimiters aligned to the byte boundary.

Frame Control

The Frame Control (FC) field is used to discriminate PDUs. For tokens, the FC field identifies Restricted and Non-restricted tokens. For frames, the FC field identifies the frame types and format and how the frame is to be processed.

The one byte FC field is formatted as shown in *Figure 4-3*.

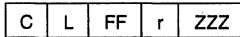


FIGURE 4-3. Frame Control Field

The C (Class) bit specifies the MAC Service Class as Asynchronous (C = 0) or Synchronous (C = 1).

The L (Length) bit specifies the length of the MAC Address as Short (L = 0) or Long (L = 1). A Short Address is a 16-bit address. A Long Address is a 48-bit address.

The FF (Format) bits specify the PDU types as shown in Table 4-3.

The r (Reserved) bit is currently not specified and should always be transmitted as Zero (Exception: SMT NSA Frames).

The ZZZ (Control) bits are used in conjunction with the C and FF bits to specify the type of PDUs. These bits may be used to affect protocol processing criteria such as the Priority, Protocol Class, Status Handling, etc.

TABLE 4-3. Frame Control Format Bits

FC,FF		PDU Types
0	0	SMT/MAC
0	1	LLC
1	0	Reserved for Implementer
1	1	Reserved for Future Standardization

When the Frame Control Format bits (FC,FF) indicate a SMT or MAC PDU, the frame type is identified as shown in Table 4-4.

TABLE 4-4. MAC/SMT Frames Types

CLFF	rZZZ	PDU Type
1000	0000	Non-Restricted Token
1100	0000	Restricted Token
0L00	0000	Void Frame
0L00	0001 to 1110	SMT Frame
0L00	1111	SMT Next Station Addressing Frame
1L00	0001	Other MAC Frame
1L00	0010	MAC Beacon Frame
1L00	0011	MAC Claim Frame
1L00	0100 to 1111	Other MAC Frame

Destination Address

The Destination Address (DA) field is used to specify the station(s) that should receive and process the frame.

The DA can be an Individual or Group address. This is determined by the Most Significant Bit of the DA (DA,IG). When DA,IG is 0 the DA is an Individual Address, when DA,IG is 1 the DA is a Group Address. The Broadcast/Universal address is a Group Address.

The DA field can be a Long or Short Address. This is determined by the L bit in the FC field (FC,L). If FC,L is 1, the DA is a 48-bit Long Address. If FC,L is 0, the DA is a 16-bit Short Address.

The Ring Engine maintains both a 16-bit Individual Address, My Short Address (MSA) and a 48-bit Individual Address, My Long Address (MLA).

On the receive side, if DA,IG is 0 the incoming DA is compared with MLA (if FC,L = 1) or MSA (if FC,L = 0). If the received DA matches MLA or MSA the frame is intended for this station and the address recognized flag (A_Flag) is set. If DA,IG is 1, the DA is a Group Address and is compared with the set of Group Addresses recognized by the Ring Engine. If a match occurs the address recognized flag (A_Flag) is set. The A_Flag is used by system interface logic as part of the criteria (with FC,L, DA,IG and M_Flag) to determine whether or not to copy the frame. If the A_flag is set, the system interface will normally attempt to copy the frame.

On the transmit side, the DA is provided by the system interface logic as part of the data stream. The length of the address to be transmitted is determined by the L bit of the FC field. (The FC field is also passed in the data stream.) The Destination Address can be an Individual, Group, or Broadcast Address.

Source Address

The Source Address (SA) field is used to specify the address of the station that originally transmitted the frame.

4.0 FDDI MAC Facilities (Continued)

The Source Address has the same length as the Destination Address (i.e., if the DA is a 16-bit Address, the SA is a 16-bit Address; if the DA is a 48-bit Address, the SA is a 48-bit Address).

On the receive side, the incoming SA is compared with either MSA or MLA. If a match occurs between the incoming SA and this station's MLA or MSA, the M_Flag is set. This flag is used to indicate that the frame is recognized as having been transmitted by this station and is stripped. The most significant bit of the SA (SA.IG) is not evaluated in the comparison.

On the transmit side, the station's individual address is transmitted as the SA. Since the SA field is normally used for stripping frames from the ring, the SA stored by the Ring Engine normally replaces the SA from the data stream. The length of the address to be transmitted is determined by the L bit of the FC field. (The FC field is passed in the data stream.) The most significant bit of the SA (SA.IG) is normally transmitted as 0, independent of the value passed through the data stream.

As a transmission option, the SA may also be transmitted transparently from the data stream. When the SA Transparency option is used, an alternate stripping mechanism is necessary to remove these frames from the ring. (The Ring Engine provides a Void Stripping Option. See Section 7.4.2.4 for further information.)

As a separate and independent transmission option, the MSB of the SA may also be transmitted transparently from the data stream. This is useful for end stations participating in the Source Routing protocol.

Information

The Information field (Info) contains the Service Data Unit (SDU). A SDU is the unit of data transfer between peer users of the MAC data service (SMT, LLC, etc). There is no INFO field in a Token.

The INFO field contains zero or more bytes.

On the receive side, the INFO field is checked to ensure that it has at least the minimum length for the frame type and contains an even number of symbols, as required by the Standard.

The first 4 bytes of the INFO field of MAC frames (e.g., MAC Beacon or MAC Claim) are stored in an internal register and compared against the INFO field of the next MAC frame. If the data of the two frames match, the SameInfo signal is generated. This signal may be used to copy MAC frames only when new information is present.

On the transmit side, the Ring Engine does not limit the maximum size of the INFO field, but it does insure that frames are transmitted with a valid DA and SA.

Frame Check Sequence

The Frame Check Sequence (FCS) is a 32-bit Cyclic Redundancy Check that is used to check for data corruption in frames. There is no FCS field in a Token.

On the receive side, the Ring Engine checks the FCS to determine whether the frame is valid or corrupted.

On the transmit side, the FCS field is appended to the end of the INFO field. As a transmission option, appending the FCS to the frame can be inhibited (FCS Transparency).

End of Frame Sequence

The End of Frame Sequence (EFS) always begins with a T symbol and should always contain an even number of symbols. For Tokens an additional T symbol is added. For frames the *Ending Delimiter* (ED) is followed by one or more *Frame Status* Indicators (FS).

The Frame Status (FS) field is used to indicate the status of the frame. The FS field consists of three Indicators: Error Detected (E), Address Recognized (A), and Frame Copied (C). These Indicators are created and modified as specified in the Standard.

For frames transmitted by the Ring Engine, the E, A and C Indicators are appended to all frames and are transmitted as R symbols. No provisions are made to generate additional trailing control indicators.

For frames repeated by the Ring Engine, the E, A and C Indicators are handled as specified in the Standard. Additional trailing control indicators are repeated unmodified provided they are properly aligned. See Section 5.5 for details on Frame Status Processing.

4.2.2 Token Formats

The Ring Engine supports non-restricted and restricted Tokens. See *Figures 4-4 and 4-5*.

SFS	FC	EFS
SD	80	ED

FIGURE 4-4. Non-Restricted Token Format

SFS	FC	ED
SD	C0	ED

FIGURE 4-5. Restricted Token Format

Non-Restricted

A non-restricted token is used for synchronous and non-restricted asynchronous transmissions.

Each time the non-restricted token arrives, a station is permitted to transmit one or more frames in accordance with its synchronous bandwidth allocation regardless of the status of the token (late or early).

Asynchronous transmissions occur only if the token is early (usable token) and the Token Holding Timer has not reached the selected threshold.

Restricted

A restricted token is used for synchronous and restricted asynchronous transmissions only.

A station which initiates the restricted dialogue captures a non-restricted token and releases a restricted token. Stations that participate in the restricted dialogue are allowed to capture the restricted token. A station ends the restricted dialogue by capturing the restricted token and releasing a non-restricted token.

4.2.3 Frame Formats

The Ring Engine supports all of the frame formats permitted by the FDDI standard. All frame types may be created external to the BMAC device and be passed through the MAC Request Interface to the Ring. The BMAC device also has the ability to generate Void, Beacon and Claim frames internally.

4.0 FDDI MAC Facilities (Continued)

Frames Generated Externally

The Ring Engine transmits frames passed to it from the System Interface. The data portion of the frame is created by the System Interface. This begins with the FC field and ends with the last byte of the INFO field. The FC field is passed transparently to the ring. The length bit in the FC field is used to determine the length of the transmitted addresses. The data is passed as a byte stream across the MAC Request Interface as shown in Table 4-5.

Before the frame is transmitted, the Ring Engine inserts the Start of Frame Sequence with at least 8 bytes of Preamble but no more than 40 bytes of Preamble. The starting delimiter is transmitted as a JK symbol pair. The Source Address is normally transmitted by the Ring Engine since it uses the Source Address to strip the frame from the ring. This can be overridden by using the Source Address transparency capability. Similarly, the Frame Check Sequence (4 bytes) is normally transmitted by the Ring Engine. This can be overridden with the FCS transparency capability. With FCS transparency, the FCS is transmitted from the data stream. The End of Frame Sequence is always transmitted by the Ring Engine as TR RR.

Frames transmitted by the Ring Engine must have a valid DA and SA field. If the end of a frame is reached before a valid length is transmitted, the frame will be aborted and a Void frame will be transmitted.

TABLE 4-5. Frame Formats

Field	Size	MA_Request	PH_Request
PA	≥ 8; ≤ 40		Idle Pairs
SD	1		JK
FC	1	FC	FC
DA	2 or 6	DA	DA
SA	2 or 6	SA	MSA, MLA, or SA
INFO	≥ 0	INFO	INFO
FCS	4 if Present	FCS	FCS
ED	1		TR
FS	1		RR

Frames Generated by the Ring Engine

The Ring Engine generates and detects several frames in order to attain and maintain an operational ring.

Void Frames

Void frames are used during normal operation. The Ring Engine generates two types of void frames: regular Void frames and My_Void frames. See Table 4-6.

If short addressing is enabled, Void frames with the short address are transmitted, otherwise Void frames with the long address are transmitted.

Void frames are transmitted in order to reset the Valid Transmission timers (TVX) in other stations in order to eliminate an unnecessary entry to the Claim state. Stations are not required to copy Void frames. Void frames are transmitted by the Ring Engine in two situations:

1. While holding a token when no data is ready to be transmitted.
2. After a frame transmission is aborted.

My_Void frames are transmitted by the Ring Engine in three situations:

1. After a request to measure the Ring Latency has been made when the next early token is captured.
2. After this station wins the Claim Process before the token is issued.
3. After a frame has been transmitted with the STRIP option before the token for that service opportunity is issued.

Void frames are also detected by the Ring Engine. A Void frame with a Source Address other than MSA or MLA is considered an Other_Void frame.

Claim Frames

Claim frames are generated continuously with minimum preamble while the Ring Engine is in the Transmit Claim state.

The format of Claim frames generated by the Ring Engine is shown in Table 4-7. When long addressing is enabled, frames with the long address are transmitted, otherwise frames with the short address are transmitted.

The Ring Engine detects reception of valid claim frames. A comparison is performed between the (first) four bytes of the received INFO field and TREQ in order to distinguish Higher_Claim, Lower_Claim, and My_Claim. Details are given in Appendix A.

TABLE 4-6. Void Frames

Type	Enable	Size	SFS		FC	DA	SA	FCS	EFS
Void	ESA	Short	PA	SD	40	Null	MSA	FCS	TRRR
Void	Not ESA	Long	PA	SD	00	Null	MLA	FCS	TRRR
My_Void	ESA	Short	PA	SD	40	MSA	MSA	FCS	TRRR
My_Void	Not ESA	Long	PA	SD	00	MLA	MLA	FCS	TRRR

TABLE 4-7. Claim Frames

Type	Enable	Size	SFS		FC	DA	SA	INFO	FCS	EFS
My_Claim	Not ELA	Short	PA	SD	83	MSA	MSA	TREQ	FCS	TRRR
My_Claim	ELA	Long	PA	SD	C3	MLA	MLA	TREQ	FCS	TRRR

4.0 FDDI MAC Facilities (Continued)

Beacon Frames

Beacon frames are transmitted continuously with minimum preamble when the Ring Engine is in the Transmit Beacon state. The format of Beacon frames generated by the Ring Engine is shown in Table 4-8. When long addressing is enabled, frames with the long address are transmitted, otherwise frames with the short address are transmitted.

When the Transmit Beacon State is entered from the Transmit Claim State the first byte of the 4 byte TBT Field is transmitted as Zero.

Beacon frames that require alternative formats such as Directed Beacons must be generated externally.

The Ring Engine detects reception of valid Beacon frames and distinguishes between Beacon frames transmitted by this MAC (My_Beacon) and Beacon frames transmitted by other stations (Other_Beacon). Details are given in Appendix A.

4.3 FRAME COUNTS

To aid in fault isolation and to enhance the management capabilities of a ring, the Ring Engine maintains several frame counts. The Error and Isolated frame counts increment when a frame is received with one or more errors that were previously undetected. The Ring Engine then corrects the error such that a downstream station will not increment its count.

The size of the counters has been chosen such that minimal software intervention is required, even under marginal operating conditions.

The following counts are maintained by the Ring Engine:

FRCT	Frame Received
EICT	Error Isolated
LFCT	Lost Frame
FCCT	Frames Copied
FNCT	Frames Not Copied
FTCT	Frames Transmitted

4.3.1 Frame Received Count

The Frame Received Count (FRCT) is specified in the FDDI MAC Standard, and is the count of all complete frames received. This count includes frames stripped by this station.

4.3.2 Error Isolated Count

The Error Isolated Count (EICT) is specified in the FDDI MAC Standard, and is the count of error frames detected by this station and no previous station. It increments when:

1. An FCS error is detected and the received Error Indicator (Er) is not equal to S.
2. A frame of invalid length (i.e., off boundary T) is received and Er is not equal to S.
3. Er is not R or S.

4.3.3 Lost Frame Count

The Lost Frame Count (LFCT) is specified in the FDDI MAC Standard, and is the count of all instances where a format error is detected in a frame or token such that the credibility of PDU reception is placed in doubt. The Lost Frame Count is incremented when any symbol other than data or Idle symbols are received between the Starting and Ending Delimiters of a PDU (this includes parity errors).

4.3.4 Frame Copied Count

The Frames Copied Count (FCCT) is specified in the FDDI MAC-2 Standard, and is the count of the number of frames copied by this station. The count is incremented when an internal or external match occurs (when Option.EMIND is enabled) on the Destination Address, no errors were detected in the frame and the frame was successfully copied (VCOPY = 1). This can be used to accumulate station performance statistics. Frames copied promiscuously, MAC frames, Void frames and NSA frames received with the A indicator set are not included in this count.

4.3.5 Frames Not Copied Count

The Frames Not Copied Count (FNCT) is specified in the FDDI MAC-2 Standard, and is the count of frames intended for this station that were not successfully copied by this station. The count is incremented when an internal or external (when Option.EMIND is enabled) Destination Address match occurs, no errors were detected in the frame, and the frame was not successfully copied (VCOPY = 0). This count is an indication of insufficient buffering or frame processing capability for frames addressed to the station. MAC frames, Void frames and NSA frames received with the A indicator set are not included in this count.

4.3.6 Frames Transmitted Count

The Frames Transmitted Count (FTCT) is specified in the FDDI MAC-2 Standard, and is incremented every time a complete frame is transmitted from the MAC Request Interface. The count is provided as an aid to accumulate station performance statistics. Void and MAC frames generated by the Ring Engine are not included in the count.

4.4 TIMERS

4.4.1 Token Rotation Timer

The Token Rotation Timer (TRT) times token rotations from arrival to arrival. TRT is used to control ring scheduling during normal operation and to detect and recover from serious ring error situations.

TRT is loaded with the maximum token rotation time, TMAX, when the ring is not operational. TRT is loaded with the negotiated Target Token Rotation Time, TNEG, when the ring is operational.

TABLE 4-8. Beacon Frames

Type	Enable	Size	SFS		FC	DA	SA	INFO	FCS	EFS
My_Beacon	Not ELA	Short	PA	SD	82	Null	MSA	TBT	FCS	TRRR
My_Beacon	ELA	Long	PA	SD	C2	Null	MLA	TBT	FCS	TRRR

4.0 FDDI MAC Facilities (Continued)

4.4.2 Token Holding Timer

The Token Holding timer (THT) is used to limit the amount of ring bandwidth used by a station for asynchronous traffic once the token is captured. THT is used to determine if the captured token is (still) usable for asynchronous transmission. A token is usable for asynchronous traffic if THT has not reached the selected threshold. Two asynchronous thresholds are supported; one that is fixed at the Negotiated Target Token Rotation Time (TNEG), and one that is programmable at one of 16 Asynchronous Priority Thresholds. Requests to transmit frames at one of the priority thresholds are serviced when the Token Holding Timer (THT) has not reached the selected threshold.

4.4.3 Late Count

The Late Count (LTCT) is implemented differently than suggested by the Standard, but provides similar information. The function of the Late Count is divided between the Late_ Flag that is equivalent to the standard Late Count with a non-zero value and a separate counter. Late Flag is maintained by the Ring Engine to indicate if it is possible to send asynchronous traffic. When the ring is operational, Late Count indicates the time it took the ring to recover the last time the ring went non-operational. When the ring is non-operational, Late Count indicates the time it has taken (so far) to recover the ring.

The Late Count is incremented every time TRT expires while the ring is non-operational and Late_Flag is set (once every TMAX).

The Late Count is provided to assist Station Management, SMT, in the isolation of serious ring errors. In many situations the ring will recover very quickly and late count will be of marginal utility. However in the case of serious ring errors, it is helpful for SMT to know how long it has been since the ring went non-operational (with TMAX resolution) in order to determine if it is necessary to invoke recovery procedures. When the ring goes non operational there is no way to know how long it will stay non-operational, therefore a timer is necessary. If the Late Count were not provided, SMT would be forced to start a timer every time the ring goes non-operational even though it may seldom be used. By using the provided Late Count, an SMT implementation may be able to alleviate this additional overhead.

4.4.4 Valid Transmission Timer

The Valid Transmission Timer (TVX) is reset every time a valid PDU is received. TVX is used to increase the responsiveness of the ring to errors. Expiration of the TVX indicates that no PDU has been received within the timeout period and causes the Transmitter to invoke the recovery Claim Process.

4.4.5 Token Received Count

The Token Received Count (TKCT) is incremented every time a valid token arrives. The Token Count can be used with the Ring Latency Count to calculate the average network load over a period of time. The frequency of token arrival is inversely related to the network load.

4.4.6 Ring Latency Count

The Ring Latency Count (RLCT) is a measurement of time for PDUs to propagate around the ring. This counter contains the last measured ring latency whenever the Ring Latency Valid bit of the Token Event Register (TELR.RLVLD) is one.

The Latency Counter increments every 16 byte times (1.28 μ s) and is used to measure ring latencies up to 1.3421772 seconds directly with an accuracy of 1.2 μ s. No overflow or increment event is provided with this counter.

4.5 RING SCHEDULING

FDDI uses a timed token protocol to schedule the use of the ring. The protocol measures load on the network by timing the rotation of the token. The longer the token rotation time the greater the instantaneous load on the network. By limiting the transmission of data when the token rotation time exceeds a target rotation time, a maximum average token rotation time is realized. The protocol is used to provide different classes of service.

Multiple classes of service can be accommodated by setting different target token rotation times for each class of service.

The Ring Engine supports Synchronous, Non-Restricted Asynchronous, Restricted Asynchronous, and Immediate service classes. The Immediate service class is supported when the ring is non-operational; the other classes are supported when the ring is operational.

4.5.1 Synchronous Service Class

The Synchronous service class may be used to guarantee a maximum response time (2 times TTRT), minimum bandwidth, or both.

Each time the token arrives, a station is permitted to transmit one or more frames in accordance with its synchronous bandwidth allocation regardless of the status of the token (late or early; Restricted or Non-Restricted).

Since the Ring Engine does not provide a mechanism for monitoring a station's synchronous bandwidth utilization, the user must insure that no synchronous request requires more than the allocated bandwidth.

To help ensure that synchronous bandwidth is properly allocated after ring configuration, synchronous requests are not serviced after a Beacon frame is received. After a major reconfiguration has occurred, management software must intervene to verify or modify the current synchronous bandwidth allocation.

4.5.2 Non-Restricted Asynchronous Service Class

The Non-Restricted Asynchronous service class is typically used with interactive and background traffic. Non-restricted Asynchronous requests are serviced only if the token is early and the Token Holding Timer has not reached the selected threshold.

Asynchronous service is available at two priority thresholds, the Negotiated Target Token Rotation Time plus one programmable threshold. Management software may use the priority thresholds to discriminate additional classes of traffic based on current loading characteristics of the ring. The priority thresholds may be determined using the current TTRT and the Ring Latency. In this case, application software is only concerned with the priority level of a request.

As an option, Asynchronous Requests may be serviced with THT disabled. This is useful when it is necessary to guarantee that a multi-frame request will be serviced on a single token opportunity. Because of the possibility of causing late tokens, this capability should be used with caution, and should only be allowed when absolutely necessary.

4.0 FDDI MAC Facilities (Continued)

4.5.3 Restricted Asynchronous Service Class

The Restricted Asynchronous service class is useful for large transfers requiring all of the available Asynchronous bandwidth. The Restricted Token service is useful for large transfers requiring all of the available (remaining) asynchronous bandwidth.

The Restricted Token service may also be used for operations requiring instantaneous allocation of the remaining synchronous bandwidth when Restricted Requests are serviced with THT disabled. This is useful when it is necessary to guarantee atomicity, i.e., that a multi-frame request will be serviced on a single token opportunity.

A Restricted dialogue consists of three phases:

1. Initiation of a Restricted dialogue:
 - Capture a Non-Restricted Token
 - Transmit zero or more frames to establish a Restricted dialogue with other stations
 - Issue a Restricted Token to allow other stations in the dialogue to transmit frames
2. Continuation of a Restricted dialogue:
 - Capture a Restricted Token
 - Transmit zero or more frames to continue the Restricted dialogue
 - Issue a Restricted Token to allow other stations in the dialogue to transmit frames
3. Termination of a Restricted dialogue:
 - Capture a Restricted Token
 - Transmit zero or more frames to continue the Restricted dialogue
 - Issue a Non-restricted Token to return to the Non-restricted service class

Initiation of a Restricted dialogue will prevent all Non-restricted Asynchronous traffic throughout the ring for the duration of the dialogue, but will not affect Synchronous traffic.

To ensure that the Restricted traffic is operating properly, it is possible to monitor the use of Restricted Tokens on the ring. When a Restricted Token is received, the event is latched and under program control may generate an interrupt. In addition, a request to begin a Restricted dialogue will only be honored if both the previous transmitted Token and the current received Token were Non-restricted tokens. This is to ensure that the upper bound on the presence of a Restricted dialogue in the ring is limited to a single dialogue.

As suggested by the MAC-2 Draft standard, to help ensure that only one Restricted dialogue will be in progress at any given time, Restricted Requests are not serviced after a MAC frame is received until Restricted Requests are explicitly enabled by management software. Since the Claim Process results in the generation of a Non-restricted Token, this prevents stations from initiating another restricted dialogue without the intervention of management software.

4.5.4 Immediate Service Class

The Immediate service class facilitates several non-standard applications and is useful in ring failure recovery (e.g., Transmission of Directed Beacons). Certain ring failures may cause the ring to be unusable for normal traffic, until the failure is remedied.

Immediate requests are only serviced when the ring is non-operational. Immediate requests may be serviced from the Transmitter Data, Claim, and Beacon states Options are available to force the Ring Engine to enter the Claim or Beacon state, to prohibit it from entering the Claim state, or to remain in the Claim state when receiving My_Claim.

On the completion of an Immediate request, a Token (Non-restricted or Restricted) may optionally be issued. Immediate requests may also be used in non-standard applications such as a full duplex point to point link.

5.0 Functional Description

5.1 TOKEN HANDLING

5.1.1 Token Timing Logic

The FDDI Ring operates based on the Timed Token Rotation protocol where all stations on the ring negotiate on the maximum time that the stations have to wait before being able to transmit frames. This value is termed the Negotiated Target Token Rotation Time (TTRT). The TTRT value is stored in the TNEG Register.

Stations negotiate for TTRT based on their TREQ that is assigned to them upon initialization.

Each station keeps track of the token arrival by setting the Token Rotation Timer (TRT) to the TTRT value. If the token is not received within TTRT (the token is late), the event is recorded by setting the Late_Flag. If the token is not received within twice TTRT (TRT expires and Late_Flag is set), there is a potential problem in the ring and the recovery process is invoked.

Furthermore, the Token Holding Timer (THT) is used to limit the amount of ring bandwidth used by a station for Asynchronous traffic once the token is captured. Asynchronous traffic is prioritized based on the Late_Flag which denotes a threshold at TTRT and an additional Asynchronous Priority Threshold (THSH). The Asynchronous Threshold comparison (Apri 1) is pipelined, so a threshold crossing may not be detected immediately; however, the possible error is a fraction of the precision of the threshold values.

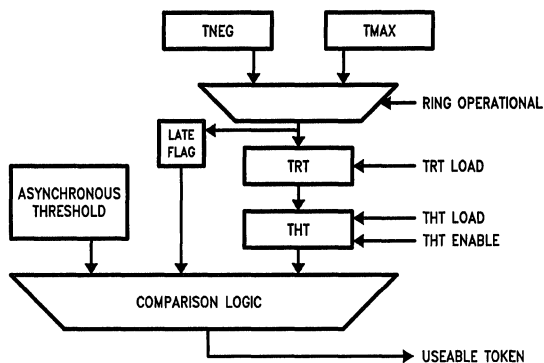
The Token Timing Logic consists of two Timers, TRT and THT, in addition to the TMAX and TNEG values loaded into these counters (See *Figure 5-1*).

The Timers are implemented as count-up counters that increment every 80 ns. The Timers are reset by loading TNEG or TMAX into the counters where TNEG and TMAX are unsigned twos complement numbers. This allows a Carry flag to denote timer expiration.

On an early token arrival (Late_Flag is not set), TRT is loaded with TNEG and counts up. On a late token arrival (Late_Flag is set), Late_Flag is cleared and TRT continues to count. When TRT expires and Late_Flag is not set, Late_Flag is set and TRT is loaded with TNEG.

THT follows the value of TRT until a token is captured. When a token is captured, TRT may be reloaded with TNEG while THT continues to count from its previous value (THT does not wrap around). THT increments when enabled. THT is disabled during synchronous transmission and a special class of asynchronous transmission. THT is used to determine if the token is usable for asynchronous requests.

5.0 Functional Description (Continued)



TL/F/10387-3

FIGURE 5-1. Token Timing Logic

If TRT expires while Late_Flag is set, TRT is loaded with TMAX and the recovery process (Claim) is invoked. When TRT expires and the ring is not operational, TRT is loaded with TMAX. TRT is also loaded with TMAX on a MAC Reset.

5.1.2 Token Recovery

While the ring is operational, every station in the ring uses the Negotiated Target Token Rotation Time, TNEG. The MAC implements the protocol for negotiation of this target token rotation time (TTRT) through the Claim Process. The shortest requested Token Rotation Time is used by all of the stations in the ring as the TNEG.

If TRT expires with Late_Flag set, a token has not been received within twice TTRT (Target Token Rotation Time). If TVX (Valid Transmission Timer) expires, the station has not received a valid token within TVX Max. Both these events require token recovery and cause the Ring Engine to enter the Claim Process.

In the Claim Process a MAC continuously transmits Claim frames containing TREQ. Should the MAC receive a Claim frame with a shorter TREQ (larger value—Higher_Claim) it leaves the Claim State. A station that receives its own Claim frame gains the right to send the first token and make the ring operational again. If the Claim Process does not complete successfully, TRT will expire and the Beacon Process is invoked.

The Beacon Process is used for fault isolation. A station may invoke the Beacon Process through an SM_Control.request(Beacon). When a station enters the Beacon Process, it continuously sends out Beacon frames. The Beacon Process is complete when a station receives its

own Beacon frame. That station then enters the Claim Process, to re-initialize the ring.

5.2 SERVICING TRANSMISSION REQUESTS

A Request to transmit one or more frames is serviced by the Ring Engine. After a Request is submitted to the Ring Engine, the Ring Engine awaits an appropriate Service Opportunity in which to service the Request. Frames associated with the Request are transmitted during the Service Opportunity. The definition of a Service Opportunity is different depending on the operational state of the ring.

A Service Opportunity begins when the criteria presented to the Ring Engine are met. This criteria contains the requested service class (synch, asynch, asynch priority, immediate) and the type of token to capture (restricted, non-restricted, any, none).

During a service opportunity, the Ring Engine guarantees that a valid frame is sent with at most 40 bytes of preamble. When data is not ready to be transmitted, Void frames are transmitted to reset the TVX timers in all stations. During an immediate request while in the Claim or Beacon States, when no Claim or Beacon frames are ready to be transmitted, the internally generated Claim or Beacon frames are transmitted.

5.2.1 Service Opportunity While Ring Operational

Beginning of Service Opportunity

Table 5-1 shows the conditions that must be true when a valid token is received in order to begin a Service Opportunity when the ring is operational.

TABLE 5-1. Beginning of Service Opportunity

Requested Service Class	Requested Token Capture Class	Criteria	Received Token Class
Asynchronous Priority	non-restricted	THT > THSH Late_Flag = 0 Ring_Op = 1	non-restricted
Asynchronous	non-restricted	Late_Flag = 0 Ring_Op = 1	non-restricted
Asynchronous	restricted	Late_Flag = 0 Ring_Op = 1	restricted
Synchronous	any	Ring_Op = 1	any

5.0 Functional Description (Continued)

In addition to the criteria mentioned above, additional criteria apply to the servicing of Synchronous and Restricted Requests.

- Synchronous Requests are not serviced if RELR.BCNR is set (See Section 4.5.1).
- Restricted requests are not serviced when RELR.BCNR, RELR.CLMR, or RELR.OTRMAC are set. (See Section 4.5.3).
- Restricted Dialogues may only begin when a non-restricted token has been received and transmitted (See Section 4.5.3).

End of Service Opportunity

The Service Opportunity continues until either a token is issued or the ring becomes non-operational.

A token is issued after the current frame, if any, is transmitted when:

1. It is no longer necessary to hold the token
 - All frames of all active requests have been transmitted
2. The token became unusable while servicing a request
 - Asynchronous Priority threshold reached (If an Asynch Priority Request is being serviced)
 - THT expired (if enabled)

When the ring becomes non-operational the current frame transmission is aborted. The ring may go non-operational while holding a token as a result of any one of the following conditions:

- A MAC Reset
- Reception of a valid MAC frame
- TRT expiration, (TRT was reset when the token was captured)

Issue Token Type

The criteria presented to the Ring Engine to begin a Service Opportunity, also contains the Issue Token Class. The Issue Token Class is used if servicing of that request was completed (the last frame of that request was transmitted), otherwise a token of the Capture Token Class is issued.

When servicing multiple requests on a single service opportunity, the Issue Token Class of the previous class becomes the capture class for the next request for purposes of determining usability.

The type of token issued depends on the service class and the type of token captured as shown in Table 5-2.

5.2.2 Service Opportunity while Ring Not Operational

While the ring is not operational, a service opportunity occurs if an immediate transmission is requested from the

transmitter Data, Claim or Beacon State, and the transmitter is in the appropriate state.

The service opportunity continues until any one of the following conditions exist:

1. No (additional) frames are to be sent
2. TMAX of time elapses on this request
3. The transmitter exits the requested state
4. The ring becomes operational while servicing an immediate request

5.2.3 Frame Transmission

Frames associated with the current request may be transmitted at any time during a Service Opportunity. In many applications, data is ready to be transmitted when the request is presented to the interface. Soon after the Service Opportunity begins, frame transmission begins. In other applications in order to minimize the effects of ring latency it is desirable to capture the token when no data is ready to be transmitted. This capability results in wasted ring bandwidth and should be used judiciously.

During transmission, a byte stream is passed from the System Interface to the MAC Request Interface. The data is passed through the Ring Engine and appears at the PHY Request Interface two byte times later.

While a frame is being transmitted, the request parameters for the next request (if different) may be presented to the interface. At the end of the current frame transmission, a decision is made to continue or cancel the current service opportunity based on the new request parameters.

During a transmission several errors can occur. A transmission may be terminated unsuccessfully because of external buffering or interface parity errors, internal Ring Engine errors, a MAC reset, or reception of a MAC frame. When a transmission is aborted due to an external error (and Option.IRPT is not set), a Void frame is transmitted to reset the TVX timers in all stations in the ring. When a frame is aborted due to a transmission error, the Service Opportunity is not automatically ended.

5.3 REQUEST SERVICE PARAMETERS

5.3.1 Request Service Class

The Request Service corresponds to the Request Service Class and the token class parameters of the (SM_)MA_DATA.request and (SM_)MA-Token.request primitives as specified in the Standard.

14 useful combinations of the Requested Service Class (Non-Restricted Asynchronous, Restricted Asynchronous, Synchronous, Immediate), the Token Capture and Issue Class, and THT Enable are supported by the Ring Engine as shown in Table 5-3.

TABLE 5-2. Token Transmission Type

Service Class	Token Captured	Token Issued
Non-Restricted	Non-Restricted	Non-Restricted
Begin Restricted	Non-Restricted	Restricted
Continue Restricted	Restricted	Restricted
End Restricted	Restricted	Non-Restricted
Immediate	None	None
Immediate Non-Restricted	None	Non-Restricted
Immediate Restricted	None	Restricted

5.0 Functional Description (Continued)

TABLE 5-3. Request Service Classes

RQRCLS	Name	Class	THT	Token Capture	Token Issue	Notes
0000	None	None				
0001	Apri_1	Async THSH1	Enabled	Non-rstr	Non-rstr	
0010	Reserved	Reserved				
0011	Reserved	Reserved				
0100	Syn	Synch	Disabled	Any	Captured	1
0101	Imm	Immediate	Disabled	None	None	4
0110	ImmN	Immediate	Disabled	None	Non-rstr	4
0111	ImmR	Immediate	Disabled	None	Rstr	4
1000	Asyn	Asynch	Enabled	Non-rstr	Non-rstr	
1001	Rbeg	Restricted	Enabled	Non-rstr	Rstr	2, 3
1010	Rend	Restricted	Enabled	Rstr	Non-rstr	2
1011	Rcnt	Restricted	Enabled	Rstr	Rstr	2
1100	AsynD	Asynch	Disabled	Non-rstr	Non-rstr	
1101	RbeginD	Restricted	Disabled	Non-rstr	Rstr	2, 3
1110	RenD	Restricted	Disabled	Rstr	Non-rstr	2
1111	RcntD	Restricted	Disabled	Rstr	Rstr	2

Note 1: Synchronous Requests are not serviced when bit BCNR of the Ring Event Latch Register is set.

Note 2: Restricted Requests are not serviced when bit BCNR, CLMR, or OTRMAC of the Ring Event Latch Register is set.

Note 3: Restricted Dialogues only begin when a Non-Restricted token has been received and transmitted.

Note 4: Immediate Requests are serviced when the ring is Non-Operational. These requests are serviced from the Data state if neither signal RQCLM nor RQBCN is asserted. If signal RQCLM is asserted, Immediate Requests are serviced from the Claim State. If signal RQBCN is asserted, Immediate Requests are serviced from the Beacon State. RQCLM and RQBCN do not cause transitions to the Claim and Beacon States.

Requests are serviced on a Service Opportunity meeting the requested criteria.

External support is required to limit the requests presented to the MAC Interface by different MAC Users (SMT, LLC, etc.).

A Token Capture Class of **non-rstr** indicates that the Transmitter Token Class must be Non-Restricted to begin servicing the request. A Token Capture Class of **rstr** indicates that the Transmitter Token Class must be Restricted to begin servicing the Request. A Token Issue Class of **non-rstr** means that the Transmitter Token Class will be Non-Restricted upon completion of the request. A Token Issue Class of **rstr** means that the Transmitter Token Class will be Restricted upon completion of the request.

5.3.2 Request Options

The Request Options provide the ability for Source Address Transparency (SAT) and FCS Transparency (FCST). In both cases, data from the request stream is transmitted in place of data from either the Ring Engine. The use of Source Address transparency has no effect on the sequencing of the interface. When Source Address transparency is not used, the SA from the internal parameter RAM is substituted for the SA bytes in the request stream, which must still be present. Since the FCS is appended to the frame, when FCS transparency is not used, no FCS bytes are present in the request stream.

Source Address Transparency

Normally the SA field in a frame is generated by the BMAC device, using either the MSA or MLA. When the SA Transparency option is selected, the SA from the data stream is transmitted in place of the MSA or MLA. The SAT option can be invoked on a per frame basis upon the assertion of the SAT signal (Pin 12).

When the SA Transparency option is selected, it is necessary to rely on an alternate stripping mechanism because stripping based on the returning SA only guarantees that frames with MSA or MLA will be stripped. Either the Void Stripping option (described below) may be invoked, or external hardware that forces stripping using the EM (External M_Flag) signal is required.

The MSB of the SA is not controlled by this option. It is normally forced to Zero. It can be controlled using the Source Address MSB Transparency option described below.

SA Transparency is possible for all frames (including MAC frames). External support is required to limit the use of SA Transparency to certain MAC Users. SA Transparency should not be used with externally generated MAC Frames in order to maintain accountability, but this is not enforced by the Ring Engine.

5.0 Functional Description (Continued)

SA Transparency also overrides the Long and Short Addressing enables. For example, if Long Addressing is not enabled, it is still possible to transmit frames with Long Addresses. Similarly, if Short Addressing is not enabled, it is still possible to transmit Frames with Short Addresses. This may be useful in full duplex point to point applications and for diagnostic purposes.

Source Address Most Significant Bit Transparency

With the Source Address MSB Transparency option, the MSB of the SA is sourced from the data stream, as opposed to being transmitted as Zero. The SA MSB Transparency option is selected by asserting signal SAIGT (Pin 11).

Unless the Source Address Transparency option is also selected, the rest of the SA is generated by the Ring Engine.

The MSB of the SA is used to denote the presence of the Routing Information Field used in Source Routing algorithms (as in the IEEE 802.5 protocol). This option is useful for stations that utilize Source Routing. In these applications, the SA can still be generated by the Ring Engine, even when routing information is inserted into the data stream.

Void Stripping

This option is useful for removing bridged and ownerless frames and remnants (fragments) from the ring.

In the Void Stripping protocol, two My_Void frames are transmitted at the end of a service opportunity. Stripping continues until one of the following conditions occur:

- One My_Void frames returns (The Second My_Void will be stripped on the basis of the SA)
- A Token is received
- An Other_Void is received
- A MAC frame other than My_Claim is received
- A MAC Reset occurs

If any frame of a Service Opportunity requests this option, then all frames on that service opportunity will be stripped using this method. Void Stripping is invoked upon the assertion of the STRIP signal (Pin 13) at the beginning of a frame transmission.

Void Stripping is also automatically invoked by this station if it wins the Claim Process before the initial token is issued. This removes all fragments and ownerless frames from the ring when the ring becomes operational.

FCS Transparency

Normally, the BMAC device generates and transmits the FCS. When the Frame Check Sequence Transparency option is selected, the Ring Engine device does not append the FCS to the end of the information field. This option is selected by asserting signal FCST (Pin 14).

The receiving stations treat the last four bytes of the data stream as the FCS.

This option may be useful for end to end FCS coverage when crossing FDDI bridges, for diagnostic purposes, or in Implementer frames.

5.4 FRAME VALIDITY PROCESSING

A valid frame is a frame that meets the minimum length criteria and contains an integral number of data symbol pairs between the Starting and Ending Delimiters as shown in Table 5-4.

On the Transmit side, frames are checked to see that they are of a minimum length. If the end of a frame is reached before a valid length is transmitted, the frame will be aborted and a Void frame will be transmitted (as with all aborted frames). A MAC frame with a zero length INFO field will not be aborted even though the Receiver will not recognize it as a valid frame. Frame lengths are not checked for the maximum allowable length (4500 bytes).

Also on the Transmit side, the L bit in the FC field is checked against the ESA and ELA bits in the Option Register (if the SA Transparency option is not selected) to insure that a frame of that address length can be transmitted. If the selected address length is not enabled, the frame is aborted at the beginning of the SA field. If SA Transparency is selected, the frame is not aborted.

TABLE 5-4. Valid Frame Length

Frame Types	Short Address	Long Address	Notes
	(Minimum Number of Bytes)		
Void	9	17	
MAC	13	21	Including a 4 Byte INFO Field
Non-MAC	9	17	Including a 0 Byte INFO Field

5.0 Functional Description (Continued)

5.5 FRAME STATUS PROCESSING

Each frame contains three or more Control Indicators. The FDDI Standard specifies three: the E, A, and C Indicators.

When a frame is transmitted, the Control Indicators are transmitted as R (Reset) symbols. If an error is detected by a station that receives the frame, the E Indicator is changed to an S (Set) symbol. If a station recognizes the DA of a frame as its own address (Individual, Group or Broadcast), the A Indicator is changed to an S symbol. If that station then copies the frame, the C Indicator is changed to an S symbol.

The received value of the Control Indicators for every frame received is reported at the MAC Indicate Interface on signals MID(7-0). On a frame transmitted by this station, the returning Control Indicators give the transmission status.

The Ending Delimiter followed by the Frame Status Indicators should begin and end on byte boundaries. Control Indicators are repeated until the first non R, S, or T is received.

The processing of properly aligned E, A, and C indicators by the Ring Engine is detailed in Table 5-5. Given the shown received Control Indicator values and the settings of the internal flags, the noted control indicator values will be transmitted.

TABLE 5-5. Control Indicators Processing

Received Indicators			Flags				Transmitted Indicators		
E	A	C	E	A	Copy	N	E	A	C
R	R	R	0	0	X	X	R	R	S
R	R	R	0	1	0	X	R	S	R
R	R	R	0	1	1	X	R	S	S
X	R	R	1	X	X	X	S	R	R
R	R	S	0	0	X	X	R	R	S
R	R	S	0	1	0	X	R	S	R
R	R	S	0	1	1	X	R	S	S
X	R	S	1	X	X	X	S	R	S
R	S	R	0	X	X	1	R	S	R
R	S	R	0	X	0	0	R	S	R
R	S	R	0	1	1	0	R	S	S
R	S	R	0	0	X	X	R	S	R
R	S	S	0	X	X	X	R	S	S
X	S	S	1	X	X	X	S	S	S
R	R	T	0	0	X	X	R	R	T
R	R	T	0	1	0	X	R	S	R
R	R	T	0	1	1	X	R	S	S
X	R	T	1	X	X	X	S	R	T
R	S	T	0	1	1	0	R	S	S
R	S	T	0	0	X	X	R	S	T
R	S	T	0	1	0	X	R	S	R
R	S	T	0	1	1	1	R	S	R
X	S	T	1	X	X	X	S	S	T

E_Flag is set when the local FCS check fails or when the E Indicator is received as anything other than R.

A_Flag is the internal _Flag or the external A Flag (pin EA) when Option.Emind is set.

The Copy Flag is a one cycle delayed version of the VCOPY input.

N_Flag indicates that an NSA frame is being received. This signal is sampled at the same time that the received A indicator is being investigated.

X Represents a Don't Care Condition.

5.0 Functional Description (Continued)

5.5.1 Odd Symbols Handling

When the first T symbol of a frame is received as the second symbol of a symbol pair (the T symbol is received off-boundary), the Ring Engine corrects this condition by sending out the symbol sequence TSII. This symbol sequence indicates the end of the frame and that an error has been detected in the frame. Note that this is a low probability error event.

Reception of symbols other than R, S, and T during the Frame Status processing is also a low probability event. This event is handled slightly differently on the first byte of the Ending Frame Status.

On the first byte of the Ending Frame Status, if the symbol following the T symbol is not [R or S], the symbol sequence TSII is transmitted and the error and frame counts are incremented.

After the first byte of the Ending Frame Status, if either the first symbol is not [R or S] or the second symbol is not [R or S or T], an Idle symbol pair (II) is transmitted.

5.6 SMT FRAME PROCESSING

All SMT frames are handled as all other frames with the exception of the SMT Next Station Addressing (SMT NSA) frame. NSA frames are used to announce this station's address to the next addressed station. The current SMT protocol requires stations to periodically (at least once every 30 seconds) transmit an NSA frame. Since the Broadcast address is used, and every station is required to recognize the broadcast address, the downstream neighbor will set the A Indicator. A station can determine its upstream neighbor by finding NSA frames received with the A Indicator received as R. By collecting this information from all stations, a map of the logical ring can be built.

Additionally, only the station that sets the A Indicator is permitted to set the C Indicator on such frames. In this way, the station that sends out the NSA frame can determine if the next addressed station copied the frame by examining the returning C Indicator.

5.7 MAC FRAME PROCESSING

Upon the reception of a valid MAC frame (Claim, Beacon, or Other), the Ring_Operational flag is reset and the Ring Engine enters the Idle, Claim or Beacon State. Received Claim and Beacon frames are processed as defined in the Standard (See Appendix A), unless inhibited by the bits in the Option Register.

5.7.1 Claim Token Process

Receive

When a Claim frame is received, its Frame Type is reported (Claim frame) along with the type of Claim frame.

There are three types of Claim frames: My_Claim, Higher_Claim, and Lower_Claim.

A My_Claim frame is a Claim frame with a Source Address that matches this station address and the T_Bid_Rc in the INFO field is equal to this station's TREQ.

A Higher_Claim frame is a Claim frame with a Source Address that does not match this station address and the T_Bid_Rc in the INFO field is greater than this station's TREQ.

A Lower_Claim frame is a Claim frame with a Source Address that does not match this station address and the T_Bid_Rc in the INFO field is less than this station's TREQ.

Transmit

Claim frames are transmitted continuously while in the Claim State.

Claim frames are generated by the Ring Engine, unless an Immediate Claim Request is present at the MAC Request Interface. Even if an Immediate Claim Request is present at the MAC Request Interface, at least one Claim frame must be generated by the Ring Engine before Claim frames from the Interface are transmitted.

For internally generated Claim frames, the Information field is transmitted as the 4-byte Requested Target Token Rotation Time.

The Information field of a Claim frame consists of the station's Requested Target Token Rotation Time. In the Ring Engine implementation, TREQ is programmable with 20.48 μ s resolution and a maximum value of 1.34 seconds.

Claim Protocol

Entry to the Claim state occurs whenever token recovery is required. The Recovery Required condition occurs when:

- TRT expires and Late_Flag is set
- TVX expires
- A Lower Claim frame or My_Beacon frame is received

Entry to the Claim state may be blocked by enabling the Inhibit Recovery Required option (bit Option.ITR).

The Claim state is entered (even if Option.IRR = 1) with a SM_MA_Control.request (Claim) (Set Function.CLM to 1).

While in the Claim state:

- Claim frames are transmitted continuously
- If a Higher Claim frame is received, the station exits the Claim state and enters the IDLE state. In this state it then repeats additional Higher Claim frames.
- If a Lower Claim frame is received, this station continues to send its own Claim frames and remains in the Claim state.

Eventually, if a logical ring exists, the station with the shortest TREQ on the ring should receive its own Claim frames, the My Claim frame. This completes the Claim Token Process. This one station then earns the right to issue a token to establish an Operational ring.

An option is provided to remain in the Claim state if this station won the Claim Token Process by enabling the Inhibit Token Release Option (bit Option.ITR).

5.0 Functional Description (Continued)

5.7.2 Beacon Process

Receive

When a Beacon frame is received, its Frame Type is reported (Beacon frame) along with the type of Beacon frame.

There are two types of Beacon frames: My__Beacon and Other__Beacon.

A Beacon frame is considered a My__Beacon if its Source Address matches this station's address (long or short).

A Beacon frame is marked as Other__Beacon if its Source Address does not match this station's address.

Transmit

Beacon frames are transmitted continuously while in the Beacon state.

Beacon frames are generated by the Ring Engine, unless an Immediate Beacon Request is present at the MAC Request Interface. Even if an Immediate Beacon Request is present at the MAC Request Interface, at least one Beacon frame must be generated by the Ring Engine before Beacon frames from the Interface are transmitted.

For internally generated Beacon frames, the Ring Engine uses the TBT in the Information field.

Beacon Protocol

Entry to the Beacon state occurs under two conditions:

- A failed Claim Process (TRT expires during the Claim process)
- An SM__MA__Control.request (Beacon) (Set Function.BCN to 1).

Beacon frames are then transmitted until the Beacon process is completed.

If an Other__Beacon frame is received, this station exits the Beacon state, stops sending its own Beacon frames, and repeats the incoming Beacon frames.

If a My__Beacon frame is received, the station has received back its own Beacon frame; thus successfully completing the Beacon process. The station then enters the Claim Process.

5.7.3 Handling Reserved MAC Frames

A Reserved MAC frame is any MAC frame aside from the Claim and Beacon frame. Tokens are not considered MAC frames even though Format bit (FC.FF) are the same as for MAC frames.

When a Reserved MAC frame (Other__MAC) is received, it is treated as a Higher Claim. If the Transmitter is in the Claim state when a Reserved MAC frame is received, the Transmitter returns to the Idle state and then repeats the next Reserved MAC frame received. If the Transmitter is in the Beacon state and a Reserved MAC frame is received, the Transmitter continues to transmit Beacon frames. If the Transmitter is in the Idle state, the Reserved MAC frame is repeated.

5.8 RECEIVE BATCHING SUPPORT

The Ring Engine stores each received SA and compares the incoming SA with the previous SA. This may be used to batch status on frames received from the same station.

The SameSA signal is asserted when:

1. The current and previous non-Void frames were not MAC frames
2. The size of the address of the current frame is the same as the size of the address of the previous non-Void frame
3. The SA of the current frame is the same as the SA of the previous non-Void frame.

On MAC frames, the Information fields are compared. This information may be useful to inhibit copying MAC frames with identical information. This is particularly useful for copying Claim and Beacon frames when new information is present.

The Same INFO signal is asserted when:

1. The current and previous non-Void frames were both MAC frames (not necessarily the same FC value).
2. The first four bytes of the INFO field of the current frame is the same as the first four bytes of the INFO field of the previous non-Void frame.

The size of the address of MAC frames is not checked.

5.9 IMMEDIATE FRAME TRANSMISSION

Immediate requests are used when it is desirable to send frames without first capturing a token. Immediate requests are typically used as part of management processes for error isolation and recovery. Immediate requests are also useful in full duplex applications. Immediate requests are serviced only when the station's Ring__Operational flag is not set (CTSR.ROP = 0).

To transmit an Immediate request, the request must first be queued at the MA__Request Interface. If the Ring is not operational (Ring__Operational flag is not set), the request will be serviced immediately. If the Ring is operational (Ring__Operational flag is set), the request will be serviced when the Ring becomes non-operational. The Ring becomes non-operational as a result of a MAC Reset (Function.MCRST is set to One) or any of the conditions causing the Reset or Recovery Actions are performed.

In addition to servicing an Immediate request from the Tx__Data State, it is also possible to service Immediate requests from the Claim or Beacon State. When transmitting from the Claim or Beacon state, in addition to requesting an Immediate Transmission Service Class, the RQCLM or RQBCN signals (pins 15 and 16) must be asserted to indicate an Immediate Claim or Immediate Beacon request. These requests will only be serviced when in the Claim or Beacon state. Entry to the Beacon State can be forced

5.0 Functional Description (Continued)

by setting bit Function.BCN to One. Entry to the Claim State can be forced by setting bit Function.CLM to One.

While in the Claim or Beacon state, the Ring Engine will transmit internally generated Claim or Beacon frames except when an Immediate Claim or Beacon request is present at the MA_Request Interface, signal RQCLM or RQBCN is asserted, and a frame is ready to be transmitted. At least one internally generated Claim or Beacon frame must be transmitted before an Immediate Claim or Beacon request is serviced. It is possible for the internally generated frame to return before the end of the requested frame has been transmitted. To allow time for the requested frame(s) to be transmitted before leaving the Claim or Beacon state, bit ITR (for Claim) or bit IRR (for Beacon) of the Option Register should be set to One.

While an Immediate request is being serviced (from any state), if bit IRPT of the Option Register is set to One (Inhibit Repeat option), all received frames (except Lower_Claim and My_Beacon frames) are ignored and the Immediate request continues. Lower_Claim and My_Beacon frames can also be ignored by setting bit IRR of the Option Register.

5.10 FULL DUPLEX OPERATION

The BMAC device supports full duplex operation by

1. Suspending the Token Management and Token Recovery protocols (set Option.IRR)
2. Inhibiting the repetition of all PDUs (set Option.IRPT)
3. Using the Immediate Service Class

Frames of any size may be transmitted or received, subject to the minimum length specified in Section 5.4.

5.11 PARITY PROCESSING

The BMAC device contains five data interfaces as shown in Table 5-6.

Through Parity is supported on the internal data paths between any Request interface and any Indicate interface.

Odd Parity is provided every clock on all data outputs and is checked every clock on all data inputs. Parity errors are not propagated through the BMAC device (from the MAC Request and PHY Indication interface to the PHY Request interface or from the PHY Indication interface to the MAC Indication interface). Parity errors are isolated and resolved.

When parity is not used on an Interface, the parity provided by the BMAC device for its outputs may be ignored. For the BMAC device's inputs, the result of the parity check is used only if parity on that Interface is enabled.

Interface parity is enabled by setting the appropriate bit in the Mode register: Mode.CBP for Control Bus Parity, Mode.PIP for PHY Indication parity and Mode.MRP for MAC Request Parity. A Master Reset (Function. MARST) disables parity on all interfaces.

On the PHY Request interface, parity is generated for internally sourced fields (such as the SA or FCS on frames when not using SA or FCS transparency, and internally generated Beacon, Claim and Void frames). In REV 1 of the BMAC device, MRP is passed transparently to PRP for externally sourced fields independent of the value of the Mode.MRP. In all later revisions, correct Odd parity is always generated for PRP. This allows through parity support at the PHY interface even if parity is not used at the MAC interface. This is very desirable since every byte of data that traverses the ring travels across the PHY Interface which is actually part of the ring.

Through parity is not supported in the Control Interface Registers and the Parameter RAM. Parity is generated and stripped at the Control Interface.

Handling Parity Errors

Parity errors are reported in the Exception Status Register when parity on that interface is enabled.

A parity error at the PHY interface (when Mode.PIP is set) is treated as a code violation and ESR.PPE is set. If the parity error occurs in the middle of a PDU (token or frame) reception, the PDU is stripped, a Format Error is signaled (FOERROR) and the Lost Count is incremented.

A parity error at the MAC Interface (when Mode.MRP is set) during a frame transmission from the MAC interface (while TXACK is asserted) causes the frame transmission to be aborted. When a frame is aborted, a Void frame is transmitted to reset every station's TVX timer. A parity error (when enabled) causes ESR.MPE to be set.

A parity error at the Control Interface (when Mode.CBP is set) will cancel the current write access. ESR.CPE is set to indicate that a parity error occurred and ESR.CCE is set to indicate that the write was not performed.

TABLE 5-6. BMAC Device Parity

Interface	Parity On	Parity	Direction
MAC Request Interface	MRD(7:0)	MRP	I
MAC Indication Interface	MID(7:0)	MIP	O
PHY Request Interface	PRD(7:0), PRC	PRP	O
PHY Indication Interface	PID(7:0), PIC	PIP	I
Control Interface	CBD(7:0)	CBP	I/O

6.0 Control Information

The Control Information includes Operation, Event, Status and Parameter Registers that are used to manage and operate the Ring Engine. A processor on the external Control Bus gains access to read and write these parameters via the Control Interface.

The Control Information Address Space is divided into 4 groups as shown in Table 6-1. An information summary is given for each group (see Tables 6-2 through 6-5) followed by a detailed description of all registers.

6.1 CONVENTIONS

When referring to multi-byte fields, byte 0 is always the most significant byte. When referring to bits within a byte, bit (7) is the most significant bit and bit (0) is the least significant bit.

When referring to the contents of a byte, the most significant bit is always referred to first.

When referring to a bit within a byte the notation register_name.bit_name is used. For example, Mode.RUN references the RUN bit in the Mode Register.

6.2 ACCESS RULES

All parameters are accessible in Diagnose Mode. Reserved address space is not accessible in any mode. Certain Status and Parameter Registers are not accessible while in Run mode.

All Control Interface accesses are checked against the current operational mode to determine if the register is currently accessible. If not currently accessible, the Control Bus Interface access is rejected (and reported in an Event Register). This means that all Control Bus Interface accesses complete in a deterministic amount of time.

The Exceptional Status Register can be checked to verify that the operation terminated normally.

TABLE 6-1. Control Information Address Space

Address Range	Description	Read Conditions	Write Conditions
00-07	Operation Registers	Always (Note 2)	Always (Note 2)
08-2F	Event Registers	Always (Note 2)	Always (Cond) (Note 2)
30-3F	Reserved	N/A	N/A
40-7F	MAC Parameters	Stop Mode (Notes 1, 3)	Stop Mode (Notes 1, 3)
80-BF	Counters/Timers	Always	Stop Mode (Note 1)
C0-FF	Reserved	N/A	N/A

Note 1: An attempt to access a currently inaccessible location because of the current mode or because it is in a reserved address space will cause a command error (bit CCE of the Exception Status Register is set to One).

Note 2: Read and write accesses to reserved location within the Operation and Event Address ranges cause a command error (bit CCE of the Exception Status Register is set to One).

Note 3: The MAC Parameter RAM is also accessible when conditions a, b, and c are true. Otherwise accesses will cause a command error (ESR.CEE set to One) and the access will not be performed.

- The MAC Transmitter is in states T0, T1 or T3;
- Bits ITC and IRR of the Option Register are set to One.
- Bits CLM and BCN of the Function Register are not set to One.

Note 4: Reserved bits in registers are always read as 0 and are not writable.

TABLE 6-2. Operation Registers

Addr	Name	D7	D6	D5	D4	D3	D2	D1	D0	Read	Write
0	Mode	DIAG	ILB	RES	RES	PIP	MRP	CBP	RUN	Always	Always
1	Option	ITC	EMIND	IFCS	IRPT	IRR	ITR	ELA	ESA	Always	Always
2	Function	RES	RES	RES	CLM	BCN	MCRST	RES	MARST	Always	Always
3-6	Reserved	RES	RES	RES	RES	RES	RES	RES	RES	N/A	N/A
7	Revision	REV(7-0)								Always	Always

Note: Attempts to access reserved locations will result in Command Rejects (ESR.CCE set to ONE).

6.0 Control Information (Continued)

TABLE 6-3. Event Registers

Addr	Name	D7	D6	D5	D4	D3	D2	D1	D0	Read	Write
8	CMP	CMP(7-0)								Always	Always
9-B	Reserved	RES	RES	RES	RES	RES	RES	RES	RES	N/A	N/A
C	CRS0	RFLG	RS2	RS1	RS0	RES	RTS2	RTS1	RTS0	Always	Ignore
D	Reserved	RES	RES	RES	RES	RES	RES	RES	RES	N/A	N/A
E	CTS0	ROP	TS2	TS1	TS0	TTS3	TTS2	TTS1	TTS0	Always	Ignore
F	Reserved	RES	RES	RES	RES	RES	RES	RES	RES	N/A	N/A
10	RELR0	RES	DUP ADD	PINV	OTR MAC	CLMR	BCNR	RNOP	ROP	Always	Condition
11	REMR0	RES	DUP ADD	PINV	OTR MAC	CLMR	BCNR	RNOP	ROP	Always	Always
12	RELR1	LOCLM	HICLM	MYCLM	RES	RES	RES	MYBCN	OTRBCN	Always	Condition
13	REMR1	LOCLM	HICLM	MYCLM	RES	RES	RES	MYBCN	OTRBCN	Always	Always
14	TELR0	RLVD	TKPASS	TKCAPT	CBERR	DUPTKR	TRTEXP	TVXEXP	ENTRMD	Always	Condition
15	TEMR0	RLVD	TKPASS	TKCAPT	CBERR	DUPTKR	TRTEXP	TVXEXP	ENTRMD	Always	Always
16-17	Reserved	RES	RES	RES	RES	RES	RES	RES	RES	N/A	N/A
18	CILR	RES	TK RCVD	FR TRX	FR NCOP	FR COP	FR LST	FREI	FR RCV	Always	Condition
19	CIMR	RES	TK RCVD	FR TRX	FR NCOP	FR COP	FR LST	FREI	FR RCV	Always	Always
1A-1B	Reserved	RES	RES	RES	RES	RES	RES	RES	RES	N/A	N/A
1C	COLR	RES	TK RCVD	FR TRX	FR NCOP	FR COP	FR LST	FREI	FR RCV	Always	Condition
1D	COMR	RES	TK RCVD	FR TRX	FR NCOP	FR COP	FR LST	FREI	FR RCV	Always	Always
1E-27	Reserved	RES	RES	RES	RES	RES	RES	RES	RES	N/A	N/A
28	IELR	RES	RES	RES	RES	TSM ERR	RSM ERR	RES	MPE	Always	Condition
29-2B	Reserved	RES	RES	RES	RES	RES	RES	RES	RES	N/A	N/A
2C	ESR	CWI	CCE	CPE	RES	RES	RES	RES	PPE	Always	Condition
2D	EMR	ZERO	CCE	CPE	RES	RES	RES	RES	PPE	Always	Always
2E	ICR	ESE	IERR	RES	RES	COE	CIE	TTE	RNG	Always	Ignore
2F	IMR	ESE	IER	RES	RES	COE	CIE	TTE	RNG	Always	Always

Note 1: Attempts to access reserved locations will result in Command Rejects (ESR.CCE set to ONE).

Note 2: Bits in the conditional write registers are only written when the corresponding bit in the Compare Register is equal to the bit to be overwritten and the bit is not changing in that cycle.

6.0 Control Information (Continued)

TABLE 6-4. MAC Parameter RAM

Address	Name	Register Contents
40	MLA0	MLA(47–40)
41	MLA1	MLA(39–32)
42	MLA2	MLA(31–24)
43	MLA3	MLA(23–16)
44	MLA4	MLA(15–8)
45	MLA5	MLA(7–0)
46	MSA0	MSA(15–8)
47	MSA1	MSA(7–0)
48	GLA0	GLA(47–40)
49	GLA1	GLA(39–32)
4A	GLA2	GLA(31–24)
4B	GLA3	GLA(23–16)
4C	GLA4	GLA(15–8)
4D	Reserved	
4E	GSA0	GSA(15–8)
4F	Reserved	
50	TREQ0	TREQ(31–24)
51	TREQ1	TREQ(23–16)
52	TREQ2	TREQ(15–8)
53	TREQ3	TREQ(7–0)
54	TBT0	TBT(31–24)
55	TBT1	TBT(23–16)
56	TBT2	TBT(15–8)
57	TBT3	TBT(7–0)
58	FGM0	FGM(7–0)
59	FGM1	FGM(F–8)
5A–5F	RES	Reserved

Note: The MAC Parameter RAM is accessible in Stop mode and in RUN mode while the MAC Transmitter is in the states T0, T1 or T3; Option.ITC and Option.IRR are set; and Function.BCN and Function.CLM are not set. Otherwise a command reject is given (ESR.CCE) and the Parameter RAM will not be read or written.

TABLE 6-4. MAC Parameter RAM (Continued)

Address	Name	Register Contents
60	PGM10	PGM(87–80)
61	PGM11	PGM(8F–88)
62	PGM12	PGM(97–90)
63	PGM13	PGM(9F–98)
64	PGM14	PGM(A7–A0)
65	PGM15	PGM(AF–A8)
66	PGM16	PGM(B7–B0)
67	PGM17	PGM(BF–B8)
68	PGM18	PGM(C7–C0)
69	PGM19	PGM(CF–C8)
6A	PGM1A	PGM(D7–D0)
6B	PGM1B	PGM(DF–D8)
6C	PGM1C	PGM(E7–E0)
6D	PGM1D	PGM(EF–E8)
6E	PGM1E	PGM(F7–F0)
6F	PGM1F	PGM(FF–F8)
70	PGM0	PGM(7–0)
71	PGM1	PGM(F–8)
72	PGM2	PGM(17–10)
73	PGM3	PGM(1F–18)
74	PGM4	PGM(27–20)
75	PGM5	PGM(2F–28)
76	PGM6	PGM(37–30)
77	PGM7	PGM(3F–38)
78	PGM8	PGM(47–40)
79	PGM9	PGM(4F–48)
7A	PGMA	PGM(57–50)
7B	PGMB	PGM(5F–58)
7C	PGMC	PGM(67–60)
7D	PGMD	PGM(6F–68)
7E	PGME	PGM(77–70)
7F	PGMF	PGM(7F–78)

6.0 Control Information (Continued)

TABLE 6-5. MAC Counters and Timer Thresholds

Address	Name	Register Contents
80–86	Reserved	
87	THSH1	Null(7–4), THSH1(3–0)
88–92	Reserved	
93	TMAX	Null(7–4), TMAX(3–0)
94–96	Reserved	
97	TVX	Null(7–4), TVX(3–0)
98	TNEG0	TNEG(31–24)
99	TNEG1	TNEG(23–16)
9A	TNEG2	TNEG(15–8)
9B	TNEG3	TNEG(7–0)
9C–9E	Reserved	
9F	LTCT	Null(7–4), LTCT(3–0)
A0	FRCT0	Zero(31–24)
A1	FRCT1	Null(7–4), FRCT(19–16)
A2	FRCT2	FRCT(15–8)
A3	FRCT3	FRCT(7–0)
A4	EICT0	Zero(31–24)
A5	EICT1	Null(7–4), EICT(19–16)
A6	EICT2	EICT(15–8)
A7	EICT3	EICT(7–0)
A8	LFCT0	Zero(31–24)
A9	LFCT0	Null(7–4), LFCT(19–16)
AA	LFCT1	LFCT(15–8)
AB	LFCT2	LFCT(7–0)
AC	FCCT0	Zero(31–24)
AD	FCCT1	Null(7–4), FCCT(19–16)
AE	FCCT2	FCCT(15–8)
AF	FCCT3	FCCT(7–0)

TABLE 6-5. MAC Counters and Timer Thresholds
(Continued)

Address	Name	Register Contents
B0	FNCT0	Zero(31–24)
B1	FNCT1	Null(7–4), FNCT(19–16)
B2	FNCT2	FNCT(15–8)
B3	FNCT3	FNCT(7–0)
B4	FTCT0	Zero(31–24)
B5	FTCT1	Null(7–4), FTCT(19–16)
B6	FTCT2	FTCT(15–8)
B7	FTCT3	FTCT(7–0)
B8	TKCT0	Zero(31–24)
B9	TKCT1	Null(7–4), TKCT(19–16)
BA	TKCT2	TKCT(15–8)
BB	TKCT3	TKCT(7–0)
BC	RLCT0	Zero(31–24)
BD	RLCT1	Null(7–4), RLCT(19–16)
BE	RLCT2	RLCT(15–8)
BF	RLCT3	RLCT(7–0)

Note: The MAC event counters and timer thresholds are always readable, and are writable in Stop mode.

Note: Null(7–4) indicates that these bits are forced to zero on reads, and are ignored on writes.

Note: The value obtained on reads from reserved locations is not specified.

The Event Counters are 20-bit counters and are read through three control accesses. In order to guarantee a consistent snapshot, whenever byte 3 of an event counter is read, byte 1 and byte 2 of the counters are loaded into a holding register. Byte 1 and byte 2 may then be read from the holding register. A single holding register is shared by all of the counters but (for convenience) is accessible at several places within the address space. Consistent readings across counters can be accomplished using the Counter Increment Latch Register (CILR).

The Event Counters are not reset as a result of a Master Reset. This may be done by either reading the counters out and keeping track relative to the initial value read, or by writing a value to all of the counters in stop mode. The counters may be written in any order. With some exceptions, interrupts are available when the counters increment or wraparound.

6.3 OPERATION REGISTERS

The Operation Registers are used to control the operation of the BMAC device. The Operation Registers include the following registers.

- Mode Register (Mode)
- Option Register (Option)
- Function Register (Function)
- Revision Register (REV)

6.0 Control Information (Continued)

Mode Register (Mode)

The Mode Register (Mode) contains the current mode of the BMAC device.

ACCESS RULES

Address	Read	Write
00h	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
DIAG	ILB	RES	RES	PIP	MRP	CBP	RUN

Bit	Symbol	Description
D0	RUN	<p>RUN/Stop:</p> <p>0: Stop Mode</p> <p>All state machines return to and remain in their zero state. All counters and timers are disabled. The Ring Engine transmits Idle symbols.</p> <p>1: Run Mode. Must be in Run Mode to achieve an operational Ring.</p>
D1	CBP	<p>Control Bus Parity: Enables Odd Parity checking on the Control Bus Data pins (CBD7–0) during write accesses.</p> <p>If a parity error occurs, the CPE bit of the Exception Status Register is set to One and an interrupt is generated. The write data will not be deposited in the register. Parity is always generated on CBD7–0 during read accesses</p>
D2	MRP	<p>MAC Request Parity: Enables Odd Parity checking on the MAC Request Data pins (MRD7–0). A parity error causes the transmission to be aborted. In REV 1 of the BMAC device MIP is always passed transparently from PIP. In all later revisions correct Odd parity is always generated on MIP.</p>
D3	PIP	<p>PHY Indicate Parity: Enables Odd Parity checking on the PHY Indicate Data pins (PID7–0). Parity errors are treated as code violations and cause the byte in error to be replaced with Idle symbols. In REV 1 of the BMAC device Parity is passed transparently between MRP and PRP during transmission. When repeating, Parity is passed transparently from PIP to PRP. Odd Parity is generated for all internally generated fields. In all later revisions correct Odd Parity is always generated on the PHY Request Data pins (PRD7–0).</p>
D4–5	RES	Reserved
D6	ILB	<p>Internal Loopback: Enables the internal loopback that connects PRP, PRC, and PRD7–0 to PIP, PIC, and PID7–0 respectively. When enabled, the PHY Indicate Interface is ignored.</p> <p>Since the Ring Engine Transmitter and Receiver work as independent processes, a ring can be made operational in this mode, albeit consisting only of a single MAC. With an operational ring many diagnostic tests can be performed to test out MAC level and system level diagnostics including: the Beacon Process, the Claim Process, Ring Engine frame generation, token timers, event counters, transmission options, test of event detection capabilities, test of addressing modes, test of state machine sequencing options, etc. In addition, a large portion of the system interface logic can be tested, such as full duplex transmission to self within the limits of the system interface performance constraints, status handling and generation, etc.</p> <p>The same system tests can also be performed at different levels of loopback including through the various paths within a station: through the PMD interface of the PLAYER device, and through the CRD device. System level tests can also be performed through the ring during normal operation.</p>
D7	DIAG	<p>Diagnose Mode: Enables access to all BMAC device registers. When set, interoperability is not guaranteed. This bit should only be set when the BMAC device is not inserted in a ring.</p> <p>In diagnose mode, should an internal error occur the Current Receive and Transmit Status Registers are frozen with the errored state until the internal state machine error condition is cleared (IELR.RSMERR and/or IELR.TSMERR).</p>

6.0 Control Information (Continued)

Option Register (Option)

The Ring Engine supports several options. These options are typically static during operation but may be altered during operation. This register is initialized to Zero after a master reset.

ACCESS RULES

Address	Read	Write
01h	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
ITC	EMIND	IFCS	IRPT	IRR	ITR	ELA	ESA

Bit	Symbol	Description
D0	ESA	<p>Enable Short Addressing: Enables the setting of A__Flag on matches of received Short Destination Addresses with MSA. Enables the setting of M__Flag and stripping on matches of received Short Source Addresses with MSA.</p> <p>Permits transmission of frames with Short Addresses. Frames with Short Addresses can be transmitted when Short Addressing is not enabled if the SA Transparency option is selected.</p> <p>Void frames are sent with the Short Address if ESA is set to One. If ESA is Zero and ELA is One, Void frames are sent with the Long Address.</p> <p>When both the ESA and ELA bits are Zero, the ring is effectively interrupted at this station. The token capture process and Error Recovery logic are suspended and no frames are repeated. Immediate requests are serviced if the SA Transparency option is selected.</p>
D1	ELA	<p>Enable Long Addressing: Enables the setting of A__Flag on matches of received Long Destination Addresses with MLA. Enables the setting of M__Flag and stripping on matches of received Long Source Address with MLA.</p> <p>Permits transmission of frames with Long Addresses. Frames with long addresses can be transmitted when long addressing is not enabled if the SA transparency option is selected.</p> <p>Claim and Beacon frames are sent with the Long Address if ELA is One. If ELA is Zero and ESA is One, Claim and Beacon frames are sent with the Short Address.</p> <p>When both ESA and ELA are Zero, the ring is effectively interrupted at this station. The token capture process and Error Recovery logic are suspended and no frames are repeated. Immediate requests are serviced if the SA Transparency option is selected.</p>
D2	ITR	<p>Inhibit Token Release: When bit ITR is set to One, the station will not issue a token after winning the Claim Process. The station remains in the Claim state while the station's Claim frames are returning to the station and it has won the Claim Process. At this point the station is in control of the ring as long as no Higher__Claim or Beacon frames are received.</p> <p>While in control of the ring, the station may transmit special Claim or Management frames for a variety of implementation specific purposes. For example, the station might send out a Claim frame with a unique identifier to make sure that another station with its address and TREQ is not also Claiming.</p>
D3	IRR	<p>Inhibit Recovery Required: When bit IRR is set to One, the Ring Engine does not take the transitions into the Claim state (T4). This option inhibits all the recovery required transitions as defined in the FDDI MAC Standard. This bit does not inhibit entry to the Claim state on a Claim Request generated at the MAC Request Interface via the Function Register.</p> <p>This option can be used to guarantee that implementation specific Beacon frames will be transmitted from the Beacon state. It is also useful in systems where Local Address Administration is used, to prohibit stations with the Null Address (or any address) from Claiming. The option could also be used to enable the use of the Ring Engine in full duplex applications (in conjunction with the Inhibit Repeat option) to disable the recovery timers.</p>

6.0 Control Information (Continued)

Option Register (Continued)

Bit	Symbol	Description
D4	IRPT	<p>Inhibit Repeat: When enabled,</p> <ol style="list-style-type: none"> the Ring Engine cannot enter the Transmitter Repeat and Issue__Token states. This causes all received PDUs to be stripped and prevents tokens from being issued. Void frames are not transmitted during a service opportunity. Idle to Repeat transition is inhibited and all received tokens and MAC frames except Lower__Claim and My__Beacon frames are ignored (Lower__Claim and My__Beacon frames may be ignored by setting Option.IRR). <p>When the ring is operational, enabling this option causes the Reset actions to occur upon the completion of the Service Opportunity, if any. When the ring is not operational, Immediate Requests are serviced and continue to completion.</p> <p>The Inhibit Repeat option can be used to scrub the ring for a period longer than the Ring Latency. The option is also useful in full duplex applications.</p>
D5	IFCS	<p>Implementer FCS: Enables use of the standard CRC as the FCS on Implementer frames (FC.FF = 10). When enabled, Implementer frames are treated like all other frames. When Implementer frames are received with bad FCS and Er = R, the E Indicator is transmitted as S and EICT is incremented.</p> <p>On Implementer frames, the Standard does not mandate the setting of the E Indicator on the result of the FCS check. This allows Implementers to use alternate Frame Check Sequences aside from the standard 32-bit CRC. Implementers may also choose not to use any FCS in applications such as packet voice.</p> <p>If other stations in the ring are using Implementer frames with a non-standard FCS, if used, this option may cause an interoperability problem.</p>
D6	EMIND	<p>External Matching Indicators: Enables the setting of the transmitted A Indicator (Ax) as an S symbol when the EA pin is set. Also enables the setting of the transmitted C Indicator (Cx) as an S symbol when the VCOPY pin is set if the A Indicator is set as a result of an external match. The Copied/Not Copied Frame Counters are also incremented as a result of external comparisons when this option is enabled.</p>
D7	ITC	<p>Inhibit Token Capture: When enabled, the Ring Engine is prohibited from transmitting any (more) frames. This option prohibits entry to the Transmit Void and Data states from the Idle state, and causes exit from the Data state after the current frame has been transmitted.</p> <p>When enabled, it is still possible to perform Immediate transmissions from the transmitter Claim and Beacon states, but not from the Data state.</p> <p>This option can be used to temporarily block normal data service. It can also be used in conjunction with the Inhibit Recovery Required option to permit access via the Control Interface to the MAC Parameter RAM during MAC operation.</p>

6.0 Control Information (Continued)

Function Register (Function)

The Ring Engine performs the MAC Reset, Claim Request, and Beacon Request using the Function Register. The Register is initialized to Zero after a master reset. A function is performed by setting the appropriate bit to One. When the function is complete, the bit is cleared by the Ring Engine.

ACCESS RULES

Address	Read	Write
02h	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	CLM	BCN	MCRST	RES	MARST

Bit	Symbol	Description
D0	MARST	<p>Master Reset: Produces the functions of an SM_CONTROL(MAC Reset) as specified by the FDDI MAC Standard. Sets all internal state machines and registers to known values.</p> <p>Master Reset causes the MCRST bit to be set. It also clears the Mode, Option, Event and Mask Registers. The Timers are set to their defaults. The Event Counters are not cleared.</p> <p>When the Master Reset function is complete, MARST is cleared. At this time, all bits in the Function Register should be Zero.</p>
D1	RES	Reserved
D2	MCRST	<p>MAC Reset: Forces the Receiver to state R0 (Listen) and the Transmitter to state T0 (Idle). TNEG (Registers 98–9B) is not loaded with TMAX (this operation can be performed as part of the MAC Reset Request actions by writing to TNEG before the MAC Reset is initiated).</p> <p>MCRST takes precedence over bits D3 (BCN) and D4 (CLM), but does not clear these bits.</p> <p>A MAC Reset that occurs while a frame is being transmitted will cause the frame to be aborted. Frames without the Frame Status are not transmitted by the Ring Engine. Whenever the byte with the Ending Delimiter is transmitted, valid frame status is transmitted as well. If a MAC Reset occurs during the byte where the Ending Delimiter and E Indicator should be transmitted, it will not be transmitted. If a MAC Reset occurs on the cycle where the A and C Indicators are transmitted, they will still be transmitted.</p>
D3	BCN	<p>Beacon Request: Produces the functions of an SM_CONTROL.request (Beacon) as required by the FDDI MAC Standard. The Ring Engine Transmitter is forced to enter the Beacon State. Beacon frames are then transmitted until the Beacon Process completes. The Beacon Process will not complete if Option.IRR = 1.</p> <p>Beacon frames are generated by the Ring Engine unless an Immediate Beacon Request is present at the MAC Request Interface and a frame is ready to be transmitted. Even with an External Immediate Beacon Request the Ring Engine transmits at least one Beacon frame before the Beacon frames from the MAC Request Interface are transmitted.</p> <p>If an external Beacon frame is to be transmitted, the Beacon frame should first be set up, then the request should be given to the MAC Request Interface and then bit BCN should be set to One.</p> <p>Writing to this bit also sets bit D2 (MCRST). This bit is cleared on entry to the Beacon state. If both bits D3 (BCN) and D4 (CLM) are set, bit D3 takes precedence.</p>
D4	CLM	<p>Claim Request: Produces the functions equivalent to an SM_CONTROL.request (Claim) and causes entry to the Claim State. The Ring Engine Transmitter is forced to enter the Claim State unless the Transmitter is in the Beacon State or bit BCN is set to One. Claim frames are then transmitted until the Claim Process completes. The Claim Process will not complete if Option.ITR = 1.</p> <p>A Claim Request is honored immediately from any state except the Beacon state. It is honored in the Beacon state when a My_Beacon returns. Claim requests are honored even when Option.IRR = 1.</p> <p>Claim frames are generated by the Ring Engine unless an Immediate Claim Request is available at the MAC Request Interface. Even with an Immediate Claim Request at the interface, the Ring Engine transmits at least one Claim frame before the Claim frames from the MAC Request Interface are transmitted.</p> <p>If an external Claim frames is to be transmitted, the Claim frame should first be set up, then the request should be given to the MAC Request Interface before the CLM bit is set to One.</p> <p>The Claim bit is reset upon entry to the Claim or Beacon state.</p>
D5–7	RES	Reserved

6.0 Control Information (Continued)

Revision Register (Rev)

The Revision Register (Rev) contains the revision number of the BMAC device.

ACCESS RULES

Address	Read	Write
02h	Always	Data Ignored

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0

Bit	Symbol	Description
D0–7	REV	<p>Revision Number: Bits D0–7 contain the version ID of the BMAC device. Software should consult this register for any software-specific issues related to the current version.</p> <p>00h reserved 01h Initial Release 02h First Revision</p> <ul style="list-style-type: none"> • Programmable Group Address Modification • Not copied count does not increment on reception of NSA frames with Ar=S • Detection of Idle on reception of nl • Generation of ODD Parity at all times • Reset of Latency Count on initiation of new measurement

6.0 Control Information (Continued)

6.4 EVENT REGISTERS

The Event Registers record the occurrence of events or series of events. Events are recorded and contribute to generating the Interrupt signal. There is a two-level hierarchy in generating this signal.

At the first level of the hierarchy, events are recorded as bits in the Latch Registers (e.g., Ring Event Latch Registers, Counter Increment Latch Register). Each Latch Register has a corresponding Mask Register (e.g., Ring Event Mask Registers, Counter Increment Mask Register). When a bit in the Latch Register is set to One and its corresponding bit in the Mask Register is also set to One, a bit in the Interrupt Condition Register is set to One.

At the second level of the hierarchy, if a bit in the Interrupt Condition Register is set to One and the corresponding bit in the Interrupt Mask Register is set to One, the Interrupt signal is asserted.

Bits in Conditional Write Registers (e.g., Ring Event Latch Registers) are only written when the corresponding bits in the Compare Register are equal to bits to be overwritten.

Servicing Interrupts

In the process of servicing an interrupt, a Management Entity may use one or both levels of condition masks to disable new interrupts while one is being serviced. Soon after the Management Entity has processed the interrupt to some extent, it is ready to rearm the interrupt in order to be notified of the next condition.

The Interrupt Control Register always contains the merged output of the masked Condition Registers as shown in *Figure 6-1*. It is only possible to remove a condition by setting the corresponding Condition Latch Register bit to Zero. By storing the events on-chip, and having the ability to selectively set bits to Zero, the need for the software to maintain a copy of the Event Registers is alleviated.

To prevent the overwriting and consequent missing of events, an interlock mechanism is used. In the period between the Read of a Condition Latch Register, and the corresponding Write to reset the condition, additional events can occur.

In order to prevent software from overwriting bits which have changed since the last read and losing interrupt events a conditional write mechanism is employed. Only bits

that have not changed since the last read can be written to a new value.

Whenever a Condition Latch Register is read, its contents are stored in the Compare Register. Each bit of the Compare Register is compared with the current contents of the Register that is to be written. Writing a bit with a new value to a Condition Register is only possible when the corresponding bit in the Compare Register matches the bit in the Condition Register. For any bit that has not changed, the new value of the bit is written into the Register. For any bit that has changed, the writing of the bit is inhibited. The fact that an attempt was made to change a modified bit in the Register is latched in the Condition Write Inhibit bit in the Exception Status Register (ESR.CWI).

In the BMAC device, the Compare Register is shared by all of the Condition Latch Registers and always reflects the most recent read of one of these registers. (In the DP83251/5 PLAYER Device, there is a Compare Register for every Event Register.) For the cases where more than one register must be read before writing a new value, the software may write the Compare Register with the most recently read value before writing the register again. Alternatively, the register may be read again before being written.

The Event Registers include the following registers as:

- Compare Register (CMP)
- Current Receiver Status Register (CRS0)
- Current Transmitter Status Register (CTS0)
- Ring Event Latch Registers (RELR0-1)
- Ring Event Mask Registers (REMR0-1)
- Token and Timer Event Latch Register (TELR0)
- Token and Timer Event Mask Register (TEMR0)
- Counter Increment Latch Register (CILR)
- Counter Increment Mask Register (CIMR)
- Counter Overflow Latch Register (COLR)
- Counter Overflow Mask Register (COMR)
- Internal Event Latch Register (IELR)
- Exception Status Register (ESR)
- Exception Mask Register (EMR)
- Interrupt Condition Register (ICR)
- Interrupt Mask Register (IMR)

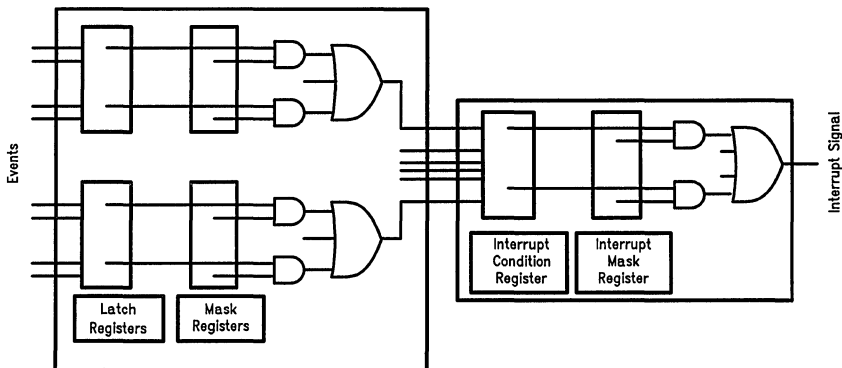


FIGURE 6-1. Event Registers Hierarchy

TL/F/10387-7

6.0 Control Information (Continued)

Compare Register (CMP)

The Compare Register (CMP) is written with the contents of a conditional event latch registers when it is read. The Compare Register may also be written to directly. During a write to any of the conditional write registers, the contents of the Compare Register (CMP) is compared with bits D0–7 of the accessed register. Only bits for which the comparison matches can be written to a new value.

ACCESS RULES

Address	Read	Write
08h	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
CMP7	CMP6	CMP5	CMP4	CMP3	CMP2	CMP1	CMP0

6.0 Control Information (Continued)

Current Receiver Status Register (CRS0)

The Current Receiver Status Register (CRS0) records the status of the Receiver state machine. It is continuously updated. It remains stable when accessed.

When in Diagnose Mode, this register is frozen on an internal error until the internal error event is cleared by resetting the RSMERR bit in the Internal Event Latch Register.

ACCESS RULES

Address	Read	Write
0Ch	Always	Data Ignored

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RFLG	RS2	RS1	RS0	RES	RTS2	RTS1	RTS0

Bit	Symbol	Description																																				
D0-2	RTS(0-2)	<p>Receive Timing State: RTS(0-2) represent the current state of the Receiver Timing state machine. The encoding is shown below.</p> <table border="1"> <thead> <tr> <th>RTS2</th> <th>RTS1</th> <th>RTS0</th> <th>Receive Timing State</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Await_SD</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Check_FC</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Check_SA</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Check_DA</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Check_INFO</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Check_MAC</td> </tr> <tr> <td>1</td> <td>1</td> <td>x</td> <td>Reserved</td> </tr> </tbody> </table>	RTS2	RTS1	RTS0	Receive Timing State	0	0	1	Await_SD	0	0	1	Check_FC	0	1	0	Check_SA	0	1	1	Check_DA	1	0	0	Check_INFO	1	0	1	Check_MAC	1	1	x	Reserved				
RTS2	RTS1	RTS0	Receive Timing State																																			
0	0	1	Await_SD																																			
0	0	1	Check_FC																																			
0	1	0	Check_SA																																			
0	1	1	Check_DA																																			
1	0	0	Check_INFO																																			
1	0	1	Check_MAC																																			
1	1	x	Reserved																																			
D3	RES	Reserved																																				
D4-6	RS(0-2)	<p>Receive State: RS(0-2) represent the current state of the Receive state machine that implements the ANSI standard MAC Receive Functions. The encoding is shown below.</p> <table border="1"> <thead> <tr> <th>RS2</th> <th>RS1</th> <th>RS0</th> <th>Receive State</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Listen</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Await_SD</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>RC_FR_CTRL (Receive FC)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>RC_FR_BODY (Rec FR Body)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>RC_FR_STATUS (A & C Ind)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>CHECK_TOKEN (Check Token)</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>RC_FR_STATUS (Optional Ind)</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Reserved</td> </tr> </tbody> </table>	RS2	RS1	RS0	Receive State	0	0	0	Listen	0	0	1	Await_SD	0	1	0	RC_FR_CTRL (Receive FC)	0	1	1	RC_FR_BODY (Rec FR Body)	1	0	0	RC_FR_STATUS (A & C Ind)	1	0	1	CHECK_TOKEN (Check Token)	1	1	0	RC_FR_STATUS (Optional Ind)	1	1	1	Reserved
RS2	RS1	RS0	Receive State																																			
0	0	0	Listen																																			
0	0	1	Await_SD																																			
0	1	0	RC_FR_CTRL (Receive FC)																																			
0	1	1	RC_FR_BODY (Rec FR Body)																																			
1	0	0	RC_FR_STATUS (A & C Ind)																																			
1	0	1	CHECK_TOKEN (Check Token)																																			
1	1	0	RC_FR_STATUS (Optional Ind)																																			
1	1	1	Reserved																																			
D7	RFLG	<p>R_Flag: Current value of the Restricted Flag. When not holding the token indicates the type of the last valid token received. When holding the token indicates the type of token that will be issued at the end of the current service opportunity.</p> <p>0: Non-restricted 1: Restricted</p>																																				

6.0 Control Information (Continued)

Current Transmitter Status Register (CTS0)

The Current Transmitter Status Register (CTS0) records the status of the Transmitter state machine. It is continuously updated. It remains stable when accessed. When in Diagnose Mode, this register is frozen on an internal error until the internal error event is cleared by resetting the TSMERR bit of the Internal Event Latch Register.

ACCESS RULES

Address	Read	Write
0Eh	Always	Data Ignored

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
ROP	TS2	TS1	TS0	TTS3	TTS2	TTS1	TTS0

Bit	Symbol	Description																																																							
D0-3	TTS(0-3)	<p>TRANSMIT TIMING STATE: TTS(0-3) represent the current state of the Transmitter Timing state machine. The encoding is shown below.</p> <table border="1"> <thead> <tr> <th>TTS3</th> <th>TTS2</th> <th>TTS1</th> <th>TTS0</th> <th>Transmit Timing State</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Idle</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Transmit Preamble</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Wait for Data (FIFO)</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>Transmit SD & FC Fields</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Transmit DA</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>Transmit SA</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>Transmit INFO</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>Transmit FCS</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>Transmit ED & FS</td></tr> <tr><td></td><td>9h-Fh</td><td></td><td></td><td>Reserved</td></tr> </tbody> </table>	TTS3	TTS2	TTS1	TTS0	Transmit Timing State	0	0	0	0	Idle	0	0	0	1	Transmit Preamble	0	0	1	0	Wait for Data (FIFO)	0	0	1	1	Transmit SD & FC Fields	0	1	0	0	Transmit DA	0	1	0	1	Transmit SA	0	1	1	0	Transmit INFO	0	1	1	1	Transmit FCS	1	0	0	0	Transmit ED & FS		9h-Fh			Reserved
TTS3	TTS2	TTS1	TTS0	Transmit Timing State																																																					
0	0	0	0	Idle																																																					
0	0	0	1	Transmit Preamble																																																					
0	0	1	0	Wait for Data (FIFO)																																																					
0	0	1	1	Transmit SD & FC Fields																																																					
0	1	0	0	Transmit DA																																																					
0	1	0	1	Transmit SA																																																					
0	1	1	0	Transmit INFO																																																					
0	1	1	1	Transmit FCS																																																					
1	0	0	0	Transmit ED & FS																																																					
	9h-Fh			Reserved																																																					
D4-6	TS(0-2)	<p>Transmit State: TS(0-2) represent the current state of the Transmit state machine that implements the ANSI standard MAC Transmit Functions. The encoding is shown below.</p> <table border="1"> <thead> <tr> <th>TS2</th> <th>TS1</th> <th>TS0</th> <th>Transmit State</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>Idle</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>Repeat</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>Data</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>Issue Token</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>Claim</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>Beacon</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>Reserved</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>Void</td></tr> </tbody> </table>	TS2	TS1	TS0	Transmit State	0	0	0	Idle	0	0	1	Repeat	0	1	0	Data	0	1	1	Issue Token	1	0	0	Claim	1	0	1	Beacon	1	1	0	Reserved	1	1	1	Void																			
TS2	TS1	TS0	Transmit State																																																						
0	0	0	Idle																																																						
0	0	1	Repeat																																																						
0	1	0	Data																																																						
0	1	1	Issue Token																																																						
1	0	0	Claim																																																						
1	0	1	Beacon																																																						
1	1	0	Reserved																																																						
1	1	1	Void																																																						
D7	ROP	Ring Operational Flag: Indicates the current value of the local Ring Operational Flag.																																																							

6.0 Control Information (Continued)

Ring Event Latch Register (RELR0)

The Ring Event Latch Register 0 (RELR0) captures conditions that occur on the Ring including the receipt of Beacon and Claim frames, transitions in the Ring Operational flag, and the receipt of duplicate addresses. Each bit may be masked via the Ring Event Mask Register 0 (REMR0).

ACCESS RULES

Address	Read	Write
10h	Always	Condition

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RES	DUPADD	PINV	OTRMAC	CLMR	BCNR	RNOP	ROP

Bit	Symbol	Description
D0	ROP	Ring Operational Set: Is set when the Local Ring Operational flag transitions from 0 to 1.
D1	RNOP	Ring Non-Operational Set: Is set when the Local Ring Operational flag transitions from 1 to 0.
D2	BCNR	Beacon Frame Received: Indicates that a valid Beacon frame was received. When set, restricted and synchronous requests are not serviced. The type of Beacon frame received is given in Register RELR1.
D3	CLMR	Claim Frame Received: Indicates that a valid Claim frame was received. When set, restricted requests are not serviced. The type of Claim frame received is given in Register RELR1.
D4	OTRMAC	Other MAC Frame Received: Indicates that a MAC frame other than a Beacon or Claim frame was received. When set, restricted requests are not serviced.
D5	PINV	PHY_Invalid Received: Indicates that a PHY_Invalid was received. This could be the result of a PLAYER device Reset operation. PHY_Invalid causes the MAC Receiver to enter state R0 (Listen).
D6	DUPADD	Duplicate Address Received: Indicates that a valid individual frame addressed to this station was received with the A indicator set. This could be caused by either a MAC using the same address (duplicate address) or a strip error at the Source (the frame was received twice).
D7	RES	Reserved

6.0 Control Information (Continued)

Ring Event Mask Register 0 (REMR0)

The Ring Event Mask Register 0 (REMR0) is used to mask bits in Register RELR0. If a bit in Register REMR0 is set to One, the corresponding bit in Register RELR0 will be applied to the Interrupt Condition Register, which can then be used to generate an interrupt.

ACCESS RULES

Address	Read	Write
11h	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RES	DUPADD	PINV	OTRMAC	CLMR	BCNR	RNOP	ROP

Bit	Symbol	Description
D0	ROP	Ring Operational Mask: This bit is used to mask RELR0.ROP.
D1	RNOP	Ring Non-Operational Mask: This bit is used to mask RELR0.RNOP.
D2	BCNR	Beacon Frame Mask: This bit is used to mask RELR0.BCNR.
D3	CLMR	Claim Frame Mask: This bit is used to mask RELR0.CLMR.
D4	OTRMAC	Other MAC Frame Mask: This bit is used to mask RELR0.OTRMAC.
D5	PINV	PHY__Invalid Mask: This bit is used to mask RELR0.PINV.
D6	DUPADD	Duplicate Address Mask: This bit is used to mask RELR0.DUPADD.
D7	RES	Reserved

6.0 Control Information (Continued)

Ring Event Latch Register 1 (REL1)

The Ring Event Latch Register 1 (REL1) captures the progress of the Beacon and Claim Processes. During the Beacon Process, it records reception of an Other__Beacon or a My__Beacon. It also identifies Claim frames as Higher, Lower, or My Claim. Each bit may be masked via the Ring Event Mask Register 1 (REMR1).

ACCESS RULES

Address	Read	Write
12h	Always	Condition

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
LOCLM	HICLM	MYCLM	RES	RES	RES	MYBCN	OTRBCN

Bit	Symbol	Description
D0	OTRBCN	Other__Beacon Received: Indicates that an Other__Beacon frame was received.
D1	MYBCN	My__Beacon Received: Indicates that a My__Beacon frame was received.
D2-4	RES	Reserved
D5	MYCLM	My__Claim Received: Indicates that a My__Claim frame was received. (This includes the comparison between the T__Bid__Rec and TREQ as specified in the standard).
D6	HICLM	Higher__Claim Received: Indicates that a Higher__Claim frame was received.
D7	LOCLM	Lower__Claim Received: Indicates that a Lower__Claim frame was received.

6.0 Control Information (Continued)

Ring Event Mask Register 1 (REMR1)

The Ring Event Mask Register 1 (REMR1) is used to mask bits in Register RELR1. If a bit in Register REMR1 is set to One, the corresponding bit in Register RELR1 will be applied to the Interrupt Condition Register, which can then be used to generate an interrupt to the CPU.

All bits in this register are set to Zero upon reset.

ACCESS RULES

Address	Read	Write
13h	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
LOCLM	HICLM	MYCLM	RES	RES	RES	MYBCN	OTRBCN

Bit	Symbol	Description
D0	OTRBCN	Other__Beacon Mask: This bit is used to mask RELR1.OTRBCN.
D1	MYBCN	My__Beacon Mask: This bit is used to mask RELR1.MYBCN.
D2-4	RES	Reserved
D5	MYCLM	My__Claim Mask: This bit is used to mask RELR1.MYCLM.
D6	HICLM	Higher__Claim Mask: This bit is used to mask RELR1.HICLM.
D7	LOCLM	Lower__Claim Mask: This bit is used to mask RELR1.LOCLM.

6.0 Control Information (Continued)

Token and Timer Event Latch Register 0 (TELRO)

The Token and Timer Event Latch Register 0 (TELRO) informs software of expirations of the Token Rotation Timer (TRT) and Valid Transmission Timer (TVX). The TELRO Register also reports token events such as duplicate token detection, restricted token reception, and general token capture and release. The completion of the Ring Latency measurement is also indicated in the TELRO Register. Each bit may be masked via the Token and Timer Event Mask Register (TEMRO).

ACCESS RULES

Address	Read	Write
14h	Always	Condition

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RLVD	TKPASS	TKCAPT	CBERR	DUPTKR	TRTEXP	TVXEXP	ENTRMD

Bit	Symbol	Description
D0	ENTRMD	Enter Restricted Mode: Indicates that a Restricted Token was received and that the R_Flag transitioned from 0 to 1.
D1	TVXEXP	TVX Expired: Indicates that a valid frame or token was not received in TVX time.
D2	TRTEXP	TRT Expired: Indicates that a valid token was not received within 2*TNEG.
D3	DUPTKR	Duplicate Token Received: Indicates that a valid token was received while the transmitter was in state T2 or T3.
D4	CBERR	Claim and/or Beacon Error: Indicates that the Claim and/or Beacon Process failed because TRT expired while the Transmitter was in state T4 or T5.
D5	TKCAPT	Token Captured: Indicates that a token has been captured.
D6	TKPASS	Token Passed: Indicates that a valid token has been passed (without capturing it) or has been issued after a service opportunity.
D7	RLVD	<p>Ring Latency Valid:</p> <p>0: This bit is set to Zero to request a new latency value from the Ring Engine. In Rev 01 and all future Revisions, the Ring Latency count is set to zero before each measurement.</p> <p>1: This bit is set to One when the Ring Latency measurement is complete.</p> <p>This bit is written unconditionally and is not protected by the Compare Register.</p>

6.0 Control Information (Continued)

Token and Timer Event Mask Register 0 (TEMR0)

The Token and Timer Event Mask Register 0 (TEMR0) is used to mask bits in Register TELR0. If a bit in Register TEMR0 is set to One, the corresponding bit in Register TELR will be applied to the Interrupt Condition Register, which can then be used to generate an interrupt.

All bits in this register are set to Zero upon reset.

ACCESS RULES

Address	Read	Write
15h	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RLVD	TKPASS	TKCAPT	CBERR	DUPTOK	TRTEXP	TVXEXP	ENTRMD

Bit	Symbol	Description
D0	ENTRMD	Enter Restricted Mode Mask: This bit is used to mask TELR0.ENTRMD.
D1	TVXEXP	TVX Expired Mask: This bit is used to mask TELR0.TVXEXP.
D2	TRTEXP	TRT Expired and Set Late__Flag Mask: This bit is used to mask TELR0.TRTEXP.
D3	DUPTOK	Duplicated Token Received Mask: This bit is used to mask TELR0.DUPTOK.
D4	CBERR	Claim/Beacon Error Mask: This bit is used to mask TELR0.CBERR.
D5	TKCAPT	Token Captured Mask: This bit is used to mask TELR0.TKCAPT.
D6	TKPASS	Token Passed Mask: This bit is used to mask TELR0.TKPASS.
D7	RLVD	Ring Latency Valid Mask: This bit is used to mask TELR0.RLVD.

6.0 Control Information (Continued)

Counter Increment Latch Register (CILR)

The Counter Increment Latch Register (CILR) records the occurrence of any increment to the Event Counters. Each bit corresponds to a counter and is set when the corresponding counter is incremented. Each bit may be masked via the Counter Increment Mask Register (CIMR).

ACCESS RULES

Address	Read	Write
18h	Always	Condition

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RES	TKRCVD	FRTRX	FRNCOP	FRCOP	FRLST	FREI	FRRCV

Bit	Symbol	Description
D0	FRRCV	Frame Received: Is set when the Frame Received Counter (FRCT) is incremented, indicating the reception of a frame.
D1	FREI	Frame Error Isolated: Is set when the Error Isolated Counter (EICT) is incremented, indicating an error has been isolated.
D2	FRLST	Frame Lost Isolated: Is set when the Lost Frame Counter (LFCT) is incremented, indicating a format error has been detected in the frame.
D3	FRCOP	Frame Copied: Is set when the Frame Copied Counter (FCCT) is incremented, indicating a frame has been copied.
D4	FRNCOP	Frame Not Copied: Is set when the Frame Not Copied Counter (FNCT) is incremented, indicating a frame could not be copied.
D5	FRTRX	Frame Transmitted: Is set when the Frame Transmitted Counter (FTCT) is incremented, indicating a frame has been transmitted.
D6	TKRCVD	Token Received: Is set when the Token Received Counter (TKCT) is incremented, indicating that a token has been received.
D7	RES	Reserved

6.0 Control Information (Continued)

Counter Increment Mask Register (CIMR)

The Counter Increment Mask Register (CIMR) is used to mask bits from the Counter Increment Latch Register (CILR). If a bit in Register CIMR is set to One, the corresponding bit in Register CILR will be applied to the Interrupt Condition Register, which can then be used to generate an interrupt.

All bits in this register are set to Zero upon reset.

ACCESS RULES

Address	Read	Write
19h	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RES	TKRCVD	FRTRX	FRNCOP	FRCOP	FRLST	FREI	FRRCV

Bit	Symbol	Description
D0	FRRCV	Frame Received Counter Increment Mask: This bit is used to mask CILR.FRRCV.
D1	FREI	Error Isolated Counter Increment Mask: This bit is used to mask CILR.FREI.
D2	FRLST	Lost Frame Counter Increment Mask: This bit is used to mask CILR.FRLST.
D3	FRCOP	Frame Copied Counter Increment Mask: This bit is used to mask CILR.FRCOP.
D4	FRNCOP	Frame Not Copied Counter Increment Mask: This bit is used to mask CILR.FRNCOP.
D5	FRTRX	Frame Transmitted Counter Increment Mask: This bit is used to mask CILR.FRTRX.
D6	TKRCVD	Token Received Counter Increment Mask: This bit is used to mask CILR.TKRCVD.
D7	RES	Reserved

6.0 Control Information (Continued)

Counter Overflow Latch Register (COLR)

The Counter Overflow Latch Register (COLR) records carry events from the 20th bit of the Event Counters. Each bit in the COLR corresponds to an individual counter. Each bit may be masked via the Counter Overflow Mask Register (COMR).

ACCESS RULES

Address	Read	Write
1Ch	Always	Condition

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RES	TKRCVD	FRTRX	FRNCOP	FRCOP	FRLST	FREI	FRCV

Bit	Symbol	Description
D0	FRCV	Frame Received: Is set to One when the Frame Received Counter (FRCT) overflows.
D1	FREI	Frame Error Isolated: Is set to One when the Error Isolated Counter (EICT) overflows.
D2	FRLST	Frame Lost Isolated: Is set to One when the Lost Frame Counter (LFCT) overflows.
D3	FRCOP	Frame Copied: Is set to One when the Frame Copied Counter (FCCT) overflows.
D4	FRNCOP	Frame Not Copied: Is set to One when the Frame Not Copied Counter (FNCT) overflows.
D5	FRTRX	Frame Transmitted: Is set to One when the Frame Transmitted Counter (FTCT) overflows.
D6	TKRCVD	Token Received: Is set to One when the Token Received Counter (TKCT) overflows.
D7	RES	Reserved

6.0 Control Information (Continued)

Counter Overflow Mask Register (COMR)

The Counter Overflow Mask Register (COMR) is used to mask bits from the Counter Overflow Latch Register (COLR). If a bit in Register COMR is set to One, the corresponding bit in Register COLR will be applied to the Interrupt Condition Register, which can then be used to generate an interrupt.

All bits in this register are set to Zero upon reset.

ACCESS RULES

Address	Read	Write
1Dh	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RES	TKRCVD	FRTRX	FRNCOP	FRCOP	FRLST	FREI	FRCV

Bit	Symbol	Description
D0	FRCV	Frame Received Counter Overflow Mask: This bit is used to mask COLR.FRCV.
D1	FREI	Error Isolated Counter Overflow Mask: This bit is used to mask COLR.FREI.
D2	FRLST	Lost Frame Counter Overflow Mask: This bit is used to mask COLR.FRLST.
D3	FRCOP	Frame Copied Counter Overflow Mask: This bit is used to mask COLR.FRCOP.
D4	FRNCOP	Frame Not Copied Counter Overflow Mask: This bit is used to mask COLR.FRNCOP.
D5	FRTRX	Frame Transmitted Counter Overflow Mask: This bit is used to mask COLR.FRTRX.
D6	TKRCVD	Token Received Counter Overflow Mask: This bit is used to mask COLR.TKRCVD.
D7	RES	Reserved

6.0 Control Information (Continued)

Internal Event Latch Register (IELR)

The Internal Event Latch Register (IELR) reports internal errors in the BMAC device. These errors include MAC Parity errors and inconsistencies in the Receiver and Transmitter state machines.

After an internal state machine is detected and reported (bit RSMERR for the receiver and TSMERR for the transmitter), the Current Receive Status Register (CRS0) and Current Transmit Status Register (CTS0) continue to be updated as normal.

Errors internal to the BMAC device cause a MAC__Reset.

ACCESS RULES

Address	Read	Write
28h	Always	Condition

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	TSMERR	RSMERR	RES	MPE

Bit	Symbol	Description
D0	MPE	MAC Interface Parity Error: Indicates a Parity Error on the MAC Request Data pins (MRD7–0) when parity is enabled on the MA_Request Interface (bits MRP of the Mode Register is set and pin TXACK is asserted).
D1	RES	Reserved
D2	RSMERR	Receive State Machine Error: Indicates inconsistency in the Receiver state machine. When set, causes bit MCRST of the Function Register to be set.
D3	TSMERR	Transmit State Machine Error: Indicates inconsistency in the Transmitter state machine. When set, causes bit MCRST of the Function Register to be set.
D4–7	RES	Reserved

6.0 Control Information (Continued)

Exception Status Register (ESR)

The Exception Status Register (ESR) reports errors to the software. Errors include PHY Interface Parity errors, illegal attempts to access currently inaccessible registers, and writing to a conditional write location if a register bit has changed since it was last read. Each bit may be masked via the Exception Mask Register (EMR).

ACCESS RULES

Address	Read	Write
2Ch	Always	Condition

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
CWI	CCE	CPE	RES	RES	RES	RES	PPE

Bit	Symbol	Description
D0	PPE	PHY Interface Parity Error: Indicates parity error detected on PID7–0. Parity errors are reported when parity is enabled on the PHY_Request Interface (bit PIP of the Mode Register is set).
D1–4	RES	Reserved
D5	CPE	Control Bus Parity Error: Indicates a Control Bus Parity Error was detected on the Control Bus Data pins (CBD7–0) during a write operation to a register. Parity errors are reported if parity is enabled on the Control Bus Interface (bit CBP of the Mode Register is set).
D6	CCE	Control Bus Command Error: Indicates that a Control Bus command was not performed due to an error, i.e., illegal command or a Control Bus Write Parity error. An illegal command is an attempt to access a currently inaccessible register.
D7	CWI	Conditional Write Inhibit: Indicates that at least one bit of the previous conditional write operation was not written. This bit is set unconditionally after each write to a conditional write register if the value of the Compare Register is not equal to the value of the register that was accessed for a write before it was written. This may indicate that the accessed register has changed since it was last read. This bit is cleared after a successful conditional write. This occurs when the value of the Compare Register is equal to the value of the register that was accessed for a write before it was written. CWI does not contribute to setting the ESE bit of the Interrupt Condition Register (it is always implicitly masked).

6.0 Control Information (Continued)

Exception Mask Register (EMR)

The Exception Mask Register (EMR) is used to mask bits in the Exception Status Register (ESR). If a bit in Register EMR is set to One, the corresponding bit in Register ESR will be applied to the Condition Register, which can then be used to generate an interrupt.

All bits in this register are set to Zero upon request.

ACCESS RULES

Address	Read	Write
2Dh	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
ZERO	CCE	CPE	RES	RES	RES	RES	PPE

Bit	Symbol	Description
D0	PPE	PHY Interface Parity Error Mask: This bit is used to mask ESR.PPE.
D1-4	RES	Reserved
D5	CPE	Control Bus Parity Error Mask: This bit is used to mask ESR.CPE.
D6	CCE	Control Bus Error Mask: This bit is used to mask ESR.CCE.
D7	ZERO	Zero: This bit is always Zero. This implies that the CWI bit never contributes to the Interrupt Signal.

6.0 Control Information (Continued)

Interrupt Condition Register (ICR)

The Interrupt Condition Register (ICR) collects unmasked interrupts from the Event Registers. Interrupts are categorized into Ring Events, Token and Timer Events, Counter Events, and Error and Exceptional Status Events. If the bit in the Interrupt Mask Register (IMR) and the corresponding bit in the ICR are set to One, the \overline{INT} pin is forced low and thus triggers an interrupt.

Note: Bits are cleared ONLY by clearing underlying conditions (Mask bit and/or Event Bit) in the appropriate Event Register.

ACCESS RULES

Address	Read	Write
2Eh	Always	Data Ignored

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
ESE	IERR	RES	RES	COE	CIE	TTE	RNG

Bit	Symbol	Description
D0	RNG	Ring Event Interrupt: Is set if corresponding bits in the Ring Event Latch and Mask Registers are set.
D1	TTE	Token and Timer Event Interrupt: Is set if corresponding bits in the Token and Timer Event Latch and Mask Registers are set.
D2	CIE	Counter Increment Event Interrupt: Is set if corresponding bits in the Counter Increment Latch and Mask Registers are set.
D3	COE	Counter Overflow Event Interrupt: Is set if corresponding bits in the Counter Overflow Latch and Mask Registers are set.
D4–5	RES	Reserved
D6	IERR	Internal Error Interrupt: Is set if any bits in the Internal Event Register are set.
D7	ESE	Exception Status Event Interrupt: Is set if corresponding bits in the Exception Status and Mask Registers are set.

6.0 Control Information (Continued)

Interrupt Mask Register (IMR)

The Interrupt Mask Register (IMR) is used to mask bits in the Interrupt Condition Register (ICR). If a bit in Register IMR and the corresponding bit in Register ICR are set to One, the INT pin is forced low and causes an interrupt. Each bit in the IMR corresponds to an Event Register or a pair of Event Registers and associated bits.

ACCESS RULES

Address	Read	Write
2Fh	Always	Always

REGISTER BITS

D7	D6	D5	D4	D3	D2	D1	D0
ESE	IERR	RES	RES	COE	CIE	TTE	RNG

Bit	Symbol	Description
D0	RNG	Ring Event Mask: This bit is used to mask ICR.RNG.
D1	TTE	Token and Timer Event Mask: This bit is used to mask ICR.TTE.
D2	CIE	Counter Increment Event Mask: This bit is used to mask ICR.CIE.
D3	COE	Counter Overflow Event Mask: This bit is used to mask ICR.COE.
D4-5	RES	Reserved
D6	IERR	Internal Error Mask: This bit is used to mask ICR.IER.
D7	ESE	Exception Status Event Mask: This bit is used to mask ICR.ESE.

6.0 Control Information (Continued)

6.5 MAC PARAMETERS

The MAC Parameters are accessible in the Stop Mode. These parameters are also accessible in the Run Mode when the following conditions are met:

- the MAC Transmitter is in state T0, T1, or T3; and
- bits ITC and IRR of the Option Register are set to One; and
- bits CLM and BCN of the Function Register are set to Zero.

Otherwise read and write accesses will cause a command error (bit CCE of the Exception Status Register is set to One) and the access will not be performed.

The MAC Parameters are stored in the MAC Parameter RAM. They include the following control information:

- Individual Addresses: My Long Address (MLA0–5) and My Short Address (MSA0–1).
- Group Addresses: Group Long Address (GLA0–4) and Group Short Address (GSA0), Programmable Group Map (PGM0–1F), and Fixed Group Map (FGM0–1).
- MAC Frame Information: Requested Target Token Rotation Time (TREQ0–3) and Transmit Beacon Type (TBT0–3).

6.5.1 Individual Addresses

The Ring Engine supports both Long and Short Individual Addresses simultaneously. The Station's Long Address is stored in registers MLA0–5. The Station's Short Address is stored in register MSA0–1.

For received frames, MLA or MSA is compared with the received DA in order to set the Address recognized Flag (A Flag) and compared with the received SA in order to set the My Address recognized Flag (M Flag). In transmitted frames, MLA or MSA normally replaces the SA from the frame data stream (exception: when SA transparency is used).

Bits MLA(47) and MSA(15) are the most significant bits of the address and are transmitted and received first. Bits MLA(0) and MSA(0) are the least significant bits of the address and are transmitted and received last.

MLA and MSA should be valid for at least 12 byte times before the Addressing Mode is enabled and should remain valid for at least 12 byte times after the Addressing Mode is disabled in order to guarantee proper detection.

Bits ELA (Enable Long Addressing) and ESA (Enable Short Addressing) in the Option Register determine the address types that may be recognized and generated by this MAC.

My Long Address

My Long Address (MLA0–MLA5) represent this station's long 48-bit address.

ACCESS RULES

Address	Read	Write
40–45h	Stop Mode	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
MLA0	MLA(47)	MLA(46)	MLA(45)	MLA(44)	MLA(43)	MLA(42)	MLA(41)	MLA(40)
MLA1	MLA(39)	MLA(38)	MLA(37)	MLA(36)	MLA(35)	MLA(34)	MLA(33)	MLA(32)
MLA2	MLA(31)	MLA(30)	MLA(29)	MLA(28)	MLA(27)	MLA(26)	MLA(25)	MLA(24)
MLA3	MLA(23)	MLA(22)	MLA(21)	MLA(20)	MLA(19)	MLA(18)	MLA(17)	MLA(16)
MLA4	MLA(15)	MLA(14)	MLA(13)	MLA(12)	MLA(11)	MLA(10)	MLA(9)	MLA(8)
MLA5	MLA(7)	MLA(6)	MLA(5)	MLA(4)	MLA(3)	MLA(2)	MLA(1)	MLA(0)

Note: MLA(47) should always be set to 0.

My Short Address

My Short Address (MSA0–MSA1) represent this station's short 16-bit address.

ACCESS RULES

Address	Read	Write
46–47h	Stop Mode	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
MSA0	MSA(15)	MSA(14)	MSA(13)	MSA(12)	MSA(11)	MSA(10)	MSA(9)	MSA(8)
MSA1	MSA(7)	MSA(6)	MSA(5)	MSA(4)	MSA(3)	MSA(2)	MSA(1)	MSA(0)

Note: MSA(15) should always be set to 0.

6.0 Control Information (Continued)

6.5.2 Group Addresses

The Ring Engine supports detection of Group Addresses within programmable and fixed blocks of consecutive addresses. The algorithm used by the Ring Engine first performs a comparison between the most significant bits of the received DA with programmable and fixed addresses. If the most significant bits match, the remaining bits are used as an index into a programmable bit map. If the indexed bit is 1, the A Flag is set to 1; if the indexed bit is 0 the A flag remains 0.

One programmable block of 128 group addresses is supported for group long addresses (GLA) and one programmable block of group addresses is supported for group short addresses (GSA). Both of the programmable ranges share the same programmable group address map (PGM).

For short addresses, the first byte of a received DA is compared with GSA0 (bits GSA(15–8)). If they match then the second byte is used as an index into the PGM. For long addresses the first 5 bytes of a received DA are compared with GLA0 through GLA4 (bits GLA(47–8)). If all 5 of these bytes match the corresponding byte in the received DA, then the 6th byte of the received DA is used as an index into the PGM. The last byte of the address is used as an index into the PGM in both long and short group addressing.

A fixed block of 16 group addresses is supported for both long and short addresses at the end of the address space that includes the Universal/Broadcast address (FF...FF). For short addresses, if the first 12 bits of the received DA are all 1's then the last 4 bits are used as an index into the 16-bit Fixed Group Map (FGM). Similarly, for long addresses if the first 44 bits are all 1's, the last 4 bits are also used as an index into the 16-bit FGM.

The Group Addresses should be valid for at least 12 byte times before the Addressing Mode is enabled and should remain valid for at least 12 byte times after the Addressing Mode is disabled in order to guarantee proper detection.

Bits ELA (Enable Long Addressing) and ESA (Enable Short Addressing) in the Option Register determine the address types that will be recognized by this MAC.

Alternative group addressing schemes may be implemented using external matching logic that monitors the byte stream at the PHY Interface. The result of the comparison is returned using the EA (External A_Flag) pin.

Group Long Address

Group Long Address (GLA0–GLA4) represents the first 5 bytes of the long address, bit GLA(47) to bit GLA(8).

To disable Long Group Address matches, bits GLA(46–8) should be set to all 1's.

ACCESS RULES

Address	Read	Write
48–4Ch	Stop Mode	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
GLA0	GLA(47)	GLA(46)	GLA(45)	GLA(44)	GLA(43)	GLA(42)	GLA(41)	GLA(40)
GLA1	GLA(39)	GLA(38)	GLA(37)	GLA(36)	GLA(35)	GLA(34)	GLA(33)	GLA(32)
GLA2	GLA(31)	GLA(30)	GLA(29)	GLA(28)	GLA(27)	GLA(26)	GLA(25)	GLA(24)
GLA3	GLA(23)	GLA(22)	GLA(21)	GLA(20)	GLA(19)	GLA(18)	GLA(17)	GLA(16)
GLA4	GLA(15)	GLA(14)	GLA(13)	GLA(12)	GLA(11)	GLA(10)	GLA(9)	GLA(8)

Note: Bit GLA(47) should always be set to ONE.

6.0 Control Information (Continued)

Group Short Address

Group Short Address (GSA0) represents the station's short 16-bit address, bit GSA(15) to bit GSA(8). It is possible to disable Short Group Addressing by programming bits GSA(14–8) to all Ones.

ACCESS RULES

Address	Read	Write
4Eh	Stop Mode	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
GSA4	GSA(15)	GSA(14)	GSA(13)	GSA(12)	GSA(11)	GSA(10)	GSA(9)	GSA(8)

Design Note: GSA(15) is not used in the comparison since the comparison will only be accomplished if the received DA(15) is a One.

Fixed Group Address MAP (FGM0–FGM1)

If the first 44 bits of a long DA, DA(47–4), or if the first 12 bits of a short DA, DA(15–4) are 1, the last 4 bits of the DA, DA(3–0), are used as an index into FGM.

The 4-bit index into FGM can be viewed in two different ways. It can be viewed as 4 bits selecting one of 16 bits where the hexadecimal equivalent of DA(3–0) can be used as the index. For example the broadcast address would index FGM(F). Alternatively it can be viewed as one bit, DA(3), selecting the byte (FGM0 or FGM1) and three bits, DA(2–0) selecting one of 8 bits within a byte.

ACCESS RULES

Address	Read	Write
58–59h	Stop Mode	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
FGM0	FGM(7)	FGM(6)	FGM(5)	FGM(4)	FGM(3)	FGM(2)	FGM(1)	FGM(0)
FGM1	FGM(F)	FGM(E)	FGM(D)	FGM(C)	FGM(B)	FGM(A)	FGM(9)	FGM(8)

Note: Bit FGM(F) must be set to One to ensure proper handling of frames with the Universal/Broadcast address including the SMT NSA frames. This is mandatory for interoperability on an FDDI Ring.

6.0 Control Information (Continued)

Programmable Group Address MAP (PGM0–PGM1F)

If the first 40 bits of a long DA, DA(47–8), match the GLA or if the first 8 bits of a short DA, DA(15–8), match the GSA, the last 8 bits of the DA are used as an index into PGM.

The 8-bit index into PGM can be viewed in two different ways.

- As 8 bits selecting one of 256 bits where the hexadecimal equivalent of DA(7–0) can be used as the index. For example a DA with the last byte as A2h indexes PGM(A2).
- As 5 bits, DA(7–3), selecting the byte (PGM0 to PGM1F) and three bits, DA(2–0) selecting one of 8 bits within a byte. For example a DA with the last byte of A2h (10100010b) selects PGM14 bit 2.

It is possible to disable Long and Short Group Addressing by filling the Group Address Map with 0's.

In REV 1 of the BMAC device, PGM(00) to PGM(7F) are hardwired to 0 and are not accessible via the Control Interface. This implies that group addresses with DA(7) = 0 can not be recognized.

In REV 2 of the BMAC device, PGM(00) to PGM(7F) are set equal to PGM(80) to PGM(FF) and are accessible via the Control Interface. This implies that DA(7) of group addresses is a don't care.

ACCESS RULES

Address	Read	Write
70–7Fh	Stop Mode	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
PGM0	PGM(7)	PGM(6)	PGM(5)	PGM(4)	PGM(3)	PGM(2)	PGM(1)	PGM(0)
PGM1	PGM(F)	PGM(E)	PGM(D)	PGM(C)	PGM(B)	PGM(A)	PGM(9)	PGM(8)
PGM2	PGM(17)	PGM(16)	PGM(15)	PGM(14)	PGM(13)	PGM(12)	PGM(11)	PGM(10)
PGM3	PGM(1F)	PGM(1E)	PGM(1D)	PGM(1C)	PGM(1B)	PGM(1A)	PGM(19)	PGM(18)
PGM4	PGM(27)	PGM(26)	PGM(25)	PGM(24)	PGM(23)	PGM(22)	PGM(21)	PGM(20)
PGM5	PGM(2F)	PGM(2E)	PGM(2D)	PGM(2C)	PGM(2B)	PGM(2A)	PGM(29)	PGM(28)
PGM6	PGM(37)	PGM(36)	PGM(35)	PGM(34)	PGM(33)	PGM(32)	PGM(31)	PGM(30)
PGM7	PGM(3F)	PGM(3E)	PGM(3D)	PGM(3C)	PGM(3B)	PGM(3A)	PGM(39)	PGM(38)
PGM8	PGM(47)	PGM(46)	PGM(45)	PGM(44)	PGM(43)	PGM(42)	PGM(41)	PGM(40)
PGM9	PGM(4F)	PGM(4E)	PGM(4D)	PGM(4C)	PGM(4B)	PGM(4A)	PGM(49)	PGM(48)
PGMA	PGM(57)	PGM(56)	PGM(55)	PGM(54)	PGM(53)	PGM(52)	PGM(51)	PGM(50)
PGMB	PGM(5F)	PGM(5E)	PGM(5D)	PGM(5C)	PGM(5B)	PGM(5A)	PGM(59)	PGM(58)
PGMC	PGM(67)	PGM(66)	PGM(65)	PGM(64)	PGM(63)	PGM(62)	PGM(61)	PGM(60)
PGMD	PGM(6F)	PGM(6E)	PGM(6D)	PGM(6C)	PGM(6B)	PGM(6A)	PGM(69)	PGM(68)
PGME	PGM(77)	PGM(76)	PGM(75)	PGM(74)	PGM(73)	PGM(72)	PGM(71)	PGM(70)
PGMF	PGM(7F)	PGM(7E)	PGM(7D)	PGM(7C)	PGM(7B)	PGM(7A)	PGM(79)	PGM(78)

6.0 Control Information (Continued)

Programmable Group Address MAP (PGM0–PGM1F) (Continued)

ACCESS RULES

Address	Read	Write
60–6Fh	Stop Mode	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
PGM10	PGM(87)	PGM(86)	PGM(85)	PGM(84)	PGM(83)	PGM(82)	PGM(81)	PGM(80)
PGM11	PGM(8F)	PGM(8E)	PGM(8D)	PGM(8C)	PGM(8B)	PGM(8A)	PGM(89)	PGM(88)
PGM12	PGM(97)	PGM(96)	PGM(95)	PGM(94)	PGM(93)	PGM(92)	PGM(91)	PGM(90)
PGM13	PGM(9F)	PGM(9E)	PGM(9D)	PGM(9C)	PGM(9B)	PGM(9A)	PGM(99)	PGM(98)
PGM14	PGM(A7)	PGM(A6)	PGM(A5)	PGM(A4)	PGM(A3)	PGM(A2)	PGM(A1)	PGM(A0)
PGM15	PGM(AF)	PGM(AE)	PGM(AD)	PGM(AC)	PGM(AB)	PGM(AA)	PGM(A9)	PGM(A8)
PGM16	PGM(B7)	PGM(B6)	PGM(B5)	PGM(B4)	PGM(B3)	PGM(B2)	PGM(B1)	PGM(B0)
PGM17	PGM(BF)	PGM(BE)	PGM(BD)	PGM(BC)	PGM(BB)	PGM(BA)	PGM(B9)	PGM(B8)
PGM18	PGM(C7)	PGM(C6)	PGM(C5)	PGM(C4)	PGM(C3)	PGM(C2)	PGM(C1)	PGM(C0)
PGM19	PGM(CF)	PGM(CE)	PGM(CD)	PGM(CC)	PGM(CB)	PGM(CA)	PGM(C9)	PGM(C8)
PGM1A	PGM(D7)	PGM(D6)	PGM(D5)	PGM(D4)	PGM(D3)	PGM(D2)	PGM(D1)	PGM(D0)
PGM1B	PGM(DF)	PGM(DE)	PGM(DD)	PGM(DC)	PGM(DB)	PGM(DA)	PGM(D9)	PGM(D8)
PGM1C	PGM(E7)	PGM(E6)	PGM(E5)	PGM(E4)	PGM(E3)	PGM(E2)	PGM(E1)	PGM(E0)
PGM1D	PGM(EF)	PGM(EE)	PGM(ED)	PGM(EC)	PGM(EB)	PGM(EA)	PGM(E9)	PGM(E8)
PGM1E	PGM(F7)	PGM(F6)	PGM(F5)	PGM(F4)	PGM(F3)	PGM(F2)	PGM(F1)	PGM(F0)
PGM1F	PGM(FF)	PGM(FE)	PGM(FD)	PGM(FC)	PGM(FB)	PGM(FA)	PGM(F9)	PGM(F8)

6.0 Control Information (Continued)

6.5.3 Claim Information: Requested Target Token Rotation Time (TREQ)

The Requested Target Token Rotation Time (TREQ) is stored in registers TREQ0–TREQ3. TREQ(31–0) is represented as a negative two's complement number. This value is transmitted in all Claim frames generated by the Ring Engine.

Bits TREQ(31–24) are always transmitted as and compared with FFh and bits TREQ(7–0) are always transmitted as and compared with 00h, independent of the value stored in the MAC Parameter RAM. TREQ is therefore programmable with 20.48 μ s resolution and a maximum value of 1.34 seconds.

ACCESS RULES

Address	Read	Write
50–53h	Stop Mode	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
TREQ0	TREQ(31)	TREQ(30)	TREQ(29)	TREQ(28)	TREQ(27)	TREQ(26)	TREQ(25)	TREQ(24)
TREQ1	TREQ(23)	TREQ(22)	TREQ(21)	TREQ(20)	TREQ(19)	TREQ(18)	TREQ(17)	TREQ(16)
TREQ2	TREQ(15)	TREQ(14)	TREQ(13)	TREQ(12)	TREQ(11)	TREQ(10)	TREQ(9)	TREQ(8)
TREQ3	TREQ(7)	TREQ(6)	TREQ(5)	TREQ(4)	TREQ(3)	TREQ(2)	TREQ(1)	TREQ(0)

6.5.4 Beacon Information: Transmit Beacon Type (TBT)

Transmit Beacon Type 0–3 (TBT0–3) represents the Transmit Beacon Type to be transmitted in the information field of a Beacon frame.

When the Beacon state is reached as a result of a failed Claim process, the first byte of the Beacon Information field, bits TBT(31–24), are forced to Zero to produce a Beacon Type 0 as required by the MAC Standard. Bits TBT(23–0) are transmitted as the rest of the Information field.

When the Beacon state is reached as a result of a Beacon Request (when Function.BCN is set), bits TBT(31–0) are transmitted as the Information field. Bit TBT(0) is transmitted last.

ACCESS RULES

Address	Read	Write
54–57h	Stop Mode	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
TBT0	TBT(31)	TBT(30)	TBT(29)	TBT(28)	TBT(27)	TBT(26)	TBT(25)	TBT(24)
TBT1	TBT(23)	TBT(22)	TBT(21)	TBT(20)	TBT(19)	TBT(18)	TBT(17)	TBT(16)
TBT2	TBT(15)	TBT(14)	TBT(13)	TBT(12)	TBT(11)	TBT(10)	TBT(9)	TBT(8)
TBT3	TBT(7)	TBT(6)	TBT(5)	TBT(4)	TBT(3)	TBT(2)	TBT(1)	TBT(0)

6.0 Control Information (Continued)

6.6 TIMER VALUES

The Ring Engine stores several timer values and thresholds used in normal operation. With the exception of TNEG, the timers use an exponential expansion on a 4-bit value to produce a negative twos complement 24-bit value used by the Timer Logic. The timer values are always readable and are writable in Stop Mode.

Asynchronous Priority Threshold (THSH1)

The Ring Engine currently supports one Asynchronous Priority Threshold (THSH1) in addition to the one at TTRT. The Asynchronous Priority Threshold is used in a magnitude comparison with THT when an Asynchronous Priority Request is presented to the MAC Request Interface.

Bits 7–4 are always written to Zero and are always read as Zero.

When more than one threshold is used, the users of THSH1 have the lowest priority. All asynchronous transmissions are limited by TTRT. If the Late Flag is set, no frames may be transmitted, regardless of the value of the Asynchronous Priority Threshold.

ACCESS RULES

Address	Read	Write
87h	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
THSH1	Zero	Zero	Zero	Zero	THSH(3)	THSH(2)	THSH(1)	THSH(0)

THSH1(3–0)	Time remaining in THT when token becomes unusable
0	10.24 μ s
1	20.48 μ s
2	40.96 μ s
3	81.92 μ s
4	163.84 μ s
5	327.68 μ s
6	655.36 μ s
7	1.3107 ms
8	2.6214 ms
9	5.2429 ms
A	10.486 ms
B	20.972 ms
C	41.943 ms (default)
D	83.886 ms
E	167.77 ms
F	335.54 ms

Warning: The default value may not be appropriate for all values of TNEG. In some cases, this could result in a request that is NEVER serviced.

6.0 Control Information (Continued)

Maximum Token Rotation Time (TMAX)

The Maximum Token Rotation Time (TMAX) denotes the maximum Target Token Rotation Time supported by this station. TMAX is stored as a 4-bit value that is expanded to a binary exponential value. Bits 7–4 are ignored during write operations and are always read as Zero.

TMAX has a maximum value of 1.34 seconds with a threshold of $40.96 \times 2^{TMAX} \mu\text{s}$. On a Master Reset (Function.MARST set to One), TMAX is set to the value of Ch which corresponds to 167.772 ms, the default specified by the FDDI MAC Standard.

ACCESS RULES

Address	Read	Write
93h	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
TMAX	Zero	Zero	Zero	Zero	TMAX(3)	TMAX(2)	TMAX(1)	TMAX(0)

TMAX(0–3)	Time
0	40.96 μs
1	81.92 μs
2	163.84 μs
3	327.68 μs
4	655.36 μs
5	1.3107 ms
6	2.6214 ms
7	5.2429 ms
8	10.486 ms
9	20.972 ms
A	41.943 ms
B	83.886 ms
C	167.77 ms (default)
D	335.54 ms
E	671.09 ms
F	1.3422s

6.0 Control Information (Continued)

Valid Transmission Time (TVX)

The Valid Transmission Timer (TVX) is used to increase the responsiveness of the ring to errors that cause ring recovery. The TVX value denotes the maximum time in which a valid frame or token should be seen by this station. TVX is stored as a 4-bit value that is expanded to a binary exponential value. Bits 7–4 are ignored during write operations and read as Zero.

TVX has a maximum value of 1.34 seconds with a threshold of $40.96 \times 2^{TVX} \mu s$. On a Master Reset (Function.MARST is set to One), TVX is set to the value of 6h which corresponds to 2.62 ms, the default by the FDDI MAC Standard.

ACCESS RULES

Address	Read	Write
97h	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
TVX	Zero	Zero	Zero	Zero	TVX(3)	TVX(2)	TVX(1)	TVX(0)

TVX(0–3)	Time
0	40.96 μs
1	81.92 μs
2	163.84 μs
3	327.68 μs
4	655.36 μs
5	1.3107 ms
6	2.6214 ms (default)
7	5.2429 ms
8	10.486 ms
9	20.972 ms
A	41.943 ms
B	83.886 ms
C	167.77 ms
D	335.54 ms
E	671.09 ms
F	1.3422s

6.0 Control Information (Continued)

Negotiated Target Rotation Time (TNEG)

The Negotiated Target Rotation Time (TNEG0–3) is a 32-bit two's complement value. It is the result of the Claim Process. TNEG is loaded either directly from the received Claim Information field (T_Bid_Rc) or via the Control Interface.

The first byte of TNEG (bits TNEG(31–24)) always contains FFh. TNEG has a maximum value of 1.34 seconds and a resolution of 80 ns.

TRT is loaded with TNEG when the Ring_Operational flag is set. TNEG is not automatically compared with TREQ when the Ring_Operational flag is set. This should be checked by software whenever the ring becomes operational to make sure that TNEG is less than or equal to TREQ.

An implementation of the SM_Control.Request (Reset) should load TNEG with TMAX to remove any possibility of the station entering Claim early.

On a Master Reset (Function.MARST is set), TNEG is set to FFE00000, which corresponds to 167.772 ms, the default TMAX specified by the FDDI MAC Standard.

ACCESS RULES

Address	Read	Write
98–9Bh	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
TNEG0	TNEG(31)	TNEG(30)	TNEG(29)	TNEG(28)	TNEG(27)	TNEG(26)	TNEG(25)	TNEG(24)
TNEG1	TNEG(23)	TNEG(22)	TNEG(21)	TNEG(20)	TNEG(19)	TNEG(18)	TNEG(17)	TNEG(16)
TNEG2	TNEG(15)	TNEG(14)	TNEG(13)	TNEG(12)	TNEG(11)	TNEG(10)	TNEG(9)	TNEG(8)
TNEG3	TNEG(7)	TNEG(6)	TNEG(5)	TNEG(4)	TNEG(3)	TNEG(2)	TNEG(1)	TNEG(0)

6.0 Control Information (Continued)

6.7 EVENT COUNTERS

The Event Counters are used to gain access to the internal 20-bit counters used to gather statistics.

The following event counters are included:

- Frame Received Counter (FRCT1-3)
- Error Isolated Counter (EICT1-3)
- Lost Frame Counter (LFCT1-3)
- Frame Copied Counter (FCCT1-3)
- Frame Not Copied Counter (FNCT1-3)
- Frame Transmitted Counter (FTCT1-3)
- Token Received Counter (TKCT1-3)
- Ring Latency Counter (RLCT1-3)
- Late Count Counter (LTCT)

6.7.1 Processing Procedures

The counters are 20-bit wrap-around counters except for the Late Count Counter which is a 4-bit sticky counter (see *Figure 6-2*).

Since the Control Bus Interface is an 8-bit interface and the counters are 20-bits wide, a register holding scheme is implemented. In order to provide a consistent snapshot of a counter, while the least significant byte is read, the upper 12 bits are loaded into a holding which can then be read. The least significant byte must be read first.

The Counters are always readable and are writable in Stop Mode. The Counters are not reset as a result of a Master Reset. This may be done by either reading the Counters out and keeping track relative to the initial value read, or by writing a value (Zero) to all of the Counters in Stop Mode. The Counters may be written in any order. Interrupts may be requested when the counters increment (except for Ring Latency Counter) or wrap-around (except for Ring Latency Counter and Late Count Counter).

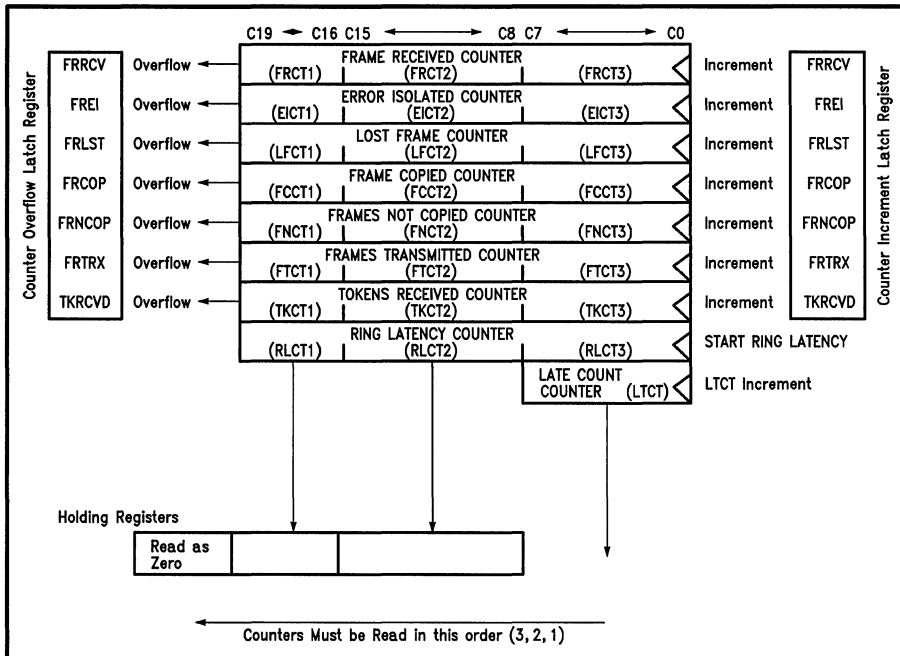


FIGURE 6-2. Event Counters

TL/F/10387-10

6.0 Control Information (Continued)

Frame Received Counter (FRCT)

The Frame Received Counter (FRCT) is specified in the FDDI MAC Standard. It is the count of all complete frames received including MAC frames, Void frames and frames stripped by this station.

Interrupts are available on increment (CILR.FRRRCV) and when the 20-bit counter overflows and wraps around (COLR.FRRRCV).

ACCESS RULES

Address	Read	Write
A0-A3h	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
FRCT0	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
FRCT1	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
FRCT2	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
FRCT3	CT(7)	CT(6)	CT(5)	CT(4)	CT(3)	CT(2)	CT(1)	CT(0)

6.0 Control Information (Continued)

Error Isolated Counter (EICT)

The Error Isolated Counter (EICT) is specified in the FDDI MAC Standard. It is the count of all error frames detected by this station and no previous station.

It is incremented when:

- 1) an FCS error is detected and the received Error Indicator (Er) is not equal to S; or
- 2) a frame of invalid length (i.e., off-boundary T) is received and Er is not equal to S; or
- 3) Er is not R or S

Interrupts are available on increment (CILR.FREI) and when the 20-bit counter overflows and wraps around (COLR.FREI).

ACCESS RULES

Address	Read	Write
A4–A7h	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
EICT0	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
EICT1	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
EICT2	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
EICT3	CT(7)	CT(6)	CT(5)	CT(4)	CT(3)	CT(2)	CT(1)	CT(0)

6.0 Control Information (Continued)

Lost Frame Counter (LFCT)

The Lost Frame Counter (LFCT) is specified in the FDDI MAC Standard. It is the count of all instances where a Format Error is detected in a frame or token such that the credibility of the PDU reception is in doubt.

The Lost Frame Counter is incremented when any symbol other than a data or Idle symbol is received between the Starting and Ending Delimiters of a PDU (this includes parity errors).

Interrupts are available on increment (CILR.FRLST) and when the 20-bit counter overflows and wraps around (COLR.FRLST).

ACCESS RULES

Address	Read	Write
A8-ABh	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
LFCT0	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
LFCT1	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
LFCT2	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
LFCT3	CT(7)	CT(6)	CT(5)	CT(4)	CT(3)	CT(2)	CT(1)	CT(0)

In REV 1 of the BMAC device the Lost Count includes frames stripped on an ODD symbol boundary. This may cause larger than expected counts in Rings where an upstream station produces valid remnants that begin on an ODD symbol boundary. (The Ring Engine converts these remnants to byte aligned remnants, so that only the downstream station would increment its Lost Count.) In subsequent revisions remnants that begin on an odd symbol boundary are not considered lost frames and do not cause the Lost Count to increment.

6.0 Control Information (Continued)

Frame Copied Counter (FCCT)

The Frame Copied Counter (FCCT) maintains the count of the number of frames successfully copied by this station. This counter can be used to accumulate station performance statistics.

The Frame Copied Counter is incremented when an internal or external match occurs on the Destination Address, no errors were detected in the frame, and the frame was successfully copied (VCOPY signal is asserted). Copied MAC and Void frames are not included in this count.

For SMT NSA frames, the Frame Copied Count only increments for NSA frames received with the A Indicator as an R symbol for which the frame was copied. SMT NSA frames received with the A Indicator as an S do not cause this count to increment, even if the frame is successfully copied.

Increments are available on increment (CILR.FRCOP) and when the 20-bit counter overflows and wraps around (COLR.FRCOP).

ACCESS RULES

Address	Read	Write
AC-AFh	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
FCCT0	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
FCCT1	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
FCCT2	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
FCCT3	CT(7)	CT(6)	CT(5)	CT(4)	CT(3)	CT(2)	CT(1)	CT(0)

6.0 Control Information (Continued)

Frame Not Copied Counter (FNCT)

The Frame Not Copied Counter (FNCT) maintains a count of the number of frames intended for this station that were not successfully copied by this station. This count can be used to accumulate station performance statistics such as insufficient buffering or deficient frame processing capabilities for frames addressed to this station.

The Frame Not Copied Counter is incremented when an internal or external match (when Option.EMIND enabled) occurs on the Destination Address, no errors were detected in the frame, and the frame was not successfully copied (VCOPY signal not asserted). Not Copied MAC frames and Void frames are not included in this count.

Interrupts are available on increment (CILR.FRNCOP) and when the 20-bit counter overflows and wraps around (COLR.FRNCOP).

ACCESS RULES

Address	Read	Write
B0-B3h	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
FNCT0	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
FNCT1	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
FNCT2	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
FNCT3	CT(7)	CT(6)	CT(5)	CT(4)	CT(3)	CT(2)	CT(1)	CT(0)

In REV 1 of the BMAC device, the Frame Not Copied Counter does increment for all NSA frames received with the A indicator as an S symbol, if it was copied or not. This will result in higher than expected values in the Not Copied Counter. To obtain a more accurate value with REV 1 of the BMAC device, the number of copied NSA frames received with the A Indicator set should be subtracted from the value in the Not Copied Counter.

In subsequent revisions of the BMAC device, NSA frames received with the A Indicator as S that are not copied will not be counted.

In REV 2 the handling of SMT NSA has been modified in accordance with the MAC-2 Draft Standard. For SMT NSA frames, the Frame Not Copied Count only increments for NSA frames received with the A Indicator as an R symbol for which the frame was not copied. SMT NSA frames received with the A Indicator as an S do not cause this count to increment, even if the frame is not successfully copied. Group addressed frames transmitted by this station and recognized by this station that are not copied will also cause this counter to increment.

6.0 Control Information (Continued)

Frame Transmitted Counter (FTCT)

The Frame Transmitted Counter (FTCT) maintains the count of frames transmitted successfully by this station. The counter can be used to accumulate station performance statistics.

The Frame Transmitted Counter is incremented every time a complete frame is transmitted from the MAC Request Interface. MAC and Void frames generated by the Ring Engine are not included in the count.

Interrupts are available on increment (CILR.FRTRX) and when the 20-bit counter overflows and wraps around (COLR.FRTRX).

ACCESS RULES

Address	Read	Write
B4-B7h	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
FTCT0	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
FTCT1	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
FTCT2	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
FTCT3	CT(7)	CT(6)	CT(5)	CT(4)	CT(3)	CT(2)	CT(1)	CT(0)

6.0 Control Information (Continued)

Token Received Counter (TKCT)

The Token Received Counter (TKCT) maintains the count of valid tokens received by this station. The counter can be used with the Ring Latency Counter to calculate the average network load over a period of time. The frequency of token arrival is inversely related to the network load.

The Token Received Counter is incremented every time a valid token arrives.

Interrupts are available on increment (CILR.TKRCVD) and when the 20-bit counter overflows and wraps around (COLR.TKRCVD).

ACCESS RULES

Address	Read	Write
B8-BBh	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
TKCT0	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
TKCT1	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
TKCT2	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
TKCT3	CT(7)	CT(6)	CT(5)	CT(4)	CT(3)	CT(2)	CT(1)	CT(0)

6.0 Control Information (Continued)

Ring Latency Counter (RLCT)

The Ring Latency Counter (RLCT) is a measurement of time for PDUs to propagate around the ring. This counter contains the last measured ring latency whenever the RLVD bit of the Token and Timer Event Latch Register (TELR.RLVD) is One.

The current ring latency is measured by timing the propagation of a My_Void frame around the ring. A new latency measurement can be requested by clearing the Ring Latency Valid bit of the Token Event Register (TELR.RLVLD).

When the ring is operational, the next early token is captured. Before the token is re-issued, a My_Void frame is transmitted and the Ring Latency Counter (RLCT) is reset. The token will not be captured if the Inhibit Token Option (Option.ITC) is set and the ring latency will not be measured.

When the ring is not operational, ring latency timing will commence at the end of the next immediate request. A My_Void is transmitted and RLCT is reset. This could be used to time how long the ring is non-operational since the My_Void frame will not return.

The Ring Latency Counter increments once every 6 byte times from when the Ending Delimiter of the My_Void frame is transmitted, until the Ending Delimiter of the My_Void frame returns. When the My_Void frame returns, the ring latency valid bit (TELR.RLVLD) is set and may cause an interrupt. When set, RELR.RLVLD indicates that RLCT will be valid within 1.28 μ s. The Ring Latency Counter can measure ring latencies up to 1.3421772 seconds with accuracy of 1.28 μ s.

The ring latency timing function is automatically disabled when exceptions are detected and retried at the next opportunity.

Since a Master Reset (Function.MARST) causes TELR.RLVLD to be cleared, the ring latency will automatically be measured on the first opportunity (at the end of the first immediate request or with the first early token).

ACCESS RULES

Address	Read	Write
BC-BFh	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
RLCT0	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
RLCT1	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
RLCT2	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
RLCT3	CT(7)	CT(6)	CT(5)	CT(4)	CT(3)	CT(2)	CT(1)	CT(0)

In REV 1 of the BMAC device, the Latency Counter is not reset to Zero when a new latency measurement is initiated. The latency count will be the difference between the value of RLCT after the measurement is complete and the value of RLCT before the measurement was initiated.

If a new latency measurement causes the latency counter to overflow, the new latency value will be less than the previous value. In this case, no subtraction is necessary. The new value is equal to the ring latency. This is because the Ring Engine recognizes the overflow condition and restarts the latency count from zero.

It is not possible to reset the Latency counter in software once the BMAC device has been put into RUN mode (Mode.Run = 1). This counter is only writable while in STOP mode (Mode.Run = 0).

6.0 Control Information (Continued)

Late Count (LTCT)

The Late Count Counter (LTCT) is implemented differently than suggested by the FDDI MAC Standard, but provides similar information. The function of the Late Count Counter is divided between the Late__Flag and a separate counter. The Late__Flag is equivalent to the Standard Late Count with a non-zero value. It is maintained by the Ring Engine to indicate if it is possible to send asynchronous traffic. When the ring is operational, Late Count indicates the time it took the ring to recover the last time the ring went non-operational. When the ring is non-operational, Late Count indicates the time it has taken (so far) to recover the ring.

Late Count is provided to assist Station Management in the isolation of serious ring errors. In many situations, it is helpful for SMT to know how long it has been since the ring went non-operational in order to determine if it is necessary to invoke recovery procedures. When the ring becomes non-operational, there is no way to know how long it will stay non-operational, therefore a timer is necessary. If the Late Count Counter is not provided, SMT would be forced to start a timer every time the ring goes non-operational even though it may seldom be used. By using the provided Late Count Counter, an SMT implementation may be able to alleviate this additional overhead.

Late Count is incremented every time TRT expires while the ring is non-operational and Late__Flag is set (once every TMAX). This counter is never writable, not even in Stop Mode. The counter is set to Zero as a result of a MAC Reset when a Beacon or Claim Request is not also present (Function.MCRST is set and Function.BCN and Function.CLM are not set) and every time the ring becomes non-operational. The Late Count Counter is a sticky counter at 15.

Events reported in the Token and Timer Event Latch Register (TEL.R.CBERR, TELR.TRTEXP) can be used to determine that Late Count Counter has incremented. No overflow event is provided.

ACCESS RULES

Address	Read	Write						
0Fh	Always	n/a						
	D7	D6	D5	D4	D3	D2	D1	D0
LTCT	Zero	Zero	Zero	Zero	CT3	CT2	CT1	CT0

7.0 Signal Descriptions

Interface Organization

The BMAC device signals are organized into five Interfaces:

Control Interface: Used for processor access to the BMAC device.

PHY Interface: Interface signals to the DP83251/55 PLAYER device.

MAC Indicate Interface: Signals for receiving and processing incoming frames.

MAC Request Interface: Signals used to capture tokens and transmit frames.

Electrical Interface: Signals associated with power supply and clocking.

Application Note 689, BMAC Device Hardware Design Guide, provides a discussion of design considerations and tradeoffs for using the BMAC Device.

7.1 CONTROL INTERFACE

The Control Interface operates asynchronous to the operation of the data services. During an access, the external Control Bus is synchronized with the internal Control Bus.

The ACK and INT signals are open drain signals to allow wire ORing several such signals.

Symbol	Pin #	I/O	Description
CBP	10	I/O	Control Bus Parity: Odd parity on CBD7-0.
CBD7-0	9-6, 3-1, 132	I/O	Control Bus Data
CBA7-0	131-129, 127-123	I	Control Bus Address: Address of a particular register.
\overline{CE}	120	I	Control Bus Enable: Handshake signal used to begin a Control Interface access. Active low signal.
R/ \overline{W}	119	I	Read/ ~ Write: Determines current direction of a Control Interface access.
\overline{ACK}	122	OD	~ Acknowledge: Acknowledges that the Control Interface access has been performed. Active low, open drain signal.
INT	121	OD	~ Interrupt: Indicates presence of one or more enabled condition(s) from the Event Registers. Active low, open drain signal.

7.0 Signal Descriptions (Continued)

7.2 PHY INTERFACE

The PHY Interface signals transfer symbol pairs between the BMAC and PLAYER devices. Transfers are synchronous using the 12.5 MHz Local Byte Clock signal (signal provided by the Clock Distribution Device).

A control bit is used to indicate if a Data symbol pair or Control symbol pair or a mixed Control/Data symbol pair are being transferred.

Parity is generated on the PHY__Indicate and MA__Indicate data. Parity is checked on the PHY__Request and MA__Request data.

Symbol	Pin #	I/O	Description
PRP	114	O	PHY Request Parity: Odd parity for PRC and PRD7-0.
PRC	112	O	PHY Request Control: 0: Indicates PRD7-0 contains a Data symbol pair. 1: Indicates PRD7-0 contains a Control or mixed Control/Data symbol pair.
PRD7-0	110, 108, 105, 103, 99, 97, 95, 92	O	PHY Request Data: Contains a Data or Control symbol pair.
PIP	115	I	PHY Indicate Parity: Odd parity for PIC and PID7-0.
PIC	113	I	PHY Indicate Control: 0: Indicates PID7-0 contains a Data symbol pair. 1: Indicates PID7-0 contains a Control or mixed Control/Data symbol pair.
PID7-0	111, 109, 107, 104, 102, 98, 96, 93	I	PHY Indicate Data: Contains a Data or a mixed Control/Data symbol pair.

7.0 Signal Descriptions (Continued)

7.2.1 PHY Interface Codes

The DP83251/155 PLAYER device converts the Standard 4B/5B FDDI symbol code to the internal code used at the PHY Interface. The PH_DATA.Indication table shows how the Ring Engine interprets the codes generated by the PLAYER device and the PH_DATA_Request table shows the codes generated by the Ring Engine.

The internal code is actually an 8B/9B code with parity where one bit is used to determine whether the symbol pair contains two data symbols or at least one control symbol.

PH_DATA.Indication

The Ring Engine interprets the byte stream the PLAYER device as defined in Table 7-1.

TABLE 7-1. Internal PHY Indicate Coding

Value	PIP	PIC	PID(7-4)	PID(3-0)	Type
0	1	0	0000	0000	Data Symbol Pair
1	0	0	0000	0001	Data Symbol Pair
:	:	:	:	:	:
254	0	0	1111	1110	Data Symbol Pair
255	1	0	1111	1111	Data Symbol Pair
JK	P	1	1101	xxxx	Start Delimiter
PI	P	1	x011	x1xx	PH_Invalid
PI	P	1	x011	xx1x	PH_Invalid
II	P	1	10xx	xxxx	Idle Symbols
ni	P	1	0000	10xx	Data/Idle Symbol
RR	P	1	0110	0110	Frame Status
RS	P	1	0110	0111	Frame Status
RT	P	1	0110	0101	Frame Status
SS	P	1	0111	0111	Frame Status
SR	P	1	0111	0110	Frame Status
ST	P	1	0111	0101	Frame Status
SX	P	1	0111	xxxx	Frame Status
TX	P	1	0101	xxxx	Ending Delimiter
TR	P	1	0101	0110	Ending Delimiter
TS	P	1	0101	0111	Ending Delimiter
TT	P	1	0101	0101	Ending Delimiter
nT	P	1	0000	0101	Mixed Symbol Pair
Parity Error	~P	0	????	????	Code Violation
Otherwise	?	1	Else		Code Violation

where:

PIP PHY Indicate Parity bit, ODD parity

PIC PHY Indicate Control bit:

0 = > data byte,

1 = > control/mixed byte

PID(7-0) PHY Indicate Data(7-0)

P represents ODD Parity (~P is Bad Parity)

x- represents a don't care and is not decoded

? represents a 1 or 0 but not both.

The PLAYER device aligns the received JK to a byte boundary. Thus, no provision is made in the internal code or by the Ring Engine for off boundary JKs.

The Idle and PH_Invalid encodings overlap. Idle symbols received while the PLAYER device is in Active Line State (ALS) or Idle Line State (ILS) are not considered PH_INVALID. Idle symbols received while the PLAYER device is in states other than ALS or ILS are treated as PH_Invalid.

7.0 Signal Descriptions (Continued)

PH_DATA.Request

The Ring Engine generates the 10 bit byte stream as defined in Table 7-2. Note that all symbol pairs are either control or data symbol pairs. Mixed data/control symbol pairs are never generated or repeated by the Ring Engine.

TABLE 7-2. Internal PHY Request Coding

Value	PRP	PRC	PRD(7-4)	PRD(3-0)	Type
0	1	0	0000	0000	Data Symbol Pair
1	0	0	0000	0001	Data Symbol Pair
:	:	:	:	:	:
254	0	0	1111	1110	Data Symbol Pair
255	1	0	1111	1111	Data Symbol Pair
JK	0	1	1101	1101	Start Delimiter
II	0	1	1010	1010	Idle Symbols
RR	0	1	0110	0110	Frame Status
RS	1	1	0110	0111	Frame Status
RT	0	1	0110	0101	Frame Status
SS	0	1	0111	0111	Frame Status
SR	1	1	0111	0110	Frame Status
ST	1	1	0111	0101	Frame Status
TR	0	1	0101	0110	Ending Delimiter
TS	1	1	0101	0111	Ending Delimiter
TT	0	1	0101	0101	Ending Delimiter

Where:

PRP PHY Request Parity bit, parity for all symbol pairs is ODD

PRC PHY Request control bit:

0 = > data byte

1 = > control byte

PRD(7-0) PHY Request Data (7-0)

The Ring Engine can repeat the RS, RT and ST symbol pairs but does not create them.

7.0 Signal Descriptions (Continued)

7.3 MAC INDICATION INTERFACE

The MAC Indication Interface provides a delayed version of the byte stream presented to the Ring Engine at the PHY Indication Interface. Every byte of all incoming frames is presented at the MAC Indication Interface. Every byte time (80 ns) one byte of data with Odd parity is presented at the MAC Indication Interface. This byte stream is interpreted by the system interface logic using the control signals that are provided in parallel with the byte stream. These control signals are used to determine frame boundaries in the byte stream, determine whether or not to (continue to) copy a frame, and to provide status on received PDUs.

In the following sections, an overview of the signals is provided (Section 7.3.1) as well as a detailed explanation (Section 7.3.2) with several example timing scenarios (Section 7.3.3).

7.3.1 Overview

The MAC Indication Interface is divided into one group of data signals and five groups of control signals.

The data signals consist of the 8 bits of MAC Indicate Data (MID) with parity.

The control signals consist of 5 groups:

- **PDU Sequencing** to aid in delimiting PDUs from the byte stream and sequencing through fields in the received PDUs.
- **PDU Flags** to aid in the decision of whether or not to continue to copy a PDU.
- **Termination Event** to determine when and how a PDU terminated.
- **Termination Status** to provide status on received frames.
- **External Flags** to allow external address comparison and copy information to be conveyed back to the Ring Engine.

The **PDU Sequencing** signals are asserted at different points within a PDU.

RCSTART when the Starting Delimiter is present on MID

FCRCVD when the Frame Control Field is on MID

DARCVD when the last byte of the DA is on MID until the next Starting Delimiter

SARCVD when the last byte of the SA is on MID until the next Starting Delimiter

INFORCVD when the fourth byte of the info field is on MID until the next Starting Delimiter

Not all of the sequencing signals would be used in a typical implementation.

The **PDU Flags** provide the input for potential copy criteria and status breakpoints. The results of the comparisons between the station's long or short address and the frame's source and destination addresses are provided in the **AFLAG** and **MFLAG** signals. The sequencing information is used to determine when this information is valid. Since the Ring Engine is capable of accomplishing four internal comparisons on any given frame, two signals give the internal comparison that was accomplished.

AFLAG Internal DA Match. There are actually four **AFLAGs** as determined by the two signals: **FCSL—Short/Long**, **DAIG—Individual/Group**. Valid with **DARCVD**.

MFLAG Internal SA Match. There are actually two **MFLAGs** as determined by the values of **FCSL**. Valid with **SARCVD**.

SAMESA SA same as in previous frame. Valid with **SARCVD** on Non-MAC frames. Can be used by external logic to batch status or reduce the number of interrupts when multiple frames are received from the same station.

SAMEINFO First four bytes of Info same as in previous frame. Valid with **INFORCVD** on MAC frames. Can be used to inhibit copying of identical MAC frames.

No temporary buffering is provided in the Ring Engine. The system interface must provide this buffering while the decision is made on whether or not to continue to copy the frame.

Termination Event: One of these signals is asserted at the end of every PDU:

EDRCVD when the Ending Delimiter is on MID until the end of the Frame Status (typically asserted for two byte times)

TKRCVD when the Ending Delimiter of a token is on MID

FRSTRP when the first Idle byte of a stripped frame is on MID

FOERROR when the byte with the format error is on MID

MACRST when a **MACRST** occurs or Ring Engine in Stop Mode

7.0 Signal Descriptions (Continued)

Termination Status: These signals provide status on reception of a valid ending delimiter on a frame.

VDL Valid Data Length.

Criteria:

1. more than the minimum number bytes
2. integral number of symbol pairs.

Valid with EDRDVD

VFCS Valid FCS Criteria: Received FCS matches with standard CRC polynomial.

Valid with EDRDVD

External Flags: These signals are used for setting the outgoing control indicators, the interface accepts:

EA For external address matches for the setting of the A indicator (bridging, Group addressing, Aliasing)

VCOPY For the setting of the C Indicator when AFLAG or EA is set.

7.0 Signal Descriptions (Continued)

7.3.2 Signals

All output signals change relative to the rising edge of the Local Byte Clock signal (provided by the Clock Distribution Device) and are active high.

7.3.2.1 Indication Data

Symbol	Pin #	I/O	Description
MIP	73	O	MAC Indicate Parity: Odd parity on MID7-0. Only valid with Data and Status Indicators.
MID7-0	74-76, 79-83	O	<p>MAC Indicate Data:</p> <p>Data: Indicates data is being presented on MID7-0 between the rising edge of Frame Control Receive FCRCVD and the rising edge of one of the following signals:</p> <ul style="list-style-type: none"> Ending Delimiter Received (EDRCVD), Token Received (TKRCVD), Format Error (FOERROR), Frame Strip (FRSTRP), or MAC Reset (MACRST). <p>Status: Indicates Status Indicators are being presented on MID7-0 while Ending Delimiter Received (EDRCVD) or Token Received (TKRCVD) is asserted.</p>

The Contents and interpretation of MID7-0 are given in Table 7-3.

TABLE 7-3. MAC Indication Coding

Contents	Value	MID(7-4)	MID(3-0)	Condition
Data	0	0	0	Between RCSTART and
	1	0	1	EDRCVD or
	2	0	2	TKRCVD or
	:	:	:	FRSTRP or
	:	:	:	FOERROR or
	254 255	F F	E F	MACRST
Status	TT	5	5	with EDRCVD or TKRCVD
	TR	5	6	with EDRCVD
	TS	5	7	with EDRCVD
	TX	5	≠5, 6 or 7	with EDRCVD
	nT	0	5	with EDRCVD
	RT	6	5	with EDRCVD
	RR	6	6	with EDRCVD
	RS	6	7	with EDRCVD
	ST	7	5	with EDRCVD
	SR	7	6	with EDRCVD
SS	7	7	with EDRCVD	
undefined	otherwise	x	x	otherwise

7.0 Signal Descriptions (Continued)

7.3.2.2 PDU Sequencing

The PDU Sequencing signals apply to the data and status available at the MAC Indicate Interface. They are used to determine the validity of the data (MID7–0) and parity (MIP). In addition the sequencing signals are used to determine the validity of the Addressing Flags, and the Frame Status such as the Control Indicators. All timing is explained relative to the byte present on the MAC Indicate Interface.

Symbol	Pin #	I/O	Description
RCSTART	51	O	Receive Start: Indicates that a MAC PDU Starting Delimiter has been received. It is asserted when the Starting Delimiter is present at the MAC Indicate Interface.
FCRCVD	52	O	Frame Control Received: Indicates that the Frame Control field is present. It is asserted when the Frame Control field is present at the MAC Indicate Interface.
DARCVD	55	O	Destination Address Received: Indicates that the Destination Address has been received. It is asserted on the last byte of the Destination Address and remains asserted until the next PDU Starting Delimiter is received.
SARCVD	59	O	Source Address Received: Indicates that the Source Address has been received. It is asserted on the last byte of the Source Address and remains asserted until the next PDU Starting Delimiter is received.
INFORCVD	62	O	Information Field Received: Indicates that four bytes of the Information field have been received. It is asserted on the fourth byte of the INFO field and remains active until the next PDU Starting Delimiter is received.

7.0 Signal Descriptions (Continued)

7.3.2.3 PDU Flags

The PDU flags may be used with the received Frame Control field to determine if an attempt should be made to copy the frame.

Symbol	Pin #	I/O	Description
AFLAG	56	O	My Destination Address Recognized: Indicates that an internal address match occurred on the Destination Address field. The internal address (MSA, MLA, GSA, GLA) match is indicated by the assertion of FCSL and DAIG. AFLAG is asserted along with DARCVD. It is reset when the next PDU Starting Delimiter is received.
DAIG	54	O	Individual/Group Address Flag: Indicates the address type. Valid on the first byte of the Destination Address. 0: Individual Address 1: Group Address
FCSL	53	O	Short/Long Address Flag: Indicates the size of the Destination Address. Signal is valid when FCRCVD is asserted. 0: Short Address 1: Long Address Used in conjunction with TKRCVD to indicate the type of token received. 0: Non-restricted token 1: Restricted token
MFLAG	60	O	My Source Address Recognized: Indicates that the received Source Address field matched the MLA or MSA. SA.IG is ignored in the comparison. MFLAG is asserted along with SARCVD. It is reset when the next PDU Starting Delimiter is received.
SAMESA	61	O	Same Source Address: Indicates three conditions: 1. The Source Address of the current frame is the same as the Source Address of the previous frame AND 2. The current and previous frames were not MAC frames AND 3. The current and previous frames have the same address field size. SAMESA is asserted along with SARCVD. It is reset when the next PDU starting delimiter is received.
SAMEINFO	63	O	Same MAC Information: Indicates two conditions: 1. The first 4 bytes of the information field of the current frame are identical to the first 4 bytes of the previous non-Void frame AND 2. The current and previous non-Void frames were MAC frames. SAMEINFO is asserted along with INFORCVD. It is reset when the next PDU Starting Delimiter is received. Note that the FC field is not checked to insure that it is the same as in the previous frame. This includes the address size comparison. In REV1 of the BMAC Device Void frames are not ignored as stated in 1 and 2 above.

7.0 Signal Descriptions (Continued)

7.3.2.4 Termination Event

The terminating event for all PDUs is provided in the PDU Status signals.

When a token is terminated by a valid Ending Delimiter (TT symbol pair), the TKRCVD signal is asserted. When a frame is terminated by a valid Ending Delimiter, the EDRCVD is asserted and remains asserted until all frame status has been passed to the MA_Indicate Interface. Every PDU is terminated by one of the following:

1. A valid Ending Delimiter (TKRCVD or EDRCVD)
2. An IDLE symbol indicating that the frame was stripped by another station (FRSTRP)
3. A symbol other than data, Idle or an Ending Delimiter indicating that a Format Error occurred (FOERROR)
4. A MAC Reset (MACRST)

Symbol	Pin #	I/O	Description
TKRCVD	69	O	Token Received: Indicates that the Ending Delimiter for a valid token is being received.
EDRCVD	66	O	Ending Delimiter Received: Indicates that the Ending Delimiter for a frame is being received. The values of the received Status Indicators are available through the MID byte stream on this and subsequent cycles while this signal is asserted.
FRSTRP	71	O	Frame Stripped: Indicates that an Idle symbol was received while expecting part of a PDU. This usually indicates that the PDU was stripped by an upstream station. This signal may be asserted anytime during reception of a frame after RCSTART is asserted.
FOERROR	70	O	Format Error: Indicates that a Format Error (non-DATA, IDLE or Ending Delimiter Symbol) was detected. This signal may be asserted anytime during reception of a frame including with RCSTART for the next frame.
MACRST	72	O	MAC Reset: Indicates that a MAC Reset has been issued. This signal is asserted as a result of a software or hardware reset, or internal errors. This signal is asserted whenever bit MCRST of the Function Register is set. This signal may be asserted anytime.

7.3.2.5 Termination Status

When a valid Ending delimiter is received after a valid starting delimiter, the termination status signals provided the results of the Frame validity check and the Frame Check Sequence Check.

The received values of the control indicators are presented in the data stream while EDRCVD is asserted.

Symbol	Pin #	I/O	Description
VDL	68	O	Valid Data Length: Indicates that a frame meeting the minimum length requirements of the Standard and of an even number of symbols was received. This signal is valid with EDRCVD.
VFCS	67	O	Valid Frame Check Sequence: Indicates that a frame with the standard CRC was received. This signal is valid with EDRCVD.

7.0 Signal Descriptions (Continued)

7.3.2.6 External Flags

The External Flags provide input to the Ring Engine in order to set the A and C indicators or in order to initiate stripping based on external logic.

Symbol	Pin #	I/O	Description
EA	38	I	External A__Flag: Indicates that an external address match occurred. The value of EA is used to alter the values of the transmitted A and C Indicators (Ax and Cx). EA must be valid one byte time before EDRCVD is asserted. When the EMIND bit in the Option Register is set, the A Indicator is repeated as set (S symbol) and either the Copied or Not Copied Frame Counter is incremented depending on the value of VCOPY.
EM	39	I	External M__Flag: Indicates that the current frame should be stripped. Three byte times after the EM signal is asserted, the Ring Engine begins to source Idle symbols and the frame is stripped.
VCOPY	85	I	Valid Copy: Indicates that the C Indicator (Cx) should be repeated as an S symbol for received frames when VCOPY is asserted, the received frame is not an SMT NSA frame received with the A indicator as Set and <ol style="list-style-type: none"> 1. the internal A__flag is set or 2. Option.EMIND= 1 and the external A__flag (EA) is set; See Section 5.5 for a complete description of the setting of the control indicators. VCOPY must be valid one byte time before EDRCVD is asserted and remain asserted until EDRCVD is asserted. The sampled value of VCOPY with EDRCVD also affects the incrementing of the frame copied and frame not copied counters. See the description of the event counters for more information.

7.0 Signal Descriptions (Continued)

7.3.3 Timing Examples

The following examples show the sequencing of signals at the MAC Indicate Interface for well formed frames, for stripped frames and for several special cases. The diagrams show the logical operation of the interface with 0 ns delays. The actual delays are specified in Section 8. Also, in place of specifying the actual values for the flags and inputs, the cycles where they are valid (for outputs) or must be valid (for inputs) are shown.

Frame Reception

The examples shown in *Figures 7-1* through *7-3* display normal frame reception for a frame with a Short Address and a frame with a Long Address.

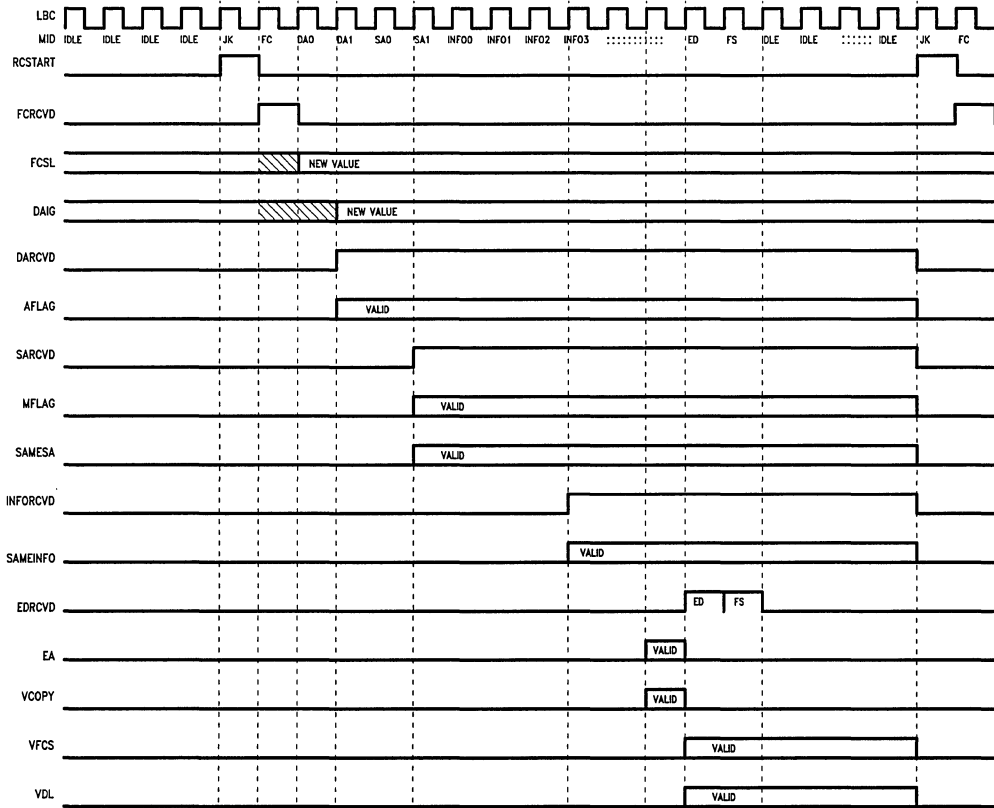
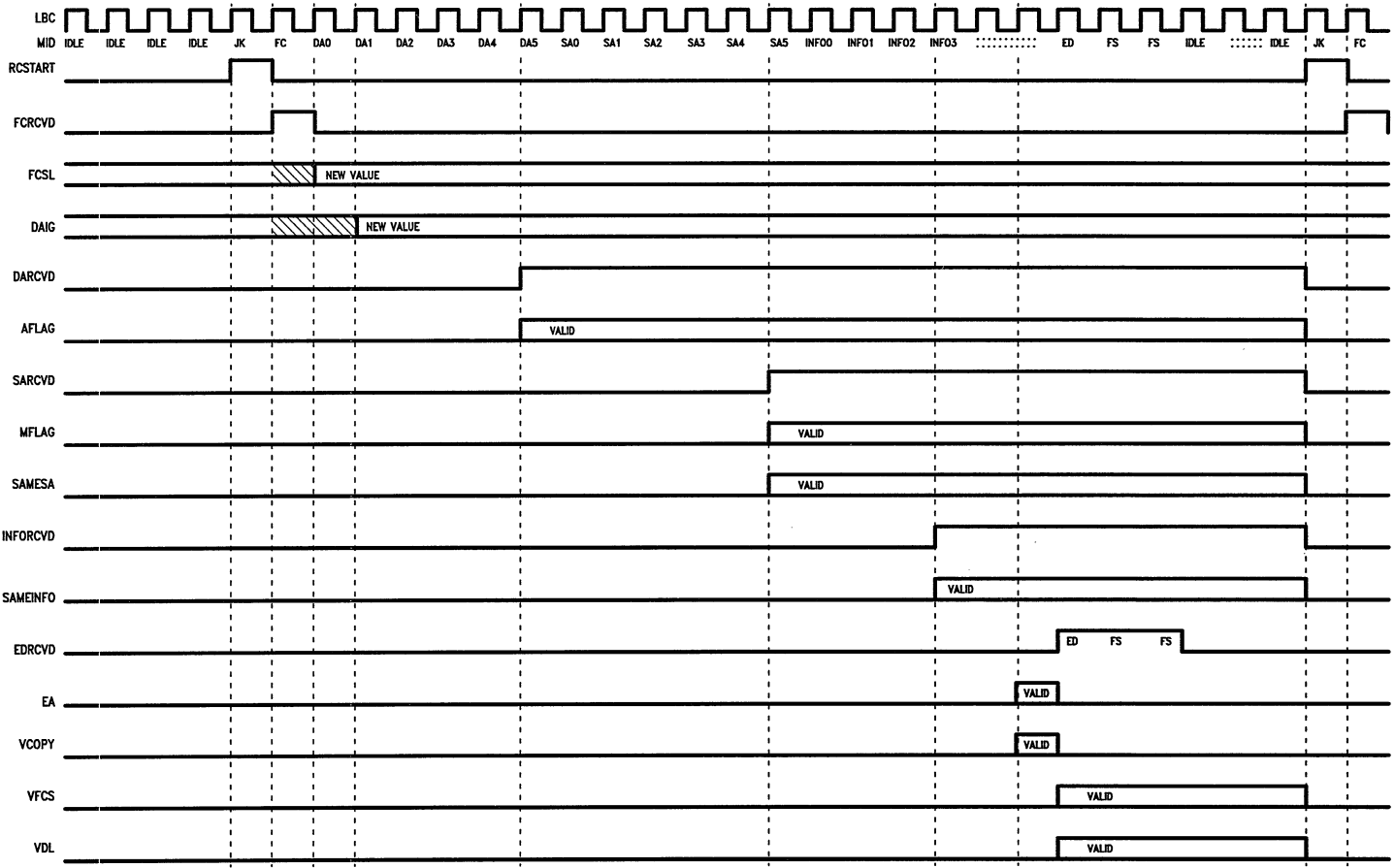


FIGURE 7-1. Frame Reception with Short Address

TL/F/10387-5



2-212

FIGURE 7-2. Frame Reception with Long Address

7.0 Signal Descriptions (Continued)

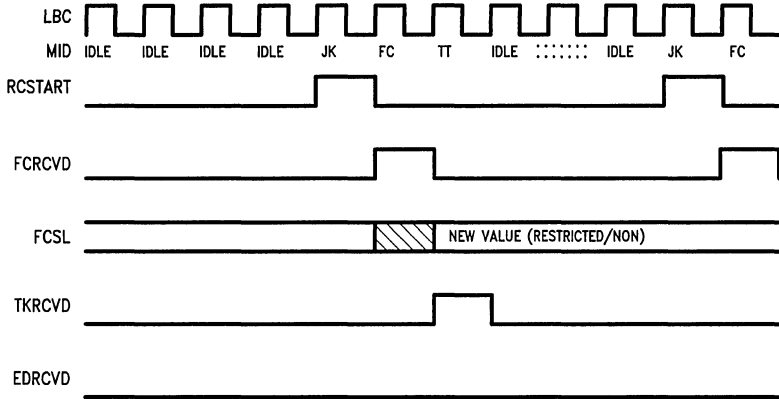


FIGURE 7-3. Token Reception

TL/F/10387-8

Remnant Reception

In these examples, the remnants of frames that were stripped by an upstream station are received. Examples are shown for frames where the strip point occurred at an upstream station before, during and after the SA field. (See Figures 7-4 through 7-6.)

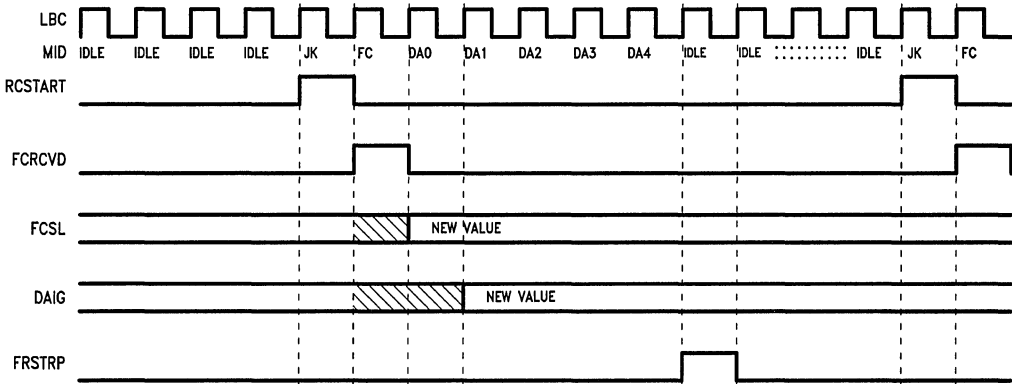


FIGURE 7-4. Frame Stripped before SA Field

TL/F/10387-9

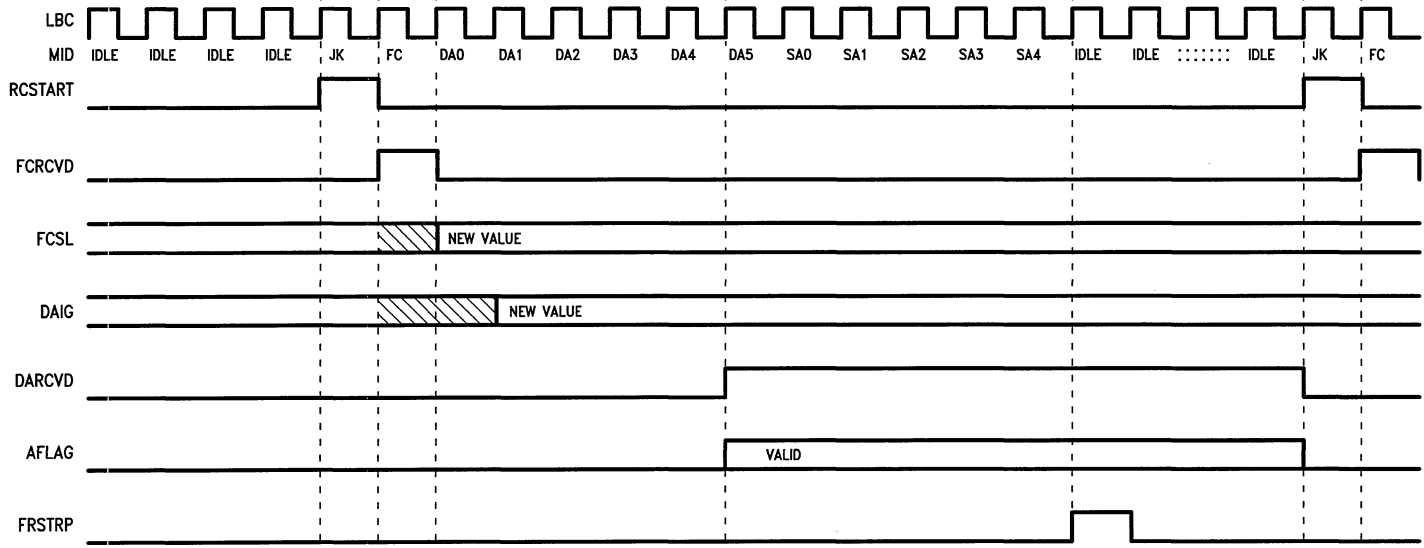


FIGURE 7-5. Frame Stripped during SA Field

TL/F/10387-12

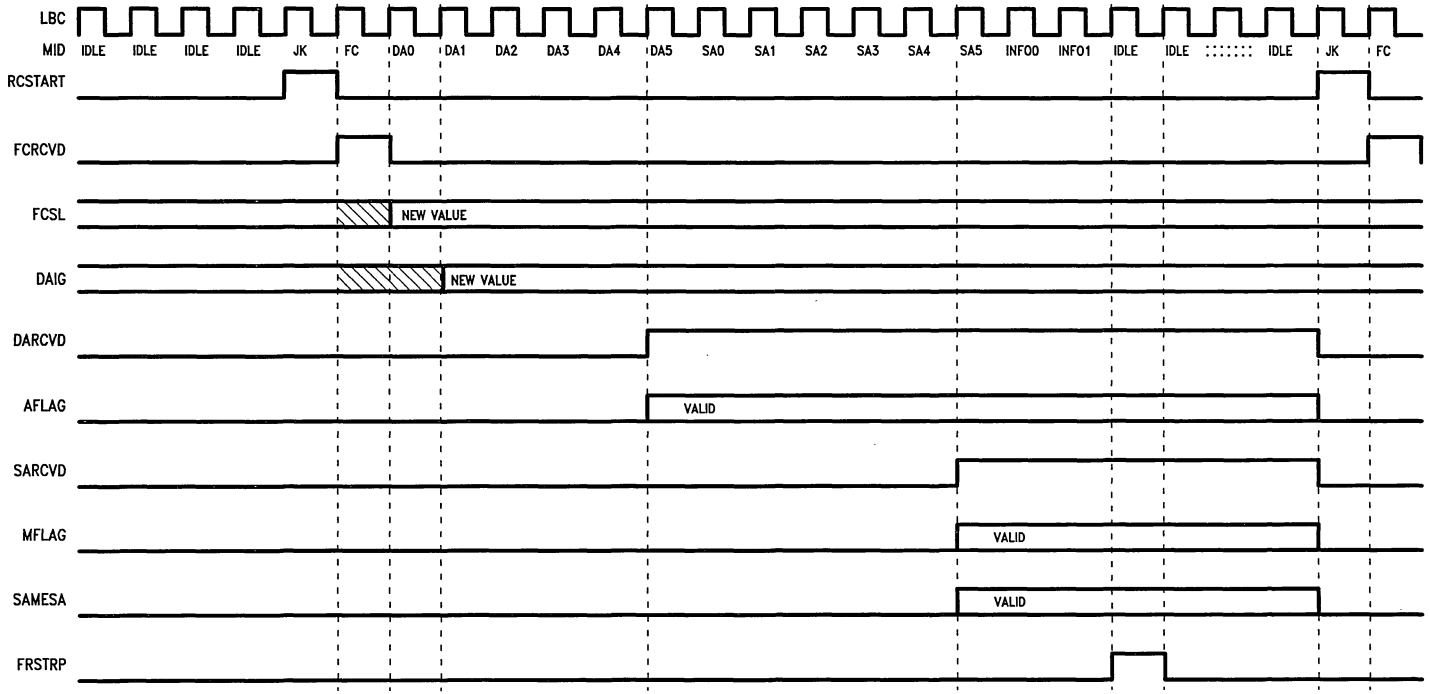


FIGURE 7-6. Frame Stripped after SA Field

TL/F/10387-13

2-215

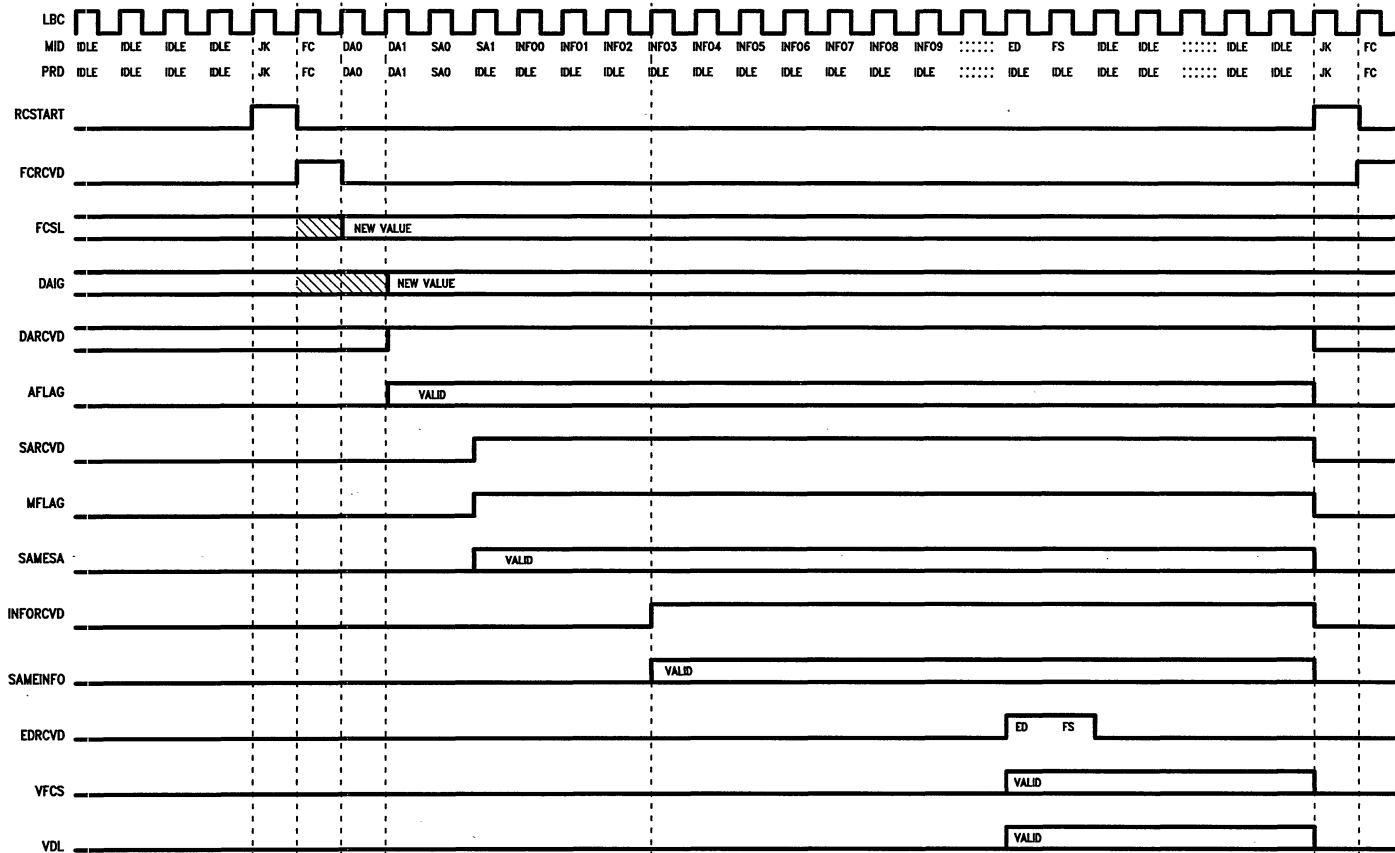
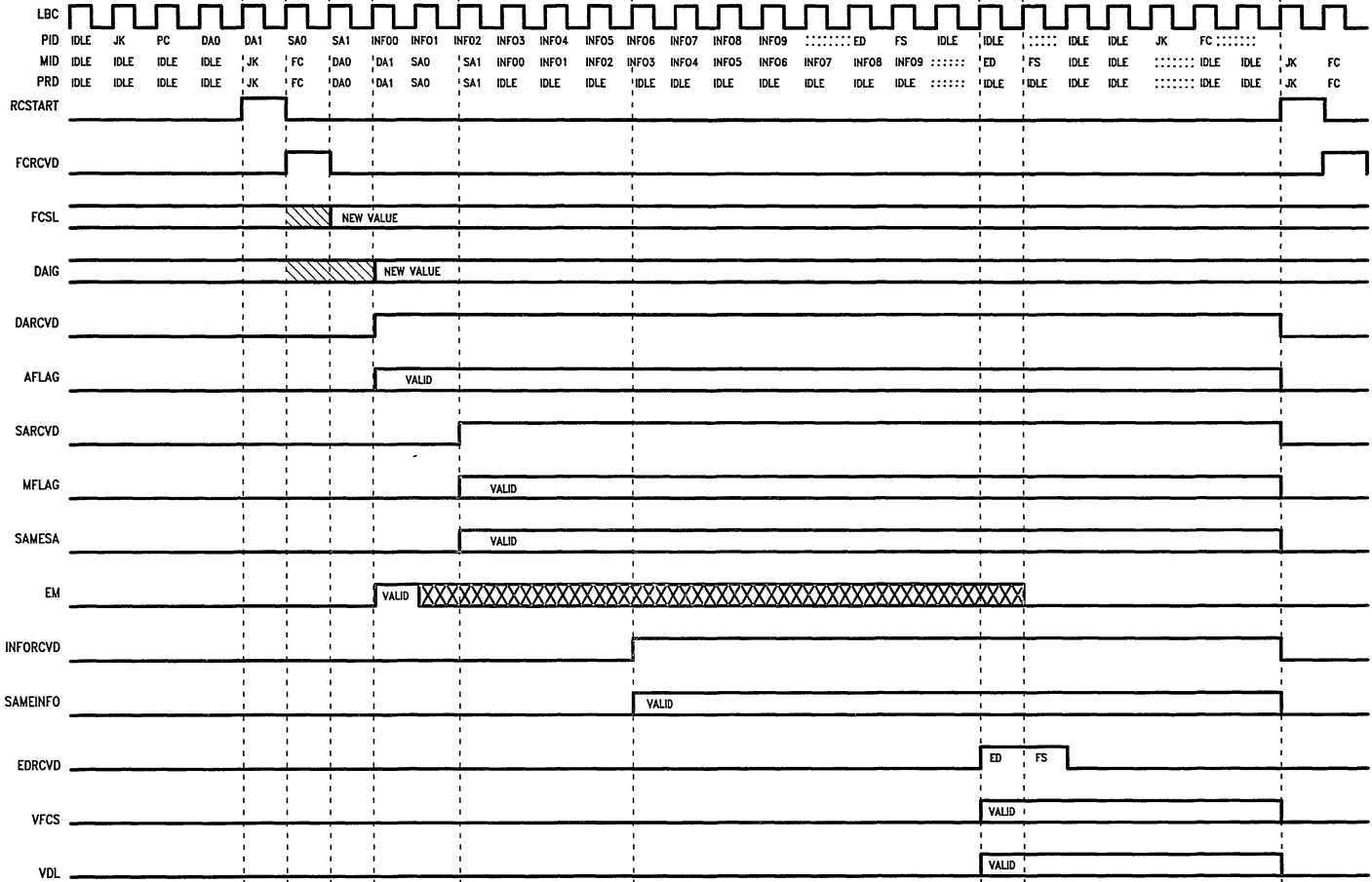


FIGURE 7-7. Stripping Based on M Flag



Note that stripping begins 3 byte times after EM is asserted.

FIGURE 7-8. Stripping Based on External M Flag

TL/F/10387-15

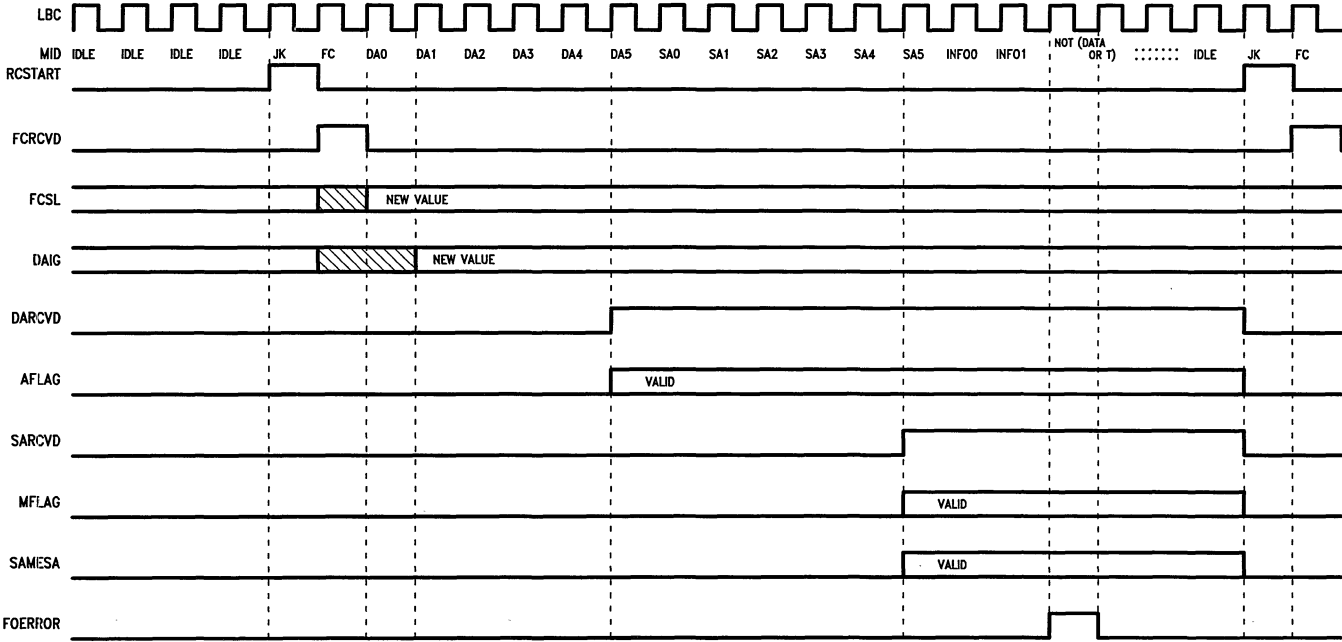
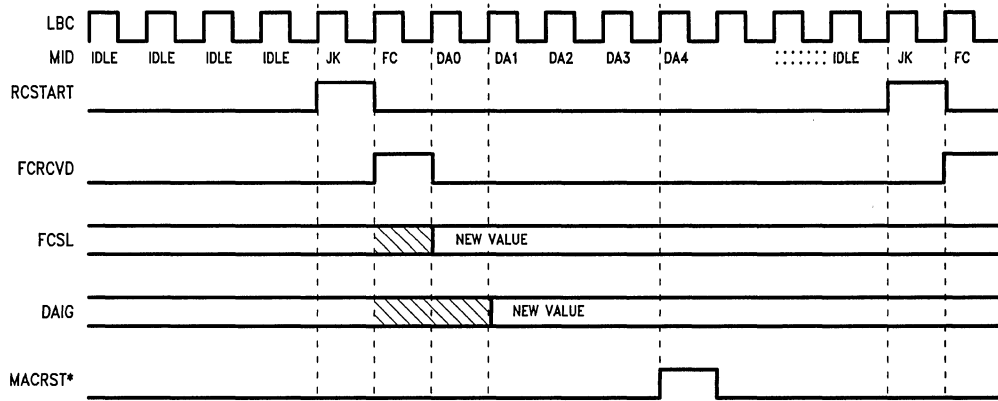


FIGURE 7-9. Format Error

7.0 Signal Descriptions (Continued)



TL/F/10387-17

*MAC Reset can be asserted at any time. When MODE0.RUN = 0 MACRST is asserted and remains asserted.

FIGURE 7-10. MAC Reset

7.0 Signal Descriptions (Continued)

7.4 MAC REQUEST INTERFACE

The MAC Request Interface is used to gain access to the ring and to transmit data into the ring. After a Request is submitted to the interface, the Ring Engine awaits for an appropriate Service Opportunity in which to service the Request. Frames associated with the Request are transmitted during an appropriate Service Opportunity. Sections 5.2 and 5.3 provide important information related to the functional operation of the interface.

In the following sections, an overview (Section 7.4.1) and detailed signal description (Section 7.4.2) are provided. The detailed description includes a signal by signal description followed by a state diagram that details the operation of the handshaking signals. Finally several example timing scenarios are shown (Section 7.4.3).

7.4.1 Overview

The MAC Request Interface signals provide the Request and Frame level handshakes required to transmit frames.

The MAC Request Interface is divided into one group of data signals and four groups of control signals.

The data signals consist of the 8 bits of MAC Request data with optional parity.

The control signals consist of:

- **Handshake** signals that implement a Request level and Frame level handshake. The state machines that specify the interface handshake are provided and described in detail in Section 7.4.3.
- **Service Parameters** that convey the requested type of Service Opportunity.
- **Frame Options** that convey the special transmission options. These are especially useful for MAC level bridging applications.
- **Transmission Status** that report the success and/or failure of the transmission.

7.4.2 Signals

Request Data

The MRD7–0 signals change on the rising edge of the Local Byte Clock signal (provided by the Clock Distribution Device).

Symbol	Pin #	I/O	Description
MRP	41	I	MAC Request Parity: Odd parity on MRD7–0.
MRD7–0	42–49	I	MAC Request Data: Data byte conveyed for transmission while the Transmit Acknowledge (TXACK) signal is asserted.

7.0 Signal Descriptions (Continued)

Handshake

The Handshake signals control the Request Interface handshaking process. They are used for token capture and transmission of PDUs.

Symbol	Pin #	I/O	Description
TXPASS	28	O	Transmit Pass: Indicates the absence of a Service Opportunity. This could result from an unusable request class, waiting for a token, timer expiration or MAC Reset. TXPASS is always asserted between service opportunities. It is deasserted when TXRDY signal is asserted at the beginning of a Service Opportunity.
TXRDY	29	O	Transmit Ready: Indicates that the Transmitter is ready for another frame. For a non-immediate request, a usable token must be held in order to transmit frames. TXRDY is asserted when: <ul style="list-style-type: none"> a) A usable token is being held or b) An immediate request becomes serviceable or c) After frame transmission if the current Service Opportunity is still usable for another frame. It is deasserted when the TXPASS or TXACK signal is asserted.
RQRDY	23	I	Request Ready: Indicates that the Transmitter should attempt to provide a Service Opportunity as indicated by the RQRCLS(3-0) signals one cycle before RQRDY is asserted. The Service Opportunity will be maintained as long as possible. If RQRDY is asserted within 6 byte times after TXRDY signal is asserted, the Transmitter will wait at least L_{Max} plus one Void frame (4.16 μ s or 4.80 μ s) for RQSEND to be asserted before releasing the token.
RQSEND	24	I	Request Send: Indicates that the Transmitter should send the next frame. The MRD(7-0) signals convey the FC byte when the RQSEND signal is asserted. If RQSEND is asserted within 6 byte times after the TXRDY signal is asserted, the Transmitter will send the frame with a minimum length preamble. If RQSEND is not asserted within L_{Max} plus one Void frame after RQRDY signal has been asserted (4.16 μ s or 4.60 μ s), the token may become unusable (due to a timer expiration). For Immediate transmissions from the Claim or Beacon State (when RQCLM or RQBCN is asserted), RQSEND must be asserted no later than 8 byte times after TXRDY is asserted. RQSEND may only be asserted when TXRDY and RQRDY signals are asserted and RQFINAL is deasserted. RQSEND must be deasserted not later than one byte time after TXRDY is deasserted.
TXACK	30	O	Transmit Acknowledge: Indicates that the Transmitter is ready for the next data byte. TXACK is asserted when the FC byte is accepted on MRD7-0, and remains asserted for each additional data byte accepted. It is deasserted one byte time after RQEOF or RQABT is asserted. The signal is also deasserted when TXABORT or TXPASS is asserted.
RQEOF	25	I	Request End of Frame: Indicates that MRD7-0 conveys the last data byte when asserted. Normally, this is the last byte of the INFO field of the frame (exceptions: FCS transparency, invalid frame length). RQEOF causes TXACK to be deasserted and is ignored if TXACK is not asserted.
RQABT	27	I	Request Abort: Indicates that the current frame should be aborted. Normally this causes the Transmitter to generate a Void, Claim, or Beacon frame. RQABT causes TXACK to be deasserted and is ignored when TXACK is not asserted.
RQFINAL	26	I	Request Final: Indicates that the final frame of the request has been presented to the MAC Interface. When asserted, the Issue Token Class (as opposed to the Capture Token Class) becomes the new Token Class (TXCLASS). RQFINAL may only be asserted when RQRDY is asserted and RQSEND is deasserted. RQFINAL is ignored unless RQRDY has been asserted for at least one byte time and the service parameters have been valid for at least three byte times. RQFINAL must be deasserted not later than two byte times after RQSEND is deasserted.

7.0 Signal Descriptions (Continued)

Service Parameters

The Service Parameters define the Service Request. They must be valid for at least one byte time before the RQRDY signal is asserted and must not change while RDRDY remains asserted. See Section 5.3.1 for the encoding of RQRCLS.

The Requested Service corresponds to the Request Service Class and the Token Class parameters of the (SM_)MA_DATA.request and (SM_)MA-Token.request primitives as specified in the Standard.

Encoded into each of the 14 possible values of RQRCLS in the Service Class (Non-Restricted Asynchronous, Restricted Asynchronous, Synchronous, Immediate), the Token Capture and Issue Class, and THT Enable.

Requests are serviced on a Service Opportunity meeting the requested criteria.

External support is required to limit the requests presented to the MAC Interface by different MAC Users (SMT, LLC, etc.).

Symbol	Pin #	I/O	Description
RQRCLS(3-0)	19-22	I	<p>Request Class: Indicates the Service Class parameters for this request (see Section 5.3.1). When RQRCLS > 0, the Transmitter will capture a usable token (for non-immediate requests) and assert TXRDY. The Service Opportunity continues as long as the token is usable with the current service parameters, even if RQRDY is not asserted. If RQRCLS indicates a service class that is not serviceable for any cycle of a service opportunity, the service opportunity will conclude after the current frame and a token of the issue token class will be issued. If RQRCLS = 0, the Service Opportunity will terminate after the current frame and a token of the issue token class will be issued (even if RQRCLS subsequently becomes non-zero). See Table 5-3.</p>
RQCLM	15	I	<p>Request Claim: Indicates that this request is to be serviced in the Claim state. Ignored for non-immediate requests.</p>
RQBCN	16	I	<p>Request Beacon: Indicates that this request is to be serviced in the Beacon state. Ignored for non-immediate requests.</p>

Frame Options

The Frame Options signals are selected for each frame. They must be valid while the RQSEND signal is asserted. These options are typically used in bridging applications.

Symbol	Pin #	I/O	Description
STRIP	13	I	<p>Void Strip: Forces two My_Void frames to be transmitted on end of current Service Opportunity. Stripping continues until a My_Void frame returns. If any frame of a Service Opportunity requests this option, then all frames on that Service Opportunity will be stripped using this method.</p>
SAT	12	I	<p>Source Address Transparency: When SA transparency is selected, the SA from the data stream is transmitted in place of the internal MSA or MLA stored in the MAC Parameter RAM.</p>
SAIGT	11	I	<p>Source Address I/G Transparency: With this option, the MSB of the SA is sourced from the data stream, as opposed to being forced to zero.</p>
FCST	14	I	<p>Frame Check Sequence Transparency: When selected, the Ring Engine generated FCS is not appended to the end of the Information field.</p>

7.0 Signal Descriptions (Continued)

Transmission Status

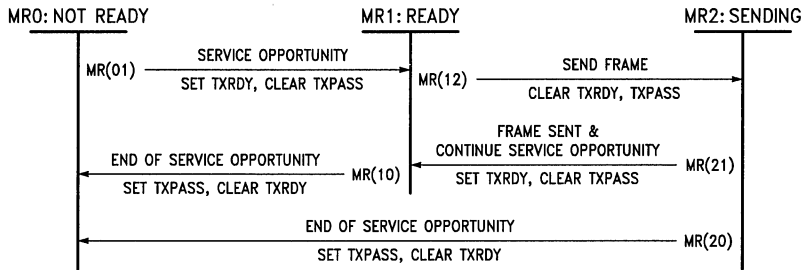
Symbol	Pin #	I/O	Description
TXED	31	O	Transmitted Ending Delimiter: Indicates that the Transmitter completed transmission of the current or previous PDU. TXED is asserted when the current PHY Request byte is a transmitted (not repeated) Ending Delimiter. It remains asserted until the beginning of either the next transmitted (not repeated) PDU or the next Service Opportunity. TXED is cleared by the Master Reset (bit MARST of the Function Register).
TXABORT	32	O	Transmission Aborted: Indicates that the Transmitter aborted transmission of the current or previous PDU before the Ending Delimiter, or that the current Service Opportunity was aborted by Reset or Recovery actions. TXABORT is asserted when the current transmitted (not repeated) PDU has been aborted, and remains valid until the beginning of either the next transmitted (not repeated) PDU or the next Service Opportunity. TXABORT is cleared by Master Reset (bit MARST of the Function Register). It is also cleared when an Immediate Claim or Beacon Service Opportunity is terminated by My__Claim or My__Beacon received (i.e., when transition T(47) or T(54) occurs during an Immediate Service Opportunity).
TXRINGOP	37	O	Ring Operational: Indicates the current value of the Ring__Operational flag.
TXCLASS	35	O	Token Class: Indicates the class of the current or previous token in the Transmitter. TXCLASS is set to R__Flag when a valid token is received. TXCLASS is set to the Issue Token Class when the RQRDY and RQFINAL signals are asserted (before Token FC time) for the current Service Opportunity. It is cleared by Reset and Recovery actions.
THTDIS	36	O	Token Holding Timer Disabled: Indicates that the Token Holding Timer was disabled when the current PHY Request byte was generated. THTDIS only changes between frames. When either signal TXRDY or TXPASS is asserted after a frame, THTDIS reflects the THT usage for that frame for at least two byte times. When TXPASS is asserted while THTDIS is asserted it indicates that TRT expired.

7.0 Signal Descriptions (Continued)

7.4.3 Operation

The MAC Request Interface has three logical states as determined by TXRDY and TXPASS. The interface state machine is shown in *Figure 7-11* followed by a description of the conditions, states and transitions.

State	Description	TXRDY	TXPASS
MR0: Not Ready	Ring Engine is not ready to service a request	0	1
MR1: Ready	Ring Engine is ready to transmit a frame	1	0
MR2: Sending	Ring Engine is sending a frame	0	0



TL/F/10387-18

FIGURE 7-11. MAC Request Interface State Machine

Conditions

- Send Frame** A frame can be sent from the interface when at least 8 bytes of preamble have been transmitted, TXRDY, RQRDY and RQSEND are asserted, and RQFINAL has not yet been asserted for this request.
- Service Opportunity** A Service Opportunity occurs when it is possible to service the current request, as defined by the current service parameters (RQRCLS, RQCLM and RQBCN). The rules for servicing requests are described in Section 5.2.
- Continue Service Opportunity** A Service Opportunity is continued after the current frame if valid service parameters continue to be presented during the frame, and the timer(s) used for the (next) requested service class have not reached their threshold.
- End of Service Opportunity** The end of a Service Opportunity occurs when it is no longer necessary or possible to continue the Service Opportunity. The service parameters are continuously compared with the current state of the Transmitter. If an unserviceable request is presented or any timer threshold is reached, the Service Opportunity will not continue after the current frame (if any).

Table 7-4 shows the timer thresholds used to determine if a Service Opportunity is possible for each service class.

TABLE 7-4. Thresholds Used to Determine Service Opportunities

Request Service Class	Threshold
All Requests	TRT Expiration
All Requests with THT Enabled	THT Expiration
Priority Asynchronous Requests	Asynchronous Priority Threshold

7.0 Signal Descriptions (Continued)

State Descriptions

MR0: Not Ready

In this state the Ring Engine does not have a Service Opportunity. If RQRCLS is not zero, the Ring Engine is trying to secure a Service Opportunity meeting the requested service parameters.

On a valid Service Opportunity, the MR(01) transition is taken. The status signals TXED and TXABORT are cleared and TXRDY is set to indicate that the Transmitter is ready to service a request.

MR1: Ready

In this state the Ring Engine has secured a Service Opportunity and is ready to service the current request. The Transmitter is sourcing Preamble, Fill or internally generated Void (from the Data state), Claim (from the Claim state) or Beacon (from the Beacon state) frames.

The Service Opportunity is governed by the requested service parameters. If an unserviceable request is presented for one or more cycles the Service Opportunity ends. If THT expires or a priority threshold is reached, the Service Opportunity will end immediately or after the next frame, depending on the state of the send window.

The send window is an opportunity to send a frame without being interrupted by time thresholds. The Service Opportunity may end during a send window if the service parameters change.

The send window opens each time TXRDY is asserted (entry to MR1). It remains open for a minimum of 6 byte times. The send window also opens if RQRDY is asserted while TXRDY is asserted, and if TXPASS is not asserted within one byte time after RQRDY is asserted.

The send window is held open until

- 1) RQSEND or RQFINAL is asserted or
- 2) until L_Max has expired (3.20 μ s), a Void frame has been sent (0.96 μ s or 1.60 μ s), and 7 more bytes of preamble have been sent (0.56 μ s). (When Option.IRPT = 1 this condition does not apply.)

At any time after RQRDY has been asserted and the final frame of the request has been sent, RQFINAL may be asserted to indicate that a token of the Issue Token Class should be transmitted at the end of the current Service Opportunity. If the MR(10) transition occurs while RQFINAL is asserted, and all the other conditions for accepting RQFINAL hold, the transmitted token will be of the Issue Token Class.

After RQFINAL has been asserted, no more frames can be sent until RQRDY has been deasserted and then reasserted. RQRDY should be deasserted and the service parameters updated to reflect the next request (if any) as soon as possible, to allow the Ring Engine to make better ring scheduling decisions. If RQRDY is not deasserted by the end of the last frame of a Service Opportunity, a Void frame will be transmitted before the token.

When the Service Opportunity ends, the MR(10) transition is taken and TXPASS is asserted to indicate the end of the current Service Opportunity. RQFINAL (and RQRDY) must be asserted not later than one byte time after TXPASS to insure that a token of the appropriate issue class is issued.

When a frame can be sent from the interface, the MR(12) transition is taken and TXACK is set to indicate that the Transmitter is sending the frame.

The state diagram for the internal substates within state MR1 is shown in *Figure 7-12*.

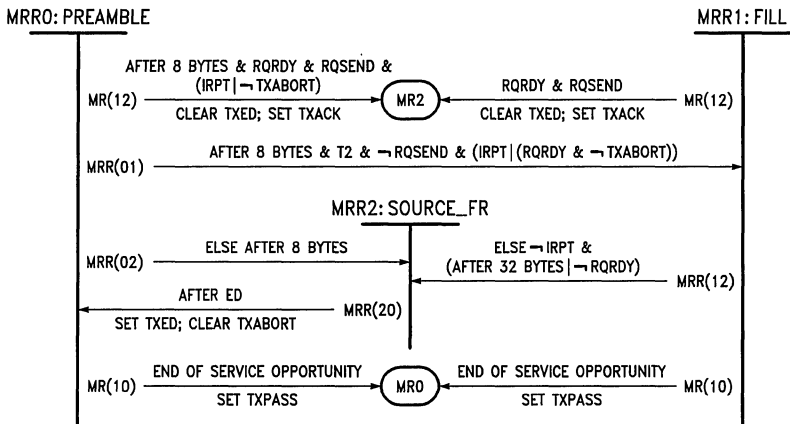


FIGURE 7-12. MR1 Substate Diagram

TL/F/10387-19

7.0 Signal Descriptions (Continued)

SUBSTATE MRR0: Preamble

Upon entry to MR1, 8 bytes of Preamble (Idles) are transmitted in substate MRR0.

After the Preamble, if a frame can be sent from the interface, transition MR12 occurs. The frame options are latched, TXED is cleared and TXACK is asserted.

If a frame cannot be sent from the interface, either Fill (additional Idles) or an internally generated frame (Void, Claim, or Beacon) is transmitted.

SUBSTATE MRR1: Fill

For requests, if RQRDY is asserted (indicating that the current request has been selected for service) or Option.IRPT is set (indicating that the ring is being interrupted), additional fill bytes (Idles) are transmitted in substate MRR1.

Fill continues until:

- 1) a frame can be sent from the interface, or
- 2) the Service Opportunity ends, or
- 3) 32 bytes of Idles are transmitted or RQRDY is deasserted. After that, an internal Void frame is generated in substate MRR2. (If Option.IRPT is set Void frames are not generated.)

If RQRDY is not asserted, if RQCLM or RQBCN is asserted, or (unless Option.IRPT is set) if the previous frame in the current Service Opportunity was aborted, an internal Void, Claim or Beacon frame is generated in substate MRR2.

At the end of an internal frame, TXED is set, TXABORT is cleared and another preamble is generated in substate MRR0.

MR2: Sending

In this state the Ring Engine is transmitting a frame from the MAC Request Interface. While the frame is being sent, if an unserviceable request is presented or any timer threshold is reached, the Service Opportunity will end after the current frame.

This implies that a Service Opportunity is never longer than TMAX plus one maximum length frame interval for Immediate Requests (unless Option.IRPT is set), or TNEG plus one maximum length frame interval for Non-immediate Requests. The maximum length of the frame interval is the maximum send window open time ($4.64 \mu\text{s}$ – $5.28 \mu\text{s}$) plus F_{max} . F_{max} is the maximum length of a frame, including 2 bytes of Preamble. The default value of F_{max} for FDDI is 4500 bytes = $360.00 \mu\text{s}$.

On entry to MR2 TXACK is asserted. It remains asserted while data is being accepted from the interface. RQSEND must be deasserted within one byte time after entering MR2.

At any time after RQSEND is deasserted for the final frame of a request, RQFINAL may be asserted to indicate that a token of the Issue Token Class should be transmitted at the end of the current Service Opportunity. If the MR(20) transition occurs while RQFINAL is asserted, and all the other conditions for accepting RQFINAL hold, the transmitted token will be the issue token class.

After RQFINAL has been asserted, no more frames can be sent until RQRDY has been deasserted and then reasserted. RQRDY should be deasserted and the service parameters updated to reflect the next requests (if any) as soon as possible, to allow the Ring Engine to make better ring scheduling decisions.

The last byte of data at the interface is indicated by RQEOF, which must be asserted with the last byte of data. After the Ending Delimiter of a frame is transmitted, TXED is asserted. When not using FCS transparency, TXED is asserted 7 byte times after RQEOF is asserted. When using FCS transparency, it is asserted 3 byte times after RQEOF is asserted. TXACK is deasserted no later than one byte time after RQEOF is asserted.

At any time during frame transmission, TXABORT may be asserted. This indicates that the frame was aborted due to internal errors, buffering errors, parity errors, RQABT, MAC reset, reception of a MAC frame etc. TXACK is deasserted no later than TXABORT is asserted. When a transmission is aborted due to an error (and Option.IRPT is not set), a Void frame is transmitted to reset the TVX timers in all stations in the ring.

After a successful or unsuccessful frame transmission, if the current Service Opportunity can be continued transition MR(21) occurs and TXRDY is asserted; otherwise transition MR(20) occurs and TXPASS is asserted.

If at any time during a frame transmission, the end of Service Opportunity condition is detected, transition MR(20) will occur after the current frame transition.

7.0 Signal Descriptions (Continued)

7.4.3.3 Transmission Status

Upon leaving MR2, transmission status is available after TXRDY or TXPASS is asserted. TXED and TXABORT are normally valid for at least 9 byte times (exception: 2 byte times when an Immediate Service Opportunity ends without issuing a token, and another Service Opportunity begins immediately upon return to state MR0). THTDIS is valid for at least 2 byte times. When TXPASS is deasserted and for at least two byte times after is reasserted, TXCLASS denotes the token that will be issued at the end of the current Service Opportunity.

TXED indicates that the Ending Delimiter of the previous PDU was transmitted. TXABORT indicates that the previous frame was aborted as a result of a request abort (RQABT), an internal error or the Reset or Recovery Required conditions became true.

If TXED is asserted, TXABORT may also be asserted (within 9 byte clocks) if this station backed off to another station after a complete Claim frame was transmitted.

When transmitting Claim/Beacon frames from the Transmitter Claim or Beacon, if TXPASS is asserted the Claim or Beacon Process has completed. In this case, TXABORT indicates if this station won (TXABORT = 0) or lost (TXABORT = 1) the Claim or Beacon process.

The interpretation of TXED and TXABORT is given in Table 7-5.

TABLE 7-5. Transmission Status

TXED	TXABORT	Condition
0	0	After a Master Reset or frame aborted during successful immediate Claim or Beacon service due to My_Claim or My_Beacon.
1	0	Complete frame transmitted.
0	1	Frame aborted.
1	1	Complete frame transmitted, followed by reset or recovery actions or unsuccessful immediate Claim or Beacon service due to timeout.

If TXPASS is asserted and THT was disabled during the last frame that was transmitted (THTDIS is asserted), TRT has expired. This is a serious error and indicates that there was

an over-allocation of synchronous bandwidth or a station used more than it was allocated. The ring will likely be claiming when this occurs.

7.4.4 Timing Examples

Several example sequences of the MAC Request Interface are provided. While this in no way is an exhaustive list of sequences, several likely sequences are shown. It is useful to follow the state diagrams of Section 7.4.3 (*Figures 7-11 and 7-12*) while examining the scenarios.

The timing is shown for all signals available at the MAC Interface.

The data shown in MRD and PRD are the data at those interfaces.

The data at PRD is duplicated with the TXED to show its relationship with the transmitted Ending Delimiter.

TXPASS and TXRDY show the relationship to the data at the transmitter. This is one byte time before the data is transmitted. The relationship to incoming tokens is shown explicitly.

RQRCLS contains the Requested service class. In several examples this is shown as generic requests (r1, r2) to make the examples more general purpose. The RQCLM and RQBCN signals are not shown, but have the same timing as the RQRCLS signals.

The Frame Options are grouped together since they have the same timing. These include the SAT, SAIGT, FCST and STRIP options.

Single Frame Transmission with Prestaging

Prestaging refers to the staging of data before the Service Opportunity begins. Prestaging occurs in interfaces where data is loaded into a FIFO or dedicated memory used as a FIFO before the token arrives.

In *Figure 7-13* RQRDY is asserted one byte time after RQRCLS has been passed to the interface. At this point the Ring Engine is awaiting an appropriate Service Opportunity. Upon capture of a usable token, TXRDY is asserted. TXRDY causes RQSEND to be asserted.

RQSEND causes TXRDY to be deasserted, which in turn causes RQSEND to be deasserted.

Notice that after RQSEND is deasserted, RQFINAL is asserted for one cycle to indicate that the issue token class should be used for the token. RQRDY is then deasserted and RQRCLS is set to zero. Since RQRCLS goes to zero, the end of Service Opportunity condition becomes true and the token is issued at the end of the current frame.

7.0 Signal Descriptions (Continued)

If RQRCLS remained asserted the token would be held as long as possible and multiple frames could be transmitted. In this case the \uparrow TXRDY \rightarrow \uparrow RQSEND \rightarrow \downarrow TXRDY \rightarrow \downarrow RQSEND handshake for the beginning of each frame remains identical.

Single Frame Transmission without Prestaging

In *Figure 7-14*, prestaging is not used. Multiple requests are present at the interface, of which only the highest priority request is presented to the interface. RQRCLS is changing because higher priority requests become ready to be serviced. The scheduling decision is made until a Service Opportunity occurs. Once TXRDY is asserted, RQRDY is asserted and the r6 request is serviced.

When the data associated with r6 is ready to be transmitted, RQSEND is asserted. This in turn causes TXRDY to be deasserted when transmission begins (entrance to MR2). The deassertion of TXRDY causes RQSEND to be deasserted.

During the first frame of the request, the end of Service Opportunity condition becomes true as a result of:

THT reaching the THT priority threshold if the request was an asynchronous priority request,

THT expiration if the request was an asynchronous request or

TRT expiration if the request was a synchronous request.

TXPASS is asserted to indicate that this Service Opportunity is complete.

If RQRCLS remains greater than 0, the next usable token will be captured and servicing of the request will continue. If RQRCLS remained asserted the token would be held as long as possible and multiple frames could be transmitted. In this case the \uparrow TXRDY \rightarrow \uparrow RQSEND \rightarrow \downarrow TXRDY \rightarrow \downarrow RQSEND handshake for the beginning of each frame remains identical.

Aborted Frame Transmission

A transmission as in *Figure 7-14* is started. During the transmission, an interface error occurs (for example) and RQABT is asserted to cause the current frame to be aborted (see *Figure 7-15*). TXACK is then deasserted and TXABORT is asserted to indicate that the frame was aborted as a result of a FIFO underrun or an equivalent reason. This is signaled with RQABT. After the frame is aborted, TXRDY is asserted to indicate that another frame may be transmitted. Since no frames are ready to be transmitted a Void fill frame is transmitted. During the Void frame transmission, the interface then sets RQRCLS to zero to indicate that the Token should be issued. TXPASS is then asserted once the Ending Delimiter of the Void frame is transmitted.

In this scenario the transmitted Void frame serves two purposes. It is transmitted because the interface was stalling waiting for another frame and also in response to the aborted frame. A Void frame is transmitted every time a transmission is aborted.

MAC Reset

In *Figure 7-16*, a MAC reset occurs during a frame transmission. This causes the current frame to be aborted and the Ring Operational Flag (TXRINGOP) to be deasserted. TXPASS is asserted with TXABORT after the frame is aborted. Since the ring is not operational, no Void frames are transmitted.

In *Figure 7-16* the MAC Reset occurs while the Ending Delimiter is being transmitted. In *Figure 7-17* the boundary case is shown where the MAC Reset occurs during the Frame Status. Note that the Ending Delimiter of the frame is transmitted with the frame status. TXRDY is asserted for one cycle followed by TXPASS with TXABORT.

Synchronous Request followed by Asynchronous Request

In *Figure 7-18*, frames from two requests are serviced on the same Service Opportunity. Once the synchronous frame is being transmitted, the RQRCLS is changed to that for the asynchronous frame. At the end of the synchronous frame TXRDY is asserted since the token is still usable for the asynchronous request. RQRDY is then asserted and the Asynchronous frame is then transmitted.

Notice that the value of THTDIS changes after the Frame Status for the synchronous frame is transmitted. THT is disabled for synchronous transmission and enabled for normal asynchronous transmission.

Restricted Begin

In *Figure 7-19*, a restricted dialogue is begun. A non-restricted Token is captured, a single frame is transmitted and a Restricted Token is issued.

An Rbeg Request is a request to capture a Non-restricted Token and issue a Restricted Token. Since there is only one frame in this example, RQFINAL is asserted during the first frame. In the example, RQFINAL is asserted one byte time after RQSEND is deasserted while RQRDY is still asserted, but it may be asserted anytime while RQRDY is asserted. Notice that TXCLASS changes to restricted after RQFINAL is asserted.

Immediate Claim

In *Figure 7-20*, an immediate Claim frame is transmitted from the Claim state.

A Lower__Claim frame is received from an upstream station, causing this station to enter its Claim state and deassert TXRINGOP. RQRCLS is set to immediate and RQCLM is asserted.

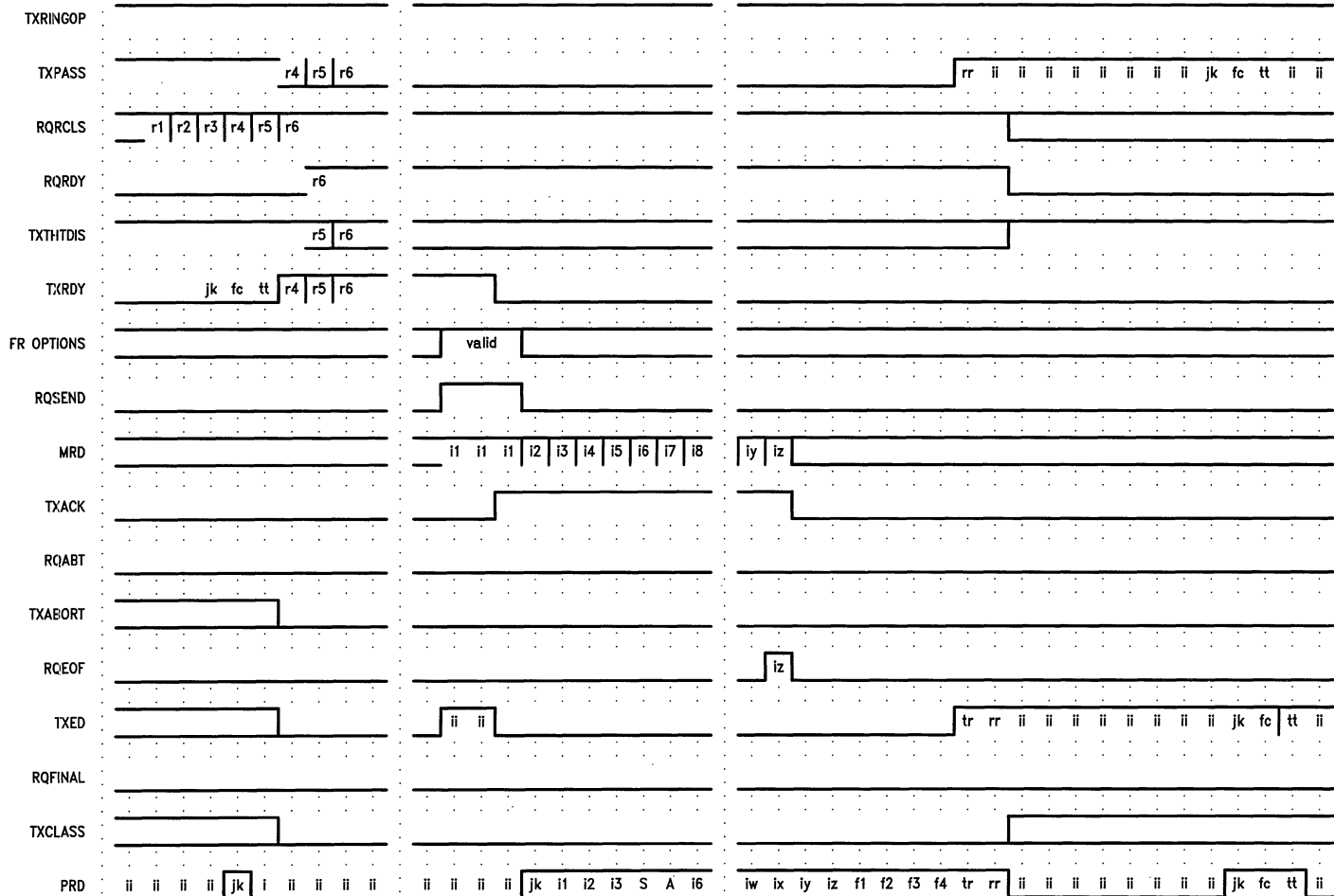
An internally generated Claim frame is first transmitted (at least one internally generated Claim or Beacon frame is always transmitted upon entry to the Claim or Beacon state). After the internally generated Claim frame is transmitted, TXRDY is asserted since the transmitter is still in the Claim state (the ring can hold more than one Claim frame). The frame is then transmitted following the normal handshake.

Similar timing applies for externally generated Beacon frames.

Remember that for Immediate Requests from the Claim and Beacon States, RQSEND must be asserted no later than 8 byte times after TXRDY is asserted. This guarantees that a minimum size preamble will be generated.

After the frame is transmitted, TXRDY is asserted again since the transmitter is still in the Claim state.

If this station wins the Claim Process TXPASS is asserted without TXABORT. If another station causes this station to backoff (this station receives a Higher__Claim), TXPASS is asserted with TXABORT.



2-230

FIGURE 7-14. Asynchronous Request without Prestaging

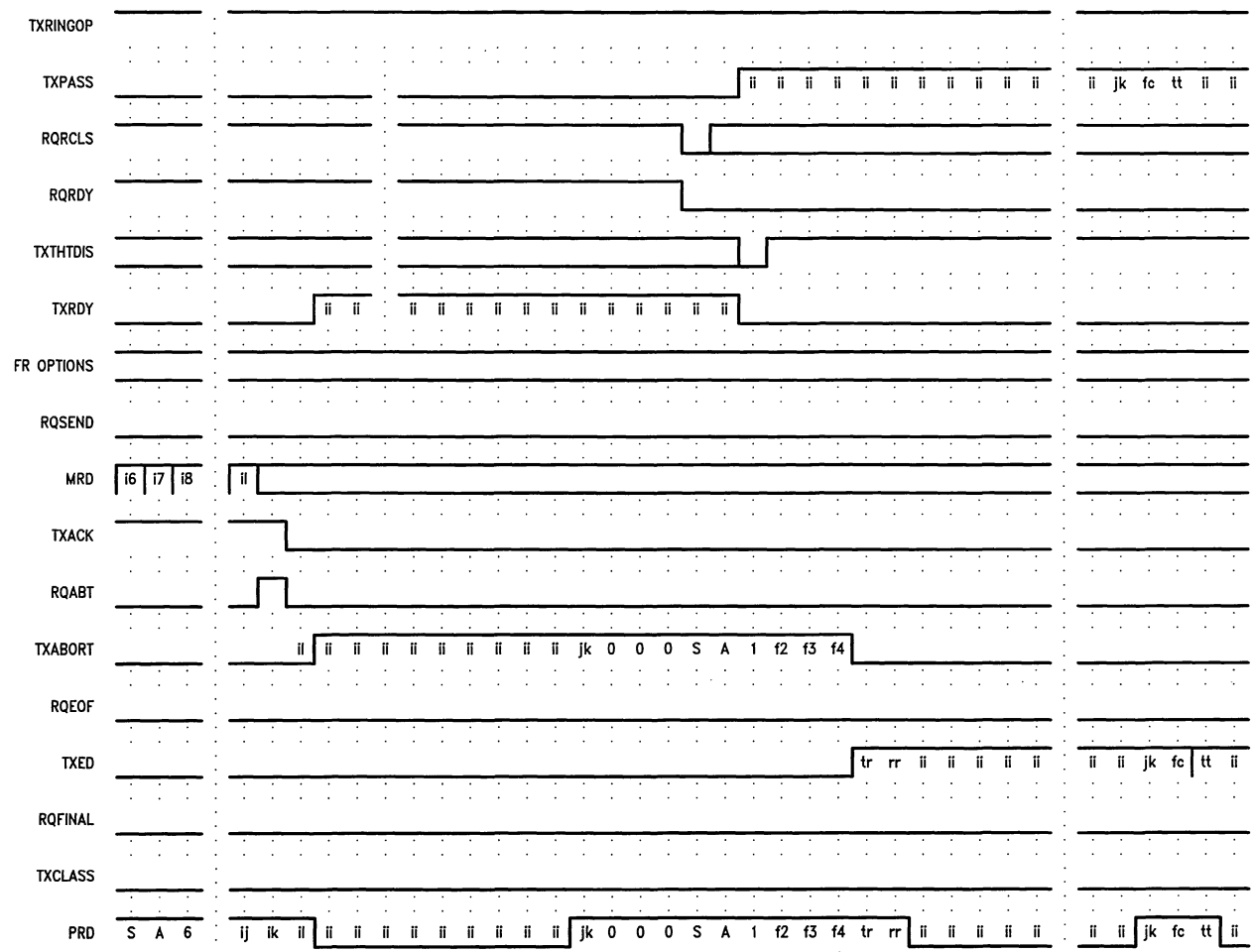


FIGURE 7-15. Frame Abort

TL/F/10387-22

2-231

7.0 Signal Descriptions (Continued)

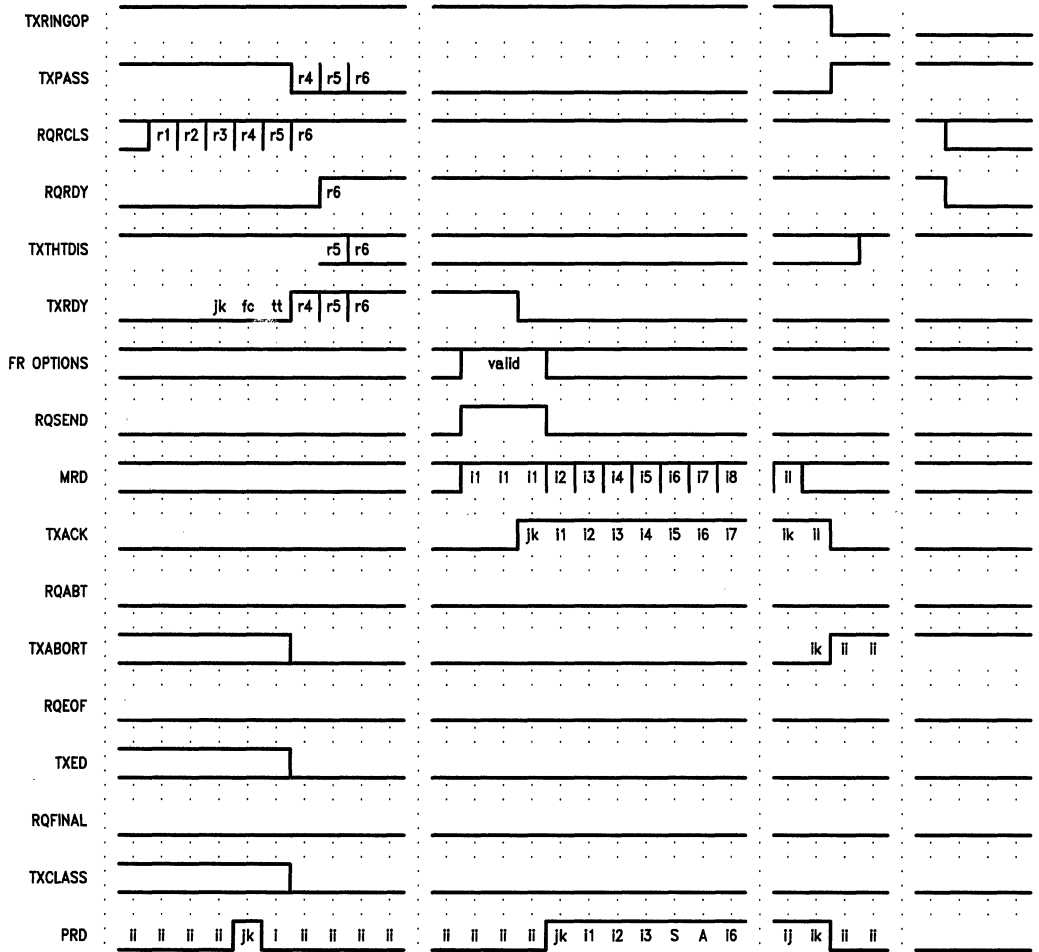


FIGURE 7-16. MAC Reset

TL/F/10387-23

7.0 Signal Descriptions (Continued)

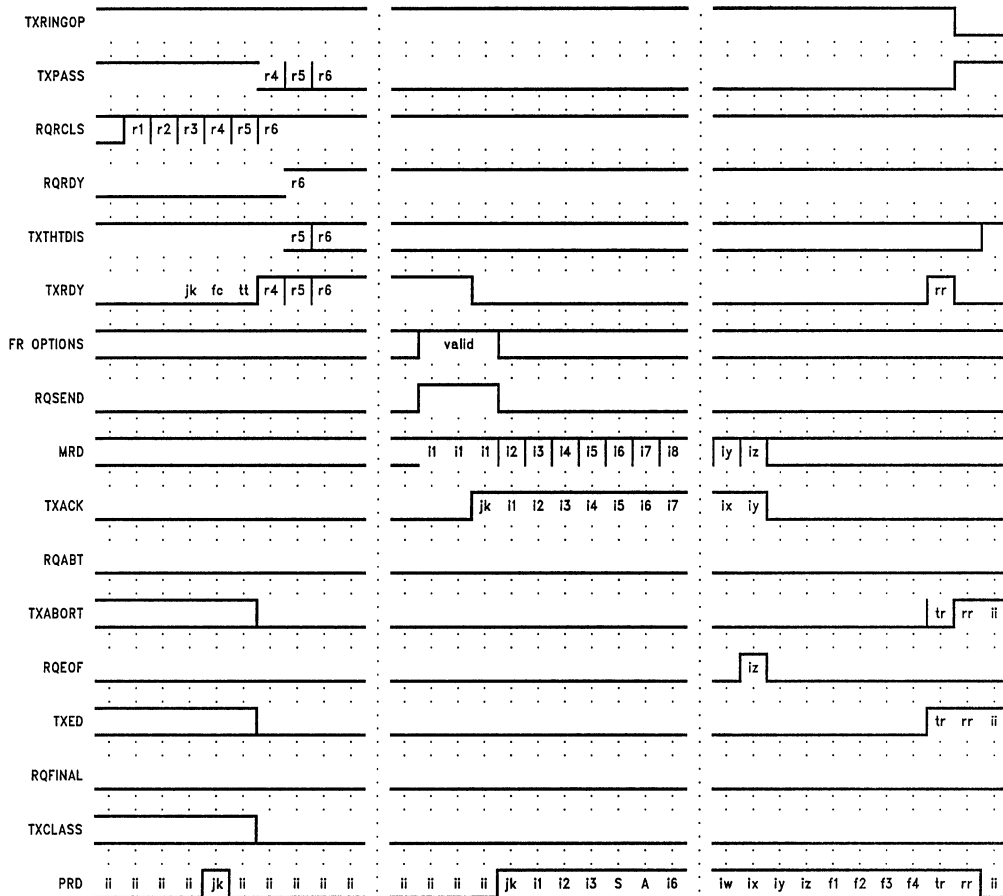


FIGURE 7-17. MAC Reset at End of Frame

TL/F/10387-24

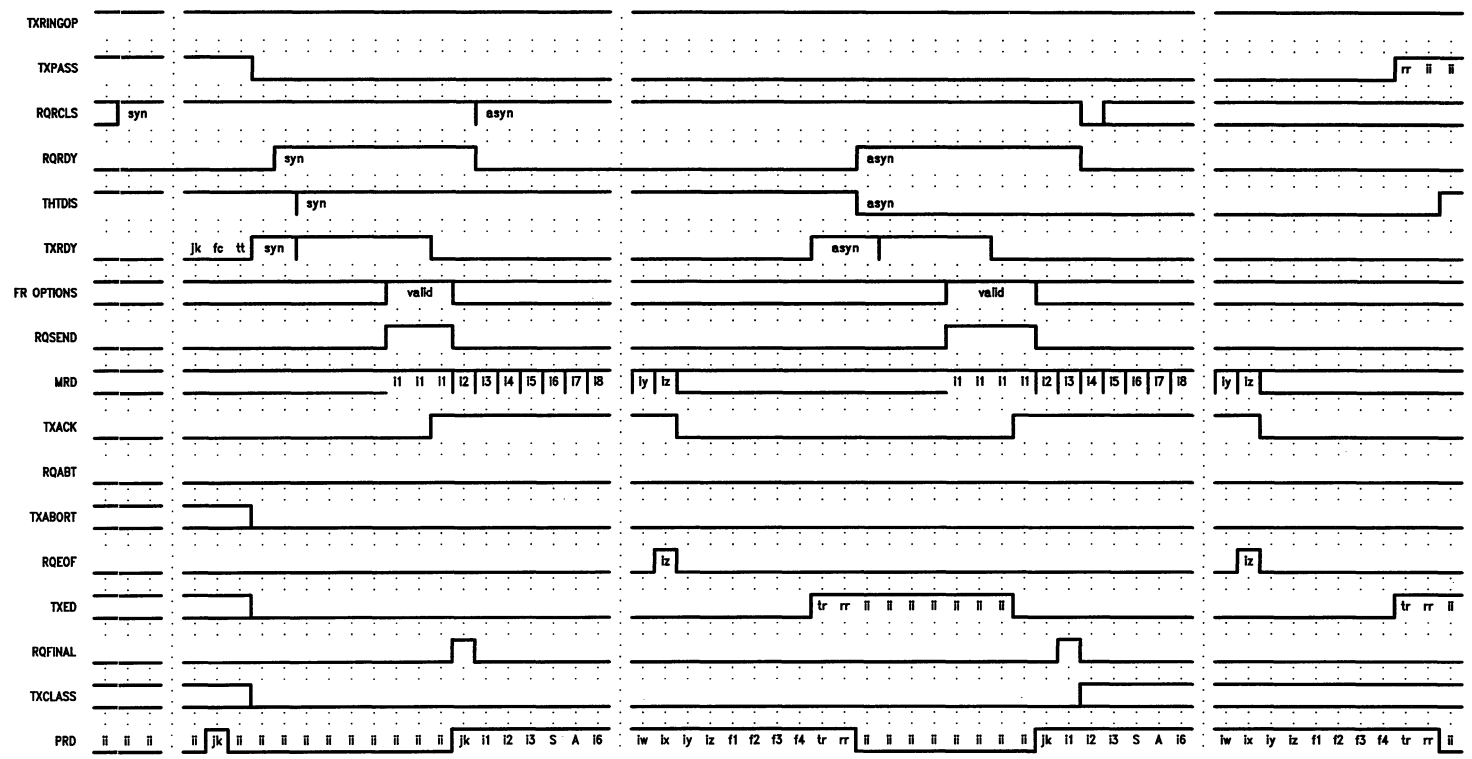


FIGURE 7-18. Synchronous Request followed by Asynchronous

2-234

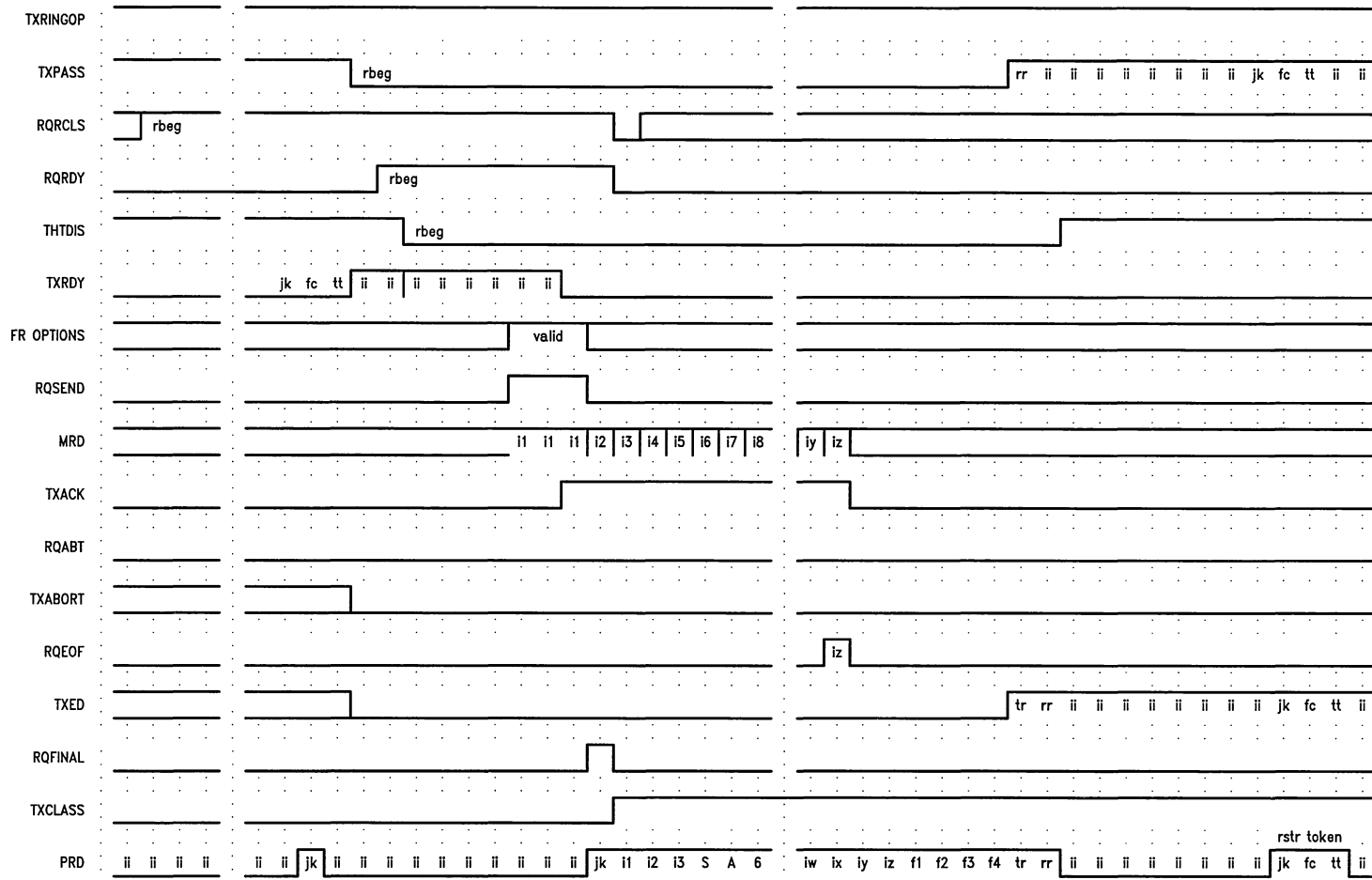


FIGURE 7-19. Restricted Begin

TL/F/10387-26

2-235

2-236

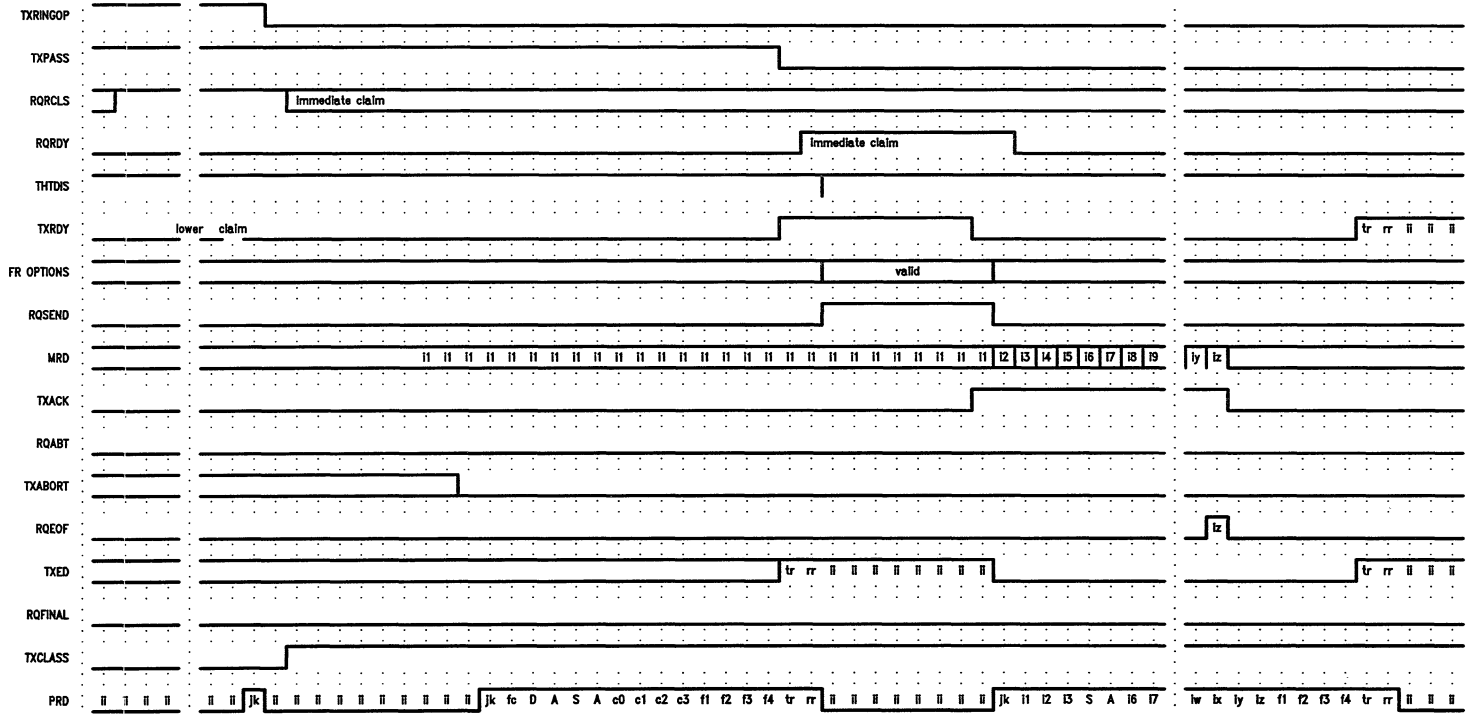


FIGURE 7-20. Immediate Claim

TL/F/10387-27

7.0 Signal Descriptions (Continued)

7.5 ELECTRICAL INTERFACE

The Electrical Interface signals comprise all of the clocking, power supply, and ground pins.

Symbol	Pin #	I/O	Description
LSC	87	I	Local Symbol Clock: 25 MHz clock with a 40/60 duty-cycle. Typically generated by the CDD.
LBC	86	I	Local Byte Clock: 12.5 MHz clock 50/50 duty-cycle in phase with LSC. Typically generated by the CDD.
RST	118	I	Master Reset: Equivalent to setting the Master Reset bit in the Function Register. An asynchronous input that must be asserted for at least 5 LSC clock cycles. When asserted, all bi-directional signals are tri-stated. Active low signal.
V _{CC} [11]	4, 17, 34 58, 78 94, 100 106, 117	I	Positive Power Supply: 5V, ±5% relative to GND.
GND[11]	5, 18, 33 57, 77, 88–91, 101, 116, 128	I	Power Supply Return

7.6 PINOUT SUMMARY

TABLE 7-6. Pinout Summary

Pin #	Signal Name	Symbol	I/O
1	Control Bus Data 1	CBD1	I/O
2	Control Bus Data 2	CBD2	I/O
3	Control Bus Data 3	CBD3	I/O
4	Positive Power Supply	V _{CC}	I
5	Ground	GND	I
6	Control Bus Data 4	CBD4	I/O
7	Control Bus Data 5	CBD5	I/O
8	Control Bus Data 6	CBD6	I/O
9	Control Bus Data 7	CBD7	I/O
10	Control Bus Parity	CBP	I/O
11	Source Address I/G Transparency	SAIGT	I
12	Source Address Transparency	SAT	I
13	Void Strip	STRIP	I
14	Frame Check Sequence Transparency	FCST	I
15	Request Claim	RQCLM	I
16	Request Beacon	RQBCN	I
17	Positive Power Supply	V _{CC}	I
18	Ground	GND	I
19	Request Class 3	RQRCLS3	I

7.0 Signal Descriptions (Continued)**7.6 PINOUT SUMMARY** (Continued)**TABLE 7-6. Pinout Summary** (Continued)

Pin #	Signal Name	Symbol	I/O
20	Request Class 2	RQRCLS2	I
21	Request Class 1	RQRCLS1	I
22	Request Class 0	RQRCLS0	I
23	Request Ready	RQRDY	I
24	Request Send	RQSEND	I
25	Request End of Frame	RQEOF	I
26	Request Final	RQFINAL	I
27	Request Abort	RQABT	I
28	Transmit Pass	TXPASS	O
29	Transmit Ready	TXRDY	O
30	Transmit Acknowledge	TXACK	O
31	Transmit Ending Delimiter	TXED	O
32	Transmit Abort	TXABORT	O
33	Ground	GND	I
34	Positive Power Supply	V _{CC}	I
35	Token Class	TXCLASS	O
36	Token Holding Timer Disabled	THTDIS	O
37	Ring Operational	TXRINGOP	O
38	External A__Flag	EA	I
39	External M__Flag	EM	I
40	Positive Power Supply	V _{CC}	I
41	MAC Request Parity	MRP	I
42	MAC Request Data 7	MRD7	I
43	MAC Request Data 6	MRD6	I
44	MAC Request Data 5	MRD5	I
45	MAC Request Data 4	MRD4	I
46	MAC Request Data 3	MRD3	I
47	MAC Request Data 2	MRD2	I
48	MAC Request Data 1	MRD1	I
49	MAC Request Data 0	MRD0	I
50	Ground	GND	I
51	Receive Start	RCSTART	O
52	Frame Control Received	FCRCVD	O
53	Short/Long Address Flag	FCSL	O
54	Individual/Group Address Flag	DAIG	O
55	Destination Address Received	DARCVD	O
56	My Destination Address Recognized	AFLAG	O
57	Ground	GND	I
58	Positive Power Supply	V _{CC}	I
59	Source Address Received	SARCVD	O

7.0 Signal Descriptions (Continued)**7.6 PINOUT SUMMARY** (Continued)**TABLE 7-6. Pinout Summary** (Continued)

Pin #	Signal Name	Symbol	I/O
60	My Source Address Recognized	MFLAG	O
61	Same Source Address	SAMESA	O
62	Information Field Received	INFORCVD	O
63	Same MAC Information	SAMEINFO	O
64	Ground	GND	I
65	Positive Power Supply	V _{CC}	I
66	Ending Delimiter Received	EDRCVD	O
67	Valid Frame Check Sequence	VFCS	O
68	Valid Data Length	VDL	O
69	Token Received	TKRCVD	O
70	Format Error	FOERROR	O
71	Frame Stripped	FRSTRP	O
72	Media Access Control Reset	MCRST	O
73	MAC Indicate Parity	MIP	O
74	MAC Indicate Data 7	MID7	O
75	MAC Indicate Data 6	MID6	O
76	MAC Indicate Data 5	MID5	O
77	Ground	GND	I
78	Positive Power Supply	V _{CC}	I
79	MAC Indicate Data 4	MID4	O
80	MAC Indicate Data 3	MID3	O
81	MAC Indicate Data 2	MID2	O
82	MAC Indicate Data 1	MID1	O
83	MAC Indicate Data 0	MID0	O
84	Ground	GND	I
85	Valid Copy	VCOPY	I
86	Local Byte Clock	LBC	I
87	Local Symbol Clock	LSC	I
88	Ground	GND	I
89	Ground	GND	I
90	Ground	GND	I
91	Ground	GND	I
92	PHY Request Data 0	PRD0	O
93	PHY Indicate Data 0	PID0	I
94	Positive Power Supply	V _{CC}	I
95	PHY Request Data 1	PRD1	O
96	PHY Indicate Data 1	PID1	I
97	PHY Request Data 2	PRD2	O
98	PHY Indicate Data 2	PID2	I
99	PHY Request Data 3	PRD3	O

7.0 Signal Descriptions (Continued)

7.6 PINOUT SUMMARY (Continued)

TABLE 7-6. Pinout Summary (Continued)

Pin #	Signal Name	Symbol	I/O
100	Positive Power Supply	V _{CC}	I
101	Ground	GND	I
102	PHY Indicate Data 3	PID3	I
103	PHY Request Data 4	PRD4	O
104	PHY Indicate Data 4	PID4	I
105	PHY Request Data 5	PRD5	O
106	Positive Power Supply	V _{CC}	I
107	PHY Indicate Data 5	PID5	I
108	PHY Request Data 6	PRD6	O
109	PHY Indicate Data 6	PID6	I
110	PHY Request Data 7	PRD7	O
111	PHY Indicate Data 7	PID7	I
112	PHY Request Control	PRC	O
113	PHY Indicate Control	PIC	I
114	PHY Request Parity	PRP	O
115	PHY Indicate Parity	PIP	I
116	Ground	GND	I
117	Positive Power Supply	V _{CC}	I
118	Master Reset	$\overline{\text{RST}}$	I
119	Read/ ~ Write	R/ $\overline{\text{W}}$	I
120	~ Control Bus Enable	$\overline{\text{CE}}$	I
121	~ Interrupt	$\overline{\text{INT}}$	O
122	~ Acknowledge	$\overline{\text{ACK}}$	O
123	Control Bus Address 0	CBA0	I
124	Control Bus Address 1	CBA1	I
125	Control Bus Address 2	CBA2	I
126	Control Bus Address 3	CBA3	I
127	Control Bus Address 4	CBA4	I
128	Ground	GND	I
129	Control Bus Address 5	CBA5	I
130	Control Bus Address 6	CBA6	I
131	Control Bus Address 7	CBA7	I
132	Control Bus Data 0	CBD0	I/O

7.0 Signal Descriptions (Continued)

7.7 PINOUT DIAGRAM

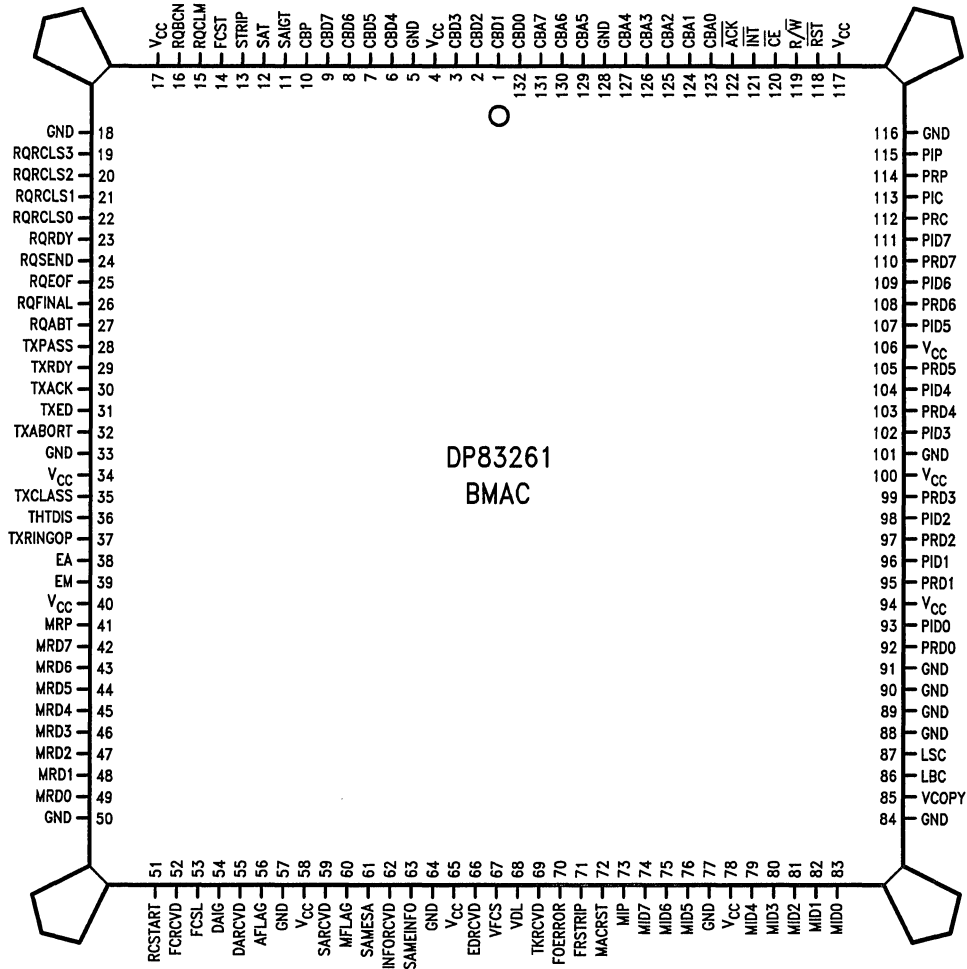


FIGURE 7-21. DP83261 132-Pin PQFP Pinout

TL/F/10387-11

Order Number DP83261AVF
See NS Package Number VF132A

8.0 Electrical Characteristics

8.1 ABSOLUTE MAXIMUM RATINGS

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Symbol	Parameter	Conditions	Min	Max	Units
V _{CC}	Supply Voltage		-0.5	7.0	V
V _{IN}	DC Input Voltage		-0.5	V _{CC} + 0.5	V
V _{OUT}	DC Output Voltage		-0.5	V _{CC} + 0.5	V
T _{STG}	Storage Temperature Range		-65	+150	°C
T _L	Lead Temperature	Soldering, 10 sec. (IR or Vapor) (Phase Reflow)		230	°C
	ESD Rating	R _{ZAP} = 1.5k, C _{ZAP} = 120 pF		800	V

8.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Conditions	Min	Max	Units
V _{CC}	Supply Voltage		4.75	5.25	V
PD	Power Dissipation			400	mW
T	Operating Temp		0	70	°C

8.3 DC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Max	Units
V _{OH1}	Minimum High Level Output Voltage	C _L = 50 pF	V _{CC} - 0.5		V
V _{OH2}	Minimum High Level Output Voltage	I _{OH} = -2 mA	2.4		V
V _{OL1}	Maximum Low Level Output Voltage	C _L = 50 pF		0.4	V
V _{OL2}	Maximum Low Level Output Voltage	I _{OL} = 4 mA		0.4	V
V _{OL3}	Maximum Low Level Output Voltage $\overline{\text{INT}}$ and $\overline{\text{ACK}}$ (Open Drain)	I _{OL} = 8 mA		0.4	V
V _{IH}	Minimum High Level Input Voltage		2.0		V
V _{IL}	Maximum Low Level Input Voltage			0.8	V
I _{IH}	Input High Current			+10	μA
I _{IL}	Input Low Current			-10	μA
I _{OZ1}	TRI-STATE Leakage for CBD(7-0) and CBP			±10	μA
I _{OZ2}	TRI-STATE Leakage for $\overline{\text{INT}}$ and $\overline{\text{ACK}}$ (Open Drain)			±10	μA
I _{OZ3}	Dynamic Supply Current	C _L = 50 pF, 12.5 MHz		70m	mA

8.0 Electrical Characteristics (Continued)

8.4 AC ELECTRICAL CHARACTERISTICS

See Figures 8-8 and 8-9 for AC Signal and TRI-STATE Testing Criteria.

8.4.1 Control Bus Interface

Symbol	Parameter	Min	Max	Units
T1	\overline{CE} Setup to LBC	15		ns
T2	LBC Period	80		ns
T3	LBC to \overline{ACK} Low		45	ns
T4	\overline{CE} Low to \overline{ACK} Low	290	540	ns
T5	LBC Low to CBD(7-0) and CBP Valid		60	ns
T6	LBC to CBD(7-0) and CBP Active		60	ns
T7	\overline{CE} Low to CBD(7-0) and CBP Active	225	475	ns
T8	\overline{CE} Low to CBD(7-0) and CBP Valid	265	515	ns
T9	LBC Pulse Width High	35	45	ns
T10	LBC Pulse Width Low	35	45	ns
T11	\overline{CE} High to \overline{ACK} High		45	ns
T12	R/ \overline{W} , CBA(7-0), CBD(7-0) and CBP Set up to \overline{CE} Low	5		ns
T13	\overline{CE} High to R/ \overline{W} , CBA(7-0), CBD(7-0) and CBP Hold Time	0		ns
T14	R/ \overline{W} , CBA (7-0), CBD(7-0) and CBP Setup to LBC	20		ns
T15	\overline{ACK} Low to \overline{CE} High Lead Time	0		ns
T16	\overline{CE} Minimum Pulse Width High	20		ns
T17	\overline{CE} High to CBD(7-0) and CBP TRI-STATE		55	ns
T18	\overline{ACK} High to \overline{CE} Low	0		ns
T19	CBD(7-0) Valid to \overline{ACK} Low Setup	20		ns
T20	LBC to \overline{INT} Low		55	ns

Asynchronous Definitions

T4 (min)	$T1 + (3 * T2) + T3$
T4 (max)	$T1 + (6 * T2) + T3$
T7 (min)	$T1 + (2 * T2) + T6$
T7 (max)	$T1 + (5 * T2) + T6$
T8 (min)	$T1 + (2 * T2) + T9 + T5$
T8 (max)	$T1 + (5 * T2) + T9 + T5$

Note: Min/Max numbers are based on T2 = 80 ns and T9 = T10 = 40 ns.

8.0 Electrical Characteristics (Continued)

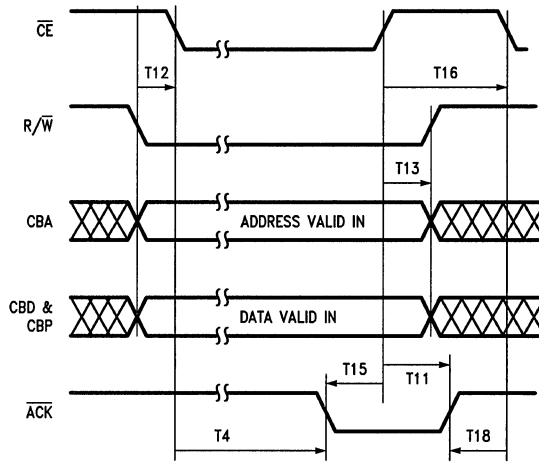


FIGURE 8-1. Control Bus Interface Write Cycle

TL/F/10387-28

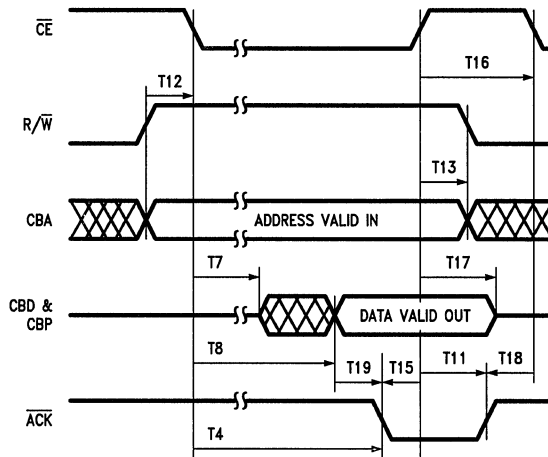
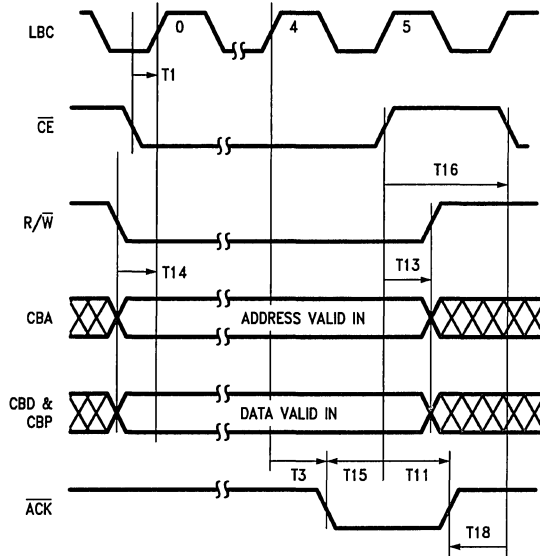


FIGURE 8-2. Control Bus Interface Read Cycle

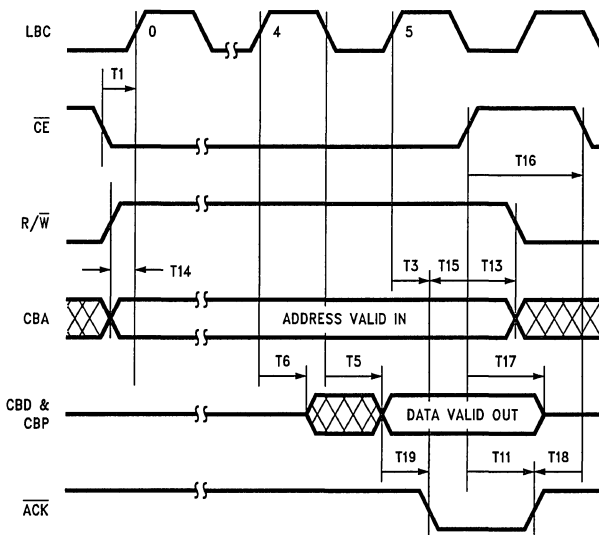
TL/F/10387-29

8.0 Electrical Characteristics (Continued)



TL/F/10387-30

FIGURE 8-3. Control Bus Interface Synchronous Write Cycle



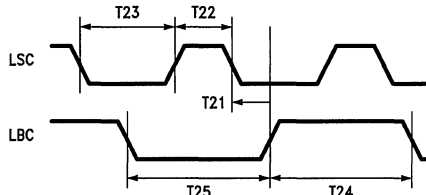
TL/F/10387-31

FIGURE 8-4. Control Bus Interface Synchronous Read Cycle

8.0 Electrical Characteristics (Continued)

8.4.2 Clock Signals

Symbol	Parameter	Min	Max	Units
T21	LSC to LBC Lead Time (Skew Left)	-4	6	ns
T22	LSC Pulse Width High	12		ns
T23	LSC Pulse Width Low	21		ns
T24	LBC Pulse Width High	35	45	ns
T25	LBC Pulse Width Low	35	45	ns

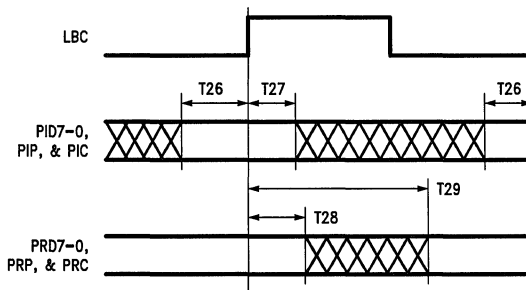


TL/F/10387-32

FIGURE 8-5. Clock Signals

8.4.3 PHY Interface

Symbol	Parameter	Min	Max	Units
T26	PHY Data Input Setup	15		ns
T27	PHY Data Input Hold	5		ns
T28	PHY Data Sustain	10		ns
T29	PHY Data Valid		45	ns



TL/F/10387-33

Note: All setup and hold testing is done on single edges only (i.e., no combined setup/hold testing is done for pulse signals. This implies that the signal makes only one low to high or high to low transition per cycle).

FIGURE 8-6. PHY Interface Timing

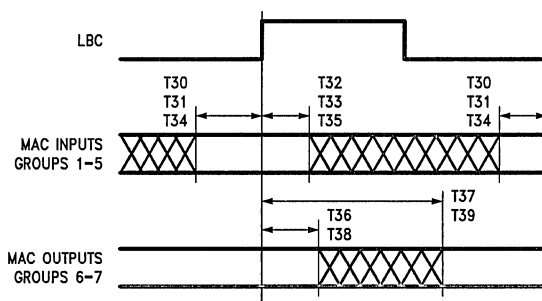
8.0 Electrical Characteristics (Continued)

8.4.4 MAC Interface

Pin Groups

Group #	I/O	Pins
1	I	SAIGT, SAT, STRIP, EA, VCOPY, RQEOF, RQSEND, RQFINAL
2	I	RQRDY
3	I	FCST, RQBCN, RQCLS(3-0), EM, RQABT
4	I	RQCLM
5	I	MRD(7-0), MRP
6	O	TXPASS, TXED, TXABORT, RCSTART, FCRCVD, SAMEA, INFOCVD, SAMEINFO, TXRDY, TXACK, TXCLASS, THTDIS, TXRINGOP, DIAG, DARCV, AFLAG, SARCV, MFLAG, EDRCVD, VFCS, VDL, TKRCVD, FOERROR, FRSTRIP, MACRST
7	O	MID(7-0), MIP

Symbol	Parameter	Min	Max	Units
T30	MAC Control Setup (Groups #1 and #3 and #4)	15		ns
T31	MAC Control Setup (Group #2)	30		ns
T32	MAC Control Hold (Group #3)	2		ns
T33	MAC Control Hold (Groups #1 and #2 and #4)	5		ns
T34	MAC Data Setup (Group #5)	15		ns
T35	MAC Data Hold (Group #5)	6		ns
T36	MAC Control Sustain (Group #6)	15		ns
T37	MAC Control Valid (Group #6)		45	ns
T38	MAC Data Sustain (Group #7)	15		ns
T39	MAC Data Valid (Group #7)		45	ns



TL/F/10387-34

Note: All setup and hold testing is done on single edges only (i.e., no combined setup/hold testing is done for pulse signals. This implies that the signal makes a single low to high or high to low transition per cycle).

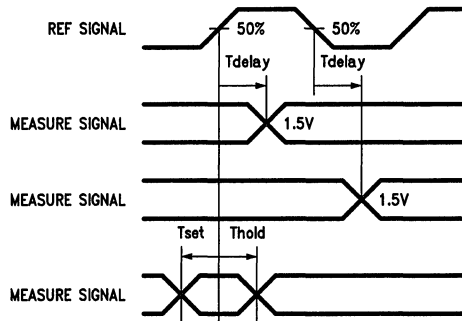
FIGURE 8-7. MAC Interface Timings

8.0 Electrical Characteristics (Continued)

Test Conditions for AC Testing

V_{IH}	3.0V
V_{IL}	0.0V
V_{OH}	1.5V
V_{OL}	1.5V
I_{OL}	8.0 mA (\overline{ACK} , \overline{INT})
C_L	50 pF

AC Signal Testing

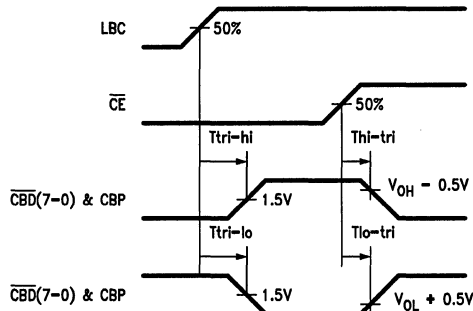


TL/F/10387-35

Note: All setup and hold testing is done on single edges only (i.e., no combined setup/hold testing is done for pulse signals. This implies that the signal makes only one single low to high or high to low transition per cycle).

FIGURE 8-8. A.C. Signal Testing

TRI-STATE Timing



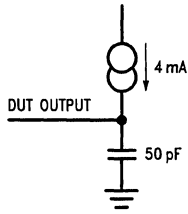
TL/F/10387-36

FIGURE 8-9. TRI-STATE Timing

8.0 Electrical Characteristics (Continued)

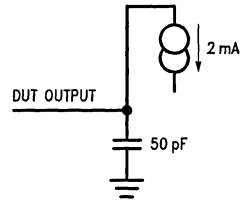
Test Equivalent Loads

V_{OL2} Testing



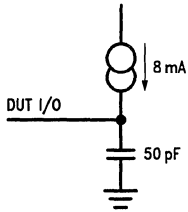
TL/F/10387-37

V_{OH2} Testing



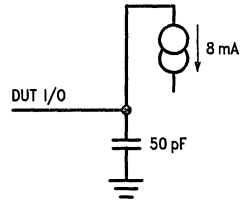
TL/F/10387-38

T_{lo-tri}



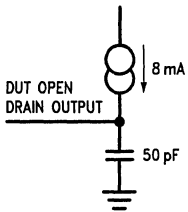
TL/F/10387-39

T_{hi-tri}



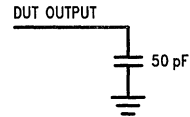
TL/F/10387-40

Open Drain V_{OL} Testing



TL/F/10387-41

AC, V_{OL1}, V_{OH1} Testing



TL/F/10387-42

FIGURE 8-10. Test Equivalent Loads

Appendix A. Ring Engine State Machines

A.1 RECEIVER

A.1.1 MAC Receiver State Diagram

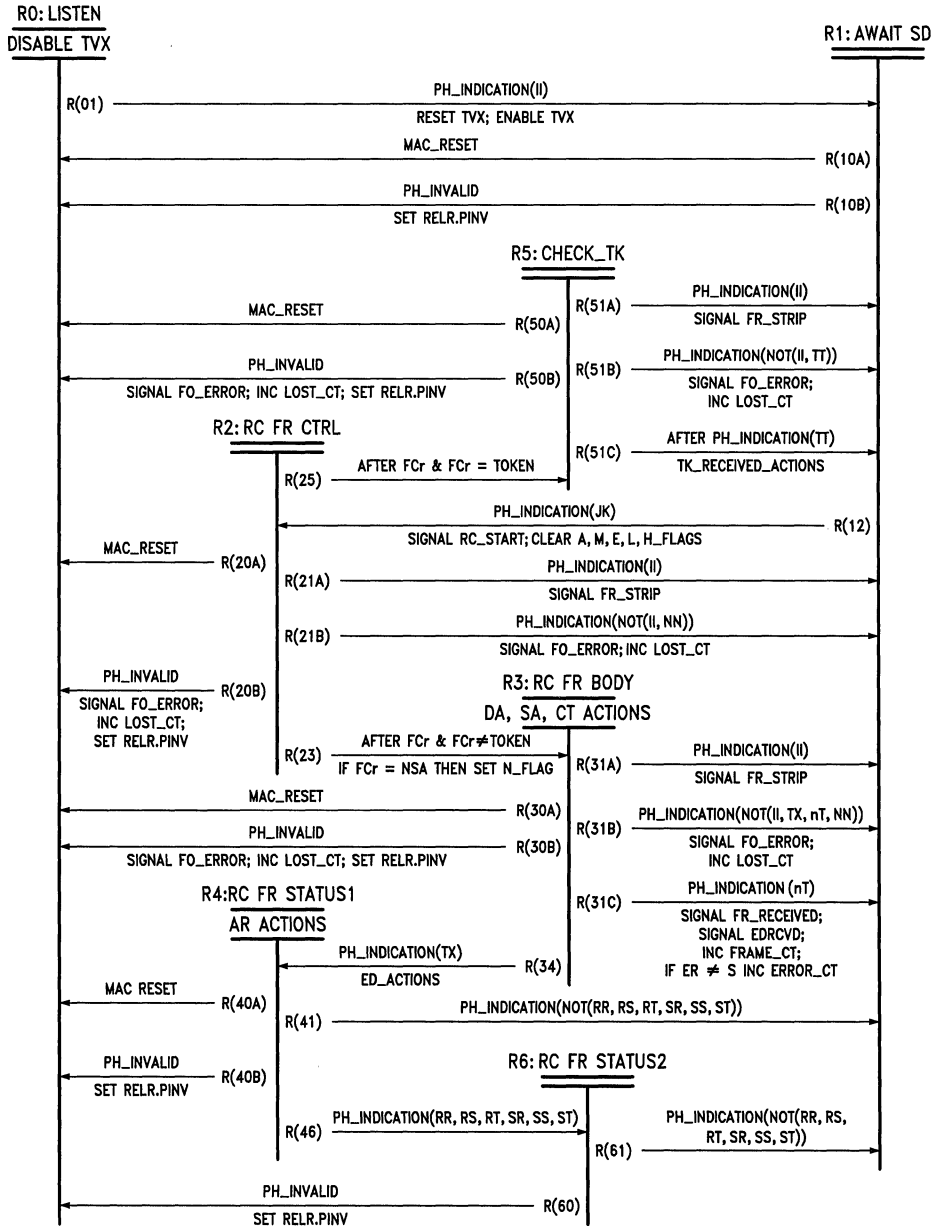


FIGURE A-1. Ring Engine Receiver State Diagram

TL/F/10387-43

Appendix A. Ring Engine State Machines (Continued)

A.1.2 MAC RECEIVER FOOTNOTES

A.1.2.1 Internal Conditions

- (1) **ESA:**
Option.Enable_Short_Address
- (2) **ELA:**
Option.Enable_Long_Address
- (3) **IRR:**
Option.Inhibit_Recovery_Required | (\neg ESA & \neg ELA)
- (4) **IFCS:**
Option.Implementer_FCS
- (5) **EMIND:**
Option.External_Matching_Indicators
- (6) **MAC_Reset:**
Function.MAC_Reset | \neg Mode.Run

A.1.2.2 Transition Conditions

- (1) **PH_Invalid:**
See encoding of PH_Invalid in Section 7.2.1.1
- (2) **PH_Indication (S1 S2):**
S1 is the first symbol received, S2 is the second symbol received. See encoding of PH_Indication in Section 7.2.1.1
- (3) **Transition R(12):**
This Transition may be a 0 time transition from any state except R0:Listen

A.1.2.3 Actions

1. DA_Actions:

```

IF FC.L = 0 CLEAR FCSL ;short address
ELSE SET FCSL ;long address

After DAOr
IF DA.IG = 0 CLEAR DAIG ;individual address
ELSE SET DAIG ;group address

IF FCSL = 0 ;short address
After DALr
    SIGNAL DARCVD
    IF DAIG = 1
        THEN IF DAR is contained in set of Group Addresses
            THEN SET A_Flag
    IF DAIG = 0
        THEN IF DAR = MSA
            THEN SET A_Flag

IF FCSL = 1 ;long address
After DA5r
    SIGNAL DARCVD
    IF DAIG = 1
        THEN IF DAR is contained in set of Group Addresses
            THEN SET A_Flag
    IF DAIG = 0
        THEN IF DAR = MLA
            THEN SET A_Flag

```

NOTE: A_Flag may be set on reception of VOID frames.

Appendix A. Ring Engine State Machines (Continued)

2. SA_Actions:

```

IF FCSL = 0; short address
  After SALr
    SIGNAL SARCVd
    IF ESA
      THEN IF SAR = MSA
        THEN SET MFLAG, Signal FR_Strip
        ELSE IF SAR > MSA THEN
          SET HFLAG
          ELSE SET LFLAG
      IF ((SAR = previous SAR) & (previous FCSL = 0) &
        (FC.FF = -MAC & previous FC.FF = -MAC) THEN
        SIGNAL SAMEESA

```

```

IF FCSL = 1; long address
  After SA5r
    SIGNAL SARCVd
    IF ELA
      THEN IF SAR = MLA
        THEN SET MFlag, Signal FR_Strip
        ELSE IF SAR > MLA
          THEN SET H_Flag
          ELSE SET L_Flag
      IF ((SAR = previous SAR) & (previous FCSL = 1) &
        (FC.FF = -MAC & previous FC.FF = -MAC)
        THEN SIGNAL SAMEESA

```

NOTE: A station with a null address may not win Claim when Option.IRR is set..

3. CT_Actions:

```

After 4_Info_Octets
  If FCr = Claim
    IF T_Bid_Rc ≠ TREQ
      CLEAR MFLAG
    IF T_Bid_Rc > TREQ
      THEN IF L_Flag
        SET H_Flag
        CLEAR L_Flag
      ELSE IF H_Flag
        SET L_Flag
        CLEAR H_Flag
    IF L_Flag
      SIGNAL FR_Strip
  IF ((INFOr = previous INFO) & (FCSL = previous FCSL) &
    (FC.FF = MAC & previous FC.FF = MAC)
    THEN SIGNAL SAMEINFO

```

4. TK_Received_Actions:

```

IF Token_Class = Restricted
  THEN IF -R_Flag
    THEN SET R_Flag
    SET TELR.ENTRMD {Entered_Restricted_Mode}
    ELSE RESET TVX
    CLEAR R_Flag
  SIGNAL TK_Received
  INC TKCT {token count}
  SET CILR.TKRCVD {Token_Received}

```

Appendix A. Ring Engine State Machines (Continued)

5. ED_Actions:

```

INC FRCT {Frame_Received_Ct}
SIGNAL FR_Received, EDRDVD
SET CILR.FRRVCV
    If Valid_Data_Length & (Valid_FCS_Rc | (FCr = Void) |
    (FCr = Implementer and ¬(Option.IFCS))
    THEN RESET TVX;
    IF (A_Flag | (EA & Option.EMIND)) & VCOPI
    THEN SET C_Flag
    ELSE SET E_Flag {This E_Flag is used during rest of the ED_Actions}
    CLEAR A_Flag, M_Flag, H_Flag, L_Flag

IF Er ≠ S & E_Flag
    THEN INC EICT {Error_Ct}
    SET CILR.FREI {Frame_Error_Isolated}

IF Er = R & ¬E_Flag
    THEN
        IF FCr = Claim
            THEN SET RELR.CLM
            IF A_Flag & M_Flag
                THEN SIGNAL My_Claim
                SET RELR.MYCLM
                CLEAR R_Flag
                SET TNEG = T_Bid_Rc

            IF H_Flag
                THEN SIGNAL Higher_Claim
                SET RELR.HICLM
                CLEAR R_Flag
                SET TNEG = T_Bid_Rc

            IF L_Flag
                THEN SIGNAL Lower_Claim
                SET RELR.LOCLM
                CLEAR R_Flag

        IF FCr = Beacon
            THEN SET RELR.BCN
            IF M_Flag
                THEN SIGNAL My_Beacon
                SET RELR.MYBCN
                CLEAR R_Flag
            IF ¬(M_Flag | E_Flag)
                THEN SIGNAL Other_Beacon
                SET RELR.OTRBCN
                CLEAR R_Flag

        IF FCr = Other_MAC
            THEN SIGNAL Other_MAC
            SET RELR.OTRMAC

        IF FCr = VOID
            THEN
                IF M_Flag & A_Flag & ¬DAIG
                    SIGNAL My_Void
                ELSE IF ¬A_Flag
                    SIGNAL Void
                ELSE IF ¬M_Flag
                    SIGNAL Other_Void

```

Appendix A. Ring Engine State Machines (Continued)**6. Ar_Actions:**

After Ar

```
IF Ar = R
    THEN CLEAR N_Flag
IF A_Flag & Ar = S & DA.IG = 0 & ¬E_Flag
    THEN SET RELR.DUPADD {Duplicate_Address | Strip Error detected}
IF REV1 & ¬E_Flag & (A_Flag | (EA & EMIND)) & FCr ≠ (MAC | Void)
    IF (VCOPY & FCr ≠ NSA) | (VCOPY & FCr = NSA & Ar = R)
        THEN SET CILR.FRCOP
            INC FCCT {Frame_Copied_Ct}
        ELSE IF ¬VCOPY | (FCr = NSA & Ar = S)
            SET CILR.FRNCOP
            INC FNCT {Frame_Not_Copied_Ct}
IF REV2 & ¬E_Flag & (A_Flag | (EA & EMIND)) & FCr ≠ (MAC | Void) & ¬(FCr =
NSA & Ar = S)
    IF VCOPY
        THEN SET CILR.FRCOP
            INC FCCT {Frame_Copied_Ct}
        ELSE SET CILR.FRNCOP
            INC FNCT {Frame_Not_Copied_Ct}
```

Appendix A. Ring Engine State Machines (Continued)

A.2 TRANSMITTER

A.2.1 MAC Transmitter State Diagram

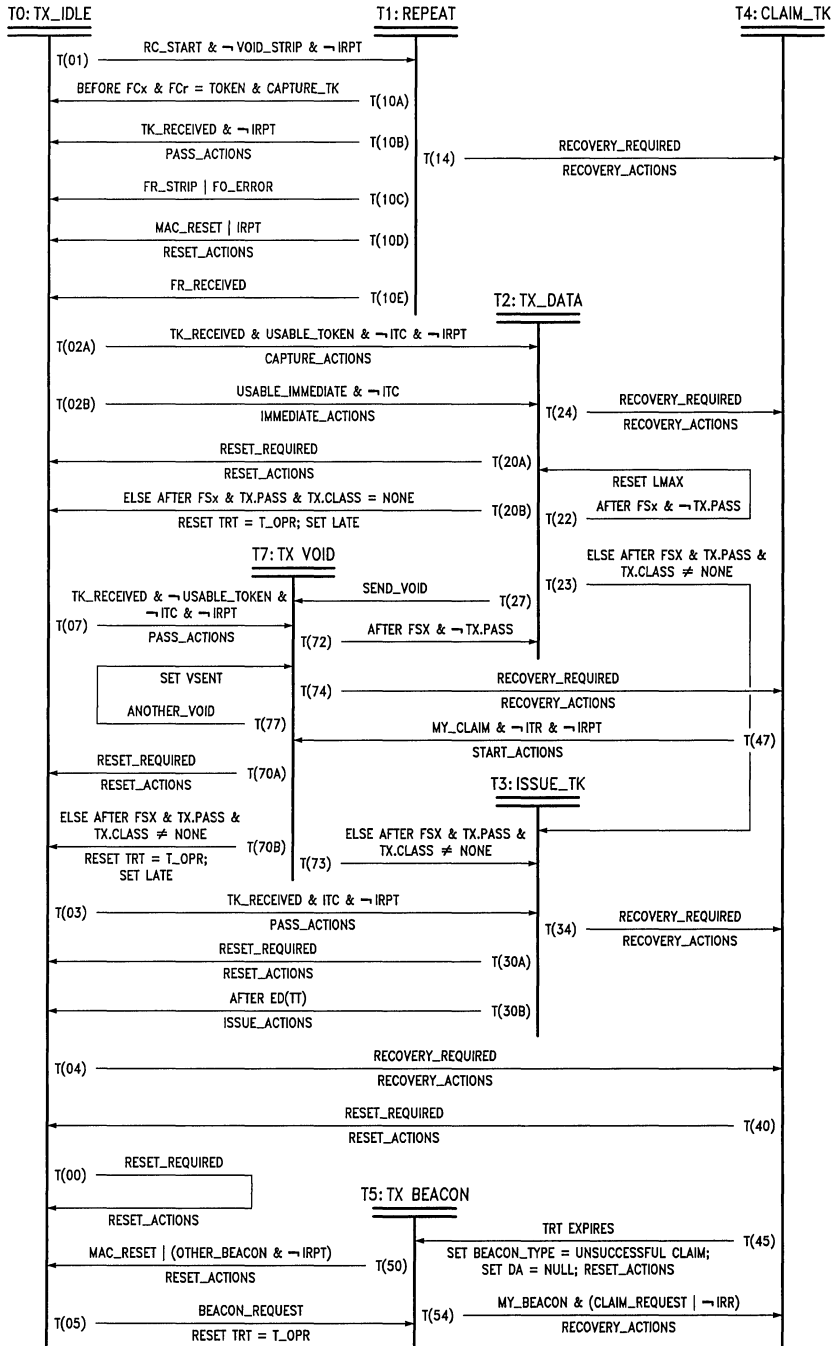


FIGURE A-2. Ring Engine Transmitter State Diagram

TL/F/10387-44

Appendix A. Ring Engine State Machines (Continued)

A.2.2 MAC TRANSMITTER FOOTNOTES

A.2.2.1 Internal Conditions

- (1) **ESA:**
Option.Enable_Short_Address
- (2) **ELA:**
Option.Enable_Long_Address
- (3) **IRPT:**
Option.Inhibit_Repeat | (¬ESA & ¬ELA)
- (4) **ITC:**
Option.Inhibit_Token_Capture
- (5) **IRR:**
Option.Inhibit_Recovery_Required | (¬ESA & ¬ELA)
- (6) **ITR:**
Option.Inhibit_Token_Release
- (7) **IFCS:**
Option.Implementer_FCS
- (8) **EMIND:**
Option.External_Matching_Indicators
- (9) **MAC_Reset:**
Function.MAC_Reset | ¬Mode.Run
- (10) **Beacon_Request:**
Function.Beacon_Request & ¬MAC_Reset
- (11) **Claim_Request:**
Function.Claim_Request & ¬Beacon_Request & ¬MAC_Reset

A.2.2.2 Transition Conditions

- (1) **Usable_Token:**
Ring_Operational & ¬RQ.Send &
((RQ.Class = synchronous & ¬RELR.Beacon_Received) |
(RQ.Class = asynchronous & ¬Late & RQ.Class.Capture = FCr.L &
(RQ.Class ≠ priority | TRT < T_Pri[RQ.Class.Priority]) &
¬(RQ.Class = restricted &
(RELR.Beacon_Received | RELR.Claim_Received |
(RQ.Class.Capture = nonrestricted & ¬RbeginOK))))
- (2) **Capture_TK:**
¬ITC & ¬IRPT & (Usable_Token |
(Ring_Operational & ¬TELR.Ring_Latency_Valid & ¬Late & ¬FCr.L))
- (3) **Immediate_Request:**
RQ.Class = immediate & ¬RQ.Claim & ¬RQ.Beacon
- (4) **Usable_Immediate:**
¬Ring_Operational & TX.Class = none &
¬(TK_Received & ¬IRPT) & ¬RQ.Send & Immediate_Request
- (5) **Send_Void:**
TX.Abort |
(After FSx &
((TX.Ready &
(¬RQ.Ready | ((Immediate_Request | Lmax expired) & ¬RQ.Send)) |
TX.Pass &
(Void_Strip | (¬TELR.Ring_Latency_Valid & Early) |
¬(TX.ED | TX.Class = none))

Appendix A. Ring Engine State Machines (Continued)

(6) Another_Void:

After FSx & TX.Pass & Void_Strip & ~Vsent

(7) Reset_Required:

```
MAC_Reset |
(-IRPT & (Higher_Claim | Other_Beacon | Other_MAC)) |
(IRPT & (T3 | (TO & (Ring_Operational | TX.Class ≠ none | ~Late))))
```

Note: Any other MAC frame received while RING_Operational must be a My_Claim or a bad frame. These frames are ignored here.

(8) Recovery_Required:

```
Claim_Request |
(~IRR &
(Lower_Claim | My_Beacon | TVX expires |
(TRT expires & Late & ~(TO | T1) & TK_Received)))
```

Note: (Ring_Operational & T_Opr < T_Req) must be detected by software!

A.2.2.3. Transition Actions

(1) Pass_Actions:

```
CLEAR TX.Ready, Void_Strip;
IF T1 THEN SET RbeginOK = ~FCr.L;
SET TX.Class = FCr.L; SET TX.Pass, TELR.Token_Passed;
If Ring_Operational
  THEN IF ~Late
    THEN RESET TRT = T_Opr
    ELSE CLEAR Late
  ELSE SET T_Opr = T_Neg; RESET TRT = T_Opr; SET Late;
  SET RELR.Ring_Operational_Set, Ring_Operational
```

(2) Capture_Actions:

```
CLEAR TX.ED, TX.Abort, TX.Pass, Void_Strip;
SET TX.Class = FCr.L; SET TX.Ready, TELR.Token_Captured;
RESET Lmax;
IF ~Late
  THEN SET Early; SET THT = TRT; RESET TRT = T_Opr
  ELSE CLEAR Early, Late
```

(3) Immediate_Actions:

```
CLEAR TX.ED, TX.Abort, TX.Pass, Void_Strip;
SET TX.Class = none; SET TX.Ready;
SET Early; RESET TRT = T_Opr; CLEAR Late
```

(4) Reset_Actions:

```
IF T4|T5|(T2 & ~TX.Ready & ~TX.Pass & ~TX.ED)
  THEN SET TX.Abort
CLEAR TX.Ready, TX.Ack, Void_Strip;
SET TX.Class = none; SET TX.Pass;
SET T_Opr = T_Max; RESET TRT = T_Opr; SET Late;
IF Ring_Operational
  THEN SET RELR.Ring_Operational_Reset
IF MAC_Reset|Ring_Operational
  CLEAR Late_Count, Ring_Operational, Function.MAC_Reset
```

Appendix A. Ring Engine State Machines (Continued)

(5) Recovery_Actions:

```

IF T2 & ¬TX.Ready & ¬TX.Pass & ¬TX.ED
    THEN SET TX.Abort

IF T5
    THEN CLEAR TX.Abort

CLEAR TX.Ready, TX.Ack, Void_Strip;
SET TX.Class = nonrestricted; SET TX.Pass;
SET T_Opr = T_Max; RESET TRT = T_Opr; SET Late;
IF Ring_Operational
    THEN SET RELR.Ring_Operational_Reset;
    CLEAR Late_Count, Ring_Operational

```

(6) Start_Actions:

```

CLEAR TX.Ready, TX.Ack, TX.Abort; SET Void_Strip, TX.Pass;
RESET TRT = T_Opr

```

(7) Issue_Actions:

```

IF ¬Ring_Operational
    THEN SET T_Opr = T_Neg; RESET TRT = T_Opr; SET Late
IF TX.Class = nonrestricted & ¬R_Flag
    THEN SET RbeginOK
    ELSE CLEAR RbeginOK

```

A.2.2.4 State Actions

(1) TRT_Actions:

```

Always:
    IF TRT expires
        THEN RESET TRT = T_Opr;
        IF ¬((T0|T1) & TK_Received & ¬IRPT)
            THEN IF ¬Late
                THEN SET Late
                ELSE SET TELR.TRT_Expired;
                IF ¬Ring_Operational
                    THEN INCREMENT Late_Count
        IF (T4 & ¬(My_Claim & ¬ITR & ¬IRPT))|
        (T5 & ¬(My_Beacon & (Claim_Request|¬IRR)))
            THEN SET TELR.Recovery_Failed

```

(2) RLCT_Actions:

```

Always:
    IF TELR.Ring_Latency_Valid|MAC_Reset|(¬ESA & ¬ELA)|
    (TRT expires & Late)|PH_Invalid|
    Lower_Claim|My_Beacon|Higher_Claim|
    Other_Beacon|Other_MAC|TK_Received|Other_Void
        THEN DISABLE Latency_Count
    IF My_Void & Latency_Count enabled
        THEN SET TELR.Ring_Latency_Valid

```

(3) TX_Idle_Actions (T0):

```

Always:
    PH_Data.request(II);
    IF My_Void|Other_Void
    THEN CLEAR Void_Strip

```

Appendix A. Ring Engine State Machines (Continued)

(4) Repeat_Actions (T1):

```

IF ¬IRPT & (Higher_Claim|Other_Beacon|Other_MAC) &
  (Ring_Operational|TX.Class ≠ none|¬Late)
  THEN SET TX.Class = none;
  SET T_Opr = T_Max; RESET TRT = T_Opr; SET Late;
  IF Ring_Operational
    THEN SET RELR.Ring_Operational_Reset;
    CLEAR Late_Count, Ring_Operational

Still_Usable:
  Ring_Operational & ¬RQ.Send &
  ((RQ.Class = synchronous & ¬RELR.Beacon_Received|
  (RQ.Class = asynchronous & ¬Late & RQ.Capture = FCr.L &
  (RQ.Class ≠ priority|TRT < T_Pri[RQ.Class.Priority]) &
  ¬(RQ.Class = restricted &
  (RELR.Beacon_Received|RELR.Claim_Received|
  (RQ.Class.Capture = nonrestricted & ¬RbeginOK))))))

```

(5) TX_Data_Actions (T2):

```

IF Lmax = expired & RQ.Ready
  THEN RESET Lmax; SET Used

IF ¬RQ.Ready
  THEN RESET Lmax; CLEAR Used

IF Abort
  THEN SET TX.Abort;
  IF Still_Usable
    THEN SET TX.Ready
    ELSE SET TX.Pass

After ED
  SET TX.ED;
  IF Still_Usable
    THEN SET TX.Ready
    ELSE SET TX.Pass

```

(6) Issue_TK_Actions (T3):

```

Always:
  IF My_Void
    THEN CLEAR Void_Strip

```

(7) Claim_TK_Actions (T4):

```

CLEAR Function.Claim_Request

```

(8) TX_Beacon_Actions (T5):

```

CLEAR Function.Beacon_Request

```

(9) TX_Void_Actions (T7):

```

Always:
  IF My_Void & Vsent
    THEN CLEAR Void_Strip

```


DP83265 BSI™ Device (FDDI System Interface)

General Description

The DP83265 BSI device implements an interface between the National FDDI BMAC™ device and a host system. It provides a multi-frame, MAC-level interface to one or more MAC Users.

The BSI device accepts MAC User requests to receive and transmit multiple frames (Service Data Units). On reception (Indicate), it receives the byte stream from the BMAC device, packs it into 32-bit words and writes it to memory. On transmission (Request), it unpacks the 32-bit wide memory data and sends it a byte at a time to the BMAC device. The host software and the BSI device communicate via registers, descriptors, and an attention/notify scheme using clustered interrupts.

Features

- 32-bit wide Address/Data path with byte parity
- Programmable transfer burst sizes of 4 or 8 32-bit words
- Interfaces to low-cost DRAMs or directly to system bus
- 2 Output and 3 Input Channels
- Supports Header/Info splitting
- Bridging support
- Efficient data structures
- Programmable Big or Little Endian alignment
- Full Duplex data path allows transmission to self
- Confirmation status batching services
- Receive frame filtering services
- Operates from 12.5 MHz to 25 MHz synchronously with host system

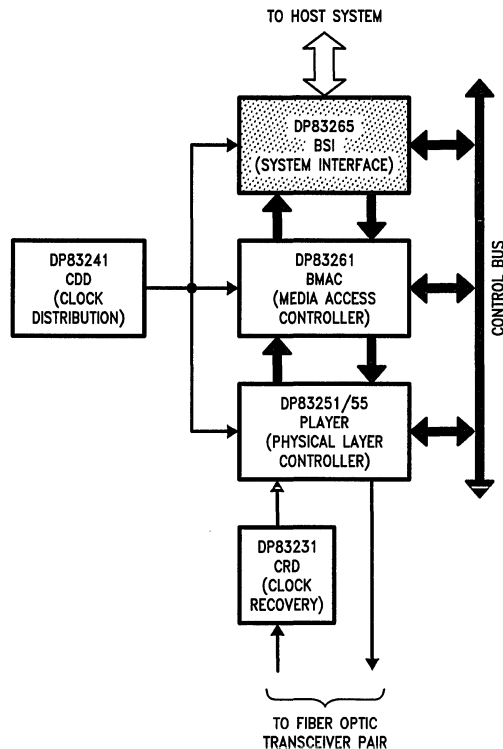


FIGURE 1. FDDI Chip Set Block Diagram

TL/F/10791-1

Table of Contents

1.0 FDDI CHIP SET OVERVIEW

2.0 ARCHITECTURE DESCRIPTION

- 2.1 Interfaces
- 2.2 Data Structures
- 2.3 Map Engine

3.0 FEATURE OVERVIEW

- 3.1 32-Bit address/Data Path to Host Memory
- 3.2 Multi-Channel Architecture
- 3.3 Support for Header/Info Splitting
- 3.4 MAC Bridging Support
- 3.5 Confirmation Status Batching Services
- 3.6 Receive Frame Filtering Services
- 3.7 Two Timing Domains
- 3.8 Clustered Interrupts

4.0 FUNCTIONAL DESCRIPTION

- 4.1 Overview
- 4.2 Operation
- 4.3 Bus Interface Unit

5.0 CONTROL INFORMATION

- 5.1 Overview
- 5.2 Operation Registers
- 5.3 Control and Event Register Descriptions
- 5.4 Pointer RAM Registers
- 5.5 Limit RAM Registers
- 5.6 Descriptors
- 5.7 Operating Rules
- 5.8 Pointer RAM Register Descriptions
- 5.9 Limit RAM Register Descriptions
- 5.10 BSI Device Descriptors

6.0 SIGNAL DESCRIPTIONS

- 6.1 Pin Organization
- 6.2 Control Interface
- 6.3 BMAC Device Indicate Interface
- 6.4 BMAC Device Request Interface
- 6.5 ABus Interface
- 6.6 Electrical Interface

1.0 FDDI Chip Set Overview

National Semiconductor's FDDI chip set consists of five components as shown in *Figure 1-1*. For more information about the other devices in the chip set, consult the appropriate data sheets and application notes.

DP83231 CRD™ Device Clock Recovery Device

The Clock Recovery Device extracts a 125 MHz clock from the incoming bit stream.

Features

- PHY Layer loopback test
- Crystal controlled
- Clock locks in less than 85 μ s

DP83241 CDD™ Device Clock Distribution Device

From a 12.5 MHz reference, the Clock Distribution Device synthesizes the 125 MHz, 25 MHz, and 12.5 MHz clock required by the BSI, BMAC, and PLAYER devices.

DP83251/55 PLAYERT™ Device Physical Layer Controller

The PLAYER device implements the Physical Layer (PHY) protocol as defined by the ANSI FDDI PHY X3T9.5 Standard.

Features

- 4B/5B encoders and decoders
- Framing logic
- Elasticity Buffer, Repeat Filter and Smoother
- Line state detector/generator
- Link error detector
- Configuration switch
- Full duplex operation
- Separate management port that is used to configure and control operation

In addition, the DP83255 contains an additional PHY__Data.request and PHY__Data.indicate port required for concentrators and dual attach stations.

DP83261 BMACTM Device Media Access Controller

The BMAC device implements the Timed Token Media Access Control protocol defined by the ANSI FDDI X3T9.5 MAC Standard.

Features

- All of the standard defined ring service options
- Full duplex operation with through parity
- Supports all FDDI Ring Scheduling Classes (Synchronous, Asynchronous, etc.)
- Supports Individual, Group, Short, Long and External Addressing
- Generates Beacon, Claim, and Void frames internally
- Extensive ring and station statistics gathering
- Extensions for MAC level bridging
- Separate management port that is used to configure and control operation
- Multi-frame streaming interface

DP83265 BSITM Device System Interface

The BSI Device implements an interface between the BMAC device and a host system.

Features

- 32-bit wide Address/Data path with byte parity
- Programmable transfer burst sizes of 4 or 8 32-bit words
- Interfaces to low-cost DRAMs or directly to system bus
- Provides 2 Output and 3 Input Channels
- Supports Header/Info splitting
- Efficient data structures
- Programmable Big or Little Endian alignment
- Full duplex data path allows transmission to self
- Confirmation status batching services
- Receive frame filtering services
- Operates from 12.5 MHz to 25 MHz synchronously with host system

2.0 Architecture Description

The BSI device is composed of three interfaces and the Map Engine.

The three interfaces are the BMAC device, the ABus, and the Control Bus. They are used to connect the BSI device to the BMAC device, Host System, and external Control Bus.

The Map Engine manages the operation of the BSI device.

2.1 INTERFACES

The BSI device connects to external components via three interfaces: the BMAC device Interface, the ABus Interface, and the Control Bus Interface (see Figure 2-1).

2.1.1 BMAC Device Interface

The BSI device connects to the BMAC device via the MA_Indicate (receive) and MA_Request (transmit) Interfaces, as shown in Figure 2-1.

Received Data is transferred from the BMAC device to the BSI device via the MA_Indicate Interface. The MA_Indicate Interface consists of a parity bit (odd parity) and byte-wide data along with flag and control signals.

Transmit Data is transferred from the BSI device to the BMAC device via the MA_Request Interface. The MA_Request Interface consists of a parity bit (odd parity) and byte-wide data along with flag and control signals.

2.1.2 ABus Interface

The BSI device connects to the Host System via the ABus Interface. The ABus Interface consists of four bits of parity (odd parity) and 32 bits of multiplexed address and data along with transfer control and bus arbitration signals.

2.1.3 Control Bus Interface

The Control Bus Interface connects the BSI device to the external Control Bus.

The Control Bus Interface is separate from the BMAC device and ABus Interfaces to allow independent operation of the Control Bus.

The host uses the Control Bus to access the BSI device's internal registers, and to manage the attention/notify logic.

2.2 DATA STRUCTURES

2.2.1 Data Types

The architecture of the BSI device defines two basic kinds of objects: Data Units and Descriptors. A Data Unit is a group of contiguous bytes which forms all or part of a frame (Service Data Unit). A Descriptor is a two-word (64-bit) control object that provides addressing information and control/status information about BSI device operations.

Data and Descriptor objects may consist of one or more parts, where each part is contiguous and wholly contained within a 1k or 4k memory page. A single-part object consists of one **Only** Part; a multiple-part object consists of one **First** Part, zero or more **Middle** Parts, and one **Last** Part. In Descriptor names, the object part is denoted in a suffix, preceded by a dot. Thus an Input Data Unit Descriptor (IDUD), which describes the last Data Unit of a frame received from the ring, is called an IDUD.Last.

A single-part Data Unit is stored in contiguous locations within a single 4k byte page in memory. Multiple-part Data Units are stored in separate, and not necessarily contiguous 4k byte pages. Descriptors are stored in contiguous locations in Queues and Lists, where each Queue or List occupies a single 1k byte or 4k byte memory page, aligned on the queue-size boundary. For both Queues and Lists, an access to the next location after the end of a page will automatically wrap-around and access the first location in the page.

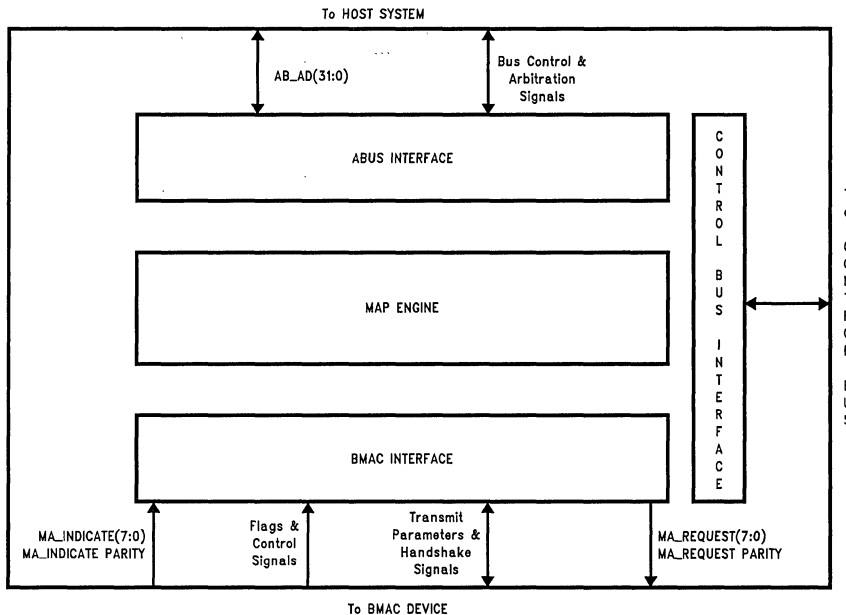


FIGURE 2-1. BSI Device Interfaces

TL/F/10791-2

2.0 Architecture Description (Continued)

Data Units (MAC Service Data Units) are transferred between the BSI device and BMAC device via five simplex Channels, three used for Indicate (receive) data and two for Request (transmit) data. Parts of frames received from the ring and copied to memory are called Input Data Units (IDUs); parts of frames read from memory to be transmitted to the ring are called Output Data Units (ODUs).

Descriptors are transferred between the BSI device and Host via the ABUS, whose operation is for the most part transparent to the user. There are five Descriptor types recognized by the BSI device: Input Data Unit Descriptors (IDUDs), Output Data Unit Descriptors (ODUDs), Pool Space Descriptors (PSPs), Request Descriptors (REQs), and Confirmation Message Descriptors (CNFs).

Input and Output Data Unit Descriptors describe a single Data Unit part, i.e., its address (page number and offset), its size in bytes, and its part (Only, First, Middle, or Last). Frames consisting of a single part are described by a Descriptor.Only; frames consisting of multiple parts are described by a Descriptor.First, zero or more Descriptor.Middles, and a Descriptor.Last.

Every Output Data Unit part is described by an Output Data Unit Descriptor (ODUD). Output Data Unit Descriptors are fetched from memory so that frame parts can be assembled for transmission.

Every Input Data Unit part is described by an Input Data Unit Descriptor (IDUD). Input Data Unit Descriptors are generated on Indicate Channels to describe where the BSI device wrote each frame part and to report status for the frame.

Request Descriptors (REQs) are written by the user to specify the operational parameters for BSI device Request operations. Request Descriptors also contain the start address of part of a stream of ODUDs and the number of frames represented by the ODUD stream part (i.e., the number of ODUD.Last descriptors). Typically, the user will define a single Request Object consisting of multiple frames of the same request and service class, frame control, and expected status.

Confirmation Messages (CNFs) are created by the BSI device to record the result of a Request operation.

Pool Space Descriptors (PSPs) describe the location and size of a region of memory space available for writing Input Data Units.

Request (transmit) and Indicate (receive) data structures are summarized in *Figures 2-2 and 2-3*.

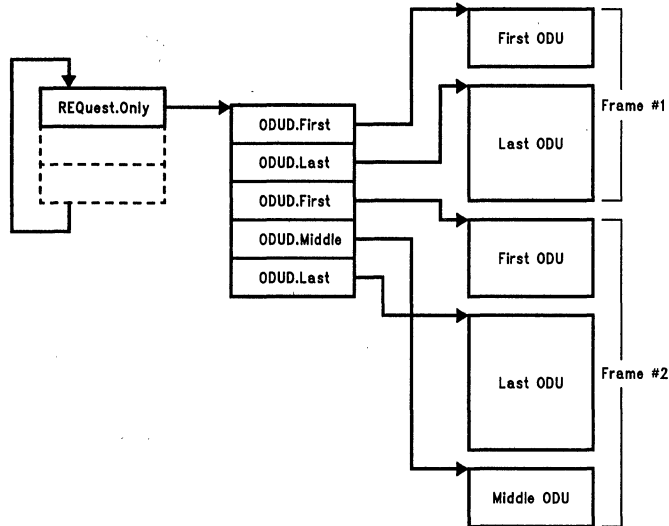
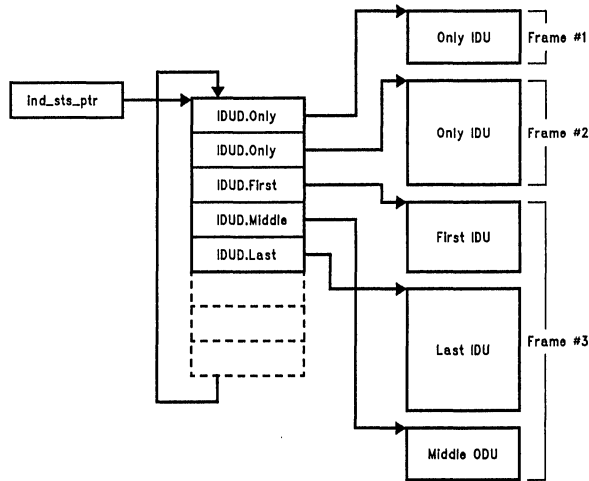


FIGURE 2-2. BSI Device Request Data Structures

TL/F/10791-3

2.0 Architecture Description (Continued)



TL/F/10791-4

FIGURE 2-3. BSI Device Indicate Data Structures

2.2.2 Descriptor Queues and Lists

The BSI device utilizes 10 Queues and two Lists. These queues and lists are circular. There are six Queues for Indicate operations, and four Queues and two Lists for Request operations. Each of the three Indicate Channels has a Data Queue containing Pool Space Descriptors (PSPs), and a Status Queue containing Input Data Unit Descriptors (IDUDs). Each Request Channel has a Data Queue containing Request Descriptors (REQs), a Status Queue containing Confirmation Messages (CNFs), and a List containing Output Data Unit Descriptors (ODUDs).

During Indicate and Request operations, Descriptor Queues and Lists are read and written by the BSI device, using registers in the Pointer and Limit RAM Register files. The Pointer RAM Queue and List Pointer Registers point to a location from which a Descriptor will be read (PSPs, REQs, and ODUDs) or written (IDUDs and CNFs).

For each Queue Pointer Register there is a corresponding Queue Limit Register in the Limit RAM Register file, which holds the Queue's limit as an offset value in units of 1 Descriptor (8 bytes). The address in the Queue Pointer is incremented before a Descriptor is read and after a Descriptor is written, then compared with the value in the corresponding Queue Limit Register. When a Descriptor is accessed from the location defined by the Queue Limit Register, an attention is generated, informing the host that the Queue is empty. When a pointer value is incremented past the end of the page, it wraps to the beginning of the page.

2.2.3 Storage Allocation

The maximum unit of contiguous storage allocation in external memory is a Page. All BSI device addresses consist of a 16-bit page number and a 12-bit offset.

The BSI device uses a page size of 1K or 4k bytes for storage of Descriptor Queues and Lists (as selected by the user), and a page size of 4K bytes for storage of Data Units. A single page may contain multiple Data Units, and multiple-part Data Units may span multiple disjoint or contiguous pages.

2.3 MAP ENGINE

The Map Engine, which manages the operation of the BSI, is comprised of seven basic blocks: Indicate Machine, Request Machine, Status/Space State Machine, Operation RAM, Pointer RAM, Limit RAM, and Bus Interface Unit. An internal block diagram of the BSI device is shown in *Figure 2-4*.

2.3.1 Indicate Machine

The Indicate Block accepts Service Data Units (frames) from the BMAC device in the byte stream format (MA_Indicate).

Upon receiving the data, the Indicate Block performs the following functions:

- Decodes the Frame Control field to determine the frame type
- Sorts the received frames onto Channels according to the Sort Mode
- Filters identical MAC frames
- Copies the received frames to memory according to Copy Criteria
- Writes status for the received frames to the Indicate Status Queue
- Issues interrupts to the host at host-defined status breakpoints

2.3.2 Request Machine

The Request Machine presents Service Data Units (MAC frames) to the BMAC device in the byte stream format (MA_Request).

The Request Machine performs the following functions:

- Reads frames from host memory and assembles them onto Request Channels
- Prioritizes active requests
- Transmits frames to the BMAC device
- Writes status for transmitted and returning frames
- Issues interrupts to the host on user-defined group boundaries

2.0 Architecture Description (Continued)

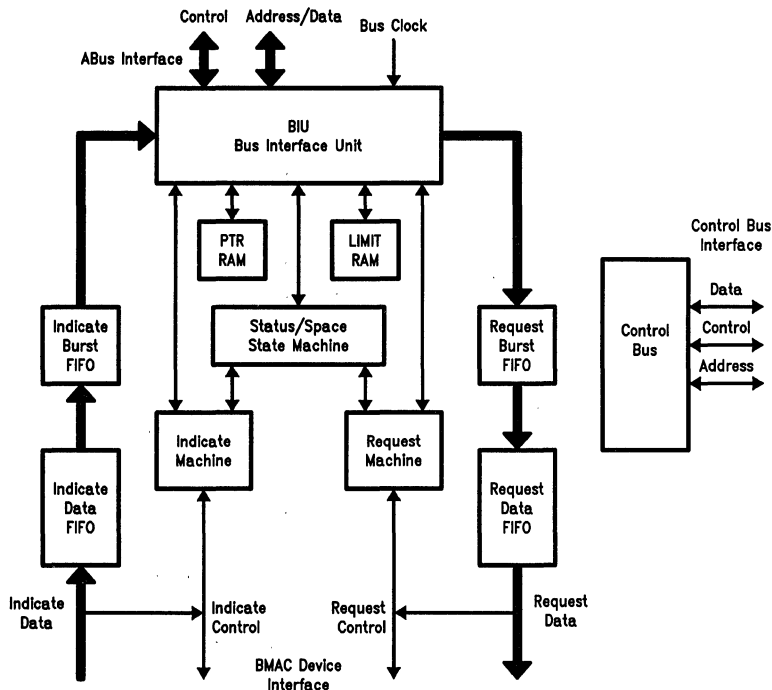


FIGURE 2-4. BSI Device Internal Block Diagram

TL/F/10791-5

2.3.3 Status/Space Machine

The Status/Space Machine is used by both the Indicate Machine and the Request Machine.

The Status/Space Machine manages all descriptor Queues and writes status for received and transmitted frames.

2.3.4 Bus Interface Unit

The Bus Interface Unit (BIU) is used by both the Indicate and Request Blocks. It manages the ABus Interface, providing the BSI device with a 32-bit data path to local or system memory.

The Bus Interface Unit controls the transfer of Data Units and Descriptors between the BSI device and Host memory via the ABus.

Data and Descriptors are transferred between the BSI device and Host memory in **streams**, where a stream is a flow of logically related information (i.e., a single type of data or descriptor object) in one direction (either to or from host memory). Each Channel supports a subset of object streams, via Subchannels. The three Indicate Channels each support three Subchannels:

1. Input Data Unit stream
2. Input Data Unit Descriptor stream
3. Pool Space Descriptor stream

The two Request Channels each support four Subchannels:

1. Output Data Unit stream
2. Output Data Unit Descriptor stream

3. Confirmation Message Descriptor stream

4. Request Descriptor stream

The BIU arbitrates between the Subchannels and issues a Bus Request when any Subchannel requests service. The priority of Subchannel bus requests is generally as follows, from highest priority to lowest priority:

1. Output Data Unit reads (highest priority)
2. Input Data Unit writes
3. Input Data Unit Descriptor writes
4. Confirmation Message Descriptor writes
5. Pool Space Descriptor reads
6. Mailbox reads/writes
7. Pointer RAM and Limit RAM Service functions (lowest priority)

Addresses for Subchannel accesses are contained in the Pointer RAM Registers.

2.3.5 Pointer RAM

The Pointer RAM Block is used by both the Indicate and Request Machines. It contains pointers to all Data Units and Descriptors manipulated by the BSI device, namely, Input and Output Data Units, Input and Output Data Unit Descriptors, Request Descriptors, Confirmation Messages, and Pool Space Descriptors.

2.0 Architecture Description

(Continued)

The Pointer RAM Block is accessed by clearing the PTOF (Pointer RAM Operation) bit in the Service Attention Register, which causes the transfer of data between the Pointer RAM Register and a mailbox location in memory.

2.3.6 Limit RAM

The Limit RAM Block is used by both the Indicate and Request Machines. It contains data values that define the limits of the ten Queues maintained by the BSI device.

Limit RAM Registers are accessed by clearing the LMOF (Limit RAM Operation) bit in the Service Attention Register, which causes the transfer of data between the Limit RAM Register and the Limit Data and Limit Address Registers.

3.0 Feature Overview

The BSI device implements a system interface for the FDDI BMAC Device. It is designed to provide a high-performance, low-cost interface for a variety of hosts.

On the system side, the BSI device provides a simple yet powerful bus interface and memory management scheme to maximize system efficiency. It is capable of interfacing to a variety of host busses/environments. The BSI device provides a 32-bit wide multiplexed address/data interface, which can be configured to share a system bus to main memory or communicate via external shared memory. The system interface supports virtual addressing using fixed-size pages.

On the network side, the BSI device performs many functions which greatly simplify the interface to the BMAC device, and provides many services which simplify network management and increase system performance and reliability. The BSI device is capable of batching confirmation and indication status, filtering out MAC frames with the same information field, and performing network monitoring functions.

3.1 32-BIT ADDRESS/DATA PATH TO HOST MEMORY

The BSI device provides a 32-bit wide synchronous multiplexed address/data interface, which permits interfacing to a standard multi-master system bus operating from 12.5 MHz to 25 MHz, or to local memory, using Big or Little Endian byte ordering. The memory may be static or dynamic. For maximum performance, the BSI device utilizes burst mode transfers, with four or eight 32-bit words to a burst. To assist the user with the burst transfer capability, the three bits of the address which cycle during a burst are output demultiplexed. Maximum burst speed is one 32-bit word per clock, but slower speeds may be accommodated by inserting wait states.

The BSI device can operate within any combination of cached/non-cached, paged or non-paged memory environments. To provide this capability, all data structures are contained within a page, and bus transactions never cross a page. The BSI device performs all bus transactions within aligned blocks to ease the interface to a cached environment.

3.2 MULTI-CHANNEL ARCHITECTURE

The BSI device provides three Input Channels and two Output Channels, which are designed to operate independently and concurrently. They are separately configured by the user to manage the reception or transmission of a particular kind of frame (for example, synchronous frames only).

3.3 SUPPORT FOR HEADER/INFO SPLITTING

In order to support high performance protocol processing, the BSI device can be programmed to split the header and information portions of (non-MAC/SMT) frames between two Indicate Channels. Frame bytes from the Frame Control field (FC) up to the user-defined header length are copied onto Indicate Channel 1, and the remaining bytes (info) are copied onto Indicate Channel 2.

3.4 MAC BRIDGING SUPPORT

Support for bridging and monitoring applications is provided by the Internal/External Sorting Mode. All frames matching the external address (frames requiring bridging) are sorted onto Indicate Channel 2, MAC and SMT frames matching the internal (BMAC device) address are sorted onto Indicate Channel 0, and all other frames matching the BMAC device's internal address (short or long) are sorted onto Indicate Channel 1.

3.5 CONFIRMATION STATUS BATCHING SERVICES

The BSI device provides confirmation status for transmitted and returning frames. Interrupts to the host are generated only at status breakpoints, which are defined by the user on a per Channel basis when the Channel is configured for operation.

The BSI device further reduces host processing time by separating received frame status from the received data. This allows the CPU to quickly scan for errors when deciding whether to copy the data to memory. If the status were embedded in the data stream, all the data would need to be read contiguously to find the Status Indicator.

3.6 RECEIVE FRAME FILTERING SERVICES

To increase performance and reliability, the BSI device can be programmed to filter out identical (same FC and Info field) MAC or SMT frames received from the ring. Filtering unnecessary frames reduces the fill rate of the Indicate FIFO, reduces CPU frame processing time, and avoids unnecessary memory bus transactions.

3.7 TWO TIMING DOMAINS

To provide maximum performance and system flexibility, the BSI device utilizes two independent clocks, one for the MAC (ring) Interface, and one for the system/memory bus. The BSI device provides a fully synchronized interface between these two timing domains.

3.8 CLUSTERED INTERRUPTS

The BSI device can be operated in a polled or interrupt-driven environment. The BSI device provides the ability to generate attentions (interrupts) at group boundaries. Some boundaries are pre-defined in hardware; others are defined by the user when the Channel is configured. This interrupt scheme significantly reduces the number of interrupts to the host, thus reducing host processing overhead.

4.0 Functional Description

The BSI device is composed of the Map Engine and Interfaces to the Control Bus (Control Bus Interface), the BMAC device (BMAC Device Interface) and the ABus (Abus Interface).

In this section, the Map Engine is described in detail to provide an in-depth look at the operation of the BSI device.

4.1 OVERVIEW

The Map Engine consists of two major blocks, the Indicate Machine and the Request Machine. These blocks share the Bus Interface Unit, Status/Space Machine, Pointer RAM, and Limit RAM blocks.

The Map Engine provides an interface between the BMAC FDDI Protocol chip and a host system. The Map Engine transfers FDDI frames (Service Data Units) between the FDDI device and host memory.

4.1.1 Indicate Machine

On the Receive side (from the ring) the Indicate Machine sequences through the incoming byte stream from the BMAC device. Received frames are sorted onto Indicate Channels and a decision is made whether or not to copy them to host memory. The Indicate Machine uses the control signals provided by the BMAC device Receive State Machine on the MAC Indicate Interface.

4.1.2 Request Machine

On the Transmit side (to the ring) the Request Machine prepares one or more frames from host memory for transmission to the BMAC device. The Request Machine provides all the control signals to drive the BMAC device Request Interface.

4.2 OPERATION

4.2.1 Indicate Operation

The Indicate Block accepts data from the BMAC device as a byte stream.

Upon receiving the data, the Indicate Block performs the following functions:

- Decodes the Frame Control field to determine the frame type
- Sorts the received frames onto Channels according to the Sort Mode
- Filters identical MAC frames
- Copies the received frames to memory according to Copy Criteria
- Writes status for the received frames to the Indicate Status Queue
- Issues interrupts to the host on host-defined status breakpoints

The Indicate Machine decodes the Frame Control (FC) field to determine the type of frame. Ten types of frames are recognized: Logical Link Control (LLC), Restricted Token, Unrestricted Token, Reserved, Station Management (SMT), SMT Next Station Addressing, MAC Beacon, MAC Claim, Other MAC, and Implementer.

The Indicate Machine sorts incoming frames onto Indicate Channels according to the frame's FC field, the state of the AFLAG signal from the BMAC device, and the host-defined sorting mode programmed in the Sort Mode field of the Indicate Mode Register. SMT and MAC Service Data Units (SDUs) are always sorted onto Indicate Channel 0. On Indicate Channels 1 and 2, frames can be sorted according to

whether they are synchronous or asynchronous, high-priority asynchronous or low-priority asynchronous, whether their address matches an internal (BMAC device) or external address, or the header and Information fields of all non-MAC/SMT frames.

The Synchronous/Asynchronous Sort Mode is intended for use in end-stations or applications using synchronous transmission.

With High-priority/Low-priority sorting, high-priority asynchronous frames are sorted onto Indicate Channel 1 and low-priority asynchronous frames are sorted onto Indicate Channel 2. The most-significant bit of the three-bit priority field within the FC field determines the priority. This Mode is intended for end stations using two priority levels of asynchronous transmission.

With External/Internal sorting, frames matching the internal address (in the BMAC device) are sorted onto Indicate Channel 1 and frames matching an external address (when the EA input is asserted) are sorted onto Indicate Channel 2. This mode is intended for bridges or ring monitors, which would utilize the ECIP/EA/EM pins with external address matching circuitry.

The proper use of the ECIP, EA, and EM pins is as follows. External address matching circuitry must assert ECIP somewhere from the assertion of FCRCVD (from the BMAC device) up to the clock cycle before the assertion of INFORCVD (from the BMAC device). Otherwise, the BSI device assumes that no external address comparison is taking place. ECIP must be negated for at least one cycle to complete the external comparison. If it has not been deasserted before EDRCVD (from the BMAC device) the frame is not copied. EA and EM are sampled on the clock cycle after ECIP is negated. ECIP is ignored after it is negated until FCRCVD is asserted again.

Note that this design allows ECIP to be a positive or negative pulse. To confirm frames in this mode, (typically with Source Address Transparency enabled), EM must be asserted within the same timeframe as EA.

With the Header/Info Sort Mode, Indicate Channels 1 and 2 receive all non-MAC/SMT frames that are to be copied, but between them split the frame header (whose length is user-defined) and the remaining portions of the frame (Info). Indicate Channel 1 copies the initial bytes up until the host-defined header length is reached. The remainder of the frame's bytes are copied onto Indicate Channel 2. Only one IDUD stream is produced (on Indicate Channel 1), but both PSP Queues are used to determine where the IDUs will be written. When a multi-part IDUD is produced, the Indicate Status field is used to determine which parts point to the header and which point to the Info. This Mode is intended for high-performance protocol processing applications.

The Indicate Machine filters identical MAC and SMT frames when the SKIP bit in the Indicate Mode Register is set, and the Indicate Configuration Register's Copy Control field (2 bits) for Indicate Channel 0 is set to 01 or 10.

Received frames are copied to memory based on the AFLAG and MFLAG, ECIP, EA, and EM input signals from external address matching logic, input signals from the BMAC device, as well as the Indicate Channel's Copy Control field. Received frames are written as a series of Input Data Units to the current Indicate page. Each frame is aligned to the start of a currently-defined, burst-size memory block (16 or 32 bytes as programmed in the Mode Register's SMLB bit). The first word contains the FC only, copied

4.0 Functional Description (Continued)

into all bytes of the first word written, with the DA, SA and INFO fields aligned to the first byte of the next word. The format differs according to the setting of the Mode Register's BIGEND (Big Endian) bit, as shown in *Figure 4-1*.

Byte 0		Byte 3	
Bit 31	Big Endian Indicate Data Unit Format	Bit 0	Bit 0
FC	FC	FC	FC
DA0	DA1	SA0	SA1

Byte 3		Byte 0	
Bit 31	Little Endian Indicate Data Unit Format	Bit 0	Bit 0
FC	FC	FC	FC
SA1	SA0	DA1	DA0

FIGURE 4-1. Indicate Data Unit Formats (Short Addresses)

For each Input Data Unit, the Indicate Machine creates an Input Data Unit Descriptor (IDUD), which contains status information about the IDU, its size (byte count), and its location in memory. For IDUs that fit within the current Indicate page, an IDUD.Only Descriptor is created. For IDUs that span more than one page, a multi-part IDUD is created, i.e., when a frame crosses a page boundary, the BSI device writes an IDUD.First; if another page is crossed, an IDUD.Middle will be written; and at the frame end, an IDUD.Last is written. IDUDs are written to consecutive locations in the Indicate Status Queue for the particular Indicate Channel, up to the host-defined queue limit.

The Indicate Machine copies IDUs and IDUDs to memory as long as there are no exceptions or errors, and the Channel has data and status space. When a lack of either data or status space is detected on a particular Channel, the Indicate Machine stops copying new frames for that Channel (only). It will set the No Status Space attention bit in the No Space Attention Register when it runs out of Status Space. It will set the Low Data Space bit in the No Space Attention Register when the last available PSP is prefetched from the Indicate Channel PSP Queue. The host allocates more data space by adding PSPs to the tail of the PSP Queue and then updating the PSP Queue Limit Register, which causes the BSI device to clear the Low Data Space attention bit and resume copying (on the same Channel). The host allocates more status space by updating the IDUD Queue Limit Register and then explicitly clearing the Channel's No Status Space bit, after which the Indicate Machine resumes copying.

The BSI device provides the ability to group incoming frames and then generate interrupts (via attentions) at group boundaries. To group incoming frames, the BSI device defines status breakpoints, which identify the end of a group (burst) of related frames. Status breakpoints can be enabled to generate an attention.

The breakpoints for Indicate Channels are defined by the host in the Indicate Mode, Indicate Notify, and Indicate Threshold registers. Status breakpoints include Channel change, receipt of a token, SA change, DA change, MAC Info change, and the fact that a user-specified number of frames have been copied on a particular Indicate Channel. Status breakpoint generation may be individually enabled for Indicate Channels 1 and 2 by setting the corresponding

Breakpoint bits (Breakpoint on Burst End, Breakpoint on Service Opportunity, and Breakpoint on Threshold) in the Indicate Mode Register, and enabling the breakpoints to generate an attention by setting the corresponding Breakpoint bit in the Indicate Notify Register.

When an Indicate exception occurs, the current frame is marked complete, status is written into an IDUD.Last, and the Channel's Exception (EXC) bit in the Indicate Attention Register is set.

When an Indicate error (other than a parity error) is detected, the Channel's Error (ERR) bit in the State Attention Register is set. The host must reset the INSTOP Attention bit to restart processing on the Indicate Channel.

When parity checking is enabled and a parity error is detected in a received frame, it is recorded in the Indicate Status field of the IDUD, and the BMAC device Parity Error (PBE) bit in the Status Attention Register is set.

4.2.2 Request Operation

The Request Block transmits frames from host memory to the BMAC device. Data is presented to the BMAC device as a byte stream.

The Request Block performs the following functions:

- Prioritizes active requests to transmit frames
- Requests the BMAC device to obtain a token
- Transmits frames to the BMAC device
- Writes status for transmitted and returning frames
- Issues interrupts to the host on user-defined group boundaries

The Request Machine processes requests by reading Request Descriptors from the REQ Queue, then assembling frames of the specified service class, frame control and expected status for transmission to the BMAC device. Request and ODUD Descriptors are checked for consistency, and the Request Class is checked for compatibility with the current ring state. When an inconsistency or incompatibility is detected, the request is aborted.

When a consistency failure occurs, the Request is terminated with appropriate status. The Request Machine then locates the end of the current object (REQ or ODUD). If the current Descriptor is not the end (Last bit not set), the Request Machine will fetch subsequent Descriptors until it detects the end, then resume processing with the next Descriptor.First or Descriptor.Only.

Requests are processed on both Request Channels simultaneously. Their interaction is determined by their priorities (Request Channel 0 has higher priority than Request Channel 1) and the Hold and Preempt/Prestage bits in the Request Channel's Request Configuration Register. An active Request Channel 0 is always serviced first, and may be programmed to preempt Request Channel 1, such that uncommitted Request Channel 1's data already in the request FIFO will be purged and then refetched after servicing Request Channel 0. When prestaging is enabled, the next frame is staged before the token arrives. Prestaging is always enabled for Request Channel 0, and is a programmable option on Request Channel 1.

When a REQ.First is loaded, the Request Machine commands the BMAC device to capture a token of the type specified in the REQ Descriptor, and concurrently fetches the first ODUD. If prestaging is enabled, or a service opportunity exists for this Request Channel, data from the first

4.0 Functional Description (Continued)

ODU is loaded into the Request FIFO, and the BSI device requests transmission from the BMAC device. When the BMAC device has captured the appropriate token and the frame is committed to transmission (the FIFO threshold has been reached or the end of the frame is in the FIFO), transmission begins. The BSI device fetches the next ODU and starts loading the ODUs of the next frame into the FIFO. This continues (across multiple service opportunities if required) until all frames for that Request have been transmitted (i.e., a REQ.ONLY or a REQ.LAST is detected), or an exception or error occurs, which prematurely ends the Request.

The BSI device will load REQ Descriptors as long as the RQSTOP bit in the State Attention Register is Zero, the REQ Queue contains valid entries (the REQ Queue Pointer Register does not exceed the REQ Queue Limit Register), and there is space in the CNF Queue (the CNF Queue Pointer Register is less than the CNF Queue Limit Register).

Request status is generated as a single confirmation object (single- or multi-part) per Request object, with each confirmation object consisting of one or more CNF Descriptors. The type of confirmation is specified by the host in the Confirmation Class field of the REQ Descriptor.

The BSI device can be programmed to generate CNF Descriptors at the end of the Request object (End Confirmation), or at the end of each token opportunity (Intermediate Confirmation), as selected in the E and I bits of the Request Class Field of the REQ Descriptor. A CNF Descriptor is always written when an exception or error occurs (regardless of the value in the Confirmation Class field), when a Request completes (for End or Intermediate Confirmation Class), or when an enabled breakpoint occurs (Intermediate Confirmation Class only).

There are three basic types of confirmation: Transmitter, Full, and None. With Transmitter Confirmation, the BSI device verifies that the Output Data Units were successfully transmitted. With Full Confirmation, the Request Machine verifies that the ODUs were successfully transmitted, that the number of (returning) frames "matches" the number of transmitted ODUs, and that the returning frames contain the expected status. When the None Confirmation Class is selected, confirmation is written only if an exception or error occurs.

For Full Confirmation, a matching frame must meet the following criteria:

1. The frame has a valid Ending Delimiter (ED).
2. The selected bits in the FC fields of the transmitted and received frames are equal (the selected bits are specified in the FCT bit of the Request Configuration Register).
3. The frame is My__SA (MFLAG or both SAT & EM asserted).
4. The frame matches the values in the Expected Frame Status Register.
5. FCS checking is disabled or FCS checking is enabled and the frame has a valid FCS.
6. All bytes from FC to ED have good parity (when the FLOW bit in the Mode Register is set, i.e., parity checking is enabled).

The confirmed frame count starts after the first Request burst frame has been committed by the BMAC device, and when a frame with My__SA is received. Void and My_Void

frames are ignored by the BSI device. The frame count ends when any of the following conditions occur:

1. All the frames have been transmitted, and the transmitted and confirmed frame counts are equal.
2. There is a MACRST (MAC Reset).
3. The state of the ring-operation has changed.
4. A stripped frame or a frame with a parity error is received.
5. A non-matching frame is received.
6. A token is received.

When Source Address Transparency is selected (by setting the SAT bit in the Request Configuration Register) and Full confirmation is enabled, confirmation begins when a frame end is detected with either MFLAG or EM asserted.

When a non-matching frame is received, the BSI device ends the Request, and generates the Request Complete (RCM), Exception (EXC), and Breakpoint (BRK) attentions. Any remaining REQs in the Request object are fetched until a REQ.Last or REQ.Only is encountered. Processing then resumes on the next REQ.First or REQ.Only (any other type of REQ would be a consistency failure).

Request errors and exceptions are reported in the State Attention Register, Request Attention Register, and the Confirmation Message Descriptor. When an exception or error occurs, the Request Machine generates a CNF and ends the Request. The Unserviceable Request (USR) attention is set to block subsequent Requests once one becomes unserviceable.

4.2.3 State Machines

There are three state machines under control of the host: the Request Machine, the Indicate Machine, and the Status/Space Machine. Each Machine has two Modes: Stop and Run. The Mode is determined by the setting of the Machine's corresponding STOP bit in the State Attention Register. The STOP bits are set by the BSI device when an error occurs or may be set by the user to place the Machine in Stop Mode.

The BSI Control Registers may be programmed only when all Machines are in Stop Mode. When the Status/Space Machine is in Stop Mode, only the Pointer RAM and Limit RAM Registers may be programmed.

When the Indicate and Request Machines are in Stop Mode, all indicate and request operations are halted. When the Status/Space Machine is in Stop Mode, only the PTOP and LMOP service functions can be performed.

4.3 BUS INTERFACE UNIT

4.3.1 Overview

The ABus provides a 32-bit wide synchronous multiplexed address/data bus for transfers between the host system and the BSI device. The ABus uses a bus request/bus grant protocol that allows multiple bus masters, supports burst transfers of 16 and 32 bytes, and supports virtual and physical addressing using fixed-size pages. The BSI is capable of operating directly on the system bus to main memory, or connected to external shared memory.

All bus signals are synchronized to the master bus clock. The maximum burst speed is one, 32-bit word per clock, but slower speeds may be accommodated by inserting wait states. The user may use separate clocks for the ring (FDDI MAC) and system (ABus) interfaces. The only restriction is that the ABus clock must be at least as fast as the ring clock

4.0 Functional Description (Continued)

(LBC). It is important to note that all ABus outputs change and all ABus inputs are sampled on the **rising** edge of AB_CLK.

Addressing Modes

The Bus Interface Unit has two Address Modes, as selected by the user: Physical Address Mode and Virtual Address Mode. In Physical Address Mode, the BSI device emits the memory address and immediately begins transferring the data. In Virtual Address Mode, the BSI device inserts two clock cycles and TRI-STATE™s the address between emitting the virtual address and starting to transfer the data. This allows virtual-to-physical address translation by an external MMU.

The BSI device interfaces to byte-addressable memory, but always transfers information in words. The BSI device uses a word width of 32 data bits plus 4 (1 per byte) parity bits. Parity may be ignored.

Bus Transfers

The bus supports several types of transactions. Simple reads and writes involve a single address and data transfer. Burst reads and writes involve a single address transfer followed by multiple data transfers. The BSI device provides the incrementing address bits during the burst transaction. Burst sizes are selected dynamically by the BSI.

On Indicate Channels, when 8-word bursts are enabled, all transactions will be 8 words until the end of the frame; the last transfer will be 4 or 8 words, depending on the number of remaining bytes. If only 4-word bursts are allowed, all Indicate Data transfers are 4 words.

On Request Channels, the BSI will use 4- or 8-word bursts to access all data up to the end of the ODU. If 8-word bursts are enabled, the first access will be an 8-word burst if the ODU begins less than 4 words from the start of an 8-word burst boundary. If 8-word bursts are not allowed, or if the ODU begins 4 or more words from the start of an 8-word burst boundary, a 4-word burst will be used. The BSI will ignore unused bytes if the ODU does not start on a burst boundary. At the end of an ODU, the BSI will use the smallest transfer size (1, 4, or 8 words) which completes the ODU read. To coexist in a system that assumes implicit wrap-around for the addresses within a burst, the BSI device never emits a burst that will wrap the 4- or 8-word boundary.

A Function Code identifying the type of transaction is output by the BSI device on the upper four address bits during the address phase of a data transfer. This can be used for more elaborate external addressing schemes, for example, to direct control information to one memory and data to another (e.g., an external FIFO). To assist the user with the burst transfer capability, the BSI device also outputs three demultiplexed address bits during a burst transfer. These indicate the next word within a burst to be accessed.

Byte Ordering

The basic addressable quantum is a byte, so request data may be aligned to any byte boundary in memory. All information is accessed in 32-bit words, however, so the BSI device ignores unused bytes when reading.

Descriptors must always be aligned to a word address boundary. Input Data Units are always aligned to a burst-size boundary. Output Data Units may be any number of bytes, aligned to any byte-address boundary, but operate most efficiently when aligned to a burst-size boundary.

Burst transfers are always word-aligned on a 16- or 32-byte (burst-size) address boundary. Burst transfers will never cross a burst-size boundary. If a 32-byte transfer size is chosen, the BSI device will perform both 16-byte and 32-byte bursts, whichever is most efficient (least number of clocks to load/store all required data).

The Bus Interface Unit can operate in either Big Endian or Little Endian Mode. The bit and byte alignments for both modes are shown in *Figure 4-2*. Byte 0 is the first byte received from the ring or transmitted to the ring.

Bus Arbitration

The ABus is a multi-master bus, using a simple Bus Request/Bus Grant protocol that allows an external Bus Arbiter to support any number of bus masters, using any arbitration scheme (e.g., rotating or fixed priority). The protocol provides for multiple transactions per tenure, and bus master preemption.

The BSI device asserts a Bus Request, and assumes mastership when Bus Grant is asserted. If the BSI device has another transaction pending, it will keep Bus Request asserted, or reassert it before the completion of the current transaction. If Bus Grant is (re)asserted before the end of the current transaction, the BSI device retains mastership and runs the next transaction. This process may be repeated indefinitely.

If the Bus Arbiter wishes to preempt the BSI device, it deasserts Bus Grant. The BSI device will complete the current bus transaction, then release the bus. From preemption to bus release is a maximum of (11 bus clocks + (8 times the number of memory wait states)) bus clocks. For example, in a 1 wait-state system, the BSI device will release the bus within a maximum of 19 bus clocks.

Big-Endian Byte Order

D[31]		D[0]	
Word			
Halfword 0		Halfword 1	
Byte 0	Byte 1	Byte 2	Byte 3

Little-Endian Byte Order

D[31]		D[0]	
Word			
Halfword 1		Halfword 0	
Byte 3	Byte 2	Byte 1	Byte 0

FIGURE 4-2. ABus Byte Orders

Parity

There are two options for parity: one for systems using parity, the other for systems not using parity. Parity checking on the ABus can be disabled by clearing the FLOW bit in the Mode Register. When parity is enabled (FLOW bit is set), it operates in flow-through mode on the main datapath, that is, parity is not checked at the ABus but simply flows between the ABus and the BMAC device interface, and is checked by the BMAC device as it is received. When the FLOW bit is set, parity checking is also enabled on the Control Bus and MAC Indicate Interfaces.

The BSI device generates parity on all addresses output on the ABus.

4.0 Functional Description (Continued)

Bandwidth

The ABus supports single reads and writes, and burst reads and writes. With physical addressing, back-to-back single reads/writes can take place every four clock cycles. Burst transactions can transfer 8, 32-bit words (32 bytes) every 11 clock cycles. With a 25 MHz clock this yields a peak bandwidth of 72.7 Mbytes/sec.

To allow the bus to operate at high frequency, the protocol defines all signals to be both asserted and deasserted by the bus master and slaves. Having a bus device actively deassert a signal guarantees a high-speed inactive transition. If this were not defined, external pull-up resistors would not be able to deassert signals fast enough. The protocol also reduces contention by avoiding cases where two bus devices simultaneously drive the same line.

The BSI device operates synchronously with the ABus clock. In general, operations will be asynchronous to the ring, since most applications will use a system bus clock that is asynchronous to the ring. The BSI device is designed to interface either directly to the host's main system bus or to external shared memory. When interfaced to the host's bus, there are two parameters of critical interest: latency and bandwidth.

Data moves between the Request and Indicate Channels and the ABus via four FIFOs, two in the receive path (Indicate) and two in the transmit path (Request). On the BMAc Device Interface, there are two, 16 x 32 bit data FIFOs for Indicate and Request data. On the ABus Interface, there are

two Burst FIFOs, each containing two banks of 32 bytes, which provide ABus bursting capability.

The amount of latency covered by the Data FIFO plus one of the banks of the Burst FIFO must meet the average and maximum bus latency of the external memory. With a new byte every 80 ns from the ring, a 64-byte FIFO provides $64 \times 80 = 5.12 \mu\text{s}$ maximum latency.

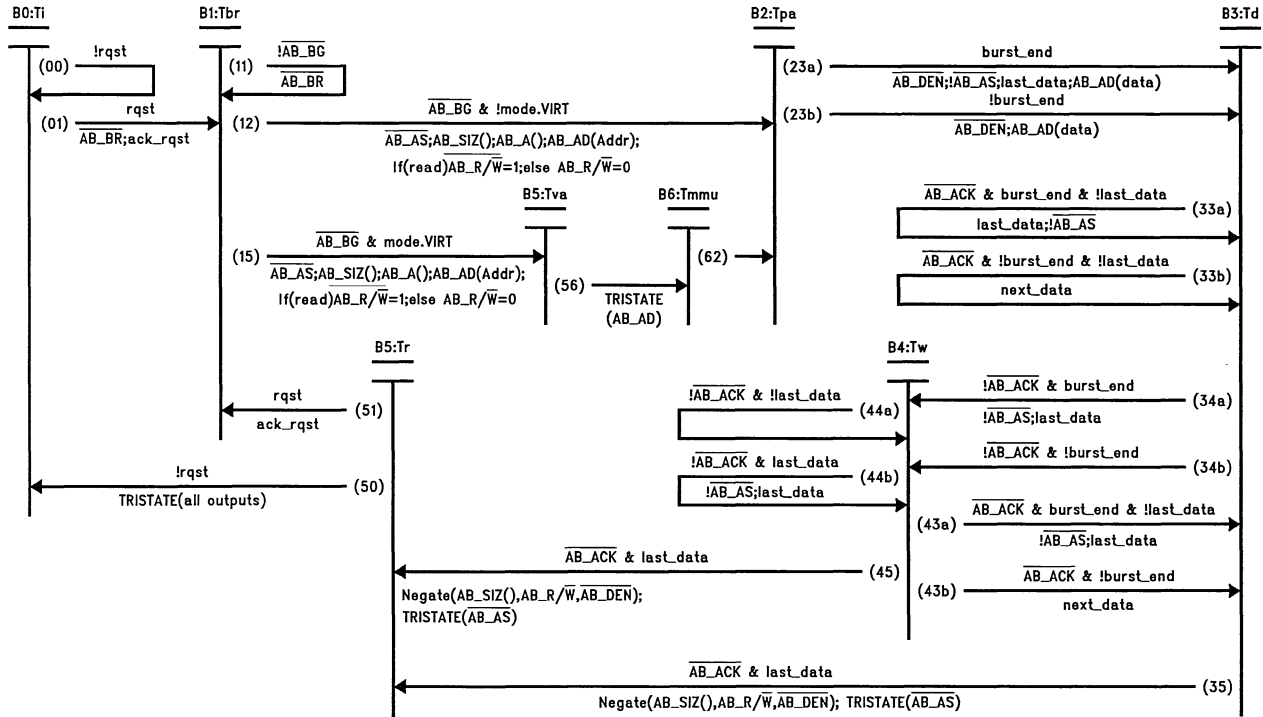
To assist latency issues, the BSI device can completely empty or fill the Burst FIFO in one bus tenure by asserting Bus Request for multiple transactions. Since one bank of the Burst FIFO is 8 words deep, if 8-word bursts are enabled, that half of the Burst FIFO can be emptied in one transaction. If the second half of the burst FIFO is also full, it can be emptied in the same bus tenure by again granting the bus to the BSI device.

The BSI device may be preempted at any time by removing Bus Grant, which causes the BSI device to complete the current transaction and release the bus. There will be a maximum of 11 clocks (plus any memory wait states) from preemption to bus release (fewer if 8-word bursts are not enabled).

4.3.2 Bus States

An ABus Master has eight states: idle (Ti), bus request (Tbr), virtual address (Tva), MMU translate (Tmmu), physical address (Tpa), data transfer (Td), wait (Tw) and recovery (Tr). The ABus Master state diagram is shown in *Figure 4-3*.

An ABus Slave has five states: idle (Ti), selected (Ts), data transfer (Td), wait (Tw), and recovery (Tr).



TL/F/10791-6

- Note 1:** If ($\overline{AB_ACK}$) next_data decrements burst_count, processes data.
- Note 2:** Burst_count is pipelined 1 ahead, so is true in last Td state.
- Note 3:** If ($\overline{AB_ACK}$) last_data negates $\overline{AB_AS}$, processes data.
- Note 4:** Timing for $\overline{AB_BR}$ not shown.

FIGURE 4-3. BSI Device ABus Master State Machine (from the BSI Device)

4.0 Functional Description (Continued)

Master States

The Ti state exists when no bus activity is required. The BIU does not drive any of the bus signals when it is in this state (all are released). If the BIU requires bus service, it may assert Bus Request.

When a transaction is run, the BIU enters Tbr and asserts Bus Request, then waits for Bus Grant to be asserted.

The state following Tbr is either Tva or Tpa. In Virtual Address Mode, the BIU enters Tva and drives the virtual address and size lines onto the bus. In Physical Address Mode, Tpa occurs next (see Section 4.3.3).

Following a Tva state is a Tmmu state. During this cycle the external MMU is performing a translation of the virtual address emitted during Tva.

Following a Tmmu state (when using virtual addressing) or a Tbr state (when using physical addressing), is the Tpa state. During the Tpa state, the BSI device drives the read/write strobes and size signals. In physical address mode, it also drives AB_AD with address. In virtual address mode, the BSI device TRI-STATES AB_BD so the host CPU or MMU can drive the address.

Following the Tpa state, the BIU enters the Td state to transfer data words. Each data transfer may be extended indefinitely by inserting Tw states. A slave acknowledges by asserting AB_ACK and transferring data in a Td state (cy-

cle). If the slave can drive data at the rate of one word per clock (in a burst), it keeps AB_ACK asserted.

Following the final Td/Tw state, the BIU enters a Tr state to allow time to turn off or turn around bus transceivers.

A bus retry request is recognized in any Td/Tw state. The BIU will go to a Tr state and then rerun the transaction when it obtains a new Bus Grant. The whole transaction is retried, i.e., all words of a burst. Additionally, no other transaction will be attempted before the interrupted one is retried. The BIU retries indefinitely until either the transaction completes successfully, or a bus error is signaled.

Bus errors are recognized in Td/Tw states.

4.3.3 Physical Addressing Bus Transactions

Bus transactions in Physical Address Mode are shown in Figure 4-4 through 4-7. BSI device signals are defined in Chapter 6.

Single Read

Tbr: BSI device asserts AB_BF to indicate it wishes to perform a transfer. Host asserts AB_BG. Moves to Tpa on the next clock.

Tpa: BSI device drives AB_A and AB_AD with the address, asserts AB_AS, drives AB_RW and AB_SIZ[2:0], negates AB_BF if another transaction is not required.

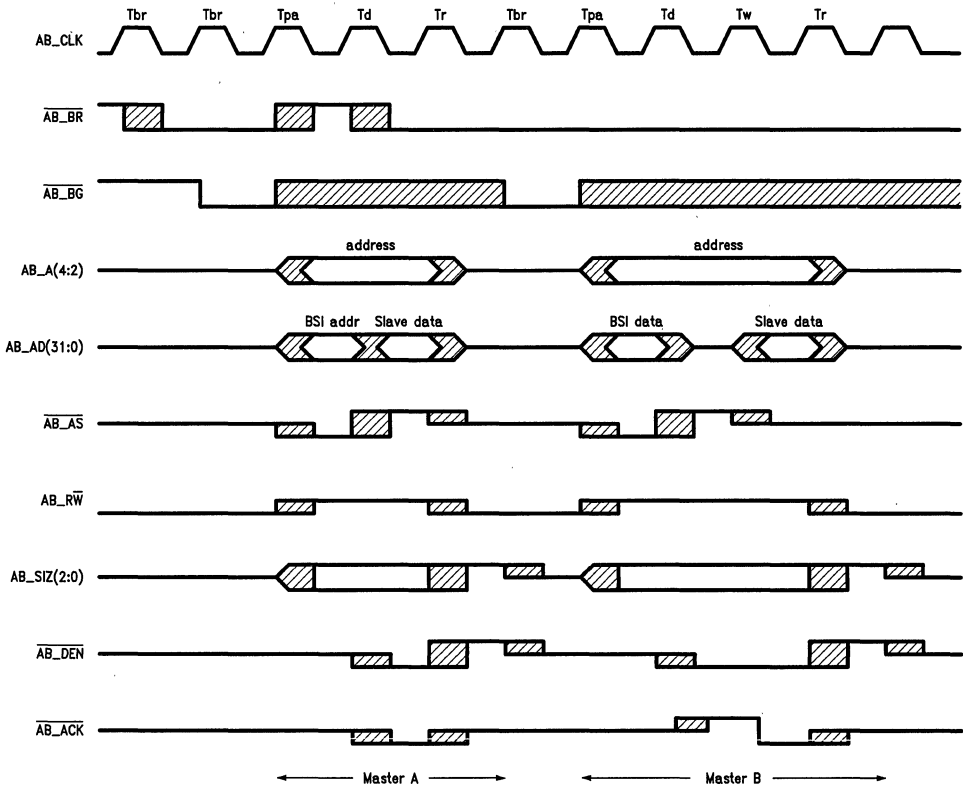


FIGURE 4-4. ABus Single Read, Physical Addressing, 0 W-S, 1 W-S, Bus Handover

TL/F/10791-7

4.0 Functional Description (Continued)

Td: BSI device negates $\overline{AB_AS}$, asserts $\overline{AB_DEN}$, samples $\overline{AB_ACK}$ and $\overline{AB_ERR}$. Slave asserts $\overline{AB_ACK}$, drives $\overline{AB_ERR}$, drives AB_AD (with data) when ready. The BSI device samples a valid $\overline{AB_ACK}$, capturing the read data. T_w states may occur after T_d .

Tr: BSI device negates AB_RW , $\overline{AB_DEN}$, $AB_SIZ[2:0]$, releases AB_A , and $\overline{AB_AS}$. Slave deasserts $\overline{AB_ACK}$ and $\overline{AB_ERR}$, releases AB_AD

Tpa: BSI device drives AB_A and AB_AD with the address, asserts $\overline{AB_AS}$, drives AB_RW and $AB_SIZ[2:0]$, and negates $\overline{AB_BR}$ if another transaction is not required.

Td: BSI device negates $\overline{AB_AS}$, asserts $\overline{AB_DEN}$, drives AB_AD with the write data and starts sampling $\overline{AB_ACK}$ and $\overline{AB_ERR}$. Slave captures AB_AD data, asserts $\overline{AB_ACK}$, drives $\overline{AB_ERR}$. T_w states may occur after T_d if the slave deasserts $\overline{AB_ACK}$.

Tr: BSI device negates AB_RW , $\overline{AB_DEN}$, $AB_SIZ[2:0]$, releases AB_A , AB_AD , $\overline{AB_AS}$. Slave deasserts $\overline{AB_ACK}$ and $\overline{AB_ERR}$, and stops driving AB_AD with data.

Single Write

Tbr: BSI device asserts $\overline{AB_BR}$ to indicate it wishes to perform a transfer. Host asserts $\overline{AB_BG}$. Moves to T_{pa} on the next clock.

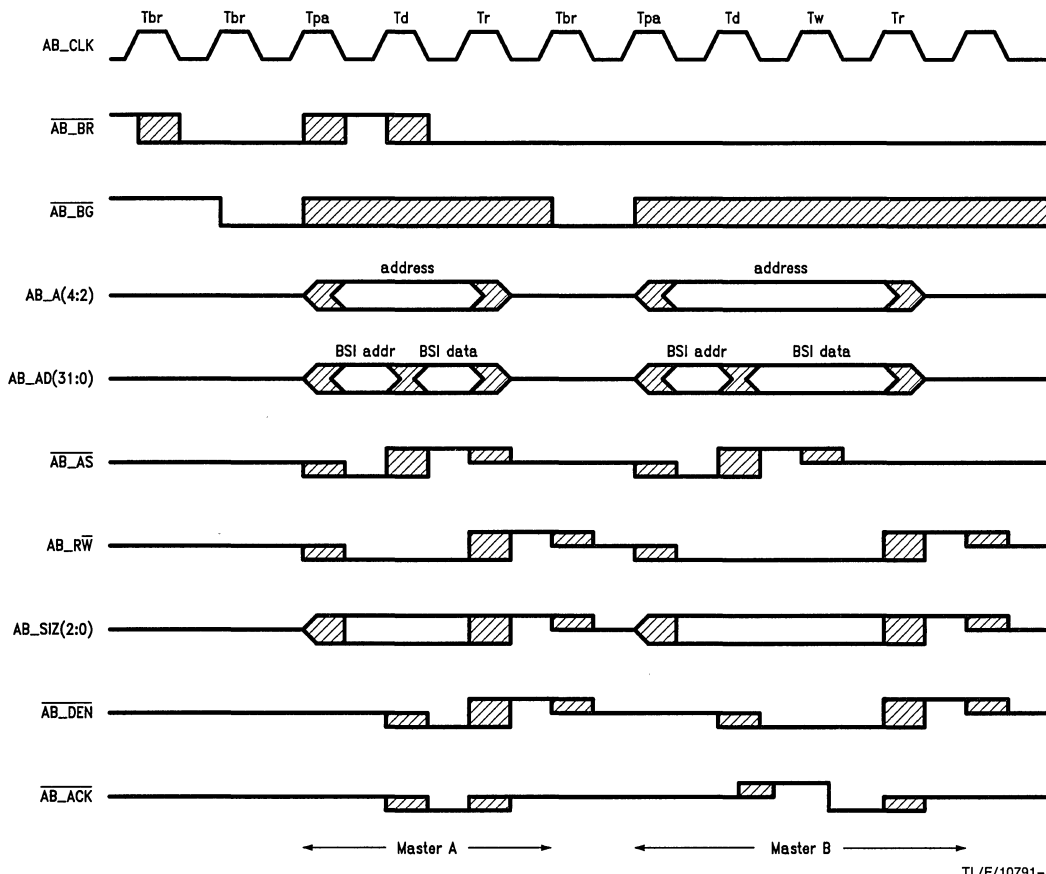


FIGURE 4-5. ABus Single Write, Physical Addressing, 0 W-S, 1 W-S, Bus Handover

TL/F/10791-8

4.0 Functional Description (Continued)

Burst Read

Tbr: BSI device asserts $\overline{AB_BR}$ to indicate it wishes to perform a transfer. Host asserts $\overline{AB_BG}$. Moves to Tpa on the next clock.

Tpa: BSI device drives $\overline{AB_A}$ and $\overline{AB_AD}$ with the address, asserts $\overline{AB_AS}$, drives $\overline{AB_RW}$ and $\overline{AB_SIZ}[2:0]$, and negates $\overline{AB_BR}$ if another transaction is not required.

Td: BSI device asserts $\overline{AB_DEN}$, samples $\overline{AB_ACK}$ and $\overline{AB_ERR}$, increments the address on $\overline{AB_A}$. Slave asserts $\overline{AB_ACK}$, drives $\overline{AB_ERR}$, and drives $\overline{AB_AD}$ (with data) when ready. BSI device samples a valid $\overline{AB_ACK}$, capturing the read data. Tw states may occur after Td. Td state is repeated four or eight times (according to the burst size). On the last Td state, the BSI device negates $\overline{AB_AS}$.

Tr: BSI device negates $\overline{AB_RW}$, $\overline{AB_DEN}$, $\overline{AB_SIZ}[2:0]$, and releases $\overline{AB_A}$ and $\overline{AB_AS}$. Slave deasserts $\overline{AB_ACK}$ and $\overline{AB_ERR}$, and releases $\overline{AB_AD}$.

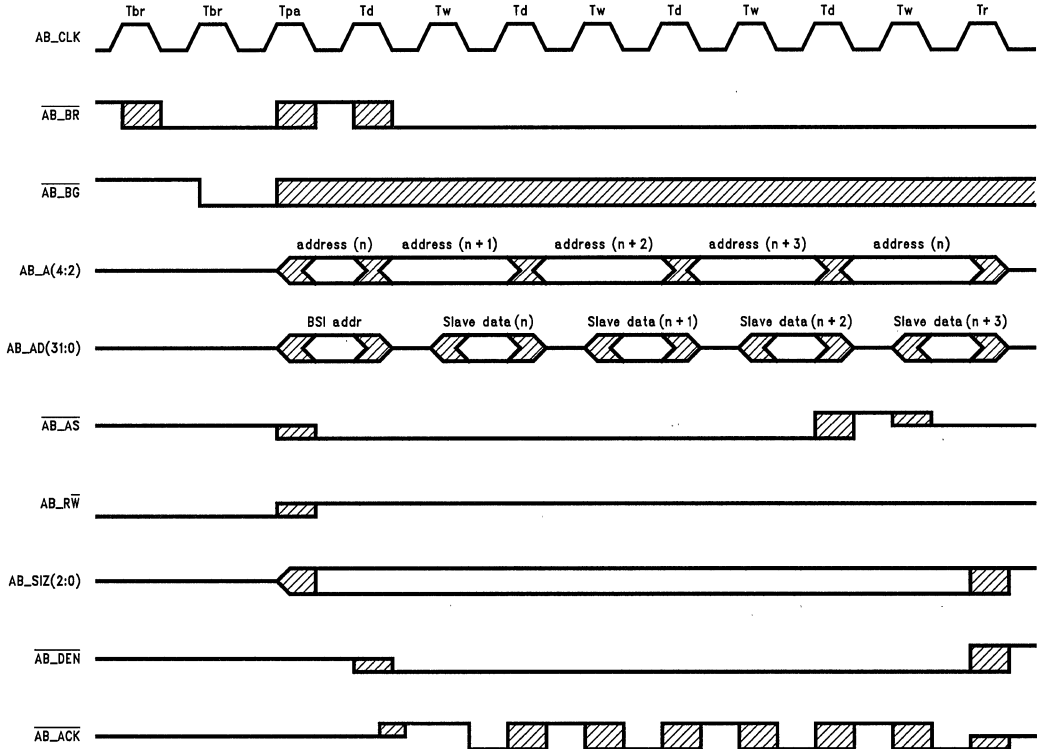


FIGURE 4-6. ABus Burst Read, Physical Addressing, 16 Bytes, 1 W-S

TL/F/10791-9

4.0 Functional Description (Continued)

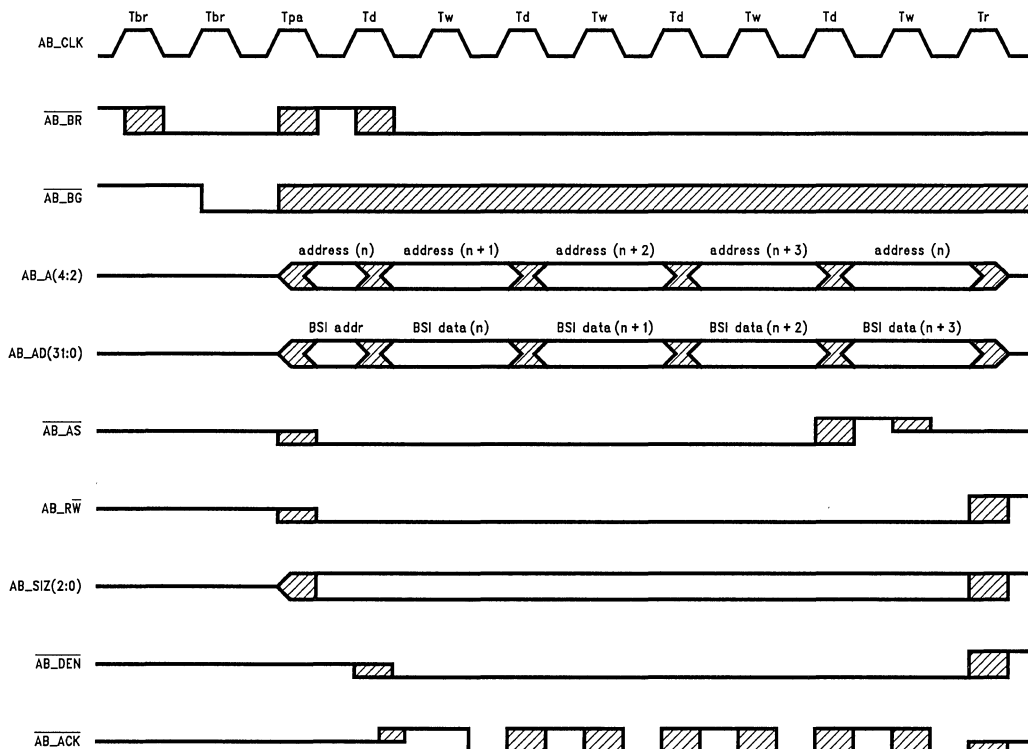
Burst Write

Tbr: BSI device asserts $\overline{AB_BR}$ to indicate it wishes to perform a transfer. Host asserts $\overline{AB_BG}$. Moves to Tpa on the next clock.

Tpa: BSI device drives AB_A and AB_AD with the address, asserts $\overline{AB_AS}$, drives AB_RW and $AB_SIZ[2:0]$, and negates $\overline{AB_BR}$ if another transaction is not required.

Td: BSI device asserts $\overline{AB_DEN}$, drives AB_AD with the write data, samples $\overline{AB_ACK}$ and AB_ERR , increments the address on AB_A . Slave captures AB_AD data, asserts $\overline{AB_ACK}$, drives $\overline{AB_ERR}$. BSI device samples a valid $\overline{AB_ACK}$. Tw states may occur after Td. Td state is repeated as required for the complete burst. On the last Td state, the BSI device negates $\overline{AB_AS}$.

Tr: BSI device negates AB_RW , $\overline{AB_DEN}$, $AB_SIZ[2:0]$, releases AB_A and $\overline{AB_AS}$. Slave deasserts $\overline{AB_ACK}$ and $\overline{AB_ERR}$, and stops driving AB_AD with data.



TL/F/10791-10

FIGURE 4-7. ABus Burst Write, Physical Addressing, 16 Bytes 1 W-S

4.0 Functional Description (Continued)

4.3.4 Virtual Addressing Bus Transactions

Single Read

- Tbr:** BSI device asserts $\overline{AB_BR}$ to indicate it wishes to perform a transfer. Host asserts $\overline{AB_BG}$, and BSI device drives AB_A and AB_AD when $\overline{AB_BG}$ is asserted. Moves to Tva on the next clock.
- Tva:** BSI device drives AB_A and AB_AD with the virtual address for one clock, negates $\overline{AB_AS}$, asserts AB_RW , drives $AB_SIZ[2:0]$, and negates $\overline{AB_BR}$ if another transaction is not required.
- Tmmu:** Host MMU performs an address translation during this clock.
- Tpa:** Host MMU drives AB_AD with the translated (physical) address, BSI device drives AB_A and asserts $\overline{AB_AS}$.
- Td:** BSI device negates $\overline{AB_AS}$, asserts $\overline{AB_DEN}$, samples $\overline{AB_ACK}$ and $\overline{AB_ERR}$. Slave asserts $\overline{AB_ACK}$, drives $\overline{AB_ERR}$, drives AB_AD (with data) when ready. BSI device samples a valid $\overline{AB_ACK}$, capturing the read data. Tw states may occur after Td.
- Tr:** BSI device negates AB_RW , $\overline{AB_DEN}$, and $AB_SIZ[2:0]$, releases AB_A and $\overline{AB_AS}$. Slave deasserts $\overline{AB_ACK}$ and $\overline{AB_ERR}$ and releases AB_AD .

Single Write

- Tbr:** BSI device asserts $\overline{AB_BR}$ to indicate it wishes to perform a transfer. Host asserts $\overline{AB_BG}$, BSI device drives AB_A and AB_AD when $\overline{AB_BG}$ is asserted. Moves to Tva on the next clock.
- Tva:** BSI device drives AB_A and AB_AD with the virtual address for one clock, negates $\overline{AB_AS}$, negates AB_RW , and drives $AB_SIZ[2:0]$.
- Tmmu:** Host MMU performs an address translation during this clock.
- Tpa:** Host MMU drives AB_AD with the address, BSI device drives AB_A asserts $\overline{AB_AS}$, and negates $\overline{AB_BR}$ if another transaction is not required.
- Td:** BSI device negates $\overline{AB_AS}$, asserts $\overline{AB_DEN}$, drives AB_AD with the write data and starts sampling $\overline{AB_ACK}$ and $\overline{AB_ERR}$. Slave captures AB_AD data, asserts $\overline{AB_ACK}$, and drives $\overline{AB_ERR}$. BSI device samples a valid $\overline{AB_ACK}$. Tw states may occur after Td.
- Tr:** BSI device negates AB_RW , $\overline{AB_DEN}$, $AB_SIZ[2:0]$, releases AB_A , AB_AD , and $\overline{AB_AS}$. Slave deasserts $\overline{AB_ACK}$ and $\overline{AB_ERR}$, and stops driving AB_AD with data.

Burst Read

- Tbr:** BSI device asserts $\overline{AB_BR}$ to indicate it wishes to perform a transfer. Host asserts $\overline{AB_BG}$, BSI drives AB_A and AB_AD when $\overline{AB_BG}$ is asserted. Moves to Tva on the next clock.
- Tva:** BSI device drives AB_A and AB_AD with the virtual address for one clock, negates $\overline{AB_AS}$, asserts AB_RW , drives $AB_SIZ[2:0]$, and negates $\overline{AB_BR}$ if another transaction is not required.
- Tmmu:** Host MMU performs an address translation during this clock.
- Tpa:** Host MMU drives AB_AD with the translated (physical) address. BSI device drives AB_A and asserts $\overline{AB_AS}$.
- Td:** BSI device asserts $\overline{AB_DEN}$, samples $\overline{AB_ACK}$ and $\overline{AB_ERR}$. Slave asserts $\overline{AB_ACK}$, drives AB_AD (with data) when ready. BSI device samples a valid $\overline{AB_ACK}$, capturing the read data. Tw states may occur after Td. This state is repeated four or eight times (according to burst size). On the last Td state the BSI device negates $\overline{AB_AS}$.
- Tr:** BSI device negates AB_RW , $\overline{AB_DEN}$, $AB_SIZ[2:0]$, releases AB_A and $\overline{AB_AS}$. Slave deasserts $\overline{AB_ACK}$ and $\overline{AB_ERR}$, and releases AB_AD .

Burst Write

- Tbr:** BSI device asserts $\overline{AB_BR}$ to indicate it wishes to perform a transfer. Host asserts $\overline{AB_BG}$, BSI device drives AB_A and AB_AD when $\overline{AB_BG}$ is asserted. Moves to Tva on the next clock.
- Tva:** BSI device drives AB_A and AB_AD with the virtual address for one clock, negates $\overline{AB_AS}$, negates AB_RW , drives $AB_SIZ[2:0]$.
- Tmmu:** Host MMU performs an address translation during this clock.
- Tpa:** Host MMU drives AB_AD with the address, BSI device drives AB_A asserts $\overline{AB_AS}$, and negates $\overline{AB_BR}$ if another transaction is not required.
- Td:** BSI device asserts $\overline{AB_DEN}$, drives AB_AD with the write data and starts sampling $\overline{AB_ACK}$ and $\overline{AB_ERR}$. Slave captures AB_AD data, asserts $\overline{AB_ACK}$ and drives $\overline{AB_ERR}$. BSI device samples a valid $\overline{AB_ACK}$. Tw states may occur after Td. This state is repeated as required for the complete burst. On the last Td state, the BSI device negates $\overline{AB_AS}$.
- Tr:** BSI device negates AB_RW , $\overline{AB_DEN}$, $AB_SIZ[2:0]$, releases AB_A , AB_AD , $\overline{AB_AS}$. Slave deasserts $\overline{AB_ACK}$ and $\overline{AB_ERR}$, stops driving AB_AD with data.

5.0 Control Information

5.1 OVERVIEW

Control information includes the parameters that are used to manage and operate the BSI device.

Control information is divided into four basic groups: Operation Registers, Pointer RAM Registers, Limit RAM Registers, and Descriptors. The Control information Register Address Space is shown in Table 5-1.

Operation registers are accessed directly via the Control Bus. Limit RAM Registers are accessed indirectly via the Control Bus, using the Limit RAM Data and Limit RAM Address Registers. The Pointer RAM Registers are accessed indirectly via the Control Bus and ABUS using the Pointer RAM Address and Control Register, the Mailbox Address Register, and a mailbox location in ABUS memory.

5.2 OPERATION REGISTERS

The Operation Registers are divided into two functional groups: Control Registers and Event Registers. They are shown in Table 5-2.

Control Registers

The Control Registers are used to configure and control the operation of the BSI device.

The Control Registers include the following registers:

- **Mode Register (MR)** establishes major operating parameters for the BSI device.
- **Pointer RAM Control and Address Register (PCAR)** is used to program the parameters for the PTO (Pointer RAM Operation) service function.
- **Mailbox Address Register (MBAR)** is used to program the memory address of the mailbox used in the data transfer of the PTO service function.

- **Limit Address Register (LAR)** is used to program the parameters and data used in the LMOP (Limit RAM Operation) service function.
- **Limit Data Register (LDR)** is used to program the data used in the LMOP service function.
- **Request Channel 0 Configuration Register (R0CR)** is used to program the operational parameters for Request Channel 0.
- **Request Channel 1 Configuration Register (R1CR)** is used to program the operational parameters for Request Channel 1.
- **Request Channel 0 Expected Frame Status Register (R0EFR)** defines the expected frame status for frames being confirmed on Request Channel 0.
- **Request Channel 1 Expected Frame Status Register (R1EFR)** defines the expected frame status for frames being confirmed on Request Channel 1.
- **Indicate Threshold Register (ITR)** is used to specify a maximum number of frames that can be copied onto an Indicate Channel before a breakpoint is generated.
- **Indicate Mode Register (IMR)** specifies how the incoming frames are sorted onto Indicate Channels, enables frame filtering, and enables breakpoints on various burst boundaries.
- **Indicate Configuration Register (ICR)** is used to program the copy criteria for each of the Indicate Channels.
- **Indicate Header Length Register (IHLR)** defines the length of the frame header for use with the Header/Info Sort Mode.

Table 5.1 Control Register Address Space

Address Range	Description	Read Conditions	Write Conditions
00–1Fh	Operation Registers	Always	Always (Conditional)
00–15h*	Pointer RAM Registers	Always	Always
0–9h**	Limit RAM Registers	Always	Always

*Bits 0–4 of Pointer RAM Address and Control Register

**Bits 4–7 of Limit RAM Address Register

5.0 Control Information (Continued)

TABLE 5-2. Control and Event Registers

Register Group	Address	Register Name	Access Rules	
			Read	Write
C	00	Mode Register (MR)	Always	Always
C	01	Reserved	N/A	N/A
C	02	Pointer RAM Control and Address Register (PCAR)	Always	Always
C	03	Mailbox Address Register (MBAR)	Always	Always
E	04	Master Attention Register (MAR)	Always	Data Ignored
E	05	Master Notify Register (MNR)	Always	Always
E	06	State Attention Register (STAR)	Always	Conditional
E	07	State Notify Register (STNR)	Always	Always
E	08	Service Attention Register (SAR)	Always	Conditional
E	09	Service Notify Register (SNR)	Always	Always
E	0A	No Space Attention Register (NSAR)	Always	Conditional
E	0B	No Space Notify Register (NSNR)	Always	Always
C	0C	Limit Address Register (LAR)	Always	Always
C	0D	Limit Data Register (LDR)	Always	Always
E	0E	Request Attention Register (RAR)	Always	Conditional
E	0F	Request Notify Register (RNR)	Always	Always
C	10	Request Channel 0 Configuration Register (ROCR)	Always	Always
C	11	Request Channel 1 Configuration Register (R1CR)	Always	Always
C	12	Request Channel 0 Expected Frame Status Register (R0EFSR)	Always	Always
C	13	Request Channel 1 Expected Frame Status Register (R1EFSR)	Always	Always
E	14	Indicate Attention Register (IAR)	Always	Conditional
E	15	Indicate Notify Register (INR)	Always	Always
C	16	Indicate Threshold Register (ITR)	Always	INSTOP Mode or EXC = 1 Only
C	17	Indicate Mode Register (IMR)	Always	INSTOP Mode Only
C	18	Indicate Configuration Register (ICR)	Always	Always
C	19	Indicate Header Length Register (IHLR)	Always	INSTOP Mode or EXC = 1 Only
	1A–C	Reserved	N/A	N/A
E	1F	Compare Register (CMP)	Always	Always

C = Control Register
E = Event Register

5.0 Control Information (Continued)

TABLE 5-3. Control and Event Registers Following Reset

Address	Register	Reset
00	Mode Register	00
02	Pointer RAM Control and Address Register	NA
03	Mailbox Address Register	*
04	Master Attention Register	00
05	Master Notify Registers	00
06	State Attention Register	07
07	State Notify Register	00
08	Service Attention Register	0F
09	Service Notify Register	00
0A	No Space Attention Register	FF
0B	No Space Notify Register	00
0C	Limit Address Register	NA
0D	Limit Data Register	NA
0E	Request Attention Register	00
0F	Request Notify Register	00
10	Request Channel 0 Configuration Register	NA
11	Request Channel 1 Configuration Register	NA
12	Request Channel 0 Expected Frame Status Register	NA
13	Request Channel 1 Expected Frame Status Register	NA
14	Indicate Attention Register	00
15	Indicate Notify Register	00
16	Indicate Threshold Register	NA
17	Indicate Mode Register	NA
18	Indicate Configuration Register	NA
19	Indicate Header Length Register	NA
1F	Compare Register	NA

* = Initialized to a silicon Revision code upon reset. The Revision code remains until it is overwritten by the host.
 NA = Not altered upon reset.

5.0 Control Information (Continued)

Event Registers

The Event Registers record the occurrence of events or series of events. Events are recorded and contribute to generating the Interrupt signal. There is a two-level hierarchy in generating this signal, as shown in *Figure 5-1*.

At the first level of the hierarchy, events are recorded as bits in the Attention Registers (e.g., No Space Attention Register). Each Attention Register has a corresponding Notify Register (e.g., No Space Notify Register). When a bit in the Attention Register is set to One and its corresponding bit in the Notify Register is also set to One, the corresponding bit in the Master Attention Register will be set to one.

At the second level of the hierarchy, if a bit in the Master Attention Register is set to One and the corresponding bit in the Master Notify Register is set to One, the Interrupt signal is asserted.

Bits in Conditional Write Registers (e.g., No Space Attention Register) are only written when the corresponding bits in the Compare Register are equal to the bits to be overwritten.

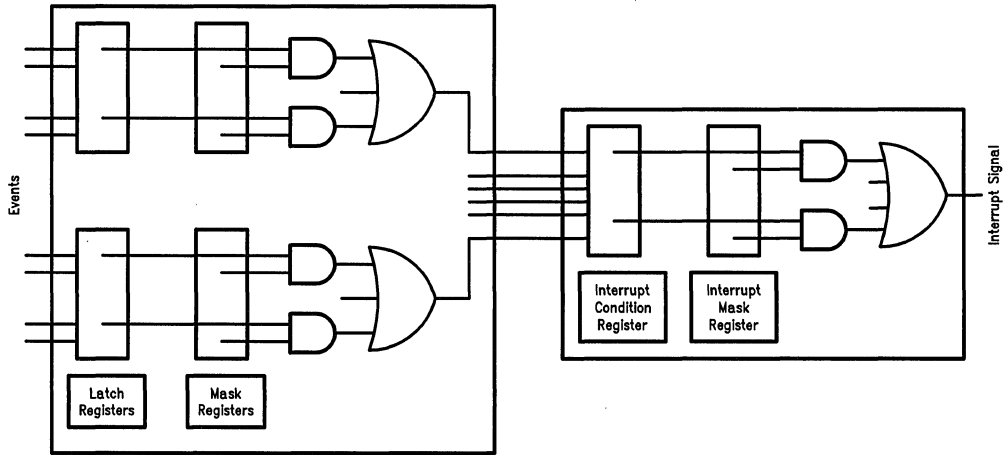


FIGURE 5-1. Event Registers Hierarchy

TL/F/10791-11

Events are recorded in Attention Registers and contribute to the Interrupt when the bit in the corresponding Notify Register is set (see Table 5-2).

The Event Registers include the following registers:

- **Master Attention Register (MAR)** collects enabled attentions from the State Attention Register, Service Attention Register, No Space Attention Register, Request Attention Register, and Indicate Attention Register.
- **Master Notify Register (NMR)** is used to selectively enable attention in the Master Attention Register.
- **State Attention Register (STAR)** presents attentions for major states within the BSI device and various error conditions.
- **State Notify Register (STNR)** is used to enable attentions in the State Attention Register.
- **Service Attention Register (SAR)** presents attentions for the PTOP and LMOP service functions.
- **Service Notify Register (SNR)** is used to enable attentions in the Service Attention Register.
- **No Space Attention Register (NSAR)** presents attentions generated when the IDUD, PSP, or CNF Queues run out of space or valid entries.
- **Request Attention Register (RAR)** presents attentions generated by both Request Channels.
- **Request Notify Register (RNR)** is used to enable attentions in the Request Attention Register.
- **Indicate Attention Register (IAR)** presents the attentions generated by the Indicate Channels.
- **Indicate Notify Register (INR)** is used to enable attentions in the Indicate Attention Register.
- **Compare Register (CMP)** is used for comparison with a write access of a conditional write (Attention) register.

5.0 Control Information (Continued)

5.3 CONTROL AND EVENT REGISTER DESCRIPTIONS

Mode Register (MR)

The Mode Register (MR) is used to program the major operating parameters for the BSI device. This register should be programmed only at power-on, or after a software Master Reset.

This register is cleared upon reset.

Access Rules

Address	Read	Write
00h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
SMLB	SMLQ	VIRT	BIGEND	FLOW	MRST	FABCLK	TEST

Bit	Symbol	Description
D0	TEST	Test Mode: Enables test logic, in which the transmitted frames counter will cause a service loss after four frames, instead of 255 frames.
D1	FABCLK	Fast ABus Clock: Determines the metastability delay period for synchronizing between the ABus clock and the Ring clock (LBC). Upon reset this bit is cleared to Zero, which selects one ABus clock period as the delay. When this bit is set to One, only 1/2 of an ABus clock delay is used. When AB__CLK = LBC, (i.e., at 12.5 MHz and in phase), this bit should be set. For any AB__CLK greater than LBC, this bit must be Zero.
D2	MRST	Master Reset: When this bit is set, the Indicate, Request, and Status/Space Macines are placed in Stop Mode, and BSI device registers are initialized to the values shown in Table 5-5. This bit is cleared after the reset is complete.
D3	FLOW	Flow Parity: When this bit is set, parity flows between the ABus and the BMAC device, that is, incoming data is not checked at the ABus interface, but is checked (by the BMAC device) as it is passed to the BMAC device. The parity check includes the frame's FC through ED fields. When this bit is set, Control Bus parity is also checked, and errors are reported in the CPE bit of the State Attention Register. When this bit is Zero, no parity is checked on the Control Bus or ABus.
D4	BIGEND	Big Endian Data Format: Selects between the Little Endian (BIGEND = 0) or Big Endian (BIGEND = 1) data format. See <i>Figure 4-2</i> .
D5	VIRT	Virtual Address Mode: Selects between virtual (VIRT = 1) or physical (VIRT = 0) address mode on the ABus.
D6	SMLQ	Small Queue: Selects the size of all Descriptor queues and lists. When SMLQ = 0, the size is 4k bytes; when SMLQ = 1, the size is 1k bytes. Note that data pages are always 4k bytes.
D7	SMLB	Small Bursts: Selects size of bursts on ABus. When SMLB = 0, the BSI device uses 1-, 4-, and 8-word transfers. When SMLB = 1, the BSI device uses 1- and 4-word transfers.

5.0 Control Information (Continued)

Pointer RAM Control and Address Register (PCAR)

The Pointer RAM Control and Address Register (PCAR) is used to program the parameters for the PTOp (Pointer RAM Operation) service function, in which data is written to or read from a Pointer RAM Register.

This register is not altered upon reset.

Access Rules

Address	Read	Write
02h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
BP1	BP0	PTRW	A4	A3	A2	A1	A0

Bit	Symbol	Description
D0-4	A0-4	Pointer RAM Address: These five bits contain the Pointer RAM Register address for a subsequent PTOp service function.
D5	PTRW	PTOp Read/Write: This bit determines whether a PTOp service function will be a read from the Pointer RAM Register to the mailbox in memory (PTRW = 1), or a write to the Pointer RAM Register from the mailbox (PTRW = 0).
D6-7	BP0-1	Byte Pointer: These two bits are used to program an internal byte pointer for accesses to the 32-bit Mailbox Address Register. They are normally set to Zero to initialize the byte pointer for four successive writes (most-significant byte first) and are automatically incremented after each write.

5.0 Control Information (Continued)

Mailbox Address Register (MBAR)

The Mailbox Address Register (MBAR) is used to program the word-aligned 28-bit memory address of the mailbox used in the data transfer of the PTOp (Pointer RAM Operation) service function.

The address of this register is used as a window into four internal byte registers. The four byte registers are loaded by successive writes to this address after first setting the BPR bits in the Pointer RAM Control and Address Register to Zero. The bytes must be loaded most-significant byte first. The BSI device increments the byte pointer internally after each write or read. Mailbox Address bits 0 and 1 forced internally to Zero.

This register is initialized to a silicon Revision code upon reset. The Revision code remains until it is overwritten by the host.

Access Rules

Address	Read	Write
03h	Always	Always

Register Bits

7	0
Mailbox Address [27:24]	
Mailbox Address [23:16]	
Mailbox Address [15:8]	
Mailbox Address [7:0]	

5.0 Control Information (Continued)

Master Attention Register (MAR)

The Master Attention Register (MAR) collects enabled attentions from the State Attention Register, Service Attention Register, No Space Attention Register, Request Attention Register, and Indicate Attention Register. If the Notify bit in the Master Notify Register and the corresponding bit in the MAR are set to One, the $\overline{\text{INT}}$ is forced to LOW and thus triggers an interrupt.

Writes to the Master Attention Register are permitted, but do not change the contents.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
04h	Always	Data Ignored

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
STA	NSA	SVA	RQA	INA	RES	RES	RES

Bit	Symbol	Description
D0-2	RES	Reserved
D3	INA	Indicate Attention Register: Is set if any bit in the Indicate Attention Register is set.
D4	RQA	Request Attention Register: Is set if any bit in the Request Attention Register is set.
D5	SVA	Service Attention Register: Is set if any bit in the Service Attention Register is set.
D6	NSA	No Space Attention Register: Is set if any bit in the No Space Attention Register is set.
D7	STA	State Attention Register: Is set if any bit in the State Attention Register is set.

5.0 Control Information (Continued)

Master Notify Register (MNR)

The Master Notify Register (MNR) is used to enable attentions in the Master Attention Register (MAR). If a bit in Register MNR and the corresponding bit in Register MAR are set to One, the $\overline{\text{INT}}$ signal is deasserted and causes an interrupt.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
05h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
STAN	NSAN	SVAN	RQAN	INAN	RES	RES	RES

Bit	Symbol	Description
D0–2	RES	Reserved
D3	INAN	Indicate Attention Register Notify: This bit is used to enable the INA bit in Register MNR.
D4	RQAN	Request Attention Register Notify: This bit is used to enable the RQA bit in Register MNR.
D5	SVAN	Service Attention Register Notify: This bit is used to enable the SVA bit in Register MNR.
D6	NSAN	No Space Attention Register Notify: This bit is used to enable the NSA bit in Register MNR.
D7	STAN	State Attention Register Notify: This bit is used to enable the STA bit in Register MNR.

5.0 Control Information (Continued)

State Attention Register (STAR)

The State Attention Register (STAR) controls the state of the Indicate, Request, and Status/Space Machines. It also records parity, internal logic, and ABus transaction errors. Each bit may be enabled by setting the corresponding bit in the State Notify Register.

Access Rules

Address	Read	Write
06h	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
ERR	BPE	CPE	CWI	CMDE	SPSTOP	RQSTOP	INSTOP

Bit	Symbol	Description
D0	INSTOP	Indicate Stop: This bit is set by the host to place the Indicate Machine in Stop Mode. This bit is set by the BSI device when the Indicate state machine detects an internal error, enters an invalid state, or when the host loads the Indicate Header Length Register with an illegal value. This bit is set upon reset.
D1	RQSTOP	Request Stop: This bit is set by the host to place the Request Machine in Stop Mode. This bit is set by the BSI device when the Request Machine detects an internal error or enters an invalid state. It is also set when an ABus error occurs while storing a Confirmation Status Message Descriptor (CNF). This bit is set upon reset.
D2	SPSTOP	Status/Space Stop: This bit is set by the host to place the Status/Space Machine in Stop Mode. This bit is set by the BSI device when the Status/Space Machine has entered STOP Mode because of an unrecoverable error. In STOP Mode, only PTOP or LMOP service functions will be performed. This bit is set upon reset.
D3	CMDE	Command Error: Indicates that the host performed an invalid operation. This occurs when an invalid value is loaded into the Indicate Header Length Register (which also sets the INSTOP attention). This bit is cleared upon reset.
D4	CWI	Conditional Write Inhibit: Indicates that at least one bit of the previous conditional write operation was not written. This bit is set unconditionally after each write to a conditional write register. It is also set when the value of the Compare Register is not equal to the value of the register that was accessed for a write before it was written. This may indicate that the accessed register has changed since it was last read. This bit is cleared after a successful conditional write. CWI bit does not contribute to setting the STA bit of the Master Attention Register because its associated Notify bit is always 0. This bit is cleared upon reset.
D5	CPE	Control Bus Parity Error: Indicates a parity error detected on CBD7–0. If there is a Control Bus parity error during a host write, the write is suppressed. Control Bus parity errors are reported when flow-through parity is enabled (the FLOW bit of the Mode Register is set). This bit is cleared upon reset.
D6	BPE	BMAC Device Parity Error: Indicates parity error detected on MID7–0. BMAC device parity is always checked during a frame. This bit is cleared upon reset.
D7	ERR	Error: This bit is set by the BSI device when a non-recoverable error occurs. These include an ABus transaction error while writing confirmation status, an internal logic error, or when any state machine enters an invalid state. This bit is cleared upon reset.

5.0 Control Information (Continued)

State Notify Register (STNR)

The State Notify Register (STNR) is used to enable bits in the State Attention Register (STAR). If a bit in Register STNR is set to One, the corresponding bit in Register STAR will be applied to the Master Attention Register, which can be used to generate an interrupt to the host.

All bits in this register are cleared to Zero upon reset.

Access Rules

Address	Read	Write
07h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
ERRN	BPEN	CPEN	CWIN	CMDEN	SPSTOPN	RQSTOPN	INSTOPN

Bit	Symbol	Description
D0	INSTOPN	Indicate Stop Notify: This bit is used to enable the INSTOP bit in Register STAR.
D1	RQSTOPN	Request Stop Notify: This bit is used to enable the RQSTOP bit in Register STAR.
D2	SPSTOPN	Status/Space Stop Notify: This bit is used to enable the SPSTOP bit in Register STAR.
D3	CMDEN	Command Error Notify: This bit is used to enable the CMDE bit in Register STAR.
D4	CWIN	Conditional Write Inhibit Notify: This bit is used to enable the CWI bit in Register STAR.
D5	CPEN	Control Bus Parity Error Notify: This bit is used to enable the CPE bit in Register STAR.
D6	BPEN	BMAC Device Parity Error Notify: This bit is used to enable the BPE bit in Register STAR.
D7	ERRN	Error Notify: This bit is used to enable the ERR bit in Register STAR.

5.0 Control Information (Continued)

Service Attention Register (SAR)

The Service Attention Register (SAR) is used to present the attentions for the service functions. Each bit may be enabled by setting the corresponding bit in the State Notify Register.

Access Rules

Address	Read	Write
08h	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	ABR0	ABR1	LMOP	PTOP

Bit	Symbol	Description
D0	PTOP	Pointer RAM Operation: This bit is cleared by the host to cause the BSI device to transfer data between a Pointer RAM Register and a mailbox location in memory. The Pointer RAM Control and Address Register contains the Pointer RAM Register address and determines the direction of the transfer (read or write). The memory address is in the Mailbox Address Register. This bit is set by the BSI device after it performs the data transfer. While PTOPI = 0, the host must not alter the Pointer RAM Address and Control Register or the Mailbox Address Register.
D1	LMOP	Limit RAM Operation: This bit is cleared by the host to cause the BSI device to transfer data between a Limit RAM Register and the Limit Data and Limit Address Registers. The Limit Address Register contains the Pointer RAM Register address and determines the direction of the transfer (read and write). This bit is set by the BSI device after it data performs the transfer. While LMOP = 0, the host must not alter either the Limit Address or Limit Data Registers.
D2	ABR1	Abort Request RCHN1: This bit is cleared by the host to abort a Request on RCHN1. This bit is set by the BSI device when RQABORT ends a request on RCHN1. The host may write a 1 to this bit, which may or may not prevent the request from being aborted. When this bit is cleared by the host, the USR1 bit in the Request Attention Register is set and further processing on RCHN1 is halted.
D3	ABR0	Abort Request RCHN0: This bit is cleared by the host to abort a Request on RCHN0. This bit is set by the BSI device when RQABORT ends a request on RCHN0. The host may write a 1 to this bit, which may or may not prevent the request from being aborted. When this bit is cleared by the host, the USR0 bit in the Request Attention Register is set and further processing on RCHN0 is halted.
D4-7	RES	Reserved

5.0 Control Information (Continued)

Service Notify Register (SNR)

The Service Notify Register (SNR) is used to enable attentions in the Service Attention Register (SAR). If a bit in Register SNR is set to One, the corresponding bit in Register SAR will be applied to the Master Attention Register, which can be used to generate an interrupt to the host.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
09h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	ABR0N	ABR1N	LMOPN	PTOPN

Bit	Symbol	Description
D0	PTOPN	Pointer RAM Operation Notify: This bit is used to enable the PTOP bit in Register SAR.
D1	LMOPN	Limit RAM Operation Notify: This bit is used to enable the LMOP bit in Register SAR.
D2	ABR1N	Abort Request RCHN1 Notify: This bit is used to enable the ABR1 bit in Register SAR.
D3	ABR0N	Abort Request RCHN0 Notify: This bit is used to enable the ABR0 bit in Register SAR.
D4–7	RES	Reserved

5.0 Control Information (Continued)

No Space Attention Register (NSAR)

The No Space Attention Register (NSAR) presents the attentions generated when the CNF, PSP, or IDUD Queues run out of space. The host may set any attention bit to cause an attention for test purposes only, though this should not be done during normal operation.

The No Data Space attentions are set and cleared by the BSI device automatically. The No Status Space attentions are set by the BSI device, and must be cleared by the host.

Upon reset this register is set to 0xffh.

Access Rules

Address	Read	Write
0Ah	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
NSR0	NSR1	LDI0	NSI0	LDI1	NSI1	LDI2	NSI2

Bit	Symbol	Description
D0	NSI2	No Status Space on ICHN2: This bit is set by the BSI device upon a Reset, or when an IDUD has been written to the next-to-last available entry in the Indicate Channel's IDUD Status Queue. When this occurs, the BSI device stops copying on ICHN2 and the last IDUD is written with special status. This bit (as well as the USR Attention bit) must be cleared by the host before the BSI device will resume copying on this Channel.
D1	LDI2	Low Data Space on ICHN2: This bit is set by the BSI device upon a Reset, or when a PSP is prefetched from ICHN2's last PSP Queue location (as defined by the PSP Queue Limit Register). Note that the amount of warning is dependent on the length of the frame. There will always be one more page (4k bytes) available for the BSI device when this attention is generated. Another FDDI maximum-length frame (after the current one) will not fit in this space. If SPS fetching was stopped because there were no more PSP entries, fetching will resume automatically when the PSP Queue Limit Register is updated.
D2	NSI1	No Status Space on ICHN1: This bit is set by the BSI device upon a Reset, or when an IDUD has been written to the next-to-last available entry in the Indicate Channel's IDUD Status Queue. When this occurs, the BSI device stops copying on ICHN1 and the last IDUD is written with special status. This bit (as well as the USR Attention bit) must be cleared by the host before the BSI device will resume copying on this Channel.
D3	LDI1	Low Data Space on ICHN1: This bit is set by the BSI device upon a Reset, or when a PSP is prefetched from ICHN1's last PSP Queue location (as defined by the PSP Queue Limit Register). Note that the amount of warning is dependent on the length of the frame. There will always be one more page (4k bytes) available for the BSI device when this attention is generated. Another FDDI maximum-length frame (after the current one) will not fit in this space. If PSP fetching was stopped because there were no more PSP entries, fetching will resume automatically when the PSP Queue Limit Register is updated.
D4	NSI0	No Status Space on ICHN0: This bit is set by the BSI device upon a Reset, or when an IDUD has been written to the next-to-last available entry in the Indicate Channel's IDUD Status Queue. When this occurs, the BSI device stops copying on ICHN0 and the last IDUD is written with special status. This bit (as well as the USR Attention bit) must be cleared by the host before the BSI device will resume copying on this Channel.
D5	LDI0	Low Data Space on ICHN0: This bit is set by the BSI device upon a Reset, or when a PSP is prefetched from ICHN0's last PSP Queue location (as defined by the PSP Queue Limit Register). Note that the amount of warning is dependent on the length of the frame. There will always be one more page (4k bytes) available for the BSI device when this attention is generated. Another FDDI maximum-length frame (after the current one) will not fit in this space. If PSP fetching was stopped because there were no more PSP entries, fetching will resume automatically when the PSP Queue Limit Register is updated.
D6	NSR1	No Status Space on RCHN1: This bit is set by the BSI device upon a Reset, or when it has written a CNF Descriptor to the next-to-last Queue location. Due to internal pipelining, the BSI device may write up to two more CNFs to the Queue after this attention is generated. Thus the Host must set the CNF Queue Limit Register to be one less than the available space in the Queue. This bit (as well as the USR attention bit) must be cleared by the Host before the BSI device will continue to process requests on RCHN1.
D7	NSR0	No Status Space on RCHN0: This bit is set by the BSI device upon a Reset, or when it has written a CNF Descriptor to the next-to-last Queue location. Due to internal pipelining, the BSI device may write up to two more CNFs to the Queue after this attention is generated. Thus the Host must set the CNF Queue Limit Register to be one less than the available space in the Queue. This bit (as well as the USR attention bit) must be cleared by the Host before the BSI device will continue to process requests on RCHN0.

5.0 Control Information (Continued)

No Space Notify Register (NSNR)

The No Space Notify Register (NSNR) is used to enable attentions in the No Space Attention Register (NSAR). If a bit in Register NSNR is set to One, the corresponding bit in Register NSAR will be applied to the Master Attention Register, which can be used to generate an interrupt to the host.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
0Bh	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
NSR0N	NSR1N	LDI0N	NSI0N	LDI1N	NSI1N	LDI2N	NSI2N

Bit	Symbol	Description
D0	NSI2N	No Status Space on ICHN2 Notify: This bit is used to enable the NSI2 in Register NSAR.
D1	LDI2N	Low Data Space on ICHN2 Notify: This bit is used to enable the LDI2 in Register NSAR.
D2	NSI1N	No Status Space on ICHN1 Notify: This bit is used to enable the NSI1 in Register NSAR.
D3	LDI1N	Low Data Space on ICHN1 Notify: This bit is used to enable the LDI1 in Register NSAR.
D4	NSI0N	No Status Space on ICHN0 Notify: This bit is used to enable the NSI0 in Register NSAR.
D5	LDI0N	Low Data Space on ICHN0 Notify: This bit is used to enable the LDI0 in Register NSAR.
D6	NSR1N	No Status Space on RCHN1 Notify: This bit is used to enable the NSR1 in Register NSAR.
D7	NSR0N	No Status Space on RCHN0 Notify: This bit is used to enable the NSR0 in Register NSAR.

5.0 Control Information (Continued)

Limit Address Register (LAR)

The Limit Address Register (LAR) is used to program the parameters for a LMOP (Limit RAM Operation) service function. This register is not altered upon reset.

Access Rules

Address	Read	Write
0Ch	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
LRA3	LRA2	LRA1	LRA0	LMRW	RES	RES	LRD8

Bit	Symbol	Description
D0	LRD8	Limit RAM Data Bit 8: This bit contains the most-significant data bit read or written from the addressed Limit RAM Register.
D1–2	RES	Reserved
D3	LMRW	LMOP Read/Write: This bit determines whether a LMOP service function will be a read (LMRW = 1) or write (LMRW = 0).
D4–7	LRA0–3	Limit RAM Register Address: Used to program the Limit RAM Register address for a subsequent LMOP service function.

5.0 Control Information (Continued)

Limit Data Register (LDR)

The Limit Data Register (LDR) is used to contain the 8 least-significant Limit RAM data bits transferred in a LMOP service function. (The most-significant data bit is in the Limit Address register.)

This register is not altered upon reset.

Access Rules

Address	Read	Write
0Dh	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
LRD7	LRD6	LRD5	LRD4	LRD3	LRD2	LRD1	LRD0

Bit	Symbol	Description
D0–7	LRD0–7	Limit RAM Data Bits 0–7: These bits contain the least-significant data bits read from or written to a Limit RAM Register in a LMOP service function.

5.0 Control Information (Continued)

Request Attention Register (RAR)

The Request Attention Register (RAR) is used to present exception, breakpoint, request complete, and unserviceable request attentions generated by each Request Channel. Each bit may be enabled by setting the corresponding bit in the Request Notify Register.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
0Eh	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
USRR0	RCMR0	EXCR0	BRKR0	USRR1	RCMR1	EXCR1	BRKR1

Bit	Symbol	Description
D0	BRKR1	Breakpoint on RCHN1: Is set by the BSI device when a CNF Descriptor is written on RCHN1. No action is taken by the BSI device if the host sets this bit.
D1	EXCR1	Exception on RCHN1: Is set by the BSI device when an exception occurs on RCHN1. No action is taken by the BSI device if the host sets this bit.
D2	RCMR1	Request Complete on RCHN1: Is set by the BSI device when it has completed processing a Request object on RCHN1, an error occurs, or a completion exception occurs. No action is taken if the Host sets this bit.
D3	USRR1	Unserviceable Request on RCHN1: Is set by the BSI device when a Request cannot be processed on RCHN1. This occurs when the Request Class is inappropriate for the current ring state, or when there is no CNF status space, or when the host aborts a request by clearing the ABR bit in the Service Attention Register. While this bit is set, no requests will be processed on RCHN1. The host must clear this bit to resume request processing.
D4	BRKR0	Breakpoint on RCHN0: Is set by the BSI device when a CNF Descriptor is written on RCHN0. No action is taken by the BSI device if the host sets this bit.
D5	EXCR0	Exception on RCHN0: Is set by the BSI device when an exception occurs on RCHN0. No action is taken by the BSI device if the host sets this bit.
D6	RCMR0	Request Complete on RCHN0: Is set by the BSI device when it has completed processing a Request object on RCHN0, an error occurs, or a completion exception occurs. No action is taken if the Host sets this bit.
D7	USRR0	Unserviceable Request on RCHN0: Is set by the BSI device when a Request cannot be processed on RCHN0. This occurs when the Request Class is inappropriate for the current ring state, or when there is no CNF status space, or when the host aborts a request by clearing the ABR bit in the Service Attention Register. While this bit is set, no requests will be processed on RCHN0. The host must clear this bit to resume request processing.

5.0 Control Information (Continued)

Request Notify Register (RNR)

The Request Notify Register (RNR) is used to enable attentions in the Request Attention Register (RAR). If a bit in Register RNR is set to One, the corresponding bit in Register RAR will be applied to the Master Attention Register, which can be used to generate an interrupt to the host.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
0Fh	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
USRR0N	RCMR0N	EXCR0N	BRKR0N	USRR1N	RCMR1N	EXCR1N	BRKR1N

Bit	Symbol	Description
D0	BRKR1N	Breakpoint on RCHN1 Notify: This bit is used to enable the BRKR1 bit in Register RAR.
D1	EXCR1N	Exception on RCHN1 Notify: This bit is used to enable the EXCR1 bit in Register RAR.
D2	RCMR1N	Request Complete on RCHN1 Notify: This bit is used to enable the RCMR1 bit in Register RAR.
D3	USRR1N	Unserviceable Request on RCHN1 Notify: This bit is used to enable the USRR1 bit in Register RAR.
D4	BRKR0N	Breakpoint on RCHN0 Notify: This bit is used to enable the BRKR0 bit in Register RAR.
D5	EXCR0N	Exception on RCHN0 Notify: This bit is used to enable the EXCR0 bit in Register RAR.
D6	RCMR0N	Request Complete on RCHN0 Notify: This bit is used to enable the RCMR0 bit in Register RAR.
D7	USRR0N	Unserviceable Request on RCHN0 Notify: This bit is used to enable the USRR0 bit in Register RAR.

5.0 Control Information (Continued)

Request Channel 0 and 1 Configuration Registers (R0CR and R1CR)

The two Request Configuration Registers (R0CR and R1CR) are programmed with the operational parameters for each of the Request Channels. These registers may only be altered between Requests, i.e., while the particular Request Channel does not have a Request loaded.

These registers are not altered upon reset.

Access Rules

Address	Read	Write
10-11h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
TT1	TT0	PRE	HLD	FCT	SAT	VST	FCS

Bit	Symbol	Description
D0	FCS	Frame Check Sequence Disable: When this bit is set, the BSI device asserts the FCST signal throughout the request. This may drive the BMAC device FCST pin, or also the SAT or SAIGT pins, depending on the application. This bit is normally used to program the BMAC device not to concatenate its generated FCS to the transmitted frame. The Valid FCS bit in the Expected Frame Status Register independently determines whether a frame needs a valid FCS to meet the matching frame criteria.
D1	VST	Void Stripping: When this bit is set, the BSI device asserts the STRIP output signal out throughout the request. This may drive the BMAC device STRIP (Void Strip) pin, or also the SAT pin, depending on the application.
D2	SAT	Source Address Transparency: When this bit is set, the BSI device asserts the SAT output signal throughout the request. This may drive the BMAC device's SAT and/or SAIGT pins, depending on the application. When SAT is set, Full Confirmation requires the use of the EM (External SA Match) signal.
D3	FCT	Frame Control Transparency: When this bit is set, the FC will be sourced from the ODU (not the REQ.First Descriptor). When Full Confirmation is enabled and FCT = 0, all bits of the FC in returning frames must match the FC field in the REQ Descriptor; if FCT = 1, only the C, L and r bits must match. Note that since the BSI device decodes the REQ.F Descriptor FC field to determine whether to assert RQCLM/RQBCN, FC transparency may be used to send Beacons or Claims in any ring non-operational state, as long as the FC in the REQ Descriptor is not set to Beacon or Claim. By programming a Beacon or Claim FC in the REQ Descriptor, then using FC transparency, any type of frame may be transmitted in the Beacon or Claim state.
D4	HLD	Hold: When this bit is set, the BSI device will not end a service opportunity until the Request is complete. When this bit is Zero, the BSI device ends the service opportunity on the Request Channel when all of the following conditions are met: <ol style="list-style-type: none"> 1. There is no valid request active on the Request Channel. 2. The service class is non-immediate. 3. There is no data in the FIFO. 4. There is no valid REQ fetched by the BSI device. This bit also affects Prestaging on RCHN1 (Request Channel 1). When HLD = 0, prestaging is enabled on RCHN1, regardless of the state of the PRE bit (except for Immediate service classes). When HLD = 1, prestaging is determined by the PRE bit. This option can potentially waste ring bandwidth, but may be required (particularly on RCHN0, Request Channel 0) if a small guaranteed service time is required. When using the Repeat option, HLD is required for small frames. If HLD is not used, the other Request Channel will be checked for service before releasing the token between frames. This may not be the desired action, particularly if there is a request on RCHN1 that needs servicing after the completion of RCHN0's Repeated Request.

5.0 Control Information (Continued)

Request Channel 0 and 1 Configuration Registers (R0CR and R1CR) (Continued)

Symbol	Bit	Description															
D5	PRE	<p>Preempt/Prestage: When this bit is set, preemption is enabled for RCHN0, and prestaging is enabled for RCHN1 (prestaging is always enabled for RCHN0). When this bit Zero, preemption is disabled and prestaging is enabled only on RCHN0.</p> <p>When preemption is enabled, RCHN0 preempts a (non-committed) frame of RCHN1 already in the FIFO, causing it to be purged and refetched after RCHN0's request has been serviced. When the Request Machine is servicing a request on RCHN1 and a request on RCHN0 becomes active, if preemption is enabled on RCHN0, the Request Machine will finish transmitting the current frame on RCHN1, then release the token and move back to the start state. This has the effect of reprioritizing the Request Channels, thus ensuring that frames on RCHN0 are transmitted at the next service opportunity. When RCHN0 has been serviced, transmission will continue on RCHN1 with no loss of data.</p> <p>When prestaging is enabled, the next frame for RCHN1 is staged (ODUs are loaded into the FIFO before the token arrives). If prestaging is not enabled, the Request Machine waits until the token is captured before staging the first frame. Once the token is captured, the Request Machine begins fetching data, and when the FIFO threshold has been reached, transmits the data on that Request Channel. For requests with an Immediate service class, prestaging is not applicable.</p> <p>When this bit is Zero, preemption is disabled for RCHN0, and requests on RCHN1 will not be prestaged unless the HLD bit is Zero, in which case RCHN1 will prestage data regardless of the setting of the PRE bit.</p> <p>Note that when prestaging is not enabled on RCHN1, data is not staged until the token is captured. Since there is no data in the FIFO (if there is no active request on RCHN0), the BSI device will immediately release the token if the HLD option is not set.</p>															
D6-7	TT0-1	<p>Transmit Threshold: Determines the threshold on the output data FIFO before the BSI device requests transmission.</p> <table border="1"> <thead> <tr> <th>TT1</th> <th>TT0</th> <th>Threshold Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>8 Words</td> </tr> <tr> <td>0</td> <td>1</td> <td>16 Words</td> </tr> <tr> <td>1</td> <td>0</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>Reserved</td> </tr> </tbody> </table>	TT1	TT0	Threshold Value	0	0	8 Words	0	1	16 Words	1	0	Reserved	1	1	Reserved
TT1	TT0	Threshold Value															
0	0	8 Words															
0	1	16 Words															
1	0	Reserved															
1	1	Reserved															

5.0 Control Information (Continued)

Request Channel 0 and 1 Expected Frame Status Registers (R0EFSR and R1EFSR)

The Expected Frame Status Registers (R0EFSR and R1EFSR) define the matching criteria used for Full Confirmation of returning frames on each Request Channel. A returning frame must meet the programmed criteria to be counted as a matching frame.

These registers are not altered upon reset.

Access Rules

Address	Read	Write
12-13h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
VDL	VFCS	EE1	EE0	EA1	EA0	EC1	EC0

Bit	Symbol	Description															
D0-1	EC0-1	Expected C Indicator: <table border="1"> <thead> <tr> <th>EC1</th> <th>EC0</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Any</td> </tr> <tr> <td>0</td> <td>1</td> <td>R</td> </tr> <tr> <td>1</td> <td>0</td> <td>S</td> </tr> <tr> <td>1</td> <td>1</td> <td>R or S</td> </tr> </tbody> </table>	EC1	EC0	Value	0	0	Any	0	1	R	1	0	S	1	1	R or S
EC1	EC0	Value															
0	0	Any															
0	1	R															
1	0	S															
1	1	R or S															
D2-3	EA0-1	Expected A Indicator: <table border="1"> <thead> <tr> <th>EA1</th> <th>EA0</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Any</td> </tr> <tr> <td>0</td> <td>1</td> <td>R</td> </tr> <tr> <td>1</td> <td>0</td> <td>S</td> </tr> <tr> <td>1</td> <td>1</td> <td>R or S</td> </tr> </tbody> </table>	EA1	EA0	Value	0	0	Any	0	1	R	1	0	S	1	1	R or S
EA1	EA0	Value															
0	0	Any															
0	1	R															
1	0	S															
1	1	R or S															
D4-5	EE0-1	Expected E Indicator: <table border="1"> <thead> <tr> <th>EE1</th> <th>EE0</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Any</td> </tr> <tr> <td>0</td> <td>1</td> <td>R</td> </tr> <tr> <td>1</td> <td>0</td> <td>S</td> </tr> <tr> <td>1</td> <td>1</td> <td>R or S</td> </tr> </tbody> </table>	EE1	EE0	Value	0	0	Any	0	1	R	1	0	S	1	1	R or S
EE1	EE0	Value															
0	0	Any															
0	1	R															
1	0	S															
1	1	R or S															
D6	VFCS	Valid FCS: When this bit is set, returning frames must have a valid FCS field to meet the confirmation criteria.															
D7	VDL	Valid Data Length: When this bit is set, returning frames must have a valid VDL field to meet the confirmation criteria.															

5.0 Control Information (Continued)

Indicate Attention Register (IAR)

The Indicate Attention Register (IAR) is used to present exception and breakpoint attentions generated by each Indicate Channel. An Attention bit is set by hardware when an exception or breakpoint occurs on the corresponding Indicate Channel. Each bit may be enabled by setting the corresponding bit in the Indicate Notify Register.

Access Rules

Address	Read	Write
14h	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	EXCIO	BRKI0	EXCI1	BRKI1	EXCI2	BRKI2

Bit	Symbol	Description
D0	BRKI2	Breakpoint on ICHN2: Is set when a breakpoint is detected on Indicate Channel 2. No action is taken if the host sets this bit.
D1	EXCI2	Exception on ICHN2: Is set by the BSI device when an exception occurs on Indicate Channel 2. May be set by the host to disable copying on ICHN2, which is convenient when updating the Indicate Header Length and Indicate Threshold registers. While this bit is set, copying is disabled on ICHN2.
D2	BRKI1	Breakpoint on ICHN1: Is set when a breakpoint is detected on Indicate Channel 1. No action is taken if the host sets this bit.
D3	EXCI1	Exception on ICHN1: Is set by the BSI device when an exception occurs on Indicate Channel 1. May be set by the host to disable copying on ICHN1, which is convenient when updating the Indicate Header Length and Indicate Threshold registers. While this bit is set, copying is disabled on ICHN1.
D4	BRKI0	Breakpoint on ICHN0: Is set when a breakpoint is detected on ICHN0. No action is taken if the host sets this bit.
D5	EXCIO	Exception on ICHN0: Is set by the BSI device when an exception occurs on Indicate Channel 0. May be set by the host to disable copying on ICHN0, which is convenient when updating the Indicate Header Length and Indicate Threshold registers. While this bit is set, copying is disabled on ICHN0.
D6–7	RES	Reserved

5.0 Control Information (Continued)

Indicate Notify Register (INR)

The Indicate Notify Register (INR) is used to enable attentions in the Indicate Attention Register (IAR). If a bit in Register INR is set to One, the corresponding bit in Register IAR will be applied to the Master Attention Register, which can be used to generate an interrupt to the host.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
15h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	EXC0N	BRK0N	EXC1N	BRK1N	EXC2N	BRK2N

Bit	Symbol	Description
D0	BRK2N	Breakpoint on ICHN2 Notify: This bit is used to enable the BRK2 bit in Register IAR.
D1	EXC2N	Exception on ICHN2 Notify: This bit is used to enable the EXC2 bit in Register IAR.
D2	BRK1N	Breakpoint on ICHN1 Notify: This bit is used to enable the BRK1 bit in Register IAR.
D3	EXC1N	Exception on ICHN1 Notify: This bit is used to enable the EXC1 bit in Register IAR.
D4	BRK0N	Breakpoint on ICHN0 Notify: This bit is used to enable the BRK0 bit in Register IAR.
D5	EXC0N	Exception on ICHN0 Notify: This bit is used to enable the EXC0 bit in Register IAR.
D6-7	RES	Reserved

5.0 Control Information (Continued)

Indicate Threshold Register (ITR)

The Indicate Threshold Register (ITR) specifies the maximum number of frames that can be received on Indicate Channel 1 or Indicate Channel 2 before an attention will be generated. This register may be written only when the INSTOP bit in the State Attention Register is set, or when the Indicate Channel's corresponding EXC bit in the Indicate Attention Register is set.

This register is not altered upon reset.

Access Rules

Address	Read	Write
16h	Always	INSTOP Mode or EXC = 1 Only

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
THR7	THR6	THR5	THR4	THR3	THR2	THR1	THR0

Bit	Symbol	Description
D0-7	THR0-7	Threshold Data Bits 0-7: The value programmed in this register is loaded into an internal counter every time the Indicate Channel changes. Each valid frame copied on the current Channel decrements the counter. When the counter reaches Zero, a status breakpoint attention is generated (i.e., the Channel's BRK bit in the Indicate Attention Register is set) if the Channel's Breakpoint on Threshold (BOT) bit in the Indicate Mode Register is set. Loading the Indicate Threshold Register with Zero generates a breakpoint after 256 consecutive frames are received on any one Indicate Channel.

5.0 Control Information (Continued)

Indicate Mode Register (IMR)

The Indicate Mode Register (IMR) defines configuration options for all three Indicate Channels, including the sort mode, frame filtering, and status breakpoints.

This register may be written only when the INSTOP bit in the State Attention Register is set. It may be written with its current value any time, which is useful for one-shot sampling.

This register is not altered upon reset.

Access Rules

Address	Read	Write
17h	Always	INSTOP Mode Only

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
SM1	SM0	SKIP	RES	BOT1	BOT2	BOB	BOS

Bit	Symbol	Description
D0	BOS	Breakpoint on Service Opportunity: Enables the end of a service opportunity to generate an Indicate breakpoint attention (i.e., set the Channel's BRK bit in the Indicate Attention Register). Service opportunities include receipt of a Token, a MAC Frame, or a ring operational change following some copied frames.
D1	BOB	Breakpoint on Burst: Enables the end of a burst to generate an Indicate breakpoint attention (i.e., set the Channel's BRK bit in the Indicate Attention Register). End of burst includes Channel change, DA change, SA change, or MAC INFO change. A Channel change is detected from the FC field of valid, copied frames. A DA change is detected when a frame's DA field changes from our address to any other. A SA change is detected when a frame's SA field is not the same as the previous one. A MAC INFO breakpoint occurs when a MAC frame does not have the identical first four bytes of INFO as the previous frame. This breakpoint always sets the BRK bit (i.e., this breakpoint is always enabled).
D2	BOT2	Breakpoint on Threshold for ICHN2: Enables the value in the Indicate Threshold Register to be used to generate an Indicate breakpoint attention on Indicate Channel 2, (i.e., set the BRK2 bit in the Indicate Attention Register).
D3	BOT1	Breakpoint on Threshold for ICHN1: Enables the value in the Indicate Threshold Register to be used to generate an Indicate breakpoint attention on ICHN1, (i.e., set the BRK1 bit in the Indicate Attention Register).
D4	RES	Reserved
D5	SKIP	Skip Enable: Enables filtering on Indicate Channel 0 when the Copy Control field for ICHN0 in the Indicate Configuration Register is set to 01 or 10. When this bit is set, only the unique MAC and SMT frames received on Indicate Channel 0 will be copied to memory, i.e., those having an FC field or first four bytes of the Information field that differs from the previous frame. A write to the Indicate Mode Register disables filtering.

5.0 Control Information (Continued)

Indicate Mode Register (IMR) (Continued)

Bit	Symbol	Description																				
SM0-1	D6-7	<p>Sort Mode: These bits determine how the BSI device sorts Indicate data onto Indicate Channels 1 and 2. (Indicate Channel 0 always receives SMT and MAC frames.)</p> <table border="1" data-bbox="283 269 689 390"> <thead> <tr> <th>SM1</th> <th>SM0</th> <th>ICHN2</th> <th>ICHN1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Asynchronous</td> <td>Synchronous</td> </tr> <tr> <td>0</td> <td>1</td> <td>External</td> <td>Internal</td> </tr> <tr> <td>1</td> <td>0</td> <td>Info</td> <td>Header</td> </tr> <tr> <td>1</td> <td>1</td> <td>Low Priority</td> <td>High Priority</td> </tr> </tbody> </table> <p>The Synchronous/Asynchronous Sort Mode is intended for use in end-stations or applications using synchronous transmission.</p> <p>The Internal/External Sorting Mode is intended for bridging or monitoring applications. MAC/SMT frames matching the internal (BMAC device) address are sorted onto ICHN0, and all other frames matching the BMAC device's internal address (short or long) are sorted onto ICHN1. All frames matching the external address (frames requiring bridging) are sorted onto ICHN2 (including MAC/SMT). This sorting mode utilizes the EM, EA, and ECIP input signals with external address matching circuitry. External address circuitry must assert ECIP sometime from the assertion of FCRCVD up to the clock before the assertion of INFORCVD. Otherwise, the BSI device assumes no external address comparison is taking place. ECIP must be negated before EDRCVD; if not, the frame is not copied. EA and EM are sampled on the clock after ECIP is negated. ECIP is ignored after it is negated, until FCRCVD is asserted again. To confirm transmitted frames in this mode (typically using SAT), EM must be asserted within the same time frame as EA. Note that internal matches have precedence over external matches.</p> <p>The Header/Info Sort Mode is intended for high performance protocol processing. MAC/SMT frames are sorted onto ICHN0, while all other frames are sorted onto ICHN1 and ICHN2. Frame bytes from the FC up to the programmed header length are copied onto ICHN1. The remaining bytes (info) are copied onto ICHN2. Only one stream of IDUDs is produced (on ICHN1), but both Indicate Channel's PSP queues are used for space (i.e., PSPs from ICHN1 for header space, and PSPs from ICHN2 for info space). Frames may comprise a header only, or a header + info. For frames with info, multi-part IDUD objects are produced. For multi-part IDUDs, the Indicate Status field in the IDUD is used to determine which part of the IDUD object points to the end of the header. The remainder of the IDUD object points to the Info. For example, if page crosses occur while writing the header and while writing out the Info, the BSI Device will generate a four part IDUD object (IDUD.First, IDUD.Middle, IDUD.Middle, IDUD.Last). The IDUD.First will have a status of "page cross". The first IDUD.Middle will have a status of "end of header". The next IDUD.Middle will have a status of "page boundary crossed". The IDUD.Last will have an "end of frame" status.</p> <p>The High Priority/Low Priority Sort Mode is intended for end stations using two priority levels of asynchronous transmission. The priority is determined by the most-significant z-bit of the FC (zzz = 0xx = low-priority; zzz = 1xx = high-priority).</p>	SM1	SM0	ICHN2	ICHN1	0	0	Asynchronous	Synchronous	0	1	External	Internal	1	0	Info	Header	1	1	Low Priority	High Priority
SM1	SM0	ICHN2	ICHN1																			
0	0	Asynchronous	Synchronous																			
0	1	External	Internal																			
1	0	Info	Header																			
1	1	Low Priority	High Priority																			

5.0 Control Information (Continued)

Indicate Configuration Register (ICR)

The Indicate Configuration Register (ICR) is used to program the copy criteria for each of the Indicate Channels. This register is not altered upon reset.

Access Rules

Address	Read	Write
18h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
CC0		RES	CC1		RES		CC2

Bit	Symbol	Description																	
D0-1	CC2	Copy Control ICHN2: <table border="1"> <thead> <tr> <th>CC1</th> <th>CC0</th> <th>Copy Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Do Not Copy</td> </tr> <tr> <td>0</td> <td>1</td> <td>Copy if (AFLAG (~ ECIP & EA)) & ~MFLAG</td> </tr> <tr> <td>1</td> <td>0</td> <td>Copy if (AFLAG (~ ECIP & EA)) MFLAG</td> </tr> <tr> <td>1</td> <td>1</td> <td>Copy Promiscuously.</td> </tr> </tbody> </table>			CC1	CC0	Copy Mode	0	0	Do Not Copy	0	1	Copy if (AFLAG (~ ECIP & EA)) & ~MFLAG	1	0	Copy if (AFLAG (~ ECIP & EA)) MFLAG	1	1	Copy Promiscuously.
CC1	CC0	Copy Mode																	
0	0	Do Not Copy																	
0	1	Copy if (AFLAG (~ ECIP & EA)) & ~MFLAG																	
1	0	Copy if (AFLAG (~ ECIP & EA)) MFLAG																	
1	1	Copy Promiscuously.																	
D2	RES	Reserved																	
D3-4	CC1	Copy Control ICHN1: <table border="1"> <thead> <tr> <th>CC4</th> <th>CC3</th> <th>Copy Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Do Not Copy</td> </tr> <tr> <td>0</td> <td>1</td> <td>Copy if (AFLAG (~ ECIP & EA)) & ~MFLAG</td> </tr> <tr> <td>1</td> <td>0</td> <td>Copy if (AFLAG (~ ECIP & EA)) MFLAG</td> </tr> <tr> <td>1</td> <td>1</td> <td>Copy Promiscuously.</td> </tr> </tbody> </table>			CC4	CC3	Copy Mode	0	0	Do Not Copy	0	1	Copy if (AFLAG (~ ECIP & EA)) & ~MFLAG	1	0	Copy if (AFLAG (~ ECIP & EA)) MFLAG	1	1	Copy Promiscuously.
CC4	CC3	Copy Mode																	
0	0	Do Not Copy																	
0	1	Copy if (AFLAG (~ ECIP & EA)) & ~MFLAG																	
1	0	Copy if (AFLAG (~ ECIP & EA)) MFLAG																	
1	1	Copy Promiscuously.																	
D5	RES	Reserved																	
D6-7	CC0	Copy Control ICHN0: <table border="1"> <thead> <tr> <th>CC7</th> <th>CC6</th> <th>Copy Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Do Not Copy</td> </tr> <tr> <td>0</td> <td>1</td> <td>Copy if (AFLAG (~ ECIP & EA)) & ~MFLAG</td> </tr> <tr> <td>1</td> <td>0</td> <td>Copy if (AFLAG (~ ECIP & EA)) MFLAG</td> </tr> <tr> <td>1</td> <td>1</td> <td>Copy Promiscuously.</td> </tr> </tbody> </table>			CC7	CC6	Copy Mode	0	0	Do Not Copy	0	1	Copy if (AFLAG (~ ECIP & EA)) & ~MFLAG	1	0	Copy if (AFLAG (~ ECIP & EA)) MFLAG	1	1	Copy Promiscuously.
CC7	CC6	Copy Mode																	
0	0	Do Not Copy																	
0	1	Copy if (AFLAG (~ ECIP & EA)) & ~MFLAG																	
1	0	Copy if (AFLAG (~ ECIP & EA)) MFLAG																	
1	1	Copy Promiscuously.																	

5.0 Control Information (Continued)

Indicate Header Length Register (IHLR)

The Indicate Header Length Register (IHLR) defines the length (in words) of the frame header, for use with the Header/Info Sort Mode.

The Indicate Header Length Register must be initialized before setting the Sort Mode to Header/Info. This register may be changed while the INSTOP bit in the State Attention Register or the EXC bit in the Indicate Attention Register is set.

This register is not altered upon reset.

Access Rules

Address	Read	Write
19h	Always	INSTOP Mode or EXC = 1 Only

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
HL7	HL6	HL5	HL4	HL3	HL2	HL1	HL0

Bit	Symbol	Description
D0–7	HL0–7	Header Length: Specifies the length (in words) of the frame header, for use with the Header/Info Sort Mode. The frame FC is written as a separate word, and thus counts as one word. For example, to split after four bytes of header data in a frame with long addresses, this register is programmed with the value 05 (1 word FC, 1.5 DA, 1.5 SA, 1 HDR_DATA). IHLR must not be loaded with a value less than 4. If it is, the BSI device sets the Command Error (CMDE) and Indicate Stop (INSTOP) attentions.

5.0 Control Information (Continued)

Compare Register (CMP)

The Compare Register (CMP) is used in comparison with a write access of a conditional write register. The Compare Register is loaded on a read of any of the conditional event Attention Registers or by directly writing to it.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
1Fh	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
CMP7	CMP6	CMP5	CMP4	CMP3	CMP2	CMP1	CMP0

Bit	Symbol	Description
D0-7	CMP0-7	Compare: These bits are compared to bits D0-7 of the accessed register, and only the bits in the Attention Register that have the same current value as the corresponding bit in the Compare register will be updated with the new value.

5.0 Control Information (Continued)

5.4 POINTER RAM REGISTERS

Pointer RAM Registers contain pointers to all data and Descriptors manipulated by the BSI device, namely, Input and Output Data Units, Input and Output Data Unit Descriptors, Request Descriptors, Confirmation Messages, and Pool Space Descriptors. Pointer RAM Registers are shown in Table 5-4.

5.5 LIMIT RAM REGISTERS

The Limit RAM Registers are used by both the Indicate and Request machines. Limit RAM Registers contain data values that define the limits of the ten queues maintained by the BSI device. Limit RAM Registers are shown in Table 5-5.

5.6 DESCRIPTORS

Descriptors are used to observe and control the operation of the BSI device. They contain address, status, and control information about Indicate and Request operations. Descriptors are stored in lists and wrap-around queues in memory external to the BSI device and accessed via the ABus.

Descriptors include the following:

- **Input Data Unit Descriptors (IDUDs)** specify the location, size, part, and status information for Input Data Units.
- **Output Data Unit Descriptors (ODUDs)** specify the location and size of Output Data Units. For multi-ODUD frames, they also specify which part of the frame is pointed to by the ODUD.
- **Pool Space Descriptors (PSPs)** describe the location and size of a region of memory space available for writing Indicate data.
- **Request Descriptors (REQs)** describe the location of a stream of Output Data Unit Descriptors and contain operational parameters
- **Confirmation Status Messages (CNFs)** describe the result of a Request operation.

5.7 OPERATING RULES

Multi-Byte Register Ordering

When referring to multi-byte fields, byte 0 is always the most significant byte. When referring to bits within a byte, bit 7 is the most significant bit and bit 0 is the least significant bit.

When referring to the contents of a byte, the most significant bit is always referred to first.

5.0 Control Information (Continued)

TABLE 5-4. Pointer RAM Registers

Group	Address	Register Name	Access Rules	
			Read	Write
P O I N T E R R A M	00	ODU Pointer RCHN1 (OPR1)	Always	Always
	01	ODUD List Pointer RCHN1 (OLPR1)	Always	Always
	02	CNF Queue Pointer RCHN1 (CQPR1)	Always	Always
	03	REQ Queue Pointer RCHN1 (RQPR1)	Always	Always
	04	ODU Pointer RCHN0 (OPR0)	Always	Always
	05	ODUD List Pointer RCHN0 (OLPR0)	Always	Always
	06	CNF Queue Pointer RCHN0 (CQPR0)	Always	Always
	07	REQ Queue Pointer RCHN0 (RQPR0)	Always	Always
	08	IDU Pointer ICHN2 (IPI2)	Always	Always
	09	IDUD Queue Pointer ICHN2 (IQPI2)	Always	Always
	0A	PSP Queue Pointer ICHN2 (PQPI2)	Always	Always
	0B	Next PSP ICHN2 (NPI2)	Always	Always
	0C	IDU Pointer ICHN1 (IPI1)	Always	Always
	0D	IDUD Queue Pointer ICHN1 (IQPI1)	Always	Always
	0E	PSP Queue Pointer ICHN1 (PQPI1)	Always	Always
	0F	Next PSP ICHN1 (NPI1)	Always	Always
	10	IDU Pointer ICHN0 (IPI0)	Always	Always
	11	IDUD Queue Pointer ICHN0 (IQPI0)	Always	Always
	12	PSP Queue Pointer ICHN0 (PQPI0)	Always	Always
	13	Next PSP ICHN0 (NPI0)	Always	Always
14	IDUD Shadow Register (ISR)	Always	Always	
15	ODUD Shadow Register (OSR)	Always	Always	
16-1F	Reserved	N/A	N/A	

TABLE 5-5. Limit RAM Registers

Group	Address	Register Name	Access Rules	
			Read	Write
L I M I T R A M	0	REQ Queue Limit RCHN1 (RQLR1)	Always	Always
	1	CNF Queue Limit RCHN1 (CQLR1)	Always	Always
	2	REQ Queue Limit RCHN0 (RQLR0)	Always	Always
	3	CNF Queue Limit RCHN0 (CQLR0)	Always	Always
	4	IDUD Queue Limit ICHN2 (IQLI2)	Always	Always
	5	PSP Queue Limit ICHN2 (PQLI2)	Always	Always
	6	IDUD Queue Limit ICHN1 (IQLI1)	Always	Always
	7	PSP Queue Limit ICHN1 (PQLI1)	Always	Always
	8	IDUD Queue Limit ICHN0 (IQLI0)	Always	Always
	9	PSP Queue Limit ICHN0 (PQLI0)	Always	Always
A-F	Reserved	N/A	N/A	

5.0 Control Information (Continued)

5.8 POINTER RAM REGISTER DESCRIPTIONS

The Pointer RAM Register set contains 32, 28-bit registers. Registers 23 through 31 are reserved, and user access of these locations produces undefined results.

Pointer RAM Registers are read and written by the host using the Pointer RAM Operation (PTOP) service function and are accessed directly by BSI device hardware during Indicate and Request operations.

During Indicate and Request operations, Pointer RAM registers are used as addresses for ABus accesses of data and Descriptors, i.e., the subchannel addresses for loads (reads) of streams of PSPs, ODUs, ODUDs, and REQs, and for stores (writes) of streams of IDUs, IDUDs, and CNFs.

Pointer RAM Registers include the following:

ODU Pointer: Contains the address of an Output Data Unit. During Request operations, this register is loaded by the BSI device from the Location Field of its Output Data Unit Descriptor.

ODUD List Pointer: Loaded by the BSI device from the Location Field of the REQ Descriptor when it is read from memory. The address is incremented by the BSI device as each ODUD is fetched from memory.

CNF Queue Pointer: Contains the current CNF Status Queue address. This register is written by the user after he has allocated space for the CNF Queue. During Request operations, this register is incremented by the BSI device after each CNF is written to the CNF Queue.

REQ Queue Pointer: Initialized by the host with the start address of the REQ Descriptor Queue after the Queue has been initialized. During Request operations, the address is incremented by the BSI device as each REQ is fetched.

IDU Pointer: Written by the BSI device with the Location Field of the PSP Descriptor when it is read from memory.

IDUD Queue Pointer: Points to the Queue location where IDUDs will be stored. Written by the user after he has allocated space for the IDUD Status Queue. Incremented by the BSI device as IDUDs are written to consecutive locations in the Queue.

PSP Queue Pointer Register: Points to the next available PSP. Initialized by the host with the start address of the PSP Queue, after the Queue has been initialized with valid PSP Descriptors. As each PSP is read from memory, this register is loaded with the address in the Next PSP Register.

Next PSP Register: Written by the BSI device with the PSP fetched from the PSP Queue.

Indicate Shadow Register: Written by the BSI device with the start address of the last IDU copied to memory.

Request Shadow Register: Written by the BSI device with the address of the current ODUD.

See Table 5-4 for Summary including address and access rules.

5.9 LIMIT RAM REGISTER DESCRIPTIONS

The Limit RAM Register set contains 16, 9-bit registers. Registers 11 through 15 are reserved, and used access of these locations produces undefined results.

The Limit RAM registers contain data values that define the limits of each of the ten queues maintained by the BSI device.

Limit RAM Registers are read and written by the host using the Limit RAM Operation (LMOP) service function when the Status/Space Machine is in STOP Mode, and are read directly by BSI device hardware during Indicate and Request operations.

Limit RAM Registers include the following:

REQ Queue Limit: Defines the last valid REQ written by the host.

CNF Queue Limit Register: Defines the last Queue location where a CNF may be written by the BSI device. Due to pipelining, the BSI device may write up to two CNFs after it detects a write to the next-to-last CNF entry (and generates a No Status Space Attention). For this reason, the host must always define the CNF queue limit to be one Descriptor less than the available space.

IDUD Queue Limit Register: Defines the last Queue location where an IDUD may be written by the BSI device.

PSP Queue Limit: Defines the last valid PSP written by the host.

See Table 5-5 for Summary including address and access rules.

5.0 Control Information (Continued)

5.10 BSI DEVICE DESCRIPTORS

Input Data Unit Descriptor (IDUD)

Input Data Unit Descriptors (IDUDs) are generated on Indicate Channels to describe where the BSI device wrote each frame part and to report status for the frame.

For multi-part IDUDs, intermediate status is written in each IDUD, and when a status event occurs, definitive status is written in the last IDUD.

A detailed description of the encodings of the Indicate Status bits is given in Table 5-6.

31	30	29	28	27	24	23	16	15	14	13	12	0
IS			FRA			FRS		VC	RES		CNT	
F-L		RES				LOC						

Word 0
Word 1

Word 0

Bit	Symbol	Description
D0-12	CNT	Byte Count: Number of bytes in the SDU.
D13-14	RES	Reserved
D15	VC	VCOPY: Reflects the state of the VCOPY signal sent to the BMAC device for this frame. 0: VCOPY was negated. 1: VCOPY was asserted.
D16-23	FRS	Frame Status: This field is valid only for Full Confirmation, and if the frame ended with an ED.
D16-17	C	C Indicator: 00: none 01: R 10: S 11: T
D18-19	A	A Indicator: 00: none 01: R 10: S 11: T
D20-21	E	E Indicator: 00: none 01: R 10: S 11: T
D22	VFCS	Valid FCS: 0: FCS field was invalid 1: FCS field was valid
D23	VDL	Valid Data Length: 0: Data length was invalid 1: Data length was valid

5.0 Control Information (Continued)

5.10 BSI DEVICE DESCRIPTORS (Continued)

Input Data Unit Descriptor (IDUD) (Continued)

Word 0 (Continued)

Bit	Symbol	Description																																																																																																									
D24–27	FRA	Frame Attributes																																																																																																									
D24–25	TC	Termination Condition: This field is valid only for Full Confirmation. 00: Other (e.g., MAC Reset/token). 01: ED. 10: Format error. 11: Frame stripped.																																																																																																									
D26	AFLAG	AFLAG: Reflects the state of the AFLAG input signal, which is sampled by the BSI device at INFORCVD. This field is valid only for Full Confirmation. 0: External DA match. 1: Internal DA match.																																																																																																									
D27	MFLAG	MFLAG: Reflects the state of the MFLG input signal, which is sampled by the BSI device at INFORCVD. This field is valid only for Full Confirmation. 0: Frame sent by another station. 1: Frame sent by this station.																																																																																																									
D28–31	IS	<p>Indicate Status: The values in this field are prioritized, with the highest number having the highest priority. A detailed description of the encodings is given in Table 5-6.</p> <table border="1"> <thead> <tr> <th>IS3</th> <th>IS2</th> <th>IS1</th> <th>IS0</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td colspan="5">Non-end Frame Status</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Last IDU of queue, page-cross.</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>Page boundary crossed.</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>End of header.</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>Page-cross with header-end.</td> </tr> <tr> <td colspan="5">Normal-end Frame Status</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>Intermediate (no breakpoints).</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>Burst boundary.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>Threshold.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>Service opportunity.</td> </tr> <tr> <td colspan="5">Copy Abort due to No Space</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>No data space.</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>No header space.</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>Good header, info not copied.</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>Not enough info space.</td> </tr> <tr> <td colspan="5">Error</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>FIFO overrun.</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>Bad frame (no VDL or no VFCS).</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>Parity error.</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>Internal error.</td> </tr> </tbody> </table>	IS3	IS2	IS1	IS0	Meaning	Non-end Frame Status					0	0	0	0	Last IDU of queue, page-cross.	0	0	0	1	Page boundary crossed.	0	0	1	0	End of header.	0	0	1	1	Page-cross with header-end.	Normal-end Frame Status					0	1	0	0	Intermediate (no breakpoints).	0	1	0	1	Burst boundary.	0	1	1	0	Threshold.	0	1	1	1	Service opportunity.	Copy Abort due to No Space					1	0	0	0	No data space.	1	0	0	1	No header space.	1	0	1	0	Good header, info not copied.	1	0	1	1	Not enough info space.	Error					1	1	0	0	FIFO overrun.	1	1	0	1	Bad frame (no VDL or no VFCS).	1	1	1	0	Parity error.	1	1	1	1	Internal error.
IS3	IS2	IS1	IS0	Meaning																																																																																																							
Non-end Frame Status																																																																																																											
0	0	0	0	Last IDU of queue, page-cross.																																																																																																							
0	0	0	1	Page boundary crossed.																																																																																																							
0	0	1	0	End of header.																																																																																																							
0	0	1	1	Page-cross with header-end.																																																																																																							
Normal-end Frame Status																																																																																																											
0	1	0	0	Intermediate (no breakpoints).																																																																																																							
0	1	0	1	Burst boundary.																																																																																																							
0	1	1	0	Threshold.																																																																																																							
0	1	1	1	Service opportunity.																																																																																																							
Copy Abort due to No Space																																																																																																											
1	0	0	0	No data space.																																																																																																							
1	0	0	1	No header space.																																																																																																							
1	0	1	0	Good header, info not copied.																																																																																																							
1	0	1	1	Not enough info space.																																																																																																							
Error																																																																																																											
1	1	0	0	FIFO overrun.																																																																																																							
1	1	0	1	Bad frame (no VDL or no VFCS).																																																																																																							
1	1	1	0	Parity error.																																																																																																							
1	1	1	1	Internal error.																																																																																																							

Word 1

Bit	Symbol	Description
D0–27	LOC	Location: 28-bit memory address of the start of an IDU. For the first IDU of a frame, the address is of the fourth FC byte of the burst-aligned frame (i.e., bits [1:0] = 11). For subsequent IDUs, the address is of the first byte of the IDU (i.e., bits [1:0] = 00).
D28–29	RES	Reserved
D30–31	F–L	First/Last Tag: Identifies the IDU object part, i.e., Only, First, Middle, or Last.

5.0 Control Information (Continued)

TABLE 5-6. Indicate Status Field (IS) of IDU Descriptor

NON-END FRAME STATUS

[0000]	Last IDUD of Queue, with a Page Cross: The last available location of the ICHN's IDUD queue was written. Since there was a page cross, there was more data to be written. Since there was no more IDUD space, the remaining data was not written. Note that this code will not be written in a IDU.MIDDLE, so that a Zero IS field with Zero F-L tags can be utilized by software as a null descriptor.
[0001]	Page Cross: Must be an IDUD.FIRST or IDUD.MIDDLE. This is part of a frame that filled up the remainder of the current page, requiring a new page for the remainder of the data.
[0010]	Header End: This refers to the last IDU of the header portion of a frame.
[0011]	Page Cross and Header End: The occurrence of a page cross and header end.

NORMAL-END FRAME STATUS

[0100]	Intermediate: A frame ended normally, and there was no breakpoint.
[0101]	Burst Boundary: A frame ended normally, and there was a breakpoint because a burst boundary was detected.
[0110]	Threshold: The copied frame threshold counter was reached when this frame was copied, and the frame ended normally.
[0111]	Service Opportunity: This (normal end) frame was preceded by a token or MACRST, a MAC frame was received, or there was a ring-op change. Any of these events marks a burst boundary.

NO SPACE COPY ABORT

[1000]	Insufficient Data Space: Not all the frame was copied because there was insufficient data space. This code is only written in non-Header/Info Sort Mode.
[1001]	Insufficient Header Space: The frame copy was aborted because there was insufficient header space (in Header/Info Sort Mode).
[1010]	Successful Header Copy, Frame Info Not Copied: There was sufficient space to copy the header, but insufficient data space to copy info, or insufficient IDU space (on ICHN2), or both. No info was copied.
[1011]	No Info Space: The frame's header was copied. When copying the data, there was insufficient data and/or IDU space.

ERROR

[1100]	FIFO Overrun: The Indicate FIFO had an overrun while copying this frame.
[1101]	Bad Frame: The frame did not have a valid data length, or had invalid FCS, or both.
[1110]	Parity Error: There was a parity error during this frame.
[1111]	Internal Error: There was an internal logic error during this frame.

5.0 Control Information (Continued)

REQ Descriptor (REQ)

Request Descriptors (REQs) contain the part, byte address, and size of one or more Output Data Unit Descriptors. They also contain parameters and commands to the BSI device associated with Request operations.

Multiple REQ Descriptors (parts) may be grouped as one Request Descriptor object by the host software, with the REQ.First defining the parameters for the entire Request object. Also, multiple Output Data Unit Descriptors may be grouped contiguously, to be described by a single REQ Descriptor.

Each REQ part is fetched by the BSI device from the Request Channel's REQ Descriptor Queue, using the REQ Queue Pointer Register. Each Request Channel processes one Request Descriptor, per service opportunity, until a REQ.Last is encountered.

The BSI device checks for the following inconsistencies when the REQ is loaded from memory:

1. REQ.F with invalid Confirmation Class (as shown in the Table 5-8).
2. REQ.First with Request Class = 0.
3. REQ.First, when the previous REQ was not a REQ.Last or REQ.Only.
4. REQ which is not a REQ.First, when the previous REQ was a REQ.Last or a REQ.Only.

When an inconsistency is detected, the BSI device aborts the Request, and reports the exception in the Request Status field of the CNF Descriptor.

The encodings of the RQCLS and CNFCLS bits are described in more detail in Tables 5-7 and 5-8 respectively.

31	30	29	28	27	24	23	16	15	12	11	8	7	0
RES		UID			SIZE		CNFCLS		RQCLS		FC		Word 0
F-L		RES			LOC								Word 1

Word 0

Bit	Symbol	Description
D0-7	FC	Frame Control: Frame control field to be used unless FC transparency is enabled. This field is decoded to determine whether to assert RQCLM or RQBCN. This decoding is always active, i.e., regardless of frame control transparency. This field is also used for comparing received frames when confirming (without FC transparency).
D8-11	RQCLS	Request/Release Class: This field encodes the Request Class for the entire Request object, and is thus only sampled on a REQ.First or REQ.Only. The field is asserted on the RQCLS output signals to the BMAC device when requesting a token. If the Request Class is incompatible with the current ring state, the BSI device sets the RCHN's USR bit in the Request Attention Register. The encoding of this field is shown in Table 5-7.
D12-15	CNFCLS	Confirmation Class: This field encodes the Confirmation Class for the entire Request object, and is only sampled on a REQ.First or REQ.Only. The encoding of this field is shown in Table 5-8.
D12	E	End: Enables confirmation on completion of request. 0: CNFs on completion disabled. 1: CNFs on completion enabled.
D13	I	Intermediate: Enables Intermediate Confirmation. 0: Intermediate CNFs disabled. 1: Intermediate CNFs enabled.
D14	F	Full/Transmitter: Selects between Transmitter and Full Confirmation. 0: Transmitter confirm. 1: Full confirm.
D15	R	Repeat: Enables repeated transmission of the first frame of the request until the request is aborted. This may be used when sending BEACON or CLAIM frames. 0: Fetch all frames of REQ. 1: Repeat transmission of first frame of REQ. A Request may use Repeat on RCHN1, and have a Request loaded on RCHN0, but not vice-versa. Specifically, when a Request with the Repeat option is loaded on RCHN0, RCHN1 must not have any REQs active or visible to the BSI device. Thus REQs on RCHN1 may be queued externally but the queue's Limit Register must not be set at or after that point. Requests with the Repeat option should only be used on one Request Channel at a time, and preferably on RCHN0.

5.0 Control Information (Continued)**REQ Descriptor (REQ)** (Continued)**Word 0** (Continued)

Bit	Symbol	Description
D16–23	SIZE	<p>Size: Count of number of frames represented by the ODUD stream pointed to by LOC. REQ Descriptors with a frame count are permitted, and are typically used to end a Request, without having to send data. For example, to end a restricted dialogue, a REQ.Last with SIZE = 0 will cause the Request Machine to command the BMAC device to capture and release the specified classes of token. The response of the BSI device to REQs with SIZE = 0 is as follows:</p> <ol style="list-style-type: none"> 1. REQ.First: BSI device latches the REQ Descriptor fields, then fetches the next REQ. REQCLS is asserted, but RQRDY remains deasserted. 2. REQ.Middle: BSI device fetches the next REQ. 3. REQ.Only: BSI device requests the capture of the appropriate token. When it is captured, the BSI asserts RQFINAL and ends the request. 4. REQ.Last: BSI device captures the token, asserts RQFINAL, then marks the request complete.
D24–29	UID	User Identification: Contains the UID field from the current REQ.First or REQ.Only.
D30–31	RES	Reserved

Word 1

Bit	Symbol	Description
D0–27	LOC	Location: Bits [27:2] are the memory word address of ODUD stream. Bits [1:0] are expected to be 00, and are not checked.
D28–29	RES	Reserved
D30–31	F–L	First/Last Tag: Identifies the ODUD stream part, i.e., Only, First, Middle, or Last.

5.0 Control Information (Continued)

TABLE 5-7. REQ Descriptor Request Class Field Encodings

RQCLS Value	RQCLS Name	Class Type	THT	Token Capture	Token Issue	Notes
0000	None	None	—	none	non	
0001	Apr1	Async pri1	E	non-r	non-r	
0010	Reserved	Reserved				
0011	Reserved	Reserved				
0100	Syn	Sync	D	any	capt	1
0101	Imm	Immed	D	none	none	4
0110	ImmN	Immed	D	none	non-r	4
0111	ImmR	Immed	D	none	restr	4
1000	Asyn	Async	E	non-r	non-r	
1001	Rbeg	Restricted	E	non-r	restr	2, 3
1010	Rend	Restricted	E	restr	non-r	2
1011	Rcnt	Restricted	E	restr	restr	2
1100	AsynD	Async	D	non-r	non-r	
1101	RbegD	Restricted	D	non-r	restr	2, 3
1110	RendD	Restricted	D	restr	non-r	2
1111	RcntD	Restricted	D	restr	restr	2

E = enabled, D = disabled, non-r = non-restricted, restr = restricted, capt = captured

Note 1: Synchronous Requests are not serviced when bit BCNR of the Ring Event Latch Register is set.

Note 2: Restricted Requests are not serviced when bit BCNR, CLMR, or OTRMAC of the Ring Event Latch Register is set.

Note 3: Restricted Dialogues only begin when a Non-Restricted token has been received and transmitted.

Note 4: Immediate Requests are serviced when the ring is Non-Operational. These requests are serviced from the Data state if neither signal RQCLM nor RQBCN is asserted. If signal RQCLM is asserted, Immediate Requests are serviced from the Claim State. If signal RQBCN is asserted, Immediate Requests are serviced from the Beacon State. RQCLM and RQBCN do not cause transitions to the Claim and Beacon States.

TABLE 5-8. REQ Descriptor Confirmation Class Field Encodings

[R]	[F]	[I]	[E]	Confirmation Class
x	0	0	0	Invalid (consistency failure)
x	x	1	0	Invalid (consistency failure)
0	x	0	0	None: Confirmation only on exception
0	0	0	1	Tend: Transmitter confirm, CNF on exception or completion
0	0	1	1	Tint: Transmitter confirm, CNF on exception, completion or intermediate
0	1	0	1	Fend: Full Confirm, CNF on exception or completion
0	1	1	1	Fint: Full Confirm, CNF on exception, completion or intermediate
1	1	0	0	NoneR: Confirmation only on exception, repeat frame
1	0	0	1	TendR: Transmitter confirm, CNF on exception or completion, repeat frame
1	0	1	1	TintR: Transmitter confirm, CNF on exception, completion or intermediate, repeat frame
1	1	0	1	FendR: Full confirmation, CNF on exception or completion, repeat frame
1	1	1	1	FintR: Full Confirmation, CNF on exception, completion, or intermediate, repeat frame

5.0 Control Information (Continued)

Output Data Unit Descriptor (ODUD)

An Output Data Unit Descriptor (ODUD) contains the part, byte address and size of an Output Data Unit. During Request operations, ODUDs are fetched by the BSI device from a list in memory, using the address in the ODUD List Pointer Register (in the Pointer RAM).

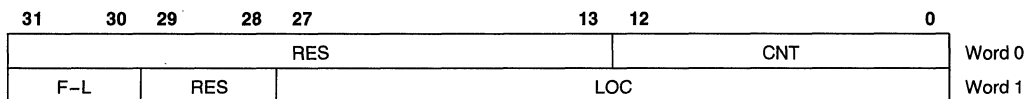
ODUDs may have a zero byte count, which is useful for fixed protocol stacks. One layer may be called, and if it has no data to add to the frame, it may add an ODUD with a zero byte count to the list.

The BSI device checks for the following inconsistencies when an ODUD is loaded from memory:

1. ODUD.First, when the previous ODUD was not an ODUD.Last or ODUD.Only.
2. ODUD which is not an ODUD.First, when the previous ODUD was an ODUD.Last or ODUD.Only.
3. ODUD.First with zero byte count.

When an inconsistency is detected, the BSI device aborts the Request, and reports the exception in the Request Status field of the CNF Descriptor.

ODUDs must contain at least 4 bytes (for short addresses).



Word 0

Bit	Symbol	Description
D0-12	CNT	Byte Count: Number of bytes in the ODU. The size may be Zero, which is useful for fixed protocol stacks.
D13-31	RES	Reserved

Word 1

Bit	Symbol	Description
D0-27	LOC	Location: Memory byte address of SDU.
D28-29	RES	Reserved
D30-31	F-L	First/Last Tag: Identifies the Output Data Unit part, i.e., Only, First, Middle, or Last.

5.0 Control Information (Continued)

Confirmation Status Message Descriptor (CNF)

A Confirmation Status Message (CNF) describes the result of a Request operation.

A more detailed description of the encoding of the RS bits is given in Table 5-9.

31	30	29	28	27	24	23	16	15	8	7	0	
RS				FRA		FRS		TFC		CFC		Word 0
F-L		UID				FC		CS		RES		Word 1

Word 0

Bit	Symbol	Description
D0-7	CFC	Confirmed Frame Count: Number of confirmed frames. Valid only for Full Confirmation.
D8-15	TFC	Transmitted Frame Count: Number of frames successfully transmitted by the BSI device and BMAC device. Valid for all confirmation classes.
D16-23	FRS	Frame Status: This field is valid only for Full Confirmation, and if the frame ended with an ED.
D16-17	C	C Indicator: 00: None 01: R 10: S 11: T
D18-19	A	A Indicator: 00: None 01: R 10: S 11: T
D20-21	E	E Indicator: 00: None 01: R 10: S 11: T
D22	VFCS	Valid FSC: 0: FSC Field was Invalid. 1: FSC Field was Valid.
D23	VDL	Valid Data Length: 0: Data Length was Invalid. 1: Data Length was Valid.
D24-27	FRA	Frame Attributes: This field is valid only for Full Confirmation.
D24-25	TC	Terminating Condition: 00: Other (e.g., MAC Reset/token). 01: Ed. 10: Format Error. 11: Frame Stripped.
D26	AFLAG	AFLAG: Reflects the state of the AFLAG input signal, which is sampled by the BSI device at INFORCVD. 0: No DA Match. 1: DA Match.
D27	MFLAG	MFLAG: Reflects the state of the MFLAG input signal, which is sampled by the BSI device at INFORCVD. 0: Frame Sent by another Station. 1: Frame Sent by this Station.

5.0 Control Information (Continued)

Confirmation Status Message Descriptor (CNF) (Continued)

Word 0 (Continued)

Bit	Symbol	Description				
D28-31	RS	Request Status: This field represents a priority encoded status value, with the highest number having the highest priority. This field is described in Table 5-9.				
		RS3	RS2	RS1	RS0	Meaning
		Intermediate				
		0	0	0	0	None
		0	0	0	1	Preempted
		0	0	1	0	Part Done
		Breakpoints				
		0	0	1	1	Service Loss
		0	1	0	0	Reserved
		Completion				
		0	1	0	1	Completed BEACON
		0	1	1	0	Completed OK
		Exception Completion				
		0	1	1	1	Bad Confirmation
		1	0	0	0	Underrun
		1	0	0	1	Host Abort
		1	0	1	0	Bad Ringop
		1	0	1	1	MAC Abort
		1	1	0	0	Timeout
		1	1	0	1	MAC Reset
		1	1	1	0	Consistency Failure
		Error				
		1	1	1	1	Internal or Fatal ABUS Error

5.0 Control Information (Continued)

Confirmation Status Message Descriptor (CNF) (Continued)

Word 1

Bit	Symbol	Description
D0-7	RES	Reserved
D8-15	CS	Confirmation Status
D8-9	FT	Frame Type: This field reflects the type of frame that ended Full Confirmation. 00: Any Other. 01: Token. 10: Other Void. 00: My Void.
D10	F	Full Confirm: This bit is set when the Request was for Full Confirmation.
D11	U	Unexpected Frame Status: This bit is set when the frame status does not match the value programmed in the Request Expected Frame Status Register. This applies only to Full Confirmation.
D12	P	Parity: This bit is set when a parity error is detected in a received frame. Parity is checked from FC to ED inclusive if the FLOW bit in the Mode Register is set.
D13	E	Exception: This bit is set when an exception occurs. The RCHN's EXC bit in the Request Attention Register is also set.
D14	R	Ring-Op: This bit is set when the ring enters a bad operational state after transmission but before all returning frames have been confirmed.
D15	T	Transmit Class: 0: Restricted. 1: Non-Restricted.
D16-23	FC	Frame Control: Frame Control field of the last frame of the last confirmed burst. Valid only for Full Confirmation.
D24-29	UID	User Identification: Contains the UID field copied from the current REQ.FIRST or REQ.ONLY.
D30-31	F-L	First/Last Tag: Identifies the CNF part, i.e., Only, First, Middle, or Last.

5.0 Control Information (Continued)

TABLE 5-9. Request Status (RS) Field of CNF Descriptor

INTERMEDIATE

[0000]	None: Non status is written. This may be used by software to identify a NULL or invalid CNF.
[0001]	Preempted: RCHN1 was preempted by RCHN0. RCHN1 will be serviced following RCHN0.
[0010]	Part None: The BSI device is servicing a Request, but it cannot hold onto a token, and the last frame of a Request.part has been transmitted.

BREAKPOINTS

[0011]	Service Loss: The THT expired during a Request with THT enabled. Only occurs for Intermediate Confirmation.
[0100]	Reserved

COMPLETION

[0101]	Completed BEACON: When transmitting from the BEACON state, this status is returned when the BMAC device receives a My__Beacon. When transmitting from the CLAIM state, this status is returned when the BMAC device wins the CLAIM process.
[0110]	Completed OK: Normal completion with good status.

EXCEPTION COMPLETION

[0111]	Bad Confirmation: There was an error during confirmation, causing the Request to complete with this status, or one of higher priority. Confirmation errors include MACRST, ring-operational change, receiving an Other__Void or My__Void or token, receiving a bad frame, or receiving a frame that did not match the programmed expected frame status.
[1000]	Underrun: There was no data in the request data FIFO when it was required to be presented to the BMAC device.
[1001]	Host Abort: The host aborted the Request on this Request Channel, either directly by clearing the ABT bit in the Service Attention Register or indirectly by having insufficient entries in the CNF queue.
[1010]	Bad Ringop: A Request was loaded with a Request Class inappropriate for the current ring operational state.
[1011]	MAC Device Abort: The BMAC device aborted the Request and asserted TXABORT. This could be from an interface parity error, or because the transmitted frame failed the FC check, or because the BMAC device received a MAC frame while transmitting in the BEACON state. This status is also returned when the BMAC device receives an Other__Beacon while the BSI device is transmitting in the BEACON state, or when the CLAIM process is lost while the BSI device is transmitting in the CLAIM state.
[1100]	Timeout: The TRT expired during a Request with THT disabled. The Request is aborted.
[1101]	MAC Reset: The BMAC device asserted MACRST.
[1110]	Consistency Failure: There was an inconsistency within the REQ or ODUD stream.

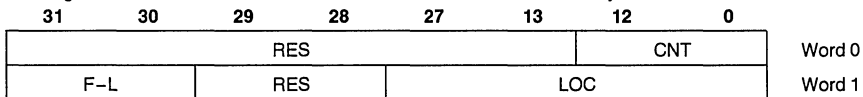
ERROR

[1111]	Internal or Fatal ABus Error: There was an internal logic error or a fatal ABus error while writing a CNF.
--------	---

5.0 Control Information (Continued)

Pool Space Descriptor (PSP)

Pool Space Descriptors (PSPs) contain the address and size (in bytes) of a free space in host memory available for writing Input Data Units. When PSPs are read by the BSI device, the address field of the PSP is loaded into the Indicate Channel's IDU Pointer Register, and is used as the subchannel address for the IDU memory write.



Word 0

Bit	Symbol	Description
D0-12	CNT	Byte Count: Number of bytes in the available memory area (up to 4k bytes).
D13-31	RES	Reserved

Word 1

Bit	Symbol	Description
D0-27	LOC	Location: Memory byte address of memory area available for writing IDUs. Normally the page offset will be Zero to simplify space management.
D30-31	RES	Reserved

6.0 Signal Descriptions

6.1 PIN ORGANIZATION

The BSI device pinout is organized into five groups:

Control Interface: Used for host microprocessor access to the BSI device.

BMAC Device Indicate Interface: Pins for receiving and processing incoming frames from the DP82361 BMAC device.

BMAC Device Request Interface: Pins for transmitting frames to the BMAC device.

ABUS Interface: Pins for transferring data and data information between system memory and the BSI device.

Electrical Interface: Pins associated with power supply, clocking, and scan test.

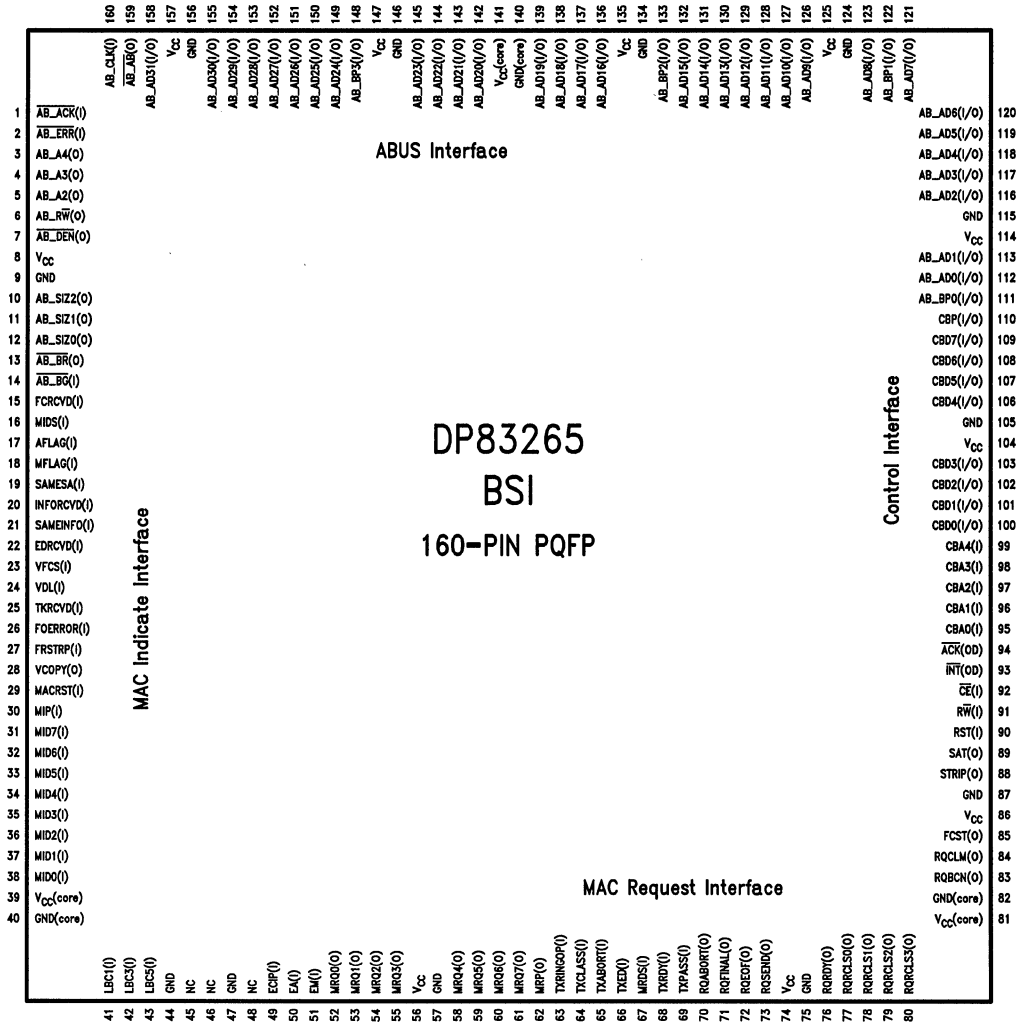


FIGURE 6-1. DP83265 160-Pin Pinout

TL/F/10791-12

Order Number DP83265VF
See NS Package Number VF160A

6.0 Signal Descriptions (Continued)

DP83265 Pinout Description

Pin	Description	I/O
1	AB_ACK	I
2	AB_ERR	I
3	AB_A4	O
4	AB_A3	O
5	AB_A2	O
6	AB_RW	O
7	AB_DEN	O
8	V _{CC}	
9	GND	
10	AB_SIZ2	O
11	AB_SIZ1	O
12	AB_SIZ0	O
13	AB_BR	O
14	AB_BG	I
15	FCRCVD	I
16	MIDS	I
17	AFLAG	I
18	MFLAG	I
19	SAMESA	I
20	INFORCVD	I
21	SAMEINFO	I
22	EDRCVD	I
23	VFCS	I
24	VDL	I
25	TKRCVD	I
26	FOERROR	I
27	FRSTRP	I
28	VCOPY	O
29	MACRST	I
30	MIP	I
31	MID7	I
32	MID6	I
33	MID5	I
34	MID4	I
35	MID3	I
36	MID2	I
37	MID1	I
38	MID0	I
39	V _{CC}	Core
40	GND	Core

Pin	Description	I/O
41	LBC1	I
42	LBC3	I
43	LBC5	I
44	GND	
45	NC	
46	NC	
47	GND	
48	NC	
49	ECIP	I
50	EA	I
51	EM	I
52	MRQ0	O
53	MRQ1	O
54	MRQ2	O
55	MRQ3	O
56	V _{CC}	
57	GND	
58	MRQ4	O
59	MRQ5	O
60	MRQ6	O
61	MRQ7	O
62	MRP	O
63	TXRINGOP	I
64	TXCLASS	I
65	TXABORT	I
66	TXED	I
67	MRDS	I
68	TXRDY	I
69	TXPASS	I
70	RQABORT	O
71	RQFINAL	O
72	RQEOF	O
73	RQSEND	O
74	V _{CC}	
75	GND	
76	RQRDY	O
77	RQRCLS0	O
78	RQRCLS1	O
79	RQRCLS2	O
80	RQRCLS3	O

Pin	Description	I/O
81	V _{CC}	Core
82	GND	Core
83	RQBCN	O
84	RQCLM	O
85	FCST	O
86	V _{CC}	
87	GND	
88	STRIP	O
89	SAT	O
90	RST	I
91	RW	I
92	CE	I
93	INT	OD
94	ACK	OD
95	CBA0	I
96	CBA1	I
97	CBA2	I
98	CBA3	I
99	CBA4	I
100	CBD0	I/O
101	CBD1	I/O
102	CBD2	I/O
103	CBD3	I/O
104	V _{CC}	
105	GND	
106	CBD4	I/O
107	CBD5	I/O
108	CBD6	I/O
109	CBD7	I/O
110	CBP	I/O
111	AB_BP0	I/O
112	AB_AD0	I/O
113	AB_AD1	I/O
114	V _{CC}	
115	GND	
116	AB_AD2	I/O
117	AB_AD3	I/O
118	AB_AD4	I/O
119	AB_AD5	I/O
120	AB_AD6	I/O

Pin	Description	I/O
121	AB_AD7	I/O
122	AB_BP1	I/O
123	AB_AD8	I/O
124	GND	
125	V _{CC}	
126	AB_AD9	I/O
127	AB_AD10	I/O
128	AB_AD11	I/O
129	AB_AD12	I/O
130	AB_AD13	I/O
131	AB_AD14	I/O
132	AB_AD15	I/O
133	AB_BP2	I/O
134	GND	
135	V _{CC}	
136	AB_AD16	I/O
137	AB_AD17	I/O
138	AB_AD18	I/O
139	AB_AD19	I/O
140	GND	Core
141	V _{CC}	Core
142	AB_AD20	I/O
143	AB_AD21	I/O
144	AB_AD22	I/O
145	AB_AD23	I/O
146	GND	
147	V _{CC}	
148	AB_BP3	I/O
149	AB_AD24	I/O
150	AB_AD25	I/O
151	AB_AD26	I/O
152	AB_AD27	I/O
153	AB_AD28	I/O
154	AB_AD29	I/O
155	AB_AD30	I/O
156	GND	
157	V _{CC}	
158	AB_AD31	I/O
159	AB_AB	I/O
160	AB_CLK	I/O

6.0 Signal Descriptions (Continued)

6.2 CONTROL INTERFACE

The Control Interface operates asynchronously to the operation of the BMAC device and ABus interfaces.

The $\overline{\text{ACK}}$ and $\overline{\text{INT}}$ signals are open drain to allow wire ORing.

Symbol	Pin #	I/O	Description
CBP	110	I/O	Control Bus Parity: Odd parity on CBD7–0.
CBD7–0	109-106, 103–100	I/O	Control Bus Data: Bidirectional Data bus.
CBA4–0	99–95	I	Control Bus Address: Address of a particular BSI device register.
$\overline{\text{CE}}$	92	I	Control Bus Enable: Handshake signal used to begin a Control Interface access. Active low signal.
R/ $\overline{\text{W}}$	91	I	Read/Write: Determines current direction of a Control Interface access.
$\overline{\text{ACK}}$	94	OD	Acknowledge: Acknowledges that the Control Interface access has been performed. Active low, open drain signal.
$\overline{\text{INT}}$	93	OD	Interrupt: Indicates presence of one or more enabled conditions. Active low, open drain signal.
$\overline{\text{RST}}$	90	I	Reset: Causes a reset of BSI device state machines and registers.

6.0 Signal Descriptions (Continued)

6.3 BMAC Device Indicate Interface

The BMAC Device Indicate Interface signals provide data and control bytes as received from the BMAC device. Each Indicate Data byte is also provided with odd parity.

MID7–0 signals are valid on the rising edge of the Local Byte Clock signal (provided by the Clock Recovery Device).

Data:

Symbol	Pin #	I/O	Description
MIP	30	I	MAC Indicate Parity: This is connected directly to the corresponding BMAC device pin of similar name. Odd parity on MID7–0. Only valid with Data and Status indicators.
MID7–0	31–38	I	MAC Indicate Data: This is connected directly to the corresponding BMAC device pin of similar name. Data: The BMAC device indicates data is being presented on MID7–0 during the time when RCSTART is asserted until one of the following signals is asserted: EDRCVD, TKRCVD, FOERROR, or MACRST. Status: The BMAC device indicates Status Indicators are being presented on MID7–0 when EDRCVD or TKRCVD is asserted.

Frame Sequencing:

The Frame Sequencing signals apply to the data available at the MAC Indicate Interface (MIP and MID7–0). The Frame Sequencing signals can be used to control the latching of appropriate Frame Status.

Symbol	Pin #	I/O	Description
FCRCVD	15	I	Frame Control Received: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the Frame Control Field has been received.
INFORCVD	20	I	Information Field Received: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that four bytes of the Information Field have been received. It is asserted by the BMAC device on the fourth byte of the INFO field and remains active until the next JK symbol pair is received.
EDRCVD	22	I	EDFS Received: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the End of Frame Sequence has been received.
MIDS	16	I	MAC Indicate Data Stroke: Asserted by BMAC device to indicate valid data. This signal should be tied to V _{CC} for FDDI-I, and used for FDDI-II.

Frame Information:

Symbol	Pin #	I/O	Description
AFLAG	17	I	My Destination Address Recognized: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that an internal address match occurred on the Destination Address field. It is reset when the next JK symbol pair is received.
MFLAG	18	I	My Source Address Recognized: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the received Source Address field matched the MLA or MSA BMAC device registers. It is reset when the next JK symbol pair is received.
SAMESA	19	I	Same Source Address: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the SA of the current frame is the same as the previous frame, that the frames were not MAC frames, and that the frames are the same size. It is reset when the next KJ symbol pair is received.
SAMEINFO	21	I	Same MAC Information: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the first four bytes of the IF of the current frame are the same as the previous frame, that the frames were MAC frames, and that their address lengths are the same. SAMEINFO is asserted along with INFORCVD. It is reset when the next JK symbol pair is received.

6.0 Signal Descriptions (Continued)

Frame Status:

Symbol	Pin #	I/O	Description
VDL	24	I	Valid Data Length: The BMAC device indicates a valid data length for the current frame.
VFCS	23	I	Valid Frame Check Sequence: The BMAC device indicates a valid FCS for the current frame.
TKRCVD	25	I	Token Received: The BMAC device indicates that a complete token was received.
FRSTRP	27	I	Frame Stripped: The BMAC device indicates that the current frame was stripped.
FOERROR	26	I	Format Error: The BMAC device indicates a standard-defined format error.
MACRST	29	I	MAC Reset: The BMAC device indicates an internal error, MAC frame, MAC reset, or hardware or software reset.
EA	50	I	External AFlag: This signal is used by external address matching to signal that a Destination Address (DA) match has occurred. Assuming that the proper timing of EA and ECIP are met, the assertion of EA will cause the BSI device to copy this frame. EA is sampled on the cycle after ECIP is deasserted. The sample window is from FCRCVD to EDRCVD.
EM	51	I	External MFlag: This signal is used by external address matching logic to signal a Source Address (SA) match. It is sampled on the clock cycle after ECIP is deasserted.
VCOPY	28	O	Valid Copy: Affects the setting of the transmitted Cx (Copied Indicator). The value of VCOPY is used to determine the value of the transmitted Cx. VCOPY must be asserted one byte time before EDRCVD is asserted.
ECIP	49	I	External Compare in Progress: This signal is asserted to indicate that external address comparison has begun. It is deasserted to indicate that the comparison has completed. EA and EM are sampled upon the deassertion of ECIP. ECIP must be asserted during the period from the assertion of FCRVCD (by the BMAC device) to the assertion of INFORCVD (by the BMAC device) in order for the BSI to recognize an external comparison. It must be deasserted for at least one cycle for the external comparison to complete. If ECIP has not been deasserted before EDRCVD (from the BMAC device), the BSI device will not copy this frame. ECIP may be implemented as a positive or negative pulse.

6.0 Signal Descriptions (Continued)

6.4 BMAC Device Request Interface

The BMAC Device Request Interface signals provide data and control bytes to the BMAC device as received from the Host System. Each Request Data byte is also provided with odd parity.

Symbol	Pin #	I/O	Description
MRP	62	O	MAC Request Parity: This is connected directly to the corresponding BMAC device pin of similar name. Odd parity on MRD7-0.
MRD7-0	61-58, 55-52	O	MAC Request Data: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates data is being presented on MRD7-0.

Service Parameters:

Symbol	Pin #	I/O	Description
RQRCLS3-0	80-77	O	Request Class: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates the service class parameters for this request. When RQRCLS > 0, the BMAC device Transmitter will capture a usable token for non-immediate requests) and assert TXRDY. The service opportunity continues as long as the token is usable with the current service parameters, even if RQRDY is not asserted. When RQRCLS = 0, the service opportunity will terminate after the current frame (even if RQRCLS subsequently becomes non-Zero).
RQCLM	84	O	Request CLAIM: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that this request is to be serviced in the Transmit CLAIM state. Ignored for non-immediate requests.
RQCBN	83	O	Request BEACON: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that this request is to be serviced in the Transmit BEACON state. Ignored for non-immediate requests.

6.0 Signal Descriptions (Continued)

Frame Options:

Symbol	Pin #	I/O	Description
STRIP	88	O	Void Strip: Connected to STRIP and possibly SAT on the BMAC device.
SAT	89	O	Source Address Transparency: Connected to SAIGT on the BMAC device and to SAT on the BMAC device if STRIP is not.
FCST	85	O	Frame Check Sequence Transparency: This is connected directly to the corresponding BMAC device pin of similar name. When selected, the BMAC device will not append FCS to the end of the Information field.

Request Handshake:

Symbol	Pin #	I/O	Description
TXPASS	69	I	Transmit Pass: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates the absence of a service opportunity. This could result from an unusable request class, waiting for a token, timer expiration, or MAC Reset. TXPASS is always asserted between service opportunities. It is deasserted when TXRDY is asserted at the beginning of a service opportunity.
TXRDY	68	I	Transmit Ready: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the BMAC device transmitter is ready for another frame. For a non-immediate request, a useable token must be held in order to transmit frames. TXRDY is asserted by the BMAC device when: a. a usable token is being held, or b. an immediate request becomes serviceable, or c. after frame transmission if the current service opportunity is still usable for another frame. TXRDY is deasserted when TXPASS or TXACK is asserted.
RQRDY	76	O	Request Ready: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the BMAC device transmitter should attempt to use a service opportunity. If RQRDY is asserted within 6 byte times after TXRDY is asserted, the BMAC device transmitter will wait at least L_Max plus one Void frame (4.16 ms – 4.80 ms) for RQSEND to be asserted before releasing the token.
RQSEND	73	O	Request Send: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the BMAC device transmitter should send the next frame. The MRD7–0 signals convey the FC byte when this signal is asserted. If RQSEND is asserted within 6 byte times after TXRDY is asserted, the BMAC device transmitter will send the frame with a minimum length preamble. If RQSEND is not asserted within L_Max plus one Void frame after RQRDY has been asserted (4.16 ms – 4.60 ms), the token may become unusable due to timer expiration. RQSEND may only be asserted when TXRDY and RQRDY are asserted and RQFINAL is deasserted. RQSEND must be deasserted not later than one byte time after TXRDY is deasserted.
MRDS	67	I	MAC Request Data Strobe: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that data on MRD7–0 is valid. This signal should be connected to the TXACK on the BMAC device.
RQEOF	72	O	Request EOF: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that MRD7–0 conveys the last data byte when asserted. Normally, this is the last byte of the INFO field of the frame (exceptions: FCS transparency, invalid frame length). RQEOF causes TXACK to be deasserted and is ignored when TXACK is not asserted.

6.0 Signal Descriptions (Continued)

Request Handshake: (Continued)

Symbol	Pin #	I/O	Description
RQABORT	70	O	<p>Request Abort: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the current frame should be aborted. Normally this causes the BMAC device transmitter to generate a Void, CLAIM, or BEACON frame.</p> <p>RQABORT causes TXACK to be deasserted and is ignored when TXACK is not asserted.</p>
RQFINAL	71	O	<p>Request Final: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the final frame of the request has been presented to the BMAC device Interface.</p> <p>When asserted, the Issue Token Class (as opposed to the Capture Token Class) becomes the new Token Class (TXCLASS). RQFINAL may only be asserted when RQRDY is asserted and RQSEND is deasserted. RQFINAL is ignored unless RQRDY has been asserted for at least one byte time and the service parameters have been valid for at least three byte times.</p> <p>RQFINAL must be deasserted not later than two byte times after TXPASS is deasserted.</p>
TXED	66	I	<p>Transmit End Delimiter: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the ED is being transmitted.</p>
TXABORT	65	I	<p>Transmit Abort: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the MAC Transmitter aborted the current frame.</p>

Transmit Status:

Symbol	Pin #	I/O	Description
TXRINGOP	63	I	<p>Transmit Ring Operational: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates the state of the MAC Transmitter.</p>
TXCLASS	64	I	<p>Transmit Token Class: This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates the class of the current token.</p>

6.0 Signal Descriptions (Continued)

6.5 ABus Interface

The ABus Interface signals provide a 32-bit multiplexed address/data bus for transfers between the host system and the BSI device. The ABus uses a bus request/bus grant protocol that allows for multiple bus masters, supports burst transfers of 4 or 8 32-bit words, and permits both physical and virtual addressing using fixed-size pages.

Address and Data:

Symbol	Pin #	I/O	Description																																		
AB_BP3-0	148, 133, 122, 111	I/O	ABus Byte Parity: These TRI-STATE signals contain the BSI device-generated parity for each address byte of AB_AD, such that AB_BP0 is the parity for AB_AD7-0, AB_BP1 is the parity for AB_AD15-8, etc.																																		
AB_AD31-0	158, 155-149, 145-142, 139-136, 132-126, 123, 121-116, 113-112	I/O	<p>ABus Address and Data: These TRI-STATE signals are the multiplexed ABus address and data lines. During the address phase of a cycle, AB_AD27-0 contain the 28-bit address, and AB_AD31-28 contain a 4-bit function code identifying the type of transaction, encoded as follows:</p> <table border="1"> <thead> <tr> <th>AB_AD[31:28]</th> <th>Transaction Type</th> </tr> </thead> <tbody> <tr><td>0</td><td>RSAP1 ODU Load</td></tr> <tr><td>1</td><td>RSAP1 ODUD Load/CNF Store</td></tr> <tr><td>2</td><td>RSAP1 REQ Load</td></tr> <tr><td>3</td><td>RSAP0 ODU Load</td></tr> <tr><td>4</td><td>RSAP0 ODUD Load/CNF Store</td></tr> <tr><td>5</td><td>RSAP0 REQ Load</td></tr> <tr><td>6</td><td>ISAP2 IDU Store</td></tr> <tr><td>7</td><td>ISAP2 IDUD Store</td></tr> <tr><td>8</td><td>ISAP2 PSP Load</td></tr> <tr><td>9</td><td>ISAP1 IDU Store</td></tr> <tr><td>A</td><td>ISAP1 IDUD Store</td></tr> <tr><td>B</td><td>ISAP1 PSP Load</td></tr> <tr><td>C</td><td>ISAP0 IDU Store</td></tr> <tr><td>D</td><td>ISAP0 IDUD Store</td></tr> <tr><td>E</td><td>ISAP0 PSP Load</td></tr> <tr><td>F</td><td>PTR RAM Load/Store</td></tr> </tbody> </table>	AB_AD[31:28]	Transaction Type	0	RSAP1 ODU Load	1	RSAP1 ODUD Load/CNF Store	2	RSAP1 REQ Load	3	RSAP0 ODU Load	4	RSAP0 ODUD Load/CNF Store	5	RSAP0 REQ Load	6	ISAP2 IDU Store	7	ISAP2 IDUD Store	8	ISAP2 PSP Load	9	ISAP1 IDU Store	A	ISAP1 IDUD Store	B	ISAP1 PSP Load	C	ISAP0 IDU Store	D	ISAP0 IDUD Store	E	ISAP0 PSP Load	F	PTR RAM Load/Store
AB_AD[31:28]	Transaction Type																																				
0	RSAP1 ODU Load																																				
1	RSAP1 ODUD Load/CNF Store																																				
2	RSAP1 REQ Load																																				
3	RSAP0 ODU Load																																				
4	RSAP0 ODUD Load/CNF Store																																				
5	RSAP0 REQ Load																																				
6	ISAP2 IDU Store																																				
7	ISAP2 IDUD Store																																				
8	ISAP2 PSP Load																																				
9	ISAP1 IDU Store																																				
A	ISAP1 IDUD Store																																				
B	ISAP1 PSP Load																																				
C	ISAP0 IDU Store																																				
D	ISAP0 IDUD Store																																				
E	ISAP0 PSP Load																																				
F	PTR RAM Load/Store																																				
AB_A4-2	3-5	O	ABus Burst Address: These TRI-STATE signals contain the word address during burst-mode accesses. They are driven from Tpa to the last Td state, negated in the following Tr state, then released. Note that the address presented allows external pipelining for optimum memory timing.																																		

6.0 Signal Descriptions (Continued)

Bus Control:

Symbol	Pin #	I/O	Description																																				
AB_AS	159	O	ABus Address Strobe: When asserted, This TRI-STATE signal indicates that data on AB_AD is valid. When this signal is inactive and AB_ACK is asserted, the next cycle is a Tr state, in which the bus arbiter can sample all bus requests, then issue a bus grant in the following cycle.																																				
AB_RW	6	O	ABus Read/Write: This TRI-STATE signal determines the current direction of an ABus access.																																				
AB_DEN	7	O	ABus Data Enable: This TRI-STATE signal indicates that data on AB_AD31-0 is valid.																																				
AB_SIZ2-0	10-12	O	<p>ABus Size: These TRI-STATE signals indicate the size of the transfer on AB_AD31-0, encoded as follows:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>AB_SIZ2</th> <th>AB_SIZ1</th> <th>AB_SIZ0</th> <th>Transfer Size</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>4 Bytes</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Reserved</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Reserved</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>16 Bytes</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>32 Bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Reserved</td> </tr> </tbody> </table>	AB_SIZ2	AB_SIZ1	AB_SIZ0	Transfer Size	0	0	0	4 Bytes	0	0	1	Reserved	0	1	0	Reserved	0	1	1	Reserved	1	0	0	16 Bytes	1	0	1	32 Bytes	1	1	0	Reserved	1	1	1	Reserved
AB_SIZ2	AB_SIZ1	AB_SIZ0	Transfer Size																																				
0	0	0	4 Bytes																																				
0	0	1	Reserved																																				
0	1	0	Reserved																																				
0	1	1	Reserved																																				
1	0	0	16 Bytes																																				
1	0	1	32 Bytes																																				
1	1	0	Reserved																																				
1	1	1	Reserved																																				
AB_ACK	1	I	ABus Acknowledge: Indicates a bus slave's response to a bus master. The meaning of this signal depends on the state of ABus Error (AB_ERR), as described below.																																				
AB_ERR	2	I	<p>ABus Error: This signal is asserted by a bus slave to cause a transaction retry or transaction abort. Together with AB_ACK, the encoding is as follows:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>AB_ACK</th> <th>AB_ERR</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>Insert Wait States</td> </tr> <tr> <td>1</td> <td>0</td> <td>Bus Error</td> </tr> <tr> <td>0</td> <td>0</td> <td>Transaction Retry</td> </tr> <tr> <td>0</td> <td>1</td> <td>Acknowledge</td> </tr> </tbody> </table>	AB_ACK	AB_ERR	Definition	1	1	Insert Wait States	1	0	Bus Error	0	0	Transaction Retry	0	1	Acknowledge																					
AB_ACK	AB_ERR	Definition																																					
1	1	Insert Wait States																																					
1	0	Bus Error																																					
0	0	Transaction Retry																																					
0	1	Acknowledge																																					

Bus Arbitration:

Symbol	Pin #	I/O	Description
AB_BR	13	O	ABus Bus Request: This signal is used by a bus master to request use of the ABus.
AB_BG	14	I	ABus Bus Grant: This signal is asserted by external bus arbitration logic to grant use of the ABus to the BSI device. When AB_BG is deasserted, the BSI device completes the current transaction and releases the bus. If AB_BG is asserted at the start of a transaction (Tbr), the BSI device will run a transaction.
AB_CLK	160	I	ABus Clock: All ABus operations are synchronized to the rising edge of AB_CLK.

6.0 Signal Descriptions (Continued)**6.6 ELECTRICAL INTERFACE**

Symbol	Pin #	I/O	Description
LBC5, 3, 1	43-41	I	Local Byte Clock: 12.5 MHz clock with a 60/40 duty-cycle. Generated by CDD.
V _{CC} [13]	8, 39, 56 74, 81, 86, 104, 114, 125, 135, 141, 147, 157		Positive Power Supply: 5V, $\pm 10\%$ relative to GND.
GND[13]	9, 40, 57 75, 82, 87, 105, 115, 124, 134, 140, 146, 156		Power Supply Return.
GND	44, 47		Must be grounded.
NC	45, 46, 48		No Connect: Must be left unconnected.



Section 3
Development Support



Section 3 Contents

DP83200EB FDDI AT Evaluation Kit	3-3
DP83200SMT XLNT Manager FDDI Station Management (SMT) Software Support Package .	3-15



DP83200EB FDDI AT Evaluation Kit

General Description

The DP83200EB FDDI is a complete design/evaluation kit (using an AT or compatible platform), that includes hardware, software and application documentation, to implement a single node compliant with an ANSI X3T9.5 FDDI network. The kit has been designed to demonstrate the capabilities of National Semiconductor's FDDI chip set.

It contains a Link card and MAC card, that together implement one FDDI Single Attach node.

The Evaluation Boards allow evaluation of the many capabilities of the chip set and serve as an educational tool for customers designing products with the FDDI chip set. High performance as a goal was sacrificed at the expense of simplicity and accessibility. There are many laboratories around the world with a spare IBM® PC® or compatible.

These boards allow users to experiment and gain experience with the FDDI chip set in order to unleash its capabilities in their own products.

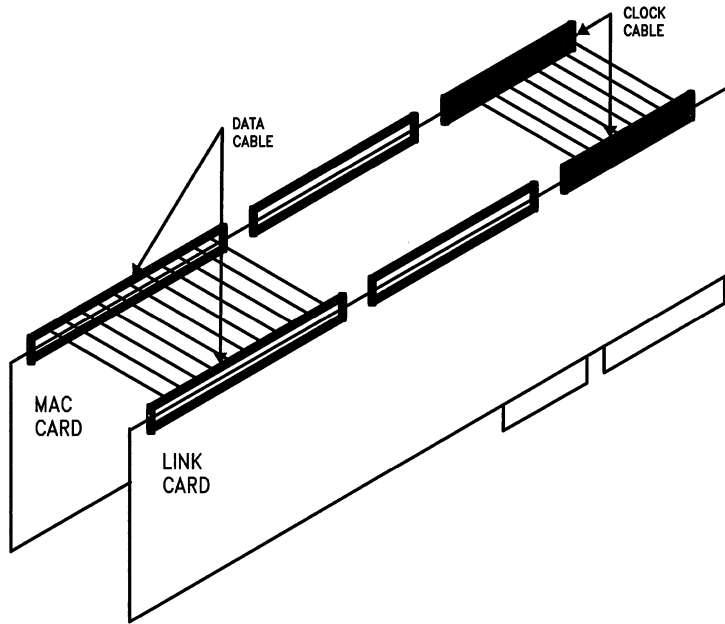
The DP83200EB can be combined with a DP83200EK Kit to create a dual attach station. The DP83200EK Kit contains the additional Link card (PHY Layer) and appropriate cables.

Features

- System modularity supports single attachment or dual attachment
- Utilizes a PC-AT® compatible form factor
- Built-in diagnostic capability for fault detection
- Supports an external optical bypass switch
- Supports asynchronous and synchronous transmission classes
- PAL based buffer management
- Supported by demonstration and diagnostic software
- Board schematics

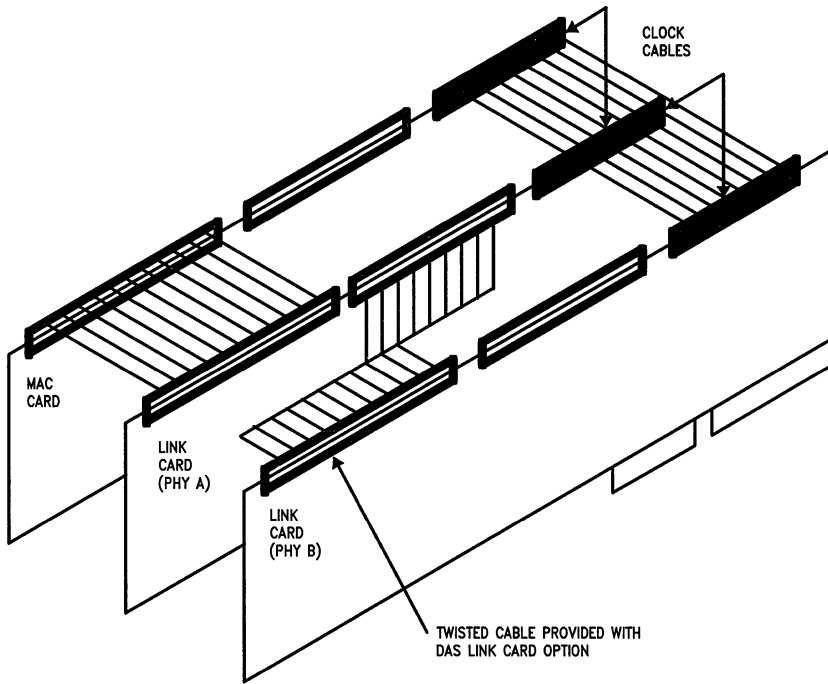


TL/F/11122-1



SAS Configuration

TL/F/11122-2



DAS Configuration

TL/F/11122-3

FIGURE 1. Single attach and optional dual attach configuration

1.0 Link Card

1.1 Link Card Description

The Link Card is intended for evaluation of the following three National Semiconductor devices which implement the FDDI Physical Layer and clock distribution.

- DP83255 Physical Layer Controller (PLAYER™ device)
- DP83241 Clock Distribution Device (CDD™ device)
- DP83231 Clock Recovery Device (CRD™ device)

The design goal of the Link Card was to allow the user to exercise the Physical Layer devices and CDD device. The Link Card can be used in tandem with the DP83291EB MAC Card (see Section 2.3). A Link Card connected to a single MAC Card implements a single attachment station. Dual attachment stations require two Link Cards. The Link Card requires a PC-AT compatible machine which is a readily available platform capable of supporting an FDDI application.

1.2 Link Card Features

The Link Card offers many features to provide a flexible and convenient evaluation platform:

- Utilizes the National FDDI Chip Set
 - DP83255 PLAYER Device
 - DP83241 CDD Device
 - DP83231 CRD Device

- System modularity supports single attachment or dual attachment configurations.
- Utilizes a PC-AT compatible form factor
- Built-in diagnostic capability for fault detection
- Supports an external optical bypass switch
- Power consumption is 1.5 amps typical per Link Card

2.0 Link Card System Description

2.1 Block Diagram Description

The Link Card block diagram is composed of the following seven blocks:

1. AT Bus Interface
2. Clock Bus Interface
3. Clock Distribution Device (CDD device)
4. Clock Recovery Device (CRD device)
5. Link Bus Interface
6. Physical Layer Controller (PLAYER device)
7. Transceiver Interface

Figure 2 is a detailed representation of the block diagram.

2.0 Link Card System Description (Continued)

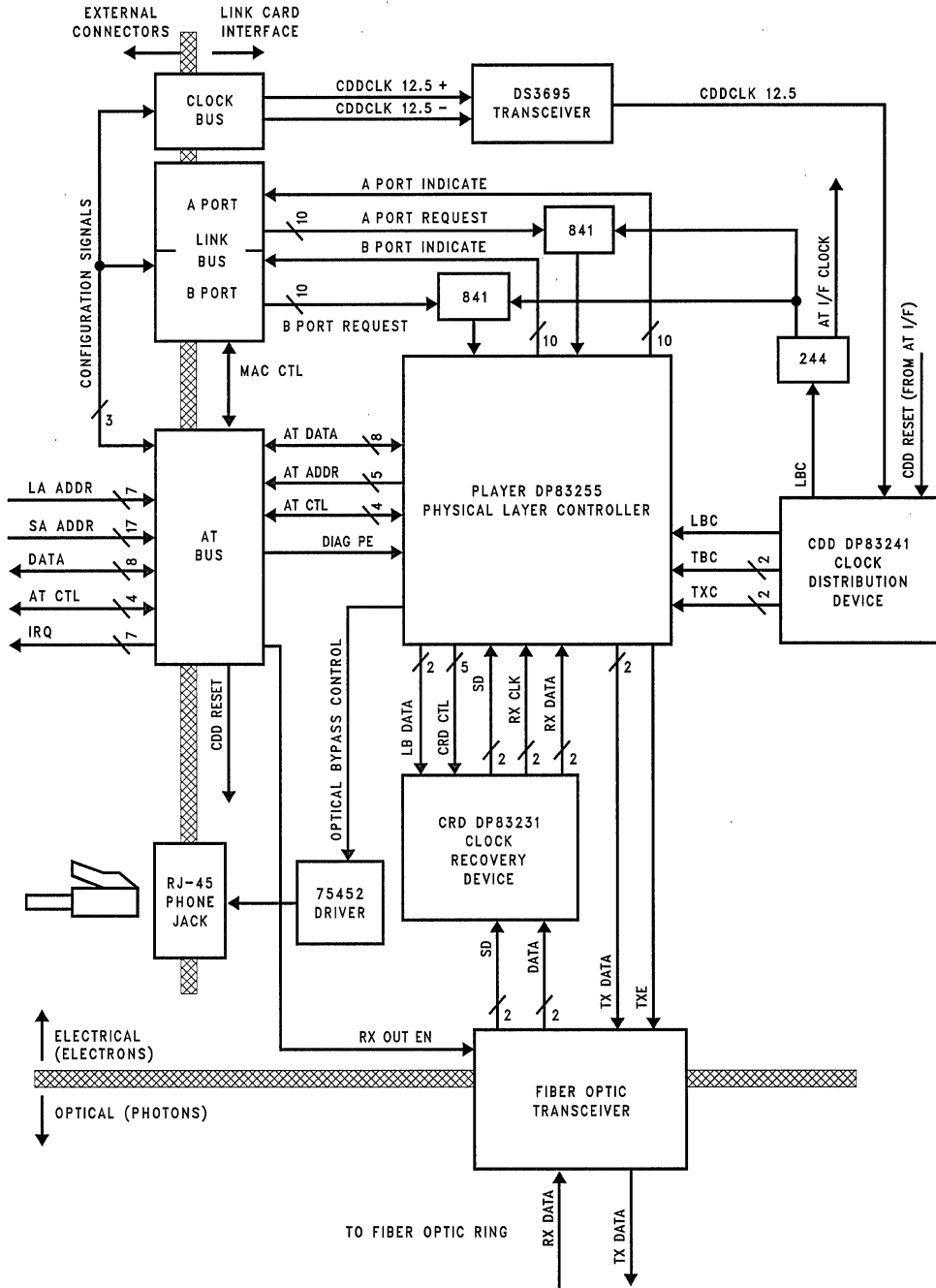


FIGURE 2. Link Card Block Diagram

TL/F/11122-4

2.0 Link Card System Description (Continued)

2.2 AT Interface Block

The function of the AT Interface Block is to interface the Link Card with the AT host. This block features a full 24-bit address bus for flexible Link Card memory map placement. The data bus is 8 bits wide which is adequate for this demo platform. Bits 8 through 15 are not used on the base Link Card, but they have been tapped to test points on the board. The test points are included in the event that an application requires a 16-bit data bus. In addition to the address and data buses, seven AT bus interrupts and the necessary control signals are included. All address and data signal lines are buffered with independent parity generation supplied for the data bus.

The AT bus block is the *sole* power supply for the Link Card. The address decoding scheme is accomplished with generic array logic devices (GALs). Equations for each of the four GAL devices are included in Appendix G of the DP83290EB FDDI Physical Layer Evaluation Board User's Guide.

Beyond these basic functions, the AT Interface offers a number of modes such as autoconfiguration, base register area select, and memory map configuration.

2.3 Clock Bus Block

The Clock Bus Block is included in the Link Card design to provide a physical bus among all Link and MAC Cards that form a station. The construction of the bus is a twenty pin ribbon cable capable of supporting 9 signals. Each signal is surrounded on either side by a ground line to reduce crosstalk.

2.4 CDD Device Block

The Clock Distribution Device is a clock generation and distribution device intended for use in FDDI networks. The device provides the complete set of clocks required to convert byte wide data to serial format for fiber medium transmission and to move byte wide data between the PLAYER and BMAC devices in various station configurations. 12.5 MHz and 125 MHz differential ECL clocks are generated for the conversion of data to serial format and 12.5 MHz and 125 MHz TTL clocks are generated for the byte wide data transfers.

2.5 CRD Device Block

The Clock Recovery Device has been designed for use in this FDDI implementation. The device receives serial data from a Fiber Optic Receiver (FORX) in differential ECL NRZI 4B/5B group code format and outputs resynchronized NRZI received data and a 125 MHz received clock in differential ECL format for use by the PLAYER device.

2.6 Link Bus Block

The function of the Link Bus is to provide a data path between the Link and MAC Cards that form an FDDI station. Each connection contains two 10-bit data buses (Indicate and Request) and station configuration signals. The pinout of the Link Bus has been designed to allow the user to build Single Attachment and Dual Attachment/Single MAC configurations. To build one of these configurations, the user must simply connect the cabling in the manner shown in Appendix E of the User's Guide.

Every other wire in the Link Bus is grounded to insure data integrity. This cabling scheme has been tested for resistance to data corruption induced by crosstalk.

2.7 PLAYER Device Block

The Physical Layer Controller is a part of National Semiconductor's FDDI Chip Set. It implements one Physical Layer entity as defined by the ANSI X3T9.5 PHY standard. The PLAYER device performs the 4B/5B encoding and decoding, serialization and deserialization of data, repeat filter, and line state control and detection. It also contains a configuration switch. The PLAYER device supports many types of station configurations as allowed by the standard.

Although tailored to the FDDI specification, the PLAYER device is also well suited for use in high speed point-to-point communication links over optical fibers and coaxial cable.

2.8 Transceiver Block

The transceiver block consists of two parts: fiber optic receiver and fiber optic transmitter. The Link Card supports the following FDDI optical transceiver modules:

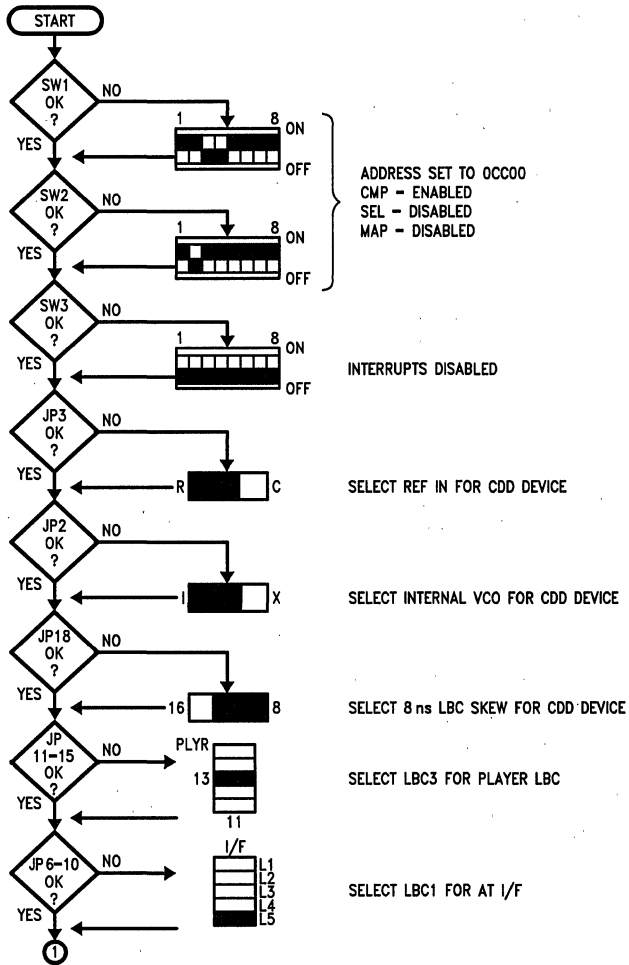
- AT&T ODL 125 Lightwave Data Links
- Sumitomo DM-742 1300nm Data Link
- Any Transceiver pair which supports the
 - AT&T footprint 2.1.8.1 pin format
 - composed of 2 independent 16-pin DIP (footprints)

See Appendix A of the User's Guide for a detailed footprint description.

2.0 Link Card System Description (Continued)

2.9 Installation

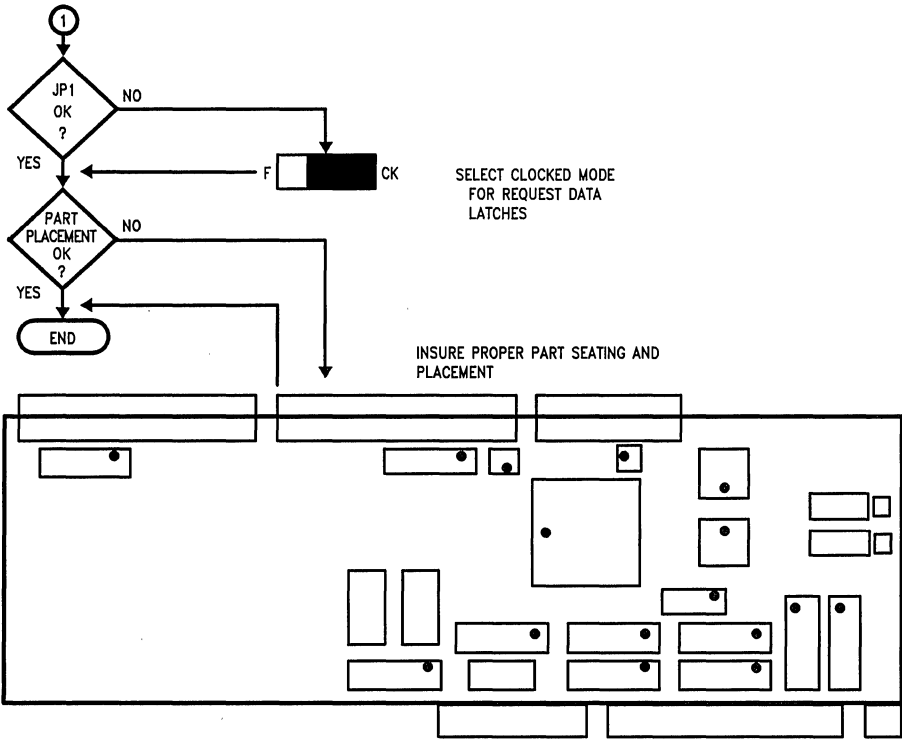
A.1.1 Setup



TL/F/11122-5

2.0 Link Card System Description (Continued)

A.1.1 Setup (Continued)



TL/F/11122-6

3.0 MAC Card

3.1 MAC CARD DESCRIPTION

The DP83291EB FDDI MAC Layer Evaluation Board is a PC-AT compatible board that implements the MAC Layer functions of the FDDI standard. The Board utilizes the National Semiconductor DP83261 BMAC™ device along with PAL®-based Buffer Management Logic to implement a simple MAC Layer.

3.2 MAC CARD FEATURES

The MAC card offers many features on a convenient platform:

- PC-AT compatible full size card
- Dual ported memory interface—full duplex data path
- Interfaces to link cards for DAS or SAS configurations
- Supported by demonstration software
- Utilizes DP83261 BMAC device
- Full network statistics
- Supports asynchronous and synchronous transmission classes
- PAL based buffer management

3.3 MAC CARD SYSTEM DESCRIPTION

The MAC Evaluation Board is designed to perform simple transmission and reception scenarios. One 8k and 8 Static RAM is used for Transmission and another is used for Reception and Status. See *Figure 3* for a detailed block diagram.

On Transmission, the Transmission Counter is used to address the Tx_RAM where the frames to be transmitted are written. The Transmit Sequencer controls the sequencing of frames across the BMAC device's MAC Request Interface.

On Reception, the Receive Sequencer controls the sequencing of frames across the BMAC device's MAC indicate interface. Transmit and Receive Status is stored and multiplexed into the Rx_RAM between frames. The Receive Counter is used to address the Rx_RAM. The Receive Counter is under the control of the Receive Sequencer. The Copy PAL monitors the addressing information across the BMAC device and determines whether to continue to copy frames. This is signaled to the Receive Sequencer which then adjusts the Receive Counter accordingly.

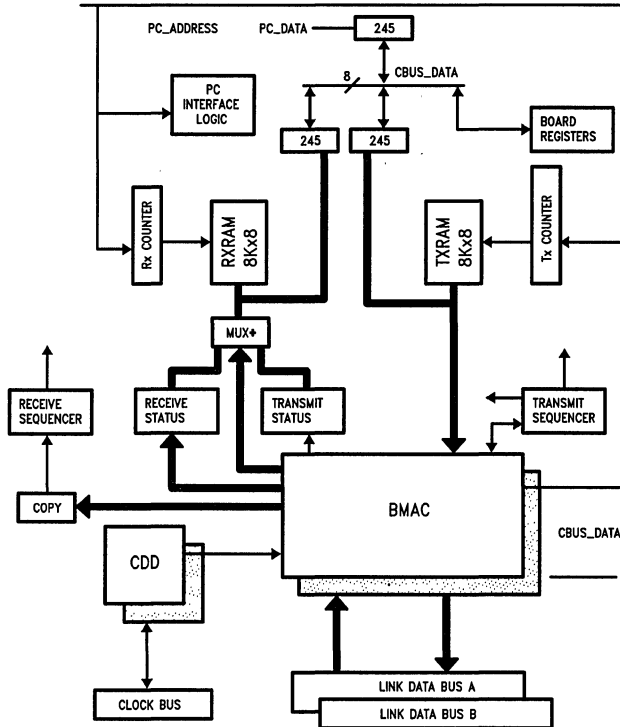


FIGURE 3. MAC Card Block Diagram

TL/F/11122-7

3.0 MAC Card (Continued)

The Tx_RAM contains either one maximum size frame or up to 16 frames of 512 bytes or less. The Rx_RAM may be filled with up to 8 frames.

The DP8570A is used as a timer on this board and provides support for Station Management. Its registers are accessed via the PC interface and is memory mapped.

The PC interface provides access to the Transmit and Receive RAM in addition to the Board Registers and BMAC device. The Board Registers include a Mode, Function and Status Register.

3.4 ADDRESS MAPPING

PC BUS INTERFACE

Board Address Mapping

The Evaluation Board Control Bus is mapped into a 64k segment within the lowest 1M of the PC address space. The 64k offset is selected (by a jumper) as shown below:

- Sel = 0 offset is C0000
- Sel = 1 offset is D0000

64k Segment Mapping

The 64k segment reserved for the evaluation board is divided as shown below:

- 0000-3FFF Used for control registers (See Figure 3)
- 4000-7FFF reserved
- 8000-9FFF used to access 8k Tx_RAM
- A000-BFFF shadow to Tx_RAM
- C000-DFFF used to access 8k Rx_RAM
- E000-FFFF shadow of Rx_RAM

Board Register Address Mapping

The address space used for control registers is divided into 512 byte pages for each PHY or MAC. The MAC is selectable as either MAC 0 or MAC 1.

- 0000-01FF MAC0
- 0200-03FF MAC1

- 0400-0FFF reserved for future use
- 1000-3FFF reserved for future use

See Figure 4.

MAC Registers

- 0000-00FF BMAC Device Registers
- 0100-01FF Board Registers

BMAC Device Registers

The BMAC Device Registers are mapped directly into the 64k segment of the address space as defined in the BMAC Device Datasheet.

MAC Evaluation Board Registers

The board registers are mapped into the 64k segment of the address space as:

- 0100 Mode Register
- 0140 Function Register
- 0180 Status Register
- 01A0 Timer Registers
- 01C0 Board Reset

Note: For MAC1 add 200h to each address.

BMAC 0 REGISTERS	0000-00FF
BOARD 0 REGISTERS	00FF-01FF
BMAC 1 REGISTERS	0200-02FF
BOARD 1 REGISTERS	0300-03FF
RESERVED	0400-0FFF
RESERVED	1000-3FFF
RESERVED	4000-7FFF
TX RAM	8000-9FFF
SHADOW TX RAM	A000-BFFF
RX RAM	C000-DFFF
SHADOW RX RAM	E000-FFFF

FIGURE 4. Board Address Map

3.0 MAC Card (Continued)

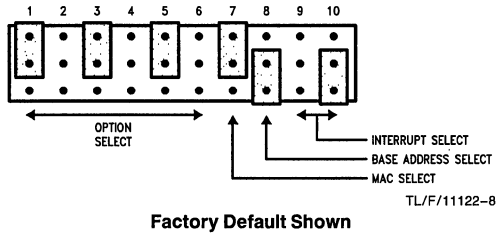
3.5 MAC Board Installation

The MAC Card requires a full length slot. It can be installed in either an AT or XT slot. There are several options that can be programmed via jumpers provided on the MAC Card. The position of the jumpers on the board are shown in Figure 5.

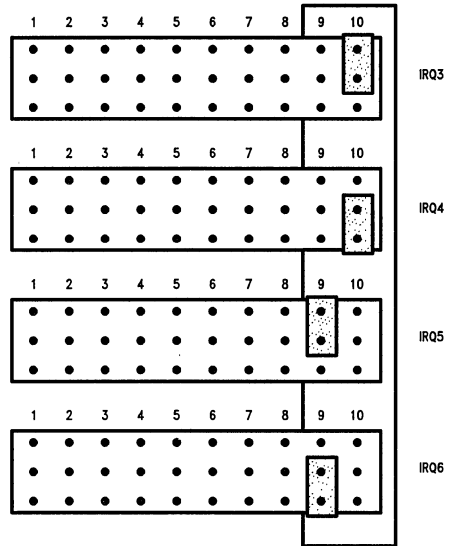
JUMPER SETTINGS

J1: Base Address, MAC Select, Interrupt and Option Selection

J1 is used to select the Base Address of the MAC Card, MAC Number, interrupt to be used for the interconnected MAC and Link Cards and Option Selections on the BMAC device. The possible settings and factory defaults follow.

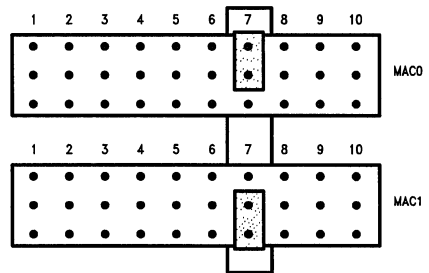


Interrupts



Factory Default = IRQ4

MAC Select



Factory Default = MAC0

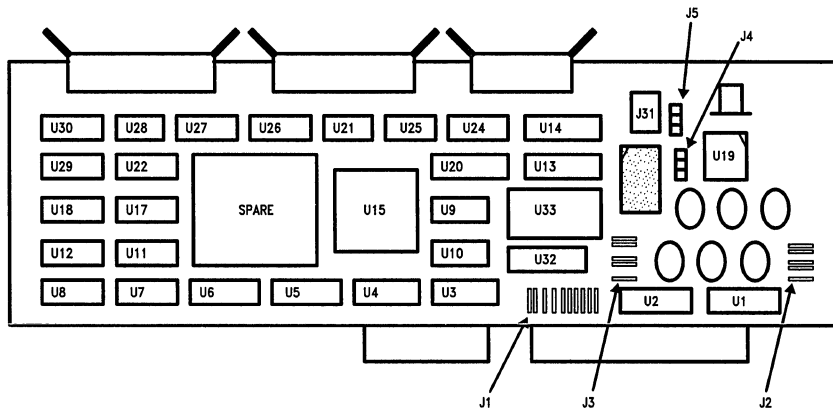
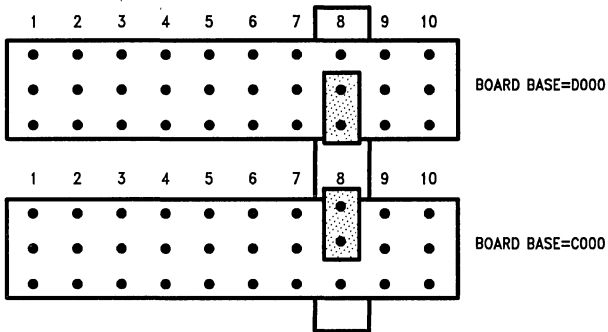


FIGURE 5. MAC Card Jumper Locations

3.0 MAC Card (Continued)

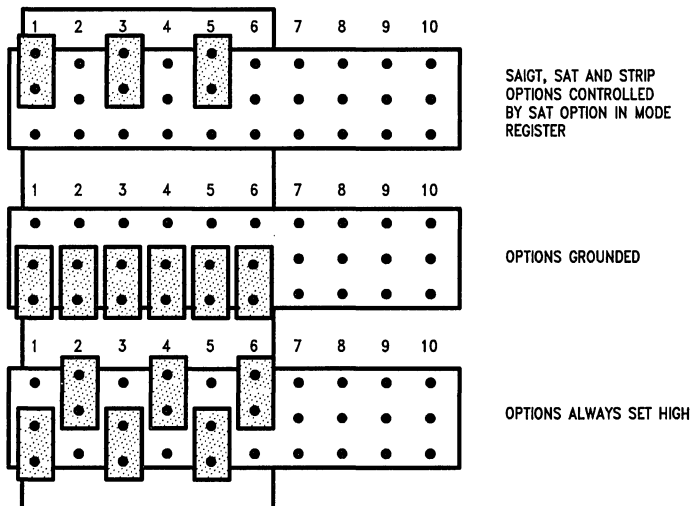
Base Address Select



Factory Default = D000

TL/F/11122-12

Options Selection



Factory Default = Options Controlled by SAT Option in Board Mode Register

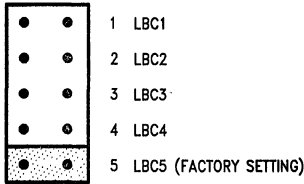
TL/F/11122-13

3.0 MAC Card (Continued)

J2: Latch Clock Select

J2 is used to select which phase of the LBC to transfer data between the Link and MAC Cards. This is used to optimize set up and hold time on data transfers between the BMAC and PLAYER devices when using a bus. **This jumper should always be left at the default setting (LBC5).**

J2: Latch Clock Select

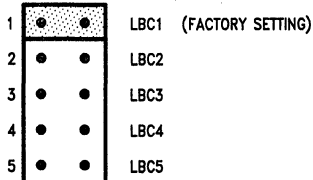


TL/F/11122-14

J3: CDD Device Feedback Select

J3 is used to select which phase of the LBC that the CDD device will use in its feedback loop. **This jumper should be left at the default setting (LBC1).**

J3: CDD Device Feedback Select

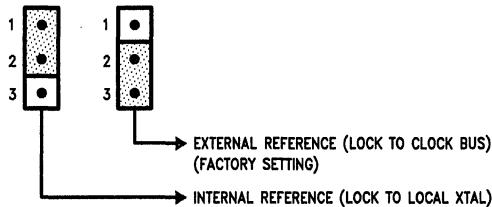


TL/F/11122-15

J4: Reference Select

J4 is used to select either the clock on the clock bus or the local crystal oscillator as a reference. In order to provide less skew between the Link and the MAC Cards, the CDD device is locked to the Clock Bus signal. **This jumper should be left at the default setting.**

J4 Reference Select

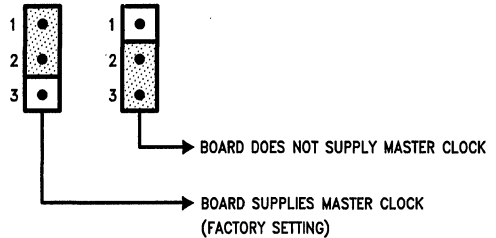


TL/F/11122-16

J5: Master Clock Select

J5 programs whether or not this board provides the master clock for the node. Typically in a DAS or SAS configuration, only a single MAC is used. The MAC card will provide the clock for the entire system. If two MACs are present in the node, only one of the MAC Boards should provide a clock. **This jumper should only be programmed when configuring Dual MAC systems.**

J5: Master Clock Select



TL/F/11122-17

INSTALLING THE BOARD WITH A LINK CARD

The MAC Card should be inserted into the PC-AT or XT slot along with the Link Card. Make sure that the board is inserted into the connectors accurately. The next step is to connect the cabling between the MAC Card and the Link Card. Two configurations are possible, Single Attach Station (SAS) and a Dual Attach Station (DAS).

SAS Configuration

To implement a SAS Configuration a MAC Card and a Link Card are inserted into two slots on the AT. Two cables are required. First connect the 20-Pin Clock Cable into the 20 Pin header on the top of both the Link Card and MAC Card. Then connect the 50-Pin Cable between the two "A" port connectors on the Cards. This is the connector farthest from the back of the AT chassis (see Figure 5). Make sure the cables are properly engaged with the pins by pressing down on the top of the cable with your thumbs. Once the cables have been attached inspect the cards to make sure that they are straight in the connectors. Pressing in the cables has a tendency to skew the cards in the slots.

DAS Configuration

To implement a DAS Configuration an additional Link Card is required. (Two Link Cards + 1 MAC Card implements a DAS). Insert a MAC Card and two Link Cards adjacent to each other as shown in Figure 1. Connect the Cable between the Link Cards as shown, then connect the Clock Bus Cable and the final Data Cable between the Link and MAC Cards as shown. Note that the clock cable must attach to all three boards. Make sure the cables are properly engaged with the pins on the cards. Also confirm that the cards are straight in the card cage after connecting the cables.

DP83200SMT XLNT Manager™ FDDI Station Management (SMT) Software Support Package

General Description

The XLNT Manager™ Software Package completely implements all required and most optional FDDI SMT Protocol functions. It comes complete with well documented source code written in the C programming language and an easy to use Integration Guide. The source code is modular so that it can be partitioned over multiple processors and can be used in all FDDI applications including concentrators, and dual and single attach stations. The porting guide includes step by step instructions how to port the software to your hardware platform. XLNT Manager FDDI SMT Software is a product of XLNT Design Inc., and is made available to National Semiconductor FDDI customers through a sublicensing agreement. Technical support is provided by National Semiconductor's fully trained FDDI Applications team. XLNT Manager FDDI SMT Software combined with National Semiconductor's FDDI chip set provides a complete FDDI solution.

The software can easily be integrated into an operating environment using minimal hardware and system support from the operating environment. The design of the software also allows for rapid updating to implement changes to the SMT Standard. Custom Management services can interface with the SMT software through a standard, well defined, soft-

ware interface. An integration sequence example is included.

Features

- Implements all required and most optional ANSI X3T9.5 Station Management Protocol functions
- Complete portability of the code to most hardware platforms
- Direct interface with the FDDI Chip Set for compact and efficient implementation
- Flexible architecture to allow code partitioning for single and multiple processors
- Standard interface between SMT software and custom management applications
- Well documented source code (C Language)
- Supports single and dual attach stations and concentrators
- Operating System Independent
- Includes easy-to-use porting guide
- Free updates through 1991

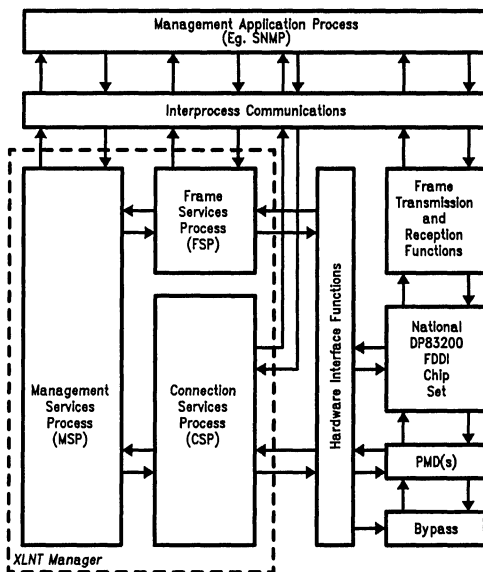


FIGURE 1. Station Management (SMT) Software Block Diagram

TL/F/11121-1

Table of Contents

- 1.0 OVERVIEW
- 2.0 FUNCTIONAL DESCRIPTION
- 3.0 INTEGRATION PROCEDURE
- 4.0 SUBLICENSE AGREEMENT AND MAINTENANCE
- 5.0 TECHNICAL SUPPORT

1.0 Overview

One of the key areas in the FDDI standardization process has been Station Management (SMT). The SMT document provides the guidelines and protocols which can be used to manage an FDDI network.

To ensure interoperability in a multi-vendor environment, some of the protocols described in the SMT document are mandatory. However, to facilitate the diverse network environments envisioned for FDDI, many of the protocols described are optional. Implementation of the optional protocols will depend upon the application need.

The basic SMT functions required are:

- Fault management for high network availability
- Reliable error detection and recovery
- Access to networked resources
- Fast and reliable connection management procedure
- Management for multi-vendor networks
- Access to individual station information
- Flexible network configuration

2.0 Functional Description

The XLNT Manager Software Package completely implements all required and most optional FDDI SMT Protocol Functions. The software package consists of three main functional blocks as shown in *Figure 1*. The three main functional blocks are 1) Management Services Process (MSP), 2) the Frame Services Process (FSP) and 3) the Connection Services Processes (CSP).

2.1 MANAGEMENT SERVICES PROCESS

The Management Services Process (MSP) processes information in the Management Information Base (MIB). The MIB is an information database which is defined by the FDDI SMT Standard. It includes information about the status and performance of a station. The information is collected from the hardware and other processes and stored in a central database. The Management Information Base (MIB) can be accessed by a Management Application Process like the Simple Network Management Protocol (SNMP) or by a remote station over the network. The Management Services Process maintains the MIB and provides a uniform interface to the Management Application Process, the Frame Services Process (FSP) and the Connection Services Process (CSP).

2.2 FRAME SERVICES PROCESS

The Frame Services Process (FSP) implements the framed based services defined by the FDDI SMT Protocol. The FDDI SMT Protocol defines several different frame types which are used to convey status and control information between stations on the network. For example, Neighbor Information Frames (NIFs) convey information about the station to its neighbor. This information can be used in the

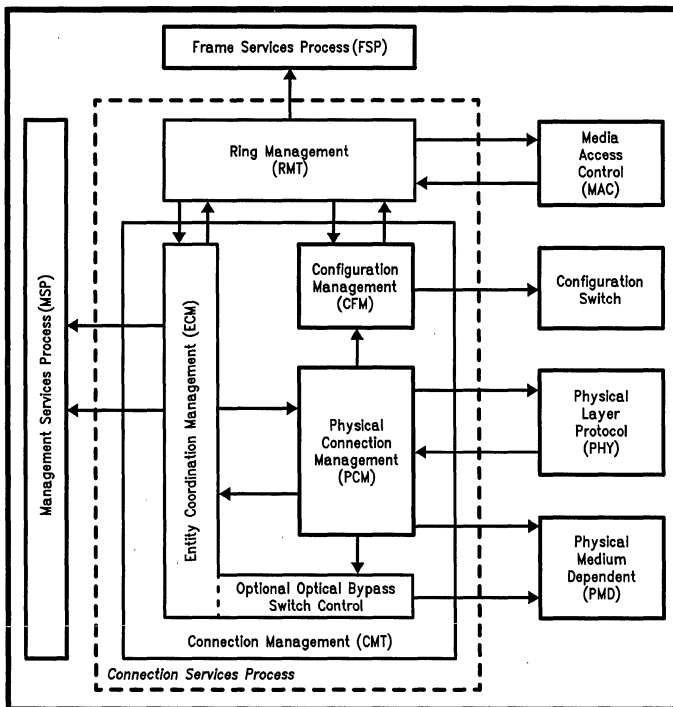


FIGURE 2. Connection Services Process Block Diagram

TL/F/11121-2

2.0 Functional Description (Continued)

development of a ring map. The Frame Services Process responds to requests (from the Management Application Process) to send frames. It also sends frames as required by SMT and responds to SMT frames from other stations. For more information on FDDI frame based services refer to the ANSI X3T9.5 FDDI SMT Documents.

2.3 CONNECTION SERVICES PROCESS

The Connection Services Process implements the Connection Management (CMT) and Ring Management (RMT) functions of the FDDI SMT Protocol. The Connection Management (CMT) Protocol consists of three functional blocks as shown in *Figure 2*.

The three main functional blocks of the Connection Management (CMT) protocol are the Configuration Management (CFM) block, the Physical Connection Management (PCM) block and the Entity Co-ordination Management (ECM) block.

The Physical Connection Management (PCM) block is responsible for establishing a photonic connection between two neighboring stations, signaling configuration information and monitoring the link errors between stations. The PCM process interacts with the National DP83251/55 which performs the physical signaling process and error monitoring functions.

The Configuration Management process controls the data paths between MAC and PHY devices. This process interacts with the configuration switch in the National DP83251/55 PLAYER Device. It also interacts with other SMT processes.

The Entity Co-ordination Management process controls the optional optical by-pass switch and acts as a well defined interface to the Management Services Process.

The primary function of the Ring Management process is to detect duplicate addresses that would inhibit the ring from becoming operational, and other network failures.

Implementation

The effective implementation of Station Management software in a design is tied directly to the ability of the FDDI chip set silicon to provide the appropriate hooks and management information. The DP83200 FDDI Chip Set provides the essential support for FDDI Station Management functions.

The PLAYER Device and the BMAC Device support the Connection Management services and the Ring Management service respectively. The BSI Device provides separate Station Management and data frame channels for the maximum support of the SMT Frame Services. See Application Note AN-726 "Station Management Simplified" for more information.

3.0 Integration Procedure

The following description presents a possible sequence for integration of the XLNT Manager SMT software package onto your system.

The steps described will, in general, have to be done to integrate the SMT software package into a system, but need not happen in the exact sequence given below.

- Hardware test
- Write network interface functions
- Define partitioning of SMT function
- Customize headers and initialization function
- Provide control interface
- Compile, link and test
- Integrate Management Application Process

3.1 HARDWARE TEST

It is imperative that the hardware platform is debugged before starting the XLNT Manager Software integration. Proper data flow and timer operation is required before starting the integration procedure. The time required to port the software could be greatly extended if the hardware platform is not fully debugged.

3.2 INTERFACE FUNCTIONS

A detailed description of how to write interface functions are described in the XLNT Manager Integration Guide. A brief explanation, with examples, follows here.

The SMT software uses a number of interface functions that deal directly with the FDDI chip set hardware, provide timer service to the FDDI functions and support transmission and reception of SMT frames. Before writing and testing the hardware interface, it is important to fully test the hardware platform.

Example system functions:

```
uint32 ReadSystemTimer (void)
void SetSystemTimer (uint32 interval)
uint StationPathTest (void)
void BypassRequest (uint bypassState)
void ConfigureStation (uint CFM__state)
```

Example PHY interface functions:

```
void TransmitPMD (uint phyID, uint state)
uint16 ReadLineState (uint phyID)
void SendLineState (uint phyID, LineState ls)
```

Example MAC interface functions:

```
void MACResetRequest (uint macID)
void SetBeacon (uint macID, char beaconType,
LongAddr *DA, uint infoLength, char *beaconInfo)
void BeaconRequest (uint macID)
```

Write Interface Code

The majority of the integrator supplied code is in the interface routines. The interface routines are those that directly deal with hardware. The number of interface functions and macros for SMT 6.2 is expected to be approximately 40. Over four thousand lines of prototype source code is supplied for these functions to aid the implementor.

The complexity of these functions varies significantly and there is little correlation of the complexity of the function with the amount of code. The most complex and important implementor code is for frame transmission and reception. This code is important because it provides the basic function of an FDDI attachment, moving data. SMT requires

3.0 Integration Procedure (Continued)

comparatively little frame transmission and reception and performance of the SMT frame processes will not be critical in most devices. An implementor supplied function that can be fairly complex is the path test, though this is a logical subset of a power on self test for the board. The majority of the functions are simple and direct. These simple functions include transmitting a line state, managing MAC frame counters, and setting interrupt masks.

A few implementor provided functions are affected by the options in the SMT Standard. Two examples are the method to be used for scrubbing, and generation of data for link confidence testing. If the implementor desires to use the MAC in these cases, the interface code complexity will be increased, since the MAC will need to be set up for generation of the desired data.

Many of these interface functions can be used in initial testing of the board. Planning ahead and using the functions as part of the basic hardware test plan can reduce the total development time.

Example Section of Interface Function Code

```

ConfigureStation (CFM_state)
    uint CFMstate;
/*****
Function: Set the station hardware to the specified CFM state.
Parameters: CFM_state =state for hardware configuration as
            defined in the CFM state machine.
Input:     Uses stationData.
Output:
Return:    No value returned.
Notes:     None.
Modification History:
*****/
{
uint16 mac1,          /*MAC 1 mode register*/
mac2,                /*MAC 2 mode register*/
phyA,                /*PHY A control register*/
phyB,                /*PHY B control register*/
mac1Ints,            /*MAC 1 interrupt mask*/
mac2Ints,            /*MAC 2 interrupt mask*/
ProcState pState; /*interrupt state*/
/*
 * Select type of station to configure.
 */
switch (stationData.cspType)
{
case SMSAS_TYPE:
    .
    .
    .
case SMDAS_TYPE:
    /*
     * Read current macMode.
     */
    mac1= ReadReg (MAC_MODE_REG);
    /*Select state */
    switch (CFM_state)
    {
case SC_ISOLATED:
    /*
     * A:PH_DATA.indication to A:PH_DATA.request
     * B:PH_DATA.indication to B:PH_DATA.request
     * MAC connections are unspecified
     */
    MChangeBits (mac1 MMODE_BITS|MSELRA_BITS,
                MMODE_Initialize|MSELRB);
    phyA = PHY_TA|PHY_Repeat;
    phyB = PHY_TA|PHY_Repeat;
    break;

```

3.0 Integration Procedure (Continued)

Example Section of Interface Function Code (Continued)

```

case SC_WRAP_A:
  /*
   * A:PH_DATA.indication to l:PH_DATA.indication
   * l:PH_DATA.request to A:PH_DATA.request
   * B:PH_DATA.indication to B:PH_DATA.request
   */
  MChangeBits (macl MMODE_BITS|MSELRA_BITS,
               MMODE_On_Line|MSELRA) :
  phya = PHY_TA|PHY_Through;
  phyb = PHY_TA|PHY_Repeat;
  break;
case SC_WRAP_B:
  .
  .
  .

```

The XLNT Manager Integration Guide includes detailed descriptions of all the functions.

3.3 PARTITION OF SMT FUNCTION

The partitioning of SMT software is quite simple. The first step is to determine the partitioning of SMT function. In most implementations, all SMT code will run on the same processor. If this is the case, there is no further effort associated with this task. If the code is to be partitioned between processors, then communication routines need to be written to transfer SMT signals between the different environments. Defining the partition of the SMT function can be simple for an adaptor card and less straight forward in a concentrator. Implementors can partition the SMT functions between different processors. This includes the ability to separate Connection Services from the rest of SMT. Those implementations partitioning SMT software onto multiple processors must provide routines for communication between the SMT functions in the separate environments.

3.4 CUSTOMIZATION OF HEADERS AND INITIALIZATION FUNCTIONS

FDDI includes a large number of possible station types with optional capability. The SMT software provides options supported by the code, and the SMT Standard. Examples, of this customization include: number of MACs, number of PHYs, default policies, and bypass relay present.

The next step is to customize header files. This includes general customization like declaring whether the processor uses big endian or little endian byte ordering.

Certain attributes need to be modified for differences in station type, (e.g., SM-DAS, DM-DAS, bridge, etc). In some cases, the implementation may be flexible, for example, an optional external optical bypass relay. In these cases, the customization of values is done in the initialization routines so that the implementor can read hardware signals to determine the current configuration. (In the bypass case, insertion of the connector for switching the bypass could be detected through loopback between two pins on the connector.) Where the configuration is stable, the customization can be done in header files or the initialization routine.

It is anticipated that in SMT 6.2 the number of attributes requiring review for a node will be less than 25. In some cases, these attributes occur multiple times (e.g., one per PHY). In a few cases, the values of occurrences of an attribute may differ (e.g., Requested Path for a MAC). These attributes requiring review are well documented with defaults for the most common implementations included and guidelines for selection of different values.

3.0 Integration Procedure (Continued)

Example Section of Initialization Code

```

InitHardware ()
/*****
Function: Initialize interface to Example boards and initialize
         hardware for SMT.
Parameters: None
Input:     None
Output:    Sets file descriptor to access boards.
Return:    0 if successful, otherwise error code is returned
*****/
{
int  pgrp;           /*process group ID*/
uint16 macInts;     /*MAC interrupt mask*/
uint16 beaconPrt;   /*buffer address of beacon prt*/
/*
 * Open driver interface.
 */
if ((Examplefd=open (FDDI_DEVICE,0_RDWR,0777))<0)
{
printf ("Error %d opening device %s\n",FDDI_DEVICE);
return (errno);
}
/*
 * If second board doesn't open, the SMSAS is assumed.
 */
Examplefd2 = open (FDDI_DEVICE2, 0_RDWR,0777);

/*
 * Initialize MAC.
 */
WriteREG (MAC_MODE_REG, 0); /*put MAC offline*/
macInts = MAC_OFF_IMASK; /*set interrupt mask*/
WriteREG (MAC_IMASK_REG, macInts);

/*
 * Initialize ENDEC on primary board.
 */
/*Reset PHY A*/
WriteReg (PHY_CMD_REG, CR2);
WriteReg (PHYA_DATA_REG, PHY_Reset);
/*Send Quiet*/
WriteReg (PHYA_CMD_REG, CRO);
WriteReg (PHYA_DATA_REG, FORCE_QUIET);

/*Select TB-Bus and Repeat mode*/
WriteReg (PHYA_CMD_REG, CR1);
WriteReg (PHYA_DATA_REG, (PHY_TB|PHY_Repeat));

```

3.0 Integration Procedure (Continued)

Write Interrupt Handler Code

The nature of this task is affected by the environment in which the code is running. The functionality of an Interrupt Handler may be included in the interrupt service routine or it may be implemented as a separate function. There are 15 interrupt classes used by SMT 6.2. Most of these classes

include only one interrupt (e.g., TVX expiration), while others include virtually identical code for multiple interrupts (e.g., MAC Counter overflow: Frame_Ct, Error_Ct, etc.). National's Chip Set includes additional interrupts that are not used by SMT. These can be added as events that are passed through the SMT code to the MAP interface.

Example Interrupt Handler Function

```

ProcessLSA ()
/*****
Function:  Process line state change on PHY A or S.
Parameters:  None
Input:  Reads hardware status.
Output:  Hardware register cleared to current line state.
         Posts signal for CMT.
Return:  None.
*****/
{
  uint16 ls;
  static uint16 prevLS = 0;
  /*
   * Read current line states shown in hardware.
   */
  ls=ReadLineState (PHY_A);
  ls &=~ (LSU|OVUF); /*remove unwanted states*/
  /*
   * Post line state change signal.
   */
  /*if (ls !=prevLS)*/
    SendSignal (SIG_Line_State, PHY_A, ls);
  prevLS = ls;
  return;
}

```

The XLNT Manager Integration Guide outlines the options and how to select them.

3.0 Integration Procedure (Continued)

3.5 PROVIDE CONTROL INTERFACE

The SMT software needs stimulus for beginning execution. In initial test, the person testing the interface should have access to basic SMT control. A simple connect and disconnect capability meets the minimum requirements. The trace capabilities of the SMT software and other facilities provided through SMT can provide a more functional control interface.

3.6 COMPILE, LINK AND TEST

After compiling and linking the code for MSP, CSP, or FSP, the testing stage is entered. The number of stages included in integration testing are largely influenced by the degree of testing that the hardware has been subjected to prior to integration, and whether the software is partitioned over multiple processors. The software includes some basic trace diagnostics to aid in integration testing. A detailed description of these functions is available in the XLNT Manager Integration Guide.

3.7 INTEGRATE MANAGEMENT APPLICATION PROCESS

The scope of defining an interface to an application process can vary in degree of complexity. If no management agent like SNMP is required the task is very simple. In some stations, the higher layer protocols used will also include a Management Application Process. TCP/IP networks will likely employ SNMP. The management agent requires access to the management information base within the FDDI attachment, and may require access to optional SMT frames that the SMT software supports. Although not all stations provide a formal management agent, the SMT software requires some external control process to provide the initial request for Connection Services and to handle catastrophic errors or conditions that the SMT components may report. This simplifies implementation in basic stations, as well as those that include an agent. A more detailed description of the procedure interface is provided in the XLNT Manager Integration Guide.

The SMT package includes a sample simulation environment written to operate under UNIX® V.3.

3.8 TIMER REQUIREMENT

The XLNT Manager implementation of SMT requires only a single timer to satisfy all of its timing requirements, and allows a great deal of flexibility in the means with which it can access that timer. The code works with either of two timer access methods:

Method 1— Fixed period clock 'tick' to a single timer handler

Method 2— Direct enqueueing of requested timeouts by the timer handler

SMT contains two classes of functions related to timer usage. The first class, constant processes that run when the station is inserted into a ring with at least one active connection, includes the SMT message timestamp, Link Error Monitor and Frame Based Management. The second class only requires timer support when Ring Management (RMT) processes exception conditions or when Connection Management (CMT) is establishing a connection. Stations should typically run only the first class of functions 99.99% of the time.

The second class of functions has variable period requirements and irregular usage. Method 1 timing needs a tick period ≤ 20 ms. For CMT and timestamp considerations, XDI recommends a tick period ≤ 8 ms. With CMT, longer timer periods retard connection activation, impacting ring availability and potentially affecting neighbor performance. The SMT standard requires timestamp resolution to within 8 ms, which comes naturally with ≤ 8 ms periods, or requires the additional ability to read a timer with that granularity with longer periods.

Method 2 timing provides generally better behavior for SMT timers, though it can produce brief bursts of timer interrupts during connection establishment. Physical Connection Management (PCM) can theoretically produce strings of these interrupts 30 ms apart up to 10 interrupts long. Actual separation depends on the speed of the neighbor station. These PCM functions generally only run when a station powers on (SAS or DAS), or when the neighbor station powers on or off (DAS). Thus, while use of Method 2 more severely impacts your cache hits for brief periods, its overall effect is less.

For proper timestamp operation, Method 2 timing requires either the ability to read a timer with ≤ 8 ms granularity or a separate Method 1 timer with a ≤ 8 ms period.

4.0 Sublicense Agreement and Maintenance

XLNT Manager is a product of XLNT Designs Inc. National Semiconductor sublicenses XLNT Manager source code to its customers. A sublicense agreement is available on request. Maintenance through 1991 is free. Maintenance upgrades are available after 1991 under the terms of the optional maintenance agreement. A maintenance agreement is available on request.

5.0 Technical Support

XLNT Manager software written in the C programming language comes complete with an easy to use integration guide. Most engineers can port XLNT Manager software to their hardware platform in less than six weeks. A detailed description of the integration procedures are outlined in the Integration Guide. Most customers can proceed with little assistance but should the need arise, National's specially trained application staff is ready to answer your technical question. National's FDDI staff has been trained by XLNT Designs as to the structure, function and porting procedures of XLNT Manager software. In addition, staff members have ported XLNT Manager to a variety of platforms including the DP83200EB and DP83200BK kits, and an internal router project. Should the need arise, XLNT Designs will provide additional support.

Product Evaluation

The XLNT Manager object code is included in DP83200 evaluation kits to simplify the evaluation phase. XLNT Manager software running on the National Hardware platform is also available for interoperability testing at the University of New Hampshire.



Section 4
**Application Notes and
System Briefs**



Section 4 Contents

AN-675 Designing FDDI Concentrators	4-3
AN-741 Differentiating FDDI Concentrators	4-11
AN-740 Incorporating FDDI MAC Level Bridging	4-28
AN-726 Station Management Simplified	4-36
AN-727 A Guide to the Implementation of Physical Connection Management	4-46
AN-728 FDDI Station Management with the National Chip Set	4-59
AN-736 Interfacing the HPC46064 to the DP83200 FDDI Chip Set	4-66
AN-678 BMAC Device Software Design Guide	4-72
AN-689 BMAC Device Hardware Design Guide	4-78
AN-730 BSI Device Software Design Guide	4-99
AN-674 Layout Recommendations for a System Using National's FDDI Chip Set	4-109
AN-679 Point-to-Point Fiber Optic Links	4-150
SB-107 High-Speed, Point-to-Point, Fiber, Data Communication	4-158
SB-112 FDDI Concentrator	4-160
SB-115 An FDDI-Ethernet Router	4-163
SB-116 FDDI-Adapter Card	4-165

Designing FDDI Concentrators

National Semiconductor
Application Note 675
Gabriel Li,
Ron Perloff, Robert Grow, XDI



INTRODUCTION

The Concentrator plays an important role in the Fiber Distributed Data Interface (FDDI) architecture. This application note introduces the concentrator, discusses its applications and describes its structure. It also illustrates how to build various kinds of concentrators with the National Semiconductor FDDI chip set. After covering Station Management requirements, it concludes with a thorough discussion of clocking considerations, including a set of rules to guide the designer through this important part of the implementation.

WHAT IS A CONCENTRATOR?

FDDI employs dual counter-rotating rings which, among other benefits, allow a network to continue working in the presence of a single failed link or node. Such a failure causes the network to go into "wrap", wherein the station directly upstream of a failure re-routes its output to the secondary ring while the station immediately downstream of the failure accepts input from the secondary ring. A subsequent failure segments the network, isolating stations on one segment from those on the other.

The standard defines two classes of stations: a dual attachment station (DAS) which connects directly to both rings and a single attachment station (SAS) which connects to just one of the rings. Concentrators support the connection of single attachment stations to the network, allowing communication between SAS and DAS stations.

A concentrator, which itself can be either a single attachment or dual attachment node, provides "drops" to individual nodes, thereby including them on one of the rings. When it senses a failure on one of its drops, it "heals" the ring by electronically bypassing that drop. Properly designed concentrators can bypass any number of drops with no degradation in ring performance.

The standard allows dual attachment nodes to connect into a concentrator in the same way as a SAS, creating redundancy at this level. Further, it allows one concentrator to connect into another, providing additional flexibility.

BENEFITS OF USING CONCENTRATORS

Employing concentrators provides multiple advantages:

Improved network reliability: If a network contains only DAS nodes, connected on the dual ring, single failures will normally cause the ring to wrap, and subsequent failures will segment the ring. Since a concentrator can bypass any number of its drops, it makes the ring tolerant to a larger number of failures. This higher reliability makes large FDDI networks more practical.

Simplicity of wiring: Connectivity between a concentrator and SAS attachments matches normal wiring topologies. Most facilities planners wire buildings in star or star-of-stars fashion rather than in rings. Concentrators allow them to install dual fiber cables between offices and wiring closets, making it easy to add, remove or power down nodes on a network with minimal disturbance to other users.

More compact desktop workstations: Manufacturers constantly sustain pressure to reduce the footprint of their desktop workstations. Concentrators, by providing connectivity to SAS nodes, reduce the logic and power requirements of workstations which, in turn, reduces their footprint.

CONCENTRATOR CONFIGURATIONS

The FDDI Station Management (SMT) standard defines the Concentrator Configuration Element (CCE), the interconnection mechanism between individual PHYs and Paths (Figure 1). Each CCE connects to an individual PHY and one or more paths. The standard requires the Primary Path, allowing the Secondary and Local Paths as optional. The PHYs attached to the CCEs can be:

PHY M: A master port which supports connection and bypass of nodes.

PHY A and PHY B: Ports used for connection to a dual ring.

PHY S: A slave port used for connection to a master port of a concentrator. A slave can be either a single attachment station or a single attachment concentrator.

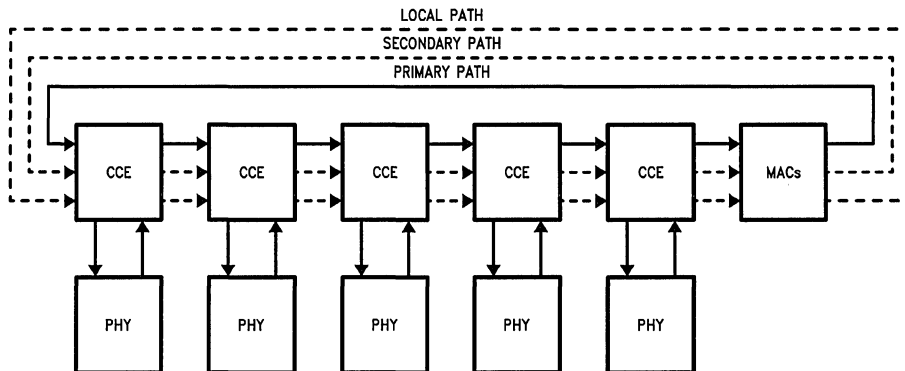


FIGURE 1. The Concentrator Configuration Element (CCE)

TL/F/10792-1

The general architecture supports many configurations, as indicated by the following examples:

1. A null MAC single attachment concentrator with master ports connecting onto the primary ring (Figure 2).
2. Single MAC single attachment concentrator or single MAC dual attachment concentrator with all master ports on the primary ring (Figure 3).
3. Single MAC dual attachment concentrator with all master ports switchable to either the primary ring or the secondary ring (Figure 4). This allows the attached nodes to participate on one of the rings. In the figure the master ports are shown switched to the secondary ring.
4. Dual MAC dual attachment concentrator with master ports individually switchable to either the primary ring or the secondary ring (Figure 5). This example places MACs on the exit ports of the primary and secondary rings.
5. Dual MAC dual attachment concentrator with master ports individually switchable to either ring or to the local path which also has a MAC (Figure 6). The local MAC, sometimes called a roving MAC, allows communication between the concentrator and nodes attached to individual master ports before inserting them into the network.

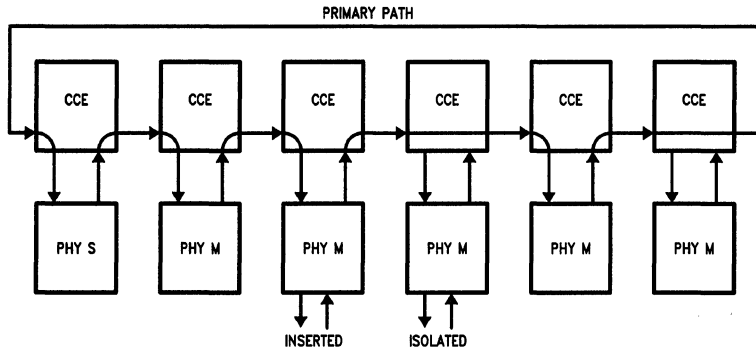


FIGURE 2. A Null MAC Concentrator

TL/F/10792-2

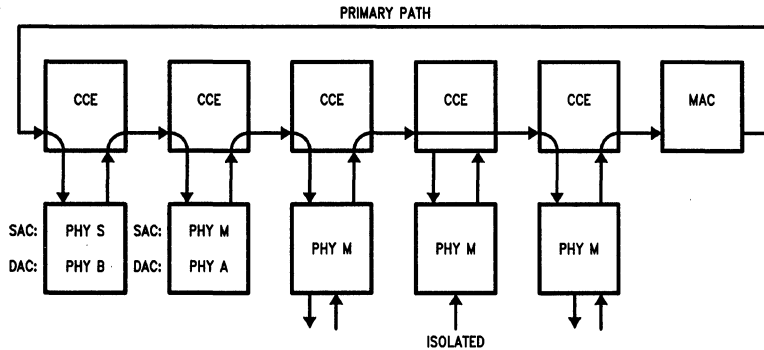


FIGURE 3. Single MAC Single Attachment Concentrator

TL/F/10792-3

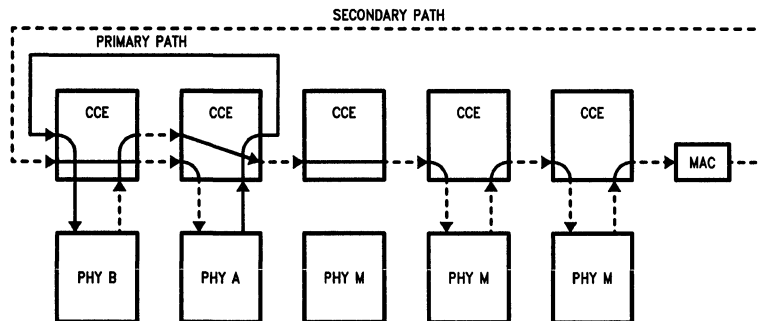


FIGURE 4. Single MAC Dual Attachment Concentrator

TL/F/10792-4

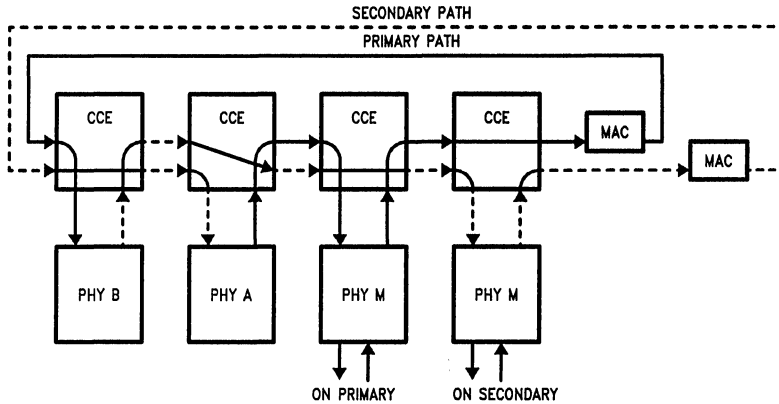


FIGURE 5. Dual MAC Dual Attachment Concentrator

TL/F/10792-5

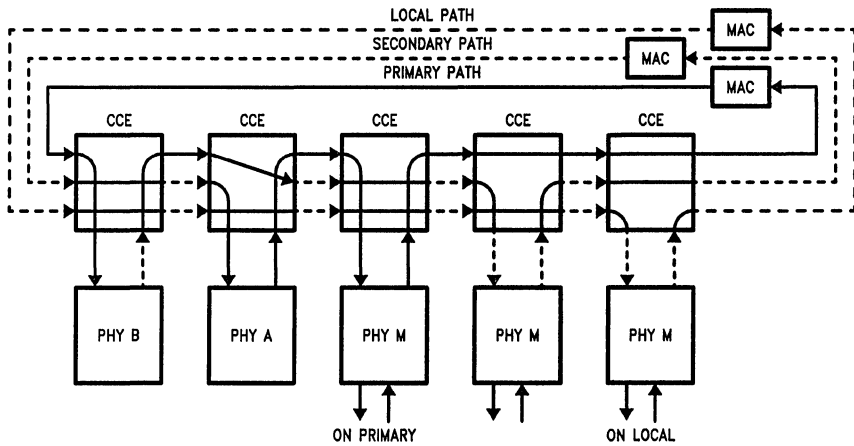


FIGURE 6. Dual Attachment Concentrator with Local MAC

TL/F/10792-6

BUILDING CONCENTRATORS WITH NATIONAL'S CHIPS

The National Semiconductor FDDI chip set supports various types of concentrators. The DP83251/55 Physical Layer Controller (PLAYER™ device) implements the Physical Layer (PHY) of FDDI. The PLAYER device interfaces to the Basic Media Access Controller (BMAC™ device) and other PLAYER devices through two full duplex parallel ports, PORTA and PORTB. The DP83251, with just a single port, provides a more economical solution for single attachment stations and certain concentrator configurations.

The PLAYER device's Configuration Switch provides the flexibility to implement most concentrator configurations without any additional external data path logic. Roving MAC configurations require only minimal external logic, consisting of two-way MUXes and registers. National Semiconductor's BiCMOS technology minimizes the PLAYER device real estate, cost, and power dissipation, making these chips ideal for concentrator applications.

National Semiconductor's FDDI chip set directly supports SMT protocols, eliminating the external "glue" otherwise required for this function.

CONCENTRATOR EXAMPLES

The following examples illustrate the simplicity of building concentrators with the National chip set. The figures show the data paths connecting the PLAYER and BMAC chips. Where a PLAYER device is shown, the implementation will also include a DP83231 Clock Recovery Device (CRD™ device) and the fiber optic transceivers. Figures 7-11 do not show the clocking structure, which depends upon the size of the concentrator. The DP83241 Clock Distribution Device (CCD™ device), discussed later in this application note, simplifies the clocking regardless of functionality or size.

A null MAC concentrator, previously shown in Figure 2, employs DP83251 PLAYER chips (Figure 7). Adding a MAC to this minimal design (Figure 8) creates the hardware necessary for either a single MAC single attachment or single MAC dual attachment concentrator that inserts all master ports in the primary ring as shown in Figure 3. The MAC includes the DP83261 BMAC device, buffer memory and a processor. This configuration requires no glue logic between any of the FDDI chips, nor does it need external hardware for SMT support. The fiber optic connector receptacle

keying and the Connection Management protocols executed while connecting to the network distinguish the two concentrator types which this basic design supports.

Figure 9 illustrates an implementation of the single MAC dual attachment concentrator described in Figure 4. The DP83255 PLAYER chips replace the DP83251 PLAYER chips for the PHYA and PHYB connections.

Figure 10 shows a dual MAC dual attachment concentrator with the master ports individually switchable to either ring.

This requires DP83255 PLAYER chips on all master ports, but again, no additional glue logic is required.

Figure 11 shows the addition of a third "roving" MAC that the local bus can connect to any port without disturbing the network. As illustrated, this configuration requires an external 10-input, 2 x 2 crossbar to augment the DP83255s internal configuration switch. A register on the second output of the crossbar avoids improper stacking of propagation delays in the local path. The implementation of each crossbar requires only a pair of PALs.

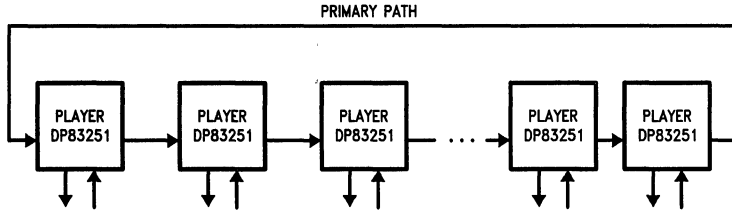


FIGURE 7. A Null MAC Concentrator Using National's FDDI Chips

TL/F/10792-7

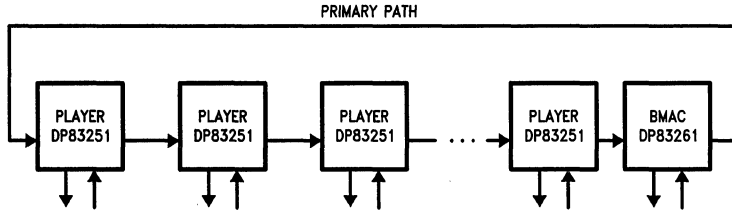


FIGURE 8. Single MAC Single Attachment Concentrator Using National's FDDI Chips

TL/F/10792-8

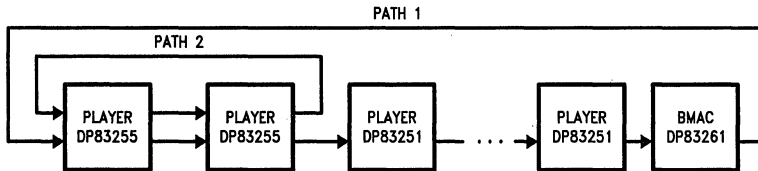


FIGURE 9. Single MAC Dual Attachment Concentrator Using National's FDDI Chips

TL/F/10792-9

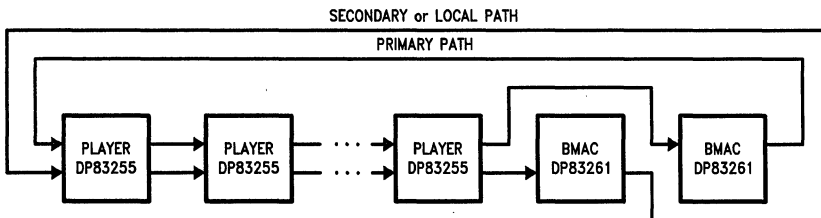
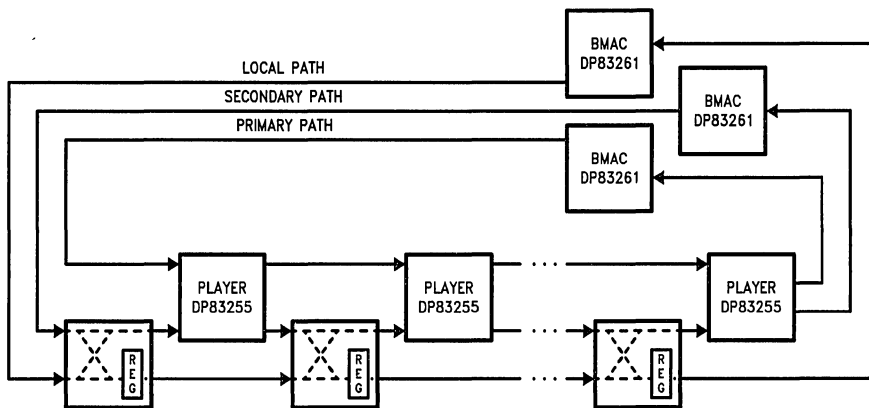


FIGURE 10. Dual MAC Dual Attachment Concentrator Using National's FDDI Chips

TL/F/10792-10



TL/F/10792-11

FIGURE 11. Local MAC Concentrator Using National's FDDI Chips

STATION MANAGEMENT FOR CONCENTRATORS

As indicated above, the National Semiconductor chip set does away with the need for external logic to support SMT. With the following partitioning, a single local processor can execute SMT software for all but the most elaborate configurations:

PHY: Each PLAYER device requires its own separate processes for Physical Connection Management (PCM) and Link Error Monitoring (LEM). The very largest configurations may require one processor for each N PLAYER devices to satisfy PCM's PC_React time. No external hardware is required because the line state history register and the link error detector are contained within the PLAYER chip.

MAC: Each BMAC device requires its own Ring Management (RMT) process.

CONCENTRATOR: The configuration switch built into each PLAYER device fulfills SMT's Configuration Management (CFM) requirement except for roving MAC systems that require the external crossbars shown in *Figure 11*.

CONCENTRATOR CLOCKING STRUCTURE

Poor card layout or clock distribution can forfeit the low error rate benefits of FDDI's fiber optic links. The greater number of PHY elements in concentrators exacerbates any problems in these areas. National Semiconductor designed the DP83241 to simplify the clock distribution.

Concentrators have more stringent packaging demands than most end attachments due to environmental, modularity and reliability considerations. They may range from a single board desktop implementation with a small but fixed number of master ports to large, modular designs that support hot insertion and removal of master port cards. Though systems at these two extremes require different clocking structures, the DP83241 Clock Distribution Device provides reliable solutions.

The DP83261 BMAC and the DP83251/55 PLAYER devices require a 12.5 MHz TTL clock for internal operation and communication between one another. The DP83251/55 PLAYER device also employs a 12.5 MHz ECL clock to drive the parallel to serial conversion in the transmitter and a 125 MHz ECL clock to shift each bit out of the transmitter. The PLAYER device imposes specific timing relationships among these clocks, with relatively tight tolerances between the two ECL clocks and substantially relaxed tolerance between the two 12.5 MHz clocks.

The CDD device guarantees the relationships of the ECL and TTL clocks that it generates to significantly tighter tolerances than the PLAYER chips require. A concentrator clocking structure must distribute and deliver these signals without introducing more skew than the PLAYER chips can accept.

Figure 12 shows the simplest concentrator clocking structure, suitable for small systems, in which the CDD device directly drives all the clocks. The delays for all clocks in this configuration come solely from capacitance and trace lengths. The TTL clock goes to the BMAC device, the buffer and processor interface logic and the PLAYER devices. The ECL clocks run from the CDD device to the PLAYER devices in differential pairs, thus minimizing the introduction of noise and skew. The prudent designer will obey the following check list:

1. Keep all clock lines as short as possible, lowering exposure to problems.
2. Match the trace lengths of all ECL signals to within 0.5", eliminating one source of skew.
3. Keep the two traces of each pair close and parallel to each other, reducing noise susceptibility and skew.
4. Escort the pairs with ground traces along both sides, further reducing noise susceptibility and providing better control of the clock edges.

- Minimize the number of layer changes of these signals to minimize spurious reflections.
- Avoid stubs that cause reflections by running the traces through the destination pins to proper terminations in the characteristic impedance of the board. *Figures 13a and 13b* illustrate the right and wrong way to avoid stubs at

- termination. When board space limitations do not permit elimination of stubs, limit them to a maximum of 0.5".
- Run the ECL signals above a virtual ground plane that does not carry any TTL current, eliminating a source of noise injection.

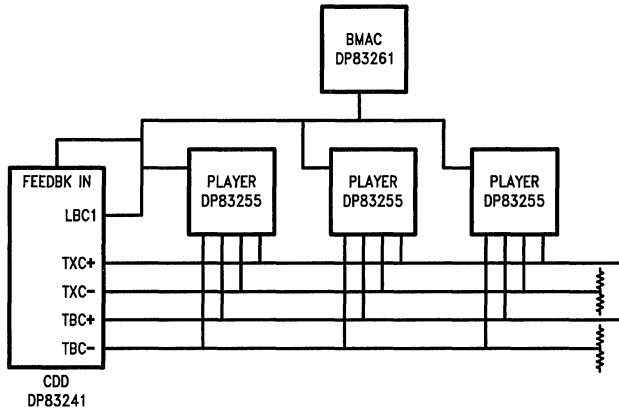


FIGURE 12. Small Concentrator Clocking Structure

TL/F/10792-12

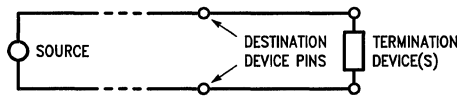


FIGURE 13a. Proper Termination—No Stub

TL/F/10792-13

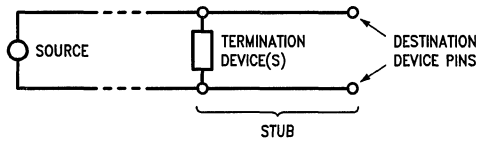


FIGURE 13b. Improper Termination

TL/F/10792-14

Figure 14 shows a slightly larger concentrator clocking structure in which the fanout on the clocks exceeds what the CDD device can directly drive. A non-inverting buffer, such as a 74F08, drives the clocks. Feedback from the output of the driver into the CDD device allows it to compensate for the driver's delay. Skew calculations for this case must include the differential delay through the TTL clock buffers. The delay through the F100115 ECL clock driver chip will also contribute to the skew. Here, the wise designer will add another rule.

8. When distributing the clock through multiple buffers, keep the capacitive and resistive load on each buffer the same. Differing loads can cause differing delays and contribute to skew.

Figure 15 shows the clocking structure on a typical board of a multi-board system which supports field increases in the number of installed boards. This modularity requires a ro-

bust clocking structure to accommodate the greater distances between boards, the potential reflections introduced by the interconnection system, and the difference in delays found in components on different boards that come from different manufacturers or different lots, have a different aging history and are operating at different temperatures. The CDD device feedback mechanism allows it to accommodate these vagaries. A single clock source drives the feedback input of a CDD located on each board. Each CDD device drives the PLAYER and BMAC devices on that board directly as shown in Figure 12. Larger board form factors that house more PLAYER or BMAC devices than the single CDD device can handle require additional DP83241s.

CONCLUSION

The Concentrator creates additional degrees of configuration freedom and greater reliability for FDDI networks. The National Semiconductor FDDI chip set provides the basis for robust concentrator designs that require virtually no glue.

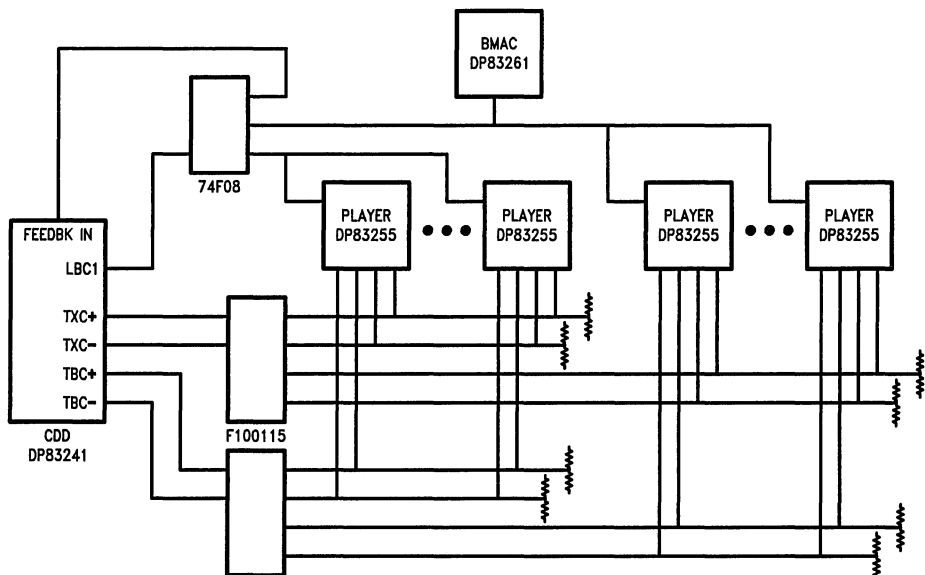


FIGURE 14. Medium Size Concentrator Clocking Structure

TL/F/10792-15

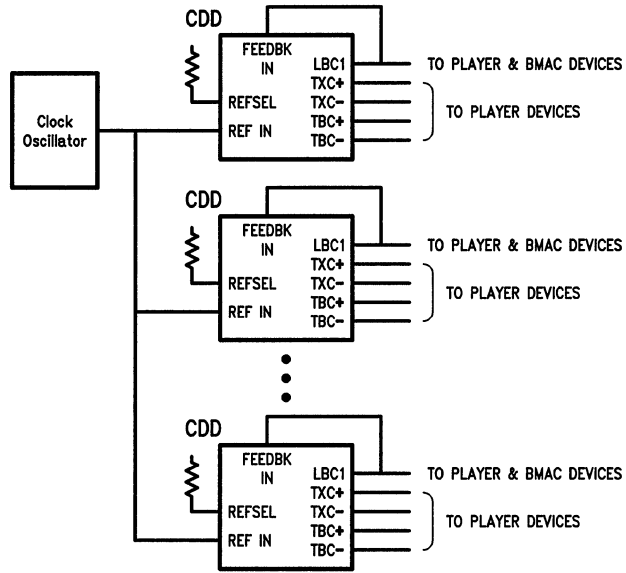


FIGURE 15. Multi-Board Concentrator Clocking Structure

TL/F/10792-16

Differentiating FDDI Concentrators

National Semiconductor
Application Note 741
David Brief, Bob Hanrahan



AN-741

INTRODUCTION

The FDDI Standard offers a broad based set of capabilities that will allow it to become the standard high performance network of choice for the '90s. The FDDI Concentrator plays a central role in providing the necessary flexibility, reliability and manageability in order for FDDI to enjoy widespread deployment. A single Concentrator design is unlikely to meet the wide range of cost and performance objectives for all installations and environments. As the FDDI marketplace develops, many different types of concentrators will be required that are differentiated on the basis of their total cost combined with their expandability, functionality, reliability and the management features they provide. This market differentiation of concentrators is just beginning.

After a short overview of the place of concentrators within FDDI, issues relating to requirements of concentrators and how to differentiate them in the marketplace are discussed. Finally, two example designs that make full use of the National Semiconductor DP83200 chipset are presented. The example designs include a single board single processor design and a multiboard multi-processor design. These examples can be used as a basis and starting point for a whole spectrum of solutions.

1.0 CONCENTRATOR OVERVIEW

1.1 FDDI Topology Considerations

The FDDI Standard defines two classes of stations: a dual attach station (DAS) which connects to both rings either directly or through a concentrator and a single attach station (SAS) which connects to one of the rings through a concentrator. A concentrator, which can be either a single attach or dual attach node, provides ports to individual nodes, thereby including them in one of the rings. Further flexibility is provided by allowing a concentrator to connect into another concentrator and create a tree structure.

The FDDI topology is defined as a dual ring of trees where any subset is a legal topology. This permits many possible physical topologies:

- 1) single concentrator with a limited number of SASs
- 2) a "tree" of concentrators with several levels of concentrators with SASs at the lowest level
- 3) a dual counter rotating ring with multiple dual attach stations
- 4) a dual ring of concentrator trees

The concentrator plays a key role in many of the permitted and preferred physical topologies, providing the required flexibility to accommodate the common building cabling practices.

The FDDI physical topology is composed of point to point full duplex links. These links are arranged in a combination of tree (star) and ring topologies. These links are then formed into a logical ring topology using a signaling technique on each duplex link. When fully functional, the logical topology consists of dual counter-rotating rings, independent of the physical topology.

The counter-rotating rings allow a network to continue working in the presence of a single failed link or node. Such a failure causes the network to go into "wrap", wherein the station directly upstream of a failure re-routes its output to the secondary ring while the station immediately downstream of the failure accepts input from the secondary ring. This then bypasses the fault and creates a single logical ring.

A concentrator has the added benefit of isolating a faulty link or node in one of the ports that it is controlling. It does not need to wrap the ring and instead it can electronically bypass whatever is connected to that port to effectively heal the ring with minimal disruption.

1.2 Benefits of Using Concentrators

Employing concentrators as part of a physical topology plant provides many advantages as detailed below.

1.2.1 Simplicity of Wiring

Connectivity between a concentrator and SAS attachments matches normal wiring topologies. Most facility planners wire buildings in star or star-of-star fashion rather than in rings. Concentrators require installation of one fiber pair between offices and wiring closets instead of 2 fiber pairs required by a DAS attachment.

1.2.2 Reduce Overall Cost

Manufacturers constantly sustain pressure to reduce the footprint of their desktop workstations. Concentrators, by providing connectivity to SAS nodes, reduce the logic and power requirements of these nodes, which in turn reduces the footprint. This is then reflected in the cost of the node (workstation). In addition, there is reduced fiber and connector cost compared with not using a concentrator.

The additional cost of the concentrator may initially offset the reduced price of the workstation. This does not account for the long term network facility costs in which ease of maintenance, manageability and reliability are crucial.

1.2.3 Improved Network Reliability

If a network contains only DAS nodes connected on the dual ring, single failures will normally cause the ring to wrap, and subsequent failures will segment the ring into one or more disjoint rings. Since a concentrator can bypass any number of the nodes connected to its ports, it makes the ring tolerant to a larger number of failures. This higher reliability makes very large FDDI networks more practical.

For additional reliability, it is also possible to connect the ports of Dual Attach Stations to different concentrators. This increases the network reliability in terms of survivability in the presence of faults (at the cost of increasing the number of physical connections (links) in the network).

1.3 What Is in a Concentrator

The FDDI concentrator provides ports for connecting the concentrator into one of the rings and for connecting nodes into one of the rings. FDDI defines 4 types of ports, the A Port, the B Port, the S Port and the M Port.

The A and B ports are typically found in the dual ring of trees (sometimes called the trunk ring). An A-B connection is called a Peer Connection. The M (Master) and S (Slave) ports are typically found in the concentrator trees. The M-S connection is a Tree connection. The M port is provided by the concentrator to connect Slaves into the ring via their S ports. As defined currently by SMT, a concentrator may have from 1 to 255 M ports. A concentrator also provides a connection to the rest of the network. To connect a dual attach ring, an A port and B port are provided. To connect to another concentrator in a tree, an S port is provided. In a complete tree physical topology, the concentrator at the top of the tree would not have any A, B or S ports.

For each port in a concentrator there is one PHY (physical) layer entity. The PHYs perform the PHY layer services including re-timing the input data stream, performing the 4B/5B encode/decode function, monitoring the link error rate on the incoming link and providing the facilities to perform the link and connection establishment. Second generation FDDI chipsets such as National Semiconductor's DP83200 chipset, provide a large level of support for these functions, but so far none alleviate the need for some external processor support for performing CMT and LEM.

Aside from the number of M Ports a concentrator provides and the way it attaches to the rest of the network, concentrators are also differentiated on the basis of the number of internal paths they provide. The paths within a station are used to connect the stations attached to the concentrator M ports to the primary or secondary ring. Some concentrators may only be able to connect M ports onto the primary ring while others may be able to connect these ports onto the secondary ring as well. A configuration switch is required with each PHY entity in order to attach the associated M Port into either the primary or secondary ring.

To take full advantage of the management capabilities afforded by a concentrator, a MAC level entity is required in a concentrator. A concentrator may include up to one MAC per path. At the simplest level, the MAC provides a frame level interface. This makes possible communication with all of the other stations in the ring. This allows the concentrator to participate in all of the management protocols such as the parameter management and status reporting protocols.

The concentrator also provides an obvious place to perform network management functions such as gathering relevant statistics for all stations, detecting degrading links, isolating faulty stations, managing legal configurations, and detecting illegal topologies. The concentrator may also serve as an interface between higher level management services such as the Simple Network Management Protocol (SNMP) and the local FDDI station management.

1.4 Station Management in Concentrators

The FDDI concentrator is an active station in an FDDI LAN unlike hubs in IEEE 802.3 and passive wiring centers in IEEE 802.5. As an active station it is responsible for monitoring and maintaining all of its connections to other stations in the network and for performing and providing the services required by the FDDI Station Management (SMT). The SMT for Concentrators is a superset of the SMT for Stations. For example the path management requirements for a station are less complicated than the path management issues for a concentrator.

SMT can be viewed as an entity that manages several objects and their interactions within a Station or Concentrator. Information related to each manageable object is maintained as described in the FDDI SMT Management Informa-

tion Base (MIB). Currently, the MIB defines four object types: MAC, PATH, PORT and ATTACHMENT. Each object may have multiple instances within a node. For example, a typical concentrator might have two MAC objects, two PATH objects and ten PORT and ATTACHMENT objects.

It is convenient to view SMT as a distributed database where each station keeps track of its own information. SMT defines the information in the MIB, how to access it, and how it changes. All SMT processes are involved with manipulating the MIB.

The SMT frame based processes typically convert information from the MIB into frames. SMT frames are defined to help build ring maps and detect duplicates (Neighbor Notification frames), provide station information, provide access to parameters (Parameter Management frames), and announce interesting conditions (Status Reporting frames). These processes are typically implemented for each MAC object.

An instance of the SMT Ring Management (RMT) process is present for each MAC object. RMT is responsible for providing information to keep the ring operational. This includes detecting Stuck Beacon conditions, detection of duplicate addresses that keep the ring from becoming operational, and notification of MAC Availability. RMT also provides recovery mechanisms for the stuck beacon condition and duplicate address conditions that prevent the ring from becoming operational. The current state of RMT is kept in the MIB. The DP83261 BMACTM device supports efficient implementations of RMT by reporting all of the necessary events and conditions.

In addition, for each MAC frame level statistics are maintained. These include the number of frames transmitted, received, copied and not copied in addition to the frame error and lost frame counts. The measured Ring Latency and Token Count are also maintained. The DP83261 BMACTM device provides extensive support for all of these statistics.

An instance of the Physical Connection Management (PCM) process is present for each PORT object. This process is responsible for the initialization of physical connections. This includes support for connection policies such as the withholding of undesirable and illegal connections, signaling of physical topology information and notification of availability for a PORT connection. PCM also includes support for the fault tracing function and testing of link confidence. The DP83251/55 PLAYERTM device supports efficient implementations of PCM with no external logic, and minimal constraints on the processor.

In addition, for each PORT continuous monitoring of the estimated Link Error Rate is also accomplished. Should this link error estimate fall below a MIB defined alarm threshold, the event is announced through Status Reporting Frames. Similarly if the link error estimate falls below a MIB defined cutoff threshold, the link is broken and an attempt is made to reconfigure the network around the physical layer fault. The DP83251/55 PLAYER device supports efficient software implementations of the Link Error Monitor with no additional components.

One instance of the Entity Coordination Management (ECM) process is present in a Node and controls all of the Attachment objects. This process is responsible for controlling the optional bypass switches and signaling PCM when the media is available. Actions that must be coordinated across several entities such as Path Testing and the Trace Function are handled in ECM.

The PATH Objects are handled slightly differently within SMT. Instances of Configuration Element Management (CEM) for each PORT are collectively responsible for managing all of the PATH objects. The DP83200 chipset and in particular the DP83255 PLAYER device supports CEM through its built in Configuration Switch.

In concentrators there are several special considerations for SMT. Handling of Dual Homing must be accounted for, as must the unique cases of propagating the Trace function. In addition, more complex configurations such as Trees must be described and managed. This includes detection of illegal topologies such as Master Slave Loops, and the notion of Root Concentrator MACs.

There is still a fair amount of confusion regarding Path Management issues in SMT, especially as they relate to a concentrator. As more experience is gained in building stations and concentrators, this process should converge to a ratified SMT Standard.

2.0 DIFFERENTIATING CONCENTRATORS

FDDI concentrators can be differentiated in many ways just as in any diversified product such as Adapter Cards, Bridges, Routers, etc.

The most obvious way in which concentrators are differentiated is in the number of each of the resources that are managed. Other ways concentrators are differentiated are in terms of the options they provide for expansion, the management features they offer, and the diagnostic capabilities they provide. The concentrator also is a logical place to implement Bridging and Routing Functions as well as Management Agent processes. Each of these ways for differentiating concentrators is discussed below.

2.1 Types of Concentrators

There are several concentrator configurations as shown in the table below:

TABLE I. Possible FDDI Concentrator Configurations

Name	Attach Count	Paths	MACs	M Ports
NAC	0	1-3	>1	>0
SAC	1	1-3	>1	>0
DAC	2	2-3	>1	>0

Naming Conventions

The concentrators are named according to the greatest number of attachments they can handle (Null, Single or Dual Attachment Concentrator NAC, SAC, DAC respectively). It is debatable whether this is the best way to name concentrators since the other parameters are more likely to vary.

Attach Count

This refers to the number of attachments that are available for use to either the trunk ring or to the concentrator tree. A concentrator with an attach count of 2 could default to a concentrator with an attach count of 0 or 1 if those attachments are not connected to anything else.

Paths

This refers to the number of token paths possible in a concentrator. These paths may then be used as part of the

Primary, Secondary or Local Ring. There must be at least one path in single and null attach concentrators and at least two paths in a dual attach concentrator. These paths are used to connect ports into the appropriate path.

With only one path in a concentrator, attached stations can only be connected into the ring that the Attachment is connected into. In a tree, if the S Port of a SAC is connected into the Primary Ring then all of the stations attached to the M Ports of the concentrator can only be connected into the Primary Ring. Similarly, if the S port is connected into the Secondary Ring the stations attached to the M Ports can only be connected into the Secondary Ring.

With two paths in a Dual Attach Concentrator, Ports can be placed in both the primary and secondary rings. Currently it is difficult to reliably put a Port on the secondary ring. PCM does not offer the necessary set of hooks to request placement on the secondary ring. The default is that the station is placed in the primary path and can only be switched to the secondary using optional services (by using Parameter Management Frames to set the Requested Paths MIB parameter).

With two paths in a Single Attach Concentrator, Ports can be placed in either the Primary or Secondary Ring depending on the ring in which the Attachment is currently connected. The second path could then be used as a Local Path.

With three paths in a Single Attach Concentrator, Ports can be placed in both the Primary and Secondary Rings. The third path could then be used as a Local Path.

Use of a Local path within a concentrator is not well defined and there is not a clear consensus on the use of and the requirements for a Local Path in a concentrator. Functions that have been suggested for a Local Path usually assume that a MAC can be placed, at least temporarily onto the Local Path. This MAC is sometimes called a roving MAC or local MAC. Suggested uses of the Local Path include:

- setting of slave station's parameters
- testing of slave before insertion into ring
- determining the topology of a concentrator tree before insertion into the Primary or Secondary ring
- graceful insertion of slave stations or concentrator trees.

There are still several problems associated with each of these uses. There is no interoperable method for setting parameters in a slave station and no protocol defined for testing a slave before inserting it into a ring. Similarly it is not clear that a roving MAC could find out more about a slave station than a station self test could.

These same problems could be solved in the context of SMT without resort to a non-standard local path. A slave could do a self test before requesting insertion into the ring by causing remote loopback at the concentrator. The station could come into the ring with default parameters that could also be changed during normal operation. And to solve the ring mapping problems, an efficient protocol using well known SMT Group Addresses could be used. There is still a fair amount of work to be done in the area of inserting slave stations into a ring. Use of the local path does not solve these problems and the extra cost and complexity of adding an additional path is questionable in terms of the minimal payoff.

MACs

In order for a Concentrator to provide management services and act as a manageable entity in a network, it should contain at least one MAC. This gives the concentrator the ability to participate in the SMT frame based protocols and help isolate, announce and recover from all types of problems. By having MACs in concentrators, better logical and physical ring maps can be built.

To simplify the relationship between MACs and Paths, it is easiest to have one MAC per Path. Otherwise it is necessary to multiplex a MAC between the paths. It is difficult to manage inserted stations on a path that has no MAC.

The data throughput requirements for a MAC implementation are rather minimal, thus low cost management MACs can easily be added. The cost associated with MACs has been one of the main points of resistance to standardizing the presence of at least one MAC in a concentrator. The incremental cost of adding a MAC compared with the benefits of the added manageability makes it worthwhile, especially in configurations where it amounts to only two extra chips. (In these configurations all of the memory and processor support is already present.)

Master Ports

A concentrator will have several Master Ports, each of which can connect to a Slave or Peer Port. The number of ports is very implementation specific, and ranges from small departmental concentrators with 4 M Ports to large multi-board concentrators with up to over 250 ports.

2.2 Expandability/Configurability

Another way in which concentrators can be differentiated is by their packaging. In many instances a low cost fixed configuration makes sense. In other cases an expandable and configurable concentrator might be attractive.

An attractive fixed configuration might consist of 8 ports that could be used as an 8 port NAC, a 7 port SAC or a 6 port DAC.

An attractive flexible configuration might consist of Slave or Peer Attach boards and Master Port boards. On each Master Port Board there might be 4 or 8 Master Ports.

An example of both a fixed and flexible concentrator are outlined as examples in Section 4.

Another dimension in which concentrators can be configurable is in the PMD that is supported. FDDI already supports both a multi-mode and a single mode fiber optic PMD and it is very likely that additional lower cost PMDs such as Shielded Twisted Pair and shorter distance fiber optic connections will be supported. By isolating the PMD to a separate board the PMD could be a configurable option.

The Concentrator could also serve as a repeater between different media domains by using different PMDs in different ports.

2.3 Management Capabilities

Concentrators can also be differentiated by the management capabilities that they provide. The SMT Standard provides many implementation options does not specify how the services that it provides will be used. The Concentrator is a convenient location to run Network Management Applications such as a ring monitor, ring mapper, traffic analyzer, traffic generator, etc. It also is a logical location to run higher level management agents such as SNMP and/or CMIP. Management agents serve as proxy agents for manage-

ment services between stations that do not implement the particular management protocol but do implement SMT and the actual management application.

2.4 Diagnostic Capabilities

Since a concentrator is one of the most complex nodes in the network, it should contain extensive diagnostic capabilities. Essential to this is the ability to perform sophisticated Path Testing and Resource testing of all resources within the concentrator. The National Semiconductor DP83200 FDDI chipset has numerous diagnostic capabilities built in to aid in isolating and resolving problems.

2.5 Bridging/Routing Capabilities

The concentrator is also a natural location to add in a MAC level bridging or routing function between similar or different media, networks, speeds, and protocols. Because of the concentrator position within an installation it may offer a natural point to segment the local traffic contained exclusively in the tree versus traffic that must go outside of the tree.

Such a concentrator would become a multi purpose network attachment that could be a repeater/bridge/router and manager where each function is added in as needed on a common backplane. You could think of it as a ring in a box.

3.0 USING THE NATIONAL DP83200 FDDI CHIPSET

The National Semiconductor DP83200 FDDI chip set was partitioned for use in all of the Standard FDDI configurations including Single and Dual Attach Stations with one or two MACs and in numerous Concentrator configurations.

The DP83251/55 Physical Layer Controller (PLAYER device) implements the Physical Layer (PHY) of FDDI and provides all of the support required by a processor running SMT. In addition to providing all of the functions required by an FDDI PHY, the PLAYER device includes support for the Link Error Monitor Function, the Noise Counter (TNE), Line State Detection and Generation, and Configuration Management. Two full duplex parallel ports are supported along with a built in Configuration Switch in order to provide the flexibility to support all of the station types (SAS, SM-DAS, DM-DAS) and configurations (THRU, WRAP, ISOLATE). Concentrator configurations with numerous M Ports and 2 internal paths can be created without any external logic. The low power dissipation and minimal real estate required by the PLAYER device makes these chips ideal for concentrator applications.

The DP83231 Clock Recovery Device (CRD™ device) provides a very high performance analog Phased Locked Loop with very high noise immunity. It can lock to the worst legal patterns in under 85 μ s and provides a very high dynamic lock range.

The DP83261 Basic MAC Layer Controller (BMAC device) implements the Media Access Control Layer (MAC) of FDDI and provides the support required by a processor running SMT. In addition to providing all of the functions required by an FDDI MAC, the BMAC device includes support for RMT, all of the required and optional frame counters, measurements of the ring/path latency and many tunable parameters.

The DP83265 BMAC System Interface (BSI™ device) is also available for providing a straightforward system memory interface. The BSI device also provides several features that are useful for SMT such as independent channels for Management Data (SMT/MAC).

The DP83241 Clock Distribution Device (CDD™ device) generates all of the clocks required by the PLAYER, BMAC and BSI devices. The CDD device accepts either an external reference or a local crystal. In a concentrator the CDD device is particularly useful in that all of the clocks required by the PLAYER device can be generated locally and only slower speed 12.5 MHz clocks need be distributed. The CDD device also provides 10 different phased clock outputs.

The chipset also provides numerous diagnostic features. These include full duplex data paths to allow diagnostic transmission to self, multiple levels of loopback to isolate individual chip and interconnect errors and the ability to inject errors and make sure that proper recovery takes place. In addition, while operational, several levels of self checking are also employed. These include through parity support, full duplex data paths so that every frame transmitted can also be received and the FCS can be checked (the frames could also be copied), and a full implementation of the FDDI protocol which is self checking in itself.

The chipset also includes several programmable features that allow compatibility with the ongoing enhancements to the ANSI specification.

For more details see the appropriate device datasheets.

4.0 EXAMPLE DESIGNS

The following examples illustrate the simplicity of building concentrators with the National Semiconductor DP83200 FDDI chipset. Two example designs are shown representing two ends of the spectrum. First a Single Board, Single Processor Concentrator is shown, followed by a Multi-Board, Multi-Processor design.

4.1 A Single Board Single Processor Concentrator

A simple FDDI concentrator contains a single processor and is designed on a single card. The clear advantage of this type is reduced cost, though forfeiting flexibility and expandability.

Figure 4-1 shows a simple single processor dual attach dual path design. The diagram also suggests a physical layout of the devices. Even the pinout of the devices in the chipset were optimized for these applications.

The concentrator shown has an A Port, B Port, 4 or more M Ports and 2 MACs. It also has two complete physical paths, one of which can be used as the Primary Path, and the other as a Secondary Path or Local Path. The M Ports connect into the Primary or Secondary Path through the PLAYER device's configuration switch as shown in *Figure 4-2*.

Extensive Path testing can be accomplished as two complete rings can be made operational as shown in *Figure 4-3*. For the ring to be operational, all data paths have to be working properly. The MAC on each ring will go through the Claim Process and if it receives its own Claim frame successfully, a token will be issued which will continue to circulate. That MAC could then send frames to itself for even more robust path testing.

The port types are distinguishable by the fiber optic connector receptacle keying. During the connection establishment procedures of SMT (Physical Connection Management—PCM), the connected port types are checked for compatibility. Typically SMT will only allow A to B, B to A and M to S connections, but SMT can be configured to override this protection mechanism.

When there is no trunk ring or concentrator tree to connect into, SMT can be configured to allow slaves to attach to the

A and B Ports (A to S and B to S connections). When there is a tree to connect into, the concentrator can be configured to allow M to A and/or M to B connections. In this way, this concentrator could be used as a Null Attach Concentrator with 10 leaf connections, a Single Attach Concentrator with 9 leaf connections or a Dual Attach Concentrator with 8 leaf connections.

With the two path design shown and correct programming of the configuration switches, the second path could be used as a local path while bypassing the ring's secondary path within the concentrator (see *Figure 4-4*). Note the subtle differences between *Figure 4-2* and *Figure 4-4*. Only the connection surrounding the Peer Ports is changed. Thus effectively, three paths are present within the concentrator, although not all of them can be active simultaneously. With additional external multiplexing, a dedicated third path could be added to the design. The third local path would then always be available for internal testing for use with a roving MAC. The benefits of a dedicated third path versus the additional cost are questionable.

The processor controls the BSI, BMAC and PLAYER devices through the common control interface. The control bus uses a simple asynchronous handshake. The processor can accomplish all of software oriented portions of SMT (CMT, RMT, monitoring functions) using this common interface. The SMT frames are typically generated and processed in a shared memory.

Moderate interrupt response time is required to assure compliance with the default requirements (3 ms) of the Physical Connection Management (T_REACT, PC_REACT, etc.). Depending on the interrupt latency of the processor, as additional M Ports are added, the processor will eventually reach its limit in terms of the number of ports that can be handled simultaneously. In concentrators with many M Ports, in order to guarantee the required response times, it may become necessary to partition tasks among multiple processors or switch to a processor with a better interrupt latency (provided that it is not saturated).

A concentrator requires a moderately high performance processor with support peripherals such as timers and interrupt control. In many designs it is preferable to use a processor that includes these support peripherals. National Semiconductor offers a full range of processors that are appropriate for this application from the HPC family of High Performance Controllers to the 32000 family of processors.

4.2 A Multi-Board Multi-Processor Concentrator

A multi-board concentrator will be used where flexibility and expandability is desired over lower cost. For an organization where adding functionality with time is attractive, the multi-board design is appropriate.

The design is partitioned into three basic cards and a backplane is defined (see *Figure 4-5*). The three cards include: a Master Port Card, a Slave Port Card, and a Peer Port Card.

A controller is present on each card to allow very large configurations beyond the constraints of a single processor. This also decreases the performance requirements of the main processor by reducing the response time requirements. Having a processor on each card coupled with a high enough performance serial bus between all cards also reduces the size and complexity of the backplane to less than 50 signal pins all at or below 12.5 MHz.

4-16

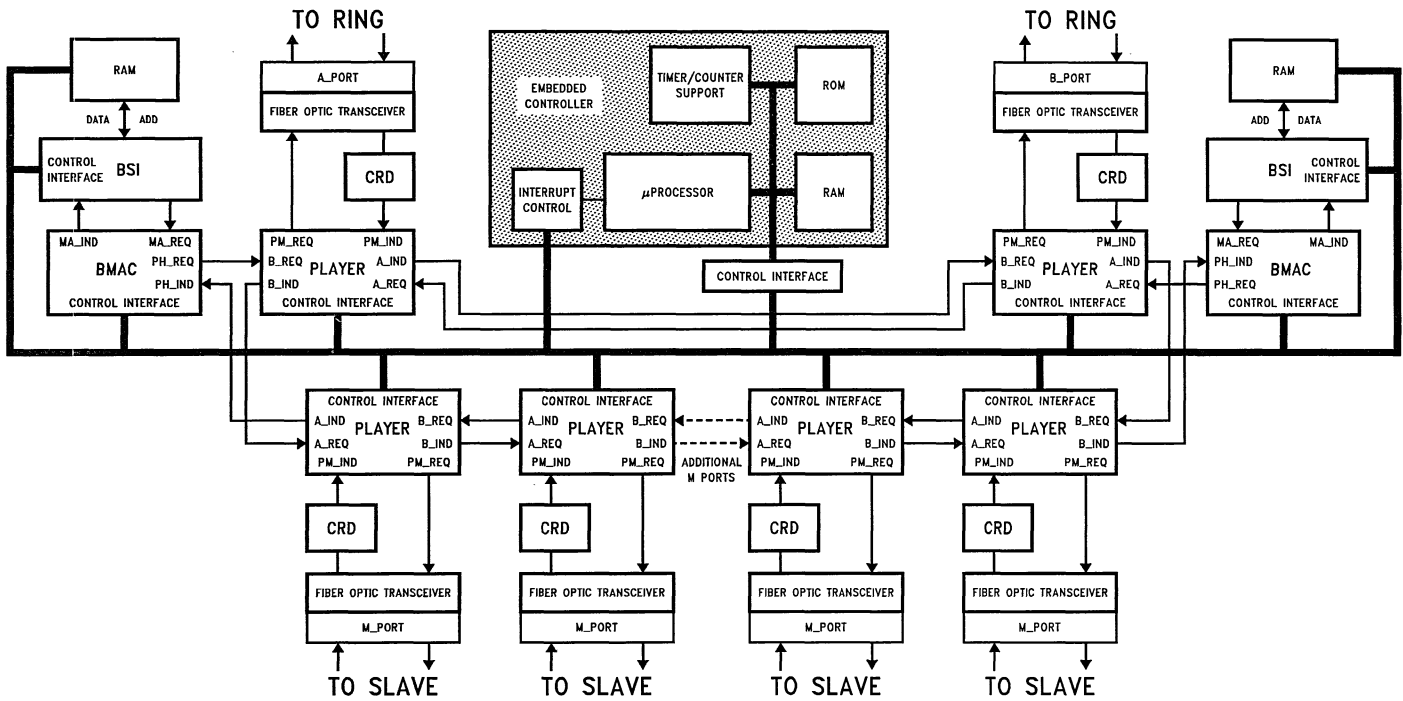


FIGURE 4-1. Single Board Concentrator

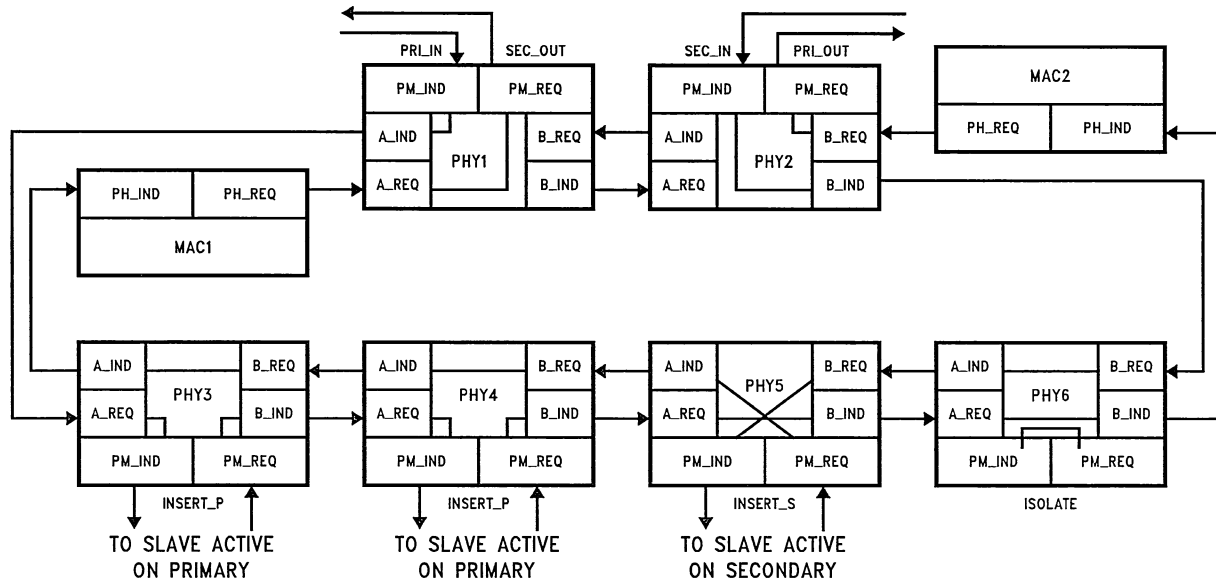


FIGURE 4-2. Configuration Switch Insertion

TL/F/11120-2

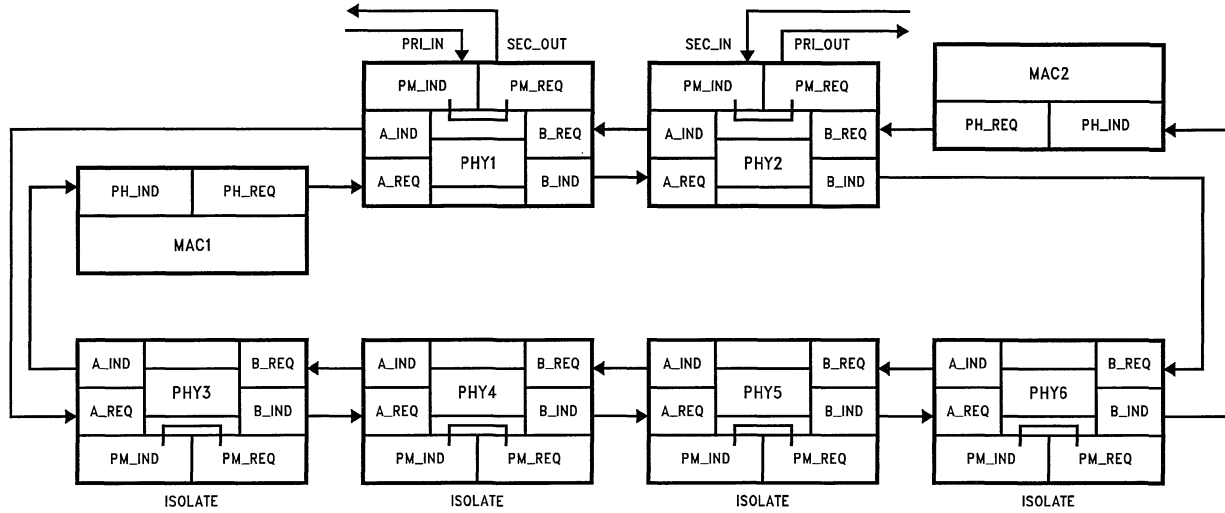


FIGURE 4-3. Internal Path Testing

TL/F/11120-3

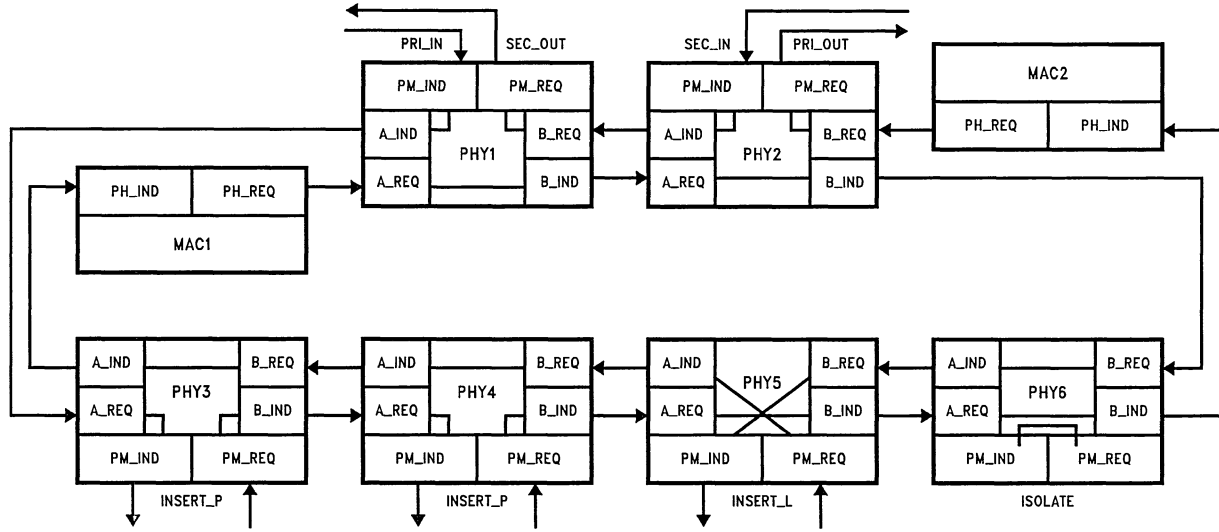


FIGURE 4-4. Use of Local Path

TL/F/11120-4

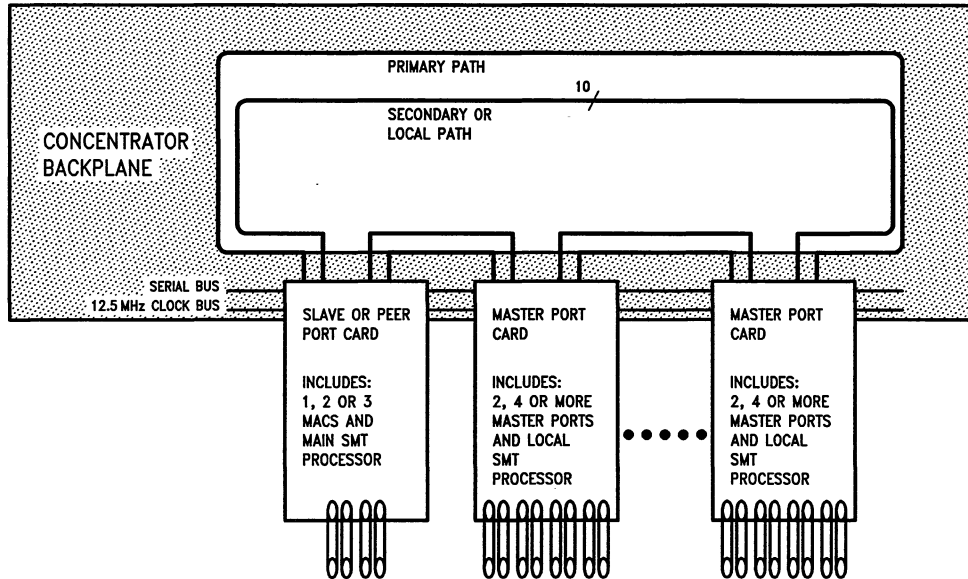


FIGURE 4-5. Concentrator Backplane

TL/F/11120-5

Master Port Card

The Master Port card provides several (2, 4, 8+) M Ports. The Card shown in *Figure 4-6* shows 4 M Ports. These M Ports are typically connected to the S Port of slave stations. The slave station could be a single attach workstation or single attach concentrator at the next lower (or same) level of the concentrator tree. The M Ports could also be used to connect Dual Attach Stations and Dual Attach Concentrators where Dual Homing is permitted. In these configurations, to maximize redundancy, the two attachments would be connected onto two different concentrators.

The M Port card can be designed with multiple PLAYER devices (PHY layers). The number of ports per card is restricted by the physical limitations of the card such as the space required by the connectors, optical components and PLAYER and CRD devices. The processor or microcontroller must service physical SMT tasks within the maximum times specified in the Standard (T_{React}, PC react, etc.). The increase in service latency that accumulates as additional PLAYER devices are connected to the control bus might also limit the number of Ports that can be supported on a card for a given processor configuration. SMT tasks are serviced locally by the processor and reported back to the main SMT processor located on the slave (or peer) card.

The National Semiconductor High Performance Microcontroller (HPC) is well suited for this local task. It provides low interrupt response time, fast 25 MHz operation, embedded timers for CMT and serial communications capabilities for implementing a serial bus.

Peer and Slave Port Cards

The Peer and Slave Port Cards provides ports for connection into the larger Ring. With the Peer Port Card, A and B Ports are provided for connection into either the trunk ring or a concentrator tree. With the Slave Port Card, an S Port is provided for connection into a concentrator tree.

Figure 4-7 shows a Peer Port Card design which includes the A and B Ports, two MACs and the main SMT processor. The Slave Port Card shown in *Figure 4-8* includes one S Port, one MAC and the main SMT processor. An advantage of placing the main SMT processor on the same card as the MAC implementations is to avoid providing support for a full databus on the backplane or on another potentially expensive special connector. With the DP83200 chipset including the DP83265 BSI, System Interface device, space efficient and cost-effective memory interfaces are possible.

The Clock Distribution Device provides all of the clocks required for the DP83200 chipset. Since there is typically only one Slave or Peer Port in a system, this card drives the 12.5 MHz reference clock onto the backplane. Each card has a local CDD device which uses the reference to generate the appropriate clocks for the card. Clock distribution in very large concentrators is discussed separately.

The SMT processor interfaces directly to the local PLAYER, BMAC and BSI devices across the 8-bit Control Bus. This bus is used for access to registers, parameters and counters (configuration switch, line state detector, statistic counters, etc.). The SMT processor also interfaces to the local

memory used for frame transactions (PDUs). The SMT processor handles all lower level SMT tasks (PCM, CFM, etc.). This main SMT processor also uses a serial controller, to interface to the other processors in the concentrator across the serial bus.

National Semiconductor offers a variety of processors and controllers for this task. The NS32GX320 is a 15 MIP CISC machine with on-chip cache, timers, interrupt control and DMA support. The NS32CG160 is a lower performance and lower cost alternative and is suitable for the SMT function as is the HPC. The HPC also includes an on-chip 1 MHz serial interface and an on-chip UART that could be used for a local RS-232 connection. This connection can be used for diagnostic purposes, or for connection to a network analyzer console.

Backplane Design

The cards plug into a common backplane as shown in *Figure 4-4*. The backplane provides the interconnection between the various cards and provides a common clock reference and the serial bus. Less than 50 signal pins are necessary, all at or below 12.5 MHz so no sophisticated backplane design is required.

At each slot, two complete token paths are provided. The 10-bit internal National code is sent over the backplane. For each path, 10 input and 10 output signals are used. For each slot, 40 signals are present. The output of one slot is connected to the input of the next slot in order to create the token paths. The two physical paths are used by SMT as in the single board design.

A backplane design can use special shorting connectors on a terminator card which would keep the daisy chain paths connected when no board is plugged into a slot.

In a single processor design, the backplane must also provide a control path for inter-board communications. The control bus would have to be extended across the backplane in order to provide access to all of the devices. Depending on the capabilities of the main processor, the limitations of its response time, the desired extendability of the concentrator, this may or may not be warranted.

The requirements placed on the serial bus/ring are rather minimal if partitioning of the SMT functions is done well. Typically, all of PCM would be done locally on each M Port Card and the simple portions of CMT as well. One very simple protocol to run on a serial bus uses a two or three wire interface where a main processor polls the other processors in the system, one at a time. Small (size and cost) RS-485 transceivers such as National's DS75176B or DS3695 can be used for a simple interface to the backplane.

Concentrator Extensions

The backplane defined could be viewed as links between stations. The SMT standard only makes assumptions about the MAC at the exit ports and makes provisions for additional MACs within concentrators. This allows additional cards to be developed to provide a bridging function either to other rings, or even between paths.

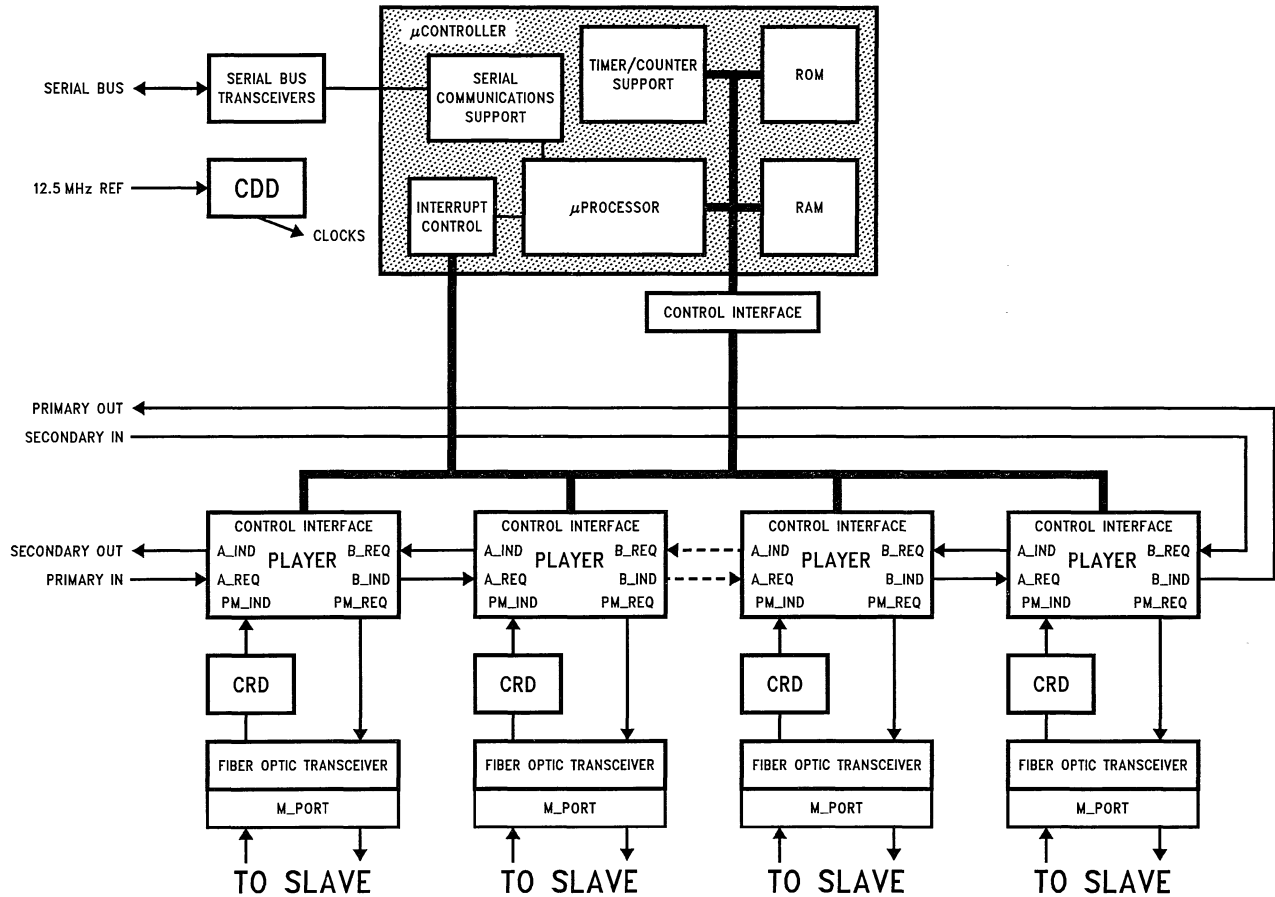


FIGURE 4-6. Master Port Card

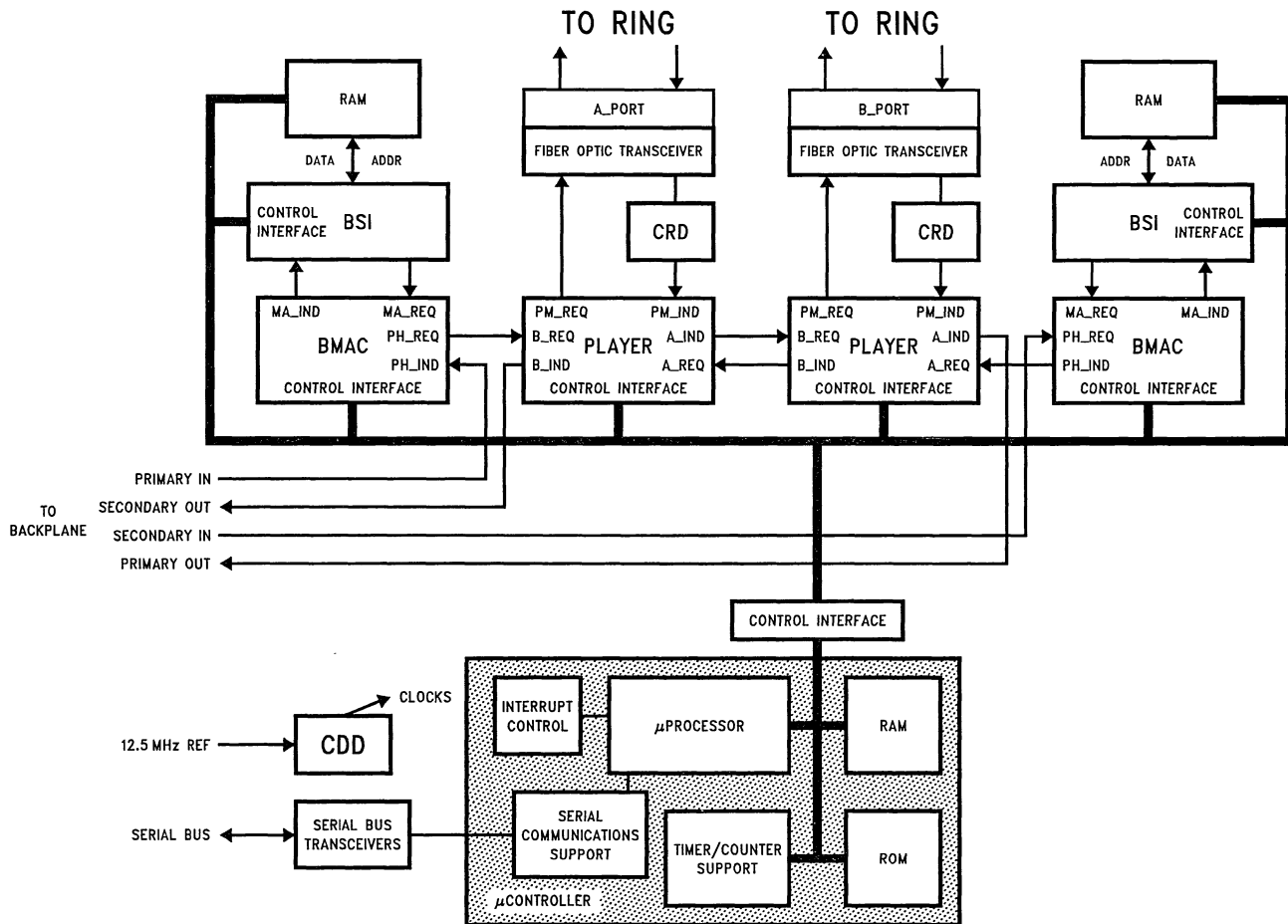
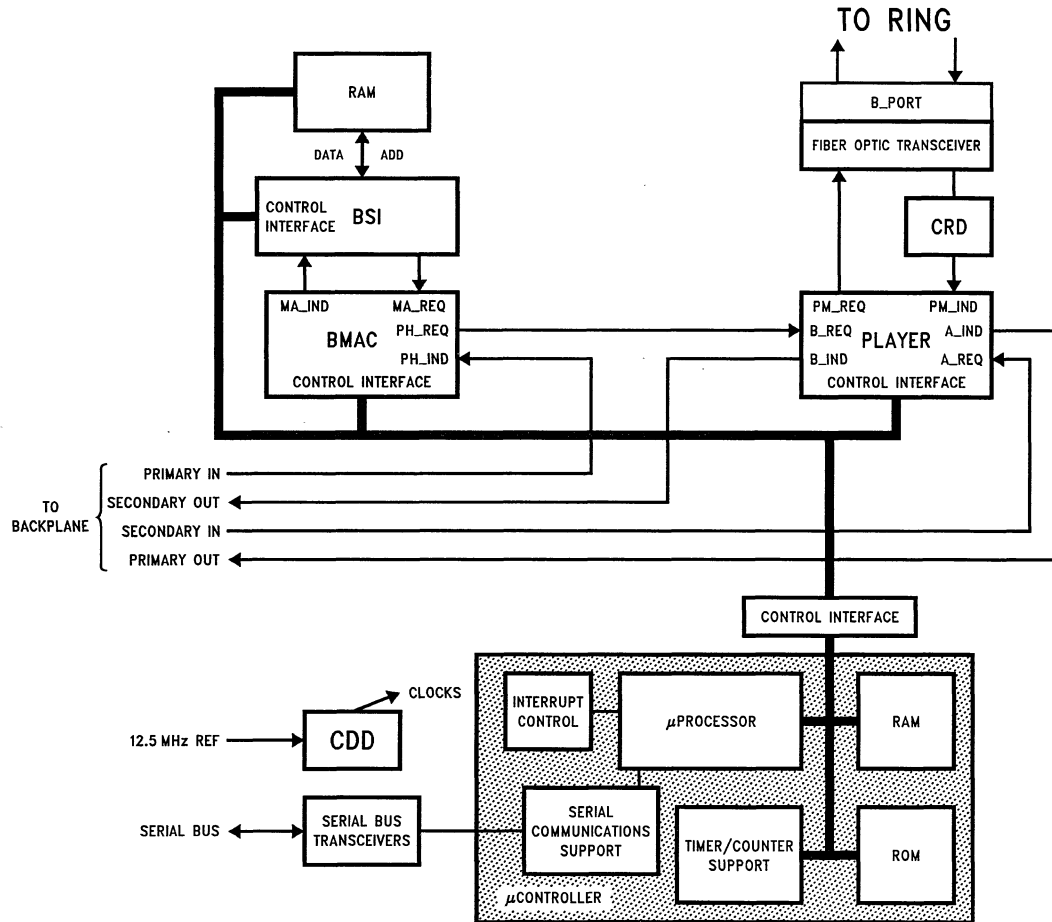


FIGURE 4-7. Peer Port Card

TL/F/11120-7



4-24

FIGURE 4-8. Port Board

Another possibility for add in cards for a concentrator are special network monitoring cards and network testing cards. Another extension is the ability to support live insertion of boards to avoid interruptions. This might even involve intelligent backplanes that could involve sophisticated multiplexing functions.

A standard for a concentrator backplane is a likely possibility at some point so that many different cards could be developed. This would allow customers to build concentrators with boards from many different vendors just as standard buses are used today to create customized computing environments.

This does cause one problem, when does a concentrator stop being a concentrator?

5.0 SMT NOTES

Several topics relating to implementations of Station Management for a Concentrator are worthy of discussion, many of these are covered below. Some of these topics relate to SMT implementations for stations as well.

5.1 Processor Performance Requirements

The most stringent requirement on an implementation of PCM using the PLAYER device is the Active to Break transition. The Standard requires that this transition takes place within 3 ms. This implies that within 3 ms after an interrupt is generated by the PLAYER device, a register is written to send out Quiet symbols and change the programming of the configuration switch. By using a low end processor to control several M Ports, the requirements on the main processor are reduced significantly. When bringing links up and going through the PCM signalling, the timing constraints are much less severe since there are no requirements for bringing up multiple links simultaneously. It is acceptable to distribute processing with several PLAYER devices controlled by a single processor.

5.2 Path Test

A complete path test in a concentrator is invaluable for isolating problems. Especially in large configurations, the ability to isolate errors, such as marginal connectors or connections, faulty components, etc., is extremely important.

The two complete token paths suggested in both example designs allows operational rings to be created within concentrators to check connectivity and components. The DP83200 chipset provides numerous features that aid in isolating problems down to the chip level. Modes such as 4 loopback paths, error insertion and statistic counters are just a few of the features included that give true network/system testing capability without complex external circuitry or equipment.

By having two complete paths, if a fault exists, the fault can be isolated precisely by careful programming of the configuration switch (see *Figure 4-3*). Once it is known which configurations are physically possible, it then becomes necessary to decide which paths should be connected and when this should be done. The current configuration must always be valid and up to date (there are still conflicting views on how this information should be kept in the MIB).

5.3 Moving MACs between Paths

In designs with fewer MACs than paths, it is necessary to move the MAC between paths. In many cases it is easier to just put a MAC on each path than to build all of the necessary software interlocks.

Before moving a MAC from one path to the other, it is necessary to remove all frames transmitted by this MAC in order not to create ownerless frames on the path that the MAC is leaving.

With the BMAC device, one way to accomplish this is to enable the Inhibit Token Capture option (Option.ITC) for at least one ring latency (after this station issues the token, if it is being held) before changing paths. Before leaving, it is also helpful to capture the token in order to avoid corrupting any frames. When placing the MAC on the new ring, it is necessary to insert while claiming. It is not possible to guarantee graceful entry if a token is not being held on that path.

When the situation exists for this station to scrub the ring of ownerless frames, several options are provided. The BMAC device provides a robust stripping mechanism that can be used without any destructive side effects, namely the STRIP option where stripping continues until a My_Void frame comes back around the ring, thus stripping all of the ownerless frames. This occurs automatically after this station wins Claim before the first token is issued. Similarly, the Inhibit Repeat option or the ability to block the MAC input from the PLAYER device for more than one ring latency also provide this function.

5.4 Graceful Insertion of Slaves

The graceful insertion (also known as smooth or bumpless insertion) of slave stations or slave concentrators into a ring allows stations to be placed into a ring without disrupting other stations in the ring. Unlike a ring consisting only of Dual Attach Stations where the ring must go through at least the Claim Process on each insertion/deinsertion and possibly cause frames to be corrupted, with concentrators it is possible to provide the capability to insert and deinsert stations gracefully without any data corruption.

Two methods of varying complexities are shown for graceful insertion.

With the first method to gracefully insert a slave the following must be accomplished:

- 1) make the slave ready to bring the ring operational on the next received token
- 2) capture a token on the path on which the slave will be inserted
- 3) while holding the token, modify configuration switch of the M Port to insert the slave
- 4) hold the token for at least actual ring latency of the new ring (the inserted slave concentrator might be bigger than the rest of the ring)
- 5) issue the token

Step 2 above requires a MAC on the path on which the slave will be inserted. The slave is inserted into the ring in step 3 above. The token must be held for a ring latency in order to guarantee that no frames are corrupted. This could occur if a slave connected to this concentrator transmitted a frame to an upstream slave connected to this concentrator. To avoid a TRT expiration in a downstream station while holding the token, a pool of synchronous bandwidth could be allocated and shared by all concentrators on the ring. This can be done using asynchronous transmission with THT disabled (one of the service classes provided by the BMAC device).

Depending on the assumptions you can rely on concerning a slave station after PCM, Step 1 may or may not require a MAC on a local path in a concentrator. If you can make the assumption that the slave will not enter claim because of TVX expirations and that its TREQ is set to TMAX, then a MAC would not be required. Otherwise a MAC would be required.

With this method, the theoretical possibility that a station will not accumulate lateness properly and will cause an unnecessary entry to Claim still exists. In the MAC, TRT is not cleared when a late token arrives, only late count is cleared. This means that if the ring is operating at heavy load, all stations in the ring will have some accumulated lateness. An inserted station cannot possibly know this and may hold the token longer than it would have if it had accumulated lateness. If it does, then a downstream station that had been accumulating lateness will consider the lack of a token arrival as a reason to enter Claim. Since this condition is very load and location dependent, and since Claiming is the only consequence (which it would be anyway if graceful entry was not attempted) it certainly doesn't hurt to try graceful entry. It is analogous to building a cache, if you can build one with a high hit rate (i.e., gracefully insert), do you? Implementors will need to decide if the payoff from graceful insertion is worth the additional complexities.

Another method to gracefully insert would be to capture a token on the path on which the slave will be inserted on and then always have the slave station go into the ring claiming or beaconing. This avoids the problems of accumulated lateness, at the expense of bringing the ring down for a potentially longer time, however no data is corrupted and only one MAC is required. Again this brings up questions regarding the assumptions that can be made about the default parameters of the slave station.

More work is necessary regarding the assumptions a concentrator can make about slaves upon insertion and how slaves should be inserted. Unfortunately, most proposals require services that are currently optional in SMT.

For example, a station could come into a ring and always default to a dual homing configuration. This would mean that A Ports are always attached to the primary ring, B Ports to the Secondary and S Ports to the primary. The Slave would be attached with default parameters and a default configuration. Once inserted into the ring the Slave would contact the Master in the concentrator using a well defined group address. In this way the Master MAC would determine its slaves quickly, and the slave would determine its Master. After this discovery process they could then use the standard parameter management frames to exchange configuration information. The Master would also have the ability to modify the Slave's default parameters.

This is currently unworkable because the ability to receive SMT group addresses is an optional facility as are the parameter management frames. In a future revision of SMT this may change.

6.0 CONCENTRATOR CLOCKING CONSIDERATIONS

Poor layout or clock distribution can forfeit the low error rate benefits of FDDI's fiber optic links. The greater number of PHY elements in concentrators exacerbates any problems in these areas. National Semiconductor designed the DP83241 to simplify the clock distribution.

Concentrators have more stringent packaging demands than most end attachments due to environmental, modularity and reliability considerations. They may range from a single board desktop implementation with a small number of master ports to large modular designs that support hot insertion and removal of master port cards. Though systems at these two extremes require different clocking structures, the DP83241 Clock Distribution Device provides reliable solutions.

The BSI, BMAC and PLAYER devices require 12.5 MHz TTL clocks for internal operation and communication between one another. The PLAYER device also requires a 12.5 MHz ECL clock to drive the parallel to serial conversion in the transmitter and a 125 MHz ECL clock to shift each bit out of the transmitter. The PLAYER device imposes specific timing relationships among these clocks, with relatively tight tolerances between the two ECL clocks and substantially relaxed tolerances between the two 12.5 MHz clocks. The CDD device guarantees the relationships of the ECL and TTL clocks that it generates to significantly tighter tolerances than the PLAYER device requires. A concentrator clocking structure must distribute and deliver these signals without introducing more skew than the PLAYER device can accept.

Appropriate layout and loading considerations must be followed in order to avoid unnecessary noise and skew. The PLAYER device at each port requires 12.5 MHz ECL and TTL clocks with a fixed relationship between them that must be preserved. In addition, the 125 MHz signals used for serial transmission have Standard defined jitter requirements.

In a single board concentrator, a single CDD device could be used to drive all devices on the board provided that the skews between the various clocks can be matched. Alternatively, one CDD device could be used for every two ports and a 12.5 MHz reference to be distributed around the card. This approach also minimizes the layout constraints as all clock lines can be kept as short as possible, with less variability. The layout recommendations for Dual Attach Stations can be applied.

In a multi-board design the above principle can be extended and on the backplane only the 12.5 MHz signal (from one of the Peer or Slave Port Cards) need be distributed to all boards in the concentrator. On each board the local CDD devices use it as a reference and generate all other clocks.

Flight Time Cancellation

Each board passes signals to the next downstream board and the propagation time between the adjacent boards is accounted for in the Set Up and Hold Times. Note that only one load is present for each board, not multiple loads, even in a fully loaded backplane. However at the endpoints of the backplane where the signals must be looped around, the margin of the setup and hold times may not be sufficient to accommodate the propagation time. In concentrators with less than 4 slots this may not pose a problem, but in concentrators with more than four slots, the backplane flight time, when added to the chip propagation delay from the local byte clock, would exceed the setup requirements at the next downstream port. This case therefore requires special attention.

The flight time that we are concerned about is the signal propagation time on the backplane of the return path. This propagation time needs to be measured once for each backplane. To compensate for this propagation delay, the CDD device provides ten different phases of the 12.5 MHz local byte clock. The appropriate phase of the CDD device could then be used to latch the data at the input to the next downstream card. The correct phase is chosen to match the set up and hold requirements of that latch such that the set and hold requirements of the next port on the PLAYER or BMAC devices are met. Five of the ten phases of the CDD device are selectable at either 8 ns or 16 ns increments to maximize the flexibility and range of its use in compensating for this design constraint.

In summary, for each path in a concentrator, an additional latch is used at the first input after the return path and is clocked by an appropriate phase of the 12.5 MHz LBC. For all other backplane connections, this additional latching is

not required because the Setup and Hold times provide sufficient margin to account for clock skew and propagation time. (If this is not the case, additional latching may be required.)

Conclusion

The Concentrator creates additional degrees of configuration and greater reliability for FDDI networks. The National Semiconductor DP83200 FDDI chipset provides the basis for a variety of robust concentrator designs and offers many ways in which National's customers can differentiate their products.

Acknowledgements

Thanks to Gabriel Li, Dave Talaski, Nanette Talaski, Mehdi Salamat, Michael Yipp, and Simon Stanley of National Semiconductor, Bob Grow and Ron Perloff of XDI and Jim Hamstra of Stanatek for their assistance in this effort.

Incorporating FDDI MAC Level Bridging

National Semiconductor
Application Note 740
David Brief



INTRODUCTION

A natural application for FDDI is as the backbone LAN in large installations with several FDDI networks and many segmented lower speed LANs. The interconnection of these networks is typically accomplished with MAC Layer Bridges and Network Layer Routers.

This Application Note focuses on the unique aspects of MAC layer bridging for FDDI and suggests ways of incorporating these functions into products using the National Semiconductor DP83200 FDDI chipset. The DP83200 chipset provides many capabilities that ease the job of incorporating MAC Layer bridging functions into products. It is likely that MAC level bridging capabilities will be resident in products together with Network Layer Routing Functions, FDDI Concentrator Functions and/or End Station capabilities.

TABLE OF CONTENTS

1.0 MAC LEVEL BRIDGING CONSIDERATIONS ON FDDI

- 1.1 Bridging Protocols
- 1.2 Filtering and Forwarding
- 1.3 Setting and Interpreting the Control Indicators
- 1.4 Stripping Transmitted Frames
- 1.5 Bridgeable Frame

2.0 DP83200 CHIPSET BRIDGING FEATURES

- 2.1 Source Address Transparency
- 2.2 Stripping Mechanism
- 2.3 FCS Transparency
- 2.4 External Matching Interface
- 2.5 Data Structures

3.0 IMPLEMENTING TRANSPARENT BRIDGING

- 3.1 Bridge Block Diagram
- 3.2 Address Matching Alternatives
- 3.3 Developing the Addressing Topology
- 3.4 Void Stripping and Learning
- 3.5 Special Control Indicator Processing

4.0 TRANSPARENT BRIDGING IMPLICATIONS FOR END STATIONS

- 4.1 Indicator Setting
- 4.2 Interpretation of Control Indicators

5.0 IMPLEMENTING SOURCE ROUTING BRIDGING

- 5.1 Address Filter
- 5.2 Discovery Process
- 5.3 Forwarding
- 5.4 End to End FCS Checking

6.0 SOURCE ROUTING BRIDGING IMPLICATIONS FOR END STATIONS

1.0 MAC LEVEL BRIDGING CONSIDERATIONS ON FDDI

The presence of bridges in an FDDI network should enhance the connectivity and performance of the overall network. This should in turn be reflected in the performance seen by end stations.

MAC layer bridging introduces several interesting quirks into the FDDI protocols. These are largely because many of the MAC layer bridging protocols were developed in environments where frame stripping was implicit (In the Ethernet bus topology, packets just disappear and in Token ring there initially was no early token release) and control indicators were not used or were not present.

There are also several options for implementing MAC layer Bridging. In this section, the issues involved with MAC Layer Bridging are discussed in general terms. In subsequent sections, the implications of these topics when using Transparent and Source Routing Bridging is discussed in further detail.

A single bridging method is not defined in FDDI since it is possible to use the same bridging protocols as in the IEEE802 LANs. However, in the areas that relate to interoperability and operation of the FDDI protocols, compromises have been reached on how to run these bridging protocols in FDDI. These compromises are captured in the ANSI X3T9.5 MAC-2 Draft Standard.

1.1 Bridging Protocols

There are two categories of MAC level bridging that are typically performed. They are named according to how much knowledge, responsibility and control an end station (e.g., workstation) has over the routing of the frame through the network. In transparent bridging, the end station has little if any control over the routing of the frame to its destination and is free from all responsibility and work of determining the route to the end station. In Source Routing bridging, as its name implies, the source is responsible for routing a frame to its ultimate destination. To end stations, the bridging function that is accomplished somewhere in the path between the two protocol endpoints appears as part of the network. This bridging function may be accomplished by one or more stations that implement the bridging function. These stations are generically referred to as bridges even though those stations do more than just the bridging function. For example, every station that implements a bridging function also acts as an addressable end station.

With Transparent bridging, the bridges are responsible for maintaining all of the routing information. The most popular, but not the only form of transparent bridging uses what is commonly known as the Spanning Tree protocol for determining how to route frames to their destination. The goal of the Spanning Tree protocol is to create a single path to connect all stations in the connected network at any given time. The Spanning Tree protocol allows extensions for multiple paths between segmented LANs. These paths can be used as backup links or alternatively to provide a form of load sharing. The load sharing is possible across different dialogues, but not across a single dialogue.

Transparent bridging protocols reduce the complexity of the end station while increasing the complexity required by the bridge function to retain the illusion of end station transparency.

With Source Routing, the job of the bridges is simplified, but the end stations must determine and include explicit routing information in every frame that they transmit. Since a station typically communicates with a small number of stations, and the routing to these stations is relatively static, this does not typically represent a large overhead for end stations.

The IEEE802.1d committee reached a compromise in order to allow Source Routing and Transparent bridging to work in the same extended LAN. The committee has specified a Source Routing-Transparent (SRT) bridge. With this compromise, end stations that participate in source routing protocols also can communicate with stations using transparent bridging provided an SRT bridge is present. For End Stations capable of source routing, this effectively makes all stations connected through the Spanning Tree appear as if they are in the local Source Routing Domain. There is little affect on bridges and end-stations that perform only transparent bridging. For more information on SRT bridging see the recent IEEE802.1d document.

1.2 Filtering and Forwarding

The decision to forward a frame from one port of a bridge to another is based on the addressing information contained in the frame. The address is used as an input to an address filter which contains the necessary information to determine if a frame should be copied and forwarded or rejected and filtered.

Transparent bridging uses the Destination Address field of an FDDI frame to determine if a frame should be forwarded. In a multi-port bridge, the Address filter might also indicate to which port to forward the frame. The set of addresses for which frames are forwarded is configured in the address filter by a learning process. In the future, information contained in the SMT Management Information Base might be used to load this Address filter when all information necessary to develop and predict ring addressing topologies is present through standard management services.

Transparent Bridging might also be implemented with a partial filter where some or all of the frames are copied. Further address filtering then indicates whether or not the frame should be forwarded.

In Source Routing Bridging, the decision to copy a frame for forwarding is based on its 16-bit ring address being listed in the routing information field contained within the INFO field of the FDDI frame. The presence of this routing field is indicated when the most significant bit of the SA field is 1. End stations participating in the Source Routing protocol are only required to recognize their own address as the destination and are not required to process the routing information field for purposes of forwarding.

1.3 Setting and Interpreting the Control Indicators

The meaning of the Address Recognized and Frame Copied control indicators (A__Ind and C__Ind) is impacted in an extended LAN. After a long and sometimes heated debate over the meaning of this frame status information, the FDDI Committee reached agreement on a compromise solution.

The A Indicator, when set, indicates that a precise address match occurred. The C Indicator, when set, indicates that the frame was copied successfully by a station and that either the frame will be delivered to the correct protocol endpoint or an error will be indicated back to the sending station.

Source routing bridges set both the A and C Indicators if it recognizes and copies the frame for forwarding just as if its

address had appeared in the DA field. End stations can therefore assume that if transmitted frames return with both the A and C Indicator set, that the frame is being forwarded to its ultimate destination. If a transmitted frame returns with the A Indicator set but the C Indicator as reset, then the station might assume that buffer congestion has occurred somewhere between the source and the destination buffer. The end station could use this as an indication to stop transmitting any more frames because there is likely buffer congestion somewhere in the path and additional frames that are transmitted would probably not be copied anyway, and would waste system and ring bandwidth. The exact interpretation and the recovery implied is dependent on the protocols used. Protocols that indicate back pressure in an extended LAN are still largely proprietary.

The interpretation of the control indicators when Transparent bridging is present is slightly more complicated. A transparent bridge will only set the A__Ind if the frame is explicitly addressed to the bridge as before. However, for frames that are copied with the intent to forward the C__Ind may optionally be set. This capability is indicated in the FDDI SMT MIB.

If a transmitted frame returns with A__Ind = R and C__Ind = S, an end station can then assume that the frame has been copied by a bridge that intends to forward the frame to its ultimate destination. If a transmitted frame returns with A__Ind = R and C__Ind = R, an end station can either assume that the frame was not forwarded by a transparent bridge if it knows that only bridges that implement the option are in the route to the ultimate destination. This can be used to stop transmission of additional frames until the buffer congestion condition subsides. If bridges that do not implement the C setting option are in the route to the ultimate destination, then the station can not assume anything about the forwarding of the frame (as in Ethernet).

If a transmitted frame returns with A__Ind = S, an end station can assume that the frame was addressed to a station on the local ring. The interpretation of the C Indicator depends on the option implemented by the addressed station. An optional status clearing capability may be implemented in stations. In this case, a bridge that has not yet learned of a station on the ring can copy a frame for forwarding, before the destination receives the frame. When a frame is received by a status clearing station with the received A__Ind = R and C__Ind = S, a station implementing the option will change the C__Ind to reset if it cannot copy the frame. The DP83261 BMACTM device implements this option. This capability preserves the meaning of the A__Ind = S, C__Ind = R combination on a local ring, and the transmitting station then knows that the destination did not copy the frame. In this way the presence of transparent bridges implementing the first option will not destroy any status used in optimizing the performance on a local ring.

The implications of the compromise that was reached for transparent bridging is that end stations wishing to take advantage of the control indicators are required to keep status information about each station they are communicating with in order to determine how to interpret the control indicators.

1.4 Stripping Transmitted Frames

In FDDI, the MAC is responsible for stripping every frame that it transmits into the ring. In the case of both Source Routing and Transparent bridging, the SA and DA fields of the frame contain the original source and ultimate destination stations for the frame. Therefore, when a bridge forwards a frame onto an FDDI ring, it is not possible to use

the SA field to recognize frames to be stripped. For this reason, a bridge is required to implement an alternative stripping mechanism. Other stations may also find it useful to implement an alternative stripping mechanism.

The MAC-2 standard does not specify a single stripping mechanism, but rather suggests examples of a number of different stripping mechanisms that can be implemented within the bridge station. Any method which is interoperable with the rest of the FDDI Standards and meets several general criteria may be employed.

One property of the ring that is exploited to implement alternate stripping mechanisms is the property that all frames transmitted will return to the station that transmitted them before any frame transmitted by any other station is received. This allows the station to use a special frame to mark the end of one or more frames that are transmitted during a service opportunity (while holding the token). If this method is used, the MAC-2 Draft standard suggests the use of a special type of Void frame that contains this stations DA and SA, this is called a My__Void Frame. This is differentiated from regular Void frames from the non-Null DA. It is differentiated from Void frames from other stations (Other__Voids) by the presence of this stations SA field. By transmitting a My__Void frame before a token is issued, and stripping until it returns, all frames transmitted by this station are removed from the ring in the forward mode of operation.

Errors in this process, such as the creation on corruption of a My__Void frame, result in either Understripping, where not enough frames stripped, or Overstripping, where additional frames are stripped unnecessarily.

Understripping is undesirable because it causes duplicate packets to be delivered to stations on the ring. Any algorithm that stops stripping upon receipt of a valid token cannot have a probability of understripping less than that of a token being erroneously created from a frame. This requires the conversion of the FC field to that of a token and the conversion of two consecutive data or Idle symbols to T symbols. Thus, a limiting factor to the probability of understripping is approximately:

$$1.25 \times 10^{-22}$$

In order to prevent understripping more than once in 100 years, the probability of understripping should be:

$$1.25 \times 10^{-22} \leq \text{probability (understripping)} \leq 1.3 \times 10^{-14}$$

Overstripping is also detrimental to ring performance. It causes frame loss that would otherwise not have occurred. Assuming any other factors which might cause frame loss (e.g., receiver congestion) are eliminated, the frame loss rate is bounded by the link bit error rate.

In order that overstripping not cause an excessive increase in total frame loss rate, it should not cause the loss rate to increase more than 10% above the loss rate due to link bit errors alone. Therefore:

$$\frac{\text{probability (frame loss due to overstripping)}}{\text{probability (frame loss due to link errors)}} < 0.1$$

Several mechanisms exist that might be used to meet these frame stripping requirements. A combination of mechanisms may be required to achieve adequate robustness in the stripping algorithm. Some examples, but not all, of mechanisms that might be used as components of stripping algorithms include:

- (a) Transmitting one or more My__Void frames before issuing the token to mark the end of a burst of transmitted frames, and stripping until a My__Void frame is received. (The BMAC device transmits two My__Void frames when the stripping option is invoked)
- (b) Stop stripping upon receipt of valid Tokens, other Void frames or MAC frames (e.g., Beacon or Claim frames). (The BMAC device implements all of these options)
- (c) Counting the number of frames transmitted and the number of frames stripped. (The BMAC device does not implement this option)

To ensure interoperability of different stripping methods, and to minimize under stripping or over stripping, the bridging appendix in the MAC-2 Draft Standard recommends that a bridge:

- (a) Transmit at least one My__Void frame immediately before issuing a token.
- (b) Stop stripping when it receives its own My__Void frame or any valid Void frame with SA other than its individual address.

In addition, an alternative stripping method is required to stop stripping when it receives a valid token or clears Ring__Operational.

Some stations, including stations using the DP83265 BSI™ device, are implemented in such a way that a shared resource (e.g., bus or memory bandwidth) is used at some point in a frame. To allow design of these stations without undesirable performance implications, the MAC-2 Draft standard requires stripping to begin no later than the seventh symbol after the end of the SA field.

In the BSI device, the copy decision can begin as early as the 4th byte of the INFO field. Since all fragments should be less than this size (they should have at most 7 symbols in the INFO field), fragments will not result in any wasted bus and memory bandwidth. Longer fragments caused by stripping errors with more than 4 bytes in the INFO field might be copied by the BSI device if an address match occurs. The termination status reported with every copied frame by the BSI device indicates these frames as being stripped frames.

For a frame whose SA is the MAC's individual address, it is only necessary to strip based on the recognition of the SA field.

1.5 Bridgeable Frame

The FDDI frame format defines the first byte of every frame transmitted as the Frame Control Field. The Frame Control Field is used to determine the Protocol Endpoint such as MAC, SMT, LLC Asynch, LLC Synch, Implementer, Reserved. Currently, only LLC Asynch frames are bridgeable. LLC Synch, Implementer and Reserved frames are beyond the scope of the standard. MAC and SMT frames must never be forwarded from one ring to another (this would cause serious problems for MAC and SMT protocols). Only LLC Asynch Frames are intended to be bridged.

The use of restricted dialogues that traverse more than one ring are also beyond the scope of the Standard.

Any LLC Asynch frame with at least 4 data bytes after the SA is considered bridgeable. In a worst case condition, many short frames with minimal preamble can be received. Using long addresses with 4 bytes of Info and B bytes of preamble, this corresponds to over 400,000 frames per second!

Large frames might also present difficulties for bridges. In Basic mode FDDI has a maximum frame size of 4500 bytes and in Hybrid mode FDDI has a maximum frame size of 8600 bytes. Without use of a segmentation/reassembly protocol, it is not possible to bridge frames that are larger than the size permitted on a LAN. For example an 8600 byte frame from a ring operating in Hybrid mode could not be forwarded into a ring operating in Basic mode. However, a 4500 byte frame could be forwarded in either direction. Similar problems appear when forwarding frames between Ethernet and FDDI.

2.0 DP83200 CHIPSET BRIDGING FEATURES

The DP83200 FDDI chipset incorporates several features to ease implementing bridging functions. These capabilities make possible simple and efficient bridge designs. The features are covered below followed by a description of how they are used in Source Routing and Transparent Bridges and End Stations.

2.1 Source Address Transparency

Normally the Source Address is transmitted from the MAC parameter RAM as added protection since the SA is used to strip frames. The SA transparency capability allows the SA to be transmitted from the Data Stream as opposed to from the parameter RAM. This is important since bridges work with the original source address on frames that are to be forwarded. This is different from the bridge station's address.

In the BSI device SA transparency is a channel parameter. Every frame transmitted on a channel either uses SA transparency or does not depending on the current programming of the channel configuration register. This register can be modified reliably between requests, not on a per frame basis. An implementation using the BSI device might transmit all of the bridged traffic on one channel and all of the local traffic on the other. (The BMAC device can handle changes on a per frame basis.)

The BMAC device also supports the capability to provide separate transparency control over the routing information indicator bit of the SA. For historical reasons the BMAC device signal is called SAIGT since its routing information indicator falls in the same relative location as the individual/group within the DA field.

2.2 Stripping Mechanism

In FDDI, every station needs to strip every frame it transmits. Typically this is accomplished by stripping based only on the Source Address. However, in bridging applications where the SA is not necessarily of this station, this station needs to either watch out for all of its frames (and use CAM technology to assist the strip decision) or use some other mechanism. The MAC-2 Draft Standard states that the strip points of all frames is before the fourth byte of the INFO field. That implies that any fragment with more than three bytes of INFO is an illegal fragment.

The National stripping mechanism accomplishes the stripping by sending out two My_Voids before the token and stripping everything until the first My_Void returns. The second My_Void is stripped on the basis of its SA, as in the normal stripping mechanism.

2.2.1 Algorithm

- Stripping of frames with SA received = MSA or MLA still occurs
- If the stripping option is requested by any Frame of a Service Opportunity (i.e., while a token is held)
 - at the end of the Service Opportunity (before the token is issued) two My_Void frames are transmitted.
- Stripping continues until:
 - a valid My_Void frame is received (normal case)
 - a valid Other_Void frame is received
 - a valid token is received
 - a MAC frame (other than My_Claim) is received

Stripping does not stop on receipt of a My_Claim in order to allow removal of all My_Claim fragments that would otherwise be present after the Claim token process is won by this station.

- a MAC Reset occurs

• Void Frames

Three types of Void frames are used in this algorithm, the Void frame, the My_Void frame and the Other_Void frame. During normal operation Void frames are transmitted when L_Max expires and when frames are aborted. My Void frames are transmitted at the end of a service opportunity when stripping has been requested for any frames of that service opportunity. Other Void frames are Void frames transmitted by some other station on the ring.

The My_Void Frame provides a convenient marker to delimit stripping. The My_Void frame is used to distinguish it from the Void frame.

Void frames are used within a service opportunity in order to make sure that valid frames are sent every F_MAX and to limit the maximum preamble size. This typically occurs after a frame is aborted or after the token has been captured when a frame is not ready to be transmitted.

- My_Void frame
 - FC = Void
 - DA = MSA or MLA (MSA used if enabled)
 - SA = MSA or MLA (MSA used if enabled)
 - FCS
 - ED, FS
- Void frame
 - FC = Void
 - DA = Null
 - SA = MSA or MLA (MSA used if enabled)
 - FCS
 - ED, FS
- Other_Void frame (Detection Criteria)
 - FC = Void
 - DA = any
 - SA = not ((MSA and MSA enable) or (MLA and MLA enabled))
 - FCS
 - ED, FS

2.2.2 Robustness

- Two My__Void frames are used to improve the robustness over using just one My__Void Frame
- Overstripping
 - Overstripping will only occur when both My__Void frames are corrupted or lost
 - even in this case, overstripping will be limited by Other__Void frames which could occur before during or after the next station's burst of frames.
 - assume
 - that noise in receiver major course of error this implies that there is no correlation of errors across frames.
 - maximum ring load
 - 100 stations
 - 100 byte frames
 - 1 frame per station per token
 - 10 e^{-10} link error rate
 - with one My__Void frame
 - overstripping occurs once every 13 minutes
 - with two My__Void frames
 - overstripping occurs once every week
- Understripping Probability:
 - this would only occur if
 - some other frame turns into a well formed My__Void or Other__Void frame
 - a My__Void or Other__Void is transmitted from the data interface.

2.2.3 Affects on Synchronous Allocation

The bandwidth used by the two My__Void frames is taken from the synchronous bandwidth allocation and effectively reduces the maximum allocatable synchronous bandwidth by the time it takes to transmit the two My__Void frames. The My__Voids can be sent with the short address to minimize the use of ring bandwidth. With the BMAC device, if short addresses are enabled, the BMAC device will transmit My__Void frames with the short address.

The bandwidth required from the synchronous pool is independent of the number of stations on a ring using the Void Stripping option and is dependent only on the maximum number of My__Void frames that can be transmitted before the token is issued. If all bridges in the ring utilize the BMAC device, then only 34 byte times of the synchronous allocations needs to be used.

In a ring where both Synchronous and Asynchronous services are being used, if THT expires while transmitting a frame, two My__Voids will then be transmitted followed by the token. At the next station and all downstream stations THT will have already expired and only synchronous requests may be serviced. In this way the token will go all the way around the ring until it reaches the station after the station where THT expired during the frame transmission. All other stations will have the opportunity to send all of their synchronous traffic. In the case where all of the stations using synchronous bandwidth line up perfectly and use all of the synchronous bandwidth on one token rotation, the recovery required conditions will still not become true at any station provided that the synchronous bandwidth was not overallocated.

2.2.4 Why Not Use a Frame Counter?

Our approach is easier to implement and less likely to overstrip than a stripping mechanism that uses a frame counter, in the presence of line errors.

The first My__Void frame provides equal protection with or without a frame counter. However, a frame counter will miss on a hit in any data frame that causes that frame to be lost. A second Void frame will only be missed on a hit on the Void frame itself. Since the length of the Void frame is significantly less than the total length of the transmitted data frame(s), it is significantly more likely that the counter will miss a data frame due to noise than that a Void frame will be missed due to noise. This implies that a frame counter approach is more likely to cause understripping than an approach using a second My__Void.

For additional protection, stripping is stopped on receipt of any well-formed Void frame from another station, which minimizes interference with other implementations and avoids cascaded overstripping when multiple stations are stripping and multiple My__Void frames from one station are lost. With the frame count approach, if both data frame(s) and the single My__Void frame are lost (much more probable than losing a pair of My__Void frames), overstripping could eat up another station's My__Void frame, as well as its data frame(s). This can cause cascaded overstripping unless additional logic is added to selectively strip frames based on their PC values. Other implementations may use some kind of selective strip filter logic; however, this results in fragments on the ring, whereas our strip mechanism can be used to clean up all garbage except multiple valid MAC or well-formed void frames (the first such frame is stripped), or tokens (which are stripped but regenerated).

2.3 FCS Transparency

FCS Transparency determines if the FCS calculated by the CRC Generator in the MAC transmitter will be appended to the frame. This allows diagnostic testing of the CRC checker and generator. This option is also used to perform end to end FCS checking. This could be used in FDDI to FDDI transparent bridges, but is not useful in transparent bridges between FDDI and Ethernet because the FC field and bit ordering of the SA and DA change the value of the FCS.

2.4 External Matching Interface

The BSI and BMAC devices accept several inputs from an external address filter in order to cause frames to be copied based on external address matches and also to affect the setting of the control indicators and incrementing of the frame counters.

The External A flag (EA) signal to the BMAC device causes the A Indicator to be set and the frame copied count to be incremented if the frame is successfully copied. The External A flag (EA) signal to the BSI device causes the frame to be copied when asserted from the beginning of the frame until ECIP is deasserted.

The External M__flag signal to the BMAC device causes the current frame to be stripped. This is useful if an alternative to the resident Void stripping algorithm is used. The External M__Flag signal to the BSI device is used when confirmation is used with external address matching.

An implementation of an address filter would typically use the interface between the BMAC and PLAYER™ devices. This address filter would detect a starting delimiter and then sequence into the frame.

The reason for using the interface between the BMAC and the PLAYER device is to allow a few more bytes of look ahead and also to be in line with the future DP83200 chipset integration strategy.

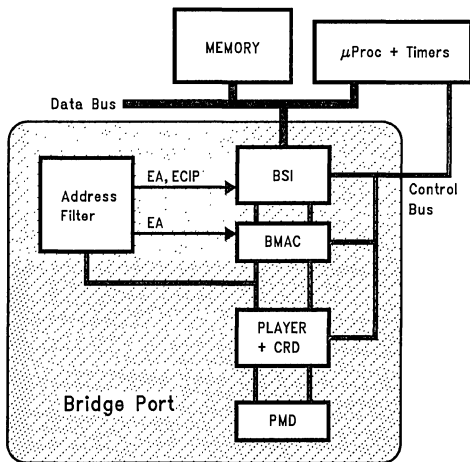
2.5 Data Structures

The data structures used by the BSI have been optimized to allow efficient transfer of data. The Output and Input data structures are symmetric and the data and descriptors are segregated.

3.0 IMPLEMENTING TRANSPARENT BRIDGING

Bridging that appears transparent to end stations can be considered the most difficult to implement. In this section topics relating specifically to transparent bridging are discussed. In subsequent sections, new topics relating to end stations and source routing bridges are discussed.

3.1 Bridge Block Diagram



TL/F/11119-1

FIGURE 1. Bridge Block Diagram

The block diagram in *Figure 1* shows a generic architecture for a multi-port bridge, highlighting the requirements for a single port in the bridge.

3.2 Address Matching Alternatives

In any bridging application, additional address recognition logic is required. This may range from totally promiscuous copying with filtering only in Software, to combinations between Hardware and Software, to complete filtering in real time with dedicated Hardware.

The advantage of doing the address matching in real time in dedicated hardware is to be able to set the Address recognized indicator accurately (in explicit bridges) and to set the Frame Copied indicator appropriately (when the intent is to forward the frame).

The address matching logic, the Address Filter, can either be implemented in very wide CAMs (to handle 48-bit addresses) or can be implemented in sophisticated SRAM lookup tables. One way of using an SRAM to get a reasonable address density is to use each successive byte or nibble as an index into the RAM implementing a TRIE structure

(M-ary tree) in hardware. Each successive byte or nibble is combined with data from the previous cycle and used together as part of the address for the next cycle.

Bridges that can handle 400,000 frames per second containing over 4K addresses are becoming available.

As an alternative to Forwarding Bridges where for each received frame, the bridge determines which port, if any, to forward it to, there are also Filtering Bridges. Filtering Bridges reject frames with certain addresses only and accept all others. For example, it is reasonable to reject all frames with an address on this ring (in the ring map), and copy all others. This approach is particularly appropriate in designs with two ports since there is only one place that the frame can be forwarded to. In bridges with more than two ports, it is appropriate if the forwarding decision is made using additional software support.

Similar logic is used to provide additional Group Addressing Capabilities and possibly Individual Addressing Aliasing.

However the address filter is implemented, the BMAC device relies on the External A flag signal (EA) in order to set the transmitted A Indicator. Similarly, the BSI device expects the External A flag signal (EA) combined with a latching signal, ECIP, External Compare In Progress. When ECIP is deasserted EA is sampled to determine if the frame should be copied by the BSI device. The BSI device will copy the frame to either Channel 1 or Channel 2 depending on its current configuration.

3.3 Developing the Addressing Topology

In order to correctly determine which frames should be copied and forwarded and which frames should be rejected and filtered, it is necessary to develop an accurate addressing topology. Once the addressing topology is established, it can be used to determine the location of every address and the route (one or more) to every address from any other address. In Spanning Tree transparent bridging there is one route between any address pair, and in most implementations just the next hop needs to be known, not the entire route.

The addressing topology can be developed in several ways. One way, popularized by Ethernet networks where there is no alternative, is to effectively learn the topology by watching the traffic. By seeing what frames come from where, it is possible to deduce what side of the bridge or to which port frames are connected.

In FDDI networks, it is also possible to develop the addressing topology using information stored in the SMT Management Information Base. Each station is responsible for maintaining a list of its Aliases. It is not clear however if all of the proper information is currently available in the SMT MIB to develop an accurate addressing topology for transparent bridging. This is a topic for future work.

Once the addressing topology is developed, it is used by the address filter to determine which frames to copy (and which port to forward them to) and which frames to reject.

The addressing topology must be maintained using some kind of aging or refreshing process. Changes in the addressing topology are relatively slow, except in the presence of configuration changes. In a configuration change (such as a wrap or an additional wrap), a small adjustment can have a large affect on routing between address pairs.

3.4 Void Stripping and Learning

When using Void Stripping, it is desirable to inhibit copying of frames transmitted by this station. A learning algorithm that monitors frames that are received to determine their source would get confused if transmitted frames are also received. In order to implement this function, additional logic is necessary between the BMAC and BSI devices to monitor when a frame is transmitted with the strip option.

In order to inhibit copying, a Stripping Flag is suggested. While this flag is set, no frames should be copied. The STRIPPING flag is set when a STRIP is asserted and TXPASS = 0 and TXRDY = 0. The STRIPPING flag is cleared when one of the following conditions occurs:

- a valid My__Void frame is received [normal case]
- a valid Other__Void frame is received
- a valid token is received
- TXRINGOP of the BMAC device is deasserted (this includes the cases where a MACRST occurs and MAC frames are received).

This requires a station to decipher the Void and Token Frame Control Fields and also monitor the BMAC device AFLAG and MFLAG signals to determine the Void frame type. Use of the BMAC device TXRINGOP is also necessary. The plan for the integrated solution is to provide this stripping flag as an output to be used in external copy decisions.

3.5 Special Control Indicator Processing

For Stations implementing transparent bridging special handling of the C Indicator is allowed by the MAC-2 Draft Standard.

The rules for setting the A and C control indicators are as shown below:

- if the frame is addressed to this station then
 - set A Indicator
 - if frame copied successfully then set C Indicator
- if the frame is not addressed to this station then
 - leave A Indicator unaltered
 - if A Indicator received as R and frame copied optionally set the C Indicator

For frames copied with the intent to forward for which a station address did not match, only the C Indicator should be set, not the A Indicator.

To accomplish this, since the current BMAC device will not set the C Indicator without setting the A Indicator as well, it is necessary to intercept the PHY Request byte stream between the BMAC device and the PLAYER device. Fortunately, the difference between the R and S symbol is only a single bit. Thus, when a frame is copied with the intent to forward it, the received A Indicator was not set, and the station's AFLAG is not set, the C Indicator must be changed from an R to an S. This occurs one byte time after EDRCVD is asserted and requires that PRD(0) be changed from a 0 to a 1.

This is only required in bridges implementing this option. In the integrated solution, an extra pin will be provided to help implement this function and remove the requirement for this small amount of external logic.

4.0 TRANSPARENT BRIDGING IMPLICATIONS FOR END STATIONS

Although the goal of transparent bridging is to remove all implication from end stations, there are a few areas that are affected as indicated below.

4.1 Indicator Setting

For End Stations and Bridges which also act as end stations, the BMAC device always implements the optional status clearing function defined in the MAC-2 Draft Standard.

Specifically, for frames received with A__Ind = R and C__Ind = S for which the station recognizes the address and intends to set the A Indicator but the frame was not successfully copied, the A Indicator is transmitted as S and the C Indicator is transmitted as R.

Support by the BMAC device of this function is in line with the desire to support all end station requirements very well and support other station types (bridges, routers, concentrators) with minimal additional logic.

4.2 Interpretation of Control Indicators

Because of the optional capabilities allowed by the MAC-2 Draft Standard for setting and clearing of the Control Indicators, their interpretation is slightly different depending on the current operating environment and the options that are implemented by stations and bridges in the LAN.

In order to make optimal use of the control indicators to accelerate throughput and run at data throughput rates close to the optimum allowed through the physical constraints of the data paths, it is necessary to know what options are being implemented by all stations participating in any particular data transfer. This may amount to as little as one bit of state for each end station with which this end station communicates.

What is of use is the knowledge that when the C Indicator comes back as R, whether that station should refrain from transmitting any additional frames. This can be used to implement effective flow control schemes that do not transmit frames when it is likely that they will not be able to be copied anyway. In these scenarios, not only is additional ring bandwidth saved, but system bandwidth in the transmitting and receiving station is saved as well. Only when a station is running close to the maximum throughput for a certain address pair can it know that it is taking full advantage of the throughput of the logical link. The BSI device provides the confirmation services necessary to stop transmitting additional frames if the expected frame status is not received.

In extended LAN environments, bridges must also be able to turn off forwarding between certain address pairs when congestion is detected.

5.0 IMPLEMENTING SOURCE ROUTING BRIDGING

The high level block diagram for a multi-port Source Routing Bridge is remarkably similar to that of the Transparent Bridge. The major differences come in the responsibilities and implementation of the Addressing Filter.

What is new with Source Routing Bridging is the presence of the Routing Information Field. The presence of the Rout-

ing Information Field is denoted when the most significant bit of the SA field is set to One. The Routing Information Field contains a string of 16-bit bridge numbers that the frame is to be routed to. These 16-bit bridge numbers are considered aliases of a certain bridge.

Source Routing Bridges fall under the category of explicit bridges used in the MAC-2 Draft Standard. In explicit bridges, the addresses that the bridge recognizes are considered its aliases, and the A Indicator is set for this class of address matches. (For this reason, in these implementations it is possible to connect the EA signals of the BMAC and BSI devices together.)

5.1 Address Filter

The Address filter is much simpler in Source Routing bridges than in transparent bridges. The address filter is required to parse the Routing Information Field, when present, and look for this bridge's 16-bit ID number. If the number is present in the Routing Information Field then the frame is copied and the control indicators are set appropriately using the EA and ECIP inputs to the BMAC and BSI devices. The frame is then forwarded to the next destination in the list.

5.2 Discovery Process

In order for an end station to determine the route to another end station, the discovery process is necessary. In one variation of the discovery process, a station transmits out several all route frames. Each bridge then adds its bridge number into the Routing Information Field, until the addressed station has been reached. The addressed station then transmits back a frame according to the "best" route. This route is then used for future transmission.

5.3 Forwarding

Once the bridge's, bridge number is detected in the Routing Information Field, the frame can be forwarded to the next bridge number using the appropriate port (in a multi-port bridge).

Since all frames that are transmitted, must be stripped before the 7th symbol of the INFO field, the VOID stripping still must be used, because it would not be tolerable to strip based on a bridge number later in the frame.

There are some subtle requirements placed on the transmission of frames in Source Routing Bridges. For example in some frames, the Most Significant Bit of the SA remains unchanged for some frames and is forced to zero in others. The SAT and SAIGT BMAC device input signals can be driven from the SAT signal on the BSI device. In addition, the STRIP option also needs to be used to strip frames forwarded by this station.

5.4 End to End FCS Checking

Between FDDI and FDDI rings complete End to End FCS checking is possible. It may also be possible to provide this type of service between IEEE802.5 and FDDI. In any event, the FCST (FCS Transparency) input is used to control this option.

6.0 SOURCE ROUTING BRIDGING IMPLICATIONS FOR END STATIONS

There are a few requirements placed on End Stations that participate in Source Routing protocols. The end station maintains the responsibility for discovering the route to its peers using the All route frame. Once the route has been discovered, it must be used in all future correspondences as part of the Routing Information Field.

An End station has no requirements for any external address matching logic. End stations can use the ability to transmit only the MSB of the SA (the Routing Indication Indicator) from the data stream using SAIGT. When using the BSI device, the BSI device SAT could be connected to the BMAC device SAIGT; similarly, SAT could be grounded and the BSI device STRIP could be connected to the BMAC device STRIP signal. The reason the rest of the SA can come from the Ring Engine is to insure proper stripping based only on the transmitted SA. This implies that in End Stations there is no need to use Void stripping.

Station Management Simplified

National Semiconductor
Application Note 726
My Le



INTRODUCTION

One of the key areas in the FDDI standardization process has been the work on Station Management (SMT). The SMT document provides the guidelines and protocols which can be used to manage an FDDI network.

To ensure interoperability in a multi-vendor environment, some of the protocols described in the SMT document are mandatory. On the other hand, to facilitate the diverse network environments envisioned for FDDI, many protocols described are optional. Thus the users need to determine the SMT protocols to be implemented based on their application and configuration requirements.

This application note provides an introduction to FDDI Station Management with the assumption that the reader is familiar with the MAC, PHY, and PMD portions of the FDDI protocol.

The following topics are included in this paper:

- Station Management Requirements
- Structure of FDDI Station Management
- Basic SMT framework to manage an FDDI network
- Optional management protocols based on configurations and applications
- SMT features provided by the National's DP83200 FDDI chip set

1.0 SMT Requirements

Before determining the SMT requirements for FDDI, let's define the major types of users of the network. Based on the requirements of the users, we can then determine the functions required by SMT.

1.1 TYPES OF USERS

Users can be divided into two main groups: End-Users and Network Administrators.

The End-Users are mainly interested in the services which the network provides; thus, SMT operations on the network should appear transparent to the End-Users.

The second major type of user in an FDDI network is the Network Administrators. While the End-Users would like to know as little as possible about the network, the Network Administrator's goal is to gather as much information about the network and its attachments as possible; thus SMT should be designed to allow the Network Administrators to control the network in a manner that is unobtrusive to the End-Users.

End-Users

The main requirements of the network by End-Users are:

• Network Services Reliability

One of the top requirements from the End-Users is the reliability of the network. The network should remain up and running with the probability of an error occurring as infrequently as possible.

When an error does occur, it should be isolated while the rest of the network that is free from the error should contin-

ue to operate. And finally, the error should be detected and removed from the network in a deterministic fashion as quickly as possible.

Thus, SMT needs to provide extensive and complete Fault Detection and Recovery procedures to satisfy the requirement of network's reliability.

• Access to All Authorized Networked Resources

SMT can be used to provide the mechanism for the End-Users to obtain the services available on the network. This service is useful, especially in a multi-vendor environment, to guarantee that all stations can receive the same types of network services regardless of their particular implementations.

An example of the services provided by the FDDI network is Synchronous Bandwidth Allocation. Using this service, stations can then obtain part of the bandwidth to transmit synchronous data.

• Plug-and-Play

Connection to a network should be made as simple as possible such that the End-Users can plug into the network without the need for complicated instructions or the possibility of bringing down the network by mistake.

This requirement is especially important in a large network such as FDDI where a large number of stations can potentially be connected, disconnected, or moved at any given time.

To satisfy this requirement, SMT needs to provide a comprehensive connection management procedure to allow stations to be connected quickly and correctly to the network.

Network Administrators

The main requirements of the network by Network Administrators are:

• Ability to Gather Information

One of the key functions of Network Administrators is to monitor the status of the network and the attached stations. To achieve this goal, the Network Administrators must have the capability of requesting and receiving information from stations on the network.

From the information gathered, the Network Administrators can then determine the status of the network and invoke any recovery mechanisms if necessary.

To meet this requirement, SMT needs to provide a monitoring procedure where the Network Administrator can gather information frequently and accurately.

• Standardized Management Services

In an open network environment where the Network Administrators have to control equipment from a large number of vendors, there is a need for standardized management services to allow the Network Administrators to communicate with any station on the network regardless of its implementation.

The standardized management services also allow the Network Administrators to interpret the information received from the stations.

• **Flexible Network Configuration**

Networks can be designed in many configurations depending on many factors such as applications used, building structure, etc. A network that can be configured in many different forms gives the Network Administrators the flexibility to design the network based on their own requirements and constraints.

FDDI Station Management provides the Connection Management procedure which allows the network to be connected into many different configurations (e.g., dual ring of trees, single tree, dual ring, etc.).

• **Ability to Manage the Network Remotely**

It is desirable for the Network Administrators to monitor activities on the network or trouble-shoot problems from a central location. It is also desirable to down-load information without physically being at the stations.

To provide these types of services, SMT needs the capabilities to control the remote stations and order them to perform certain operations.

1.2 SMT FUNCTIONS

Based on the types of users on an FDDI network and their applications as described above, a list of the SMT requirements can be drawn up as follows:

- Fault management for high network availability

- Reliable error detection and recovery management
- Access to networked resources
- Fast and reliable connection management procedure
- Management for multi-vendor networks
- Access to individual station information
- Flexible configuration

2.0 SMT Structure

SMT is the layer management service for FDDI networks which covers the Physical (PHY) and Media Access Control (MAC) Layers. SMT serves two main purposes in an FDDI network:

1. To collect information to report to the Management Agent Process which is responsible for the management of the entire station (above the PHY and MAC Layers), and
2. To manage stations on the network by starting and maintaining the PHY and MAC Layers.

SMT is divided into three main groups: Connection Management, Ring Management, and SMT Frame Services. The functions of these entities are described followed by a more detailed discussion on each.

Figure 1 shows the overall SMT Architectural model.

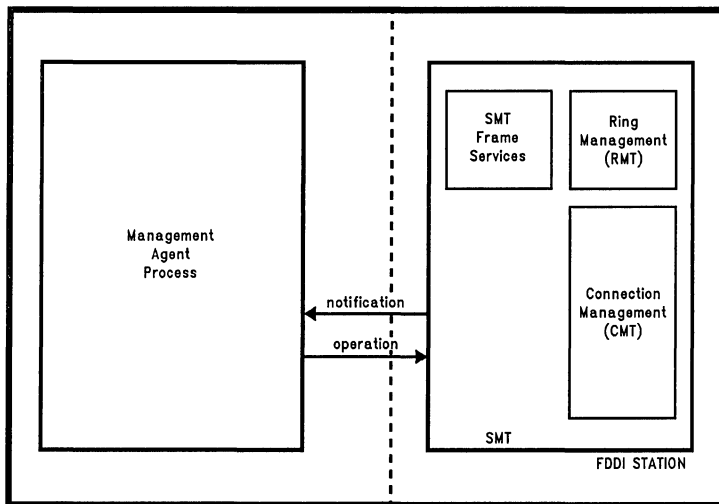


FIGURE 1. Station Management (SMT) Architectural Model

TL/F/11080-1

2.1 Connection Management

Connection Management (CMT) is the management entity in SMT that is responsible for the Ports (a Port is a PHY and PMD pair) and their interconnections to their neighboring Ports. It is also responsible for the configuration of MACs and PHYs within a station.

CMT's functions include the following:

- Establish and initialize physical connections
- Control station configuration
- Detect Physical Layer faults

The CMT entity is further divided into three sub-entities: Entity Coordination Management, Physical Connection Management, and Configuration Control Management.

Entity Coordination Management

The Entity Connection Management (ECM) indicates when the media is available (i.e. when signals can be transmitted and received). It is also used to coordinate activities of other entities within CMT and RMT.

There is one ECM entity per Station or Concentrator.

Physical Connection Management

The Physical Connection Management (PCM) initializes the connection of Ports and manages the Signaling Sequence between each physical connection.

The PCM uses the Line States available in the PHY to perform the Signaling Sequence.

There is one PCM entity for each Port.

Configuration Control Management

The Configuration Control Management (CCM) controls the interconnection of PHYs and MACs within a node to configure one of the following node types:

- Single Attach Station (SAS)
- Dual Attach Station (DAS)
- Single Attach Concentrator (SAC)
- Dual Attach Concentrator (DAC)

There is one CCM entity per Port.

2.2 Ring Management

Ring Management (RMT) is the entity in SMT that is responsible for the MACs within a station.

RMT's functions include the following:

- Notify station of MAC availability
- Detect logical MAC Layer faults

The RMT entity receives status information from the MAC and the Configuration Control Management (CCM) entity. The information is then reported to the higher-level management entity.

There is one RMT entity for each MAC in a Station or Concentrator.

Figure 2 shows the internal structure of the CMT and RMT entities.

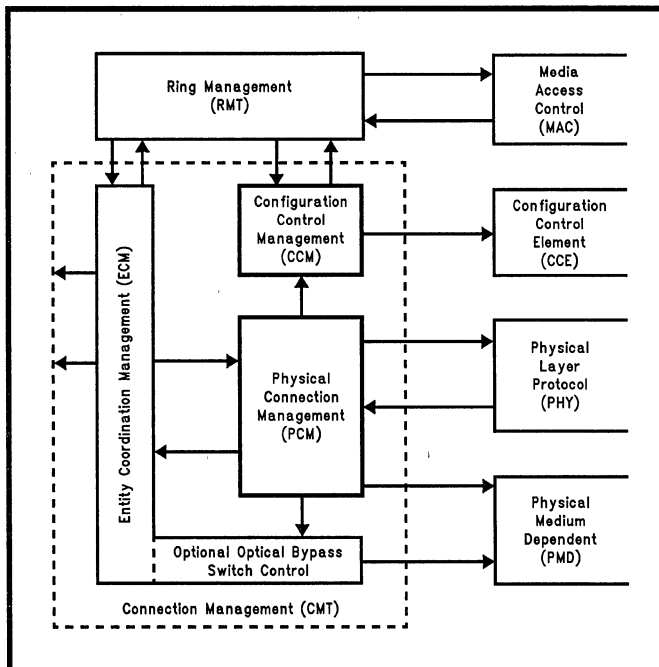


FIGURE 2. RMT and CMT Entities

TL/F/11080-2

2.3 Frame Services

Frame Services is the management entity in SMT that is responsible for providing a number of frames that may be used to gather information and control the stations attached to the network. These frames are used in SMT protocols which collect information for higher-level management entities.

Frame Services are used by the Management Agent Process to:

- Gather statistics
- Detect, isolate, and resolve network failures
- Tune performance
- Change topology

3.0 Basic SMT Framework

All entities within SMT, Connection Management, Ring Management, and Frame Services, are operated based upon the following inputs:

- Signals from higher level management
- Internal conditions triggered by the expiration of timers
- Signals received from other stations on the network

Since SMT is considered as an intelligent entity, the higher level management does not need to control every level of operation within SMT. Rather, SMT will perform the necessary procedures and protocols to accomplish a task requested when the higher layer management entities set the appropriate signals (Connect, Disconnect, Reset, etc.).

3.1 CONNECTION MANAGEMENT

CMT controls the Physical Layer (PHY and PMD) and the Configuration Control Element which connects the MACs and PHYs within a station or concentrator.

The operation of CMT is based upon the requests from SMT which in turn come from a higher level of management. The requests used to control CMT are:

- Connect Request: this request is used to signal the Physical Layer to connect to or disconnect from the network (signals Connect and Disconnect).
- Control Request: this request is used to signal the CMT to perform certain operations or to report status.

CMT communicates with SMT via Status Indication which is used to report CMT status changes.

Entity Coordination Management

The ECM is initialized when the CMT receives a Connect Control Request from SMT.

The services provided by the ECM Entity are:

- Connects the PMD to the network when CMT is initialized:
 - Allows the Transmitter and Receiver to begin to transmit and to receive.
 - Once the Fiber Optic Transmitter and Receiver are ready, a signal is set to initialize the PCM.
- Starts the Trace process to localize a Stuck Beacon condition based on the Trace__Prop signal from the Ring Management or Physical Connection Management entity.
 - After the ring is stuck in the Beacon state for a period of time (≥ 8.0 seconds), a signal is set to begin the Trace process.

— ECM performs the Trace function by invoking the PCM to transmit the appropriate Line States.

- Disconnects the PMD from the network
- A Path Test function is used to test all the components and paths within a node. Since the test occurs entirely within a node and cannot be verified, it is considered an implementation dependent issue and is not specified by the Standard.

The Path Test is used to ensure that the node will operate correctly once it joins the ring. It is also used to determine if the node causes errors on the ring.

For example a Path Test can include the following steps:

- Test all accessible data paths within the node
- Perform loopback testing of the PHY as close as possible to the PMD interface
- Confirm parameters provided to the MAC: addresses, timer values, etc.
- The MAC recovery process for this node including the resolution of the Beacon and Claim Processes

Physical Connection Management

PCM is initialized by the PC__Start signal from the ECM.

PCM provides the following services:

- Initializes a physical connection. The physical connection procedure is performed at the PHY Layer as follows:
 - Determines that a neighboring PHY exists
 - Determines that the neighboring PHY has the correct PC__Type to establish a legal connection between the two Ports. [In a typical network configuration, there are two types of legal connections: A to B (Dual Attach Stations on dual rings) and M to S (Single Attach Station connected to a Concentrator).]
- Runs the Link Confidence Test. The Link Confidence Test is used to determine if the link quality is adequate for ring operation. Its aim is to detect major link quality problems, not to determine the exact Link Error Rate.

The Link Confidence Test is performed in the PCM State Machine before the link is allowed to join the ring. A minimum Link Confidence test requires the transmission of Idle symbols for a period of 50.0 ms providing that the link has not had any recent link quality problem. Errors that occur during the testing period are recorded. If the number of errors recorded exceeds the acceptable error rate, the test fails. Otherwise, the link is considered to have passed the Link Confidence test.

The result of the Link Confidence Test is reported to higher level management. If the link fails the test, it will continue to be tested until the test is passed or until higher level management disconnects the link.

Once the Link Confidence test has been completed successfully, the link is ready to be included in the network. The PCM then signals the CCM to connect the appropriate MACs and PHYs together within a station.

The Link Error Monitor (LEM) is used to examine the link error rate of an active link. The LEM function complements the initial Link Confidence test to monitor the link quality once it has joined the ring.

LEM is performed by SMT using the facilities available in the PHY.

Link Error Events are counted to produce the LEM_CT count. A Threshold test is used to compare the current Link Error Rate with the cutoff and alarm Link Error Rate thresholds.

Once the LER reaches the alarm LER threshold, SMT reports the status to higher level management. If the LER is equal to or greater than the cutoff LER thresholds, the link is automatically removed from the ring and this event is reported to higher level management.

- Performs the Trace function.

Upon the reception of signal PC__Trace from ECM, PCM transmits the appropriate Line States required by the Trace function.

The Trace function provides a recovery mechanism for Stuck Beacon conditions on the FDDI ring. Whereas PCM is designed to recover from most physical faults that occur between two nodes, the Trace function is intended to provide recovery from a Stuck Beacon condition which cannot be localized to a single link.

The Trace function causes all stations and concentrators in the suspected fault domain to leave the ring and complete a Path test, so that the fault may be localized. The fault domain is defined as the area between the Beaconsing MAC and its nearest upstream neighbor MAC.

The Trace function is performed as follows:

- When a station enters the Beacon State, a timer is reset. If the station is still in the Beacon state when the timer expires, a Stuck Beaconsing condition has occurred and the RMT sets the Trace__Prop signal to the ECM to initialize the Trace function.

- The Trace function starts at the node with the Beaconsing MAC and traverses to the nearest upstream MAC.

- The ECM controls the configuration information and sets the PC__Trace signal to the appropriate PCM. When a PCM receives a PC__Trace signal, it transitions to the Trace state to transmit a special line state that indicates Trace.

- The Port at the other end of the link receives the "Trace" Line State and will set the Trace__Prop flag to indicate that the Trace function is to be propagated upstream.

- When the Trace Line State arrives at a Port that is connected to the input of a MAC, the Trace has been completed.

The node with the MAC that receives the Trace removes itself from the ring from Path Test.

- The removal of this node causes the node downstream from it to remove itself also.

Thus, all nodes in the Trace domain will eventually remove themselves from the ring to perform Path Tests. This process should take less than the Trace__Max timer value (7 seconds).

If the Trace function has not been completed within the Trace__Max time, the process has failed and manual intervention is required.

Once the Trace function has been completed, it will indicate the result to ECM.

- Supports Maintenance.

In the Maintenance state, the PCM can transmit any sequence of symbols. This feature is useful to ensure that the PHY Transmitter can transmit all the symbols in the FDDI Code. It is also used to force the other end of the connection into a particular state manually without going through the Connection Sequences.

Configuration Control Management

The CCM is initialized by the CF__Join signal from the PCM.

The services provided by the CCM Entity are:

- Inserts Single Attachment Station or Concentrator to the Primary Path.

- Connects the MAC of Dual Attachment Station or Concentrator with a Single MAC to the Primary Path.

In this configuration, all Dual Attach Stations with one MAC are connected to the Primary Ring. Only Dual Attach Stations with two MACs can transmit and receive frames on both the Primary and Secondary Rings.

Single Attach Stations are connected to the Primary ring as the default configuration.

- Performs the Scrub function.

The Scrub function is used to remove PDUs sourced by MACs that no longer form part of the same token path. These MACs may have been removed from the token path internally within its node or due to a network topology change. It is controlled by the CEM entity.

The Scrub function removes left over PDUs after a reconfiguration to ensure that all PDUs on the ring have been created since the last reconfiguration.

The Scrub function may be performed by using one of several mechanisms listed below.

- Transmit Beacon or Claim frames for a sufficient time while the input to the MAC is blocked (stripping old frames while transmitting Beacon or Claim frames)

- Transmit Idle symbols for a sufficient time while discarding input stream received at the PHY. This method may be used for a node that does not have a MAC after reconfiguration.

- Frames can also be stripped by the node that is performing the Scrub function.

3.2 RING MANAGEMENT

RMT manages the basic information and condition of each MAC. The operation of RMT is based upon the control request from SMT, which in turn comes from the higher level of management. This request is used to signal RMT to reset, to change the basic information in the MAC, or to report its status. RMT is also responsible for initiating fault recovery actions to recover the ring.

RMT communicates with SMT via the Status Indication which is used to report its status changes.

Services provided by RMT are:

- **Identification of a Stuck Beacon condition**
If the ring remains in the Beaconing state for a long time (≥ 8.0 seconds), the Stuck Beacon condition has occurred. RMT will report this error condition to higher level management as well as starting a new error recovery mechanism.
RMT uses a timer (T_{Stuck}) to keep track of the Struck Beacon condition.
- **Initiation of the Trace function**
Once the ring has been identified to be in the Stuck Beacon condition, RMT starts the Trace function by setting the Trace_Prop signal to ECM.
- **Notification of MAC availability**
After the CCM sets the RM_Join signal to indicate that the MAC is connected to the appropriate PHY in a station (or concentrator), the RMT can then set the MAC_Available signal to higher level management to indicate that the MAC is ready to transmit and receive data.
- **Detection of Duplicate Addresses**
By observing the order in which Beacon and Claim frames are received at the MAC, RMT can detect Duplicate Addresses which can prevent the ring from becoming operational.
Upon detecting this condition, RMT will notify higher level management of the condition. It will also take actions to resolve the Duplicate Address problem.
- **Resolution of Duplicate Address Problem**
One of three possible solutions can be taken by RMT to eliminate the Duplicate Address problem:
 1. Change the MAC's address to a unique universal address
 2. Change the bidding time to guarantee that this station will lose the Claim Process
 3. Remove the station from the ring

3.3 FRAME SERVICES

A number of frames are specified as SMT frames. These frames are used to gather information and control the operation of the stations on the network. There are four types of mandatory SMT frames:

- Neighbor Information Frame
- Resource Allocation Frame
- Request Denied Frame
- Status Report Frame

Neighbor Information Frame

A Neighbor Information Frame (NIF) is used by a station for periodic announcement of its basic operating information.

Based on the Upstream Neighbor Address provided in NIF frames, a station can then build a ring map of the stations' locations and their connections to other stations.

There are three types of NIFs: Announcement, Request, and Response.

• Announcement

A NIF Announcement frame is broadcast to the entire ring. A station can choose to transmit a NIF Announcement or NIF Request. If a NIF Announcement is to be transmitted, it will be sent every 30 seconds when the ring is operational and under zero load conditions.

• Request

A NIF Request is sent to a station, a group of stations, or the entire ring. The NIF Request announces the station's information while requesting that the corresponding station(s) respond with a NIF Response.

If an NIF Request is to be transmitted, it will be sent every 30 seconds when the ring is operational, under zero load conditions.

• Response

A NIF Response is sent in response to a NIF Request.

Upon receiving a NIF Request, a station is required to send a NIF Response within 30 seconds if the ring is operational, under zero load conditions.

In addition to the Upstream Neighbor Address, the NIF Response frame also provides the Downstream Neighbor Address, and the mechanism to detect Duplicate Addresses.

Resource Allocation Frame

A Resource Allocation Frame (RAF) is defined to support a variety of network policies for allocation of resources. At this point, only the Synchronous Bandwidth is identified as the only resource supported by the Resource Allocation Frames. However, the protocol can also be used to support other types of resource allocation which have yet to be specified in the Standard.

Request Denied Frame

A Request Denied Frame (RDF) is used to respond to optional frames that the station does not support. It is also used to respond to an SMT frame with a Version ID that this station does not support.

Status Report Frame

The Status Report Frame (SRF) is used to periodically announce the station's status which may be of interest to the Network Administrator.

Two types of information are included in the SRF: Conditions and Events. Conditions include the station state which may be of interest to a network manager as long as the condition remains asserted. Events are instantaneous occurrences which are of interest to a network manager.

4.0 Optional Protocols

Aside from the mandatory functions listed in Section 2.0, FDDI SMT also provides many optional protocols that can be implemented in addition to the mandatory ones.

4.1 CONNECTION MANAGEMENT

Entity Coordination Management

ECM has two optional features, the Optical Bypass Switch Control and Hold Policy.

• Optical Bypass Option

The Optical Bypass is used to allow a Dual Attachment Station or Concentrator to be inserted and deinserted from a dual ring without disrupting the operation of the ring.

If the Optical Bypass Option is available, the ECM allows for the switching time of the optical bypass switch during the Insertion process. It also allows time for the optical bypass switch to deinsert during the Deinsertion Process.

• Hold Policy Option

When the Hold Policy is invoked, it prevents the dual rings from wrapping when a fault occurs on one of the two rings. The Hold Policy may be used in Dual Attachment Stations and Concentrators.

The Hold Policy is useful in preventing the disruption of a ring when an error occurs on the other ring of the dual rings (disruption occurs when the ring attempts to wrap).

Physical Connection Management

PCM has the following two optional Features

• Physical Connection

In a normal dual ring of trees structure, there are two types of physical connections between two ports: A-B (A port to B port) and M-S (Master port to Slave port). In addition to these two connections, other connections can also be acceptable as legal:

- A port to A port (A-A)
- B port to B port (B-B)
- A port to Master port (A-M)
- B port to Master port (B-M)
- Slave port to Slave port (S-S)

The A-A and B-B connections may be used when two Dual Attachment Stations are connected together to form a ringlet (a dual ring with two stations).

The A-M and B-M connections may be used when a Dual Attachment Station is used as two Single Attachment Station. In this case, the station can only be connected to the ring via a Concentrator. This scenario is called Dual-Homing.

The S-S connection may be used to connect two Single Attachment Stations together to form a link. The two stations thus form a single ring.

Although these connections are considered legal, higher level management needs to be notified so that the link can then be rejected.

• Link Confidence Test

Aside from the minimum Link Confidence Test described in Section 3.4, other types of Link Confidence tests can be performed.

The two PHYs of the link need to agree beforehand which type of Link Confidence test is to be carried out. This information is exchanged via a bit in the PCM Signaling Sequence.

Other Link Confidence tests to be considered include the following:

- Transmitting PDUs and counting link errors. Errors are detected and counted at the PHY.
This Link Confidence test requires at least one MAC connected to one of the two PHYs.
- Transmitting PDUs and counting Frame Check Sequence errors. Errors are detected and counted as frame errors at the MAC.
This Link Confidence test requires at least one MAC connected to one of the two PHYs.
- Looping back symbols received from the other end of a connection and counting link errors on reception. Errors are detected and counted at the MAC.

This Link Confidence test is performed at the PHY layer.

The length of the Link Confidence Test can be:

- Short (50.0 ms)
- Medium (500.0 ms)
- Long (5.0 sec.)
- Extended (no maximum time specified)

The length of the Link Confidence test is indicated by two bits of the PCM Signaling Sequence.

Configuration Control Management

Aside from the Primary Path, there are two other optional paths available in CCM: Secondary and Local.

- A PHY or a MAC can be connected to the Local Path. While connected to the Local Path, these entities are removed from the ring and can be used to perform local testing.
- A Single Attachment Station can initially be connected to the Secondary Path. Single Attach Stations can then choose to transmit and receive frames on either the Primary or Secondary ring depending upon the initial connection.
- The MAC of a Dual Attachment Station with a Single MAC can also optionally be connected to the Secondary Path. Stations with one MAC can then choose to transmit and receive frames on either the Primary or Secondary ring.

4.2 FRAME SERVICES

The following SMT frames are provided by the Frame Services to gather status and control the nodes on the ring:

- Station Information Frames
- Echo Frames
- Extended Service Frames
- Status Report Frames
- Parameter Management Frames

Station Information Frame (SIF)

Station Information Frames (SIFs) are used to request and provide, in response, a station's configuration and operating information. There are two classes of SIFs: Configuration and Operation.

• Configuration SIF

A station can request a station, a group of stations, or all stations on the ring to respond with its (their) configuration information using the SIF Configuration Request. The transmission of these Request frames is optional.

A station is required to respond to SIF Configuration Request frames with a SIF Configuration Response frame within 30 seconds of receiving a Request frame, under zero load conditions. Stations can also deny the request by sending back a Request Denied Frame.

The SIF Configuration Response provides the configuration structure of the node by describing the connections of the PHYs and MACs within the node. It is used to build the full ring map (both logical and physical).

• Operation SIF

A station can request a station, a group of stations, or all stations on the ring to respond with its (their) operation information using the SIF Operation Request. The transmission of these Request frames is optional.

A station is required to respond to SIF Operation Request frames with a SIF Operation Response frame within 30 seconds of receiving a Request frame, under zero load conditions.

The SIF Operation Response provides the operating parameters in a node; information such as timer values, counter values, etc. It is used to detect faults by monitoring the station's status and counter values.

Echo Frames

A station can request another station on the ring to re-transmit a test pattern using the Echo Request Frame (ECF). This test pattern is stored in the Information field of the Echo Request Frame.

Upon receiving the Echo Request Frame, the recipient builds an Echo Response Frame and sends it to the Request Frame's Source Address.

The Response Frame is required to be transmitted within 30 seconds after receiving the Request frame if the ring is operational and there is zero load. The recipient can also send a Request Denied Frame instead of the Response Frame.

Echo Frames can be used to test for data-sensitive network failures by placing the suspect data pattern in the Echo Information field. It can also confirm that a station's Port, MAC, and SMT are partially operational.

Extended Service Frame

The Extended Service Frame (ESF) can be used to test new SMT services that are intended for inclusion in later versions of the FDDI SMT document.

The structure of the ESF is defined by the owner of the Extended__Type.

Parameter Management Frames

The Parameter Management Frames (PMF) are used to get, change, add or remove parameters in a node. There are 4 classes of PMFs: PMF Get, PMF Change, PMF Add, and PMF Remove.

There are two types of frames for each class: Request and Response.

PMFs are transmitted with an optional authorization code to provide a type of security check.

PMF Get

A station can issue a PMF Get Request Frame to query the value of one or a group of attributes in the Management Information Base (MIB) of an individual, group, or all stations.

The receiving station can respond with the current value of the requested attributes. If the protocol is not supported, a station can transmit a Request Denied Frame in return.

PMF Change

A station can issue a PMF Change Request Frame to change the value of a single attribute in the Management Information Base of an individual, group, or all stations.

The receiving station can act to change the requested attribute. It can then respond with a PMF Change Response. A station could also transmit a Request Denied Frame in return.

PMF Add

A station can issue a PMF Add Request Frame to add a value of a single attribute in the Management Information Base of an individual, group, or all stations.

The receiving station can act to add the requested value. It can then respond with a PMF Add Response. A station could also transmit a Request Denied Frame in return.

PMF Remove

A station can issue a PMF Remove Request Frame to remove the value of a single attribute in the Management Information Base of an individual, a group, or all stations.

The receiving station can act to remove the requested value. It can then respond with a PMF Remove Response. The station could transmit a Request Denied Frame in return instead.

5.0 National's FDDI Chip Set

National's DP83200 FDDI Chip Set has been designed to provide maximum support to the Station Management functions. Both the PLAYER and BMAC devices have separate management interfaces via the Control Bus. Furthermore, each chip has many registers on-board to provide the information required by the different SMT entities.

5.1 PLAYER DEVICE

Connection Management

The Connection Management Entities (ECM, PCM, and CCM) can control the operation of the PMD and PHY using the registers available on the PLAYER Device.

• Entity Coordination Management

The user can control the operation of the PMD by setting or resetting bits 4 to 7 of the MODE Register.

• Physical Connection Management

The PCM Signaling Sequence can be implemented using the Current Transmit State Register and Current Receive State Register in the PLAYER device.

Line States can be transmitted by setting the appropriate bits in the Current Transmit State Register. Line States received can be monitored observing the Current Receive State Register.

Furthermore, a historical record of the Line States received is kept in the Receive Condition Registers. This information is useful for keeping track of the Signaling Sequence.

The Noise Threshold and Noise Prescale Threshold Registers are used to ensure that the noise conditions do not persist beyond the maximum tolerated level.

• Configuration Control Management

Using the Configuration Register, the CCM can control the connection of the PHYs and MACs in a node. Each PLAYER Device can be connected to the Primary Path, Secondary Path, and Local Path. In addition, it can also be connected to the PHY Invalid Bus where the PLAYER Device can continuously transmit PHY Invalid to the ring or indicate PHY Invalid to the entity it is connected to internally within the node (i.e., a BMAC Device or another PLAYER Device). The PLAYER Device can be configured via the Configuration Register without external logic.

Link Error Monitor

SMT can use the following registers to perform the Link Error Monitor functions once the PLAYER Device is connected to the ring:

- Current Noise Count Register
- Current Noise Prescale Count Register
- Link Error Threshold Register

These registers enable the user to implement different methods of monitoring link errors according to their requirements.

Loopback

The PLAYER Device can be programmed to perform Internal or External Loopback. These Loopback operations are useful during Path Testing.

The Internal Loopback mode can be used to test the functionality of the PLAYER Device or to test the data path between the PLAYER and BMAC Devices.

The External Loopback mode can be used to test the functionality of the PLAYER Device and to test the data paths between the PLAYER Device, Clock Recovery Device, and BMAC Device. This mode is especially useful when the Path Test requires testing as close to the PMD as possible.

5.2 BMAC Device

The BMAC Device provides extensive ring and station statistics via the on-board Timer and Counter Registers. Furthermore, it can internally generate Claim and Beacon frames that are used in the FDDI MAC Protocol to detect errors.

Timers and Counters

Information can be provided to RMT and other SMT entities to represent the operating status of the node using the following Counters and Timers:

- Late Count Counter
- Frame Received Counter
- Error Isolated Counter
- Lost Frame Counter
- Frame Copied Counter
- Frame Not Copied Counter
- Frame Transmitted Counter
- Token Received Counter
- Ring Latency Counter
- Negotiated Target Rotation Timer
- Maximum Token Rotation Timer
- Valid Transmission Timer
- Asynchronous Priority Threshold

The information provided can then be transmitted to other stations on the ring in the Station Information Operation Response Frame and Status Report Frame.

Loopback

The BMAC Device can be programmed to perform loopback testing.

There are three Self-Test Paths:

- Internal to the BMAC Device
- Through the PLAYER Device(s)
- Through the CRD Device

These paths allow the user to perform Path Tests on the BMAC, PLAYER, and CRD Devices.

Stripping Protocol

A special stripping protocol can be invoked by asserting the STRIP signal (pin 13). The stripping protocol starts with the transmission of two My_Void frames at the end of a current service opportunity. The stripping will continue until a My_Void frame returns. The stripping protocol can be invoked when the initial token is issued after a successful Claim to remove all fragments and ownerless frames from the ring as required by the Scrub function of the Configuration Control Management entity.

Inhibit Recovery

By setting bit 3 of the Option Register, the MAC can be prevented from entering the Claim state.

This option is useful in allowing the ring to recover from the Duplicate Address scenario where two stations with the same address also have the winning Claim frames. By prohibiting one station to enter the Claim state, the other station can then win the Claim process thus allowing the ring to become operational.

Claim and Beacon Frames

The BMAC Device reports the reception of a Claim or Beacon frame by setting the appropriate bit in the Ring Event Latch Registers.

By keeping track of the received Claim and Beacon frames, the user can then determine if the Error Recovery process (Claim or Beacon) has succeeded or failed.

Duplicate Address Detection

Upon the detection of a Duplicate Address, the BMAC Device reports the incident by setting the appropriate bit in the Ring Event Latch Registers.

Duplicate Token Detection

Upon the detection of a Duplicate Token, the BMAC Device reports the incident by setting the appropriate bit in the Token and Timer Event Latch Register.

6.0 Summary

The Station Management (SMT) facilities, an essential part of an FDDI network, provide a rich set of tools to manage an FDDI network. As a summary, the Connection Management services of SMT manage the configuration of the station and the link between the station's Ports and their neighboring Ports; the Ring Management facilities provide control of

the MACs of a station in the FDDI rings; the Frame Services provide a tool to manage the complete FDDI network, the services are the most flexible and extensive part of SMT.

The implementation of Station Management software can be rather complicated without adequate support from the hardware. As a result, the National Semiconductor Corporation DP83200 FDDI Chip Set integrated many essential functions on the chip set and provides maximum support to FDDI Station Management functions. The PLAYER Device and the BMAC Device support the Connection Management services and the Ring Management service respectively. The BSI Device provides separate Station Management channel and data frame channels for the maximum support of the SMT Frame Services. The invaluable Station Management features in the DP83200 Chip Set can shorten the Station Management software development cycle and provide higher reliability of the FDDI network.

A Guide to the Implementation of Physical Connection Management

National Semiconductor
Application Note 727
My Le, Michael Yip



1.0 Introduction

The FDDI Station Management (SMT) standard provides the necessary control of an FDDI station (node) so that the node may work cooperatively as a part of an FDDI network. To effectively implement the functions required, SMT is divided into three entities, namely the Connection Management entity (CMT), the Ring Management entity (RMT) and also the Frame Based Services. The Connection Management is further divided into three entities, the Physical Connection Management (PCM), Configuration Element Management (CEM), and Entity Coordination Management (ECM).

The Physical Connection Management is an entity within Connection Management whose functions include:

- Initialization of the connection of neighboring ports pair where each port is comprised of one PMD entity and one PHY entity.
- Enforcement of port connection policies and withholding of unacceptable connections
- Testing of link confidence and monitoring of link quality between neighboring ports
- Detection of physical connection level faults between two ports and invocation of path test
- Support of maintenance line state
- Participate in the Trace action

For a general description of SMT, please consult the application note entitled "SMT Simplified", AN-726.

In this document, the operation of the Physical Connection Management entity is described along with a guide to the implementation of the PCM using the PLAYER™ device. In addition, one implementation of the Link Error Monitor using the PLAYER device is also discussed. For a more detailed description of PCM and other SMT entities, please consult the ANSI FDDI Station Management Standard.

2.0 PCM Functions Overview

Many instances of PCM can exist on a FDDI node and each instance of PCM controls one port in a node. For example, two separate instances of PCM exist in a Dual Attached Station (DAS) and only one instance of PCM exists in a Single Attached Station (SAS). Each PCM communicates with the ECM and CEM entities and directly controls the PHY layer device of a port.

One of the most important functions of PCM is to establish a connection between two ports. The connection process is achieved through a lock-step handshaking procedure. The handshaking procedure controlled by PCM is divided into three stages:

- Initialization sequence
- Signaling sequence
- Join sequence

The initialization sequence is used to indicate the beginning of the PCM handshaking process. It forces the neighboring PCM into a known state so that the two PCM state machines can run in a lock-step fashion.

Following the initialization sequence is the signaling sequence. The signaling sequence communicates information about the port and the node with the neighboring port. A Link Confidence Test (LCT) is also conducted during the signaling sequence to test the link quality between the two ports. If the link quality is not acceptable or the type of connection is not supported by the nodes then the connection will be withheld.

If the connection is not withheld during the signaling sequence, the PCM state machine can move onto the join sequence and establish the connection between the two neighboring ports.

In addition to establishing the connection, PCM also supports the Maintenance function and performs the Trace function. During the Maintenance function, PCM forces the PHY layer device to transmit a specified line state continuously. The Maintenance function allows SMT to force the PCM of the neighboring port into a known state manually and line faults may be traced when both end nodes are in known states and do not change state due to line noise.

The operation of PCM can be implemented by a state machine. The state machine is comprised of ten states: Off, Break, Trace, Connect, Next, Signal, Join, Verify, Active and Maintenance. The next section of this document describes the PCM State Machine.

3.0 Detailed PCM Description

3.1 CONNECTION SEQUENCE

The connection sequence starts with the reception of the PC_Start signal from the Entity Coordination Management. PC_Start indicates that the physical media is available for communication. The PC_Start signal causes the PCM state machine to enter the Break state which is the beginning of the initialization sequence.

In each sequence the two PCM state machines exchange a sequence of line states. The PCM state machine sends the line state to its neighboring PCM by directing the PHY layer devices to transmit a continuous stream of line state symbols. This line state is transmitted for a duration of time to ensure that the neighboring PCM receives the information. While transmitting, the PCM state machine also expects to receive a particular line state from the neighboring PCM. If things go as planned, the connection progresses through the three sequences and PCM signals the Configuration Element Management entity to include the connection into the token path.

The remainder of this section describes the detail operation of each sequence in the connection process.

Initialization Sequence

The Break state and the Connect state are used in the initialization sequence to start the connection.

The Break state is the entry point in the start of a PCM connection. In the Break state, a continuous stream of Quiet Symbols is transmitted to force the other end of the connection to break any existing connection and restart the connection sequence. The Break state is entered upon the reception of the PC__Start signal from ECM or other PCM states when external events force a reinitialization of the connection. Reinitialization is required if the Link Confidence Test fails, the expected line state is not received, a noise condition occurs for a period of time, or the neighboring port is forced to break the connection.

When Quiet line state or Halt line state is entered during Break state, the connection has been initialized successfully in the Break state and the PCM state machine can transition to the Connect state. The Connect state is used to synchronize the two ends of the connection to begin the signaling sequence. In the Connect state, a continuous stream of Halt symbols is transmitted for a sufficient amount of time for clock acquisition by the receiving PHY.

Signaling Sequence

The second stage of the connection sequence is the signaling sequence. The Next state and the Signal state is used in the signaling sequence to exchange port information with the neighboring PCM. Link Confidence Test and/or MAC Loop Back Test are also performed during the signaling sequence.

The PHY line states are used to signal bit information and provide handshaking during the signaling. During the signaling sequence, three line states are used. The Idle line state is used as a bit delimiter, the Halt line state is used to represent a logical one (set) and the Master line state is used to represent a logical zero (clear). If Quiet line state is entered during any part of the signaling sequence, the PCM state machine should make a transition to the Break state and restart the connection sequence again. During the signaling sequence, ten bits of information are exchanged with the neighboring PCM. Section 5.2 has a list of the format of the 10 bits of information.

The Initialization sequence leaves the PCM state machine in the Connect state while the PHY Port is transmitting Halt symbols. The reception of Halt line state in the Connect state causes the PCM state machine to transition to the Next state. The signaling sequence starts upon the first transition to the Next state. In the Next state, a continuous stream of Idle symbols is transmitted. The Next state is used to separate the bit signaling performed in the Signal state.

The reception of Idle line state in the Next state causes the PCM state machine to transition to the Signal state. In the Signal state, a continuous stream of Halt symbols or Halt-Quiet (Master) symbol pairs are transmitted. Therefore one bit of information is transferred each time when the Signal state is entered. The reception of Halt line state or Master line state in the Signal state will cause the PCM state machine to transition to the Next state again. The Next state and Signal state cycle repeats ten times to exchange all ten bits of information in the signaling sequence.

Join Sequence

The final stage of the connection sequence is the join sequence. The join sequence is comprised of three states running in sequence. The three states are Join state, Verify state and finally the Active stage. The join sequence is a unique sequence of transmitted symbol streams received as line states (HLS-MLS-ILS) that leads to an active connection. At the end of the join sequence, the PCM state machine signals the CEM entity and CEM will incorporate the connection into the token path.

The PCM state machine enters the Join state when the signaling sequence finishes exchanging the ten bits of information. After the tenth bit is transmitted, the PCM state machine enters the Next state. The reception of Idle line state in the Next state causes the PCM state machine to transition to the Join state. In the Join state, a continuous stream of Halt symbols is transmitted.

The reception of Halt line state in the Join state causes the PCM state machine to transition to the Verify state. In the Verify state, a continuous stream of Halt-Quiet (Master) symbol pairs is transmitted.

The last state of the connection sequence is the Active state. The reception of Master line state in the Verify state causes the PCM state machine to transition to the Active state. In the Active state, a continuous stream of Idle symbols is transmitted. Upon the reception of Idle line state in the Active state, the PCM state machine directs the PHY device into the Active Transmit Mode, which transmits the PDUs from the PHY device's request port.

Many timers are used to ensure the connection sequence proceeds in lock-step fashion on the two ports of the link. The timers used in PCM are listed in Section 5.3 together with a brief explanation of each timer. The state diagram of the PCM state machine and a list of PCM states are also presented in Section 5.1.

3.2 MAINTENANCE FUNCTION

The Maintenance state is used to perform the Maintenance Function. In the Maintenance state, the symbol stream specified by higher level management agent shall be forced. During the Maintenance state, the PCM state machine is insensitive to the received line state.

The Maintenance state is useful to ensure that the PHY and PMD devices can transmit all the symbols specified in the FDDI standard. The Maintenance state is also used to force the other end of the connection to a particular state manually (without going through the Connection Sequence).

3.3 TRACE SUPPORT

Trace support is needed to localize a Stuck Beacon condition. The Trace function is used to recover from the Stuck Beacon condition by propagating the Trace signal along the ring, causing all the stations and concentrators in the suspected fault domain to leave the ring and perform a Path Test. In the Trace state, a continuous stream of Halt-Quiet (Master) symbol pairs is transmitted to the upstream station.

Two possible sequences of events can cause the PCM state machine to enter the Trace state. In the first case, the reception of Master (or Trace) line state in the Active state causes the PCM state machine to send the Trace__Prop signal to ECM. ECM will direct the appropriate PCM state machine to transition to the Trace state by sending the PC__Trace signal to that PCM. Upon the reception of the PC__Trace signal, a continuous stream of Halt-Quiet (Master) symbol pairs is transmitted.

The second situation that causes PCM state machine to enter the Trace state is a bit more involved. When the Ring Management entity detects a Stuck Beacon condition, RMT sends the Trace_Prop signal to the ECM. Like the first case, the ECM entity will direct the appropriated PCM state machine to transition to the Trace state.

3.4 Link Confidence Test

The Link Confidence Test (LCT) tests the link quality before the link is inserted into the token path. If the link quality is not adequate for ring operation then LCT will fail. Failure of LCT will cause PCM to return to the Break state and the connection is reinitialized. As a result, the LCT and the whole connection sequence will be repeated until the LCT is passed. The LCT is intended to detect major link problems and not to determine the exact Link Error Rate.

The Link Confidence Test is performed after bit 6 is transmitted in the Signaling sequence. During the duration of the test, either Idle symbols or valid PDUs are transmitted. The reception of Master line state signifies the successful completion of the LCT, while the reception of Halt line state signifies the failure of the LCT. If Quiet line state is entered during the LCT, the test is aborted and PCM returns to the Break state.

The duration of the LCT and the type of LCT is determined by the bit signaling process in the signaling sequence. LCT is defined to have four durations, namely LC_Short (50 ms), LC_Medium (500 ms), LC_Long (5 sec) and LC_Extended (50 sec). When a port is started the first time, it requests the shortest test duration. Every time the LCT fails, the duration is increased up to a maximum duration of LC_Extended. If the two ports on the link request different LCT duration in the Signaling sequence, the longer LCT duration is used to ensure a better measurement of link quality.

The LCT can be performed using one of the following methods:

- Transmit Idle symbols and count link errors. The error measurement is taken from the PHY device.
- Transmit valid PDUs and count link errors. Again, the error measurement is taken from the PHY layer device.
- Transmit valid PDUs and count Frame Check Sequence (FCS) errors from the MAC frames. In this method, the MAC layer device is used to source the PDUs and also take error measurements.
- The PHY will retransmit what is received and the transmitter will check for link errors. This method requires at least one MAC device connected to the link to send PDUs. The error measurement is taken from the PHY layer device.

Because multiple levels of the LCT can be performed, a minimum capability of transmitting Idle symbols and counting link errors is required for each port.

3.5 LINK ERROR MONITOR

The Link Error Monitor (LEM) continuously examines the link error rate (LER) of an Active connection. Its function complements the LCT to ensure that the link quality is adequate for ring operation at all times. When the LER exceeds the cutoff value, the connection is flagged as faulty and shall be removed from the token path.

The LEM and LER are based on the link error count from the PHY level device. However, the implementation of LEM and LER is not specified in the FDDI SMT Standard. One way of calculating LER is to keep a running time average of link error events occurred in a given time period. See Table I.

TABLE I. Link Error Rate (LER) Calculation Example

Time (in ms)	LEC in	Starting Time	Ending Time	Total Time (in ms)	Total LEC	LER (in Bit/Sec)
T+100	0	T	T+100	100	0	$<8.00 \times 10^{-8}$
T+200	1	T+100	T+200	200	1	4.00×10^{-8}
T+300	0	T+100	T+300	300	1	2.67×10^{-8}
T+400	0	T+200	T+400	300	1	2.67×10^{-8}
T+500	0	T+300	T+500	300	0	$<2.67 \times 10^{-8}$
T+600	2	T+400	T+600	300	2	5.33×10^{-8}
T+700	1	T+500	T+700	300	3	8.00×10^{-8}
T+800	0	T+600	T+800	300	3	8.00×10^{-8}
T+900	0	T+700	T+900	300	1	2.67×10^{-8}

Notes: The Time field is in ms and the measurement is started from certain time and ended 100 ms later.

The Link Error Count (LEC) field measures the number of Link Error occurring in the 100 ms interval.

The Starting and Ending Time is the starting and ending period for the LER calculation.

LER is calculated by the formula:

$$\text{LER} = \frac{\text{Total LEC}}{\text{Total Time (seconds)} \times 125 \times 10^6}$$

Note that LER cannot be zero even if Total LEC is zero and the LEM can only indicate the LER is less than when Total LEC is one.

4.0 PLAYER Device

The PLAYER device was designed with SMT in mind, and therefore the PLAYER device offers a larger number of registers to be used by the PCM software. These registers can be divided into four groups based on their functions. The groups are general control and system interface registers, line state reception and symbol generation registers and event counter registers.

The general control and system interface registers provide control to different types of optical transmitters and receivers, control of different types of transmission and reception and also control of interrupt generation for the system. The line state reception and symbol generation registers allow the transmission of different line states and the reporting of line state reception (with or without interrupt generation). The event counter registers consist of three sets of registers, one set keeps track of noise events, one set for link error monitoring and another set for timing of the incoming line state.

The following sections discuss the PLAYER registers in detail and also some ideas on how to implement PCM using the PLAYER device. For a complete list of PLAYER registers, please refer to the DP83251/55 PLAYER device data-sheet.

4.1 PLAYER REGISTERS

The PLAYER device includes many registers which ease the implementation of PCM. This section provides an explanation of the registers which are used to implement PCM and how to use those registers. Consult the PLAYER data-sheet for the address of those registers. Note that not all the functions in each registers are discussed in this section.

MODE Register

The Mode Register (MR) controls the mode of operation of the PLAYER device. MR enables and disables the PLAYER device, which is a very important function! It also sets the transmission level of the Quiet symbols to either low level or high level so that the PLAYER device can interface with different types of optical transmitters without extra external logic.

Interrupt Condition Register

The Interrupt Condition Register (ICR) records the occurrence of an interrupt condition. It is set by the hardware and can be clear by the software. Each bit in the ICR represents an interrupt condition, the interrupt conditions are:

- PHY Request Interface Parity Error. This error occurs when there is a parity error at the PHY request interface and the parity checking is enabled by setting the PHY Request Data Parity Enable bit in the Current Transmit State Register.
- Control Bus Data Parity Error. This error occurs when there is a parity error in the incoming data from the Control Bus and the parity checking is enabled by using the Control Bus Parity Enable pin.

- Control Bus Write Command Reject. This error occurs when the software is trying to write into one of the read-only registers in the PLAYER device.
- Conditional Write Inhibit Condition. This bit is set to one, when the software tries to write to a conditional register when that register is updated internally. This feature protects the system from accidentally erasing an exception condition.
- Link Error Monitor Threshold Reached. This bit is set to one when the internal 8-bit Link Error Counter reaches zero. It is used to implement the alarm limit and the cutoff limit of the LEM.
- Receive Condition A. This bit is set to one by the PLAYER when the Receive Condition A occurs. See also the Receive Condition Register A and Receive Condition Mask Register A.
- Receive Condition B. This bit is set to one by the PLAYER when the Receive Condition B occurs. See also the Receive Condition Register B and Receive Condition Mask Register B.
- User Definable Interrupt. The bit is set when one or both of the Sense Bits in the User Definable Register is set to one. In order to clear this bit, both the Sense Bits must be set to zero.

An interrupt is not generated unless the bit(s) in the ICR is set and the corresponding bit(s) in the Interrupt Condition Mask Register (ICMR) is also set.

Interrupt Condition Mask Register

The Interrupt Condition Mask Register (ICMR) is used to determine which event in the Interrupt Condition Register can generate an interrupt.

Interrupt Condition Comparison Register

The Interrupt Condition Comparison Register (ICCR) ensures that any changes in the Interrupt Condition Register (ICR) is recorded while it is being read by or written to the Control Bus Interface.

This register can also be used to reset all bits in the Interrupt Condition Register. This task is accomplished by setting all bits in ICCR to one and reset all the bits in ICR to zero. If the interrupt condition is not cleared before resetting bits in ICR, those bits will be set again by the PLAYER device.

Current Transmit State Register

The Current Transmit State Register (CTSR) can be used to program the following functions:

- Generate and transmit symbols
- Control Injection Mode
- Enable the Smoother in the Transmit block
- Enable parity at the PHY Request Interface

The symbol generation and transmission function is used by the PCM state machine to send out FDDI control symbols. The symbols can be generated by setting the three least significant bits in CTSR. Table II shows the Transmit Mode and the corresponding bit assignments.

TABLE II. Transmit Mode Bit Assignment

Transmit Mode	Bit(2)	Bit(1)	Bit(0)
Active	0	0	0
Idle	0	0	1
Master	1	0	0
Halt	1	0	1
Quiet	1	1	0

When the Transmit Mode is set to Idle mode or Halt mode, the PLAYER generates and transmits the Idle symbols or Halt symbols respectively. When it is set to Master mode, the PLAYER transmits the Halt-Quiet symbol pair. In the Quiet Transmit Mode, a stream of Quiet symbols is transmitted using the Quiet Transmit Level programmed in the Mode Register. During the Active Transmit Mode, the transmit block repeats the request data onto its outputs, therefore, Idle symbols and PDUs can be transmitted.

Current Transmit State Comparison Register

The Current Transmit State Comparison Register (CTSCR) ensures that any changes in the Current Transmit State Register (CTSR) are recorded while it is being read or written to by the Control Bus Interface. This register can also be used to reset all bits in the Current Transmit State Register.

Current Receive State Register

Line states being received by the PLAYER device are reported via the Current Receive State Register (CRSR). At the reception of the new line state, information about the previous line state is cleared. This register is different from the two Receive Condition Registers in that the Receive Condition Registers create a historical record of all the line states received whereas the CRSR only reflects the current line state.

Table III shows the line states and the decoded bit assignments.

TABLE III. Line State Bit Assignment

Line State	Bit(2)	Bit(1)	Bit(0)
Active	0	0	0
Idle	0	0	1
No Signal	0	1	0
Master	1	0	0
Halt	1	0	1
Quiet	1	1	0
Noise	1	1	1

The No Signal Detected condition and the reception of Quiet line state condition are very similar. In Quiet line state, a stream of Quiet symbols is received. However, the No Signal Detected condition is reported when the Signal Detect pin (TTLSD) has been deasserted. When the TTLSD pin is deasserted, the receiver probably also receives a stream of Quiet symbols.

Note that CRSR is a read-only register and is often used in a polling application since it cannot be used to generate interrupts.

Receive Condition Register A and Receive Condition Register B

The Receive Condition Registers A and B (RCRA and RCRB) are used to maintain a historical record of the line states received by the PLAYER device.

When a new line state is received, the bit corresponding to the line state is set without clearing any other bits. As a result, line states previously received are also maintained. Although all line states received are being recorded, the registers do not keep track of the sequence of line state reception.

Since the function of the Receive Condition Registers is to record line state changes, not the current line state, it is possible to completely clear the registers by the code sequence in Table IV. Note that RCRA is zero at the end of the code sequence even when a Halt line state is being received.

TABLE IV. Clearing the Receive Condition Register

Time	CRSR	RCRA	Action
T0	QLS	0000 0010	None
T1	QLS	0000 0010	None
T2	HLS	0100 0110	None
T3	HLS	0100 0110	Read RCRA
T4	HLS	0100 0110	Write RCRA = 0
T5	HLS	0000 0000	None

Receive Condition Mask Register A and Receive Condition Mask Register B

The PLAYER can be programmed to generate an interrupt when a line state of interest to the PCM State Machine is received. When a bit in the Receive Condition Register is set by the PLAYER device and the corresponding bit in the Receive Condition Mask Register (RCMRA or RCMRB) is also set, then an interrupt condition is generated and is registered in the Interrupt Condition Register. However, an interrupt is not generated unless the corresponding bits (Bit5 and Bit6) are also set.

Receive Condition Comparison Register A and Receive Condition Comparison Register B

The Receive Condition Comparison Registers (RCCRA and RCCRB) ensures that any changes in the Receive Condition Registers is recorded while they are being read or written by the Control Bus Interface.

These registers can also be used to reset all bits in the Receive Condition Registers. This task is accomplished by setting all bits in RCCRs to one and writing zero to the Receive Condition Registers.

Noise Threshold Register and Noise Prescale Threshold Register

The Noise Threshold Register (NTR) and Noise Prescale Threshold Register (NPTR) are used to set the threshold value of the internal 15-bit noise counter. The noise counter counts the duration that the PLAYER device receives noise.

The Noise Threshold Register is a 7-bit register. It forms the most significant bits of the noise threshold value. The Noise Prescale Threshold Register is an 8-bit register. It forms the least significant bits of the noise threshold value. Therefore, the noise counter takes

$$(NPTR + 1) \times (NTR + 1) \times 80 \text{ ns}$$

to reach zero. For example, if NPTR is 1100 0111 (199),

then NTR is set to 0110 0011 (99) for a threshold value of 1.6 ms. Note that 1.6 ms is the default value of the NS_Max timer used in PCM. Also note that the noise counter can be set at the maximum threshold value of 2.62 ms. When the noise threshold is reached, the Noise Threshold (Bit5) in the Receive Condition Register is set to one, which can in turn generate an interrupt.

Current Noise Count Register and Current Noise Prescale Count Register

The Current Noise Count Register (CNCR) and Current Noise Prescale Count Register (CNPCR) takes a snap shot of the internal noise counter. CNCR reports the most significant 7 bits of the noise counter, whereas the CNPCR is for the least significant 8 bits. Both registers are read-only registers.

Link Error Threshold Register

The Link Error Threshold Register (LETR) contains the starting value of the internal Link Error Monitor counter. The LEM counter is an 8-bit down-counter which decrements if link errors are detected.

The LETR value is loaded into the LEM counter when a value is written to the LETR or when the internal LEM counter reaches zero. When the internal LEM counter reaches zero, an interrupt condition is generated. The interrupt condition is registered in the Interrupt Condition Register. If the corresponding bit (Bit4) in the Interrupt Condition Mask Register is also set, an interrupt is generated.

Current Link Error Count Register

The Current Link Error Count Register (CLECR) serves a function similar to the Current Noise Count Registers. The CLECR takes a snap shot of the internal LEM counter, thus allowing the control process to read the LEM counter without interrupting the LEM counter. The CLECR can be used to calculate the Link Error Rate and it is also needed in the Link Confidence Test during the PCM connection sequence.

State Threshold Register and State Prescale Threshold Register

The State Threshold Register (STR) and the State Prescale Threshold Register (CPTR) are used to set the threshold value of the internal 15-bit state counter. The state counter counts the duration that the PLAYER device receives a line state.

The State threshold Register is a 7-bit register. It forms the most significant bits of the state counter threshold value. The State Prescale Threshold Register is an 8-bit register. It forms the least significant bits of the state counter threshold value. Like the Noise Threshold Registers, the STR and SPTR together can specify a threshold value of 2.62 ms. When the internal state counter reaches zero, the State Threshold (Bit1) is set to one, therefore, an interrupt can be generated.

The internal state counter can be used to keep track of the length of time a particular line state is required to be transmitted or received before the PCM state machine can take appropriate actions. For example, while the PCM state machine is in the Next state, the State Counter can be set at a threshold of 1.6 ms to ensure that the state machine has sent a sufficient number of line state symbols in the Connection state before it moves to the Next state.

Current State Count Register and Current State Prescale Count Register

The Current State Count Register (CSCR) and the Current State Prescale Count Register (CSPCR) take a snap shot of the internal state counter. CSCR reports the most significant 7 bits of the state counter, whereas the CNPCR is for the least significant 8 bits. Both registers are read-only registers.

4.2 POLLING TECHNIQUE

Many techniques can be used by the PCM state machine software to monitor the line state information and the polling technique is one of them.

A sample of a polling algorithm is shown in *Figure 1*. In the example, the PCM state machine just moved to the Next state. In the Next state, the PCM state machine needs to transmit Idle symbols, poll for Quiet line state or Idle line state, and transmit a "bit" signal if idle line state is received.

```

01 CTSR = ITM
02 Loop
03 RxLS = CRSR
04 IF RxLS = QLS
   THEN BreakCondition = TRUE, Done = TRUE
05 IF RxLS = SILS
   THEN LSFlag = TRUE, Done = TRUE
06 IF TPC > T.Out
   THEN BreakCondition = TRUE, Done = TRUE
07 End Loop if Done = TRUE
08 Wait (TL_Min)
09 IF BreakCondition = TRUE
   THEN DO BreakAction
10 IF LSFlag = TRUE
   THEN TxLS = RCode ( n ) , CTSR = TxLS

```

FIGURE 1. Polling Technique Example Code

Let's examine the code in more detail. Line 01 instructs the PLAYER device to transmit Idle symbols. Line 02 is the beginning of the polling loop which ends at line 07.

In the polling loop, line 03 polls the CRSR register and stores the received line state in the variable RxLS. If received line state is Quiet line state, line 04 sets the BreakCondition variable to indicate the PCM state machine needs to return to the Break state. If the received line state is Super Idle line state, it sets the variable LSFlag, indicating that the correct line state is received. Super Idle line state is tested instead of Idle line state because it is required to wait for at least 12 Idle symbols in the Next state before any action. Line 06 checks for the time out condition to ensure the state machine does not get stuck in the Next state waiting for Idle symbols. The polling loop is repeated until one of the conditions is met.

After existing the polling loop, the PCM state machine has to wait for 30 ms (TL_Min) before starting the next action (Line 08). It is used to make sure that the PHY device on the other side has enough time to recognize the line state symbols. If the BreakCondition is TRUE, line 09 will transit to the Break state by executing a subroutine BreakAction. If everything is fine and the expected line state is received, the PCM Pseudo Code machine determines the next transmission mode and directs the PCM state machine to transmit the new line state symbols.

During the polling loop, the PCM software needs to monitor the Current Receive State Register very frequently so that line state information will not be missed. Failure to recognize all the changes of line state can cause failure in the connection sequence and reinitialization of the connection.

4.3 INTERRUPT TECHNIQUE

The interrupt technique can also be used in implementing the PCM software. This section explains a simple sample of a portion of the PCM software using the interrupt capability built in to the PLAYER device.

The code segment shown in *Figure 2* is part of an interrupt handler that responds to the line state reception during the signaling sequence. It serves the same purpose as the code shown in the previous section.

```

01 RCMRA = 0, RMMRB = 0
02 tmp = ICR
03 tmp = tmp AND 1001 0000
04 ICR = tmp
05 RxLS = CRSR
06 IF RxLS = QLS
    THEN DO BreakAction, RETURN
07 IF RxLS = ILS
    THEN TxLS = RCode(n),
09         tmp = RCRA, RCRA = 0,
10         tmp = RCRB, RCRB = 0,
11         RxLS = CRSR,
12         RCRA = RCRARxmSk [RxLS],
13         RCRB = RCRBRxmSk [RxLS],
14         RCMRA = 0010 1111,
15         RCMRB = 0000 0000,
16         CTSR = TxLS
17 RETURN

```

FIGURE 2. Interrupt Technique Example Code

Line 01 is first line of this part of the interrupt handler. The interrupt handler has to examine the device and condition that generates the interrupt before executing the code.

Lines 01 to 04 stop the PLAYER device from generating the interrupt. It first stops the interrupt condition by clearing both RCMRA and RCMRB. Then, it clears the Receive Condition A (Bit5) and the Receive Condition B (Bit6) in the Interrupt Condition Register. Note that the ICR is a conditional write register, as a result, it is read in line 02 before a value is written to it.

Lines 05 to 07 decide what to do with the received line state. If the current line state is Quiet line state, BreakAction is executed and this section of the interrupt handler is completed. However, if the received line state is Idle line state, the interrupt handler needs to prepare the PLAYER for the reception of the next line state and also transmits the next line state.

It first runs the PCM Psuedo Code machine by calling the subroutine RCode(n), where n is the nth bit during the signaling sequence. Lines 09 and 10, prepare the reception of the next line state by clearing the registers RCRA and RCRB. However, clearing all the line states in the RCRA and RCRB register means that the current line state information is also cleared from the two registers. As a result, the line state that PCM is expecting may also be erased. Lines 11 to 13, prevent such a condition by writing the current line state information back to RCRA or RCRB.

At this point, the PLAYER device is ready to receive new line states. Line 14 sets up the mask for RCRA so that when the expected line state is received, an interrupt can be generated. The value 0010 1111 written to RCMRA in this example allows the PLAYER to watch for Halt line state, Master line state, Quiet line state and also the Noise condition. All these are required in the Next state. Line 15 clears the mask for RCRB. This line is not necessary since RCMRB is cleared at the beginning of the code. It is only included as a reminder for other PCM states.

The next line state is finally transmitted in line 16. After the new line state is transmitted, this code segment ends and returns to the calling process.

4.4 OTHER TECHNIQUES

Interrupt and polling are by far the most popular techniques used to implement PCM software. However, implementation of the PCM software is not limited by the two techniques. One other approach is to use the interrupt technique together with an event queue scheduler.

The mix design that uses the interrupt technique and the event queue provides less interrupt handling time and decouples the interrupt handling code from the PCM software. When an event or line state occurs and an interrupt is generated, the interrupt handler unmask the interrupt condition and enqueues the line state event into the event queue. Events in the event queue are taken care of one after another outside of the interrupt handling routines.

Many more possible designs can be used to implement PCM with the PLAYER device. The implementations depend on the underlying hardware platform and cannot be covered by this paper.

5.0 Appendix

This section provides explanations and definitions of terms needed to help understand PCM. However, the ANSI SMT document shall be used as the standard reference. The State diagram from the ANSI SMT document is reprinted in *Figures 3* and *4*.

5.1 PCM STATES

The PCM State Machine is comprised of ten states. Section 4.1 explains the meaning and functions of each PCM state.

PC0: Off State

The Off state is the initial state of the PCM State Machine. The PCM returns to this state upon the reception of the PC_Stop signal from the Entity Coordination Management Entity (ECM).

In the Off state, the PMD optical transmitter is optionally disabled and the PHY device is required to transmit Quiet symbols.

PC1: Break State

The Break state is the entry point of the PCM connection sequence. The Break state is entered upon the reception of the PC_Start signal from ECM. The Break state is also entered from any other state when the connection sequence cannot be completed and a reinitialization is required.

In the Break state, the PMD optical transmitter is disabled and the PHY device is required to transmit Quiet symbols. A few variables are also cleared and initialized during the Break state.

Physical Connection Management Footnotes.

1. Off_Actions:
 - SM_PM_CONTROL.request(Transmit_Disable)*
 - SM_PH_Line-State.request(TRANSMIT_QUIET)
 - CLEAR CF_Loop
 - CLEAR CF_Join
 - CLEAR BS_Flag
2. Break_Actions:
 - SM_PM_CONTROL.request(Transmit_Disable)*
 - SM_PH_Line-State.request(TRANSMIT_QUIET)
 - CLEAR CF_Loop
 - CLEAR CF_Join
 - CLEAR BS_Flag
 - PC_Mode = N
3. Trace_Actions:
 - CLEAR LS_Flag
 - SM_PH_Line-State.request(TRANSMIT_MASTER)
4. Connect_Actions:
 - CLEAR LS_Flag
 - CLEAR BS_Flag
 - SM_PM_CONTROL.request(Transmit_Enable)*
 - SM_PH_Line-State.request(TRANSMIT_HALT)
5. Next_Actions:
 - CLEAR LS_Flag, RC_Flag, TC_Flag, TD_Flag
 - SM_PH_Line-State.request(TRANSMIT_IDLE)
6. Signal_Actions:
 - CLEAR LS_Flag
 - IF T_Val(n)
 - THEN SM_PH_Line-State.request(TRANSMIT_HALT)
 - ELSE SM_PH_Line-State.request(TRANSMIT_MASTER)
7. Join_Actions:
 - CLEAR LS_Flag
 - SM_PH_Line-state.request(TRANSMIT_HALT)
8. Verify_Actions:
 - CLEAR LS_Flag
 - SM_PH_Line-State.request(TRANSMIT_MASTER)
9. Active_Actions:
 - CLEAR LS_Flag, TR_Flag
 - SM_PH_Line-State.request(TRANSMIT_IDLE)
10. Maint_Actions:
 - IF maintls = QUIET
 - THEN SM_PM_CONTROL.request(Transmit_Disable)*
 - ELSE SM_PM_CONTROL.request(Transmit_Enable)*
 - SM_PH_Line-State.request(TRANSMIT_maintls)
11. Break_Required:
 - PC_LS=QLS or (not LS_Flag & TPC>T_Out)
 - Optionally, Break_Required may include the condition TNE>NS_Max:
 - PC_LS=QLS or (not LS_Flag & TPC>T_Out) or TNE>NS_Max
12. Restart_Required:
 - PC_LS=QLS or (not LS_Flag & n≠0 & TPC>T_Out)
 - Optionally, Restart_Required may include the condition TNE>NS_Max:
 - PC_LS=QLS or (not LS_Flag & n≠0 & TPC>T_Out) or TNE>NS_Max
13. Transitions effected by the Break_Required or Restart_Required conditions shall take precedence over other transitions. The reception of QLS while in states 4 to 8 shall cause the PCM to transition to the Break State within PC_React time.
14. RESET TPC on every transition. This includes transitions where the destination state is the same as the originating state.

*This primitive is not required for all PMD implementations.

FIGURE 4. PCM State Diagram Footnote

Upon the reception of Quiet line state or Halt line state, the PCM state machine leaves the Break state and enters the Connect state. If the state machine is stuck in the Break state for a long time (TB_Max), the state machine sets the BS_Flag so that other management agents can examine the condition and takes the appropriate action.

PC2: Trace

The Trace state is used to localize the fault domain of a Stuck Beacon condition where the ring cannot recover from its Beacon state. The state machine enters the Trace state when receiving the PC_Trace signal from ECM while in the Active state.

During the Trace state, the PHY entity transmits a stream of Halt-Quiet (Master) symbol pairs.

The only way to leave the Trace state is receiving the PC_Start or PC_Off signals from ECM.

PC3: Connection State

The Connection state is used to synchronize the two ends of the connection to begin the signaling sequence. It is also used for clock recovery since the Break state does not transmit any clocking information through the optical transmitter.

In the Connect state, the PMD level optical transmitter is enabled and a continuous stream of Halt symbols is transmitted.

Upon the reception of Halt line state, the state machine leaves the Connect state to the Next state. If Idle line state is received before Halt line state, then the connection is not synchronized and the state machine transmits to the Break state to restart the connection sequence.

PC4: Next State

The Next state is one of the two states used in the signaling sequence. The main purpose of the Next state is to separate the "bit" signaling performed in the Signal state. The Next state is also used to transmit PDUs while MAC Local Loop or Link Confidence Test is performed. The PCM Pseudo Code machine is also started in the Next state.

On initial entry into the Next state, a continuous stream of Idle symbols is transmitted. While in the Next state, either a continuous stream of Idle symbols or PDU symbol stream is transmitted.

The Next state terminates and the state machine transits to the Signal state upon the reception of Halt or Master line state or when the PC_Signal signal is received from the Pseudo Code machine. The state machine transit to the Break state upon the reception of Quiet line state.

PC5: Signal State

The Signal state is one of the two states used for the signaling sequence. In the Signal state, individual bits of information are communicated across the connection by transmitting either Halt symbols or Halt-Quiet (Master) symbols pair. Transmitting the Halt symbols is equated to the transmission of a logical one and the transmission of the Master symbols pairs is a logical zero.

Once each individual bit has been transmitted and received, the state machine moves to the Next state before returning to the Signal state for the next transmission. Thus the Next state is used as the bit delimiter between two signaling bits. When all signaled bits have been transmitted and received, the Signaling Sequence ends.

PC6: Join State

The Join state is the first of three states in the join sequence that leads to an active connection. The join sequence assures that both ends of a connection enter the Active state together at the completion of the sequence.

The Join state is entered upon the completion of the signaling sequence when the PC_Join signal is issued from the Pseudo Code machine. In Join state, a continuous stream of Halt symbols is transmitted.

PC7: Verify State

The Verify is the second of three states in the join sequence. The Verify state is entered when Halt line state is received when the state machine is in the Join state.

A continuous stream of Halt-Quiet (Master) symbols pairs is transmitted during the Verify state.

PC8: Active State

The Active state is the last of three states in the join sequence. In this state, the port is incorporated into the token path.

On initial entry into the Active state, a continuous stream of Idle symbols is transmitted. Upon the reception of Idle line state during the Active state, the PHY device is allowed to enter the Active Transmit Mode and PDUs presented to the PHY Port Request interface can be transmitted.

In addition to the normal break conditions, when Halt line state is received in place of Idle line state, the connection is not synchronized and a reinitialization of the connection is required. If the Link Error Rate is too high, the connection also needs to be reinitialized upon entering the Active state.

PC9: Maintenance State

The Maintenance state is used to perform the Maintenance function. The Maintenance state is entered upon the reception of the PC_Maint or PC_Disable signal from other management agency.

5.2 PCM PSEUDO CODE

The PCM Pseudo Code provides and processes the information that is sent between the neighboring PCMs during the signaling sequence. As mentioned in the previous sections, the information is communicated via the bit signaling technique whereas a bit value is represented by a stream of line state symbols. The Halt symbols are used to represent a logical one and the Master symbols pairs for a logical zero.

This section explains the meaning of the ten bits of information communicated during the signaling sequence.

Bit(0): Escape Bit

Bit(0) is called the Escape Bit. It's value is zero for SMT Version 6.2. The setting of Bit(0) is reserved for future assignment by the standard.

Bit(1,2): PC_Type

Bit(1) and Bit(2) together state the PC_Type of this port. The four PC_Types defined in the FDDI standard are encoded as shown in Table V.

TABLE V. PC_Type Encoding

PC_Type	Bit(1)	Bit(2)
PHY_A	0	0
PHY_B	0	1
PHY_S	1	0
PHY_M	1	1

The PC_Type is communicated during the signaling sequence so that the connection policy can be enforced before the station is inserted into the token path.

Bit(3): Port Compatible

Bit(3) is set to one if the two ports on the link are compatible which means that this connection is allowed and supported by both stations. Connections that are not allowed are withheld until Bit(9) is received and a PC_Start signal is generated to reinitialize the connection. All the connections are allowed in the standard except a M-M connection, however certain types of connection can be rejected depending on the implementation.

Bit(4,5): LCT Duration

Four durations of Link Confidence Test are allowed and Bit(4) and Bit(5) specify the LCT duration suggested by this port. If the suggested values from the two ports are different, then the longer duration is used for the Link Confidence Test. The LCT duration is encoded into Bit(4) and Bit(5) as shown in Table VI.

TABLE VI. LCT Duration Encoding

LCT Duration	Bit(4)	Bit(5)
Short	0	0
Medium	0	1
Long	1	0
Extended	1	1

The Short LCT duration is used when there is no recent history of excessive link errors. The Medium LCT duration is used when a failure in LCT occurred. The Long LCT duration is used after the rejection of a link due to an excessive Link Error Rate. And finally, the Extended LCT duration is used when LCT is used to withhold an undesirable connection.

Bit(6): MAC Available for LCT

Bit(6) is set to indicate that a MAC will be placed at this end of the connection during the Link Confidence Test.

If the other port on the link does not have a MAC available for the Link Confidence Test, then this end may optionally source valid PDUs.

If this end cannot connect to a MAC during the Link Confidence Test, then this PHY device is configured to repeat any received PDUs.

Note that LCT is performed after Bit(6) is communicated, during the Next state.

Bit(7): LCT Failed

Bit(7) is set to indicate that the Link Confidence Test was failed by this end of the connection.

If either side signals that the LCT failed, the connection is restarted and a long Link Confidence test is used next time.

Bit(8): MAC for Local Loop

Bit(8) is set to indicate that this end of the connection will provide a MAC for the MAC Local Loop. MAC Local Loop may optionally be used to verify MAC recovery processes, token passing and the neighbor notification process.

If neither side set Bit(8), then the MAC Local Loop is omitted. If this end does not support MAC Local Loop and the other end does, this end of the connection has the option of not performing the MAC local Loop.

Bit(9): MAC on Port Output

Bit(9) is set to indicate that this end of the connection intends to place a MAC in the output token path.

**5.3 PCM TIMERS AND
TIMER EXPIRATION VALUES**

There are only a few timers used in PCM to determine the length of certain operations and the time in which an appropriate response is expected. However, one timer can have different expiration values depending on the state of the PCM software.

For most of the time, the timers are used to measure the maximum timer to wait for the reception of certain line states, the minimum duration to transmit certain line states, the duration for LCT, etc. Therefore, the timer operations help to ensure the two PCMs are synchronized.

The following timers are employed in PCM:

TPC

Physical Connection Timer is the main timer that PCM uses. It is used to ensure state transitions proceed at the desired rate.

TID

The Timer for Idle Detection is used by PCM to measure the time of continuous ILS reception.

TNE

The Timer for Noise Event is used by PCM to detect the length of noise events. If an excessive number of noise is received, the PCM state machine can optionally reinitialize the connection by moving to the Break state.

The rest of this section lists the timer expiration values used in the PCM state machine. All of the expiration values are used for TPC timer unless specified otherwise. The default value is also highlighted for the ease of referencing.

PC_React: 3.0 ms

PC_React states the maximum timer for PCM to make a state transition to the Break state when the connection needs to be restarted. The reinitialization of the connection is needed upon the reception of Quiet line state, when a fault condition is detected, or PC_Start is presented.

TB_Min: 0 ms

The minimum time that PCM transmits Quiet symbols and receives Quiet line state during Break state. This value is rather large because PCM has to wait for the other end to respond.

TB_Max: 50 ms

The maximum time the state machine is allowed to remain in the Break state before an error flag (BS_Flag) is set to indicate that the PCM is stuck in the Break state.

C_Min: 1.6 ms

The minimum time the state machine is required to send Halt symbols after the reception of Halt line state during the Connect state. This timer is used to assure that the other end of the connection has recognized the Halt symbols being transmitted in the Connect state.

LS_Min: 480 ns

The minimum time of continuous reception of Idle line state before Idle line state is recognized by the PCM state machine. The duration of LS_Min is greater than or equal to 12 symbol times to ensure robustness. This expiration value to be used in the Next state and the Active state. It is measured by the TID timer.

TL_Min: 0.3 ms

TL_Min is the minimum time to transmit a line state before advancing to the next state. It is used in the following states: Next, Signal, Join, Verify and Active. The TL_Min value is set to twice the time required for a line state to be recognized by the PHY device in the other end of the link.

T_Out: 100 ms

T_Out specifies the signaling timeout value. It is defined as the minimum time the state machine is required to remain in a state to wait for the line state reception before reinitialization of the connection.

The expiration of T_Out indicates that a line state change is expected but did not happen, the connection has failed and needs to be re-started in the Break state.

T_Next(n): 100 ms

T_Next(n) is the same as the T_Out values but it is used in the Next state since LCT and MAC Local Loop is to be performed during the Next state. T_Next(n) specifies the timeout value for the nth bit in the signaling sequence. T_Next(7) and T_Next(9) specifies a different timeout value other than the default ones.

T_Next(7): LCT Duration

The T_Next(7) value is the negotiated value of the Link Confidence Test duration after the signaling of Bit(4) and Bit(5). It can take on one of the following values: LC_Short (50 ms), LC_Medium (500 ms), LC_Long (5.0 sec) or LC_Extended (50 sec).

T_Next(9): 200 ms

The maximum time for the optional MAC Local Loop to be performed. This timer is used to prevent deadlock while allowing sufficient time for MAC recovery process completion and exchange of neighbor information frames.

NS_Max: 1.3 ms

The maximum time that noise is tolerated before the connection is considered to be unreliable and needs to be re-started. This timeout value is based on the TNE timer.

5.4 SIGNALS

A signal is used to initiate a state change within PCM. It is also used to communicate with other SMT entities. The signals can be generated by PCM or other entities.

The following signals are used in PCM:

PC_Start

A signal that is set by the Entity Coordination Management Entity (ECM) to PCM. PC_Start is used to signal the PCM to initialize a connection.

PC_Start is also signaled by PCM when the Link Confidence Test fails and the connection is re-initialized.

PC_Maint

A signal that is set by a higher-level management entity to PCM. PC_Maint is used to signal PCM to enter the Maintenance state.

PC_Trace

A signal that is set by ECM to PCM. PC_Trace is used to signal PCM to enter the Trace State.

PC_Stop

A signal that is set by ECM to PCM. PC_Stop is used to signal PCM to enter the Off state.

PC_Enable

A signal that is set by a management entity to PCM. PC_Enable is used to signal PCM to move from the Maintenance state to the Off state.

PC_Disable

A signal that is set by a management entity to PCM. PC_Disable is used to signal PCM to move from any state to the Maintenance state.

PC_Signal

A signal that is set by PCM for its internal use during the Signaling Sequence. PC_Signal is used to indicate that the next value is available and ready for transmission.

PC_PDR

A signal that is set by PCM for its internal use. PC_PDR is used to indicate that a PDR is to be transmitted.

PC_Join

A signal that is set by PCM for its internal use. PC_Join is used to indicate that the Signaling Sequence has been completed successfully and the Join Sequence is started.

5.5 PCM FLAGS AND VARIABLES

A flag is a variable that shall take one of two values: set(1) or cleared(0). Flags are used to reflect the status of the state machine and also to signal other entities of the PCM status.

Variables can take on a wider but still a limited set of values. Variables serve the same function of Flags.

The following is a list of flags and variables used in PCM:

PC__MAC__LCT

A flag that is available internally to PCM. This flag is used to indicate that a MAC will be used for the Link Confidence test.

PC__MAC__Loop

A flag that is available internally to PCM. This flag is used to indicate that the MAC Local Loop will be performed before the connection is made active.

CF__MAC

A flag that is set by the Configuration Control Management (CCM) to PCM. This flag is used to indicate that a MAC is available for the Link Confidence test or MAC Local Loop.

BS__Flag

The Break State Flag (BS_Flag) is set by PCM to indicate that the PCM state machine is not leaving the Break state in an expected time interval and a problem is suspected. It can be used by other management agencies to indicate a problem in PCM or the link that needs to be resolved.

LEM__Fail

A flag that is set by PCM to other management entities. It is set to indicate that the Link Error Rate exceeds the LER__Cutoff threshold. The flag is cleared when the Link Error Rate Threshold test is passed. It is used to remove connection with excessive Link Error Rate.

PC__Type

A variable that is set by the higher level management agency to PCM. It is used to specify the type of port being managed by the PCM.

Four different ports are defined:

- A The port in a Dual Attachment Station or Concentrator which attaches to the Primary In and Secondary Out when attaching to the dual ring.
- B The port in a Dual Attachment Station or Concentrator which attaches to the Secondary In and Primary Out when attaching to the dual ring.
- S The port in a Single Attachment Station (SAS) or one of the port in a Single Attachment Concentrator (SAC).
- M The port in a Concentrator that is used to connect with a SAS or SAC.

PC__Neighbor

A variable that is set by PCM to other management entities. This variable is set to indicate the PC__Type of the PHY at the other end of the connection. PC__Neighbor is set during the signaling sequence.

PC__Neighbor can have one of five values: A, B, S, M, or None.

PC__Mode

A variable that is set from PCM to other management entities. This variable is set after the signaling sequence has been completed to indicate the mode of physical connection that has been performed.

PC__Mode can have one of three values:

- Peer PC__Mode is set to Peer when neither the port under control or the port at the other end are of type M. It indicates that this connection exists within the trunk ring.
- Tree PC__Mode is set to Tree when one of the port is of type M. It indicates that the connection exists with a concentrator tree.
- None The connection is neither of the two previous values. PC__Mode is set to None when the connection type is yet unknown.

PC__Withold

A variable that is set by PCM to other management entities. This variable is used to indicate the reason the connection did not get incorporated in the ring.

PC__Withold can have one of the following three values: None, Port M to Port M, or Other incompatible Port Types.

Maint__LS

A variable that is set by other management entities to PCM. This variable is used to indicate the symbol stream to be transmitted when the PCM is in the Maintenance state.

Maint__LS has one of the following values: Quiet, Halt, Idle, Master or PDR.

PC__LS

A variable that is set by PCM to another management entity. This variable is set to indicate the line states received by the PHY.

PC__LS can have one of the following values: QLS, HLS, MLS, ILS, ALS, NLS or LSU.

PC__LCT__Fail

A variable that is set by PCM to other management entities. This variable is used to indicate the number of consecutive failures of the Link Confidence Test.

n

A variable that is set by PCM for its internal use. This variable is used to indicate the number of the next value to be signaled in the Next state and the current value being signaled in the Signal state.

FDDI Station Management with the National Chip Set

National Semiconductor
Application Note 728
Robert M. Grow, Jim Schuessler



1.0 INTRODUCTION

The National DP83200 FDDI Chip Set includes special features that aid in the management of an FDDI station as well as the management of an FDDI ring. An attempt is made here to guide you through some of the details of Station Management (SMT) using National's DP83200 FDDI Chip Set with as little pain as possible. Special circumstances for non-standard applications are also discussed.

Due to the universally acknowledged complexity of the FDDI Standard, it is always a wise idea to have ready access to the original documents when reading collateral material—this is no exception! We recommend obtaining the ANSI X3T9.5 FDDI Standards set¹.

The BMACTM device and PLAYERTM device are two of the devices comprising the National DP83200 FDDI Chip Set.

They both provide a rich set of fully maskable interrupts. These interrupts are used to drive SMT protocols, including Frame Based Management, Connection Management and Ring Management. The BMACTM device includes counters for fault isolation and station and ring performance monitoring that ease the implementation of SMT protocols. The PLAYER device includes multiplexing capability to implement the node configurations called out in the SMT Configuration Management state machine (the most popular being a Single Attach Station (SAS) and Dual Attach Station (DAS)), internal hardware for link error monitoring, and noise timing (see *Figure 1*). The full duplex architecture of the chip set provides for comprehensive testing and fault isolation.

DP83261 BMACTM DEVICE (BASIC MEDIA ACCESS CONTROLLER)

- SMT multicast addressing on-chip
- Full duplex architecture
- Auto generation of Beacon and Claim frames
- Multiple transmit immediate modes
- Multiple diagnostic counters
- Ring latency timer
- 4-bit late counter
- Same information field detection for MAC frame filtering
- Duplicate address detection
- Multiple token detection

DP83251/55 PLAYERTM DEVICE (PHYSICAL LAYER CONTROLLER)

- On-chip configuration switch
- Line state history registers
- Link error detector
- Noise threshold timer
- Unique injection register
- Full duplex architecture

FIGURE 1. National DP83200 FDDI Chip Set SMT Features

¹ There are four documents comprising the FDDI Standard: PMD, PHY, MAC and SMT. The first three are published standards available through ANSI (phone: 212-642-4900), the last, SMT, is still in draft form at the publication time of this Application Note. It can be obtained through Global Engineering Documents, phone 800-854-7179, or 714-261-1455.

Four general areas of management will be discussed in this paper: how to select values of operational parameters, how to use the BMAC device for fault isolation, how to use the PLAYER device for implementing Connection Management (CMT), and how to monitor network status and performance.

2.0 OPERATIONAL PARAMETERS

FDDI supports a broad range of network configurations, and the FDDI standard specifies default parameters for operation of large configurations. The National FDDI Chip Set is designed to operate over an even larger range of configurations for special applications. The default values of these parameters must be changed for larger configurations. In a few systems, it is also valuable to optimize parameters in small ring configurations.

At the core of the timed token protocol implemented in the BMAC device are timers used to control the transmission of information on the network and to detect when ring recovery is required. These timers are loaded with values which ultimately determine the performance of the ring. Ring recovery/startup is a function which can be performed by the BMAC device automatically with default timer values. Other values for these timers can be loaded, but be warned: changing these values from the defaults may cause poor performance, (high usable token latency or low throughput) or worse yet, oscillation between Claim and operational states. For instance, a shorter Valid Transmission Timer (TVX) value might be used on a small ring to accelerate ring recovery, but if the shorter value was used on a large ring it could cause the above problems.

In most cases, the operation and performance of FDDI is determined by the most demanding (typically the shortest!) parameter in all stations on the ring. For example, the station with the shortest TVX value will frequently be the station starting a Claim Process. This is because any timed out TVX will cause the BMAC device to start the Claim Process (TVX timeout indicates no frame received within TVX time).

When powering up, or after a hardware reset, the BMAC device has been designed to revert to default values for many critical parameters; though, a station must initialize the Parameter RAM before participating in an FDDI ring. Parameter RAM values include the long and short addresses, for example.

The use of the BMAC device's programmable timers is discussed below to help illustrate the different alternatives for management of the parameters and selection of proper values for desired operational characteristics.

2.1 The Claim Process

Claim is a ring state which must be completed before the ring can become operational. The objective is to create an interoperable environment in which all stations may communicate with both asynchronous and synchronous traffic. The process does this by setting the ring's Target Token Rotation Time (TTRT) and determining who will issue the first token (the "winner" of claim). Following FDDI rules for the Claim Process, the station with the shortest Requested Target Rotation Time (TREQ) is the "winner", and will determine the Negotiated Token Rotation Time (TNEG) for the ring. TNEG is used as the operational value for the Token Rotation Time (TRT) in all stations on the ring.²

² In many cases, stations will have the same TREQ value. In this case other information in the Claim frame is used to select the "winner".

A simplified Claim Process flow is shown in Table I:

TABLE I. Timer Values in the Claim Process

Value	Becomes	Value
TREQ	→	Tbid in a station's Claim frames
Shortest Tbid	→	TNEG in all stations
TNEG	→	Token Rotation Timer (TRT) when the ring becomes operational

Note: See Recovery Required in MAC Standard

The BMAC device is capable of automatically generating Claim frames. It starts the Claim Process when TVX times out or Tlate=2 (Tlate is a 4-bit counter which increments once each time TRT goes to zero). When the frames are transmitted, the BMAC device places the TREQ value stored in the BMAC device Parameter RAM into the Claim frame. This value then becomes Tbid to the next station on the ring. The receiving BMAC device saves the Tbid from the Claim frame as the potential TNEG, while comparing it to its TREQ. If the received Tbid time is shorter than its own TREQ, it stores the Tbid value as its new TNEG, stops transmitting Claim frames and repeats incoming Claim frames. If the received Tbid is larger than the current TREQ, the station keeps (or starts) transmitting its current Claim frame. The Claim Process completes when the BMAC device receives Claim frames with both source and destination address equal to its own, (its own Claim frame) as well as the Tbid value equal to its TREQ. This process insures that the shortest Tbid value of any node participating in the Claim process ends up as TNEG for all nodes. If an implementation externally generates Claim frames instead of letting the BMAC device generate them, then it is very important that TREQ in the BMAC device Parameter RAM be equal to the first four bytes of the Claim frame (Tbid). If this isn't done, the Claim Process may not complete, and a false duplicate address condition may be detected by the SMT Ring Management protocol.

2.2 Selection of TREQ

Two major factors are used in selecting a TREQ value. The first is the delay and segment size for synchronous traffic. The second is the desired queuing delay for all traffic, both asynchronous and synchronous.³ The first factors: delay and segment size are just another way of specifying the throughput necessary for the synchronous application. The data source is usually isochronous (periodic) and therefore must be serviced or "disaster" will strike. (An example might be voice data, where a delay would result in a noticeable blank spot in continuous speech.)

An application must be able to rely on the stability of TNEG, therefore, TREQ should not be changed frequently. It is generally a bad idea to change TREQ as application processes start and stop. In fact, there are really only two reasons for changing TREQ from the default TMAX: To create a synchronous service period, or to adjust the asynchronous maximum usable__token latency—both relate to token latency.

³ For a discussion of Asynchronous versus Synchronous traffic, see the ANSI X3T9.5 FDDI MAC Standard.

2.2.1 Selection of TREQ for Synchronous Traffic

Changing TNEG, and thus TRT, has significant implications on synchronous traffic. Synchronous service is usually viewed as some number of bits per second; but in FDDI, synchronous service is provided in bytes per token rotation. Each time a token is received, a station may transmit X bytes based on its synchronous allocation. The total synchronous bandwidth allocation for the ring may not exceed TNEG less overhead.⁴

A synchronous application will generally determine the bandwidth requirement on application layer (OSI Layer 7) data; but the synchronous allocation requested in SMT protocols must include overhead for placing the application data in a frame. A change in TNEG changes the framing overhead. This is because the number of bytes of overhead is the same for 1 byte of application data or 4 kBytes of application data. Since an objective of synchronous service is to guarantee bandwidth to an application, a shorter TTRT will cause the application data to be segmented into smaller sizes. So as TNEG is lowered, response time decreases (faster token rotation) but overhead increases and therefore overall throughput decreases. This is the tradeoff between response time and throughput.

For example, the synchronous requirement for an application layer requiring 10,000 bytes per second above the MAC layer would increase 37% on MAC overhead alone when TNEG changes from 20 ms to 5 ms. A little math will illustrate this.

Remember that all bytes transmitted must be counted in the synchronous allocation. The fixed bytes of overhead per frame are shown in Table II.

TABLE II. Fixed Bytes of Overhead Per Frame

Number of Bytes	Portion of Frame
8	Preamble
1	SD (JK Symbol)
1	FC
12	Addresses: SA, DA
...	DATA
4	FCS
2	ED
28	Total

If TRT is 20 ms that means the token will circulate 50 times per second (1/20 ms) at the maximum network utilization. Our requirement is for 10,000 bytes/s which means 200 bytes per frame (10,000 bytes/s)/(50 frames/s). If TRT is 5 ms, the other alternative, the token rotates 200 times per second (1/5 ms). The same 10,000 bytes are delivered in 50 bytes per frame (10,000 bytes/s)/(200 frames/s). Therefore:

5 ms Case:

28 bytes overhead + 50 bytes data = 78 bytes/frame → 36% overhead

78 bytes/frame * 200 frames = 15,600 total bytes (10,000 bytes data)

⁴ Actual time is Target Token Rotation Time (TTRT) less the sum of maximum Ring Latency (D_MAX = 1.617 ms), maximum Frame Time (F_MAX = 0.361 ms), and Token Time (0.00088 ms).

20 ms Case:

28 bytes overhead + 200 bytes data = 228 bytes/frame → 12% overhead

228 bytes/frame * 50 frames = 11,400 total bytes (10,000 bytes data)

There is a 37% decrease in bandwidth in the 5 ms TRT case versus the 20 ms TRT.

If other protocol information like an LLC is transmitted on each frame, the increase would be even worse. In addition, a change in TNEG also creates ring stability problems for previously enqueued synchronous information. Frames which are queued at 20 bytes, for example, would cause the ring to go into the Claim Process if enough of them were queued when TRT changed to a lower value requiring 50 byte frames.

Synchronous service is not well specified in current FDDI documents. If each synchronous application is allowed to pick its own TREQ, then as applications start and stop, TNEG will increase or decrease. In most cases, it is better for an application to learn what the synchronous target time is and segment to that size, rather than dynamically changing it as applications start and stop. This simpler model of operation can be handled in the ring's synchronous bandwidth allocation algorithm.

All this is to say that applications (OSI Layer 7) should not specify or have control over TREQ. Synchronous allocation should be done by a process which has global knowledge of the throughput and latency requirements of all stations. SMT is uniquely qualified for this purpose.

2.2.2 Selection of TREQ for Asynchronous Traffic

Asynchronous traffic is not effected significantly by the value of TNEG in typical systems. Here, the desired TNEG is based on a tradeoff between network throughput and response time. For example, on a large ring of 150 stations (n) at 1 μs latency per station, the maximum throughput is 99% at 20 ms. TNEG, and 97% at 5 ms TNEG.

$$\frac{n(\text{TNEG} - \text{Ring Latency})}{n(\text{TNEG}) + \text{Ring Latency}} = \text{Percentage Throughput} \quad (1)$$

$$\text{Therefore: } 150 (20 \text{ ms} - 150 \mu\text{s}) / (150(20 \text{ ms}) + 150 \mu\text{s}) = 99\%$$

$$\text{and: } 150 (5 \text{ ms} - 150 \mu\text{s}) / (150(5 \text{ ms}) + 150 \mu\text{s}) = 97\%$$

Another important performance characteristic is the maximum usable_token latency. Usable_token latency is the time for a token to return to a particular station, and be usable for asynchronous traffic. This means the TRT has not timed out once since the station last saw the token. (This is opposed to maximum token latency which is 2 times TNEG, a much smaller time.)

It is possible, though highly improbable, that each station in an overloaded ring could use the token for TNEG time (minus ring latency) when the token is captured. This is a worst case scenario. The maximum usable_token latency would then be (n - 1) (TNEG) + 2 (Ring Latency). For this improbable event to occur each station on the ring must transmit for the maximum allowable time—in this case TNEG minus ring latency.

In the above configuration, the maximum usable_token latency is 3 seconds at 20 ms TNEG, and 0.75 seconds at 5 ms TNEG.

$(n - 1) (\text{TNEG}) + 2 (\text{Ring Latency}) = \text{Max. Usable Token Latency}$ (2)

Therefore: $(150 - 1) * 20 \text{ ms} + 2(150 \mu\text{s}) = 3.0\text{s}$

and: $(150 - 1) * 5 \text{ ms} + 2(150 \mu\text{s}) = 0.75\text{s}$

This kind of tradeoff can best be made by a network management station as described later, since a station that attempts to minimize the usable_token latency can adversely effect network throughput in some configurations. For example, if the Ring_latency were 1 ms, instead of the 150 μs above, the change in TNEG from 20 ms to 5 ms would change the maximum throughput from 95% to 85%.

TABLE III. Example Configurations and TNEG Value

Stations	TNEG	Maximum Throughput	Maximum Usable_Token Latency
10	8	0.9986	0.072 ms
	64	0.9998	0.576 ms
	167	0.9999	1.503 s
150	8	0.9811	1.192 s
	64	0.9976	9.536 s
	167	0.9991	24.883 s
500	8	0.9374	3.993 s
	64	0.9922	31.937 s
	167	0.9970	83.334 s

Note: At 1 μs per station latency (including optical fiber propagation delay), actual latency may be different.

Note that a TNEG (TREQ) of 8 ms can significantly reduce the usable_token latency (a good thing) and still not decrease throughput significantly for rings of under about 150 stations.

A robust method of controlling the TNEG of a ring can be created using the operational characteristics described above, and the management features of the BMAC device. Normal stations operate with the default TMAX (approximately 167 ms) and TREQ equal to the TMAX. A network management station determines the desired TNEG for the ring based on knowledge of synchronous applications requirements and ring throughput implications. The network management station sets its TREQ to the desired TNEG value, thus determining the TNEG resulting from the Claim Process. See Table III.

If implementors allow stations to set TREQ independently, then it is advisable that a lower bound on TREQ be enforced to protect the network from serious denial of service problems. When TREQ is changed, the station can cause a Claim Process by setting the CLM bit of the Function Register (FR) in the BMAC device. If this is not done, then it may be a long time before the desired change in TNEG would occur.

2.3 Selection of Asynchronous Priority Thresholds

Asynchronous priorities are set in the THSH1, THSH2, and THSH3 Registers. These priorities are of greatest value when the ring latency is large. Two approaches can be used for setting the thresholds. The first sets the threshold at a load factor, for example 50% load. In the majority of systems, the latency will be small enough that all load factors default to the same effective level, namely transmit if no

frames were transmitted on the last rotation. In the large 150 station ring described earlier, a 1 kByte frame would represent 50% network load to the MAC timers. A longer latency would require a larger frame to get the same 50% load factor. The latency timing feature of the BMAC device allows determination of the appropriate threshold for a load factor. (See Network Monitoring, Section 5, for an example load factor calculation).

The threshold may also be viewed as a reservation of time for transmission of frames at higher priority. In this case the thresholds can be set directly from the tables contained in the BMAC device datasheet (end of Section 5).

2.4 Selection of TMAX and TVX

Stations should only change the TVX and TMAX values when there are critical responsiveness requirements. No improvement in throughput will be achieved, and in most rings, no improvement in network availability will be achieved. However, a discussion of their selection criteria is provided here as guidance for tuning these parameters to the latency of the network.

2.4.1 Selection of TMAX

The BMAC device has been designed so that it can be applied to rings with extremely large latencies. In these cases, the value of TMAX will need to be longer than the default. Making TMAX shorter has a statistically insignificant impact on ring availability, therefore, most implementors need only use the default specified in the FDDI Standard. The BMAC device sets TMAX to default on reset. Special closed systems may benefit from a shorter TMAX value, since a very small set of ring failures is detected by timing for TMAX. In either the longer or shorter TMAX case, the BMAC device can be loaded with the desired value after reset.⁵

2.4.2 Selection of TVX

The Valid Transmission Time (TVX) register is used to hold the FDDI parameter TVX_valid. The token is assumed to be lost if the time between receiving valid frames or non-restricted tokens is longer than TVX. The BMAC device loads TVX with the default value recommended in the FDDI MAC Standard upon reset. TVX need only be changed in rings with latencies larger than 1.7 ms. If a station has a TVX value that is too small, the likely symptom will be a ring that oscillates between the Claim Process and being operational. The optional SMT Parameter Management Frame (PMF) capability can be used by a network manager to attempt to fix an oscillation condition. The National FDDI chip set has been designed with larger than default counters to allow large latency networks, other implementations may not be able to interoperate in one of these very large networks.

In stations which need a non-default TVX value, station implementors can provide a non-volatile storage location. This feature would avoid potential oscillation between the ring being operational and being in the Claim Process, when stations are powered off and on in a larger network than supported by the default value.

2.5 Adjustment of Value for Parameter Encoding

Two representations are used for timer values in the BMAC device. Where accuracy and resolution are important, the chip uses binary encoding. Where network manageability

⁵ See the FDDI Standard for the calculation of these values. The value of DMAX should be computed from the equations in the PHY Standard. TVX and TMAX equations are given in the MAC Standard.

would not be compromised, an exponential representation was used. Tables for converting exponential values are included in the BMAC device datasheet. This optimization saved circuitry which allowed other functions to be included in the BMAC device. When desired values cannot be represented exactly in the chip, the guidelines shown in Table IV can be used.

**TABLE IV. BMAC Device
Parameter Encoding Guidelines**

TREQ	Loaded Time \leq Desired Time
TMAX	Loaded Time \geq Desired Time
TVX	Loaded Time \geq Desired Time
THSH	Loaded Time Is the Closest to Desired Time

2.6 Changing Addresses

The BMAC device has been designed to reduce the number of things that an implementor needs to worry about. The setting of the station addresses, unfortunately, is not one of those things. Addresses can be changed by the SMT processor through the control interface, and here lies potential danger. Dangers exist for both Group and Individual Addresses; but the more serious implications are in changing an Individual Address. If an Individual Address is changed while a frame is on the ring, or still enqueued, then a no owner frame can be created, since the address is changed one byte at a time (a no owner frame will eventually be stripped when it runs into a station which is transmitting). This can be avoided by waiting for the transmit queue to become empty, then disable the Individual Address with the BMAC Device Option Register, until the change is complete. If the implementation uses the optional external Claim or Beacon frames, the address must be changed in those frames also.

The BMAC device also includes an on-chip SMT group address recognition capability. The SMT committee has request addresses for use in SMT frame protocols. These reserved addresses are for the exclusive use of SMT processes. Changes to the base group addresses may result in frames being copied as the result of comparison against a partially changed address. If the group address capability is used for SMT addresses, individual addresses can be enabled and disabled without this problem.

2.7 Denial of Service Protection

Some of the discussion on individual parameters indicated how the ring can become unusable with the improper setting of that parameter. Improper use of parameters or frames can result in disruption of service to the entire ring, or in some cases, to a single station. These conditions can be grouped together as denial of service problems. In general, it is prudent that an implementor only allow operational changes by trusted software. This includes the setting of parameters, as well as the ability to source MAC frames (e.g., Claims and Beacons) and SMT frames. This is simplified by features in the BMAC device like internal Claim and Beacon generation, transmission of Source Address from the Parameter RAM, and reset to default values of important parameters.

3.0 USE OF THE BMAC DEVICE FOR FAULT ISOLATION

In some failure cases, communication on an FDDI ring will be impossible because of a low probability failure within some station on the network. Sometimes, this may result in

the ring being non-operational, and other times the ring may be oscillating between operational and non-operational states. The BMAC device is designed to support network management applications that correct or isolate these rare faults. Described below are several methods which facilitate fault isolation.

3.1 How to Perform Transmit Immediate

Transmit Immediate is a BMAC device feature which allows the transmission of any frame without the ring becoming operational. In other words, no token needs to be received, the station just strips anything received and transmits its frame—thus: Transmit Immediate.

The transmit immediate capability can be used to isolate many faults. One tool that is useful is to allow a network manager to segment the ring. Using this capability, faults can be localized by monitoring the symptom of the failure. For example, if the ring cannot become operational, an application can segment the ring by forcing a configuration change in a remote station(s). If the symptom goes away in the segment containing the network management station, the fault is probably in the isolated segment of the network, if it doesn't, the fault is probably in the remaining segment of the network. The same procedure can then be used with a different remote station until the fault domain is located.

In the case of timer parameter faults, the problem may be corrected directly by performing a transmit immediate of an SMT PMF Request Frame to the station with the invalid parameter.

Applications using the transmit immediate capability must take into account three important items. Differing implementations of transmit immediate, transmission of MAC frames, and the effect of the Ring being Operational.

The BMAC device has the capability to perform transmit immediate under all ring conditions; other implementations do not. Therefore, the application cannot expect a response from other stations.

The fault isolation protocol must take into account that in a ring stuck at the Beacon Process, each repeating station will enter Claim every TMAX, destroying traffic being repeated at that time. As a result, the source or destination of the frame, or any station between may make the transition to Claim, causing an abort of the management frame. The probability of getting a frame to the destination is improved with a few techniques. Setting the Inhibit Recovery Required option (IRR bit in the BMAC device Option Register) will allow the station to transmit complete frames independent of the station's TRT expirations. Setting the IRPT option will stop MAC frames generated by other stations from aborting transmission. A short frame has a statistically smaller chance of being aborted by other stations; but retry of the frame may also be necessary.

3.2 Implementing SMT Events

The BMAC device and PLAYER device are designed to allow for reporting of significant network events. This includes timer expirations, received frame conditions, and counter increments and overflows. All of these conditions are implemented as maskable interrupts. Use of these interrupt conditions can eliminate any need for polling of status in the FDDI logic. SMT frames are transmitted as the result of some of these events. The ability for generation of interrupts on increment of a BMAC device statistical counter (e.g., Error Counter) allows for generation of event report frames directly from BMAC device interrupts.

4.0 USE OF THE PLAYER DEVICE FOR CONNECTION MANAGEMENT

The interface between the PLAYER device and the FDDI connection management (CMT) protocol is designed so that time critical operations are performed by the PLAYER device. The most time critical operation to be performed by the CMT software is PC_React. PC_React is equal to 3 ms and is defined by the standard as the maximum time for the Physical Connection Management (PCM) state machine (implemented by a combination of hardware and software) to make a transition from the active state to the break state in response to the Quiet Line state (QLS), a fault condition, or a request to start the PCM protocol (PC_Start).

Important fault isolation features are provided in the Connection Management (CMT) protocols specified in the SMT standard. The PLAYER device includes counter and interrupt logic to aid in implementing these protocols. The line state reporting of the PLAYER device includes individual reporting of each state transitioned, thus providing a history of all line states seen since the register was cleared. (Registers: Receive Condition A and B (RCRA, RCRB)). This is important since the Physical Connection Management (PCM) is intended to run, even when the optical link has an extremely high error rate. The line state history thus provides greater flexibility to the software implementing these protocols. In addition, individual masking of line state interrupts can eliminate interrupts for line states that are ignored in a specified operational state.

Another fault isolation capability supported directly by the PLAYER device is the noise timer, which detects jabber conditions (continuous transmission) and other faults that may occur in a system. This timer is referred to as the TNE timer in the standard. The PLAYER device has prescale and count registers which load countdown counters. Each Noise Line State, Active Line State or Line State Unknown symbol will decrement the noise counter. The noise counter is used by the Physical Connection Management (PCM) state machine to detect the length of the noise events.

When the TNE timer threshold is exceeded, an interrupt can be generated by the PLAYER device which causes the PCM state machine to transition from the active to the break state. The PLAYER device also includes hardware that implements the Link Error detector function of SMT. The LETR and CLECR registers are also used for performing the Link Confidence Test during link establishment.

5.0 NETWORK MONITORING

Maximum network throughput can be calculated as:

$$n(TTRT - \text{Ring Latency}) / (n(TTRT) + \text{Ring Latency}) \quad (1)$$

Again

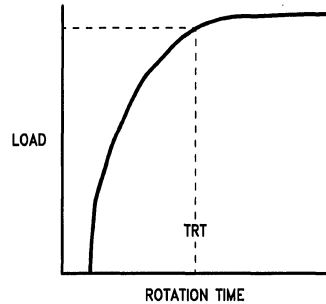
Where n is the number of stations in the network.

This basic equation can be used to determine the proper values for TREQ. This function determines the asymptote for network throughput. The actual utilization of the network can be calculated as:

$$(TRT - \text{Ring Latency}) / TRT \quad (3)$$

Where TRT is the actual token rotation time.

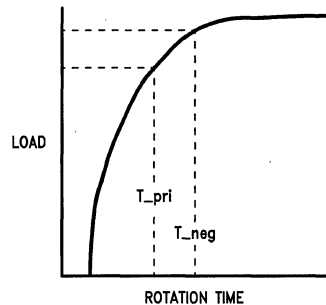
This function produces the curve shown in *Figure 2*.



TL/F/11084-1

FIGURE 2. Timed Token Protocol

If the Token Rotation Time is equal to TNEG, then the network is at the maximum utilization. In a similar way, a T_pri value can be determined for loading into a THSH register by picking the desired load factor and determining the corresponding time value. Conversely, a time value can be picked to determine the maximum utilization at that T_pri.



TL/F/11084-2

FIGURE 3. Asynchronous Priorities

The same basic equation (Eqn. 3) is used to monitor the throughput of the network. Either monitoring the throughput or setting timer parameters requires determination of the ring latency. The BMAC device includes hardware that performs a latency measurement of the ring. The latency of a ring will vary slightly through expansion and contraction of the elasticity buffers and smoothers required in PHY components like the PLAYER device, and more significantly through stations entering and exiting the network. Therefore periodic measurement may be necessary for network monitoring. The latency measurement, obtained through the RLCT counter in the BMAC device, can be used in conjunction with the token counter (TKCT) to determine average load over time. Average load is then represented by the equation.

$$(\text{Time} - (\text{Token_Ct} * \text{Ring Latency})) / \text{Time} \quad (4)$$

For example:

$$(5 \text{ sec} - (5000 \text{ Tokens} * 150 \mu\text{s})) / 5 \text{ sec} = 85\% \text{ Utilization}$$

Over a measurement period of 5 seconds, 5000 tokens were counted in TKCT and the ring latency counter read 150 μ s. Working the formula through results in an 85% load on the ring.

The BMAC and PLAYER devices also include required and optional statistical counters that can be used to evaluate traffic in terms of frames. One of the important counters added is the Frame Not Copied (FNCT) count of the BMAC device. This counter is very useful for evaluation of overload on stations in a network, since it indicates inability of the station to keep up with frames sent to it. Network managers need this type of information for proper administration of server stations. The BMAC device transmit and receive frame counters (FTCT and FRCT) provide valuable traffic information with virtually no software overhead. The error counters in the BMAC and PLAYER devices are useful indications of error rate problems on communication links in the ring. All of these counters are designed to simplify the implementation of station software.

REFERENCES

1. National Chip Specifications.
2. Fiber Distributed Data Interface (FDDI)—Token Ring Media Access Control (MAC), ANSI X3.139—1987.
3. Fiber Distributed Data Interface (FDDI)—Token Ring Physical Layer Protocol (PHY), ANSI X3.148—1988.
4. Fiber Distributed Data Interface (FDDI)—Token Ring Physical Layer Medium Dependent (PMD), ANSI X3.166—1990.
5. Fiber Distributed Data Interface (FDDI)—Station Management (SMT), X3T9.5/84-49, Rev. 6.2.
6. R. Jain, Performance Analysis of FDDI Token Ring Networks: Effect of Parameters and Guidelines for Setting TTRT, DEC-TR-655, September 1989.

Interfacing the HPC46064 to the DP83200 FDDI Chip Set

National Semiconductor
Application Note 736
Simon Stanley



TABLE OF CONTENTS

- 1.0 INTRODUCTION
- 2.0 FDDI INTELLIGENT STATION ARCHITECTURE
- 3.0 DP83200 FDDI CHIPSET
- 4.0 HPC46064 HIGH PERFORMANCE MICROCONTROLLER (HPC™)
- 5.0 CONTROL BUS INTERFACE

1.0 INTRODUCTION

Fiber Distributed Data Interface (FDDI) is a high bandwidth (100 Mbits per second) local area network (LAN) which uses a dual redundant ring architecture. The network consists of a number of point to point links connected to form a ring. The physical and electrical characteristics of the ring protocol are covered by the Physical Media Dependent (PMD), Physical (PHY) and Media Access Control (MAC) Standards as defined by the American National Standards Institute (ANSI) X3T9.5 committee, and can be implemented using the DP83200 FDDI chipset from National Semiconductor. The on-line verification of these point to point links, and the formation of a ring is covered by the FDDI Station Management (SMT) Standard.

This application note covers the use of the HPC46064 High Performance Microcontroller from National Semiconductor to provide the local processing power required within a Fiber Distributed Data Interface (FDDI) node for Station Management (SMT). The note covers the interface between the HPC and the FDDI chipset from National Semiconductor and a possible system architecture.

2.0 FDDI INTELLIGENT STATION ARCHITECTURE

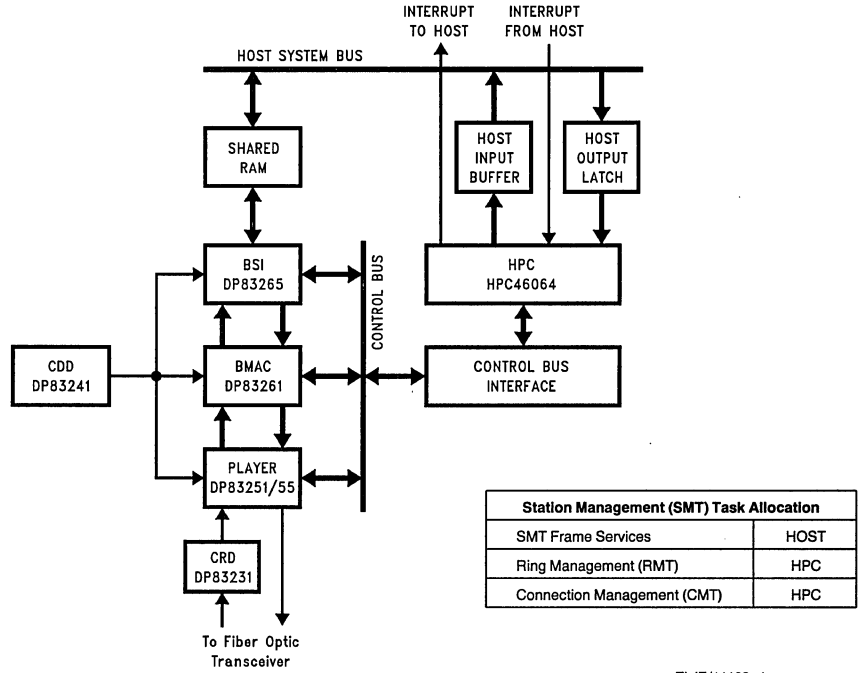
In FDDI, the Station Management (SMT) service is split into three main sections, SMT Frame Services, Ring Management (RMT) and Connection Management (CMT). Within the National Semiconductor implementation of FDDI, any controller handling RMT and CMT services need only access the Control Bus directly, although a communications channel to the host is also required. An architecture using the HPC from National Semiconductor is shown in *Figure 1*. This can be implemented directly using the interface shown in *Figure 2*. Using this architecture, the SMT frame services are provided by the host.

The architecture shown is one of several that could be implemented with the HPC and the National FDDI chipset. This architecture connects the HPC multiplexed address/data bus to the control bus of the FDDI devices, allowing the HPC to access these devices directly with single instructions. The 16 Kbyte ROM of the HPC46064 is used to minimize chip count, though this architecture allows external ROM or RAM to be used if additional functions are implemented.

The HPC has sufficient performance to handle all of SMT, including frame based management, but this would require that the HPC have access to the frame buffer memory. This requires an arbitration scheme to resolve conflicts between the host and the HPC in accessing the buffer RAM. The arbitration scheme could be simply implemented using the HOLD and HLDA (Hold Acknowledge) pins of the HPC.

An advantage of putting all of SMT on the board is that the SMT function need not be resident in the host memory. The removal of TSRs or daemons from host memory leaves more room for applications.

An alternative architecture would run the HPC in single chip mode, reducing the chip count and allowing the Universal Peripheral Interface (UPI) port to be used as the host interface. This would require software drivers to simulate the FDDI control bus signals using the I/O ports of the HPC.



TL/F/11103-1

FIGURE 1. FDDI Station Architecture Using HPC for Partial Station Management

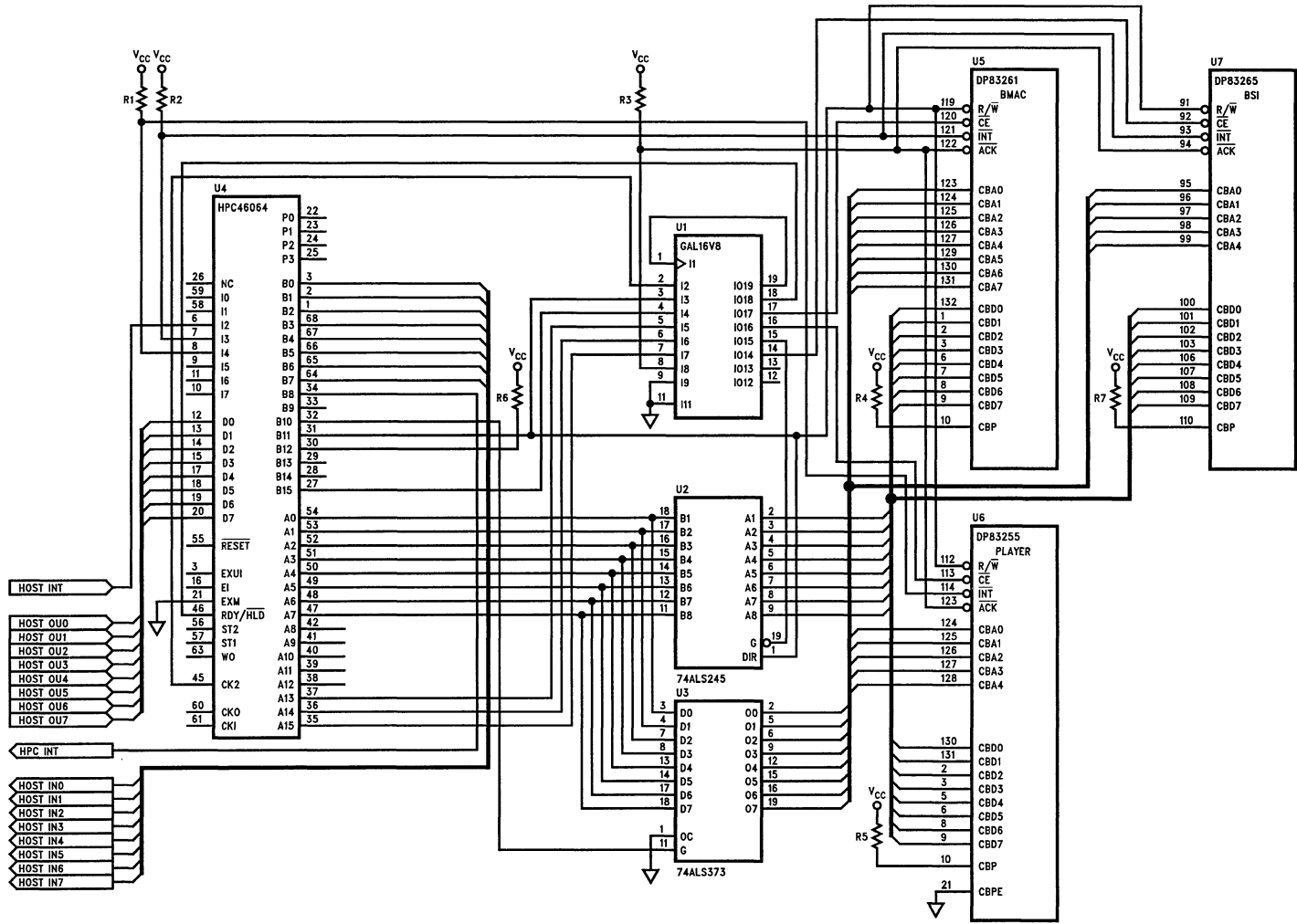


FIGURE 2. HPC to FDDI Chipset Interface

3.0 DP83200 FDDI CHIPSET

The DP83200 FDDI chipset from National Semiconductor implements the PHY and MAC Standards as defined by the American National Standards Institute (ANSI) X3T9.5 committee. The chipset includes the following devices.

DP83231 Clock Recovery Device (CRD™ DEVICE)

The Clock Recovery Device extracts a 125 MHz clock from the incoming data of the upstream station. Its features include on-chip loopback control, on-chip PLL, ability to lock to a Master Line State in less than 100 μ s, and a single +5V supply.

DP83241 Clock Distribution Device (CDD™ DEVICE)

From a 12.5 MHz reference, the CDD generates the 125 MHz, 25 MHz and 12.5 MHz clocks required by the PLAYER™ and BMACT™ devices.

DP83251/DP83255 Physical Layer Controller (PLAYER™ DEVICE)

The PLAYER device converts the BMAC 12.5 Mbyte/s stream into a 125 Mbaud 4B/5B encoded bit stream as specified in the FDDI PHY Standard. It synchronizes the received bit stream to the local 12.5 MHz clock and decodes the 4B/5B data into internal code. The DP83255 PLAYER device also contains a configuration switch for use in dual attachment stations and concentrators.

DP83261 Basic Media Access Controller (BMACT™ DEVICE)

The BMAC implements the functions defined by ANSI X3T9.5 FDDI Media Access Control Standard. The BMAC consists of the transmit and receive state machines, an address magnitude compare unit, a CRC checker, a CRC generator, protocol timers and diagnostic counters.

DP83265 BMAC System Interface (BSI™ DEVICE)

The BSI device provides a multiframe, multiple channel interface between the BMAC device and a host system. The BSI device interfaces directly to the system bus or through low-cost DRAMs. The efficient data structures employed provide high throughput with minimal host intervention.

For more information on these and other devices in the chipset please consult the appropriate data sheets and application notes.

4.0 HPC46064 High Performance Microcontroller (HPC DEVICE)

The HPC46064 is a member of the HPC family of High Performance Microcontrollers. Each member of the family has the same core CPU with a unique memory and I/O configuration to suit specific applications. The HPC46064 has 16 Kbytes of on-chip ROM and is fabricated in National's microCMOS technology. This process combined with the advanced architecture provides fast, flexible I/O, efficient data manipulation, and high speed computation.

The HPC46064 features include 16 bit internal architecture, 16- or 8-bit external data bus, 16 bit address and up to 52 general purpose I/O lines. The HPC46064 also includes a full duplex UART, four 16-bit timers, 16 Kbytes of ROM and 512 bytes of RAM.

The HPC46064 is used in this design because its 16 Kbytes of on-chip ROM remove the need for an external EPROM.

For prototyping or small production runs the pin compatible HPC467064 may be used, which has 16 Kbytes of on-chip EPROM.

Devices in the HPC family can directly address 64 Kbytes of external memory. This address range can be expanded to over 500 Kbytes without additional devices by using I/O pins to implement a simple bank switching technique.

This additional addressing range may be needed if the HPC implements the whole SMT function. The HPC development tools support this bank switching technique, allowing this low-cost microcontroller to handle the whole SMT function while minimizing the software development effort.

The performance of the system will not be affected by using this bank switching approach because many of the SMT functions are never performed simultaneously. The HPC could, for example, run the Physical Connection Management (PCM) from one memory bank, then switch to the bank containing the frame-based management functions at its leisure.

5.0 CONTROL BUS INTERFACE

The FDDI chipset from National Semiconductor provides the PHY and MAC services as detailed in the FDDI Standard. The chipset does not provide the Station Management (SMT) services, but does provide access to all the necessary data through a large array of 8-bit registers located on-board the PLAYER, BMAC, and BSI devices. The interface to these registers is via the Control Bus defined in the data sheet for the BMAC Device (Media Access Controller). The interface consists of an 8-bit address bus, 8-bit data bus and several control lines. The control lines consist of a read/write signal, a chip enable signal and an acknowledge signal. There is also an interrupt signal and a parity bit, which for this application, has been disabled. The interrupt signal and the acknowledge signal are open drain signals and can be wire-ORed.

A transfer cycle on the bus is started when the processor drives the Chip Enable (CE) signal low. Within 20 ns, the data (write cycle only), address and read/write signals must all be valid. The device being accessed will respond by driving the Acknowledge (ACK) signal low when data is valid (read cycle) or when the data has been clocked in (write cycle). The processor can now end the transfer by driving CE high and the device will complete the cycle by driving ACK TRI-STATE®.

The HPC provides a multiplexed 16 bit bus for interfacing to external memory. As only 8 bits are required for the Control Bus, the interface can be achieved with minimal components (see *Figure 2*). The multiplexed address is latched by the flow-through latch, U3. The data bus is buffered by the bi-directional buffer, U2, which is enabled whenever either read or write signals is asserted by the HPC. The direction is controlled by the Write (\overline{WR}) signal. Two chip enables are provided (PCE and BCE) by decoding the read, write and upper address bits. The chip enables are delayed until the next Clock (CK2) falling edge so that during a write cycle, the data is available within 20 ns of the chip enable. The Ready (RDY) signal forces the HPC to insert wait states until an \overline{ACK} is received from the Control Bus. (See *Figures 3* and *4* for interface waveforms.)

No polling of the Control Bus is required as the interrupt signals for each device are brought into the HPC separately. A simple host interface is provided consisting of two interrupt signals and two eight bit ports, one for read and one for write.

Device U1 is a GAL16V8 (a programmable logic device) and performs all of the logic functions required to generate the control signals used by this system. The programming of this device using the ABEL language is straightforward, as shown by *Figure 5*.

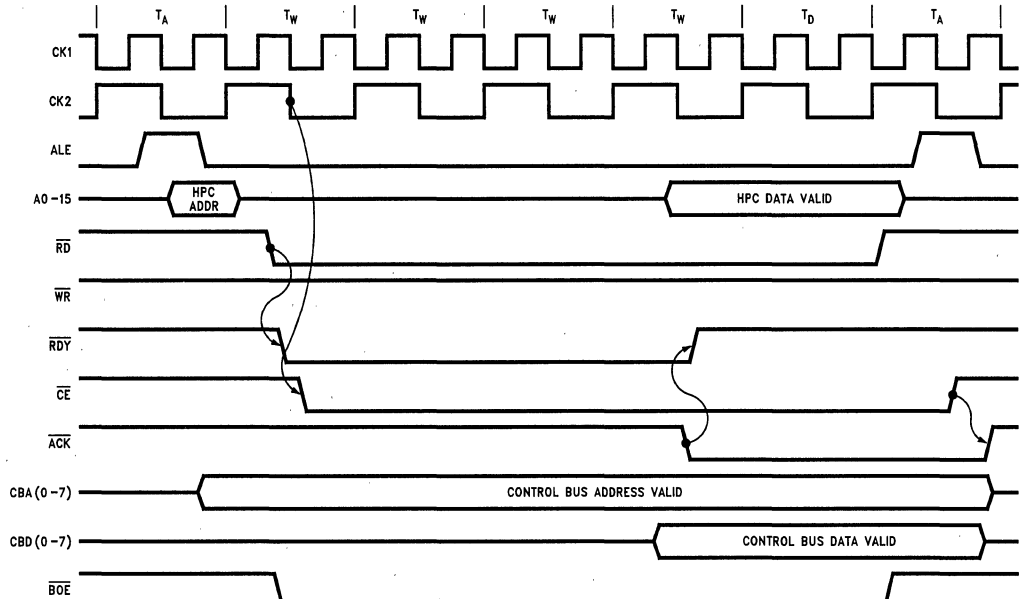
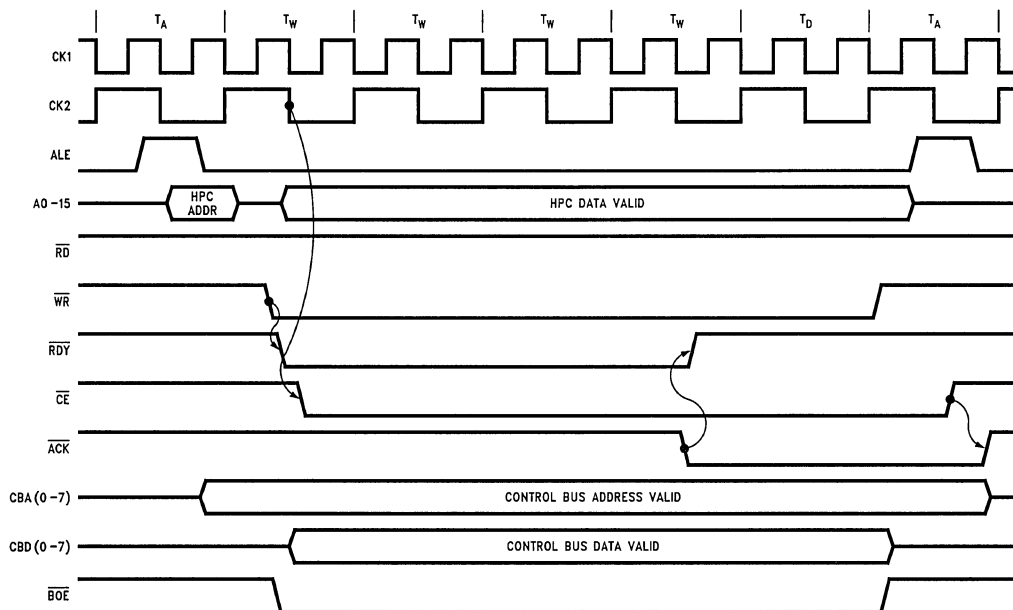


FIGURE 3. Control Bus Interface READ Cycle

TL/F/11103-3



TL/F/11103-4

FIGURE 4. Control Bus Interface WRITE Cycle

Title 'HPC to C Bus Interface
Simon Stanley

National Semiconductor 12 Dec. 1990

```
U01          device 'GAL16V8';
CK2,WR,RD,ACK  pin 2,3,4,8;
A15,A14,A13    pin 5,6,7;
RDY,NCK2,PCE,BCE, BSICE, BOE  pin 18,19,16,17,14,15;
Address = [A15,A14,A13,x,x,x,x,x,x,x,x,x,x,x,x];
```

equations

```
!PCE := (!RD# !WR) & (Address >= h8000) & (Address <= h9FFF);
!BCE := (!RD# !WR) & (Address >= hA000) & (Address <= hBFFF);
RDY = (WR & RD)# !ACK;
!BOE = !WR# !RD;
NCK2 = !CK2;
```

FIGURE 5. ABEL Source File for HPC FDDI Interface PAL

BMAC™ Device Software Design Guide

National Semiconductor
Application Note 678
David Brief



Table of Contents

1.0 INTRODUCTION

2.0 CONTROL SERVICES

3.0 INITIALIZATION PROCEDURE

- 3.1 Put the BMAC Device in Stop Mode
- 3.2 Load the MAC Parameter RAM
- 3.3 Clear the MAC Event Counters
- 3.4 Clear the Event and Mask Registers—Optional
- 3.5 Modify the Timer Thresholds—Optional
- 3.6 Set the Option Register
- 3.7 Set the Mode Register
- 3.8 Loopback Testing—Optional

4.0 DURING OPERATION

- 4.1 Control Interface Event Registers
- 4.2 Event Control
- 4.3 Servicing Interrupts
- 4.4 Event Counters

5.0 EXAMPLE PROCEDURES

- 5.1 Getting the Ring Operational
- 5.2 Receiving and Copying a Frame
- 5.3 Initiating Claim

6.0 DIAGNOSTIC SCENARIOS

- 6.1 For (At Least) One Station
- 6.2 For Two or More Stations
- 6.3 Path Tests

1.0 INTRODUCTION

This application note describes how to initialize the BMAC device and what to do while it is inserted in a ring. The software support to implement the Media Access Control (MAC) level protocol and the necessary services to a Station Management (SMT) entity in an FDDI node are described. The necessary data service support for transmitting and copying frames is not covered in this application note.

The MAC protocol and all of the services required by SMT (except the SMT data services) are supported through the BMAC device's Control Interface. The processor running this software must have access to the Control Bus and have the ability to respond to interrupts.

2.0 CONTROL SERVICES

The Control Services provided by the BMAC device are accessed through the Control Interface. A more detailed description of the facilities and services provided by the BMAC device is given in the BMAC device datasheet.

The BMAC Control Bus address space is divided into 4 address ranges:

The Operation registers control the current mode of operation of the BMAC device (Run/Stop, Internal Loopback) including the options that are being used (Short/Long Addressing, MAC state machine options). In addition several functions may be initiated (Master Reset, MAC Reset, Claim, Beacon).

The Event registers record the occurrence of events which may cause interrupts (each event bit has a corresponding mask bit). These include Ring, Token and Counter Increment/Overflow Events.

The MAC Parameter RAM contains all of the MAC related parameters such as this station's long and short addresses.

The MAC Counter/Timer Thresholds contain the event counters (Frame, Error, Lost, Copied, Not Copied, Transmitted, Token) in addition to the programmed thresholds for the various MAC timers such as TMAX and TVX.

The various ranges may be accessed for reading and/or writing either always or only in stop mode as shown below.

Address Range	Description	Read Cond	Write Cond
00–07	Operation Registers	always ²	always ²
08–2F	Event Registers	always ²	always (Cond) ²
40–7F	MAC Parameter RAM	stop ^{1,3}	stop ^{1,3}
80–BF	MAC Counters/Thresholds	always	stop ¹

Note 1: An attempt to access a currently inaccessible location because of the current mode or because it is a reserved address space will cause a command error (ESR.CCE set to One).

Note 2: Read and write accesses to reserved locations within the Operation, Event Address ranges cause a command error (ESR.CCE set to One).

Note 3: The MAC Parameter RAM is also accessible when:

- a) the MAC Transmitter is in states T0, T1 or T3;
- b) Option.ITC and Option.IRR are set
- c) Function.CLM and Function.BCN are not set otherwise accesses will cause a command error (ESR.CCE set to One).

Note 4: Reserved bits in registers are always read as 0 and are not writable.

3.0 INITIALIZATION

Before being inserted into a ring, the BMAC device must be initialized. To initialize the BMAC device the following steps should be followed. Each action is explained further below.

Put the BMAC device in Stop Mode
 Load the MAC Parameter RAM
 Individual Addresses (MLA, MSA)
 Group Addresses (GLA, GSA, MAP, SGM)
 Requested Token Rotation Time—TREQ
 Beacon Information—TBT
 Clear the MAC Event Counters
 Clear the Event and Mask Registers—Optional
 Modify the Timer Thresholds—Optional
 TVX
 TMAX
 Asynchronous Priority
 Set the Option Register
 Set the Mode Register
 Loopback Testing—Optional

3.1 Put the BMAC Device in Stop Mode

The Mode Register is programmed first to place the BMAC device into STOP mode so that all registers can be accessed.

The Parity for the different interfaces is enabled here as is the ability to be in MAC Loopback. At Initialization it doesn't matter if the part is configured in loopback, but since loopback testing will probably be done after initialization it could be set now as well.

In a system without parity checking on the Control Bus or the MAC Interface the Mode register would be set to 44h. This enables the parity checking on the PHY interface which is actually part of the Ring. (Parity is always generated by the BMAC device to the PLAYER™ device.)

3.2 Load the MAC Parameter RAM

Load RAM with values as indicated in the following passages.

Individual Addresses (MLA, MSA)

MLA—the 48-bit address
 MSA—the 16-bit address—Optional

Group Addresses (GLA, GSA, MAP, SGM)

The same MAP is used for the short and long group addresses. To disable Group Addressing, GLA and/or GSA must be set to all ONE's.

Long Addresses—GLA plus MAP (Optional)

Short Addresses—GSA plus MAP (Optional)

Fixed Group Address—FGM(15:1) This is located at FF (FF FF FF FF) 0x where the last nibble 0x is fanned out using the Fixed Group Map (SGM).

The Broadcast Address—FGM(0) **must** be set to One in order to participate properly in the Next Station Addressing protocols that rely on the Broadcast Address.

Requested Token Rotation Time—TREQ

This should be loaded with FF000000 unless this station is using/managing Synchronous Bandwidth.

When this station is using Synchronous Bandwidth and needs a faster average response time for its Synchronous Bandwidth, the value of TREQ is used in the Claim process to negotiate the target timer rotation time.

If this station wins the Claim process, every station will use this station's value of TREQ as TNEG.

Determining TREQ

TREQ is this MAC's requested value for the token rotation time, i.e., in the worst case, this station wants to "see" a token at least once every 2*TREQ. For example if a station wanted to be guaranteed to capture a token every 2 ms it would set TREQ to 1 ms.

1 ms is 12,500 ticks of the 80 ns clock or 30D4 hex. Subtracting this from 1-0000-0000 yields. FF-FF-CF-2B for TREQ since TRT is an unsigned twos complement up counter.

Since the least significant byte of TREQ is transmitted as 0, the value should be rounded up in order to guarantee that this station will see the token as often as it needs to. In this case TREQ would be written to FF FF D0 00.

Beacon Information—TBT

TBT(31:0) should be loaded with 00 00 00 00.

This is only modified in the case of specialized Beacon Frames. Additional Beacon frames are being defined by the FDDI standards committee.

The BMAC device has two limitations on the Beacon Frames it can transmit. Firstly, only frames with a Null DA with four bytes of information are transmitted by the BMAC device. This precludes the use of the internally generated Beacon Frames for the directed Beacon because it can not be sent to the SMT multicast address. Secondly the size restrictions on Beacon Frames also preclude their use for conveying useful information. The Beacon Frame is the penultimate immediate transmission. (Blocking the MAC Indication Input with Option.IRPT will allow transmission of Beacon Frames in the presence of an upstream Beaconer.)

3.3 Clear the MAC Event Counters

The counters are 20-bit counters, but SMT requires 32-bit counters. This implies that the upper 12 bits are maintained by software.

In order to use the low order bits directly without having to calculate how much they have changed since the last time they were read, the counters should be cleared at initialization.

3.4 Clear the Event and Mask Registers—Optional

The Event and Mask registers are actually cleared on a Master Reset. If you did not do a Master Reset before the initialization sequence it is good practice to clear these registers.

3.5 Modify the Timer Thresholds—Optional

At Master Reset the timer thresholds are set to the defaults recommended by the standard. In most applications there is little incentive to modify the defaults.

TVX

In most applications, the valid transmission timer, TVX, would remain at its default value. The value of TVX determines in how long a valid transmission should be seen. If a valid frame is not seen in this period of time the Claim process is started. This is one of the recovery required conditions.

TMAX

In most applications, TMAX would also remain at its default value. The value of TMAX is used to determine how long to stay in the Claim process before starting to Beacon.

Asynchronous Priority

If not using an Asynchronous priority, the threshold register THSH1 should be set to 0. If using one, the threshold registers should be set appropriately.

3.6 Set the Option Register

There are three types of options that are configurable via this register: the addressing modes, the state machine transitions, and the frame status indicators.

In a typical application only the Long Address would be enabled. In this case the Option Register would be programmed with 02h.

Additional detail on the use of the state machine Options and frame status options is given in the BMAC device data-sheet.

3.7 Set the Mode Register

The Mode Register is set last so that the RUN bit can be set after all other initialization is complete.

The Parity for the different interfaces is enabled here as is the ability to be in MAC Loopback.

During operation in a system without parity checking on the Control Bus or the MAC Interface, the Mode Register would be set to 05h. If Loopback testing is to be done following initialization the Mode register should be set to 45h.

3.8 Loopback Testing—Optional

Loopback testing can be considered an integral part of the initialization sequence. Because the BMAC device is full duplex, loopback testing can be used to check most of the operation of the BMAC device. In loopback it is possible to get an operational ring and check that the claim process is entered and that a token is issued. The token can then be captured and frames transmitted. This allows frames to be sent to oneself to check all of the frame handling logic, address comparison logic, etc. See the diagnostic section below for more details.

Even though the "ring" is very small (2 bytes in internal loopback) an operational ring can still be reached. The BMAC device transmits void frames between tokens since the ring is not big enough to hold a token (and its preamble).

4.0 DURING OPERATION

After the BMAC device is inserted into an operational ring, various SMT processes monitor the event counters and respond to error and exceptional events.

The Ring Management (RMT) process is responsible for getting and keeping the logical ring of MACs operational, thereby allowing the MACs to provide data services to their users. RMT uses the events generated by the BMAC device such as:

- Timer Expirations (TVX, TRT)
- Reception of MAC frames
- Capturing/Passing a Token
- Duplicate Token/Duplicate Address
- Losing a frame
- Not copying a frame

The ring changing operational states in order to monitor the ring and take the appropriate actions when the ring cannot provide data services.

To that end, RMT provides higher level recovery mechanisms than the MACs Claim and Beacon processes, including Directed Beacons. It has the ability to initiate Configuration Management (CMT) recovery including the CMT Trace process. RMT also can take advantage of some of the BMAC device state machine transition options, such as Inhibit Recovery Required, to prevent this station from entering and potentially winning the Claim process.

RMT is also responsible for resolving any duplicate address detection problems that prevent the ring from becoming operational and managing the use of restricted tokens on the ring.

Management Information Base

The station keeps track of all of its information in the Management Information Base (MIB) and updates the information in its database and in the MAC and PHY entities it controls when appropriate. SMT uses the MIB to generate frames conveying information about the station configuration and its "neighborhood". Other stations may optionally access and modify this station's MIB using the optional frame based parameter management protocol (sometimes referred to as the remote set/get protocol).

The MAC related information kept in the MIB includes:

- The current and previous values of the event counters
- Which PLAYER devices the MAC is connected to in the station
- The most recently determined upstream and downstream neighbors (this is useful for creating logical ring maps)
- The addresses that the MAC can interpret
- The current values of TMAX, TVX, TREQ
- Current Synchronous Bandwidth allocations
- Current Asynchronous priority threshold

The station's current estimate of the ring latency is also kept in the MIB. The BMAC device is well suited for this function since it contains a latency counter to measure the ring latency. This is necessary to calculate the ring load and to set meaningful asynchronous priorities. The ring latency is also useful for tuning default timer values that assume a maximum default size ring (TMAX for example).

In addition to the SMT processes occurring within a station, other performance monitoring processes and data service related processes may be active. An example performance monitoring process could measure the load on the network over a period of time by reading the token count periodically and deducing the load from that based on the ring latency and the number of tokens received over a period of time. The ring latency could also be used to optimize certain timing parameters in the station and for accurately setting the asynchronous priorities.

Additionally, the buffering capabilities of the station could be deduced by looking at the number of frames not copied compared with the number of frames that were copied.

4.1 Control Interface Event Registers

The event registers record the occurrence of events. Events are recorded in condition latch registers and contribute to the Interrupt when the bit in the corresponding mask register is enabled. See *Figure 1*.

4.2 Event Control

All events are latched in condition latch registers, and may generate interrupts. Events are grouped into classes according to their probable usage. Event classes are enabled via mask registers in groups at the Interrupt Register and individually at the Condition Registers. Enabled events may generate interrupts. Each condition latch register has a corresponding mask register.

Conditions are used to signify that an event or series of events has occurred. The Interrupt signal becomes active to notify the managing entity that a condition or set of conditions exist. Only enabled conditions contribute to the Interrupt signal.

The interrupt signal is an open drain signal to allow it to be combined with a similar signal from other BMAC and PLAY-ER devices in the station. In this case, software would determine which device is causing the interrupt and process it accordingly.

4.3 Servicing Interrupts

After an interrupt has occurred, the source of the interrupt must first be determined in order to decide how to service the interrupt. In the process of servicing an interrupt, a management entity may use one or both levels of condition masks to disable new interrupts while one is being serviced. Soon after the managing entity has processed the interrupt to some extent, it is ready to rearm the interrupt in order to be notified the next time the event occurs.

The Interrupt Condition Register (ICR) always contains the merged output of the masked condition registers. It is only possible to remove a condition by clearing the corresponding condition latch register bit. Condition latch registers are cleared by writing 0's to the appropriate bits. By storing the events on chip, and having the ability to selectively clear bits, the need for the software to maintain a copy of the event registers is alleviated.

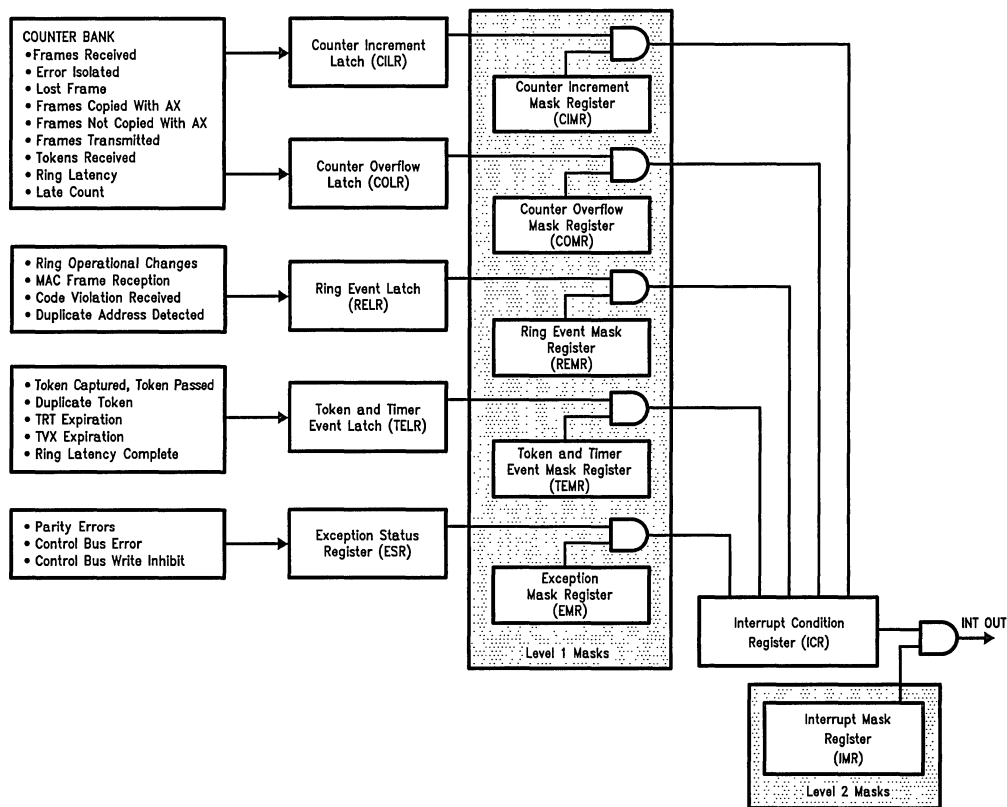


FIGURE 1. Control Interface Event Registers

TL/F/10796-1

Example Interrupt Service Routine

1. Disable Interrupts
2. Determine which event is triggering the interrupt
3. Determine which condition(s) exist that need(s) attention
 - a. Read ICR
 - b. Read appropriate Condition Register
4. Process event
 - a. Complete Processing for the event or
 - b. Queue a process to handle the event
5. Clear or Mask the condition
 - a. To Clear:
 - i. Will only clear conditions that have not changed since last read
 - ii. Make sure that last value read is in the Compare Register
 - b. To Mask:
 - i. Clear the appropriate Mask Bit
 - ii. Before the Mask Bit is set to reenable interrupts, the condition must be cleared as shown above.
6. Reenable interrupts

Additional Notes

1. Nesting of Interrupts:

Nesting of interrupts may be of use in driver level software. For example if an error condition occurs while "processing" a frame it may be prudent to stop processing the frame and handle the error condition. Alternatively, once processing of frames begins, it may not be necessary to reenable frame related interrupts until all copied frames have been processed. This is especially true with token ring protocols where bursts of frames between stations is common (or at least should be common to optimize performance of the media and the software. Software performance would be increased because the software performance can be related closely to the number of interrupts that need to be processed).

2. Conditional Writes:

In the period between the Read of a condition latch register, and the corresponding Write to reset the condition, additional events could occur. To prevent the overwriting and consequent missing of events, an interlock mechanism is used.

Whenever a condition latch register (RELRO, RELR1, TELR, CILR, COLR, or ESLR) is Read, its contents are stored in the Compare Register.

Each bit of the Compare Register is compared with the current contents of the register that is to be written. For any bit that has not changed, the new value of the bit is written into the register. For any bit that has changed the writing of the bit is inhibited. This prevents the software from overwriting bits which have changed since the last read and losing interrupt events. The fact that an attempt was made to modify a changed bit in the register is latched in the Conditional Write Inhibit bit of the Exceptional Status Register (ESR.CW). This bit is written unconditionally after each write to a conditional write register. This is different than in the PLAYER device.

The Compare Register may also be written unconditionally by software. There is a single compare register for all of the conditional write registers in the BMAC device. This is different than in the PLAYER device where each conditional write register has its own compare register.

After a conditional write register is read, if another conditional write register is read before a condition is cleared, the compare register will no longer have the appropriate value in it. In such cases the compare register should be written with the previously read value of the register.

The compare register may also be useful for software compare/update sequences and for diagnostic purposes.

4.4 Event Counters

The event counters are 20-bit counters, but the SMT MIB and SMT frames requires 32-bit counters. This implies that the upper 12 bits are maintained by software.

The counters may be read either periodically or upon an event. The fact that individual counters incremented or overflowed is reported as an event in the CILR or COLR event registers respectively.

In order to use the low order bits directly without having to calculate how much they have changed since the last time they were read, the counters should be cleared at initialization.

Some uses of the counter may require that a consistent value be obtained across two counters. Since the event that a counter incremented is stored, software can tell if a consistent reading was obtained.

When reading individual counters, the upper 12 bits of the counter are latched when the low order 8 bits are read. This allows consistent readings of a single counter and implies that the low order byte must be read first.

At least one of the event counters is incremented for every Starting Delimiter (JK) received.

After a Starting Delimiter (JK) is detected:

```
If Token Ending Delimiter—Increment Token Count
{or} If Format Error—Increment Lost Count
{or} If Frame Ending Delimiter—Increment Frame Count
  If Er = R and FCS error detected—Increment Error Isolated Count
  Else If AFLAG and VCOPY—Increment Frame Copied Count
  Else If AFLAG and not VCOPY—Increment Frame Not Copied Count
```

5.0 EXAMPLE PROCEDURES

5.1 Getting the Ring Operational

To get the ring operational requires setting the Run bit in the Mode Register to a ONE. Once TVX expires, the Claim process will be entered, and if a single token path exists, the Claim process will quickly complete, a token will be issued and the ring will become operational. This occurs even when in internal loopback.

e.g.:

- Set Mode.Run = 1.
- TVX will soon expire causing entrance to Claim.
- Claim will resolve and a token will be issued.
- The reception of a valid token causes the ring to become operational.
- Once the ring is operational the station should check to make sure that $TNEG > T_{min}$ to ensure that it can operate on the ring as an equal station (if this is not true it may be denied service for excessively long periods of time).

5.2 Receiving and Copying a Frame

All frames are received by the BMAC device, but only frames addressed to this station are copied. Every frame received by this station causes the frame received counter to be incremented. In addition for frames addressed to this station (when the A flag is set) either the frame copied (FRCOP) or frame not copied (FRNCOP) bit is set and the appropriate counter is incremented.

e.g.:

- CILR.FRCOP is set indicating that a frame was copied by external_logic
- Can be used to wake up driver software
- Software that receives this interrupt should
 - Process the Frame Status
 - Process some of the frame
 - Pass remainder of frame to another process to be processed

5.3 Initiating Claim

- Enter stop mode (this breaks the ring)
- Load a new value of TREQ into the parameter RAM.
- Load TNEG with TMAX.
- Clear the events related to claim in RELR0 and RELR1
- Enter run mode.
- Initiate the claim process by writing the Function Register with 14.
- Wait until Function Register is zero.
- Check that TNEG \geq TREQ when claim completes.
- See if this station won claim:
 - See if RELR1.MYCLM is set
 - If set see that TNEG = TREQ.

6.0 DIAGNOSTIC SCENARIOS

6.1 For (At Least) One Station

- Control Interface Checkout
- Internal Frame Generation Tests
- State Machine Sequencing Tests
- External Frame Transmissions
- Test of Token Timers
- Test of Transmission Options
- Full Duplex Operation

6.2 For Two or More Stations

- Beacon Scenarios
- Claim Scenarios (Need Three To Do Complete Testing)
- Duplicate Token Conditions
- Duplicate Address Conditions
- Abnormal Frame Termination (Format Errors, FCS Errors, etc.)

6.3 Path Tests

A path test is performed to determine that everything is working in that path. By doing successive path tests on parts of the same path, the Fault Domain can more accurately be determined (i.e., which chip/connector is broken).

The fact that the chip set is full duplex greatly aids the path tests. This allows identical tests to be run at all levels.

Paths that can be tested in a node:

- Through the BMAC Device
- Through All of the Station Paths
- Through the CRD device for SAS and through Each CRD device for DAS and Concentrators
- Through Every Station on the (Logical) Ring

BMAC™ Device Hardware Design Guide

National Semiconductor
Application Note 689
David Brief



Introduction

The BMAC device provides the basic facilities required to implement the Media Access Control functions required by the ANSI X3T9.5 FDDI MAC standard. This document describes how to use the BMAC device to implement the MAC level functionality required by stations in an FDDI network.

A short overview of the BMAC device is given followed by a discussion of design considerations and tradeoffs for using the BMAC device.

Familiarity with the ANSI standard and BMAC Device Datasheet is recommended before using this document.

The state machines and timing shown in this document provide illustrative examples only. Should there be a discrepancy, the BMAC Device Datasheet is the overriding authority.

Table of Contents

- 1.0 OVERVIEW
- 2.0 CONTROL INTERFACE
- 3.0 PHY INTERFACE
- 4.0 MAC INTERFACE
- 5.0 BRIDGING/EXTERNAL MATCHING SUPPORT

1.0 Overview

The BMAC device interfaces to one or more PLAYER™ components, a Control Bus and to external logic that tunes the MAC interface to the requirements of the system.

The BMAC device is comprised of the Ring Engine, the PHY Interface, the Control Interface and the MAC Interface as shown in *Figure 1*.

The BMAC device utilizes a full duplex, byte wide (symbol pair) architecture. There are two bytes of delay in the transmit path, and three bytes of delay in the receive and repeat paths. Two bytes of delay are present in the loopback path.

1.1 RING ENGINE

The Ring Engine implements the FDDI MAC protocol for transmitting, receiving, repeating and stripping frames in a

ring of stations. CLAIM, BEACON and Void frames are generated by the Ring Engine when appropriate. The Ring Engine transmits, strips or repeats Protocol Data Units (PDUs, i.e., Tokens and Frames) and handles the token management functions required by the timed token protocol in accordance with the FDDI MAC standard.

On output (to the ring), interface logic prepares one or more frames for transmission and requests a service opportunity. Based on the requested service class and requested token type, the Ring Engine waits for a token meeting the requested criteria. When a token is captured, the Ring Engine signals the interface and soon thereafter transmission begins. After traversing the ring, frames are stripped based on the Source Address. Frames with a Source Address matching one of the station individual addresses are stripped by the Ring Engine. Status is available at the MAC interface for every transmitted frame.

On input, the Ring Engine sequences through the incoming byte stream, comparing against the station short or long addresses. The results of these comparisons are made available at the MAC Interface. Interface logic then decides how to handle the frame. In the normal case, a frame with a Destination Address matching one of the station addresses is copied and passed to the system.

1.2 CONTROL INTERFACE

The Control Interface implements the interface to the Control Bus by which to initialize, monitor and diagnose the operation of the BMAC device. The Control Interface of the BMAC device is identical to the control interface of the PLAYER device. Typically, all of the BMAC and PLAYER devices within a station will all be addressable on a shared Control Bus.

1.3 PHY INTERFACE

The PHY Interface provides a byte stream to the PLAYER device (PHY Request) and receives a byte stream from the PLAYER device (PHY Indication). The Configuration Switch in the PLAYER device allows the BMAC device to be switched into the Primary and Secondary rings as desired.

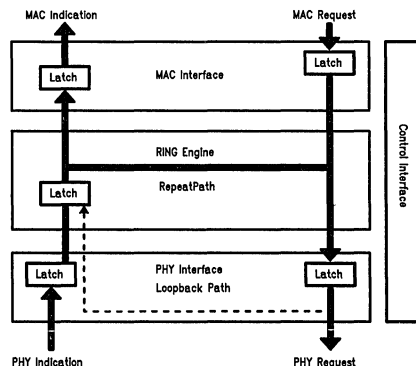


FIGURE 1. BMAC Device Block Diagram

TL/F/10825-15

1.4 MAC INTERFACE

The MAC Interface provides the interface to external buffering and control logic. A byte stream is provided to interface logic with appropriate control signals (MAC Indication), and a byte stream is provided to the BMAC device with appropriate handshake control signals (MAC Request).

It is the job of the external interface logic to transform the byte streams to and from the BMAC device to implement the data services required by the system interface.

2.0 Control Interface

The Control Interface provides an 8-bit asynchronous interface to the Control Bus. Through the Control Interface, access is provided to all internal registers. These registers control the operation of the BMAC device as described below. The interface to the Control Bus is identical to that of the PLAYER device.

The Control Interface provides the synchronization between the asynchronous Control Bus and the synchronous operation of the Ring Engine. The Ring Engine is a synchronous device running exclusively on the 12.5 MHz Local Byte Clock and the in phase 25 MHz Local Symbol Clock.

FDDI components within a station will typically share the same Control Bus. The processor on this bus may be a dedicated microcontroller that is running time critical portions of SMT (i.e., CMT), or it may be a more general purpose microprocessor which gets called on to handle interrupts.

The Control Interface is separated completely from the MAC Interface in order to allow the data paths to run at full speed and not be shared with the slower control transfers. Access to registers may occur simultaneously with the data transfer.

All communication that is not synchronized with the (high speed) data services uses the Control Interface. For example, error conditions are reported through the Control Interface, while frame reception status is reported at the MAC interface synchronized with the data stream. Likewise, options for frame transmission that can change with every frame are submitted to the interface with every frame.

All events are latched in condition latch registers, and may generate interrupts. Only enabled events may generate interrupts. Events are reported through a two level logical hierarchy. Events are grouped into classes according to their probable usage. Each event of a class may be enabled individually in the mask registers and event classes are enabled via mask registers at the Interrupt Register.

Conditions are used to signify that an event or series of events has occurred. The Interrupt signal becomes active to notify the managing entity that a condition or set of conditions exist.

All of the events suggested by the MAC standard are reported by the BMAC device. In addition, the BMAC device provides events on each counter increment and overflow.

The Control Interface also manages access to shared registers. Certain Status and Parameter Registers are not accessible while in Run mode because the Ring Engine may be accessing those locations. All Control Interface accesses are checked against the current operational mode to determine if the register is currently accessible. If not currently accessible, the Control Interface access is rejected (and re-

ported in an Event Register). This means that all Control Interface accesses complete in a deterministic amount of time. This can be useful in the design of certain interfaces to simplify synchronization circuitry. The Exceptional Status Register must be checked to insure that the operation terminated normally.

The registers accessible through the Control Interface maintain the Operation, Event, Status and MAC Parameters.

The Operation Registers are used to control the operation of the BMAC device. The Operation Registers include the Mode, Option and Function Registers. The Mode Register determines the current operational mode (run/stop, loop-back, etc.). The Option Register determines how the BMAC device will work while in run mode. The Function Register initiates functions and provides polled status on their completion. These include a Software Master Reset, a MAC reset and the CLAIM and BEACON requests.

The Event Registers are used to report the occurrence of events. The Event Registers are used to generate interrupts when selected conditions occur (under program control).

The Status Registers are used to access either current status or long term status from event counters.

The Parameter Registers are used to set up parameters used by the Ring Engine such as the station's address and the group address maps.

3.0 PHY Interface

The PHY Interface provides the data paths to connect the BMAC device to one or more PHY Layer devices. The interface is synchronous; with every rising edge of the Local Byte Clock ten bits are transferred in each direction—from the BMAC device to the PLAYER device on the PHY Request Interface and from the PLAYER device to the BMAC device on the PHY Indicate Interface. The elasticity buffer of the PLAYER device removes the asynchronous element from the data stream and allows the BMAC device to work as a synchronous device with synchronous interfaces.

One BMAC device may drive one or more PLAYER devices while only one PLAYER device should be driving the BMAC device at any given time. TRI-STATE® control and pullups are provided within the PLAYER device to allow several PLAYER devices to share one PHY Indicate Bus. This is very important in most Dual Attach and many Concentrator configurations.

FDDI uses a 4B/5B symbol encoding scheme that is normally byte aligned. The PLAYER device aligns starting delimiters of all PDUs to the byte boundary. This allows the BMAC device to detect the JK starting delimiter as a single code point and greatly simplifies the alignment issues. The BMAC device aligns delimiters of all transmitted or repeated PDUs to the byte boundary.

The 10 bits transferred with each clock consist of 8 bits of data and one bit of control to say if the data represents a data code point or a control code point. Mixed data/control symbol pairs are encoded to a set of control code points with the data being given as zero regardless of the actual data symbol value. Odd parity is also provided with every byte.

Through parity is important in this path because every station's data goes through this path. Parity errors detected in the MAC are treated as all other code violations (format errors).

4.0 MAC Interface

4.1 OVERVIEW

The BMAC device is partitioned at the MAC Interfaces to allow a full spectrum of system interface organizations. We discovered early on that it would be difficult, if not impossible to develop an interface that met cost and performance goals for all potential uses of FDDI. However, in crafting the MAC Interface, we did not just guess what a good interface would include. We modeled a complete System Interface Architecture including a real time multi-tasking operating system at the register transfer level on top of the MAC under the assumption that all lower performance interfaces would be some subset of a very high performance interface architecture.

The MAC Interface provides independent Indication (receive) and Request (transmit) Interfaces. Separate signals for control and data are presented at the interface to allow overlapping/pipelining of data and control (status/command) processing. A byte stream is transferred in each direction. On input, the MAC Indication Data byte stream (MID(7:0)) is handled by interface logic using the provided sequencing and status signals. On output, the MAC Request Data byte stream (MRD(7:0)) is generated by interface logic and passed through the BMAC device to the ring on a service opportunity.

The interface is synchronous using the 12.5 MHz Local Byte Clock (LBC). All Outputs change and all Inputs are latched on the rising edge of LBC.

The remainder of this section describes the structure of the MAC Interface.

4.2 MAC INDICATION INTERFACE

Overview

Every byte of all incoming frames is presented at the MAC Indication interface. The MID byte stream is effectively a three byte time delayed version of the PID byte stream. Sequencing signals and addressing flags are provided to help make the decision of whether or not to (continue to) copy the frame.

The Sequencing Signals are asserted at different points within the frame. They are asserted under the following conditions:

RCSTART	when the Starting Delimiter is present on MID
FCRCVD	when the Frame Control Field is on MID
DARCVD	when the last byte of the DA is on MID until the next JK
SARCVD	when the last byte of the SA is on MID until the next JK

INFORCVD when the fourth byte of the Info Field is on MID until the next JK

Not all of the signals would be used by a typical implementation. The sequencing signals are reset on a Master Reset. One of these Termination Event signals is asserted at the end of every PDU as described below:

EDRCVD	when the Ending Delimiter of a frame is on MID until the end of the Frame Status (typically asserted for two byte times)
TKRCVD	when the Ending Delimiter of a token is on MID for one byte time
FRSTRP	when the first Idle byte of a stripped frame is on MID
FOERROR	when the byte with the format error is on MID
MACRST	when a MACRST occurs or BMAC device is in Stop Mode

The Flags provide the input for potential copy criteria and status breakpoints as follows:

AFLAG	Internal DA Match (Short/Long: FCSL); Individual/Group: DIAG); Valid with DARCVD.
MFLAG	INTERNAL SA Match (Short/Long: FCSL); Valid with SARCVD
SAMESA	SA Same as in previous frame: Valid with SARCVD on non-MAC frames.
SAMEINFO	First four bytes of Info same as in previous frame; Valid with INFORCVD on MAC frames.
VDL	Valid Data Length; Criteria—more than the minimum number of bytes and an integral number of symbol pairs: Valid with EDRCVD.
VFCS	Valid FCS: Criteria—received FCS matches with standard CRC polynomial: Valid with EDRCVD.

Note: The Flags are only valid when the corresponding sequencing signal is also set.

For setting the outgoing control indicators, the interface accepts the following:

EA	For external address matches for the setting of the A Indicator (Bridging Group addressing, Aliasing)
VCOPY	For the setting of the C Indicator

When AFLAG or EA is set and the EMIND bit of the Option Register is set, the frame copied counter or the frame not copied counter will be incremented depending on the value of VCOPY for all repeated frames.

Ten MAC Indication Timing Examples are shown in *Figures 2–11*.

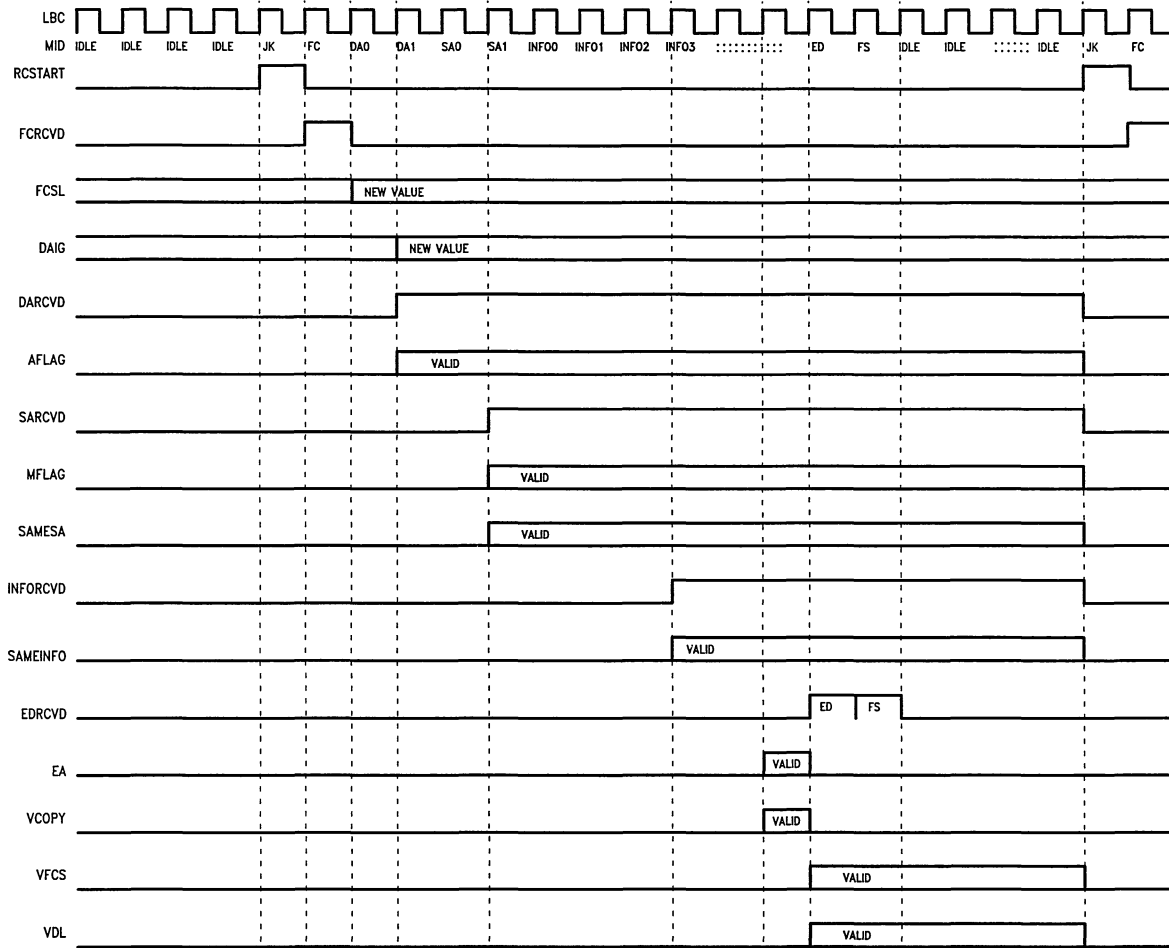


FIGURE 2. MAC Indication Timing Example—Short Frame

TL/F/10825-1

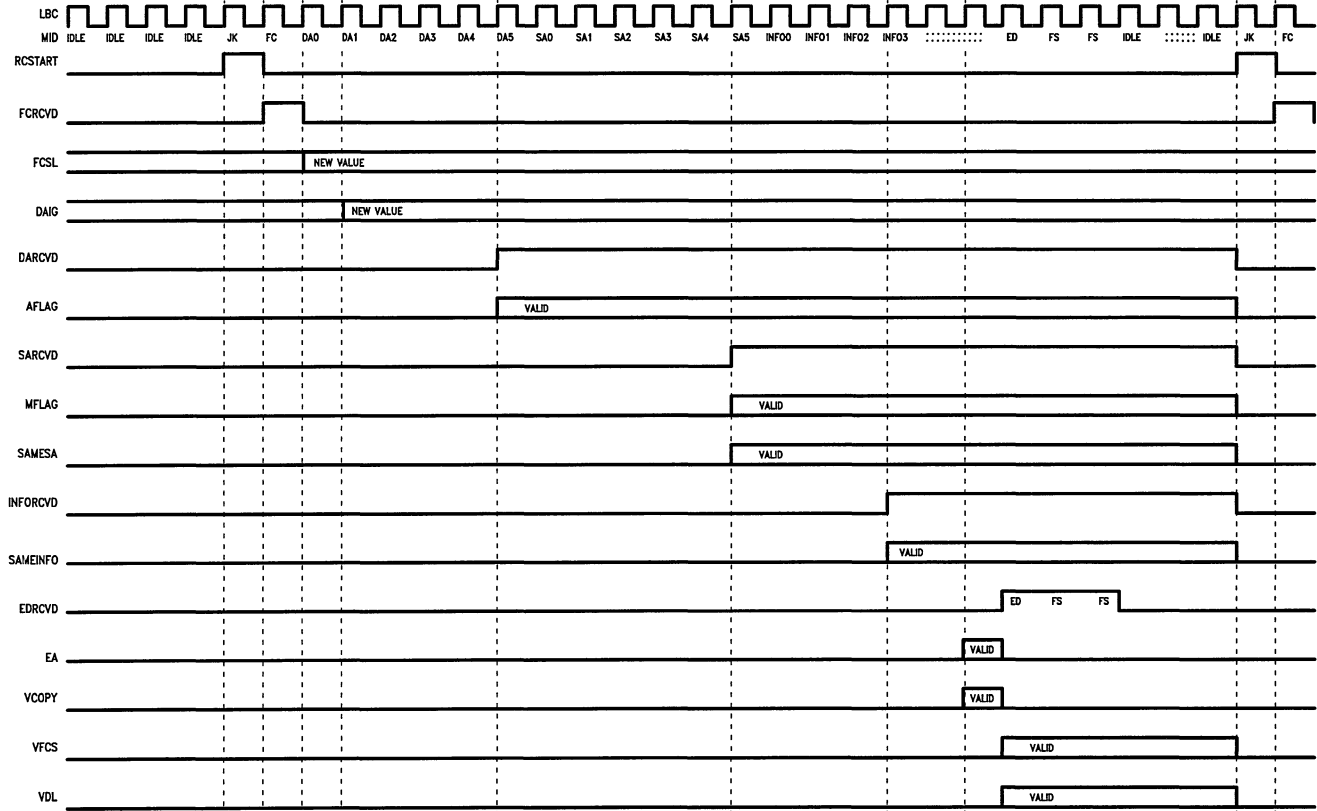
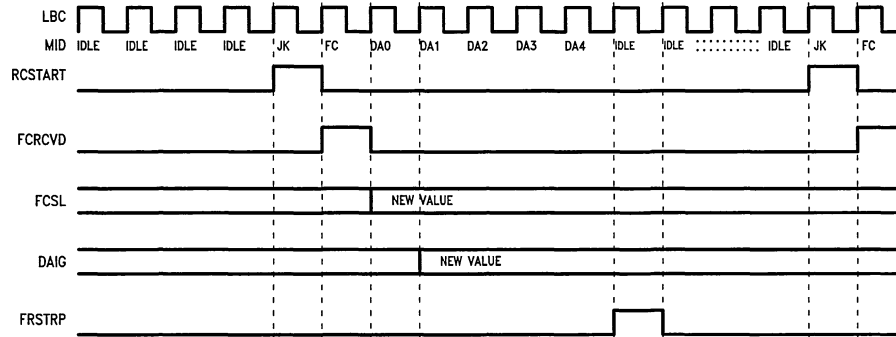


FIGURE 3. MAC Indication Timing Example—Long Frame

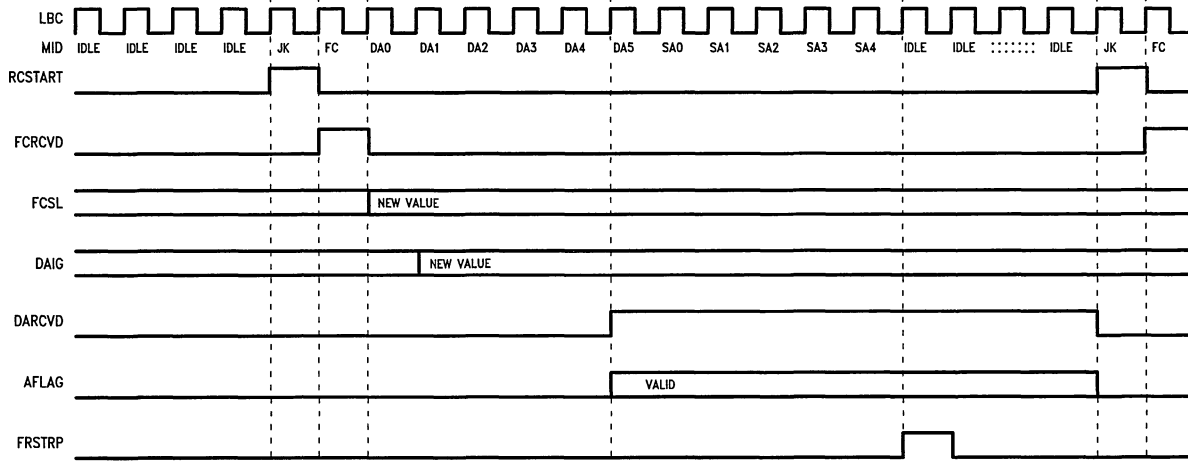
TL/F/10825-2



TL/F/10825-3

Note: FRSTRP may be asserted any time after RCSTART; All signals are held at their current value until the next RCSTART.

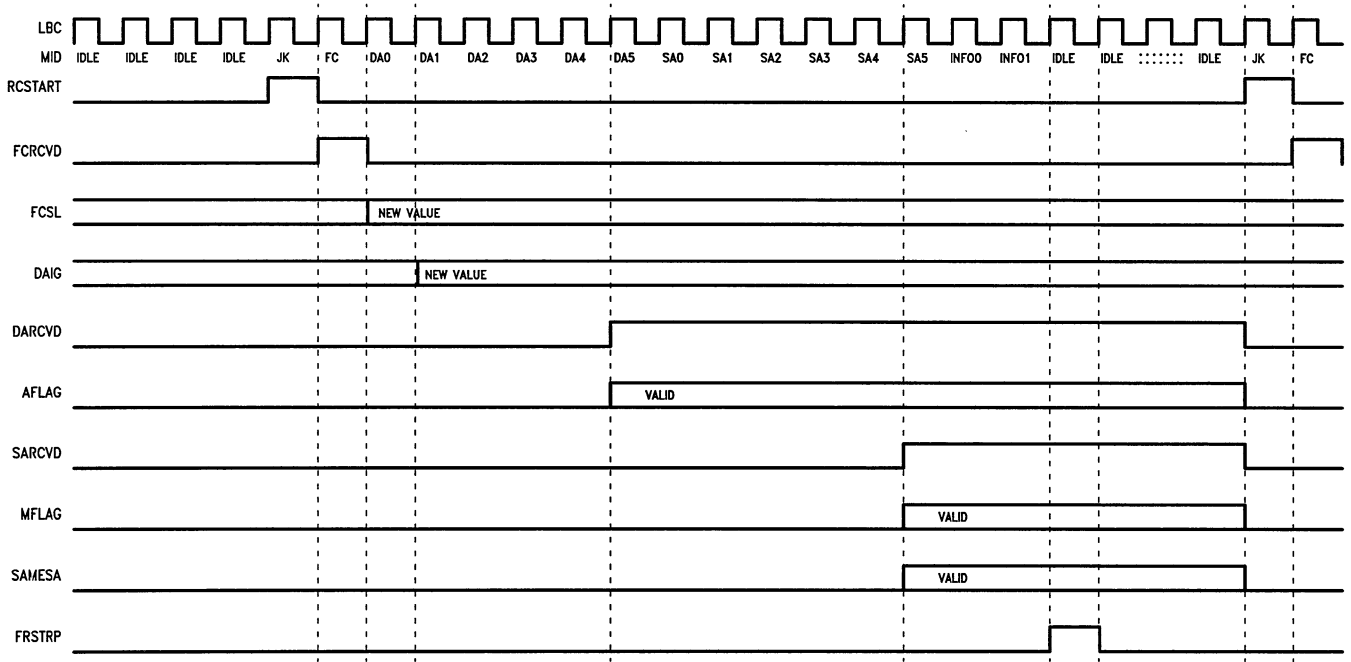
FIGURE 4. MAC Indication Timing Example—Stripped Frame before SA



TL/F/10825-4

Note: FRSTRP may be asserted any time after RCSTART; All signals are held at their value until the next RCSTART.

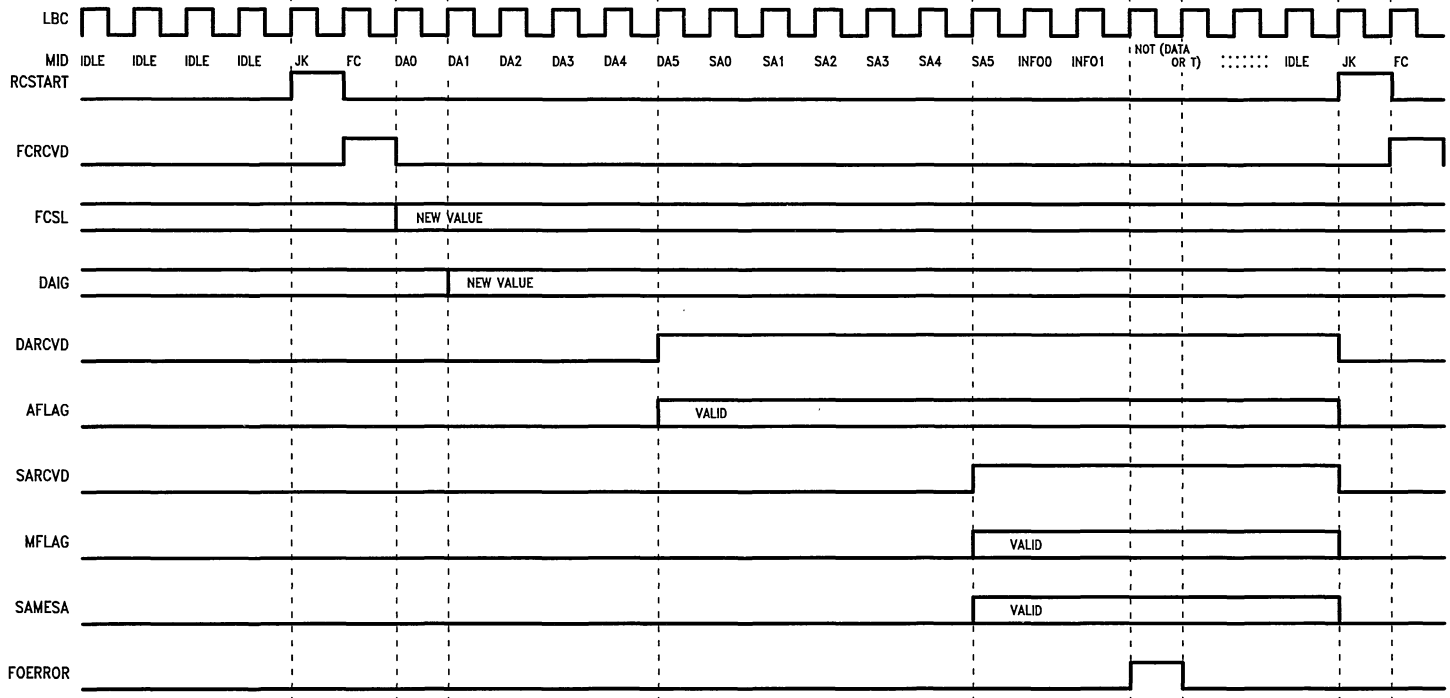
FIGURE 5. MAC Indication Timing Example—Stripped Frame during SA



4-84

Note: FRSTRP may be asserted any time after RCSTART; All signals are held at their value until the next RCSTART.

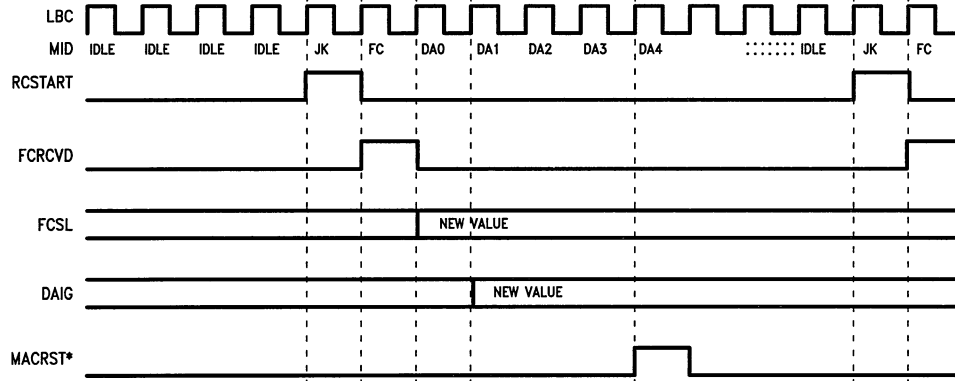
FIGURE 6. MAC Indication Timing Example—Stripped Frame after SA



TL/F/10825-6

Note: FOERROR may be asserted at any time.

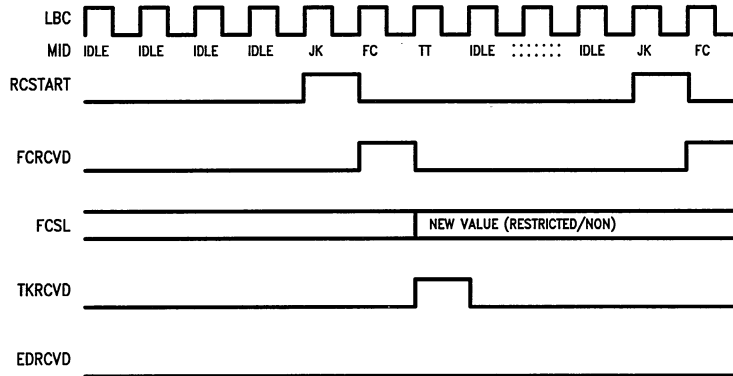
FIGURE 7. MAC Indication Timing Example—Format Error



TL/F/10825-7

Note: MACRST may be asserted if Mode Run = 0.

FIGURE 8. MAC Indication Timing Example—MAC Reset



TL/F/10825-8

FIGURE 9. MAC Indication Timing Example—Token Reception

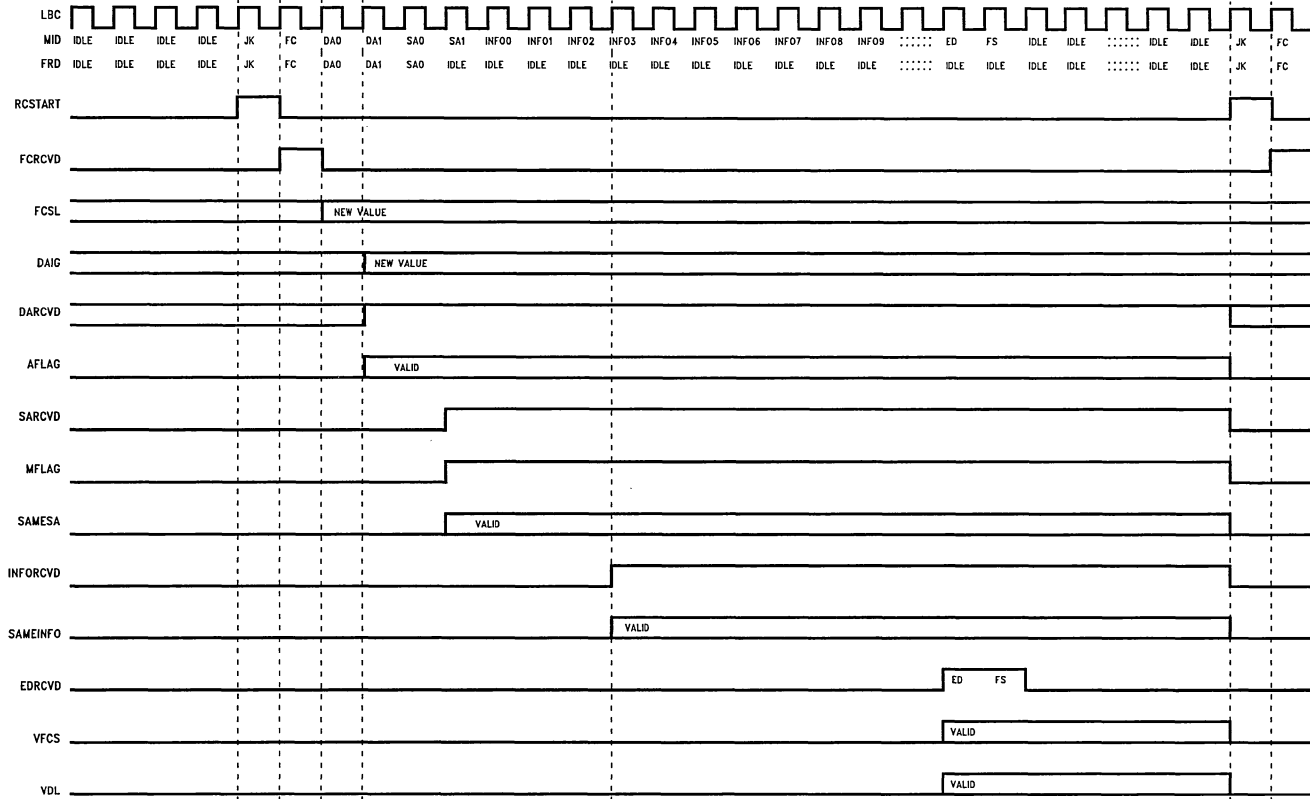
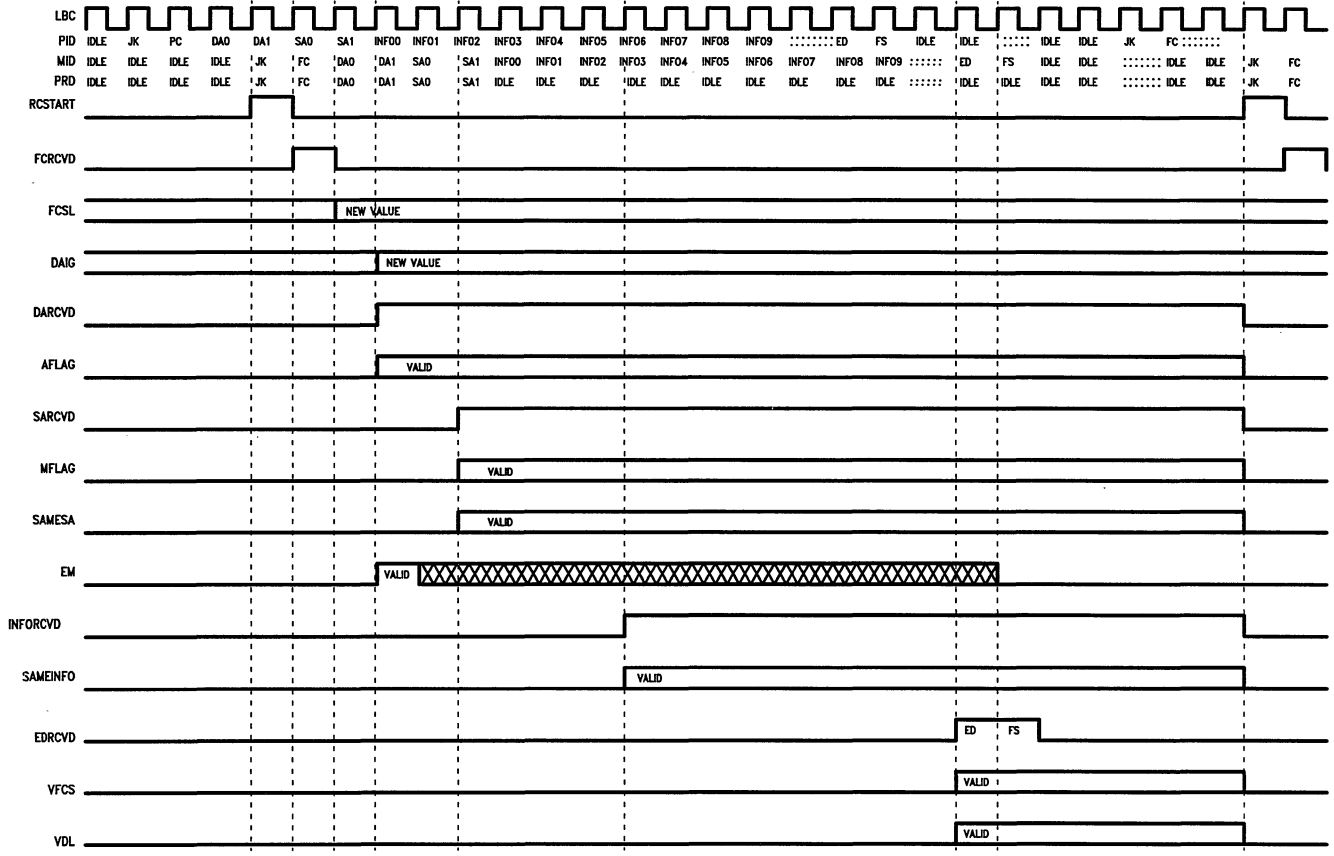


FIGURE 10. MAC Indication Timing Example—Internal Stripping based on MFLAG

TL/F/10825-9



4-88

Note: EM may be asserted at any time; Stripping begins 3 byte times after EM is asserted.

FIGURE 11. MAC Indication Timing Example—External Stripping based on External MFLAG

COPY CRITERIA

The decision to copy or not copy a frame is made by external hardware that looks at flags from the BMAC device qualified by the sequencing signals. The Copy decision may also use inputs from additional address matching logic. The Copy Criteria may be different for frames with different FC values. For example, SMT frames might only be copied on the basis of internal address matches while LLC frames might be copied on the basis of external matches as well. When using different frame copying criteria for different FC values, the copy logic must latch the relevant bits of the FC, or alternatively map the FC, or other fields within the frame, to one of several groups, each of which share the same copy criteria. The frame copying criteria might also be programmable in each group. The BMAC device thus provides the mechanisms by which to create very complicated or simple copy criteria.

A simple copy criteria might be as follows:

All Frames AFLAG and not MFLAG (so that you don't copy frames that you sent, i.e., broadcast and multicast)

A more sophisticated copy criteria might have different copy criteria for each of several groups. In this example, groups are differentiated by the FC.

MAC Not (SAMEINFO and SAMEFC)
 SMT AFLAG or MFLAG
 LLC Synch Short and AFLAG and Not MFLAG
 LLC Asynch Long and AFLAG and Not MFLAG
 Reserved Do not copy
 Implementer Short and external address match

The copy criteria for MAC frames allows a station to skip copying MAC frames with the same (first four bytes of) information and the same FC value.

RECEIVE SEQUENCER

An example receive sequencer is shown in *Figure 12* for the Indicate byte stream.

RS0: Idle

Remain in the Idle state while not copying any data or writing any status. Leave Idle when there is a place to put an incoming frame and the beginning of a frame (RCSTART) is received. At this point the Stage state is entered.

RS1: Stage

Remain in the Stage state while deciding whether or not to copy a frame. The decision is made at the commit point. The commit point occurs any time after both the relevant address comparisons are complete, and after the point in the frame where it is not considered a legal fragment (4 bytes into the Information Field, when INFORCVD is asserted). At the commit point the results of the address comparisons as indicated by the Flags from the BMAC device are compared against the copy criteria. If the criteria matches and no frame termination event has occurred, the decision is made to continue copying the frame and the Copy state is entered. Otherwise, the Idle state is entered and the rest of the frame is not copied.

RS2: Copy

Remain in the Copy state while copying a frame into memory or temporary storage. Once committed to copying, continue copying until a Termination Event (TE) occurs. Upon a TE, status is written. For every frame or partial frame copied, there should be a place to record status.

RS3: Status

Remain in the Status State while Status is being written. After Status has been written, return to the Idle state.

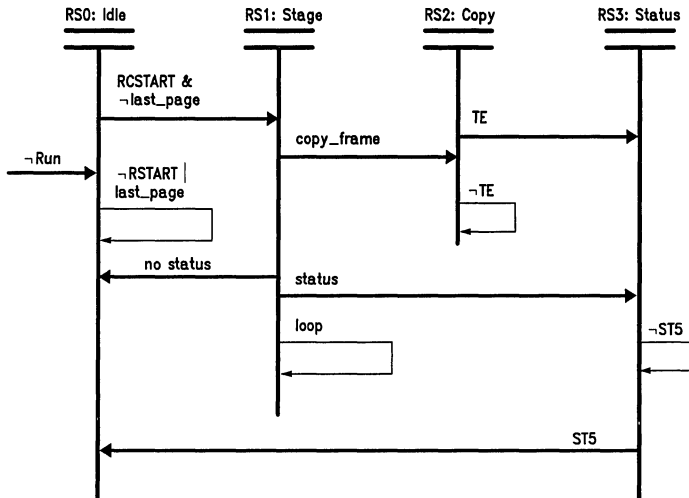


FIGURE 12. Example Receive Sequencer

TL/F/10825-16

CONDITIONS

- $TE = \neg TE \ \& \ | \ (OutofSpace \ | \ EDRCDV \ | \ FRSTRP \ | \ FOERROR \ | \ MACRST \ | \ TKRCVD) \ \{ \text{this is the termination event} \}$
- $copy = \neg TE$ and copydecision and commit_point {copydecision is result of copy criteria}
- status = TE and copydecision
- $nostatus = \neg copydecision$ and $(TE \ | \ \neg TE$ and commit_point)
- $write_status = TE$ and $((RS1$ and copydecision)|RS2 | RS3)
- last_page = {nospace condition}
- SCMP = {status recording is complete}

COMMIT LOGIC

The decision to copy or not is made early in the frame in order to minimize the bus bandwidth used and to minimize the amount of temporary buffering (FIFOing) required.

Status is written at the end of every committed frame, even if an error occurs late in the frame. Since FDDI provides a very reliable transmission system, almost all frames (99.99%+) will be received without errors. In this way all committed frames are handled in exactly the same way.

With this scheme the hardware doesn't handle any exceptions in this time critical and performance sensitive area. It could be a real challenge to recover space on the fly by backing up pointers as when trying to guarantee that back to back frames with minimum preamble can be copied.

RECEPTION STATUS

For each frame received, the following status could be written.

Frame Status [8-Bits]

- Values of received control indicators E, A and C_ind, for each report values of {R, S, T, other} {two bits per indicator}
- Value of FCS check at this station
- Result of Valid Data Length check

Frame Attributes [8-Bits]

- DA Match—Short/Long Ind/Group
- SA Match—Short/Long
- External Match Criteria
- Terminating Condition [2]
 - MAC Reset
 - Valid ED
 - Format Error
 - Frame Stripped by another station

Location and size of the frame (each Data Unit)

The location and size of the frame is either as one contiguous Data Unit or as multiple separate Data Units.

INDICATION BREAKPOINT PROCESSING

Since the performance of a station is proportional to the number of interrupts that must be taken and the amount of status that must be processed, it is desirable to limit the number of interrupts and the amount of status that must be processed (and written).

In one approach, status is always written, but interrupts are only generated at defined breakpoints. In a more sophisticated approach, status is only generated at breakpoints, and interrupts at an even less frequent interval.

As a burst of frames is being received, several options are available for reporting/generating status. A status breakpoint could be requested whenever a frame is received or whenever a burst of frames is received. When status is requested on a burst of frames, the burst is delimited by either events that happen during a burst as a result of a single token opportunity (at the transmitting stations), or events that happen as a result of more than one token opportunity. Within a single token opportunity, status might be generated as a result of the end of the opportunity e.g., a token or MAC frame is received) or if an FC change is detected.

An Indication threshold provides a method of requesting status every n frames. In a large burst of frames status could be requested every n frames. Status could also be requested if a frame is received with unexpected frame status.

Across more than one token opportunity, status can be generated on operations relative to a SAP (Service Access Point). Such status might be generated relative to the last FC, DA or SA received on that SAP as opposed to just the last FC, DA or SA received.

4.3 MAC Request Interface**Overview**

The MAC Request Interface is used to gain access to the ring and to transmit data into the ring according to the requested service class. The interface utilizes a handshake that separates token capture from frame transmission. Upon gaining a service opportunity, a byte stream is passed from interface logic to the MAC Request Interface. The data is passed through the Ring Engine and appears at the PHY Request Interface two byte times later.

While a frame is being transmitted, the Request parameters for the next Request (if different) may be presented to the interface. At the end of the current frame transmission, a decision is made to continue or cancel the current service opportunity based on the new Request parameters.

The MAC Request Interface allows several options on a frame by frame basis. The Request options provide the ability for Source Address transparency and FCS transparency. In both cases, data from the Request stream is transmitted in place of data from the Ring Engine. The use of Source Address transparency has no effect on the sequencing of the interface. When Source Address transparency is not used, the SA from the internal parameter RAM is substituted for the SA bytes in the request stream, which must still be present. As a separate and independent option, the most significant bit of the SA, which is the escape bit for the routing information fields in long SAs, may also come transparently from the data stream, either with a transparent SA or the internal SA.

Since the FCS is appended to the frame, when FCS transparency is not used, no FCS bytes are present in the request stream.

The strip option allows frames with SAs other than this station's to be stripped based on a Void stripping. At the end of a service opportunity where any frame was transmitted with the strip option, two My_Void frames are transmitted. Stripping continues until one of them returns. In this way all frames transmitted on the service opportunity will be stripped. The second My_Void is then stripped based on the SA. The second Void frame is included just in case the first Void frame is lost. This mechanism has the desirable property that the probability of understripping is extremely low. For LANs, frames should not have any possibility of either arriving out of order or being delivered twice.

Reporting Status related to a transmission is one of the most challenging aspects of designing an interface to the BMAC device. During a transmission several errors can occur. A transmission may be terminated unsuccessfully because of external buffering or interface parity errors, internal Ring Engine errors, a MAC reset, or reception of a MAC frame. When a transmission is aborted due to an external error (and Option.IRPT is not set), a Void frame is transmitted to reset the TVX timers in all stations in the ring.

The BMAC device also guarantees that a valid frame is sent with at most $\underline{\hspace{1cm}}$ MAX preamble (3.20 μ s of preamble). This alleviates the requirement from being handled by the interface logic. During a service opportunity when data is not ready to be transmitted, Void frames are transmitted to re-

set the TVX timers in all stations. During an immediate request from the CLAIM or BEACON states, when no CLAIM or BEACON frames are ready, the internally generated CLAIM or BEACON frames are transmitted.

SERVICE CLASSES

A token capture class of "non-rstr" indicates that the transmitter token class must be non-restricted to begin servicing the request. A token capture class of "rstr" indicates that the transmitter token class must be restricted to begin servicing the request. A token issue class of "non-rstr" means that the transmitter token class will be non-restricted upon completion of the request. A token issue class of "rstr" means that the transmitter token class will be restricted upon completion of the request. (See Figure 13.)

RQRCLS (3:0)	Name	Class	THT	Token Capture	Token Issue	Notes
0000	None	None				
0001	April_1	Async thsh1	enabled	non-rstr	non-rstr	
0010	April_2	Async thsh2	enabled	non-rstr	non-rstr	
0011	April_3	Async thsh3	enabled	non-rstr	non-rstr	
0100	Syn	Synch	disabled	any	captured	1
0101	Imm	Immediate	disabled	none	none	4
0110	ImmN	Immediate	disabled	none	non-rstr	4
0111	ImmR	Immediate	disabled	none	rstr	4
1000	Asyn	Async	enabled	non-rstr	non-rstr	
1001	Rbeg	Restricted	enabled	non-rstr	rstr	2, 3
1010	Rend	Restricted	enabled	rstr	non-rstr	2
1011	Rcnt	Restricted	enabled	rstr	rstr	
1100	AsynD	Async	disabled	non-rstr	non-rstr	2
1101	RbegD	Restricted	disabled	non-rstr	rstr	2, 3
1110	RendD	Restricted	disabled	rstr	non-rstr	2
1111	RcntD	Restricted	disabled	rstr	rstr	2

Note 1: Synchronous requests are not serviced when RELR.BCNR is set.

Note 2: Restricted requests are not serviced when RELR.BCNR or RELR.CLMR are set.

Note 3: Restricted Dialogues only begin when a non-restricted token has been received and transmitted.

Note 4: Immediate Requests are serviced when the ring is non-operational. These requests are serviced from the Data state if neither RQCLM nor RQBCN are asserted. If RQCLM is asserted, Immediate requests are serviced from the CLAIM state and if RQBCN is asserted, Immediate requests are serviced from the BEACON state. RQCLM and RQBCN do not cause transitions to the CLAIM and BEACON states. Function.CLM and Function.BCN cause these transitions.

FIGURE 13. Request Service Classes

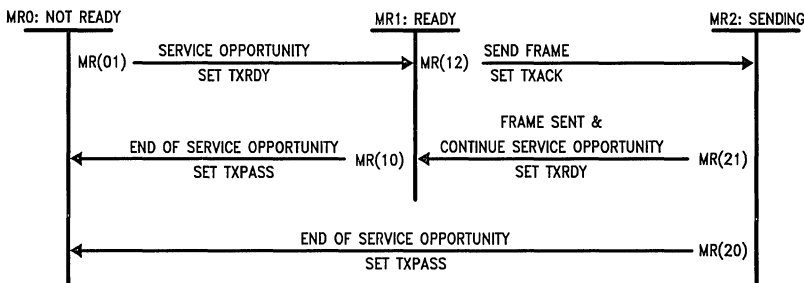


FIGURE 14. MAC Request Interface State Diagrams

TL/F/10825-12

The Logical States of the MAC Request Interface

The MAC Request Interface has three logical states: either the Ring Engine is not ready to service a request, the Ring Engine is ready for the next frame from the interface, or the Ring Engine is sending a frame from the interface. See *Figure 14*.

The values of TXRDY and TXPASS associated with these three states are shown below.

State	TXRDY	TXPASS
Not ready	= 0	= 1
Ready	= 1	= 0
Sending	= 0	= 0
Internal error	= 1	= 1

Conditions

Service Opportunity: A service opportunity occurs when it is possible to service the current request, as defined by the current service parameters (RQRCLS, RQCLM and RQBCN).

The last latched version of RQRCLS is used when RQRDY is asserted.

Send Frame: A frame can be sent from the interface when at least 8 bytes of preamble have been transmitted, TXRDY,

RQRDY and RQSEND are asserted, and RQFINAL has not yet been asserted for this request.

Continue Service Opportunity: The Service Opportunity is continued after the current frame if valid service parameters continue to be presented during the frame, and the timer(s) used for the (next) requested service class have not reached their threshold.

The table below shows the timer thresholds used for each service class:

Service Class	Threshold
All Requests	TRT Expiration
All Requests with THT Enabled	THT Expiration
Priority Asynchronous Requests	Asynchronous Priority Threshold

End of Service Opportunity: The end of a service opportunity occurs when it is no longer necessary or possible to continue the service opportunity. The service parameters are continuously compared with the current state of the Transmitter. If an unserviceable request is presented or any time threshold is reached, the service opportunity will not continue after the current frame (if any).

Two MAC Request Timing Examples are shown in *Figures 15 and 16*.

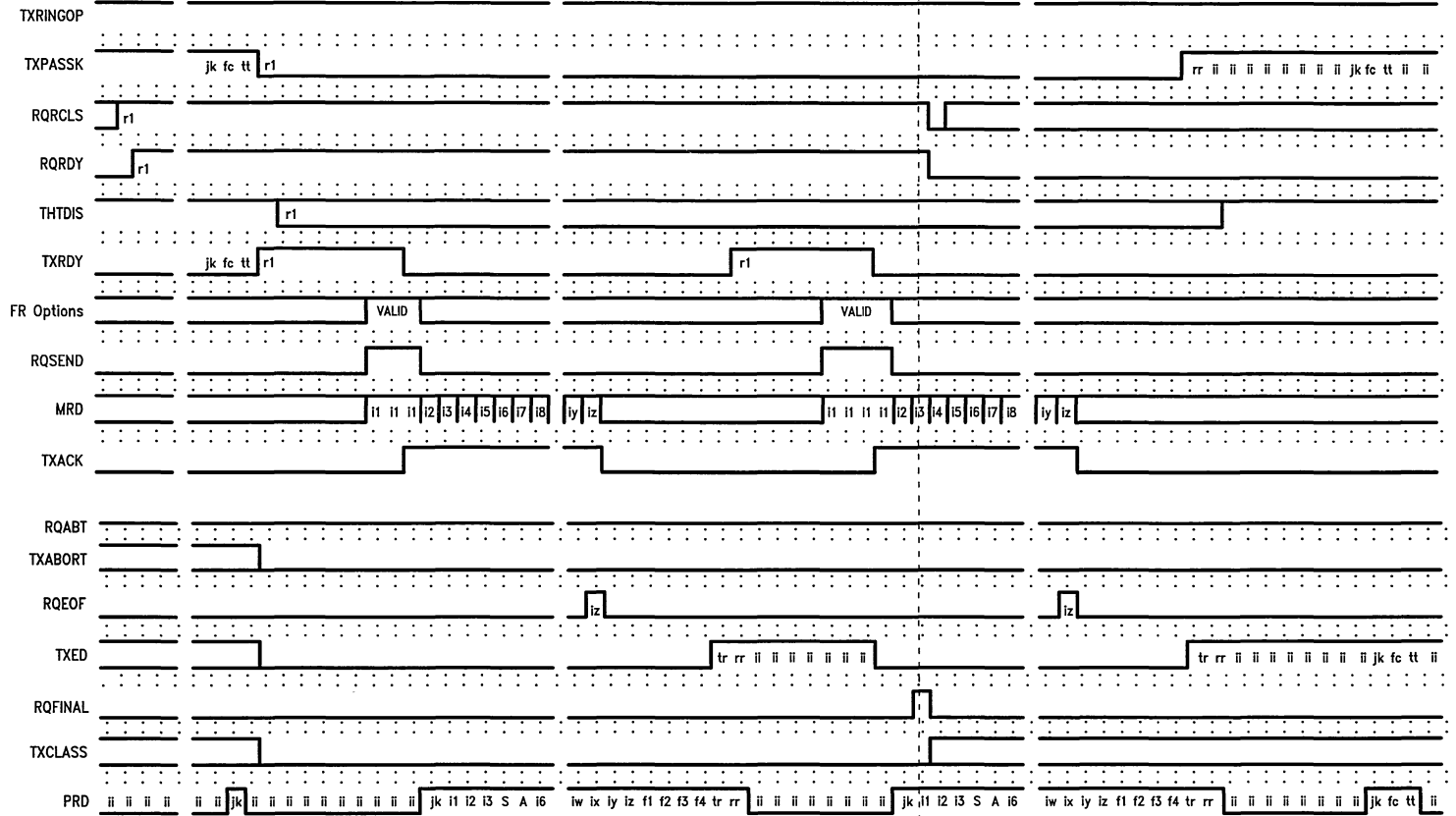


FIGURE 15. Asynchronous Request with Prestaging (normal completion after two frames)

TL/F/10825-13

REQUEST MACHINE

The Request Machine shown in *Figure 17* is a simplified version of a Transmit Sequencer. This could be used as the basis for many designs.

Note that all signals that begin with TX come from the BMAC Device Transmitter, and all signals that begin with RQ come from the PAL Sequencer (the Request Machine or Transmit Sequencer).

RQ0: IDLE

While in the Idle State, there are no frames to be sent and all of the status from previous frames has been reported. When a frame is to be sent, it is loaded into the temporary buffer. Once a frame is ready to be sent RQRCLS is loaded with the appropriate value. This requests a service opportunity as described by the 4-bit RQRCLS field in addition to the RQCLM and RQBCN which are used with Immediate transmissions. A transition to the Ready State then occurs.

RQ1: READY

RQRDY is asserted to indicate that the interface has or will soon have a frame that is ready to be sent. RQRDY causes the RQRCLS to be latched into the BMAC device and used while RQRDY is asserted. Once the token is captured a transition to the Send State occurs. Alternatively, enough data could be loaded into temporary storage to keep it not empty over transmission of the frame.

RQ2: SEND

Upon entry to the Send State RQSEND is asserted and the Frame Options are latched by the MAC interface. TXRDY is deasserted and TXACK asserted when the beginning of the frame is sent. A flag is set that causes us to remain in the Send State until the end of the frame as denoted by the

rising edge of TXPASS or TXRDY. While TXACK is asserted, data is transmitted (TXACK could be used as an enable for a byte counter). On the last byte of the frame, RQEOF is asserted and in response TXACK is deasserted. TXACK is used by the frame level handshake and RQEOF is returned from the frame level handshake. The Request Machine does not use TXACK or provide RQEOF.

The Last Frame and Last Request signals help control the sequencing for multiframe requests or multirequest service opportunities. Only between requests can the Requested Service Class change. The frame options may change on each frame.

RQ3: STATUS

After the frame is sent, status related to that frame is written or accumulated as appropriate either on the transmitted frame only or the returning frame as well. In short rings, this should be several byte times after the frame is transmitted, but in large rings there could be a several microsecond delay.

Conditions

Frame__Ready

- Frame__Ready is True when a frame is ready to be sent
- The Frame options (SAT, SAIGT, STRIP, FCST) must be valid at this time.

Instead of loading the entire frame into the temporary buffer before telling the MAC Interface, it would be possible to load in enough of the frame to guarantee that the temporary buffering will have enough data to keep up with the ring over the duration of the frame. An implementation like this could employ a FIFO or an equivalent speed matching device to match or smooth the bandwidth characteristics of the hardware feeding the temporary buffering.

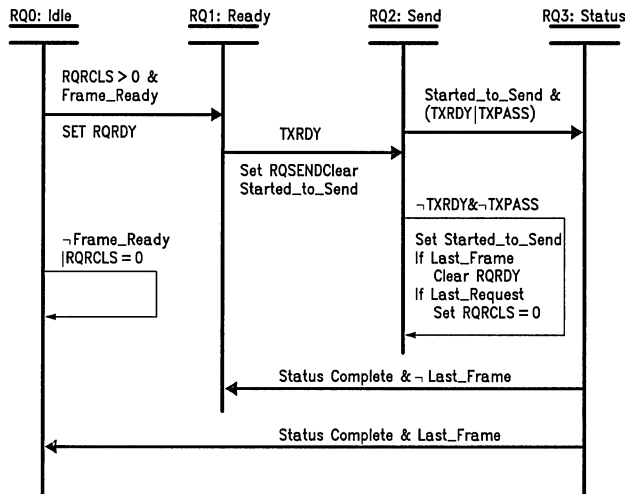


FIGURE 17. Example Transmit Sequencer

TL/F/10825-17

Notice that on the transition to the Ready State RQRCLS>0 is also used as a condition. This implies that the token could be captured before the data is actually ready. In other words, from the word go a race takes place between the data becoming ready in the temporary buffering and the MAC capturing a useable token.

Started__to__Send

- This is true any time after the MAC transmitter has started to send a frame.

Status__Complete

- This is true when status for the transmitted frame has been recorded.
- This may be a result of the transmitted frame, and optionally the returning frame.

Notice that in the case where status is written for every frame, additional frames may not be transmitted until after status is written. A further alternative approach would allow status on frames of a request to be stored in a temporary location and then only write status between requests. An extension to this idea would be to store temporary status separately for each individual request. This last approach would allow both of the Status Complete transitions to occur immediately. At some point however the interlock with the actual writing of status would have to occur. For example, even in the case where status is written into temporary storage for each request, a new request could not be started until status for the previous request has been written.

A pipelined multiple request machine could even take the status__complete and last__frame transition before the end of the frame and begin to load in the new parameters for the next frame. This would allow frames of the next Request to be transmitted very soon after the frames of the previous Request.

TRANSMISSION STATUS

Transmission status concerning the previous transmission is valid in all cases one cycle after TXRDY or TXPASS is asserted.

Transmission Completion

If an Ending Delimiter was transmitted TXED will be active one cycle after TXRDY or TXPASS is asserted until the starting delimiter of the next frame is transmitted. If the frame was aborted TXABORT is active one cycle after TXRDY or TXPASS is asserted.

The frame could have been aborted for several reasons:

- a MAC Reset
- the ring going non operational
- an internal BMAC device error
- an RQABORT during the frame

TRT Expiration

If TXPASS is asserted and THT was disabled during the last frame that was transmitted (THTDIS is asserted), TRT has expired. This is a serious error and indicates that there was an overallocation of synchronous bandwidth or a station used more than it was allocated. The ring will likely be in the CLAIM state when this occurs.

External CLAIM/BEACON Transmission

When transmitting CLAIM/BEACON frames from the CLAIM/BEACON state, if TXPASS is asserted the CLAIM/BEACON process is complete. In this case TXABORT indicates if this station won (TXABORT=0) or lost (TXABORT=1) the CLAIM/BEACON process.

Holding the Token

At end of the frame, TXRDY is asserted if possible (THT or TRT has not expired) or desirable (there are more frames to transmit on this service opportunity). Otherwise TXPASS is asserted.

Issue Token Class

If RQRCLS goes to zero at any time during a service opportunity a token will be issued.

TXCLASS indicates the class of token that was issued, in the case of TXPASS, or would have been issued in the case of TXRDY. Since the issue token class is given in the RQRCLS encoding, this is a sanity check, especially in a non-pipelined implementation.

For certain SMT frames (NSA frames), the returning status is of considerable interest. In these cases, it may be easier to just turn on copying of SMT frames based on MFLAG (i.e., AFLAG or MFLAG instead of the normal AFLAG and not MFLAG). The status can be the same as the reception status.

PROVIDING A TRANSMISSION CONFIRMATION SERVICE

Providing a Transmission Confirmation Service is one of the most challenging aspects of designing an interface to the BMAC device. Some useful hints for providing such a service are given below.

Overview

A transmission confirmation service returns ending status for a request to transmit one or more frames. As frames of a request are transmitted, the status on the returning frames is accumulated. When all of the frames of the request return around the ring, a positive status is given. Should anything other than the expected frames (with the correct FC, SA, FCS, and frame status values) return before all of the transmitted frames return, a negative status is generated. Status is thus generated when either all of the frames return, or an abnormal condition is detected.

The transmission confirmation service is similar to the (SM_)MA_UNITDATA_STATUS.indication (formerly called (SM_)MA_DATA.confirmation) defined in the ANSI FDDI MAC standard.

Ring Assumptions

- Returning frames can be associated with transmitted frames.

At most one request can be serviced (for each SAP) during a given token opportunity. Since the order of SAP serviced is fixed, the order for matching returning frames with their corresponding requests is also known. The returning FC, SA and FCS could be used to distinguish a frame transmitted by this station and match it with its corresponding SAP. Additional fields may be used to increase confidence in the association.

- Frames return in the order transmitted.

We assume that no frames are maliciously inserted into the ring.

- After a token is captured by a station for transmission, the next returning frames will be those transmitted by this station.

Since there is a single token, the next frame received by this station should be the (first) one it transmitted unless another station has failed to strip its frame(s).

- Synchronization points:

At these synchronization points, the number of frames transmitted and the number of frames received should be equal.

- Token received
- Frame with a different SA is received, after receipt of a frame with my SA.

Error Conditions

- Frame not recognized by addressed station

No station recognizes the Destination Address, and as a result the A indicator is not set. Either the station is not present or a line hit corrupted the Destination Address.

- Frame not copied by addressed station

The addressed station runs out of buffering resources and as a result does not set the C indicator. This is probably the most likely of the error conditions.

- Frame Error due to line hit

For example when a line hit causes an FCS error.

- Frame Lost due to line hit

For example the starting or ending delimiter is destroyed, or a data symbol becomes a non-data symbol. This may have different effects when the lost frame is:

- first frame of request
- a middle frame of a request
- the last frame of a request

- Frame not recognized by source station

A line hit corrupted the FC or the Source Address.

- Frame incorrectly recognized by source station

Frame insertion or misdirection can occur during reconfiguration, but CMT will remove such frames by scrubbing. An alias could be caused by a line hit on the FC or SA, but it is highly unlikely to contain the proper FCS value (this would require multiple errors or an unfortunate pattern on a transparent FCS transmission). Indiscriminate use of transparent FCS transmission could diminish the integrity of the confirmation service.

Error Handling

For each request, an expected status parameter is given either explicitly with the request or implicitly (it was given explicitly previously and has not changed). The expected status contains the expected value of the returning FCS, E, A and C indicators. If the expected value of these indicators is not returned, a negative confirmation is generated.

- Frame not recognized by addressed station

This condition can be handled using the expected status. The returning A indicator is not set, so the expected frame status is not returned. A negative confirmation is generated and the request is optionally terminated.

- Frame not copied by addressed station

This condition can be handled using the expected status. The returning C indicator is not set, so the expected frame status is not returned. A negative confirmation is generated and the request is optionally terminated.

- Frame Error due to line hit

This condition can be handled using the expected status. The returning E indicator is not reset or the FCS is invalid, so the expected frame status is not returned. A negative confirmation is generated and the request is optionally terminated.

- Frame Lost due to line hit

This type of error is more difficult to handle because it cannot be detected until the next synchronization point. At the next synchronization point the received and transmitted counts for the current request will not be equal. A negative confirmation is generated and the request is optionally terminated.

- Frame not recognized by source station

This type of error is more difficult to handle because it creates a false synchronization point. At this point the received and transmitted counts for the current request will not be equal. A negative confirmation is generated and the request is optionally terminated.

- Frame incorrectly recognized by source station

This condition will be detected unless the line hit caused a formerly invalid FCS to become valid. When detected, this condition may produce a false negative confirmation, but it will not produce a false positive confirmation.

5.0 Bridging/External Matching Support

There are two major considerations for MAC level Bridging products on FDDI:

1. How stripping will be accomplished and
2. How the control indicators will be handled.

The FDDI standard does not have a recommended Bridging algorithm like 802.3's Transparent Bridging and 802.5's Source Routing Bridging.

The issue with stripping is that every station needs to strip every frame it transmits. Typically this is accomplished by stripping based only on the Source Address. However, in bridging applications where the SA is not necessarily of this station, this station needs to either watch out for all of its frames (and use CAM technology to assist the strip decision) or use some other mechanism. Currently, the FDDI X3T9.5 Committee has agreed that the strip points of all frames is before the fourth byte of the INFO field. That implies that any fragment with more than three bytes of INFO is an illegal fragment.

The BMAC device provides an alternate stripping mechanism to accomplish the stripping by sending out two My_Void frames before the token and stripping everything until the first My_Void returns.

The definition of a bridgeable frame also has impact in bridging applications. Only LLC frames are bridgeable. Management frames (MAC and SMT frames) are not bridgeable. BEACON frames, CLAIM frames and SMT frames are local to a single ring. Likewise, synchronous and restricted dialogs are local to a single ring.

5.1 EXTERNAL MATCHING LOGIC

External Matching Logic is used to allow this station to recognize additional addresses. These additional addresses may be aliases (additional addresses that are not this station's management address) or additional group addresses not recognized by the BMAC device. In addition, more perfect filtering or routing could be done on certain classes of frames. Certain bridges and routers (not 802.1d transparent bridges since these address matches do not set the A_ind) can be viewed as having a set of aliases.

The result of the address comparison is then fed into the copy criteria and used to set the A indicator (the BMAC device EA input). The C indicator should be set as usual unless the frames recognized by the external matching logic are sent to a different buffer memory and generate a separate VCOPY signal.

There are two choices of where the external matching logic may tap into the data stream, either at the PHY Indicate Interface or at the MAC Indicate Interface. We recommend using the PHY Indicate Interface between the PLAYER device and BMAC device, because it provides a three byte time advantage over the alternative.

When using alias addressing it is also possible to strip frames on the basis of these additional aliased source addresses if the STRIP option is not used or desired. Three byte times after the EM signal is asserted, Idle symbols are transmitted into the ring and the frame is stripped.

5.2 SA, SAIG, FCS TRANSPARENCY OPTIONS

These transparency options are particularly useful in bridging applications. The SA transparency options are useful in certain Transparent Bridging algorithms and in Source Routing Protocols. The SA and SAIG transparency options allow the SA to be sent transparently from the data stream as opposed to being sent from the MAC Parameter RAM. The FCS transparency is particularly useful between bridged rings (end to end FCS checking with FDDI to FDDI bridges). The FCS transparency could also be useful for diagnostic purposes and with implementer defined FCSs on implementer frames.

5.3 IMPLEMENTING THE BRIDGING COMMITTEE RECOMMENDATIONS

The meaning of the control indicators also is impacted in bridging applications. The ANSI committee decided that a

bridge will only set the A__IND if the frame is explicitly addressed to the bridge and may optionally set the C__IND for all frames copied with the intent to be forwarded. If an end station receives a frame addressed to it with the A indicator reset, the C indicator set, and it does not copy the frame, the station sends it out with the A indicator set but the C indicator reset.

END STATIONS

For end stations and bridges which also act as end stations, the BMAC device always implements the option recommended by the Bridging Working Committee to X3T9.5. For frames with A not set and C set for which the station intends to Set the A indicator but not the C, the A indicator is transmitted as set and the C indicator is transmitted as Reset.

BRIDGES

For stations acting as bridges, the committee recommends that for frames copied with the intent to forward (for which a station address or alias match did not occur), only the C indicator should be set, not the A indicator.

To accomplish this, since the BMAC device will not set the C indicator without setting the A indicator as well, it is necessary to intercept the byte stream between the BMAC device and the PLAYER device. Fortunately, the difference between the R and S symbol is only a single bit. Thus, when a frame is copied with the intent to forward it, the received A indicator is not set, and the station's AFLAG is not set, the C indicator must be changed from an R to an S. This occurs one byte time after EDRCVD is asserted and requires that PRD(0) be changed from a 0 to a 1.

This easily fits into a small slice of a PAL and is only required in bridges implementing this option.



Table of Contents

- 1.0 INTRODUCTION
- 2.0 INITIALIZATION
- 3.0 SERVICING INTERRUPTS
- 4.0 MEMORY MANAGEMENT SCHEMES
- 5.0 SENDING FRAMES
- 6.0 RECEIVING FRAMES
- 7.0 QUEUE MANIPULATION

1.0 INTRODUCTION

This application note describes how to initialize the National Semiconductor BSI device (DP83265) and interact with it. Initialization and data service support occur through the Control Bus and via memory that is accessible by both the BSI device and the host. The host processor must be able to respond to interrupts and have access to both the BSI device Control Bus and some mutually accessible memory. This application note should be read in conjunction with the BSI datasheet.

2.0 INITIALIZATION

Before BSI device operation can begin, the device must be initialized. The BMACTM and PLAYERTM devices must also be individually initialized. To initialize the BSI device, the steps shown below should be followed. Each action is explained further in the subsections that follow.

- Put the BSI Device in Stop Mode
- Set the Mailbox Address Register
- Load the Pointer RAM
- Set the Event Notify Registers
- Set the Mode Register
- Set the Request Configuration Registers
- Set the Request Expected Frame Status Registers
- Set the Indicate Configuration Register
- Set the Indicate Mode Register
- Set the Indicate Threshold Register
- Set the Indicate Header Length Register
- Load the Limit RAM
- Clear the Attention Register
- Put the BSI Device in Run Mode

2.1 Put the BSI Device in Stop Mode

During initialization the BSI device must be in "Stop Mode". This is necessary to prevent the device from attempting to perform any actions or respond to any external stimulus prior to the completion of the initialization sequence. The BSI device may be placed in Stop Mode by setting three bits in the State Attention Register (STAR).

Since the State Attention Register (STAR) is a conditional write register, it must be read before it is written. By doing so the original contents of the STAR register are loaded into the Compare Register (CMP). When a subsequent write to the STAR register occurs, only those bits that match the

corresponding bits in the CMP register will be actually stored in STAR. With the BSI device in "Stop Mode", and no intervening accesses to the BSI device Control Bus, it is guaranteed that all 8 bits will match.

Finally, write 0x07 to the STAR. This clears all the error bits and sets all the stop bits (the STAR is automatically loaded with 0x07 upon reset of the BSI device).

2.2 Set the Mailbox Address Register

When loading Pointer RAM data into the BSI device, a "mailbox" mechanism is used. The mailbox is a 32-bit word in off-chip memory which the BSI device uses to load or dump the Pointer RAM Registers. This mailbox may be located anywhere within the 28-bit ABus address space of the BSI device and accordingly its address must be explicitly defined. This is accomplished via the 8-bit Mailbox Address Register (MBAR).

To load the Mailbox Address Register (MBAR) first load 0x00 into the Pointer RAM Control and Address Register (PCAR). This tells the BSI device to internally point to the first byte of the mailbox address. Then execute four successive writes to the MBAR to load the full mailbox address; writing the most significant bytes first. Each write automatically increments the byte pointer to the next byte.

When the BSI device is reset, the Mailbox Address Register (MBAR) is loaded with a hardware revision code. The host may obtain this revision code by sequentially reading four bytes from the MBAR before loading the mailbox address.

2.3 Load the Pointer RAM

The BSI device maintains pointer registers for accessing and manipulating the various queues. Prior to normal operation, some of these pointer registers must be initialized with queue addresses (see Table I). For active Request queues the host software must load the Confirmation Message (CNF) Queue Pointer Register and the Request (REQ) Queue Pointer Register. For Indicate queues the host software must load the Indicate Data Unit Descriptor (IDUD) Queue Pointer Register and the Pool Space (PSP) Queue Pointer Register. For information about choosing initial Pointer RAM values see Section 7 on Queue Manipulation.

TABLE I. Pointer Registers Used during Queue Initialization

Pointer Registers		Addr
CNF	Queue Pointer Register (RCHN1)	0x02
REQ	Queue Pointer Register (RCHN1)	0x03
CNF	Queue Pointer Register (RCHN0)	0x06
REQ	Queue Pointer Register (RCHN0)	0x07
IDUD	Queue Pointer Register (ICHN2)	0x09
PSP	Queue Pointer Register (ICHN2)	0x0a
IDUD	Queue Pointer Register (ICHN1)	0x0d
PSP	Queue Pointer Register (ICHN1)	0x0e
IDUD	Queue Pointer Register (ICHN0)	0x11
PSP	Queue Pointer Register (ICHN0)	0x12

Before loading a Pointer RAM Register, first read the Service Attention Register (SAR) to verify that the PTOp bit is

set; signifying that a previous Pointer RAM operation has completed. If this bit is not set wait for the previous operation to finish.

Write the value one wishes to store in the Pointer RAM Register (i.e., the base address of the relevant queue) in the memory location selected as the BSI device Mailbox.

Next configure the Pointer RAM Control and Address Register (PCAR) with the PTRW bit cleared and the address of the Pointer RAM Register placed in the least significant five bits. A zero value in the PTRW bit specifies that the next Pointer RAM operation will read from the Mailbox and write to the Pointer RAM Register. The two most significant bits in the PCAR (BP0, BP1) are not used in this context and may be loaded with 0's. For example, when loading the PSP Queue Pointer for Indicate Channel 0, one would write 0x12 to the PCAR.

Finally, clear the PTOPI bit in the Service Attention Register (SAR). The SAR is a conditional write register, so it is necessary to read it immediately before writing to it. Clearing the PTOPI bit causes the BSI device to perform the actual Pointer RAM operation. The device signals the completion of the operation by setting the PTOPI bit in the SAR.

The above steps must be done for all pointers associated with those channels that will be used.

2.4 Set the Event Notify Registers

You may specify which events will trigger an interrupt by setting the corresponding bit in the Notify Registers; where a 1 enables interrupts from that event and a 0 disables those interrupts. The Notify Registers may be written without being read previously (not conditional write registers).

See Section 3, Servicing Interrupts, for a more complete treatment of this subject.

2.5 Set the Mode Register

Load the BSI device Mode Register (MR) to configure the BSI device with global bus and queue parameters. For example a value of 0x52 causes the BSI device to generate 32 byte bursts when accessing the data bus, use 1k (small) queues, operate in a physical memory environment, use "big-endian" data alignment, check parity on access to the ABus and Control Bus and optimize operation for clock speeds over 12.5 MHz.

2.6 Set the Request Configuration Registers

Load the Request Configuration Registers (R0CR and R1CR) for both Request Channels (RCHN0 and RCHN1) to establish channel specific operating parameters; such as Source Address and Frame Control Transparency.

2.7 Set the Request Expected Frame Status Registers

Load the Request Expected Frame Status Registers (R0EFR and R1EFR) for both Request Channels (RCHN0 and RCHN1) to set up the expected status for frame confirmation services. A value of 0x00 in these registers means that any frame status is acceptable.

2.8 Set the Indicate Configuration Register

Load the Indicate Configuration Register (ICR) to establish copy control parameters for each Indicate Channel. A typical register value is 0x49; which instructs the BSI device to copy frames addressed for the owned MAC address or to an externally matched group address.

2.9 Set the Indicate Mode Register

Load the Indicate Mode Register (IMR) to set the frame sorting mode, skip option and the desired Indicate breakpoints.

Indicate breakpoints are instances that generate interrupts. You may configure the BSI device to interrupt at the end of each service opportunity, at the end of a burst (i.e., channel change) or after a user defined number of frames have been received. Prudent use of Indicate breakpoints can significantly reduce interrupt processing overhead by reducing the number of interrupts generated by the BSI device.

2.10 Set the Indicate Threshold Register

The Indicate Threshold Register (ITR) specifies how many frames must be received before a threshold breakpoint is realized. The value in this register is only used when the appropriate bits are set in the IMR.

Loading the ITR with 0x00 specifies a value of 256. This value is loaded into an internal working register each time the state of any Indicate Channels change.

2.11 Set the Indicate Header Length Register

If the Header/Info frame sorting mode is specified, one must load the Indicate Header Length Register (IHLR) with the length (in units of four byte words) of the header portion of the frame. The FC field occupies an entire word. For example, to separate an 8 octet header when using long, six-octet MAC addresses, one would load a value of 6 (FC = 1, DA/SA = 3, header = 2) into this register.

2.12 Load the Limit RAM

During normal operation of the BSI device, the CNF and IDUD queues must be given status space. This may be done as part of the initialization procedure. For information about choosing initial Limit RAM values see Section 7 on Queue Manipulation.

Before loading a Limit RAM Register, first read the Service Attention Register (SAR) to verify that the LMOP bit is set (signifying that the previous Limit RAM operation has completed). If this bit is not set wait for the previous operation to finish.

Next load the Limit Address Register (LAR). The top four bits of the LAR define the target Limit RAM Register, the LMRW bit specifies what the next Limit RAM operation will be (LMRW = 0 means a write to the Limit RAM) and the MSBD bit contains the most-significant data bit of the 9-bit Limit value.

Next load the Limit Data Register (LDR) with the lower 8 bits of the limit value.

Finally, write a 0 into the LMOP bit in the Service Attention Register (SAR). The SAR is a conditional write register, making it necessary to read it immediately before writing to it. Clearing the LMOP bit causes the BSI device to perform the actual Limit RAM operation. The BSI device signals the completion of the operation by setting the LMOP bit in the SAR.

Repeat the above steps for all desired limits.

2.13 Clear the Attention Registers

Clear the Request Attention (RAR) and Indicate Attention Registers (IAR) by first reading the register, to load the Compare Register (CMP), and then writing a 0x00 value to the register. Both of these registers are automatically initialized to 0 upon BSI device reset.

The No Space Attention Register (NSAR) should be initialized to reflect the state of space of all the queues. If space was given to all of the CNF and PSP queues, read and write 0x00 into NSAR.

2.14 Put the BSI Device in Run Mode

Initialization of the BSI device is now complete. The device may be made fully operational by reading the State Attention Register (STAR) and immediately writing 0x00 to it. This will clear the stop bits for the Indicate, Request and Status/Space machines; putting them in "Run Mode".

The BSI device should immediately begin fetching PSP Descriptors for the Indicate Channels to use for frame reception. At this point a write to one of the REQ queue Limit RAM Registers would cause the BSI device to begin fetching REQ Queue Descriptors for frame transmission.

3.0 SERVICING INTERRUPTS

The BSI device provides facilities for selecting which events will generate an interrupt and a mechanism for determining which events are present after an interrupt has been raised.

3.1 Event Registers

The BSI device supports a two-level hierarchy of Event Registers; where the presence of attention signals in lower level attention registers is recorded in a single upper level attention register. Attention signals may be disabled at either of the two levels. Events may only be cleared by resetting the attention bits in the lower level registers.

The upper level attention register is called the Master Attention Register (MAR). It contains five attention bits that indicate the presence or absence of any events recorded in each of the five corresponding attention lower level registers. Those registers are listed in Table II.

TABLE II. Attention Registers

Master Attention Register (MAR)
State Attention Register (STAR)
No Space Attention Register (NSAR)
Service Attention Register (SAR)
Request Attention Register (RAR)
Indicate Attention Register (IAR)

The host may control which attention bits will generate an interrupt by configuring the Notify Registers (see Table III).

TABLE III. Notify Registers

Master Notify Register (MNR)
State Notify Register (STNR)
No Space Notify Register (NSNR)
Service Notify Register (SNR)
Request Notify Register (RNR)
Indicate Notify Register (INR)

For each Attention Register a corresponding Notify Register exists. Each Attention Register is ANDed with its corresponding Notify Register and then all of the resulting signals are ORed together and presented to the next level (see Figure 1).

For example, to disable all interrupts caused by service events: **clear** the Service Attention Register Notify (SVAN) bit in the Master Notify Register (MNR). To disable only interrupts caused by Pointer RAM Operations: **set** the SVAN bit in the MNR and **clear** the PTOPN bit in the Service Notify Register (SNR).

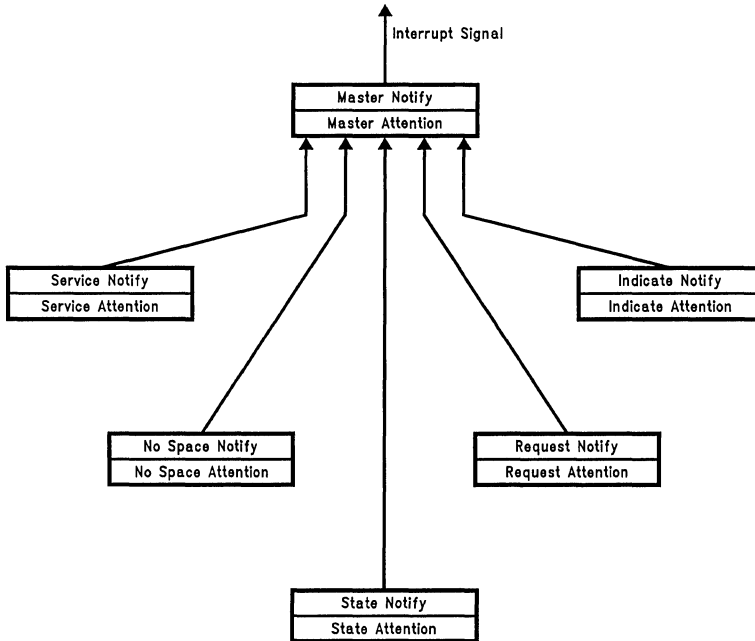


FIGURE 1. BSI Device Event/Notify Registers

TL/F/11088-1

When checking attention registers for the cause of an interrupt, one should perform a bit-wise AND operation between the attention and notify registers and examine the result. Just checking the attention registers may be misleading. For example, to disable an Indicate Channel one may wish to leave its PSP queue empty and mask off the "Low Data Space" attention bit for that channel; via the Indicate Notify Register (INR). Under these circumstances the IAR, by itself, may contain misleading information.

3.2 Example Procedure

A typical procedure for servicing BSI device interrupts is as follows:

- disable host interrupts
- determine the event that triggered the interrupt by checking the Master Attention Register and then querying the appropriate lower level attention register
- process the event (or post the event to a service queue)
- clear the attention bit (or mask the attention bit)
- enable host interrupts

4.0 MEMORY MANAGEMENT SCHEMES

The BSI device may be configured to use memory shared between itself and the host or it may be configured to use the host's memory. In addition, it can be made to operate in a virtual memory environment.

Although the BSI device manages space for incoming data (from channel specific Pool Space (PSP) queues); the host must implement a memory management mechanism to replenish the PSP queues and manage the space needed to hold output data (ODU) and ODU Descriptors (ODUDs).

4.1 Memory Requirements

Up to ten distinct queues may be established; two for each channel. Depending upon the value of the Small Queue (SMLQ) bit in the Mode Register (MR), these queues will each consume 1k or 4k of memory; collectively occupying either 10k or 40k of memory.

A 4 byte word must be allocated as the BSI device Mailbox. This word is only used when accessing the BSI device Pointer RAM Registers.

Space must also be allocated for buffering frames. Any buffers drawn from this space must be no larger than 4 kbytes and may not cross a 4 kbyte boundary. At the current time the Count (CNT) field in the PSP Descriptor is ignored by the BSI device. Pool space is intended to be allocated in 4 kbyte pages.

Space must be allocated for buffering ODU Descriptors (ODUDs). As with frame data, buffers drawn from this space must be no larger than 4 kbytes and may not cross a 4 kbyte boundary.

4.2 Dedicated Buffer Pools

To simplify memory management, buffer pages may be dedicated to individual PSP queues. When using dedicated buffers, the Indicate buffer management task becomes a matter of:

- detecting page boundary crosses; as an indication that the BSI device has finished filling the previous page
- obtaining confirmation that the host has finished processing all frames in the previous page

- attaching the previous page to a PSP Descriptor of the same queue
- incrementing the PSP Limit Register for that queue

The management of space for ODUs (outgoing frame data) and ODU Descriptors (ODUDs) must be done by the host.

A fully allocated 1k PSP queue consumes 512 kbytes of buffer space. A fully allocated 4k PSP queue uses 2 MB of buffer space.

4.3 Shared Buffer Pool

To maximize memory utilization, multiple Indicate Channels may share a single pool of data buffers. This does **not** mean that Indicate Channels can be made to share a Pool Space (PSP) queue, but rather that data buffers attached to the various PSP queues are allocated and freed from a global buffer pool on an "as needed" basis. When using a shared buffer pool, the Indicate buffer management becomes the following:

- Detecting page boundary crosses; to determine when the BSI device is finished filling the previous page
- Obtaining confirmation that the host has finished processing all frames in that page
- Returning that page to a shared buffer pool or, upon determining that the page has been dedicated to a given channel, reattach the page to the channel's PSP queue
- Responding to interrupts caused by a "Low Space" condition by allocating buffer space to one or more PSP Descriptors and incrementing the PSP Limit Register for that queue

Again, the management of space for ODUs (outgoing frame data) and ODU Descriptors (ODUDs) must be done by the host.

One danger with sharing pool space is that a heavily used low priority channel may starve a high priority channel by consuming all of the buffer space. This is contrary to the idea of priority. It is recommended that some mechanism be implemented for reserving memory for a given channel and that at least four buffer pages be dedicated to Indicate Channel 0 (ICHNO). This is to ensure that FDDI-SMT frames will not be dropped when there is a great deal of activity on the other Indicate Channels.

5.0 SENDING FRAMES

This section describes how to use the BSI device to queue a frame for transmission on an FDDI ring. It is assumed that the BSI device has been initialized and that the Request Configuration Registers (R0CR and R1CR) and the Request Expected Frame Status Registers (R0EFRS and R1EFRS) have been previously loaded with the desired values.

The mechanism for sending a frame is as follows:

- Obtain space for data structures
- Load the ODU(s)
- Process previous CNFs (optional)
- Build the ODUD(s)
- Build the REQ
- Signal the BSI device

Subsection 5.7 describes some special considerations for sending multiple frames in a single request object.

5.1 Obtain Space for Data Structures

BSI device addressable memory must be obtained to:

- hold the frame data
- hold the ODU Descriptor(s)
- hold the Request Descriptor

It is the responsibility of the host software to manage space for frame data and the ODUDs. Section 4, Memory Management Schemes, describes two simple memory allocation methods. If multiple ODUDs are required, these must be contiguously allocated in the form of an array of ODUDs.

Space for the Request Descriptor must be located within the area designated as the Request queue (see Section 2.3). It must also be allocated in a serially contiguous fashion; immediately following the previously allocated descriptor. All BSI device queue pointers "wrap" to the first location upon reaching the end of the queue area.

The frame data may need to be divided between multiple ODUs. Each ODU may start anywhere within a 4k page, but it must end at or before the next 4k boundary. Multiple ODUs must be generated for frames over 4k in length.

For each ODU, space must be allocated for a corresponding ODU Descriptor (ODUD). System configurations in which the BSI device directly addresses host memory, may be able to contrive ODU Descriptors (ODUDs) that refer directly to host specific memory buffers.

5.2 Process Confirmation Status Message Descriptors (CNF)—Optional

When the BSI device processes a request it places confirmation messages in the CNF queue for that Request Channel. By examining these messages, the host may determine when the BSI device has finished using ODU, ODUD and REQ queue space.

If the BSI device has been instructed to generate interrupts after writing confirmation messages, then an autonomous interrupt handler should be available to asynchronously process CNFs. Conversely, if these interrupts have been disabled, then CNFs should be processed when attempting to send a frame.

5.3 Copy Frame Data to Buffer

If the ODU buffers are distinct from the host specific memory buffers, copy the frame data from the host buffer(s) to the ODU buffer(s).

5.4 Build the ODU Descriptor(s)

An ODU Descriptor (ODUD) must be written for each ODU. The address and size of the ODU must be recorded in the ODUD.LOC and ODUD.CNT fields, respectively.

When only a single ODUD is needed both the First and Last bits should be set (only). With multiple ODUDs the first ODUD should have the just First bit set (first) and the last ODUD should have the just Last bit set (last). Any intervening ODUDs should have both bits cleared (middle).

5.5 Build the Request Descriptor

A Request Descriptor must be constructed which references the ODUD(s) that were just built.

The User Identification (UID) field may be assigned a host defined value. This UID value will reemerge in one or more CNFs and may be useful when processing a CNF (i.e., deallocating ODUD and buffer space). The SIZE field should

be set to a value of 1; since, in this case, only a single frame will be transmitted.

The Confirmation Class (CNFCLS) field defines the level of request confirmation that the BSI device will use and should be set as needed. To turn off request confirmation put a hex value of 0x4 in the CNFCLS field; although, the BSI device will **always** generate CNF Descriptors whenever an exception is encountered. Please note that request processing will halt for a given channel should that channel's CNF queue become full. Thus, a provision for processing CNF Descriptors must be included in all applications; even those applications that do not wish to receive confirmation for most requests.

The Request Class (RQCLS) field defines the class of the request (i.e., asynchronous, synchronous, restricted token, etc.). The FC field should be loaded with an appropriate FDDI Frame Control value.

When sending a single frame, both bits in the First and Last bits should be set; indicating that this is the only REQ Descriptor in this request object. Finally, the address of the first ODUD should be put into the LOC field.

5.6 Signal the BSI Device about the Request

The BSI device may be caused to examine either of its REQ queues by writing to the corresponding Limit RAM Register with a value that raises the limit to reference the new REQ Descriptor.

5.7 Sending Multiple Frames in a Single Request Object

The BSI device is capable of transmitting multiple frames in a single service opportunity. This feature becomes important on a heavily loaded FDDI ring, with relatively infrequent service opportunities. The BSI device can be caused to process multiple frames by:

- building an ODUD list that contains multiple frames
- building a request object that contains multiple REQ Descriptors or
- a combination of the above two methods.

The first method is extremely simple. Allocate and fill the ODU buffers for all of the frames. Build multiple ODU Descriptor objects (demarcated by the First and Last bits in each ODUD) and concatenate the ODUDs together into one array of descriptors. Build the REQ Descriptor, as before, except load the frame count into the SIZE field.

The second method consists of a creating a REQ Descriptor marked as being "first", zero or more REQ Descriptors marked as "middle" descriptors and an ending REQ Descriptor marked as being "last". The Limit RAM Register, for the given request queue, must be set beyond the last REQ Descriptor. The parameter fields in first REQ Descriptor are used for the entire request object.

5.8 Batching Single Frame Requests

On a heavily loaded FDDI ring service opportunities occur less frequently than on an FDDI ring with only light traffic. On a loaded network it makes sense to send multiple frames per service opportunity. However, many network communication systems send only a single frame at a time. This subsection tells how one may use the capabilities of the BSI device to batch single frame requests into a larger request object.

The BSI device will only attempt to send a single request object in any given service opportunity. A request object is defined here to consist of one or more REQ Descriptors delimited using the First and Last bits found inside each

descriptor. The BSI device interface software needs to build different types of REQ Descriptors when queuing a frame such that:

- A single frame request object is generated when the queue is empty
- The resulting request object is limited to a maximum size
- Optionally the resulting request object is closed whenever a service opportunity is detected.

The following pseudo-code may be used to satisfy the above requirements.

Transmit Logic

```

Req_Size = Req_Size + 1
if Queued_Cnt = 0
  mark as REQ.ONLY
  Open_Req = FALSE
  Queued_Cnt = 1
  REQ_Size = 0
else
  if Open_Req = FALSE
    mark as REQ.FIRST
    Open_Req = TRUE
    Queued_Cnt = Queued_Cnt + 1
    Req_Size = 1
  else
    if Req_Size ≥ Max_Req
      mark as REQ.LAST
      Open_Req = FALSE
      Req_Size = 0
    else
      mark as REQ.MIDDLE
    endif
  endif
endif
endif

```

Interrupt Service Routine Logic (Optional)

```

if Open_Req = TRUE
  generate empty REQ.LAST
  Open_Req = FALSE
  Req_Size = 0
endif

```

Queue_Cnt is the number of queued request objects on the request queue and is set to 0 queue initialization time. **Req_Size** is the number of frames in the currently open request object and is set to 0 at queue initialization time. **Open_Req** is a boolean variable indicating the presence or absence of an open request object and is set to FALSE at queue initialization time. **Max_Req** is a maximum number of frames per request object defined by the host software. Please note that the BSI device will **not** hold the token unnecessarily when processing an open request object. It will only hold the token when explicitly instructed to do so, via the RQCLS field in the Request Descriptor (REQ).

6.0 RECEIVING FRAMES

This section describes how to process incoming frames, using the BSI device's data structures. It is assumed that the Indicate Mode Register (IMR), Indicate Threshold Register (ITR), Indicate Configuration Register (ICR) and Indicate Header Length Register (IHLR) have been previously configured. It is also assumed that the host has already selected a particular Indicate Channel for processing; perhaps by examining the attention register hierarchy.

The host must maintain a minimum of two variables depicting the state of the Indicate machine: current IDUD queue pointer and the current buffer page address. To reduce accesses to the BSI device Control Bus, the host software may wish to also keep its own copy of the channel's Limit RAM Register.

For a visual description of the Indicate machine, see *Figures 2, 3, and 4.*

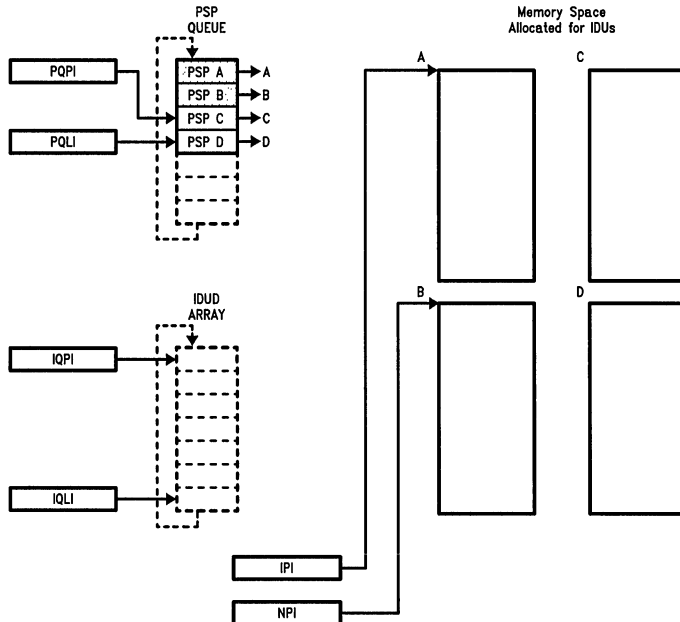


FIGURE 2. BSI Device Indicate Memory Structure

TL/F/11088-2

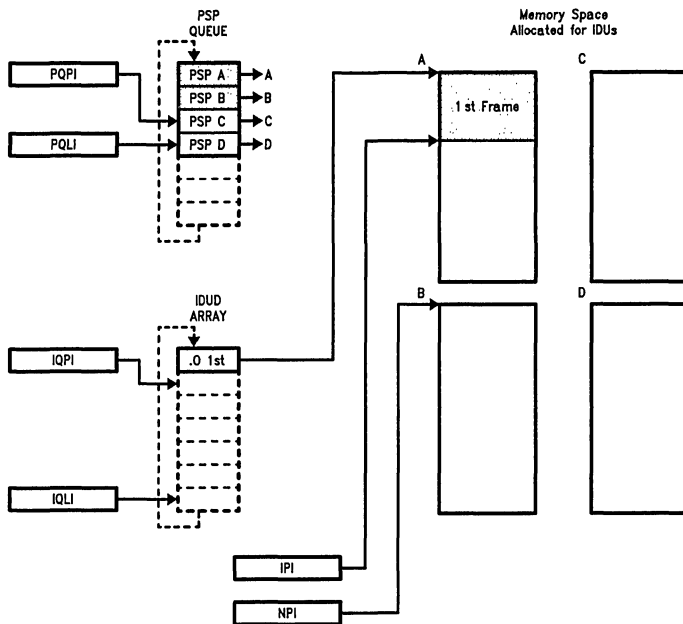


FIGURE 3. BSI Device Indicate Memory Structure

TL/F/11088-3

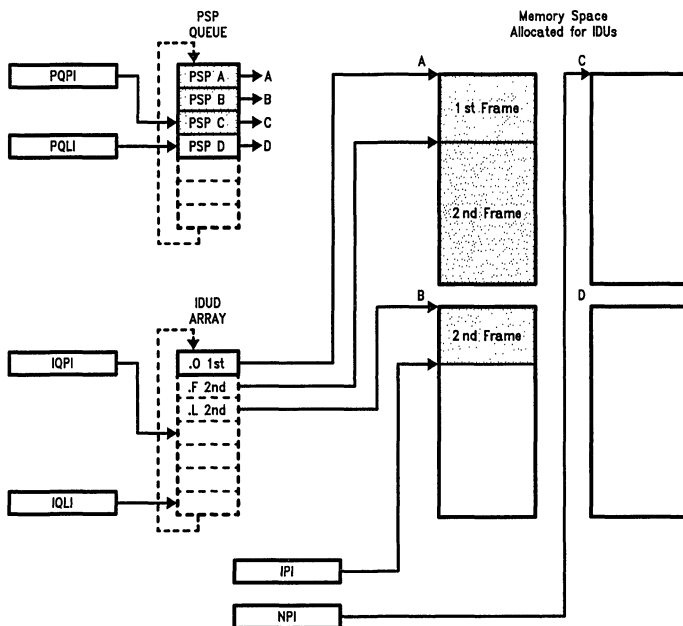


FIGURE 4. BSI Device Indicate Memory Structure

TL/F/11088-4

Figures 2 through 4 describe the operation of an Indicate Channel when receiving two frames.

Key for Figures 2-4

- | | | |
|--------------------------|------------------------|-----------|
| PQPI —PSP Queue Pointer | IQLI —IDUD Queue Limit | .O —Only |
| PQLI —PSP Queue Limit | IPI —IDU Pointer | .F —First |
| IQPI —IDUD Queue Pointer | NPI —Next PSP Pointer | .L —Last |

6.1 Disable Interrupts for the Indicate Channel

Host manipulation of BSI device queues must be atomic; meaning, in this context, that only a single host agent (process, task, thread, etc.) may actively dequeue frames from a given Indicate Channel at a given time. In support of atomicity, four different granularities of interrupt masking may be achieved: host level, BSI device level, Indicate service level or Indicate Channel level. To mask interrupts at the Indicate Channel level, modify the Indicate Notify Register (INR) to clear the breakpoint and exception bits for the target channel.

6.2 Collect an Indicate Object

Indicate objects may be represented by one or more IDU Descriptors (IDUDs) on the given channel's IDUD queue. The host must maintain a queue pointer for the next queue position. When collecting an Indicate object the host should scan forward from this position until an entire Indicate object has been found. If a null descriptor is found in the first position, there are no Indicate objects on the queue. The beginning of an Indicate object is marked by an IDUD with the "First" bit set and, conversely, the end is marked by an IDUD with the "Last" bit set. An IDUD with both of these bits set ("Only") completely describes an incoming frame. Note that it is the responsibility of the host to nullify descriptors after processing the frame.

6.3 Determine Acceptability of the Frame

There are three fields defined in the IDU Descriptor (IDUD) that are of use in determining the acceptability of a given frame. These fields are valid in the last IDUD in the Indicate object.

The Frame Status field may be examined to determine validity of the data length and FDDI FCS fields; as well as the values of the E, A, and C Indicators in the FDDI Frame Status field.

The Frame Attribute field can be queried to determine how the frame was recognized (MFLAG, AFLAG) and what the terminating condition was.

The Indicate Status field contains encoded status information (See Table IV).

TABLE IV. Indicate Status Codes

Code	Status
0x0	Last IDUD of Queue, Page Cross
0x1	Page Cross
0x2	Header End
0x3	Page Cross and Header End
0x4	Intermediate Frame
0x5	Burst Boundary
0x6	Threshold
0x7	Service Opportunity
0x8	Insufficient Data Space
0x9	Insufficient Header Space
0xa	Successful Header Copy, No Info Copy
0xb	No Info Space
0xc	FIFO Overrun
0xd	Bad Frame
0xe	Parity Error
0xf	Internal Error

6.4 Process the Frame Data

The Location (IDUD.LOC) and Byte Count (IDUD.CNT) fields in each IDUD describe the address and length of each Indicate Data Unit (IDU). There may be multiple IDUDs in a given Indicate object. The frame data referenced by these IDUDs must be logically concatenated to construct a single frame. The method of presenting frame data to upper level software is highly host dependent.

6.5 Reclaim Data Buffer Space

A 4 kbyte data page is available for reuse when the BSI device has filled it with IDUs and the host has finished processing all the IDUs on the page. The BSI device is guaranteed to consume space on a channel's PSP queue in a serial manner; thus it is possible to tell when the BSI device has finished using a page by detecting when the device starts filling a new page. When the host is also done processing all of the IDUs on the given page that data space may be reused. See Section 4 on Memory Management Schemes for ideas on managing buffer space for the BSI device.

6.6 Update IDUD Queue Pointers

After extracting all of the needed data from an Indicate Channel, that channel's IDUD queue may be updated to allow reuse of queue space. Three operations should be performed:

- Mark the processed IDU Descriptors (IDUDs) as null descriptors. A safe method is overlaying the eight byte queue slot with binary zeros.
- Update the host resident current IDUD queue pointer to point beyond the processed IDUDs.
- Update the Limit RAM Register for the given channel. If this value is maintained by the host, the value may be incremented and stored in the channel's Limit RAM Register; otherwise it is necessary to read the register first. The Limit RAM Operation (LMOP) is described above in Section 2.12.

6.7 Enable Interrupts for the Indicate Channel

Now that the given channel's queues have been updated, interrupts may be enabled again. If interrupts were masked at the Indicate Channel level, the Indicate Notify Register (INR) should be modified to set the breakpoint and exception bits for the target channel.

7.0 QUEUE MANIPULATION

The BSI device has two basic classes of queues: those that it uses to consume descriptors (REQ, PSP) and those that it uses to produce descriptors (IDUD, CNF). Conversely, the host software must consume descriptors (IDUD, CNF) and produce descriptors (REQ, PSP) on the opposite queues.

For Request Channel operation the BSI device **reads** from the channel's Request Descriptor (REQ) queue and **writes** to the channel's Confirmation Message (CNF) queue. For Indicate Channel operation the BSI device **reads** from the channel's Pool Space (PSP) queue and **writes** to the channel's Indicate Data Unit Descriptor (IDUD) queue.

It is necessary to understand how the BSI device interacts with each type of queue so that one may design host software that interacts with the queues in a complementary fashion. When writing software that interfaces with the BSI device, one minimally needs to understand the following:

- How the queues are organized
- How to initialize each type of queue

- How to handle queue “wraps”
- How to detect boundary conditions (empty queue, full queue)

7.1 Queue Organization

A BSI device queue consists of a contiguous block of BSI device addressable memory logically sub-divided into eight byte queue slots. Queues sized at 1 kbytes must be aligned on 1 kbyte boundaries, while queues sized at 4 kbytes must be aligned on 4 kbytes boundaries.

The BSI device embodies two indexing variables for each queue: a pointer to the **next** available queue position to read or write (stored in BSI device Pointer RAM) and a queue limit (stored in BSI device Limit RAM). The BSI device increments the pointer variable after reading from a queue or writing to a queue. The host software may control channel operation by manipulating the value of the limit variable. With this in mind there are a few simple rules governing queue manipulation by the BSI device.

1. Pointer and limit variables reference a given queue slot by pointing to the first word of the descriptor.
2. The pointer variable always points to the **next** available position on the queue.
3. The BSI device always increments the pointer variable **after** a read or write operation.
4. The BSI device halts channel processing when the pointer and limit variables are logically **equal**.
5. The BSI device tests for pointer/limit equality **during** queue read operations (REQ, PSP) and **after** queue write operations (IDUD, CNF). When detecting pointer/limit equality during a read operation, the current read operation and no further read operations are made.
6. Read operations are triggered by Limit RAM updates.

There are also some special considerations caused by the pipelined processing of CNF, IDUD and PSP Descriptors.

- The BSI device may generate two additional CNF/IDUD Descriptors after detecting pointer/limit equality. It is necessary to set the limit value such that it references the penultimate available queue slot.

- Each active Indicate Channel should have at least two buffers placed on its PSP queue. With only one buffer, the BSI device will immediately raise the Low Data Space attention bit for that channel.

The host software must maintain its own pointer and limit variables for each queue. For REQ and PSP queues, the limit variable should reference the next available queue slot for writing a new descriptor; while the pointer variable should correspond to the pointer variable on BSI device. For CNF and IDUD queues, the software pointer should reference the next queue slot to be used when reading a new descriptor. The software limit variable must always reflect the limit variable on the BSI device. The software pointer variable must be maintained independently from the BSI device queue pointer.

7.2 Queue Initialization

To initialize queues that the BSI device reads (REQ, PSP) simply set the pointer (BSI device and software versions) and software limit to reference the first queue slot. Do not update the queue’s Limit RAM Register until actually queuing the queue’s first descriptor.

To initialize queues that the BSI device writes (IDUD, CNF) one must set the limit variable to reference the penultimate available queue slot (required due to pipelining). For example, to make all but one of the queue slots available one could set the pointer variable to reference the first queue slot and the limit variable to reference the “next to the next to the last” queue slot. Also, one should clear the queue area by overwriting it with binary zeroes; effectively marking all queue slots as “null descriptors”.

See Figure 5 for a pictorial description of initialized queues.

7.3 Queue “Wraps”

Upon reaching the end of the queue memory block, queue indexing variables “wrap” to the beginning of the queue memory block. The BSI device automatically performs queue “wraps” for pointer variables; while the host software must perform queue “wraps” for limit variables and software queue pointers. The method for calculating the next

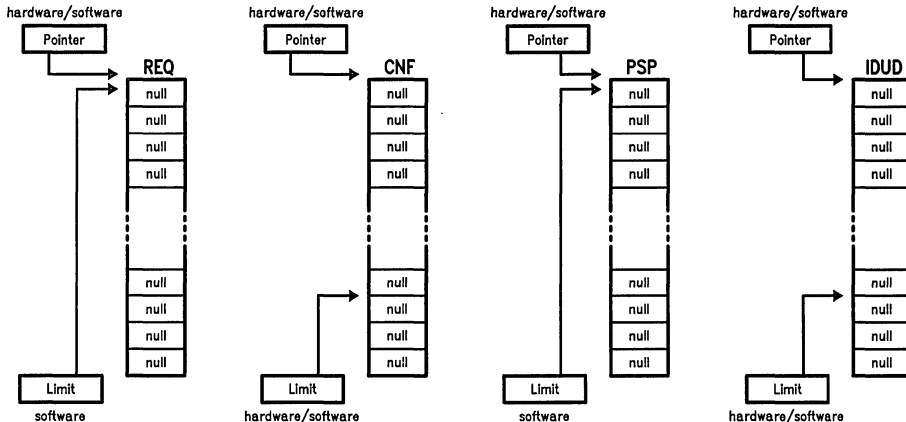


FIGURE 5. Suggested Queue Initialization

TL/F/11088-5

queue position is dependent upon the form of data representation that the host software uses (i.e., BSI device addressable pointers, queue byte offset, . . .). For example, when representing the limit variable as a queue offset one could use simple modulo arithmetic. When the limit variable is maintained as a pointer into BSI device addressable memory the host software might use the following method to increment the variable (specified with C code using 4 kbyte queues).

```
new = ((old + 8) & 0xff) + (old & 0xffff000)
```

7.4 Detecting Boundary Conditions

The BSI device detects both queue empty (when reading) and queue full conditions (when writing) by testing for pointer/limit variable equality. As noted above, this test is done during read operations and after write operations.

When reading, a queue empty condition **cannot** be determined by comparing the pointer and limit variables. Instead the host software may recognize the presence of a "null descriptor" on the queue. To ensure that there will always be at least one "null descriptor" to demarcate the queue empty condition; the host software must never set the limit variable to indicate that all of the queue slots are available and must mark each available queue slot as a "null descriptor" (binary zeroes are recommended).

The host software can detect a queue full condition using the same basic mechanism as the BSI device. When writing, a queue may be considered full when the software limit pointer references the queue slot immediately "before" the slot referenced by the software pointer variable. Queue "wraps" must be taken into account in the queue variable arithmetic.

See *Figure 6* for a graphical representation of the various queue boundary conditions.

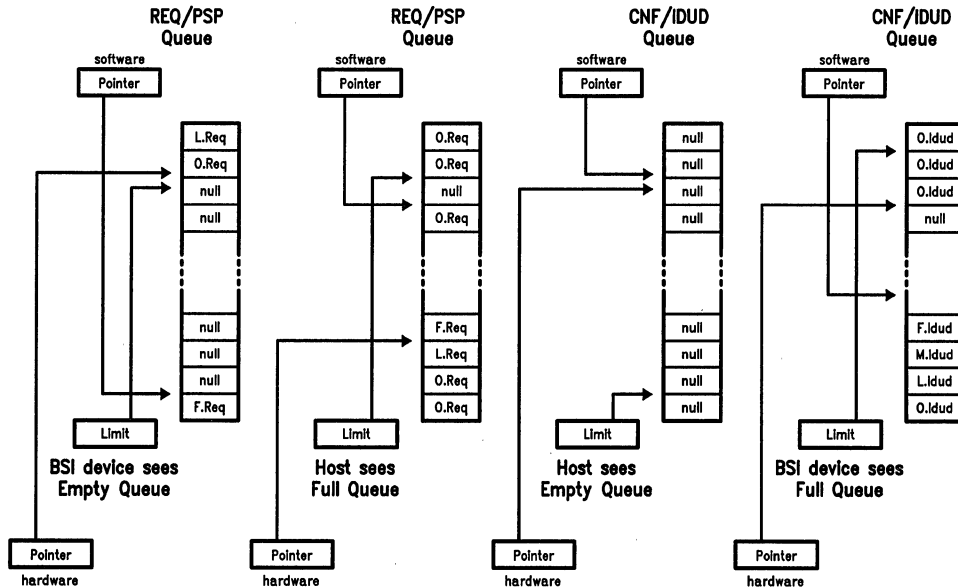


FIGURE 6. Queue Boundary Conditions

TL/F/11088-6

Layout Recommendations for a System Using National's FDDI Chip Set

National Semiconductor
Application Note 674
Bruce Wolfson



This application note covers basic PCB layout recommendations and design techniques for high speed signal distribution in National's FDDI system. Due to the high signal speeds in FDDI, proper layout is critical. Many digital designers are not aware of problems that can arise in a high speed system from improper routing, incorrect termination, and poor power and ground layout.

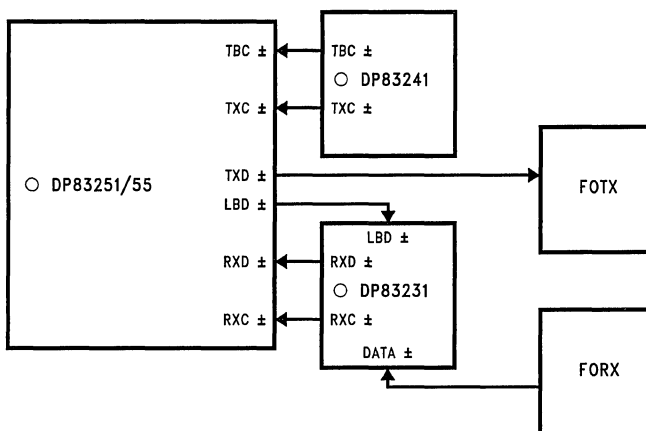
ROUTING

Line reflections are a reflection of the signal waveform in a transmission line. They are caused by a difference in impedance between the transmission line and the load at the end of the line. If the line propagation delay is small compared to the rise time of the signal, the reflection is hidden during the rise time and is not seen as overshoot or ringing. However with the fast edge rates of the signals in a FDDI system, the line length becomes critical. The distortion that results from reflections can give false triggering or data errors. The signals most susceptible to reflections in a FDDI system are

the differential ECL signals. In National's FDDI chip set these include the following:

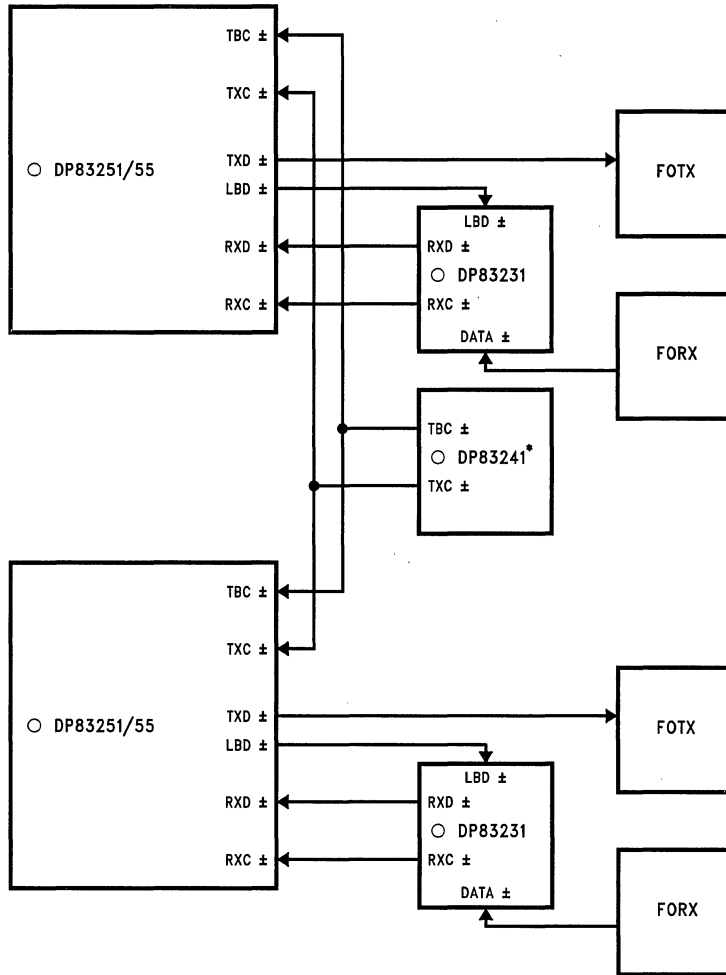
- 125 MHz and 12.5 MHz ECL clocks from the DP83241 (CDD™ device)
- Receive Data and Receive Clock from the DP83231 (CRD™ device)
- Data to and from the fiber optic transceiver, and
- Loopback data from the DP83251/55 (PLAYER™ device)

It is imperative that these signal lines be kept as short as possible. To achieve this, the DP83241 should be placed close to the DP83251/55, the DP83231 should be placed close to the fiber optic receiver and the DP83251/55, and the fiber optic transmitter should be placed close to the DP83251/55 (Figures 1 and 2). The pinout of the PLAYER device determines the placement of the CDD and CRD devices. All the ECL signals that travel between these devices line up perfectly. To keep the ECL signals traveling between these three devices as short as possible, the CDD and CRD devices should be oriented with pin 1 facing the



TL/F/10790-1

FIGURE 1. Recommended Component Placement for a Single Attach Station (SAS)



*See DP83241 Datasheet for driving multiple DP83251/55 devices.

TL/F/10790-2

FIGURE 2. Recommended Component Placement for a Dual Attach Station (DAS)

PLAYER device. In addition, the PLAYER device must have its ECL signals facing toward the fiber optic transceiver.

A Single Attach Station (SAS) in an AT form factor was chosen to demonstrate recommended chip placement and signal routing. Figures 3 through 7 show the basic connections between National's FDDI chips in a Single Attach Station. The silkscreen showing chip and component placement is shown in Figure 8 and the actual signal traces are shown in Figure 9. The FOTX and FORX placement is set by the PMD specification. The CDD, CRD and PLAYER devices should be placed as described in the previous paragraph. However, due to the constraints of the AT form factor, it is necessary to rotate the CDD device and its external components 90 degrees counter clockwise. The only remaining chip to be placed was the DP83261, BMAC™ device. Once again the form factor determined the placement,

but the orientation was chosen to allow an easy connection to the system interface, the control logic and the PLAYER device. Since the system interface logic is on the end of the board opposite the fiber optic transceiver, it was logical to have these signals facing that end of the board. It also allowed the control signals of the BMAC device to be near the control signals of the PLAYER device. All the TTL signals in this design can be autorouted. However, none of these signals should pass through the CDD or CRD device circuitry areas to avoid the possibility of noise due to crosstalk. In a Dual Attach Station (DAS) or a Concentrator design, the CRD device should still be placed as close as possible to the PLAYER device but the one CDD device responsible for clocking all the PLAYER and BMAC devices should be placed so as to minimize the skews between each PHY layer. More information regarding the CDD device driving multiple PLAYER devices can be found in the DP83241 datasheet.

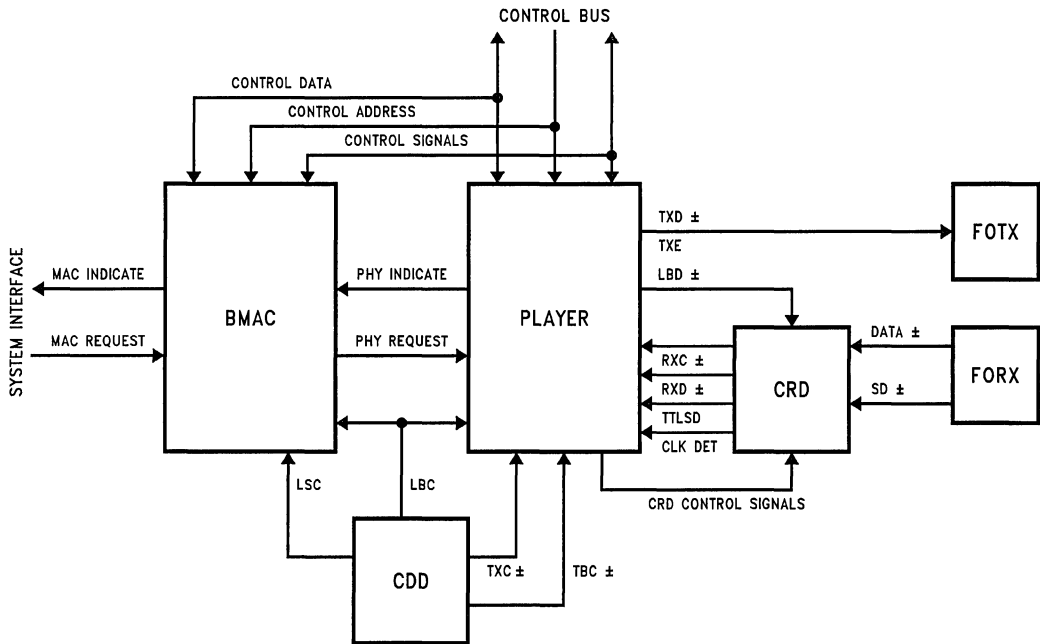


FIGURE 3. Basic Block Diagram of a Single Attach Station Using National's FDDI Chips

TL/F/10790-3

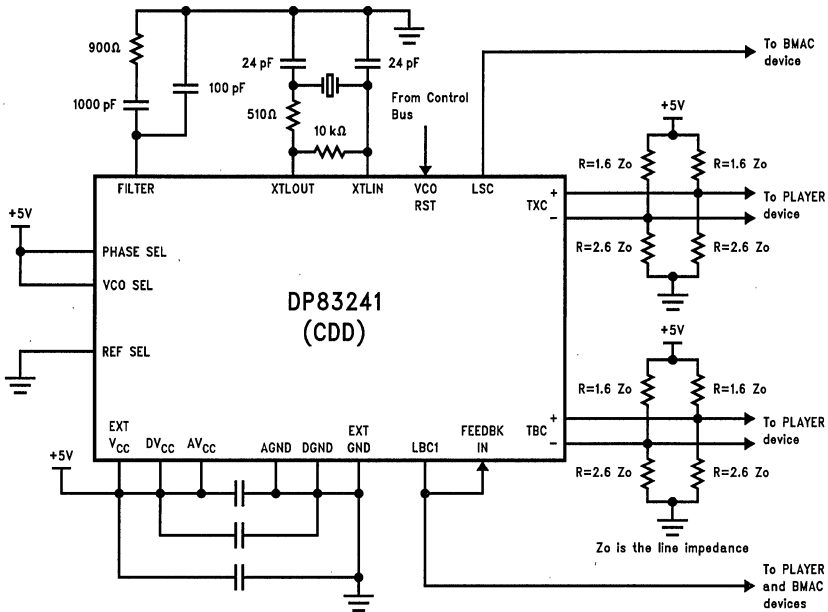


FIGURE 4. Basic DP83241 Connections in a Single Attach Station (SAS)

TL/F/10790-4

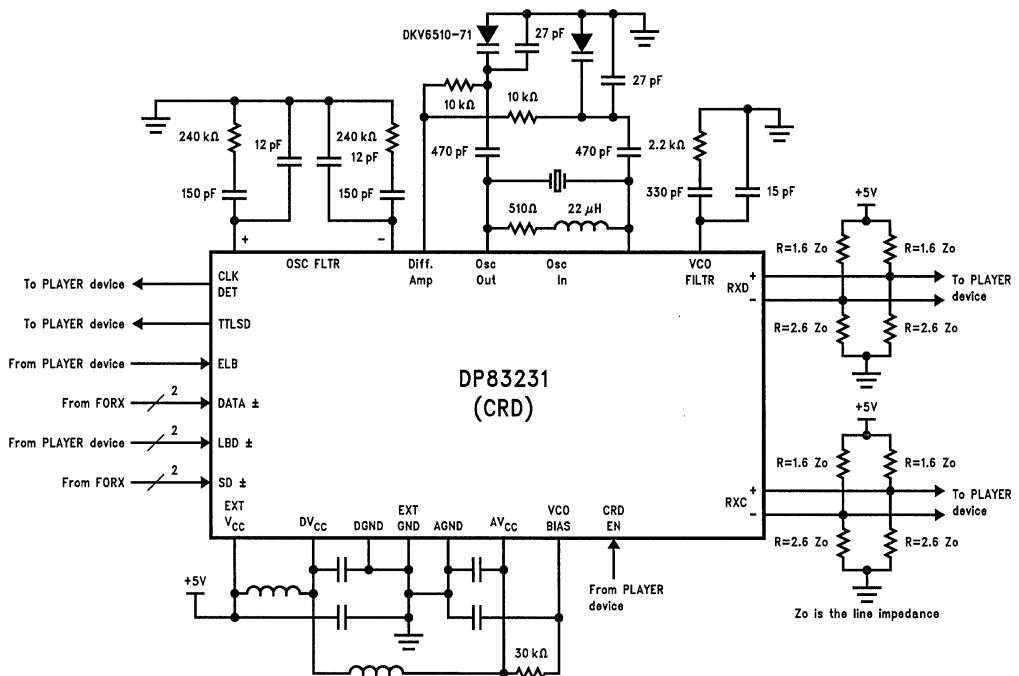


FIGURE 5. Basic DP83231 Connections in a Single Attach Station (SAS)

TL/F/10790-5

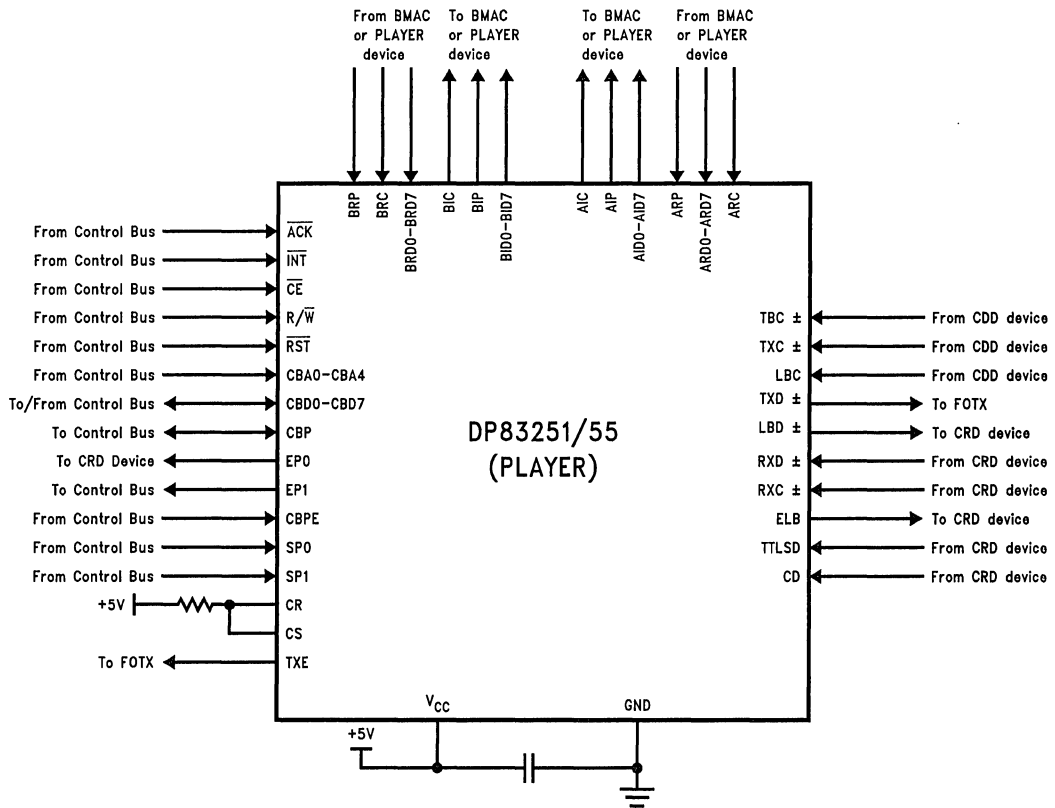


FIGURE 6. Basic DP83251/55 Connections for either an SAS or DAS

TL/F/10790-6

TERMINATION

In addition to keeping the ECL signal traces as short as possible to avoid reflections, it is necessary to properly terminate all of the ECL signal traces. If the ECL signals are not properly terminated, line reflections will occur. There are several methods for terminating signal traces but it is recommended that a Thevenin equivalent of the proper parallel termination be used. This method has the advantages of using a single power supply voltage and providing a pull-down resistor for the driver circuit.

Reflections occur not only from mismatched load and source impedances but also from changes in line impedance. The impedance of a signal trace can be set to a fairly accurate degree by setting the trace width to a particular value. The equation in Appendix B, System Considerations,

shows how the line impedance is determined by the dielectric constant of the PCB, the thickness of the trace, the width of the trace and the distance of the trace from the ground plane. The dielectric constant will vary depending on the board manufacturer that is being used, so it is advisable to get this information from the board manufacturer before the ECL signals are routed. In an attempt to prevent line reflections, sharp bends (changes in the characteristic impedance) in the signal line should be avoided. It is recommended that all bends in high speed signal traces be forty-five degrees or less. More information on reflections and termination schemes can be found in Appendix A, Transmission Line Concepts, and Appendix B, System Considerations.

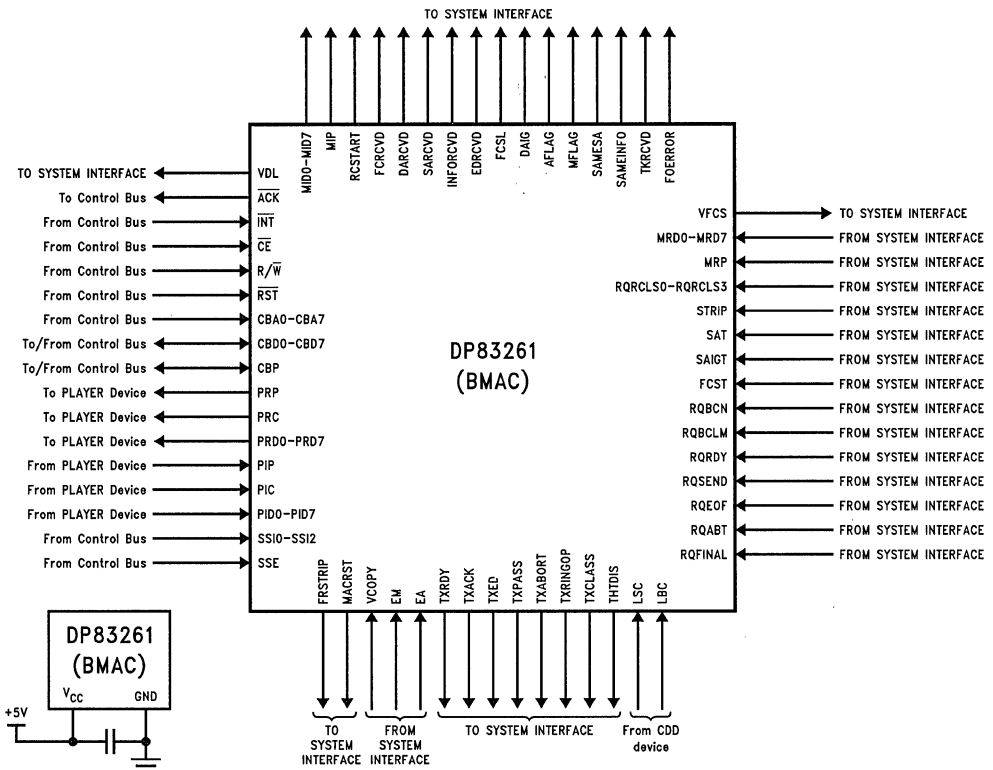


FIGURE 7. Basic DP83261 Connections

TL/F/10790-7

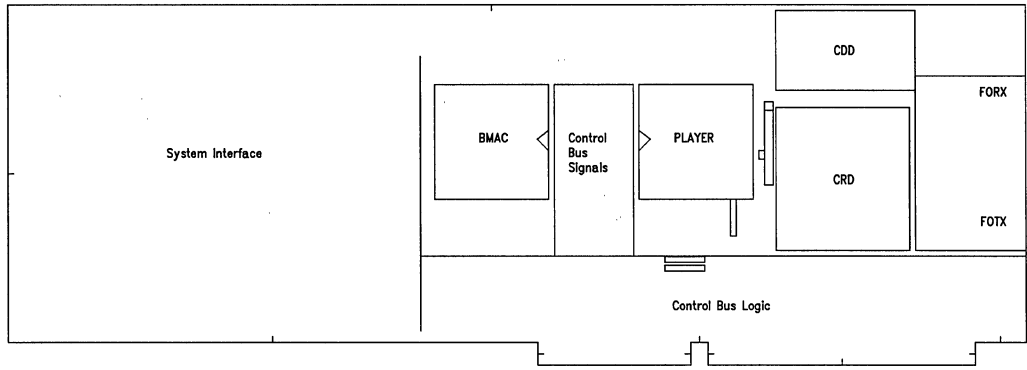
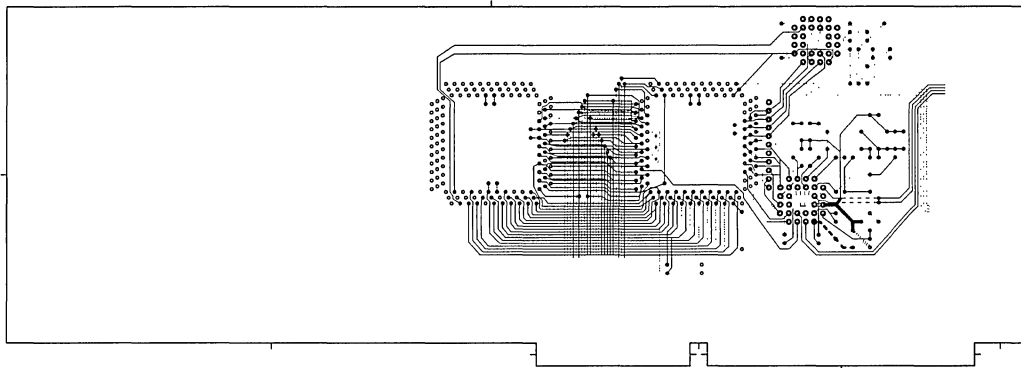


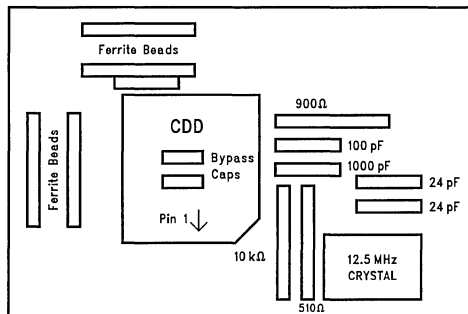
FIGURE 8. Silkscreen Showing Placement of National's FDDI Chip Set in an SAS for an AT Form Factor

TL/F/10790-8

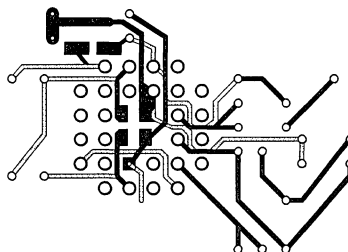


TL/F/10790-9

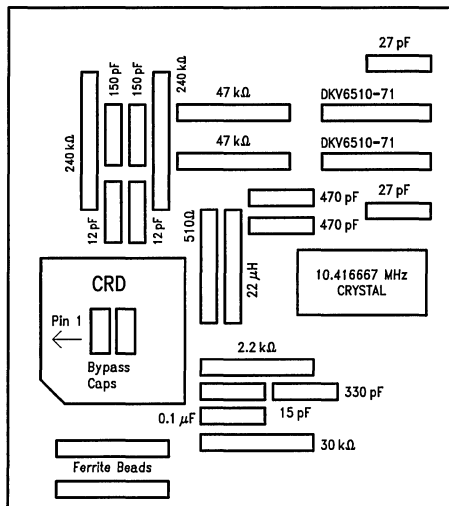
FIGURE 9. Signal Traces Connecting National's FDDI Chip Set in an SAS for an AT Form Factor



TL/F/10790-10

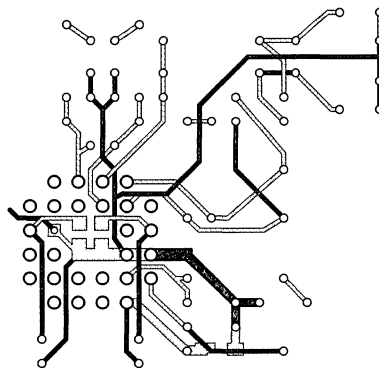


TL/F/10790-11



TL/F/10790-12

FIGURE 8a. Close Up of Silkscreen for CDD and CRD Portion of Figure 8



TL/F/10790-13

FIGURE 9a. Close Up of Signal Traces for CDD and CRD Portion of Figure 9. Black traces represent the component side signal layer and the shaded traces represent the solder side signal layer.

POWER AND GROUND

In the design of a system, most people concern themselves with the layout of the signal lines and components and not with the ground and power layout. In high speed design, the shortest ground return path and the least inductive ground help maintain large signal noise margins. A common technique used to reduce ground noise is to create separate ground islands that are connected at only one point on the board. This point, at which the ground islands should connect is where the ground enters the PCB. The goal of this technique is to separate the noise on the ground plane, due to high current switching devices, from the parts which are susceptible to noise. Since voltage is equal to the inductance multiplied by the change in current divided by the change in time, a high current switching device will affect the noise margin of a signal. It is not necessary to create different ground islands for each part but only to create a separate ground island for the high current switching devices such as line or bus drivers. In the Single Attach Station design example described earlier in this application note there was a ground plane with no islands. The only exceptions to this involved some ground pins on the CDD and CRD devices. These connections are explained in the recommended layout sections of their datasheets.

The ground return path should be kept as short as possible by keeping the ground for an output device and the load (termination) on the same ground island. High speed signals should not be routed across different ground islands since a change in impedance will occur from the break in the ground planes. Ground should be routed on signal layers between signals that run parallel to each other for long distances.

Proper power supply bypassing is also recommended so that V_{CC} levels are not reduced when a sudden switch in current occurs. This reduction of V_{CC} will not be global, but instead localized and could affect the noise margin of a signal.

Remember that in ECL design, V_{CC} is connected to a logic ground but is actually the highest voltage level. The V_{EE} supply level should be treated as the ground and V_{CC} as the power supply when using National's FDDI chip set since it uses positive referenced ECL signals rather than negative.

More information on power and ground can be found in Appendix C, Power Distribution and Thermal Considerations.

APPENDIX A ECL DESIGN GUIDE: TRANSMISSION LINE CONCEPTS

INTRODUCTION

The interactions between wiring and circuitry in high-speed systems are more easily determined by treating the interconnections as transmission lines. A brief review of basic concepts is presented and simplified methods of analysis are used to examine situations commonly encountered in digital systems. Since the principles and methods apply to any type of logic circuit, normalized pulse amplitudes are used in sample waveforms and calculations.

SIMPLIFYING ASSUMPTIONS

For the great majority of interconnections in digital systems, the resistance of the conductors is much less than the input and output resistance of the circuits. Similarly, the insulating materials have very good dielectric properties. These circumstances allow such factors as attenuation, phase distortion, and bandwidth limitations to be ignored. With these simplifications, interconnections can be dealt with in terms of characteristic impedance and propagation delay.

CHARACTERISTIC IMPEDANCE

The two conductors that interconnect a pair of circuits have distributed series inductance and distributed capacitance between them, and thus constitute a transmission line. For any length in which these distributed parameters are constant, the pair of conductors have a characteristic impedance Z_0 . Whereas quiescent conditions on the line are determined by the circuits and terminations, Z_0 is the ratio of transient voltage to transient current passing by a point on the line when a signal charge or other electrical disturbance occurs. The relationship between transient voltage, transient current, characteristic impedance, and the distributed parameters is expressed as follows:

$$\frac{V}{I} = Z_0 = \sqrt{\frac{L_0}{C_0}} \quad (\text{eq. 1})$$

where L_0 = inductance per unit length, C_0 = capacitance per unit length. Z_0 is in ohms, L_0 in Henries, C_0 in Farads.

PROPAGATION VELOCITY

Propagation velocity ν and its reciprocal, delay per unit length δ , can also be expressed in terms of L_0 and C_0 . A consistent set of units is nanoseconds, microhenries and picofarads, with a common unit of length.

$$\nu = \frac{1}{\sqrt{L_0 C_0}} \quad \delta = \sqrt{L_0 C_0} \quad (\text{eq. 2})$$

Equations 1 and 2 provide a convenient means of determining the L_0 and C_0 , of a line when delay, length and impedance are known. For a length l and delay T , δ is the ratio T/l . To determine L_0 and C_0 , combine Equations 1 and 2.

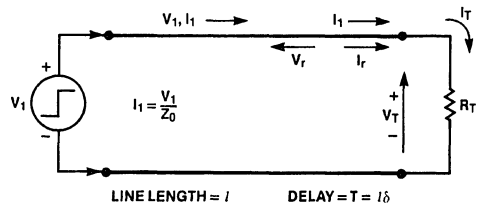
$$L_0 = \delta Z_0 \quad (\text{eq. 3})$$

$$C_0 = \frac{\delta}{Z_0} \quad (\text{eq. 4})$$

More formal treatments of transmission line characteristics, including loss effects, are available from many sources.¹⁻³

TERMINATION AND REFLECTION

A transmission line with a terminating resistor is shown in Figure 1. As indicated, a positive step function voltage travels from left to right. To keep track of reflection polarities, it is convenient to consider the lower conductor as the voltage reference and to think in terms of current flow in the top conductor only. The generator is assumed to have zero internal impedance. The initial current i_1 is determined by V_1 and Z_0 .



TL/F/10790-14

FIGURE 1. Assigned Polarities and Directions for Determining Reflections

If the terminating resistor matches the line impedance, the ratio of voltage to current traveling along the line is matched by the ratio of voltage to current which must, by Ohm's law, always prevail at R_T . From the viewpoint of the voltage step generator, no adjustment of output current is ever required; the situation is as though the transmission line never existed and R_T had been connected directly across the terminals of the generator. From the R_T viewpoint, the only thing the line did was delay the arrival of the voltage step by the amount of time T .

When R_T is not equal to Z_0 , the initial current starting down the line is still determined by V_1 and Z_0 but the final steady state current, after all reflections have died out, is determined by V_1 and R_T (ohmic resistance of the line is assumed to be negligible). The ratio of voltage to current in the initial wave is not equal to the ratio of voltage to current demanded by R_T . Therefore, at the instant the initial wave arrives at R_T , another voltage and current wave must be generated so that Ohm's law is satisfied at the line-load interface. This *reflected* wave, indicated by V_r and i_r in Figure 1, starts to return toward the generator. Applying

Kirchoff's laws to the end of the line at the instant the initial wave arrives, results in the following.

$$I_1 + I_r = I_T = \text{current into } R_T \quad (\text{eq. 5})$$

Since only one voltage can exist at the end of the line at this instant of time, the following is true:

$$V_1 + V_r = V_T$$

$$\text{thus } I_T = \frac{V_T}{R_T} = \frac{V_1 + V_r}{R_T} \quad (\text{eq. 6})$$

$$\text{also } I_1 = \frac{V_1}{Z_0} \text{ and } I_r = -\frac{V_r}{Z_0}$$

with the minus sign indicating that V_r is moving toward the generator.

Combining the foregoing relationships algebraically and solving for V_r yields a simplified expression in terms of V_1 , Z_0 and R_T .

$$\frac{V_1 - V_r}{Z_0} - \frac{V_r}{Z_0} = \frac{V_1 + V_r}{R_T} = \frac{V_1}{R_T} + \frac{V_r}{R_T}$$

$$V_1 \left(\frac{1}{Z_0} - \frac{1}{R_T} \right) = V_r \left(\frac{1}{R_T} + \frac{1}{Z_0} \right) \quad (\text{eq. 7})$$

$$V_r = V_1 \left(\frac{R_T - Z_0}{R_T + Z_0} \right) = \rho_L V_1$$

The term in parenthesis is called the coefficient of reflection ρ . With R_T ranging between zero (shorted line) and infinity (open line), the coefficient ranges between -1 and $+1$ respectively. The subscript L indicates that ρ refers to the coefficient at the load end of the line.

Equation 7 expresses the amount of voltage sent back down the line, and since

$$V_T = V_1 + V_r \quad (\text{eq. 8})$$

then $V_T = V_1 (1 + \rho_L)$.

V_T can also be determined from an expression which does not require the preliminary step of calculating ρ_L . Manipulating $(1 + \rho_L)$ results in

$$1 + \rho_L = 1 + \frac{R_T - Z_0}{R_T + Z_0} = 2 \left(\frac{R_T}{R_T + Z_0} \right)$$

Substituting in Equation 8 gives

$$V_T = 2 \left(\frac{R_T}{R_T + Z_0} \right) V_1 \quad (\text{eq. 9})$$

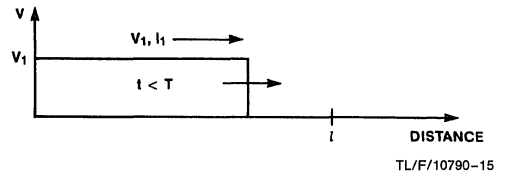
The foregoing has the same form as a simple voltage divider involving a generator V_1 with internal impedance Z_0 driving a load R_T , except that the amplitude of V_T is doubled.

The arrow indicating the direction of V_r in Figure 1 correctly indicates the V_r direction of travel, but the direction of I_r flow depends on the V_r polarity. If V_r is positive, I_r flows toward the generator, opposing I_1 . This relationship between the polarity of V_r and the direction of I_r can be deduced by noting in Equation 7 that if V_r is positive it is because R_T is

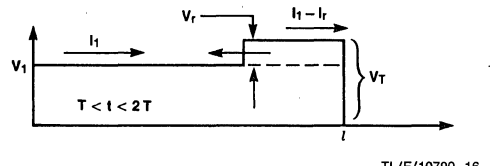
greater than Z_0 . In turn, this means that the initial current I_1 is larger than the final quiescent current, dictated by V_1 and R_T . Hence, I_r must oppose I_1 to reduce the line current to the final quiescent value. Similar reasoning shows that if V_r is negative, I_r flows in the same direction as I_1 .

It is sometimes easier to determine the effect of V_r on line conditions by thinking of it as an independent voltage generator in series with R_T . With this concept, the direction of I_r is immediately apparent; its magnitude, however, is the ratio of V_r to Z_0 , i.e., R_T is already accounted for in the magnitude of V_r . The relationships between incident and reflected signals are represented in Figure 2 for both cases of mismatch between R_T and Z_0 .

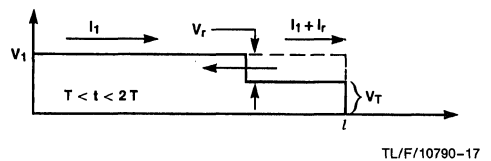
The incident wave is shown in Figure 2a, before it has reached the end of the line. In Figure 2b, a positive V_r is returning to the generator. To the left of V_r the current is still I_1 , flowing to the right, while to the right of V_r the net current in the line is the difference between I_1 and I_r . In Figure 2c, the reflection coefficient is negative, producing a negative V_r . This, in turn, causes an increase in the amount of current flowing to the right behind the V_r wave.



a. Incident Wave



b. Reflected Wave for $R_T > Z_0$



c. Reflected Wave for $R_T < Z_0$
FIGURE 2. Reflections for $R_T \neq Z_0$

SOURCE IMPEDANCE, MULTIPLE REFLECTIONS

When a reflected voltage arrives back at the source (generator), the reflection coefficient at the source determines the response to V_r . The coefficient of reflection at the source is governed by Z_0 and the source resistance R_S .

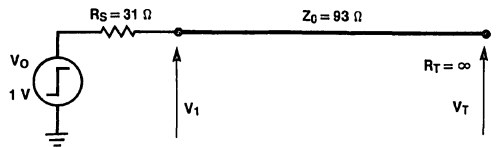
$$\rho_S = \frac{R_S - Z_0}{R_S + Z_0} \quad (\text{eq. 10})$$

If the source impedance matches the line impedance, a reflected voltage arriving at the source is not reflected back toward the load end. Voltage and current on the line are stable with the following values.

$$V_T = V_1 + V_r \text{ and } I_T = I_1 - I_r \quad (\text{eq. 11})$$

If neither source impedance nor terminating impedance matches Z_0 , multiple reflections occur; the voltage at each end of the line comes closer to the final steady state value with each succeeding reflection. An example of a line mismatched on both ends is shown in *Figure 3*. The source is a step function of 1V amplitude occurring at time t_0 . The initial value of V_1 starting down the line is 0.75V due to the voltage divider action of Z_0 and R_S . The time scale in the photograph shows that the line delay is approximately 6 ns. Since neither end of the line is terminated in its characteristic impedance, multiple reflections occur.

The amplitude and persistence of the ringing shown in *Figure 3* become greater with increasing mismatch between the line impedance and source and load impedances. Re-

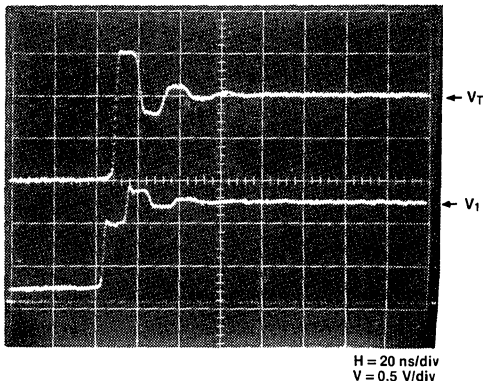


TL/F/10790-18

$$\rho_S = \frac{31 - 93}{31 + 93} = -0.5$$

$$\rho_L = \frac{\infty - 93}{\infty + 93} = +1$$

$$\text{Initially: } V_1 = \frac{Z_0}{Z_0 + R_S} \cdot V_0 = \frac{93}{124} \cdot 1 = 0.75\text{V}$$



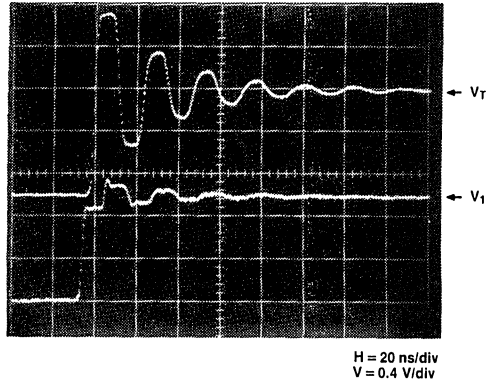
TL/F/10790-19

FIGURE 3. Multiple Reflections Due to Mismatch at Load and Source

ducing R_S (*Figure 3*) to 13Ω increases ρ_S to -0.75V , and the effects are illustrated in *Figure 4*. The initial value of V_T is 1.8V with a reflection of 0.9V from the open end. When this reflection reaches the source, a reflection of $0.9\text{V} \times -0.75\text{V}$ starts back toward the open end. Thus, the second increment of voltage arriving at the open end is negative going. In turn, a negative-going reflection of $0.9\text{V} \times -0.75\text{V}$ starts back toward the source. This negative increment is again multiplied by -0.75 at the source and returned toward the open end. It can be deduced that the difference in amplitude between the first two positive peaks observed at the open end is

$$V_T - V'_T = (1 + \rho_L)V_1 - (1 + \rho_L)V_1\rho_L^2\rho_S^2 \\ = (1 + \rho_L)V_1(1 - \rho_L^2\rho_S^2) \quad (\text{eq. 12})$$

The factor $(1 - \rho_L^2\rho_S^2)$ is similar to the damping factor associated with lumped constant circuitry. It expresses the attenuation of successive positive or negative peaks of ringing.

H = 20 ns/div
V = 0.4 V/div

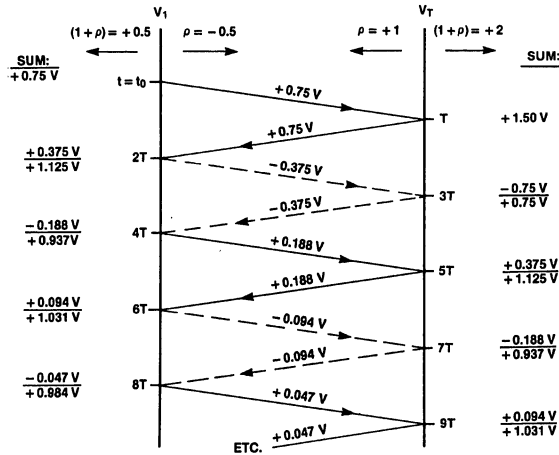
TL/F/10790-20

FIGURE 4. Extended Ringing when R_S of *Figure 3* is Reduced to 13Ω

LATTICE DIAGRAM

In the presence of multiple reflections, keeping track of the incremental waves on the line and the net voltage at the ends becomes a bookkeeping chore. A convenient and systematic method of indicating the conditions which combines magnitude, polarity and time utilizes a graphic construction called a lattice diagram.⁴ A lattice diagram for the line conditions of *Figure 3* is shown in *Figure 5*.

The vertical lines symbolize discontinuity points, in this case the ends of the line. A time scale is marked off on each line in increments of $2T$, starting at t_0 for V_1 and T for V_T . The diagonal lines indicate the incremental voltages traveling between the ends of the line; solid lines are used for positive voltages and dashed lines for negative. It is helpful to write the reflection and transmission multipliers ρ and $(1 + \rho)$ at each vertical line, and to tabulate the incremental and net voltages in columns alongside the vertical lines. Both the lattice diagram and the waveform photograph show that V_1 and V_T asymptotically approach 1V , as they must with a 1V source driving an open-ended line.



TL/F/10790-21

FIGURE 5. Lattice Diagram for the Circuit of Figure 3

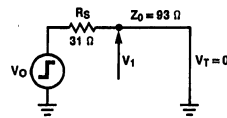
SHORTED LINE

The open-ended line in Figure 3 has a reflection coefficient of +1 and the successive reflections tend toward the steady state conditions of zero line current and a line voltage equal to the source voltage. In contrast, a shorted line has a reflection coefficient of -1 and successive reflections must cause the line conditions to approach the steady state conditions of zero voltage and a line current determined by the source voltage and resistance.

Shorted line conditions are shown in Figure 6a with the reflection coefficient at the source end of the line also negative. A negative coefficient at both ends of the line means that any voltage approaching either end of the line is reflected in the opposite polarity. Figure 6b shows the response to an input step-function with a duration much longer than the

line delay. The initial voltage starting down the line is about +0.75V, which is inverted at the shorted end and returned toward the source as -0.75V. Arriving back at the source end of the line, this voltage is multiplied by (1 + rho_s), causing a -0.37V net change in V1. Concurrently, a reflected voltage of +0.37V (-0.75V times rho_s of -0.5) starts back toward the shorted end of the line. The voltage at V1 is reduced by 50% with each successive round trip of reflections, thus leading to the final condition of zero volts on the line.

When the duration of the input pulse is less than the delay of the line, the reflections observed at the source end of the line constitute a train of negative pulses, as shown in Figure 6c. The amplitude decreases by 50% with each successive occurrence as it did in Figure 6b.

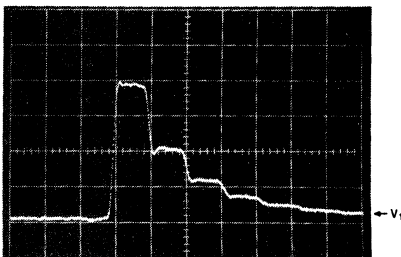


TL/F/10790-99

$$\rho_s = -0.5$$

$$\rho_L = \frac{0 - 93}{0 + 93} = -1$$

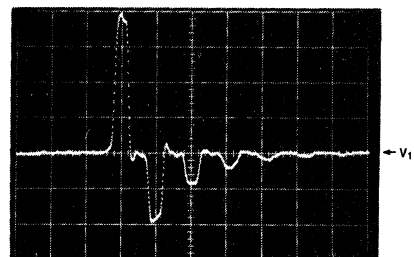
a. Reflection Coefficients for Shorted Line



H = 10 ns/div
V = 0.2 V/div

TL/F/10790-22

b. Input Pulse Duration > Line Delay



H = 10 ns/div
V = 0.2 V/div

TL/F/10790-23

c. Input Pulse Duration < Line Delay

FIGURE 6. Reflections of Long and Short Pulses on a Shorted Line

SERIES TERMINATION

Driving an open-ended line through a source resistance equal to the line impedance is called series termination. It is particularly useful when transmitting signals which originate on a PC board and travel through the backplane to another board, with the attendant discontinuities, since reflections coming back to the source are absorbed and ringing thereby controlled. Figure 7 shows a 93Ω line driven from a 1V generator through a source impedance of 93Ω. The photograph illustrates that the amplitude of the initial signal sent down the line is only half of the generator voltage, while the voltage at the open end of the line is doubled to full amplitude ($1 + \rho_L = 2$). The reflected voltage arriving back at the source raises V_1 to the full amplitude of the generator signal. Since the reflection coefficient at the source is zero, no further changes occur and the line voltage is equal to the generator voltage. Because the initial signal on the line is only half the normal signal swing, the loads must be connected at or near the end of the line to avoid receiving a 2-step input signal.

An ECL output driving a series terminated line requires a pull-down resistor to V_{EE} , as indicated in Figure 8. The resistor R_0 shown in Figure 8 symbolizes the output resistance of the ECL gate. The relationships between R_0 , R_S , R_E and Z_0 are discussed in Appendix B.

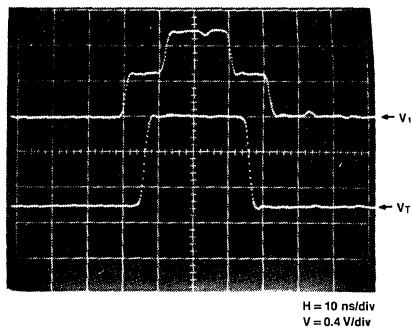
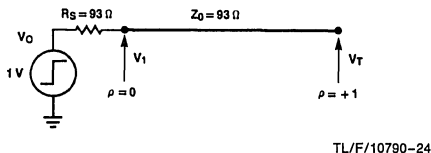


FIGURE 7. Series Terminated Line and Waveforms

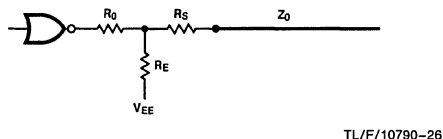
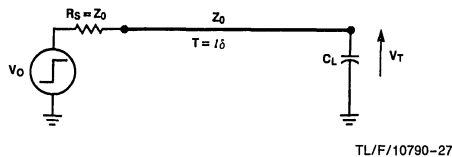


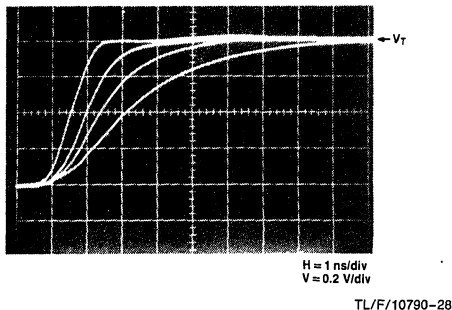
FIGURE 8. ECL Element Driving a Series Terminated Line

EXTRA DELAY WITH TERMINATION CAPACITANCE

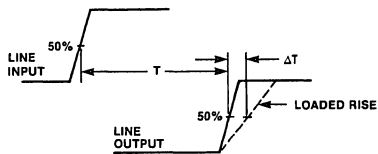
Designers should consider the effect of the load capacitance at the end of the line when using series termination. Figure 9 shows how the output waveform changes with increasing load capacitance. Figure 9b shows the effect of load capacitances of 0, 12, 24, 48 pF. With no load, the delay between the 50% points of the input and output is just the line delay T . A capacitive load at the end of the line causes an extra delay ΔT due to the increase in rise time of the output signal. The midpoint of the output is used as a criterion because the propagation delay of an ECL circuit is measured between the 50% points of the input and output signals.



a. Series Terminated Line with Load Capacitance

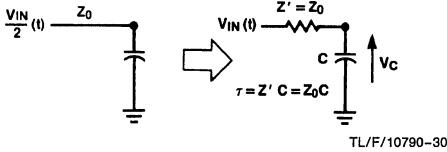


b. Output Rise Time Increase with Increasing Load Capacitance

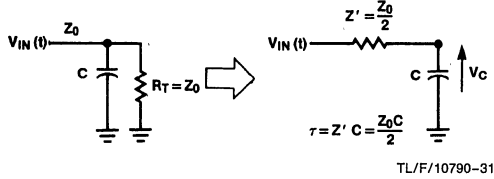


c. Extra Delay ΔT Due to Rise Time Increase

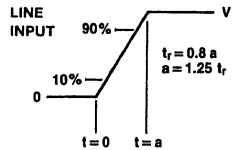
FIGURE 9. Extra Delay with Termination Capacitance



a. Thevenin Equivalent for Series Terminated Case



b. Thevenin Equivalent for Parallel Terminated Case



TL/F/10790-32

$$v_{in}(t) = \frac{V}{a} [tu(t) - (t-a)u(t-a)]$$

$$u(t) = \begin{cases} 0 & \text{for } t < 0 \\ 1 & \text{for } t > 0 \end{cases}$$

$$u(t-a) = \begin{cases} 0 & \text{for } t < a \\ 1 & \text{for } t > a \end{cases}$$

$$V_{IN}(S) = \frac{V}{as^2} (1 - e^{-as})$$

$$V_C(S) = \frac{V}{ar} \cdot \frac{1}{s^2(s + 1/\tau)} (1 - e^{-as})$$

$$v_c(t) = \frac{V}{a} [t - \tau(1 - e^{-t/\tau})] u(t)$$

$$- \frac{V}{a} [(t-a)$$

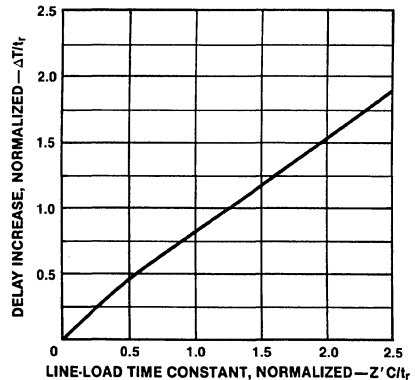
$$- \tau(1 - e^{-\frac{t-a}{\tau}})] u(t-a)$$

c. Equations for Input and Output Voltages

FIGURE 10. Determining the Effect of End-of-Line Capacitance

The increase in propagation delay can be calculated by using a ramp approximation for the incident voltage and characterizing the circuit as a fixed impedance in series with the load capacitance, as shown in Figure 10. One general solution serves both series and parallel termination cases by using an impedance Z' and a time constant τ , defined in Figure 10a and 10b. Calculated and observed increases in delay time to the 50% point show close agreement when τ is less than half the ramp time. At large ratios of τ/a (where a = ramp time), measured delays exceed calculated values by approximately 7%. Figure 11, based on measured values, shows the increase in delay to the 50% point as a function of the Z'C time constant, both normalized to the 10% to 90% rise time of the input signal. As an example of using the graph, consider a 100Ω series terminated line with 30 pF load capacitance at the end of the line and a no-load rise time of 3 ns for the input signal. From Figure 10a, Z' is equal to 100Ω; the ratio $Z'C/t_r$ is 1. From the graph, the ratio $\Delta T/t_r$ is 0.8. Thus the increase in the delay to the 50% point of the output waveform is 0.8 t_r , or 2.4 ns, which is then added to the no-load line delay T to determine the total delay.

Had the 100Ω line in the foregoing example been parallel rather than series terminated at the end of the line, Z' would be 50Ω. The added delay would be only 1.35 ns with the same 30 pF loading at the end. The added delay would be only 0.75 ns if the line were 50Ω and parallel terminated. The various trade-offs involving type of termination, line impedance, and loading are important considerations for critical delay paths.



TL/F/10790-33

FIGURE 11. Increase in 50% Point Delay Due to Capacitive Loading at the End of the Line, Normalized to T_r

DISTRIBUTED LOADING EFFECTS ON LINE CHARACTERISTICS

When capacitive loads such as ECL inputs are connected along a transmission line, each one causes a reflection with a polarity opposite to that of the incident wave. Reflections from two adjacent loads tend to overlap if the time required for the incident wave to travel from one load to the next is equal to or less than the signal rise time.⁵ *Figure 12a* illustrates an arrangement for observing the effects of capacitive loading, while *Figure 12b* shows an incident wave followed by reflections from two capacitive loads. The two capacitors causing the reflections are separated by a distance requiring a travel time of 1 ns. The two reflections return to the source 2 ns apart, since it takes 1 ns longer for the incident wave to reach the second capacitor and an additional 1 ns for the second reflection to travel back to the source. In the upper trace of *Figure 12b*, the input signal rise time is 1 ns and there are two distinct reflections, although the trailing edge of the first overlaps the leading edge of the second. The input rise time is longer in the middle trace, causing a greater overlap. In the lower trace, the 2 ns input rise time causes the two reflections to merge and appear as a single reflection which is relatively constant (at $\approx -10\%$) for half its duration. This is about the same reflection that would occur if the 93Ω line had a middle section with an impedance reduced to 75Ω.

With a number of capacitors distributed all along the line of *Figure 12a*, the combined reflections modify the observed input waveform as shown in the top trace of *Figure 12c*. The reflections persist for a time equal to the 2-way line delay (15 ns), after which the line voltage attains its final value. The waveform suggests a line terminated with a resistance greater than its characteristic impedance ($R_T > Z_0$).

This analogy is strengthened by observing the effect of reducing R_T from 93Ω to 75Ω, which leads to the middle waveform of *Figure 12c*. Note that the final (steady state) value of the line voltage is reduced by about the same amount as that caused by the capacitive reflections. In the lower trace of *Figure 12c* the source resistance R_S is reduced from 93Ω to 75Ω, restoring both the initial and final line voltage values to the same amplitude as the final value in the upper trace. From the standpoint of providing a desired signal voltage on the line and impedance matching at either end, the effect of distributed capacitive loading can be treated as a reduction in line impedance.

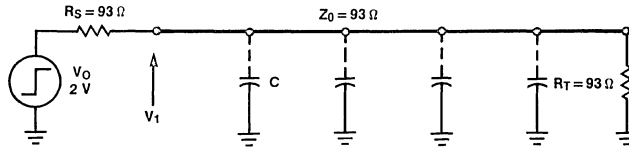
The reduced line impedance can be calculated by considering the load capacitance C_L as an increase in the intrinsic line capacitance C_0 along that portion of the line where the loads are connected.⁶ Denoting this length of line as l , the distributed value C_D of the load capacitance is as follows.

$$C_D = \frac{C_L}{l}$$

C_D is then added to C_0 in *Equation 1* to determine the reduced line impedance Z_0' .

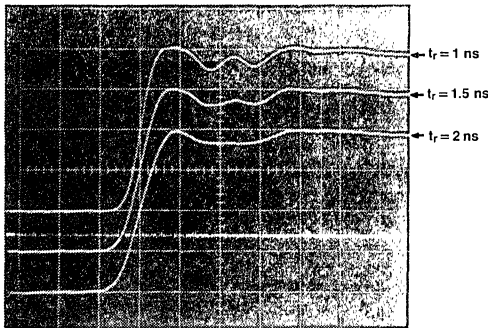
$$Z_0' = \sqrt{\frac{L_0}{C_0 + C_D}} = \sqrt{\frac{L_0}{C_0 \left(1 + \frac{C_D}{C_0}\right)}} \tag{eq. 13}$$

$$Z_0' + \frac{\sqrt{\frac{L_0}{C_0}}}{\sqrt{1 + \frac{C_D}{C_0}}} = \frac{Z_0}{\sqrt{1 + \frac{C_D}{C_0}}}$$



TL/F/10790-34

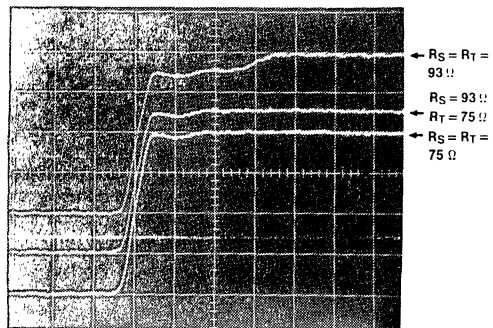
a. Arrangement for Observing Capacitive Loading Effects



H = 2 ns/div
V = 0.25 V/div

TL/F/10790-35

b. Capacitive Reflections Merging as Rise Time Increases



H = 5 ns/div
V = 0.25 V/div

TL/F/10790-36

c. Matching the Altered Impedance of a Capacitively Loaded Line

FIGURE 12. Capacitive Reflections and Effects on Line Characteristics

In the example of *Figure 12c*, the total load capacitance is 33 pF while the total intrinsic line capacitance $/C_0$ is 60 pF. (Note that the ratio C_D/C_0 is the same as C_L/C_0 .) The calculated value of the reduced impedance is thus

$$Z'_0 = \frac{93}{\sqrt{1 + \frac{33}{60}}} = \frac{93}{\sqrt{1.55}} = 75\Omega \quad (\text{eq. 14})$$

This correlates with the results observed in *Figure 12c* when R_T and R_S are reduced to 75Ω.

The distributed load capacitance also increases the line delay, which can be calculated from *Equation 2*.

$$\begin{aligned} \delta' &= \sqrt{L_0(C_0 + C_D)} = \sqrt{L_0 C_0} \sqrt{1 + \frac{C_D}{C_0}} \\ &= \delta \sqrt{1 + \frac{C_D}{C_0}} \end{aligned} \quad (\text{eq. 15})$$

The line used in the example of *Figure 12c* has an intrinsic delay of 6 ns and a loaded delay of 7.5 ns which checks with *Equation 15*.

$$l\delta' = l\delta \sqrt{1.55} = 6 \sqrt{1.55} = 7.5 \text{ ns} \quad (\text{eq. 16})$$

Equation 15 can be used to predict the delay for a given line and load. The ratio C_D/C_0 (hence the loading effect) can be minimized for a given loading by using a line with a high intrinsic capacitance C_0 .

A plot of Z' and δ' for a 50Ω line as a function of C_D is shown in *Figure 13*. This figure illustrates that relatively modest amounts of load capacitance will add appreciably to the propagation delay of a line. In addition, the characteristic impedance is reduced significantly.

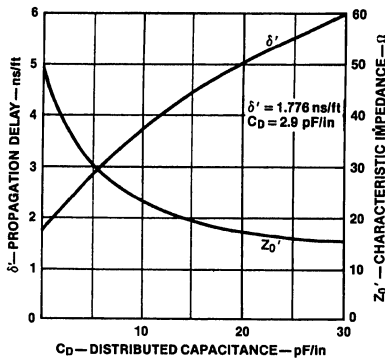


FIGURE 13. Capacitive Loading Effects on Line Delay and Impedance

TL/F/10790-37

Worst case reflections from a capacitively loaded section of transmission line can be accurately predicted by using the modified impedance of *Equation 9.6*. When a signal originates on an unloaded section of line, the effective reflection coefficient is as follows.

$$\rho = \frac{Z'_0 - Z_0}{Z'_0 + Z_0} \quad (\text{eq. 17})$$

MISMATCHED LINES

Reflections occur not only from mismatched load and source impedances but also from changes in line impedance. These changes could be caused by bends in coaxial cable, unshielded twisted-pair in contact with metal, or mismatch between PC board traces and backplane wiring. With the coax or twisted-pair, line impedance changes run about 5% to 10% and reflections are usually no problem since the percent reflection is roughly half the percent change in impedance. However, between PC board and backplane wiring, the mismatch can be 2 or 3 to 1. This is illustrated in *Figure 14* and analyzed in the lattice diagram of *Figure 15*. Line 1 is driven in the series terminated mode so that reflections coming back to the source are absorbed.

The reflection and transmission at the point where impedances differ are determined by treating the downstream line as though it were a terminating resistor. For the example of *Figure 14*, the reflection coefficient at the intersection of lines 1 and 2 for a signal traveling to the right is as follows.

$$\rho_{12} = \frac{Z_2 - Z_1}{Z_2 + Z_1} = \frac{93 - 50}{143} = +0.3 \quad (\text{eq. 18})$$

Thus the signal reflected back toward the source and the signal continuing along line 2 are, respectively, as follows.

$$V_{1r} = \rho_{12} V_1 = +0.3V_1 \quad (\text{eq. 19a})$$

$$V_2 = (1 + \rho_{12}) V_1 = +1.3 V_1 \quad (\text{eq. 19b})$$

At the intersection of lines 2 and 3, the reflection coefficient for signals traveling to the right is determined by treating Z_3 as a terminating resistor.

$$\rho_{23} = \frac{Z_3 - Z_2}{Z_3 + Z_2} = \frac{39 - 93}{132} = -0.41 \quad (\text{eq. 20})$$

When V_2 arrives at this point, the reflected and transmitted signals are as follows.

$$\begin{aligned} V_{2r} &= \rho_{23} V_2 = -0.41 V_2 \\ &= (-0.41)(1.3) V_1 \\ &= -0.53 V_1 \end{aligned} \quad (\text{eq. 21a})$$

$$\begin{aligned} V_3 &= (1 + \rho_{23}) V_2 = 0.59 V_2 \\ &= (0.59)(1.3) V_1 \\ &= 0.77 V_1 \end{aligned} \quad (\text{eq. 21b})$$

Voltage V_3 is doubled in magnitude when it arrives at the open-ended output, since ρ_L is +1. This effectively cancels the voltage divider action between R_S and Z_1 .

$$\begin{aligned} V_4 &= (1 + \rho_L) V_3 = (1 + \rho_L)(1 + \rho_{23}) V_2 \\ &= (1 + \rho_L)(1 + \rho_{23})(1 + \rho_{12}) V_1 \\ &= (1 + \rho_L)(1 + \rho_{23})(1 + \rho_{12}) \frac{V_0}{2} \end{aligned} \quad (\text{eq. 22})$$

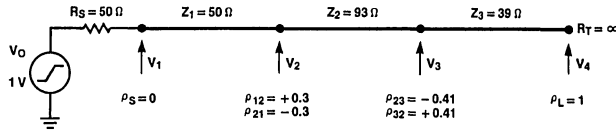
$V_4 = (1 + \rho_{23})(1 + \rho_{12}) V_0$
Thus, *Equation 22* is the general expression for the initial step of output voltage for three lines when the input is series terminated and the output is open-ended.

Note that the reflection coefficients at the intersections of lines 1 and 2 and lines 2 and 3 in *Figure 15* have reversed signs for signals traveling to the left. Thus the voltage reflected from the open output and the signal reflecting back and forth on line 2 both contribute additional increments of output voltage in the same polarity as V_O . Lines 2 and 3 have the same delay time; therefore, the two aforementioned increments arrive at the output simultaneously at time $5T$ on the lattice diagram (*Figure 15*).

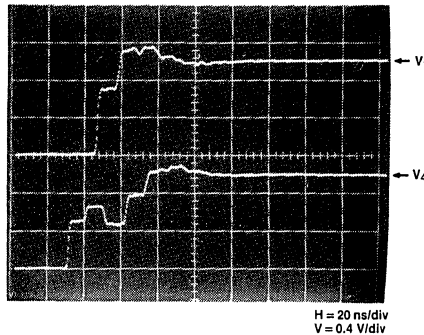
In the general case of series lines with different delay times, the vertical lines on the lattice diagram should be spaced apart in the ratio of the respective delays. *Figure 16* shows this for a hypothetical case with delay ratios 1:2:3. For a sequence of transmission lines with the highest imped-

ance line in the middle, at least three output voltage increments with the same polarity as V_O occur before one can occur of opposite polarity. On the other hand, if the middle line has the lowest impedance, the polarity of the second increment of output voltage is the opposite of V_O . The third increment of output voltage has the opposite polarity, for the time delay ratios of *Figure 16*.

When transmitting logic signals, it is important that the initial step of line output voltage pass through the threshold region of the receiving circuit, and that the next two increments of output voltage augment the initial step. Thus in a series terminated sequence of three mismatched lines, the middle line should have the highest impedance.



TL/F/10790-38



TL/F/10790-39

FIGURE 14. Reflections from Mismatched Lines

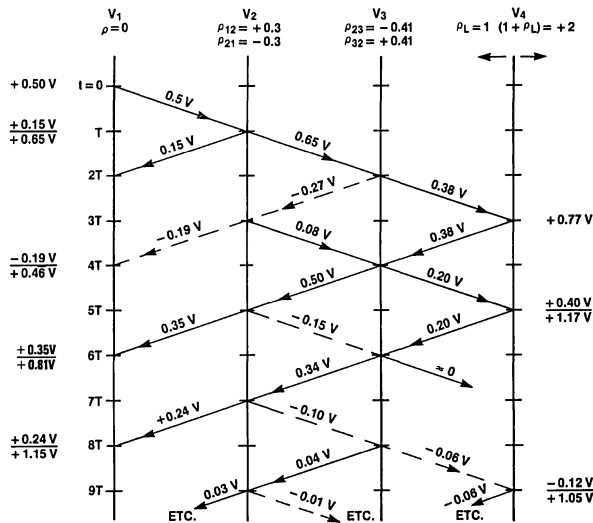
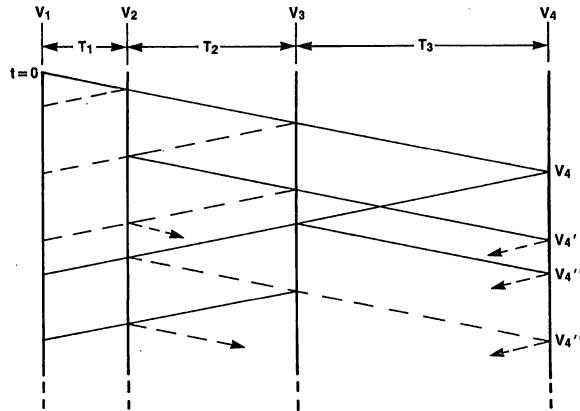


FIGURE 15. Lattice Diagram for the Circuit of *Figure 14*

TL/F/10790-40



TL/F/10790-41

FIGURE 16. Lattice Diagram for Three Lines with Delay Ratios 1:2:3

RISE TIME VERSUS LINE DELAY

When the 2-way line delay is less than the rise time of the input wave, any reflections generated at the end of the line are returned to the source before the input transition is completed. Assuming that the generator has a finite source resistance, the reflected wave adds algebraically to the input wave while it is still in transition, thereby changing the shape of the input. This effect is illustrated in Figure 17, which shows input and output voltages for several comparative values of rise time and line delay.

In Figure 17b where the rise time is much shorter than the line delay, V_1 rises to an initial value of 1V. At time T later, V_T rises to 0.5V, i.e., $1 + \rho_L = 0.5$. The negative reflection arrives back at the source at time $2T$, causing a net change of $-0.4V$, i.e., $(1 + \rho_S)(-0.5) = -0.4$.

The negative coefficient at the source changes the polarity of the other 0.1V of the reflection and returns it to the end of the line, causing V_T to go positive by another 50 mV at time $3T$. The remaining 50 mV is inverted and reflected back to the source, where its effect is barely distinguishable as a small negative change at time $4T$.

In Figure 17c, the input rise time (0% to 100%) is increased to such an extent that the input ramp ends just as the negative reflection arrives back at the source end. Thus the input rise time is equal to $2T$.

The input rise time is increased to $4T$ in Figure 17d, with the negative reflection causing a noticeable change in input slope at about its midpoint. This change in slope is more visible in the double exposure photo of Figure 17e, which shows V_1 (t_r still set for $4T$) with and without the negative reflection. The reflection was eliminated by terminating the line in its characteristic impedance.

The net input voltage at any particular time is determined by adding the reflection to the otherwise unaffected input. It must be remembered that the reflection arriving back at the input at a given time is proportional to the input voltage at a time $2T$ earlier. The value of V_1 in Figure 17d can be calculated by starting with the 1V input ramp.

$$V_1 = \frac{1}{t_r} \cdot t \quad \text{for } 0 \leq t \leq 4T \quad \text{(eq. 23)}$$

$$= 1V \quad \text{for } t \geq 4T$$

The reflection from the end of the line is

$$V_r = \frac{\rho_L (t - 2T)}{t_r} \quad \text{(eq. 24)}$$

the portion of the reflection that appears at the input is

$$V'_r = \frac{(1 + \rho_S) \rho_L (t - 2T)}{t_r} \quad \text{(eq. 25)}$$

the net value of the input voltage is the sum.

$$V'_1 = \frac{t}{t_r} + \frac{(1 + \rho_S) \rho_L (t - 2T)}{t_r} \quad \text{(eq. 26)}$$

The peak value of the input voltage in Figure 17d is determined by substituting values and letting t equal $4T$.

$$V'_1 = 1 + \frac{(0.8)(-0.5)(4T - 2T)}{t_r} \quad \text{(eq. 27)}$$

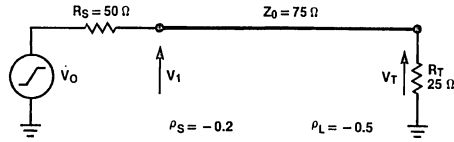
$$= 1 - 0.4(0.5) = 0.8V$$

After this peak point, the input ramp is no longer increasing but the reflection is still arriving. Hence the net value of the input voltage decreases. In this example, the later reflections are too small to be detected and the input voltage is thus stable after time $6T$. For the general case of repeated reflections, the net voltage $V_1(t)$ seen at the driven end of the line can be expressed as follows, where the signal caused by the generator is $V_1(t)$.

$$\begin{aligned}
 V'_1(t) &= V_1(t) && \text{for } 0 < t < 2T \\
 V'_1(t) &= V_1(t) + (1 + \rho_S) \rho_L V_1(t-2T) && \text{for } 2T < t < 4T \\
 V'_1(t) &= V_1(t) + (1 + \rho_S) \rho_L V_1(t-2T) && + (1 + \rho_S) \rho_S \rho_L^2 V_1(t-4T) && \text{for } 4T < t < 6T \\
 V'_1(t) &= V_1(t) + (1 + \rho_S) \rho_L V_1(t-2T) && + (1 + \rho_S) \rho_S \rho_L^2 V_1(t-4T) && + (1 + \rho_S) \rho_S^2 \rho_L^3 V_1(t-6T) && \text{for } 6T < t < 8T, \text{ etc.}
 \end{aligned}
 \tag{eq. 28}$$

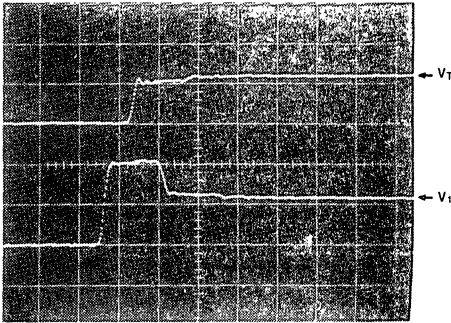
The voltage at the output end of the line is expressed in a similar manner.

$$\begin{aligned}
 V_T(t) &= 0 && \text{for } 0 < t < T \\
 V_T(t) &= (1 + \rho_L) V_1(t-T) && \text{for } T < t < 3T \\
 V_T(t) &= (1 + \rho_L) V_1(t-T) && + (1 + \rho_L) \rho_S \rho_L V_1(t-3T) && \text{for } 3T < t < 5T \\
 V_T(t) &= (1 + \rho_L) V_1(t-T) && + (1 + \rho_L) \rho_S \rho_L V_1(t-3T) && + (1 + \rho_L) \rho_S^2 \rho_L^2 V_1(t-5T) && \text{for } 5T < t < 7T, \text{ etc.}
 \end{aligned}
 \tag{eq. 29}$$



TL/F/10790-42

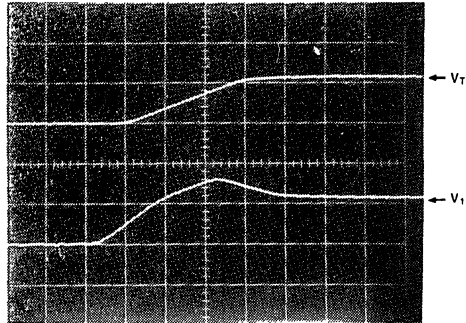
a. Test Arrangement for Rise Time Analysis



H = 10 ns/div
V = 0.5 V/div

TL/F/10790-43

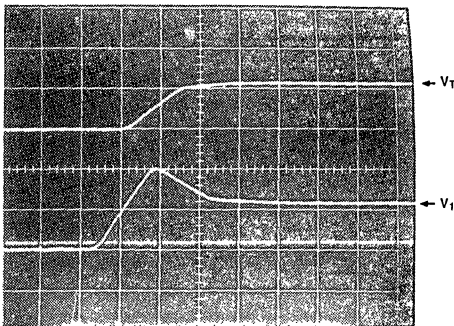
b. Line Voltages for $t_r \leq T$



H = 10 ns/div
V = 0.5 V/div

TL/F/10790-45

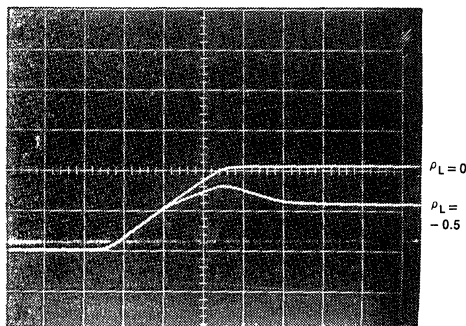
d. Line Voltages for $t_r = 4T$



H = 10 ns/div
V = 0.5 V/div

TL/F/10790-44

c. Line Voltages for $t_r = 2T$



H = 10 ns/div
V = 0.5 V/div

TL/F/10790-46

e. Input Voltage with and without Reflection

FIGURE 17. Line Voltages for Various Ratios of Rise Time to Line Delay

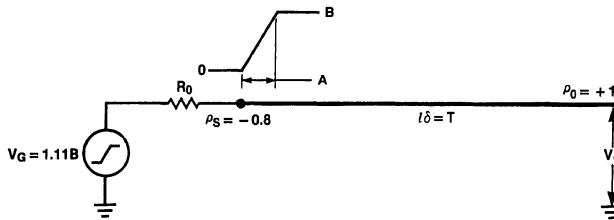
RINGING

Multiple reflections occur on a transmission line when neither the signal source impedance nor the termination (load) impedance matches the line impedance. When the source reflection coefficient ρ_S and the load reflection coefficient ρ_L are of opposite polarity, the reflections alternate in polarity. This causes the signal voltage to oscillate about the final steady state value, commonly recognized as ringing.

When the signal rise time is long compared to the line delay, the signal shape is distorted because the individual reflections overlap in time. The basic relationships among rise time, line delay, overshoot and undershoot are shown in a simplified diagram, *Figure 18*. The incident wave is a ramp of amplitude B and rise duration A. The reflection coefficient at the open-ended line output is +1 and the source reflection coefficient is assumed to be -0.8, i.e., $R_0 = Z_0/9$.

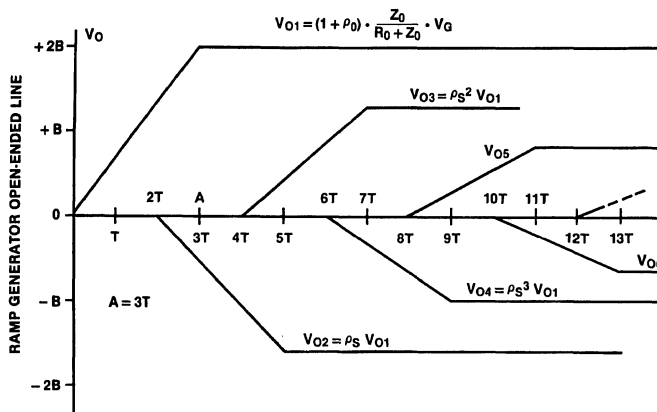
Figure 18b shows the individual reflections treated separately. Rise time A is assumed to be three times the line delay T. The time scale reference is the line output and the first increment of output voltage V_0 rises to 2B in the time interval A. Simultaneously, a positive reflection (not shown) of amplitude B is generated and travels to the source, whereupon it is multiplied by -0.8 and returns toward the end of the line. This negative-going ramp starts at time 2T (twice the line delay) and doubles to -1.6B at time 2T + A.

The negative-going increment also generates a reflection of amplitude -0.8B which makes the round trip to the source and back, appearing at time 4T as a positive ramp rising to +1.28B at time 4T + A. The process of reflection and re-reflection continues, and each successive increment changes in polarity and has an amplitude of 80% of the preceding increment.



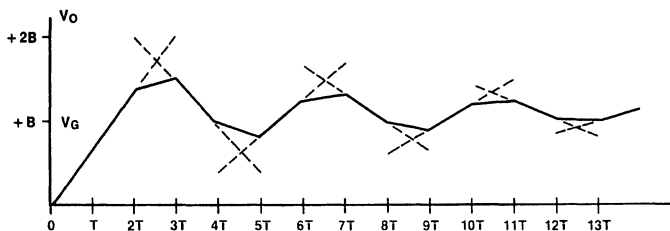
a. Ramp Generator Driving Open-Ended Line

TL/F/10790-47



b. Increments of Output Voltage Treated Individually

TL/F/10790-48



c. Net Output Signal Determined by Superposition

TL/F/10790-49

FIGURE 18. Basic Relationships Involved in Ringing

In *Figure 18c*, the output increments are added algebraically by superposition. The starting point of each increment is shifted upward to a voltage value equal to the algebraic sum of the quiescent levels of all the preceding increments (i.e., 0, 2B, 0.4B, 1.68B, etc.). For time intervals when two ramps occur simultaneously, the two linear functions add to produce a third ramp that prevails during the overlap time of the two increments.

It is apparent from the geometric relationships, that if the ramp time A is less than twice the line delay, the first output increment has time to rise to the full 2B amplitude and the second increment reduces the net output voltage to 0.4B. Conversely, if the line delay is very short compared to the ramp time, the excursions about the final value V_G are small.

Figure 18c shows that the peak of each excursion is reached when the earlier of the two constituent ramps reaches its maximum value, with the result that the first peak occurs at time A. This is because the earlier ramp has a greater slope (absolute value) than the one that follows.

Actual waveforms such as produced by ECL or TTL do not have a constant slope and do not start and stop as abruptly as the ramp used in the example of *Figure 18*. Predicting the time at which the peaks of overshoot and undershoot occur is not as simple as with ramp excitation. A more rigorous treatment is required, including an expression for the driving waveform which closely simulates its actual shape. In the general case, a peak occurs when the sum of the slopes of the individual signal increment is zero.

SUMMARY

The foregoing discussions are by no means an exhaustive treatment of transmission line characteristics. Rather, they

are intended to focus attention on the general methods used to determine the interactions between high-speed logic circuits and their interconnections. Considering an interconnection in terms of distributed rather than lumped inductance and capacitance leads to the line impedance concept, i.e., mismatch between this characteristic impedance and the terminations causes reflections and ringing.

Series termination provides a means of absorbing reflections when it is likely that discontinuities and/or line impedance changes will be encountered. A disadvantage is that the incident wave is only one-half the signal swing, which limits load placement to the end of the line. ECL input capacitance increases the rise time at the end of the line, thus increasing the effective delay. With parallel termination, i.e., at the end of the line, loads can be distributed along the line. ECL input capacitance modifies the line characteristics and should be taken into account when determining line delay.

REFERENCES

1. Metzger, G. and Vabre, J., *Transmission Lines with Pulse Excitation*, Academic Press, (1969).
2. Skilling, H., *Electric Transmission Lines*, McGraw-Hill, (1951).
3. Matick, R., *Transmission Lines for Digital and Communication Networks*, McGraw-Hill, (1969).
4. Millman, J. and Taub, H., *Pulse Digital and Switching Waveforms*, McGraw-Hill, (1965).
5. "Time Domain Reflectometry", *Hewlett-Packard Journal*, Vol. 15, No. 6, (February 1964).
6. Feller, A., Kaupp H., and Digiacom, J., "Crosstalk and Reflections in High-Speed Digital Systems", *Proceedings, Fall Joint Computer Conference*, (1965).

APPENDIX B ECL DESIGN GUIDE: SYSTEM CONSIDERATIONS

INTRODUCTION

All of National's ECL input and output impedances are designed to accommodate various methods of driving and terminating interconnections. Controlled wiring impedance makes it possible to use simplified equivalent circuits to determine limiting conditions. Specific guidelines and recommendations are based on assumed worst-case combinations. Many of the recommendations may seem conservative, compared to typical observations, but the intent is to help the designer achieve a reliable system in a reasonable length of time with a minimum amount of redesign.

PC BOARD TRANSMISSION LINES

Strictly speaking, transmission lines are not always required for F100K ECL but, when used, they provide the advantages of predictable interconnect delays as well as reflection and ringing control through impedance matching. Two common types of PC board transmission lines are microstrip and stripline, *Figure 1*. Stripline requires multilayer construction techniques; microstrip uses ordinary double-clad boards. Other board construction techniques are wire wrap, stitch weld and discrete wired.

Stripline, *Figure 1b*, is used where packing density is a high priority because increasing the interconnect layers provides short signal paths. Boards with as many as 14 layers have been used in ECL systems.

Microstrip offers easier fabrication and higher propagation velocity than stripline, but the routing for a complex system may require more design effort. In *Figure 1a*, the ground plane can be a part of the V_{EE} distribution as long as adequate bypassing from V_{EE} to V_{CC} (ground) is provided. Also, signal routing is simplified and an extra voltage plane is obtained by bonding two microstrip structures back to back, *Figure 1c*.

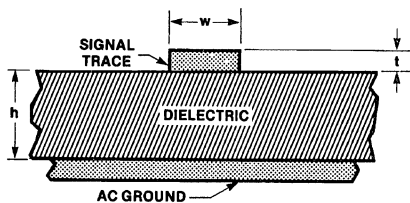
Microstrip

Equation 1 relates microstrip characteristic impedance to the dielectric constant and dimensions.¹ Electric field fringing requires that the ground extend beyond each edge of the signal trace by a distance no less than the trace width.

$$Z_0 = \left(\frac{60}{\sqrt{0.475 \epsilon_r + 0.67}} \right) \ln \left(\frac{4h}{0.67(0.8w + t)} \right) \quad (\text{eq. 1})$$

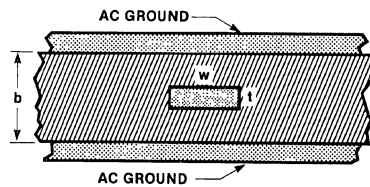
$$= \left(\frac{87}{\sqrt{\epsilon_r + 1.41}} \right) \ln \left(\frac{5.98h}{0.8w + t} \right)$$

where h = dielectric thickness, w = trace width, t = trace thickness, ϵ_r = board material dielectric constant relative to air.



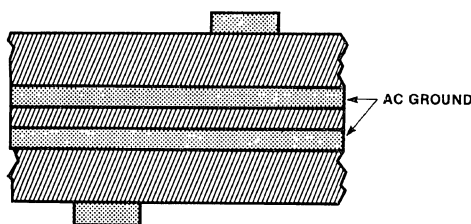
TL/F/10790-50

a. Microstrip



TL/F/10790-51

b. Stripline



TL/F/10790-52

c. Composite Microstrip

FIGURE 1. Transmission Lines on Circuit Boards

Equation 1 was developed from the impedance formula for a wire over ground plane transmission line, Equation 2.

$$Z_0 = \left(\frac{60}{\sqrt{\epsilon_r}} \right) \ln \left(\frac{4h}{d} \right) \quad (\text{eq. 2})$$

where d = wire diameter, h = distance from ground to wire center.

Comparing Equation 1 and 2, the term $0.67 (0.8 w + t)$ shows the equivalence between a round wire and a rectangular conductor. The term $0.475 \epsilon_r + 0.67$ is the *effective dielectric constant* for microstrip ϵ_e , considering that a microstrip line has a compound dielectric consisting of the board material and air. The effective dielectric constant is determined by measuring propagation delay per unit of line length and using the following relationship.

$$\delta = 1.016 \cdot \sqrt{\epsilon_e} \text{ ns/ft} \quad (\text{eq. 3})$$

where δ = propagation delay, ns/ft.

Propagation delay is a property of the dielectric material rather than line width or spacing. The coefficient 1.016 is the reciprocal of the velocity of light in free space. Propagation delay for microstrip lines on glass-filled G-10 epoxy boards is typically 1.77 ns/ft, yielding an effective dielectric constant of 3.04.

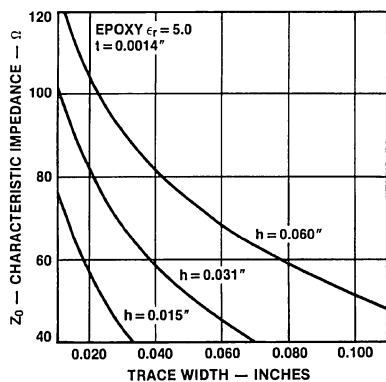


FIGURE 2. Microstrip Impedance Versus Trace Width, G-10 Epoxy

TL/F/10790-53

Using $\epsilon_r = 5.0$ in Equation 1, *Figure 2* provides microstrip line impedance as a function of width for several G-10 epoxy board thicknesses. *Figure 3* shows the related C_0 values, useful for determining capacitive loading effects on line characteristics, (Equation 15).

System designers should ascertain tolerances on board dimensions, dielectric constant and trace width etching in order to determine impedance variations. If conformal coating is used the effective dielectric constant of microstrip is increased, depending on the coating material and thickness.

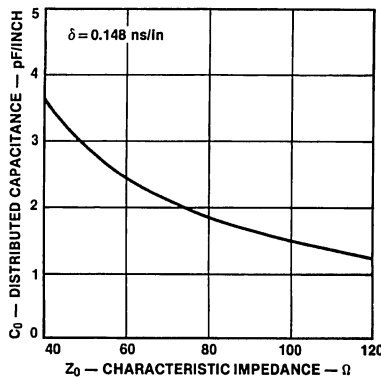


FIGURE 3. Microstrip Distributed Capacitance Versus Impedance, G-10 Epoxy

TL/F/10790-54

Stripline

Stripline conductors are totally embedded. As a result, the board material determines the dielectric constant. G-10 epoxy boards have a typical propagation delay of 2.26 ns/ft. Equation 4 is used to calculate stripline impedances.^{1,2}

$$Z_0 = \left(\frac{60}{\sqrt{\epsilon_r}} \right) \ln \left(\frac{4b}{0.67 \pi (0.8 w + t)} \right) \quad (\text{eq. 4})$$

where b = distance between ground planes, w = trace width, t = trace thickness, $w/(b-t) < 0.35$ and $t/b < 0.25$.

Figure 4 shows stripline impedance as a function of trace width, using Equation 4 and various ground plane separations for G-10 glass-filled epoxy boards. Related values of C_0 are plotted in *Figure 5*.

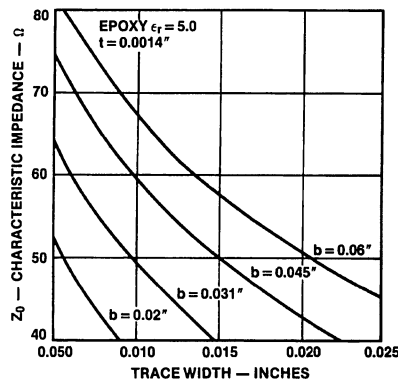
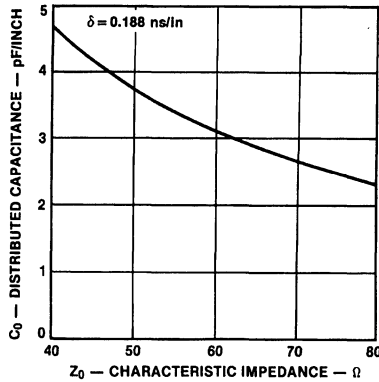


FIGURE 4. Stripline Impedance Versus Trace Width, G-10 Epoxy

TL/F/10790-55



TL/F/10790-56

FIGURE 5. Stripline Distributed Capacitance Versus Impedance, G-10 Epoxy

Wire Wrap

Wire-wrap boards are commercially available with three voltage planes, positions for several 24-pin Dual-In-Line Packages (DIP), terminating resistors, and decoupling capacitors. The devices are mounted on socket pins and interconnected with twisted pair wiring. One wire at each end of the twisted pair is wrapped around a signal pin, the other around a ground pin. The #30 insulated wire is uniformly twisted to provide a nominal 93Ω impedance line. Positions for Single-In-Line Package (SIP) terminating resistors are close to the inputs to provide good termination characteristics.

Stitch Weld

Stitch-weld boards are commercially available with three voltage planes and buried resistors between planes. The devices are mounted on terminals and interconnected with insulated wires that are welded to the backside of the terminals. The insulated wires are placed on a controlled thickness over the ground plane to provide a nominal impedance of 50Ω. The boards are available for both DIPs and flatpaks. Use of flatpaks can increase package density and provide higher system performance.

Discrete Wired

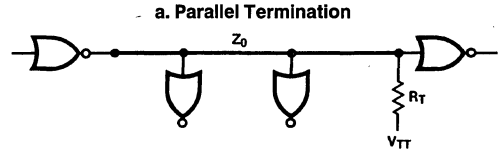
Custom Multiwire* boards are available with integral power and ground planes. Wire is placed on a controlled thickness above the ground plane to obtain a nominal impedance line of 55Ω. Then holes are drilled through the wire and board. Copper is deposited in the drilled holes by an additive-electrolysis process which bonds each wire to the wall of the holes. Devices are soldered on the board to make connection to the wires.

*Multiwire is a registered trademark of the Multiwire Corporation.

Parallel Termination

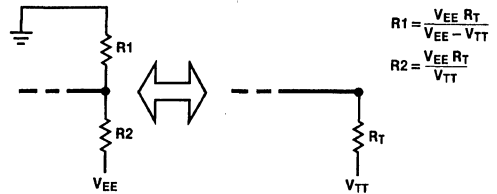
Terminating a line at the receiving end with a resistance equal to the characteristic line impedance is called parallel termination, *Figure 6a*. F100K circuits do not have internal pull-down resistors on outputs, so the terminating resistor must be returned to a voltage more negative than V_{OL} to establish the LOW-state output voltage from the emitter follower. A -2V termination return supply is commonly used. This minimizes power consumption and correlates with standard test specifications for ECL circuits. A pair of resistors connected in series between ground (V_{CC}) and the V_{EE}

supply can provide the Thevenin equivalent of a single resistor to -2V if a separate termination supply is not available, *Figure 6b*. The average power dissipation in the Thevenin equivalent resistors is about 10 times the power dissipation in the single resistor returned to -2V, as shown in *Figures 10 and 13*. For either parallel termination method, decoupling capacitors are required between the supply and ground (Chapter 6).



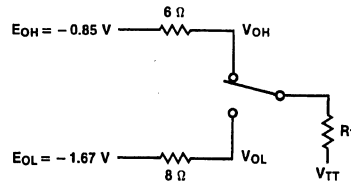
TL/F/10790-57

b. Thevenin Equivalent of R_T and V_{TT}



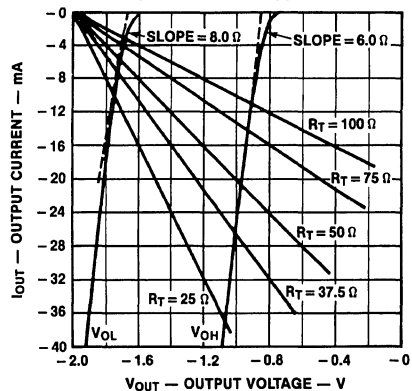
TL/F/10790-58

c. Equivalent Circuit for Determining Approximate V_{OH} and V_{OL} Levels



TL/F/10790-59

d. F100K Output Characteristic with Terminating Resistor R_T Returned to $V_{TT} = -2.0V$



TL/F/10790-60

FIGURE 6. Parallel Termination

F100K output transistors are designed to drive low-impedance loads and have a maximum output current rating of 50 mA. The circuits are specified and tested with a 50Ω load returned to -2V. This gives nominal output levels of -0.955V at 20.9 mA and -1.705V at 5.9 mA. Output levels will be different with other load currents because of the transistor output resistance. This resistance is nonlinear with load current since it is due, in part, to the base-emitter voltage of the emitter follower, which is logarithmic with output current. With the standard 50Ω load, the effective source resistance is approximately 6Ω in the HIGH state and 8Ω in the LOW state.

The foregoing values of output voltage, output current, and output resistance are used to estimate quiescent output levels with different loads. An equivalent circuit is shown in *Figure 6c*. The ECL circuit is assumed to contain two internal voltage sources E_{OH} and E_{OL} with series resistances of 6Ω and 8Ω respectively. The values shown for E_{OH} and E_{OL} are -0.85V and -1.67V respectively.

The linearized portion of the F100K output characteristic can be represented by two equations:

$$\text{For } V_{OH}: V_{OUT} = -850 - 6 I_{OUT}$$

$$\text{For } V_{OL}: V_{OUT} = -1670 - 8 I_{OUT}$$

where I_{OUT} is in mA, V_{OUT} is in mV.

If the range of I_{OUT} is confined between 8 mA to 40 mA for V_{OH} , and 2 mA to 16 mA for V_{OL} , the output voltage can be estimated within ± 10 mV (*Figure 6d*).

An ECL output can drive two or more lines in parallel, provided the maximum rated current is not exceeded. Another consideration is the effect of various loads on noise margins. For example, two parallel 75Ω terminations to -2V (*Figure 6d*) give output levels of approximately -1.000V and -1.716V. Noise margins are thus 35 mV less in the HIGH state and 11 mV more in the LOW state, compared to 50Ω load conditions. Conversely, a single 75Ω load to -2V causes noise margins 38 mV greater in the HIGH state and 11 mV less in the Low state, compared to a 50Ω load.

The magnitude of reflections from the terminated end of the line depends on how well the termination resistance R_T matches the line impedance Z_0 . The ratio of the reflected voltage to the incident voltage V_i is the reflection coefficient ρ .

$$\frac{V_r}{V_i} = \rho = \frac{R_T - Z_0}{R_T + Z_0} \quad (\text{eq. 5})$$

The initial signal swing at the termination is the sum of the incident and reflected voltages. The ratio of termination signal to incident signal is thus:

$$\frac{V_T}{V_i} = 1 + \rho = \frac{2R_T}{R_T + Z_0} \quad (\text{eq. 6})$$

The degree of reflections which can be tolerated varies in different situations, but to allow for worst-case circuits, a good rule of thumb is to limit reflections to 15% to prevent excursions into the threshold region of the ECL inputs connected along the line. The range of permissible values of R_T as a function of Z_0 and the reflection coefficient limitations can be determined by rearranging Equation 5.

$$R_T = Z_0 \frac{1 + \rho}{1 - \rho} \quad (\text{eq. 7})$$

Using 15% reflection limits as examples, the range of the R_T/Z_0 ratio is as follows.

$$\frac{1.15}{0.85} > \frac{R_T}{Z_0} > \frac{0.85}{1.15} \quad 1.35 > \frac{R_T}{Z_0} > 0.74 \quad (\text{eq. 8})$$

The permissible range of the R_T/Z_0 ratio determines the tolerance ranges for R_T and Z_0 . For example, using the foregoing ratio limits, R_T tolerances of $\pm 10\%$ allow Z_0 tolerance limits of +22% and -19%; R_T tolerances of $\pm 5\%$ allow Z_0 tolerance limits of +28% and -23%.

An additional requirement on the maximum value of R_T is related to the value of quiescent I_{OH} current needed to insure sufficient negative-going signal swing when the ECL driver switches from the HIGH state to the LOW state. The npn emitter-follower output of the ECL circuit cannot act as a voltage source driver for negative-going transitions. When the voltage at the base of the emitter follower starts going negative as a result of an internal state change, the output current of the emitter follower starts to decrease. The transmission line responds to the decrease in current by producing a negative-going change in voltage. The ratio of the voltage change to the current change is, of course, the characteristic impedance Z_0 . Since the maximum decrease in current that the line can experience is from I_{OH} to zero, the maximum negative-going transition which can be produced is the product $I_{OH} Z_0$.

If the $I_{OH} Z_0$ product is greater than the normal negative-going signal swing, the emitter follower responds by limiting the current change, thereby controlling the signal swing. If, however, the $I_{OH} Z_0$ product is too small, the emitter follower is momentarily turned off due to insufficient forward bias of its base-emitter junctions, causing a discontinuous negative-going edge such as the one shown in *Figure 14*. In the output-LOW state the emitter follower is essentially non-conducting for V_{OL} values more positive than about -1.55V. Using this value as a criterion and expressing I_{OH} and V_{OH} in terms of the equivalent circuit of *Figure 6c*, an upper limit on the value of R_T can be developed.

$$\Delta V = I_{OH} Z_0 > 1.55 - |V_{OH}|$$

$$\left(\frac{E_{OH} - V_{TT}}{R_0 + R_T} \right) Z_0 > 1.55 - \frac{|V_{TT} R_0 = E_{OH} R_T|}{R_0 + R_T}$$

$$R_T < \frac{(E_{OH} - V_{TT}) Z_0 - (1.55 - |V_{TT}|) R_0}{1.55 - |E_{OH}|} \quad (\text{eq. 9})$$

For a V_{TT} of -2V, R_0 of 6Ω and E_{OH} of -0.85V, Equation 4-9 reduces to

$$R_T < 1.64 Z_0 + 3.86\Omega$$

For $Z_0 = 50\Omega$, the emitter follower cuts off during a negative-going transition if R_T exceeds 86Ω. Changing the voltage level criteria to -1.60V to insure continuous conduction in the emitter follower gives an upper limit of 77Ω for a 50Ω line. For a line terminated at the receiving end with a resistance to -2V, a rough rule-of-thumb is that termination resistance should not exceed line impedance by more than 50%. This insures a satisfactory negative-going signal swing to ECL inputs connected along the line. The quiescent V_{OL} level, after all reflections have damped out, is determined by R_T and the ECL output characteristic.

INPUT IMPEDANCE

The input impedance of ECL circuits is predominately capacitive. A single-function input has an effective value of about 1.5 pF for F100K flatpak, as determined by its effect on reflected and transmitted signals on transmission lines.

In practical calculations, a value of 2 pF should be used. Approximately one third of this capacitance is attributed to the internal circuitry and two thirds to the flatpak pin and internal bonding.

For F100K flatpak circuits, multiple input lines may appear to have up to 3 pF to 4 pF but never more. For example, in the F100102, an input is connected internally to all five gates, but because of the philosophy of buffering these types of inputs in the F100K family this input appears as a unit load with a capacitance of approximately 2 pF. For applications such as a data bus, with two or more outputs connected to the same line, the capacitance of a passive-LOW output can be taken as 2 pF.

Capacitive loads connected along a transmission line increase the propagation delay of a signal along the line. The modified delay can be determined by treating the load capacitance as an increase in the intrinsic distributed capacitance of the line, discussed in Chapter 3. The intrinsic capacitance of any stubs which connect the inputs to the line should be included in the load capacitance. The intrinsic capacitance per unit length for G-10 epoxy boards is shown in Figure 3 and 5 for microstrip and stripline respectively. For other dielectric materials, the intrinsic capacitance C_0 can be determined by dividing the intrinsic delay δ (Equation 3) by the line impedance Z_0 .

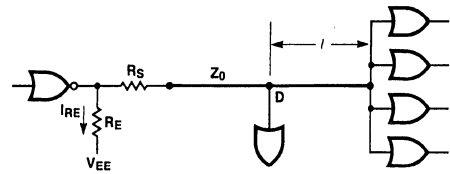
The length of a stub branching off the line to connect an input should be limited to insure that the signal continuing along the line past the stub has a continuous rise, as opposed to a rise (or fall) with several partial steps. The point where a stub branches off the line is a low impedance point. This creates a negative coefficient of reflection, which in turn reduces the amplitude of the incident wave as it continues beyond the branch point. If the stub length is short enough, however, the first reflection returning from the end of the stub adds to the attenuated incident wave while it is still rising. The sum of the attenuated incident wave and the first stub reflection provides a step-free signal, although its rise time will be longer than that of the original signal. Satisfactory signal transitions can be assured by restricting stub lengths according to the recommendations for unterminated lines (Figure 10). The same considerations apply when the termination resistance is not connected at the end of the line; a section of line continuing beyond the termination resistance should be treated as an unterminated line and its length restricted accordingly.

SERIES TERMINATION

Series termination requires a resistor between the driver and transmission line, Figure 7. The receiving end of the line has no termination resistance. The series resistor value should be selected so that when added to the driver source resistance, the total resistance equals the line impedance. The voltage divider action between the net series resistance and the line impedance causes an incident wave of half amplitude to start down the line. When the signal arrives at the unterminated end of the line, it doubles and is thus restored to a full amplitude. Any reflections returning to the source are absorbed without further reflection since the line and source impedance match. This feature, source absorption, makes series termination attractive for interconnection paths involving impedance discontinuities, such as occur in backplane wiring.

A disadvantage of series termination is that driven inputs must be near the end of the line to avoid receiving a 2-step

signal. The initial signal at the driver end is half amplitude, rising to full amplitude only after the reflection returns from the open end of the line. In Figure 7, one load is shown connected at point D, away from the line end. This input receives a full amplitude signal with a continuous edge if the distance l to the open end of the line is within recommended lengths for unterminated line (Figure 10).



TL/F/10790-61

FIGURE 7. Series Termination

The signal at the end has a slower rise time than the incident wave because of capacitive loading. The increase in rise time to the 50% point effectively increases the line propagation delay, since the 50% point of the signal swing is the input signal timing reference point. This added delay as a function of the product line impedance and load capacitance is discussed in Chapter 3.

Quiescent V_{OH} and V_{OL} levels are established by resistor R_E (Figure 7), which also acts with V_{EE} to provide the negative-going drive into R_S and Z_0 when the driver output goes to the LOW state. To determine the appropriate R_E value, the driver output can be treated as a simple mechanical switch which opens to initiate the negative-going swing. At this instant, Z_0 acts as a linear resistor returned to V_{OH} . Thus the components form a simple circuit of R_E , R_S and Z_0 in a series, connected between V_{EE} and V_{OH} . The initial current in this series circuit must be sufficient to introduce a 0.38V transient into the line, which then doubles at the load end to give 0.75V swing.

$$I_{RE} = \frac{V_{OH} - V_{EE}}{R_E + R_S + Z_0} \geq \frac{0.38}{Z_0} \quad (\text{eq. 10})$$

Any I_{OH} current flowing in the line before the switch opens helps to generate the negative swing. This current may be quite small, however, and should be ignored when calculating R_E .

Increasing the minimum signal swing into the line by 30% to 0.49V insures sufficient pull-down current to handle reflection currents caused by impedance discontinuities and load capacitance. The appropriate R_E value is determined from the following relationship.

$$\frac{V_{OH} - V_{EE}}{R_E + R_S + Z_0} \geq \frac{0.49}{Z_0} \quad (\text{eq. 11})$$

For the R_E range normally used, quiescent V_{OH} averages approximately 0.955V and $V_{EE} = -4.5V$. The value of R_S is equal to Z_0 minus R_0 (R_0 averages 7 Ω). Inserting these values and rearranging Equation 11 gives the following.

$$R_E \leq 5.23 Z_0 + 7\Omega \quad (\text{eq. 12})$$

Power dissipation in R_E is listed in Figure 14. The power dissipation in R_E is greater than in R_T of a parallel termination to $-2V$, but still less than the two resistors of the Thevenin equivalent parallel termination, see Figure 10, 13 and 14.

The number of driven inputs on a series terminated line is limited by the voltage drop across R_S in the quiescent HIGH state, caused by the finite input currents of the ECL loads. I_{IH} values are specified on data sheets for various types of

inputs, with a worst-case value of 265 μA for simple gate inputs. The voltage drop subtracts from the HIGH-state noise margin as outlined in *Figure 8a*.

However, there is more HIGH-state noise margin initially, because there is less I_{OH} with the R_E load than with the standard 50 Ω load to -2V . This makes V_{OH} more positive; the increase ranges from 43 mV for a 50 Ω line to 82 mV for a 100 Ω line. Using this V_{OH} increase as a limit on the voltage drop across R_S assures that the HIGH-state noise margin is as good as in the parallel terminated case. Dividing the V_{OH} increase by $R_S + R_0 (= Z_0)$ gives the allowed load input current (I_x in *Figure 8a*). This works out to 0.86 mA for a 50 Ω line, 0.92 mA for a 75 Ω line and 0.82 mA for a 100 Ω line. Load input current greater than these values can be tolerated at some sacrifice in noise margin. If, for example, an additional 50 mV loss is feasible, the maximum values of current become 1.86 mA, 1.59 mA and 1.32 mA for 50 Ω , 75 Ω and 100 Ω lines respectively.

An ECL output can drive more than one series terminated line, as suggested in *Figure 8b*, if the maximum rated output current of 50 mA is not exceeded. Also, driving two or more lines requires a lower R_E value. This makes the quiescent I_{OH} higher and consequently V_{OH} lower, due to the voltage drop across R_0 . This voltage drop decreases the HIGH-state noise margin, which may become the limiting factor (rather than the maximum rated current), depending on the particular application.

The appropriate R_E value can be determined using Equation 13 for $V_{EE} = -4.5\text{V}$.

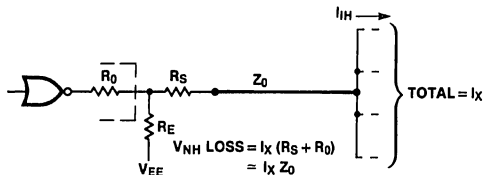
$$\frac{1}{R_E} \geq \frac{1}{6.23 Z_1 - R_{S1}} + \frac{1}{6.23 Z_2 - R_{S2}} + \frac{1}{6.23 Z_3 - R_{S3}} \quad (\text{eq. 13})$$

Circuits with multiple outputs (such as the F100112) provide an alternate means of driving several lines simultaneously (*Figure 8c*). Note, each output should be treated individually when assigning load distribution, line impedance, and R_E value.

UNTERMINATED LINES

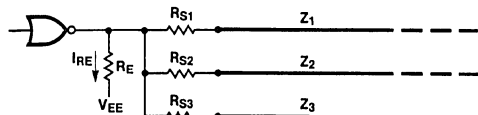
Lines can be used without series or parallel termination if the line delay is short compared to the signal rise time. Ringing occurs because the reflection coefficient at the open (receiving) end of the line is positive (nominally +1) while the reflection coefficient at the driving end is negative (approximately -0.8). These opposite polarity reflection coefficients cause any change in signal voltage to be reflected back and forth, with a polarity change each time the signal is reflected from the driver. Net voltage change on the line is thus a succession of increments with alternating polarity and decreasing magnitude. The algebraic sum of these increments is the observed ringing. The general relationships among rise time, line delay, overshoot and undershoot are discussed in Chapter 3, using simple waveforms for clarity.

Excessive overshoot on the positive-going edge of the signal drives input transistors into saturation. Although this does not damage an ECL input, it does cause excessive recovery times and makes propagation delays unpredictable.



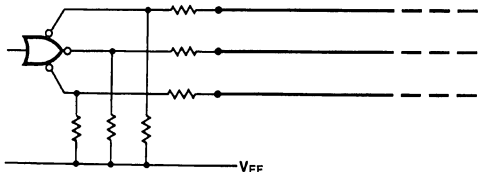
TL/F/10790-62

a. Noise Margin Loss Due to Load Input Current



TL/F/10790-63

b. Driving Several Lines from one Output



TL/F/10790-64

c. Using Multiple Output Element for Load Sharing

FIGURE 8. Loading Considerations for Series Termination

able. Undershoot (following the overshoot) must also be limited to prevent signal excursions into the threshold region of the loads. Such excursions could cause exaggerated transition times at the driven circuit outputs, and could also cause multiple triggering of sequential circuits. Signal swing, exclusive of ringing, is slightly greater on unterminated lines than on parallel terminated lines; I_{OH} is less and I_{OL} is greater with the R_E load, (*Figure 9a*) making V_{OH} higher and V_{OL} lower.

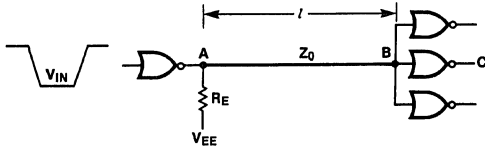
For worst case combinations of driver output and load input characteristics, a 35% overshoot limit insures that system speed is not compromised either by saturating an input on overshoot or extending into the threshold region on the following undershoot.

For distributed loading, ringing is satisfactorily controlled if the 2-way modified line delay does not exceed the 20% to 80% rise time of the driver output. This relationship can be expressed as follows, using the symbols from Chapter 3 and incorporating the effects of load capacitance on line delay.

$$t_r = 2T' = 2\ell \delta' = 2\ell \delta \sqrt{1 + \frac{C_L}{\ell C_0}}$$

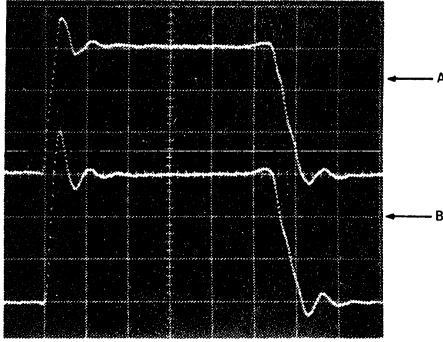
Solving this expression for the line length (ℓ):

$$\ell_{\max} = \frac{1}{2} \sqrt{\left(\frac{C_L}{C_0}\right)^2 + \left(\frac{t_r}{\delta}\right)^2} - \frac{C_L}{2C_0} \quad (\text{eq. 14})$$



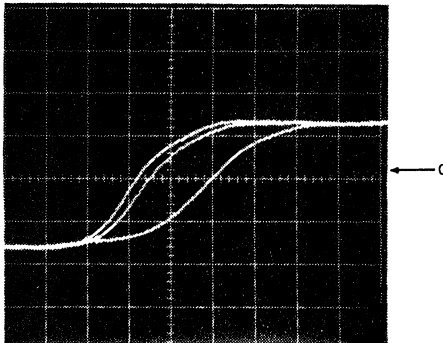
a. Unterminated Line

TL/F/10790-65



b. Line Voltages Showing Stair-step Trailing Edges

TL/F/10790-66



c. Load Gate Output Showing Net Propagation Increase for Increasing Values of R_E : 330 Ω , 510 Ω , 1 k Ω

TL/F/10790-67

FIGURE 9. Effect on R_E Value on Trailing-Edge Propagation

The shorter the rise time, the shorter the permissible line length. For F100K ECL, the minimum rise time from 20% to 80% is specified as 0.5 ns. Using this rise time and 2 pF per fan-out load, calculated maximum line lengths for G-10 epoxy microstrip are listed in Figure 10a. The length (l) in the table is the distance from the terminating resistor to the input of the device(s). For F100K ECL the case described in Figure 10a is the only one calculated, since all other combinations are approximately the same. For other combinations of rise time, impedance, fan-out or line characteristics

(δ and C_0), maximum lengths are calculated using Equation 14. For the convenience of those who are also using 10K ECL, maximum recommended lengths of unterminated lines are listed in Figure 10b to 10e.

Z_0	Number of Fan-Out Loads			
	1	2	3	4
50	1.37*	1.13	0.95	0.81
62	1.33	1.07	0.87	0.70
75	1.25	0.95	0.75	0.61
90	1.18	0.85	0.66	0.53
100	1.15	0.82	0.61	0.49

*Length in inches.
Unit load = 2 pF, δ = 0.148 ns/inch

FIGURE 10a. F100K Maximum Worst-Case Line Lengths for Unterminated Microstrip, Distributed Loading

Z_0	Number of Fan-Out Loads				
	2	3	4	6	8
50	4.15*	3.75	3.45	2.85	2.45
62	3.95	3.50	3.15	2.55	2.10
75	3.75	3.25	2.85	2.25	1.85
90	3.55	3.00	2.60	2.00	1.60
100	3.45	2.85	2.45	1.85	1.45

*Length in inches.
Unit load = 3 pF, δ = 0.148 ns/in.

FIGURE 10b. 10K Maximum Worst-Case Line Lengths for Unterminated Microstrip, Distributed Loading

Z_0	Number of Fan-Out Loads				
	1	2	4	6	8
50	4.40*	3.65	2.60	1.90	1.40
62	4.30	3.45	2.30	1.60	1.15
75	4.20	3.20	2.05	1.40	0.95
90	4.05	2.95	1.75	1.05	0.65
100	3.90	2.80	1.60	0.90	0.50

*Length in inches.
Unit load = 3 pF, δ = 0.148 ns/in.

FIGURE 10c. 10K Maximum Worst-Case Line Lengths for Unterminated Microstrip, Concentrated Loading

Z_0	Number of Fan-Out Loads				
	2	3	4	6	8
50	3.30*	3.00	2.70	2.25	2.90
62	3.15	2.80	2.50	2.00	1.65
75	3.00	2.60	2.25	1.80	1.45
90	2.80	2.40	2.05	1.55	1.25

*Length in inches.
Unit load = 3 pF, δ = 0.188 ns/in.

FIGURE 10d. 10K Maximum Worst-Case Line Lengths for Unterminated Stripline, Distributed Loading

Z_0	Number of Fan-Out Loads				
	1	2	4	6	8
50	3.45*	2.85	2.00	1.50	1.10
62	3.40	2.70	1.80	1.30	0.90
75	3.30	2.55	1.60	1.10	0.75
90	3.15	2.35	1.40	0.85	0.50
100	3.10	2.20	1.25	0.70	0.40

*Length in inches.
Unit load = 3 pF, $\delta = 0.188$ ns/in.

FIGURE 10e. 10K Maximum Worst-Case Line Lengths for Unterminated Stripline, Concentrated Loading

A load capacitance concentrated at the end of the line restricts line length more than a distributed load does. Maximum recommended lengths for fiberglass epoxy dielectric and a 0.5 ns rise time are listed in *Figure 10* for microstrip. For line impedances not listed, linear interpolation can be used to determine appropriate line lengths. Appropriate line lengths for dielectric materials with a different propagation constant δ can be determined by multiplying the listed values by the fiberglass epoxy δ and then dividing by the δ of the other material. For example, a line length for a material which has a microstrip δ of 0.1 ns/inch is determined by multiplying the length given in the microstrip table (for a desired impedance and load) by 0.148 and dividing by 0.1.

Resistor R_E must provide the current for the negative-going signal at the driver output. Line input and output waveforms are noticeably affected if R_E is too large, as shown in *Figure 9b*. The negative-going edge of the signal falls in stair-step fashion, with three distinct steps visible at point A. The waveform at point B shows a step in the middle of the negative-going swing. The effect of different R_E values on the net propagation time through the line and the driven loads is evident in *Figure 9c* which shows the output signal of one driven gate in a multiple exposure photograph. The horizontal sweep (time axis) was held constant with respect to the input signal of the driver. The earliest of the three output signals occurs with an R_E value of 330 Ω . Changing R_E to 510 Ω increases the net propagation delay by 0.3 ns, the horizontal offset between the first and second signals. Changing R_E to 1 k Ω produces a much greater increase in net propagation delay, indicating that the negative-going signal at B contains several steps. In practice, a satisfactory negative-going signal results when the R_E value is chosen to give an initial negative-going step of 0.6V at the driving end of the line. This gives an upper limit on the value of R_E , as shown in Equation 15.

$$\text{initial step} = \Delta I \cdot Z_0 = \frac{(V_{OH} - V_{EE}) Z_0}{R_E + Z_0} \geq 0.6$$

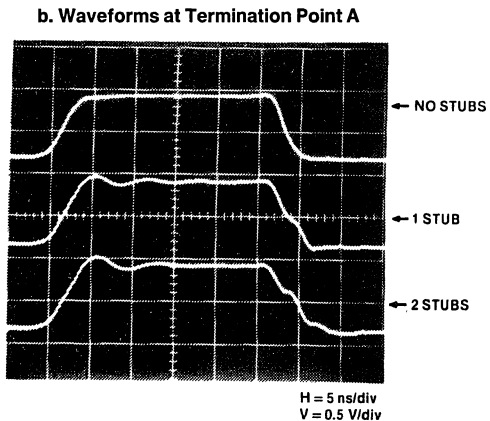
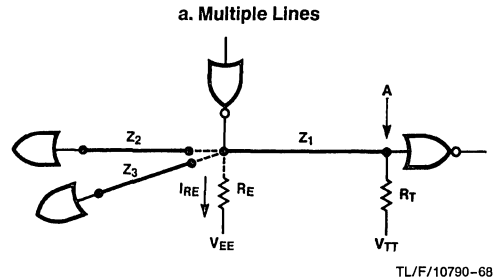
$$R_E \leq 6.25 Z_0 \quad (\text{eq. 15})$$

An ECL output can drive two or more unterminated lines, provided each line length and loading combination is within the recommended constraints. The appropriate R_E value is determined from Equation 15, using the parallel impedance of the two or more lines for Z_0 .

An ECL output can simultaneously drive terminated and unterminated lines, although the negative-going edge of the signal shows two or more distinct steps when the stubs are long unless some extra pull-down current is provided. *Figure 11a* shows an ECL circuit driving a parallel terminated line, with provision for connecting two worst-case unterminated

lines to the driver output. Waveforms at the termination resistor (point A) are shown in the multiple exposure photograph of *Figure 11b*. The upper trace shows a normal signal without stubs connected to the driver. The middle trace shows the effect of connecting one stub to the driver. The step in the negative-going edge indicates that the quiescent I_{OH} current through R_T is not sufficient to cause a full signal for both lines. The relationship between the quiescent I_{OH} current through R_T and the negative-going signal swing was discussed earlier in connection with parallel termination.

The bottom trace in *Figure 11* shows the effect of connecting two stubs to the driver output. The steps in trailing edge are smaller and more pronounced. The deteriorated trailing edge of either the middle or lower waveform increas-



c. Equivalent Circuit for Determining Initial Negative Voltage Step at the Driver Output

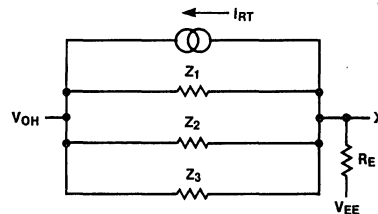


FIGURE 11. Driving Terminated and Unterminated Lines in Parallel

es the switching time of the circuit connected to point A. If this extra delay cannot be tolerated, additional pull-down current must be provided. One method uses a resistor to V_{EE} as suggested in *Figure 11a*. The initial negative-going step at point A should be about 0.7V to insure a good fall rate through the threshold region of the driven gate. The initial step at the driver output should also be 0.7V. If the driver output is treated as a switch that opens to initiate the negative-going signal, the equivalent circuit of *Figure 11c* can be used to determine the initial voltage step at the driver output (point X). The value of the current source I_{RT} is the quiescent I_{OH} current through R_T . Using Z' to denote the parallel impedance of the transmission lines and ΔV for the desired voltage step at X, the appropriate value of R_E can be determined from the following equation, using absolute values to avoid polarity confusion.

$$R_E = \left(|V_{EE}| - |V_{OH}| - \Delta V \right) \cdot \left(\frac{Z'}{|\Delta V| - |I_{RT}Z'|} \right)$$

For a sample calculation, assume that R_T and the line impedances are each 100 Ω , V_{OH} is -0.955V, ΔV is 0.750V, V_{EE} is -4.5V and V_{TT} is -2V. I_{RT} is thus 10.45 mA and the calculated value of R_E is 232 Ω . In practice, this value is on the conservative side and can be increased to the next larger (10%) standard value with no appreciable sacrifice in propagation through the gate at point A.

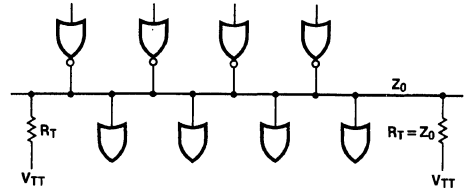
Again, the foregoing example is based on worst-case stub lengths (the longest permissible). With shorter stubs, the effects are less pronounced and a point is reached where extra pull-down current is not required because the reflection from the end of the stub arrives back at the driver while the original signal is still falling. Since the reflection is also negative going, it combines with and reinforces the falling signal at the driver, eliminating the steps. The net result is a smoothly falling signal but with increased fall time compared to the stubless condition.

The many combinations of line impedance and load make it practically impossible to define just with stub length begins to cause noticeable steps in the falling signal. A rough rule-of-thumb would be to limit the stub length to one-third of the values given in *Figure 10*.

DATA BUSSING

Data bussing involves connecting two or more outputs and one or more inputs to the same signal line, (*Figure 12*). Any one of the several drivers can be enabled and can apply data to the line. Load inputs connected to the line thus receive data from the selected source. This method of steering data from place to place simplifies wiring and tends to minimize package count. Only one of the drivers can be enabled at a given time; all other driver outputs must be in the LOW state. Termination resistors matching the line impedance are connected to both ends of the line to prevent reflections. For calculating the modified delay of the line (*Chapter 3*) the capacitance of a LOW (unselected) driver output should be taken as 2 pF.

An output driving the line sees an impedance equal to half the line impedance. Similarly, the quiescent I_{OH} current is higher than with a single termination. For line impedance less than 100 Ω , the I_{OH} current is greater than the data sheet test value, with a consequent reduction of HIGH-state noise margin. This loss can be eliminated if necessary by



TL/F/10790-71

FIGURE 12. Data Bus or Party Line

using multiple output gates (F100112) and paralleling two outputs for each driver. In the quiescent LOW state, termination current is shared among all the output transistors on the line. This sharing makes V_{OL} more positive than if only one output were conducting all of the current. For example, a 100 Ω line terminated at both ends represents a net 50 Ω DC load, which is the same as the data sheet condition for V_{OL} . If one worst-case output were conducting all the current, the V_{OL} would be -1.705V. If another output with identical DC characteristics shares the load current equally, the V_{OL} level shifts upward by about 25 mV. Connecting two additional outputs for a total of four with the same characteristics shifts V_{OL} upward another 22 mV. Connecting four more identical outputs shifts V_{OL} upward another 20 mV. Thus the V_{OL} shift for eight outputs having identical worst-case V_{OL} characteristics is approximately 67 mV. In practice, the probability of having eight circuits with worst-case V_{OL} characteristics is quite low. The output with the highest V_{OL} tends to conduct most of the current. This limits the upward shift to much less than the theoretical worst-case value. In addition, the LOW-state noise margin is specified greater than the HIGH-state margin to allow for V_{OL} shift when outputs are paralleled.

In some instances a single termination is satisfactory for a data bus, provided certain conditions are fulfilled. The single termination is connected in the middle of the line. This requires that for each half of the line, from the termination to the end, the line length and loading must comply with the same restrictions as unterminated lines to limit overshoot and undershoot to acceptable levels. The termination should be connected as near as possible to the electrical mid-point of the line, in terms of the modified line delay from the termination to either end. Another restriction is that the time between successive transitions, i.e., the nominal bit time, should not be less than 15 ns. This allows time for the major reflections to damp out and limits additive reflections to a minor level.

WIRED-OR

In general-purpose wired-OR logic connections, where two or more driver outputs are expected to be in the HIGH state simultaneously, it is important to minimize the line length between the participating driver outputs, and to place the termination as close as possible to the mid-point between the two most widely separated sources. This minimizes the negative-going disturbances which occur when one HIGH output turns off while other outputs remain HIGH. The driver output going off represents a sudden decrease in line current, which in turn generates a negative-going voltage on the line. A finite time is required for the other driver outputs (quiescently HIGH) to supply the extra current. The net re-

sult is a "V" shaped negative glitch whose amplitude and duration depend on three factors: current that the off-going output was conducting, the line impedance, and the line length between outputs. If the separation between outputs is kept within about one inch, the transient will not propagate through the driven load circuits.

If a wired-OR connection cannot be short, it may be necessary to design the logic so that the signal on the line is not sampled for some time after the normal propagation delay (output going negative) of the element being switched. Normal propagation delay is defined as the case where the element being switched is the only one on the line in the HIGH state, resulting in the line going LOW when the element switches. In this case, the propagation delay is measured from the 50% point on the input signal of the off-going element to the 50% point of the signal at the input farthest away from the output being switched. The extra wiring time required in the case of a severe negative glitch is, in a worst-case physical arrangement, twice the line delay between the off-going output and the nearest quiescently HIGH output, plus 2 ns.

An idea of how the extra waiting time varies with physical arrangement can be obtained by qualitatively comparing the signal paths in *Figure 13*. With the outputs at A and B quiescently HIGH, the duration of the transient observed at C is longer if B is the off-going output than if A is the off-going element. This is because the negative-going voltage generated at B must travel to A, whereupon the corrective signal is generated, which subsequently propagates back toward C. Thus the corrective signal lags behind the initial transient, as observed at C, by twice the line delay between A and B. On the other hand, if the output at A generates the negative-going transient, the corrective response starts when the

transient reaches point B. Consequently, the transient duration observed at C is shorter by twice the line delay from A to B.

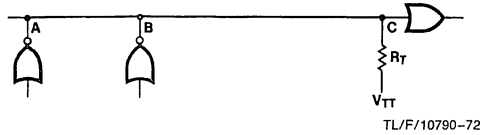
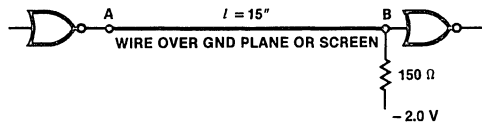


FIGURE 13. Relative to Wired-OR Propagation

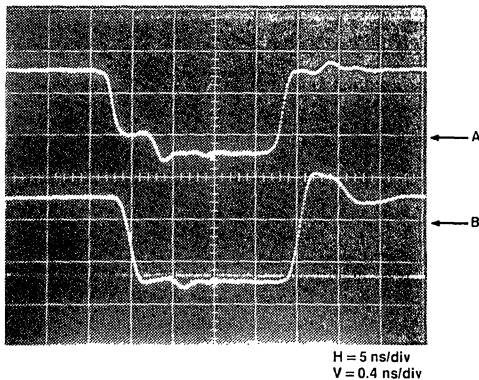
BACKPLANE INTERCONNECTIONS

Several types of interconnections can be used to transmit a signal between logic boards. The factors to be considered when selecting a particular interconnection for a given application are cost, impedance discontinuities, predictability of propagation delay, noise environment, and bandwidth. Single-ended transmission over an ordinary wire is the most economical but has the least predictable impedance and propagation delay. At the opposite end of the scale, coaxial cable is the most costly but has the best electrical characteristics. Twisted pair and similar parallel wire interconnection cost and quality fall in between.

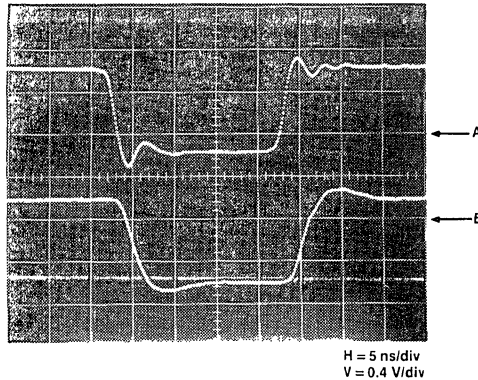
For single-wire transmission through the backplane, a ground plane or ground screen (Chapter 5) should be provided to establish a controlled impedance. A wire over a ground plane or screen has a typical impedance of 150Ω with variations on the order of ±33%, depending primarily on the distance from ground and the configuration of the ground. *Figure 14* illustrates the effects of impedance variations with a 15-inch wire parallel terminated with 150Ω to -2V. *Figure 14b* shows source and receiver waveforms when the wire is in contact with a continuous ground plane.



a. Wire over Ground Plane or Screen



b. Wire in Contact with Ground Plane



c. Wire Spaced 1/8" from Ground Screen

FIGURE 14. Parallel Terminated Backplane Wire

The negative-going signal at the source shows an initial step of only 80% of a full signal swing. This occurs because the quiescent HIGH-state current I_{OH} (about 7 mA) multiplied by the impedance of the wire (approximately 90Ω) is less than the normal signal swing, and this condition allows the driver emitter follower to turn off. The negative-going signal at the receiving end is greater by 25% ($1 + \rho = 1.25$). The receiving end mismatch causes a negative-going reflection which returns to the source and establishes the V_{OL} level. The positive-going signal at the source shows a normal signal swing, with the receiving end exhibiting approximately 25% overshoot.

Figure 14c shows waveforms for a similar arrangement, but with the wire about $\frac{1}{8}$ inch from a ground screen. The impedance of the wire is greater than 150Ω termination, but small variations in impedance along the wire cause intermediate reflections which tend to lengthen the rise and fall times of the signal. As a result, the received signal does not exhibit pronounced changes in slope as would be expected if a 200Ω constant impedance line were terminated with 150Ω .

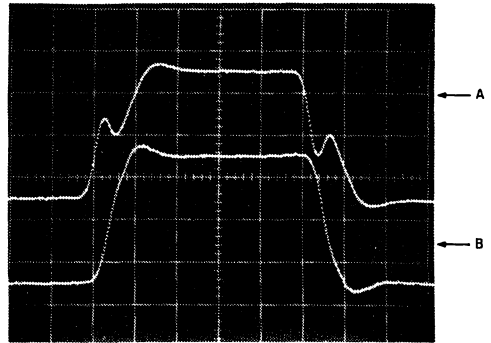
Series source resistance can also be used with single wire interconnections to absorb reflection. Figure 15a shows a 16-inch wire with a ground screen driven through a source resistance of 100Ω . The waveforms (Figure 15b) show that although reflections are generated, they are largely absorbed by the series resistor, and the signal received at the load exhibits only slight changes and overshoot. Series termination techniques can also be used when the signal into the wire comes from the PC board transmission line. Figure 16a illustrates a 12-inch wire over a ground screen, with 12-inch microstrip lines at either end of the wire. The output is heavily loaded (fan-out of 8) and the combination of impedances produces a variety of reflections at the input to the first microstrip line, shown in the upper trace of Figure 16b. The lower trace shows the final output; a comparison between the two traces shows the effectiveness of damping in maintaining an acceptable signal at the output. Figure 16c shows the signals at the input to the driving gate and at the output of the load gate, with a net through-put time of 8.5 ns. The circuit in Figure 16a is a case of mismatched transmission lines, discussed in Appendix A.

Signal propagation along a single wire tends to be fast because the dielectric medium is mostly air. However, impedance variations along a wire cause intermediate reflections which tend to increase rise and fall times, effectively increasing propagation delay. Effective propagation delays are in the range of 1.5 to 2.0 ns per foot of wire. Load capacitance at the receiving end also increases rise and fall time (Appendix A), further increasing the effective propagation delay.



TL/F/10790-76

a. Wire over Ground Screen

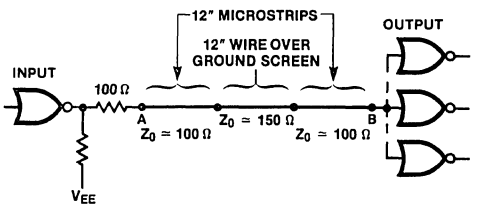


H = 10 ns/div
V = 0.3 V/div

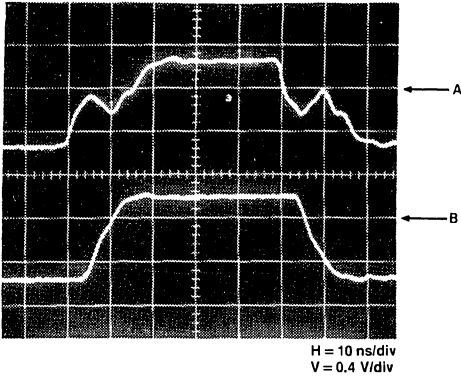
TL/F/10790-77

b. Series Terminated Waveform

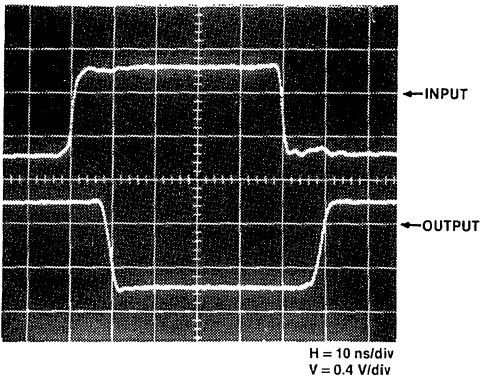
FIGURE 15. Series Terminated Backplane Wire



a. Backplane Wire Interconnecting PC Board Lines
TL/F/10790-78



b. Signals into the First Microstrip and at the Loads
TL/F/10790-79

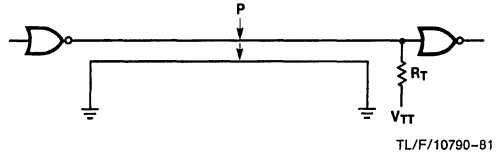


c. Input to Driving Gate and Output of Load Gate
TL/F/10790-80

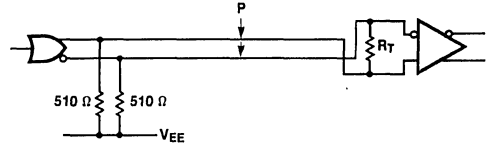
FIGURE 4-16. Signal Path with Sequence of Microstrip, Wire, Microstrip

Better control of line impedance and faster propagation can be achieved with a twisted pair. A twisted pair of AWG 26 Teflon* insulated wires, two twists per inch, exhibits a propagation delay of 1.33 ns/ft and an impedance of 115Ω. Twisted pair lines are available in a variety of sizes, impedances and multiple-pair cables. *Figure 4-17a* illustrates sin-

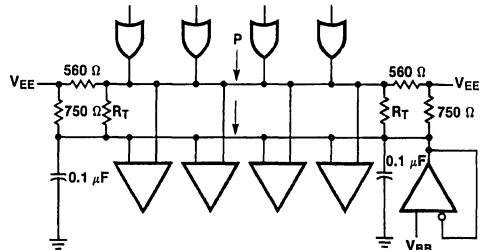
*Teflon is a registered trademark of E.I. du Pont de Nemours Company.



a. Single-ended Twisted Pair
TL/F/10790-81



b. Differential Transmission Reception
TL/F/10790-82



c. Backplane Data Bus
TL/F/10790-83

FIGURE 4-17. Twisted Pair Connections

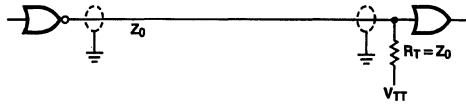
gle-ended driving and receiving. In addition to improved propagation velocity, the magnetic fields of the two conductors tend to cancel, minimizing noise coupled into adjacent wiring.

Differential line driving and receiving complementary gates as the driver and an F100114 line receiver is illustrated in *Figure 4-17b*. Differential operation provides high noise immunity, since common mode input voltages between -0.55V and -3.0V are rejected. The differential mode is recommended for communication between different parts of a system, because it effectively nullifies ground voltage differences. For long runs between cabinets or near high power transients, interconnections using shielded twisted pair are recommended.

Twisted pair lines can be used to implement party line type data transfer in the backplane, as indicated in *Figure 4-17c*. Only one driver should be enabled at a given time; the other outputs must be in the VOL state. The VBB reference voltage is available on pin 22 of the flatpak and pin 19 of the dual-in-line package for the F100114.

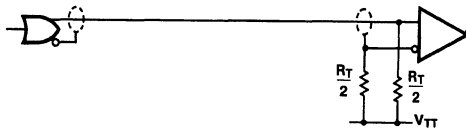
In the differential mode, a twisted pair can send high-frequency symmetrical signals, such as clock pulses, of 100 MHz over distances of 50 to 100 feet. For random data, however, bit rate capability is reduced by a factor of four or five due to line rise effects on time jitter.³

Coaxial cable offers the highest frequency capability. In addition, the outer conductor acts as a shield against noise, while the uniformity of characteristics simplifies the task of matching time delays between different parts of the system. In the single-ended mode, *Figure 4-18a*, 50 MHz signals can be transferred over distances of 100 feet. For 100 MHz operation, lengths should be 50 feet or less. In the differential mode, *Figures 4-18b, c*, the line receiver can recover smaller signals, allowing 100 MHz signals to be transferred up to 100 feet. The dual cable arrangement of *Figure 4-18c* provides maximum noise immunity. The delay of coaxial cables depends on the type of dielectric material, with typical delays of 1.52 ns/ft for polyethylene and 1.36 ns/ft for cellular polyethylene.



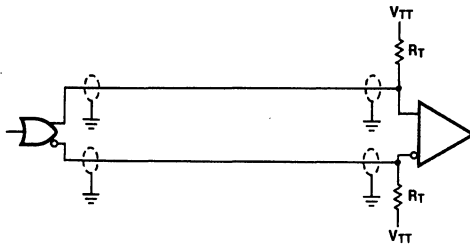
TL/F/10790-84

a. Single-Ended Coaxial Transmission



TL/F/10790-85

b. Differential Coaxial Transmission



TL/F/10790-86

c. Differential Transmission with Grounded Shields

FIGURE 4-18. Coaxial Cable Connections

REFERENCES

1. Kaupp, H. R., "Characteristics of Microstrip Transmission Lines," *IEEE Transaction on Electronic Computers*, Vol. EC-16 (April, 1967).
2. Harper, C. A., *Handbook of Wiring, Cabling and Interconnections for Electronics*. New York: McGraw-Hill, 1972.
3. True, K. M., "Transmission Line Interface Elements," *The TTL Applications Handbook*, Chapter 14 (August 1973), pp. 14-1-14-14.

APPENDIX C ECL DESIGN GUIDE: POWER DISTRIBUTION AND THERMAL CONSIDERATIONS

INTRODUCTION

High-speed circuits generally consume more power than similar low-speed circuits. At the system level, this means that the power supply distribution system must handle the larger current flow; the larger power dissipation places a greater demand on the cooling system. The direct current (DC) voltage drop along ground busses affects noise margins for all types of ECL circuits. Voltage drops along V_{EE} busses have only a slight effect on F100K circuits, but they require consideration to obtain the performance available from the family.

LOGIC CIRCUIT GROUND, V_{CC}

The positive potential V_{CC} and V_{CCA} in ECL circuits is the reference voltage for output voltages and input thresholds and should therefore be the ground potential. When two circuits are connected in a single-ended mode, any difference in ground potentials decreases the noise margins, as discussed in *Chapter 1*. This effect for TTL/DTL circuits, as well as for ECL circuits, is illustrated in *Figure 1*. The following analysis assumes some average value of current flowing through the distributed resistance along the ground path between two circuits. For the indicated direction of I_G , the shift in ground potential *decreases* the LOW-state noise margin of the TTL/DTL circuits and the HIGH-state noise margin of the ECL circuits. If I_G is flowing in the opposite direction, it *increases* these noise margins, but *decreases* the noise margins when the drivers are in the opposite state. For tabulation of ground currents in ECL, the designs must include termination currents as well as I_{EE} operating currents. ECL logic boards which use microstrip or stripline techniques generally have large areas of ground metal. This causes the ground resistance to be quite low and thus minimizes noise margin loss between pairs of circuits on the same board.

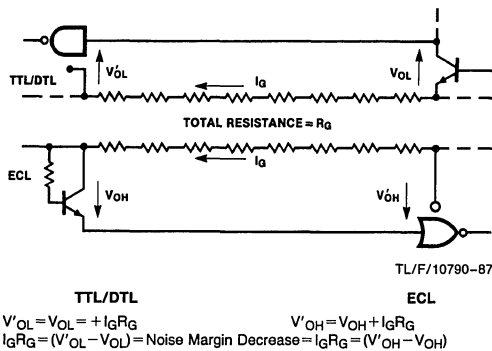
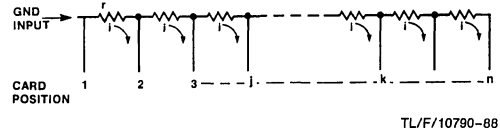


FIGURE 1. Effect of Ground Resistance on Noise Margins

In practice, two communicating circuits might be located on widely separated PC cards with other PC cards in between. The net resistance then includes the incremental resistance of the ground distribution bus from card to card, while the ground current is successively increased by the contribution from each card. *Figure 2* illustrates a distribution bus for a row of cards with incremental resistances along the bus.



r = Incremental Bus Resistance between Positions
 i = Average Ground Current per Card

FIGURE 2. Ground Shift Along a Row of PC Cards

The ground shift can be estimated by first determining an average value of current per card based on the number of packages, the mix of SSI and MSI, and the number and types of terminations. With n cards in the row, an average ground current (i) per card, and an incremental bus resistance (r) between card positions, the bus voltage drops between the various positions can be determined as follows:

$$\begin{aligned} \text{between positions 1 and 2: } v_{1-2} &= (n-1)ir \\ \text{between positions 1 and 3: } v_{1-3} &= (n-1)ir + (n-2)ir \\ \text{between positions 1 and 4: } v_{1-4} &= (n-1)ir + (n-2)ir + (n-3)ir \\ \text{between 1 and } n: & v_{1-n} = ir \{ (n-1) + (n-2) + (n-3) + \dots + [n - (n-1)] \} \\ &= ir [1 + 2 + 3 + \dots + (n-1)] \\ &= ir \sum_{1}^{n-1} n \end{aligned}$$

For a row of 15 cards, for example, the total ground shift between positions 1 and 15 is expressed as in Equation 1.

$$\begin{aligned} v_{1-15} &= ir \sum_{1}^{14} n = ir (1 + 2 + 3 + \dots + 13 + 14) \\ &= 105 ir \end{aligned} \tag{eq. 1}$$

The ground shift between any two card positions j and k can be determined as follows for the general case.

$$\begin{aligned} v_{j-k} &= (n-j)ir + [n-(j+1)]ir + \\ &\quad [n-(j+2)]ir \\ &\quad + \dots + \{n-[j+(k-j-1)]\}ir \\ &= (k-j)nir - ir\{j+(j+1)+(j+2) \\ &\quad + \dots + [j+(k-j-1)]\} \end{aligned} \quad (\text{eq. 2})$$

$$v_{j-k} = (k-j)nir - ir \sum_j^{k-1} n = ir[(k-j)n - \sum_j^{k-1} n]$$

In a row of 15 cards, the ground shift between positions four and nine, for example, is determined as follows.

$$\begin{aligned} v_{j-k} &= ir[(9-4)15 - (4+5+6+7+8)] \\ &= ir(75-30) = 45ir \end{aligned} \quad (\text{eq. 3})$$

The ground shift between the same number of positions further down the row is less because of the decreasing current along the row. Consider the ground shift between card positions 10 and 15.

$$\begin{aligned} v_{10-15} &= ir[(15-10)15 - \\ &\quad (10+11+12+13+14)] \\ &= ir(75-60) = 15ir \end{aligned} \quad (\text{eq. 4})$$

These examples illustrate several principles the designer should consider regarding the ground distribution bus and assignment of card positions. The bus resistance should be kept as low as possible by making the cross-sectional areas as large as practical. Logic cards which represent the heaviest current drain should be located nearest the end where ground comes into the row of cards. Cards with single-ended logic wiring between them should be assigned to positions as close together as possible. Conversely, if the ground shift between two card positions represents an unacceptable loss of noise margin, then the differential transmission and reception method i.e., twisted pair, should be used for logic wiring between them, thereby eliminating ground shift as a noise margin factor.

CONDUCTOR RESISTANCES

Conductors with large cross-sectional areas are required to maintain low voltage drops along power busses. For convenience, *Figure 3* lists the resistance per foot and the cross-sectional area for more common sizes of annealed copper wire. Other characteristics and a complete list of sizes can be found in standard wire tables. A useful rule-of-thumb regarding resistances and, hence, areas is: as gauge numbers increase, resistance doubles with every third gauge number; e.g., the resistance per foot of #10 wire is 1 m Ω , for #13 wire it is 2 m Ω . Similarly, the resistance per foot of #0 wire is 0.078 m Ω , which is half that of #2 wire.

For calculations involving conductors having rectangular cross sections, it is often convenient to work with sheet resistance, particularly for power distribution on PC cards. Copper resistivity is usually given in ohm-centimeters, indicating the resistance between opposing faces of a 1 cm cube. The sheet resistance of a conductor is obtained by dividing the resistivity by the conductor thickness. These relationships follow.

AWG B & S Gauge	Resistance m Ω Per Foot	Cross-Sectional Area Square Inches
#2	0.156	$5,213 \times 10^{-2}$
#6	0.395	$2,062 \times 10^{-2}$
#10	0.999	$8,155 \times 10^{-3}$
#12	1.588	$5,129 \times 10^{-3}$
#18	6.385	$1,276 \times 10^{-3}$
#22	16.14	$5,046 \times 10^{-4}$
#26	40.81	$1,996 \times 10^{-4}$
#30	103.2	$7,894 \times 10^{-5}$

FIGURE 3. Resistance and Cross-Sectional Area of Several Sizes of Annealed Copper Wire

Copper resistivity = $\rho = 1.724 \times 10^{-6} \Omega\text{cm}$ @ 20°C

$$\text{Resistance of a conductor} = \rho \frac{l}{A} = \rho \frac{l}{tw}$$

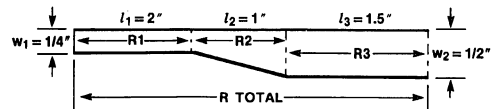
where: l = length t = thickness w = width

$$\text{Sheet resistance } \rho_S = \frac{\rho}{t} \Omega \text{ per } \frac{l}{w}$$

The length/width ratio (l/w) is dimensionless; therefore, the resistance of a length of conductor of uniform thickness can be calculated by first determining the number of "squares," then multiplying by the sheet resistance. For example, a conductor one-eighth inch wide and three inches long has 24 squares; its resistance is 24 times the sheet resistance. Since many thickness dimensions are given in inches, it is convenient to express the resistivity in ohm-inch, as follows.

$$\rho(\Omega\text{in.}) = \rho(\Omega\text{cm}) \div 2.54 = 6.788 \times 10^{-7} \Omega\text{in.}$$

The use of sheet resistance and the "squares" concept is illustrated by calculating the resistance of the conductor shown in *Figure 4*. Assume the conductor is a 1 oz. copper cladding with a 0.0012 inch minimum thickness on a PC card.



TL/F/10790-89

FIGURE 4. Conductor of Uniform Thickness but Non-Uniform Cross Section

$$\begin{aligned} \text{Sheet resistance} &= \rho_S = \frac{\rho}{t} \\ &= 5.657 \times 10^{-4} \Omega \text{ per square} \end{aligned}$$

The number of squares S for the rectangular sections are as follows.

$$S_1 = \frac{l_1}{w_1} = 8 \quad S_3 = \frac{l_3}{w_2} = 3$$

The middle average segment of the conductor has a trapezoidal shape. The average of w_1 and w_2 can be used as the effective width, within 1% accuracy, if the w_2/w_1 ratio is 1.5 or less. Otherwise, a more exact result is obtained as follows.

$$S_2 = \frac{l_2}{w_2 - w_1} \ln \left(\frac{w_2}{w_1} \right) = 4 \ln 2 = 2.77 \text{ squares} \quad (\text{eq. 5})$$

$$\begin{aligned} \text{Total } R &= R_1 + R_2 + R_3 = \rho_S(S_1 + S_2 + S_3) \\ &= 7.51 \text{ m}\Omega \end{aligned}$$

As another example, assume that a 1 oz. trace must carry a 200 mA current six inches with a voltage drop less than 10 mV.

$$R_{\max} = \frac{V_{\max}}{I} = \frac{0.01}{0.2} = 0.05\Omega$$

$$0.05 = \rho_s \frac{l}{w} \tag{eq. 6}$$

$$\frac{w}{l} = 20 \rho_s$$

$$w = 120 \rho_s = (120) 5.657 \times 10^{-4} = 67.9 \times 10^{-3}$$

∴ minimum trace width, w = 68 mils

At a higher current level, consider the voltage drop in a conductor 20 mils thick, 1.25 inches wide and 3 feet long carrying a 50A current.

$$\rho_s = \frac{6.788 \times 10^{-7}}{2 \times 10^{-2}} = 3.364 \times 10^{-5} \Omega \text{ per square}$$

$$V = IR = (50) (3.364 \times 10^{-5}) \frac{36}{1.25} \tag{eq. 7}$$

$$= 0.0484 = 48.4 \text{ mV}$$

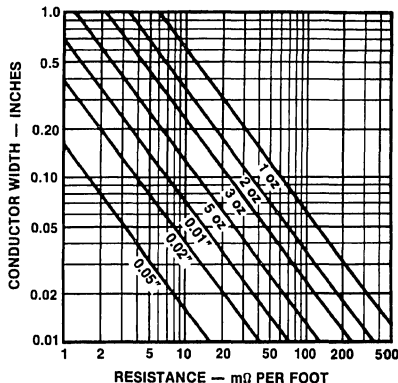
Sheet resistances for various copper thicknesses are listed in Figure 5. Standard thicknesses and tolerances for copper cladding are tabulated in Figure 6 and resistance per foot as a function of width is shown in Figure 7.

Weight or Thickness	Sheet Resistance Ω per Square	Thickness	Sheet Resistance Ω per Square
2 oz.	2.715×10^{-4}	0.02 in.	3.364×10^{-5}
3 oz.	1.886×10^{-4}	0.05 in.	1.358×10^{-5}
5 oz.	1.077×10^{-4}	1/16 in.	1.086×10^{-5}
0.01 in.	6.788×10^{-5}	1/4 in.	2.715×10^{-6}

FIGURE 5. Sheet Resistance for Various Thicknesses of Copper

Nominal Thickness		Nominal Weight oz/ft ²	Tolerances By	
in.	mm		Weight, %	in.
0.0007	0.0178	1/2	+ 10	+ 0.0002
0.0014	0.0355			+ 0.0004
0.0028	0.0715	2	+ 10	- 0.0002
				+ 0.0007
0.0042	0.1065	3	+ 10	- 0.0003
				+ 0.0006
0.0056	0.1432	4	+ 10	+ 0.0006
				+ 0.0007
0.0070	0.1780	5	+ 10	+ 0.0007
				+ 0.0008
0.0084	0.2130	6	+ 10	+ 0.0008
				+ 0.001
0.0098	0.2460	7	+ 10	+ 0.001
				+ 0.0014
0.014	0.3530	10	+ 10	+ 0.0014
				+ 0.002
0.0196	0.4920	14	+ 10	+ 0.002

FIGURE 6. Thickness and Tolerances for Copper Cladding



TL/F/10790-90

FIGURE 7. Conductor Resistance vs Thickness and Width

TEMPERATURE COEFFICIENT

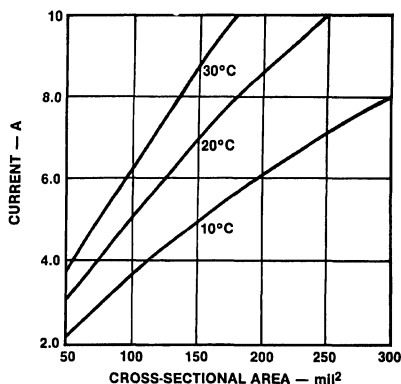
The resistances in Figures 3, 5, and 7, as well as those used in the sample calculations, are 20°C values. Since copper resistivity has a temperature coefficient of approximately 0.4%/°C, the resistance at a temperature (T) can be determined as follows.

$$R_T = R_{20^\circ\text{C}} [1 + 0.004 (T + 20^\circ\text{C})] \tag{eq. 8}$$

At 55°C:

$$R = R_{20^\circ\text{C}} [1 + 0.004 (55^\circ\text{C} - 20^\circ\text{C})] = 1.14 R_{20^\circ\text{C}}$$

When specifying power bus dimensions for PC cards containing many IC packages, designers should bear in mind that excessive current densities can cause the copper temperature to rise appreciably. Figure 8 illustrates the ohmic heating effect of various current densities.¹



TL/F/10790-91

FIGURE 8. Temperature Rise with Current Density in PC Board Traces

DISTRIBUTION IMPEDANCE

Power busses should have low AC impedance, as well as low DC resistance, to prevent propagation of extraneous disturbances along the distribution system. As far as current or voltage changes are concerned, power and ground busses appear as transmission lines; thus their impedances can be affected by shape, spacing and dielectric. The effect of geometry on impedance is illustrated in the two arrangements of *Figure 9*. The same cross-sectional area of copper is used, but the two round wires have an impedance of about 75Ω while the flat conductors have an impedance determined as follows.

$$Z_0 = \frac{377 d}{\sqrt{\epsilon} h} \text{ for } \frac{d}{h} < 0.1$$

With a Mylar® or Teflon® dielectric ($\epsilon = 2.3$) two mils thick, impedance of the flat conductor pair is only 0.5Ω . Power line impedance can be reduced by periodically connecting RF-type capacitors across the line.

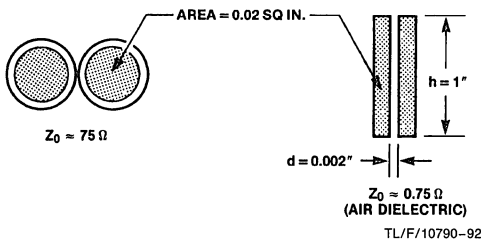


FIGURE 9. Effect of Geometry on Power Bus Impedance

*Mylar and Teflon are registered trademarks of E.I. du Pont de Nemours Company.

GROUND ON PC CARDS

It is essential to assign one layer of copper cladding almost exclusively to ground. This provides low-impedance, non-interfering return paths for the current changes which travel along signal traces when the IC outputs change state. These currents flow from the V_{CCA} pins of the IC packages, through the output transistors, then into the loads and the stray capacitances. These stray capacitances exist from an output to V_{EE} , output to ground, and to other signal lines. Thus, displacement currents through stray capacitances flow in many paths, but must ultimately return through ground to the output transistor where they originated. To reduce the length and impedance of the return path, the ground metal should cover as large an area as possible and one decoupling capacitor should be provided for every one to two IC packages. Additional capacitors may be needed for multiple output devices. These capacitors should be ceramic, monolithic or other RF types in the $0.01 \mu\text{F}$ to $0.1 \mu\text{F}$ range.

The load current returning to an IC package through ground metal is predictable, both in magnitude and in the return path. Since the magnetic and capacitive coupling between a signal trace and the underlying ground provides the transmission line characteristic, it follows that the load current flowing through the signal trace is accompanied by a ground return current equal in magnitude but opposite in direction. For example, in a 50Ω terminator I_{OL} is 5.9 mA , I_{OH} is 20.9 mA . Then signal change will cause about 15 mA current change and, as this current change propagates along the signal trace, a current of -15 mA advances along the

ground directly underneath the signal trace. Therefore, if there is an interruption in the ground, the return current is forced to go around it. The 15 mA current change can be reduced by terminating the complementary output of the signal. Then a signal change will direct the current from true output to the complement output reducing the Δ currents in the ground plane. When it is necessary to interrupt the ground plane, the interruptions should be kept as short as possible; every effort should be made to locate them away from overlying signal lines. When the ground plane is interrupted for short signal lines between packages, these lines should be at right angles to signal lines on the other side to minimize coupling. V_{EE} and V_{TT} distribution lines can also act as the return side of transmission lines, as long as decoupling capacitors to ground are placed in the immediate areas where the signal return current must continue through ground.

Several connections along the edge of a PC card should be assigned to ground to accommodate backplane signal ground. These should be spaced at one-half to one inch intervals to minimize the average path length for signal return currents and to simulate a distributed connection to the backplane signal ground.

Not enough emphasis can be placed on the requirement for a good ground. All input signals are referenced to internal V_{BB} and the V_{BB} is referenced to V_{CC} (ground). Any variation from one side of the board to the other affects the noise margins. To help eliminate some of the variations a separate V_{CCA} is provided on F100K ECL circuits to power the output drivers and leave the V_{CC} going to internal circuitry unaffected.

BACKPLANE CONSTRUCTION

In order to take complete advantage of the speeds inherent in F100K ECL it is desirable to construct the backplane as a multilayer printed circuit board. Generally, two internal layers are devoted to ground and V_{EE} and the signals occupy the outside layers. Where power densities are very high, it may be necessary to supplement the power layers with external busses (see Backplane Interconnections, *Chapter 4*). If it is necessary to use wires to augment the interconnection provided by the traces, less critical signals should use the wires. The wires will exhibit an impedance which can be calculated with the wire-over-ground formula

$$Z_0 = \frac{138}{\sqrt{\epsilon}} \log_{10} \frac{4h}{d} \quad (\text{eq. 9})$$

where d is diameter, h is distance to ground, and ϵ is dielectric constant.

Bear in mind that if the ground plane is buried inside the board, then both h and ϵ are made up of multiple components.

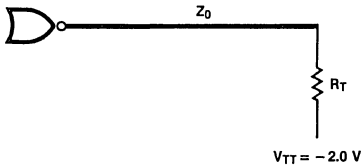
TERMINATION SUPPLY, V_{TT}

A separate return voltage for the termination resistors offers a way to minimize power dissipation in systems extensively using parallel termination techniques. A $-2V$ V_{TT} value represents an optimum speed/power trade-off, allowing sufficient termination current to discharge load capacitances while minimizing the average power consumption. *Figure 10* shows the average values of current, IC power dissipation and resistor power dissipation for various values of the termination resistor R_T returned to $-2V$. Average values are determined by calculating the output HIGH and output LOW values, then taking the average. These 50% duty cycle

values are useful in determining the current drain on the -2V supply and the contribution to dissipation on the logic boards. Peak values of termination current are approximately 60% greater than the average values listed.

DC regulation of the -2V supply is not critical; a variation of ±5% causes a change in output levels of ±12 mV for 50Ω terminations or ±7 mV for 100Ω terminations.

The high frequency characteristics of the V_{TT} distribution are extremely important. Ideally, a solid voltage plane should be devoted to V_{TT} . If this is not feasible, the V_{TT} distribution should form a grid using orthogonal traces. In any case, decoupling capacitors to ground should be used to reduce the high frequency impedance.



TL/F/10790-93

R_T Ω	I_{avg} mA	P_D (avg) mW	
		IC Output	Resistor
50	14	14	13
62	11	12	11
75	9.3	9.5	9.1
90	8.1	8.2	7.9
100	7.3	7.3	7.1
150	5.0	4.9	5.0

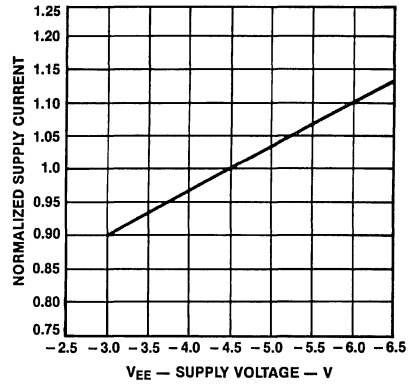
FIGURE 10. Average Current and Power Dissipation for Parallel Termination to -2V

If the terminators used are in Single In-line Packages (SIP) or Dual-In-line Packages (DIP) as opposed to discrete resistors, particular attention must be given to decoupling in order to maintain a solid V_{TT} voltage inside the package. This is necessary to avoid crosstalk due to mutual inductance to V_{TT} . SIPs have been developed which have multiple V_{TT} connections and on-board decoupling capacitors.

V_{EE} SUPPLY

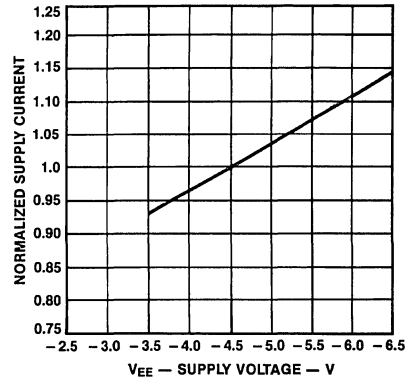
The value of V_{EE} is not critical for F100K since all circuits in the family operate over the range of -4.2V to -5.7V. Decoupling capacitors to ground should be used on each card, as previously discussed in connection with the ground on PC cards. In addition, each card should use 1 μF to 10 μF decoupling capacitors near the points where V_{EE} enters the card.

The current drain for the V_{EE} supply for each circuit type can be determined from the data sheet specifications. For V_{EE} values other than -4.5V, the current drain varies as shown in Figure 11 and 12 for SSI and MSI elements respectively. These graphs are made from data from the F100101 and F100179.



TL/F/10790-94

FIGURE 11. Supply Current vs Supply Voltage for F100101

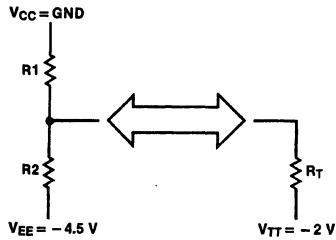


TL/F/10790-95

FIGURE 12. Supply Current vs Supply Voltage for F100179

Series dividers used to obtain Thevenin equivalent parallel terminations increase the current load on the V_{EE} supply, as do the pull-down resistors to V_{EE} used with series termination. Average V_{EE} current and resistor dissipation for Thevenin equivalent terminations are listed in Figure 13 for several representative values of equivalent resistance. The average values apply for 50% duty cycle. Peak current values are approximately 11% greater. Dissipation in the IC output transistor is the same as in Figure 10. Average dissipation and I_{EE} current for several values of pull-down resistance to V_{EE} are listed in Figure 14. The R_E values are appropriate for series termination of transmission lines with impedances listed in the Z_0 column, determined from Equation 12. Peak current values are approximately 12% greater than average values.

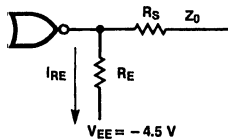
Figures 10, 13 and 14 show that the Thevenin equivalent parallel termination method leads to ten times as much dissipation in the resistors as in the single resistor returned to -2V. Similarly, the dissipation in R_E for series termination is three times the dissipation in the parallel termination resistor to -2V.



TL/F/10790-96

R_T Ω	$R_{1\Omega}$ $= 1.80 R_T$	$R_{2\Omega}$ $= 2.25 R_T$	I_{EE} (avg) mA	P_D (avg) mW Resistors
50	90	113	28.2	109
62	112	140	22.7	87.9
75	135	169	18.8	72.7
82	148	185	17.2	66.5
90	162	203	15.7	60.5
100	180	225	14.1	54.5
120	216	270	11.7	45.4
150	270	338	9.4	36.3

FIGURE 13. Series Divider for Thevenin Equivalent Terminations



TL/F/10790-97

Z_0 Ω	R_E Ω	I_{EE} (avg) mA	P_D (avg) mW	
			IC Output	R_E
50	269	9.8	12.9	25.8
62	331	7.9	10.4	20.6
75	399	6.5	8.6	16.8
90	477	5.4	7.1	13.9
100	530	4.9	6.5	12.7
120	634	4.1	5.4	10.6
150	791	3.2	4.2	8.1

FIGURE 14. Average Current and Power Dissipation Using Pull-Down Resistor to V_{EE}

THERMAL CONSIDERATIONS

System cooling requirements for ECL circuits are based on three considerations: (1) the need to minimize temperature gradients between circuits communicating in the single-ended mode, (2) the need to control the temperature environment of each circuit to assure that the parameters stay within guaranteed limits, and (3) the need to insure that the maximum rated junction temperature is not exceeded.

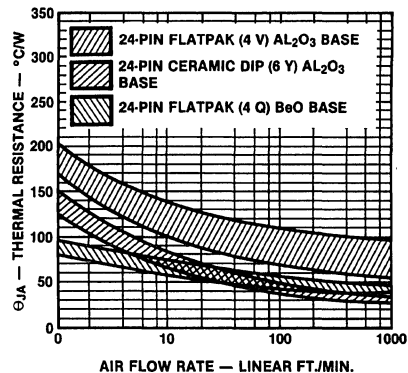
Temperature gradients are of no practical concern with F100K circuits since they are temperature compensated;

their output voltage levels and input thresholds change very little with temperature, as discussed in *Chapter 1*. With uncompensated ECL circuits, output voltage levels and input thresholds vary with temperature. This causes a loss of noise margin when driving and receiving circuits are operating at different temperatures. Loss of HIGH-state noise margin occurs when the receiving circuit is at the higher temperature, amounting to approximately 1 mV/°C of temperature gradient. When the driving circuit is at the higher temperature, the LOW-state margin decreases by approximately 0.5 mV/°C of gradient. The system designer must consider noise margin loss, due to temperature gradients.

Each DC parameter limit on the F100K data sheets applies over the entire 0°C to +85°C case temperature. For uncompensated ECL circuits, parameter limits have different values for different ambient temperatures. Further, ambient temperature specifications are based on a minimum air flow rate of 400 linear feet per minute. Thermal equilibrium must be established for incoming test results of uncompensated ECL circuits to be valid. The time required to attain equilibrium can vary considerably, depending on the internal dissipation of the particular IC type and details of the thermal arrangement. Normally, an adequate waiting time is three to five minutes after power is applied.

The maximum rated junction temperature of F100K circuits is +150°C. An individual IC junction temperature can be determined by multiplying power dissipation by the junction-to-air thermal resistance θ_{JA} and adding the result to the ambient air temperature. The power dissipation is V_{EE} times I_{EE} , from the data sheet, plus the dissipation in the output transistors from *Figure 10* or *14*. Thermal resistance is shown in *Figure 15* as a function of cooling air flow rate. This figure applies when the IC is mounted on a board with the air flowing in a plane parallel to the board and perpendicular to the long axis of the IC package. When air temperature, flow rate and package power dissipation are known, junction temperature is determined as follows.

$$T_J = T_A + P_D \theta_{JA} \quad (\text{eq. 10})$$



TL/F/10790-98

FIGURE 15. Junction-to-Air Thermal Resistance vs Air Flow Rate

Conversely, when the maximum rate junction temperature (+ 150°C), the package power dissipation, and the air temperature are known, the minimum flow rate can be determined by first determining the maximum thermal resistance.

$$\text{Maximum } \theta_{JA} = \frac{(150^\circ - T_A)}{P_D} \quad (\text{eq. 11})$$

For this value of θ_{JA} the minimum flow rate is determined from *Figure 15*.

When the system designer plans to depend on natural convection for cooling, it is recommended that thermal tests be conducted to determine actual conditions. The effectiveness of natural convection for cooling varies greatly. For

instance, on a densely packed logic board in a horizontal attitude in still air, the effective ambient temperature for an IC varies with its position. An IC in the middle of the board is subjected to air that is partially heated by surrounding ICs. Additionally, the temperature of the board rises due to heat flow through the component leads. These effects can cause a much higher junction temperature than might be expected.

REFERENCE

1. Harper, C.A., Editor, *Handbook of Wiring, Cabling and Interconnecting for Electronics*, McGraw-Hill, 1972.

Point-to-Point Fiber Optic Links

National Semiconductor
Application Note 679
Filipe Sanna
Louise Yeung



TABLE OF CONTENTS

1.0 POINT-TO-POINT APPLICATIONS

2.0 SYSTEM OVERVIEW

3.0 CHANNEL SYNCHRONIZATION

3.1 Synchronization Timing Examples

4.0 PHY LAYER COMPONENTS

4.1 System Block Diagram

4.2 Channel Block Diagram

5.0 HOST INTERFACE CONSIDERATIONS

5.1 Data Interface

5.2 Control Bus Interface

6.0 TIMING BUDGET FOR WIRED AND STRUCTURE

6.1 Pull-Up Scheme

6.2 GAL Scheme

INTRODUCTION

A station using the ANSI X3T9.5 (FDDI) physical layer standard can transmit and receive data at 100 Mbits/sec through a fiber optic cable. However, with several physical layers connected together in parallel, each with its own fiber optic cables for transmission and reception, the station can transmit and receive data at much higher speeds. National Semiconductor's physical (PHY) layer devices can be connected together in parallel to achieve such high bandwidth point-to-point links. The National devices required to implement a PHY layer are the DP83231 Clock Recovery Device (CRD™ device), the DP83241 Clock Distribution Device (CDD™ device), and the DP83251/55 Physical Layer Controller Device (PLAYER™ device). The bandwidth of the system depends on the number of PHY layers used—each set of PHY layer devices contributes 100 Mbits/sec to the system.

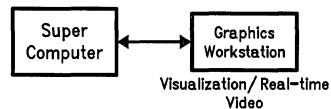
1.0 POINT-TO-POINT APPLICATIONS

The use of parallel FDDI PHY layers is a cost effective method to increasing the data throughput in multiples of 100 Mb/s. This task can be accomplished utilizing an existing FDDI fiber plant.

FDDI PHY layers in parallel require only one pair of fiber optic cables, one pair of transceivers, and one set of PHY layer chips per channel, while yielding a typical data throughput of 800 Mbits/sec (for a system with eight channels).

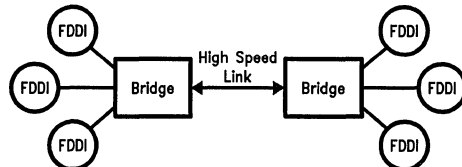
Any application where data throughput is the limiting factor to system performance is a candidate for a high-speed point-to-point link. For example, a point-to-point link can be installed between a CPU and a disk controller to speed up information storage and retrieval times. Another application area could be in the display capabilities of a graphics workstation which can be combined with the data processing power of a supercomputer to achieve visualization for data intensive simulations (*Figure 1*). As a third example, a high-speed networking backbone using a point-to-point link is depicted in *Figure 2*. Here, separate FDDI rings are connected together with a high speed link which provides bridging between rings without loss of performance.

Fiber optics afford a greater physical separation between stations than electrical signals. Hence, a fiber point-to-point link can be used to extend SCSI or IPI transmissions up to one kilometer.



TL/F/10797-1

FIGURE 1. Point-To-Point Link between a Supercomputer and a Graphics Workstation



TL/F/10797-2

FIGURE 2. High Speed Networking Backbone Using a Fiber Optic Point-To-Point Link

2.0 SYSTEM OVERVIEW

A system using four PHY layers in parallel is shown in *Figure 3*. The diagram demonstrates conceptually how data is passed from Station I to Station II at 400 Mbits/sec. Each of the four PHY layers in Station I is connected to a PHY layer in Station II with fiber optic cables. Since National Semiconductor PHY layers are full duplex, each pair of PHY layers is linked by two fibers, one for transmission in each direction.

The system interface shown contains two parts. SYS_REQ, which handles data transmission, and SYS_IND, which handles data reception.

Suppose that Station I wants to transmit a 32-bit word of data to Station II. SYS_REQ in Station I takes the data and splits it into four bytes, one for each PHY channel. Each PHY layer reads its byte from the PHYn_REQ ports of SYS_REQ (at 12.5 MHz) and sends the data out across the fiber as an 8-bit serial stream at 100 Mbits/sec. (Note that due to the 4B/5B encoding scheme used by the PLAYER devices the data actually passes through the fiber as a 10-bit serial stream at 125 MHz.)

After travelling through the fibers, the data arrives at Station II. Each PHY layer on the receiving end reads data from the fiber and presents its byte to the corresponding PHYn_IND port (at 12.5 MHz) in Station II. SYS_IND then rejoins the bytes back into the 32-bit word sent by Station I and can present the data to the host (at 12.5 MHz). This demonstrates how Station I can send 32 bits to Station II in 80 ns, giving an effective data throughput of 400 Mbits/sec.

Even though this is a non-FDDI application, the general rules for FDDI framing must be followed. In particular, each frame must start with a JK symbol and end with valid FDDI ending delimiter (*Figure 4*). Furthermore, the frame size must be between three and 4500 bytes long (see PLAYER device datasheet for more detail). At least four pairs of idle symbols should be inserted between the frames to allow for readjustment of the PLAYER device's elasticity buffer. However, to guarantee at least one opportunity to recenter the elasticity buffer between frames in the event of clock drift or a single line hit in the interframe gap, the user is advised to insert eight idle symbol pairs.

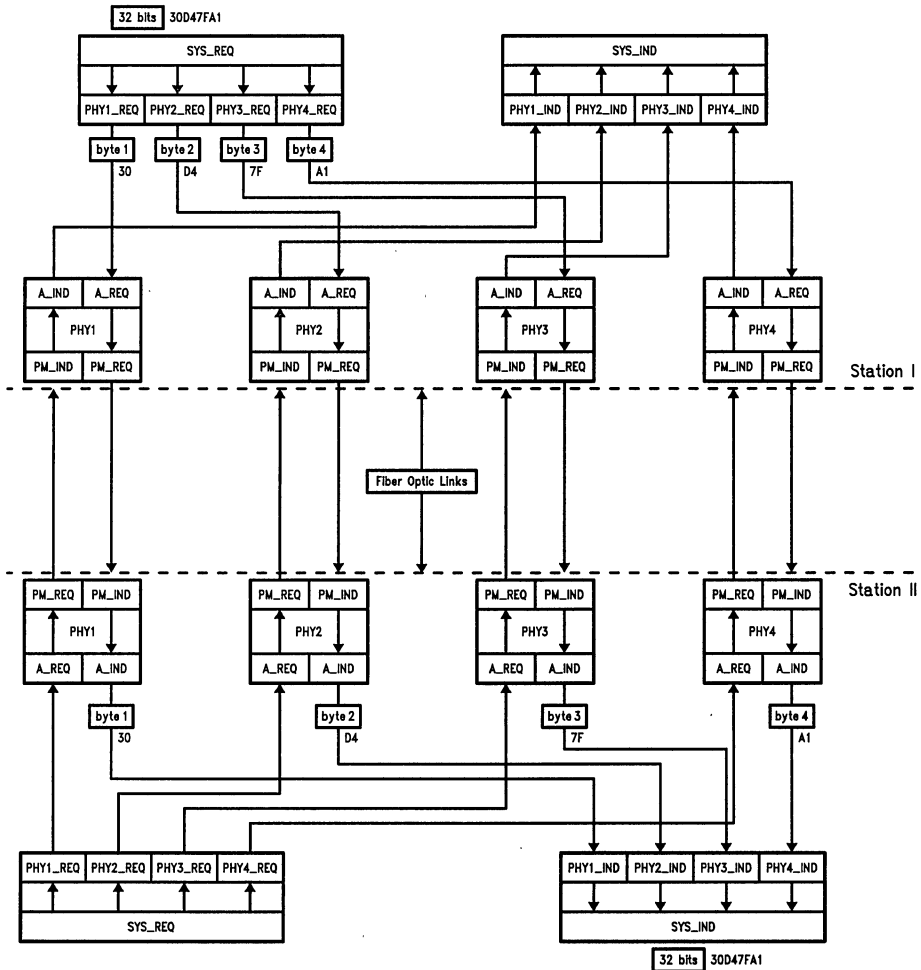
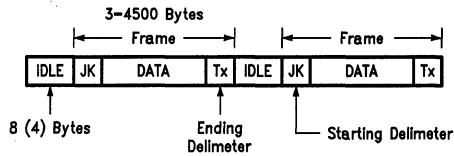


FIGURE 3. System Using Four PHY Layers in Parallel

TL/F/10797-3



TL/F/10797-4

FIGURE 4. Valid Frame Format

The FDDI standard specifies the maximum clock drift between two stations to be 100 parts per million (ppm) peak-to-peak and the maximum frame size to be 4500 bytes. However, it is possible to transmit more than 4500 bytes per packet if tighter clock tolerances are observed. The equation to determine the maximum allowable frame size is given below:

$$\text{Frame}_{\text{MAX}} = 40 \text{ ns} \frac{1}{\text{Clock Drift per Bit}}$$

The clock drift per bit is calculated by taking the maximum difference in the CDD device frequencies between the transmitting station and the receiving station multiplied by 8 ns. For example, if the 12.5 MHz crystal used by the CDD device in the transmitting station had a +25 ppm accuracy error and the 12.5 MHz crystal used by the CDD device in the receiving station had a -30 ppm accuracy error, then the clock drift per bit would be 55 ppm (peak-to-peak) x 8 ns or 4.4 ps. The maximum frame size that could be transmitted using these crystals is thus 40 ns/4.4 ps or 9090 bytes, significantly more than the 4500 specified.

3.0 CHANNEL SYNCHRONIZATION

Since the data traveling from Station I to Station II passes through different devices and different fibers, it may not arrive at each of the PHY channels on the receiving end at exactly the same time. Hence there must be some method of aligning the incoming bit streams so that data is passed to the PHYn_IND ports in the correct sequence. The JK symbol serves as the reference point for synchronizing the bit streams among channels. National Semiconductor PLAYER devices have an open drain output called Cascade Ready (CR-pin 48) which is released when a JK is received on that channel. ANDING the Cascade Ready (CR) pins of all the PLAYER devices creates a signal (called Cascade Start) which indicates that all of the channels have received a JK symbol. This signal is tied to the Cascade Start (CS-pin 47) input of each PLAYER device and indicates when all of the devices achieve synchronization.

The first PLAYER device that receives a JK symbol pair will present that pair to the host (through the A Indicate Port). Meanwhile, it will activate the open drain CR output. If needed, it will output a second JK to the host as it waits for synchronization from the other PLAYER devices. During this time, the incoming data can be temporarily stored in the elasticity buffer. The ability of the PLAYER devices to output two consecutive JK symbols yields an 80 ns synchronization window. Each PLAYER device that receives a JK symbol will present a JK symbol to the host and release its CR line.

Once all of the PLAYER devices have released their CR lines, the CS signal feeding each PLAYER device will go high. At this point the read pointers of all the PLAYER device's elasticity buffers will be aligned and all of them will output JK symbols to the host. Simultaneous reception of JK symbols on every channel informs the host that synchronization has occurred, and that the subsequent data bytes will be properly aligned.

The synchronization process is repeated with each new frame received, and may not always proceed exactly as described above. Depending on the skew between the fastest and slowest channels, the PLAYER devices will either synchronize the bit streams or generate an error. Figure 5 shows four different scenarios for the synchronization of several PHY channels. Each is described in the following section.

3.1 Synchronization Timing Examples

Figure 5a presents the timing waveforms for a single PLAYER device which was the first device to receive a JK at the Elasticity Buffer (EB) in a situation where the last JK received by another PLAYER device is 60 ns behind it. PMD_IND shows a simplified version of the data coming into the PLAYER device from the CRD device. The Local Byte Clock (LBC) is a 12.5 MHz TTL signal from the CDD device which is used by the PLAYER device. After a propagation delay, data appears at the A Indicate Port of the PLAYER device on each rising edge of LBC. The CR line is pulled high on the first falling edge of LBC after the PLAYER device completely receives the JK symbol. The delay in this signal (t_1) is due primarily to the external propagation delay of the CR line pullup structure and some delay time within the PLAYER device itself.

The important point to note in this scenario is that the CS signal does not go high for some time (t_2) after the CR signal goes high. This indicates that this PLAYER device was one of the first PLAYER devices in the system to receive its JK symbol, since CS = 1 only when all of the PLAYER devices have received a JK symbol. Since the CS signal failed to go high before the first falling edge of LBC after the CR line is released, this PLAYER device outputs a second JK symbol to the host through the A Indicate Port. By the next falling edge of LBC, the CS signal has gone high and all of the PLAYER devices outputting the first byte of data after the JK symbol to the host.

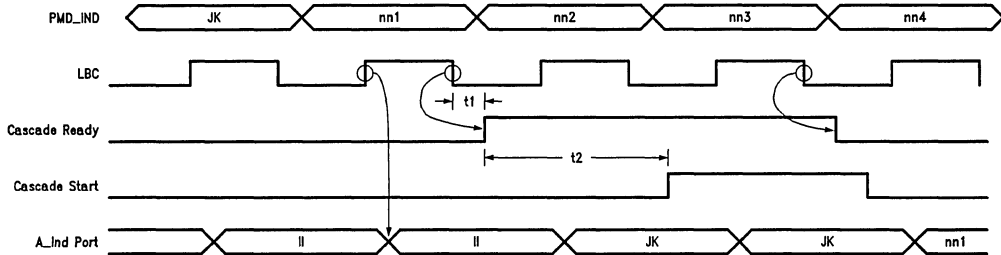
Figure 5b depicts the timing waveforms for a single PLAYER device which was the first device to receive a JK symbol in a situation where the last JK symbol received from another PLAYER device is less than 20 ns after the first JK symbol. Again, PMD_IND shows the beginning of a frame coming from the CRD device. The PLAYER device releases the CR line upon reception of the JK symbol and after a short delay (t_3), the CR signal goes high. In this scenario, however, the CS signal goes high within one LBC period, so that the PLAYER device shown only has to report one JK symbol to the host before outputting data. This indicates that all of the symbols are coming in with a skew of less than 40 ns between the slowest and fastest channels. The delay between the release of CR and the assertion of CS (t_4) depends on skews in LBC between the channels, the reaction time of the wired-AND structure used to create the CS signal, and the skew between the data coming in on the different channels.

Figure 5c demonstrates the timing waveforms for a channel which receives the JK symbol pair last. Here, the CS signal goes high immediately after the PLAYER device releases the CR line. The only delay (t_5) is due to skews in LBC between the channels and the reaction time of the external wired-AND structure. Since this PLAYER device senses the CS signal within the first falling edge of LBC, it only needs to output one JK symbol to the host before outputting the data stream.

Figure 5d shows an error situation where one or more of the channels never receives a JK symbol. In this case, the

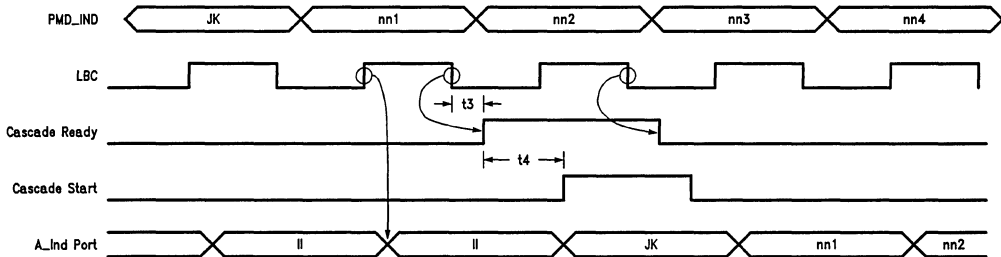
PLAYER device shown recognizes its JK symbol and releases the CR line. Since one of the PLAYER devices never gets a JK symbol and hence never releases the CR pin, the CS signal fails to go high. The PLAYER device shown in this diagram attempts to compensate for skew by outputting two

JK symbols to the host, but when the CS signal fails to rise the device is forced to output the incoming data and report a Cascade Sync Error (CSE) in the Receive Condition Register B (RCRB).



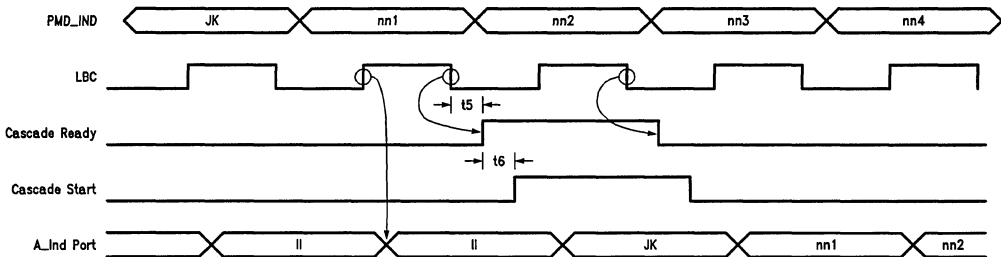
TL/F/10797-5

FIGURE 5a. First PLAYER device to receive a JK symbol in a large skew scenario. The CS signal goes high about 60 ns after the CR line is asserted. This indicates that some of the channels received JK symbols more than one LBC period after the channel shown.



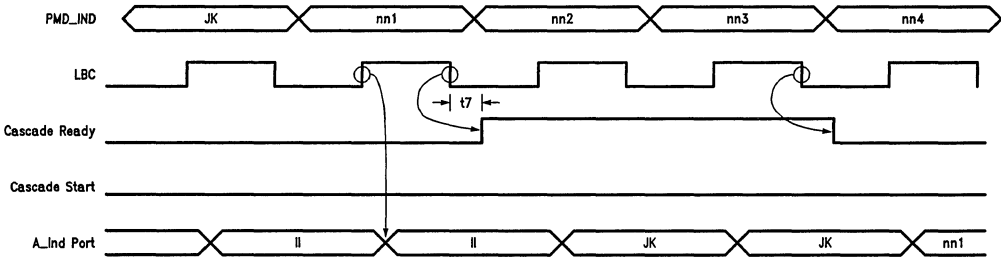
TL/F/10797-6

FIGURE 5b. First PLAYER device to receive a JK symbol in a small skew scenario. The CS signal goes high shortly after assertion of the CR signal, indicating that all of the channels received JK symbols within t_4 .



TL/F/10797-7

FIGURE 5c. Last PLAYER device to receive JK symbol (in any scenario). Since this device was the last to assert its CR line, t_6 is due only to the delay of the wired-AND structure.



TL/F/10797-8

FIGURE 5d. PLAYER device which receives JK symbol in scenario where one or more channels never receive JK symbols.

4.0 PHY LAYER COMPONENTS

4.1 System Block Diagram

The number of PHY layers connected together in parallel is limited only by the timing budget of the CS line (explained in Section 5.2) and the timing skews between channels. As an example, a system level block diagram using four PHY layers connected together in parallel is presented in Figure 6. All of the PHY layers within a given station are driven with a single set of clock signals, and all are controlled and monitored by the host system through the Control Bus interface. Each channel has two dedicated fibers, one for transmission and one for reception. The full duplex architecture eliminates the need for complex handshaking between the two stations. The four channels communicate through the CR and CS signals. For simplicity, the CR lines are shown connected to a pullup resistor—a more detailed look at the connection of these pins is given in Section 6.

The global clock scheme should be arranged to minimize the skews in the clock signals between PHY layers. Smaller clock skews between channels will leave more tolerance for device skews and fiber optic variations. For further recommendations concerning the CDD device in a multiple PLAYER device environment, see the CDD device datasheet (CDD Device Driving Multiple PLAYER Devices).

Figure 7 illustrates the source of timing deviations between the channels and demonstrates the need to minimize timing skews between the channels wherever possible. In Station I, we are concerned with TXC and TBC while in Station II we examine LBC, since Station I is transmitting to Station II. The time parameters shown in the figure represent the maximum deviations in propagation delay between channels. For example, if t_1 were 10 ns, this would mean that TXC/TBC could arrive at PHY1 up to 10 ns before arriving at PHY2.

t_1 represents the skews in TXC/TBC between the channels, t_2 encompasses the skews in the PHY layer's transmitting path, t_3 represents the differential skews amongst the fibers, t_4 includes all of the skews inherent in the PHY layer's receiving path, and t_5 represents the skews in LBC between the channels in the receiving station. All of the skews together must not exceed 80 ns in order to prevent synchronization errors, and smaller total skews will provide greater stability across temperature and power fluctuations.

In a worst case scenario where all devices were badly skewed, t_2 together with t_4 yields a base 30 ns of skew between the channels. This leaves 50 ns available for differential skews in the clock signals and the fiber. It is recommended that t_1 be held to 4 ns and t_5 kept under 8 ns to prevent misclocking of the data. Hence, the maximum skew among fibers should be less than 38 ns.

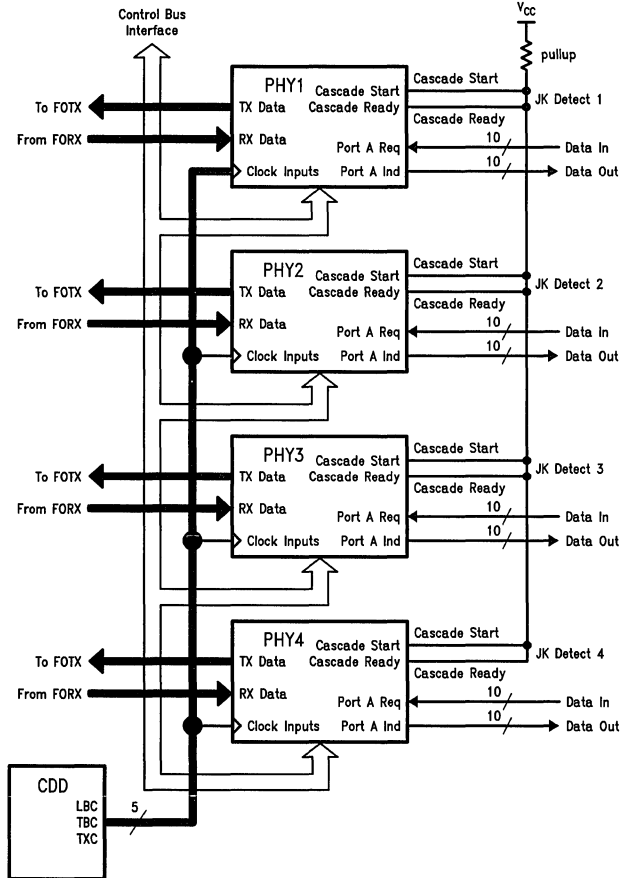


FIGURE 6. PHY Layer Block Diagram for an Example System Using Four PHY Layers in Parallel

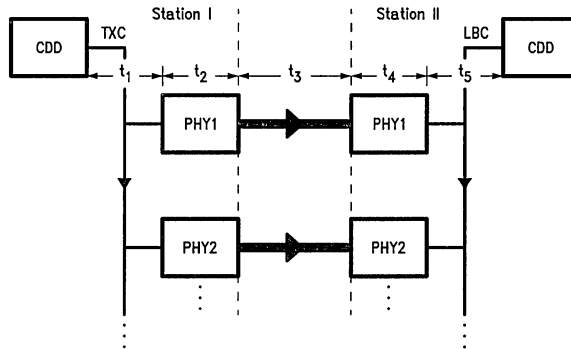
TL/F/10797-9

The following equation summarizes the tradeoff between cable length and variance:

$$\frac{l(1 + v)}{s(1 - w)} - \frac{l(1 - v)}{s(1 + w)} < 38 \text{ ns}$$

- where: **s** is the speed of the signal in the cable
- l** is the average length of the fiber in meters
- v** is the variance in the length of the cable
- w** is the variance in the speed of the signal through the fiber

For example, for a typical FDDI fiber optic cable, $s = 1.9 \times 10^8$ m/s, $v = 0.005$, and $w = 0.001$. Solving for **l** with these parameters results in a length of 667 meters.



TL/F/10797-10

Where:

- t_1 = Worst case clock skew between two PHYs
- t_2 = Worst case skew between PLAYER device propagation delays
- t_3 = Worst case skew between Fibers
- t_4 = Worst case skew between PLAYER device propagation delays
- t_5 = Worst case clock skew between two PHYs
- Total skew = $t_1 + t_2 + t_3 + t_4 + t_5$

Note: Total skew must not exceed 80 ns in order to prevent synchronization errors.

FIGURE 7. Origin of skew between channels. Adding all of the skews (t_1 through t_5) gives the total possible skew for the system.

4.2 Channel Block Diagram

Figure 8 shows the components which constitute a single PHY channel (the CDD device is common to all channels so it is not shown here). The fiber optic transceivers are standard FDDI devices which translate electric signals to light pulses and vice versa. The fiber optic receiver accepts data from the fiber and sends two pairs of differential ECL signals to the CRD device, namely signal detect and data. The CRD device extracts a clock signal from the incoming data and passes a resynchronized equivalent of this data and a recovered clock signal to the PLAYER device, as well as signal detect and clock detect signals.

Within the PLAYER device, the incoming data stream is decoded (from 5B to 4B) and placed in the elasticity buffer. When in cascade mode, the elasticity buffer is used not only to absorb variations between the received clock and the local clock, but also to smooth out skews between incoming data presented to the different PHY channels. If all of the PLAYER devices receive JK symbols within 80 ns of each other and release their CR pins, then the CS pin will go high and all of the PLAYER devices will read from the first data location of the elasticity buffer. This cell contains the first byte of data received after the JK symbol. Hence, the elasticity buffers facilitate the coordination of data output between the different PHY channels. If the last PLAYER device receiving a JK does so more than 80 ns after the first

PLAYER device, then the Cascade Sync Error (CSE) bit will be set in the Control Bus register RCRB by the first PLAYER device to have recognized a JK symbol.

5.0 HOST INTERFACE CONSIDERATIONS

5.1 Data Interface

The system interface should consist of a transmit holding register and buffer for transmission and a buffer and receive register for incoming data. A state machine is required to decode the symbols coming from the PLAYER device so that only data is stored. Furthermore, a controller will be required to monitor and manage the PLAYER device through the Control Bus interface. This controller must handle the initialization of the PLAYER device and report error conditions to the host.

Each PLAYER device takes ten bits of data at the A Request Port, a pair of 4-bit data symbols plus a parity and control bit. (See the PLAYER device datasheet for the PHY_MAC byte wide interface table.) The system interface can thus generate parity and control for each PLAYER device separately and check control and parity coming from each channel. To simplify the system interface, however, the parity pins can be tied to ground and parity checking can be disabled in the Current Transmit State Register (CTSR). Parity information coming from the PLAYER device can similarly be ignored.

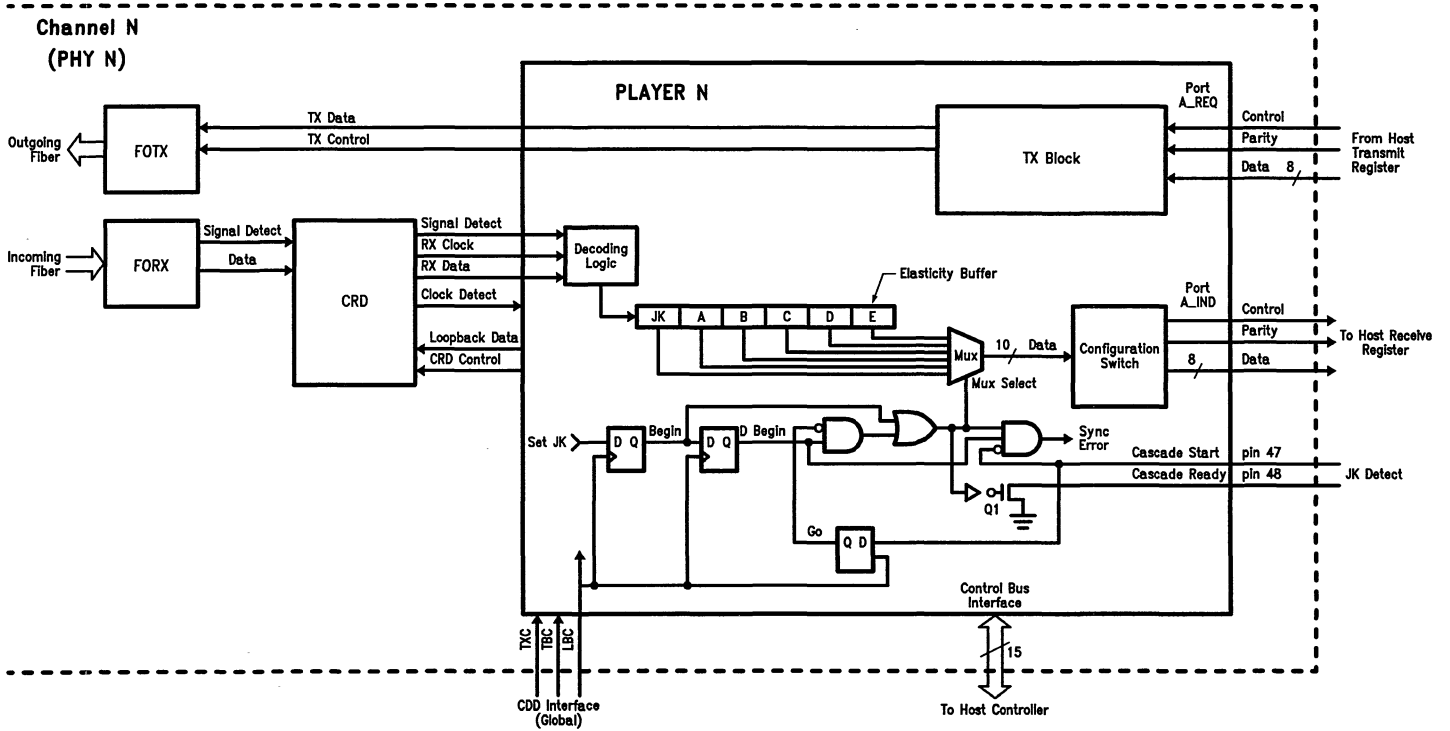


FIGURE 8. Block diagram of one PHY channel. The components shown here are repeated for each separate channel in the system.

In error situations, one or more PLAYER devices may report a Cascade Sync Error, but they may not do so simultaneously depending on when they receive JK symbols. The Cascade Sync Error (CSE) bit of the Receive Condition Register B (RCRB) will be set by each PLAYER device which receives a JK but does not sense the CS pin go high before the second falling edge of LBC from when CR was released. CS has to be set approximately within 80 ns of CR release. If a JK symbol is completely corrupted from a line hit or bad connection, the PLAYER device on that channel will not report a CSE. Only the data on the channel(s) which did not report a CSE will be corrupt, however, these are the channels which were unable to synchronize with the rest of the group. All of the PLAYER devices which receive a JK symbol (and release the CR pin) will read data from the first cell of the Elasticity Buffer. Therefore, a line hit on a single fiber will not wipe out the entire frame. The rest of the channels may still output synchronized data. This is particularly important in applications where partial data reception is still useful. For example, during screen updates in high resolution graphics systems, only one line of pixels would be lost instead of an entire block of the screen blanking out.

5.2 Control Bus Interface

If no JK symbols are corrupted, but they arrive with more than 80 ns of skew, all of the PLAYER devices will eventually report a CSE error. Hence the control microprocessor has the ability to pin point the corrupted channel or determine if the problem is due to excessive skew between the channels. Note that the Control Bus registers can be programmed to assert the interrupt (INT) pin upon detection of the CSE flag.

To place the PLAYER devices in Cascade Mode, the Mode Register (MR) must have the Cascade Mode (CM) bit set to one. The Cascade Synchronization Error (CSE) of the Receive Condition Register B (RCRB) is set to one if the CS signal fails to go high within 80 ns of recognizing the JK symbol. The RCRB also reports Elasticity Buffer errors through the EBOU bit, signaling a loss of data from the fiber. These bits must be cleared by the Control Bus controller.

When the number of PLAYER devices and the total capacitance is small, it may be possible to tie all CR pins and CS pins together and use a single pullup resistor. The lower limit of the pullup resistor is calculated as follows. The CR pins typically sink a 13 mA maximum, so the equation for the smallest resistor which should be used is:

$$R_{MIN} = V_{CC} / 0.013\Omega$$

Hence for a voltage supply of 5V, the resistor value is $5 / 0.013 = 385\Omega$. The upper limit of the pullup resistor depends on the capacitance of the system and the number of PLAYER devices used. Restricting the timing budget (t_b) to 20 ns (worst case) for the AND function, we arrive at the following equation:

$$R_{MAX} = t_b / (m \times n) \Omega$$

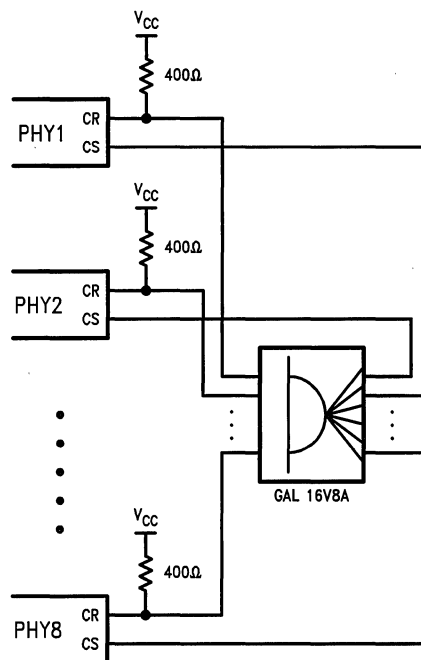
where: **m** is the capacitance associated with each PLAYER device's CR line (including the IC capacitance (4 pF), the socket capacitance, and the trace capacitance) measured in pF

n is the number of PHY channels (number of cascaded PLAYER devices).

Thus if **m** is 20 pF and **n** is 2, the maximum pullup resistor is 500 Ω , which meets the specification for the minimum resistor size. However, it is apparent that for many PLAYER devices a passive pullup resistor is too slow.

6.2 GAL Scheme

As a result, we recommend using external logic devices for any system with three or more cascaded PLAYER devices. The choice of devices is limited by the propagation delay as well as fan in or fan out. Each CR pin should be pulled up with a 400 Ω resistor and fed into the AND gate. A recommended device to perform the AND function is the GAL16V8A chip, which offers 8 inputs and supplies 8 outputs with a propagation delay of 10 ns. This chip will allow up to eight PLAYER devices to be cascaded together while still maintaining the necessary delay, fan in and fan out characteristics. The device should be placed in the electrical center of the cascaded devices to prevent excessive timing skews among the chips. Figure 9 depicts an eight channel system using the GAL16V8A to AND the CR signals together.

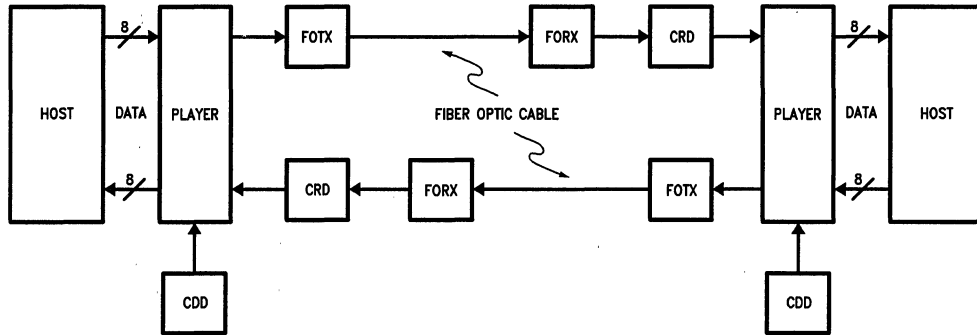


TL/F/10797-12

FIGURE 9. Example of an eight channel system using a GAL16V8A chip to perform the AND function on the CR lines. The resulting signal (CS) is fed back into each of the PLAYER devices.

High Speed, Point to Point, Fiber, Data Communication

National Semiconductor
System Brief 107



Serial Point-to-Point Data Communications

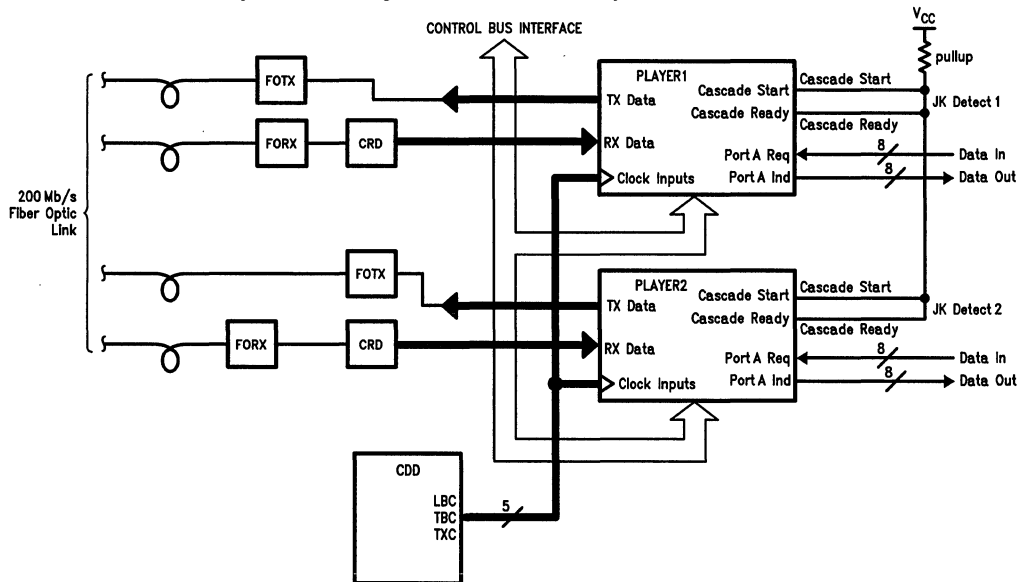
TL/F/10857-1

SYSTEM DESCRIPTION

Point-to-Point links are needed in any application where data throughput is the limiting factor to system performance. They can be installed between a CPU and disk controller to speed up information storage and retrieval times or the display capabilities of a graphics workstation can be combined with a supercomputer to achieve visualization for data intensive simulation. Point-to-Point links can also be used in a high-speed networking backbone where separate FDDI rings are connected with a fiber link to provide bridging with-

out loss of performance. Fiber optics can be used to extend SCSI or IPI transmissions. High bandwidth, greater than 100 Mb/s, point-to-point applications can use parallel FDDI PHY layers. High bandwidth point-to-point solutions prove parallel FDDI PHY layers to be cost effective, simple systems with minimal logic, and requiring only standard FDDI fiber optic transceivers (which operating at 125 MHz, are less expensive than a GHz laser).

System Block Diagram: Two Parallel PHY Layer in Cascade Mode



TL/F/10857-2

DESIGN CHALLENGES

1. Need for high bandwidth, high throughput, and high reliability.
2. Need to interface with a variety of system applications and maintain security and reliability.

KEY COMPONENTS

1. Clock Distribution Device (CDD™) provides the clocks needed for the PLAYER™ device and the host if needed. It provides 125 MHz differential ECL signals from an inexpensive 12.5 MHz crystal.

2. Physical Layer Controller Device (PLAYER™) performs the encoding and serialization of transmitted data, and deserialization and decoding of received data. It is compatible with 4B/5B and NRZI transmission code, and supports both single and dual attach station configurations.
3. Clock Recovery Device (CRD™) extracts the clock signal from incoming data and passes a resynchronized equivalent and a recovered clock signal to the PLAYER Device.
4. Transceiver provides electrical to light conversions.

BILL OF MATERIAL

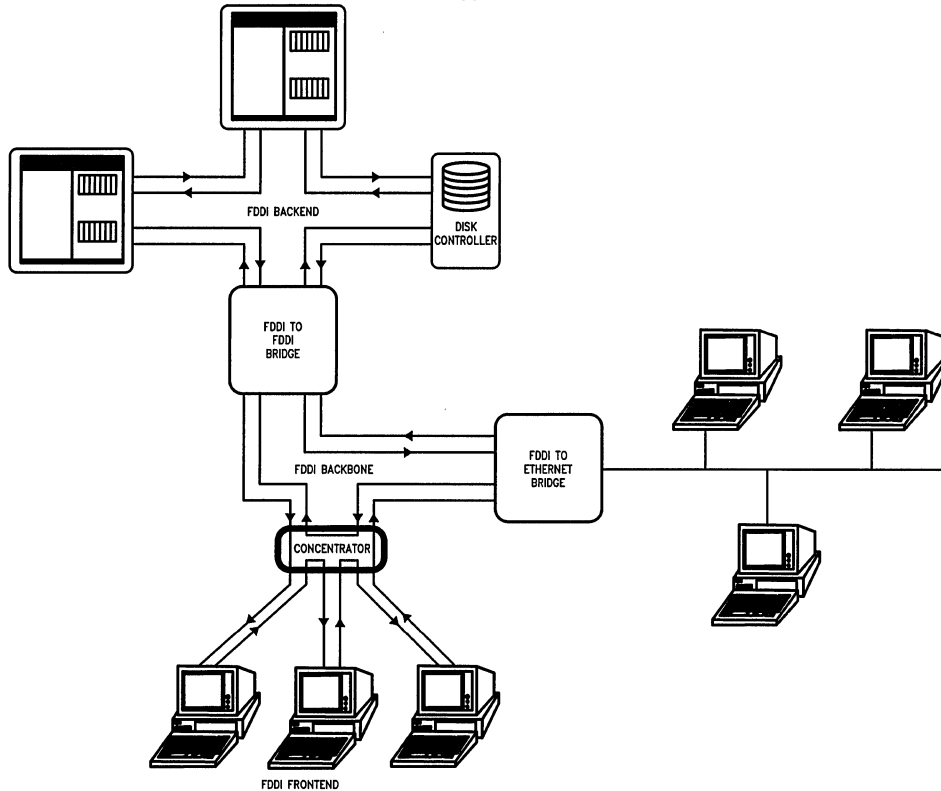
Function	Description	NSC	Other Mfg.	Qty
Decoding	PLAYER	DP83251/55		2
Clock Recovery	CRD	DP83231		2
Clock Distribution	CDD	DP83241		1
Transceiver	FOTX			2
	FORX			2
AND Function	GAL	GAL16V8A		1-8

FDDI Concentrator

National Semiconductor
System Brief 112



FDDI Applications



TL/F/11005-1

SYSTEM DESCRIPTION

The Concentrator plays an important role in the Fiber Distributed Data Interface (FDDI) architecture. FDDI offers a whole range of network topology alternatives. The concentrator simplifies the wiring of networks and allows logical ring topologies to be created from the typical star wiring configuration. The concentrator provides a very reliable and economic method of obtaining fault tolerance. The concentrator provides drops to individual nodes in order to include them in the network. When the concentrator senses a failure on one of the drops, it 'heals' the ring by electronically bypassing that station. Properly designed concentrators can bypass any number of drops with no degradation in performance.

The concentrator is an extremely chip intensive system. The small footprint, low power consumption, and special bridging features provide the ideal solution for concentrator applications. Concentrators are ideal for the needs of interconnectivity as addressed through high performance FDDI networks.

KEY DESIGN CHALLENGES

Management Software

Developing the Network Management software to manage all aspects of the concentrator and participate in the network management protocols is not a trivial task. The concentrator is also the best location for network diagnostic support including network monitors.

Modular Design

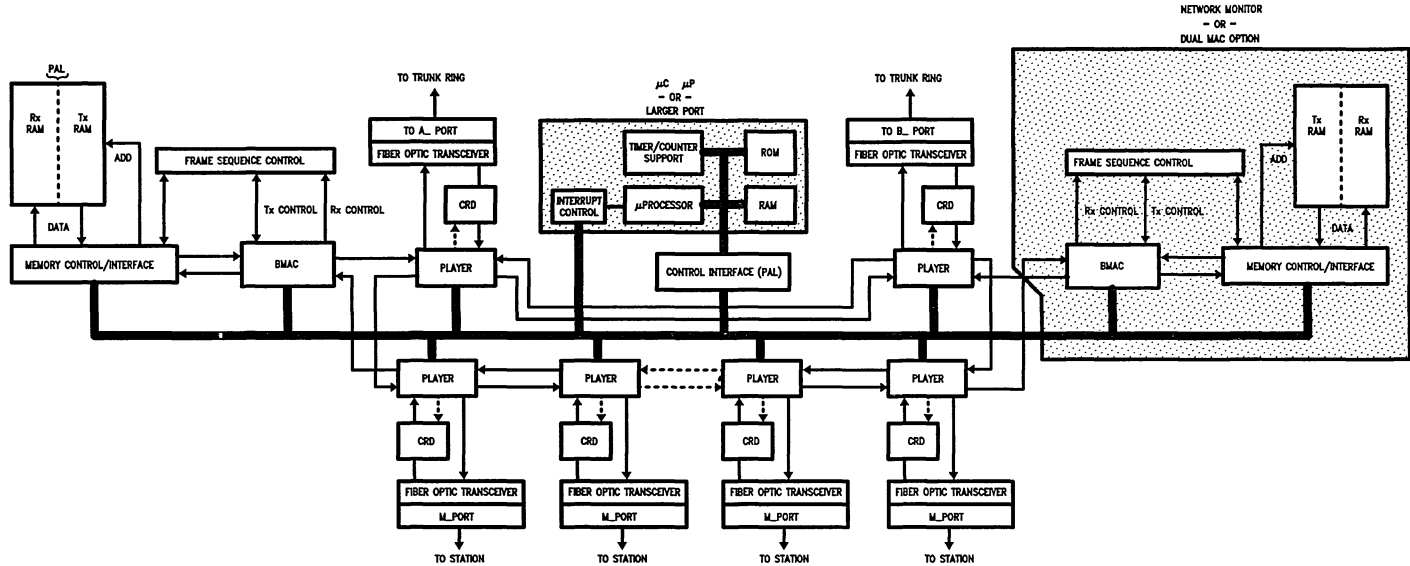
Keeping the design modular, while maintaining its manageability and flexibility, can save design time and manufacturing costs. Key to the architecture is to provide high throughput and flexibility to interface to a variety of system configurations.

In a multiboard design several other design challenges are present including other clock distribution, multi-processor communication, and backplane design issues.

Clock Distribution

Each port within a concentrator requires 125 MHz and 12.5 MHz clocks; the distribution of these clocks is not a simple task; the CDD device provides an elegant solution.

System Diagram



Single Board Concentrator Design

TL/F/11005-2

KEY COMPONENTS

PLAYER™ (Physical Layer Controller) device converts the BMAC device Mbyte stream into an encoded bit stream as specified in the FDDI PHY standard. It synchronizes the received bit stream to the local clock and decodes the 4B/5B data into internal code. The PLAYER device also contains a configuration switch for use in dual attachment stations and concentrators.

BMACTM (Basic Media Access Controller) device implements the functions defined by the ANSI X3T9.5 FDDI Media Access Control (MAC) standard. The device consists of the transmit and receive state machines, an address magnitude compare unit, a CRC generator and checker, protocol timers, and diagnostic counters.

CDD™ (Clock Distribution Device) device generates the clocks required by the PLAYER and BMAC devices, one per board.

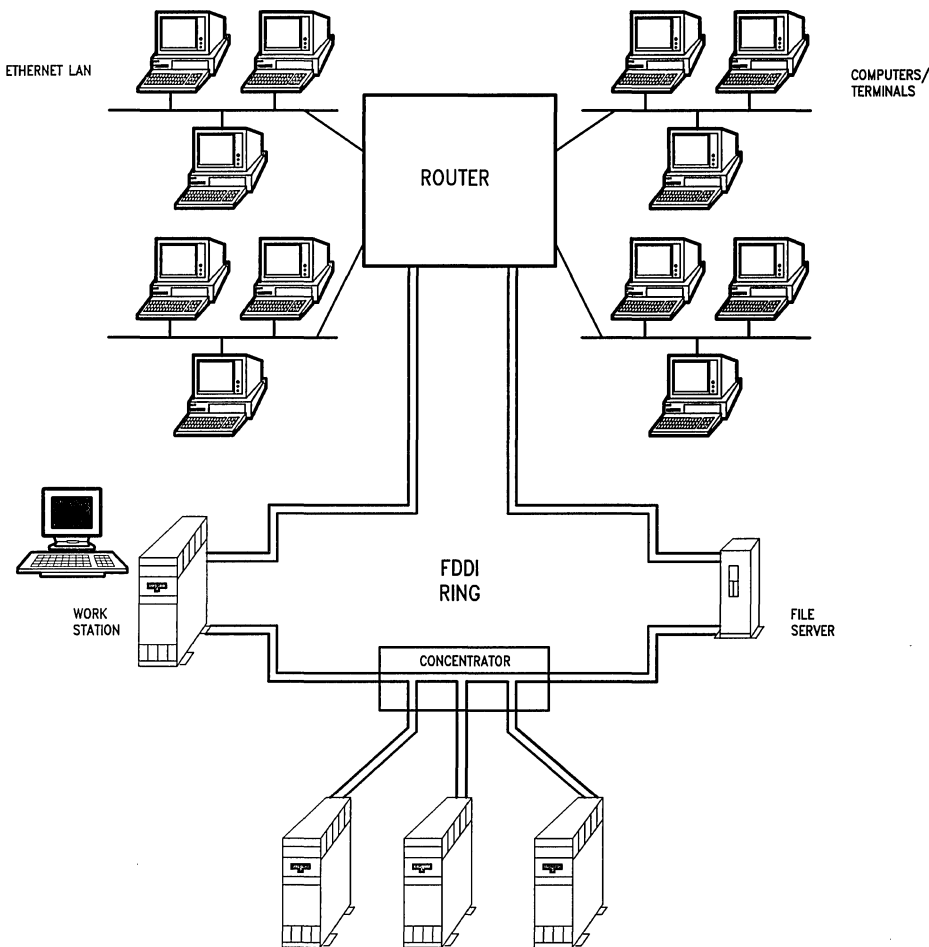
CRD™ (Clock Recovery Device) device extracts specific incoming clock data from the upstream station. Its features include on-chip loopback control, crystal control, the ability to lock to a master line state in less than 100 μ s, and a single +5V supply.

NS32GX32
or HPC

Performs the control interface with fast and flexible I/O control, efficient data manipulation, and high speed computation.

BILL OF MATERIAL

Function	Description	Part No.	Quantity
Controller	BMAC	DP83261	1/2
Controller	PLAYER	DP83251/55	4
Clock Distribution	CDD	DP83241	1(4)
Clock Recovery	CRD	DP83231	4
Controller or	HPC or	HPC16400	1
Processor	GX	NS32GX32	1
Logic	PAL		1
RAM	8k DRAM or		1
	16k SRAM		1
Power	+5V Supply		1



TL/F/11047-1

FIGURE 1. A Router Configuration in a Typical Network

SYSTEM DESCRIPTION

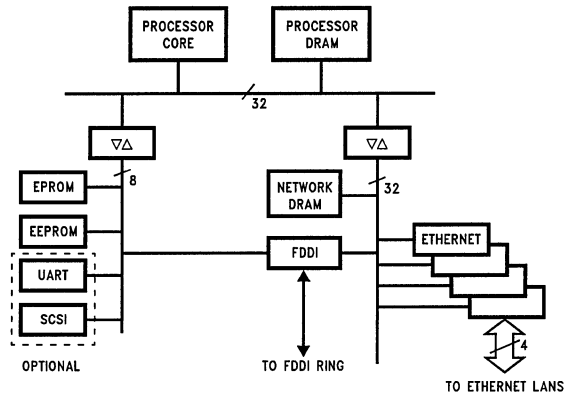
A router connects multiple networks and routes packets between them. *Figure 1* illustrates a typical router configuration. Here, a dual attach FDDI node and four Ethernet ports enable the router to interconnect an FDDI ring and up to four Ethernet LANs. The FDDI and Ethernet interfaces are implemented using high performance peripherals: the DP83200 FDDI chip set and the DP83932 Systems Oriented Network Interface Controller (SONIC™).

TCP/IP is an industry standard for networking and many routers implement the IP protocol. The router presented here implements TCP/IP in full, providing reliable routing of packets across multiple networks. In doing so it offers such services as confirmation of packet delivery, packet routing

and fragmentation, error reporting, address filtering and so on. Such a software intensive application requires a highly integrated, high performance embedded processor and therefore the NS32GX320 was chosen.

Figure 2 illustrates the router architecture. It consists of two 32-bit wide buses that separate network traffic from the processor bus activity. Each contains a 4/16 Mbyte bank of DRAM: one for the GX320 to run its application software and the second for buffering received packets and queuing new ones to be transmitted.

In addition, there is a third "bus" for accessing the FDDI chip set registers and all the 8-bit system devices, such as EPROMs, an EEPROM, a UART and a SCSI controller.



TL/F/11047-2

FIGURE 2. The Router Block Diagram

DESIGN CHALLENGES

Throughput and Bandwidth Considerations

A router must be able to process frames arriving at a peak rate of 100 Mbits/sec over the FDDI ring and 10 Mbits/sec on each of the four Ethernet LANs and then route them back onto the network with minimum latency. This is achieved using the above architecture and National Semiconductor's advanced chips.

Application Software

The application software plays a dominant role in a router. The software design consists of a router program and TCP/IP protocol (compatible with BSD UNIX 4.3 implementations of Routed and TCP/IP, respectively), buffer management and device drivers. Also included is FDDI SMT (Station Management) software. All these modules must run coherently to produce high performance and throughput.

Future Expansion

The router's design accommodates future enhancement and expansion. It may also serve as a hardware platform for other applications such as a multiple network file server or a network printer interface.

KEY COMPONENTS

a. DP83200 FDDI Product.

Includes: BSI, BMAC, PLAYER, CRD, CDD Devices and SMT software. The FDDI chip set fully implements a 32-bit wide system interface, all MAC (Media Access Control) functions and the physical layer interface to the fiber optic ring. The FDDI Product also includes FDDI SMT software.

b. DP83932 SONIC

The Systems Oriented Network Interface Controller provides a 32-bit wide system interface, implements all MAC functions and includes an ENDEC (Encoder-Decoder) for the serial interface to an Ethernet transceiver.

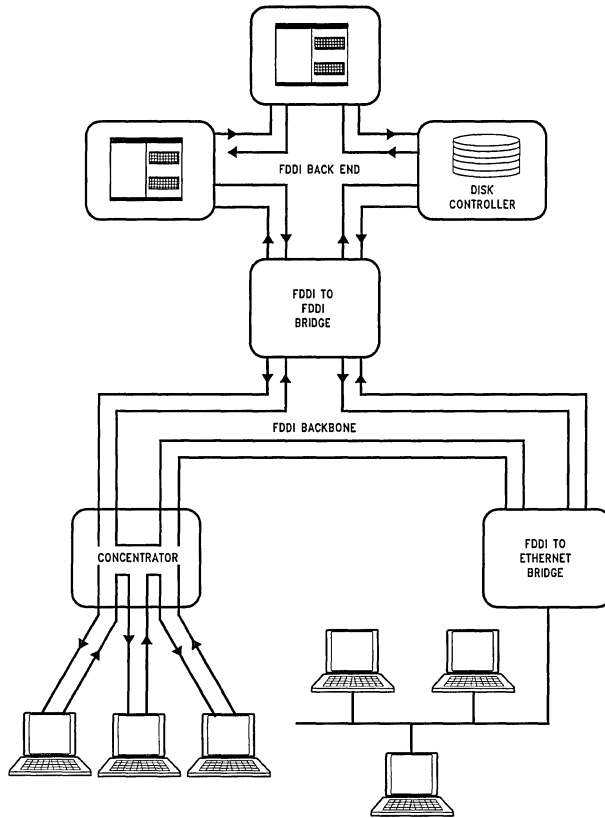
c. NS32GX320

The GX320 is a highly integrated, high performance embedded system processor designed for computation intensive applications. It incorporates a four stage instruction pipeline, on chip instruction and data caches and a hardware multiplier unit. The internal organization allows for a high degree of parallelism in instruction execution.

Integrated on the same chip with the CPU are also a two channel DMA controller, a fifteen level Interrupt Controller Unit and three 16-bit timers.

BILL OF MATERIALS

Function	Description	Part No.	Quantity
System Processor	Embedded Controller	NS32GX320	1
Controller	BSIT [™]	DP83265	1
Controller	BMAC [™]	DP83261	1
Controller	PLAYER [™]	DP83255	2
Clock Recovery	CRD [™]	DP83231	2
Clock Distribution	CDD [™]	DP83241	1
Controller	SONIC	DP83932	1
Memory	4/16 Mbyte Bank of DRAM		2
Logic	PAL [®] s/GAL [®] s		17
	Octels		12
Optical	Transceivers		2
Power	+5V, +12V, -12V		1



TL/F/11048-1

FIGURE 1. System Diagram of Adapter Cards Found in WS/PCs and the Concentrator

SYSTEM DESCRIPTION

Computer vendors have unique system architectures like countries of the world have different languages. The goal of an FDDI adapter card is to bridge the "language barrier" between the host and an FDDI network in a speedy and efficient manner. Choosing the interface that best fits the application is the key to achieving this goal.

The National Semiconductor FDDI chipset provides a simple but powerful interface that can deliver the potential bandwidth of an FDDI network to a wide variety of system architectures. This interface gives the designer the flexibility to define systems which require high network bandwidth with minimal latency or systems in which footprint size and system cost are the most important constraints.

An example of the need for a high bandwidth network can be found in a factory environment. Such a network would be responsible for tying together time critical tasks in a highly reliable manner. An FDDI network, which is fiber based and inherently reliable, is ideally suited for this application. The interface from the FDDI network to the system host must provide low latency and high throughput. With the National Semiconductor FDDI chipset, it is possible to connect di-

rectly to the system bus as a bus master with a peak bandwidth of 96 megabytes per second. In this configuration, the constraints of the design are met with a solution that is efficient and requires little or no external logic.

Workstations that pack supercomputer power, fit into a footprint that fits on a desktop, and cost under \$10,000 are leading edge example of the evolutionary growth of computer technology. This ability to process information at break neck speeds has increased the need for high bandwidth, cost effective networking solutions that can effectively connect these systems. Features of the National Semiconductor chipset allow the designer to tailor the network interface to satisfy constraints imposed by the host architecture. For example, the chipset may be connected directly as a bus master on the system bus or through shared memory which may use low-cost DRAMs or faster SRAMs. Bit ordering and high-speed protocol processing are also handled by the National Semiconductor FDDI implementation. Additionally, the chipset can easily be used to implement an FDDI concentrator which delivers the power of FDDI at a lower cost by reducing the number of networking ICs built into each end station.

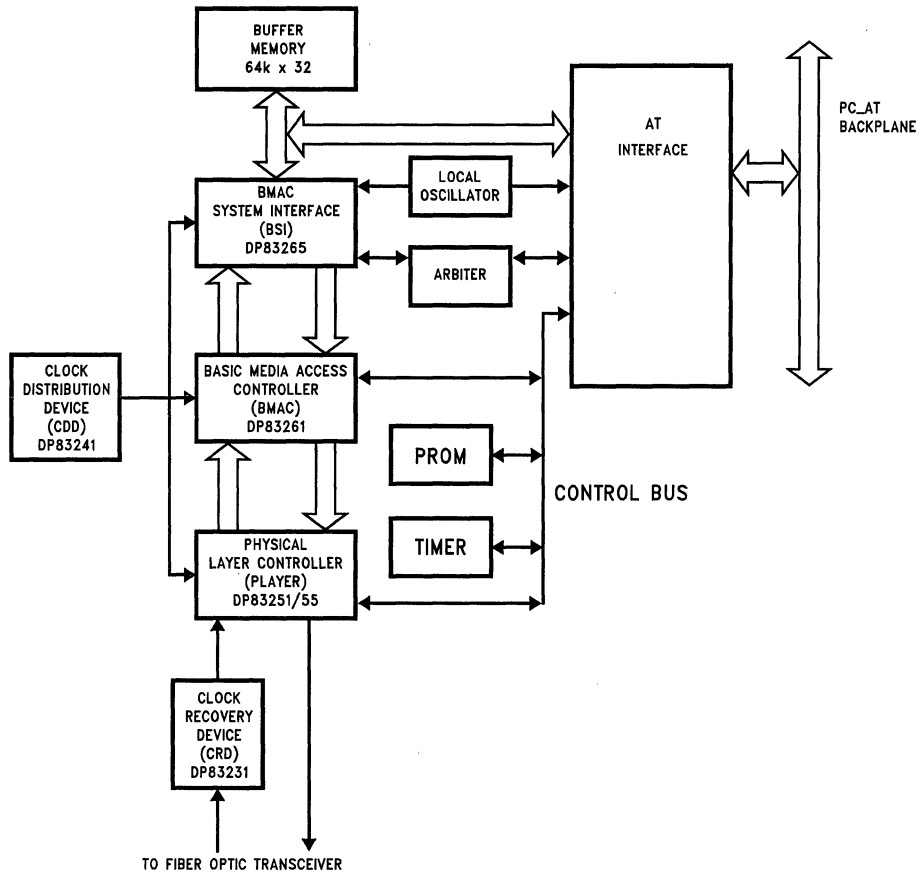


FIGURE 2. Example of AT Based FDDI Adaptor Card

TL/F/11048-2

MAJOR CHALLENGES

1. Choice of an FDDI implementation

A successful adapter card design must first start with the best FDDI solution. The National Semiconductor FDDI chipset offers a full-featured and complete solution.

2. Design of the network/host interface

The design of an FDDI interface must eliminate data bottle necks without demanding excessive design complexity or component count. The National Semiconductor FDDI chipset provides full FDDI bandwidth through a simple but powerful system interface. This interface can be tailored to create an optimal interface to a variety of system architectures.

3. Future integration

In order to maintain a leadership position in FDDI networking, an FDDI vendor must follow the same evolutionary path on the performance/value curve that has been defined by the computer industry. National Semiconductor has developed an aggressive strategy to provide the user with a consistent interface to work with while driving toward a one chip FDDI solution.

KEY COMPONENTS

DP83261 Basic Media Access Controller (BMAC™ Device)

DP83255/51 Physical Layer Controller (PLAYER™ Device)

DP83231 Clock Recovery Device (CRD™ Device)

DP83241 Clock Distribution Device (CDD™ Device)

The four devices listed above compose an FDDI-compliant, full-featured networking solution. The solution offers a full-duplex data pipe that delivers maximum FDDI bandwidth. Additional features include a thorough SMT interface and provisions for the straight forward design of bridges, routers, and concentrators.

DP83265 BMAC System Interface (BSI™ Device)

The BSI provides a simple but powerful system interface. The architecture can be connected directly to the host bus as a bus master or connected to the host through a shared memory architecture which uses low-cost DRAMs or faster SRAMs.

BILL OF MATERIAL

Function	Description	Part No.	Quantity
System I/F	BSI	DP83265	1
Controller	BMAC	DP83261	1
Controller	PLAYER	DP83251/55	1
Clock Recovery	CRD	DP83231	1
Clock Distribution	CDD	DP83241	1
SMT Node Processor	Embedded Controller	HPC46003	1
SMT Timer	Real Time Clock	DP8570A	1
Memory	DRAM/SRAM PROM	(if necessary)	1
Logic	PALS/GALS Octels	(if necessary) (if necessary)	
Fiber Optic Transceivers	125 MHz RX/TX		2



Section 5
Appendix/
Physical Dimensions



Section 5 Contents

FDDI and Networking Acronyms	5-3
Glossary of Terms for FDDI	5-4
Physical Dimensions	5-7
Bookshelf	
Distributors	

FDDI and Networking Acronyms

Acronym	Description
ALS	Active Line State
ANSI	American National Standards Institute
BMAC	Basic Media Access Controller— Part of National's Solution
BSI	BMAC System Interface— Part of National's Solution
CCE	Configuration Control Element
CDD	Clock Distribution Device— Part of National's Solution
CFM	Configuration Management
CMT	Connection Management
CRC	Cyclic Redundancy Check
CRD	Clock Recovery Device— Part of National's Solution
CSMA/CD	Carrier-Sense Multiple Access with Collision Detection
DA	Destination Address
DAC	Dual Attach Concentrator
DAS	Dual Attach Station
DLL	Data Link Layer
ECM	Entity Coordination Management
FC	Frame Control
FCS	Frame Check Sequence
FDDI	Fiber Distributed Data Interface
FTP	File Transfer Protocol
HLS	Halt Line State
IEEE	Institute of Electrical and Electronic Engineers
ILS	Idle Line State
ISO	International Standards Organization
LAN	Local Area Network
LCT	Link Confidence Test
LED	Light Emitting Diode
LEM	Link Error Monitor
LLC	Logical Link Control

Acronym	Description
LSU	Line State Unknown
MAC	Media Access Control Layer
MIB	Management Information Base
MIC	Media Interface Connector
MLS	Master Line State
MPM	MAC Placement Management
NFS	Network File System
NIF	Neighborhood Information Frame
NLS	Noise Line State
NRZ	Non-Return to Zero
NRZI	Non-Return to Zero Invert on Ones
OSI	Open Systems Interconnection
PCM	Physical Connection Management
PDU	Protocol Data Unit
PHY	Physical Layer
PLAYER	Physical Layer Controller— Part of National's Solution
PMD	Physical Medium Dependent Layer
QLS	Quiet Line State
RMT	Ring Management
SA	Source Address
SAS	Single Attach Station
SDU	Service Data Unit
SMT	Station Management
SNMP	Simple Network Management Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
THT	Token Holding Time
TNE	Noise Events Timer
TRT	Token Rotation Timer
TVX	Valid Transmission Timer
WAN	Wide Area Network



Glossary of Terms for FDDI (listed alphabetically)

4B/5B: The symbol coding method specified by the FDDI standard where each set of four bits is encoded as five bits as compared with the Manchester coding method which requires eight bits of coding for each four bit set.

ANSI: The American National Standards Institute. They are responsible for setting many standards, including FDDI.

Asynchronous: A class of data transmission service whereby all requests for service contend for a pool of dynamically allocated ring bandwidth and response time.

Attenuation: Level of optical power loss expressed in units of dB.

Average Power: The optical power measured using an average reading power meter when the FDDI station is retransmitting a stream of Halt symbols.

Backbone Network: A network which interconnects networks via gateways, bridges, and concentrators.

Back End Network: A network which interconnects mainframe computers to high performance mass storage devices, high speed controllers and file servers.

Bandwidth: The level of communication capability of a transmission link. The greater the bandwidth, the greater the volume of information the link can carry in a given time.

BEACON Frame: Frame sent during the BEACON process to indicate and recover from a break in the ring.

Bit: A single character of a language having just two characters, as either of the binary digits 0 or 1.

Bypass: The ability of a station to be optically isolated from the network while maintaining the integrity of the ring.

Cable Plant: This term refers to the installed cabling, connectors, splices and patch panels within a given plant.

Carrier Source: The component of a fiber optic system which generates the light wave, or carrier, on which information is transmitted. e.g., LEDs or LDs.

Capture: The act of removing a Token from the ring for the purpose of Frame transmission.

Center Wavelength: The average of the two wavelengths measured at the half amplitude points of the power spectrum.

CLAIM Frame: Frame sent during the CLAIM process whereby one or more stations bid for the right to initialize the ring.

Code-Bit: The smallest signaling element used by the Physical Layer for transmission on the medium.

Code Group: The specific sequence of five code bits representing an FDDI symbol.

Concentrator: A Node on the FDDI ring, which in turn provides connections for multiple FDDI stations so that they may communicate with other attachments to the FDDI ring. A concentrator has at least two Physical Layer entities and may or may not have one or more Data Link Layer entities. It provides a logical star topology while stations are physically connected as a ring. The concentrator (or center of the star topology) can actively bypass a station connected to it.

Connection Management (CMT): That portion of the Station Management (SMT) function that controls network insertion, removal, and connection of PHY and MAC entities within a station.

Connector Plug: A device used to terminate an optical conductor(s) cable.

Connector Receptacle: The fixed or stationary half of a connection that is mounted on a panel/bulkhead. Receptacles mate with plugs.

Dual Attachment Concentrator (DAC): A concentrator that offers two attachments to the FDDI network which are capable of accommodating a dual (counter-rotating) ring.

Dual Station (or dual attachment station): A station that offers two attachments to the FDDI network which are capable of accommodating a dual (counter-rotating) ring. It may offer additional attachments (see concentrator).

Entity: An active element, or functional agent, within an Open System Interconnection (OSI) layer, or sublayer, or SMT, in a specific station, including both operational and management functions.

Extinction Ratio: The ratio of the low, or off optical power level, (PL) to the high, or on optical power level, (PH) when the station is transmitting a stream of Halt symbols.

$$\text{Extinction Ratio (\%)} = (PL/PH) * 100$$

FDDI: Fiber Distributed Data Interface.

Fiber: Dielectric material that guides light; waveguide.

Fiber Optic Cable: A jacketed fiber(s).

Fiber Optics: A technology whereby signals are transmitted over an optical waveguide medium through the use of light-generating transmitters and light-detecting receivers.

Frame: A Protocol Data Unit (PDU) transmitted between cooperating MAC entities on a ring, consisting of a variable number of octets.

Front End Network: A network which interconnects workstations, word processors, personal computers, facsimiles, terminals, and printers.

Hub: An active fiber optic device which allows multiple connections. Signals are transmitted on all connections or ports. Signals are not retimed.

IEEE: The Institute of Electrical and Electronic Engineers (American). They are active in setting LAN standards. It has established a number of technical committees, prefixed by IEEE 802 (e.g., 802.3, 802.5, 802.6).

Interchannel Isolation: The ability to prevent undesired optical energy from appearing in one signal path as a result of coupling from another signal path: cross talk.

Jitter, Data Dependent (DDJ): Jitter that is related to the transmitted symbol sequence. DDJ is caused by the limited bandwidth characteristics and imperfections in the optical channel components. DDJ results from non-ideal individual pulse responses and from variation in the average value of the encoded pulse sequence which may cause base-line wander and may change the sampling threshold level in the receiver.

Jitter, Duty Cycle Distortion (DCD): Distortion usually caused by propagation delay differences between low-to-high and high-to-low transitions. DCD is manifested as a pulse width distortion of the nominal baud time.

Jitter, Random (RJ): RJ is due to thermal noise and may be modeled as a Gaussian process. The peak-peak value of RJ is of a probabilistic nature and thus any specific value requires an associated probability.

LAN: A Local Area Network is a communications network that provides interconnection of a variety of data communicating devices within a small area (e.g., a single site or group of buildings).

LED: A Light Emitting Diode is an electrical component which produces light when stimulated by electricity. It is commonly used as a method for transmitting infra-red light along an optical fiber.

LLC: Logical Link Control (IEEE 802.2) is the part of the ISO 7 layer model which is responsible for controlling the flow of information over the link between stations.

Logical Ring: A network which is treated logically as a ring even though it may be cabled as a physical star topology.

Media Access Control (MAC): The Data Link Layer (DLL) responsible for scheduling and routing data transmissions on a shared medium Local Area Network (LAN) (e.g., an FDDI ring).

Media Interface Connector (MIC): An optical fiber connector which connects the fiber media to the FDDI attachment. The MIC consists of two halves. The MIC plug is the male half used to terminate an optical fiber signal transmission cable. The MIC receptacle is the female half which is associated with the FDDI attachment.

MIC Plug: The male half of the MIC which terminates an optical signal transmission cable.

MIC Receptacle: The fixed or stationary female half of the MIC which is part of an FDDI station.

Modulator: The component of a fiber optic system which converts the electrical message into the proper format.

Multimode: An optical fiber which allows the signal carrying light to travel along more than one path.

Network: A set of communication channels interconnecting several or many locations.

Node: A generic term applying to any FDDI network attachment (station, concentrator, or repeater).

Nonrestricted Token: A Token denoting the normal mode of asynchronous bandwidth allocation, wherein the available bandwidth is time-sliced among all requesters.

Nonreturn to Zero (NRZ): A technique in which a polarity level high, or low, represents a logical "1" (one), or "0" (zero).

Nonreturn to Zero Invert on Ones (NRZI): A technique in which a polarity transition represents a logical "1" (one). The absence of a polarity transition denotes a logical "0" (zero).

Numerical Aperture (NA): The sine of the radiation or acceptance half-angle of an optical fiber, multiplied by the refractive index of the material in contact with the exit or entrance face.

Octet: A data unit composed of eight ordered bits (a pair of data symbols).

Optical Fall Time: The time interval for the falling edge of an optical pulse to transition from 90% to 10% of the pulse amplitude.

Optical Reference Plane: The plane that defines the optical boundary between the MIC Plug and the MIC Receptacle.

Optical Rise Time: The time interval for the rising edge of an optical pulse to transition from 10% to 90% of the pulse amplitude.

OSI: The Open Systems Interconnect is an objective designed to obtain the most effective internetworking. Namely, it allows different devices to communicate without regard to the manufacturer. It is obtained by adhering to appropriate international standards throughout the system.

Physical (PHY): The Physical Layer responsible for delivering a symbol stream produced by an upstream MAC Transmitter to the logically adjacent downstream MAC Receiver in an FDDI ring.

Physical Connection: The full-duplex physical layer association between adjacent PHY entities (in concentrators, repeaters, or stations) in an FDDI ring, i.e., a pair of Physical Links.

Physical Link: The simplex path (via PMD and attached medium) from the transmit function of one PHY entity to the receive function on an adjacent PHY entity (in concentrators, repeaters, or stations) in an FDDI ring.

Physical Medium Independent (PMD): The portion of the FDDI protocol which provides the digital baseband point to point communication between stations in the FDDI network. It specifies the point of interconnection requirements for conforming station and cable plants (i.e., optical power budget, MIC receptacle mating, optic cable specifications, and services provided to the PHY and SMT layers).

Primitive: An element of the services provided by one entity to another.

Protocol Data Unit (PDU): Information delivered as a unit between peer entities that may contain control information, address information, and data (e.g., a Service Data Unit (SDU) from a higher layer) or any combination of the three. The FDDI MAC PDUs are Tokens and Frames.

Receive: The action of a station of accepting a frame, token, or control sequence from the medium.

Receiver: An optoelectronic circuit that converts an optical signal to an electrical logic signal.

Repeat: The action of a station in receiving a code-bit stream (e.g., frame or token) from an upstream station and placing it on the medium to the next station. The station repeating the code-bit stream examines it and may copy it into a buffer and modify control indicators as appropriate.

Repeater: An FDDI node that minimally comprises the functionality of two PMDs and provides only a repeat function. A repeater does not have any MACs or concentrator functionality.

Restricted Token: A Token denoting a special mode of asynchronous bandwidth allocation, wherein the bandwidth available for the asynchronous class of service is dedicated to a single extended dialogue between specific requesters.

Ring: Two or more stations connected by a physical medium wherein information is passed sequentially between active stations, each station in turn examining or copying the information, finally returning it to the originating station.

Ring Management: The part of SMT which ensures the integrity of unique addresses on the FDDI ring.

Service Data Unit (SDU): The unit of data transfer between a service user and a service provider.

Services: A set of functions, or services, provided by one OSI layer or sublayer entity, for use by a higher layer or sublayer entity or by management entities.

Singlemode: An optical fiber which allows the signal carrying light to travel along only one path. It is also called monomode.

Single Attachment Concentrator (SAC): A concentrator that offers one attachment to the FDDI network.

Single Station (or Single Attachment Station): A station that offers one attachment to the FDDI network.

Spectral Width, Full Width Half Maximum (FWHM): The absolute difference between the wavelengths at which the spectral radiant intensity is 50.0 percent of the maximum power.

ST: One type of connector used for terminating optical fibers.

Star: A network configuration where all the nodes are connected to one common central point via individual cables.

Station: An addressable logical and physical node on an FDDI ring capable of transmitting, repeating and receiving information.

Station Management (SMT): The supervisory entity within an FDDI station that monitors station activity and exercises overall appropriate control of station activity.

Symbol: The smallest signaling element used by the Data Link Layer (DLL). The symbol set consists of 16 data symbols and 8 control symbols. Each symbol corresponds to a specific sequence of code bits (code group) to be transmitted by the Physical Layer.

Synchronous: A class of data transmission service whereby each requester is preallocated a maximum bandwidth and guaranteed a response time not to exceed a specific delay.

TCP/IP: A widely used de facto standard transport/network protocol. It requires a Type Field when carried over Ethernet V2 networks.

Token: An explicit indication of the right to transmit on a shared medium. On a Token Ring, the Token circulates sequentially through the stations in the ring. At any time, it may be held by zero or one station. FDDI uses two classes of Tokens: restricted and non-restricted.

Topology: The layout or configuration of a network. The principal network topologies are star, bus, and ring.

Transmit: The action of a station that consists of generating a frame, token, or control sequence, and placing it on the medium to the next station.

Transmitter: An optoelectronic circuit that converts an electrical logic signal to an optical signal.

TRRT: The Target Token Rotation Time is the target time for the token to pass every FDDI node in the token path.

WAN: A Wide Area Network is a network that uses links provided by the postal, telegraph, and telephone (PT&T) administrations and usually connects dispersed locations (e.g., greater than 50 miles).

Wavelength: A measurement of the length of any electromagnetic wave. The shorter the wavelength, the higher the frequency.

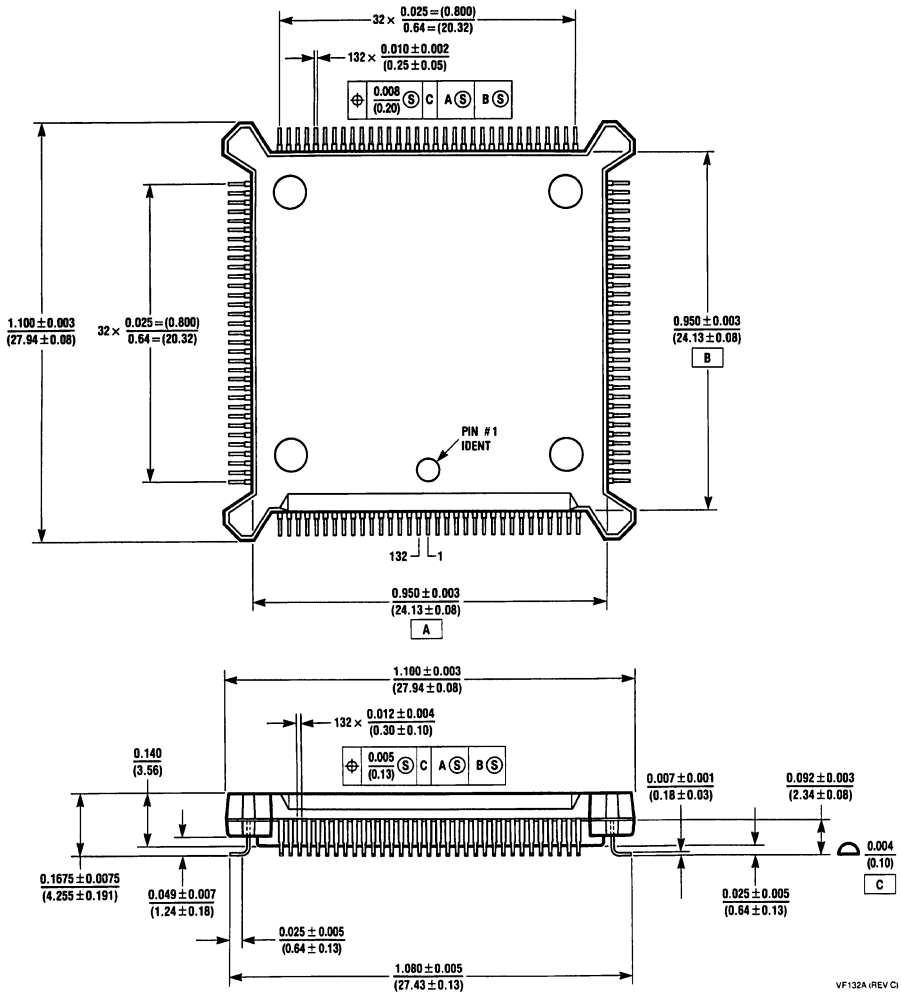
Workstation: An end user device typically comprised of high resolution screens, local graphics processing, keyboard, pointing device, and network connection.

X3T9.5: The ANSI committee responsible for specifying the FDDI standard.

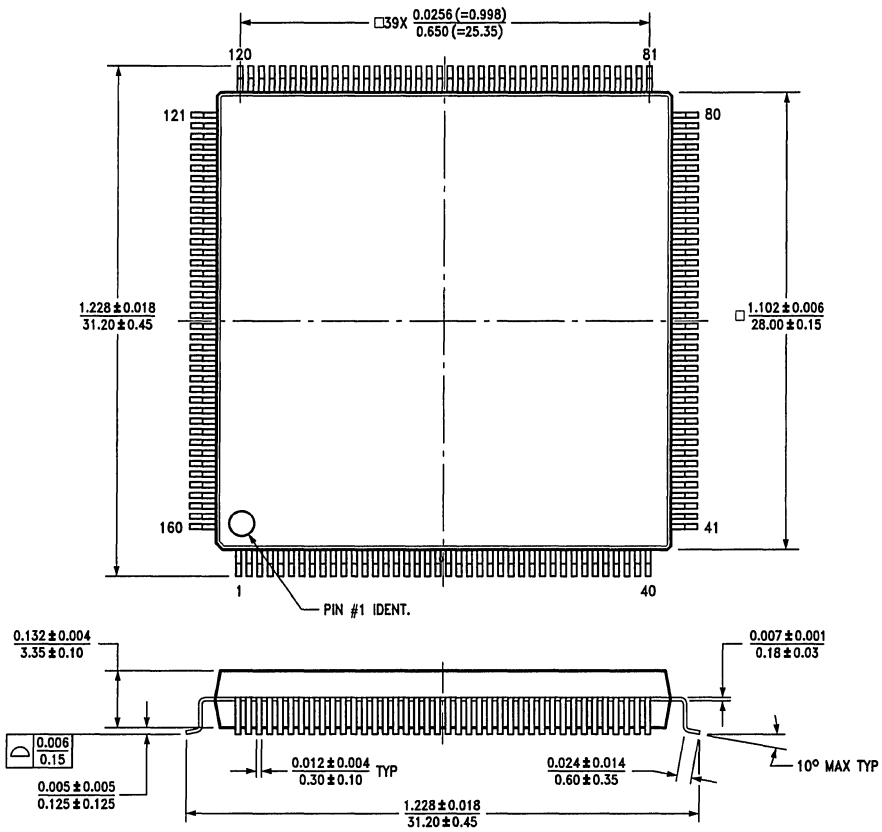
Sources

1. ANSI X3T9.5 Protocol Documents for PMD, PHY, MAC and SMT Layers.
2. BICC Booklet "FDDI, Business Communications for the 1990's".
3. Handbook of Computer Communications Standards, V2 (W. Stallings).

132 Lead Plastic Quad Flatpack (VF) NS Package Number VF132A



160 Lead Plastic Quad Flatpack (VF) NS Package Number VF160A



NOTES



Bookshelf of Technical Support Information

National Semiconductor Corporation recognizes the need to keep you informed about the availability of current technical literature.

This bookshelf is a compilation of books that are currently available. The listing that follows shows the publication year and section contents for each book.

Please contact your local National sales office for possible complimentary copies. A listing of sales offices follows this bookshelf.

We are interested in your comments on our technical literature and your suggestions for improvement.

Please send them to:

Technical Communications Dept. M/S 16-300
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090

ALS/AS LOGIC DATABOOK—1990

Introduction to Advanced Bipolar Logic • Advanced Low Power Schottky • Advanced Schottky

ASIC DESIGN MANUAL/GATE ARRAYS & STANDARD CELLS—1987

SSI/MSI Functions • Peripheral Functions • LSI/VLSI Functions • Design Guidelines • Packaging

CMOS LOGIC DATABOOK—1988

CMOS AC Switching Test Circuits and Timing Waveforms • CMOS Application Notes • MM54HC/MM74HC
MM54HCT/MM74HCT • CD4XXX • MM54CXXX/MM74CXXX • Surface Mount

DATA ACQUISITION LINEAR DEVICES—1989

Active Filters • Analog Switches/Multiplexers • Analog-to-Digital Converters • Digital-to-Analog Converters
Sample and Hold • Temperature Sensors • Voltage Regulators • Surface Mount

DATA COMMUNICATION/LAN/UART DATABOOK—1990

LAN IEEE 802.3 • High Speed Serial/IBM Data Communications • ISDN Components • UARTs
Modems • Transmission Line Drivers/Receivers

DISCRETE SEMICONDUCTOR PRODUCTS DATABOOK—1989

Selection Guide and Cross Reference Guides • Diodes • Bipolar NPN Transistors
Bipolar PNP Transistors • JFET Transistors • Surface Mount Products • Pro-Electron Series
Consumer Series • Power Components • Transistor Datasheets • Process Characteristics

DRAM MANAGEMENT HANDBOOK—1989

Dynamic Memory Control • Error Detection and Correction • Microprocessor Applications for the
DP8408A/09A/17/18/19/28/29 • Microprocessor Applications for the DP8420A/21A/22A
Microprocessor Applications for the NS32CG821

EMBEDDED SYSTEM PROCESSOR DATABOOK—1989

Embedded System Processor Overview • Central Processing Units • Slave Processors • Peripherals
Development Systems and Software Tools

FDDI DATABOOK—1991

FDDI Overview • DP83200 FDDI Chip Set • Development Support • Application Notes and System Briefs

F100K ECL LOGIC DATABOOK & DESIGN GUIDE—1990

Family Overview • 300 Series (Low-Power) Datasheets • 100 Series Datasheets • 11C Datasheets
ECL BiCMOS SRAM, ECL PAL, and ECL ASIC Datasheets • Design Guide • Circuit Basics • Logic Design
Transmission Line Concepts • System Considerations • Power Distribution and Thermal Considerations
Testing Techniques • Quality Assurance and Reliability • Application Notes

FACT™ ADVANCED CMOS LOGIC DATABOOK—1990

Description and Family Characteristics • Ratings, Specifications and Waveforms
Design Considerations • 54AC/74ACXXX • 54ACT/74ACTXXX • Quiet Series: 54ACQ/74ACQXXX
Quiet Series: 54ACTQ/74ACTQXXX • 54FCT/74FCTXXX • FCTA: 54FCTXXXA/74FCTXXXA

FAST® ADVANCED SCHOTTKY TTL LOGIC DATABOOK—1990

Circuit Characteristics • Ratings, Specifications and Waveforms • Design Considerations • 54F/74FXXX

FAST® APPLICATIONS HANDBOOK—1990

Reprint of 1987 Fairchild FAST Applications Handbook

Contains application information on the FAST family: Introduction • Multiplexers • Decoders • Encoders
Operators • FIFOs • Counters • TTL Small Scale Integration • Line Driving and System Design
FAST Characteristics and Testing • Packaging Characteristics

GENERAL PURPOSE LINEAR DEVICES DATABOOK—1989

Continuous Voltage Regulators • Switching Voltage Regulators • Operational Amplifiers • Buffers • Voltage Comparators
Instrumentation Amplifiers • Surface Mount

GRAPHICS HANDBOOK—1989

Advanced Graphics Chipset • DP8500 Development Tools • Application Notes

INTERFACE DATABOOK—1990

Transmission Line Drivers/Receivers • Bus Transceivers • Peripheral Power Drivers • Display Drivers
Memory Support • Microprocessor Support • Level Translators and Buffers • Frequency Synthesis • Hi-Rel Interface

LINEAR APPLICATIONS HANDBOOK—1986

The purpose of this handbook is to provide a fully indexed and cross-referenced collection of linear integrated circuit applications using both monolithic and hybrid circuits from National Semiconductor.

Individual application notes are normally written to explain the operation and use of one particular device or to detail various methods of accomplishing a given function. The organization of this handbook takes advantage of this innate coherence by keeping each application note intact, arranging them in numerical order, and providing a detailed Subject Index.

LS/S/TTL DATABOOK—1989

Contains former Fairchild Products

Introduction to Bipolar Logic • Low Power Schottky • Schottky • TTL • TTL—Low Power

MASS STORAGE HANDBOOK—1989

Rigid Disk Pulse Detectors • Rigid Disk Data Separators/Synchronizers and ENDECs
Rigid Disk Data Controller • SCSI Bus Interface Circuits • Floppy Disk Controllers • Disk Drive Interface Circuits
Rigid Disk Preamplifiers and Servo Control Circuits • Rigid Disk Microcontroller Circuits • Disk Interface Design Guide

MEMORY DATABOOK—1990

PROMs, EPROMs, EEPROMs • TTL I/O SRAMs • ECL I/O SRAMs

MICROCONTROLLER DATABOOK—1989

COP400 Family • COP800 Family • COPS Applications • HPC Family • HPC Applications
MICROWIRE and MICROWIRE/PLUS Peripherals • Microcontroller Development Tools

MICROPROCESSOR DATABOOK—1989

Series 32000 Overview • Central Processing Units • Slave Processors • Peripherals
Development Systems and Software Tools • Application Notes • NSC800 Family

PROGRAMMABLE LOGIC DATABOOK & DESIGN MANUAL—1990

Product Line Overview • Datasheets • Designing with PLDs • PLD Design Methodology • PLD Design Development Tools
Fabrication of Programmable Logic • Application Examples

REAL TIME CLOCK HANDBOOK—1989

Real Time Clocks and Timer Clock Peripherals • Application Notes

RELIABILITY HANDBOOK—1986

Reliability and the Die • Internal Construction • Finished Package • MIL-STD-883 • MIL-M-38510
The Specification Development Process • Reliability and the Hybrid Device • VLSI/VHSIC Devices
Radiation Environment • Electrostatic Discharge • Discrete Device • Standardization
Quality Assurance and Reliability Engineering • Reliability and Documentation • Commercial Grade Device
European Reliability Programs • Reliability and the Cost of Semiconductor Ownership
Reliability Testing at National Semiconductor • The Total Military/Aerospace Standardization Program
883B/RETS™ Products • MILS/RETSTM Products • 883/RETS™ Hybrids • MIL-M-38510 Class B Products
Radiation Hardened Technology • Wafer Fabrication • Semiconductor Assembly and Packaging
Semiconductor Packages • Glossary of Terms • Key Government Agencies • AN/ Numbers and Acronyms
Bibliography • MIL-M-38510 and DESC Drawing Cross Listing

SPECIAL PURPOSE LINEAR DEVICES DATABOOK—1989

Audio Circuits • Radio Circuits • Video Circuits • Motion Control Circuits • Special Function Circuits
Surface Mount

TELECOMMUNICATIONS—1990

Line Card Components • Integrated Services Digital Network Components • Analog Telephone Components
Application Notes



National Semiconductor Corporation

2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1-800-272-9959
TWX: (910) 339-9240

SALES OFFICES (Continued)

INTERNATIONAL OFFICES

Electronica NSC de Mexico SA

Juventino Rosas No. 118-2
Col Guadalupe Inn
Mexico, 01020 D.F. Mexico
Tel: 52-5-524-9402

National Semicondutores

Do Brasil Ltda.

Av. Brig. Faria Lima, 1383
6.0 Andor-Conj. 62
01451 Sao Paulo, SP, Brasil
Tel: (55/11) 212-5066
Fax: (55/11) 211-1181 NSBR BR

National Semiconductor GmbH

Industriestrasse 10
D-8080 Furstenfeldbruck
West Germany
Tel: (0-81-41) 103-0
Telex: 527-649
Fax: (08141) 103554

National Semiconductor (UK) Ltd.

The Maple, Kembrey Park
Swindon, Wiltshire SN2 6UT
United Kingdom
Tel: (07-93) 61-41-41
Telex: 444-674
Fax: (07-93) 69-75-22

National Semiconductor Benelux

Vorstlaan 100
B-1170 Brussels
Belgium
Tel: (02) 6-61-06-80
Telex: 61007
Fax: (02) 6-60-23-95

National Semiconductor (UK) Ltd.

Ringager 4A, 3
DK-2605 Brøndby
Denmark
Tel: (02) 43-32-11
Telex: 15-179
Fax: (02) 43-31-11

National Semiconductor S.A.

Centre d'Affaires-La Boursidiere
Bâtiment Champagne, B.P. 90
Route Nationale 186
F-92357 Le Plessis Robinson
France
Tel: (1) 40-94-88-88
Telex: 631065
Fax: (1) 40-94-88-11

National Semiconductor (UK) Ltd.

Unit 2A
Clonskeagh Square
Clonskeagh Road
Dublin 14
Tel: (01) 69-55-89
Telex: 91047
Fax: (01) 69-55-89

National Semiconductor S.p.A.

Strada 7, Palazzo R/3
I-20089 Rozzano
Milanofiori
Italy
Tel: (02) 8242046/7/8/9
Twx: 352647
Fax: (02) 8254758

National Semiconductor S.p.A.

Via del Cararaggio, 107
00147 Rome
Italy
Tel: (06) 5-13-48-80
Fax: (06) 5-13-79-47

National Semiconductor (UK) Ltd.

Isveien 45
Postboks 57
N-1393 Ostenstad
Norway
Tel: (2) 796500
Fax: (2) 796040

National Semiconductor AB

P.O. Box 1009
Grosshandlarvaegen 7
S-121 23 Johanneshov
Sweden
Tel: 46-8-7228050
Fax: 46-8-7229095
Telex: 10731 NSC S

National Semiconductor GmbH

Calle Agustin de Foxa, 27 (9°D)
28036 Madrid
Spain
Tel: (01) 733-2958
Telex: 46133
Fax: (01) 733-8018

National Semiconductor Switzerland

Alte Winterthurerstrasse 53
Postfach 567
Ch-8304 Wallisellen-Zurich
Switzerland
Tel: (01) 830-2727
Telex: 828-444
Fax: (01) 830-1900

National Semiconductor

Kaupparkatanonkatu 7 A22
SF-00930 Helsinki
Finland
Tel: (90) 33-80-33
Telex: 126116
Fax: (90) 33-81-30

National Semiconductor

Postbus 90
1380 AB Weesp
The Netherlands
Tel: (0-29-40) 3-04-48
Telex: 10-956
Fax: (0-29-40) 3-04-30

National Semiconductor Japan Ltd.

Sanseido Bldg. 5F
4-15 Nishi Shinjuku
Shinjuku-ku
Tokyo 160 Japan
Tel: 3-299-7001
Fax: 3-299-7000

National Semiconductor

Hong Kong Ltd.
Suite 513, 5th Floor,
Chinachem Golden Plaza,
77 Mody Road, Tsimshatsui East,
Kowloon, Hong Kong
Tel: 3-7231290
Telex: 52996 NSSEA HX
Fax: 3-3112536

National Semiconductor

(Australia) PTY, Ltd.
1st Floor, 441 St. Kilda Rd.
Melbourne, 3004
Victoria, Australia
Tel: (03) 267-5000
Fax: 61-3-2677458

National Semiconductor (PTE), Ltd.

200 Cantonment Road 13-01
Southpoint
Singapore 0208
Tel: 2252226
Telex: RS 33877

National Semiconductor (Far East) Ltd.

Taiwan Branch

P.O. Box 68-332 Taipei
7th Floor, Nan Shan Life Bldg.
302 Min Chuan East Road,
Taipei, Taiwan R.O.C.
Tel: (86) 02-501-7227
Telex: 22837 NSTWI
Cable: NSTW TAIPEI

National Semiconductor (Far East) Ltd.

Korea Branch

13th Floor, Dai Han Life Insurance
63 Building,
60, Yoido-dong, Youngdeungpo-ku,
Seoul, Korea 150-763
Tel: (02) 784-8051/3, 785-0696/8
Telex: 24942 NSPKLO
Fax: (02) 784-8054