

## 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-2™ 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

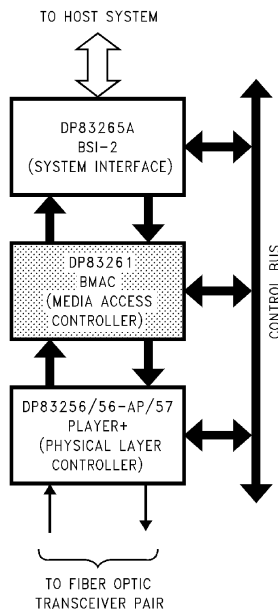


FIGURE 1-1. FDDI Chip Set Block Diagram  
TL/F/10387-1

TRI-STATE® is a registered trademark of National Semiconductor Corporation.  
BSI-2™, BMAC™, PLAYER+™, CDD™ and CRD™ are trademarks of National Semiconductor Corporation.

## 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.

### DP83256/56-AP/57 PLAYER+™ Device Physical Layer Controller

The PLAYER+ device implements the Physical Layer (PHY) protocol as defined by the ANSI FDDI PHY X3T9.5 Standard, along with all the necessary clock recovery and clock regeneration functions.

#### Features

- Single chip FDDI Physical Layer (PHY) solution
- Integrated Digital Clock Recovery Module provides enhanced tracking and greater lock acquisition range
- Integrated Clock Generation Module provides all necessary clock signals for an FDDI system from an external 12.5 MHz reference
- Alternate PMD Interface (DP83256-AP/57) supports UTP twisted pair FDDI PMDs with no external clock recovery or clock generation functions required
- No External Filter Components
- Connection Management (CMT) Support (LEM, TNE, PC\_React, CF\_React, Auto Scrubbing)
- Full on-chip configuration switch
- Low Power CMOS-BIPOLAR design using a single 5V supply
- Full duplex operation with through parity
- Separate management interface (Control Bus)
- Selectable Parity on PHY-MAC Interface and Control Bus Interface
- Two levels of on-chip loopback
- 4B/5B encoder/decoder
- Framing logic
- Elasticity Buffer, Repeat Filter and Smoother
- Line state detector/generator
- Supports single attach stations, dual attach stations and concentrators with no external logic
- DP83256/56-AP for SAS/DAS single path stations
- P83257 for SAS/DAS single/dual path stations

In addition, the DP83257 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

### DP83265A BSI-2 Device System Interface

The BSI-2 device implements an interface between the BMAC device and a host system.

#### Features

- Fully software and pin compatible with the original BSI device
- Over 2 kbytes of on-chip FIFO
- Operates from 12.5 MHz to 33 MHz synchronously with host system
- Provides Address bit swapping capability
- Reduces interface logic for SBus adapters
- 32-bit wide Address/Data path with byte parity
- Programmable transfer burst sizes of 4 or 8 32-bit words
- Interfaces to DRAMs or directly to system bus
- 2 Output and 3 Input Channels
- Supports Header/Info splitting
- Bridging support
- Programmable Big or Little Endian alignment
- Full duplex data path
- 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.

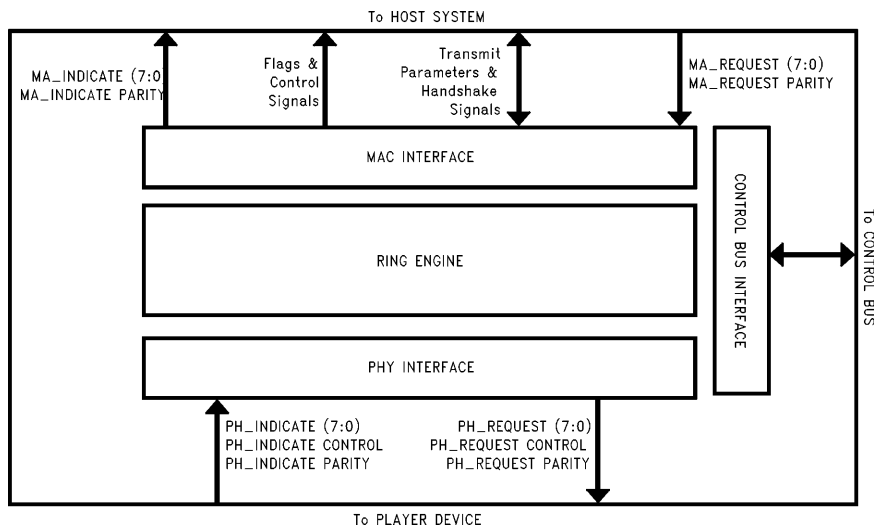


FIGURE 2-1. BMAC Device Interfaces

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.

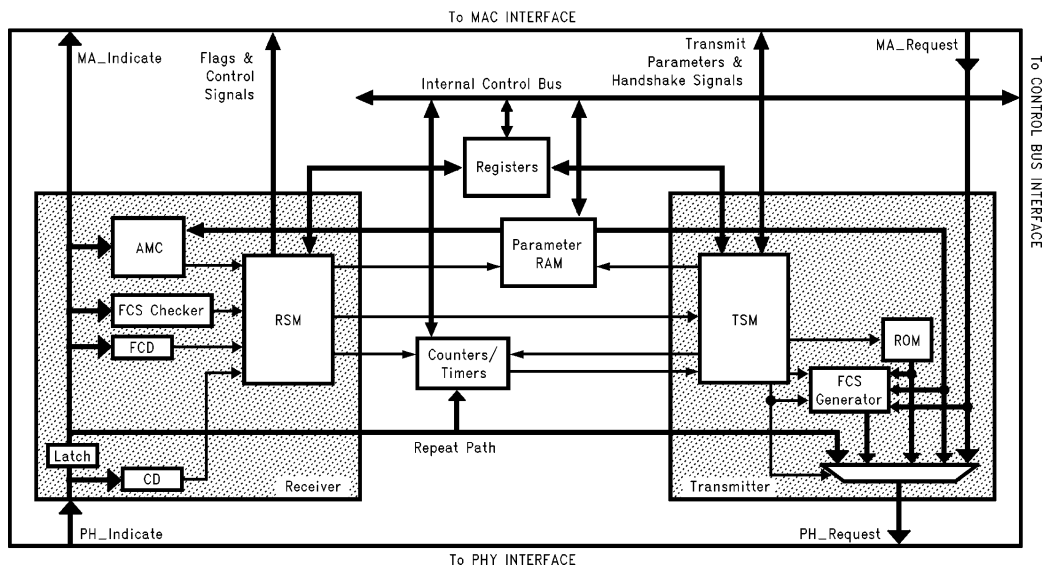


FIGURE 2-2. Ring Engine Overview Block Diagram

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 DP83265A BSI-2) 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.

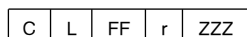
The Starting Delimiter is used to indicate the start of a new PDU. The Starting Delimiter is the JK symbol pair.

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	00	Null	MSA	FCS	TRRR
Void	Not ESA	Long	PA	SD	40	Null	MLA	FCS	TRRR
My_Void	ESA	Short	PA	SD	00	MSA	MSA	FCS	TRRR
My_Void	Not ESA	Long	PA	SD	40	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 (TELRLVLD) is one.

The Latency Counter increments every 16 byte times (1.28  $\mu$ s) and is used to measure ring latencies up to 1.3421772 seconds directly with an accuracy of 1.2  $\mu$ 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)

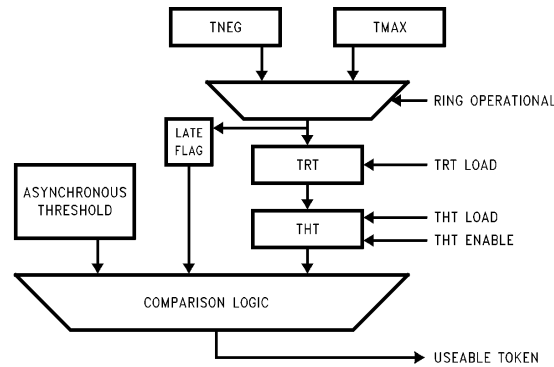


FIGURE 5-1. Token Timing Logic

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	R
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.

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  $\mu$ 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.

- a. The MAC Transmitter is in states T0, T1 or T3;
- b. Bits ITC and IRR of the Option Register are set to One.
- c. Bits CLM and BCN of the Function Register are not set to One.

**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	LFCT1	Null(7–4), LFCT(19–16)
AA	LFCT2	LFCT(15–8)
AB	LFCT3	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><b>RUN/Stop:</b></p> <p>0: Stop Mode 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><b>Control Bus Parity:</b> 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><b>MAC Request Parity:</b> 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><b>PHY Indicate Parity:</b> 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	<b>Reserved</b>
D6	ILB	<p><b>Internal Loopback:</b> 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><b>Diagnose Mode:</b> 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><b>Enable Short Addressing:</b> 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><b>Enable Long Addressing:</b> 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><b>Inhibit Token Release:</b> 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><b>Inhibit Recovery Required:</b> 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><b>Inhibit Repeat:</b> When enabled,</p> <ol style="list-style-type: none"> <li>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.</li> <li>Void frames are not transmitted during a service opportunity.</li> <li>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).</li> </ol> <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><b>Implementer FCS:</b> 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><b>External Matching Indicators:</b> 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><b>Inhibit Token Capture:</b> 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><b>Master Reset:</b> 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	<b>Reserved</b>
D2	MCRST	<p><b>MAC Reset:</b> 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><b>Beacon Request:</b> 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><b>Claim Request:</b> 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	<b>Reserved</b>

## 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
07h	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><b>Revision Number:</b> 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"> <li>• Programmable Group Address Modification</li> <li>• Not copied count does not increment on reception of NSA frames with Ar = S</li> <li>• Detection of Idle on reception of nl</li> <li>• Generation of ODD Parity at all times</li> <li>• Reset of Latency Count on initiation of new measurement</li> </ul>

## 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 Mask 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 (TEMRO)
- 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)

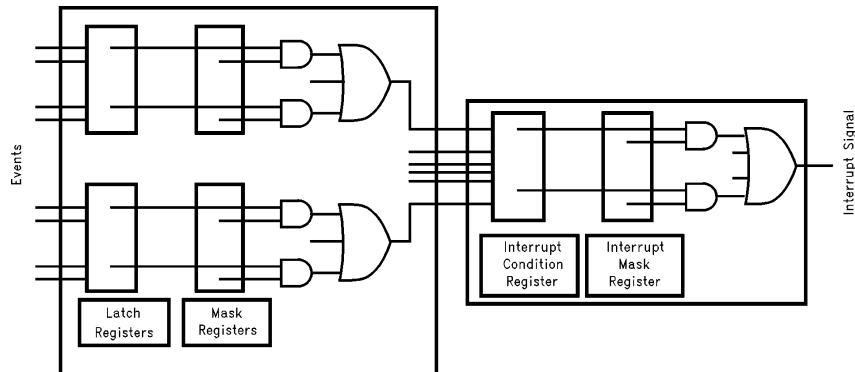


FIGURE 6-1. Event Registers Hierarchy

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><b>Receive Timing State:</b> 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	<b>Reserved</b>																																				
D4–6	RS(0–2)	<p><b>Receive State:</b> 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 &amp; 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><b>R__Flag:</b> 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><b>TRANSMIT TIMING STATE:</b> 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 &amp; 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 &amp; FS</td> </tr> <tr> <td colspan="4" style="text-align: center;">9h–Fh</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><b>Transmit State:</b> 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	<b>Ring Operational Flag:</b> 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	<b>Ring Operational Set:</b> Is set when the Local Ring Operational flag transitions from 0 to 1.
D1	RNOP	<b>Ring Non-Operational Set:</b> Is set when the Local Ring Operational flag transitions from 1 to 0.
D2	BCNR	<b>Beacon Frame Received:</b> 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	<b>Claim Frame Received:</b> 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	<b>Other MAC Frame Received:</b> Indicates that a MAC frame other than a Beacon or Claim frame was received. When set, restricted requests are not serviced.
D5	PINV	<b>PHY_Invalid Received:</b> 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	<b>Duplicate Address Received:</b> 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	<b>Reserved</b>

## 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	<b>Ring Operational Mask:</b> This bit is used to mask RELR0.ROP.
D1	RNOP	<b>Ring Non-Operational Mask:</b> This bit is used to mask RELR0.RNOP.
D2	BCNR	<b>Beacon Frame Mask:</b> This bit is used to mask RELR0.BCNR.
D3	CLMR	<b>Claim Frame Mask:</b> This bit is used to mask RELR0.CLMR.
D4	OTRMAC	<b>Other MAC Frame Mask:</b> This bit is used to mask RELR0.OTRMAC.
D5	PINV	<b>PHY_Invalid Mask:</b> This bit is used to mask RELR0.PINV.
D6	DUPADD	<b>Duplicate Address Mask:</b> This bit is used to mask RELR0.DUPADD.
D7	RES	<b>Reserved</b>

## 6.0 Control Information (Continued)

### Ring Event Latch Register 1 (RELR1)

The Ring Event Latch Register 1 (RELR1) 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	<b>Other__Beacon Received:</b> Indicates that an Other__Beacon frame was received.
D1	MYBCN	<b>My__Beacon Received:</b> Indicates that a My__Beacon frame was received.
D2-4	RES	<b>Reserved</b>
D5	MYCLM	<b>My__Claim Received:</b> 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	<b>Higher__Claim Received:</b> Indicates that a Higher__Claim frame was received.
D7	LOCLM	<b>Lower__Claim Received:</b> 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	<b>Other__Beacon Mask:</b> This bit is used to mask RELR1.OTRBCN.
D1	MYBCN	<b>My__Beacon Mask:</b> This bit is used to mask RELR1.MYBCN.
D2-4	RES	<b>Reserved</b>
D5	MYCLM	<b>My__Claim Mask:</b> This bit is used to mask RELR1.MYCLM.
D6	HICLM	<b>Higher__Claim Mask:</b> This bit is used to mask RELR1.HICLM.
D7	LOCLM	<b>Lower__Claim Mask:</b> 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	<b>Enter Restricted Mode:</b> Indicates that a Restricted Token was received and that the R_Flag transitioned from 0 to 1.
D1	TVXEXP	<b>TVX Expired:</b> Indicates that a valid frame or token was not received in TVX time.
D2	TRTEXP	<b>TRT Expired:</b> Indicates that a valid token was not received within $2 \cdot TNEG$ .
D3	DUPTKR	<b>Duplicate Token Received:</b> Indicates that a valid token was received while the transmitter was in state T2 or T3.
D4	CBERR	<b>Claim and/or Beacon Error:</b> Indicates that the Claim and/or Beacon Process failed because TRT expired while the Transmitter was in state T4 or T5.
D5	TKCAPT	<b>Token Captured:</b> Indicates that a token has been captured.
D6	TKPASS	<b>Token Passed:</b> Indicates that a valid token has been passed (without capturing it) or has been issued after a service opportunity.
D7	RLVD	<b>Ring Latency Valid:</b> 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. 1: This bit is set to One when the Ring Latency measurement is complete. This bit is written unconditionally and is not protected by the Compare Register.



## 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	<b>Enter Restricted Mode Mask:</b> This bit is used to mask TELR0.ENTRMD.
D1	TVXEXP	<b>TVX Expired Mask:</b> This bit is used to mask TELR0.TVXEXP.
D2	TRTEXP	<b>TRT Expired and Set Late__Flag Mask:</b> This bit is used to mask TELR0.TRTEXP.
D3	DUPTOK	<b>Duplicated Token Received Mask:</b> This bit is used to mask TELR0.DUPTOK.
D4	CBERR	<b>Claim/Beacon Error Mask:</b> This bit is used to mask TELR0.CBERR.
D5	TKCAPT	<b>Token Captured Mask:</b> This bit is used to mask TELR0.TKCAPT.
D6	TKPASS	<b>Token Passed Mask:</b> This bit is used to mask TELR0.TKPASS.
D7	RLVD	<b>Ring Latency Valid Mask:</b> 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	<b>Frame Received:</b> Is set when the Frame Received Counter (FRCT) is incremented, indicating the reception of a frame.
D1	FREI	<b>Frame Error Isolated:</b> Is set when the Error Isolated Counter (EICT) is incremented, indicating an error has been isolated.
D2	FRLST	<b>Frame Lost Isolated:</b> Is set when the Lost Frame Counter (LFCT) is incremented, indicating a format error has been detected in the frame.
D3	FRCOP	<b>Frame Copied:</b> Is set when the Frame Copied Counter (FCCT) is incremented, indicating a frame has been copied.
D4	FRNCOP	<b>Frame Not Copied:</b> Is set when the Frame Not Copied Counter (FNCT) is incremented, indicating a frame could not be copied.
D5	FRTRX	<b>Frame Transmitted:</b> Is set when the Frame Transmitted Counter (FTCT) is incremented, indicating a frame has been transmitted.
D6	TKRCVD	<b>Token Received:</b> Is set when the Token Received Counter (TKCT) is incremented, indicating that a token has been received.
D7	RES	<b>Reserved</b>

## 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	<b>Frame Received Counter Increment Mask:</b> This bit is used to mask CILR.FRRCV.
D1	FREI	<b>Error Isolated Counter Increment Mask:</b> This bit is used to mask CILR.FREI.
D2	FRLST	<b>Lost Frame Counter Increment Mask:</b> This bit is used to mask CILR.FRLST.
D3	FRCOP	<b>Frame Copied Counter Increment Mask:</b> This bit is used to mask CILR.FRCOP.
D4	FRNCOP	<b>Frame Not Copied Counter Increment Mask:</b> This bit is used to mask CILR.FRNCOP.
D5	FRTRX	<b>Frame Transmitted Counter Increment Mask:</b> This bit is used to mask CILR.FRTRX.
D6	TKRCVD	<b>Token Received Counter Increment Mask:</b> This bit is used to mask CILR.TKRCVD.
D7	RES	<b>Reserved</b>

## 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	FRRCV

Bit	Symbol	Description
D0	FRRCV	<b>Frame Received:</b> Is set to One when the Frame Received Counter (FRCT) overflows.
D1	FREI	<b>Frame Error Isolated:</b> Is set to One when the Error Isolated Counter (EICT) overflows.
D2	FRLST	<b>Frame Lost Isolated:</b> Is set to One when the Lost Frame Counter (LFCT) overflows.
D3	FRCOP	<b>Frame Copied:</b> Is set to One when the Frame Copied Counter (FCCT) overflows.
D4	FRNCOP	<b>Frame Not Copied:</b> Is set to One when the Frame Not Copied Counter (FNCT) overflows.
D5	FRTRX	<b>Frame Transmitted:</b> Is set to One when the Frame Transmitted Counter (FTCT) overflows.
D6	TKRCVD	<b>Token Received:</b> Is set to One when the Token Received Counter (TKCT) overflows.
D7	RES	<b>Reserved</b>

## 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	FRRCV

Bit	Symbol	Description
D0	FRRCV	<b>Frame Received Counter Overflow Mask:</b> This bit is used to mask COLR.FRRCV.
D1	FREI	<b>Error Isolated Counter Overflow Mask:</b> This bit is used to mask COLR.FREI.
D2	FRLST	<b>Lost Frame Counter Overflow Mask:</b> This bit is used to mask COLR.FRLST.
D3	FRCOP	<b>Frame Copied Counter Overflow Mask:</b> This bit is used to mask COLR.FRCOP.
D4	FRNCOP	<b>Frame Not Copied Counter Overflow Mask:</b> This bit is used to mask COLR.FRNCOP.
D5	FRTRX	<b>Frame Transmitted Counter Overflow Mask:</b> This bit is used to mask COLR.FRTRX.
D6	TKRCVD	<b>Token Received Counter Overflow Mask:</b> This bit is used to mask COLR.TKRCVD.
D7	RES	<b>Reserved</b>

## 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	<b>MAC Interface Parity Error:</b> 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	<b>Reserved</b>
D2	RSMERR	<b>Receive State Machine Error:</b> Indicates inconsistency in the Receiver state machine. When set, causes bit MCRST of the Function Register to be set.
D3	TSMERR	<b>Transmit State Machine Error:</b> Indicates inconsistency in the Transmitter state machine. When set, causes bit MCRST of the Function Register to be set.
D4–7	RES	<b>Reserved</b>

## 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	<b>PHY Interface Parity Error:</b> 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	<b>Reserved</b>
D5	CPE	<b>Control Bus Parity Error:</b> 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	<b>Control Bus Command Error:</b> 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	<b>Conditional Write Inhibit:</b> 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	<b>PHY Interface Parity Error Mask:</b> This bit is used to mask ESR.PPE.
D1-4	RES	<b>Reserved</b>
D5	CPE	<b>Control Bus Parity Error Mask:</b> This bit is used to mask ESR.CPE.
D6	CCE	<b>Control Bus Error Mask:</b> This bit is used to mask ESR.CCE.
D7	ZERO	<b>Zero:</b> 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	<b>Ring Event Interrupt:</b> Is set if corresponding bits in the Ring Event Latch and Mask Registers are set.
D1	TTE	<b>Token and Timer Event Interrupt:</b> Is set if corresponding bits in the Token and Timer Event Latch and Mask Registers are set.
D2	CIE	<b>Counter Increment Event Interrupt:</b> Is set if corresponding bits in the Counter Increment Latch and Mask Registers are set.
D3	COE	<b>Counter Overflow Event Interrupt:</b> Is set if corresponding bits in the Counter Overflow Latch and Mask Registers are set.
D4–5	RES	<b>Reserved</b>
D6	IERR	<b>Internal Error Interrupt:</b> Is set if any bits in the Internal Event Register are set.
D7	ESE	<b>Exception Status Event Interrupt:</b> 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  $\overline{\text{INT}}$  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	<b>Ring Event Mask:</b> This bit is used to mask ICR.RNG.
D1	TTE	<b>Token and Timer Event Mask:</b> This bit is used to mask ICR.TTE.
D2	CIE	<b>Counter Increment Event Mask:</b> This bit is used to mask ICR.CIE.
D3	COE	<b>Counter Overflow Event Mask:</b> This bit is used to mask ICR.COE.
D4–5	RES	<b>Reserved</b>
D6	IERR	<b>Internal Error Mask:</b> This bit is used to mask ICR.IERR.
D7	ESE	<b>Exception Status Event Mask:</b> 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
<b>MLA0</b>	MLA(47)	MLA(46)	MLA(45)	MLA(44)	MLA(43)	MLA(42)	MLA(41)	MLA(40)
<b>MLA1</b>	MLA(39)	MLA(38)	MLA(37)	MLA(36)	MLA(35)	MLA(34)	MLA(33)	MLA(32)
<b>MLA2</b>	MLA(31)	MLA(30)	MLA(29)	MLA(28)	MLA(27)	MLA(26)	MLA(25)	MLA(24)
<b>MLA3</b>	MLA(23)	MLA(22)	MLA(21)	MLA(20)	MLA(19)	MLA(18)	MLA(17)	MLA(16)
<b>MLA4</b>	MLA(15)	MLA(14)	MLA(13)	MLA(12)	MLA(11)	MLA(10)	MLA(9)	MLA(8)
<b>MLA5</b>	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
<b>MSA0</b>	MSA(15)	MSA(14)	MSA(13)	MSA(12)	MSA(11)	MSA(10)	MSA(9)	MSA(8)
<b>MSA1</b>	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
<b>GLA0</b>	GLA(47)	GLA(46)	GLA(45)	GLA(44)	GLA(43)	GLA(42)	GLA(41)	GLA(40)
<b>GLA1</b>	GLA(39)	GLA(38)	GLA(37)	GLA(36)	GLA(35)	GLA(34)	GLA(33)	GLA(32)
<b>GLA2</b>	GLA(31)	GLA(30)	GLA(29)	GLA(28)	GLA(27)	GLA(26)	GLA(25)	GLA(24)
<b>GLA3</b>	GLA(23)	GLA(22)	GLA(21)	GLA(20)	GLA(19)	GLA(18)	GLA(17)	GLA(16)
<b>GLA4</b>	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.

1. 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).
2. 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
<b>PGM0</b>	PGM(7)	PGM(6)	PGM(5)	PGM(4)	PGM(3)	PGM(2)	PGM(1)	PGM(0)
<b>PGM1</b>	PGM(F)	PGM(E)	PGM(D)	PGM(C)	PGM(B)	PGM(A)	PGM(9)	PGM(8)
<b>PGM2</b>	PGM(17)	PGM(16)	PGM(15)	PGM(14)	PGM(13)	PGM(12)	PGM(11)	PGM(10)
<b>PGM3</b>	PGM(1F)	PGM(1E)	PGM(1D)	PGM(1C)	PGM(1B)	PGM(1A)	PGM(19)	PGM(18)
<b>PGM4</b>	PGM(27)	PGM(26)	PGM(25)	PGM(24)	PGM(23)	PGM(22)	PGM(21)	PGM(20)
<b>PGM5</b>	PGM(2F)	PGM(2E)	PGM(2D)	PGM(2C)	PGM(2B)	PGM(2A)	PGM(29)	PGM(28)
<b>PGM6</b>	PGM(37)	PGM(36)	PGM(35)	PGM(34)	PGM(33)	PGM(32)	PGM(31)	PGM(30)
<b>PGM7</b>	PGM(3F)	PGM(3E)	PGM(3D)	PGM(3C)	PGM(3B)	PGM(3A)	PGM(39)	PGM(38)
<b>PGM8</b>	PGM(47)	PGM(46)	PGM(45)	PGM(44)	PGM(43)	PGM(42)	PGM(41)	PGM(40)
<b>PGM9</b>	PGM(4F)	PGM(4E)	PGM(4D)	PGM(4C)	PGM(4B)	PGM(4A)	PGM(49)	PGM(48)
<b>PGMA</b>	PGM(57)	PGM(56)	PGM(55)	PGM(54)	PGM(53)	PGM(52)	PGM(51)	PGM(50)
<b>PGMB</b>	PGM(5F)	PGM(5E)	PGM(5D)	PGM(5C)	PGM(5B)	PGM(5A)	PGM(59)	PGM(58)
<b>PGMC</b>	PGM(67)	PGM(66)	PGM(65)	PGM(64)	PGM(63)	PGM(62)	PGM(61)	PGM(60)
<b>PGMD</b>	PGM(6F)	PGM(6E)	PGM(6D)	PGM(6C)	PGM(6B)	PGM(6A)	PGM(69)	PGM(68)
<b>PGME</b>	PGM(77)	PGM(76)	PGM(75)	PGM(74)	PGM(73)	PGM(72)	PGM(71)	PGM(70)
<b>PGMF</b>	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
<b>PGM10</b>	PGM(87)	PGM(86)	PGM(85)	PGM(84)	PGM(83)	PGM(82)	PGM(81)	PGM(80)
<b>PGM11</b>	PGM(8F)	PGM(8E)	PGM(8D)	PGM(8C)	PGM(8B)	PGM(8A)	PGM(89)	PGM(88)
<b>PGM12</b>	PGM(97)	PGM(96)	PGM(95)	PGM(94)	PGM(93)	PGM(92)	PGM(91)	PGM(90)
<b>PGM13</b>	PGM(9F)	PGM(9E)	PGM(9D)	PGM(9C)	PGM(9B)	PGM(9A)	PGM(99)	PGM(98)
<b>PGM14</b>	PGM(A7)	PGM(A6)	PGM(A5)	PGM(A4)	PGM(A3)	PGM(A2)	PGM(A1)	PGM(A0)
<b>PGM15</b>	PGM(AF)	PGM(AE)	PGM(AD)	PGM(AC)	PGM(AB)	PGM(AA)	PGM(A9)	PGM(A8)
<b>PGM16</b>	PGM(B7)	PGM(B6)	PGM(B5)	PGM(B4)	PGM(B3)	PGM(B2)	PGM(B1)	PGM(B0)
<b>PGM17</b>	PGM(BF)	PGM(BE)	PGM(BD)	PGM(BC)	PGM(BB)	PGM(BA)	PGM(B9)	PGM(B8)
<b>PGM18</b>	PGM(C7)	PGM(C6)	PGM(C5)	PGM(C4)	PGM(C3)	PGM(C2)	PGM(C1)	PGM(C0)
<b>PGM19</b>	PGM(CF)	PGM(CE)	PGM(CD)	PGM(CC)	PGM(CB)	PGM(CA)	PGM(C9)	PGM(C8)
<b>PGM1A</b>	PGM(D7)	PGM(D6)	PGM(D5)	PGM(D4)	PGM(D3)	PGM(D2)	PGM(D1)	PGM(D0)
<b>PGM1B</b>	PGM(DF)	PGM(DE)	PGM(DD)	PGM(DC)	PGM(DB)	PGM(DA)	PGM(D9)	PGM(D8)
<b>PGM1C</b>	PGM(E7)	PGM(E6)	PGM(E5)	PGM(E4)	PGM(E3)	PGM(E2)	PGM(E1)	PGM(E0)
<b>PGM1D</b>	PGM(EF)	PGM(EE)	PGM(ED)	PGM(EC)	PGM(EB)	PGM(EA)	PGM(E9)	PGM(E8)
<b>PGM1E</b>	PGM(F7)	PGM(F6)	PGM(F5)	PGM(F4)	PGM(F3)	PGM(F2)	PGM(F1)	PGM(F0)
<b>PGM1F</b>	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  $\mu$ 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
<b>TREQ0</b>	TREQ(31)	TREQ(30)	TREQ(29)	TREQ(28)	TREQ(27)	TREQ(26)	TREQ(25)	TREQ(24)
<b>TREQ1</b>	TREQ(23)	TREQ(22)	TREQ(21)	TREQ(20)	TREQ(19)	TREQ(18)	TREQ(17)	TREQ(16)
<b>TREQ2</b>	TREQ(15)	TREQ(14)	TREQ(13)	TREQ(12)	TREQ(11)	TREQ(10)	TREQ(9)	TREQ(8)
<b>TREQ3</b>	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
<b>TBT0</b>	TBT(31)	TBT(30)	TBT(29)	TBT(28)	TBT(27)	TBT(26)	TBT(25)	TBT(24)
<b>TBT1</b>	TBT(23)	TBT(22)	TBT(21)	TBT(20)	TBT(19)	TBT(18)	TBT(17)	TBT(16)
<b>TBT2</b>	TBT(15)	TBT(14)	TBT(13)	TBT(12)	TBT(11)	TBT(10)	TBT(9)	TBT(8)
<b>TBT3</b>	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 $\mu$ s
1	20.48 $\mu$ s
2	40.96 $\mu$ s
3	81.92 $\mu$ s
4	163.84 $\mu$ s
5	327.68 $\mu$ s
6	655.36 $\mu$ 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  $40.96 \times 2^{\text{TMAX}} \mu\text{s}$ . On a Master Reset (Function.MARST set to One), TMAX is set to the value of Ch which corresponds to 167.77 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 $\mu\text{s}$
1	81.92 $\mu\text{s}$
2	163.84 $\mu\text{s}$
3	327.68 $\mu\text{s}$
4	655.36 $\mu\text{s}$
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  $40.96 \times 2^{\text{TVX}} \mu\text{s}$ . 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 $\mu\text{s}$
1	81.92 $\mu\text{s}$
2	163.84 $\mu\text{s}$
3	327.68 $\mu\text{s}$
4	655.36 $\mu\text{s}$
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
<b>TNEG0</b>	TNEG(31)	TNEG(30)	TNEG(29)	TNEG(28)	TNEG(27)	TNEG(26)	TNEG(25)	TNEG(24)
<b>TNEG1</b>	TNEG(23)	TNEG(22)	TNEG(21)	TNEG(20)	TNEG(19)	TNEG(18)	TNEG(17)	TNEG(16)
<b>TNEG2</b>	TNEG(15)	TNEG(14)	TNEG(13)	TNEG(12)	TNEG(11)	TNEG(10)	TNEG(9)	TNEG(8)
<b>TNEG3</b>	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).

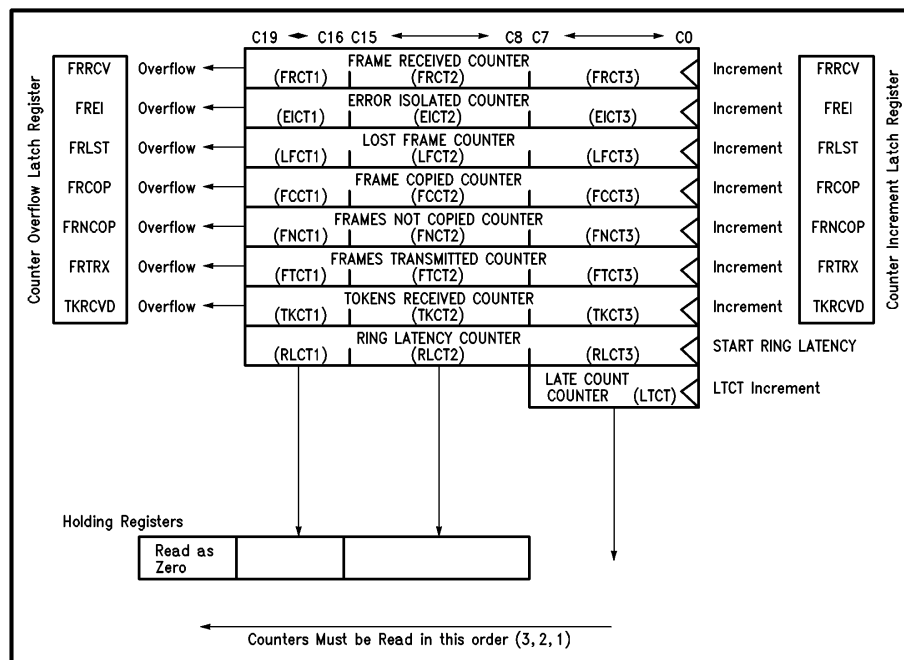


FIGURE 6-2. Event Counters

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.FRRVCV) and when the 20-bit counter overflows and wraps around (COLR.FRRVCV).

### 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
<b>EICT0</b>	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
<b>EICT1</b>	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
<b>EICT2</b>	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
<b>EICT3</b>	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
<b>LFCT0</b>	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
<b>LFCT1</b>	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
<b>LFCT2</b>	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
<b>LFCT3</b>	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
<b>FCCT0</b>	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
<b>FCCT1</b>	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
<b>FCCT2</b>	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
<b>FCCT3</b>	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.FRNCOPI) and when the 20-bit counter overflows and wraps around (COLR.FRNCOPI).

### ACCESS RULES

Address	Read	Write
B0-B3h	Always	Stop Mode

	D7	D6	D5	D4	D3	D2	D1	D0
<b>FNCT0</b>	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
<b>FNCT1</b>	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
<b>FNCT2</b>	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
<b>FNCT3</b>	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
<b>FTCT0</b>	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
<b>FTCT1</b>	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
<b>FTCT2</b>	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
<b>FTCT3</b>	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
<b>TKCT0</b>	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
<b>TKCT1</b>	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
<b>TKCT2</b>	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
<b>TKCT3</b>	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 (TEL.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 (TEL.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 16 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 (TEL.RLVLD) is set and may cause an interrupt. When set, RELR.RLVLD indicates that RLCT will be valid within 1.28  $\mu$ s. The Ring Latency Counter can measure ring latencies up to 1.3421772 seconds with accuracy of 1.28  $\mu$ 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
<b>RLCT0</b>	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
<b>RLCT1</b>	Zero	Zero	Zero	Zero	CT(19)	CT(18)	CT(17)	CT(16)
<b>RLCT2</b>	CT(15)	CT(14)	CT(13)	CT(12)	CT(11)	CT(10)	CT(9)	CT(8)
<b>RLCT3</b>	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 (TELR.CBERR, TELR.TRTEXP) can be used to determine that Late Count Counter has incremented. No overflow event is provided.

### ACCESS RULES

Address	Read	Write							
9Fh	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	<b>Control Bus Parity:</b> Odd parity on CBD7–0.
CBD7–0	9–6, 3–1, 132	I/O	<b>Control Bus Data</b>
CBA7–0	131–129, 127–123	I	<b>Control Bus Address:</b> Address of a particular register.
$\overline{CE}$	120	I	<b>Control Bus Enable:</b> Handshake signal used to begin a Control Interface access. Active low signal.
R/ $\overline{W}$	119	I	<b>Read/ ~ Write:</b> Determines current direction of a Control Interface access.
$\overline{ACK}$	122	OD	<b>~ Acknowledge:</b> Acknowledges that the Control Interface access has been performed. Active low, open drain signal.
$\overline{INT}$	121	OD	<b>~ Interrupt:</b> 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	<b>PHY Request Parity:</b> Odd parity for PRC and PRD7–0.
PRC	112	O	<b>PHY Request Control:</b> 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	<b>PHY Request Data:</b> Contains a Data or Control symbol pair.
PIP	115	I	<b>PHY Indicate Parity:</b> Odd parity for PIC and PID7–0.
PIC	113	I	<b>PHY Indicate Control:</b> 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	<b>PHY Indicate Data:</b> 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
nl	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 EDRCVD

VFCS Valid FCS Criteria: Received FCS matches with standard CRC polynomial.

Valid with EDRCVD

**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	<b>MAC Indicate Parity:</b> Odd parity on MID7–0. Only valid with Data and Status Indicators.
MID7–0	74–76, 79–83	O	<p><b>MAC Indicate Data:</b></p> <p><b>Data:</b> 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"> <li>Ending Delimiter Received (EDRCVD),</li> <li>Token Received (TKRCVD),</li> <li>Format Error (FOERROR),</li> <li>Frame Strip (FRSTRP),</li> <li>or MAC Reset (MACRST).</li> </ul> <p><b>Status:</b> 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	<b>Receive Start:</b> 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	<b>Frame Control Received:</b> 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	<b>Destination Address Received:</b> 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	<b>Source Address Received:</b> 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	<b>Information Field Received:</b> 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	<b>My Destination Address Recognized:</b> 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	<b>Individual/Group Address Flag:</b> Indicates the address type. Valid on the first byte of the Destination Address. 0: Individual Address 1: Group Address
FCSL	53	O	<b>Short/Long Address Flag:</b> 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	<b>My Source Address Recognized:</b> 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	<b>Same Source Address:</b> 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	<b>Same MAC Information:</b> 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	<b>Token Received:</b> Indicates that the Ending Delimiter for a valid token is being received.
EDRCVD	66	O	<b>Ending Delimiter Received:</b> 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	<b>Frame Stripped:</b> 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	<b>Format Error:</b> 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	<b>MAC Reset:</b> 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	<b>Valid Data Length:</b> 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	<b>Valid Frame Check Sequence:</b> 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	<b>External A__Flag:</b> 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	<b>External M__Flag:</b> 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	<b>Valid Copy:</b> 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"> <li>1. the internal A__flag is set or</li> <li>2. Option.EMIND = 1 and the external A__flag (EA) is set;</li> </ol> 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.

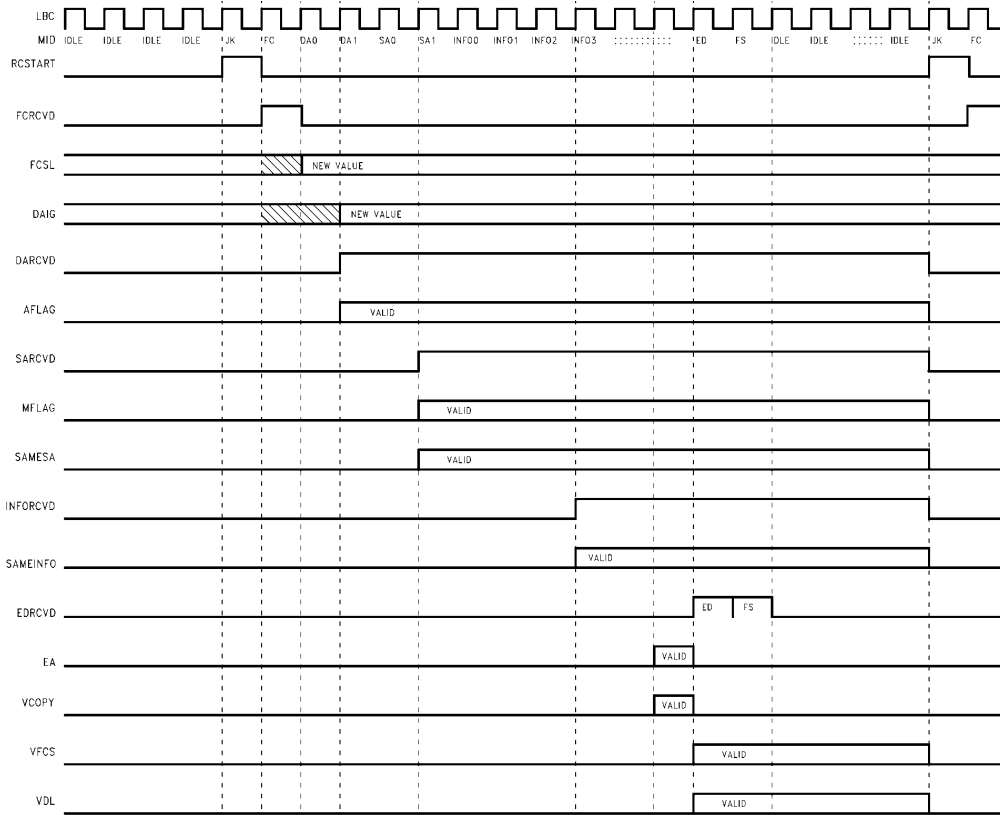
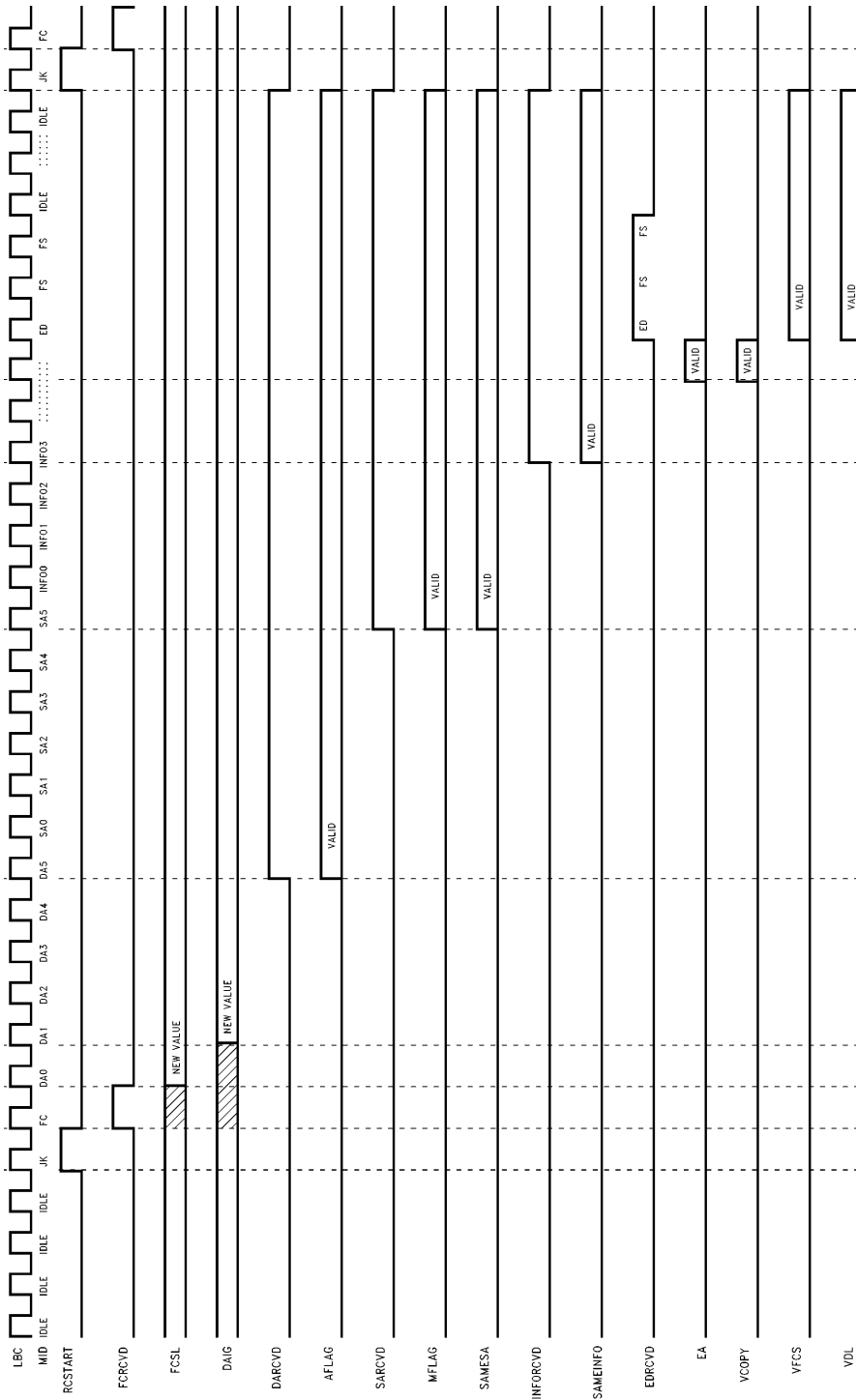


FIGURE 7-1. Frame Reception with Short Address

TL/F/10387-5

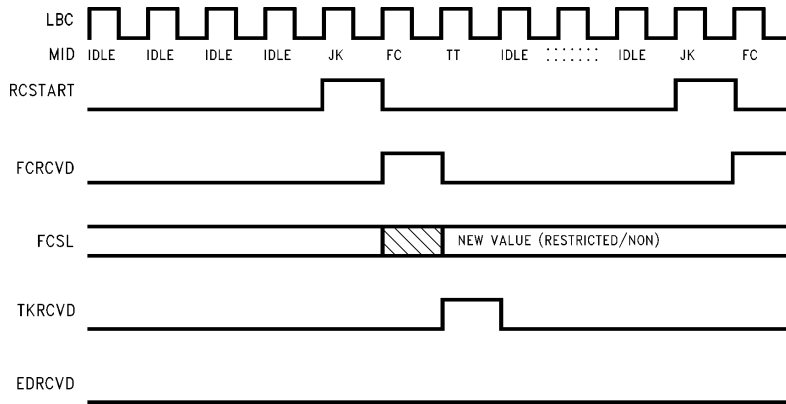
## 7.0 Signal Descriptions (Continued)



TL/F/10387-6

**FIGURE 7-2. Frame Reception with Long Address**

## 7.0 Signal Descriptions (Continued)

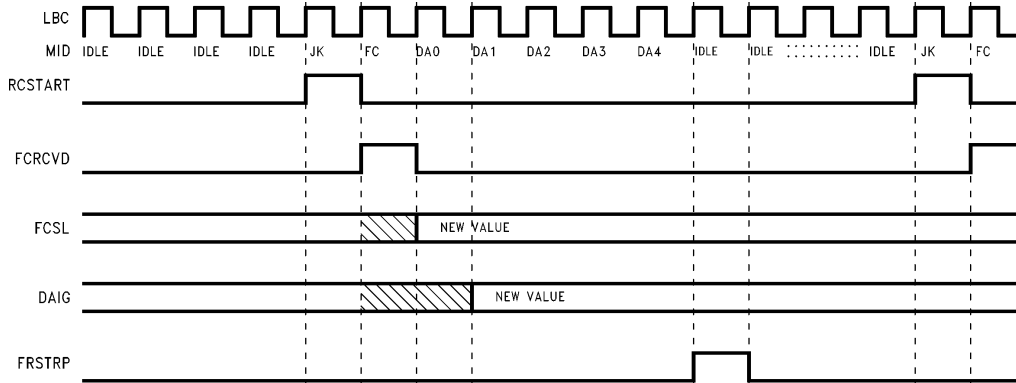


TL/F/10387-8

FIGURE 7-3. Token Reception

### Remnant Reception

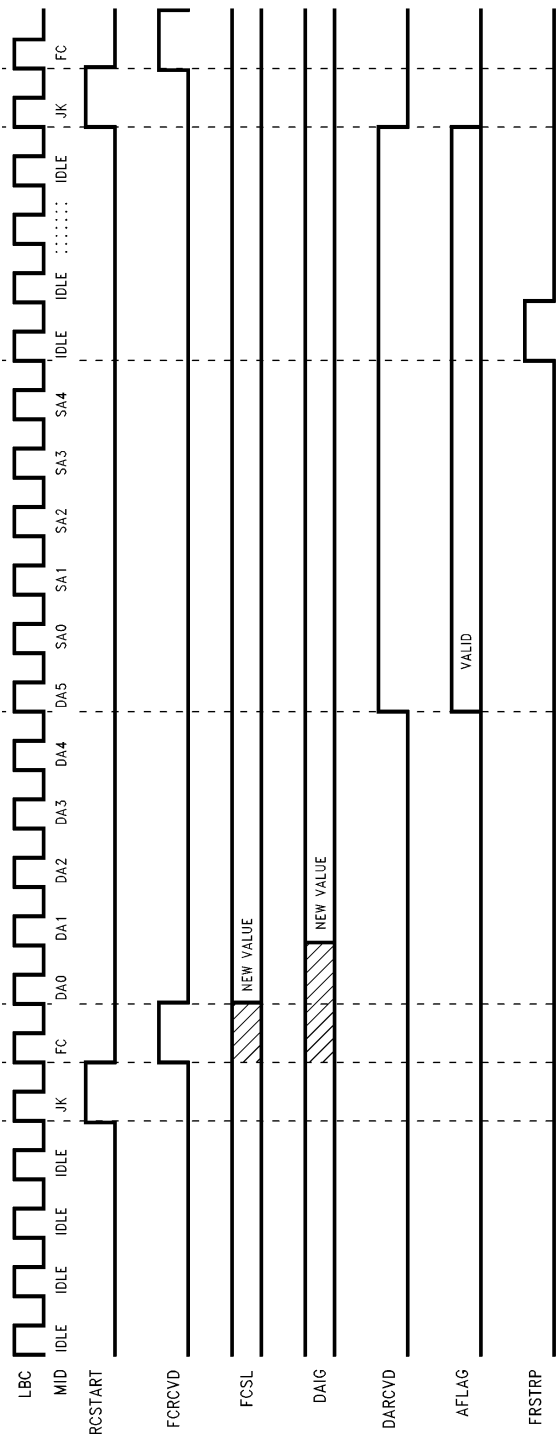
In these examples, the remnants of frames that were stripped by an upstream station are received. Examples are shown for frames where the strip point occurred at an upstream station before, during and after the SA field. (See Figures 7-4 through 7-6.)



TL/F/10387-9

FIGURE 7-4. Frame Stripped before SA Field

## 7.0 Signal Descriptions (Continued)



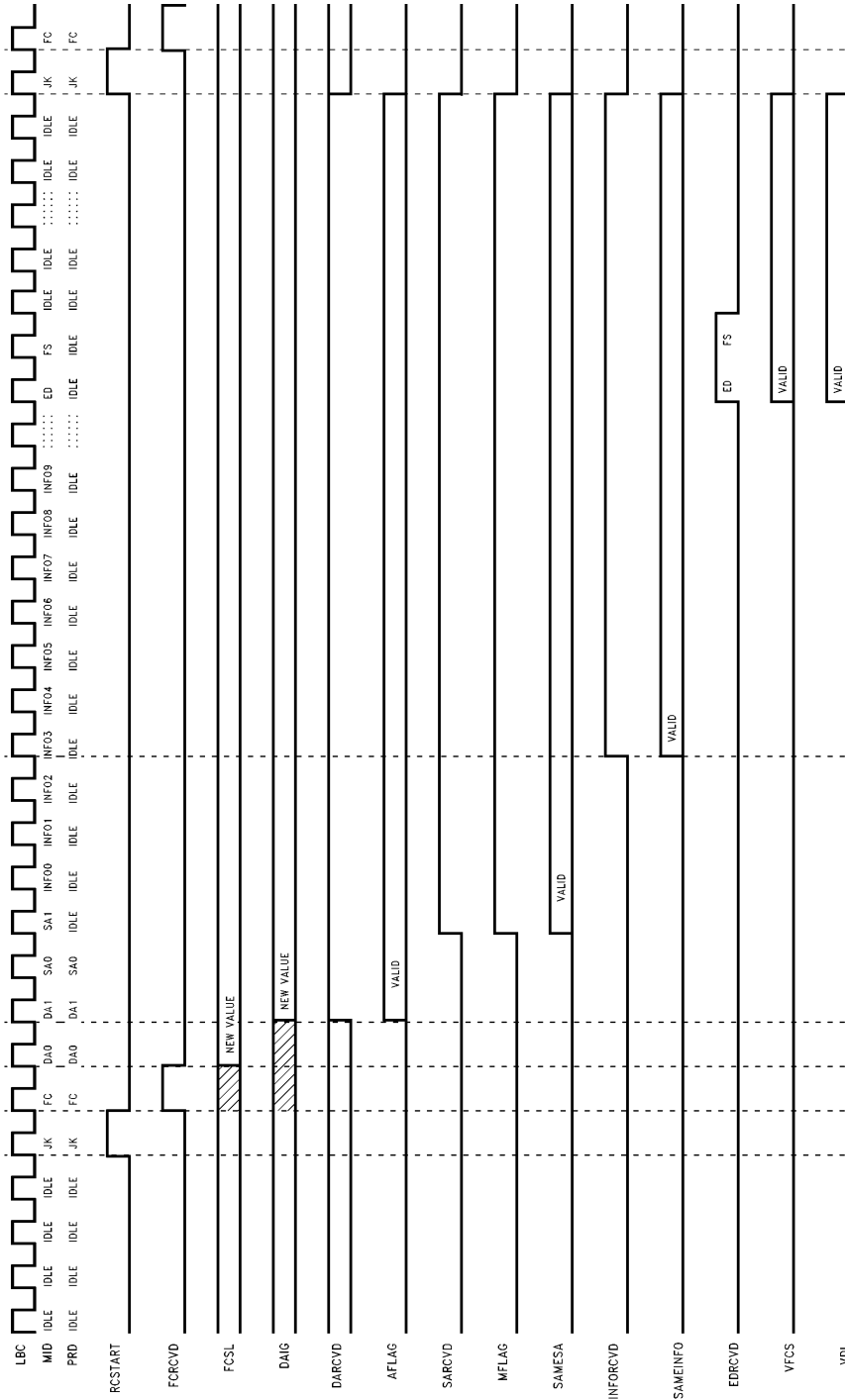
TL/F/10387-12

FIGURE 7-5. Frame Stripped during SA Field



## 7.0 Signal Descriptions (Continued)

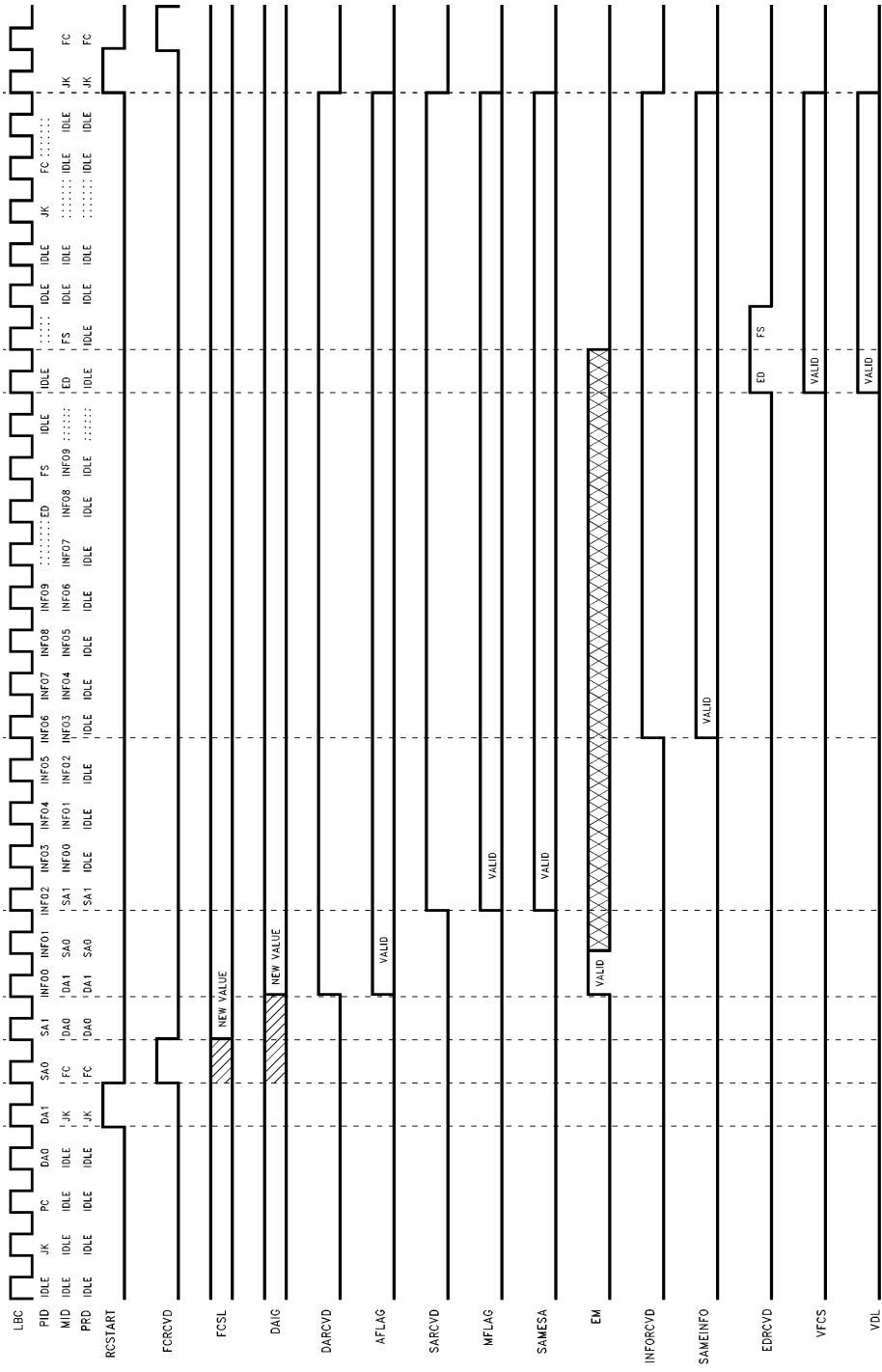
### Frame Stripping



TL/F/10887-14

FIGURE 7-7. Stripping Based on M Flag

## 7.0 Signal Descriptions (Continued)



TL/F/10387-15

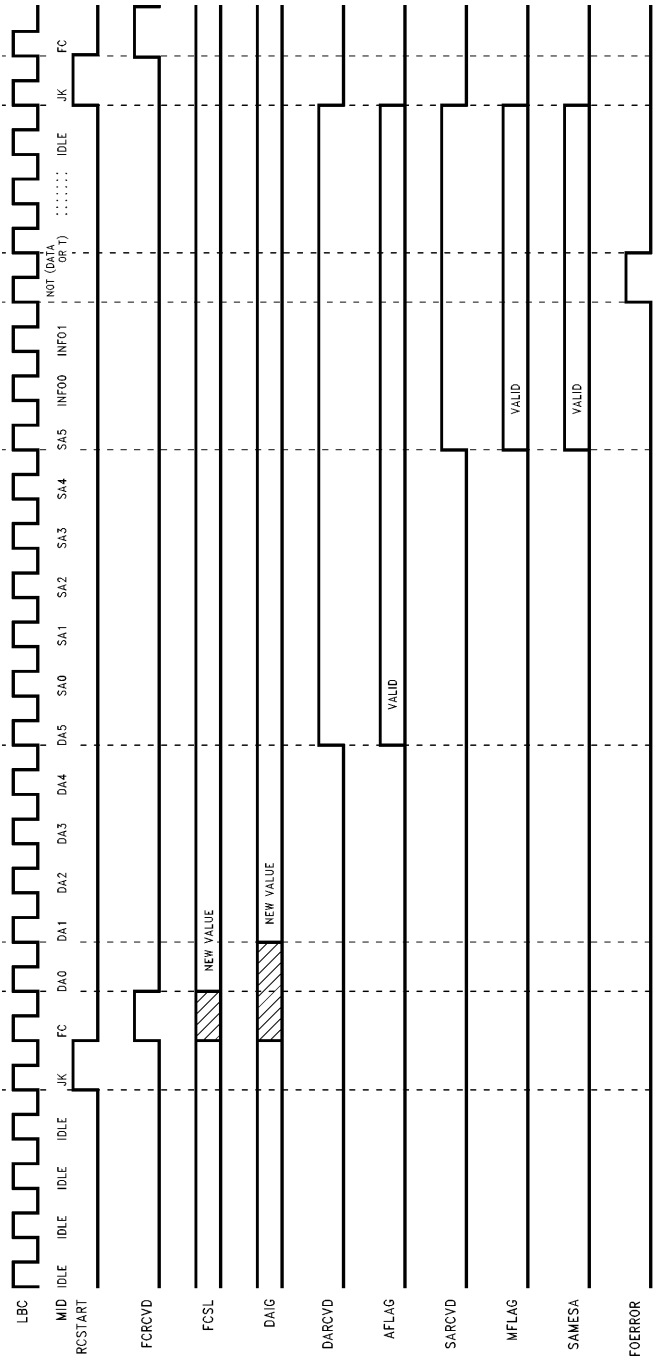
FIGURE 7-8. Stripping Based on External M Flag

Note that stripping begins 3 byte times after EM is asserted.



## 7.0 Signal Descriptions (Continued)

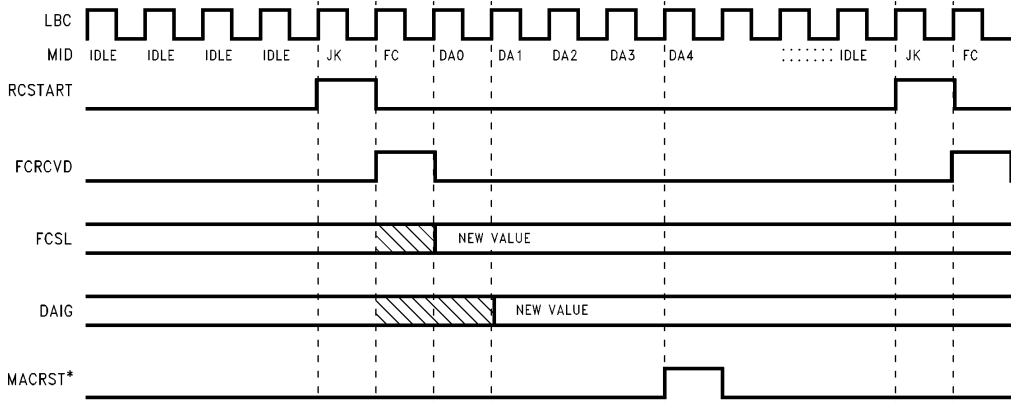
### Abnormal Termination Conditions



TL/F/10387-16

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	<b>MAC Request Parity:</b> Odd parity on MRD7-0.
MRD7-0	42-49	I	<b>MAC Request Data:</b> 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	<b>Transmit Pass:</b> 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	<b>Transmit Ready:</b> 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: 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	<b>Request Ready:</b> 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 $\mu$ s or 4.80 $\mu$ s) for RQSEND to be asserted before releasing the token.
RQSEND	24	I	<b>Request Send:</b> 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 $\mu$ s or 4.60 $\mu$ 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	<b>Transmit Acknowledge:</b> 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	<b>Request End of Frame:</b> 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	<b>Request Abort:</b> 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 will prevent TXACK assertion. The BMAC will ignore RQABT if asserted with RQEOF.
RQFINAL	26	I	<b>Request Final:</b> 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	<b>Request Class:</b> 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 will be issued (even if RQRCLS subsequently becomes non-zero). See Table 5-3.
RQCLM	15	I	<b>Request Claim:</b> Indicates that this request is to be serviced in the Claim state. Ignored for non-immediate requests.
RQBCN	16	I	<b>Request Beacon:</b> Indicates that this request is to be serviced in the Beacon state. Ignored for non-immediate requests.

### 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	<b>Void Strip:</b> 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.
SAT	12	I	<b>Source Address Transparency:</b> 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.
SAIGT	11	I	<b>Source Address I/G Transparency:</b> With this option, the MSB of the SA is sourced from the data stream, as opposed to being forced to zero.
FCST	14	I	<b>Frame Check Sequence Transparency:</b> When selected, the Ring Engine generated FCS is not appended to the end of the Information field.

## 7.0 Signal Descriptions (Continued)

### Transmission Status

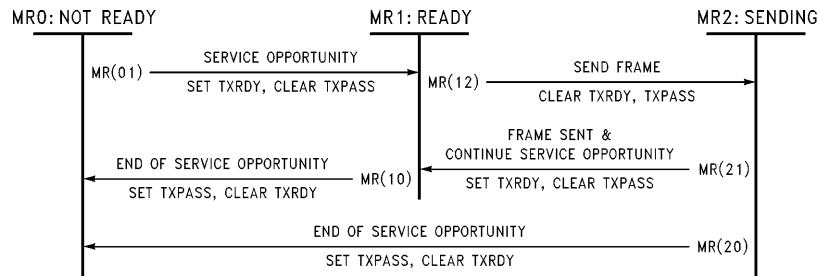
Symbol	Pin #	I/O	Description
TXED	31	O	<b>Transmitted Ending Delimiter:</b> Indicates that the Transmitter completed transmission of the current or previous PDU. TXED is asserted when the current PHY Request byte is a transmitted (not repeated) Ending Delimiter. It remains asserted until the beginning 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	<b>Transmission Aborted:</b> 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 end of the transmitted Void frame. 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	<b>Ring Operational:</b> Indicates the current value of the Ring_Operational flag.
TXCLASS	35	O	<b>Token Class:</b> 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	<b>Token Holding Timer Disabled:</b> 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  $\mu$ s), a Void frame has been sent (0.96  $\mu$ s or 1.60  $\mu$ s), and 7 more bytes of preamble have been sent (0.56  $\mu$ 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.

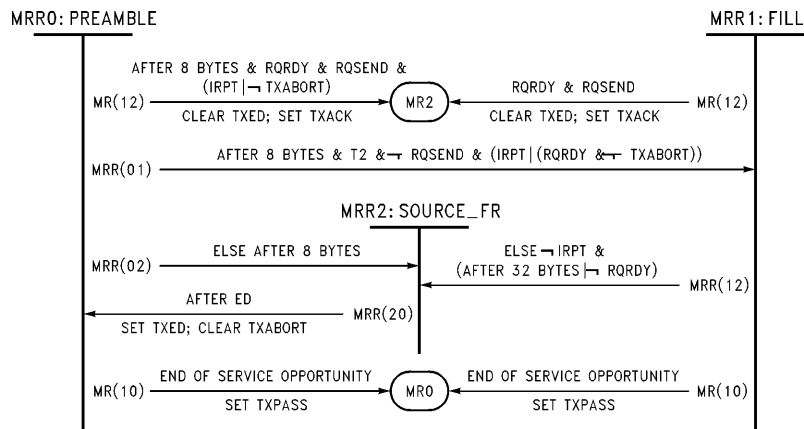


FIGURE 7-12. MR1 Substate Diagram

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  $\mu$ s–5.28  $\mu$ 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  $\mu$ 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 MRO). 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

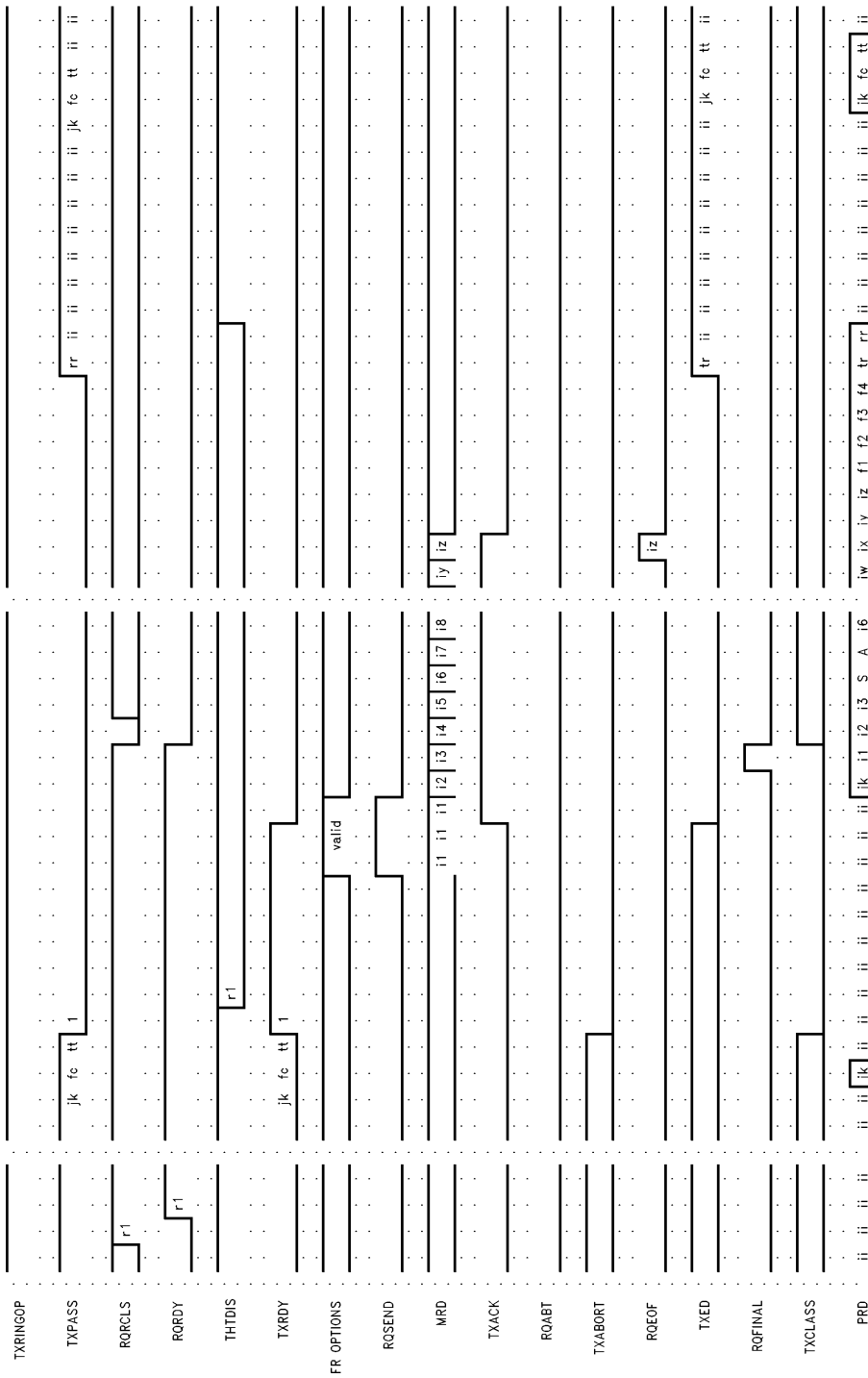
Prestaging refers to the staging of data before the Service Opportunity begins. Prestaging occurs in interfaces where data is loaded into a FIFO or dedicated memory used as a FIFO before the token arrives.

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)



TL/F/10387-20

FIGURE 7-13. Asynchronous Request with Prestaging

## 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  $\uparrow\text{TXRDY} \rightarrow \uparrow\text{RQSEND} \rightarrow \downarrow\text{TXRDY} \rightarrow \downarrow\text{RQSEND}$  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  $\uparrow\text{TXRDY} \rightarrow \uparrow\text{RQSEND} \rightarrow \downarrow\text{TXRDY} \rightarrow \downarrow\text{RQSEND}$  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.

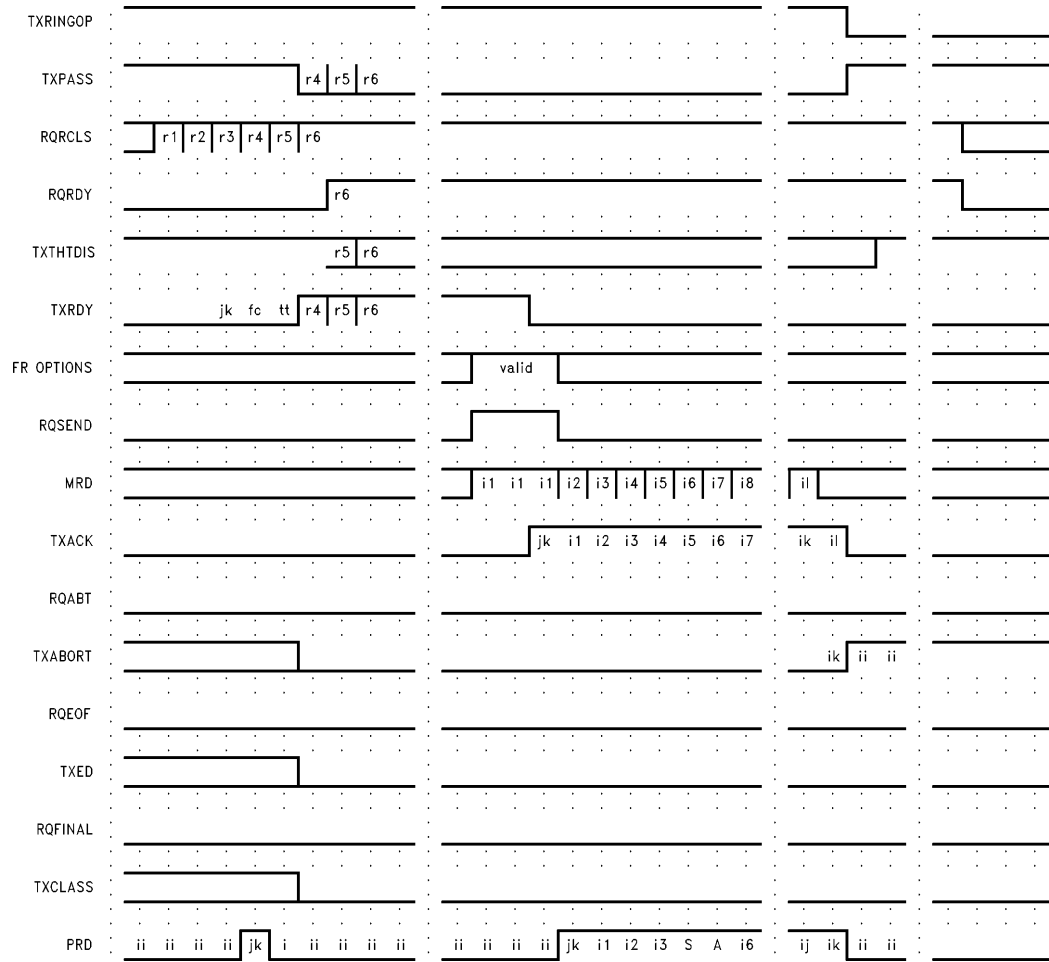
After the frame is transmitted, TXRDY is asserted again since the transmitter is still in the Claim state.

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.





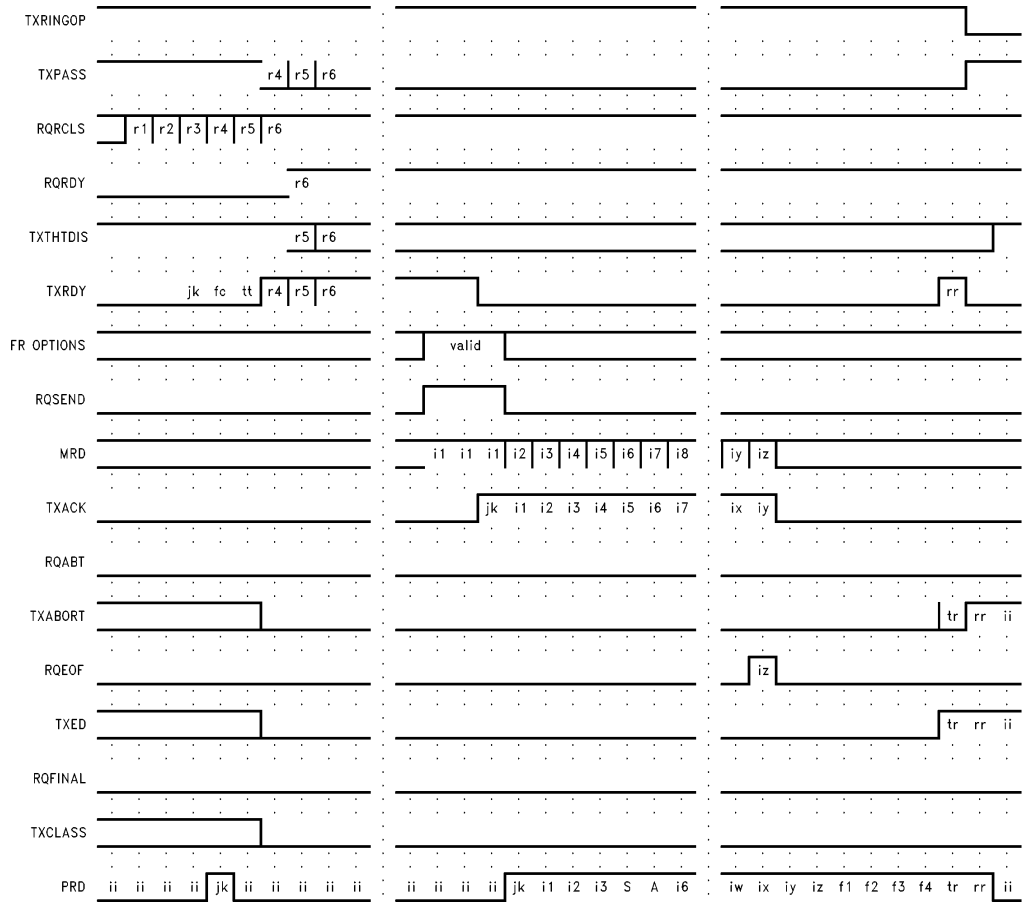
## 7.0 Signal Descriptions (Continued)



TL/F/10387-23

FIGURE 7-16. MAC Reset

## 7.0 Signal Descriptions (Continued)

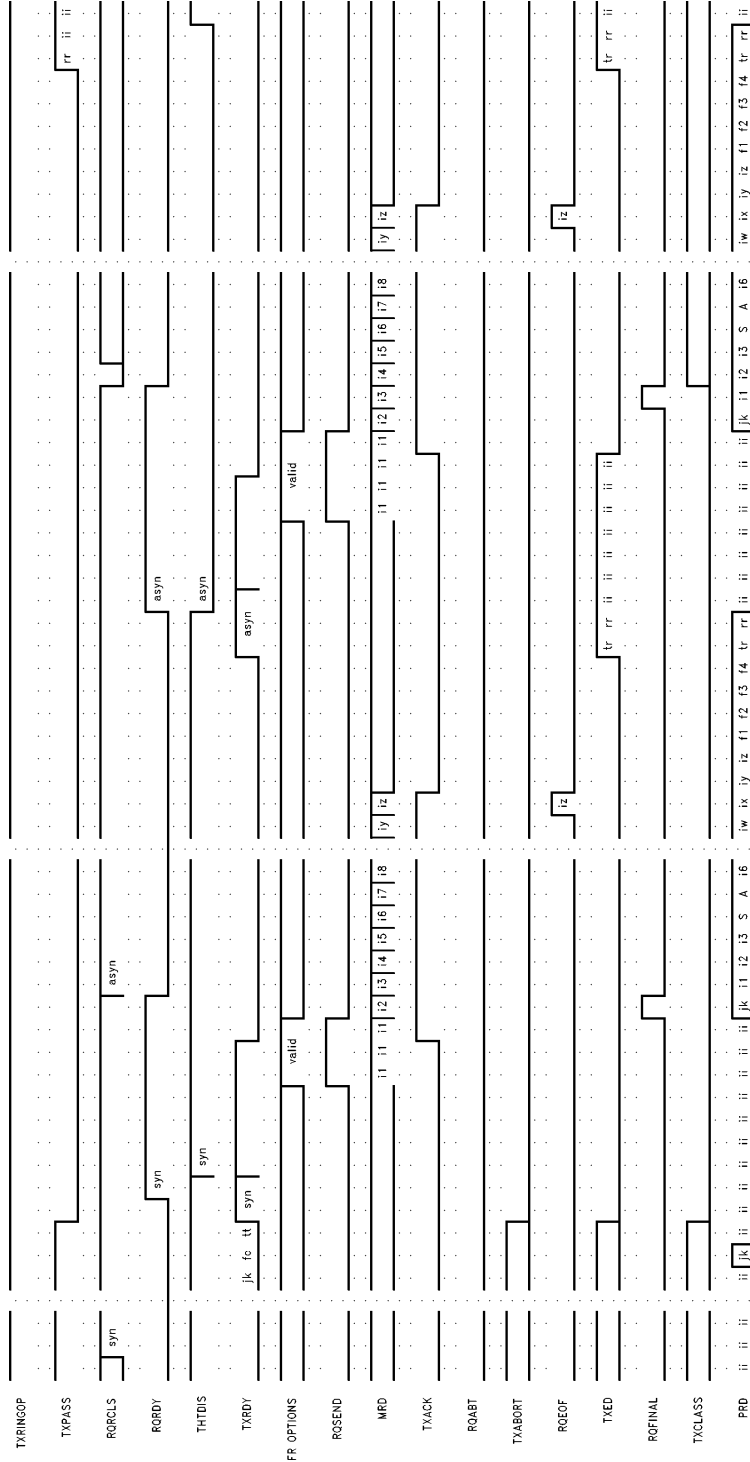


TL/F/10387-24

FIGURE 7-17. MAC Reset at End of Frame



## 7.0 Signal Descriptions (Continued)

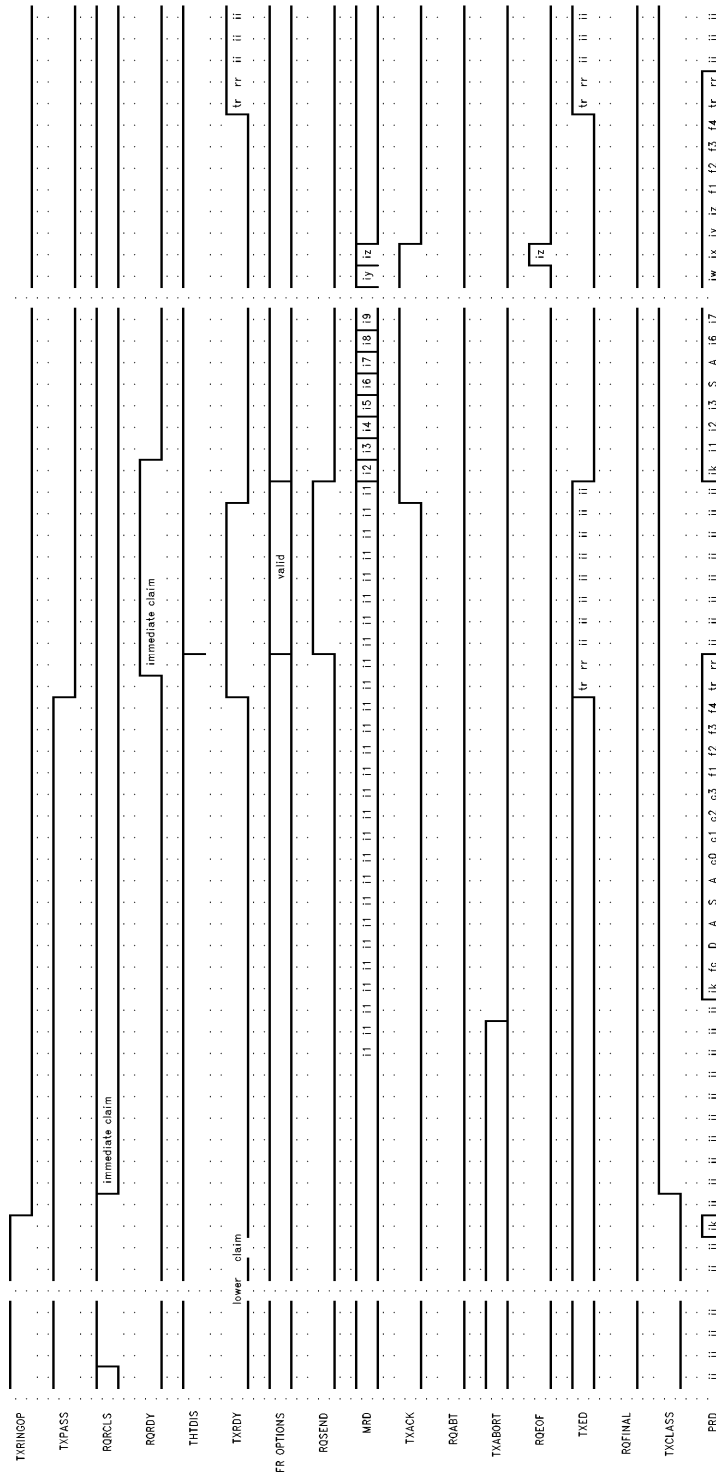


TLF/10387-25

FIGURE 7-18. Synchronous Request followed by Asynchronous



## 7.0 Signal Descriptions (Continued)



TL/F/10387-27

FIGURE 7-20. Immediate Claim

## 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	<b>Local Symbol Clock:</b> 25 MHz clock with a 40/60 duty-cycle. Typically generated by the CDD.
LBC	86	I	<b>Local Byte Clock:</b> 12.5 MHz clock 50/50 duty-cycle in phase with LSC. Typically generated by the CDD.
$\overline{\text{RST}}$	118	I	<b>Master Reset:</b> 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 <sub>CC</sub> [11]	4, 17, 34 58, 78 94, 100 106, 117	I	<b>Positive Power Supply:</b> 5V, $\pm 5\%$ relative to GND.
GND[11]	5, 18, 33 57, 77, 88–91, 101, 116, 128	I	<b>Power Supply Return</b>

### 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 <sub>CC</sub>	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 <sub>CC</sub>	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 <sub>CC</sub>	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 <sub>CC</sub>	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 <sub>CC</sub>	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 <sub>CC</sub>	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 <sub>CC</sub>	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 <sub>CC</sub>	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)

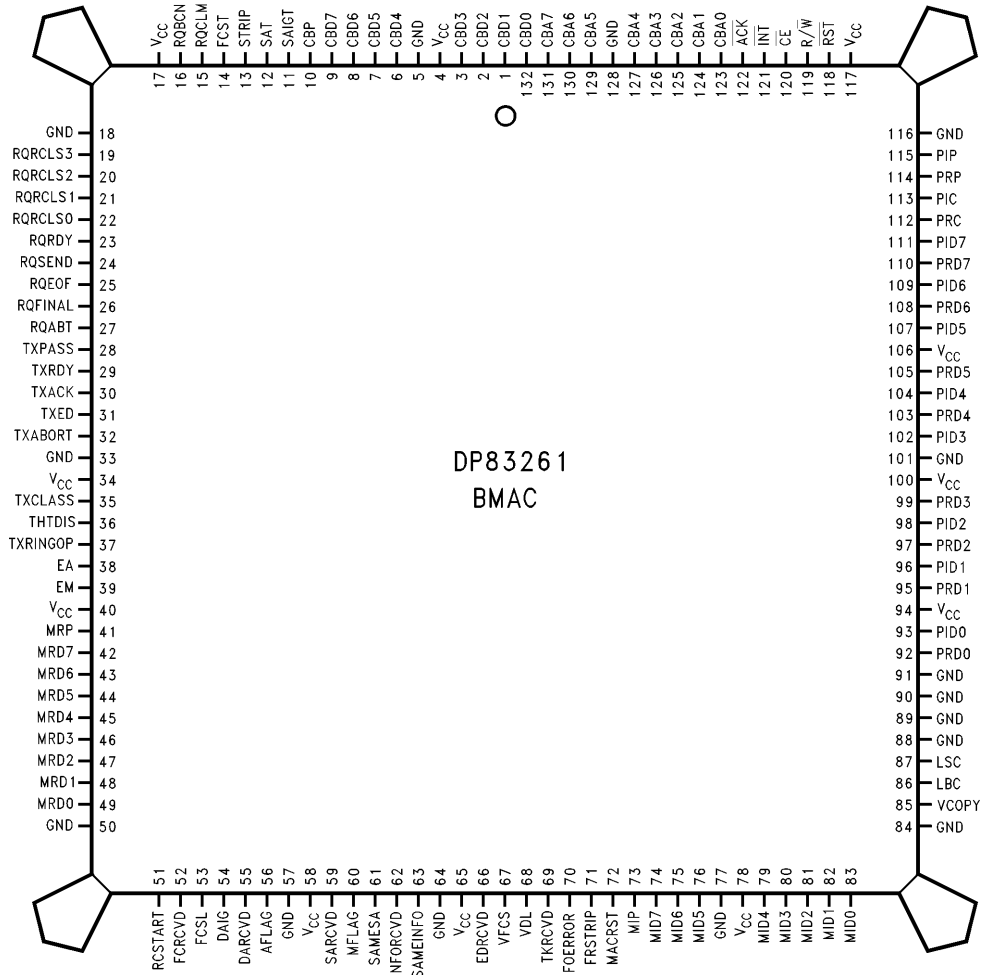
### 7.6 PINOUT SUMMARY (Continued)

TABLE 7-6. Pinout Summary (Continued)

Pin #	Signal Name	Symbol	I/O
100	Positive Power Supply	V <sub>CC</sub>	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 <sub>CC</sub>	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 <sub>CC</sub>	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



TL/F/10387-11

FIGURE 7-21. DP83261 132-Pin PQFP Pinout

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 <sub>CC</sub>	Supply Voltage		-0.5	7.0	V
V <sub>IN</sub>	DC Input Voltage		-0.5	V <sub>CC</sub> + 0.5	V
V <sub>OUT</sub>	DC Output Voltage		-0.5	V <sub>CC</sub> + 0.5	V
T <sub>STG</sub>	Storage Temperature Range		-65	+150	°C
T <sub>L</sub>	Lead Temperature	Soldering, 10 sec. (IR or Vapor) (Phase Reflow)		230	°C
	ESD Rating	R <sub>ZAP</sub> = 1.5k, C <sub>ZAP</sub> = 120 pF		800	V

### 8.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Conditions	Min	Max	Units
V <sub>CC</sub>	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 <sub>OH1</sub>	Minimum High Level Output Voltage	C <sub>L</sub> = 50 pF	V <sub>CC</sub> - 0.5		V
V <sub>OH2</sub>	Minimum High Level Output Voltage	I <sub>OH</sub> = -2 mA	2.4		V
V <sub>OL1</sub>	Maximum Low Level Output Voltage	C <sub>L</sub> = 50 pF		0.4	V
V <sub>OL2</sub>	Maximum Low Level Output Voltage	I <sub>OL</sub> = 4 mA		0.4	V
V <sub>OL3</sub>	Maximum Low Level Output Voltage $\overline{\text{INT}}$ and $\overline{\text{ACK}}$ (Open Drain)	I <sub>OL</sub> = 8 mA		0.4	V
V <sub>IH</sub>	Minimum High Level Input Voltage		2.0		V
V <sub>IL</sub>	Maximum Low Level Input Voltage			0.8	V
I <sub>IH</sub>	Input High Current			+10	μA
I <sub>IL</sub>	Input Low Current			-10	μA
I <sub>oz1</sub>	TRI-STATE Leakage for CBD(7-0) and CBP			±10	μA
I <sub>oz2</sub>	TRI-STATE Leakage for $\overline{\text{INT}}$ and $\overline{\text{ACK}}$ (Open Drain)			±10	μA
I <sub>oz3</sub>	Dynamic Supply Current	C <sub>L</sub> = 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

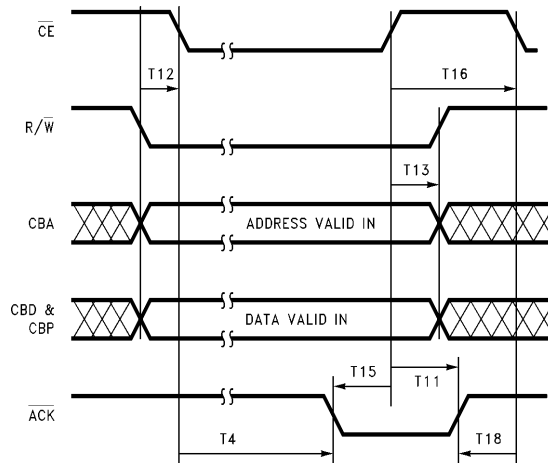
Symbol	Parameter	Min	Max	Units
T1	$\overline{CE}$ Setup to LBC	15		ns
T2	LBC Period	80		ns
T3	LBC to $\overline{ACK}$ Low		45	ns
T4	$\overline{CE}$ Low to $\overline{ACK}$ Low	290	540	ns
T5	LBC Low to CBD(7-0) and CBP Valid		60	ns
T6	LBC to CBD(7-0) and CBP Active		60	ns
T7	$\overline{CE}$ Low to CBD(7-0) 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)	$T1 + (5 * T2) + T9 + T5$

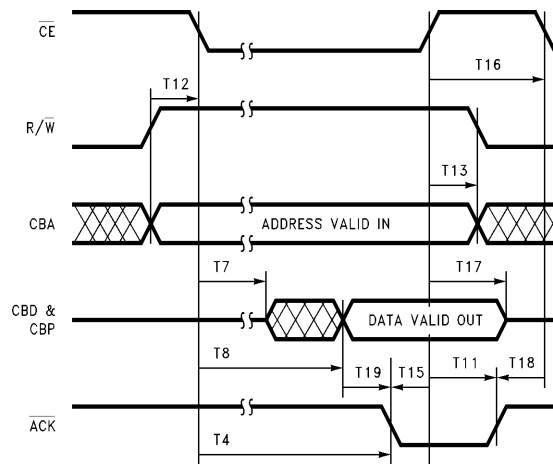
**Note:** Min/Max numbers are based on T2 = 80 ns and T9 = T10 = 40 ns.

## 8.0 Electrical Characteristics (Continued)



TL/F/10387-28

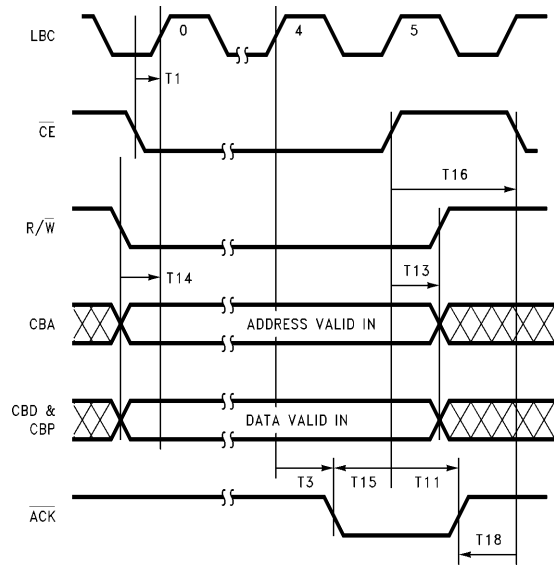
FIGURE 8-1. Control Bus Interface Write Cycle



TL/F/10387-29

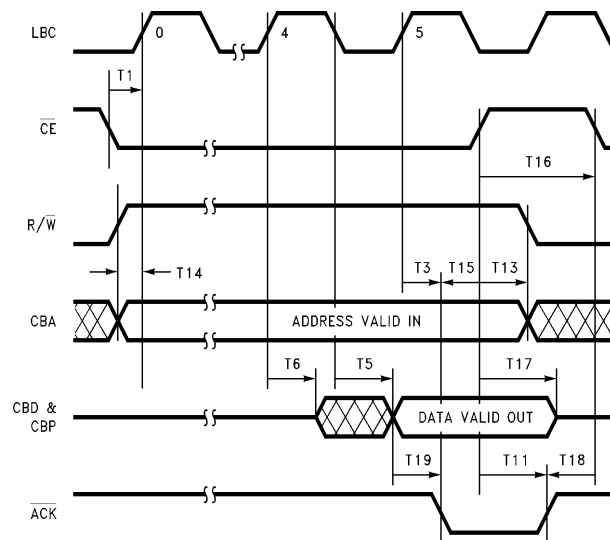
FIGURE 8-2. Control Bus Interface Read Cycle

## 8.0 Electrical Characteristics (Continued)



TL/F/10387-30

FIGURE 8-3. Control Bus Interface Synchronous Write Cycle



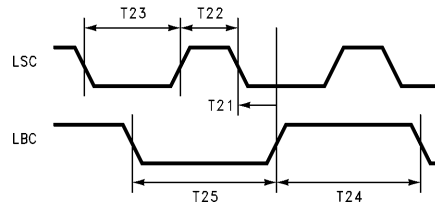
TL/F/10387-31

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

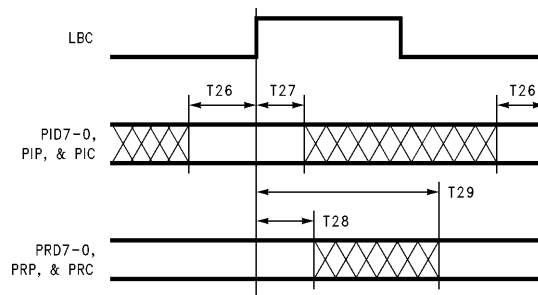


TL/F/10387-32

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

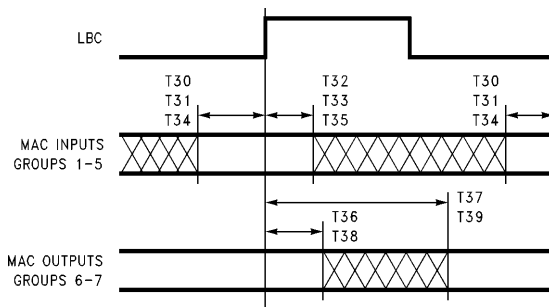
## 8.0 Electrical Characteristics (Continued)

### 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, SAMESA, INFORCVD, SAMEINFO, TXRDY, TXACK, TXCLASS, THTDIS, TXRINGOP, DIAG, DARCV D, AFLAG, SARCV D, 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).

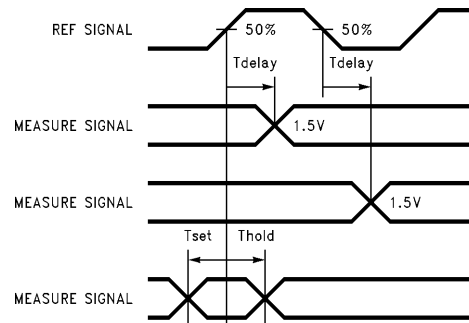
**FIGURE 8-7. MAC Interface Timings**

## 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 (ACK, INT)
$C_L$	50 pF

### AC Signal Testing

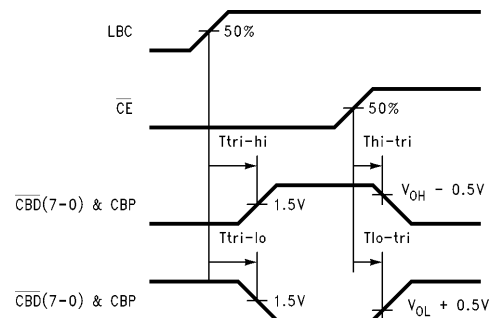


TL/F/10387-35

**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

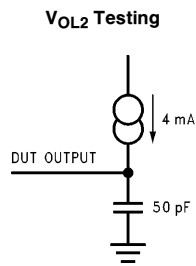


TL/F/10387-36

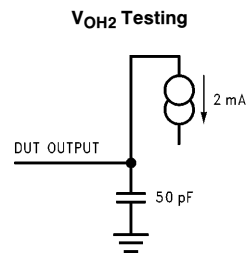
FIGURE 8-9. TRI-STATE Timing

## 8.0 Electrical Characteristics (Continued)

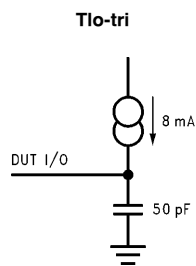
### Test Equivalent Loads



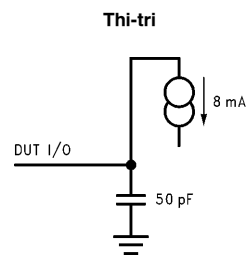
TL/F/10387-37



TL/F/10387-38

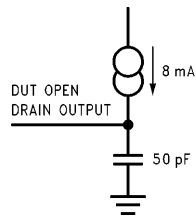


TL/F/10387-39



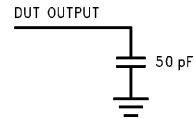
TL/F/10387-40

### Open Drain V<sub>OL</sub> Testing



TL/F/10387-41

### AC, V<sub>OL1</sub>, V<sub>OH1</sub> Testing



TL/F/10387-42

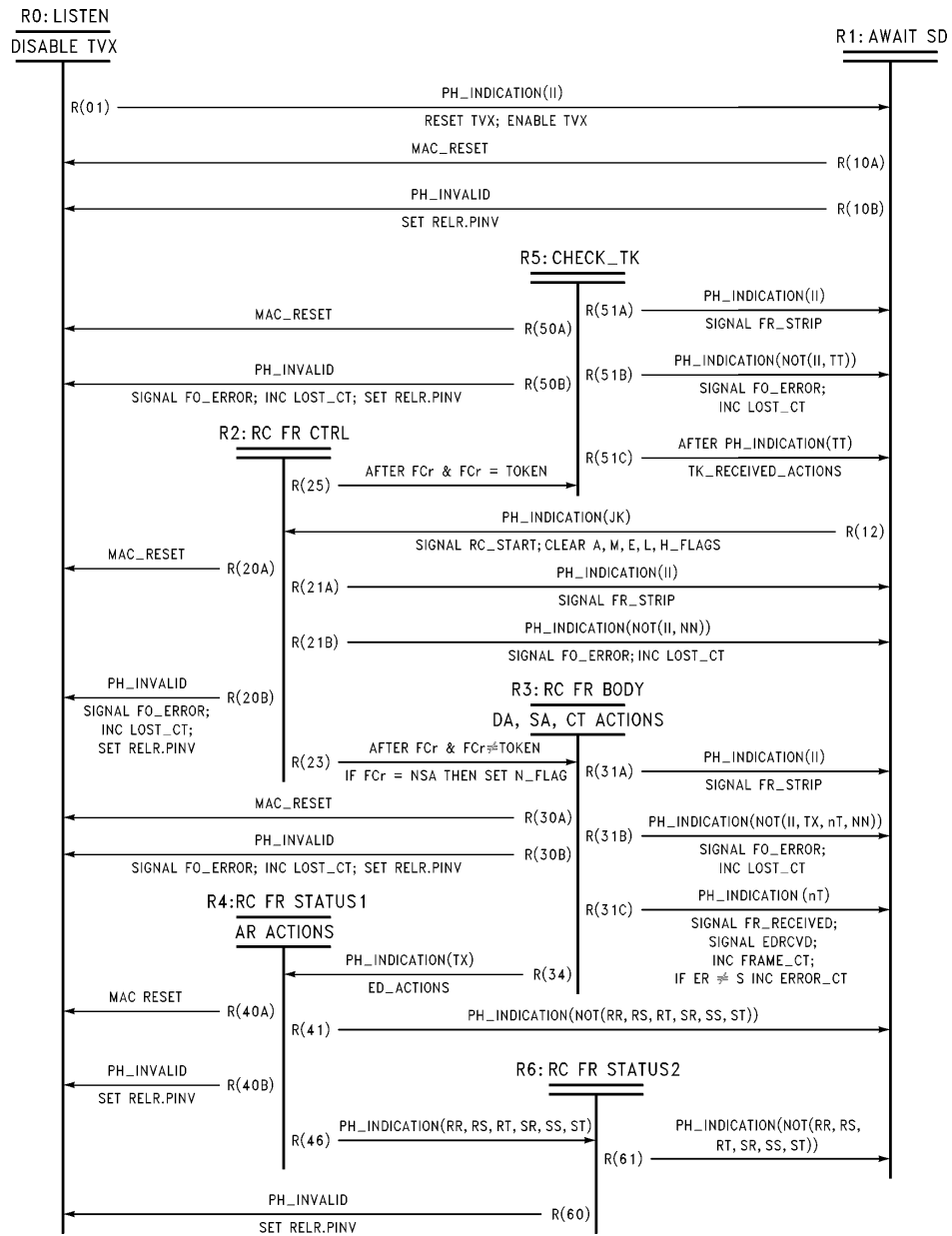
**FIGURE 8-10. Test Equivalent Loads**



# Appendix A. Ring Engine State Machines

## A.1 RECEIVER

### A.1.1 MAC Receiver State Diagram



TL/F/10387-43

FIGURE A-1. Ring Engine Receiver State Diagram

## Appendix A. Ring Engine State Machines (Continued)

### A.1.2 MAC RECEIVER FOOTNOTES

#### A1.2.1 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 DA0r

```
IF DA.IG = 0 CLEAR DAIG ;individual address
ELSE SET DAIG ;group address
```

IF FCSL = 0 ;short address

After DA1r

```
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 SAMEsA

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 SAMEsA
```

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.FRRCV
  IF Valid_Data_Length & (Valid_FCS_Rc | (FCr = Void) |
  (FCr = Implementer and ¬(Option.IFCS))
  THEN RESET TVX;
  IF (A_Flag | (EA & Option.EMIND)) & VCOPY
  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 (VCOFY & FCr ≠ NSA) | (VCOFY & FCr = NSA & Ar = R)
      THEN SET CILR.FRCOP
        INC FCCT {Frame_Copied_Ct}
      ELSE IF ¬VCOFY | (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 VCOFY
      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

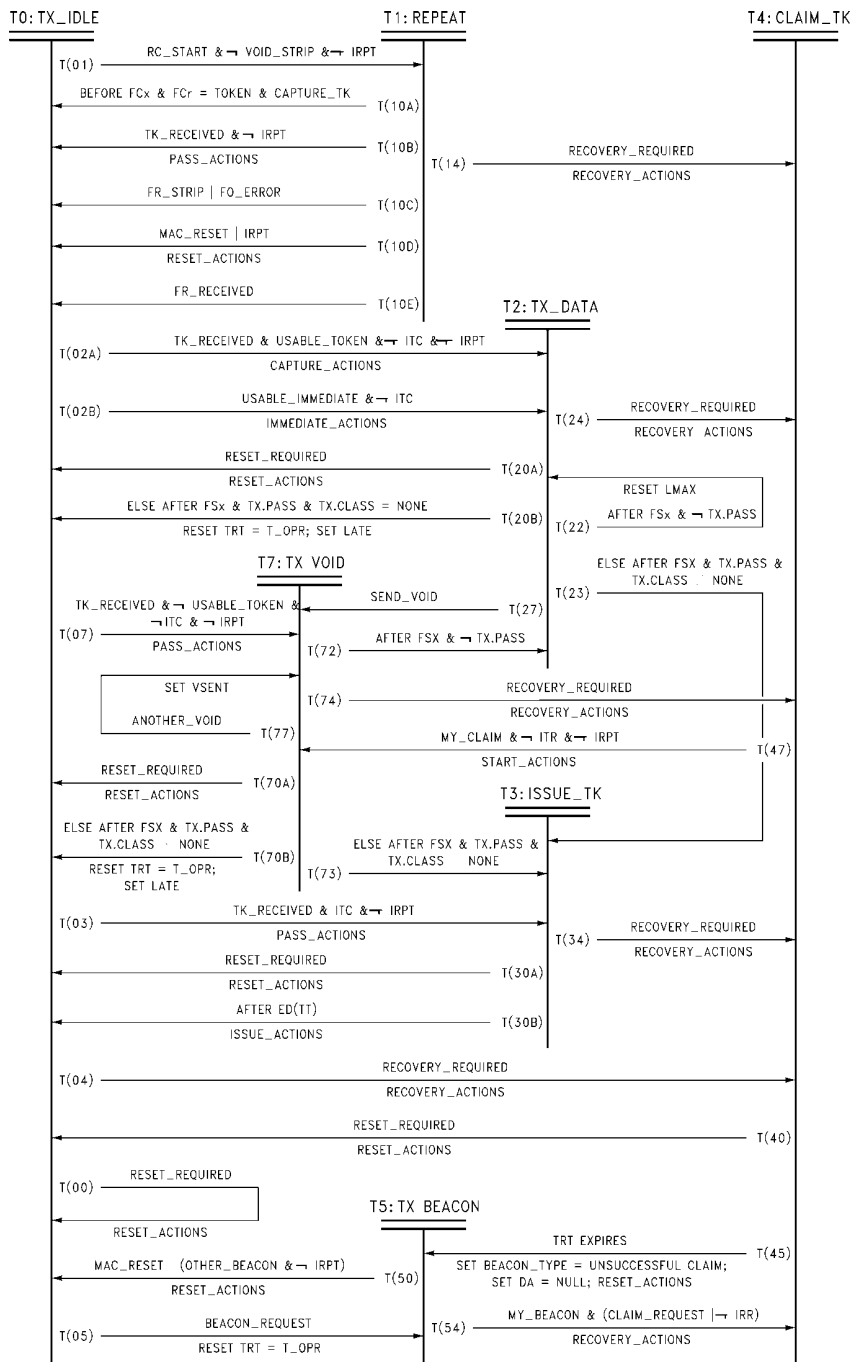


FIGURE A-2. Ring Engine Transmitter State Diagram

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 | (T0 & (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 & ¬((T0 | 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):

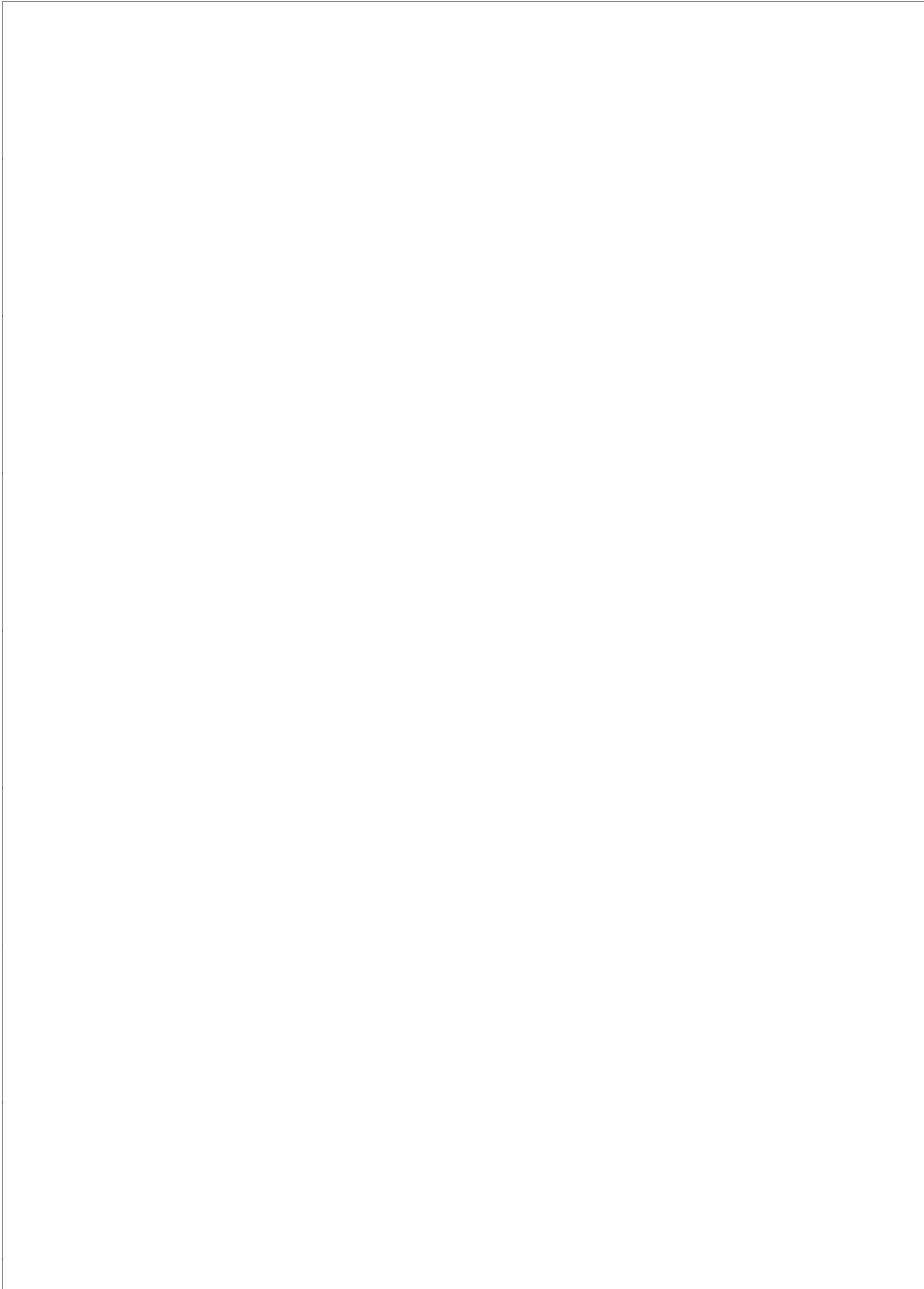
```
CLEAR Function.Claim_Request
```

### (8) TX\_Beacon\_Actions (T5):

```
CLEAR Function.Beacon_Request
```

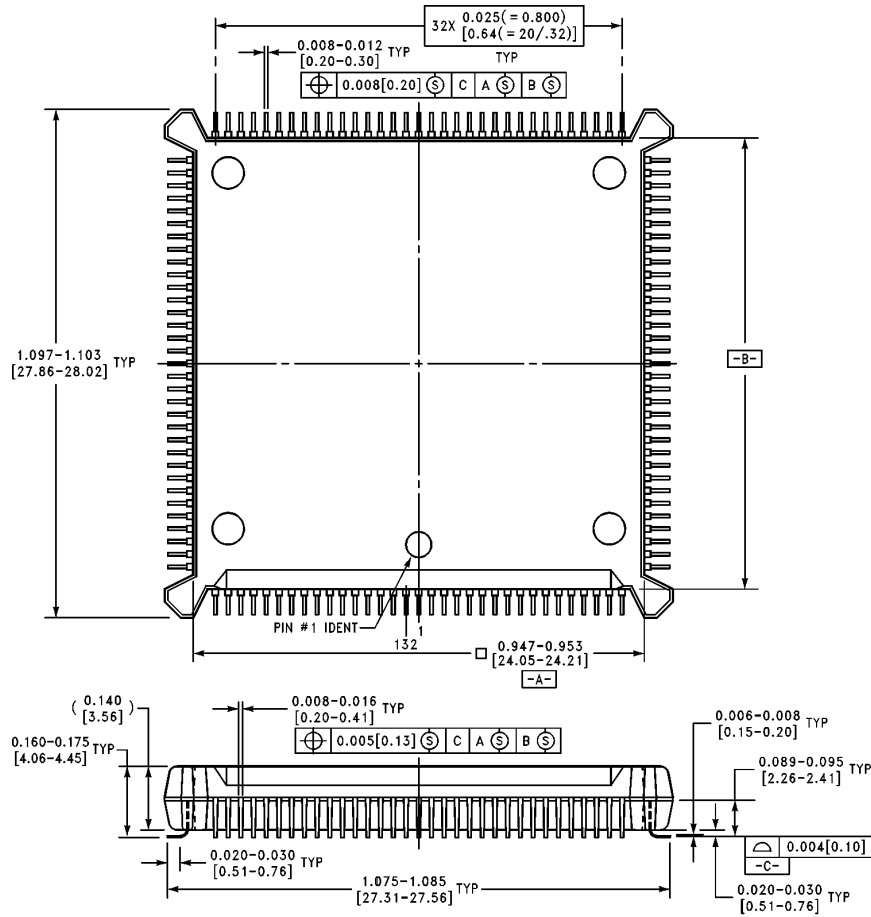
### (9) TX\_Void\_Actions (T7):

```
Always:
  IF My_Void & Vsent
    THEN CLEAR Void_Strip
```



**DP83261 BMAC Device (FDDI Media Access Controller)**

**Physical Dimensions** inches (millimeters)



VF132A (REV D)

**132-Pin PQFP**  
**Order Number DP83261AVF**  
**NS Package Number VF132A**

**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**  
 1111 West Bardin Road  
 Arlington, TX 76017  
 Tel: 1(800) 272-9959  
 Fax: 1(800) 737-7018

**National Semiconductor Europe**  
 Fax: (+49) 0-180-530 85 86  
 Email: cnjwge@tevm2.nsc.com  
 Deutsch Tel: (+49) 0-180-530 85 85  
 English Tel: (+49) 0-180-532 78 32  
 Français Tel: (+49) 0-180-532 93 58  
 Italiano Tel: (+49) 0-180-534 16 80

**National Semiconductor Hong Kong Ltd.**  
 19th Floor, Straight Block,  
 Ocean Centre, 5 Canton Rd.  
 Tsimshatsui, Kowloon  
 Hong Kong  
 Tel: (852) 2737-1600  
 Fax: (852) 2736-9960

**National Semiconductor Japan Ltd.**  
 Tel: 81-043-299-2309  
 Fax: 81-043-299-2408

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 and specifications.