National Semiconductor

PRELIMINARY
November 1994

DP83265A BSI-2™ Device (FDDI System Interface)

General Description

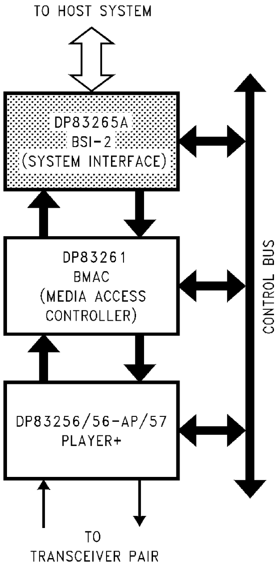
The DP83265A BSI-2 device implements an interface between the National FDDI BMAC™ device and a host system. It provides a multi-frame, MAC-level interface to one or more MAC Users. It is an enhanced version of the DP83265 BSI™ device.

The BSI-2 device accepts MAC User requests to receive and transmit multiple frames (Service Data Units). On reception (Indicate), it receives the byte stream from the BMAC device, packs it into 32-bit words and writes it to memory. On transmission (Request), it unpacks the 32-bit wide memory data and sends it a byte at a time to the BMAC device. The host software and the BSI-2 device communicate via registers, memory-resident descriptors, and an attention/notify scheme using clustered interrupts.

Features

- Fully software and pin compatible with the original BSI
- 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

Block Diagram



TL/F/11407–1

FIGURE 1-1. FDDI Chip Set

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

Table of Contents

1.0 FDDI Chip Set Overview

National Semiconductor's FDDI chip set includes the three components as shown in *Figure 1-1.* For more information about the other devices in 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.

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 the additional PHY_Data.request and PHY_Data.indicate ports required for concentrators and dual attach, dual path stations.

DP83261 BMAC Device Media Access Controller

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

Features

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

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 General Features

The BSI-2 device implements a system interface for the FDDI BMAC Device. It is designed to provide a high-performance, low-cost interface for a variety of hosts.

On the system side, the BSI-2 device provides a simple yet powerful bus interface and memory management scheme to maximize system efficiency. It is capable of interfacing to a variety of host busses/environments. The BSI-2 device provides a 32-bit wide multiplexed address/data interface, which can be configured to share a system bus to main memory or communicate via external shared memory. The system interface supports virtual addressing using fixed-size pages.

On the network side, the BSI-2 device performs many functions which greatly simplify the interface to the BMAC device, and provides many services which simplify network management and increase system performance and reliability. The BSI-2 device is capable of batching confirmation and Indication status, filtering out MAC frames with the same Information field and VOID frames, and performing network monitoring functions.

2.1 32-BIT ADDRESS/DATA PATH TO HOST MEMORY

The BSI-2 device provides a 32-bit wide synchronous multiplexed address/data interface, which permits interfacing to a standard multi-master system bus operating from 12.5 MHz to 33 MHz, or to local memory, using Big or Little Endian byte ordering. The memory may be static or dynamic. For maximum performance, the BSI-2 device utilizes burst mode transfers, with four or eight 32-bit words to a burst. To assist the user with the burst transfer capability, the three bits of the address which cycle during a burst are output demultiplexed. Maximum burst speed is one 32-bit word per clock, but slower speeds may be accommodated by inserting wait states.

The BSI-2 device can operate within any combination of cached/non-cached, paged or non-paged memory environments. To provide this capability, all data structures are contained within a page, and bus transactions never cross a page. The BSI-2 device performs all bus transactions within aligned blocks to ease the interface to a cached environment.

2.2 MULTI-CHANNEL ARCHITECTURE

The BSI-2 device provides three Input Channels and two Output Channels, which are designed to operate independently and concurrently. They are separately configured by the user to manage the reception or transmission of a particular kind of frame (for example, synchronous frames only).

2.3 SUPPORT FOR HEADER/INFO SPLITTING

In order to support high performance protocol processing, the BSI-2 device can be programmed to split the header and information portions of (non-MAC/SMT) frames between two Indicate Channels. Frame bytes from the Frame Control field (FC) up to the user-defined header length are copied onto Indicate Channel 1, and the remaining bytes (Info) are copied onto Indicate Channel 2. This is useful for separating protocol headers and data. It also allows them to be stored in different regions of memory which can prevent unnecessary copying. In addition, a protocol monitor application may decide to copy only the header portion of each frame.

2.4 MAC BRIDGING SUPPORT

Support for bridging and monitoring applications is provided by the Internal/External Sorting Mode. All frames matching the external address (frames requiring bridging) are sorted onto Indicate Channel 2, MAC and SMT frames matching the internal (BMAC device) address are sorted onto Indicate Channel 0, and all other frames matching the BMAC device's internal address (short or long) are sorted onto Indicate Channel 1.

2.5 ADDRESS BIT SWAPPING

The BSI-2 contains the necessary logic for swapping the address fields within each frame between FDDI and IEEE Canonical bit order. This involves a bit reversal within each byte of the address field (e.g. 08-00-17-C2-A1-03 would become 10:00:E8:43:85:C0). This option is selectable on a per channel basis and is supported on all channels, both transmit and receive. This is useful for bridging FDDI to Ethernet or for swapping addresses for higher level protocols.

2.6 STATUS BATCHING SERVICES

The BSI-2 device provides status for transmitted and received frames. Interrupts to the host are generated only at status breakpoints, which are defined by the user on a per Channel basis. Breakpoints are selected when the Channel is configured for operation. To allow batching, the BSI-2 provides a status option called Tend which causes the device to generate a single Confirmation Message Descriptor (CNF) for one or more Request Descriptors (REQs).

The BSI-2 device further reduces host processing time by separating received frame status from the received data. This allows the CPU to scan quickly for errors when deciding whether further processing should be done on received frames. If the status were embedded in the data stream, all the data would need to be read contiguously to find the Status Indicator.

2.7 RECEIVE FRAME FILTERING SERVICES

To increase performance and reliability, the BSI-2 device can be programmed to filter out identical MAC (same FC and Info field) or SMT frames received from the ring. VOID frames are filtered out automatically. Filtering unnecessary frames reduces the fill rate of the Indicate FIFO, reduces CPU frame processing time, and avoids unnecessary memory bus transactions.

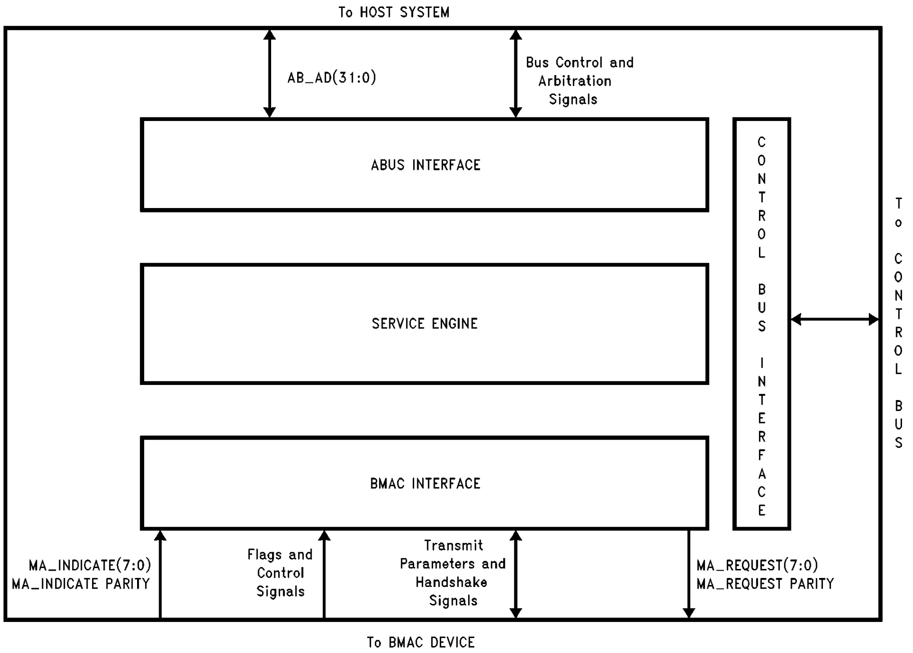
2.8 TWO TIMING DOMAINS

To provide maximum performance and system flexibility, the BSI-2 device utilizes two independent clocks, one for the MAC (ring) Interface, and one for the system/memory bus. The BSI-2 device provides a fully synchronized interface between these two timing domains.

2.9 CLUSTERED INTERRUPTS

The BSI-2 device can be operated in a polled or interrupt-driven environment. The BSI-2 device provides the ability to generate attentions (interrupts) at group boundaries. Some boundaries are pre-defined in hardware; others are defined by the user when the Channel is configured. This interrupt scheme significantly reduces the number of interrupts to the host, thus reducing host processing overhead.

3.0 Architecture Description



TL/F/11407–2

FIGURE 3-1. BSI-2 Device Interfaces

The BSI-2 device is composed of three interfaces and the Service Engine.

The three interfaces are the BMAC device, the ABus, and the Control Bus Interfaces. They are used to connect the BSI-2 device to the BMAC device, Host System, and external Control Bus respectively.

The Service Engine manages the operation of the BSI-2 device.

3.1 INTERFACES

The BSI-2 device connects to external components via three interfaces: the BMAC device Interface, the ABus Interface, and the Control Bus Interface (see *Figure 3-1*).

3.1.1 BMAC Device Interface

The BSI-2 device connects to the BMAC device via the MA__Indicate (receive) and MA__Request (transmit) Interfaces, as shown in *Figure 3-1.*

Received Data is transferred from the BMAC device to the BSI-2 device via the MA__Indicate Interface. The MA__Indicate Interface consists of a parity bit (odd parity) and byte-wide data along with flag and control signals.

Transmit Data is transferred from the BSI-2 device to the BMAC device via the MA__Request Interface. The MA__Request Interface consists of a parity bit (odd parity) and byte-wide data along with flag and control signals.

3.1.2 ABus Interface

The BSI-2 device connects to the Host System via the ABus Interface. The ABus Interface consists of four bits of parity (odd parity) and 32 bits of multiplexed address and data along with transfer control and bus arbitration signals.

3.1.3 Control Bus Interface

The Control Bus Interface connects the BSI-2 device to the external Control Bus.

The Control Bus Interface is separate from the BMAC device and ABus Interfaces to allow independent operation of the Control Bus.

The host uses the Control Bus to access the BSI-2 device's internal registers, and to manage the attention/notify logic.

3.2 DATA STRUCTURES

3.2.1 Data Types

The architecture of the BSI-2 device defines two basic kinds of objects: Data Units and Descriptors. A Data Unit is a group of contiguous bytes which forms all or part of a frame. A Descriptor is a two-word (64-bit) control object that provides addressing information and control/status information about BSI-2 device operations.

Data and Descriptor objects may consist of one or more parts, where each part is contiguous and wholly contained within a memory page. Descriptor pages are selectable as all 1 kbytes or all 4 kbytes. Data Units are described by Descriptors with a pointer and a count. A single Data Unit may not cross a 4k boundary. All Descriptors may be marked as **First**, **Middle**, **Last**, or **Only**. Thus, multiple Descriptors may be combined to describe a single entity (i.e. Frame). A single-part object consists of one **Only** Part; a multiple-part object consists of one **First** Part, zero or more **Middle** Parts, and one **Last** Part. In Descriptor names, the object part is denoted in a suffix, preceded by a dot. Thus an Input Data Unit Descriptor (IDUD), which describes the last Data Unit of a frame received from the ring, is called an IDUD.Last.

3.0 Architecture Description (Continued)

A Data Unit is stored in contiguous locations within a single 4 kbyte page in memory. Multiple-part Data Units are stored in separate, and not necessarily contiguous 4 kbyte pages. Descriptors are stored in contiguous locations in Queues and Lists, where each Queue occupies a single 1 kbyte or 4 kbyte memory page, aligned on the queue-size boundary. For Queues, an access to the next location after the end of a page will automatically wrap-around and access the first location in the page.

Data Units are transferred between the BSI-2 device and BMAC device via five simplex Channels, three used for Indicate (receive) data and two for Request (transmit) data. Parts of frames received from the ring and copied to memory are called Input Data Units (IDUs); parts of frames read from memory to be transmitted to the ring are called Output Data Units (ODUs).

Descriptors are transferred between the BSI-2 device and Host via the ABus, whose operation is for the most part transparent to the user. There are five Descriptor types recognized by the BSI-2 device: Input Data Unit Descriptors (IDUDs), Output Data Unit Descriptors (ODUDs), Pool Space Descriptors (PSPs), Request Descriptors (REQs), and Confirmation Message Descriptors (CNFs).

Input and Output Data Unit Descriptors describe a single Data Unit part, i.e., its address (page number and offset), its size in bytes, and its part (Only, First, Middle, or Last). Frames consisting of a single part are described by a Descriptor.Only; frames consisting of multiple parts are described by a Descriptor.First, zero or more Descriptor.Middles, and a Descriptor.Last.

Every Output Data Unit part is described by an Output Data Unit Descriptor (ODUD). Output Data Unit Descriptors are fetched from memory so that frame parts can be assembled for transmission.

Every Input Data Unit part is described by an Input Data Unit Descriptor (IDUD). Input Data Unit Descriptors are generated on Indicate Channels to describe where the BSI-2 device wrote each frame part and to report status for the frame.

Request Descriptors (REQs) are written by the user to specify the operational parameters for BSI-2 device Request operations. Request Descriptors also contain the start address of part of a stream of ODUDs and the number of frames represented by the ODUD stream part (i.e., the number of ODUD.Last descriptors). Typically, the user will define a single Request Object consisting of multiple frames of the same request and service class, frame control, and expected status.

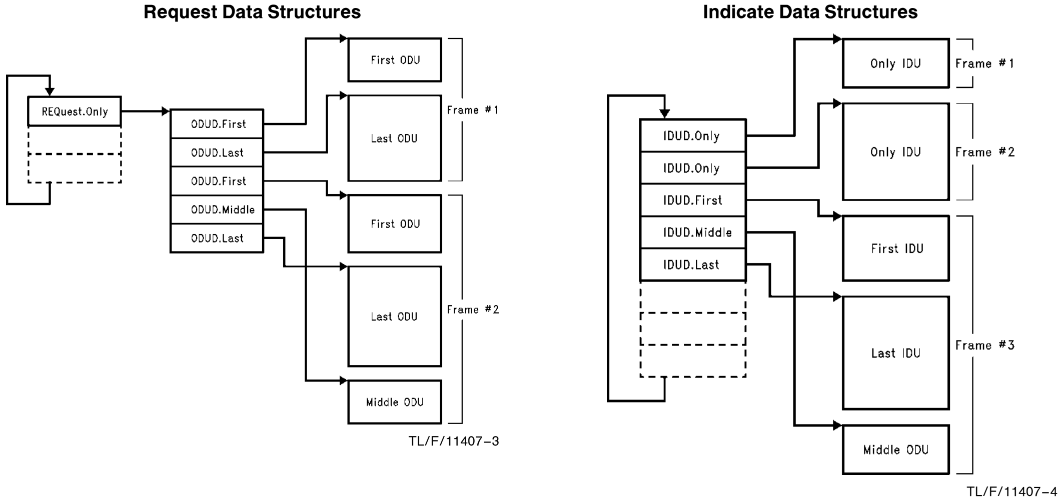
Confirmation Messages (CNFs) are created by the BSI-2 device to record the result of a Request operation.

Pool Space Descriptors (PSPs) describe the location and size of a region of memory space available for writing Input Data Units.

Request (transmit) and Indicate (receive) data structures are summarized in *Figure 3-2.*

3.2.2 Descriptor Queues and Lists

The BSI-2 device utilizes 10 Queues and two Lists. These queues and lists are circular. There are six Queues for Indicate operations, and four Queues and two Lists for Request operations. Each of the three Indicate Channels has a Data Queue containing Pool Space Descriptors (PSPs), and a Status Queue containing Input Data Unit Descriptors



TL/F/11407–3

TL/F/11407–4

FIGURE 3-2. BSI-2 Device Data Structures

(IDUDs). Each Request Channel has a Data Queue containing Request Descriptors (REQs), a Status Queue containing Confirmation Messages (CNFs), and a List containing Output Data Unit Descriptors (ODUDs).

During Indicate and Request operations, Descriptor Queues and Lists are read and written by the BSI-2 device, using registers in the Pointer and Limit RAM Register files. The Pointer RAM Queue and List Pointer Registers point to a location from which a Descriptor will be read (PSPs and REQs) or written (IDUDs and CNFs). All of the Queues and Lists are strictly unidirectional. The BSI-2 consumes objects in those queues which are produced by the Host. The Host consumes objects in those queues which are produced by the BSI-2.

For each Queue Pointer Register there is a corresponding Queue Limit Register in the Limit RAM Register file, which holds the Queue's limit as an offset value in units of 1 Descriptor (8 bytes). The address in the Queue Pointer is incremented before a Descriptor is read and after a Descriptor is written, then compared with the value in the corresponding Queue Limit Register. When a Queue Pointer Register becomes equal to the Queue Limit Register, an attention is generated, informing the host that the Queue is empty. When a pointer value is incremented past the end of the page, it wraps to the beginning of the page.

3.2.3 Storage Allocation

The maximum unit of contiguous storage allocation in external memory is a Page. All BSI-2 device addresses consist of a 16-bit page number and a 12-bit offset.

The BSI-2 device uses a page size of 1 kbyte or 4 kbytes for storage of Descriptor Queues and Lists (as selected by the user), and a page size of 4 kbytes for storage of Data Units. A single page may contain multiple Data Units, and multiple-part Data Units may span multiple, disjoint or contiguous pages.

3.3 SERVICE ENGINE

The Service Engine, which manages the operation of the BSI-2, is comprised of seven basic blocks: Indicate Machine, Request Machine, Status/ Space State Machine, Operation RAM, Pointer RAM, Limit RAM, and Bus Interface Unit. An internal block diagram of the BSI-2 device is shown in *Figure 3-3.*

3.3.1 Indicate Machine

The Indicate Block accepts Service Data Units (frames) from the BMAC device in the byte stream format (MA__Indicate).

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

- Decodes the Frame Control field to determine the frame type
- Sorts the received frames onto Channels according to the Sort Mode
- Optionally filters identical MAC frames
- Filters VOID frames
- Copies the received frames to memory according to Copy Criteria
- Writes status for the received frames to the Indicate Status Queue
- Issues interrupts to the host at host-defined status breakpoints

3.3.2 Request Machine

The Request Machine presents Service Data Units (MAC frames) to the BMAC device in the byte stream format (MA__Request).

The Request Machine performs the following functions:

- Reads frames from host memory and assembles them onto Request Channels
- Prioritizes active requests
- Transmits frames to the BMAC device
- Optionally Writes status for transmitted and returning frames
- Issues interrupts to the host on user-defined group boundaries

3.3.3 Status/Space Machine

The Status/Space Machine is used by both the Indicate Machine and the Request Machine.

The Status/Space Machine manages all descriptor Queues and writes status for received and transmitted frames.

3.3.4 Bus Interface Unit

The Bus Interface Unit (BIU) is used by both the Indicate and Request Blocks. It manages the ABus Interface, providing the BSI-2 device with a 32-bit data path to local or system memory.

The Bus Interface Unit controls the transfer of Data Units and Descriptors between the BSI-2 device and Host memory via the ABus.

Data and Descriptors are transferred between the BSI-2 device and Host memory. Each Channel type handles a set of Data and Descriptor objects. The three Indicate (Receive) Channels use the following objects:

1. Input Data Units (written by BSI-2)
2. Input Data Unit Descriptors (written by BSI-2)
3. Pool Space Descriptors (read by BSI-2)

The two Request (Transmit) Channels each use the following objects:

1. Output Data Units (read by BSI-2)
2. Output Data Unit Descriptors (read by BSI-2)
3. Confirmation Message Descriptors (written by BSI-2)
4. Request Descriptors (read by BSI-2)

Each Channel will only process one object type at a time. The BIU arbitrates between the Channels and issues a Bus Request when any Channel requests service. The priority of Channel bus requests is generally as follows, from highest priority to lowest priority:

1. Indicate Data Unit writes (highest priority)
2. Output Data Unit fetches
3. Request Descriptor and Output Data Unit Descriptor fetches
4. Input Data Unit Descriptor writes
5. Confirmation Message Descriptor writes
6. Next Pool Space Descriptor transfer to Current Pool Space Descriptor (internal operation)
7. Pool Space Descriptor fetches
8. Limit RAM Operations (internal operation)
9. Pointer RAM Operations (lowest priority)

Addresses for Channel accesses are contained in the Pointer RAM Registers.

3.0 Architecture Description (Continued)

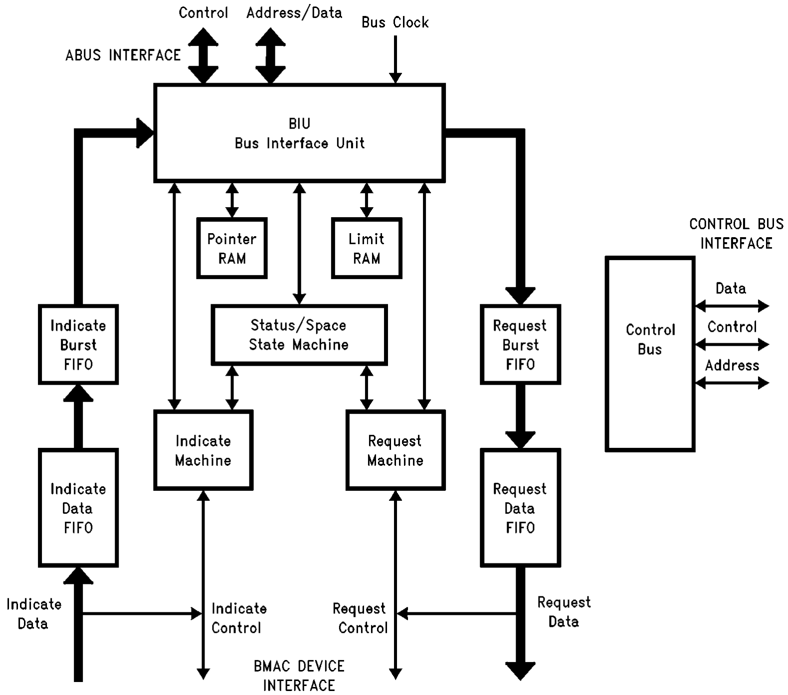


FIGURE 3-3. BSI-2 Device Internal Block Diagram

3.3.5 Pointer RAM

The Pointer RAM Block is used by both the Indicate and Request Machines. It contains pointers to all Data Units and Descriptors manipulated by the BSI-2 device, namely, Input and Output Data Units, Input and Output Data Unit Descriptors, Request Descriptors, Confirmation Message Descriptors, and Pool Space Descriptors.

The Pointer RAM Block is accessed by clearing the PTOP (Pointer RAM Operation) bit in the Service Attention Register, which causes the transfer of data between the Pointer RAM Register and a mailbox location in memory.

3.3.6 Limit RAM

The Limit RAM Block is used by both the Indicate and Request Machines. It contains data values that define the limits of the ten Queues maintained by the BSI-2 device.

Limit RAM Registers are accessed by clearing the LMOP (Limit RAM Operation) bit in the Service Attention Register, which causes the transfer of data between the Limit RAM Register and the Limit Data and Limit Address Registers.

4.0 Functional Description

The BSI-2 device is composed of the Service Engine and Interfaces to the Control Bus (Control Bus Interface), the BMAC device (BMAC Device Interface) and the ABus (ABus Interface).

In this section, the Service Engine is described in detail to provide an in-depth look at the operation of the BSI-2 device.

4.1 OVERVIEW

The Service Engine consists of two major blocks, the Indicate Machine and the Request Machine. These blocks share the Bus Interface Unit, Status/Space Machine, Pointer RAM, and Limit RAM blocks.

The Service Engine provides an interface between the BMAC FDDI Media Access Control Protocol chip and a host system. The Service Engine transfers FDDI frames between the FDDI device and host memory.

4.1.1 Indicate Machine

On the Receive side (from the ring) the Indicate Machine sequences through the incoming byte stream from the BMAC device. Received frames are sorted onto Indicate Channels and a decision is made whether or not to copy them to host memory. The Indicate Machine uses the control signals provided by the BMAC device Receive State Machine on the MAC Indicate Interface to make this decision.

4.1.2 Request Machine

On the Transmit side (to the ring) the Request Machine prepares one or more frames from host memory for transmission to the BMAC device. The Request Machine provides all the control signals to drive the BMAC device Request Interface.

4.2 OPERATION

4.2.1 Indicate Operation

The Indicate Block accepts data from the BMAC device as a byte stream.

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

- Decodes the Frame Control field to determine the frame type
- Sorts the received frames onto Channels according to the Sort Mode
- Optionally filters identical MAC frames
- Filters VOID frames
- Copies the received frames to memory according to Copy Criteria
- Writes status for the received frames to the Indicate Status Queue
- Issues interrupts to the host on host-defined status breakpoints

The Indicate Machine decodes the Frame Control (FC) field to determine the type of frame. The following types of frames are recognized: Logical Link Control (LLC), Restricted Token, Unrestricted Token, Reserved, Station Management (SMT), SMT Next Station Addressing, MAC Beacon, MAC Claim, Other MAC, and Implementer.

The Indicate Machine sorts incoming frames onto Indicate Channels according to the frame's FC field, the state of the AFLAG signal from the BMAC device (which indicates that the BMAC had an address match), and the host-defined sorting mode programmed in the Sort Mode field of the Indicate Mode Register. SMT and MAC frames are always sorted onto Indicate Channel 0. On Indicate Channels 1 and 2, frames can be sorted according to whether they are synchronous or asynchronous, high-priority asynchronous or low-priority asynchronous, whether their address matches an internal (BMAC device) or external address, or based on header and Information fields for all non-MAC/SMT frames.

The Synchronous/Asynchronous Sort Mode is intended for use in end-stations or applications using synchronous transmission.

With High-priority/Low-priority sorting, high-priority asynchronous frames are sorted onto Indicate Channel 1 and low-priority asynchronous frames are sorted onto Indicate Channel 2. The most-significant bit of the three-bit priority field within the FC field determines the priority. This Mode is intended for end stations using two priority levels of asynchronous transmission. Synchronous frames are sorted to Indicate Channel 1 in this mode.

With External/Internal sorting, frames matching the internal address (Individual or Group addresses in the BMAC device) are sorted onto Indicate Channel 1 and frames matching an external address (when the EA input is asserted) are sorted onto Indicate Channel 2. Note that under some conditions it is possible to sort internal address match and SMT/MAC frames to Indicate Channel 2. Please see Section 4.3 for full details on the External Matching Interface. This sort mode is intended for bridges or ring monitors, which would use the ECIP/EA/EM pins with external address matching circuitry. However, designers should be aware of the functioning of the ECIP pin even if external matching will not be used. If ECIP is left in an improper state (e.g floating or tied high), it will affect the operation of the BSI-2 device even when External/Internal sorting is not enabled.

With the Header/Info Sort Mode, Indicate Channels 1 and 2 receive all non-MAC/SMT frames that are to be copied, but between them split the frame header (whose length is user-defined) and the remaining portions of the frame (Info). Indicate Channel 1 copies the initial bytes up until the host-defined header length is reached. The remainder of the frame's bytes are copied onto Indicate Channel 2. Only one IDUD stream is produced (on Indicate Channel 1), but both PSP Queues are used to determine where the IDUs will be written. When a multi-part IDUD is produced, the Indicate Status field is used to determine which parts point to the header and which point to the Info. This Mode is intended for high-performance protocol processing applications.

The Indicate Machine filters identical MAC and SMT frames when the SKIP bit in the Indicate Mode Register is set, and the Indicate Configuration Register's Copy Control field (2 bits) for Indicate Channel 0 is set to 01 or 10.

Received frames are copied to memory based on the AFLAG and MFLAG, ECIP, EA, and EM input signals from external address matching logic, input signals from the BMAC device, as well as the Indicate Channel's Copy Control field. Received frames are written as a series of Input Data Units to the current Indicate page. Each frame is aligned to the start of a currently-defined, burst-size memory block (16 or 32 bytes as programmed in the Mode Register's SMLB bit). The first word contains the FC only, copied into all bytes of the first word written, with the DA, SA and INFO fields aligned to the first byte of the next word. The format differs according to the setting of the Mode Register's BIGEND (Big Endian) bit, as shown in *Figure 4-1.*

Byte 0 **Byte 3**

Bit 31 **Big Endian Indicate Data Unit Format** **Bit 0**

FC	FC	FC	FC
DA0	DA1	SA0	SA1

Byte 3 **Byte 0**

Bit 0 **Little Endian Indicate Data Unit Format** **Bit 31**

FC	FC	FC	FC
SA1	SA0	DA1	DA0

FIGURE 4-1. Indicate Data Unit Formats (Short Address)

For each Input Data Unit, the Indicate Machine creates an Input Data Unit Descriptor (IDUD), which contains status information about the IDU, its size (byte count), and its location in memory. For IDUs that fit within the current Indicate page, an IDUD.Only Descriptor is created. For IDUs that span more than one page, a multi-part IDUD is created, i.e., when a frame crosses a page boundary, the BSI-2 device writes an IDUD.First; if another page is crossed, an IDUD.Middle will be written; and at the frame end, an IDUD.Last is written. IDUDs are written to consecutive locations in the Indicate Status Queue for the particular Indicate Channel, up to the host-defined queue limit.

The BSI-2 has two modes for storing IDUs into Pool Space Pages. In the first mode, the BSI-2 will assemble as many frames into a 4 kbyte page as will fit. Thus, a single page of Pool Space may contain multiple frames and have many IDUDs pointing to it. In the second mode, the BSI-2 forces a page break after the end of each frame. This means that a single page of Pool Space will have at most a single IDUD pointing to it. This mode greatly simplifies space reclamation in those systems which do not process incoming frames in order of receipt and supports systems in which the cache line size is greater than 32 bytes.

The Indicate Machine copies IDUs and IDUDs to memory as long as there are no exceptions or errors, and the Channel has data and status space. When a lack of either data or status space is detected on a particular Channel, the Indicate Machine stops copying new frames for that Channel (only). It will set the No Status Space attention bit in the No Space Attention Register when it runs out of Status Space. It will set the Low Data Space bit in the No Space Attention Register when the last available PSP is prefetched from the Indicate Channel PSP Queue. The host allocates more data space by adding PSPs to the tail of the PSP Queue and then updating the PSP Queue Limit Register, which causes the BSI-2 device to clear the Low Data Space attention bit and resume copying (on the same Channel). The user should **never** clear the Low Data Space attention bits directly. The host allocates more status space by updating the IDUD Queue Limit Register and then explicitly clearing the Channel's No Status Space bit, after which the Indicate Machine resumes copying. Note that the No Status Space Attention bit must be cleared **after** the appropriate limit register is updated.

The BSI-2 device provides the ability to group incoming frames and then generate interrupts (via attentions) at group boundaries. To group incoming frames, the BSI-2 device defines status breakpoints, which identify the end of a group (burst) of related frames. Status breakpoints can be enabled to generate an attention.

The breakpoints for Indicate Channels are defined by the host in the Indicate Mode, Indicate Notify, and Indicate Threshold registers. Status breakpoints include Channel change, receipt of a token, SA change, DA change, MAC Info change, and the fact that a user-specified number of frames has been copied on a particular Indicate Channel.

Status breakpoint generation may be individually enabled for Indicate Channels 1 and 2 by setting the corresponding Breakpoint bits (Breakpoint on Burst End, Breakpoint on Service Opportunity, and Breakpoint on Threshold) in the Indicate Mode Register, and enabling the breakpoints to generate an attention by setting the corresponding Breakpoint bit in the Indicate Notify Register.

When an Indicate exception occurs, the current frame is marked complete, status is written into an IDUD.Last, and the Channel's Exception (EXC) bit in the Indicate Attention Register is set.

When an Indicate error (other than a parity error) is detected, the Error (ERR) bit in the State Attention Register is set. The host must reset the INSTOP Attention bit to restart processing.

When parity checking is enabled and a parity error is detected in a received frame, it is recorded in the Indicate Status field of the IDUD, and the BMAC device Parity Error (PBE) bit in the Status Attention Register is set.

A frame which is stripped after the fourth byte of the Information Field (this may occur because an upstream station detected an error within the frame) will be copied to memory but the status will show that the frame was stripped.

4.2.2 Request Operation

The Request Block transmits frames from host memory to the BMAC device. Data is presented to the BMAC device as a byte stream.

The Request Block performs the following functions:

- Prioritizes active requests to transmit frames
- Requests the BMAC device to obtain a token
- Transmits frames to the BMAC device
- Writes status for transmitted and returning frames
- Issues interrupts to the host on user-defined group boundaries

The Request Machine processes requests by reading Request Descriptors from the REQ Queue, then assembling frames of the specified service class, Frame Control (FC) and expected status for transmission to the BMAC device. Request and ODUD Descriptors are checked for consistency, and the Request Class is checked for compatibility with the current ring state. When an inconsistency or incompatibility is detected, the request is aborted.

When a consistency failure occurs, the Request is terminated with appropriate status. The Request Machine then locates the end of the current object (REQ or ODUD). If the current Descriptor is not the end (Last bit not set), the Request Machine will fetch subsequent Descriptors until it detects the end, then resume processing with the next Descriptor.First or Descriptor.Only.

Requests are processed on both Request Channels simultaneously. Their interaction is determined by their priorities (Request Channel 0 has higher priority than Request Channel 1) and the Hold and Preempt/Prestage bits in the Request Channel's Request Configuration Register. An active Request Channel 0 is always serviced first, and may be programmed to preempt Request Channel 1, such that uncommitted Request Channel 1's data already in the request FIFO will be purged and then refetched after servicing Request Channel 0. When prestaging is enabled, the next frame is staged before the token arrives. Prestaging is always enabled for Request Channel 0, and is a programmable option on Request Channel 1. The BSI-2 will process at most, one Request per Channel per Service Opportunity.

The BSI-2 contains an option bit which controls the timing of the Token capture. This bit is the Early Token Request bit (ETR) which is in R0CR1 (for RCHN0) and R1CR1 (for RCHN1). When the ETR bit is disabled for a channel, the BSI-2 will fetch a Request descriptor, then fetch the first ODUD and begin filling the transmit FIFO for that channel. When the FIFO threshold is reached (R0CR0.TT or R1CR0.TT) the BSI-2 presents a Request Class to the BMAC which causes the BMAC to capture a Token of the specified class.

When the ETR bit is enabled, a REQ.First is loaded, the Request Machine commands the BMAC device to capture a token of the type specified in the REQ Descriptor, and concurrently fetches the first ODUD. This mode is useful for systems which need tight control of the Token capture timing (e.g. systems using Synchronous traffic). Note that use of the Early Token Request mechanism may under certain circumstances waste ring bandwidth, (i.e. holding the Token while filling the FIFO). Therefore, it should be enabled only in those systems where the feature is specifically required.

If prestaging is enabled, or a Service Opportunity exists for this Request Channel, data from the first ODU is loaded into the Request FIFO, and the BSI-2 device requests transmission from the BMAC device. When the BMAC device has captured the appropriate token and the frame is committed to transmission (the FIFO threshold has been reached or the end of the frame is in the FIFO), transmission begins. The BSI-2 device fetches the next ODUD and starts loading the ODUs of the next frame into the FIFO. This continues (across multiple service opportunities if required) until all frames for that Request have been transmitted (i.e., an REQ.ONLY or an REQ.LAST is detected), or an exception or error occurs, which prematurely ends the Request.

The BSI-2 device will load REQ Descriptors as long as the RQSTOP bit in the State Attention Register is Zero, the REQ Queue contains valid entries (the REQ Queue Pointer Register does not exceed the REQ Queue Limit Register), and there is space in the CNF Queue (the BSI-2 has not detected equality of the CNF Queue Pointer Register and the CNF Queue Limit Register).

Request status is generated as a single confirmation object (single- or multi-part) per Request object, with each confirmation object consisting of one or more CNF Descriptors. The type of confirmation is specified by the host in the Confirmation Class field of the REQ Descriptor.

The BSI-2 device can be programmed to generate CNF Descriptors at the end of the Request object (End Confirmation), or at the end of each token opportunity (Intermediate Confirmation), as selected in the E and I bits of the Confirmation Class Field of the REQ Descriptor. A CNF Descriptor is always written when an exception or error occurs (regardless of the value in the Confirmation Class field), when a Request completes (for End or Intermediate Confirmation Class), or when a Service Opportunity ends (Intermediate Confirmation Class only).

There are three basic types of confirmation: Transmitter, Full, and None. With Transmitter Confirmation, the BSI-2 device verifies that the Output Data Units were successfully transmitted. With Full Confirmation, the Request Machine verifies that the ODUs were successfully transmitted, that the number of (returning) frames ‘‘matches’’ the number of transmitted ODUs, and that the returning frames contain the expected status. Full confirmation takes advantage of the fact that the sending station also removes its frames from the network. When the None Confirmation Class is selected, confirmation is written only if an exception or error occurs.

For Full Confirmation, a matching frame must meet the following criteria:

1. The frame has a valid Ending Delimiter (ED).
2. The selected bits in the FC fields of the transmitted and received frames are equal (the selected bits are specified in the FCT bit of the Request Configuration Register).
3. The frame is My__SA (MFLAG or both SAT & EM asserted).
4. The frame status indicators match the values in the Expected Frame Status Register.
5. FCS checking is disabled or FCS checking is enabled and the frame has a valid FCS.
6. All bytes from FC to ED have good parity (when the FLOW bit in the Mode Register is set, i.e., parity checking is enabled).

The confirmed frame count starts after the first Request burst frame has been committed by the BMAC device, and when a frame with My__SA is received. Void and My__Void frames are ignored by the BSI-2 device. The frame count ends when any of the following conditions occur:

1. All the frames have been transmitted, and the transmitted and confirmed frame counts are equal.
2. There is a MACRST (MAC Reset).
3. The state of the ring-operation has changed.
4. A stripped frame or a frame with a parity error is received.
5. A non-matching frame is received.
6. A token is received.

When Source Address Transparency is selected (by setting the SAT bit in the Request Configuration Register) and Full confirmation is enabled, confirmation begins when a frame end is detected with either MFLAG or EM asserted.

When a non-matching frame is received, the BSI-2 device ends the Request, and generates the Request Complete (RCM), Exception (EXC), and Breakpoint (BRK) attentions. Any remaining REQs in the Request object are fetched until a REQ.Last or REQ.Only is encountered. Processing then resumes on the next REQ.First or REQ.Only (any other type of REQ would be a consistency failure).

Request errors and exceptions are reported in the State Attention Register, Request Attention Register, and the Confirmation Message Descriptor. When an exception or error occurs, the Request Machine generates a CNF and ends the Request. The Unserviceable Request (USR) attention is set to block subsequent Requests once one becomes unserviceable.

4.2.3 State Machines

There are three state machines under control of the host: the Request Machine, the Indicate Machine, and the Status/Space Machine. Each Machine has two Modes: Stop and Run. The Mode is determined by the setting of the Machine's corresponding STOP bit in the State Attention Register. The STOP bits are set by the BSI-2 device when an error occurs or may be set by the user to place the Machine in Stop Mode.

The BSI-2 Control Registers may be programmed only when all Machines are in Stop Mode. When the Status/Space Machine is in Stop Mode, only the Pointer RAM and Limit RAM Registers may be programmed. When the Indicate and Request Machines are in Stop Mode, all indicate and request operations are halted. When the Status/Space Machine is in Stop Mode, only the PTOP and LMOP service functions can be performed.

4.3 EXTERNAL MATCHING INTERFACE

This interface consists of three pins which are used to give the BSI-2 additional address status about the incoming frame. ECIP is the timing signal. EA and EM are used to report external matches of destination and source addresses respectively.

For the purposes of external matching, it is recommended that the frame data be viewed at the PLAYER-BMAC Receive interface, (PID<7:0>, PIP, PIC). Using this interface will ensure design compatibility with future integrated versions of the chipset. In addition it is recommended that the user design the circuit to detect the JK symbol at the BMAC-PLAYER interface to start the address matching. Although there are timing handshake signals between the BMAC and BSI-2, these may not be available on future integrated versions. Relying on signals such as FCRCVD and INFORCVD may limit the user's ability to take advantage of these future versions.

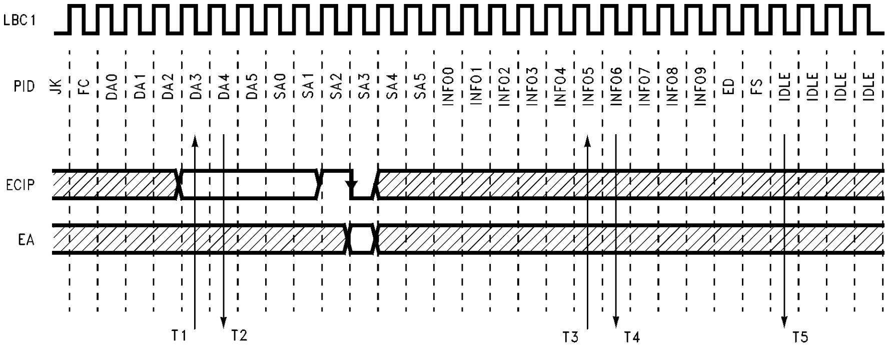
The proper use of the ECIP, EA, and EM pins is as follows. External address matching circuitry must assert ECIP somewhere from the arrival of the start delimiter (JK) to the 6th byte of the INFO field (as measured at the PLAYER-BMAC interface. Otherwise, the BSI-2 device assumes that no external address comparison is taking place. ECIP must be negated for at least one cycle to complete the external comparison. If it has not been deasserted by the 2nd byte after the End Delimiter (ED) the frame is not copied. EA and EM are sampled on the clock cycle after ECIP is negated. ECIP is ignored after it is negated until the arrival of the next JK.

Note that this design allows ECIP to be a positive or negative pulse. To confirm frames in this mode, (typically with Source Address Transparency enabled), EM must be asserted within the same timeframe as EA.

It is important to note that ECIP functions as an indicator to the internal BSI-2 Device Indicate Machine to hold off the copying of incoming data until the ECIP line is negated. Therefore, even if a design does not intend to take advantage of the BSI-2 Device's External Address Matching interface, the user must still ensure that the ECIP signal line is properly negated. Also important is the fact that the BSI-2 Device samples the ECIP signal line in order to detect just two conditions: it looks at whether ECIP is asserted at any time during the period between the start delimiter (JK) and the 6th byte of the INFO field; and it then waits until the deassertion of ECIP, at which point the BSI-2 Device samples the EA and EM signal lines for their status on this particular frame.

In the timing diagram, the specific cycles shown for the assertion and deassertion of ECIP comprise only one possible valid timing. Other timings are valid as well, within the limits of the timing parameters to be described below. Shaded areas indicate cycles where the BSI-2 Device is not sampling the signal lines for this particular pattern. Note that the sampling of ECIP is level sensitive, and synchronous with LBC1.

Note that there are five timing parameters, T1–T5. T1 and T3 are limits as to when the initial assertion of ECIP will be recognized. Once ECIP is asserted, T2, T4, and T5 become timing limits on the deassertion of ECIP. Once deasserted,



TL/F/11407-6

FIGURE 4-2. BSI-2 Device External Matching Interface Timing

ECIP is not sampled further (until the start of the next frame). T1–T5 are explained in greater detail below.

T1 is the earliest cycle where the assertion of ECIP will be recognized. ECIP may be asserted earlier than this, the BSI-2 Device simply will not sample it during these earlier periods. T1's timing is fixed as the fourth cycle following the FC data byte at the PLAYER-BMAC interface.

T2 is the earliest cycle where the deassertion of ECIP will be recognized. This can occur as soon as one cycle following ECIP's assertion. ECIP needs to be asserted for a minimum of one full clock cycle.

T3 is the latest cycle where the initial assertion of ECIP will still be recognized. T3 must occur before the 6th byte of the INFO field, not afterward. If ECIP is asserted later than this cycle, an external match will not be recognized; i.e., the frame will be copied only if it is an internal match, and will not be copied otherwise. When ECIP is not asserted until after T3, it is not recognized at all; i.e., it becomes the only case where maintaining ECIP's assertion well into the frame will have no effect at all.

T4 is the latest cycle where the deassertion of ECIP will be recognized in ‘‘regular’’ fashion. That is, if ECIP is held asserted beyond T4, a special case is created within the BSI-2 Device where the external compare has persisted to the point where it takes precedence over all other copy modes. In this case all frames which are copied, regardless of whether it was an external or an internal match, OR even an SMT frame, **are copied to ICHN2**. Note that even if an internal match has already occurred, ECIP must still become deasserted for the frame copy to complete.

It is important to note that the timing shown for T4 is dependent on the setting of the SMLB bit of the BSI-2 Device's Mode Register 0 (MR0). The timing shown is for frame copies in small-burst mode only. T4 signifies the boundary condition internal to the BSI-2 Device where the first full burst of data has been received. The ABus write access for this data will then automatically default to ICHN2 if an external copy decision is still pending, **regardless of sort mode**. When the SMLB bit is not set, i.e., in large burst mode, T4 would occur 16 cycles later than shown in the timing diagram.

T5 is the final cycle where the deassertion of ECIP can be recognized, and it occurs two cycles after the end delimiter is transferred between the PLAYER and BMAC. If ECIP is held high beyond this point, the frame will not be copied at all, **even if an internal match occurred**. Note that this is true even if Internal/External sorting is not enabled. Therefore, for applications which do not use external address matching, ECIP should be tied low. Note also that if ECIP remains asserted to the point where the incoming frame data completely fills the BSI-2 Device's Indicate Data FIFO (1024 bytes for a BSI-2 Device), then the frame will be dropped due to a FIFO overrun.

4.4 BUS INTERFACE UNIT

4.4.1 Overview

The ABus provides a 32-bit wide synchronous multiplexed address/data bus for transfers between the host system and the BSI-2 device. The ABus uses a bus request/bus grant protocol that allows multiple bus masters, supports burst transfers of 16 and 32 bytes, and supports virtual and physical addressing using fixed-size pages. The BSI-2 is capable of operating directly on the system bus to main memory, or connected to external shared memory.

All bus signals are synchronized to the master bus clock. The maximum burst speed is one 32-bit word per clock, but slower speeds may be accommodated by inserting wait states. The user may use separate clocks for the ring (FDDI MAC) and system (ABus) interfaces. The only restriction is that the ABus clock must be at least as fast as the ring clock (LBC). It is important to note that all ABus outputs change and all ABus inputs are sampled on the rising edge of AB__CLK.

The BSI-2 has two major modes of the ABus operation. The default, or ‘‘normal’’ mode, corresponds to the original BSI device and is completely backward compatible. The second mode is the Enhanced ABus mode. This mode is intended to reduce the logic required to interface to the SBus originally developed by Sun Microsystems, Inc. When the enhanced mode is selected, the BSI-2 device timing and interface signals are modified slightly to create a closer fit to the SBus. This lowers the cost and eases design of SBus FDDI adapter cards. This new mode is accessible by programming a mode bit in a register (MR1.EAM = 1). This bit is set to an inactive state upon reset to maintain backward compatibility with the original BSI device.

Addressing Modes

In the default ABus mode, the Bus Interface Unit has two Address Modes, as selected by the user: Physical Address Mode and Virtual Address Mode. In Physical Address Mode, the BSI-2 device emits the memory address and immediately begins transferring the data. In Virtual Address Mode, the BSI-2 device TRI-STATEs the Address/Data bus for two cycles between emitting the virtual address and starting to transfer the data. This allows virtual-to-physical address translation by an external MMU.

The BSI-2 has a new mode for controlling the ABus Address Strobe ($\overline{\text{AB\_\_AS}}$). In the default mode, $\overline{\text{AB\_\_AS}}$ is driven active during the address cycle and remains low throughout the access. In the new mode (MR1.ASM = 1), $\overline{\text{AB\_\_AS}}$ is driven active during the address cycle and driven inactive during the remainder of the access. This allows $\overline{\text{AB\_\_AS}}$ to be used directly as a clock enable to the address latching device which reduces the number of external components.

In Enhanced ABus Mode (EAM), the BSI-2 has fixed address timing which is similar to the Physical address timing described above. The SBus MMU does not require extra cycles for the address translation.

The BSI-2 device interfaces to byte-addressable memory, but always transfers information in words. The BSI-2 device uses a word width of 32 data bits plus 4 (1 per byte) parity bits. Parity may be ignored.

Bus Transfers

The bus supports several types of transactions, single word accesses and 4-word and 8-word bursts. Simple reads and writes involve a single address and data transfer. Burst reads and writes involve a single address transfer followed by multiple data transfers. Burst sizes are selected dynamically by the BSI-2. The user can disable 8-word bursts by setting MR0.SMLB to a 1. This forces the BSI-2 to use 1-word and 4-word transactions only.

4.0 Functional Description (Continued)

A new option for burst mode addressing is available. The ABus provides the demultiplexed low order address bits (AB__A[4:2]) during burst accesses. These demultiplexed addresses are generated a cycle before the associated data transfer to allow pipelining and to give greater address set-up time. The BSI-2 device provides a mode bit (MR1.ATM) which causes the burst addresses to be generated in the same cycle as the associated data. In systems which do not require the additional speed of the look-ahead addresses, this saves an address latch. As with all the new mode bits, this bit is set to an inactive state upon reset which maintains both hardware and software compatibility with the original BSI device.

On Indicate Channels, when 8-word bursts are enabled, all transactions will be 8 words until the end of the frame; the last transfer will be 4 or 8 words, depending on the number of remaining bytes. If only 4-word bursts are allowed, all Indicate Data transfers are 4 words.

On Request Channels, the BSI-2 will use 4- or 8-word bursts to access all data up to the end of the ODU. If 8-word bursts are enabled, the first access will be an 8-word burst if the ODU begins less than 4 words from the start of an 8-word burst boundary. If 8-word bursts are not allowed, or if the ODU begins 4 or more words from the start of an 8-word burst boundary, a 4-word burst will be used. The BSI-2 will ignore unused bytes if the ODU does not start on a burst boundary. At the end of an ODU, the BSI-2 will use the smallest transfer size (1, 4, or 8 words) which completes the ODU read. To coexist in a system that assumes implicit wrap-around for the addresses within a burst, the BSI-2 device never emits a burst that will wrap the 4- or 8-word boundary.

A Function Code identifying the type of transaction is output by the BSI-2 device on the upper four address bits during the address phase of a data transfer. This can be used for more elaborate external addressing schemes, for example, to direct control information to one memory and data to another (e.g., an external FIFO). Note that the function code is not available when using the enhanced ABus mode for SBus.

Byte Ordering

The basic addressable quantum is a byte, so request data may be aligned to any byte boundary in memory. All information is accessed in 32-bit words, however, so the BSI-2 device ignores unused bytes when reading.

Descriptors must always be aligned to a 32-bit word address boundary. Input Data Units are always aligned to a burst-size boundary. Output Data Units may be any number of bytes, aligned to any byte-address boundary, but operate most efficiently when aligned to a burst-size boundary. Pool Space Descriptors (PSPs) **must** point to a burst boundary or the Receive data will not be written to memory correctly.

Burst transfers are always word-aligned on a 16- or 32-byte (burst-size) address boundary. Burst transfers will never cross a burst-size boundary. If a 32-byte transfer size is enabled (MR0.SMLB = 0), the BSI-2 device will perform both 16-byte and 32-byte bursts, whichever is most efficient (least number of clocks to load/store all required data). If the BSI-2 has less than a full burst of data to complete a frame, it will write a full burst. Random data is written to the unused locations. The host uses the IDUD length field to determine where the valid data bytes end.

The Bus Interface Unit can operate in either Big Endian or Little Endian Mode. The bit and byte alignments for both modes are shown in *Figure 4-3.* Byte 0 is the first byte received from the ring or transmitted to the ring. This mode affects the placement of frame data bytes but it does not affect the order of Descriptor bytes.

Big-Endian Byte Order

D[31]			D[0]
Word			
Halfword 0		Halfword 1	
Byte 0	Byte 1	Byte 2	Byte 3

Little-Endian Byte Order

D[31]			D[0]
Word			
Halfword 1		Halfword 0	
Byte 3	Byte 2	Byte 1	Byte 0

FIGURE 4-3. ABus Byte Orders

Bus Arbitration

The ABus is a multi-master bus, using a simple Bus Request/Bus Grant protocol that allows an external Bus Arbiter to support any number of bus masters, using any arbitration scheme (e.g., rotating or fixed priority). The protocol provides for multiple transactions per tenure, and bus master preemption.

The BSI-2 device asserts a Bus Request, and assumes mastership when Bus Grant is asserted. If the BSI-2 device has another transaction pending, it will keep Bus Request asserted, or reassert it before the completion of the current transaction. Note that Bus Request is guaranteed to be deasserted for at least two cycles when MR1.EAM is enabled. It is a requirement of the SBus specification that Bus Request be deasserted between each transaction. If Bus Grant is (re)asserted before the end of the current transaction, the BSI-2 device retains mastership and runs the next transaction. This process may be repeated indefinitely.

It is important to note that in default ABus mode (MR1.EAM = 0), the BSI-2 may take up to two cycles to detect the assertion of Bus Grant. Therefore, the external interface logic must not remove Bus Grant or latch addresses until a signal such as Address Strobe is asserted indicating that the BSI-2 has recognized the Bus Grant.

If the Bus Arbiter wishes to preempt the BSI-2 device, it deasserts Bus Grant. The BSI-2 device will complete the current bus transaction, then release the bus. From preemption to bus release is a maximum of (11 bus clocks + (8 times the number of memory wait states)) bus clocks. For example, in a 1 wait-state system, the BSI-2 device will release the bus within a maximum of 19 bus clocks.

Parity

The state of the FLOW bit in Mode Register 0 (MR0.FLOW) controls two BSI-2 modes: one for systems using parity, the other for systems not using parity.

Regardless of the state of MR0.FLOW, the BSI-2 always provides odd parity on ABus address cycles, and Control Bus read cycles. Parity is never checked by the BSI-2 on

ABus read cycles. The ABus data parity, (except for Descriptor data parity which is generated internally) flows directly from the BMAC interface. If parity checking is enabled within the BMAC, parity flows up from the PLAYER interface. If parity checking is disabled within the BMAC, good parity is generated at the BMAC/BSI-2 interface. Note that this implies that the data parity line between the BMAC and BSI-2 must be connected to have good data parity at the ABus interface.

For transmit frame data, the parity from the ABus flows directly to the BMAC interface when MR0.FLOW is asserted, (hence the name). The data integrity across the ABus and through the BSI-2 may be checked by enabling parity checking within the BMAC device. If the MR0.FLOW bit is deasserted, the BSI-2 will generate good parity at the transmit data (MR7–0) BMAC interface.

When MR0.FLOW is enabled, parity is checked on Control Bus write operations. If a parity error is detected, the write access is rejected and the STAR.CPE bit is asserted. When MR0.FLOW is disabled Control Bus write parity is ignored.

When MR0.FLOW is enabled, parity is checked on MAC Indicate interface (receive data from the BMAC). If an error is detected, the STAR.BPE will be asserted and the IDUD for that frame will carry a status of ‘‘parity error’’. When MR0.FLOW is disabled, the parity bit at the MAC Indicate interface is ignored.

Bandwidth

The ABus supports single reads and writes, and burst reads and writes. With physical addressing, back-to-back single reads/writes can take place every four clock cycles. Burst transactions can transfer 8, 32-bit words (32 bytes) every 11 clock cycles. With a 33 MHz clock this yields a peak bandwidth of 96 Mbytes/sec.

To allow the bus to operate at high frequency, the protocol defines all signals to be both asserted and deasserted by the bus master and slaves. Having a bus device actively deassert a signal guarantees a high-speed inactive transition. If this were not defined, external pull-up resistors would not be able to deassert signals fast enough. The protocol also reduces contention by avoiding cases where two bus devices simultaneously drive the same line.

The BSI-2 device operates synchronously with the ABus clock. In general, operations will be asynchronous to the ring, since most applications will use a system bus clock that is asynchronous to the ring. The BSI-2 device is designed to interface either directly to the host's main system bus or to external shared memory. When interfaced to the host's bus, there are two parameters of critical interest: latency and bandwidth.

Data moves between the Request and Indicate Channels and the ABus via four FIFOs, two in the receive path (Indicate) and two in the transmit path (Request). On the BMAC Device Interface, there are two, 256 x 36-bit data FIFOs for Indicate and Request data (1 kbyte of data FIFO in each direction). On the ABus Interface, there are two Burst FIFOs, each containing two banks of 32 bytes, which provide ABus bursting capability.

The amount of latency covered by the Data FIFO plus one of the banks of the Burst FIFO must meet the average and maximum bus latencies of the external memory. With a new byte every 80 ns from the ring, a 1 kbyte FIFO provides 1024×80 ns $= 81.9$ μs maximum latency. The two 32-byte burst FIFO banks provide an additional 5.1 μs of latency.

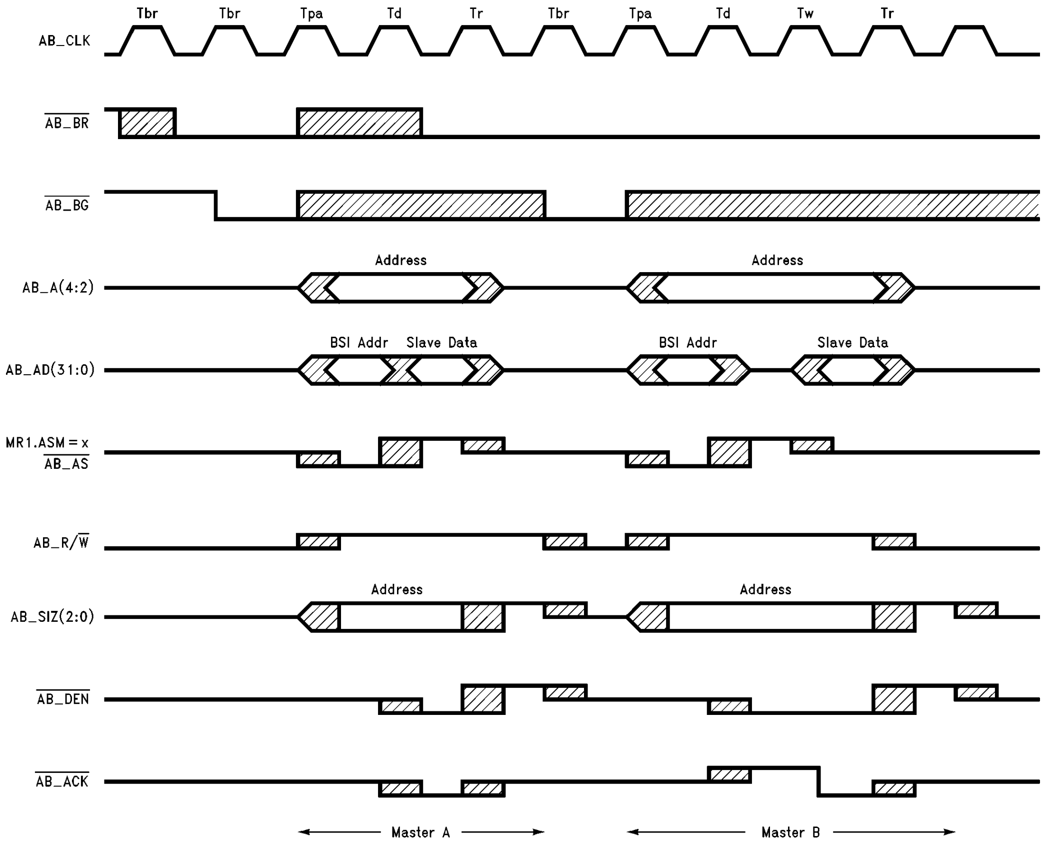
To assist latency issues, the BSI-2 device can completely empty or fill the Burst FIFO in one bus tenure by asserting Bus Request for multiple transactions. Since one bank of the Burst FIFO is 8 words deep, if 8-word bursts are enabled, that half of the Burst FIFO can be emptied in one transaction. If the second half of the burst FIFO is also full, it can be emptied in the same bus tenure by again granting the bus to the BSI-2 device.

The BSI-2 device may be preempted at any time by removing Bus Grant, which causes the BSI-2 device to complete the current transaction and release the bus. There will be a maximum of 11 clocks (plus any memory wait states) from preemption to bus release (fewer if 8-word bursts are not enabled).

4.4.2 Bus States

An ABus Master has eight states: idle (Ti), bus request (Tbr), virtual address (Tva), MMU translate (Tmmu), physical address (Tpa), data transfer (Td), wait (Tw) and recovery (Tr).

An ABus Slave has five states: idle (Ti), selected (Ts), data transfer (Td), wait (Tw), and recovery (Tr).


Tbr
Tbr
Tpa
Td
Tr
Tbr
Tpa
Td
Tw
Tr
AB_CLK
$\overline{\text{AB\_BR}}$
$\overline{\text{AB\_BG}}$
Address
Address
AB_A(4:2)
BSI Addr
Slave Data
BSI Addr
Slave Data
AB_AD(31:0)
MR1.ASM = x
$\overline{\text{AB\_AS}}$
AB_R/$\overline{\text{W}}$
Address
Address
AB_SIZ(2:0)
$\overline{\text{AB\_DEN}}$
$\overline{\text{AB\_ACK}}$
Master A
Master B

TL/F/11407–7

FIGURE 4-4. ABus Single Read, Physical Addressing, 0 w/s, 1 w/s

Master States

The Ti state exists when no bus activity is required. The BIU does not drive any of the bus signals when it is in this state (all are released). If the BIU requires bus service, it may assert Bus Request.

When a transaction is run, the BIU enters Tbr and asserts Bus Request, then waits for Bus Grant to be asserted.

The state following Tbr is either Tva or Tpa. In Virtual Address Mode, the BIU enters Tva and drives the virtual address and size lines onto the bus. In Physical Address Mode, Tpa occurs next (see Section 4.4.3).

Following a Tva state is a Tmmu state. During this cycle the external MMU is performing a translation of the virtual address emitted during Tva.

Following a Tmmu state (when using virtual addressing) or a Tbr state (when using physical addressing), is the Tpa state. During the Tpa state, the BSI-2 device drives the read/write strobes and size signals. In physical address mode, it also drives AB__AD with address. In virtual address mode, the BSI-2 device TRI-STATEs AB__BD so the host CPU or MMU can drive the address.

Following the Tpa state, the BIU enters the Td state to transfer data words. Each data transfer may be extended indefinitely by inserting Tw states. A slave acknowledges by asserting $\overline{\text{AB\_\_ACK}}$ and transferring data in a Td state (cycle). If the slave can drive data at the rate of one word per clock (in a burst), it keeps $\overline{\text{AB\_\_ACK}}$ asserted.

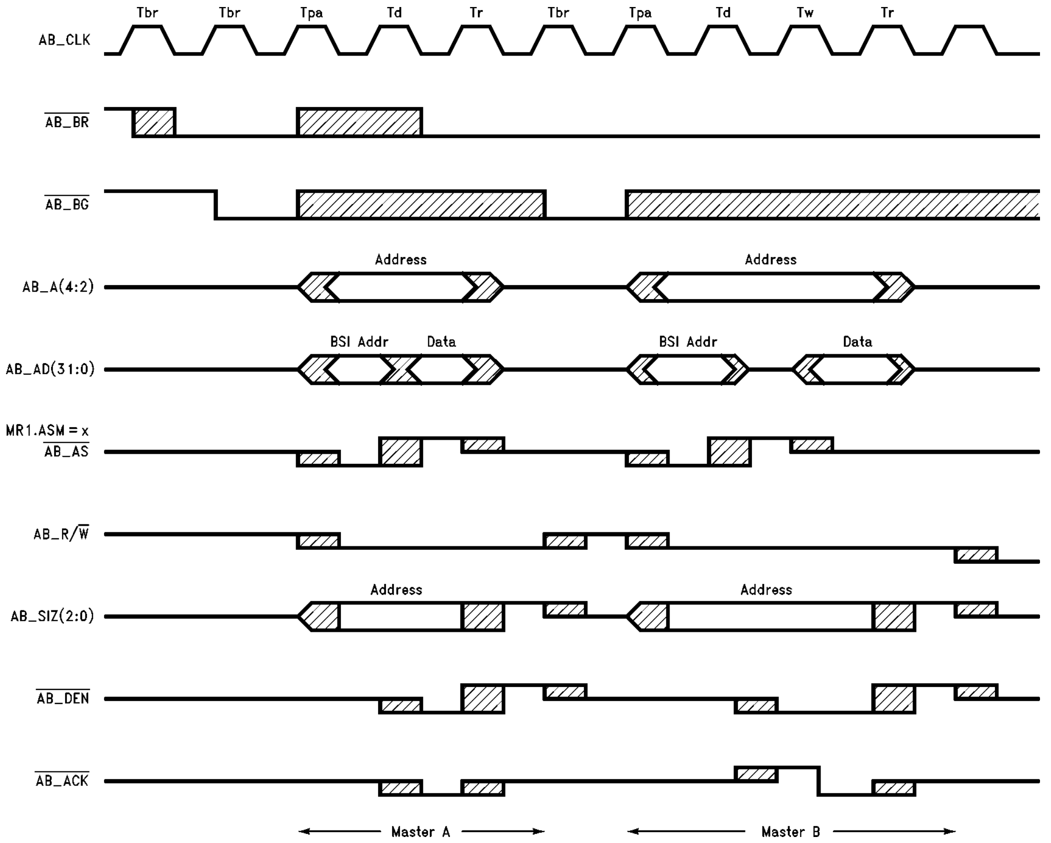
Following the final Td/Tw state, the BIU enters a Tr state to allow time to turn off or turn around bus transceivers.

A bus retry request is recognized in any Td/Tw state. The BIU will go to a Tr state and then rerun the transaction when it obtains a new Bus Grant. The whole transaction is retried, i.e., all words of a burst. Additionally, no other transaction will be attempted before the interrupted one is retried. The BIU retries indefinitely until either the transaction completes successfully, or a bus error is signaled.

Bus errors are recognized in Td/Tw states.

4.4.3 Physical Addressing Bus Transactions

Bus transactions in Physical Address Mode are shown in *Figure 4-4* through *Figure 4-7.* BSI-2 device signals are defined in Section 6.



TL/F/11407-8

FIGURE 4-5. ABus Single Write, Physical Addressing, 0 w/s, 1 w/s

Single Read

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG. The BSI-2 moves to Tpa within the next three clocks.

Tpa: BSI-2 device drives AB__A and AB__AD with the address, asserts AB__AS, drives AB__R/W and AB__SIZ[2:0], negates AB__BR if another transaction is not required.

Td: BSI-2 device negates AB__AS, asserts AB__DEN, samples AB__ACK and AB__ERR. Slave asserts AB__ACK, drives AB__ERR, drives AB__AD (with data) when ready. The BSI-2 device samples a valid AB__ACK, capturing the read data. Tw states may occur after Td.

Tr: BSI-2 device negates AB__R/W, AB__DEN, AB__SIZ[2:0], releases AB__A, and AB__AS. Slave deasserts AB__ACK and AB__ERR, releases AB__AD.

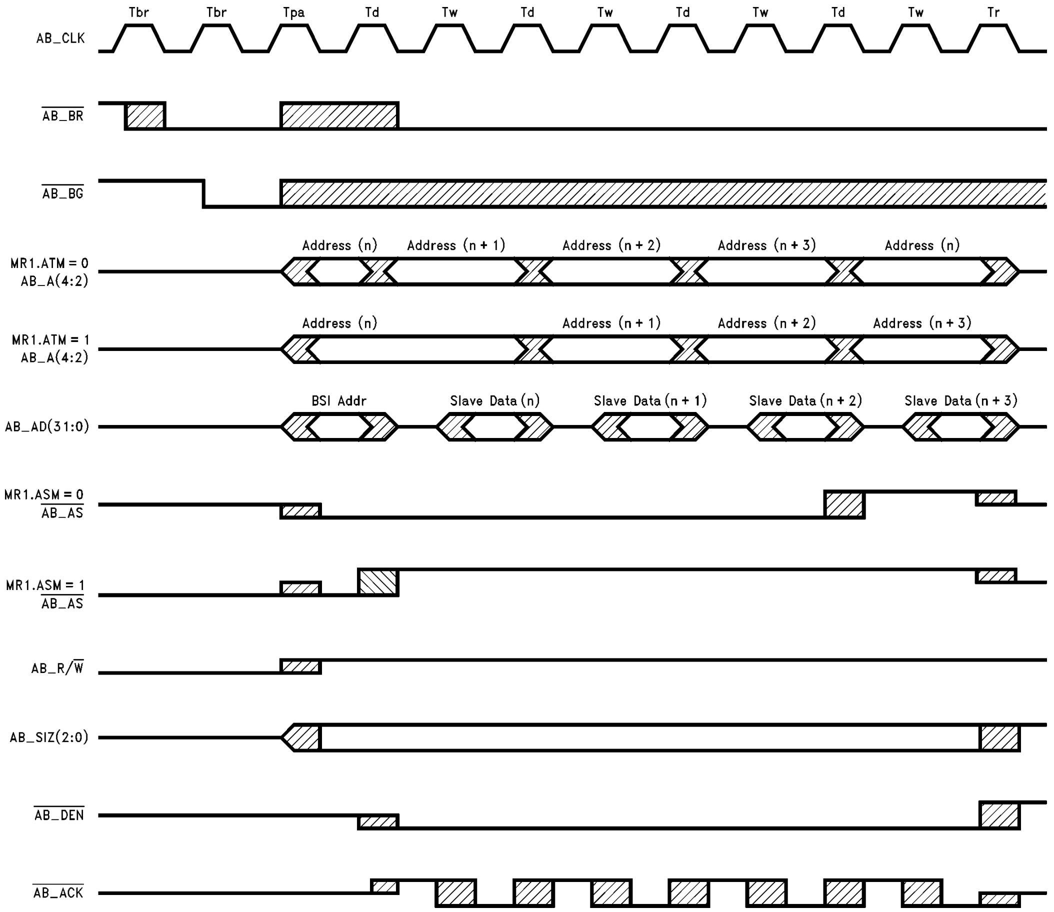
Single Write

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG. The BSI-2 moves to Tpa within the next three clocks.

Tpa: BSI-2 device drives AB__A and AB__AD with the address, asserts AB__AS, drives AB__R/W and AB__SIZ[2:0], and negates AB__BR if another transaction is not required.

Td: BSI-2 device negates AB__AS, asserts AB__DEN, drives AB__AD with the write data and starts sampling AB__ACK and AB__ERR. Slave captures AB__AD data, asserts AB__ACK, drives AB__ERR. Tw states may occur after Td if the slave deasserts AB__ACK.

Tr: BSI-2 device negates AB__R/W, AB__DEN, AB__SIZ[2:0], releases AB__A, AB__AD, AB__AS. Slave deasserts AB__ACK and AB__ERR, and stops driving AB__AD with data.



TL/F/11407–9

FIGURE 4-6. ABus Burst Read, Physical Addressing, 16 bytes, 1 w/s

Burst Read

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG. The BSI-2 moves to Tpa within the next three clocks.

Tpa: BSI-2 device drives AB__A and AB__AD with the address, asserts AB__AS, drives AB__R/W and AB__SIZ[2:0], and negates AB__BR if another transaction is not required.

Td: BSI-2 device asserts AB__DEN, samples AB__ACK and AB__ERR, increments the address on AB__A. Slave asserts AB__ACK, drives AB__ERR, and drives AB__AD (with data) when ready. BSI-2 device samples a valid AB__ACK, capturing the read data. Tw states may occur after Td. Td state is repeated four or eight times (according to the burst size). If MR1.ASM = 0, the BSI-2 device negates AB__AS in the last Td cycle. If MR1.ASM = 1, the BSI-2 will negate AB__AS in the first Td cycle.

Tr: BSI-2 device negates AB__R/W, AB__DEN, AB__SIZ[2:0], and releases AB__A and AB__AS. Slave deasserts AB__ACK and AB__ERR, and releases AB__AD.

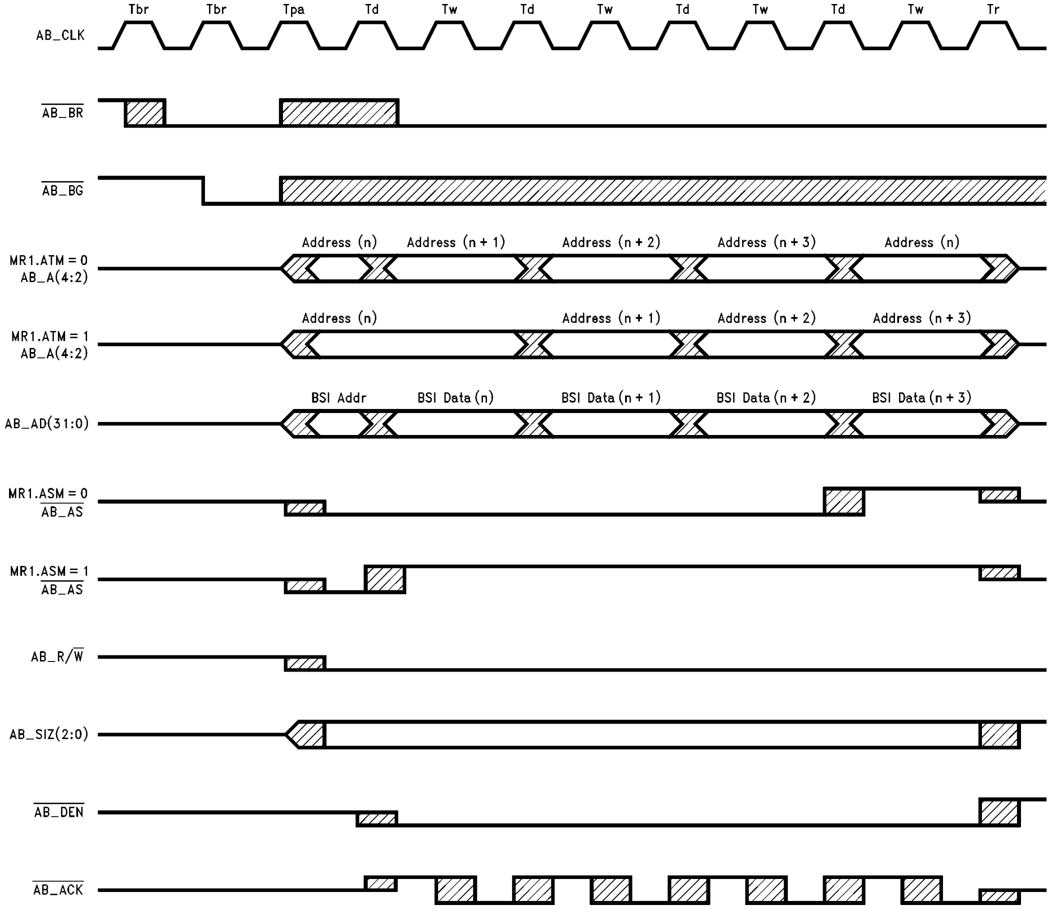
Burst Write

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG. The BSI-2 moves to Tpa within the next three clocks.

Tpa: BSI-2 device drives AB__A and AB__AD with the address, asserts AB__AS, drives AB__R/W and AB__SIZ[2:0], and negates AB__BR if another transaction is not required.

Td: BSI-2 device asserts AB__DEN, drives AB__AD with the write data, samples AB__ACK and AB__ERR, increments the address on AB__A. Slave captures AB__AD data, asserts AB__ACK, drives AB__ERR. BSI-2 device samples a valid AB__ACK. Tw states may occur after Td. Td state is repeated as required for the complete burst. If MR1.ASM = 0, the BSI-2 device negates AB__AS in the last Td cycle. If MR1.ASM = 1, the BSI-2 will negate AB__AS in the first Td cycle.

Tr: BSI-2 device negates AB__R/W, AB__DEN, AB__SIZ[2:0], releases AB__A and AB__AS, and stops driving AB__AD with data. Slave deasserts AB__ACK and AB__ERR.



TL/F/11407–10

FIGURE 4-7. ABus Burst Write, Physical Addressing, 16 bytes, 1w/s

4.4.4 Virtual Addressing Bus Transactions

Single Read

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG, and BSI-2 device drives AB__A and AB__AD when AB__BG is asserted. Moves to Tva within the next three clocks.

Tva: BSI-2 device drives AB__A and AB__AD with the virtual address for one clock, negates AB__AS, asserts AB__R/W, drives AB__SIZ[2:0], and negates AB__BR if another transaction is not required.

Tmmu: Host MMU performs an address translation during this clock.

Tpa: Host MMU drives AB__AD with the translated (physical) address, BSI-2 device drives AB__A and asserts AB__AS.

Td: BSI-2 device negates AB__AS, asserts AB__DEN, samples AB__ACK and AB__ERR. Slave asserts AB__ACK, drives AB__ERR, drives AB__AD (with data) when ready. BSI-2 device samples a valid AB__ACK, capturing the read data. Tw states may occur after Td.

Tr: BSI-2 device negates AB__R/W, AB__DEN, and AB__SIZ[2:0], releases AB__A and AB__AS. Slave deasserts AB__ACK and AB__ERR and releases AB__AD.

Single Write

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG, BSI-2 device drives AB__A and AB__AD when AB__BG is asserted. Moves to Tva within the next three clocks.

Tva: BSI-2 device drives AB__A and AB__AD with the virtual address for one clock, negates AB__AS, negates AB__R/W, and drives AB__SIZ[2:0].

Tmmu: Host MMU performs an address translation during this clock.

Tpa: Host MMU drives AB__AD with the address, BSI-2 device drives AB__A asserts AB__AS, and negates AB__BR if another transaction is not required.

Td: BSI-2 device negates AB__AS, asserts AB__DEN, drives AB__AD with the write data and starts sampling AB__ACK and AB__ERR. Slave captures AB__AD data, asserts AB__ACK, and drives AB__ERR. BSI-2 device samples a valid AB__ACK. Tw states may occur after Td.

Tr: BSI-2 device negates AB__R/W, AB__DEN, AB__SIZ[2:0], releases AB__A, AB__AD, and AB__AS. Slave deasserts AB__ACK and AB__ERR, and stops driving AB__AD with data.

Burst Read

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG, BSI-2 drives AB__A and AB__AD when AB__BG is asserted. Moves to Tva within the next three clocks.

Tva: BSI-2 device drives AB__A and AB__AD with the virtual address for one clock, negates AB__AS, asserts AB__R/W, drives AB__SIZ[2:0], and negates AB__BR if another transaction is not required.

Tmmu: Host MMU performs an address translation during this clock.

Tpa: Host MMU drives AB__AD with the translated (physical) address. BSI-2 device drives AB__A and asserts AB__AS.

Td: BSI-2 device asserts AB__DEN, samples AB__ACK and AB__ERR. Slave asserts AB__ACK, drives AB__ERR, drives AB__AD (with data) when ready. BSI-2 device samples a valid AB__ACK, capturing the read data. Tw states may occur after Td. This state is repeated four or eight times (according to burst size). If MR1.ASM=0 the BSI-2 device negates AB__AS in the last Td cycle. If MR1.ASM=1, the BSI-2 will negate AB__AS in the first Td cycle.

Tr: BSI-2 device negates AB__R/W, AB__DEN, AB__SIZ[2:0], releases AB__A and AB__AS. Slave deasserts AB__ACK and AB__ERR, and releases AB__AD.

Burst Write

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG, BSI-2 device drives AB__A and AB__AD when AB__BG is asserted. Moves to Tva within the next three clocks.

Tva: BSI-2 device drives AB__A and AB__AD with the virtual address for one clock, negates AB__AS, negates AB__R/W, drives AB__SIZ[2:0].

Tmmu: Host MMU performs an address translation during this clock.

Tpa: Host MMU drives AB__AD with the address, BSI-2 device drives AB__A asserts AB__AS, and negates AB__BR if another transaction is not required.

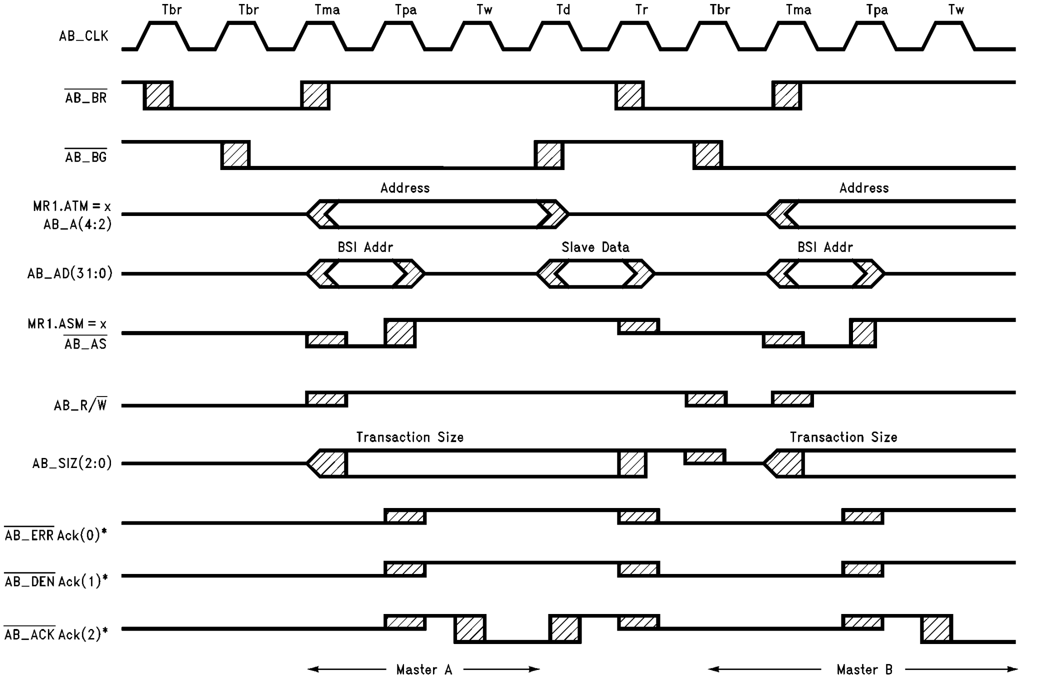
Td: BSI-2 device asserts AB__DEN, drives AB__AD with the write data and starts sampling AB__ACK and AB__ERR. Slave captures AB__AD data, asserts AB__ACK and drives AB__ERR. BSI-2 device samples a valid AB__ACK. Tw states may occur after Td. This state is repeated as required for the complete burst. If MR1.ASM=0, the BSI-2 device negates AB__AS in the last Td cycle. If MR1.ASM=1, the BSI-2 will negate AB__AS in the first Td cycle.

Tr: BSI-2 device negates AB__R/W, AB__DEN, AB__SIZ[2:0], releases AB__A, AB__AD, AB__AS, stops driving AB__AD with data. Slave deasserts AB__ACK and AB__ERR.

4.5 ENHANCED ABUS MODE

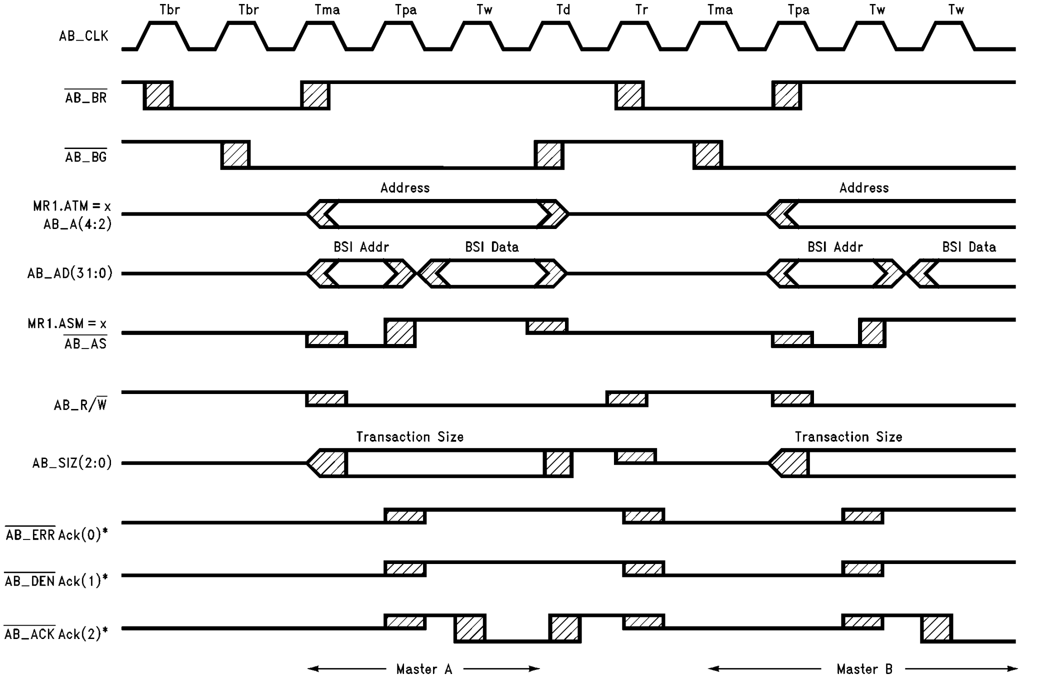
When this enhanced mode is selected, the following changes occur. The timing of AB__ACK is modified during read accesses. In this mode, read data is expected one cycle after the AB__ACK signal (see *Figure 4-8* and *Figure 4-11*). Also in this mode, channel information is no longer supplied on the upper four bits of the Address/Data lines during the address cycle. Instead, the value of this nibble of address is supplied from a programmable register within the BSI-2 Device (for a full description of these bits please see Mode Register1 (MR1)). Finally, the AB__DEN signal becomes an input in this enhanced mode. This signal along with AB__ACK and AB__ERR are used to encode a subset of the acknowledge, retry, and error functions supported on the SBus.

These enhancements make it easier to connect the BSI-2 to the SBus as a bus master. However, a full FDDI adapter design requires the design of a slave interface from the SBus to the control bus of the BSI-2 and the other FDDI components.



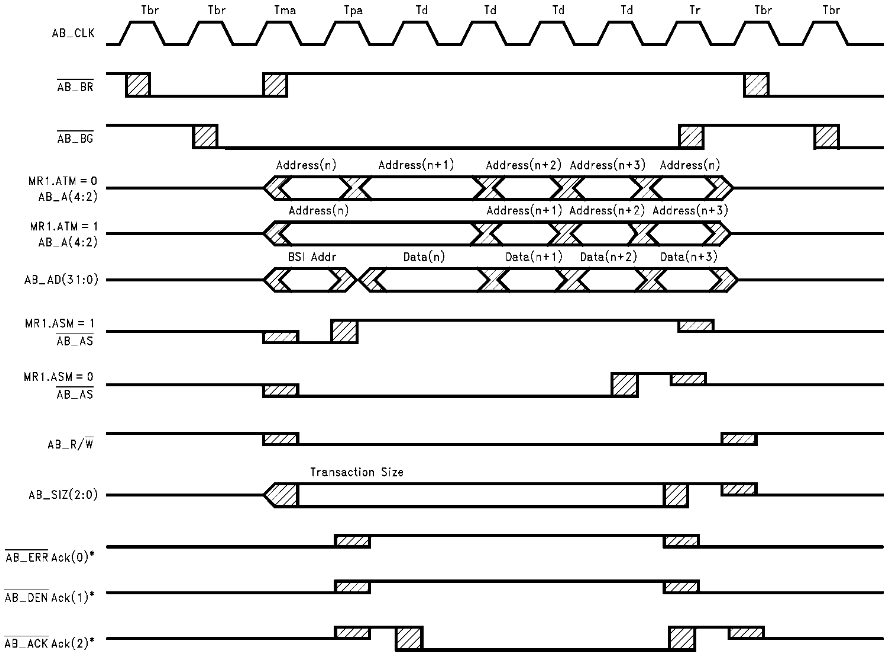
TL/F/11407–11

FIGURE 4-8. Enhanced ABus Read Timing



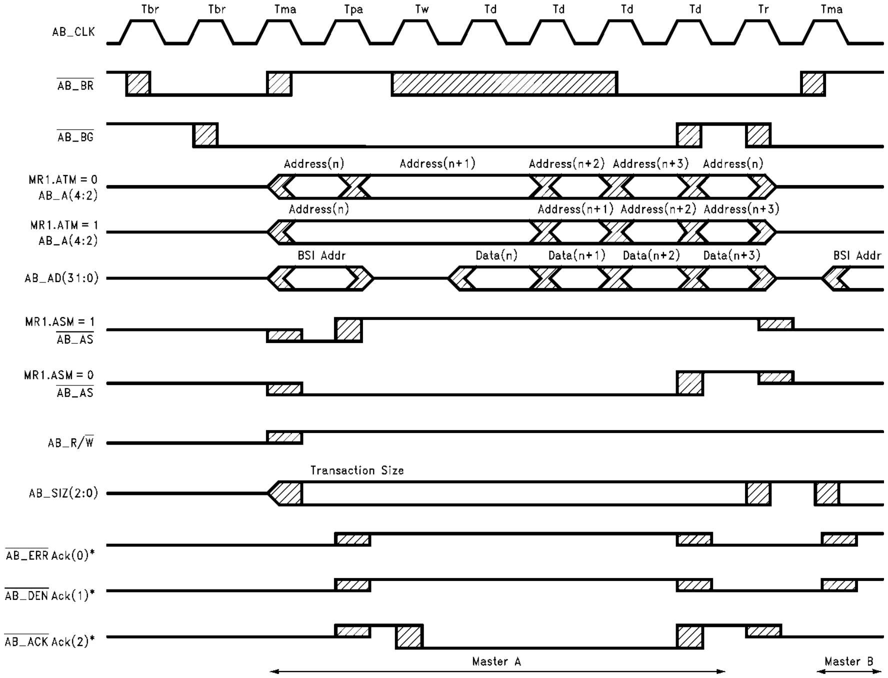
TL/F/11407–12

FIGURE 4-9. Enhanced ABus Mode Write Timing



TL/F/11407–13

FIGURE 4-10. Enhanced ABus Mode Burst Write Timing



TL/F/11407–14

FIGURE 4-11. Enhanced ABus Mode Back-to-Back Read Timing

4.5.1 Enhanced ABus Mode Bus Transactions

Bus transactions in the Enhanced Abus Mode are shown in *Figure 4-8* through *Figure 4-11.* In the Enhanced ABus mode, the Bus Request signal (AB__BR) will be deasserted after the bus is granted until the completion of the bus transaction. The only exception to this may occur when the BSI-2 is attempting back-to-back burst reads. In this case AB__BR may be deasserted for a few as two cycles.

Single Read

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG. The BSI-2 moves to Tma on the next cycle.

Tma: BSI-2 device drives AB__A and AB__AD with the master address, asserts AB__AS, drives AB__R/W and AB__SIZ[2:0], negates AB__BR until the end of this transaction.

Tpa: The Physical address is asserted by the MMU.

Td: BSI-2 device negates AB__AS, samples AB__ACK, AB__ERR, and AB__DEN. Slave asserts AB__ACK, AB__ERR, and AB__DEN with the appropriate acknowledgment. The BSI-2 device samples a valid acknowledgment and moves to Tr. Tw states may occur after Td.

Tr: BSI-2 device negates AB__R/W, AB__DEN, AB__SIZ[2:0], releases AB__A, and AB__AS, and samples AB__AD. Slave drives AB__AD (with data) deasserts AB__ACK, AB__ERR, and AB__DEN, releases AB__AD.

Single Write

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG. The BSI-2 moves to Tma in the next cycle.

Tma: BSI-2 device drives AB__A and AB__AD with the master address, asserts AB__AS, drives AB__R/W and AB__SIZ[2:0], and negates AB__BR until the end of this transaction.

Tpa: The Physical address is asserted by the MMU.

Td: BSI-2 device negates AB__AS, drives AB__AD with the write data and starts sampling AB__ACK, AB__ERR, and AB__DEN. Slave captures AB__AD data, acknowledges with AB__ACK, AB__ERR, and AB__DEN. Tw states may occur after Td if the slave does not acknowledge.

Tr: BSI-2 device negates AB__R/W, AB__SIZ[2:0], releases AB__A, AB__AD, AB__AS, and stops driving AB__AD with data. Slave deasserts AB__ACK, AB__ERR, and AB__DEN.

Burst Read

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG. The BSI-2 moves to Tma in the next cycle. This cycle may be skipped if AB__BR was asserted during the previous access. This allows for back-to-back burst reads.

Tma: BSI-2 device drives AB__A and AB__AD with the master address, asserts AB__AS, drives AB__R/W and AB__SIZ[2:0], and negates AB__BR for at least two cycles.

Tpa: The Physical Address is asserted by the MMU.

Td: BSI-2 device samples AB__ACK, AB__ERR, and AB__DEN, increments the address on AB__A. Slave acknowledges using AB__ACK, AB__ERR, and AD__DEN. BSI-2 device samples a valid AB__ACK and latches data in the *following* cycle. Tw states may occur after Td. Td state is repeated four or eight times (according to the burst size). If MR1.ASM = 0, the BSI-2 device negates AB__AS in the last Td cycle. If MR1.ASM = 1, the BSI-2 will negate AB__AS in the first Td cycle.

Tr: BSI-2 device negates AB__R/W, AB__SIZ[2:0], and releases AB__A and AB__AS. Slave drives AB__AD (with data), deasserts AB__ACK, and AB__ERR, AB__DEN. If another request is pending (AB__BR asserted) and the Bus is regranted in this cycle, the BSI-2 will proceed directly to the Tpa state of the next burst. The normal Tbr state is skipped allowing back-to-back burst reads.

Burst Write

Tbr: BSI-2 device asserts AB__BR to indicate it wishes to perform a transfer. Host asserts AB__BG. The BSI-2 moves to Tma in the next cycle.

Tma: BSI-2 device drives AB__A and AB__AD with the master address, asserts AB__AS, drives AB__R/W and AB__SIZ[2:0], and negates AB__BR until this transaction is completed.

Tpa: The Physical Address is asserted by the MMU.

Td: BSI-2 device drives AB__AD with the write data, samples AB__ACK, AB__ERR, and AB__DEN, increments the address on AB__A. Slave captures AB__AD data, acknowledges using AB__ACK, AB__ERR, and AB__DEN. BSI-2 device samples a valid acknowledge. Tw states may occur after Td. Td state is repeated as required for the complete burst. If MR1.ASM = 0, the BSI-2 device negates AB__AS in the last Td cycle. If MR1.ASM = 1, the BSI-2 will negate AB__AS in the first Td cycle.

Tr: BSI-2 device negates AB__R/W, AB__SIZ[2:0], releases AB__A and AB__AS, and stops driving AB__AD with data. Slave deasserts AB__ACK, AB__ERR, AB__DEN.

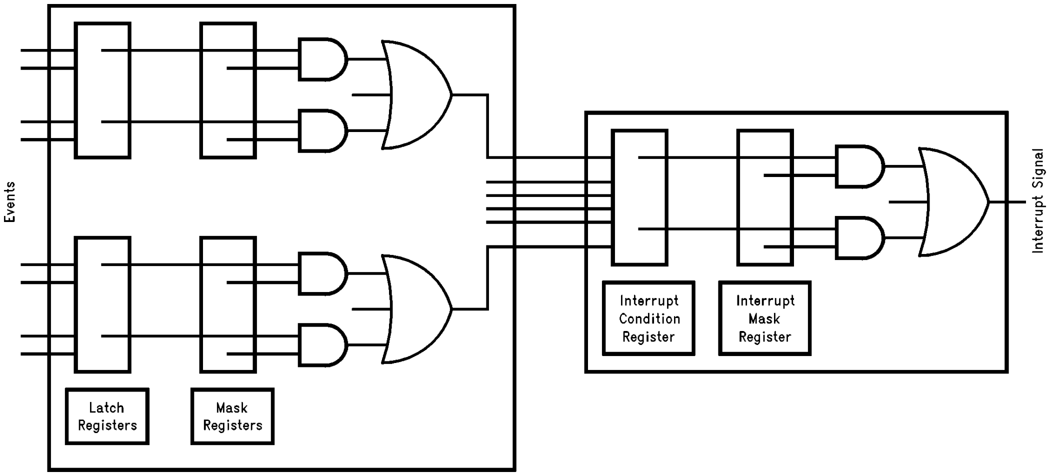
5.0 Control Information

5.1 OVERVIEW

Control information consists of registers and data structures that are used to manage and operate the BSI-2 device.

Control information is divided into four basic groups: Operation Registers, Pointer RAM Registers, Limit RAM Registers, and Descriptors.

Operation registers are accessed directly via the Control Bus. Limit RAM Registers are accessed indirectly via the Control Bus, using the Limit RAM Data and Limit RAM Address Registers. The Pointer RAM Registers are accessed indirectly via the Control Bus and ABus using the Pointer RAM Address and Control Register, the Mailbox Address Register, and a mailbox location in ABus memory.



TL/F/11407–15

FIGURE 5-1. Event Registers Hierarchy

5.2 OPERATION REGISTERS

The Operation Registers are divided into two functional groups: Control Registers and Event Registers. They are summarized in *Figure 5-2.*

Control Registers

The Control Registers are used to configure and control the operation of the BSI-2 device.

The Control Registers include the following registers:

- **Mode Register (MR)** establishes major operating parameters for the BSI-2 device.
- **Pointer RAM Control and Address Register (PCAR)** is used to program the parameters for the PTOP (Pointer RAM Operation) service function.
- **Mailbox Address Register (MBAR)** is used to program the memory address of the mailbox used in the data transfer of the PTOP service function.
- **Limit Address Register (LAR)** is used to program the parameters and data used in the LMOP (Limit RAM Operation) service function.
- **Limit Data Register (LDR)** is used to program the data used in the LMOP service function.
- **Request Channel 0 Configuration Register (R0CR)** is used to program the operational parameters for Request Channel 0.
- **Request Channel 1 Configuration Register (R1CR)** is used to program the operational parameters for Request Channel 1.
- **Request Channel 0 Expected Frame Status Register (R0EFSR)** defines the expected frame status for frames being confirmed on Request Channel 0.
- **Request Channel 1 Expected Frame Status Register (R1EFSR)** defines the expected frame status for frames being confirmed on Request Channel 1.
- **Indicate Threshold Register (ITR)** is used to specify a maximum number of frames that can be copied onto an Indicate Channel before a breakpoint is generated.
- **Indicate Mode Register (IMR)** specifies how the incoming frames are sorted onto Indicate Channels, enables frame filtering, and enables breakpoints on various burst boundaries.
- **Indicate Configuration Register (ICR)** is used to program the copy criteria for each of the Indicate Channels.
- **Indicate Header Length Register (IHLR)** defines the length of the frame header for use with the Header/Info Sort Mode.

Event Registers

The Event Registers record the occurrence of events or series of events. Events are recorded and contribute to generating the Interrupt signal. There is a two-level hierarchy in generating this signal, as shown in *Figure 5-1.*

At the first level of the hierarchy, events are recorded as bits in the Attention Registers (e.g., No Space Attention Register). Each Attention Register has a corresponding Notify Register (e.g., No Space Notify Register). When a bit in the Attention Register is set to One and its corresponding bit in the Notify Register is also set to One, the corresponding bit in the Master Attention Register will be set to one.

5.0 Control Information (Continued)

At the second level of the hierarchy, if a bit in the Master Attention Register is set to One and the corresponding bit in the Master Notify Register is set to One, the Interrupt signal is asserted.

To ensure that events are not missed when updating the attention registers, all attention registers are conditional write registers. Bits in Conditional Write Registers (e.g., No Space Attention Register) are only written when the corresponding bits in the Compare Register are equal to the bits to be overwritten. Read operations for conditional write registers automatically cause the Compare Register to be loaded with the contents of the conditional write register being accessed.

Events are recorded in Attention Registers and contribute to the Interrupt when the bit in the corresponding Notify Register is set (see *Figure 5-1*). Bits in the Master Attention Register (MAR) are not cleared directly. They are cleared by clearing the lower level attention and/or notify register.

The Event Registers include the following registers:

- **Master Attention Register (MAR)** collects enabled attentions from the State Attention Register, Service Attention Register, No Space Attention Register, Request Attention Register, and Indicate Attention Register.
- **Master Notify Register (NMR)** is used to selectively enable attention in the Master Attention Register.
- **State Attention Register (STAR)** presents attentions for major states within the BSI-2 device and various error conditions.
- **State Notify Register (STNR)** is used to enable attentions in the State Attention Register.
- **Service Attention Register (SAR)** presents attentions for the PTOP and LMOP service functions.
- **Service Notify Register (SNR)** is used to enable attentions in the Service Attention Register.
- **No Space Attention Register (NSAR)** presents attentions generated when the IDUD, PSP, or CNF Queues run out of space or valid entries.
- **Request Attention Register (RAR)** presents attentions generated by both Request Channels.
- **Request Notify Register (RNR)** is used to enable attentions in the Request Attention Register.
- **Indicate Attention Register (IAR)** presents the attentions generated by the Indicate Channels.
- **Indicate Notify Register (INR)** is used to enable attentions in the Indicate Attention Register.
- **Compare Register (CMP)** is used for comparison with a write access of a conditional write (Attention) register.

5.0 Control Information (Continued)

Register Group	Address	Register Name	Access Rules	
			Read	Write
C	00	Mode Register 0 (MR0)	Always	Always
C	01	Mode Register 1 (MR1)	Always	Always
C	02	Pointer RAM Control and Address Register (PCAR)	Always	Always
C	03	Mailbox Address Register (MBAR)	Always	Always
E	04	Master Attention Register (MAR)	Always	Data Ignored
E	05	Master Notify Register (MNR)	Always	Always
E	06	State Attention Register (STAR)	Always	Conditional
E	07	State Notify Register (STNR)	Always	Always
E	08	Service Attention Register (SAR)	Always	Conditional
E	09	Service Notify Register (SNR)	Always	Always
E	0A	No Space Attention Register (NSAR)	Always	Conditional
E	0B	No Space Notify Register (NSNR)	Always	Always
C	0C	Limit Address Register (LAR)	Always	Always
C	0D	Limit Data Register (LDR)	Always	Always
E	0E	Request Attention Register (RAR)	Always	Conditional
E	0F	Request Notify Register (RNR)	Always	Always
C	10	Request Channel 0 Configuration Register 0 (R0CR0)	Always	Always
C	11	Request Channel 1 Configuration Register 0 (R1CR0)	Always	Always
C	12	Request Channel 0 Expected Frame Status Register (R0EFSR)	Always	Always
C	13	Request Channel 1 Expected Frame Status Register (R1EFSR)	Always	Always
E	14	Indicate Attention Register (IAR)	Always	Conditional
E	15	Indicate Notify Register (INR)	Always	Always
C	16	Indicate Threshold Register (ITR)	Always	INSTOP Mode or EXC = 1 Only
C	17	Indicate Mode Register (IMR)	Always	INSTOP Mode Only
C	18	Indicate Configuration Register (ICR)	Always	Always
C	19	Indicate Header Length Register (IHLR)	Always	INSTOP Mode or EXC = 1 Only
C	1A	Address Configuration Register (ACR)	Always	Always
C	1B	Request Channel 0 Configuration Register 1 (R0CR1)	N/A	N/A
C	1C	Request Channel 1 Configuration Register 1 (R1CR1)	N/A	N/A
E	1F	Compare Register (CMP)	Always	Always

C = Control Register
E = Event Register

FIGURE 5-2. Control and Event Registers

5.0 Control Information (Continued)

Address	Register	Reset
00	Mode Register 0	00
01	Mode Register 1	00
02	Pointer RAM Control and Address Register	NA
03	Mailbox Address Register	*
04	Master Attention Register	00
05	Master Notify Registers	00
06	State Attention Register	07
07	State Notify Register	00
08	Service Attention Register	0F
09	Service Notify Register	00
0A	No Space Attention Register	FF
0B	No Space Notify Register	00
0C	Limit Address Register	NA
0D	Limit Data Register	NA
0E	Request Attention Register	00
0F	Request Notify Register	00
10	Request Channel 0 Configuration Register 0	NA
11	Request Channel 1 Configuration Register 0	NA
12	Request Channel 0 Expected Frame Status Register	NA
13	Request Channel 1 Expected Frame Status Register	NA
14	Indicate Attention Register	00
15	Indicate Notify Register	00
16	Indicate Threshold Register	NA
17	Indicate Mode Register	NA
18	Indicate Configuration Register	NA
19	Indicate Header Length Register	NA
1A	Address Configuration Register	00
1B	Request Channel 0 Configuration Register 1	NA
1C	Request Channel1 Configuration Register 1	NA
1F	Compare Register	NA

* = Initialized to a silicon Revision code upon reset. The Revision code remains until it is overwritten by the host.

NA = Not altered upon reset.

FIGURE 5-3. Control and Event Registers Following Reset

Mode Register 0 (MR0)

The Mode Register 0 (MR0) is used to program major operating parameters for the BSI-2 device. This register should be programmed only at power-on, or after a software Master Reset.

This register is cleared upon reset.

Access Rules

Address	Read	Write
00h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
SMLB	SMLQ	VIRT	BIGEND	FLOW	MRST	FABCLK	TEST

Bit	Symbol	Description
D0	TEST	**TEST MODE:** Enables test logic, in which the transmitted frames counter will cause a service loss after four frames, instead of 255 frames.
D1	FABCLK	**FAST ABUS CLOCK:** For any AB__CLK frequency greater than LBC (12.5 MHz), this bit must be Zero. For an AB__CLK frequency equal to LBC, the User may optionally set this bit. Setting this bit causes a slight optimization of the internal MACSI synchronization timing valid only for the case where AB__CLK = LBC = 12.5 MHz. National recommends that all Users leave this bit as Zero.
D2	MRST	**MASTER RESET:** When this bit is set, the indicate, Request, and Status/Space Machines are placed in Stop Mode, and BSI-2 device registers are initialized to the values shown in *Figure 5-3*. This bit is cleared after the reset is complete.
D3	FLOW	**FLOW PARITY:** When this bit is set, parity checking is enabled at the Control Bus and MAC Indicate Data (Receive Data) interfaces. The BSI-2 uses Odd parity at all interfaces. Parity errors are reported in the STAR.CPE and STAR.BPE respectively. Parity is never checked at the ABus interface. When this bit is set, the parity bit for each ABus data byte flows with the data byte through the internal FIFO and across the MAC Request (Transmit) interface where it is checked by the BMAC device. Good parity is always generated on ABus and Control Bus writes. If this bit is reset, good parity is generated at the MAC Request interface. For the MAC Indicate Data interface, the parity check includes the frame's FC through ED fields. When this bit is Zero, no parity is checked on the Control Bus or the MAC Indicate Data interface.
D4	BIGEND	**BIG ENDIAN DATA FORMAT:** Selects between the Little Endian (BIGEND = 0) or Big Endian (BIGEND = 1) data format. See *Figure 4-3.*
D5	VIRT	**VIRTUAL ADDRESS MODE:** Selects between virtual (VIRT = 1) or physical (VIRT = 0) address mode on the ABus.
D6	SMLQ	**SMALL QUEUE:** Selects the size of all Descriptor queues and lists. When SMLQ = 0, the size is 4 kbytes; when SMLQ = 1, the size is 1 kbyte. Note that data pages are always 4 kbytes.
D7	SMLB	**SMALL BURSTS:** Selects size of bursts on ABus. When SMLB = 0, the BSI-2 device uses 1-, 4-, and 8-word transfers. When SMLB = 1, the BSI-2 device uses 1- and 4-word transfers.

5.0 Control Information (Continued)

Mode Register1 (MR1)

The Mode Register 1 (MR1) is used to program major operating parameters for the BSI-2 device. This register should be programmed only at power-on, or after a software Master Reset.

This register is cleared upon reset.

Access Rules

Address	Read	Write
01h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
AB_A31	AB_A30	AB_A29	AB_A28	ATM	ASM	RES	EAM

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th></tr>
<tr><td>D0</td><td>EAM</td><td><b>ENHANCED ABUS MODE:</b> This bit controls the Enhanced ABus Mode (EAM). This enhanced mode is intended to reduce the amount of logic required to interface to the SBus. When this bit is reset, the ABus operates as it does on the original BSI device (normal ABus mode). When this bit is set, the AB_A(31:28) bits within this register are sourced on the upper nibble of the address/data lines during the address cycle. Read Data is strobed the cycle after the assertion of AB_ACK. The AB_BR signal is guaranteed to be deasserted for at least two cycles (see <i>Figure 4-8</i> through <i>Figure 4-11</i>).
The AB_DEN pin becomes an input and the Error/Acknowledgment combinations are reencoded.
<table>
<tr><th colspan="2">EAM = 0</th><th colspan="3">EAM = 1</th><th rowspan="2">Description</th></tr>
<tr><th>AB_ACK</th><th>AB_ERR</th><th>AB_ACK
Ack(2)*</th><th>AB_DEN
Ack(1)*</th><th>AB_ERR
Ack(0)*</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>Wait Cycle</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>Word Acknowledge</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>Retry</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Error</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>0</td><td>Not Supported</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>Not Supported</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td>Not Supported</td></tr>
<tr><td></td><td></td><td>1</td><td>0</td><td>1</td><td>Not Supported</td></tr>
</table></td></tr>
<tr><td>D1</td><td>RES</td><td><b>RESERVED</b></td></tr>
<tr><td>D2</td><td>ASM</td><td><b>ADDRESS STROBE MODE:</b> The ASM bit controls the Address Strobe Mode. When this bit is reset, the AB_AS signal operates as it does on the BSI. When this bit is set, the BSI-2 generates an AB_AS signal which is designed to drive an address latch control line without additional logic (see <i>Figure 4-6</i> and <i>Figure 4-7</i>).</td></tr>
<tr><td>D3</td><td>ATM</td><td><b>ADDRESS TIMING MODE:</b> The ATM bit controls the Address Timing Mode. When this bit is reset, the AB_A(4:2) lines operate as they do on the BSI. These lines provide the demultiplexed address of the <i>next</i> word to be accessed on the ABus. When the ATM bit is set, the BSI-2 provides the address of the <i>current</i> word being accessed on the ABus (see <i>Figure 4-6</i> and <i>Figure 4-7</i>).</td></tr>
<tr><td>D7-4</td><td>AB_A(31:28)</td><td><b>AB_ADDRESS(31:28):</b> In normal operation (MR1.EAM = 0), the BSI-2 encodes channel information on the upper nibble of the AB_AD bus during the address cycle. In Enhanced ABus mode (MR1.EAM = 1), the upper nibble of the address lines are driven with the data pattern which the user has stored in these four bits, AB_A(31:28).</td></tr>
</table>

5.0 Control Information (Continued)

Pointer RAM Control and Address Register (PCAR)

The Pointer RAM Control and Address Register (PCAR) is used to program the parameters for the PTOP (Pointer RAM Operation) service function, in which data is written to or read from a Pointer RAM Register.

This register is not altered upon reset.

Access Rules

Address	Read	Write
02h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
BP1	BP0	PTRW	A4	A3	A2	A1	A0

Bit	Symbol	Description
D4-0	A4-0	**POINTER RAM ADDRESS:** These five bits contain the Pointer RAM Register address for a subsequent PTOP service function.
D5	PTRW	**PTOP READ/WRITE:** This bit determines whether a PTOP service function will be a read from the Pointer RAM Register to the mailbox in memory (PTRW = 1), or a write to the Pointer RAM Register from the mailbox (PTRW = 0).
D7-6	BP1-0	**BYTE POINTER:** These two bits are used to program an internal byte pointer for accesses to the 28-bit Mailbox Address Register. They are normally set to Zero to initialize the byte pointer for four successive writes (most-significant byte first) and are automatically incremented after each write. Because this register is not altered upon reset, it is important that these bits be explicitly configured before accessing the Mailbox Address Register.

Mailbox Address Register (MBAR)

The Mailbox Address Register (MBAR) is used to program the word-aligned 28-bit memory address of the mailbox used in the date transfer of the PTOP (Pointer RAM Operation) service function.

The address of the register is used as a window into four internal byte registers. The four byte registers are loaded by successive writes to the address after first setting the BP1-0 bits in the Pointer RAM Control and Address Register to Zero. The bytes must be loaded most-significant byte first. The BSI-2 device increments the byte pointer internally after each write or read. Mailbox Address bits 0 and 1 forced internally to Zero.

The four internal byte registers are initialized to a 28-bit silicon Revision code upon reset. The Revision code remains until it is overwritten by the host. The BP1-0 bits in the PCAR must be initialized before accessing the MBAR to fetch the silicon revision code.

Access Rules

Address	Read	Write
03h	Always	Always

Register Bits

7	0
Mailbox Address [27:24]	
Mailbox Address [23:16]	
Mailbox Address [15:8]	
Mailbox Address [7:0]	

Master Attention Register (MAR)

The Master Attention Register (MAR) collects enabled attentions from the State Attention Register, Service Attention Register, No Space Attention Register, Request Attention Register, and Indicate Attention Register. If the Notify bit in the Master Notify Register and the corresponding bit in the MAR are set to One, the INT is forced to LOW and thus triggers an interrupt.

Writes to the Master Attention Register are permitted, but do not change the contents.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
04h	Always	Data Ignored

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
STA	NSA	SVA	RQA	INA	RES	RES	RES

Bit	Symbol	Description
D2-0	RES	**RESERVED**
D3	INA	**INDICATE ATTENTION REGISTER:** Is set if any bit in the Indicate Attention Register is set.
D4	RQA	**REQUEST ATTENTION REGISTER:** Is set if any bit in the Request Attention Register is set.
D5	SVA	**SERVICE ATTENTION REGISTER:** Is set if any bit in the Service Attention Register is set.
D6	NSA	**NO SPACE ATTENTION REGISTER:** Is set if any bit in the No Space Attention Register is set.
D7	STA	**STATE ATTENTION REGISTER:** Is set if any bit in the State Attention Register is set.

Master Notify Register (MNR)

The Master Notify Register (MNR) is used to enable attentions in the Master Attention Register (MAR). If a bit in Register MNR and the corresponding bit in Register MAR are set to One, the $\overline{\text{INT}}$ signal is asserted to cause an interrupt.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
05h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
STAN	NSAN	SVAN	RQAN	INAN	RES	RES	RES

Bit	Symbol	Description
D2-0	RES	**RESERVED**
D3	INAN	**INDICATE ATTENTION REGISTER NOTIFY:** This bit is used to enable the INA bit in Register MAR.
D4	RQAN	**REQUEST ATTENTION REGISTER NOTIFY:** This bit is used to enable the RQA bit in Register MAR.
D5	SVAN	**SERVICE ATTENTION REGISTER NOTIFY:** This bit is used to enable the SVA bit in Register MAR.
D6	NSAN	**NO SPACE ATTENTION REGISTER NOTIFY:** This bit is used to enable the NSA bit in Register MAR.
D7	STAN	**STATE ATTENTION REGISTER NOTIFY:** This bit is used to enable the STA bit in Register MAR.

5.0 Control Information (Continued)

State Attention Register (STAR)

The State Attention Register (STAR) controls the state of the Indicate, Request, and Status/Space Machines. It also records parity, internal logic and ABus transaction errors. Each bit may be enabled by setting the corresponding bit in the State Notify Register.

Access Rules

Address	Read	Write
06h	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
ERR	BPE	CPE	CWI	CMDE	SPSTOP	RQSTOP	INSTOP

Bit	Symbol	Description
D0	INSTOP	**INDICATE STOP:** This bit is set by the host to place the Indicate Machine in Stop Mode. It is also set upon reset. Three different conditions cause the BSI-2 Device to set this bit. The first is an internal error. This is caused by a bad tag read out of the Indicate Data FIFO. This is a hardware error. The next condition is an invalid state. This is a hardware error where the Indicate Machine state bits contain an illegal pattern. The final condition is when the user programs an illegal header length for Header/Info sorting mode. An invalid value is any value less than four words. This bit is set by serious hardware failures or illegal software operations. Therefore it is recommended that the entire BSI-2 device be reset if this bit should get set during normal operation.
D1	RQSTOP	**REQUEST STOP:** This bit is set by the host to place the Request Machine in Stop Mode. It is also set upon reset. The BSI-2 device will set this bit if it detects that the Request Machine has entered an illegal state. This is a hardware error. The BSI-2 device will also set this bit if an ABus error is detected during any Request Operation. This includes REQ, ODUD, and ODU fetches, and CNF writes. This bit is set by serious hardware failures or ABus errors. Therefore it is recommended that the entire BSI-2 device be reset if this bit should get set during normal operation.
D2	SPSTOP	**STATUS/SPACE STOP:** This bit is set by the host to place the Status/Space Machine in Stop Mode. It is also set upon reset. In addition, the BSI-2 device will set this bit upon detecting an unrecoverable error. An unrecoverable error is an ABus error during a PSP fetch or a Pointer RAM Operation, (PTOP). In STOP Mode, only PTOP or LMOP service functions may be performed. This bit is set by ABus errors during critical Status/Space operations. Therefore it is recommended that the entire BSI-2 device be reset if this bit should get set during normal operation. This reset should include reloading of Pointer RAM values and restarting the PSP queues.
D3	CMDE	**COMMAND ERROR:** Indicates that the host performed an invalid operation. This occurs when an invalid value is loaded into the Indicate Header Length Register (which also sets the INSTOP attention). This bit is cleared upon reset. This bit is set when software performs an illegal operation. This indicates either a software bug or the improper operation of the processor. Therefore it is recommended that the entire BSI-2 device be reset if this bit should get set during normal operation.
D4	CWI	**CONDITIONAL WRITE INHIBIT:** Indicates that at least one bit of the previous conditional write operation was not written. This bit is set unconditionally after each write to a conditional write register. It is also set when the value of the Compare Register is not equal to the value of the register that was accessed for a write before it was written. This may indicate that the accessed register has changed since it was last read. This bit is cleared after a successful conditional write. CWI bit does not contribute to setting the STA bit of the Master Attention Register because its associated Notify bit is always 0. This bit is cleared upon reset.
D5	CPE	**CONTROL BUS PARITY ERROR:** Indicates a parity error detected on CBD7-0. If there is a Control Bus parity error during a host write, the write is suppressed. Control Bus parity errors are reported when flow-through parity is enabled (the FLOW bit of the Mode Register is set). This bit is cleared upon reset.
D6	BPE	**BMAC DEVICE PARITY ERROR:** Indicates parity error detected on MID7-0. This bit is cleared upon reset. This bit is only set if FLOW (parity enable) is set and the error occurred on a frame that the BSI-2 device has decided to copy or if it occurred before the copy decision was made.
D7	ERR	**ERROR:** This bit is set by the BSI-2 device when a non-recoverable error occurs. This includes ABus transaction errors or an internal state machine error. This bit is cleared upon reset.

State Notify Register (STNR)

The State Notify Register (STNR) is used to enable bits in the State Attention Register (STAR). If a bit in the STNR is set to One, the corresponding bit in Register STAR will be applied to the Master Attention Register, which can be used to generate an interrupt to the host.

All bits in this register are cleared to Zero upon reset.

Access Rules

Address	Read	Write
07h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
ERRN	BPEN	CPEN	CWIN	CMDEN	SPSTOPN	RQSTOPN	INSTOPN

Bit	Symbol	Description
D0	INSTOPN	**INDICATE STOP NOTIFY:** This bit is used to enable the INSTOP bit in Register STAR.
D1	RQSTOPN	**INDICATE STOP NOTIFY:** This bit is used to enable the RQSTOP bit in Register STAR.
D2	SPSTOPN	**STATUS/SPACE STOP NOTIFY:** This bit is used to enable the SPSTOP bit in Register STAR.
D3	CMDEN	**COMMAND ERROR NOTIFY:** This bit is used to enable the CMDE bit in Register STAR.
D4	CWIN	**CONDITIONAL WRITE INHIBIT NOTIFY:** This bit is always Zero. CWI is always masked.
D5	CPEN	**CONTROL BUS PARITY ERROR NOTIFY:** This bit is used to enable the CPE bit in Register STAR.
D6	BPEN	**BMAC DEVICE PARITY ERROR NOTIFY:** This bit is used to enable the BPE bit in Register STAR.
D7	ERRN	**ERROR NOTIFY:** This bit is used to enable the ERR bit in Register STAR.

Service Attention Register (SAR)

The Service Attention Register (SAR) is used to present the attentions for the service functions. Each bit may be enabled by setting the corresponding bit in the State Notify Register.

Access Rules

Address	Read	Write
08h	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	ABR0	ABR1	LMOP	PTOP

Bit	Symbol	Description
D0	PTOP	**POINTER RAM OPERATION:** This bit is cleared by the host to cause the BSI-2 device to transfer data between a Pointer RAM Register and a predefined mailbox location in memory. The Pointer RAM Control and Address Register contains the Pointer RAM Register address and determines the direction of the transfer (read or write). The memory address is defined via the Mailbox Address Register. This bit is set by the BSI-2 device after it performs the data transfer. While PTOP = 0, the host must not alter the Pointer RAM Address and Control Register of the Mailbox Address Register.
D1	LMOP	**LIMIT RAM OPERATION:** This bit is cleared by host to cause the BSI-2 device to transfer data between a Limit RAM Register and the Limit Data and Limit Address Registers. The Limit Address Register contains the Limit RAM Register address and determines the direction of the transfer (read and write). This bit is set by the BSI-2 device after it performs the data transfer. While LMOP = 0, the host must not alter either the Limit Address or Limit Data Register.
D2	ABR1	**ABORT REQUEST RCHN1:** This bit is cleared by the host to abort a Request on RCHN1. This bit is set by the BSI-2 device when RQABORT ends a request on RCHN1. The host may write a 1 to this bit, which may or may not prevent the request from being aborted. When this bit is cleared by the host, the USR1 bit in the Request Attention Register is set and further processing on RCHN1 is halted.
D3	ABR0	**ABORT REQUEST RCHN0:** This bit is cleared by the host to abort a Request on RCHN0. This bit is set by the BSI-2 device when RQABORT ends a request on RCHN0. The host may write a 1 to this bit, which may or may not prevent the request from being aborted. When this bit is cleared by the host, the USR0 bit in the Request Attention Register is set and further processing on RCHN0 is halted.
D7–4	RES	**RESERVED**

Service Notify Register (SNR)

The Service Notify Register (SNR) is used to enable attentions in the Service Attention Register (SAR). If a bit in Register SNR is set to One, the corresponding bit in Register SAR will be applied to the Master Attention Register, which can be used to generate a interrupt to the host.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
09h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	ABR0N	ABR1N	LMOPN	PTOPN

Bit	Symbol	Description
D0	PTOPN	**POINTER RAM OPERATION NOTIFY:** This bit is used to enable the PTOP bit in Register SAR.
D1	LMOPN	**LIMIT RAM OPERATION NOTIFY:** This bit is used to enable the LMOP bit in Register SAR.
D2	ABR1N	**ABORT REQUEST RCHN1 NOTIFY:** This bit is used to enable the ABR1 bit in Register SAR.
D3	ABR0N	**ABORT REQUEST RCHN0 NOTIFY:** This bit is used to enable the ABR0 bit in Register SAR.
D7–4	RES	**RESERVED**

No Space Attention Register (NSAR)

The No Space Attention Register (NSAR) presents the attentions generated when the CNF, PSP, or IDUD Queues run out of space. The host may set any attention bit to cause an attention for test purposes only, though this should not be done during normal operation.

The No Data Space attentions are set and cleared by the BSI-2 device automatically. The No Status Space attentions are set by the BSI-2 device, and must be cleared by the host.

Upon reset this register is set to 0xff.

Access Rules

Address	Read	Write
0Ah	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
NSR0	NSR1	LDI0	NSI0	LDI1	NSI1	LDI2	NSI2

Bit	Symbol	Description
D0	NSI2	**NO STATUS SPACE ON ICHN2:** This bit is set by the BSI-2 device upon a Reset, or when an IDUD has been written to the next-to-last available entry in the Indicate Channel's IDUD Status Queue. When this occurs, the BSI-2 device stops copying on ICHN2 and the last IDUD is written with special status. This bit must be cleared by the host before the BSI-2 device will resume copying on this Channel. Note that this bit should only be cleared after the appropriate limit register has been updated to give the BSI-2 more status space.
D1	LDI2	**LOW DATA SPACE ON ICHN2:** This bit is set by the BSI-2 device upon Reset, or when a PSP is prefetched from ICHN2's last PSP Queue location (as defined by the PSP Queue Limit Register). Note that the amount of warning is dependent on the length of the frame. There will always be one more page (4 kbytes) available for the BSI-2 device when this attention is generated. Another FDDI maximum-length frame (after the current one) will not fit in this space. If PSP fetching was stopped because there were no more PSP entries, fetching will resume automatically when the PSP Queue Limit Register is updated. This bit will be cleared automatically when the new PSP Descriptors are fetched. This bit should never be cleared directly by software. Clearing this bit can cause the BSI-2 to fetch invalid PSP descriptors.
D2	NSI1	**NO STATUS SPACE ON ICHN1:** This bit is set by the BSI-2 device upon a Reset, or when an IDUD has been written to the next-to-last available entry in the Indicate Channel's IDUD Status Queue. When this occurs, the BSI-2 device stops copying on ICHN1 and the last IDUD is written with special status. This bit must be cleared by the host before the BSI-2 device will resume copying on this Channel. Note that this bit should only be cleared after the appropriate limit register has been updated to give the BSI-2 more status space.
D3	LDI1	**LOW DATA SPACE ON ICHN1:** This bit is set by the BSI-2 device upon Reset, or when a PSP is prefetched from ICHN1's last PSP Queue location (as defined by the PSP Queue Limit Register). Note that the amount of warning is dependent on the length of the frame. There will always be one more page (4 kbytes) available for the BSI-2 device when this attention is generated. Another FDDI maximum-length frame (after the current one) will not fit in this space. If PSP fetching was stopped because there were no more PSP entries, fetching will resume automatically when the PSP Queue Limit Register is updated. This bit will be cleared automatically when the new PSP Descriptors are fetched. This bit should never be cleared directly by software. Clearing this bit can cause the BSI-2 to fetch invalid PSP descriptors.
D4	NSI0	**NO STATUS SPACE ON ICHN0:** This bit is set by the BSI-2 device upon a Reset, or when an IDUD has been written to the next-to-last available entry in the Indicate Channel's IDUD Status Queue. When this occurs, the SBI device stops copying on ICHN0 and the last IDUD is written with special status. This bit must be cleared by the host before the BSI-2 device will resume copying on this Channel. Note that this bit should only be cleared after the appropriate limit register has been updated to give the BSI-2 more status space.
D5	LDI0	**LOW DATA SPACE ON ICHN0:** This bit is set by the BSI-2 device upon Reset, or when a PSP is prefetched from ICHN0's last PSP Queue location (as defined by the PSP Queue Limit Register). Note that the amount of warning is dependent on the length of the frame. There will always be one more page (4 kbytes) available for the BSI-2 device when this attention is generated. Another FDDI maximum-length frame (after the current one) will not fit in this space. If PSP fetching was stopped because there were no more PSP entries, fetching will resume automatically when the PSP Queue Limit Register is updated. This bit will be cleared automatically when the new PSP Descriptors are fetched. This bit should never be cleared directly by software. Clearing this bit can cause the BSI-2 to fetch invalid PSP descriptors.

5.0 Control Information (Continued)

Bit	Symbol	Description
D6	NSR1	**NO STATUS SPACE ON RCHN1:** This bit is set by the BSI-2 device upon a Reset, or when it has written a CNF Descriptor to the next-to last Queue location. Due to internal pipelining, the BSI-2 device may write up to two more CNFs to the Queue after this attention is generate. Thus the Host must set the CNF Queue Limit Register to be one less than the available space in the Queue. This bit (as well as the USR attention bit) must be cleared by the Host before the BSI-2 device will continue to process requests on RCHN1. Note that this bit should only be cleared after the appropriate limit register has been updated to give the BSI-2 more status space.
D7	NSR0	**NO STATUS SPACE ON RCHN0:** This bit is set by the BSI-2 device upon Reset, or when it has written a CNF Descriptor to the next-to-last Queue location. Due to internal pipelining, the BSI-2 device may write up to two more CNFs to the Queue after this attention is generated. Thus the Host must set the CNF Queue Limit Register to be one less than the available space in the Queue. This bit (as well as the USR attention bit) must be cleared by the Host before the BSI-2 device will continue to process requests on RCHN0. Note that this bit should only be cleared after the appropriate limit register has been updated to give the BSI-2 more status space.

No Space Notify Register (NSNR)

The No Space Notify Register (NSNR) is used to enable attentions in the No Space Attention Register (NSAR). If a bit in Register NSNR is set to One, the corresponding bit in Register NSAR will be applied to the Master Attention Register, which can be used to generate an interrupt to the host.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
0Bh	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
NSR0N	NSR1N	LDI0N	NSI0N	LDI1N	NSI1N	LDI2N	NSI2N

Bit	Symbol	Description
D0	NSI2N	**NO STATUS SPACE ON ICHN2 NOTIFY:** This bit is used to enable the NSI2 in Register NSAR.
D1	LDI2N	**LOW DATA SPACE ON ICHN2 NOTIFY:** This bit is used to enable the LDI2 in Register NSAR.
D2	NSI1N	**NO STATUS SPACE ON ICHN2 NOTIFY:** This bit is used to enable the NSI1 in Register NSAR.
D3	LDI1N	**LOW DATA SPACE ON ICHN2 NOTIFY:** This bit is used to enable the LDI1 in Register NSAR.
D4	NSI0N	**NO STATUS SPACE ON ICHN2 NOTIFY:** This bit is used to enable the NSI0 in Register NSAR.
D5	LDI0N	**LOW DATA SPACE ON ICHN2 NOTIFY:** This bit is used to enable the LDI0 bit in Register NSAR.
D6	NSR1N	**NO STATUS SPACE ON ICHN2 NOTIFY:** This bit is used to enable the NSR1 bit in Register NSAR.
D7	NSR0N	**LOW DATA SPACE ON ICHN2 NOTIFY:** This bit is used to enable the NSR0 bit in Register NSAR.

Limit Address Register (LAR)

The Limit Address Register (LAR) is used to program the parameters for a LMOP (Limit RAM Operation) service function. This register is not altered upon reset.

Access Rules

Address	Read	Write
0Ch	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
LRA3	LRA2	LRA1	LRA0	LMRW	RES	RES	LRD8

Bit	Symbol	Description
D0	LRD8	**LIMIT RAM DATA BIT 8:** This bit contains the most-significant data bit read or written from the addressed limit RAM Register. Bits LDR8 and LDR7 are ''don't cares'' when using small (1 kbyte) queues.
D2−1	RES	**RESERVED**
D3	LMRW	**LMOP READ/WRITE:** This bit determines whether a LMOP service function will be a read (LMRW = 1) or write (LMRW = 0).
D7−4	LRA3−0	**LIMIT RAM REGISTER ADDRESS:** Used to program the Limit RAM Register address for a subsequent LMOP service function.

Limit Data Register (LDR)

The Limit Data Register (LDR) is used to hold the 8 least-significant Limit RAM data bits transferred in an LMOP service function. (The most-significant data bit is in the Limit Address Register.)

This register is not altered upon reset.

Access Rules

Address	Read	Write
0Dh	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
LRD7	LRD6	LRD5	LRD4	LRD3	LRD2	LRD1	LRD0

Bit	Symbol	Description
D7−0	LRD7−0	**LIMIT RAM DATA BIT 7−0:** These bits contain the least-significant data bits read or written from or a Limit RAM Register in an LMOP service function. Bit LDR7 is a ''don't care'' when using small (1 kbyte) queues.

Request Attention Register (RAR)

The Request Attention Register (RAR) is used to present exception, breakpoint, request complete, and unserviceable request attentions generated by each Request Channel. Each bit may be enabled by setting the corresponding bit in the Request Notify Register.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
0Eh	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
USRR0	RCMR0	EXCR0	BRKR0	USRR1	RCMR1	EXCR1	BRKR1

Bit	Symbol	Description
D0	BRKR1	**BREAKPOINT ON RCHN1:** Is set by the BSI-2 device when a CNF Descriptor is written on RCHN1. No action is taken by the BSI-2 device if the host sets this bit.
D1	EXCR1	**EXCEPTION ON RCHN1:** Is set by the BSI-2 device when an exception occurs on RCHN1. An exception condition consists of one of the following events: ABus Error, Consistency Failure (REQ or ODUD), BMAC MAC Reset, Timeout (TRT expires), BMAC abort (MAC frame received or FC/Request Class inconsistency), RINGOP change, host abort (via SAR register), FIFO underrun, or confirmation exception (for full Confirmation). No action is taken by the BSI-2 device if the host sets this bit. This bit indicates that a request on RCHN1 did not complete normally. This implies that an exception event occurred or that the Request is improperly formed. The corresponding Confirmation (CNF) Descriptor will give more status about the failure. The additional information in the CNF descriptor should be used to make a decision about the severity of the error. If the exception was caused by an ABus error, the RQSTOP will also be set.
D2	RCMR1	**REQUEST COMPLETE ON RCHN1:** Is set by the BSI-2 device when it has completed processing a Request object on RCHN1, (note that the BSI-2 may set this bit before writing the CNF descriptor to the CNF queue). This completion may be a normal completion where a request object was transmitted without error. It may also be an abnormal completion due to one of the exception conditions listed above in EXCR1. No action is taken if the Host sets this bit.
D3	USRR1	**UNSERVICEABLE REQUEST ON RCHN1:** Is set by the BSI-2 device when a Request cannot be processed on RCHN1. This occurs when the Request Class is inappropriate for the current ring state (e.g. Asynchronous transmission while RINGOP = 0), or when there is no CNF status space, or when the host aborts a request by clearing the ABR bit in the Service Attention Register. While this bit is set, no requests will be processed on RCHN1. The host must clear this bit to resume request processing. If the USRR1 was set due to lack of CNF space, this condition must be corrected by giving the BSI-2 device more CNF space before restarting the channel. If it was due to a request class/ RINGOP incompatibility, then the reason for the incompatibility must be resolved.
D4	BRKR0	**BREAKPOINT ON RCHN0:** Is set by the BSI-2 device when a CNF Descriptor is written on RCHN0. No action is taken by the BSI-2 device if the host sets this bit.
D5	EXCR0	**EXCEPTION ON RCHN0:** This bit is set by the BSI-2 device when an exception occurs on RCHN0. An exception condition consists of one of the following events: ABus Error, Consistency Failure (REQ or ODUD), BMAC MAC Reset, Timeout (TRT expires), BMAC abort (MAC frame received or FC/Request Class inconsistency), RINGOP change, host abort (via SAR register), FIFO underrun, or confirmation exception (for full Confirmation). No action is taken by the BSI-2 device if the host sets this bit. This bit indicates that a request on RCHN0 did not complete normally. This implies that an exception event occurred or that the Request is improperly formed. The corresponding Confirmation (CNF) Descriptor will give more status about the failure. The additional information in the CNF descriptor should be used to make a decision about the severity of the error. If the exception was caused by an ABus error, the RQSTOP will also be set.
D6	RCMR0	**REQUEST COMPLETE ON RCHN0:** Is set by the BSI-2 device when it has completed processing a Request object on RCHN0. This completion may be a normal completion where a request object was transmitted without error. It may also be an abnormal completion due to one of the exception conditions listed above in EXCR0. This bit, (together with the Breakpoint bit BRKR0) indicates that there are CNF descriptors to be processed. No action is taken if the Host sets this bit.

Bit	Symbol	Description
D7	USRR0	**UNSERVICEABLE REQUEST ON RCHN0:** This bit is set by the BSI-2 device when a Request cannot be processed on RCHN0. This occurs when the Request Class is inappropriate for the current ring state (e.g. Asynchronous transmission while RINGOP = 0), or when there is no CNF status space, or when the host aborts a request by clearing the ABR bit in the Service Attention Register. While this bit is set, no requests will be processed on RCHN0. The host must clear this bit to resume request processing. If the USRR0 was set due to lack of CNF space, this condition must be corrected by giving the BSI-2 device more CNF space before restarting the channel. If it was due to a request class/ RINGOP incompatibility, then the reason for the incompatibility must be resolved.

Request Notify Register (RNR)

The Request Notify Register (RNR) is used to enable attentions in the Request Attention Register (RAR). If a bit in Register RNR is set to One, the corresponding bit in Register RAR will be applied to the Master Attention Register, which can be used to generate an interrupt to the host.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
0Fh	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
USRR0N	RCMR0N	EXCR0N	BRKR0N	USRR1N	RCMR1N	EXCR1N	BRKR1N

Bit	Symbol	Description
D0	BRKR1N	**BREAKPOINT ON RCHN1 NOTIFY:** This bit is used to enable the BRKR1 bit in Register RAR.
D1	EXCR1N	**EXCEPTION ON RCHN1 NOTIFY:** This bit is used to enable the EXCR1 bit in Register RAR.
D2	RCMR1N	**REQUEST COMPLETE ON RCHN1 NOTIFY:** This bit is used to enable the RCMR1 bit in Register RAR.
D3	USRR1N	**UNSERVICEABLE REQUEST ON RCHN1 NOTIFY:** This bit is used to enable the USRR1 bit in Register RAR.
D4	BRKR0N	**BREAKPOINT ON RCHN0 NOTIFY:** This bit is used to enable the BRKR0 bit in Register RAR.
D5	EXCR0N	**EXCEPTION ON RCHN0 NOTIFY:** This bit is used to enable the EXCR0 bit in Register RAR.
D6	RCMR0N	**REQUEST COMPLETE ON RCHN0 NOTIFY:** This bit is used to enable the RCMR0 bit in Register RAR.
D7	USRR0N	**UNSERVICEABLE REQUEST ON RCHN0 NOTIFY:** This bit is used to enable the USRR0 bit in Register RAR.

5.0 Control Information (Continued)

Request Channel 0 and 1 Configuration Registers 0 (R0CR0 and R1CR0)

The two Request Configuration Registers 0 (R0CR0 and R1CR0) are programmed with operational parameters for each of the Request Channels. Additional Request Channel parameters are configured in Request Configuration Registers 1. These registers may only be altered between Requests, i.e., while the particular Request Channel does not have a Request loaded.

These registers are not altered upon reset.

Access Rules

Address	Read	Write
10–11h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
TT1	TT0	PRE	HLD	FCT	SAT	VST	FCS

Bit	Symbol	Description
D0	FCS	**FRAME CHECK SEQUENCE DISABLE:** When this bit is set, the BSI-2 device asserts the FCST signal throughout the request. This may drive the BMAC device FCST pin, or also the SAT or SAIGT pins, depending on the application. This bit is normally used to program the BMAC device not to concatenate its generated FCS to the transmitted frame. The Valid FCS bit in the Expected Frame Status Register independently determines whether a frame needs a valid FCS to meet the matching frame criteria.
D1	VST	**VOID STRIPPING:** When this bit is set, the BSI-2 device asserts the STRIP output signal out throughout the request. This may drive the BMAC device STRIP (Void Strip) pin, or also the SAT pin, depending on the application.
D2	SAT	**SOURCE ADDRESS TRANSPARENCY:** When this bit is set, the BSI-2 device asserts the SAT output signal throughout the request. This may drive the BMAC device's SAT and/or SAIGT pins, depending on the application. When SAT is set, Full Confirmation requires the use of the EM (External SA Match) signal.
D3	FCT	**FRAME CONTROL TRANSPARENCY:** When this bit is set, the FC will be sourced from the ODU (not the REQ.First or REQ.Only Descriptor), When Full Confirmation is enabled and FCT = 0, all bits of the FC in returning frames must match the FC field in the REQ Descriptor; if FCT = 1, only the C, L and r bits must match. Note that since the BSI-2 device decodes the REQ.F Descriptor FC field to determine whether to assert RQCLM/RQBCN, FC transparency may be used to send Beacons or Claims in any ring non-operational state, as long as the FC in the REQ Descriptor is not set to Beacon or Claim. By programming a Beacon or Claim FC in the REQ Descriptor, then using FC transparency, any type of frame may be transmitted in the Beacon or Claim state.
D4	HLD	**HOLD:** When this bit is set, the BSI-2 device will not end a service opportunity until the Request is complete. When this bit is Zero, the BSI-2 device ends the service opportunity on the Request Channel when all of the following conditions are met: 1. There is no valid request active on the Request Channel. 2. The service class is non-immediate. 3. There is no data in the FIFO. 4. There is no valid REQ fetched by the BSI-2 device This bit also affects Prestaging on RCHN1 (Request Channel 1). When HLD = 0, prestaging is enabled on RCHN1, regardless of the state of the PRE bit (except for Immediate service classes). When HLD = 1, prestaging is determined by the PRE bit. This option can potentially waste ring bandwidth, but may be required (particularly on RCHN0, Request Channel 0) if a guaranteed service time is required. When using the Repeat option, HLD is required for small frames. If HLD is not used, the other Request Channel will be checked for service before releasing the token between frames. This may not be the desired action, particularly if there is a request on RCHN1 that needs servicing after the completion of RCHN0's Repeated Request.

5.0 Control Information (Continued)

Bit	Symbol	Description
D5	PRE	**PREEMPT/PRESTAGE:** When this bit is set, preemption is enabled for RCHN0, and prestaging is enabled for RCHN1 (prestaging is always enabled for RCHN0). When this bit Zero, preemption is disabled and Prestaging is enabled on the RCHN0. When preemption is enabled, RCHN0 preempts a (non-committed) frame of RCHN1 already in the FIFO, causing it to be purged and refetched after RCHN0's request has been serviced. When the Request Machine servicing on RCHN1 and a request on RCHN0 becomes active, if preemption is enabled on RCHN0, the Request Machine will finish transmitting the current frame on RCHN1, then release the token and more back to the start state. This has the effect of reprioritizition the Request Channels, thus ensuring that frames on RCHN0 are transmitted at the next service opportunity. When RCHN0 has been serviced, transmission will continue on RCHN1 with no loss of data. When prestaging is enabled, the next frame for RCHN1 is staged (ODUs are loaded into the FIFO before the token arrives). If prestaging is not enabled, the Request Machine waits until the token is captured before staging the first frame. Once the token is captured, the Request Machine begins fetching data, and when the FIFO threshold has been reached, transmits the data on the Request Channel. For requests with an Immediate service class, prestaging is not applicable. When this bit is Zero, preemption is disabled for RCHN0, and request on RCHN1 will not be prestaged unless the HLD bit is Zero, in which case RCHN1 will prestage data regardless of the setting of the PRE bit. Note that when prestaging is not enabled on RCHN1, data is not staged until the token is captured. Since there is no data in the FIFO (if there is no active request on RCHN0), the BSI-2 device will immediately release the token if the HLD option is not set.
D7–6	TT1–0	**TRANSMIT THRESHOLD:** Determines the threshold on the output data FIFO before the BSI-2 device requests transmission. **TT1 TT0 Threshold Value** 0 0 8 Words 0 1 16 Words 1 0 128 Words 1 1 256 Words

Request Channel 0 and 1 Expected Frame Status Registers (R0EFSR and R1EFSR)

The Expected Frame Status Registers (R0EFSR and R1EFSR) define the matching criteria used for Full Confirmation of returning frames on each Request Channel. A returning frame must meet the programmed criteria to be counted as a matching returning frames on each Request Channel.

These registers are not altered upon reset.

Access Rules

Address	Read	Write
12–13h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
VDL	VFCS	EE1	EE0	EA1	EA0	EC1	EC0

Bit	Symbol	Description
D1–0	EC1–0	**EXPECTED C INDICATOR:** **EC1 EC0 Value** 0 0 Any 0 1 R 1 0 S 1 1 R or S
D3–2	EA1–0	**EXPECTED A INDICATOR:** **EA1 EA0 Value** 0 0 Any 0 1 R 1 0 S 1 1 R or S
D5–4	EE1–0	**EXPECTED E INDICATOR:** **EE1 EE0 Value** 0 0 Any 0 1 R 1 0 S 1 1 R or S
D6	VFCS	**VALID FCS:** When this bit is set, returning frames must have a valid FCS field to meet the confirmation criteria.
D7	VDL	**VALID DATA LENGTH:** When this bit is set, returning frames must have a valid VDL field to meet the confirmation criteria.

5.0 Control Information (Continued)

Indicate Attention Register (IAR)

The Indicate Attention Register (IAR) is used to present exception and breakpoint attentions generated by each Indicate Channel. An Attention bit is set by hardware when an exception or breakpoint occurs on the corresponding Indicate Channel.

Each bit may be enabled by setting the corresponding bit in the Indicate Notify Register.

Access Rules

Address	Read	Write
14h	Always	Conditional

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	EXCI0	BRKI0	EXCI1	BRKI1	EXCI2	BRKI2

Bit	Symbol	Description
D0	BRKI2	**BREAKPOINT ON ICHN2:** This bit is set when a breakpoint is detected on Indicate Channel 2. No action is taken if the host sets this bit.
D1	EXCI2	**EXCEPTION ON ICHN2:** While this bit is set, copying is disabled on ICHN2. This bit is set by the BSI-2 device when an exception occurs on Indicate Channel 2. An exception consists of an ABus error during an IDU or IDUD write for ICNH2. It may be set by the host to disable copying on ICHN2, which is convenient when updating the Indicate Header Length and Indicate Threshold register. When this bit is set by the device it signifies that the last frame received on this channel had an ABus error. It is the last frame because the setting of this bit disables further copying on this channel. The last frame should be discarded.
D2	BRKI1	**BREAKPOINT ON ICHN1:** This bit is set when a breakpoint is detected on Indicate Channel 1. No action is taken if the host sets this bit.
D3	EXCI1	**EXCEPTION ON ICHN1:** While this bit is set, copying is disabled on ICHN1. This bit is set by the BSI-2 device when an exception occurs on Indicate Channel 1. An exception consists of an ABus error during an IDU or IDUD write for ICNH1. It may be set by the host to disable copying on ICHN1, which is convenient when updating the Indicate Header Length and Indicate Threshold register. When this bit is set by the device it signifies that the last frame received on this channel had an ABus error. It is the last frame because the setting of this bit disables further copying on this channel. The last frame should be discarded.
D4	BRKI0	**BREAKPOINT ON ICHN0:** This bit is set when a breakpoint is detected on ICHN0. No action is taken if the host sets this bit. The User must set IMR.BOS to use the BRKI0 breakpoint.
D5	EXCIO	**Exception on ICHN0:** While this bit is set, copying is disabled on ICHN0. This bit is set by the BSI-2 device when an exception occurs on Indicate Channel 0. An exception consists of an ABus error during an IDU or IDUD write for ICNH0. It may be set by the host to disable copying on ICHN0. When this bit is set by the device it signifies that the last frame received on this channel had an ABus error. It is the last frame because the setting of this bit disables further copying on this channel. The last frame should be discarded.
D7–6	RES	**RESERVED**

Indicate Notify Register (INR)

The Indicate Notify Register (INR) is used to enable attentions in the Indicate Attention Register (IAR). If a bit in Register INR is set to One, the corresponding bit in Register IAR will be applied to the Master Attention Register, which can be used to generate an interrupt to the host.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
15h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	EXC0N	BRK0N	EXC1N	BRK1N	EXC2N	BRK2N

Bit	Symbol	Description
D0	BRK2N	**BREAKPOINT ON ICHN2 NOTIFY:** This bit is used to enable the BRK2 bit in Register IAR.
D1	EXC2N	**EXCEPTION ON ICHN2 NOTIFY:** This bit is used to enable the EXC2 bit in Register IAR.
D2	BRK1N	**BREAKPOINT ON ICHN1 NOTIFY:** This bit is used to enable the BRK1 bit in Register IAR.
D3	EXC1N	**EXCEPTION ON ICHN1 NOTIFY:** This bit is used to enable the EXC1 bit in Register IAR.
D4	BRK0N	**BREAKPOINT ON ICHN0 NOTIFY:** This bit is used to enable the BRK0 bit in Register IAR.
D5	EXC0N	**EXCEPTION ON ICHN0 NOTIFY:** This bit is used to enable the EXC0 bit in Register IAR.
D7–6	RES	**RESERVED**

Indicate Threshold Register (ITR)

The Indicate Threshold Register (ITR) specifies the maximum number of frames that can be received on Indicate Channel 1 or Indicate Channel 2 before an attention will be generated. This register may be written only when the INSTOP bit in the State Attention Register is set, or when the Indicate Channel's corresponding EXC bit in the Indicate Attention Register is set.

This register is not altered upon reset.

Access Rules

Address	Read	Write
16h	Always	INSTOP Mode or EXC = 1 Only

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
THR7	THR6	THR5	THR4	THR3	THR2	THR1	THR0

Bit	Symbol	Description
D7–0	THR7–0	**THRESHOLD DATA BITS 7–0:** The value programmed in this register is loaded into an internal counter every time the Indicate Channel changes. Each valid frame copied on the current Channel decrements the counter. When the counter reaches Zero, a status breakpoint attention is generated (i.e., the Channel's BRK bit in the Indicate Attention Register is set) if the Channel's Breakpoint on Threshold (BOT) bit in the Indicate Mode Register is set. Loading the Indicate Threshold Register with Zero generates a breakpoint after 256 consecutive frames are received on any one Indicate Channel.

Indicate Mode Register (IMR)

The Indicate Mode Register (IMR) defines configuration options for all three indicate Channels, including the sort mode, frame filtering, and status breakpoints.

This register may be written only when the INSTOP bit in the State Attention Register is set. It may be written with its current value any time, which is useful for one-shot sampling.

This register is not altered upon reset.

Access Rules

Address	Read	Write
17h	Always	INSTOP Mode Only

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
SM1	SM0	SKIP	FPP	BOT2	BOT1	BOB	BOS

Bit	Symbol	Description
D0	BOS	**BREAKPOINT ON SERVICE OPPORTUNITY:** Enables the end of a service opportunity to generate an Indicate breakpoint attention (i.e., set the Channel's BRK bit in the Indicate Attention Register). Service opportunities include receipt of a Token, a MAC Frame, or a ring operational change following some copied frames. This bit should be set to 1 if BRKI0 will be used.
D1	BOB	**BREAKPOINT ON BURST:** Enables the end of a burst to generate an Indicate breakpoint attention (i.e., set the Channel's BRK bit in the Indicate Attention Register). End of burst includes Channel change, DA change, SA change, or MAC INFO change. A Channel change is detected from the FC field of valid, copied frames. A DA change is detected when a frame's DA field changes from our address to any other. An SA change is detected when a frame's SA field is not the same as the previous one. A MAC INFO change occurs when a MAC frame does not have the identical first four bytes of INFO as the previous frame. This breakpoint always sets the BRK bit (i.e., this breakpoint is always enabled).
D2	BOT1	**BREAKPOINT ON THRESHOLD FOR ICHN1:** Enables the value in the Indicate Threshold Register to be used to generate an Indicate breakpoint attention on Indicate Channel 1, (i.e., set the BRK1 bit in the Indicate Attention Register).
D3	BOT2	**BREAKPOINT ON THRESHOLD FOR ICHN2:** Enables the value in the Indicate Threshold Register to be used to generate an Indicate breakpoint attention on Indicate Channel 2, (i.e., set the BRK2 bit in the Indicate Attention Register).
D4	FPP	**FRAME-PER-PAGE:** This bit controls how received frames are packed into the Pool Space Pages which are provided via the PSP Descriptors. When this bit is reset, the BSI-2 assembles multiple frames into a single page of Pool Space when possible, (i.e. the frames are smaller than the 4 kbyte page size). When this bit is set, the BSI-2 will force a page break and fetch a new PSP for each frame received. This guarantees that no page will contain more than one frame. This mode is useful for systems where received frames are not processed in order of receipt. This is because space reclamation is greatly simplified. A side affect of this mode is that no frame will span more than two pages, (i.e. a frame will have at most two IDUDs).
D5	SKIP	**SKIP ENABLE:** Enables filtering on Indicate Channel 0 when the Copy Control field for ICHN0 in the Indicate Configuration Register is set to 01 or 10. When this bit is set, only the unique MAC frames received on Indicate Channel 0 will be copied to memory (i.e., those having an FC field or first four bytes of the Information field that differs from the previous frame). When this bit is 0, the BSI-2 will copy all MAC frames that meet the Copy Criteria. When this bit changes from a 0 to a 1, the BSI-2 will copy the next MAC frame, even if it is the same as the previous frame. Note that the ''Promiscuous'' Copy Mode overrides this SKIP mode, (when the User selects Promiscuous copy and the SKIP mode, the BSI-2 will copy all MAC frames).

5.0 Control Information (Continued)

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th></tr>
<tr><td>D7–6</td><td>SM1–0</td><td><b>SORT MODE:</b> These bits determine how the BSI-2 device sorts Indicate data onto Indicate Channels 1 and 2.
<table>
<tr><th>SM1</th><th>SM0</th><th>ICHN2</th><th>ICHN1</th></tr>
<tr><td>0</td><td>0</td><td>Asynchronous</td><td>Synchronous</td></tr>
<tr><td>0</td><td>1</td><td>External</td><td>Internal</td></tr>
<tr><td>1</td><td>0</td><td>Info</td><td>Header</td></tr>
<tr><td>1</td><td>1</td><td>Low Priority</td><td>High Priority</td></tr>
</table>
The Synchronous/Asynchronous Sort Mode is intended for use in end-stations or applications using synchronous transmission.

The Internal/External Sorting Mode is intended for bridging or monitoring applications. MAC/SMT frames matching the internal (BMAC device) address are sorted onto ICHN0, and all other frames matching the BMAC device's internal address (short or long) are sorted onto ICHN1. All frames matching the external address (frames requiring bridging) are sorted onto ICHN2. This sorting mode uses the EM, EA, and ECIP input signals with ex- ternal address matching circuitry. See the section on External address matching for a full description of the timing required on these signals (Section 4.3). Promiscuous mode on ICHN2 does not require any external matching logic to copy frames.

The Header/Info Sort Mode is intended for high performance protocol processing. MAC/ SMT frames are sorted onto ICHN0, while all other frames are sorted onto ICHN1 and ICHN2. Frame bytes from the FC up to the programmed header length are copied onto ICHN1. The remaining bytes(info) are copied onto ICHN2. Only one stream of IDUDs is produced (on ICHN1), but both Indicate Channel's PSP queues are used for space (i.e., PSPs from ICHN1 for header space, and PSPs from ICHN2 for info space). Frames may comprise a header only, or a header+info. For frames with info, multi-part IDUD objects are produced. For multi-part IDUDs, the Indicate Status field in the IDUD is used to determine which part of the IDUD object points to the end of the header. The remainder of the IDUD is used to determine which part of the IDUD object points to the end of the header. The remainder of the IDUD object points to the Info. If Pool Space is only declared for ICNH1, then only the Headers will be written to memory. This is useful for protocol monitor applications.

For example, if page crosses occur while writing the header and while writing out the Info, the BSI-2 Device will generate a four part IDUD object (IDUD.First, IDUD.Middle, IDUD.Middle, IDUD.Last). The IDUD.First will have a status of "page cross". The first IDUD. Middle will have a status of "end of header". The next IDUD.Middle will have a status of "page cross". The IDUD.Last will have an "end of frame" status.

The High Priority/Low Priority Sort Mode is intended for end stations using two priority levels of asynchronous transmission. The priority is determined by the most-significant z-bit of the FC (zzz = 0xx = low-priority; zzz = 1xx = high priority). Synchronous frames are sorted onto ICHN1 and MAC/SMT frames are sorted onto ICHN0.</td></tr>
</table>

5.0 Control Information (Continued)

Indicate Configuration Register (ICR)

The Indicate Configuration Register (ICR) is used to program the copy criteria for each of the Indicate Channels. This register is not altered upon reset.

Access Rules

Address	Read	Write
18h	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
CC0		RES	CC1		RES	CC2	

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th></tr>
<tr><td>D1–0</td><td>CC2</td><td><b>COPY CONTROL ICHN2:</b>
<b>CC1 CC0 Copy Mode</b>
0 0 Do Not Copy
0 1 Copy if (AFLAG | (~ECIP & EA)) & ~MFLAG
1 0 Copy if (AFLAG | (~ECIP & EA))
1 1 Copy Promiscuously.</td></tr>
<tr><td>D2</td><td>RES</td><td><b>RESERVED</b></td></tr>
<tr><td>D4–3</td><td>CC1</td><td><b>COPY CONTROL ICHN1:</b>
<b>CC4 CC3 Copy Mode</b>
0 0 Do Not Copy
0 1 Copy if (AFLAG | (~ECIP & EA)) & ~MFLAG
1 0 Copy if (AFLAG | (~ECIP & EA))
1 1 Copy Promiscuously.</td></tr>
<tr><td>D5</td><td>RES</td><td><b>Reserved</b></td></tr>
<tr><td>D7–6</td><td>CC0</td><td><b>COPY CONTROL ICHN0:</b>
<b>CC7 CC6 Copy Mode</b>
0 0 Do Not Copy
0 1 Copy if (AFLAG | (~ECIP & EA)) & ~MFLAG
1 0 Copy if (AFLAG | (~ECIP & EA))
1 1 Copy Promiscuously.</td></tr>
</table>

Indicate Header Length Register (IHLR)

The Indicate Header Length Register (IHLR) defines the length (in words) of the frame header, for use with the Header/Info Sort Mode.

The Indicate Header Length Register must be initialized before setting the Sort Mode in Header/Info. This register may be changed while the INSTOP bit in the State Attention Register or the EXC bit in the Indicate Attention Register is set.

This register is not altered upon reset.

Access Rules

Address	Read	Write
19h	Always	INSTOP Mode or EXC = 1 Only

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
HL7	HL6	HL5	HL4	HL3	HL2	HL1	HL0

Bit	Symbol	Description
D7–0	HL7–0	**HEADER LENGTH:** Specifies the length (in words) of the frame header, for use with the Header/Info Sort Mode. The frame FC is written as a separate word, and thus counts as one word. For example, to split after eight bytes of FDDI INFO in a frame with long addresses, this register is programmed with the value 06 (1 word FC, 1.5 DA, 1.5 SA, 2HDR__DATA). IHLR must not be loaded with a value less than 4. If it is, the BSI-2 device sets the Command Error (CMDE) and Indicate Stop (INSTOP) attentions. This Register only affects the Header/Info sort mode.

Address Configuration Register (ACR)

This register contains bits for configuring the address swapping logic.

All bits in this register are set to Zero upon reset.

Access Rules

Address	Read	Write
1Ah	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RSWP1	RSWP0	ISWP2	ISWP1	ISWP0

Bit	Symbol	Description
D0	ISWP0	**INDICATE SWAP 0:** This bit controls the address swapping logic for Indicate Channel 0 (ICHN0). If this bit is reset, no address swapping takes place. If this bit is set, both Destination (DA) and Source address (SA) fields are swapped from FDDI bit ordering to canonical bit ordering. This involves a bit reversal within each byte.
D1	ISWP1	**INDICATE SWAP 1:** This bit controls the address swapping logic for Indicate Channel 1 (ICHN1). If this bit is reset, no address swapping takes place. If this bit is set, both Destination (DA) and Source address (SA) fields are swapped from FDDI bit ordering to canonical bit ordering. This involves a bit reversal within each byte.
D2	ISWP2	**INDICATE SWAP 2:** This bit controls the address swapping logic for Indicate Channel 2 (ICHN2). If this bit is reset, no address swapping takes place. If this bit is set, both Destination (DA) and Source address (SA) fields are swapped from FDDI bit ordering to canonical bit ordering. This involves a bit reversal within each byte.
D3	RSWP0	**REQUEST SWAP 0:** This bit controls the address swapping logic for Request Channel 0 (RCHN0). If this bit is reset, no address swapping takes place. If this bit is set, both Destination (DA) and Source address (SA) fields are swapped from canonical bit ordering to FDDI bit ordering. This involves a bit reversal within each byte.
D4	RSWP1	**REQUEST SWAP 1:** This bit controls the address swapping logic for Request Channel 1 (RCHN1). If this bit is reset, no address swapping takes place. If this bit is set, both Destination (DA) and Source address (SA) fields are swapped from canonical bit ordering to FDDI bit ordering. This involves a bit reversal within each byte.
D7–5	RES	**RESERVED**

Request Channel 0 and 1 Configuration Registers 1 (R0CR1 and R1CR1)

The two Request Configuration Registers 1 (R0CR1 and R1CR1) are reserved for future use.

Access Rules

Address	Read	Write
1B−1Ch	N/A	N/A

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	RES	RES	RES	RES

Bit	Symbol	Description
D7−0	RES	**RESERVED**

Compare Register (CMP)

The Compare Register (CMP) is used in comparison with a write access of a conditional write register. The Compare Register is loaded on a read of any of the conditional event Attention Registers or by directly writing to it.

This register is not altered upon reset.

Access Rules

Address	Read	Write
1Fh	Always	Always

Register Bits

D7	D6	D5	D4	D3	D2	D1	D0
CMP7	CMP6	CMP5	CMP4	CMP3	CMP2	CMP1	CMP0

Bit	Symbol	Description
D7−0	CMP7−0	**COMPARE:** These bits are compared to bits D7−0 of the accessed register, and only the bits in the Attention Register that have the same current value as the corresponding bit in the Compare register will be updated with the new value.

5.3 POINTER RAM REGISTERS

Pointer RAM Registers contain pointers to all data and Descriptors manipulated by the BSI-2 device, namely, Input and Output Data Units, Input and Output Data Unit Descriptors, Request Descriptors, Confirmation Messages, and Pool Space Descriptors. Pointer RAM Registers are shown in *Figure 5-4.*

5.4 LIMIT RAM REGISTERS

The Limit RAM Registers are used by both the Indicate and Request machines. Limit RAM Registers contain data values that define how far the BSI-2 device may advance in each of its ten queues. The Limit RAM Registers do not define the wrap point for each queue which is fixed at either 1 kbytes or 4 kbytes. Limit RAM Registers are shown in *Figure 5-5.*

5.5 DESCRIPTORS

Descriptors are used to observe and control the operation of the BSI-2 device. They contain address, status, and control information about Indicate and Request operations. Descriptors are stored in lists and wrap-around queues in memory external to the BSI-2 device and accessed by the BSI-2 device via the ABus. Descriptors include the following: Input Data Unit Descriptors (IDUDs) specify the location, size, part, and status information for Input Data Units. Output Data Unit Descriptors (ODUDs) specify the location and size of Output Data Units. For multi-ODUD frames, they also specify which part of the frame is pointed to by the ODUD. Pool Space Descriptors (PSPs) describe the location and size of a region of memory space available for writing Indicate data. Request Descriptors (REQs) describe the location of a stream of Output Data Unit Descriptors and contain operational parameters. Confirmation Status Messages (CNFs) describe the result of a Request operation.

5.6 OPERATING RULES

Multi-Byte Register Ordering

When referring to multi-byte fields, byte 0 is always the most significant byte. When referring to bits within a byte, bit 7 is the most significant bit and bit 0 is the least significant bit. When referring to the contents of a byte, the most significant bit is always referred to first.

5.7 POINTER RAM REGISTER DESCRIPTIONS

The Pointer RAM Register set contains 32, 28-bit registers. Registers 23 through 31 are reserved, and user access of these locations produces undefined results.

Pointer RAM Registers are read and written by the host using the Pointer RAM Operation (PTOP) service function and are accessed directly by BSI-2 device hardware during Indicate and Request operations. After initialization, the Pointer RAMRegisters are maintained by the BSI-2 device and do not require host intervention.

During Indicate and Request operations, Pointer RAM registers are used as addresses for ABus accesses of data and Descriptors, i.e., the subchannel addresses for loads (reads) of streams of PSPs, ODUs, ODUDs, and REQs, and for stores (writes) of streams of IDUs, IDUDs, and CNFs.

Pointer RAM Registers include the following:

ODU Pointer: Contains the address of an Output Data Unit. During Request operations, this register is loaded by the BSI-2 device from the Location Field of its Output Data Unit Descriptor.

ODUD List Pointer: Loaded by the BSI-2 device from the Location Field of the REQ Descriptor when it is read from memory. The address is incremented by the BSI-2 device as each ODUD is fetched from memory.

CNF Queue Pointer: Contains the current CNF Status Queue address. This register is written by the user after he has allocated space for the CNF Queue. During Request operations, this register is incremented by the BSI-2 device after each CNF is written to the CNF Queue.

REQ Queue Pointer: Initialized by the host with the start address of the REQ Descriptor Queue after the Queue has been initialized. During Request operations, the address is incremented by the BSI-2 device as each REQ is fetched.

IDU Pointer: Written by the BSI-2 device with the Location Field of the PSP Descriptor when it is loaded from the PSP pre-fetch register.

IDUD Queue Pointer: Points to the Queue location where IDUDs will be stored. Written by the user after he has allocated space for the IDUD Status Queue. Incremented by the BSI-2 device as IDUDs are written to consecutive locations in the Queue.

PSP Queue Pointer Register: Points to the next available PSP. Initialized by the host with the start address of the PSP Queue. As each PSP is read from memory, this register is incremented.

Next PSP Register: Written by the BSI-2 device with the PSP fetched from the PSP Queue.

Indicate Shadow Register: Written by the BSI-2 device with the start address of the last IDU copied to memory.

Request Shadow Register: Written by the BSI-2 device with the address of the current ODUD.

See *Figure 5-4* for Summary including address and access rules.

5.8 LIMIT RAM REGISTER DESCRIPTIONS

The Limit RAM Register set contains 16, 9-bit registers. Registers 11 through 15 are reserved, and access of these locations produces undefined results.

The Limit RAM registers contain data values that define the limits of each of the ten queues maintained by the BSI-2 device.

Limit RAM Registers are read and written by the host using the Limit RAM Operation (LMOP) service function when the Status/Space Machine is in STOP Mode, and are read directly by BSI-2 device hardware during Indicate and Request operations.

5.0 Control Information (Continued)

Limit RAM Registers include the following:

REQ Queue Limit: Defines the last valid REQ written by the host.

CNF Queue Limit Register: Defines the last Queue location where a CNF may be written by the BSI-2 device. Due to pipelining, the BSI-2 device may write up to two CNFs after it detects a write to the next-to-last CNF entry (and generates a No Status Space Attention). For this reason, the host must always define the CNF queue limit to be one Descriptor less than the available space.

IDUD Queue Limit Register: Defines the last Queue location where an IDUD may be written by the BSI-2 device.

PSP Queue Limit: Defines the last valid PSP written by the host.

See *Figure 5-5* for Summary including address and access rules.

			Access Rules	
Group	Address	Register Name	Read	Write
POINTER RAM	00	ODU Pointer RCHN1 (OPR1)	Always	Always
	01	ODUD List Pointer RCHN1 (OLPR1)	Always	Always
	02	CNF Queue Pointer RCHN1 (CQPR1)	Always	Always
	03	REQ Queue Pointer RCHN1 (RQPR1)	Always	Always
	04	ODU Pointer RCHN0 (OPR0)	Always	Always
	05	ODUD List Pointer RCHN0 (OLPR0)	Always	Always
	06	CNF Queue Pointer RCHN0 (CLPR0)	Always	Always
	07	REQ Queue Pointer RCHN0 (RQPR0)	Always	Always
	08	IDU Pointer ICHN2 (IPI2)	Always	Always
	09	IDUD Queue Pointer ICHN2 (IQPI2)	Always	Always
	0A	PSP Queue Pointer ICHN2 (PQPI2)*	Always	Always
	0B	Next PSP ICHN2 (NPI2)	Always	Always
	0C	IDU Pointer ICHN1 (IPI1)	Always	Always
	0D	IDUD Queue Pointer ICHN1 (PQPI1)	Always	Always
	0E	PSP Queue Pointer ICHN1 (PQPI1)*	Always	Always
	0F	Next PSP ICHN1 (NPI1)	Always	Always
	10	IDU Pointer ICHN0 (IPI0)	Always	Always
	11	IDUD Queue Pointer ICHN0 (IQPI0)	Always	Always
	12	PSP Queue Pointer ICHN0 (PQPI0)*	Always	Always
	13	Next PSP ICHN0 (NPI0)	Always	Always
	14	IDUD Shadow Register (ISR)	Always	Always
	15	ODUD Shadow Register (OSR)	Always	Always
	16–1F	Reserved	N/A	N/A

*Note that bit postion D2 of these Pointer RAM Locations is always forced to a 1, (The first word of a PSP is not fetched).

FIGURE 5-4. Pointer RAM Registers

			Access Rules	
Group	Address	Register Name	Read	Write
LIMIT RAM	0	REQ Queue Limit RCHN1 (RQLR1)	Always	Always
	1	CNF Queue Limit RCHN1 (CQLR1)	Always	Always
	2	REQ Queue Limit RCHN0 (RQLR0)	Always	Always
	3	CNF Queue Limit RCHN0 (CQLR0)	Always	Always
	4	IDUD Queue Limit ICHN2 (IQLI2)	Always	Always
	5	PSP Queue Limit ICHN2 (PQLI2)	Always	Always
	6	IDUD Queue Limit ICHN1 (IQLI1)	Always	Always
	7	PSP Queue Limit ICHN1 (PQLI1)	Always	Always
	8	IDUD Queue Limit ICHN0 (IQLI0)	Always	Always
	9	PSP Queue Limit ICHN0 (PQLI0)	Always	Always
	A–F	Reserved	N/A	N/A

FIGURE 5-5. Limit RAM Registers

5.0 Control Information (Continued)

Input Data Unit Descriptor (IDUD)

Input Data Unit Descriptors (IDUDs) are generated on Indicate Channels to describe where the BSI-2 device wrote each frame part and to report status for the frame.

For multi-part IDUDs, intermediate status is written in each IDUD, and when a status event occurs, definitive status is written in the last IDUD.

A detailed description of the encodings of the Indicate Status bits is given in *Figure 5-6*.

31 30	29 28	27 24	23 16	15	14 13	12 0	
IS		FRA	FRS	VC	RES	CNT	**Word 0**
F−L	RES	LOC					**Word 1**

Word 0

Bit	Symbol	Description
D12−0	CNT	**BYTE COUNT:** Number of bytes in the IDU to which this IDUD points. This count includes the FDDI Frame Check Sequence (4 byte FCS) but it does not include the three FC pad bytes which are written.
D14−13	RES	**RESERVED**
D15	VC	**VCOPY:** Reflects the state of the VCOPY signal sent to the BMAC device for this frame. 0: VCOPY was negated 1: VCOPY was asserted
D23−16	FRS	**FRAME STATUS:** This C, E, and A fields are valid only if the frame ended with an ED.
D17−16	C	**C INDICATOR:** 00: none 01: R 10: S 11: T
D19−18	A	**A INDICATOR:** 00: none 01: R 10: S 11: T
D21−20	E	**E INDICATOR:** 00: none 01: R 10: S 11: T
D22	VFCS	**VALID FCS:** 0: FCS field was invalid 1: FCS field was valid
D23	VDL	**VALID DATE LENGTH:** 0: Data length was invalid 1: Data length was valid

5.0 Control Information (Continued)

Word 0 (Continued)

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th></tr>
<tr><td>D27–24</td><td>FRA</td><td><b>FRAME ATTRIBUTES:</b> This field gives termination and address information.</td></tr>
<tr><td>D25–24</td><td>TC</td><td><b>TERMINATION CONDITION:</b>
00: Other (e.g., MAC Reset/token).
01: ED
10: Format error
11: Frame stripped</td></tr>
<tr><td>D26</td><td>AFLAG</td><td><b>AFLAG:</b> Reflects the state of the AFLAG input signal, which is sampled by the BSI-2 device at INFORCVD.
0: External DA match
1: Internal DA match</td></tr>
<tr><td>D27</td><td>MFLAG</td><td><b>MFLAG:</b> Reflects the state of the MFLAG input signal, which is sampled by the BSI-2 device at INFORCVD.
0: Frame sent by another station
1: Frame sent by this station</td></tr>
<tr><td>D31–28</td><td>IS</td><td><b>INDICATE STATUS:</b> The values in this field are prioritized, with the highest number having the highest priority. A detailed description of the encodings are given in <i>Figure 5-6.</i>
<table>
<tr><th>IS3</th><th>IS2</th><th>IS1</th><th>IS0</th><th>Meaning</th></tr>
<tr><td>D31</td><td>D30</td><td>D29</td><td>D28</td><td></td></tr>
<tr><td colspan="5"><b>Non-End Frame Status</b></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Last IDUD of queue, page-cross.</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Page boundary crossed.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>End of header.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>Page-cross with header-end.</td></tr>
<tr><td colspan="5"><b>Normal-End Frame Status</b></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Intermediate (no breakpoints).</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>Burst boundary.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>Threshold.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>Service opportunity.</td></tr>
<tr><td colspan="5"><b>Copy Abort Due to No Space</b></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>No data space.</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>No header space.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>Good header, info not copied.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>Not enough info space.</td></tr>
<tr><td colspan="5"><b>Error</b></td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>FIFO overrun.</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>Bad frame (no VDL or no VFCS).</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>Parity error.</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>Internal error.</td></tr>
</table>
</td></tr>
</table>

Word 1

Bit	Symbol	Description
D27–0	LOC	**LOCATION:** 28-bit memory address of the start of an IDU. For the first IDU of a frame, the address is of the fourth FC byte of the burst-aligned frame (i.e., bits [1:0] = 11). For subsequent IDUs, the address is of the first byte of the IDU (i.e, bits [1:0] = 00).
D29–28	RES	**RESERVED**
D31–30	F-L	**FIRST/LAST TAG:** Identifies the IDU object part, i.e., Only, First, Middle, or Last. FL=10=First, FL=00=Middle, FL=01=Last, FL=11-Only.

5.0 Control Information (Continued)

NON-END FRAME STATUS	
[0000]	**Last IDUD of Queue, with a Page Cross:** The last available location of the ICHN's IDUD queue was written. Since there was a page cross, there was more data to be written. Since there was no more IDUD space, the remaining data was not written. Note that this code will not be written in an IDUD.Middle, so that a Zero IS field with Zero F-L tags can be used by software as a null descriptor.
[0001]	**Page Cross:** Must be an IDUD.First or IDUD.Middle. This is part of a frame that filled up the remainder of the current page, requiring a new page for remainder of the data.
[0010]	**Header End:** This refers to the last IDU of the header portion of a frame.
[0011]	**Page Cross and Header End:** The occurrence of a page cross and header end.
NORMAL-END FRAME STATUS	
[0100]	**Intermediate:** A frame ended normally, and there was no breakpoint.
[0101]	**Burst Boundary:** A frame ended normally, and there was a breakpoint because a burst boundary was detected.
[0110]	**Threshold:** The copied frame threshold counter was reached when this frame was copied, and the frame ended normally.
[0111]	**Service Opportunity:** This (normal end) frame was preceeded by a token or MACRST, a MAC frame was received, or there was a ring-op change. Any of these events marks a burst boundary.
NO SPACE COPY ABORT	
[1000]	**Insufficient Data Space:** Not all the frame was copied because there was insufficient data space. This code is only written in non-Header/Info Sort Mode.
[1001]	**Insufficient Header Space:** The frame copy was aborted because there was insufficient header space (in Header/Info Sort Mode).
[1010]	**Successful Header Copy, Frame Info Not Copied:** There was sufficient space to copy the header, but insufficient data space to copy info, or insufficient IDU space (on ICHN2), or both. No info was copied.
[1011]	**No Info Space:** The frame's header was copied. When copying the data, there was insufficient data and/ or IDU space.
ERROR	
[1100]	**FIFO Overrun:** The Indicate FIFO had an overrun while copying this frame. This exception is caused when the memory interface does not allow the BSI-2 device to empty the data FIFO as quickly as it is being filled. This frame should not be processed because data has been lost.
[1101]	**Bad Frame:** This exception is caused when the incoming frame contains an invalid data length (too short or an odd number of symbols), or an invalid Frame Check Sequence (FCS). This implies that the frame was not a valid FDDI frame. Therefore, this frame should not be processed.
[1110]	**Parity Error:** This exception is caused when the BSI-2 device detects a parity error at the BSI-BMAC interface (the MR.FLOW bit must be set to enable parity checking). This implies a data corruption error within the frame. Therefore, this frame should not be processed.
[1111]	**Internal Error:** This exception is caused when the BSI-2 device detects an internal hardware error (e.g. illegal state machine state), in the receive logic while receiving a frame. This implies that the frame data may have been corrupted. Therefore, this frame should not be processed. In addition, the BSI-2 device should be reset and reinitialized.

FIGURE 5-6. Indicate Status Field (IS) of IDU Descriptor

REQ Descriptor (REQ)

Request Descriptors (REQs) contain the part, byte address, and size of one or more Output Data Unit Descriptors. They also contain parameters and commands to the BSI-2 device associated with Request operations.

Multiple REQ Descriptors (parts) may be grouped as one Request Descriptor object by the host software, with the REQ.First defining the parameters for the entire Request object. Also multiple Output Data Unit Descriptors may be grouped contiguously, to be described by a single REQ Descriptor.

Each REQ part is fetched by the BSI-2 device from the Request Channel's REQ Descriptor Queue, using the REQ Queue Pointer Register. Each Request Channel processes one Request Object (REQ.Only or REQ.First to REQ.Last set), per service opportunity.

The BSI-2 device checks for the following inconsistencies when the REQ is loaded from memory:

1. REQ.First with invalid Confirmation Class (as shown in the Table 5-8).
2. REQ.First with Request Class = 0.
3. REQ.First, when the previous REQ was not a REQ.Last or REQ.Only.
4. REQ which is not a REQ.First or REQ.Only when the previous REQ was a REQ.Last or a REQ.Only.

When an inconsistency is detected, the BSI-2 device aborts the Request, and reports the exception in the Request Status field of the CNF Descriptor.

The encodings of the RQCLS and CNFCLS bits are described in more detail in *Figure 5-7* and *Figure 5-8* respectively.

31 30	29 28	27 24	23 16	15 12	11 8	7 0	
RES	UID		SIZE	CNFCLS	RQCLS	FC	**Word 0**
F-L	RES	LOC					**Word 1**

Word 0

Bit		Symbol	Description
D7–0		FC	**FRAME CONTROL:** This specifies the Frame control field to be used unless FC transparency is enabled. This field is decoded to determine whether to assert RQCLM or RQBCN. This decoding is always active, i.e., regardless of frame control transparency. This field is also used for comparing received frames when confirming (without FC transparency).
D18–11		RQCLS	**REQUEST/RELEASE CLASS:** This field encodes the Request Class for the entire Request object, and is thus only sampled on a REQ.First or REQ.Only. The field is asserted on the RQRCLS output signals to the BMAC device when requesting a token. If the Request Class is incompatible with the current ring state, the BSI-2 device sets the RCHN's USR bit in the Request Attention Register. The encoding of this field is shown in *Figure 5-7.*
D15–12		CNFCLS	**CONFIRMATION CLASS:** This field encodes the Confirmation Class for the entire Request object, and is only sampled on a REQ.First or REQ.Only. The encoding of this field is shown in *Figure 5-8.*
	D12	E	**END:** Enables confirmation on completion of request. 0: CNFs on completion disabled. 1: CNFs on completion enabled.
	D13	I	**INTERMEDIATE:** Enables Intermediate (at the end of each Service Opportunity) Confirmation. 0: Intermediate CNFs disabled. 1: Intermediate CNFs enabled.
	D14	F	**FULL/TRANSMITTER:** Selects between Transmitter and Full Confirmation. 0: Transmitter confirm. 1: Full confirm.
	D15	R	**REPEAT:** Enables repeated transmission of the first frame of the request until the request is aborted. This may be used when sending BEACON or CLAIM frames. 0: Fetch all frames of REQ. 1: Repeat transmission of first frame of REQ. A Request may use Repeat on RCHN1, and have a Request loaded on RCHN0, but not vice-versa. Specifically, when a Request with the Repeat option is loaded on RCHN0, RCHN1 must not have any REQs active or visible to the BSI-2 device. Thus REQs on RCHN1 may be queued externally but the queue's Limit Register must not be set at or after that point. Requests with the Repeat option should only be used on one Request Channel at a time, and preferably on RCHN0. The Repeat option will only work on a REQ. First.

5.0 Control Information (Continued)

Word 0 (Continued)

Bit	Symbol	Description
D23–16	SIZE	**SIZE:** Count of number of frames represented by the ODUD stream pointed to by REQ.LOC field. Descriptors with a null frame count are permitted, and are typically used to end a Request, without having to send data. For example, to end a restricted dialogue, a REQ.Last with SIZE = 0 will cause the Request Machine to command the BMAC device to capture and release the specified classes of token. The response of the BSI-2 device to REQs with SIZE = 0 is as follows: 1. REQ.First: BSI-2 device latches the REQ Descriptor fields, then fetches the next REQ. RQRCLS is asserted, but RQRDY remains deasserted. 2. REQ.Middle: BSI-2 device fetches the next REQ. 3. REQ.Only: BSI-2 device requests the capture of the appropriate token. When it is captured, the BSI-2 asserts RQFINAL and ends the request. 4. REQ.Last: BSI-2 device captures the token, asserts RQFINAL, then marks the request complete.
D29–24	UID	**USER IDENTIFICATION:** Contains the UID field from the current REQ.First or REQ.Only.
D31–30	RES	**RESERVED**

Word 1

Bit	Symbol	Description
D27–0	LOC	**LOCATION:** Bits [27:2] are the memory word address of the ODUD stream. Bits [1:0] are expected to be 00, and are not checked.
D29–28	RES	**RESERVED**
D31–30	F-L	**FIRST/LAST TAG:** Identifies the REQ stream part, i.e., Only, First, Middle, or Last. FL=10=First, FL=00=Middle, FL=01=Last, FL=11-Only.

RQCLS Value	RQCLS Name	Class Type	THT	Token Capture	Token Issue	Notes
0000	None	None	-	none	none	
0001	Apr1	Async pri1	E	non-r	non-r	
0010	Reserved	Reserved				
0011	Reserved	Reserved				
0100	Syn	Sync	D	any	capt	1
0101	Imm	Immed	D	none	none	4
0110	ImmN	Immed	D	none	non-r	4
0111	ImmR	Immed	D	none	restr	4
1000	Asyn	Async	E	non-r	non-r	
1001	Rbeg	Restricted	E	non-r	restr	2, 3
1010	Rend	Restricted	E	restr	non-r	2
1011	Rcnt	Restricted	E	restr	restr	2
1100	AsynD	Async	D	non-r	non-r	
1101	RbegD	Restricted	D	non-r	restr	2, 3
1110	RendD	Restricted	D	restr	non-r	2
1111	RcntD	Restricted	D	restr	restr	2

E = enabled, D = disabled, non-r = non-restricted, restr = restricted, capt = captured

Note 1: Synchronous Requests are not serviced when bit BCNR of the Ring Event Latch Register is set.

Note 2: Restricted Requests are not serviced when bit BCNR, CLMR, or OTRMAC of the Ring Event Latch Register is set.

Note 3: Restricted Dialogues only begin when a Non-Restricted token has been received and transmitted.

Note 4: Immediate Requests are serviced when the ring is Non-Operational. These requests are serviced from the Data state unless the Request contains a Beacon or Claim FC. If a Claim FC is used, Immediate Requests are serviced from the Claim State. If a Beacon FC is used, Immediate Request are serviced from the Beacon State.

FIGURE 5-7. REQ Descriptor Request Class Encoding

[R]	[F]	[I]	[E]	Confirmation Class
x	0	0	0	Invalid (consistency failure)
x	x	1	0	Invalid (consistency failure)
0	1	0	0	None: Confirmation only on exception
0	0	0	1	Trend: Transmitter confirm, CNF on exception or completion
0	0	1	1	Tint: Transmitter confirm, CNF on exception, completion, or intermediate
0	1	0	1	Fend: Full Confirm, CNF on exception or completion
0	1	1	1	Fint: Full Confirm, CNF on exception, completion, or intermediate
1	1	0	0	NoneR: Confirmation only on exception, repeat frame
1	0	0	1	TendR: Transmitter confirm, CNF on exception or completion, repeat frame
1	0	1	1	TintR: Transmitter confirm, CNF on exception, completion, or intermediate, repeat frame
1	1	0	1	FendR: Full confirmation, CNF on exception or completion, repeat frame
1	1	1	1	FintR: Full Confirmation, CNF on exception, completion, or intermediate, repeat frame

FIGURE 5-8. REQ Descriptor Confirmation Class Field Encodings

Output Data Unit Descriptor (ODUD)

An Output Data Unit Descriptor (ODUD) contains the part, byte address and size of an Output Data Unit. During Request operations, ODUDs are fetched by the BSI-2 device from a list in memory, using the address in the ODUD List Pointer Register (in the Pointer RAM).

ODUD.Firsts and ODUD.Middles may have a zero byte count, which is useful for fixed protocol stacks. One layer may be called, and if it has no data to add to the frame, it may add an ODUD with a zero byte count to the list. ODUD.Onlys and ODUD.Lasts may not have a zero byte count.

The BSI-2 device checks for the following inconsistencies when an ODUD is loaded from memory:

1. ODUD.First, when previous ODUD was not an ODUD.Last or ODUD.Only.
2. ODUD which is not an ODUD.First, when the previous ODUD was ODUD.Last or ODUD.Only.
3. ODUD.Last or ODUD.Only with a zero byte count.

When an inconsistency is detected, the BSI-2 device aborts the Request, and reports the exception in the Request Status field of the CNF Descriptor.

The entire ODUD object must contain at least 4 bytes (for short addresses).

31 30	29 28	27 13	12 0	
RES			CNT	**Word 0**
F-L	RES	LOC		**Word 1**

Word 0

Bit	Symbol	Description
D12–0	CNT	**BYTE COUNT:** Number of bytes in the ODU. For an ODUD.First or ODUD.Middle the size may be Zero, which is useful for fixed protocol stacks.
D31–13	RES	**RESERVED**

Word 1

Bit	Symbol	Description
D27–0	LOC	**LOCATION:** Pointer to the first byte of the corresponding ODU.
D29–28	RES	**RESERVED**
D31–30	F-L	**FIRST/LAST TAG:** Identifies the Output Data Unit part, i.e., Only, First, Middle, or Last. FL = 10 = First, FL = 00 = Middle, FL = 01 = Last, FL = 11-Only.

Confirmation Status Message Descriptor (CNF)

A Confirmation Status Message (CNF) describes the result of a Request operation.

A more detailed description of the encoding of the RS bits is given in *Figure 5-9.*

<table>
<tr><th>31 30</th><th>29 28</th><th>27 24</th><th>23 16</th><th>15 8</th><th>7 0</th><th></th></tr>
<tr><td colspan="2">RS</td><td>FRA</td><td>FRS</td><td>TFC</td><td>CFC</td><td>Word 0</td></tr>
<tr><td>F-L</td><td colspan="2">UID</td><td>FC</td><td>CS</td><td>RES</td><td>Word 1</td></tr>
</table>

Word 0

Bit	Symbol	Description
D7−0	CFC	**CONFIRMED FRAME COUNT:** Number of confirmed frames. Valid only for Full Confirmation. This count is cummulative for Fint.
D15−13	TFC	**TRANSMITTED FRAME COUNT:** Number of frames successfully transmitted by the BSI-2 device and BMAC device. Valid for all confirmation classes. This count is cummulative for Tint and Fint.
D23−16	FRS	**FRAME STATUS:** This field is valid only for Full Confirmation, and if the frame ended with an ED.
D17−16	C	**C INDICATOR:** 00: none 01: R 10: S 11: T
D19−18	A	**A INDICATOR:** 00: none 01: R 10: S 11: T
D21−20	E	**E INDICATOR:** 00: none 01: R 10: S 11: T
D22	VFCS	**VALID FCS:** 0: FCS field was invalid 1: FCS field was valid
D23	VDL	**VALID DATE LENGTH:** 0: Data length was invalid 1: Data length was valid
D27−24	FRA	**FRAME ATTRIBUTES:** This field is valid only for Full Confirmation.
D25−24	TC	**TERMINATION CONDITION:** 00: Other (e.g., MAC Reset/token). 01: ED 10: Format error. 11: Frame stripped.
D26	AFLAG	**AFLAG:** Reflects the state of the AFLAG input signal, which is sampled by the BSI-2 device at INFORCVD. 0: No DA Match. 1: DA Match.
D27	MFLAG	**MFLAG:** Reflects the state of the MFLAG input signal, which is sampled by the BSI-2 device at INFORCVD. 0: Frame Sent by another station. 1: Frame Sent by this station.

5.0 Control Information (Continued)

Word 0 (Continued)

Bit	Symbol	Description
D31–28	RS	**REQUEST STATUS:** This field represents a priority encoded status value, with the highest number having the highest priority. This field is described in *Figure 5-9.*

	RS3	**RS2**	**RS1**	**RS0**	**Meaning**
	D31	D30	D29	D28	
Intermediate					
	0	0	0	0	None
	0	0	0	1	Preempted
	0	0	1	0	Part Done
Breakpoints					
	0	0	1	1	Service Loss
	0	1	0	0	Reserved
Completion					
	0	1	0	1	Completed BEACON
	0	1	1	0	Completed OK
Exception Completion					
	0	1	1	1	Bad Confirmation
	1	0	0	0	Underrun
	1	0	0	1	Host Abort
	1	0	1	0	Bad Ringop
	1	0	1	1	MAC Abort
	1	1	0	0	Timeout
	1	1	0	1	MAC Reset
	1	1	1	0	Consistency Failure
Error					
	1	1	1	1	Internal or Fatal ABus Error

Word 1

Bit	Symbol	Description
D7–0	RES	**RESERVED**
D15–8	CS	**CONFIRMATION STATUS**
D9–8	FT	**FRAME TYPE:** This field reflects the type of frame that ended Full Confirmation. 00: Any Other. 01: Token. 10: Other Void. 11: My Void.
D10	F	**FULL CONFIRM:** This bit is set when the Request was for Full Confirmation.
D11	U	**UNEXPECTED FRAME STATUS:** This bit is set when the frame status does not match the value programmed in the Request Expected Frame Status Register. This applies only to Full Confirmation.
D12	P	**PARITY:** This bit is set when a parity error is detected ina received frame. Parity is checked from FC to ED inclusive if the FLOW bit in the Mode Register is set.
D13	E	**EXCEPTION:** This bit is part of the BSI-2's hierarchical status reporting. It is set when an exception occurs during confirmation. An exception is any one of the nine error or exception codes described in the RS Field. The RCHN's EXC bit in the Request Attention Register is also set.
D14	R	**RING-OP:** This bit is set when the ring changes operational state after transmission but before all returning frames have been confirmed.
D15	T	**TRANSMIT CLASS:** 0: Restricted. 1: Non-Restricted.
D23–16	FC	**FRAME CONTROL:** Frame Control field of the last frame of the last confirmed burst. Valid only for Full Confirmation.
D29–24	UID	**USER IDENTIFICATION:** Contains the UID field copied from the current REQ.First or REQ.Only.
D31–30	F-L	**FIRST/LAST TAG:** Identifies the CNF part, i.e., Only, First, Middle, or Last. FL = 10 = First, FL = 00 = Middle, FL = 01 = Last, FL = 11-Only.

5.0 Control Information (Continued)

INTERMEDIATE	
[0000]	**NONE:** Non status is written. This may be used by software to identify a NULL or invalid CNF.
[0001]	**PREEMPTED:** RCHN1 was preempted by RCHN0. RCHN1 will be serviced following RCHN0.
[0010]	**PART NONE:** The BSI-2 device is servicing a Request, but it cannot hold onto a token, and the last frame of a Request.part has been transmitted.
BREAKPOINTS	
[0011]	**SERVICE LOSS:** The THT expired during a Request with THT enabled. Only Occurs for Intermediate Confirmation.
[0100]	**RESERVED**
COMPLETION	
[0101]	**COMPLETED BEACON:** When transmitting from the BEACON state, this status is returned when the BMAC device receives a My_Beacon. When transmitting from the CLAIM state, this status is returned when the BMAC device wins the CLAIM process.
[0110]	**COMPLETED OK:** Normal completion with good status.
EXCEPTION COMPLETION	
	In all of the exception and error cases it is likely that at least some of the frames from the associated request were not transmitted properly. Therefore, retransmission may be required. In the case of bad confirmation [0111], the frames may have been transmitted properly but lost on the ring. A consistency failure [1110] means that there is a problem in the request queues. It is recommended that they be reinitialized. The Internal or Abus error code [1111] is very severe and it recommended that the BSI-2 device be reinitialized.
[0111]	**BAD CONFIRMATION:** This status is reported when there was an error during confirmation. For confirmation, the BSI-2 compares the returning frame to the Expected Frame Status (EFS). If these values do not match, the ‘‘Bad Confirmation’’ value is returned in the RS field. If the transmitted frame does not return, (My_Void, Other_Void, or Token received instead) or if the ring state changes, (MAC Reset or the Ring_Operational flag changes), the Bad Confirmation value is also returned.
[1000]	**UNDERRUN:** This exception is caused when the memory interface does not allow the BSI-2 device to fill the transmit data FIFO as quickly as it is being emptied. It implies that the frame was aborted during transmission.
[1001]	**HOST ABORT:** This exception is caused when the host software clears the SAR.ABT bit to force an abort or when there is not enough space in the confirmation (CNF) queue. This implies that the Request did not complete normally.
[1010]	**BAD RINGOP:** This exception is reported when the Request Class for a Request object is incompatible with the current ring state, (i.e. Immediate class with an operational ring or Async, Sync, or restricted class when the ring state is non-operational). The Request was aborted.
[1011]	**MAC DEVICE ABORT:** The expection indicates that the BMAC device aborted the Request and asserted TXABORT. This could be from an interface parity error, or because the transmitted frame failed the FC check, or because the BMAC device received a MAC frame while transmitting in the DATA state. This status is also returned when the BMAC device receives an Other_Beacon while the BSI-2 device is transmitting in the BEACON state, or when the CLAIM process is lost while the BSI-2 device is transmitting in the CLAIM state. It implies that the Request did not complete normally.
[1100]	**TIMEOUT:** This exception code indicates that the TRT timer expired during the transmission of a Request with THT disabled. Normally the BMAC will finish the current frame and release the Token when the Token Holding Timer (THT) expires. However, for certain requests, the THT can be disabled. In this case, the Token Rotation Timer (TRT) may expire because the station has made the Token Late. The BMAC will abort the request and a Timeout will be signalled to the BSI.
[1101]	**MAC RESET:** This code indicates that the BMAC underwent a MAC Reset during this request. A MAC Reset can be generated by software, (i.e. requested via the control bus), or caused by hardware, (the BMAC state machines entered and illegal state). In either case the Request is aborted.
[1110]	**CONSISTENCY FAILURE:** This code indicates that the BSI-2 detected an inconsistency in the REQ or ODUD descriptor queues. For example, if a frame started with on ODUD.First it should be followed by an ODUD.Middle or ODUD.Last. If the next ODUD was another ODUD.First this would be a consistency error. The Request is aborted when a consistency error is detected.
ERROR	
[1111]	**INTERNAL OR FATAL ABUS ERROR:** This exception is caused when the BSI-2 device detects an internal hardware error (e.g. illegal state machine state), in the transmit logic while transmitting a frame. It is also set when an ABus error occurs during frame transmission. It implies that the frame data may not have been transmitted properly.

FIGURE 5-9. Request Status Field (RS) of CNF Descriptor

5.0 Control Information (Continued)

Pool Space Descriptor (PSP)

Pool Space Descriptors (PSPs) contain the address of a free space in host memory available for writing Input Data Units. The count field is not used. The space is assumed to end at the next 4 kbyte boundary. When PSPs are read by the BSI-2 device, the address field of the PSP is loaded into the Indicate Channel's IDU Pointer Register, and is used as the address for the IDU memory write.

<table>
<tr><th>31 30</th><th>29 28</th><th>27 13</th><th>12 0</th><th></th></tr>
<tr><td colspan="3">RES</td><td>CNT</td><td>Word 0</td></tr>
<tr><td>F-L</td><td>RES</td><td colspan="2">LOC</td><td>Word 1</td></tr>
</table>

Word 0

Bit	Symbol	Description
D12–0	CNT	**BYTE COUNT:** Number of bytes of available memory area (this field is currently not used by the BSI-2). To ensure software compatibility with future devices which may use this field, this field may be written with the number of bytes from PSP.LOC to the next 4 kbyte boundary.
D31–13	RES	**RESERVED**

Word 1

Bit	Symbol	Description
D27–0	LOC	**LOCATION:** Memory byte address of memory area available for writing IDUs. Normally the page offset will be Zero to simplify space management. Must be burst aligned to the size of the largest burst enabled (4 word or 8 word).
D29–28	RES	**RESERVED**
D31–30	F-L	**FIRST/LAST TAG:** Identifies the PSP part, should be PSP.Only (i.e., F-L = 11).

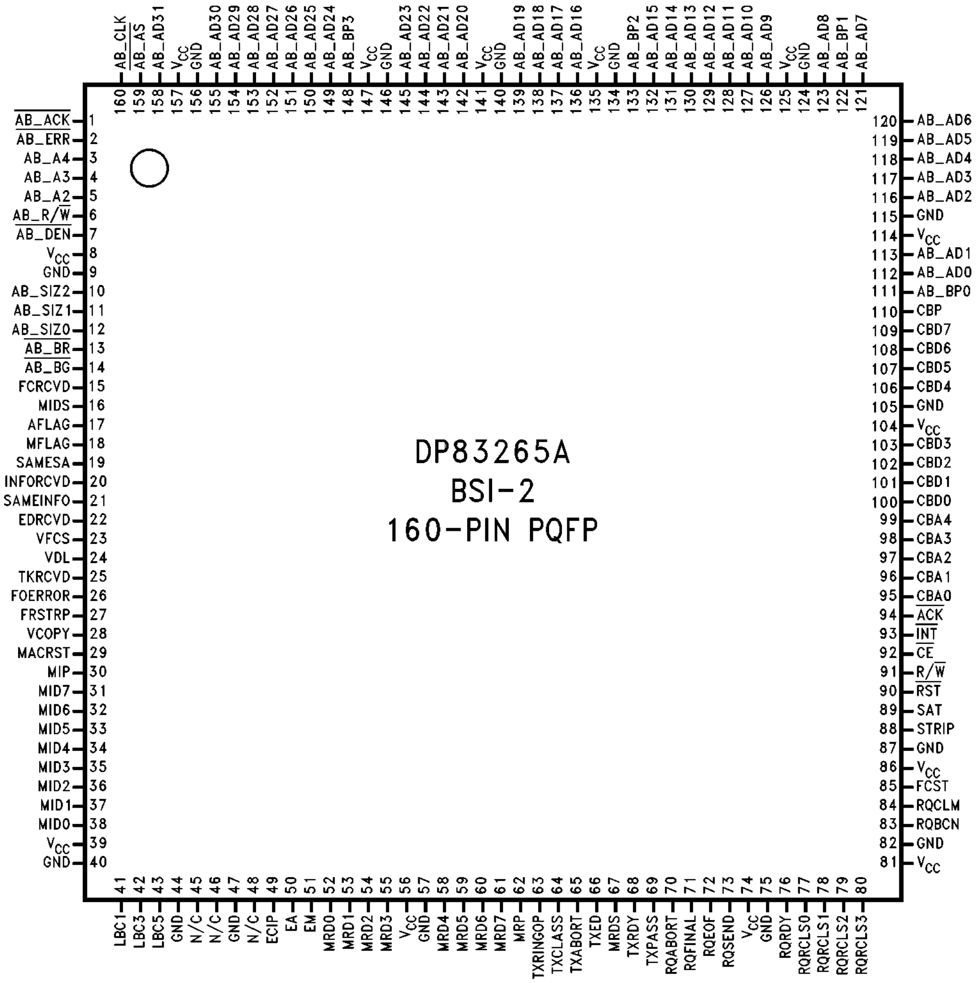
6.0 Signal Descriptions

Pin Table

Pin	Description	I/O	Pin	Description	I/O	Pin	Description	I/O	Pin	Description	I/O
1	$\overline{\text{AB\_ACK}}$	I	41	LBC1	I	81	V_{CC}	Core	121	AB_AD7	I/O
2	$\overline{\text{AB\_ERR}}$	I	42	LBC3	I	82	GND	Core	122	AB_BP1	I/O
3	AB_A4	O	43	LBC5	I	83	RQBCN	O	123	AB_AD8	I/O
4	AB_A3	O	44	GND		84	RQCLM	O	124	GND	
5	AB_A2	O	45	NC		85	FCST	O	125	V_{CC}	
6	AB_R$\overline{\text{W}}$	O	46	NC		86	V_{CC}		126	AB_AD9	I/O
7	$\overline{\text{AB\_DEN}}$	O	47	GND		87	GND		127	AB_AD10	I/O
8	V_{CC}		48	NC		88	STRIP	O	128	AB_AD11	I/O
9	GND	O	49	ECIP	I	89	SAT	O	129	AB_AD12	I/O
10	AB_SIZ2	O	50	EA	I	90	$\overline{\text{RST}}$	I	130	AB_AD13	I/O
11	AB_SIZ1	O	51	EM	I	91	R$\overline{\text{W}}$	I	131	AB_AD14	I/O
12	AB_SIZ0	O	52	MRD0	O	92	$\overline{\text{CE}}$	I	132	AB_AD15	I/O
13	$\overline{\text{AB\_BR}}$	O	53	MRD1	O	93	$\overline{\text{INT}}$	OD	133	AB_BP2	I/O
14	$\overline{\text{AB\_BG}}$	I	54	MRD2	O	94	$\overline{\text{ACK}}$	OD	134	GND	
15	FCRCVD	I	55	MRD3	O	95	CBA0	I	135	V_{CC}	
16	MIDS	I	56	V_{CC}		96	CBA1	I	136	AB_AD16	I/O
17	AFLAG	I	57	GND		97	CBA2	I	137	AB_AD17	I/O
18	MFLAG	I	58	MRD4	O	98	CBA3	I	138	AB_AD18	I/O
19	SAMESA	I	59	MRD5	O	99	CBA4	I	139	AB_AD19	I/O
20	INFORCVD	I	60	MRD6	O	100	CBD0	I/O	140	GND	Core
21	SAMEINFO	I	61	MRD7	O	101	CBD1	I/O	141	V_{CC}	Core
22	EDRCVD	I	62	MRP	O	102	CBD2	I/O	142	AB_AD20	I/O
23	VFCS	I	63	TXRINGOP	I	103	CBD3	I/O	143	AB_AD21	I/O
24	VDL	I	64	TXCLASS	I	104	V_{CC}		144	AB_AD22	I/O
25	TKRCVD	I	65	TXABORT	I	105	GND		145	AB_AD23	I/O
26	FOERROR	I	66	TXED	I	106	CBD4	I/O	146	GND	
27	FRSTRP	I	67	MRDS	I	107	CBD5	I/O	147	V_{CC}	
28	VCOPY	O	68	TXRDY	I	108	CBD6	I/O	148	AB_BP3	I/O
29	MACRST	I	69	TXPASS	I	109	CBD7	I/O	149	AB_AD24	I/O
30	MIP	I	70	RQABORT	O	110	CBP	I/O	150	AB_AD25	I/O
31	MID7	I	71	RQFINAL	O	111	AB_BP0	I/O	151	AB_AD26	I/O
32	MID6	I	72	RQEOF	O	112	AB_AD0	I/O	152	AB_AD27	I/O
33	MID5	I	73	RQSEND	O	113	AB_AD1	I/O	153	AB_AD28	I/O
34	MID4	I	74	V_{CC}		114	V_{CC}		154	AB_AD29	I/O
35	MID3	I	75	GND		115	GND		155	AB_AD30	I/O
36	MID2	I	76	RQRDY	O	116	AB_AD2	I/O	156	GND	
37	MID1	I	77	RQRCLS0	O	117	AB_AD3	I/O	157	V_{CC}	
38	MID0	I	78	RQRCLS1	O	118	AB_AD4	I/O	158	AB_AD31	I/O
39	V_{CC}	Core	79	RQRCLS2	O	119	AB_AD5	I/O	159	$\overline{\text{AB\_AB}}$	I/O
40	GND	Core	80	RQRCLS3	O	120	AB_AD6	I/O	160	AB_CLK	I/O

6.0 Signal Descriptions (Continued)

Pin Diagram



TL/F/11407–1

6.0 Signal Descriptions (Continued)

6.1 CONTROL INTERFACE

The Control Interface operates asynchronously to the operation of the BMAC Interface and the ABus Interface.
The $\overline{\text{ACK}}$ and $\overline{\text{INT}}$ signals are open drain to allow wire ORing.

Symbol	Pin #	I/O	Description
CBP	110	I/O	**CONTROL BUS PARITY:** Odd parity on CBD7–0.
CBD7–0	109–106, 103–100	I/O	**CONTROL BUS DATA:** Bidirectional Data bus.
CBA4–0	99–95	I	**CONTROL BUS ADDRESS:** Address of a particular BSI-2 device register.
$\overline{\text{CE}}$	92	I	**CONTROL BUS ENABLE:** Handshake signal used to begin a Control Inter- face access. Active low signal.
R/$\overline{\text{W}}$	91	I	**READ/WRITE:** Determines current direction of a Control Interface access.
$\overline{\text{ACK}}$	94	OD	**ACKNOWLEDGE:** Acknowledges that the Control Interface access has been performed. Active low, open drain signal.
$\overline{\text{INT}}$	93	OD	**INTERRUPT:** Indicates presence of one or more enabled conditions. Active low, open drain signal.
$\overline{\text{RST}}$	90	I	**RESET:** Causes a reset of BSI-2 device state machines and registers.

6.0 Signal Descriptions (Continued)

6.2 BMAC Device Indicate Interface

The BMAC Device Indicate Interface signals provide data and control bytes as received from the BMAC device. Each Indicate Data byte is also provided with odd parity.

MID7−0 signals are valid on the rising edge of the Local Byte Clock signal (provided by the Clock Distribution Device).

Symbol	Pin #	I/O	Description
MIP	30	I	**CONTROL BUS PARITY:** This is connected directly to the corresponding BMAC device pin of similar name. Odd parity on MID7−0. Only valid with data and Status indicators.
MID7−0	31−38	I	**MAC INDICATE DATA:** These are connected directly to the corresponding BMAC device pins of similar name. **DATA:** The BMAC device indicates data is being presented on MID7-0 during the time when RCSTART is asserted until one of the following signals is asserted: EDRCVD, TKRCVD, FOERROR, or MACRST. **STATUS:** The BMAC device indicates Status Indicators are being presented on MID7−0 when EDRCVD or TKRCVD is asserted.

Frame Sequencing:

The Frame Sequencing signals apply to the data available at the MAC Indicate Interface (MIP and MID7-0). The Frame Sequencing signals can be used to control the latching of appropriate Frame Status.

Symbol	Pin #	I/O	Description
FCRCVD	15	I	**FRAME CONTROL RECEIVED:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the Frame Control Field has been received.
INFORCVD	20	I	**INFORMATION FIELD RECEIVED:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that four bytes of the information Field have been received. It is asserted by the BMAC device on the fourth byte of the INFO field and remains active until the next JK symbol pair is received.
EDRCVD	22	I	**EDFS RECEIVED:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the End of Frame Sequence has been received.
MIDS	16	I	**MAC INDICATE DATA STROBE:** Reserved for future functionality. This pin must be tied high.

Frame Information:

Symbol	Pin #	I/O	Description
AFLAG	17	I	**MY DESTINATION ADDRESS RECOGNIZED:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that an internal address match occurred on the Destination Address field. The BSI-2 uses this information when making a copy decision. It is reset when the next JK symbol pair is received.
MFLAG	18	I	**MY SOURCE ADDRESS RECOGNIZED:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the received Source Address field matched the MLA or MSA BMAC device registers. The BSI-2 uses this information when making a copy decision. It is reset when the next JK symbol pair is received.
SAMESA	19	I	**SAME SOURCE ADDRESS:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the SA of the current frame is the same as the previous frame, that the frames were not MAC frames, and that the addresses are the same length. The BSI-2 uses this to generate receive breakpoints. It is reset when the next JK symbol pair is received.

6.0 Signal Descriptions (Continued)

Frame Information (Continued)

Symbol	Pin #	I/O	Description
SAMEINFO	21	I	**SAME MAC INFORMATION:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the first four bytes of the INFO field of the current frame are the same as the previous frame, that the frames were MAC frames, and that their address lengths are the same. SAMEINFO is asserted along with INFORCVD. It is reset when the next JK symbol pair is received. The BSI-2 uses this status when doing MAC frame filtering.
VDL	24	I	**VALID DATA LENGTH:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates if the current frame had a valid data length. The BSI-2 provides this status to the user in the IDUD.Last for each frame.
VFCS	23	I	**VALID FRAME CHECK SEQUENCE:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates a valid FCS for the current frame. The BSI-2 provides this status to the user in the IDUD.Last for each frame.
TKRCVD	25	I	**TOKEN RECEIVED:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that a complete token was received. The BSI-2 uses this information for receive break- points and confirmation processing.
FRSTRP	27	I	**FRAME STRIPPED:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the current frame was stripped. The BSI-2 ignores frames stripped before the INFORCVD point. Frames stripped after this point (due to an error upstream) have an abnormal terminating status of ‘‘Frame Stripped’’ written in the IDUD. This signal may also affect confirmation.
FOERROR	26	I	**FORMAT ERROR:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates a standard- defined format error. The BSI-2 will use this status and indicate an abnormal terminating condition if it is attempting to copy this frame. This signal may also affect confirmation.
MACRST	29	I	**MAC RESET:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates an internal error, MAC frame, MAC reset, or hardware or software reset. The BSI-2 will use this status and indicate an abnormal terminating condition if it is attempting to copy this frame. This signal may also affect confirmation.
EA	50	I	**EXTERNAL AFLAG:** This signal is used by external address matching logic to signal that a Destination Address (DA) match has occurred. Assuming that the proper timing of EA and ECIP are met, the assertion of EA will cause the BSI-2 device to copy this frame. EA is sampled on the cycle after ECIP is deasserted. The sample window is from FCRCVD to EDRCVD.
EM	51	I	**EXTERNAL MFLAG:** This signal is used by external address matching logic to signal a Source Address (SA) match. It is sampled on the clock cycle after ECIP is deasserted.
VCOPY	28	O	**VALID COPY:** This is connected directly to the corresponding BMAC device pin of similar name. It affects the BMAC’s setting of the transmitted Cx (Copied Indicator). The BSI-2 asserts VCOPY when it attempts to copy a frame. This BSI-2 will not assert VCOPY if it is using the ‘‘promiscuous’’ copy mode.
ECIP	49	I	**EXTERNAL COMPARE IN PROGRESS:** This signal is asserted to indicate that external address comparison has begun. It is deasserted to indicate that the comparison has completed. EA and EM are sampled upon the deassertion of ECIP. ECIP must be asserted during the period from the assertion of FCRVCD (by the BMAC device) to the assertion of INFORCVD (by the BMAC device) in order for the BSI-2 to recognize an external comparison. It must be deasserted for at least one cycle for the external comparison to complete. If ECIP has not been deasserted before EDRCVD (from the BMAC device), the BSI-2 device will not copy this frame. ECIP may be implemented as a positive or negative pulse. Note that ECIP will affect the operation of the BSI-2 device even if the external copy mode is not specifically selected. See Section 4.3 for more details on the external matching interface.

6.0 Signal Descriptions (Continued)

6.3 BMAC Device Request Interface

The BMAC Device Request Interface signals provide data and control bytes to the BMAC device as received from the Host System. Each Request Data Byte is also provided with odd parity.

Symbol	Pin #	I/O	Description
MRP	62	O	**MAC REQUEST PARITY:** This is connected directly to the corresponding BMAC device pin of similar name. Odd parity on MRD7–0.
MRD7–0	61–58, 55–52	O	**MAC REQUEST DATA:** These are connected directly to the corresponding BMAC device pins of similar name. The BSI-2 device provides transmit (Request) data on MRD7–0.

Service Parameters:

Symbol	Pin #	I/O	Description
RQRCLS3–0	80–77	O	**REQUEST CLASS:** These are connected directly to the corresponding BMAC device pins of similar name. The BSI-2 device indicates the service class parameters for this request. These are obtained from the REQ.First descriptor for this Request object. When RQRCLS = 0, the BMAC device Transmitter will capture a usable token (for non-immediate requests) and assert TXRDY. The service opportunity continues as long as the token is usable with the current service parameters, even if RQRDY is not asserted. When RQRCLS = 0, the service opportunity will terminate after the current frame (even if RQRCLS subsequently becomes non-Zero).
RQCLM	84	O	**REQUEST Claim:** This is connected directly to the corresponding BMAC device pin of similar name. The BSI-2 device indicates that this request is to be serviced in the Transmit Claim state. Ignored for non-immediate request.
RQCBN	83	O	**REQUEST Beacon:** This is connected directly to the corresponding BMAC device pin of similar name. The BSI-2 device indicates that this request is to be serviced in the Transmit Beacon state. Ignored for nonimmediate request.

Frame Options:

Symbol	Pin #	I/O	Description
STRIP	88	O	**VOID STRIP:** Connected to STRIP and possibly SAT on the BMAC device.
SAT	89	O	**SOURCE ADDRESS TRANSPARENCY:** Connected to SAIGT on the BMAC device and to SAT on the BMAC device is STRIP is not.
FCST	85	O	**FRAME CHECK SEQUENCE TRANSPARENCY:** This is connected directly to the corresponding BMAC device pin of similar name. When the BSI-2 device asserts this signal, the BMAC device will not append FCS to the end of the Information field.

Request Handshake:

Symbol	Pin #	I/O	Description
TXPASS	69	I	**TRANSMIT PASS:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates the absence of a service opportunity. This could result from an unusable request class, waiting for a token, timer expiration, or MAC Reset. TXPASS is always asserted between service opportunities. It is deasserted when TXRDY is asserted at the beginning of a service opportunity.
TXRDY	68	I	**TRANSMIT READY:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the BMAC device transmitter is ready for another frame. For a non-immediate request, a usable token must be held in order to transmit frames. TXRDY is asserted by the BMAC device when: a. a usable token is being held, or b. an immediate request becomes serviceable, or c. after frame transmission if the current service opportunity is still usable for another frame. TXRDY is deasserted when TXPASS or TXACK ia asserted.

6.0 Signal Descriptions (Continued)

Request Handshake (Continued)

Symbol	Pin #	I/O	Description
RQRDY	76	O	**REQUEST READY:** This is connected directly to the corresponding BMAC device pin of similar name. The BSI-2 device indicates that the BMAC device transmitter should attempt to use a service opportunity. If RQRDY is asserted within 6 byte times after TXRDY is asserted, the BMAC device transmitter will wait at least L__Max plus one Void frame (4.16 μs–4.80 μs) for RQSEND to be asserted before releasing the token.
RQSEND	73	O	**REQUEST SEND:** This is connected directly to the corresponding BMAC device pin of similar name. The BSI-2 device indicates that the BMAC device transmitter should send the next frame. The MRD7–0 signals convey the FC byte when this signal is asserted. IF RQSEND is asserted within 6 byte times after TXRDY is asserted, the BMAC device transmitter will send the frame with a minimum length preamble. If RQSEND is not asserted within L__Max plus one Void frame after RQRDY has been asserted (4.16 μs–4.60 μs), the token may become unusable due to timer expiration.
MRDS	67	I	**MAC REQUEST DATA STROBE:** This signal should be connected to TXACK on the BMAC device. When this signal is asserted, it causes the BSI-2 to advance the Transmit (Request) FIFO pointer and present a new transmit data byte on MRD7–0.
RQEOF	72	O	**REQUEST EOF:** This is connected directly to the corresponding BMAC device pin of similar name. The BSI-2 device indicates that MRD7–0 conveys the last data byte when asserted. Normally, this is the last byte of the INFO field of the frame (exceptions: FCS transparency, invalid frame length).
RQABORT	70	O	**REQUEST ABORT:** This is connected directly to the corresponding BMAC device pin of similar name. The BSI-2 device indicates that the current frame should be aborted. Normally this causes the BMAC device transmitter to generate a Void frame.
RQFINAL	71	O	**REQUEST FINAL:** This is connected directly to the corresponding BMAC device pin of similar name. The BSI-2 device indicates that the final frame of the request has been presented to the BMAC device Interface. When asserted, the Issue Token Class (as opposed to the Capture Token Class) becomes the new Token Class (TXCLASS).
TXED	66	I	**TRANSMIT END DELIMITER:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the ED is being transmitted. The BSI-2 uses this information to determine that the entire frame was transmitted without error, (i.e. the ED was transmitted with no error signal being asserted).
TXABORT	65	I	**TRANSMIT ABORT:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates that the MAC Transmitter aborted the current frame. The BSI-2 uses this signal to detect an abnormal end to an attempted frame transmission.

Transmit Status

Symbol	Pin #	I/O	Description
TXRINGOP	63	I	**TRANSMIT RING OPERATIONAL:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates the state of the MAC Transmitter. The BSI-2 uses this signal to compare requested transmission class (RQRCLS) with the current ring state. An incompatible ring state will cause a Request to be aborted.
TXCLASS	64	I	**TRANSMIT TOKEN CLASS:** This is connected directly to the corresponding BMAC device pin of similar name. The BMAC device indicates the class of the current token. The BSI-2 uses this information in the confirmation descriptor.

6.0 Signal Descriptions (Continued)

6.4 ABus Interface

The ABus interface signals provide a 32-bit multiplexed address/data bus for transfers between the host system and the BSI-2 device. The ABus uses a bus request/bus grant protocol that allows for multiple bus masters, supports burst transfers of 4 or 8 32-bit words, and permits both physical and virtual addressing using fixed-size pages.

Address and Data

Symbol	Pin #	I/O	Description
AB_BP3–0	148, 133, 122, 111	I/O	**ABUS BYTE PARITY:** These TRI-STATE signals contain the parity for each address and data byte of AB_AD, such that AB_BP0 is the parity for AB_AD7–0, AB_BP1 is the parity for AB_AD15–8, etc.
AB_AD31–0	158, 155–149, 145–142, 139–136, 132–126, 123, 121–116, 113–112	I/O	**ABUS ADDRESS AND DATA:** These TRI-STATE signals are the multiplexed ABus address and data lines. During the address phase of a cycle, AB_AD27–0 contain the 28-bit address, and AB_AD31–28 contain a 4-bit function code identifying the type of transaction, encoded as follows: **AB_AD[31:28]** **Transaction Type** 0 RCHN1 ODU Load 1 RCHN1 ODUD Load/CNF Store 2 RCHN1 REQ Load 3 RCHN0 ODU Load 4 RCHN0 ODUD Load/CNF Store 5 RCHN0 REQ Load 6 ICHN2 IDU Store 7 ICHN2 IDUD Store 8 ICHN2 PSP Load 9 ICHN1 IDU Store A ICHN1 IDUD Store B ICHN1 PSP Load C ICHN0 IDU Store D ICHN0 IDUD Store E ICHN0 PSP Load F PTR RAM Load/Store
AB_A4–2	3–5	O	**ABUS BURST ADDRESS:** These TRI-STATE signals contain the word address during burst-mode accesses. They are driven from Tpa to the last Td state, negated in the following Tr state, then released. Note that the timing of these signals is under software control via Mode Register 1 (MR1).

6.0 Signal Descriptions (Continued)

Bus Control

<table>
<tr><th>Symbol</th><th>Pin #</th><th>I/O</th><th>Description</th></tr>
<tr><td>$\overline{\text{AB\_AS}}$</td><td>159</td><td>O</td><td><b>ABUS ADDRESS STROBE:</b> When first asserted, these TRI-STATE signals indicate that the address on AB_AD is valid. When this signal is inactive and $\overline{\text{AB\_ACK}}$ is asserted, the next cycle is a Tr state, in which the bus arbiter can sample all bus requests, then issue a bus grant in the following cycle. Note that the timing of this signal is under software control via Mode Register 1 (MR1).</td></tr>
<tr><td>AB_R/$\overline{\text{W}}$</td><td>6</td><td>O</td><td><b>ABUS READ/WRITE:</b> This TRI-STATE signal determines the current direction of an ABus access. A high level indicates a read access and a low level indicates a write access.</td></tr>
<tr><td>$\overline{\text{AB\_DEN}}$</td><td>7</td><td>I/O</td><td><b>ABUS DATA ENABLE:</b> In normal ABus mode, this TRI-STATE signal indicates that data on AB_AD31–0 is valid. In the enhanced ABus mode for SBus, this signal is an additional Acknowledgment input.</td></tr>
<tr><td>AB_SIZ2–0</td><td>10–12</td><td>O</td><td><b>ABUS SIZE:</b> These TRI-STATE signals indicate the size of the transfer on AB_AD31–0, encoded as follows:
<table>
<tr><th>AB_SIZ2</th><th>AB_SIZ1</th><th>AB_SIZ0</th><th>Transfer Size</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>4 Bytes</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Reserved</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>16 Bytes</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>32 Bytes</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>$\overline{\text{AB\_ACK}}$</td><td>1</td><td>I</td><td><b>ABUS ACKNOWLEDGE:</b> Indicates a bus slave's response to a bus master. The meaning of this signal depends on the state of ABus Error ($\overline{\text{AB\_ERR}}$) as well as the ABus mode selected (normal or enhanced). The exact function is described below.</td></tr>
<tr><td>$\overline{\text{AB\_ERR}}$</td><td>2</td><td>I</td><td><b>ABUS ERROR:</b> In normal ABus mode this signal is asserted by a bus slave to cause a transaction retry or transaction abort. In the enhanced ABus mode for SBus, this signal together with $\overline{\text{AB\_ACK}}$ and $\overline{\text{AB\_DEN}}$ encode the acknowledgment type. The encoding is as follows:
<table>
<tr><th colspan="2">EAM = 0</th><th colspan="3">EAM = 1</th><th rowspan="2">Function</th></tr>
<tr><th>$\overline{\text{AB\_ACK}}$</th><th>$\overline{\text{AB\_ERR}}$</th><th>$\overline{\text{AB\_ACK}}$
Ack(2)*</th><th>$\overline{\text{AB\_DEN}}$
Ack(1)*</th><th>$\overline{\text{AB\_ERR}}$
Ack(0)*</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>Wait Cycle</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>Word Acknowledgement</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>Retry</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Error</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>0</td><td>Not Supported</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>Not Supported</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td>Not Supported</td></tr>
<tr><td></td><td></td><td>1</td><td>0</td><td>1</td><td>Not Supported</td></tr>
</table></td></tr>
</table>

6.0 Signal Descriptions (Continued)

Bus Arbitration

Symbol	Pin #	I/O	Description
$\overline{\text{AB\_BR}}$	13	O	**ABUS BUS REQUEST:** This signal is used by the BSI-2 to request use of the ABus.
$\overline{\text{AB\_BG}}$	14	I	**ABUS GRANT:** This signal is asserted by external bus arbitration logic to grant use of the ABus to the BSI-2 device. If $\overline{\text{AB\_BG}}$ is asserted at the start of a transaction (Tbr), the BSI-2 device will run a transaction. Note that in normal ABus mode (MR1.EAM = 0), the BSI-2 may take up to two cycles to respond to $\overline{\text{AB\_BG}}$. Therefore, $\overline{\text{AB\_BG}}$ should not be removed until the BSI-2 has indicated that it has sampled $\overline{\text{AB\_BG}}$ and taken the bus, (this can be determined with $\overline{\text{AB\_AS}}$ for example).
AB_CLK	160	I	**ABUS CLOCK:** All ABus operations are synchronized to the rising edge of AB_CLK.

6.5 Electrical Interface

Symbol	Pin #	I/O	Description
LBC5, 3, 1	43–41	I	**LOCAL BYTE CLOCK:** 12.5 MHz clocks with a 50/50 duty-cycle, generated by CDD
VCC[13]	8, 39, 56, 74, 81, 86, 104, 114, 125, 135, 141, 147, 157		**POSITIVE POWER SUPPLY:** 5V, 10% relative to GND
GND[13]	9, 40, 57, 75, 82, 87, 105, 115, 124, 134, 140, 146, 156		**GROUND:** Power Supply Return
GND	44, 47		**GROUND:** Must be grounded.
NC	45,46,48		**NO CONNECT:** Must be left unconnected.

7.0 Electrical Characteristics

7.1 ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage		−0.5		7.0	V
DC_{IN}	Input Voltage		−0.5		V_{CC} + 0.5	V
DC_{OUT}	Output Voltage		−0.5		V_{CC} + 0.5	V
T_{STG}	Storage Temperature		−65		150	°C
T_L	Lead Temperature	Soldering, 10s (IR or Vapor) (Phase Reflow)			230	°C
	ESD Protection		2000			V

7.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage		4.75		5.25	V
T_A	Operating Temperature		0		70	°C
PD	Power Dissipation				800	mW

7.3 DC ELECTRICAL CHARACTERISTICS

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

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OH}	Output High Voltage	I_{OH} = −8 mA	2.4			V
V_{OL1}	Output Low Voltage	I_{OL} = 8 mA			0.4	V
V_{OL2}	Output Low Voltage for $\overline{\text{INT}}$ and $\overline{\text{ACK}}$ (open drain)	I_{OL} = 8 mA			0.4	V
V_{IH}	Input High Voltage		2.0			V
V_{IL}	Input Low Voltage				0.8	V
I_{IL}	Input Low Current	V_{IN} = GND			−10	μA
I_{IH}	Input High Current	V_{IN} = V_{CC}			+10	μA
I_{OZ1}	TRI-STATE Leakage				±10	μA
I_{OZ2}	TRI-STATE Leakage for $\overline{\text{INT}}$ and $\overline{\text{ACK}}$ (open drain)				±10	μA
I_{CC}	Dynamic Supply Current	CL = 50 pf, LBC = 12.5 MHz, AB__CLK = 33 MHz			150	mA

7.0 Electrical Characteristics (Continued)

7.4 AC ELECTRICAL CHARACTERISTICS

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

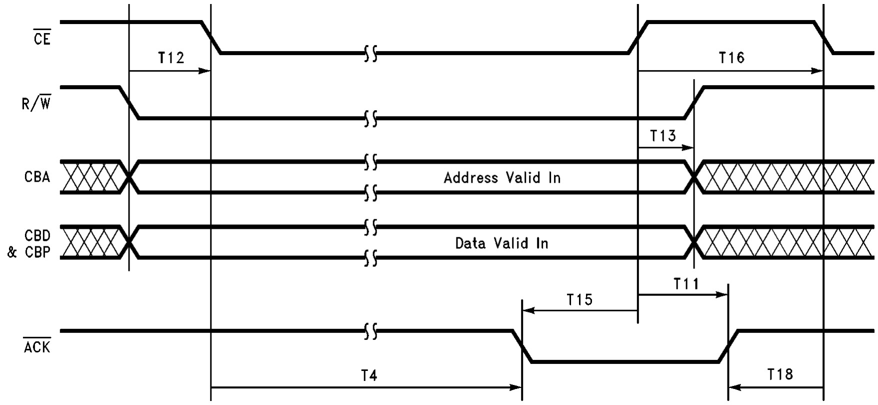
AC Characteristics for the Control Bus Interface

Symbol	Parameter	Min	Max	Units
T1	$\overline{\text{CE}}$ Setup to LBC	15		ns
T2	LBC Period	80		ns
T3	LBC1 to $\overline{\text{ACK}}$ Low		45	ns
T4	$\overline{\text{CE}}$ Low to $\overline{\text{ACK}}$ Low	290	540	ns
T5	LBC1 Low to CBD(7–0) and CBP Valid		60	ns
T6	LBC1 to CBD(7–0) and CBP Active	5		ns
T7	$\overline{\text{CE}}$ Low to CBD(7–0) and CBP Active	225	475	ns
T8	$\overline{\text{CE}}$ 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	$\overline{\text{CE}}$ High to $\overline{\text{ACK}}$ High		45	ns
T12	R/$\overline{\text{W}}$, CBA(7–0), CBD(7–0) and CBP Setup to $\overline{\text{CE}}$ Low	5		ns
T13	$\overline{\text{CE}}$ High to R/$\overline{\text{W}}$, CBA(7–0), CBD(7–0) and CBP Hold Time	0		ns
T14	R/$\overline{\text{W}}$, CBA(7–0), CBD(7–0) and CBP to LBC1 Setup Time	20		ns
T15	$\overline{\text{ACK}}$ Low to $\overline{\text{CE}}$ High Lead Time	0		ns
T16	$\overline{\text{CE}}$ Minimum Pulse Width High	20		ns
T17	$\overline{\text{CE}}$ High to CBD(7–0) and CBP TRI-STATE		55	ns
T18	$\overline{\text{ACK}}$ High to $\overline{\text{CE}}$ Low	0		ns
T19	CBD(7–0) Valid to $\overline{\text{ACK}}$ Low Setup	20		ns
T20	LBC1 to $\overline{\text{INT}}$ Low		55	ns

Asynchronous Definitions

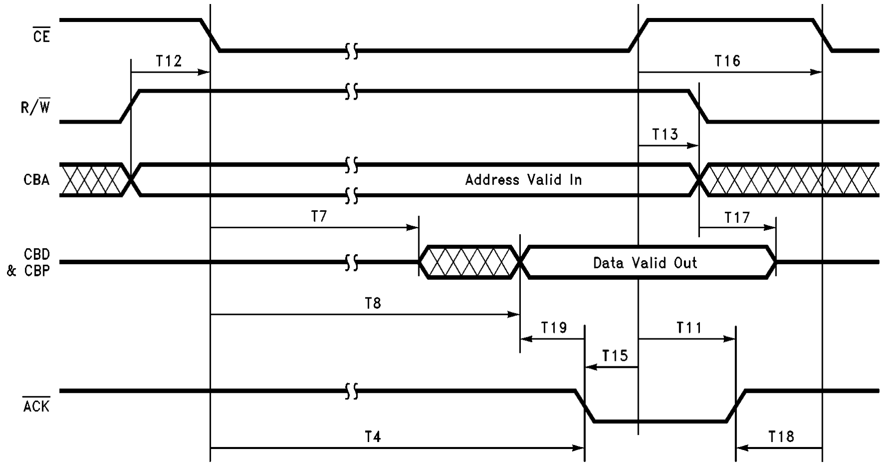
T4 (Min)	T1 + (3 * T2) + T3
T4 (Max)	T1 + (6 * T2)+ T3
T7 (Min)	T1 + (2 * T2) + T6
T7 (Max)	T1 + (5 * T2)+ T6
T8 (Min)	T1 +(2 * T2) + T9 + T5
T8 (Max)	T1 +(5 * T2) + T9 + T5

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



TL/F/11407–17

FIGURE 7-1. Asynchronous Control Bus Write Cycle Timing



TL/F/11407–18

FIGURE 7-2. Asynchronous Control Bus Read Cycle Timing

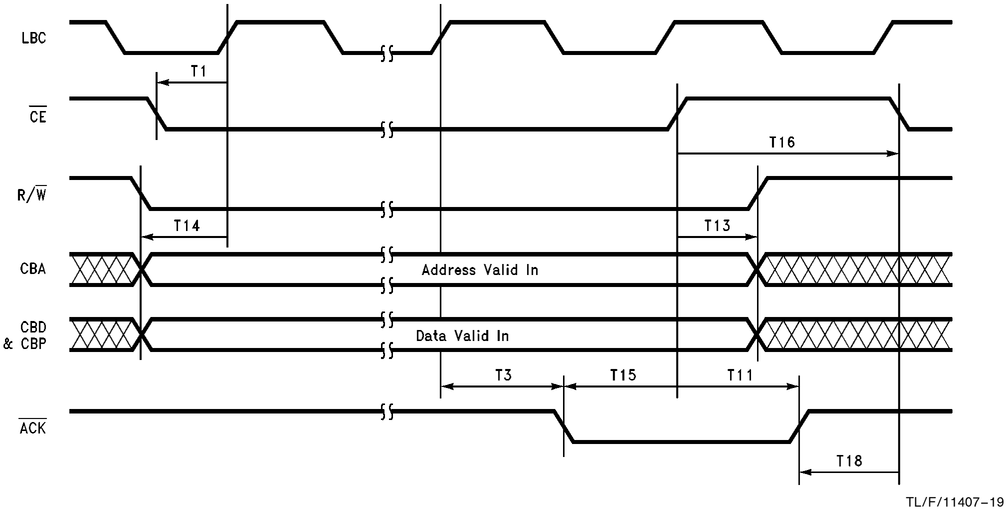


FIGURE 7-3. Control Bus Synchronous Write Cycle Timing

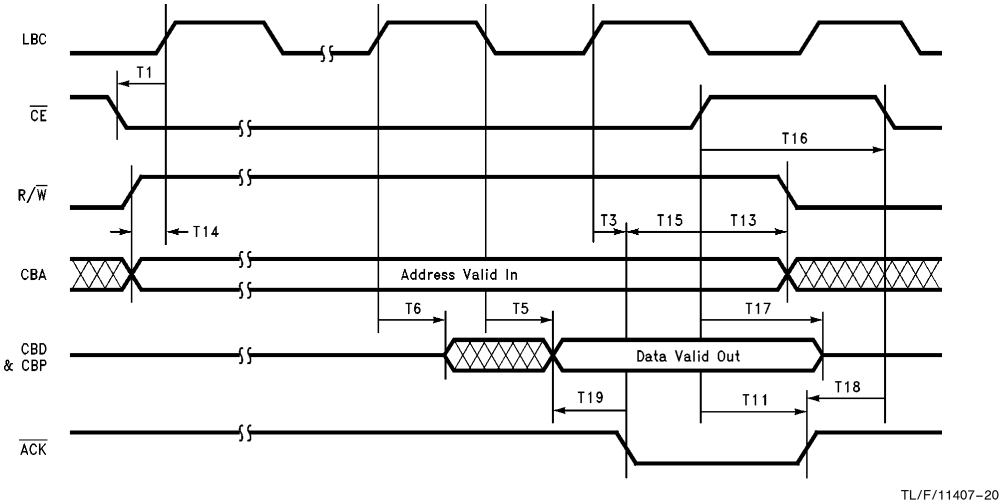
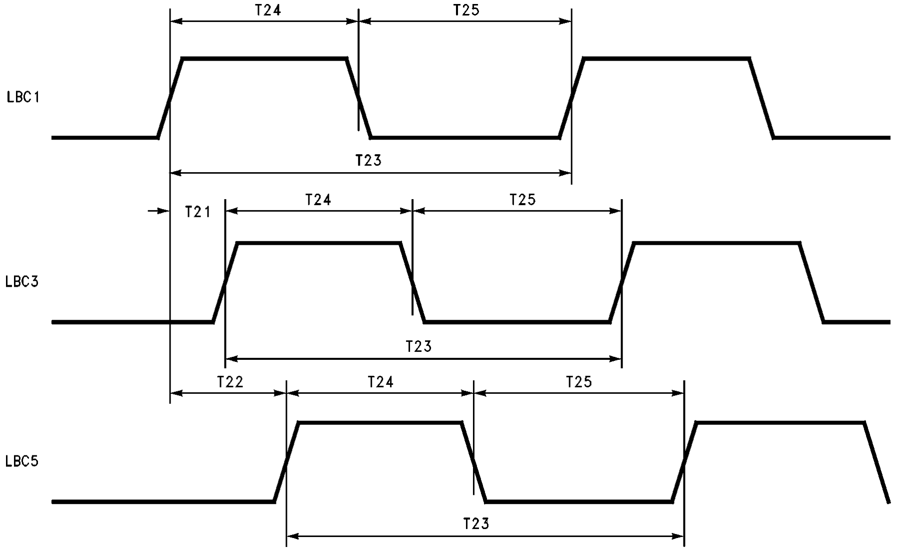


FIGURE 7-4. Control Bus Synchronous Read Cycle Timing

7.0 Electrical Characteristics (Continued)

AC Characteristics for the Clock Interface Signals

Symbol	Parameter	Min	Typ	Max	Units
T21	LBC1 to LBC3 Lead Time	11		19	ns
T22	LBC1 to LBC5 Lead Time	27		35	ns
T23	LBC1, LBC3, and LBC5 Period	80			ns
T24	LBC1, LBC3, and LBC5 Pulse Width High	35		45	ns
T25	LBC1, LBC3, and LBCS Pulse Width Low	35		45	ns



TL/F/11407–21

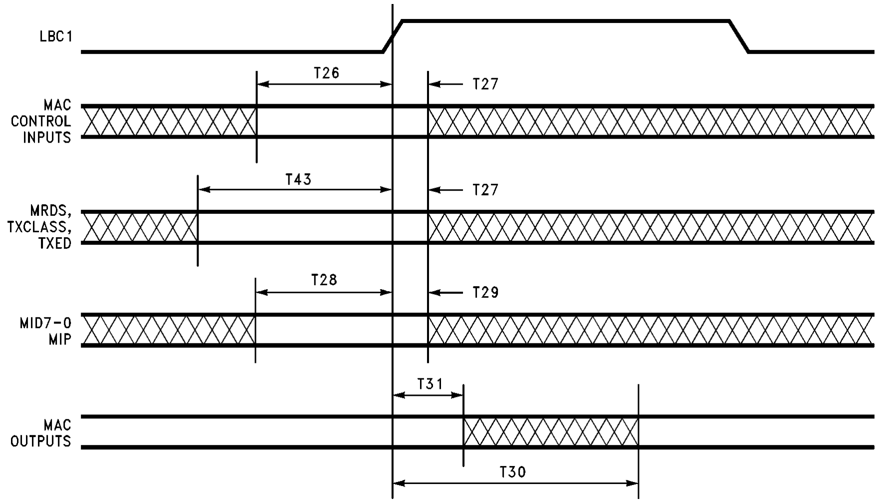
FIGURE 7-5. Clock Interface Timing Diagram

7.0 Electrical Characteristics (Continued)

AC Characteristics for Port A Interface and Port B Interface

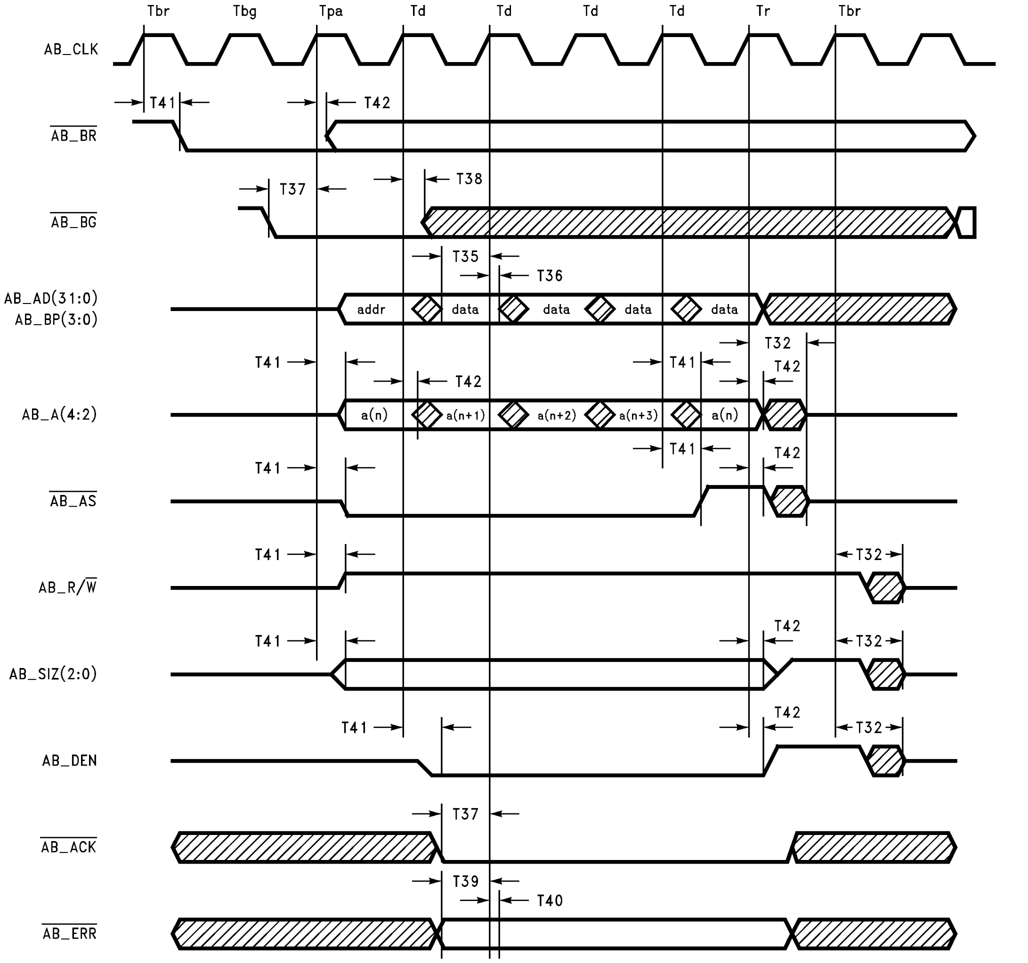
Symbol	Parameter	Min	Typ	Max	Units
T26A	MAC Control Inputs Setup to LBC1 (Except MRDS)	15			ns
T26B	MRDS Setup to LBC1	35			ns
T27	MAC Control Inputs Hold from LBC1	2			ns
T28	MAC Data Inputs Setup to LBC1	10			ns
T29	MAC Data Inputs Hold from LBC1	2			ns
T30	MAC Control and Data Outputs LBC1 to Data Valid			35	ns
T31	MAC Control and Data Outputs Sustain from LBC1	10			ns
T32	ABus Outputs AB_CLK to TRI-STATE			20	ns
T33A	AB_AD(31:0) Output, AB_CLK to Data Valid			17	ns
T33B	AB_BP Output, AB_CLK to Data Valid			20	ns
T34	AB_AD(31:0), AB_BP Output, Sustain from AB_CLK	5			ns
T35	AB_AD(31 :0), AB_BP Input, Setup from AB_CLK	12			ns
T36	AB_AD(31 :0), AB_BP, $\overline{\text{AB\_DEN}}$ Input, Hold from AB_CLK	1			ns
T37	$\overline{\text{AB\_ACK}}$, $\overline{\text{AB\_BG}}$, $\overline{\text{AB\_DEN}}$, Setup to AB_CLK	15			ns
T38	$\overline{\text{AB\_ACK}}$, $\overline{\text{AB\_BG}}$, Hold from AB_CLK	0			ns
T39	$\overline{\text{AB\_ERR}}$, Setup to AB_CLK	15			ns
T40	$\overline{\text{AB\_ERR}}$, Hold from AB_CLK	0			ns
T41	$\overline{\text{AB\_AS}}$, AB_SIZ(2:0), AB_R$\overline{\text{W}}$, $\overline{\text{AB\_DEN}}$, $\overline{\text{AB\_BR}}$, AB_A, Data Valid from AB_CLK			20	ns
T42	$\overline{\text{AB\_AS}}$, AB_SIZ(2:0), AB_R$\overline{\text{W}}$, $\overline{\text{AB\_DEN}}$, $\overline{\text{AB\_BR}}$, AB_A, Data sustain from AB_CLK	5			ns
F1	AB_CLK Frequency	12.5		25*	MHz

*Contact National for information regarding the DP83265AVF-33 (33 MHz) BSI-2 device.



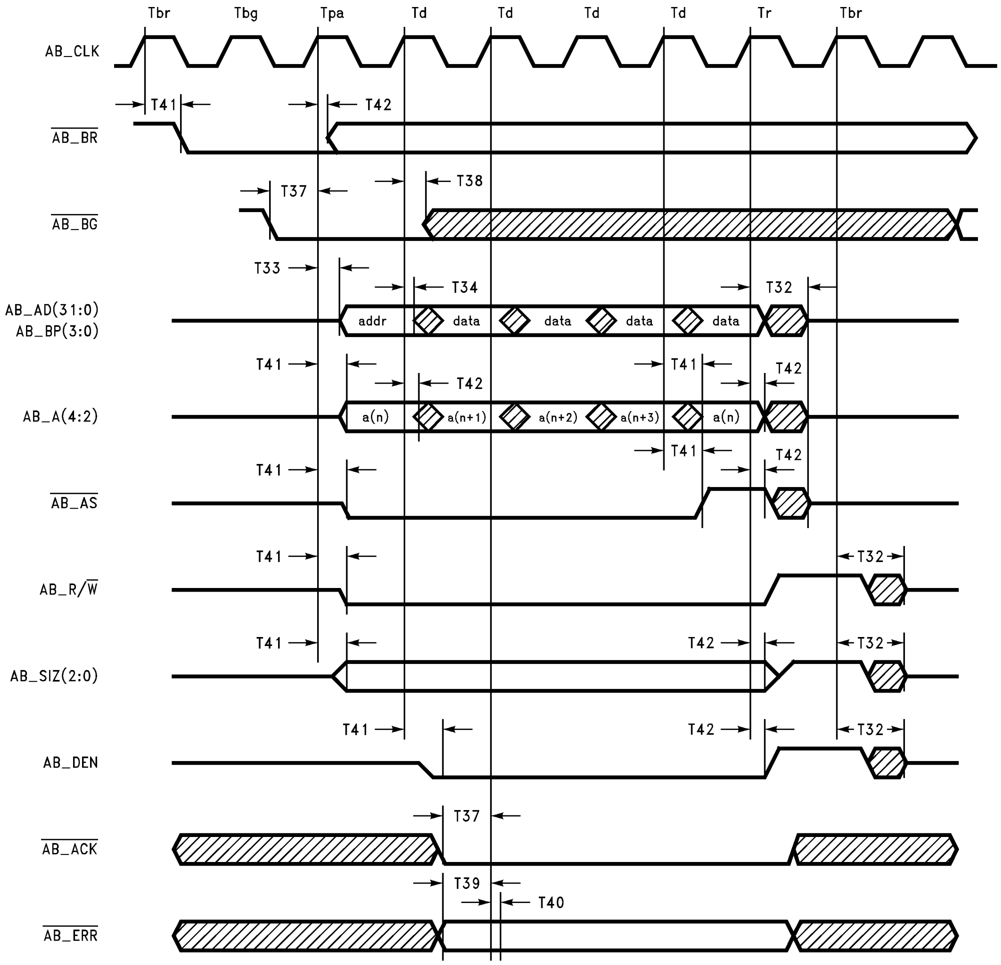
TL/F/11407–22

FIGURE 7-6. MAC Interface Timing



TL/F/11407–23

FIGURE 7-7a. ABus Read Cycle Timing Diagram

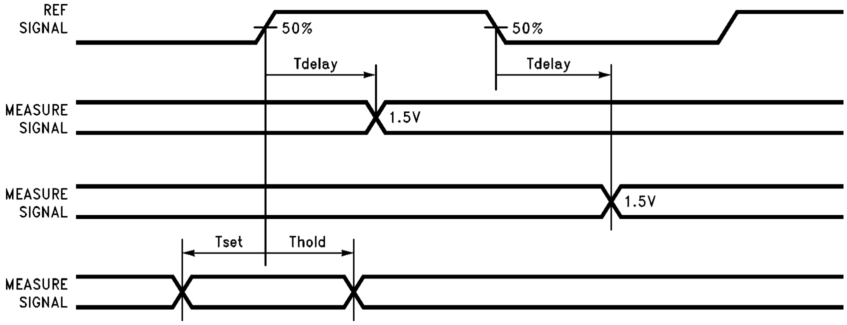


TL/F/11407–24

FIGURE 7-7b. ABus Write Cycle Timing Diagram

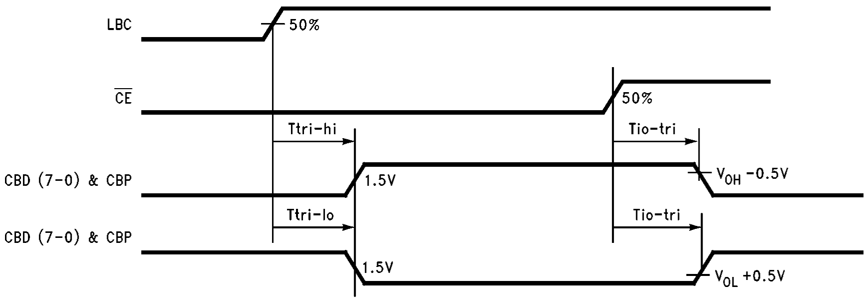
7.0 Electrical Characteristics (Continued)

AC Signal Testing



TL/F/11407–25

FIGURE 7-8. A.C. Signal Testing



TL/F/11407–26

FIGURE 7-9. TRI-STATE Timing

7.0 Electrical Characteristics (Continued)

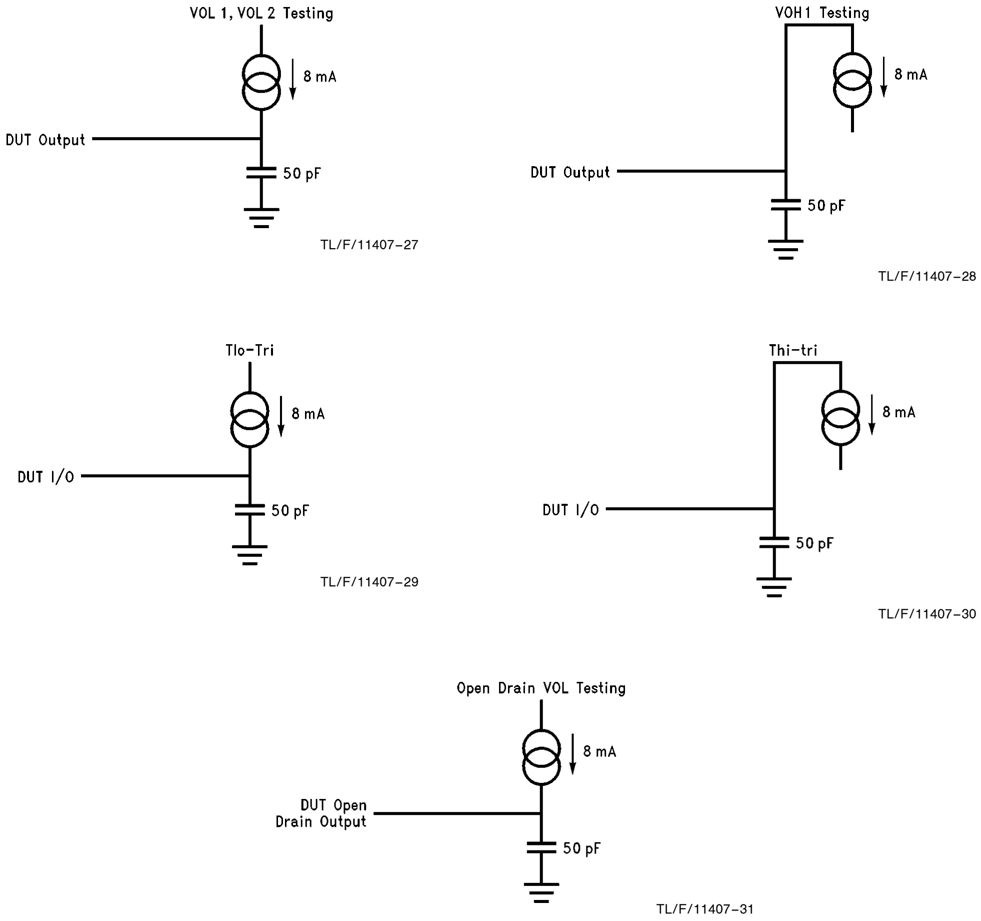
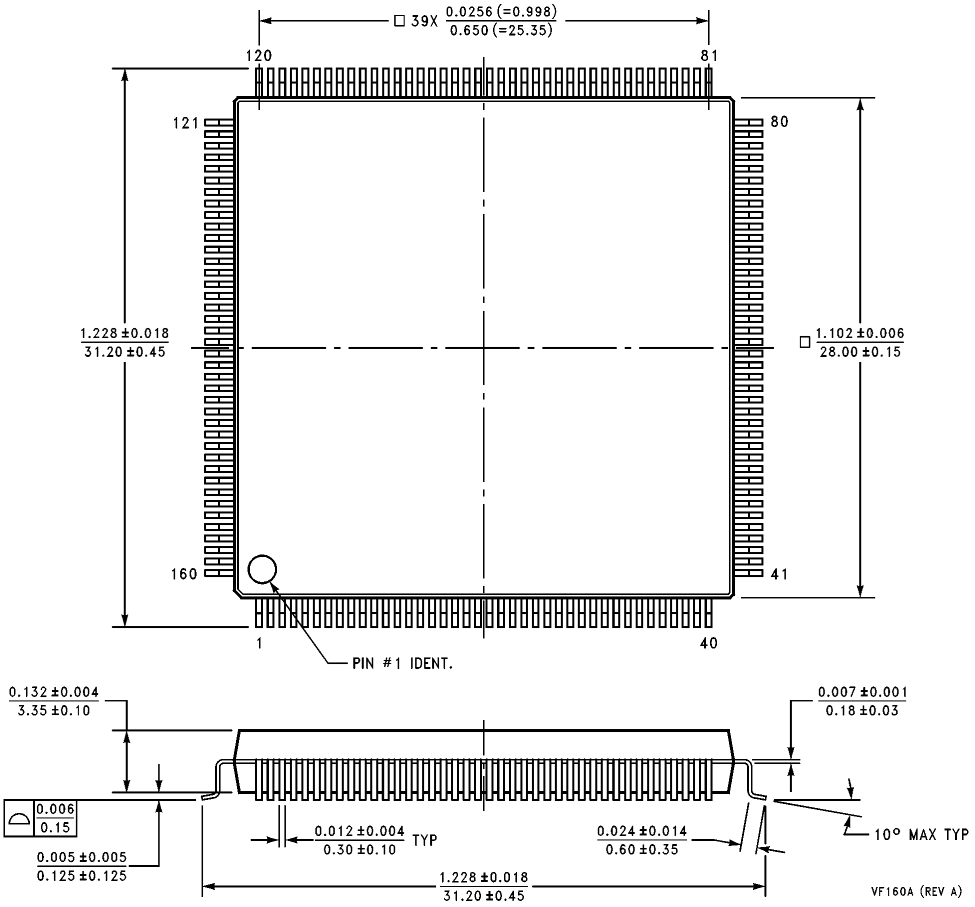


FIGURE 7-10. Test Equivalent Loads

DP83265A BSI-2 Device (FDDI System Interface)

Physical Dimensions $\frac{\text{inches}}{\text{millimeters}}$



160 Lead Plastic Quad Flat Pack
NS Package Number VF160A

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.
13th 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.