TECHNICAL MANUAL

1394 Node Controller Core

August 2001

Preliminary

LSI LOGIC®

Order #

This document is preliminary. As such, it contains data derived from functional simulations and performance estimates. LSI Logic has not verified either the functional descriptions, or the electrical and mechanical specifications using production parts.

This document contains proprietary information of LSI Logic Corporation. The information contained herein is not to be used by or disclosed to third parties without the express written permission of an officer of LSI Logic Corporation.

Document DB14-000075-02, Third Edition (August 2001)
This document describes the LSI Logic Corporation 1394 Node Controller Core and will remain the official reference source for all revisions/releases of this product until rescinded by an update.

LSI Logic Corporation reserves the right to make changes to any products herein at any time without notice. LSI Logic does not assume any responsibility or liability arising out of the application or use of any product described herein, except as expressly agreed to in writing by LSI Logic; nor does the purchase or use of a product from LSI Logic convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of LSI Logic or third parties.

TRADEMARK ACKNOWLEDGMENT
The LSI Logic logo design, CoreWare, G10, GigaBlaze, FlexStream, and Right-First-Time are trademarks or registered trademarks of LSI Logic Corporation. All other brand and product names may be trademarks of their respective companies.

EH

To receive product literature, visit us at http://www.lsilogic.com.

For a current list of our distributors, sales offices, and design resource centers, view our web page located at

http://www.lsilogic.com/contacts/na_salesoffices.html

Preface

This book is the primary reference and technical manual for the 1394 Node Controller core. It contains preliminary information regarding the functional description of the 1394 Node Controller core.

Audience

This document assumes that you have some familiarity with the IEEE 1394 high-performance serial bus and related support devices. The people who benefit from this book are:

- Engineers and managers who are evaluating the 1394 Node Controller core for possible use in a system
- Engineers who are designing the 1394 Node Controller core into a system

Organization

This document has the following chapters:

- Chapter 1, **Introduction**, summarizes the key features and applications of the 1394 Node Controller core.
- Chapter 2, **Data Formats**, describes the different data formats that the 1394 Node Controller core sends and receives.
- Chapter 3, **Signal Descriptions**, describes the signals that comprise the external interface of the 1394 Node Controller core.
- Chapter 4, **Registers**, describes the internal registers of the 1394 Node Controller core.
- Chapter 5, **Operation Overview**, provides functional waveforms, which show the operation of the core.

- Chapter 6, **Application Operation**, describes transmit and receive operations from the viewpoint of the application.

Related Publications

1394 Physical Layer (PHY) Core Technical Manual,
Document No. DB14-000036-01

IEEE Standard for a High Performance Serial Bus (IEEE Standard 1394-1995)

P1394a Draft Standard for a High Performance Serial Bus (Supplement)

Conventions Used in This Manual

1. All signals have the module name as prefix. For example:

   LLC_Lreq: This signal is part of the LLC module.

2. All signals ending with 'N' are LOW-asserted signals. For example:

   APP_RstN: Reset is an active-LOW signal.

Abbreviations:

ABUF	Asynchronous Buffer Block
APPIF	Application Interface Module
AV/C	Audio Video Control (IEC-61883)
CHF	CIP Header Field (IEC-61883)
CIP	Common Isochronous Packet (IEC-61883)
CMP	Connection Management Procedures (IEC-61883)
CSR	Command and Status Register (ISO/IEC-13213)
CSU	Control and Status Unit
CRC	Cyclic Redundancy Check code
DBUF	DMA Buffer Block
DVCR	Digital Video Cassette Recorder
EOH	End Of CIP Header (IEC-61883)

FCP	Function Control Protocol (IEC-61883)
IEC-61883	Standard for consumer digital AV equipment using 1394
IMPR	Input Master Plug Register
IPCR	Input Plug Control Register (IEC-61883)
LLC	Link Layer Controller
MPEG	Motion Picture Experts Group
OPCR	Output Plug Control Register
OMPR	Output Master Plug Register
ORB	Operation Request Block
PCR	Plug Control Register
PHT	Packet Header Transformation
SBP-2	Serial Bus Protocol 2
SPH	Source Packet Header (IEC-61883)
UBUF	Universal Asynchronous Buffer
URF	Universal Receive FIFO
UTF	Universal Transmit FIFO

The first time a word or phrase is defined in this manual, it is *italicized.*

The word *assert* means to drive a signal true or active. The word *deassert* means to drive a signal false or inactive.

Hexadecimal numbers are indicated by the prefix “0x” —for example, 0x32CF. Binary numbers are indicated by the prefix “0b” —for example, 0b0011.0010.1100.1111.

Contents

Figures

Tables

Chapter 1
Introduction

This chapter contains the following sections:

The 1394 Node Controller core conforms with the requirements of the IEEE 1394-1995 Standard for High Performance Serial Bus and the 1394a supplement. It provides full link layer functionality and easily interfaces to the LSI Logic 1394 Physical Layer Core or any other standard 1394a-compliant PHY device.

The core uses a DMA interface to reduce CPU utilization for large data transfers. The core can process incoming 1394 read/write requests to a programmable 4 Gbyte range without CPU intervention using DMA transfers. This feature is useful when the core is used in SBP-2 (serial bus protocol) initiator applications or DPP (Direct Print Protocol) receiver mode. The core can generate automatic and sequential 1394 read/write requests. This feature is useful when the core is used in SBP-2 target applications or DPP sender mode.

Isochronous transfers use the DMA interface. CIP (common isochronous packet) headers can be generated automatically when transmitting. The core supports hardware-assisted transmit data flow control (for example, DV empty CIP packet generation). The core also supports simultaneous transmission and reception of one IEC-61883 format talk stream and two listen streams.

For unformatted streams (1394 tag equals zero), the core supports any combination of simultaneous talk and listen channels.

When the core is a 1394 cycle master, it supports external 1394 cycle timer control by using the cycle clock input signal. The core provides a clock output signal that is based on a 1394 cycle timer for application synchronization when the core is a 1394 cycle slave.

1.1 LSI Logic CoreWare® Program

An LSI Logic core is a fully defined, optimized, and reusable block of logic. It supports industry-standard functions and has predefined timing and layout. The core is also an encrypted RTL simulation model for a wide range of VHDL and Verilog simulators.

The CoreWare library contains an extensive set of complex cores for the communications, consumer, and computer markets. The library consists of high-speed interconnect functions such as the GigaBlaze® G10® core, MIPS embedded microprocessors, a USB core, a PCI core, and many more.

The library also includes megafunctions or *building blocks*, which provide useful functions for developing a system on a chip. Through the CoreWare program, you can create a system on a chip uniquely suited to your applications.

Each core has an associated set of deliverables, including:

- RTL simulation models for both Verilog and VHDL environments
- A System Verification Environment (SVE) for RTL-based simulation
- Synthesis and timing shells
- Netlists for full timing simulation
- Complete documentation
- LSI Logic FlexStream software support

The LSI Logic FlexStream software provides seamless connectivity between products from leading electronic design automation (EDA) vendors and LSI Logic manufacturing environment. Standard interfaces for formats and languages such as VHDL, Verilog, Waveform Generation Language (WGL), Physical Design Exchange Format (PDEF), and Standard Delay Format (SDF) allow a wide range of tools to interoperate within the LSI Logic FlexStream environment. In addition to design

capabilities, full scan Automatic Test Pattern Generation (ATPG) tools and LSI Logic specialized test solutions can be combined to provide high-fault coverage test programs that assure a fully functional design.

Because your design requirements are unique, LSI Logic is flexible in working with you to develop your system-on-a-chip CoreWare design. Three different work relationships are available:

- You provide LSI Logic with a detailed specification and LSI Logic performs all design work.
- You design some functions while LSI Logic provides you with the cores and megafunctions, and LSI Logic completes the integration.
- You perform the entire design and integration, and LSI Logic provides the core and associated deliverables.

Whatever the work relationship, the LSI Logic advanced CoreWare methodology and ASIC process technologies consistently produce Right-First-Time™ silicon.

1.2 1394 Node Controller Core Overview

The 1394 Node Controller core is targeted for embedded ASIC designs using the IEEE 1394 serial bus. The core provides full link layer functionality. The link layer provides a one-way data transfer with confirmation of request to the transaction layer. The link layer also provides addressing, data checking, and data framing for transmitting and receiving of packets.

The 1394 Node Controller core supports asynchronous and isochronous transfers. You can configure a wide range of isochronous support capabilities, including transmission and reception of several isochronous channels in parallel. The 1394 Node Controller core seamlessly connects to the LSI Logic 1394 Physical Layer core or any other PHY device that conforms to the 1394a specification.

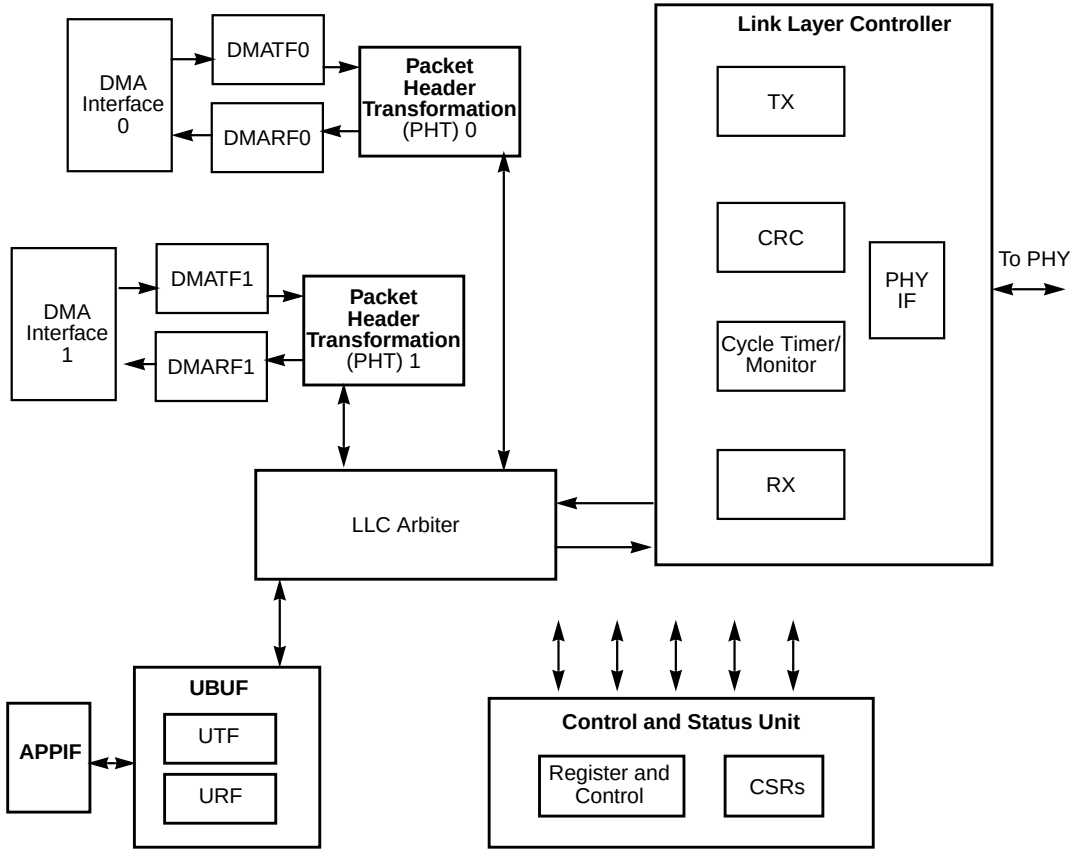
The configurable core design is ideal for a wide range of embedded applications, from basic low-cost devices to sophisticated high-performance ASICs. The core design consists of the Link Layer Controller (LLC), Control and Status Unit, LLC Arbiter, UBUF, APPIF, two PHTs, and two DMATF, DMARF, and DMA interfaces. The Packet Header

Transformation (PHT) module is capable of large isochronous or asynchronous DMA transfers. The universal asynchronous buffer is primarily intended for protocol (SBP-2, DPP, AV/C, IICP, etc.) command and status transfers.

1.3 Block Diagram Description

Figure 1.1 shows a block diagram of the 1394 Node Controller core.

Figure 1.1 1394 Node Controller Core Block Diagram



The **Link Layer Controller (LLC)** implements the complete functionality of the Link Layer in the 1394 Node Controller core. This block interfaces

to the PHY, the Arbiter, the Packet Header Transformation module (PHT), and the Control and Status Unit.

The **PHY Interface (PHYIF)** handles all the protocols between the PHY and the LLC.

The **Universal Asynchronous Buffer (UBUF)** block contains a set of FIFOs for transmitting and receiving asynchronous packets. Only one packet is allowed in each of the UBUF Transmit FIFO (UTF).

The **Packet Header Transformation (PHT)** modules handle the communication between the LLC and the DMA interfaces.

The **Application Interface (APPIF)** module connects the application with the 1394 for nonautomated asynchronous packet transfers. The application uses this interface to access the 1394 register space.

The **Control and Status Unit (CSU)** implements the Node Controller registers.

The **DMA Buffer Block (DBUF)** contains a set of FIFOs for isochronous data transmission and reception for different channels or automated asynchronous transfers.

1.4 Features

The features of the 1394 Node Controller core are summarized in the following subsections.

1.4.1 General

- Supports IEEE Standard 1394-1995 and the 1394a supplement
- Implements full link layer
- Reduces CPU utilization by use of DMA transfers
- Provides automatic asynchronous request packet generation for sequential data buffer accesses
- Generates response to received requests in a designated 4 Gbyte address range without CPU involvement
- Supports 1394 cycle time synchronization with application modules

- Generates IEC-61883 CIP header for isochronous transmission
- Provides hardware-assisted isochronous data flow control
- Provides full packet handling, and packing and unpacking for asynchronous and isochronous packet transmission and reception
- Supports 100/200/400 Mbits/s bus speeds
- Contains a standard PHY interface with up to eight data lines for 400 Mbits/s operation
- Provides 32-bit CRC generation and error detection
- Is configurable to application requirements by optional module selection, compilation parameters (FIFO sizes), and programmable configuration registers
- Connects to the LSI Logic 1394 PHY Core or any other standard 1394a-compliant PHY device

1.4.2 Application Interface

- Application interface for both control and data transfers to be used with a microcontroller or a hardwired controller
- Application interface module with 32-bit wide bus (1394 Node Controller core is bus slave) in burst mode as well as nonburst mode
- Controller interface provides access to all CSRs and control register set
- Extensive maskable interrupt register set provides status reporting

1.4.3 Asynchronous Operation

- Supports full link layer for asynchronous transmission and reception
- Provides configurable asynchronous buffer bank size during compilation; size allocation among the asynchronous FIFOs configurable by the end-user
- Supports single-phase retry sequence

1.4.4 Isochronous Operation

- Supports isochronous bandwidth of the IEEE-1394 bus (up to 400 Mbits/s implementation)
- Transmits and receives multiple isochronous channels

- Provides programmable isochronous data routing with header insertion for transmission and separation for reception
- Supports external DMA controller for isochronous data transmit or receive
- Provides DMA transmit and/or receive FIFO units whose total size is configured during compilation. Size allocation among the FIFOs is configurable by the end-user.
- Provides a byte pack/unpack module for DMA interface

1.4.5 Bus Management

- Transaction and isochronous capable node
- Cycle master capable node
- Bus Manager capable node

1.5 Applications

Some applications for the 1394 Node Controller core include:

- Digital still cameras
- Video conferencing cameras
- Printers
- Storage devices
- Scanners
- Digital audio devices (consumer and professional)
- Digital VCR/TV
- Digital set-top box

Chapter 2
Data Formats

This chapter describes the data formats that the 1394 Node Controller core accepts for transmission and reception over the 1394 bus. This chapter contains the following sections:

- Section 2.1, “Asynchronous Data Formats”
- Section 2.2, “Isochronous DMA Data Formats”
- Section 2.3, “PHY Packet Data Formats”
- Section 2.4, “Miscellaneous Packet Data Formats”

In this chapter, the term *quadlet* refers to a 32-bit data size.

2.1 Asynchronous Data Formats

This section describes the asynchronous data formats for the transmit and receive FIFO operations.

2.1.1 Asynchronous Transmit Data Formats

Asynchronous data is transmitted in one of three basic formats:

1. 3 Quadlet (used for quadlet read requests and quadlet/block write responses)
2. 4 Quadlet (used for block read requests, quadlet write requests, and quadlet read responses)
3. N Quadlet (used for block write requests, lock requests and responses, and block read responses)

2.1.1.1 3-Quadlet Transmit Format

The 3-quadlet data format is shown in Figures 2.1 and 2.2. The first quadlet contains the packet control information. The second and third quadlets contain the 16-bit destination ID and either the 48-bit quadlet aligned destination offset (used for read requests) or the response code (used for write responses).

Figure 2.1 Read Request for Data Quadlet Packet Transmit Format

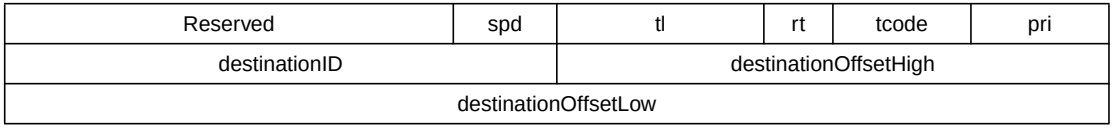
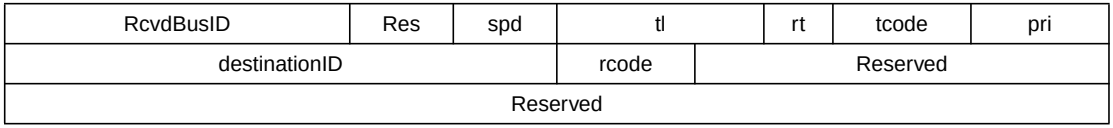


Figure 2.2 Write Response Packet Transmit Format



The field names and their descriptions are described below.

pri **Priority**
This four-bit field is the priority level for the current packet. This field should always be zero.

tcode **Transaction Code**
This four-bit field is the transaction code for the current packet. The following table lists valid encoding for this field.

tcode	Description
0100	Read Request for Data Quadlet
0010	Write Response

rt **Retry Code**
This two-bit field contains the valid retry code for the current packet as listed in the following table.

rt	Retry Code
01	retry_X

tl **Transaction Label**
These six bits indicate the transaction label for the packet. They are used for tracking requests with responses. UBUF transmitted requests should never use 0x3F, 0x3E (reserved for use by the PHT modules).

RcvdBusID **Received Bus ID**
This 10-bit field contains the Destination Bus ID used in the write request. This value is used as the Source Bus ID in the write response.

spd **Speed**
This three-bit field indicates the speed at which the current packet is to be sent.

spd	Speed
000	100 Mbits/s
010	200 Mbits/s
100	400 Mbits/s
all others	Reserved

destinationID **Destination ID**
This 16-bit field specifies the destination ID, which is the concatenation of the 10-bit destination bus ID with the 6-bit destination node ID.

destinationOffset
Destination Offset
The 48-bit destination offset field is the quadlet-aligned destination address.

rcode **Response Code**
This field contains the four-bit response code to an earlier corresponding request.

2.1.1.2 4-Quadlet Transmit Format

This data format is shown in Figures 2.3 through 2.5. The first quadlet contains the packet control information. The second and third quadlets contain the 16-bit destination ID and either the 48-bit quadlet aligned destination offset (used for read/write requests) or the response code (used for read responses). The fourth quadlet contains the quadlet data for read responses and quadlet write requests or the data length for the block read request.

Figure 2.3 Write Request for Data Quadlet Packet Transmit Format

Reserved	spd	tl	rt	tcode	pri
destinationID		destinationOffsetHigh			
destinationOffsetLow					
quadlet_data					

Figure 2.4 Read Response for Quadlet Data Packet Transmit Format

RcvdBusID	Res	spd	tl	rt	tcode	pri
destinationID			rcode	Reserved		
Reserved						
quadlet_data						

Figure 2.5 Read Request for Block Data Packet Transmit Format

Reserved	spd	tl	rt	tcode	pri
destinationID		destinationOffsetHigh			
destinationOffsetLow					
data_length		Reserved			

The field names and descriptions are described below.

pri **Priority**
This four-bit field is the priority level for the current packet. This field should always be zero.

tcode **Transaction Code**
This four-bit field is the transaction code for the current packet. The following table lists the valid encoding for this field.

tcode	Description
0000	Write Request for Data Quadlet
0110	Read Response for Data Quadlet
0101	Read Request for Data Block

rt **Retry Code**
This two-bit field contains the valid retry code for the current packet as listed in the following table.

rt	Retry Code
01	retry_X

tl **Transaction Label**
These six bits indicate the transaction label for the packet. They are used for tracking requests with responses. UBUF transmitted requests should never use 0x3F, 0x3E (reserved for use by the PHT modules).

RcvdBusID **Received Bus ID**
This 10-bit field contains the Destination Bus ID used in the read request. This value is used as the Source Bus ID in the read response.

spd **Speed**
This three-bit field indicates the speed at which the current packet is to be sent.

spd	Speed
000	100 Mbits/s
010	200 Mbits/s
100	400 Mbits/s
all others	Reserved

destinationID **Destination ID**
This 16-bit field specifies the destination ID, which is the concatenation of the 10-bit destination bus ID with the 6-bit destination node ID.

DestinationOffset
Destination Offset
The 48-bit destination offset field is the quadlet-aligned destination address.

rcode **Response Code**
This field contains the four-bit response code to an earlier corresponding request.

quadlet_data **Quadlet Data**
This 32-bit field contains the quadlet data to be transmitted for either a read response or a write request.

data_length **Data Length**
The 16-bit data length field specifies the length of the data block size in bytes for block-data read requests.

2.1.1.3 N-Quadlet Transmit Format

The N-Quadlet data format is shown in Figures 2.6 through 2.7. The first quadlet contains the packet control information. The second and third quadlets contain the 16-bit destination ID and either the 48-bit quadlet aligned destination offset (used for read/write requests) or the response code (used for read/lock responses). The fourth quadlet contains the data length and the extended transaction code for the block write/read/lock.

The block data, if any, follows the extended code. Block data must be quadlet aligned; that is, you might have to append zeros to the block data for the last quadlet.

Figure 2.6 Lock/Write Request for Data Block Packet Transmit Format

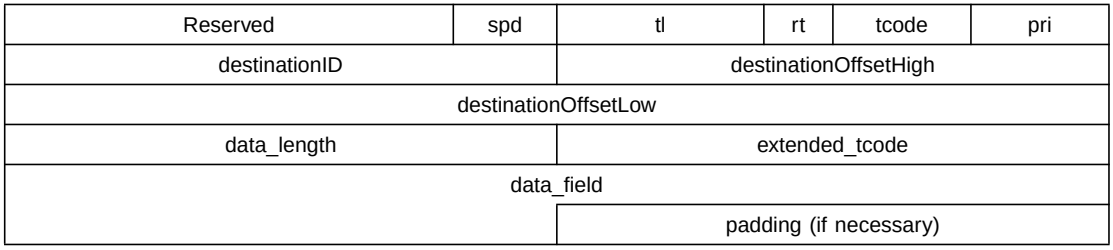
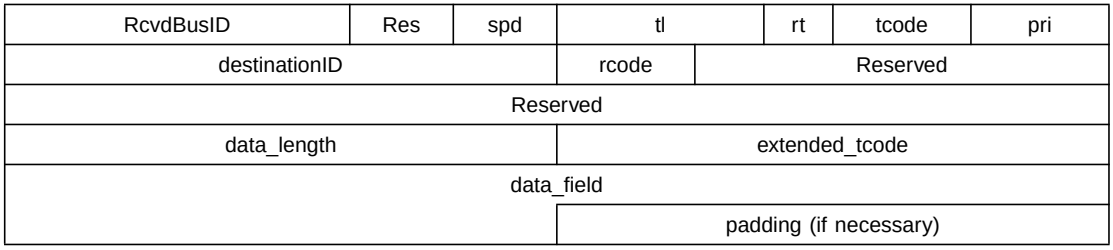


Figure 2.7 Lock/Read Response for Block Data Packet Transmit Format



The field names and their descriptions are described below.

pri **Priority**
This four-bit field is the priority level for the current packet. This field should always be zero.

tcode **Transaction Code**
This four-bit field is the transaction code for the current packet. The following table lists valid encoding for this field.

tcode	Description
0001	Write Request for Data Block
1001	Lock Request
1011	Lock Response
0111	Read Response for Data Block

rt **Retry Code**
This two-bit field contains the valid retry code for the current packet as listed in the following table.

rt	Retry Code
01	retry_X

tl **Transaction Label**
These six bits indicate the transaction label for the packet. They are used for tracking requests with responses. UBUF transmitted requests should never use 0x3F, 0x3E (reserved for use by the PHT modules).

RcvdBusID **Received Bus ID**
This 10-bit field contains the Destination Bus ID used in the read request. This value is used as the Source Bus ID in the read response.

spd **Speed**
This three-bit field indicates the speed at which the current packet is to be sent.

spd	Speed
000	100 Mbits/s
010	200 Mbits/s
100	400 Mbits/s
all others	Reserved

destinationID **Destination ID**
This 16-bit field specifies the destination ID, which is the concatenation of the 10-bit destination bus ID with the 6-bit destination node ID.

DestinationOffset
Destination Offset
The 48-bit destination offset field is the quadlet-aligned destination address.

rcode **Response Code**
This field contains the four-bit response code to an earlier corresponding request.

data_length **Data Length**
The 16-bit data length field specifies the length of the data block size in bytes for block-data read requests.

data_field **Data Field**
This field contains block data that needs to be transmitted. Its length is specified by the data_length field. If data_length is zero, there is no data_field. If the value in the data_length field is not a multiple of four, the data field must be padded with zeros to make it quadlet aligned.

extended_tcode
Extended Transaction Code
The extended transaction code is used only for lock transactions. The following table lists the encoding of this field. Refer to the IEEE-1394 document for detailed explanations of this decoding.

extended_tcode	Description
0x0	Reserved (reads and writes)
0x1	mask_swap
0x2	compare_swap
0x3	fetch_add
0x4	little_add
0x5	bounded_add
0x6	wrap_add
0x7	vendor_dependent
0x8–0xF	Reserved

2.1.2 Asynchronous Receive Data Formats

Asynchronous data is received in the receive FIFO in one of four basic formats:

1. 4 Quadlet (used for read requests for data quadlets and write responses)
2. 5 Quadlet (used for read requests for block data, write requests for data quadlets, and read responses for data quadlets)
3. N Quadlet (used for write requests for data blocks, lock requests and responses, and read responses for data blocks)
4. Self-ID Data Packets

2.1.2.1 4-Quadlet Receive Format

These data formats are shown in Figures 2.8 and 2.9. The first quadlet contains the destination node ID and the rest of the packet header information. The second and third quadlets contain the 16-bit source ID and either the 48-bit quadlet-aligned destination offset (used for read requests) or the response code (used for write responses).

Figure 2.8 Read Request Receive Format for Data Quadlet

destinationID			tl	rt	tcode	pri
sourceID			destinationOffsetHigh			
destinationOffsetLow						
Reserved		spd	Reserved			

Figure 2.9 Write Response Packet Receive Format

destinationID			tl	rt	tcode	pri
sourceID			rcode	Reserved		
Reserved						
Reserved		spd	Reserved			

The field names and their descriptions are described below.

pri **Priority**
This four-bit field is the priority level for the current packet. This field should always be zero.

tcode **Transaction Code**
This four-bit field is the transaction code for the current packet. The following table lists the valid encoding for this field.

tcode	Description
0100	Read Request for Data Quadlet
0010	Write Response

rt **Retry Code**
This two-bit field contains the retry code for the current packet as listed in the following table.

rt	Retry Code
01	retry_X

tl **Transaction Label**
These six bits indicate the transaction label for the packet. They are used for tracking requests with responses.

spd **Speed**
This three-bit field indicates the speed at which the current packet was received.

spd	Speed
000	100 Mbits/s
010	200 Mbits/s
100	400 Mbits/s
all others	Reserved

destinationID **Destination ID**
This 16-bit field specifies the destination ID, which is the concatenation of the 10-bit destination bus ID with the 6-bit destination node ID.

DestinationOffset
Destination Offset
The 48-bit destination offset field is the quadlet-aligned destination address.

rcode **Response Code**
This field contains the four-bit response code to an earlier corresponding request.

sourceID **Source ID**
This 16-bit field indicates the node ID and bus ID of the sender.

2.1.2.2 5-Quadlet Receive Format

These data formats are shown in Figures 2.10 through 2.12. The first quadlet contains the destination ID and the rest of the packet header information. The second and third quadlets contain the 16-bit source ID and either the 48-bit quadlet-aligned destination offset (used for read/write requests) or the response code (used for read responses). The fourth quadlet contains either the quadlet data for read responses and quadlet write requests or the data length for the block read request. The last quadlet contains the packet reception status.

Figure 2.10 Write Quadlet Request Receive Format

destinationID	tl	rt	tcode	pri
sourceID	destinationOffsetHigh			
destinationOffsetLow				
quadlet_data				
Reserved	spd	Reserved		

Figure 2.11 Read Quadlet Response Packet Receive Format

destinationID	tl	rt	tcode	pri
sourceID	rcode	Reserved		
Reserved				
quadlet_data				
Reserved	spd	Reserved		

Figure 2.12 Read Block Request Packet Receive Format

The field names and their descriptions are described below.

pri **Priority**
This four-bit field is the priority level for the current packet. This field should always be zero.

tcode **Transaction Code**
This four-bit field is the transaction code for the current packet. The following table lists the valid encoding for this field.

tcode	Description
0100	Read Request for Data Quadlet
0010	Write Response

rt **Retry Code**
This two-bit field contains the retry code for the current packet as listed in the following table.

rt	Retry Code
01	retry_X

tl **Transaction Label**
These six bits indicate the transaction label for the packet. They are used for tracking requests with responses.

spd **Speed**
This three-bit field indicates the speed at which the current packet was received.

spd	Speed
000	100 Mbits/s
010	200 Mbits/s
100	400 Mbits/s
all others	Reserved

destinationID **Destination ID**
This 16-bit field specifies the destination ID, which is the concatenation of the 10-bit destination bus ID with the 6-bit destination node ID.

DestinationOffset
Destination Offset
The 48-bit destination offset field is the quadlet-aligned destination address.

rcode **Response Code**
This field contains the four-bit response code to an earlier corresponding request.

sourceID **Source ID**
This 16-bit field indicates the node ID and bus ID of the sender.

quadlet_data **Quadlet Data**
This 32-bit field contains the quadlet data to be transmitted for either a read response or a write request.

data_length **Data Length**
The 16-bit data length field specifies the length of the data block size in bytes for block-data read requests.

2.1.2.3 N-Quadlet Receive Format

These data formats are shown in Figures 2.13 and 2.14. The first quadlet contains the destination ID and the rest of the packet header information. The second and third quadlets contain the 16-bit source ID and either the 48-bit quadlet-aligned destination offset (used for read/write requests) or the response code (used for read/lock responses). The

fourth quadlet contains the data length and the extended transaction code for the block write/read/lock.

The block data, if any, follows the extended code. Block data must be quadlet aligned; that is, there may be appended zeros to the block data for the last quadlet of the data field.

The EHdr and IHdr bits in the PHT Control and Status Register 0 affect the format of the data written to the DBuf Transmit and Receive FIFOs when EnDMAS is asserted. If IHdr is not set, only the data field portion of the incoming request is written to the DBuf Receive FIFO. If IHdr is set and EHdr is not set, the 1394 header is written to the DBuf Receive FIFO in addition to the data field. If IHdr and EHdr are set, in addition to the 1394 header and data field, a quadlet containing the speed of the received request is written to the DBuf Receive FIFO.

Note: If IHdr and EHdr are set, the format of the data in the DBuf is as shown in Figure 2.13 (Write Request for Data Block Receive Format). The maximum data field size that can be received is the DBuf Receive FIFO size less 20 bytes.

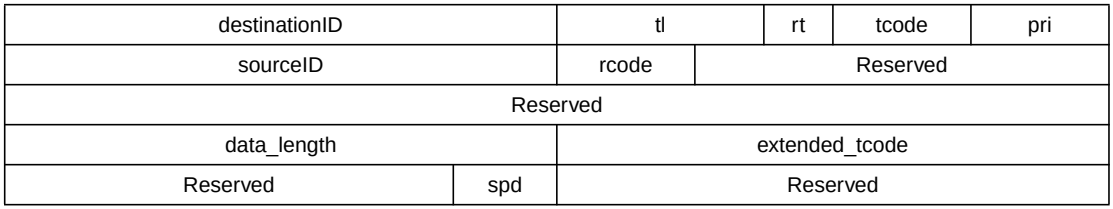
If EHdr is not set, the DBuf Transmit FIFO must contain only the data field for read response packets to be sent in response to received read requests. If EHdr is set, the format followed is that of Figure 2.4 or Figure 2.7.

An appropriate setting for EHdr and IHdr depends on the capability of the external DMA controller.

Figure 2.13 Lock/Write Request for Data Block Receive Format

<table>
<tr><td colspan="3">destinationID</td><td colspan="2">tl</td><td>rt</td><td>tcode</td><td>pri</td></tr>
<tr><td colspan="3">sourceID</td><td colspan="5">destinationOffsetHigh</td></tr>
<tr><td colspan="8">destinationOffsetLow</td></tr>
<tr><td colspan="3">data_length</td><td colspan="5">extended_tcode</td></tr>
<tr><td colspan="8">data_field</td></tr>
<tr><td colspan="3"></td><td colspan="5">padding (if necessary)</td></tr>
<tr><td colspan="2">Reserved</td><td>spd</td><td colspan="5">Reserved</td></tr>
</table>

Figure 2.14 Lock/Read Response for Block Data Receive Format



The field names and their descriptions are described below.

pri **Priority**
This four-bit field is the priority level for the current packet. This field should always be zero.

tcode **Transaction Code**
This four-bit field is the transaction code for the current packet. The following table lists valid encoding for this field.

tcode	Description
0001	Write Request for Data Block
1001	Lock Request
1011	Lock Response
0111	Read Response for Data Block

rt **Retry Code**
This two-bit field contains the valid retry code for the current packet as listed in the following table.

rt	Retry Code
01	retry_X

tl **Transaction Label**
These six bits indicate the transaction label for the packet. They are used for tracking requests with responses. UBUF transmitted requests should never use 0x3F, 0x3E (reserved for use by the PHT modules).

spd **Speed**
This three-bit field indicates the speed at which the current packet is to be sent.

spd	Speed
000	100 Mbits/s
010	200 Mbits/s
100	400 Mbits/s
all others	Reserved

destinationID **Destination ID**
This 16-bit field specifies the destination ID, which is the concatenation of the 10-bit destination bus ID with the 6-bit destination node ID.

sourceID **Source ID**
This 16-bit field indicates the node ID and bus ID of the sender.

DestinationOffset
Destination Offset
The 48-bit destination offset field is the quadlet-aligned destination address.

rcode **Response Code**
This field contains the four-bit response code to an earlier corresponding request.

data_length **Data Length**
The 16-bit data length field specifies the length of the data block size in bytes for block-data read requests.

data_field **Data Field**
This field contains block data that needs to be transmitted. Its length is specified by the data_length field. If data_length is zero, there is no data_field. If the value in the data_length field is not a multiple of four, the data field must be padded with zeros to make it quadlet aligned.

extended_tcode
Extended Transaction Code
The extended transaction code is used only for lock transactions. The following table lists the encoding of this

field. Refer to the IEEE-1394 document for detailed explanations of this decoding.

extended_tcode	Description
0x0	Reserved (reads and writes)
0x1	mask_swap
0x2	compare_swap
0x3	fetch_add
0x4	little_add
0x5	bounded_add
0x6	wrap_add
0x7	vendor_dependent
0x8–0xF	Reserved

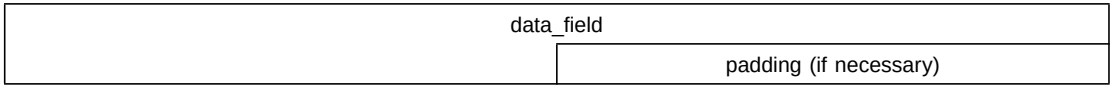
2.2 Isochronous DMA Data Formats

This section describes the isochronous data transmit/receive formats. The data formats for the transmits and receives are described separately.

2.2.1 Isochronous Transmit Data Formats

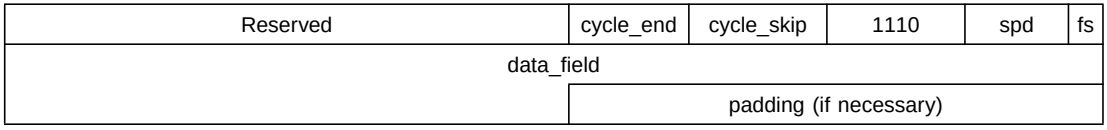
The data formats for an isochronous transmit using the DMA transmit FIFO are shown from Figure 2.15 through Figure 2.17.

Figure 2.15 Isochronous Transmit Format 1



data_field **Data Field**
This field contains the isochronous data.

Figure 2.16 Isochronous Transmit Format 2



Reserved **Reserved**
This 19-bit field is reserved and should be zeros.

cycle_end **Cycle End**
This one-bit field determines whether this is the last packet for which an isochronous request should be made during the current isochronous cycle (this bit must be set if hardware-generated CIP headers are enabled). If hardware-generated CIP headers are not enabled, this bit can be used to group several packets (with different isochronous channel numbers) that are to be transmitted during the same isochronous cycle. If N packets with different channel numbers are to be transmitted during the same isochronous cycle, the first N – 1 packets would have cycle end equal to zero and the Nth packet would have cycle end equal to one. Thus N packets are transmitted during the same isochronous cycle.

cycle_skip **Cycle Skip**
This four-bit field is used for flow rate control. Once the core has queued and is ready to transmit the packet, the core waits N isochronous cycles before requesting the 1394 bus (where N is the value in the cycle skip field), if GenCIP is not set. If GenCIP is set, the core sends empty CIPs for N isochronous cycles before requesting the 1394 bus to send the data_field following the embedded control quadlet.

spd **Speed**
This three-bit field is the speed at which the packet is to be sent.

Note: If CIP format isochronous data with hardware-generated CIP headers is being transmitted, in addition to setting the spd field in the embedded control quadlet, the speed field in the Stream Transmit Channel Header register must be set to the same value.

fs **Frame Start**
When set, this one-bit field causes an SYT time stamp to be generated when the packet is DMAed to the core if certain PHT module control bits are set.

data_field **Data Field**
This field contains the isochronous data.

Figure 2.17 Isochronous Transmit Format 3

Note: CIP format isochronous data with hardware-generated CIP headers is not supported when using Isochronous Transmit Format 3.

Reserved **Reserved**
This 19-bit field is reserved and must be zeros.

ce **Cycle End**
This one-bit field determines whether this is the last packet for which an isochronous request should be made during the current isochronous cycle. This bit can be used to group several packets (with different isochronous channel numbers) that are to be transmitted during the same isochronous cycle. If N packets with different channel numbers were desired to be transmitted during the same isochronous cycle, the first N – 1 packets would have cycle end equal to zero and the Nth packet would have cycle end equal to one. This would result in the N packets being transmitted during the same isochronous cycle.

cycle_skip **Cycle Skip**
This four-bit field is used for flow rate control. Once the core has queued and is ready to transmit the packet, the core waits N isochronous cycles before requesting the 1394 bus (where N is the value in the cycle skip field).

spd **Speed**
This three-bit field is the speed at which the packet is to be sent.

fs **Frame Start**
This one-bit field when set causes an SYT time stamp to be generated when the packet is DMAed to the core, if certain PHT module control bits are set.

data_length **Data Length**
This field contains the 16-bit data length for isochronous data.

tag **Tag**
This field contains the two-bit isochronous tag data format.

tag	Description
00	Data Field Unformatted
01	IEC-61883 format
10 –11	Reserved

channel **Channel**
This field contains the six-bit isochronous channel number.

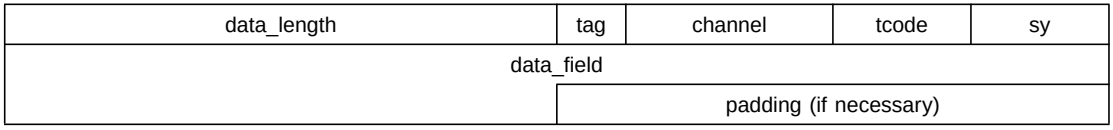
sy **Synchronization Code**
This four-bit synchronization code is application dependent.

data_field **Data Field**
This field contains the isochronous data.

2.2.2 Isochronous Receive Data Formats

The data formats for isochronous receives are shown in Figure 2.18 and Figure 2.19. The first quadlet in Figure 2.18 contains the isochronous channel number, data length, and synchronization code. Subsequent quadlets contain the isochronous data received.

Figure 2.18 Isochronous Receive Format 1



The field names and their descriptions are described below.

data_length **Data Length**
This field contains the 16-bit data length for Isochronous data.

tag **Tag**
This field contains the two-bit Isochronous tag data format.

tag	Description
00	Data Field Unformatted
01	IEC-61883 Format
10 –11	Reserved

channel **Channel**
This field contains the six-bit isochronous channel number.

tcode **Transaction Code**
This four-bit field is the transaction code for the current packet.

sy **Synchronization Code**
This four-bit synchronization code is application dependent.

data_field **Data Field**
This field contains the isochronous data.

Figure 2.19 Isochronous Receive Format 2

The field names and their descriptions are described below.

data_field **Data Field**
This field contains the isochronous data.

2.3 PHY Packet Data Formats

This section describes the PHY packet data transmit/receive formats. The data formats for the transmit and receive are described separately.

2.3.1 PHY Packet Transmit Data Format

The data formats for PHY packet transmissions are shown in Figure 2.20. The least significant eight bits of the first quadlet are fixed

at 0b1110.0000. The second quadlet contains the PHY packet data. The third quadlet is the inverse of the second quadlet.

Figure 2.20 PHY Packet Transmit Data Format

Reserved	1110	0000
PHY Packet Quadlet		
Inverse of PHY Packet Quadlet		

2.3.2 PHY Packet Receive Data Format

The data formats for PHY packet receptions are shown in Figure 2.21. The least significant eight bits of the first quadlet are fixed at 0b1110.0000. The second quadlet contains the PHY packet data.

Figure 2.21 PHY Packet Receive Data Format

Reserved	1110	0000
PHY Packet Quadlet		

2.4 Miscellaneous Packet Data Formats

This section describes the remaining packet data formats that the 1394 Node Controller core receives. The core, if programmed, will receive the self-ID packets.

2.4.1 Self-ID Packet Receive Data Format

The data format for the self-ID packets is shown in Figure 2.22. The first quadlet contains the packet header information with tcode equal to 0b1110. The remaining quadlets contain the data that is received from the time the bus is reset to the first subaction gap. This data is the concatenation of all the self-ID packets received. The SIDF bit in the PDC Control Register determines whether or not the inverse quadlet of the PHY packet is stored in addition to the PHY packet quadlet itself.

The ResultCode field is either 0xE (an error occurred) or 0x1 (no error occurred). If the last quadlet in the Universal Receive FIFO does not follow this format, it indicates that an overflow condition occurred.

Figure 2.22 Self-ID Receive Format

Reserved	1110	0001
Self-ID Packet Data		
Reserved		Result Code

2.4.2 Unformatted Data Transmit Format

The unformatted data transmit format is shown in Figure 2.23. It is sent as is without any CRCs. The first quadlet is not included in the transmission. No ack is expected.

Figure 2.23 Unformatted Data Transmit Format

Reserved	1110	0001
Unformatted Packet Data		

Chapter 3
Signal Descriptions

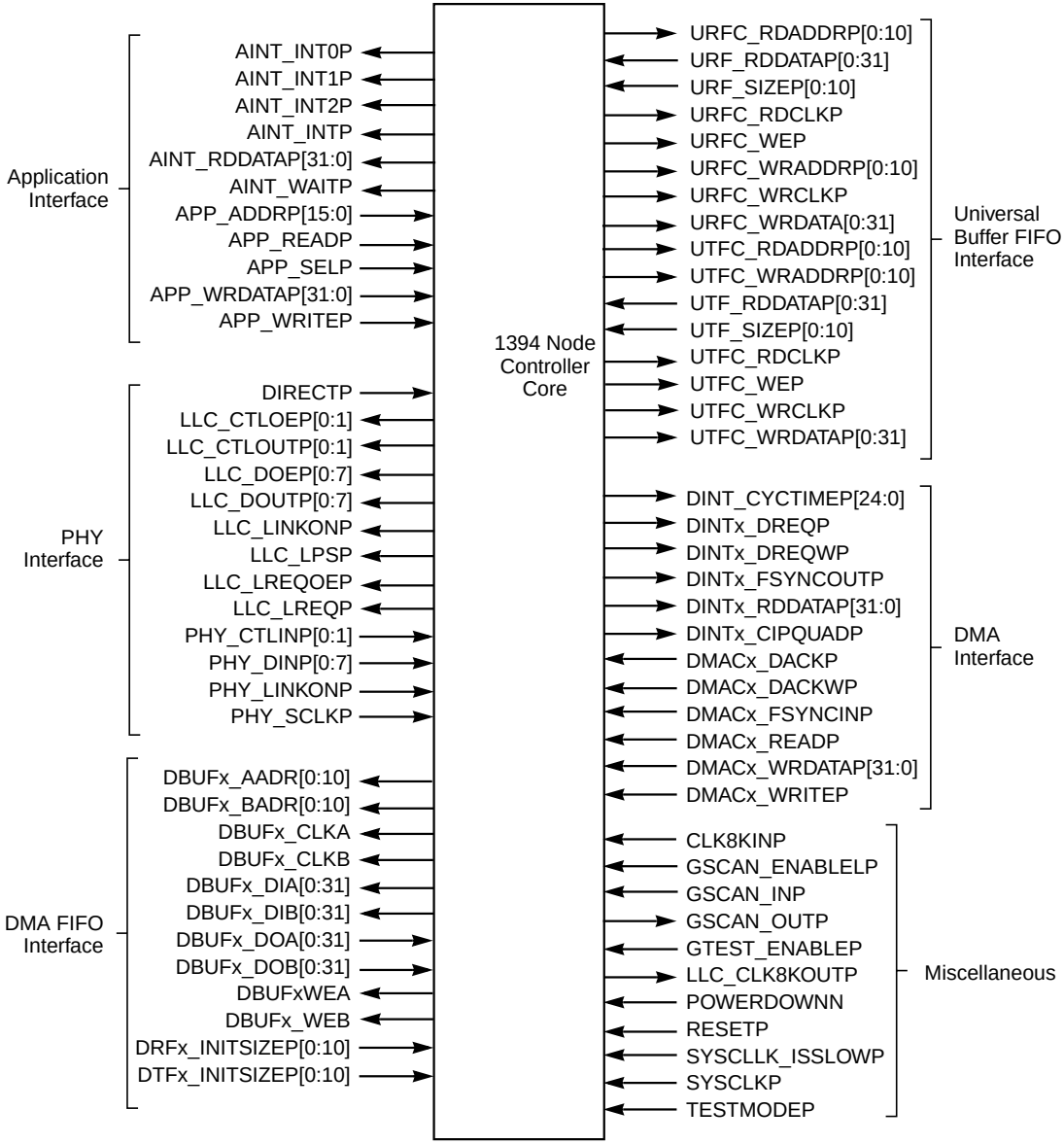
This chapter describes the signals for the 1394 Node Controller core. The signals are categorized according to interface. They are described in alphabetical order by mnemonic. The mnemonic for an active LOW signal ends in an “N”; the mnemonic for an active HIGH signal ends in a “P.”

In the following descriptions, the verb *assert* means to drive TRUE or active. The verb *deassert* means to drive FALSE or inactive.

This chapter contains the following sections:

Figure 3.1 is a logic diagram of the 1394 Node Controller Core.

Figure 3.1 1394 Node Controller Core Logic Diagram



3.1 Application Interface

This interface is used for application software device access. The 1394 Node Controller core is a slave on this bus.

AINT_INT0P **Interrupt Register 0 Interrupt** **O**
The core asserts this signal HIGH to indicate an “Interrupt Register 0” interrupt occurred.

AINT_INT1P **Interrupt Register 1 Interrupt** **O**
The core asserts this signal HIGH to indicate an “Interrupt Register 1” interrupt occurred.

AINT_INT2P **Interrupt Register 2 Interrupt** **O**
The core asserts this signal HIGH to indicate an “Interrupt Register 2” interrupt occurred.

AINT_INTP **Interrupt Register Interrupt** **O**
The core asserts this signal HIGH to indicate an interrupt in Interrupt Register 0, 1, or 2 occurred.

AINT_RDDATAP[31:0]
Data Read Port to Application **O**
The 1394 Node Controller core puts the data on this bus when the application device is reading an internal register or FIFO.

AINT_WAITP **Wait Cycle Request** **O**
The 1394 Node Controller core asserts this signal HIGH to indicate that the current application interface transaction is not finished. When asserted, this signal tells the application device to continue the current read/write from/to the core.

APP_ADDRP[15:0]
Address Bus **I**
The application device puts the address on this bus when it wants to read or write a 1394 Node Controller core’s internal register or FIFO.

APP_READP **Application Data Read from Registers/FIFO** **I**
The application device asserts this signal and APP_SELP HIGH when it wants to read the 1394 Node Controller core’s internal registers or FIFOs.

APP_SELP **Select Signal** **I**
When this control signal is asserted HIGH, access to the registers and FIFOs is enabled.

APP_WRDATAP[31:0]
Data Write Port **I**
The application device puts the data on this bus for a write to a 1394 Node Controller core internal register or FIFO.

APP_WRITEP **Application Data Write to Registers/FIFO** **I**
The application device asserts this signal and APP_SELP HIGH when it wants to write the 1394 Node Controller core's internal registers or FIFOs.

3.2 Universal Buffer FIFO Interface

URFC_RDADDRP[0:10]
Receive FIFO Read Address Port **O**
When reading the UBUF Receive FIFO, the core places the read address on this bus.

URF_RDDATAP[0:31]
Receive FIFO Read Data Port **I**
The 1394 Node Controller core receives data from the UBUF Receive FIFO on this bus.

URF_SIZEP[0:10]
Size of Receive FIFO Connected **I**
This bus indicates to the core the size (in quadlets) of the UBUF Receive FIFO.

URFC_RDCLKP
Receive FIFO Read Clock **O**
This signal is the Read Clock for the UBUF Receive FIFO.

URFC_WEP **Receive FIFO Write Enable Signal** **O**
The 1394 Node Controller core asserts this signal HIGH to request a write to the UBUF Receive FIFO.

URFC_WRADDRP[0:10]

Receive FIFO Write Address Port **O**

When writing to the UBUF Receive FIFO, the core places the write address on this bus.

URFC_WRCLKP

Receive FIFO Write Clock **O**

This signal is the Write Clock for the connected UBUF Receive FIFO.

URFC_WRDATAP[0:31]

Receive FIFO Write Data Port **O**

The core places data to be written to the UBUF Receive FIFO on this bus.

UTFC_RDADDRP[0:10]

Transmit FIFO Read Address Port **O**

The 1394 Node Controller core places the address for a read of the UBUF Transmit FIFO on this bus.

UTFC_WRADDRP[0:10]

Transmit FIFO Write Address Port **O**

The 1394 Node Controller core places the address for a write of the UBUF Transmit FIFO on this bus.

UTF_RDDATAP[0:31]

Transmit FIFO Read Data Port **I**

Data read from the UBUF Transmit FIFO is input to the core on this bus.

UTF_SIZEP[0:10]

Size of Transmit FIFO Connected **I**

This bus indicates to the core the size (in quadlets) of the UBUF Transmit FIFO.

UTFC_RDCLKP

Transmit FIFO Read CLOCK **O**

This signal is the Read Clock for the UBUF Transmit FIFO.

UTFC_WEP **Transmit FIFO Write Enable Signal** **O**

The core asserts this signal HIGH to perform a write to the UBUF Transmit FIFO.

UTFC_WRCLKP

Receive FIFO Write CLOCK **O**

This signal is the Write Clock for the connected UBUF Transmit FIFO.

UTFC_WRDATAP[0:31]

Transmit FIFO Write Data Port **O**

The core places data to be written to the UBUF Transmit FIFO on this bus.

3.3 PHY Interface

The PHY interface connects to a PHY device. This interface conforms to IEEE 1394a, Revision 2.0.

DIRECTP **Nondifferentiated PHY Interface** **I**

Drive this signal LOW when an isolation barrier is present.

LLC_CTLOEP[0:1]

PHY Control Enable **O**

In designs that use 3-state buffers for LLC_CTLOUTP, use these signals to enable the 3-state buffers.

LLC_CTLOUTP[0:1]

PHY Control Out **O**

LLC_CTLOUTP[0:1] are the control lines from the core to the PHY device.

LLC_CTLOUTP[0:1]	Description
00	Idle. Transmission complete, bus released.
01	Hold. The core is either holding the bus while preparing data or indicating that it wants to reacquire the bus without arbitrating to send another packet.
10	Transmit. The core is sending a packet to the PHY device.
11	Reserved.

LLC_DOEP[0:7]

PHY Data Enable **O**

In designs that use 3-state buffers for LLC_DOUTP, use these signals to enable the 3-state buffers.

LLC_DOUTP[0:7]

PHY Data Out **O**

The core places data to be written to the PHY device on this bus. LLC_DOUTP0 is the most-significant bit; LLC_DOUTP7 is the least-significant bit.

LLC_LINKONP

LLC Link On **O**

The core asserts this signal to indicate it has received a link on indication from the PHY device.

LLC_LPSP **Link Power Status** **O**

The core asserts this signal to indicate the core is active.

LLC_LREQOEP

Link Request Enable **O**

In designs that use a 3-state buffer for LLC_LREQP, use this signal to enable the 3-state buffer. When DIRECTP is not asserted, LLC_LREQOEP acts as a digital differentiator that enables the 3-state buffer only when LLC_LREQP changes.

LLC_LREQP **Link Request** **O**

The 1394 Node Controller core asserts this signal HIGH to request access to the PHY device.

PHY_CTLINP[0:1]

PHY Control In **I**

These signals are the control lines from the PHY device to the 1394 Node Controller core.

PHY_CTLINP[0:1]	Description
00	Idle. No activity.
01	Status. The PHY device is sending status information to the core.
10	Receive. A packet is being transferred from the PHY device to the core.
11	Grant. The core has been granted the bus and can send a packet.

PHY_DINP[0:7]

PHY Data In **I**

This bus receives data from the PHY device. PHY_DINP0 is the most-significant bit; PHY_DINP7 is the least-significant bit.

PHY_LINKONP

PHY Link On **I**

When this signal is asserted, another node is requesting activation of the core.

PHY_SCLKP **PHY Clock** **I**

This input connects to the 49.152 MHz input clock from the PHY device.

3.4 DMA 0 Interface

This interface is used for DMA data transfers. Data paths can be configured as 8, 16, or 32 bits wide.

DINT_CYCTIMEP[24:0]

Cycle Time **O**

These signals reflect the least significant 25 bits of the Cycle Time Register synchronized to SYSCLKP. Bit 24 is the most significant bit of the cycle count field.

DINT0_DREQP

DMA Read Request for Transmit and Receive Channels **O**

When the RACtl bit in the DMA Control and Status Register is set, packets read from the PHT0 DBUF use DINT0_DREQP while packets written to the PHT0 DBUF use DINT0_DREQWP.

If simultaneous read/write access is not required, RACtl can be reset to zero. In this case, packets read from or written to the PHT0 DBUF use DINT0_DREQP.

DINT0_DREQWP

DMA Write Request for Transmit and Receive Channels **O**

When the RACtl bit in the DMA Control and Status Register is set, packets read from the PHT0 DBUF use

DINT0_DREQP while packets written to the PHT0 DBUF use DINT0_DREQWP.

If simultaneous read/write access is not required, RACtl can be reset to zero. In this case, packets read from or written to the PHT0 DBUF use DINT0_DREQP.

DINT0_FSYNCOUTP

Fsync Output **O**

If the core is receiving a CIP format stream containing SYT time stamps, then, when an SYT time stamp is detected, DINT0_FSYNCOUTP is asserted during the read of the quadlet following the SYT time stamp.

DINT0_RDDATAP[31:0]

Data Read Port **O**

The 1394 Node Controller core places data from the DBUF Receive FIFO on this bus.

DINT0_CIPQUADP

CIP QUADLET **O**

The quadlet output on **DINT0_RDDATAP** is a IEC-61883 CIP quadlet.

DMAC0_DACKP

DMA Read Acknowledge **I**

When the RACtl bit in the DMA Control and Status register is set, packets read from the PHT0 DBUF use DMAC0_DACKP while packets written to the PHT0 DBUF use DMAC0_DACKWP.

If simultaneous read/write access is not required, RACtl can be reset to zero. In this case, packets read from or written to the PHT0 DBUF use DMAC0_DACKP.

DMAC0_DACKWP

DMA Write Acknowledge **I**

When the RACtl bit in the DMA Control and Status register is set, packets read from the PHT0 DBUF use DMAC0_DACKP while packets written to the PHT0 DBUF use DMAC0_DACKWP.

If simultaneous read/write access is not required, RACtl can be reset to zero. In this case, packets read from or written to the PHT0 DBUF use DMAC0_DACKP.

DMAC0_FSYNCINP

Frame Sync Input **I**

When the FSSEL bit of the PHT Control and Status Register is set, the FSYNCIN input signal is used to control time stamping when the GenCIP bit is set and the most-significant bit of the FMT field of the Stream Control Transmit 1 Register is 0 (DVCR). This input should be asserted during the first write strobe of a packet in which an SYT time stamp is generated by the core.

DMAC0_READP

EDMAC Data Read from FIFO **I**

The external DMA controller asserts this signal HIGH when it wants to read the DBUF Receive FIFO.

DMAC0_WRDATAP[31:0]

Data Write Port **I**

The DMA device places data to be written to the DBUF Transmit FIFO on this bus.

DMAC0_WRITEP

EDMAC Data Write to FIFO **I**

The external DMA controller asserts this signal HIGH to request a write to the DBUF Transmit FIFO.

3.5 DMA 1 Interface

This interface is used for DMA data transfers. Data paths can be configured as 8, 16, or 32 bits wide.

DINT1_DREQP

DMA Read Request for Transmit and Receive Channels **O**

When the RACtl bit in the DMA Control and Status Register is set, packets read from the PHT1 DBUF use DINT1_DREQP while packets written to the PHT1 DBUF use DINT1_DREQWP.

If simultaneous read/write access is not required, RACtl can be reset to zero. In this case, packets read from or written to the PHT1 DBUF use DINT1_DREQP.

DINT1_DREQWP

DMA Write Request for Transmit and Receive Channels **O**

When the RACtl bit in the DMA Control and Status register is set, packets read from the PHT1 DBUF use DINT1_DREQP while packets written to the PHT1 DBUF use DINT1_DREQWP.

If simultaneous read/write access is not required, RACtl can be reset to zero. In this case, packets read from or written to the PHT1 DBUF use DINT1_DREQP.

DINT1_FSYNCOUTP

Fsync Output **O**

If the core is receiving a CIP format stream containing SYT time stamps, then, when an SYT time stamp is detected, DINT1_FSYNCOUTP is asserted during the read of the quadlet following the SYT time stamp.

DINT1_RDDATAP[31:0]

Data Read Port **O**

The 1394 Node Controller core places data from the DBUF Receive FIFO on this bus.

DINT1_CIPQUADP

CIP QUADLET **O**

The quadlet output on **DINT1_RDDATAP** is a IEC-61883 CIP quadlet.

DMAC1_DACKP

DMA Read Acknowledge **I**

When the RACtl bit in the DMA Control and Status register is set, packets read from the PHT1 DBUF use DMAC1_DACKP while packets written to the PHT1 DBUF use DMAC1_DACKWP.

If simultaneous read/write access is not required, RACtl can be reset to zero. In this case, packets read from or written to the PHT1 DBUF use DMAC1_DACKP.

DMAC1_DACKWP

DMA Write Acknowledge **I**

When the RACtl bit in the DMA Control and Status register is set, packets read from the PHT1 DBUF use DMAC1_DACKP while packets written to the PHT1 DBUF use DMAC1_DACKWP.

If simultaneous read/write access is not required, RACtl can be reset to zero. In this case, packets read from or written to the PHT1 DBUF use DMAC1_DACKP.

DMAC1_FSYNCINP

Frame Sync Input **I**

When the FSSEL bit of the PHT Control and Status register is set, the FSYNCIN input signal is used to control time stamping when the GenCIP bit is set and the most-significant bit of the FMT field of the Stream Control Transmit 1 Register is 0 (DVCR). This input should be asserted during the first write strobe of a packet in which an SYT time stamp is generated by the core.

DMAC1_READP

EDMAC Data Read from FIFO **I**

The external DMA controller asserts this signal HIGH when it wants to read the DBUF Receive FIFO.

DMAC1_WRDATAP[31:0]

Data Write Port **I**

The DMA device places data to be written to the DBUF Transmit FIFO on this bus.

DMAC1_WRITEP

EDMAC Data Write to FIFO **I**

The external DMA controller asserts this signal HIGH to request a write to the DBUF Transmit FIFO.

3.6 DMA 0 FIFO Interface

DBUF0_AADR[0:10]

PHT0 DMA Side FIFO Address Port **O**

The 1394 Node Controller core places the address on this bus for an access of the DMA side of the DBUF Transmit/Receive FIFO.

DBUF0_BADR[0:10]

PHT0 PHT Side FIFO Address Port **O**

The 1394 Node Controller core places the address on this bus for an access of the PHT side of the DBUF Transmit/Receive FIFO.

DBUF0_CLKA

PHT0 DMA Side FIFO Clock **O**

Clock for an access of the DMA side of the DBUF Transmit/Receive FIFO.

DBUF0_CLKB

PHT0 PHT Side FIFO Clock **O**

This signal is the clock for an access of the PHT side of the DBUF Transmit/Receive FIFO.

DBUF0_DIA[0:31]

PHT0 Transmit FIFO Write Data Port **O**

Data to be written from the 1394 Node Controller core to the Transmit FIFO is output on this bus.

DBUF0_DIB[0:31]

PHT0 Receive FIFO Write Data Port **O**

Data to be written from the 1394 Node Controller core to the Receive FIFO is output on this bus.

DBUF0_DOA[0:31]

PHT0 Receive FIFO Read Data Port **I**

This 32-bit bus receives data from the Receive FIFO on this bus.

DBUF0_DOB[0:31]

PHT0 Transmit FIFO Read Data Port **I**

The 1394 Node Controller core receives data from the Transmit FIFO on this bus.

DBUF0_WEA **PHT0 Transmit FIFO Write Enable Signal** **O**

The 1394 Node Controller core asserts this output to request a write of the Transmit FIFO.

DBUF0_WEB **PHT0 Receive FIFO Write Enable Signal** **O**

The 1394 Node Controller core asserts this signal to request a write to the Receive FIFO.

DRF0_INITSIZEP[0:10]

Size of PHT0 Receive FIFO **I**

This bus indicates the size of the Receive FIFO.

DTF0_INITSIZEP[0:10]

Size of PHT0 Transmit FIFO **I**

This 11-bit bus indicates the size of the Transmit FIFO.

3.7 DMA 1 FIFO Interface

DBUF1_AADR[0:10]

PHT1 DMA SIDE FIFO Address Port **O**

The 1394 Node Controller core places the address on this bus for an access of the DMA side of the DBUF Transmit/Receive FIFO.

DBUF1_BADR[0:10]

PHT1 PHT SIDE FIFO Address Port **O**

The 1394 Node Controller core places the address on this bus for an access of the PHT side of the DBUF Transmit/Receive FIFO.

DBUF1_CLKA

PHT1 DMA SIDE FIFO CLOCK **O**

This signal is the clock for accesses to the DMA side of the DBUF Transmit/Receive FIFO.

DBUF1_CLKB

PHT1 PHT SIDE FIFO CLOCK **O**

This signal is the clock for accesses to the PHT side of the DBUF Transmit/Receive FIFO.

DBUF1_DIA[0:31]

PHT1 Transmit FIFO Write Data Port **O**

Data to be written from the 1394 Node Controller core to the Transmit FIFO is output on this bus.

DBUF1_DIB[0:31]

PHT1 Receive FIFO Write Data Port **O**

Data to be written from the 1394 Node Controller core to the Receive FIFO is output on this bus.

DBUF1_DOA[0:31]

PHT1 Receive FIFO Read Data Port **I**

This 32-bit bus receives data from the Receive FIFO.

DBUF1_DOB[0:31]

PHT1 Transmit FIFO Read Data Port **I**

The 1394 Node Controller core receives data from the Transmit FIFO on this bus.

DBUF1_WEA **PHT1 Transmit FIFO Write Enable Signal** **O**
The 1394 Node Controller core asserts this output to request a write of the Transmit FIFO.

DBUF1_WEB **PHT1 Receive FIFO Write Enable Signal** **O**
The 1394 Node Controller core asserts this signal to request a write to the Receive FIFO.

DRF1_INITSIZEP[0:10]
Size of PHT1 Receive FIFO **I**
This bus indicates the size of the Receive FIFO.

DTF1_INITSIZEP[0:10]
Size of PHT1 Transmit FIFO **I**
This 11-bit bus indicates the size of the Transmit FIFO.

3.8 Miscellaneous

CLK8KINP **8-KHz Clock Input** **I**
This optionally selectable clock is used for the cycle timer.

GSCAN_ENABLEP
Scan Enable **I**
This input is used during manufacturer's testing only. Hold it LOW during normal operation.

GSCAN_INP **Scan Data In** **I**
This input is used during manufacturer's testing only. During normal operation, it is ignored.

GSCAN_OUTP
Scan Data Out **O**
This output is used during manufacturer's testing only. During normal operation, it is ignored.

GTEST_ENABLEP
Test Enable **I**
This input is used during manufacturer's testing only. Hold it LOW during normal operation.

LLC_CLK8KOUTP
8 kHz Clock Output **O**
This signal changes at the same rate as the cycle count field of the cycle timer register (nominally 8 kHz).

POWERDOWNN

Power Down **I**

When asserted, this signal causes the LLC_LPSP signal to be deasserted.

RESETP **System Reset** **I**

This input is the core initialization signal. To reset the 1394 Node Controller core, assert this input HIGH for a minimum of two SYSCLKP cycles.

SYSCLK_ISSLOWP

SYSCLKP Range **I**

When asserted, this input indicates SYSCLKP is in the 11 to 2 MHz range.

SYSCLKP **Application Domain Clock** **I**

SYSCLKP is the application interface clock. Its frequency range is 2 to 50 MHz.

TESTMODEP **Test Mode** **I**

This input is used during manufacturer's testing only. Hold it LOW during normal operation.

Chapter 4
Registers

This chapter describes the 1394 Node Controller core registers. The Control and Status registers and the 1394 registers are directly accessible from the application interface.

This chapter contains the following sections:

4.1 Ordering Conventions

The ordering conventions of the 1394 Serial Bus are used in the representation of data structures. In order to promote interoperability with memory buses that may have different ordering conventions, the definition of the order and significance of bits within bytes, bytes within quadlets, and quadlets with octlets is in terms of their relative position and not their physically addressed position.

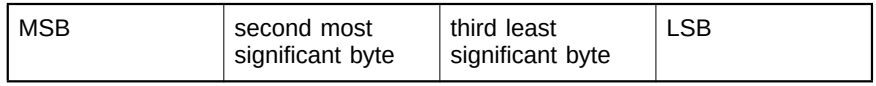
Within a byte, the most significant bit, msb, is that which is transmitted first; the least significant bit, lsb, is that which is transmitted last on the Serial Bus, as illustrated in Figure 4.1. The significance of the interior bits uniformly decreases in progression from msb to lsb.

Figure 4.1 Interior Bit Significance

msb	interior bits (decreasing significance left to right)	lsb

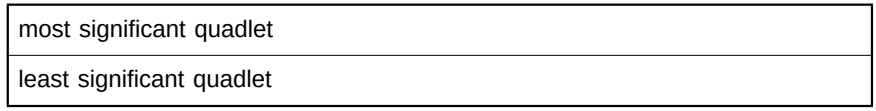
Within a quadlet, the most significant byte (MSB) is that which is transmitted first; the least significant byte (LSB) is that which is transmitted last on the Serial Bus, as shown in Figure 4.2.

Figure 4.2 Interior Byte Significance



Within an octlet, the most significant quadlet is that which is transmitted first, and the least significant quadlet is that which is transmitted last on the Serial Bus, as Figure 4.3 indicates.

Figure 4.3 Internal Quadlet Order



4.2 Control and Status Registers (CSRs)

This section describes the CSRs. Table 4.1 shows the implemented CSRs in this block with their respective register space addresses.

Table 4.1 CSR Registers

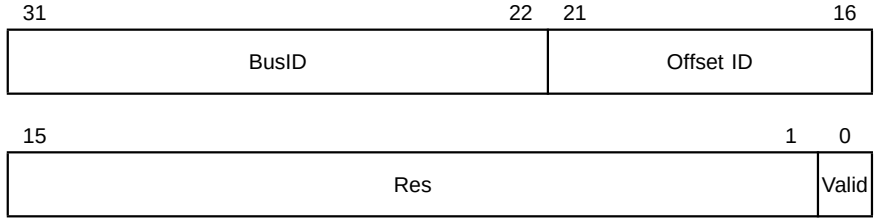
Register Space Address (Application side)	Register
0x000	Node ID
0x004	Cycle Timer

Accesses from the 1394 bus to the CSRs are served with split asynchronous subactions.

4.2.1 Node ID Register (0x000)

The Node ID register is used to identify and modify the 16-bit identifier of this node. It identifies the current bus ID and physical ID values, which directly affect the initial node address. One of the fields in this register is the offset_id field, known as the physical-ID value of the node. The PHY

determines the physical ID after a bus reset during the SelfID process. When the SelfID process is complete, the physical ID has been determined, and it is set in a register inside the PHY. In addition, the PHY automatically initiates a status transfer to the link, which includes the physical ID.



BusID **Bus ID** **[31:22]**
Default: 0x03FF **RW**
This read/write field allows multiple bus configurations to distinguish node addresses on one bus from those on another. The initial value of this field is 0x03FF.

OffsetID **Offset ID** **[21:16]**
Default: 0x00 **RO**
This read-only field contains the physical ID of the node. It is generated during the SelfID process. The initial value of this field is 0.

Res **Reserved** **[15:1]**
These bits are reserved and must be written as zeros.

Valid **Valid** **0**
This bit is cleared from when a bus reset is detected until the Offset ID has been automatically updated and from a Power On Reset until the Offset ID has been automatically updated.

4.2.2 Cycle Time Register (0x004)

The fields in this register specify the current time value. Reception of a Cycle Start packet when the device is not cycle master and the Cycle Timer is enabled is treated as a write to the Cycle Start register. The register can also be modified using the application interface.

SecCnt **Second Count** **[31:25]**
Default: 0x00 **RW**
This field increments on each carry from the Cycle Count. An increment from value 127 causes a wraparound to zero.

CycCnt **Cycle Count** **[24:12]**
Default: 0x0000 **RW**
This field increments on each carry from the Cycle Offset field when ExtCyc is zero or for each positive transition of the CLK8KIN input signal when ExtCyc and CMstr are one. An increment from value 7999 causes a wraparound to zero and carries into the Second Count field (this field has 125 µs resolution).

CycOff **Cycle Offset** **[11:0]**
Default: 0x000 **RW**
This field increments every other SCLK from the PHY. When ExtCyc is zero, an increment from when the CycOff field matches the value in the OffMatch field of the Control 1 Register (default value 3071) causes a wraparound to zero and carries into the Cycle Count field. Each positive transition of CLK8KIN when ExtCyc and CMstr are one causes a wraparound to zero.

4.3 1394 Registers

These registers control and monitor the operation of the 1394 Node Controller core. Table 4.2 shows these registers with their respective address offsets.

Table 4.2 1394 Register Summary

Address Offset	Register Name	Page #
0x008	Control 0 Register	page 4-7
0x00C	Control 1 Register	page 4-11
0x010	Control 2 Register	page 4-12
0x014	PHY Access Register	page 4-13
0x020	Interrupt 0 Register	page 4-14
0x024	Interrupt 0 Mask Register	page 4-19
0x028	Interrupt 1 Register	page 4-21
0x02C	Interrupt 1 Mask Register	page 4-22
0x030	Interrupt 2 Register	page 4-23
0x034	Interrupt 2 Mask Register	page 4-23
0x038	DMA Space Register	page 4-24
0x03C	Acknowledge Status Register	page 4-25
0x040	UBUF Transmit Next	page 4-25
0x044	UBUF Transmit Last	page 4-25
0x048	UBUF Transmit Clear	page 4-26
0x04C	UBUF Receive Clear	page 4-26
0x050	UBUF Receive	page 4-26
0x054	UBUF Receive Level	page 4-26
0x058	Encoded Interrupt Priority	page 4-26
0x080	Packet Header Transformation Control and Status Register 0	page 4-28
0x084	PHT Split Time-out/Empty CIP Interval Register 0	page 4-35
0x088	PHT Request/Response/CIP Receive Header 0 Register 0	page 4-37
0x08C	PHT Request/Response/CIP Receive Header 1 Register 0	page 4-39

Table 4.2 1394 Register Summary

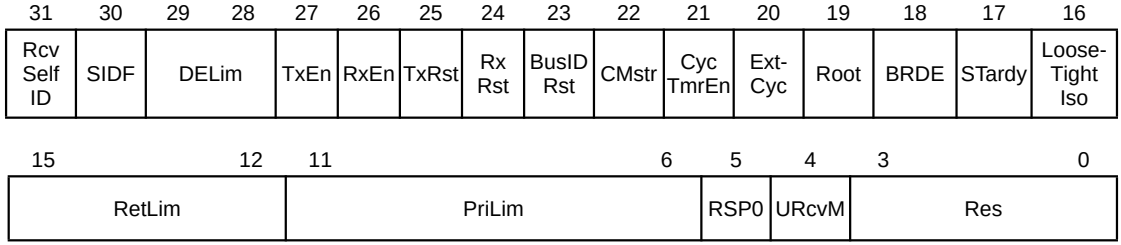
Address Offset	Register Name	Page #
0x090	PHT Request Header 2/SPH Receive Register 0	page 4-42
0x094	Stream Receive Channel/NodeID Selection 0 Register 0	page 4-43
0x098	Stream Receive Channel/NodeID Selection 1 Register 0	page 4-44
0x09C	Stream Receive Channel Header Register 0	page 4-44
0x0A0	Stream Transmit Channel Header Register 0	page 4-45
0x0A4	Data Transfer Control Register 0	page 4-47
0x0A8	CIP Header Transmit 0 Register 0	page 4-48
0x0AC	CIP Header Transmit 1 Register 0	page 4-49
0x0B4	Stream Transmit Time Stamp Offset Register 0	page 4-50
0x0B8	DMA Control and Status Register 0	page 4-52
0x0BC	DMA Transfer Threshold Register 0	page 4-53
0x0C0	DBUF FIFOs Level Register 0	page 4-54
0x0C4	DBUF Tx Data Register 0	page 4-54
0x0C8	DBUF Rx Data Register 0	page 4-55
0x0CC	DBUF Watermarks Register 0	page 4-56
0x0D0	DBUF FIFO Size Register 0	page 4-57
0x100	Packet Header Transformation Control and Status Register 1	page 4-57
0x104	PHT Split Time-out/Empty CIP Interval Register 1	page 4-64
0x108	PHT Request/Response/CIP Receive Header 0 Register 1	page 4-66
0x10C	PHT Request/Response/CIP Receive Header 1 Register 1	page 4-68
0x110	PHT Request Header 2/SPH Receive Register 1	page 4-71
0x114	Stream Receive Channel/NodeID Selection 0 Register 1	page 4-73
0x118	Stream Receive Channel/NodeID Selection 1 Register 1	page 4-73
0x11C	Stream Receive Channel Header Register 1	page 4-74
0x120	Stream Transmit Channel Header Register 1	page 4-75
0x124	Data Transfer Control Register 1	page 4-76
0x128	CIP Header Transmit 0 Register 1	page 4-77
0x12C	CIP Header Transmit 1 Register 1	page 4-78

Table 4.2 1394 Register Summary

Address Offset	Register Name	Page #
0x134	Stream Transmit Time Stamp Offset Register 1	page 4-79
0x138	DMA Control and Status Register 1	page 4-81
0x13C	DMA Transfer Threshold Register 1	page 4-82
0x140	DBUF FIFOs Level Register 1	page 4-83
0x144	DBUF Tx Data Register 1	page 4-84
0x148	DBUF Rx Data Register 1	page 4-84
0x14C	DBUF Watermarks Register 1	page 4-85
0x150	DBUF FIFO Size Register 1	page 4-86

4.3.1 Control 0 Register (0x008)

The bits within the Control 0 Register enable/disable various functions of the 1394 Node Controller core. All bits in this register can be read and written from the application interface.



RcvSelfID **Receive SelfID** **31**
Default: 0 **RW**
When this bit is set, all SelfID packets from PHYs during bus initialization are placed in the Universal Receive or DMARF0 FIFO. When this bit is cleared, these packets are ignored.

SIDF **SelfID Format** **30**
Default: 0 **RW**
When this bit is set, both quadlets of all SelfID packets from PHYs during bus initialization are placed in the Receive FIFO. Setting this bit is useful when SelfID packets are received corrupted and determining suspect

nodes is desired. When this bit is cleared, the first quadlet of all SelfID packets from PHYs during bus initialization is placed in the Universal Receive FIFO.

DELim **Data Error Retry Limit** **[29:28]**
Default: 0 **RW**
The value in this field determines how many automatic retry attempts are made for an asynchronous transmit packet when an ack_data_error is received in response to a transmission attempt. A value of zero means only the original transmission of the packet is made.

TxEn **Transmitter Enable** **27**
Default: 0 **RW**
When this bit is cleared, the transmitter does not arbitrate or send any packets.

RxEn **Receiver Enable** **26**
Default: 0 **RW**
When this bit is cleared, the receiver stops receiving packets.

TxRst **Transmitter Reset** **25**
Default: 0 **RW**
When this bit is set, the entire transmitter resets synchronously. This bit clears itself.

RxRst **Receiver Reset** **24**
Default: 0 **RW**
When this bit is set, the entire receiver resets synchronously. This bit clears itself.

BusIDRst **Bus ID Reset** **23**
Default: 0 **RW**
When this bit is set, the Bus ID field is cleared to 0x03FF, when a local bus reset is received.

CMstr **Cycle Master** **22**
Default: 0 **RW**
When this bit is set, the core transmits a Cycle Start packet subsequent to the CycCnt field of the Cycle Timer register being incremented. Subsequent to a bus reset, if the root bit is zero (as seen in the status from the PHY), then this bit is automatically cleared.

CycTmrEn **Cycle Timer Enable** **21**
Default: 1 **RW**
When this bit is set, the cycle timer's cycle offset field is incremented.

ExtCyc **ExtCyc** **20**
Default: 0 **RW**
When this bit and the CMstr are set, the cycle timer's CycCnt field is incremented subsequent to each positive transition of the CLK8KIN input signal.

Root **Root** **19**
Default: 0 **RO**
A one on this bit indicates this node is the root on the bus. It automatically updates after the SelfID phase.

BRDE **Busy Received Data Errors** **18**
Default: 0
When this bit is set, the LLC sends an Ack_Busy_x acknowledge to any rejected incoming packet with data CRC or data length mismatch errors. Setting this bit can improve the throughput from OHCI links that are experiencing temporary conditions causing underflow of the OHCI link's Transmit FIFO.

STardy **Send Tardy** **17**
Default: 0 **RW**
When this bit is set, the receiver sends an Ack Tardy acknowledge to any incoming packet addressed to the core.

LooseTightIso
Loose ISO Cycles **16**
Default: 1 **RW**
When this bit is reset to zero, loose ISO cycles are supported. When this bit is one, tight ISO cycles are supported. Tight ISO means that isochronous format packets are only accepted if received during the interval between a cycle start packet and a subaction gap.

RetLim **Retry Limit** **[15:12]**
Default: 0xF **RW**
The value in this field determines how many automatic retry attempts are made for an asynchronous transmit packet when a busy ack is received in response to a

transmission attempt. A value of zero means only the original transmission of a packet is made.

PriLim **Priority Request Limit** **[11:6]**
Default: 0x00 **RW**

The value in this field determines how many priority requests the core can make in a fairness interval for PHY packets and packets with the following asynchronous tcodes (0x0, 0x1, 0x4, 0x5, 0x9, 0xA). In addition asynchronous packets with tcodes of (0x2, 0x6, 0x7, 0xB, 0x8) can use priority requests without regard to this limit. At each arbitration reset gap indication from the PHY, a decrementer is loaded with the value of this field. As long as the counter is nonzero (it is decremented after each transmission of a packet subject to the limit) and a busy acknowledge has not been received subsequent to packet transmission, priority requests are used for the packets listed. If the decrementer reaches zero or a busy ack is received in response to a packet transmitted by the core, the core only uses fair arbitration requests (with the exception of cycle start packets) until reception of another arbitration reset gap indication from the PHY.

RSP0 **Route SelfID Packets to PHT0** **5**
Default: 0

When this bit is set, SelfID packets are routed to PHT0/DMARF0. Setting this bit is useful when the total size of the SelfID packets exceeds the size of the UBUF. When this condition occurs, the application can disable the DMA interface (and also clear ELis, EnDMAS, ERReq, EWReq, ETalk), set this bit, and cause the PHY to initiate a bus reset. Once the SubActGap interrupt occurs, this bit can be cleared by the application and the SelfID packets can be read from the DMARF0 using DBUF Rx FIFO Data Register 0.

URcvM **UBuf Receive Multiple Packets** **4**
Default: 0

When this bit is set, the Universal Receive Buffer accepts multiple packets.

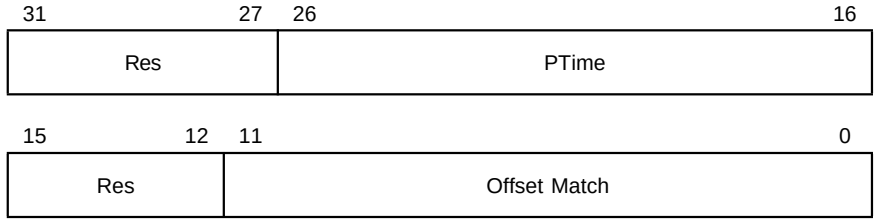
When this bit is cleared, the Universal Receive Buffer will send Busy Acks in response to all incoming packets, if there is already a packet in the UBuf Receive FIFO. An exception is made for PHY packets and SelfID packets.

Multiple SelfID and PHY packets are accepted if the previously accepted packets are PHY or SelfID packets.

Res **Reserved** **[3:0]**
Default: 0
These bits are reserved and must be written as zeros.

4.3.2 Control 1 Register (0x00C)

The settings within the Control 1 Register enable/disable various functions of the 1394 Node Controller core.



Res **Reserved** **[31:27], [15:12]**
These bits are reserved for future use. They must be written as zeros.

PTime **Ping Timer** **[26:16]**
Default: 0x7FF **RO**
Each time the last bit of a PHY packet is transmitted from the core to the PHY, this field is initialized to zero. PTime is incremented every other Sclk until the core receives the first bit of the response to the PHY packet. The bus manager uses the resulting value to optimize the gap count.

OffMatch **Offset Match** **[11:0]**
Default: 3071 **RW**
This field contains the value compared with the Cycle Offset field of the Cycle Time register to determine carry generation for the Cycle Count field of the Cycle TIme register.

4.3.3 Control 2 Register (0x010)

The settings within the Control 2 Register enable/disable various functions of the 1394 Node Controller core. All bits in this register can be read and written from the application interface. This register's clock is the application clock, unlike most of the other registers.

Res **Reserved** **[31:4]**
Default: 0
These bits are reserved and must be written as zeros.

SOK **SCLK OK** **3**
Default: 0 **RO**
When this bit is set, the clock from the PHY is operational. This bit can be polled after asserting LPSEn to determine when accesses to all SCLK-based registers can be successfully made.

SRst **SCLK Domain Reset** **2**
Default: 1 **RW**
When this bit is set, all logic that runs off the clock from the PHY (most of the core) is reset. This bit is automatically cleared after the reset has occurred (which can only occur while LPSEn is set). Reset the FIFOs before using them after this bit is set.

LPSEn **Link Power Status Enable** **1**
Default: 0 **RW**
When this bit is set, the LPS output is asserted (HIGH in undifferentiated mode, pulsing in differentiated mode).

Note: Sensible deassertion of this bit results in power savings due to the clock supplied from the PHY (which is used for most of the link logic) stopping if this bit is not asserted.

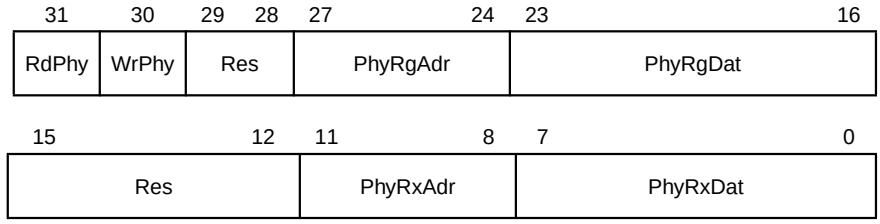
Important: Only the Control 2, Interrupt 2, and Interrupt 2 Mask Registers can be accessed when LPSEn is deasserted. All

other register accesses will not return the actual value in those registers when read and any writes will not affect the contents of those registers. After asserting LPSEn, valid accesses to other registers cannot be immediately made because it is dependent upon the clock from the PHY being restored.

LPSRst **Link Power Status Reset** **0**
Default: 0 **RW**
When this bit is set, the LPS output is deasserted (LOW in either undifferentiated mode or differentiated mode) for at least 2.75 microseconds but no more than 24 microseconds. This bit is automatically cleared. Do not set this bit during packet transmission in order to ensure compliance with Section 5.1 of the 1394a standard with regards to PHY/link interface signal conditioning.

4.3.4 PHY Access Register (0x014)

The PHY Access Register provides access to the internal registers of the attached PHY.



When reading a PHY register, the address of the register is written to the PhyRgAdr field with the RdPhy bit set. This bit is cleared once the request has been sent to the PHY. The clearing of the bit does not mean that the data has been received in the register. When the PHY returns the data for the read through status, the PHY Register Received (PhyRRx) interrupt is set, indicating that this register contains the data from the PHY read access.

When writing a PHY register, the address and data of the PHY register are written into PhyRgAdr and PhyRgDat fields with the WrPhy bit set. This bit is cleared once the write request has been sent to the PHY. The application should not try to write into this register as long as RdPhy or WrPhy is set. When the read PHY bit is set, the read data and the

address of the PHY register are written into the PhyRxDat and PhyRxAdr fields eventually and the PhyRRx interrupt bit is set.

All bits in this register can be read from and written into the application interface side except PhyRxAdr and PhyRxDat, which are read only.

RdPhy **Read PHY** **31**
Default: 0 **RW**
When this bit is set, the 1394 Node Controller core initiates a register read to the attached PHY.

WrPhy **Write PHY** **30**
Default: 0 **RW**
When this bit is set, the 1394 Node Controller core initiates a register write to the attached PHY.

Res **Reserved** **[29:28], [15:12]**
These bits are reserved and must be written as zeros.

PhyRgAdr **PHY Access Address** **[27:24]**
Default: 0 **RW**
This field contains the address of the PHY register to be accessed.

PhyRgDat **Write PHY Register Data** **[23:16]**
Default: 0 **RW**
This field contains the data to be written into the PHY register indicated by PhyRgAdr field.

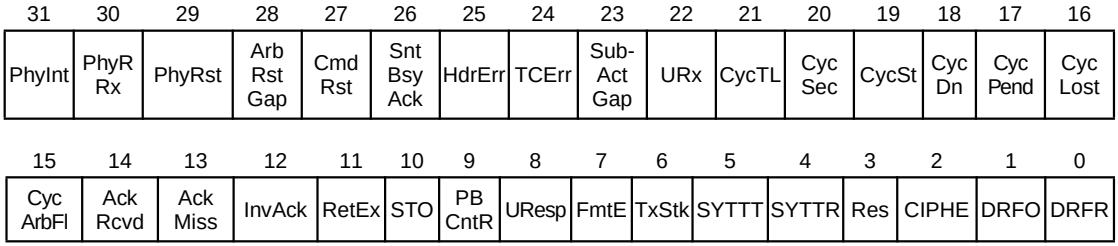
PhyRxAdr **Receive PHY Address** **[11:8]**
Default: 0 **RO**
This read-only field contains the address of the register from which PhyRxDat came.

PhyRxDat **Receive PHY Register Data** **[7:0]**
Default: 0 **RO**
This read-only field contains received register data from the PHY.

4.3.5 Interrupt 0 Register (0x020)

The fields within the Interrupt 0 Register inform the application when the state of the 1394 Node Controller core changes. When a bit is set in the Interrupt 0 Register and its corresponding bit in the Interrupt 0 Mask Register is set, the 1394 Node Controller core generates an interrupt to

the application. The application acknowledges the request, and writes a one to the corresponding interrupt bit. This action resets the corresponding interrupt bit. The default value for this register is 0.



PhyInt **PHY Interrupt** **31**
A one on this bit indicates an interrupt status bit was received from the PHY.

PhyRRx **PHY Register Received** **30**
A one on this bit indicates a register value has been transferred to the PHY Access register from the PHY interface.

PhyRst **PHY Reset** **29**
A one on this bit indicates the PHY reconfiguration has started (bus reset). The PHY Reset interrupt acts as a Tx UBUF reset. The Tx UBUF is reset as long as the interrupt bit is on (prevents sending UBUF packets when the intended destination node ID might have changed).

ArbRstGap **Arbitration Reset Gap** **28**
A one on this bit indicates the serial bus has been idle for an arbitration reset gap.

CmdRst **Command Reset** **27**
A one on this bit indicates that the receiver has been sent a quadlet write request addressed to the Reset_Start CSR register.

SntBsyAck **Sent Busy/Tardy Acknowledge** **26**
A one on this bit indicates the receiver was forced to send the busy acknowledge to a packet addressed to this node because the Receive FIFO contains a previously received packet or a tardy ACK was sent and the packet was not accepted.

HdrErr **Header Error** **25**
A one on this bit indicates the receiver detected a header CRC error on an incoming packet that may have been addressed to this node.

TCErr **Transaction Code Error** **24**
A one on this bit indicates the transmitter detected an invalid transaction code in the data at the Transmit FIFO interface.

SubActGap **Subaction Gap** **23**
A one on this bit indicates the PHY detected a subaction gap.

URx **UBUF Packet Received** **22**
A one on this bit indicates that an asynchronous packet has been received and is confirmed in the UBUF.

CycTL **Cycle Too Long** **21**
This bit is set, if after transmitting a cycle start packet, a subaction gap or bus reset is not detected within 116 microseconds. The CMstr bit is automatically cleared upon this bit being set. The CMstr bit cannot be set while this bit is set.

CycSec **Cycle Second** **20**
This bit is set whenever the cycle-second field of the Cycle Timer Register is updated.

CycSt **Cycle Start** **19**
This bit is set whenever the cycle start packet has been sent or received.

CycDn **Cycle Done** **18**
This bit is set when an arbitration gap has been detected on the bus after the cycle-start packet. This bit indicates that the isochronous cycle is over.

CycPnd **Cycle Pending** **17**
This bit is set when a cycle start packet is expected to be sent or received.

CycLost **Cycle Lost** **16**
This bit is set when no cycle start packet is sent/received between two successive cycle sync events.

CycArbFl **Cycle Arbitration Failed** **15**
This bit is set if the arbitration to send the Cycle-Start packet has failed.

AckRcvd **Acknowledge Received** **14**
A one on this bit indicates the ack code and transaction label for the last request or response packet transmitted from the UBUF are received and are present in the Ack Status register.

AckMiss **Acknowledge Missing** **13**
After the core has transmitted an asynchronous packet, this bit is set if a subaction gap indication is received from the PHY without an intervening acknowledge reception. This bit is not set after PHY or broadcast packet transmission.

InvAck **Invalid Acknowledge** **12**
This bit is set upon receipt of an acknowledge that is reserved.

RetEx **Retry Attempts Exhausted** **11**
This bit is set if a packet has been retransmitted RetLim or DELim number of times and received a busy/data-error ack each time.

STO **Split Time Out** **10**
This bit is set when a PHT request packet has been transmitted that received a pending ACK that was not followed by a response packet before the Split Time Out limit was reached (the PAct bit is cleared).

PBCntR **Packet/Byte Count Reached** **9**
This bit is set if the PBCnt field of the Data Transfer Control Register is decremented to zero, meaning a PBCnt number of request packets or PBCnt number of bytes has been transmitted successfully (the PAct bit is cleared). For a series of read requests, the data from the last packet that caused the PBCntR to be set might still be in the DBUF Receive FIFO (the DAct bit of the DMA Control and Status Register can be examined to determine when the final packet has been completely transferred).

UResp	**Unexpected Response** This bit is set when a bus reset occurs while the PAct bit was set and a PHT was in one of the asynchronous modes (EWReq or ERReq, EAST or EnDMAS set) or if a request packet generated by a PHT module receives a final response other than ‘complete’ from the destination node (the PAct bit is cleared). The PHT Control and Status Register can be read to determine the final response received to the request.	**8**
FmtE	**Format Error** This bit is set whenever the PHT module enters the stuck state, because either it detected the Control Quadlet or the isochronous header that was expected to be prepended to the data field of an isochronous packet from the DBUF Transmit FIFO contained an unexpected value (transmission stopped and PHT stuck).	**7**
TxStk	**Tx Stuck** This bit is set whenever the Tx enters the stuck state, because the Transmit FIFO becomes empty in the middle of a packet transmission.	**6**
SYTTT	**SYT Time Stamp Transmitted** This bit is set whenever a packet containing an SYT time stamp has been transmitted.	**5**
SYTTR	**SYT Time Stamp Received** This bit is set whenever a packet containing an SYT time stamp has been received.	**4**
Res	**Reserved** These bits are reserved and must be written as zeros.	**3**
CIPHE	**CIP Header Error** This bit is set whenever the eoh or form bits of a received CIP header are unexpected (packet discarded).	**2**
DRFO	**DBUF Receive FIFO Overflow** This bit is set when during an isochronous format packet reception the DBUF receive FIFO overflows. The packet that caused the overflow is removed from the FIFO and reception continues.	**1**

DRFR **DBUF Receive FIFO Reception** **0**
A one on this bit indicates a packet has been confirmed into the DMARF.

4.3.6 Interrupt 0 Mask Register (0x024)

The Interrupt 0 Mask Register determines whether the interrupts specified in the Interrupt 0 Register are activated or not. Each bit in this register corresponds to an interrupt flag bit in the Interrupt 0 Register. To mask a particular interrupt, clear its corresponding bit in the Interrupt 0 Mask Register. The application interface can read and write all bits in this register. The default value for each bit in this register is zero.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Phy IntM	PhyR-RxM	Phy Rst M	Arb Rst GapM	Cmd RstM	Snt Bsy AckM	Hdr ErrM	TC Err M	Sub Act GapM	URxM	Cyc TLM	Cyc SecM	Cyc StM	Cyc DnM	Cyc PendM	Cyc LostM

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cyc Arb FlM	Ack Rcvd M	Ack MissM	Inv AckM	RetEx M	STOM	PB Cnt RM	U Resp M	FmtE M	TxStk M	SYT TTM	SYT TRM	Res	CIPH EM	DRFO M	DRFR M

PhyIntM **PHY Interrupt Mask** **31**
Clearing this bit masks the PHY Interrupt interrupt.

PhyRRxM **PHY Register Received Mask** **30**
Clearing this bit masks the PHY Register Received interrupt.

PhyRstM **PHY Reset Mask** **29**
Clearing this bit masks the PHY Reset interrupt.

ArbRstGapM **Arbitration Reset Gap Mask** **28**
Clearing this bit masks the Arbitration Reset Gap interrupt.

CmdRstM **Command Reset Mask** **27**
Clearing this bit masks the Command Reset interrupt.

SntBsyAckM **Sent Busy Acknowledge Mask** **26**
Clearing this bit masks the Sent Busy Acknowledge interrupt.

HdrErrM **Header Error Mask** **25**
Clearing this bit masks the Header Error interrupt.

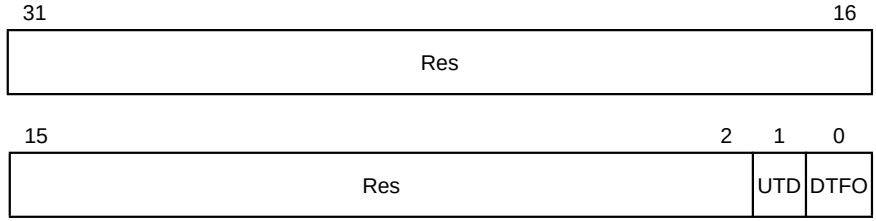
TCErrM	**Transaction Code Error Mask** Clearing this bit masks the Transaction Code Error interrupt.	**24**
SubActGapM	**Subaction Gap Mask** Clearing this bit masks the Subaction Gap interrupt.	**23**
URxM	**UBUF Packet Received Mask** Clearing this bit masks the UBUF packet Received interrupt.	**22**
CycTLM	**Cycle Too Long Mask** Clearing this bit masks the Cycle Too Long interrupt.	**21**
CycSecM	**Cycle Second Mask** Clearing this bit masks the Cycle Second interrupt.	**20**
CycStM	**Cycle Start Mask** Clearing this bit masks the Cycle Start interrupt.	**19**
CycDnM	**Cycle Done Mask** Clearing this bit masks the Cycle Done interrupt.	**18**
CycPndM	**Cycle Pending Mask** Clearing this bit masks the Cycle Pending interrupt.	**17**
CycLostM	**Cycle Lost Mask** Clearing this bit masks the Cycle Lost interrupt.	**16**
CycArbFlM	**Cycle Arbitration Failed Mask** Clearing this bit masks the Cycle Arbitration Failed interrupt.	**15**
AckRcvdM	**Acknowledge Received Mask** Clearing this bit masks the Acknowledge Received interrupt.	**14**
AckMissM	**Acknowledge Missing Mask** Clearing this bit masks the Acknowledge Missing interrupt.	**13**
InvAckM	**Invalid Acknowledge Mask** Clearing this bit masks the Invalid Acknowledge interrupt.	**12**
RetExM	**Retry attempts Exhausted Mask** Clearing this bit masks the Retry attempts Exhausted interrupt.	**11**

STOM	**Split Time Out Mask** Clearing this bit masks the Split Time Out interrupt.	**10**
PBCntRM	**Packet/Byte Count Reached Mask** Clearing this bit masks the Packet/Byte Count Reached interrupt.	**9**
URespM	**Unexpected Response Mask** Clearing this bit masks the Unexpected Response interrupt.	**8**
FmtEM	**Format Error Mask** Clearing this bit masks the Format Error interrupt.	**7**
TxStkM	**Tx Stuck Mask** Clearing this bit masks the Tx Stuck interrupt.	**6**
SYTTTM	**SYT Time Stamp Transmitted Mask** Clearing this bit masks the SYT Time Stamp Transmitted interrupt.	**5**
SYTTRM	**SYT Time Stamp Received Mask** Clearing this bit masks the SYT Time Stamp Received interrupt.	**4**
Res	**Reserved** This bit is reserved and must be written as zero.	**3**
CIPHEM	**CIP Header Error Mask** Clearing this bit masks the CIP Header Error interrupt.	**2**
DRFOM	**DBUF Receive FIFO Overflow Mask** Clearing this bit masks the DBUF transmit FIFO overflow interrupt.	**1**
DRFRM	**DBUF Receive FIFO Reception Mask** Clearing this bit masks the DBUF receive FIFO reception interrupt.	**0**

4.3.7 Interrupt 1 Register (0x028)

The fields within the Interrupt 1 Register inform the application when the state of the 1394 Node Controller core changes. When a bit is set in both the Interrupt 1 Register and its corresponding bit in the Interrupt 1 Mask Register, the 1394 Node Controller core generates an interrupt to the application. The application acknowledges the request, and writes a one

to the corresponding interrupt bit. This action resets the corresponding interrupt bit. The default value for this register is 0.



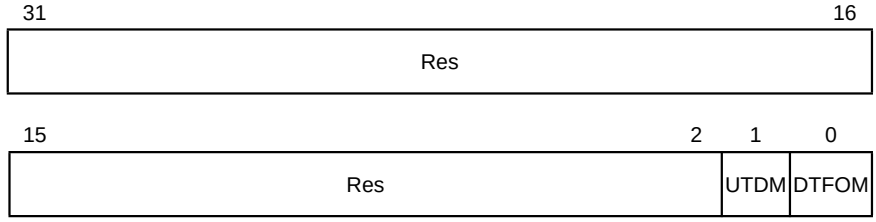
Res **Reserved** **[31:2]**
These bits are reserved and must be written as zeros.

UTD **UBUF Transmission Done** **1**
This bit is set if transmission attempts of a packet in the UBUF transmit FIFO are done (due to receipt of a pending, complete, or missing ack or busy/data error retries are exhausted, a bus reset is detected, a PHY or broadcast packet is transmitted, etc.).

DTFO **DBUF Transmit FIFO Overflow** **0**
This bit is set if during a transmission, the DBUF transmit FIFO overflows (PAct is cleared (if ELis not set) and PStk is set).

4.3.8 Interrupt 1 Mask Register (0x02C)

The Interrupt 1 Mask Register determines whether the interrupts specified in the Interrupt 1 Register are activated or not. To mask a particular interrupt, clear its corresponding bit in the Interrupt 1 Mask Register. The application interface can read and write all bits in this register. The default value for each bit in this register is zero.



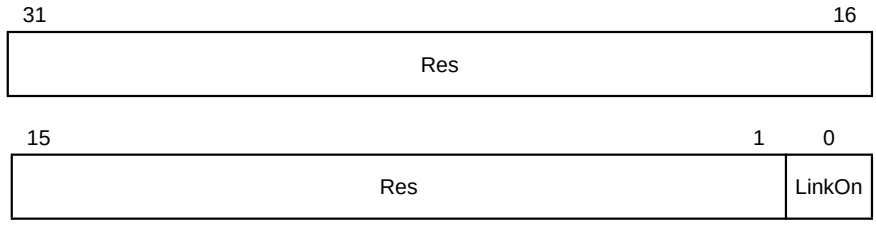
Res **Reserved** **[31:2]**
These bits are reserved and must be written as zeros.

UTDM **UBUF Transmission Done Mask** **1**
Clearing this bit masks the UBUF Transmission Done interrupt.

DTFOM **DBUF Transmit FIFO Overflow Mask** **0**
Clearing this bit masks the DBUF Transmit FIFO overflow interrupt.

4.3.9 Interrupt 2 Register (0x030)

The fields within the Interrupt 2 Register inform the application when the state of the 1394 Node Controller core changes. When a bit is set in the Interrupt 2 Register and its corresponding bit is set in the Interrupt 2 Mask Register, the 1394 Node Controller core generates an interrupt to the application. The application acknowledges the request and writes a one to the corresponding interrupt bit. This action resets the corresponding interrupt bit. The default value for this register is 0. The clock of this register is the application clock unlike most of the other registers.



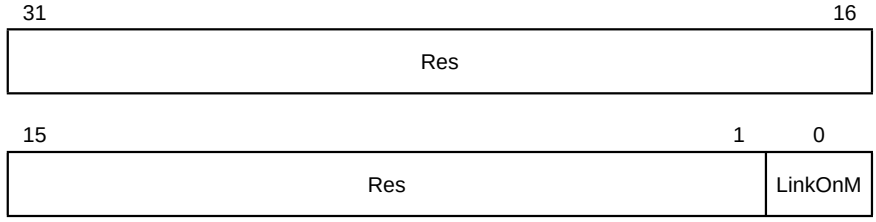
Res **Reserved** **[31:1]**
These bits are reserved and must be written as zeros.

LinkOn **Link On** **0**
This bit is set whenever the LinkOn input has been asserted. Clear this bit only after setting LPSEn or this bit will be set again because the LinkOn input will still be asserted.

4.3.10 Interrupt 2 Mask Register (0x034)

The Interrupt 2 Mask Register determines whether the interrupts specified in the Interrupt 2 Register are activated or not. Each bit in this register corresponds to an interrupt flag bit in the Interrupt 2 Register. To mask a particular interrupt, clear its corresponding bit in the Interrupt 2

Mask Register. The application interface can read and write all bits in this register. The default value for each bit in this register is zero. The clock of this register is the application clock unlike most of the other registers.

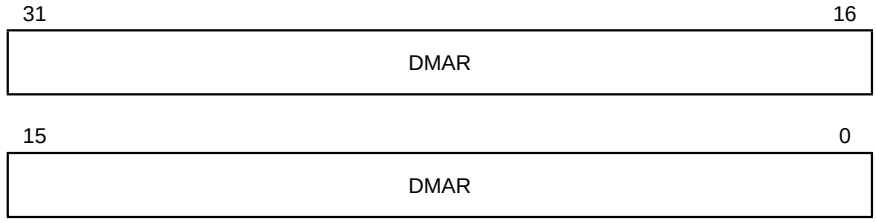


Res **Reserved** **[31:1]**
These bits are reserved and must be written as zeros.

LinkOnM **Link On Mask** **0**
This bit is the mask bit for the LinkOn interrupt.

4.3.11 DMA Space Register (0x038)

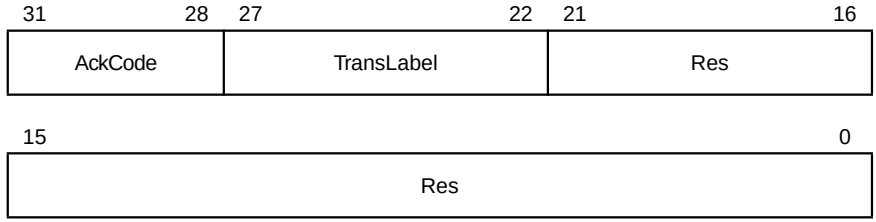
This register contains information used in the transfer of a DMA Space response packet. This information is used for transfers to/from the DBUF only.



DMAR **DMA Region** **[31:0]**
Default: 0 **RW**
This field contains the value compared against the upper 32 bits of the destination offset field of a received read or write request to determine if a DMA space access is being made. A DMA space access is being made if the upper 32 bits of the destination offset field of a received read or write request is less than DMAR.

4.3.12 Acknowledge Status Register (0x03C)

This register stores the acknowledge received for the last request or response packet transmitted from the UTF FIFO. This register also stores the transaction label for the last UTF packet transmitted. The application can read this register, but cannot write to it.



Ack Code **Acknowledge Code** **[31:28]**
Default: 0 **RO**
This field contains the acknowledge code received. This field is not updated after transmissions of PHY, broadcast, or any other transmitted packets for which an ack is not received.

TransLabel **Transaction Label** **[27:22]**
Default: 0 **RO**
This field contains the transaction label.

Res **Reserved** **[21:0]**
These bits are reserved for future use. They must be written as zeros.

4.3.13 UBUF Transmit Next (0x040)

This register is not physically within the 1394 Node Controller core. The core treats a write access to this register like a write to the UTF FIFO. The application writes a transmit packet into this FIFO for the first (n – 1) quadlets of the packet (packet size is n quadlets).

4.3.14 UBUF Transmit Last (0x044)

This register is not physically within the 1394 Node Controller core. The core treats a write access to this register like a write to the UTF FIFO. The application writes the last quadlet for the transmit packet into this address, which confirms that this address is the last quadlet of the

packet. Once a quadlet is written into the location, the status signals are updated on the LLC side of the FIFO.

4.3.15 UBUF Transmit Clear (0x048)

This register is not physically within the 1394 Node Controller core. The core treats a write access to this register like a request to initialize the UTF FIFO pointers. The application writes this location first to prepare the UTF FIFO before loading a packet into the UTF FIFO for transmission.

4.3.16 UBUF Receive Clear (0x04C)

This register is not physically within the 1394 Node Controller core. The core treats a write access to this register like a request to initialize the URF FIFO pointers. The application need not write to this location during normal operation.

4.3.17 UBUF Receive (0x050)

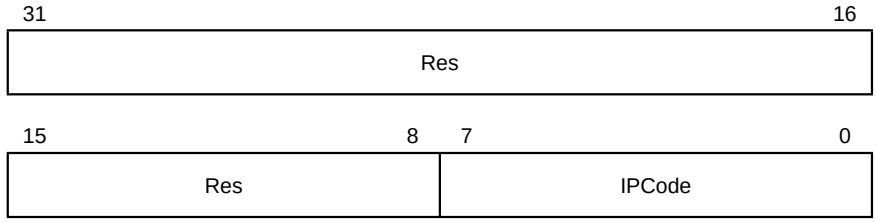
This register is not physically within the 1394 Node Controller core. The core treats a read to this register like a read of the URF FIFO.

4.3.18 UBUF Receive Level (0x054)

This register is the UBUF Receive Level, synchronized to the application interface clock. After a packet is received in the URF FIFO, this register should be read to determine the size of the packet in quadlets before reading the packet from the URF FIFO.

4.3.19 Encoded Interrupt Priority (0x058)

This register contains a code that indicates the highest priority unmasked interrupt that is set.



Res **Reserved** **[31:8]**

These bits are reserved for future use. They must be written as zeros.

IPCode **Interrupt Priority Code** **[7:0]**

Default: 0 **RW**

The IPCode field contains the code associated with the highest priority unmasked interrupt that is set. The interrupts are prioritized as shown in the following table.

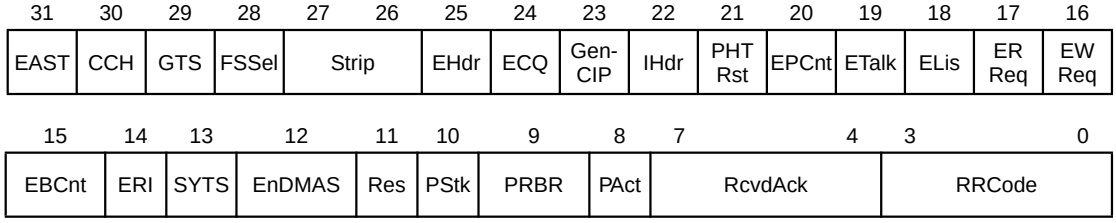
Interrupt	Priority Code
PhyRst	0x84
CmdRst	0x80
HdrErr	0x7C
TCErr	0x78
TxStk	0x74
FmtE	0x70
InvAck	0x6C
AckMiss	0x68
RetEx	0x64
Uresp	0x60
DRFO	0x5C
DFTO	0x58
CycTL	0x54
CycLost	0x50
CycArbFl	0x4C
PhyInt	0x48
PhyRRx	0x44
Urx	0x40
CIPHE	0x3C
SntBsyAck	0x38
DRFR	0x34
UTD	0x30
AckRcvd	0x2C
PCntR	0x28
SYTTT	0x24
SYTTR	0x20
ArbRstGap	0x1C

Interrupt	Priority Code
SubActGap	0x18
CycSt	0x14
CycDn	0x10
STO	0x0C
CycPnd	0x08
CycSec	0x04
No interrupt	0x00

4.3.20 PHT Control and Status Register 0 (0x080)

The settings within the PHT Control register enable/disable various functions of the 1394 Packet Header Transformation module. All bits in this register can be read and written from the application interface.

Note: Program all PHT configuration bits in other registers before enabling one of (EAST, ERReq, EWReq, EnDMAS) OR (ELis and/or ETalk).



EAST **Enable Asynchronous Stream Transmit** **31**
Default: 0 **RW**

When this bit is set, 1394 asynchronous stream packets (isochronous transmit format 1 packets sent during the asynchronous interval) are generated while EPCnt is set and the PBCnt field of the Data Transfer Control Register is not zero, and the threshold condition of the DFill field of the Data Transfer Control Register has been met. When this bit is set, 1394 asynchronous stream packets are generated while EPCnt is not set and the threshold condition of the DFill field of the Data Transfer Control Register has been met. EWReq, ERReq, EnDMAS, and ETalk must not be set when this bit is set. This bit is only sampled between transfers. If the PRBR bit becomes set while this bit and the PAct bit are set, the PAct bit is

cleared and the PStk bit is set because the channel number must be reallocated.

CCH **Check CIP Header** **30**
Default: 0 **RW**
When this bit is set, received isochronous packets are checked to determine if the EOH and Form bits of the CIP header conform to the two-quadlet CIP Header format. If the packet does not conform, the CIPHE interrupt is set.

GTS **Generate Time Stamp** **29**
Default: 0 **RW**
When this bit, GenCIP, and ETalk are set, then a time stamp is generated and is included in the transmitted packet. The most-significant bit of the FMT field of the CIP Header Transmit 1 Register determines if there is a SYT time stamp. The SPH bit of the CIP Header Transmit 0 Register determines if there is a source packet header time stamp.

FSSel **Frame Sync Select** **28**
Default: 0 **RW**
When this bit is set, the FSYNCIN input signal is used to control time stamping when GenCIP and the most-significant bit of the FMT field of the Stream Control Transmit 1 Register is 0 (for example, DVCR). The FSYNCIN input should be asserted during the first write strobe of a packet in which a SYT time stamp should be generated by the core. If this bit is not set and SYT time stamping is desired, the ECQ bit must be set to enable the alternative method (SYT time stamping occurs for each packet whose frame start bit is set in the application prepended embedded control quadlet).

Strip **Strip CIP Header** **[27:26]**
Default: 0 **RW**

This field controls how CIP headers are handled when listening to an isochronous channel.

Strip	Description
00	Unchanged.
01	Strip all CIP headers (also subtract 8 bytes from data packet length in received isochronous header sent to DBUF)
10	Strip only empty CIPs (if IHdr is set, when an empty CIP is encountered the header is not written to the DBUF)
11	Reserved.

EHdr **Embedded Header** **25**
Default: 0 **RW**

When this bit is set, it is required that the header of the packet (as shown in isochronous transmit format 3 or the read response for block data packet transmit format) is prepended to the data field of a packet to be transmitted. The header registers are not used. For unformatted isochronous data fields (tag = 0 in header), multichannel talk capability can be supported. When this bit is set and ETalk is set, ECQ should also be set. The embedded control quadlet is needed to determine the speed of transmission of the packet.

ECQ **Embedded Control Quadlet** **24**
Default: 0 **RW**

When this bit is set, it is required that a control quadlet is prepended to the data field of an isochronous packet to be transmitted (the control quadlet enables software-controlled frame synchronization and hardware assisted isochronous data flow control (useful for DV frame rate transmission control). When GenCIP and this bit are set and the cycle skip field of the embedded control quadlet is N, N empty CIP packets are transmitted before the packet containing the data_field after the embedded control quadlet. In addition, for unformatted data fields (tag = 0), multichannel talk capability can be supported if this bit is set in addition to the EHdr bit.

GenCIP **Generate CIP Header** **23**
Default: 0 **RW**
When this bit is set, a CIP header is prepended to the data field of each isochronous packet transmitted. Do not set this bit if talking on more than one isochronous channel. If this bit is set, for each DMA request assertion, the DMA controller must write less than or equal to the number of source packets transmitted in a 1394 payload.

IHdr **Include Header** **22**
Default: 0 **RW**
When this bit is set, the header is written to the DBUF FIFO in addition to the data field of the 1394 packet (useful when listening to more than one isochronous channel or DMA space requests enabled).

PHTRst **PHT Reset** **21**
Default: 0 **RW**
If the PHT becomes stuck, this bit should be set (this bit automatically clears itself).

EPCnt **Enable Packet Counter** **20**
Default: 0 **RW**
When this bit is set, PHT packet transmissions stop when the PBCnt field of the Data Transfer Control Register is decremented to zero. If neither this bit nor EBCnt is set and either ERReq, EWReq, ETalk, or EAST is set, 1394 packets are transmitted indefinitely. EBCnt and EPCnt must not be set at the same time.

ETalk **Enable Isochronous Talk** **19**
Default: 0 **RW**
When this bit is set, isochronous packets are transmitted. EAST, ERReq, EnDMAS, and EWReq must not be set if this bit is set. When generating CIPs, set ETalk before setting DEn (DMA Enable). This bit is only sampled between transfers.

ELis **Enable Isochronous Listen** **18**
Default: 0 **RW**
When this bit is set, isochronous packets transmitted on channels enabled in the Stream Receive Channel Selection Registers are received. ERReq and EWReq must not be set if this bit is set. This bit is only sampled between transfers.

ERReq **Enable Read Requests** **17**
Default: 0 **RW**
When this bit is set, 1394 Read Requests are generated while EPCnt or EBCnt is set and the PBCnt field of the Data Transfer Control Register is not cleared. If this bit is set, 1394 Read Requests are generated indefinitely while neither EPCnt nor EBCnt is set and the DBUF receive FIFO has enough space to accept another response. EAST, EWReq, ETalk, EnDMAS, and ELis must not be set if this bit is set. This bit is only sampled between transfers. If the PRBR bit becomes set while this bit and the PAct bit are set, the PAct bit is cleared and the PStk bit is set because the node IDs might have changed.

EWReq **Enable Write Requests** **16**
Default: 0 **RW**
When this bit is set, 1394 Write Requests are generated while EPCnt or EBCnt is set and the PBCnt field of the Data Transfer Control Register is not cleared. If this bit is set, 1394 Write Requests are generated indefinitely while neither EPCnt nor EBCnt is set. EAST, ERReq, ETalk, EnDMAS, and ELis must not be set when this bit is set. This bit is only sampled between transfers. If the PRBR bit becomes set while this bit and the PAct bit are set, the PAct bit is cleared and the PStk bit is set because the node IDs may have changed.

EBCnt **Enable Byte Count** **15**
Default: 0 **RW**
If this bit is set, asynchronous PHT packet transmissions stop when the PBCnt field of the Data Transfer Control register is decremented to zero. The data length field of the transmitted 1394 header of the last packet transferred is adjusted appropriately if the PBCnt field is not divisible by the programmed data length. If neither this bit nor EPCnt is set and either ERReq or EWReq is set, 1394 packets are transmitted indefinitely. Both EBCnt and EPCnt must not be set at the same time. This bit must not be set when ETalk or EAST is set.

ERI **Enable Request Interval** **14**
Default: 0 **RW**
If this bit and EWReq or ERReq are set, the time interval between consecutive PHT requests as measured from the reception of an ACK or response packet of ‘complete’

or ‘busy’ for the first packet is determined by the value programmed in the RqIntrvl field.

SYTS **SYT Start** **13**
Default: 0 **RW**

If this bit is set, incoming CIP format isochronous packets are not written to the DBUF until a packet containing an SYT time stamp is received (which indicates the start of a frame).

EnDMAS **Enable DMA Space** **12**
Default: 0 **RW**

When this bit is set, automatic handling of received read and write requests is enabled. The PHT automatically handles a received request packet whose quadlet aligned address matches the DMAR field criteria. EAST, ERReq, ETalk, EWReq, and ELis must not be set when this bit is set. This bit is only sampled between transfers. If the PRBR bit becomes set while this bit and the PAct bit are set, the PAct bit is cleared and the PStk bit is set because the node IDs might have changed.

Automatic handling of read and write requests occurs when the EnDMAS bit is set, an incoming request packet's quadlet aligned address matches the DMAR field criteria, and the incoming request packet's source node ID corresponds to a selected node for the PHT (the corresponding bit in the Stream Receive Channel/NodeID Selection registers is set).

The EHdr and IHdr bits in PHT Control and Status Register 0 affect the format of the data written to the DBuf Transmit and Receive FIFOs when EnDMAS is asserted. If IHdr is not set, only the data field portion of the incoming request is written to the DBuf Receive FIFO. If IHdr is set and EHdr is not set, the 1394 header is written to the DBuf Receive FIFO in addition to the data field. If IHdr and EHdr are both set, in addition to the 1394 header and data field, a quadlet containing the speed of the received request is written to the DBuf Receive FIFO.

Note: If EHdr is set, care must be taken to guarantee that all response packets have been sent prior to disabling EnDMAS. One possibility would be to first clear the nodeID Selection registers to disable further acceptance

of incoming requests by the PHT. When it is determined that all previously accepted requests have been processed, EnDMAS can be safely cleared.

Note: If IHdr and EHdr are set, the format of the data in the DBuf is as shown in Figure 2.13 (Write Request for Data Block Receive Format). The maximum data field size that can be received is the DBuf Receive FIFO size less 20 bytes.

If EHdr is not set, the DBuf Transmit FIFO must contain only the data field for read response packets to be sent in response to received read requests. If EHdr is set, the format followed is that of Figure 2.4 or Figure 2.7.

An appropriate setting for EHdr and IHdr depends on the capability of the external DMA controller.

Res **Reserved** **11**
This bit is reserved and must be written as zero.

PStk **PHT Module Stuck** **10**
Default: 0 **RO**
When this bit is set, software must set the PHTRst bit to reinitialize the module.

PRBR **PHT Received Bus Reset** **9**
Default: 0 **RO**
This bit is set when the PAct bit is set and a Bus Reset is detected.

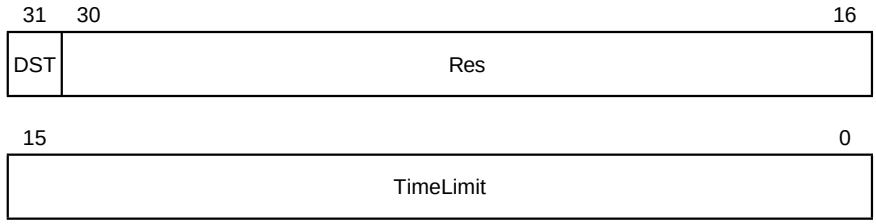
PAct **PHT Active** **8**
Default: 0 **RO**
This bit is automatically set when the PHT module is not in the idle or stuck state with the exception that if both ETalk and Elis are set and the PHT transmitter becomes stuck, the PAct bit remains set because the PHT receiver never becomes stuck.

RcvdAck **Received Acknowledge** **[7:4]**
Default: 0xF **RO**
This field contains the ack code received for the last packet transmitted. It is initialized to 0xF when a request for transmission is made.

RRCode **Received Response Code** **[3:0]**
Default: 0xF **RO**
This field contains the rcode in the last response packet received because of a request packet transmitted. It is initialized to 0xF when a request for transmission is made.

4.3.21 PHT Split Time-out/Empty CIP Interval Register 0 (0x084)

When ERReq or EWReq is set, this register contains the maximum time allowed from reception of an ack pending in reply to a request initiated by the PHT block until a response to that request is received. Application software is responsible for split time-out detection for packets transmitted using the UTF FIFO. This detection can be done using the CycSt and AckRcvd interrupts.



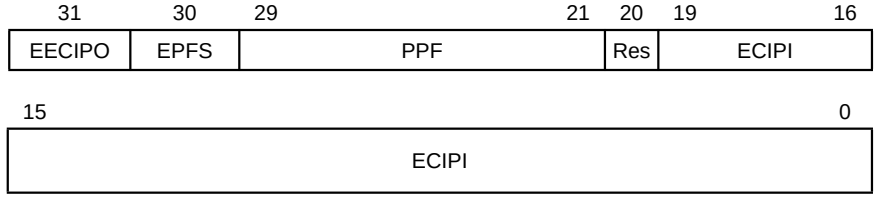
DST **Disable Split Time-Out** **31**
When this bit is set, the time limit field is ignored and there is no hardware time-out. It might be useful to set this bit when it is known the destination node is not on the local 1394 bus. In such a bridged environment, the remote split time-out might be known to be longer than the maximum value to which the TimeLimit field can be programmed, so other methods must be used for time-out detection.

Res **Reserved** **[30:16]**
These bits are reserved for future use. They must be written as zeros.

TimeLimit **Time Limit** **[15:0]**
Default: 0x0000 **RW**
Bits [15:13] correspond to seconds. Bits [12:0] correspond to fractional seconds with a 125 µs resolution (that is, equivalent resolution to the cycle count field of the Cycle Timer register). Bits [12:0] must be

programmed with a value less than 8000. The actual duration of time between reception of an ack pending and nonreception of a response packet that causes the STO interrupt to be set can differ as much as 125 microseconds less than the programmed value.

When ETalk is set, the register is defined as follows.



EECIPO **Enable Empty CIP Override** **31**
Default: 0 **RW**
When this bit, GenCIP, and ETalk, etc. are set, an empty CIP packet is sent during the isochronous cycle subsequent to each time an internal counter increments to the same value as the ECIPI field.

EPFS **Enable Programmable Frame Synchronization** **30**
Default: 0 **RW**
When this bit, GenCIP, ETalk, etc. are set, an SYT time stamp is generated for the first nonempty CIP packet transmitted and every N+1 nonempty CIP packets transmitted thereafter, where N is the value of the PPF field.

PPF **Packets Per Frame** **[29:21]**
Default: 0 **RW**
This field is only used when the EPFS bit is set. It is programmed with the number of packets per frame minus one (for example, for a 525-60 system, the value is 249 decimal).

Res **Reserved** **20**
These bits are reserved for future use. They must be written as zeros.

ECIPI **Empty CIP Interval** **[19:0]**
Default: 0 **RW**
This field is compared against an internal counter that increments every other SClk (49.152 MHz clock) from the PHY. The counter is reset to zero when a match occurs.

4.3.22 PHT Request/Response/CIP Receive Header 0 Register 0 (0x088)

This register contains information used in the transfer of a request packet if ERReq or EWReq is set or a response packet if EnDMAS is set or a received isochronous packet if ELis is set. This information is used for transfers to/from the DBUF only.

If ERReq or EWReq is set, the register is defined as follows.

DID **Destination ID** **[31:16]**
Default: 0 **RW**
This field is the concatenation of the Bus ID and the Node ID of the destination. The broadcast node ID of 0b111111 is not supported.

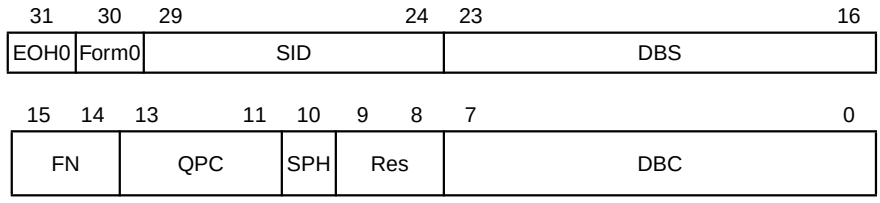
DOffHi **Destination Offset Hi** **[15:0]**
Default: 0 **RW**
This field contains the most-significant 16 bits of the destination offset address. This field must not be programmed with a value that would result in an overflow when this field is automatically updated during multipacket transmission.

If EnDMAS is set, the definition is as follows (note that if EHdr is not set, the contents of this register do not always reflect the last request received and should not be relied on for that purpose).

DestID **Destination ID** **[31:16]**
Default: 0x0000 **RW**
This field is the concatenation of the Bus ID and the Node ID of the node that initiated the DMA space request.

DLen **Data Length** **[15:0]**
Default: 0 **RW**
This field contains the data length in bytes of the received DMA Space request, which is reused if a read response is generated.

When ELis is set, this register is defined as follows.



EOH0 **End Of CIP Header 0** **31**
Default: 0 **RO**
A value of 0 means another header quadlet will follow (always zero for a standard two-quadlet CIP).

Form0 **Form 0** **30**
Default: 0 **RO**
This field is always zero for a standard two-quadlet CIP.

SID **Source ID** **[29:24]**
Default: 0 **RO**
This field contains the node ID of the transmitter.

DBS **Data Block Size** **[23:16]**
Default: 0 **RO**
This field contains the data block size in quadlets.

FN	**Fraction Number** **Default: 0** This field contains the number of data blocks into which a source packet is divided.	**[15:14]** **RO**
QPC	**Quadlet Padding Count** **Default: 0** This field contains the number of dummy quadlets padded to a source packet to equalize the size of divided data blocks.	**[13:11]** **RO**
SPH	**Source Packet Header** **Default: 0** A value of one on this bit indicates that the source packet has its own header.	**10** **RO**
Res	**Reserved** These bits are reserved for future use.	**[9:8]**
DBC	**Data Block Counter** **Default: 0** This field contains the continuity counter of data blocks to detect a loss of data blocks. The counter value for the first data block in a bus packet is shown in this field.	**[7:0]** **RO**

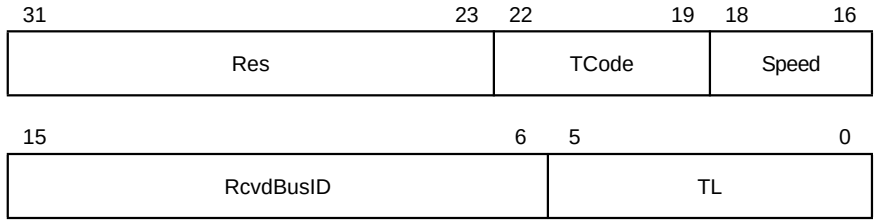
4.3.23 PHT Request/Response/CIP Receive Header 1 Register 0 (0x08C)

This register contains information used in the transfer of a request packet (if ERReq or EWReq is set) or a response packet (if EnDMAS is set), or a received isochronous packet (if ELis is set). This information is used for transfers to/from the DBUF only.

When ERReq or EWReq is set, this register is defined as follows.

DOffLo **Destination Offset Lo** **[31:0]**
Default: 0 **RW**
This field contains the least-significant 32 bits of the destination offset address.

If EnDMAS is set, this register is defined as follows (note that if EHdr is not set, the contents of this register do not always reflect the last request received and must not be relied on for that purpose).



Res **Reserved** **[31:23]**
These bits are reserved for future use. They must be written as zeros.

TCode **Transaction Code** **[22:19]**
Default: 0 **RW**
This field contains the transaction code of the DMA space read or write request.

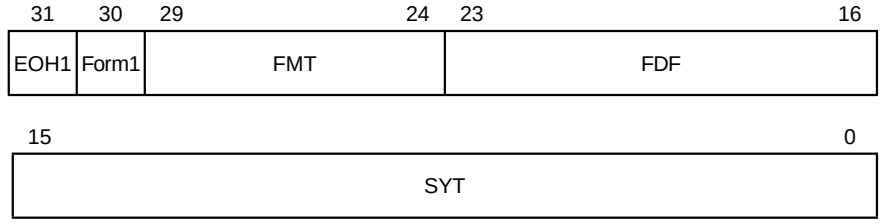
Speed **Speed** **[18:16]**
Default: 0 **RW**
This field contains the transmit speed of the DMA space read or write request.

Speed[13:15]	Encoding
000	S100 Speed
010	S200 Speed
100	S400 Speed

RcvdBusID **Received Bus ID** **[15:6]**
Default: 0 **RW**
This field contains the Destination Bus ID used in a DMA space read request. This value is used as the Source Bus ID in the read response

TL **Transaction Label** **[5:0]**
Default: 0x0000 **RW**
This field contains the transaction label of the received DMA space read or write request.

When ELis is set, this register is defined as follows.



EOH1 **End Of CIP Header 1** **31**
Default: 0 **RO**
A value of one on this bit means another header quadlet does not follow (should always be one for standard two-quadlet CIP).

Form1 **Form 1** **30**
Default: 0 **RO**
This bit must always be zero for a standard two-quadlet CIP.

FMT **Format ID** **[29:24]**
Default: 0 **RO**
The encoding for this field is shown in the table below.

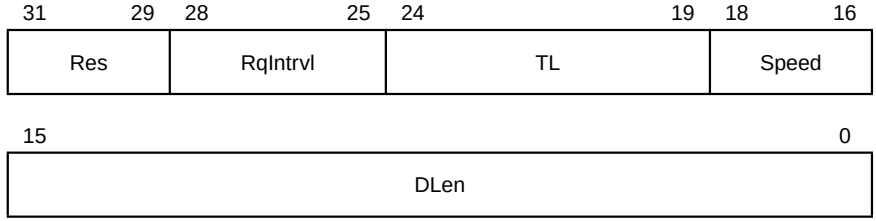
FMT	Description
0x00	When the most-significant bit (bit 2) is zero, a time stamp may be in the SYT field (such as in the DVCR format).
0x20	When the most-significant bit (bit 2) is one, a time stamp is not in the SYT field (such as in the MPEG format).

FDF **Format Dependent Field** **[23:16]**
Default: 0 **RO**
This field is defined for each format.

SYT **Synchronization Time Field** **[15:0]**
Default: 0 **RO**
For DVCR, this field can contain a time stamp of the frame synchronization pulse. For MPEG2, this field is 0.

4.3.24 PHT Request Header 2/Stream SPH Receive Register 0 (0x090)

If ERReq or EWReq is set, this register contains information used in the transfer of a request packet. This information is used for transfers to/from the DBUF only.



Res **Reserved** **[31:29]**
These bits are reserved for future use.

RqIntrvl **Request Interval** **[28:25]**
This four-bit field contains the minimum number of desired nominal 125 µs cycles between reception of an ack or response of 'complete' or 'busy' for a previous PHT generated request and transmission of the next PHT generated request. The actual observed time between any two requests can be as much as 125 µs less than the programmed value (to guarantee at least a 125 µs interval, the programmed value should be two).

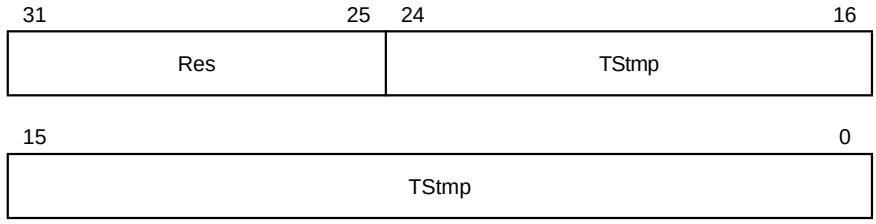
TL **Transaction Label** **[24:19]**
Default: 0x0 **RW**
These six bits indicate the transaction label for the packet. They are used for tracking requests with responses. PHT module 0 must be programmed with 0x3F. PHT module 1 must be programmed with 0x3E.

Speed **Speed** **[18:16]**
Default: 0 **RW**
This field contains the transmit speed of the DBUF read or write request as shown in the following table.

Speed[13:15]	Encoding
000	S100 Speed
010	S200 Speed
100	S400 Speed

DLen **Data Length** **[15:0]**
Default: 0 **RW**
This field contains the data length in bytes of the payload to be written or read. After an ack or rcode of ‘complete’ is received in response to a transmitted request, the destination offset field of the PHT Request Header Register is updated with the addition of the Data Length field.

When ELis is set, the register is defined as follows. This register contains the time stamp from the last received packet with an isochronous header tag field of 0b01 and a CIP header SPH bit of 0b1.

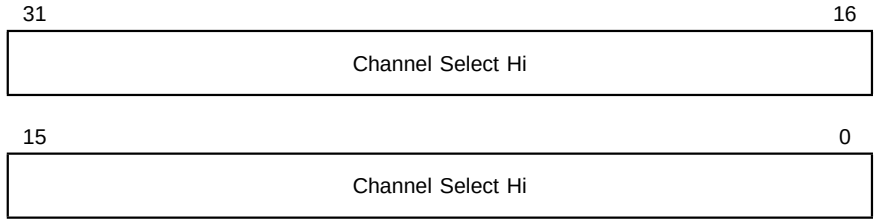


Res **Reserved** **[31:25]**
These bits are reserved for future use. They must be written as zeros.

TStmp **Time Stamp** **[24:0]**
Default: 0 **RW**
This field contains the time stamp of the packet.

4.3.25 Stream Receive Channel/NodeID Selection 0 Register 0 (0x094)

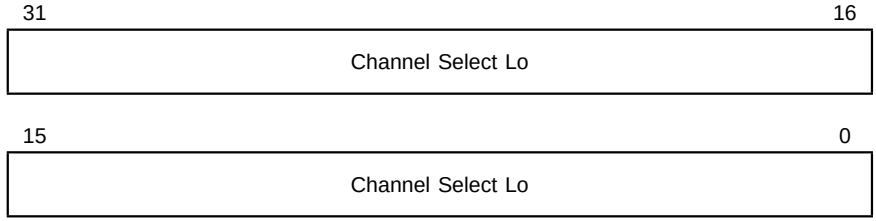
The settings within the Stream Receive Channel Selection registers enable/disable isochronous channel reception when ELis is set or enable/disable DMA Space read/write requests from other nodes when EnDMAS is set.



ChSelHi **Channel Select Hi** **[31:0]**
Default: 0 **RW**
This field contains the reception enable bits for stream channels 63 (bit 31) to 32 (bit 0) or the read/write request reception enable bits for broadcasts (bit 31) or nodes 62 (bit 30) to 32 (bit 0).

4.3.26 Stream Receive Channel/NodeID Selection 1 Register 0 (0x098)

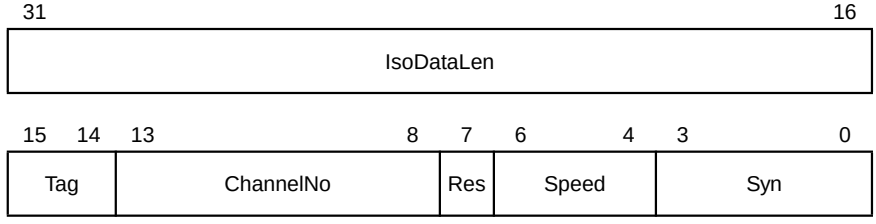
The settings within the Stream Receive Channel Selection registers enable/disable isochronous channel reception when ELis is set or enable/disable DMA Space read/write requests from other nodes when EnDMAS is set.



ChSelLo **Channel Select Lo** **[31:0]**
Default: 0 **RW**
This field contains the reception enable bits for stream channels 31 (bit 31) to 0 (bit 0) or the read/write request reception enable bits for nodes 31 (bit 31) to 0 (bit 0).

4.3.27 Stream Receive Channel Header Register 0 (0x09C)

This register contains the received information in the header quadlet of a stream packet (tcode equals 0xA Isoch or Async).



IsoDataLen **Iso Packet Data Length** **[31:16]**
Default: 0 **RO**
This field indicates the number of data bytes in the current packet. Transmission of zero data length packets are not supported.

Tag **Tag** **[15:14]**
Default: 0 **RO**
This field contains the tag of the stream packet.

ChannelNo **Channel Number** **[13:8]**
Default: 0 **RO**
This field contains the channel number of the stream packet.

Res **Reserved** **7**
This bit is reserved for future use. It must be written as zero.

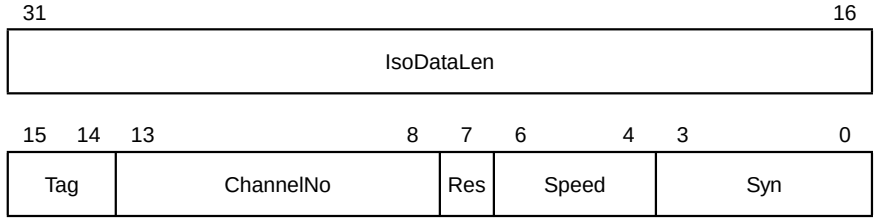
Speed **Speed** **[6:4]**
Default: 0 **RO**
This field contains the speed at which the current packet was received as shown in the following table.

Speed[25:27]	Encoding
000	S100 Speed
010	S200 Speed
100	S400 Speed

Syn **Sync** **[3:0]**
Default: 0 **RO**
This field is the sync field of the stream packet.

4.3.28 Stream Transmit Channel Header Register 0 (0x0A0)

This register contains the information in the header quadlet of a stream packet (tcode equals 0xA Isoch or Async). This information is used for stream transmits only if the Embedded Header bit is not enabled (otherwise the header information is obtained from the DBUF). This register is used for single stream transmit mode in which the DBUF contains isochronous transmit format 1 or 2 information to be transmitted. The speed field generates speed bits when sending an Lreq on the PHY/link interface.



IsoDataLen **Iso Packet Data Length** **[31:16]**
Default: 0 **RW**
This field indicates the number of data bytes in the current packet. This device does not support transmission of zero data length packets.

Tag **Tag** **[15:14]**
Default: 0 **RW**
This field contains the tag of the stream packet.

ChannelNo **Channel Number** **[13:8]**
Default: 0 **RW**
This field contains the channel number of the stream packet.

Res **Reserved** **7**
This bit is reserved for future use. It must be written as zero.

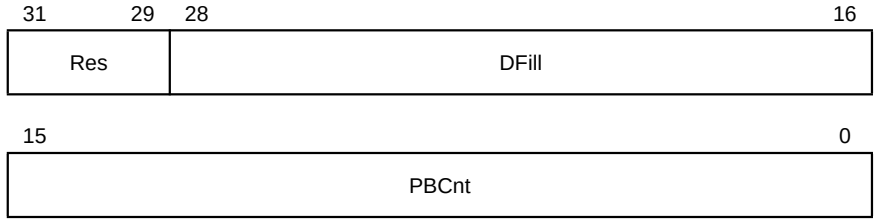
Speed **Speed** **[6:4]**
Default: 0 **RW**
This field contains the speed at which the current Isochronous packet has to be transmitted.

Speed[25:27]	Encoding
000	S100 Speed
010	S200 Speed
100	S400 Speed

Syn **Sync** **[3:0]**
Default: 0 **RW**
This field is the sync field of the stream packet.

4.3.29 Data Transfer Control Register 0 (0x0A4)

This register contains information used in the transfer of packets. This information is used for outbound 1394 DBUF transfers only.



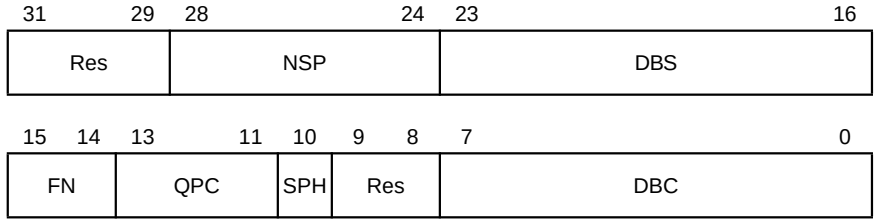
Res **Reserved** **[31:29]**
These bits are reserved for future use. They must be written as zeros.

DFill **DBUF Threshold Fill Trigger** **[28:16]**
Default: 0 **RW**
The number of bytes in the DBUF transmit FIFO must be greater than this value before a stream request is made. When in isochronous talk mode, if the DBUF transmit FIFO level is not greater than this value, an empty CIP packet is sent if GenCIP is set. No isochronous packet is sent in isochronous talk mode if GenCIP is not set and the DBUF transmit FIFO level is not greater than this value. This field is useful when the stream packet size or sum of simultaneous isochronous streams is greater than the FIFO size, and it is known that the sustained DMA data rate is sufficient to prevent underflow once transmission starts. When the ECQ bit is set, DFill must be programmed no higher than the total number of bytes to be read from the FIFO per isochronous phase less five bytes.

PBCnt **Packet/Byte Count** **[15:0]**
Default: 0 **RW**
This field can be written with the desired packet/byte transmit count before enabling PHT transfers. This field is decremented after each packet transmission (when EPCnt or EBCnt is set). When the field reaches zero, the PBCntR interrupt is set and transfers stop.

4.3.30 CIP Header Transmit 0 Register 0 (0x0A8)

This register contains information used in the isochronous transfer of an IEC-61883 format packet. This information is used for transfers from the DBUF only. This register must not be written while ETalk is one. This register is not affected by PHTRst.



Res **Reserved** **[31:29], [9:8]**
These bits are reserved and read as zeros.

NSP **Number of Source Packets** **[28:24]**
Default: 1 **RW**
This field specifies the number of source packets transmitted when a nonempty CIP packet is sent (used to control DBC field incrementing). This value should be consistent with the IsoDataLen field of the stream transmit control header register (for example, if five 188-byte MPEG + 4-byte time stamp (core generated source packet header) packets are to be transmitted in one 1394 packet, then NSP should be 5 and IsoDataLen should be 5 x (4 + 188) + 8 = 968. NSP must be one or more.

DBS **Data Block Size** **[23:16]**
Default: 0 **RW**
This field contains the data block size in quadlets.

FN **Fraction Number** **[15:14]**
Default: 0 **RW**
This field contains the number of data blocks into which a source packet is divided.

FN	Description
0b00	Not Divided (for example, DVCR). The DBC LSB increments after every nonempty CIP packet transmission.
0b11	1/8 source packet (for example, MPEG). The three LSBs of DBC are always 0b000 because an integer multiple of source packets is transmitted or an empty CIP per cycle occurs.

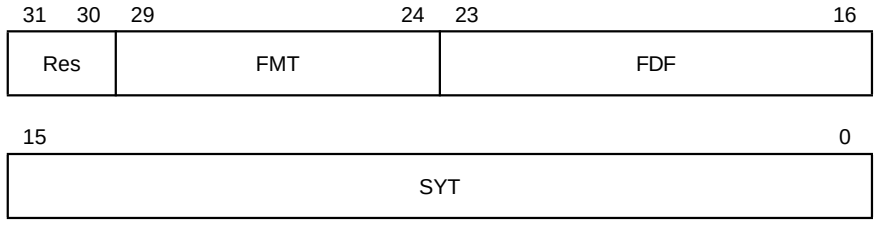
QPC **Quadlet Padding Count** **[13:11]**
Default: 0 **RW**
This field contains the number of dummy quadlets padded to a source packet by the application to equalize the size of divided data blocks.

SPH **Source Packet Header** **10**
Default: 0 **RW**
A value of one on this bit indicates that the source packet has its own header.

DBC **Data Block Counter** **[7:0]**
Default: 0 **RO**
This field contains the continuity counter of data blocks to detect a loss of data blocks. The counter value for the first data block in a 1394 packet is shown in this field.

4.3.31 CIP Header Transmit 1 Register 0 (0x0AC)

This register contains information used in the transfer of a IEC-61883 format packet. This information is used for transfers to the DBUF only. This register must not be written while ETalk is one. This register is not affected by PHTRst.



Res **Reserved** **[31:30]**
These bits are reserved and read as zeros.

FMT **Format ID** **[29:24]**
Default: 0 **RW**
The value for this field is application dependent.

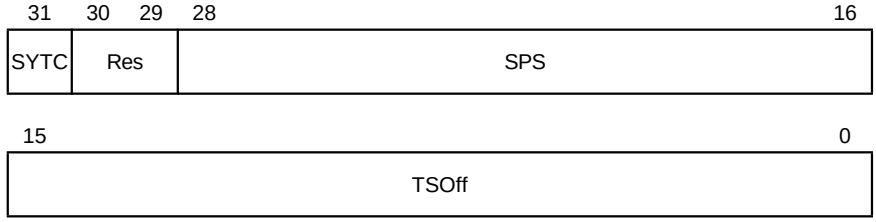
FMT	Description
0x00 (DVCR)	If the most-significant bit (bit 2) is a 0, it indicates that a time stamp may be generated by the PHT module in the transmitted SYT field (such as DVCR format).
0x20 (MPEG)	If the most-significant bit (bit 2) is a 1, the bits in the SYT field are always transmitted unmodified (used for MPEG format).

FDF **Format Dependent Field** **[23:16]**
Default: 0 **RW**
This field is defined for each format.

SYT **Synchronization Time Field** **[15:0]**
Default: 0 **RW**
For DVCR format, the transmitted version of this field is sometimes replaced with a PHT-module-generated time stamp of the frame synchronization pulse. For MPEG2 format, this field should be 0x0000.

4.3.32 Stream Transmit Time Stamp Offset Register 0 (0x0B4)

This register contains the time stamp offset used for transmitting a packet with an isochronous header tag field of 0b01, if the GTS bit is set. This register is not affected by PHTRst.



SYTC **SYT Carry** **31**
When this bit is set, a carry occurred between the cycle offset and cycle count field when generating SYT time stamps (see the definition of the TSOff field below).

Res **Reserved** **[30:29]**

These bits are reserved for future use. They must be written as zeros.

SPS **Source Packet Size** **[28:16]**

This field contains the size in bytes of a source packet less any source packet header. For example, in the case of MPEG2 transport stream packets, the appropriate value is 188 bytes. This field is only used when transmitting IEC-61883 format streams.

TSOff **Time Stamp Offset** **[15:0]**

Default: 0 **RW**

This field contains the time stamp offset added to the Cycle Time Register to produce the time stamp.

For SYT time stamps (when the SYTC bit of the PHT Control and Status register is zero), bits [11:0] are added to the cycle_offset field[1] and bits [15:12] are added to the four LSBs of the cycle_count field to produce the time stamp.

For SYT time stamps (when the SYTC bit of the PHT Control and Status register is one), bits [11:0] are added to the cycle_offset field[2] and bits [15:12] are added to the four LSBs of the cycle_count field to produce the time stamp.

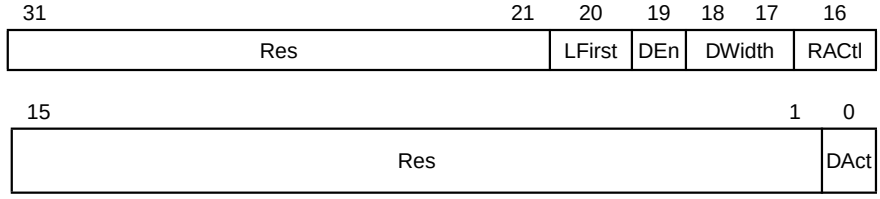
For SPH time stamps, bits [11:0] are added to the cycle_offset field of the cycle timer. If the result is greater than 3071, the resulting SPH cycle offset value transmitted is (cycle offset sum – 3072). A carry of one is made to the addition of bits [15:12] and the cycle count field of the cycle timer. If the sum of bits [15:12], the cycle count field of the cycle timer, and the carry (if any) is greater than 7999, then the resulting SPH cycle count value is (cycle count sum – 8000). The cycle offset portion (bits [11:0]) of TSOff must not be programmed with a value greater than 3071.

---

1. There is no carry to the next significant nibble of the time stamp, and the result is limited to 0xBFF (any addition that would result in a value greater than 0xBFF does not wrap to zero - the result is 0xBFF).
2. Carry to the next significant nibble of the time stamp (any addition that would result in a value greater than 0xBFF does wrap around).

4.3.33 DMA Control and Status Register 0 (0x0B8)

The settings within the DMA Control Register enable/disable various functions of the 1394 slave DMA module. All bits in this register can be read and written from the application interface.



Res **Reserved** **[31:21], [15:1]**
These bits are reserved for future use. They must be written as zeros.

LFirst **Least Significant First** **20**
Default: 0 **RW**
Data is processed on a quadlet basis within the device. If this bit is not set, then on an 8-bit wide DMA interface, the first byte output/input from/to the device is the most significant byte. If the least significant byte is output/input first, this bit must be set. Similarly, on a 16-bit interface, the first two bytes output/input from/to the device are the two most-significant bytes, if this bit is not set.

DEn **DMA Enable** **19**
Default: 0 **RW**
When this bit is set, the DMA interface is enabled.

DWidth **DMA Width** **[18:17]**
Default: 0b10 **RW**
The following table lists the encoding for this field.

DWidth	Description
00	8-bit interface
01	16-bit interface
10	32-bit interface
11	Reserved value

RACtl **Request Acknowledge Control** **16**
Default: 1 **RW**
When this bit is cleared, all DMA transfers use the DINT0_DREQP and DMAC0_DACKP signals. When this bit is set, all packets read from the DBUF use DINT0_DREQP and DMAC0_DACKP while packets written to the DBUF use DINT0_DREQWP and DMAC0_DACKWP.

DAct **DMA Active** **0**
Default: 0 **RO**
When this bit is set, the DMA interface is active (due to data to be read by external controller or data needed to be written from external controller).

4.3.34 DMA Transfer Threshold Register 0 (0x0BC)

This register contains information used in the transfer of packets. This information is used to control DMA transfers.

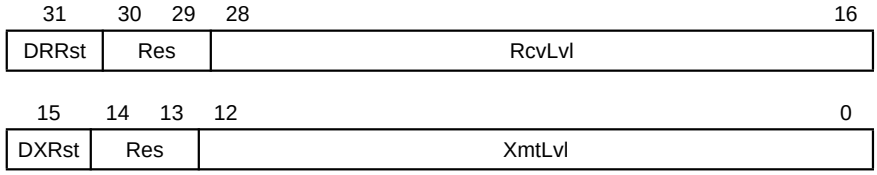
Res **Reserved** **[31:29], [15:13]**
These bits are reserved for future use. They must be written as zeros.

RdLvl **Read Level** **[28:16]**
Default: 0 **RW**
The number of bytes in the DBUF receive FIFO must be greater than this value before the appropriate DINT_DREQP signal is asserted to notify the external DMA controller to read the FIFO.

WrLvl **Write Level** **[12:0]**
Default: 0 **RW**
The number of bytes empty in the DBUF transmit FIFO must be greater than this value before the appropriate DINT_DREQP signal is asserted to notify the external DMA controller to write the FIFO.

4.3.35 DBUF FIFOs Level Register 0 (0x0C0)

This register contains the number of bytes in the DBUF FIFOs.



DRRst **DBUF Receive FIFO Reset** **31**
Default: 0 **RW**
When set, this bit clears the DBUF Receive FIFO pointers. This bit automatically clears itself.

Res **Reserved** **[30:29], [14:13]**
These bits are reserved for future use. They must be written as zeros.

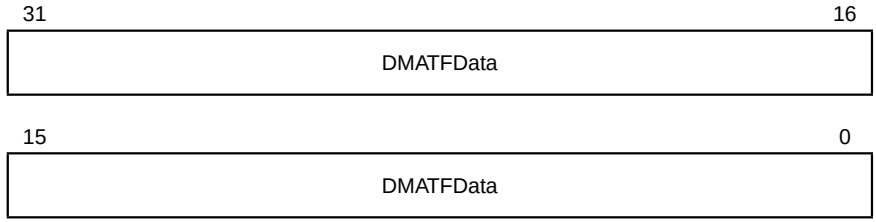
RcvLvl **Receive Level** **[28:16]**
Default: 0 **RO**
This field contains the number of bytes in the DBUF receive FIFO.

DXRst **DBUF Transmit FIFO Reset** **15**
Default: 0 **RW**
When set, this bit clears the DBUF Transmit FIFO pointers. This bit automatically clears itself.

XmitLvl **Transmit Level** **[12:0]**
Default: 0 **RO**
This field indicates the number of bytes in the DBUF transmit FIFO.

4.3.36 DBUF Tx Data Register 0 (0x0C4)

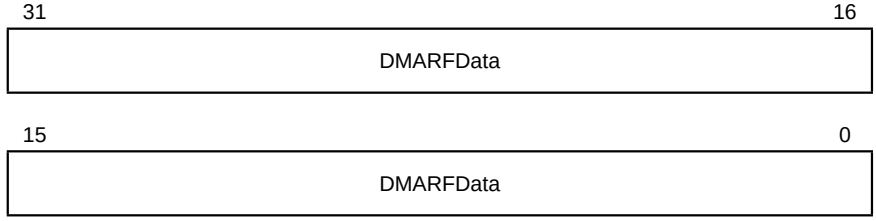
When the DMA interface is disabled, the software writes to this register in order to perform writes to the Tx DBUF FIFO. The software uses this register to directly place data in the Tx DBUF, which the PHT can then transmit.



DMATFData **DMA Transmit FIFO Data** **[31:0]**
Default: None **WO**
This field contains the data value to be written to the Tx DBUF FIFO.

4.3.37 DBUF Rx Data Register 0 (0x0C8)

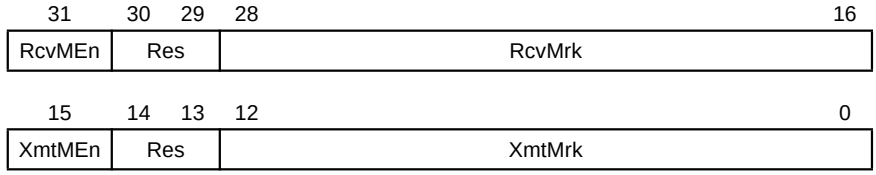
When the DMA interface is disabled, the software reads from this register in order to perform reads from the Rx DBUF FIFO. The software uses this register to directly read data that the PHT has received.



DMARFData **DMA Receive FIFO Data** **[31:0]**
Default: None **RO**
This field contains the data value that is read from the Rx DBUF FIFO.

4.3.38 DBUF FIFOs Watermark Level Register 0 (0x0CC)

This register contains the highest number of bytes in the DBUF FIFOs.



RcvMEn **DBUF Receive FIFO Watermark Enable** **31**
Default: 0 **RW**
When this bit is cleared, the RcvMrk field is reset to 0. When this bit is set, the RcvMrk field records the highest level of the receive FIFO.

Res **Reserved** **[30:29], [14:13]**
These bits are reserved for future use. They must be written as zeros.

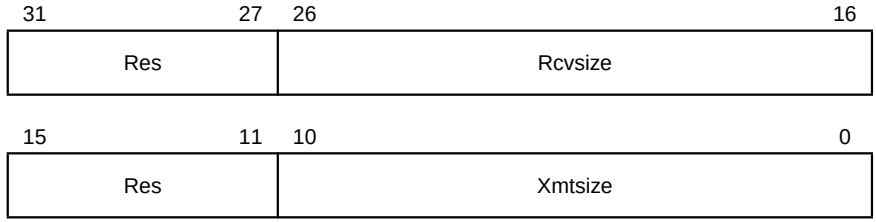
RcvMrk **Receive Watermark Level** **[28:16]**
Default: 0 **RO**
This field contains the highest number of bytes in the DBUF Receive FIFO since RcvMEn was set.

XmtMEn **DBUF Transmit FIFO Watermark Enable** **15**
Default: 0 **RW**
When this bit is cleared, the XmitMrk field is reset to 0. When this bit is set, the XmitMrk field records the highest level of the transmit FIFO.

XmitMrk **Transmit Watermark Level** **[12:0]**
Default: 0 **RO**
This field indicates the highest number of bytes in the DBUF Transmit FIFO since XmtMEn was set.

4.3.39 DBUF FIFOs Size Register 0 (0x0D0)

This register contains the size in number of quadlets of the DBUF FIFOs.



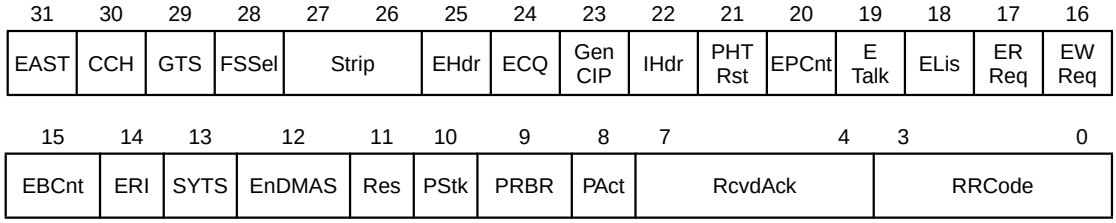
Res **Reserved** **[31:27], [15:11]**
These bits are reserved for future use. They must be written as zeros.

Rcvsize **Receive DBUF Size** **[26:16]**
Default: Based on the size inputs to the core at reset
RW
This field contains the size in number of quadlets of the DBUF Receive FIFO.

Xmitsize **Transmit DBUF Level** **[10:0]**
Default: Based on the size inputs to the core at reset
RW
This field indicates the size in number of quadlets of the DBUF Transmit FIFO.

4.3.40 PHT Control and Status Register 1 (0x100)

The settings within the PHT Control register enable/disable various functions of the 1394 Packet Header Transformation module. All bits in this register can be read and written from the application interface. Program all PHT configuration bits in other registers before enabling one of (EAST, ERReq, EWReq, EnDMAS) OR (ELis and/or ETalk).



EAST **Enable Asynchronous Stream Transmit** **31**
Default: 0 **RW**
When this bit is set, 1394 asynchronous stream packets (isochronous transmit format 1 packets sent during the asynchronous interval) are generated while EPCnt is set and the PBCnt field of the Data Transfer Control Register is not zero, and the threshold condition of the DFill field of the Data Transfer Control Register has been met. When this bit is set, 1394 asynchronous stream packets are generated while EPCnt is not set and the threshold condition of the DFill field of the Data Transfer Control Register has been met. EWReq, ERReq, EnDMAS, and ETalk must not be set when this bit is set. This bit is only sampled between transfers. If the PRBR bit becomes set while this bit and the PAct bit are set, the PAct bit is cleared and the PStk bit is set because the channel number must be reallocated.

CCH **Check CIP Header** **30**
Default: 0 **RW**
When this bit is set, received isochronous packets are checked to determine if the EOH and Form bits of the CIP header conform to the two-quadlet CIP Header format. If the packet does not conform, the CIPHE interrupt is set.

GTS **Generate Time Stamp** **29**
Default: 0 **RW**
When this bit, GenCIP, and ETalk are set, a time stamp is generated and included in the transmitted packet. The most-significant bit of the FMT field of the CIP Header Transmit 1 Register determines whether there is a SYT time stamp. The SPH bit of the CIP Header Transmit 0 Register determines if there is a source packet header time stamp.

FSSel **Frame Sync Select** **28**
Default: 0 **RW**
When this bit is set, the FSYNCIN input signal is used to control time stamping when GenCIP and the most-significant bit of the FMT field of the Stream Control Transmit 1 Register are 0s (for example, DVCR). The FSYNCIN input must be asserted during the first write strobe of a packet in which the core generates an SYT time stamp. If this bit is not set and SYT time stamping

is desired, the ECQ bit must be set to enable the alternative method (SYT time stamping occurs for each packet whose frame start bit is set in the application prepended embedded control quadlet).

Strip **Strip CIP Header** **[27:26]**
Default: 0 **RW**

This field controls how CIP headers are handled when listening to an isochronous channel.

Strip	Description
00	Unchanged.
01	Strip all CIP headers (also subtract 8 bytes from data packet length in received isochronous header sent to DBUF)
10	Strip only empty CIPs (if IHdr is set, when an empty CIP is encountered the header is not written to the DBUF)
11	Reserved.

EHdr **Embedded Header** **25**
Default: 0 **RW**

When this bit is set, the header of the packet (as shown in isochronous transmit format 3 or the read response for block data packet transmit format) must be prepended to the data field of a packet to be transmitted. The header registers are not used. For unformatted isochronous data fields (tag = 0 in header), multichannel talk capability is supported. When this bit is set and ETalk is set, ECQ also must be set. The embedded control quadlet is needed to determine the speed of transmission of the packet.

ECQ **Embedded Control Quadlet** **24**
Default: 0 **RW**

When this bit is set, a control quadlet must be prepended to the data field of an isochronous packet to be transmitted (the control quadlet enables software-controlled frame synchronization and hardware-assisted isochronous data flow control (useful for DV frame rate transmission control). When GenCIP is set in addition to this bit and the cycle skip field of the embedded control quadlet is N, N empty CIP packets are transmitted before the packet containing the data_field

after the embedded control quadlet. In addition, for unformatted data fields (tag = 0), multichannel talk capability is supported if this bit is set in addition to the EHdr bit.

GenCIP **Generate CIP Header** **23**
Default: 0 **RW**
When this bit is set, a CIP header is prepended to the data field of each isochronous packet transmitted. This bit must not be set when talking on more than one isochronous channel. If this bit is set, for each DMA request assertion, the DMA controller must write less than or equal to the number of source packets transmitted in a 1394 payload.

IHdr **Include Header** **22**
Default: 0 **RW**
When this bit is set, the header is written to the DBUF FIFO in addition to the data field of the 1394 packet (useful when listening to more than one isochronous channel or DMA space requests enabled).

PHTRst **PHT Reset** **21**
Default: 0 **RW**
If the PHT becomes stuck, this bit must be set. This bit automatically clears itself.

EPCnt **Enable Packet Counter** **20**
Default: 0 **RW**
When this bit is set, PHT packet transmissions stop when the PBCnt field of the Data Transfer Control Register is decremented to zero. If neither this bit nor EBCnt is set and either ERReq, EWReq, ETalk, or EAST is set, 1394 packets are transmitted indefinitely. EBCnt and EPCnt must not be set at the same time.

ETalk **Enable Isochronous Talk** **19**
Default: 0 **RW**
When this bit is set, isochronous packets are transmitted. EAST, ERReq, EnDMAS, and EWReq must not be set when this bit is set. When generating CIPs, set ETalk before setting DEn (DMA Enable) in the DMA Status and Control Register. This bit is only sampled between transfers.

ELis **Enable Isochronous Listen** **18**
Default: 0 **RW**
When this bit is set, isochronous packets transmitted on channels enabled in the Stream Receive Channel Selection Registers are received. ERReq and EWReq must not be set when this bit is set. This bit is only sampled between transfers.

ERReq **Enable Read Requests** **17**
Default: 0 **RW**
When this bit is set, 1394 Read Requests are generated while EPCnt or EBCnt is set and the PBCnt field of the Data Transfer Control Register is not zero. If this bit is set, 1394 Read Requests are generated indefinitely while neither EPCnt nor EBCnt is set and the DBUF Receive FIFO has enough space to accept another response. EAST, EWReq, ETalk, EnDMAS, and ELis must not be set when this bit is set. This bit is only sampled between transfers. If the PRBR bit becomes set while this bit and the PAct bit are set, the PAct bit is cleared and the PStk bit is set because the node IDs might have changed.

EWReq **Enable Write Requests** **16**
Default: 0 **RW**
When this bit is set, 1394 Write Requests are generated while EPCnt or EBCnt is set and the PBCnt field of the Data Transfer Control Register is not zero. If this bit is set, 1394 Write Requests are generated indefinitely while neither EPCnt nor EBCnt is set. EAST, ERReq, ETalk, EnDMAS, and ELis must not be set when this bit is set. This bit is only sampled between transfers. If the PRBR bit becomes set while this bit and the PAct bit are set, the PAct bit is cleared and the PStk bit is set because the node IDs might have changed.

EBCnt **Enable Byte Count** **15**
Default: 0 **RW**
If this bit is set, asynchronous PHT packet transmissions will stop when the PBCnt field of the data transfer control register is decremented to zero. The data length field of the transmitted 1394 header of the last packet transferred is adjusted appropriately if the PBCnt field is not divisible by the programmed data length. If neither this bit nor EPCnt is set and either ERReq or EWReq is set, 1394 packets are transmitted indefinitely. Both EBCnt and

EPCnt must not be set. This bit must not be set when ETalk or EAST is set.

ERI **Enable Request Interval** **14**
Default: 0 **RW**
If this bit and EWReq or ERReq are set, the value programmed in the RqIntrvl field determines the time interval between consecutive PHT requests as measured from the reception of an ACK or response packet of ‘complete’ or ‘busy’ for the first packet.

SYTS **SYT Start** **13**
Default: 0 **RW**
If this bit is set, incoming CIP format isochronous packets are not written to the DBUF until a packet containing a SYT time stamp is received (indicating the start of a frame).

EnDMAS **Enable DMA Space** **12**
Default: 0 **RW**
When this bit is set, automatic handling of received read and write requests is enabled. A received request packet whose quadlet aligned address matches the DMAR field criteria is handled automatically by the PHT. EAST, ERReq, ETalk, EWReq, and ELis must not be set when this bit is set. This bit is only sampled between transfers. If the PRBR bit becomes set while this bit and the PAct bit are set, the PAct bit is cleared and the PStk bit is set because the node IDs might have changed.

Automatic handling of read and write requests occurs when the EnDMAS bit is set, an incoming request packet's quadlet aligned address matches the DMAR field criteria, and the incoming request packet's source node ID corresponds to a selected node for the PHT (the corresponding bit in the Stream Receive Channel/NodeID Selection registers is set).

The EHdr and IHdr bits in PHT Control and Status Register 0 affect the format of the data written to the DBuf Transmit and Receive FIFOs when EnDMAS is asserted. If IHdr is not set, only the data field portion of the incoming request is written to the DBuf Receive FIFO. If IHdr is set and EHdr is not set, the 1394 header is written to the DBuf Receive FIFO in addition to the data field. If IHdr and EHdr are set, in addition to the

1394 header and data field, a quadlet containing the speed of the received request is written to the DBuf Receive FIFO.

Note: If EHdr is set, special care must be taken to guarantee that all response packets have been sent prior to disabling EnDMAS. One possibility would be to first clear the nodeID Selection registers to disable further acceptance of incoming requests by the PHT. When it is determined that all previously accepted requests have been processed, EnDMAS can be safely deasserted.

Note: If IHdr and EHdr are set, the format of the data in the DBuf is as shown in Figure 2.13 (Write Request for Data Block Receive Format). The maximum data field size that can be received is the DBuf Receive FIFO size less 20 bytes.

If EHdr is not set, the DBuf Transmit FIFO must contain only the data field for read response packets to be sent in response to received read requests. If EHdr is set, the format followed is that in Figure 2.4 or Figure 2.7.

An appropriate setting for EHdr and IHdr depends on the capability of the external DMA controller.

Res **Reserved** **11**
This bit is reserved and must be written as zero.

PStk **PHT Module Stuck** **10**
Default: 0 **RO**
When this bit is set, software must set the PHTRst bit to reinitialize the module.

PRBR **PHT Received Bus Reset** **9**
Default: 0 **RO**
This bit is set when the PAct bit was set and a Bus Reset was detected.

PAct **PHT Active** **8**
Default: 0 **RO**
This bit is automatically set when the PHT module is not in the idle or stuck state with the exception that if both ETalk and ELis are set and the PHT transmitter becomes stuck, the PAct bit remains set because the PHT receiver never becomes stuck.

RcvdAck **Received Acknowledge** **[7:4]**
Default: 0xF **RO**
This field contains the ack code received for the last packet transmitted. It is initialized to 0xF when a request for transmission is made.

RRCode **Received Response Code** **[3:0]**
Default: 0xF **RO**
This field contains the rcode in the last response packet received because of a request packet transmitted. It is initialized to 0xF when a request for transmission is made.

4.3.41 PHT Split Time-Out/Empty CIP Interval Register 1 (0x104)

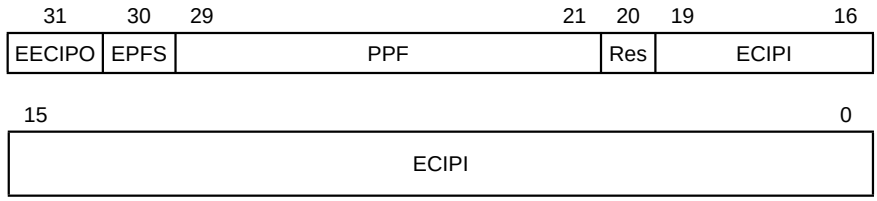
When ERReq or EWReq is set, this register contains the maximum time allowed from reception of an ack pending in reply to a request initiated by the PHT block until a response to that request is received. Application software is responsible for split time-out detection for packets transmitted using the UTF FIFO. This detection can be done using the CycSt and AckRcvd interrupts.

DST **Disable Split Time-Out** **31**
When this bit is set, the time limit field is ignored and there is no hardware time-out. It can be useful to set this bit when it is known the destination node is not on the local 1394 bus. In such a bridged environment the remote split time-out can be known to be longer than the maximum value that the TimeLimit field can be programmed to and time-out detection must be handled by other means.

Res **Reserved** **[30:16]**
These bits are reserved for future use. They must be written as zeros.

TimeLimit **Time Limit** **[15:0]**
Default: 0x0000 **RW**
Bits [16:18] correspond to seconds. Bits [19:31] correspond to fractional seconds with a 125 µs resolution (equivalent resolution to the cycle count field of the Cycle Timer register).

When ETalk is set, this register is defined as follows.



EECIPO **Enable Empty CIP Override** **31**
Default: 0 **RW**
When this bit, GenCIP, and ETalk, etc. are set, an empty CIP packet is sent during the isochronous cycle subsequent to each time an internal counter increments to the same value as the ECIPI field.

EPFS **Enable Programmable Frame Synchronization** **30**
Default: 0 **RW**
When this bit, GenCIP, ETalk, etc. are set, an SYT time stamp is generated for the first nonempty CIP packet transmitted and every N + 1 nonempty CIP packets transmitted thereafter, where N is the value of the PPF field.

PPF **Packets Per Frame** **[29:21]**
Default: 0 **RW**
This field is only used when the EPFS bit is set and must be programmed with the number of packets per frame minus one (for example, for a 525-60 system, the value is 249 decimal).

Res **Reserved** **20**
This bit is reserved for future use. They must be written as zeros.

ECIPI **Empty CIP Interval** **[19:0]**
Default: 0 **RW**
This field is compared against an internal counter that increments every other SClk (49.152 MHz clock) from the PHY. The counter is reset to zero when a match occurs.

4.3.42 PHT Request/Response/CIP Receive Header 0 Register 1 (0x108)

This register contains information used in the transfer of:

- a request packet if ERReq or EWReq is set
- a response packet if EnDMAS is set, or
- a received isochronous packet if ELis is set.

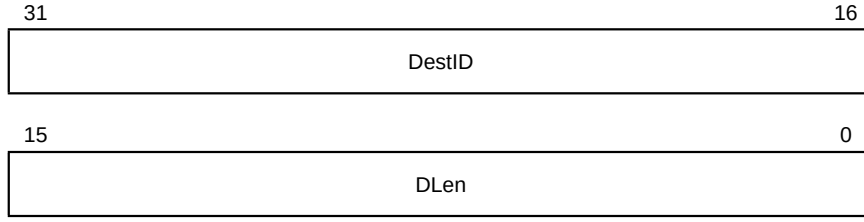
This information is used for transfers to/from the DBUF only.

If ERReq or EWReq is set, the register is defined as follows.

DID **Destination ID** **[31:16]**
Default: 0 **RW**
This field is the concatenation of the Bus ID and the Node ID of the destination. The broadcast node ID of 0b111111 is not supported.

DOffHi **Destination Offset Hi** **[15:0]**
Default: 0 **RW**
This field contains the most-significant 16 bits of the destination offset address. This field must not be programmed with a value that would result in an overflow when this field is automatically updated during multipacket transmission.

If EnDMAS is set, the definition is as follows (note that if EHdr is not set, the contents of this register do not always reflect the last request received and must not be relied on for that purpose).



DestID **Destination ID** **[31:16]**
Default: 0x0000 **RW**
This field is the concatenation of the Bus ID and the Node ID of the node that initiated the DMA space request.

DLen **Data Length** **[15:0]**
Default: 0 **RW**
This field contains the data length in bytes of the received DMA Space request, which is reused if a read response is generated.

When ELis is set, this register is defined as follows.

EOH0 **End Of CIP Header 0** **31**
Default: 0 **RO**
A value of 0 means another header quadlet will follow (always zero for a standard two-quadlet CIP).

Form0 **Form 0** **30**
Default: 0 **RO**
This field is always zero for a standard two-quadlet CIP.

SID **Source ID** **[29:24]**
Default: 0 **RO**
This field contains the node ID of transmitter.

DBS	**Data Block Size** **Default: 0** This field contains the data block size in quadlets.	**[23:16]** **RO**
FN	**Fraction Number** **Default: 0** This field contains the number of data blocks into which a source packet is divided.	**[15:14]** **RO**
QPC	**Quadlet Padding Count** **Default: 0** This field contains the number of dummy quadlets padded to a source packet to equalize the size of divided data blocks.	**[13:11]** **RO**
SPH	**Source Packet Header** **Default: 0** A value of one on this bit indicates that the source packet has its own header.	**10** **RO**
Res	**Reserved** These bits are reserved for future use.	**[9:8]**
DBC	**Data Block Counter** **Default: 0** This field contains the continuity counter of data blocks to detect a loss of data blocks. The counter value for the first data block in a bus packet is shown in this field.	**[7:0]** **RO**

4.3.43 PHT Request/Response/CIP Receive Header 1 Register 1 (0x10C)

This register contains information used in the transfer of:

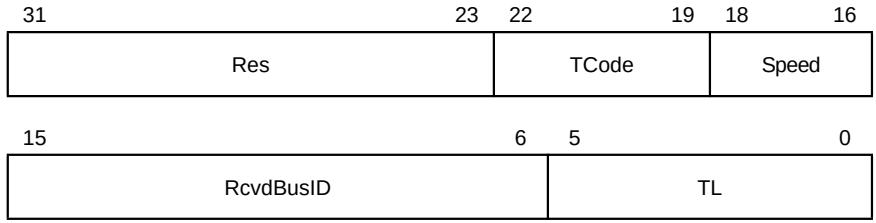
- a request packet (if ERReq or EWReq is set),
- a response packet (if EnDMAS is set), or
- a received isochronous packet (if ELis is set).

This information is used for transfers to/from the DBUF only.

When ERReq or EWReq is set, this register is defined as follows.

DOffLo **Destination Offset Lo** **[31:0]**
Default: 0 **RW**
This field contains the least-significant 32 bits of the destination offset address.

If EnDMAS is set, this register is defined as follows (note that if EHdr is not set, the contents of this register do not always reflect the last request received and must not be relied on for that purpose).



Res **Reserved** **[31:23]**
These bits are reserved for future use. They must be written as zeros.

TCode **Transaction Code** **[22:19]**
Default: 0 **RW**
This field contains the transaction code of the DMA space read or write request.

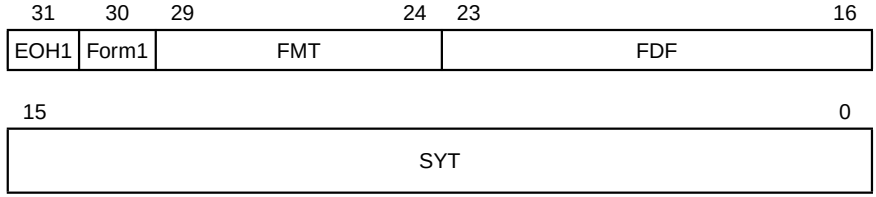
Speed **Speed** **[18:16]**
Default: 0 **RW**
This field contains the transmit speed of the DMA space read or write request.

Speed[13:15]	Encoding
000	S100 Speed
010	S200 Speed
100	S400 Speed

RcvdBusID **Received Bus ID** **[15:6]**
Default: 0 **RW**
This field contains the Destination Bus ID used in a DMA space read request. This value is used as the Source Bus ID in the read response.

TL **Transaction Label** **[5:0]**
Default: 0x0000 **RW**
This field contains the transaction label of the received DMA space read or write request.

When ELis is set, this register is defined as follows.



EOH1 **End Of CIP Header 1** **31**
Default: 0 **RO**
A value of one on this bit means another header quadlet does not follow (must always be one for standard 2-quadlet CIP).

Form1 **Form 1** **30**
Default: 0 **RO**
This bit must always be zero for a standard 2-quadlet CIP.

FMT **Format ID** **[29:24]**
Default: 0 **RO**
The encoding for this field is shown in the table below.

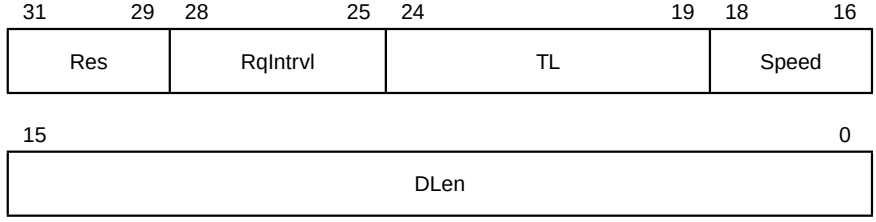
FMT	Description
0x00	When the MSB (bit 2) is zero, a time stamp might be in the SYT field (such as in the DVCR format).
0x20	When the MSB (bit 2) is one, a time stamp is not in the SYT field (such as in the MPEG format).

FDF **Format Dependent Field** **[23:16]**
Default: 0 **RO**
This field is defined for each format.

SYT **Synchronization Time Field** **[15:0]**
Default: 0 **RO**
For DVCR, this field can contain a time stamp of the frame synchronization pulse. For MPEG2, this field is zero.

4.3.44 PHT Request Header 2/Stream SPH Receive Register 1 (0x110)

If ERReq or EWReq is set, this register contains information used in the transfer of a request packet. This information is used for transfers to/from the DBUF only.



Res **Reserved** **[31:29]**
These bits are reserved for future use.

RqIntrvl **Request Interval** **[28:25]**
This four-bit field contains the minimum number of desired nominal 125 µs cycles between reception of an ack or response of ‘complete’ or ‘busy’ for a previous PHT generated request and transmission of the next PHT generated request. The actual observed time between any two requests can be as much as 125 µs less than the programmed value (to guarantee at least 125 µs interval, the programmed value must be two).

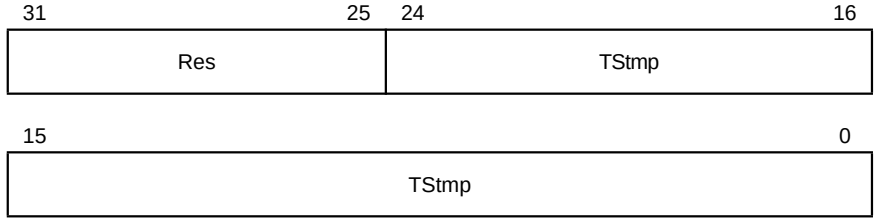
TL **Transaction Label** **[24:19]**
Default: 0x0 **RW**
These six bits indicate the transaction label for the packet. They are used for tracking requests with responses. PHT module 0 must be programmed with 0x3F. PHT module 1 must be programmed with 0x3E.

Speed **Speed** **[18:16]**
Default: 0 **RW**
This field contains the transmit speed of the DBUF read or write request as shown in the following table.

Speed[13:15]	Encoding
000	S100 Speed
010	S200 Speed
100	S400 Speed

DLen **Data Length** **[15:0]**
Default: 0 **RW**
This field contains the data length in bytes of the payload to be written or read. After an ack or rcode of ‘complete’ is received in response to a transmitted request, the destination offset field of the PHT Request Header Register is updated with the addition of the Data Length field.

When ELis is set, the register is defined as follows. This register contains the time stamp from the last received packet with an isochronous header tag field of 0b01 and a CIP header SPH bit of 0b1.

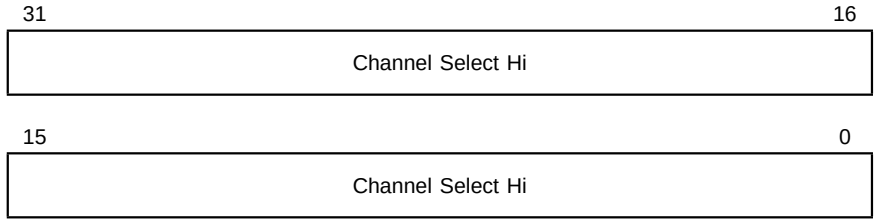


Res **Reserved** **[31:25]**
These bits are reserved for future use. They must be written as zeros.

TStmp **Time Stamp** **[24:0]**
Default: 0 **RW**
This field contains the time stamp of the packet.

4.3.45 Stream Receive Channel/NodeID Selection 0 Register 1 (0x114)

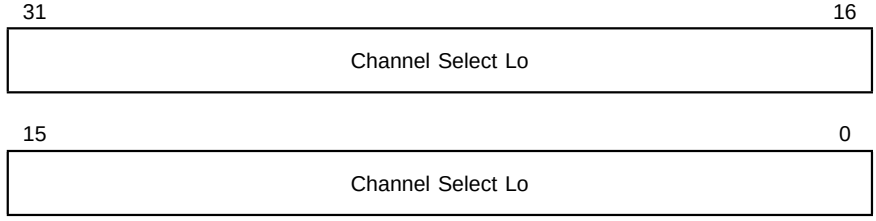
The settings within the Stream Receive Channel Selection registers enable/disable isochronous channel reception when ELis is set or enable/disable DMA Space read/write requests from other nodes when EnDMAS is set.



ChSelHi **Channel Select Hi** **[31:0]**
Default: 0 **RW**
This field contains the reception enable bits for stream channels 63 (bit 31) to 32 (bit 0) or the read/write request reception enable bits for broadcasts (bit 31) or nodes 62 (bit 30) to 32 (bit 0).

4.3.46 Stream Receive Channel/NodeID Selection 1 Register 1 (0x118)

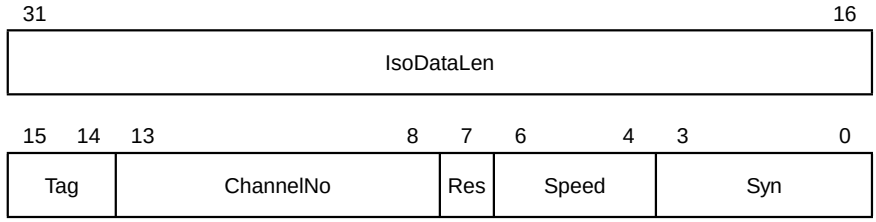
The settings within the Stream Receive Channel Selection registers enable/disable isochronous channel reception when ELis is set or enable/disable DMA Space read/write requests from other nodes when EnDMAS is set.



ChSelLo **Channel Select Lo** **[31:0]**
Default: 0 **RW**
This field contains the reception enable bits for stream channels 31 (bit 31) to 0 (bit 0) or the read/write request reception enable bits for nodes 31 (bit 31) to 0 (bit 0).

4.3.47 Stream Receive Channel Header Register 1 (0x11C)

This register contains the received information in the header quadlet of a stream packet (tcode equals 0xA Isoch or Async).



IsoDataLen **Iso Packet Data Length** **[31:16]**
Default: 0 **RO**
This field indicates the number of data bytes in the current packet. Transmission of zero data length packets is not supported.

Tag **Tag** **[15:14]**
Default: 0 **RO**
This field contains the tag of the stream packet.

ChannelNo **Channel Number** **[13:8]**
Default: 0 **RO**
This field contains the channel number of the stream packet.

Res **Reserved** **7**
This bit is reserved for future use. It must be written as zero.

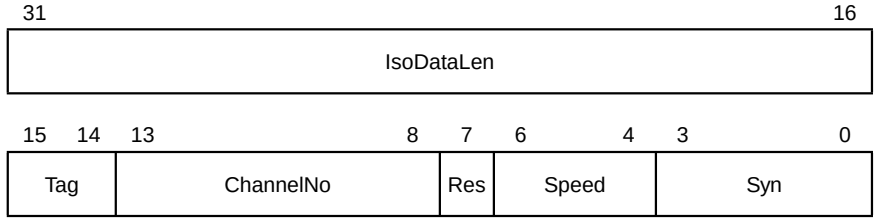
Speed **Speed** **[6:4]**
Default: 0 **RO**
This field contains the speed at which the current packet was received.

Speed[25:27]	Encoding
000	S100 Speed
010	S200 Speed
100	S400 Speed

Syn **Sync** **[3:0]**
Default: 0 **RO**
This field is the sync field of the stream packet.

4.3.48 Stream Transmit Channel Header Register 1 (0x120)

This register contains the information in the header quadlet of a stream packet (tcode equals 0xA Isoch or Async). This information is used for stream transmits only if the Embedded Header bit is not enabled (otherwise the header information is obtained from the DBUF). This register is used for single stream transmit mode in which the DBUF contains isochronous transmit format 1 or 2 information to be transmitted. The speed field generates speed bits when sending an Lreq on the PHY/link interface.



IsoDataLen **Iso Packet Data Length** **[31:16]**
Default: 0 **RW**
This field indicates the number of data bytes in the current packet.

Tag **Tag** **[15:14]**
Default: 0 **RW**
This field contains the tag of the stream packet.

ChannelNo **Channel Number** **[13:8]**
Default: 0 **RW**
This field contains the channel number of the stream packet.

Res **Reserved** **7**
This bit is reserved for future use. It must be written as zero.

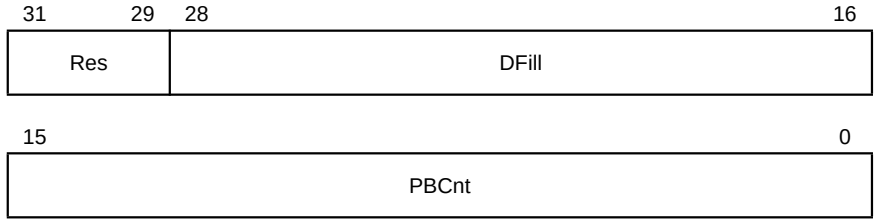
Speed **Speed** **[6:4]**
Default: 0 **RW**
This field contains the speed at which the current Isochronous packet has to be transmitted.

Speed[25:27]	Encoding
000	S100 Speed
010	S200 Speed
100	S400 Speed

Syn **Sync** **[3:0]**
Default: 0 **RW**
This field is the sync field of the stream packet.

4.3.49 Data Transfer Control Register 1 (0x124)

This register contains information used in the transfer of packets. This information is used for outbound 1394 DBUF transfers only.



Res **Reserved** **[31:29]**
These bits are reserved for future use. They must be written as zeros.

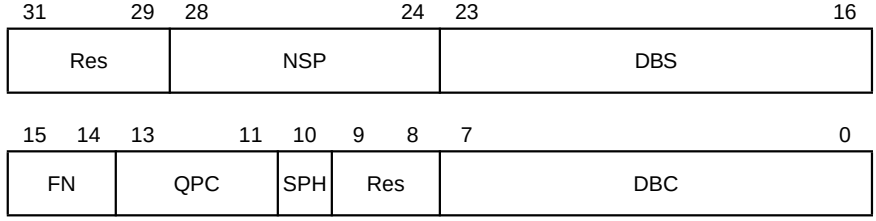
DFill **DBUF Threshold Fill Trigger** **[28:16]**
Default: 0 **RW**
The number of bytes in the DBUF transmit FIFO must be greater than this value before a stream request is made. When in isochronous talk mode, if the DBUF transmit FIFO level is not greater than this value, an empty CIP packet is sent if GenCIP is set. No isochronous packet is sent in isochronous talk mode if GenCIP is not set and the DBUF transmit FIFO level is not greater than this value. This field is useful when the stream packet size or sum of simultaneous isochronous streams is greater than the FIFO size and it is known that the sustained DMA data rate is sufficient to prevent underflow once

transmission starts. When the ECQ bit is set, DFill must be programmed no higher than the total number of bytes to be read from the FIFO per isochronous phase less five bytes.

PBCnt **Packet/Byte Count** **[15:0]**
Default: 0 **RW**
This field can be written with the desired packet/byte transmit count before enabling PHT transfers. This field is decremented after each packet transmission (when EPCnt or EBCnt is set). When the field reaches zero, the PBCntR interrupt is set and transfers stop.

4.3.50 CIP Header Transmit 0 Register 1 (0x128)

This register contains information used in the isochronous transfer of an IEC-61883 format packet. This information is used for transfers from the DBUF only. This register must not be written while ETalk is one. This register is not affected by PHTRst.



Res **Reserved** **[31:29], [9:8]**
These bits are reserved and read as zeros.

NSP **Number of Source Packets** **[28:24]**
Default: 1 **RW**
This field specifies the number of source packets transmitted when a nonempty CIP packet is sent (used to control DBC field incrementing). This value must be consistent with the IsoDataLen field of the stream transmit control header register (for example, if five 188-byte MPEG + four-byte time stamp (core generated source packet header) packets are to be transmitted in one 1394 packet, then NSP should be five and IsoDataLen should be 5 x (4 + 188) + 8 = 968. NSP must be one or more.

DBS **Data Block Size** **[23:16]**
Default: 0 **RW**
This field contains the data block size in quadlets.

FN **Fraction Number** **[15:14]**
Default: 0 **RW**
This field contains the number of data blocks into which a source packet is divided.

FN	Description
0b00	Not Divided (for example, DVCR). The DBC LSB increments after every nonempty CIP packet transmission.
0b11	1/8 source packet (for example, MPEG). The three LSBs of DBC are always 0b000 because an integer multiple of source packets is transmitted or an empty CIP per cycle occurred.

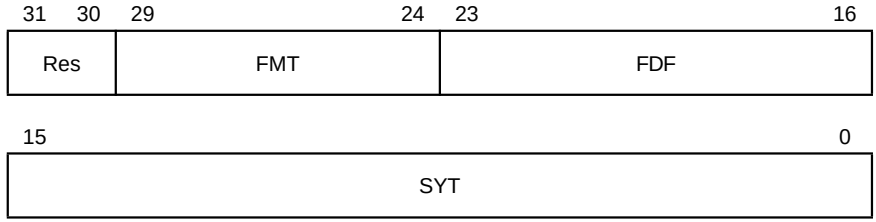
QPC **Quadlet Padding Count** **[13:11]**
Default: 0 **RW**
This field contains the number of dummy quadlets padded to a source packet by the application to equalize the size of divided data blocks.

SPH **Source Packet Header** **10**
Default: 0 **RW**
A value of one on this bit indicates that the source packet has its own header.

DBC **Data Block Counter** **[7:0]**
Default: 0 **RO**
This field contains the continuity counter of data blocks to detect a loss of data blocks. The counter value for the first data block in a 1394 packet is shown in this field.

4.3.51 CIP Header Transmit 1 Register 1 (0x12C)

This register contains information used in the transfer of a IEC-61883 format packet. This information is used for transfers to the DBUF only. This register must not be written while ETalk is one. This register is not affected by PHTRst.



Res **Reserved** **[31:30]**
These bits are reserved and read as zeros.

FMT **Format ID** **[29:24]**
Default: 0 **RW**
The appropriate value for this field is application dependent.

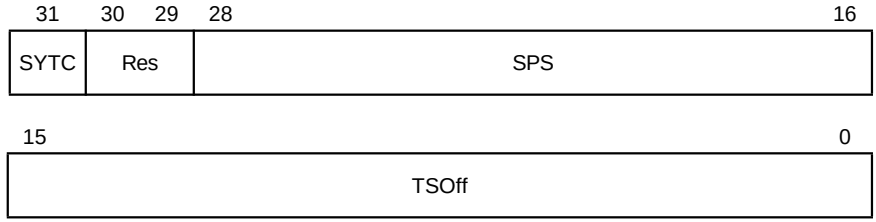
FMT	Description
0x00 (DVCR)	If the most-significant bit (bit 2) is a 0, it indicates that the PHT module can generate a time stamp in the transmitted SYT field (such as DVCR format).
0x20 (MPEG)	If the most-significant bit (bit 2) is a 1, the bits in the SYT field are always transmitted unmodified (used for MPEG format).

FDF **Format Dependent Field** **[23:16]**
Default: 0 **RW**
This field is defined for each format.

SYT **Synchronization Time Field** **[15:0]**
Default: 0 **RW**
For DVCR format, the transmitted version of this field is sometimes replaced with a PHT-module-generated time stamp of the frame synchronization pulse. For MPEG2 format, this field is 0x0000.

4.3.52 Stream Transmit Time Stamp Offset Register 1 (0x134)

This register contains the time stamp offset used for transmitting a packet with an isochronous header tag field of 0b01, if the GTS bit is set. This register is not affected by PHTRst.



SYTC **SYT Carry** **31**

When this bit is set, there is a carry between the cycle offset and cycle count field when generating SYT time stamps (see the definition of the TSOff field below).

Res **Reserved** **[30:29]**

These bits are reserved for future use. They must be written as zeros.

SPS **Source Packet Size** **[28:16]**

This field contains the size in bytes of a source packet less any source packet header. For example, in the case of MPEG2 transport stream packets, the appropriate value is 188 bytes. This field is only used when transmitting IEC-61883 format streams.

TSOff **Time Stamp Offset** **[15:0]**

Default: 0 **RW**

This field contains the time stamp offset added to the Cycle Time Register to produce the time stamp.

For SYT time stamps (when the SYTC bit of the PHT Control and Status register is zero), bits [11:0] are added to the cycle_offset field[1] and bits [15:12] are added to the four LSBs of the cycle_count field to produce the time stamp.

For SYT time stamps (when the SYTC bit of the PHT Control and Status register is one), bits [11:0] are added to the cycle_offset field[2] and bits [15:12] are added to the

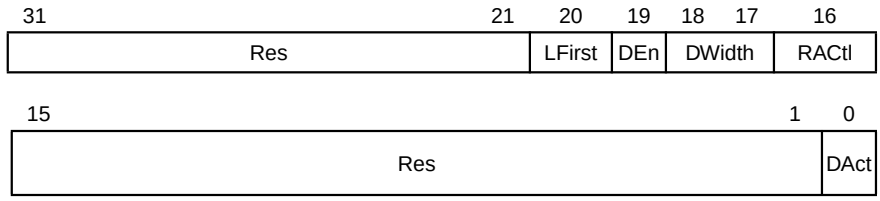
1. There is no carry to the next significant nibble of the time stamp, and the result is limited to 0xBFF (any addition that would result in a value greater than 0xBFF does not wrap to zero - the result is 0xBFF).
2. Carry to the next significant nibble of the time stamp (any addition that would result in a value greater than 0xBFF does wrap around).

four LSBs of the cycle_count field to produce the time stamp.

For SPH time stamps, bits [11:0] are added to the cycle_offset field of the cycle timer. If the result is greater than 3071, the resulting SPH cycle offset value transmitted is (cycle offset sum – 3072). A carry of one is made to the addition of bits [15:12] and the cycle count field of the cycle timer. If the sum of bits [15:12], the cycle count field of the cycle timer, and the carry (if any) is greater than 7999, then the resulting SPH cycle count value is (cycle count sum – 8000). The cycle offset portion (bits [11:0]) of TSOff must not be programmed with a value greater than 3071.

4.3.53 DMA Control and Status Register 1 (0x138)

The settings within the DMA Control Register enable/disable various functions of the 1394 slave DMA module. All bits in this register can be read and written from the application interface.



Res **Reserved** **[31:21], [15:1]**
These bits are reserved for future use. They must be written as zeros.

LFirst **Least Significant First** **20**
Default: 0 **RW**
Data is processed on a quadlet basis within the device. If this bit is not set, then on an eight-bit wide DMA interface, the first byte output/input from/to the device is the most-significant byte. If the least-significant byte is output/input first, this bit must be set. Similarly on a 16-bit interface, the first two bytes output/input from/to the device are the two most-significant bytes, if this bit is not set.

DEn **DMA Enable** **19**
Default: 0 **RW**
When this bit is set, the DMA interface is enabled.

DWidth **DMA Width** **[18:17]**
Default: 0b10 **RW**
The following table lists the encoding for this field.

DWidth	Description
00	8-bit interface
01	16-bit interface
10	32-bit interface
11	Reserved value

RACtl **Request Acknowledge Control** **16**
Default: 1 **RW**
When this bit is cleared, all DMA transfers use the DINT1_DREQP and DMAC1_DACKP signals. When this bit is set, all packets read from the DBUF use DINT1_DREQP and DMAC1_DACKP while packets written to the DBUF use DINT1_DREQWP and DMAC1_DACKWP.

DAct **DMA Active** **0**
Default: 0 **RO**
When this bit is set, the DMA interface is active (due to data to be read by external controller or data needed to be written from external controller).

4.3.54 DMA Transfer Threshold Register 1 (0x13C)

This register contains information used in the transfer of packets. This information is used to control DMA transfers.

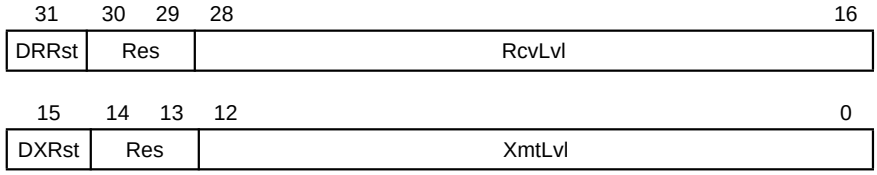
Res **Reserved** **[31:29], [15:13]**
These bits are reserved for future use. They must be written as zeros.

RdLvl **Read Level** **[28:16]**
Default: 0 **RW**
The number of bytes in the DBUF Receive FIFO must be greater than this value before the appropriate DINT_DREQP signal is asserted to notify the external DMA controller to read the FIFO.

WrLvl **Write Level** **[12:0]**
Default: 0 **RW**
The number of bytes empty in the DBUF Transmit FIFO must be greater than this value before the appropriate DINT_DREQP signal is asserted to notify the external DMA controller to write the FIFO.

4.3.55 DBUF FIFOs Level Register 1 (0x140)

This register contains the number of bytes in the DBUF FIFOs.



DRRst **DBUF Receive FIFO Reset** **31**
Default: 0 **RW**
When set, this bit clears the DBUF Receive FIFO pointers. This bit automatically clears itself.

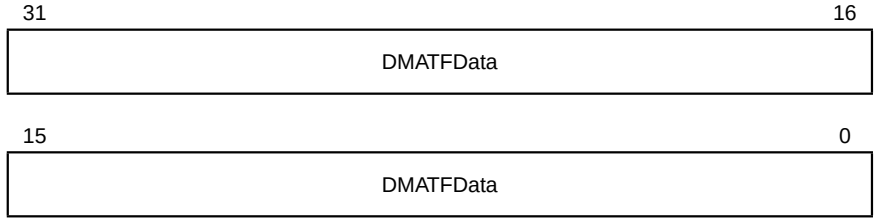
Res **Reserved** **[30:29], [14:13]**
These bits are reserved for future use. They must be written as zeros.

RcvLvl **Receive Level** **[28:16]**
Default: 0 **RO**
This field contains the number of bytes in the DBUF Receive FIFO.

DXRst	**DBUF Transmit FIFO Reset** **Default: 0** When set, this bit clears the DBUF Transmit FIFO pointers. This bit automatically clears itself.	**15** **RW**
XmitLvl	**Transmit Level** **Default: 0** This field indicates the number of bytes in the DBUF Transmit FIFO.	**[12:0]** **RO**

4.3.56 DBUF Tx Data Register 1 (0x144)

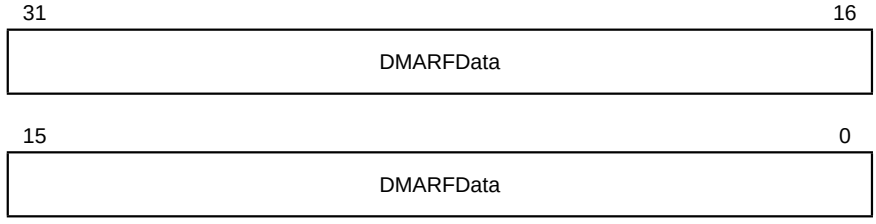
When the DMA interface is disabled, the software writes to this register in order to perform writes to the Tx DBUF FIFO. The software uses this register to directly place data in the Tx DBUF, which the PHT can then transmit.



DMATFData	**DMA Transmit FIFO Data** **Default: None** This field contains the data value to be written to the Tx DBUF FIFO.	**[31:0]** **WO**

4.3.57 DBUF Rx Data Register 1 (0x148)

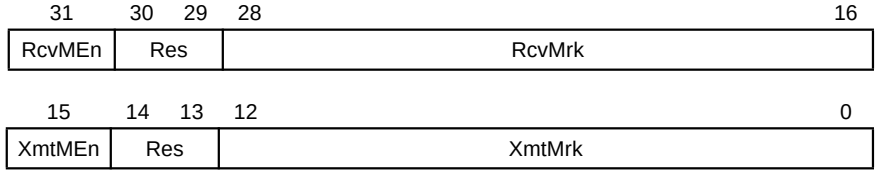
When the DMA interface is disabled, the software reads from this register in order to perform reads from the Rx DBUF FIFO. The software uses this register to directly read data that the PHT has received.



DMARFData **DMA Receive FIFO Data** **[31:0]**
Default: None **RO**
This field contains the data value that is read from the Rx DBUF FIFO.

4.3.58 DBUF FIFOs Watermark Level Register 1 (0x14C)

This register contains the highest number of bytes in the DBUF FIFOs.



RcvMEn **DBUF Receive FIFO Watermark Enable** **31**
Default: 0 **RW**
When this bit is cleared, the RcvMrk field is reset. When this bit is set, the RcvMrk field records the highest level of the Receive FIFO.

Res **Reserved** **[30:29], [14:13]**
These bits are reserved for future use. They must be written as zeros.

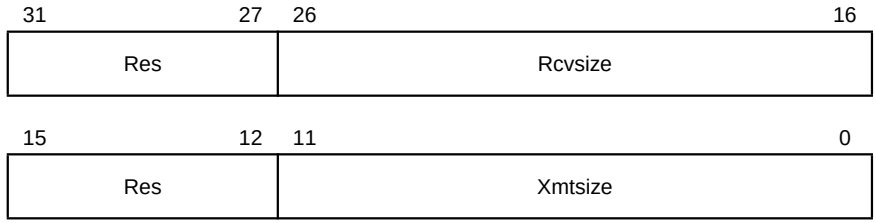
RcvMrk **Receive Watermark Level** **[28:16]**
Default: 0 **RO**
This field contains the highest number of bytes in the DBUF Receive FIFO since RcvMEn was set.

XmtMEn **DBUF Transmit FIFO Watermark Enable** **15**
Default: 0 **RW**
When this bit is cleared, the XmitMrk field is reset. When this bit is set, the XmitMrk field records the highest level of the Transmit FIFO.

XmitMrk **Transmit Watermark Level** **[12:0]**
Default: 0 **RO**
This field indicates the highest number of bytes in the DBUF Transmit FIFO since XmtMEn was set.

4.3.59 DBUF FIFOs Size Register 1 (0x150)

This register contains the size in number of quadlets of the DBUF FIFOs.



Res **Reserved** **[31:27], [15:12]**
These bits are reserved for future use. They must be written as zeros.

Rcvsize **Receive DBUF Size** **[26:16]**
Default: Based on the size inputs to the core at reset
RW
This field contains the size in number of quadlets of the DBUF Receive FIFO.

Xmitsize **Transmit DBUF Level** **[11:0]**
Default: Based on the size inputs to the core at reset
RW
This field indicates the size in number of quadlets of the DBUF Transmit FIFO.

Chapter 5
Operation Overview

This chapter discusses the operation of the blocks within the 1394 Node Controller core through written descriptions and waveforms. This chapter contains the following sections:

5.1 Link Layer Controller

The Link Layer Controller contains the transmitter and receiver blocks, the retry mechanism, and the PHY interface. This section describes the operation of these functions within the Link Layer Controller module.

5.1.1 Transmitter

The transmitter block starts arbitrating for the bus as soon as the packet is confirmed into the Transmit FIFO. Once the PHY grants the device controller, it will reformat the transmit packet as a 1394 data packet format and will start transmitting at the indicated speed. The transmitter also adds the header and data CRCs to the packet.

Once the packet is transmitted, the transmitter will wait for the acknowledge packet. If the acknowledge packet is missing, it waits until it receives the subaction gap indication from the PHY. Once the acknowledge packet is received, the Ack Code and the packet’s transaction label are stored into the Acknowledge Status Register. The device controller generates the AckRcvd interrupt.

The core implements the single phase retry protocol. The retry code in the transmit packet is Retry_X. The Transmit FIFO can hold the full transmit packet.

Depending on the acknowledge code received (for example, ACK_BUSY), if the receiver is busy, then the transmitter retransmits the packet until the retry limit has been reached. If the receiver is still issuing the busy acknowledge after the retry limit has been reached, the transmitter discards the packet and sets the RetEx interrupt.

NOTE: The application software architecture must ensure that the software request and response queues are managed to guarantee no deadlocks can occur (that is, the request and response queues must be processed independently of each other).

5.1.2 Receiver

The receiver block decodes the incoming packet for a valid tcode. The receiver block accepts all incoming packets if the destination ID matches with local node ID or if the incoming packet is a broadcast packet and automatically sends the corresponding acknowledgment.

The receiver block accepts all incoming packets if the tcode of the packet matches with one of the following tcodes (see Table 5.1).

Table 5.1 Tcodes

Type of Asynchronous Packet	tcode
Write Request for Quadlet Data	0x0000
Write Request for Block Data	0x0001
Read Request for Quadlet Data	0x0100
Read Request for Block Data	0x0101
Write Response	0x0010
Read Response for Quadlet Data	0x0110
Read Response for Block Data	0x0111

Table 5.1 Tcodes (Cont.)

Type of Asynchronous Packet	tcode
Cycle Start	0x1010
Lock Request	0x1001
Lock Response	0x1011

The receiver block receives all the packets, if the destination ID decode is successful in one of the following conditions:

- Destination ID field of the incoming packet matches the local node ID
- Destination Bus ID field of the incoming packet is 0x3FF and the physical ID matches the local physical ID
- Destination ID field of the incoming packet is 0xFFFF
- Destination physical ID field of the incoming packet is 0x3F and the destination bus ID matches the local bus ID

The receiver sets the TCErr Interrupt on the application interface if it decodes the Destination ID of the incoming packet and there is a tcode error detected (the incoming packet tcode does not match the ones listed above). In this case, the receiver just ignores the incoming packet.

The receiver also checks for mismatches in the incoming packet CRC and the calculated CRC from the CRC block. If the Header CRC mismatch occurs for the incoming packet, the receiver sets the HdrErr interrupt on the application interface. In this case, the receiver just ignores the incoming packet. The receiver block will not send any Ack for these packets. In the packets for which a Data CRC mismatch occurs, the receiver discards the packet. It also sends the Ack_Data_Error type acknowledgment in response to the discarded packet.

Once the tcode is valid and the destination ID decoding is successful, the receiver formats the incoming packets into the application data packet formats and writes into the Asynchronous receive FIFOs.

The receiver will not confirm the packet into the FIFO if the packet header is not received and decoded properly. Once the packet is decoded properly with a header CRC match, then the packet is confirmed after the data payload CRC is verified (note that the

application will not see the packet received until the receiver confirms anything into the FIFO). At the end of each packet reception, the receiver block will append a status quadlet to the packets indicating the speed at which the packet is received. The application needs this information to generate the response packets for incoming requests.

Once the incoming packet is confirmed into the FIFO, the corresponding interrupt is set (URx). The application can recognize this interrupt and can read the packet from the corresponding FIFO. The length of the packet is determined by reading the UBUF receive write level.

5.1.3 PHY Interface

The 1394 Node Controller core communicates with the PHY through the PHY interface. This interface handles the PHY arbitration and PHY data services. The PHY interface communicates the arbitration request to gain control of the bus to the PHY. The request contains the type of arbitration to perform. The PHY communicates back the arbitration result, which tells whether the arbitration was won or lost for the request.

The PHY interface also handles the data communication between the PHY and the transmitter and receiver. The PHY interface controls when data is presented to the receiver, and it also controls when the transmitter may present data to the PHY.

Four basic operations may occur on the PHY interface when communicating with the PHY: Request, Status, Transmit, or Receive. The PHY initiates all requests. The PHY interface module uses the request operation to read or write the PHY register space or to ask the PHY to initiate a transmit action by arbitrating for the bus. The PHY initiates a receive action whenever a packet is received from the serial bus.

The PHY interface module has two major functional blocks:

- Request Controller

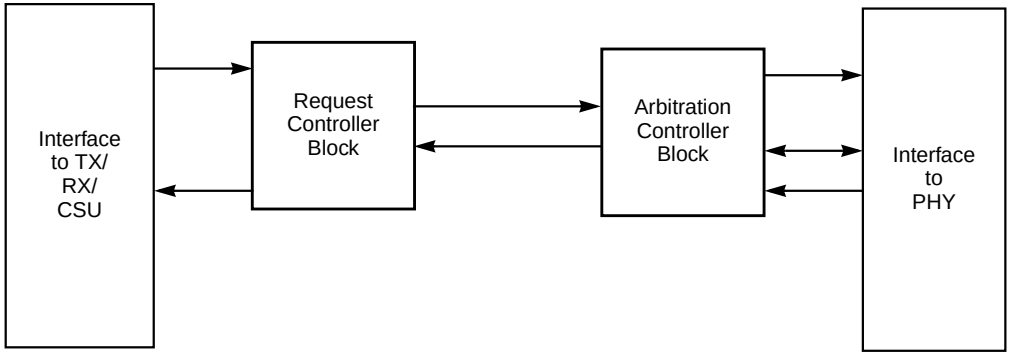
  This block interfaces with the transmitter, receiver, and the CSU.

- Arbitration Controller

  The Arbitration controller has the PHY interface on one side. This interface complies with the IEEE P1394a Draft 2.0 Specification.

Figure 5.1 shows the communication between these blocks.

Figure 5.1 Block Diagram of the PHY Interface Block

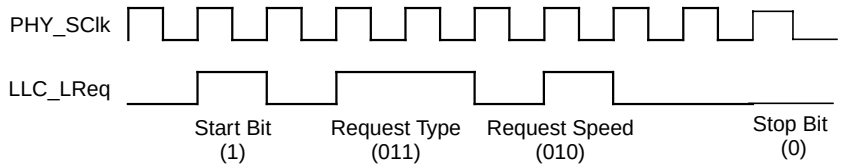


5.1.3.1 Request

To request the bus or to access a PHY register, the PHY interface module sends a short stream to the PHY on the LLC_LReq pin. The information sent includes the type of request, the speed at which the packet is to be sent, or a read or write command. The transfer size is variable, depending on whether it is a bus request, a read access, or a write access, respectively. A stop bit of ‘0’ is required after each type of request transfer before another transfer may begin.

Figure 5.2 shows the bus request for an asynchronous packet (uses fair request), which is transmitted at 200 Mbits/s. The request formats and handling for different requests are given in the IEEE P1394a Draft 2.0 Specification.

Figure 5.2 Bus Request using LLC_LReq



5.1.3.2 Status

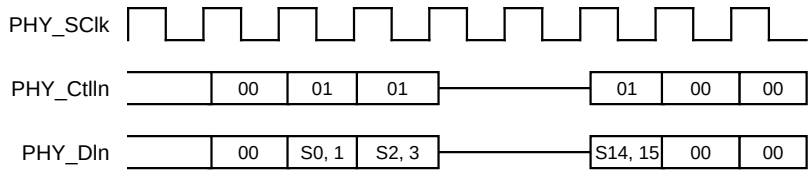
The PHY initiates this transfer whenever it has some status information and when it finds the interface idle. To initiate the transfer, the PHY places the status code (0b01) on the PHY_CtlIn pins and the first two bits of status information on PHY_DIn[0:1]. The PHY_CtlIn pins maintain

this value throughout the status transfer. The status transfer is interrupted in the middle if the PHY receives a packet from another node.

The PHY normally sends the first four status bits to the PHY interface. These bits are status flags that are needed by the 1394 state machines. The PHY sends an entire status packet of 16 bits to the PHY interface after a request transfer that contains the read request, or whenever the PHY has pertinent information to send. The only defined condition where the PHY automatically sends a register to the link is after self-identification, where it sends the Physical_ID register that contains the new node address.

Figure 5.3 shows the status timing waveforms. The status bit descriptions are described following the figure.

Figure 5.3 Status Timing



Data **[15:8]**

These bits contain the PHY register data.

Address **[7:4]**

These bits contain the PHY register address.

State Time-out **3**

A one on this bit indicates a loop was detected in the cable topology.

Bus Reset **2**

This bit indicates whether or not the PHY has entered the reset state.

Subaction Gap **1**

This bit is set when the PHY detects the Subaction gap time.

Arbitration Reset Gap **0**

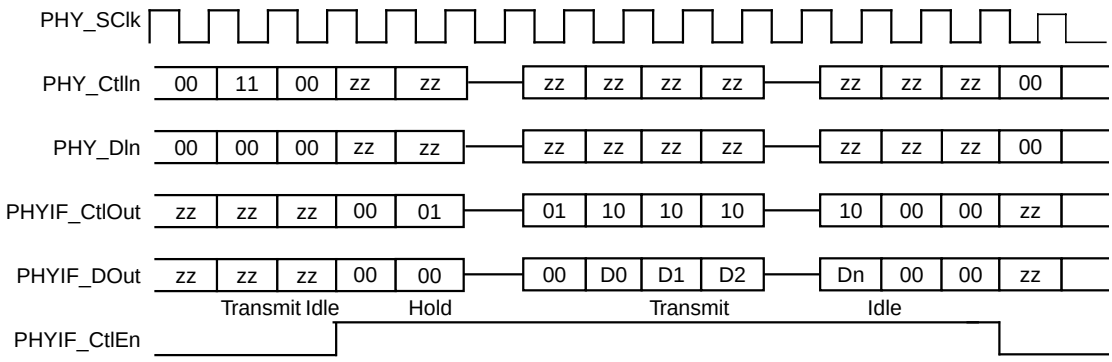
This bit is set when the PHY detects the arbitration reset gap time.

5.1.3.3 Transmit

After the PHY wins the arbitration for the request, it grants the bus to the PHY interface of the LLC by placing the transmit code (0b10) on PHY_CtlIn for one PHY_SClk cycle. The PHY places the idle code on PHY_CtlIn on the following clock. After sampling the transmit state, the PHY interface takes over the interface by placing either the hold code (0b01) or transmit code (0b10) on PHYIF_CtlOut. The PHY interface starts transmitting the packet on the PHYIF_DOut pins when the control pins are set to transmit. When the control pins are set for the hold state, the PHY retains ownership of the bus by placing the data-on state on the bus.

Figure 5.4 shows the timing for packet transmission.

Figure 5.4 Transmit Timing



5.1.3.4 Receive

When the PHY detects the data-on state on the serial bus, it initiates a receive operation by placing the receive state on the PHY_CtlIn pins and holds the PHY_DIn signals HIGH. The PHY indicates the start of a packet by placing the speed code on the PHY_DIn pins, followed by the contents of the packet. The PHY holds the receive state on PHY_CtlIn until the last symbol of the data has been transferred. The idle state on the PHY_CtlIn signals indicates the end of the packet is detected.

Figure 5.5 shows the receive timing.

Figure 5.5 Receive Timing

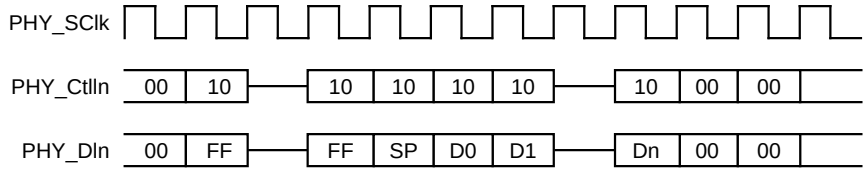


Table 5.2 shows the encoding of the speed code (SP), which is the first received data, as shown in Figure 5.5 above.

Table 5.2 Speed Codes

PHY_DIn[0:7]	Data Rate
00xxxxxx	100 Mbits/s
0100xxxx	200 Mbits/s
01010000	400 Mbits/s
11xxxxxx	Data-on Indication

5.1.4 Single-Phase Retry Mechanism

This section describes the single-phase retry mechanism within the 1394 Node Controller core.

All packets should be sent with a retry_x code. If the received response is ack_busy then the retry block resends the packet. This process continues until the busy retry limit is reached, in which case the retry block flushes the packet from the FIFO and sets the RetEx interrupt bit.

If the retry mechanism is turned off (RetLim field of the PDC Control Register is zero), the packet is transmitted once. If the transmitter receives the busy response, the FIFO is flushed and the RetEx interrupt bit is set. The application can read the Acknowledge Status register to find out what occurred.

5.2 Application Interface Module

This section describes the operation of the Application Interface Module (APPIF). The APPIF module implements the following functions:

- Asynchronous packet transmit/receive interface to application
- Register access from application
- Interrupt generation

The 1394 Node Controller core acts as a slave for the application on this interface; that is, only the application initiates all accesses. The 1394 Node Controller core can interrupt the application and give it status information, so that the application can take appropriate action.

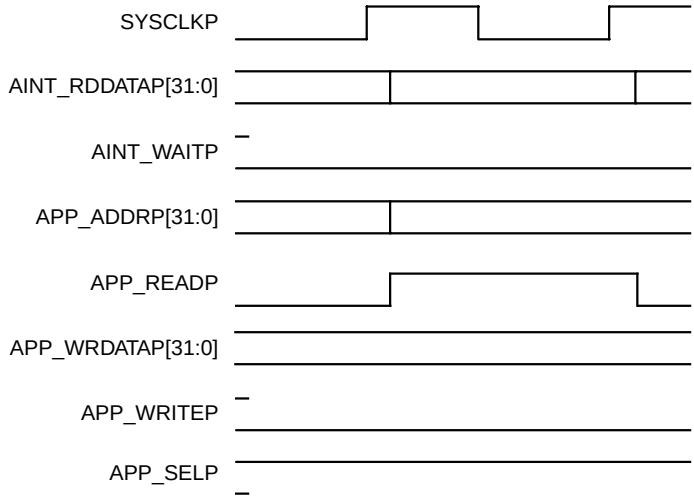
An application can access the 1394 register space or the FIFOs. The application clock and the 1394 local clock (49.152 MHz) are asynchronous to each other. 1394 Node Controller core register accesses are to the buffers or the registers. Depending on the address from the application, the APPIF diverts the transactions to the buffers or the registers. However, every buffer or register access to the application is considered a 1394 register access.

5.2.1 Read Accesses

To start a read cycle, the application asserts APP_SELP and drives APP_READP HIGH. It places the address of the register to read on APP_ADDRP[31:0]. The APPIF module latches the address and provides the address for reading. The APPIF synchronizes the data and presents it to the application with AINT_WAITP deasserted.

Figure 5.6 shows a typical read operation to a 1394 register.

Figure 5.6 Read from the Application

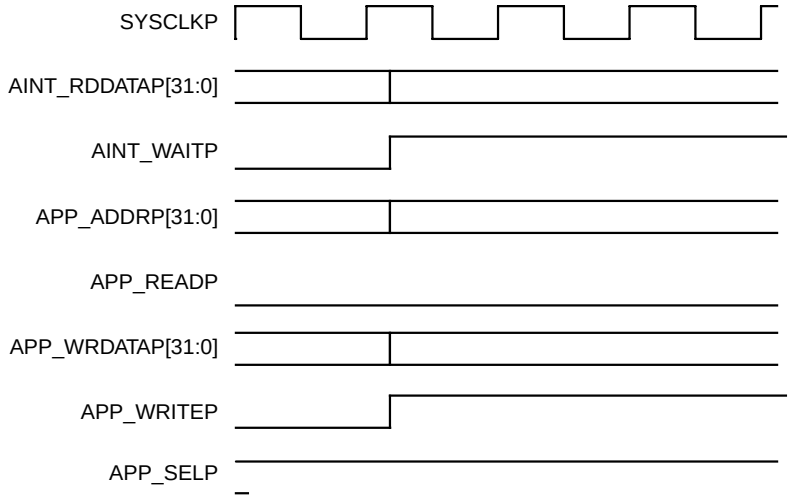


5.2.2 Write Accesses

To initiate a write cycle, the application asserts APP_SELP and holds APP_WRITEP HIGH. It places the address of the register on APP_ADDRP[31:0] and the data to be written on APP_WRDATAP[31:0]. The APPIF deasserts AINT_WAITP to indicate that the write cycle is complete.

Figure 5.7 shows a typical write cycle to the 1394 register.

Figure 5.7 Write from Application



5.3 Packet Header Transformation Modules (PHTs)

Functionally, each of the two PHT modules is divided into the following blocks:

1. Isochronous transmission
2. Isochronous reception (Tight and Loose)
3. Asynchronous request transmission
4. Asynchronous request reception

5.3.1 Isochronous Transmission

Internally, an isochronous transmission cycle starts whenever a cycle start packet is transmitted or received. The LLC arbitrates for the bus if needed to transmit isochronous packets from the PHT.

The PHT asserts the isochronous request (PHT_IsoReqN) signal to the LLC for data transmission.

When a channel is being serviced, if any more channels are to be serviced, the PHT informs the transmitter about the next pending channel and its speed (PHT-NxtChPendN and PHT-NxtChSpd). The transmitter

uses this information to send an L-Request to the cable PHY, when the cable is transmitting the current isochronous channel data (required to meet the gap timings).

The transmitter concatenates two consecutive isochronous packets, if their respective speeds of transmission allow it to do so.

The PHT gets the header from the stream transmit header register or from the DMA Transmit FIFO, depending upon the EHdr bit of the PHT Control and Status Register. If EHdr is set, then the application must provide the header in the required format (as specified in Chapter 2, "Data Formats") before presenting the data. If the EHdr is not set, the application should program the Isochronous Transmit Channel Header Register, in which case the PHT prepends the header from the stream transmit header register to the data in the DMA Transmit FIFO.

5.3.2 Isochronous Reception

The LooseTightISO bit in the PDC Control Register determines whether reception is loose or tight.

5.3.3 Tight Isochronous Cycles

The isochronous receive is triggered on a cycle sync (whenever a cycle start is received or transmitted) and ends on a subaction gap.

If the IHdr bit is not set, the header is only written to the stream receive channel header register by the PHT. If the IHdr bit is set, the header is also clocked into the FIFO before the data.

The isochronous receive is triggered on a cycle sync event (when a cycle start is received or transmitted). The PHT receives the data from the receiver block, and decodes the channel number in the header against the channel number in the stream receive channel selection registers and then stores the data if the channel is enabled.

5.3.4 Loose Isochronous Packets

The loose isochronous packets are implemented just like the tight isochronous packets. However, a loose isochronous packet can be received any time as long as the channel is enabled.

5.3.5 Asynchronous Request Transmission

The PHT can generate read or write request packets. These transfers should always use a transaction label of 0x3F for PHT 0 and 0x3E for PHT 1, which is used to distinguish the received responses from those that should go to the UBUF. Requests that are transmitted using the Universal Transmit FIFO must never use a transaction label of 0x3F or 0x3E.

PHT asynchronous request packet transfers use the DMA interface to transfer the data to/from the core. In addition, the PHT can automatically initiate sequential request transmit transfers after being initially programmed. This operation increases the data transfer rate compared to request transmit transfers, which use the UBUF. The request transmit transfer must interrupt the CPU after each 1394 packet transfer to initiate the next sequential transfer.

The PHT asynchronous request transmission mode can be used for accelerating SBP-2 target-initiated data buffer transfers (especially when the data buffer associated with an ORB was larger than the allowable 1394 packet transfer size that can take advantage of automatic request generation).

5.3.6 Asynchronous Request Reception

Read and write request packets whose address is in a range specified by the DMA space register are routed to a PHT for processing. This rerouting can be useful in SBP-2 applications where the application is an SBP-2 initiator and the DMA space register is programmed to the address region associated with the SBP-2 data buffers and/or ORBs. Using the DMA interface associated with a PHT unburdens the CPU from having to process these transfers (which is the case if the UBUF Receive FIFO was used to process these requests).

5.3.7 Scheduling of Packets for Transmission

The transmitter can transmit one of the following packets depending on the state of the core and the cable:

- Cycle Start, if the core is a Cycle Master
- Isochronous packet

- Asynchronous packet

If the node is a cycle master and the cycle monitor has asserted a request to send a cycle start when the cycle offset counts 125 µs, the transmitter transmits the cycle start packet. Then the transmitter samples the isochronous request signal from the arbiter to see if there are any isochronous packets to be transmitted.

If the core is not the cycle master, it looks for the isochronous request as soon as it receives a cycle start packet. When the core is transmitting the current isochronous packet, the transmitter samples the arb-ISO packet PendN signal to look for additional pending isochronous cycles. If this signal is asserted, the transmitter sends a request to the PHY when it is transmitting the current packet or is receiving another packet. This requesting has to be done to meet the gap timings. Once all the isochronous packets are transmitted and there are no more pending isochronous packets, the transmitter samples the requests from the asynchronous buffers.

5.4 Interrupt Mechanism

The 1394 Node Controller core provides a single interrupt signal per interrupt register and one global interrupt signal to connect to an application on the APP bus. Status indications and state changes from different areas of the core activate interrupts.

The Interrupt and Interrupt Mask registers work in tandem to generate the interrupt on the APP bus when the state of the 1394 Node Controller core changes.

Each bit of the Interrupt registers represents a unique interrupt. A particular interrupt can be masked off when the corresponding bit in the Interrupt Mask registers is zero.

5.4.1 Setting Up the Interrupt Mask Registers

To set up the Interrupt Mask registers, the application writes ones into the bits within the Interrupt Mask registers for all the interrupts the application needs.

5.4.2 Determining and Clearing the Interrupt

Once the application detects assertion of an interrupt signal, it must read the relevant Interrupt register to determine what status has changed. Once the application determines which status change caused the interrupt, it can clear the interrupt at any time after that. To clear the particular interrupt, write a one into that bit location in the relevant Interrupt register.

Chapter 6
Application Operation

This chapter describes how the application handles incoming and outgoing asynchronous packets. This chapter contains the following sections:

6.1 Asynchronous Packet Reception

To enable the reception of asynchronous packets, the application sets the RxEn bit and clears the RxRst bit in the Control 0 Register.

A typical request packet is shown in Figure 6.1.

Figure 6.1 Write Quadlet Request Receive Format

destinationID	tl	rt	tcode	pri
sourceID	destinationOffsetHigh			
destinationOffsetLow				
quadlet_data				
Reserved	spd	Reserved		

To decode this packet, the application decodes the tcode field of the packet. The spd field indicates the speed at which the packet is received. The source_id field indicates the Node ID of the device that sent this packet. Once the application processes the received packet, it can form the response packet shown in Figure 6.2.

Figure 6.2 Write Quadlet Response Packet Transmit Format

RcvdBusID	Reserved	spd	tl	rt	tcode	pri
destinationID			rcode	Reserved		
Reserved						

Here, the destination ID is the same as the source ID from the request packet. The RcvdBusID is the same as the destination bus ID from the request packet and will be used as the source bus ID in the response packet. The spd and tl fields are the same as the request packet fields.

The application can similarly decode other packets.

6.2 Asynchronous Packet Transmission

To enable the transmission of asynchronous packets, the application sets the TxEn bit in the Control 0 Register and clears the TxRst bit. The application can start sending the packets immediately after reset.

When writing an asynchronous packet into the UBUF Transmit FIFO, the application should comply with the data formats that are defined in Chapter 2, "Data Formats." The application must use three addresses to transmit request and response packets. For example, the application must use the UBUF Transmit Clear, UBUF Transmit Next, and UBUF Transmit Last addresses. The 1394 Node Controller core only starts the arbitration for the request once the packet is confirmed into the FIFO. The application can follow the steps given below for writing a packet into the FIFO:

- Write to UBUF Transmit Clear (0x48) (ensures the FIFO is in an appropriate state).
- Write to UBUF Transmit Next (0x40). Data is not confirmed (typically the first 'n – 1' quadlets of the packet are written to this address in a packet with header and payload length of 'n').
- Write to UBUF Transmit Last (0x44). Data is confirmed (typically the last quadlet of the packet is written here).

6.3 Asynchronous Stream Packet Transmission

Asynchronous stream packets are isochronous packets transmitted during a nonisochronous period. The core arbitrates using a fair or priority request (subject to the PriLim field of the Control 0 Register) whenever there is an asynchronous packet to be transmitted, including packets with a tcode of 0xA. Configuring the appropriate registers and then setting the EAST bit in the PHT Control and Status Register is the primary method for transmitting asynchronous stream packets. Also the UBUF can be used to send asynchronous stream packets by using the unformatted data transmit format (the application must generate a proper 1394 isochronous packet including CRCs as the unformatted packet data).

6.4 Example Register Configurations

This section describes eight example register configurations.

6.4.1 Automatic Asynchronous Write Request Packet Generation

To transmit a 16 Kbyte buffer to local node 0x89 at address offset 0x1234.5678.9ABC using 512-byte 1394 data fields at S100 speed, follow these steps:

1. Write 0x0XXX.8XXX to 0xC0 (DBUF FIFOs Level Register). This step initializes the DMATF by setting the DXRst bit (which automatically clears itself).
2. Write 0x0XXX.0020 to 0xA4 (Data Transfer Control Register). This step programs PCnt to 32. A total of 32 write requests are made then the PCntR interrupt is set.
3. Write 0x1FFF.003F To 0xBC (DMA Transfer Threshold Register). In this example, the external DMA controller performs a minimum of 16 quadlet transfers or multiple of 16 for each assertion of the DINT0_DREQP output signal. Therefore DINT0_DREQP must not be asserted if there are less than 64 bytes available to be written in the DMATF (the RdLvl field is programmed such that the DINT0_DREQP associated with the DMARF is never asserted).

4. Write a one to the DEn bit in the DMA Control and Status Register (0xB8). This step enables the DMA interface to assert the DINT0_DREQP signal.
5. Program the Split Time-out Register, which is used for automatically generated requests. For example, write 0x0000.0800 (initial value specified by 1394-1995) to 0x84 (PHT Split Time-out Register).
6. Program the PHT Request Header Registers with the destination address, transmission speed, and 1394 data field length. Write 0xFFE9.1234 to 0x88 (PHT Request Header 0), 0x5678.9ABC to 0x8C (PHT Request Header 1), 0x01F8.0200 to 0x90 (PHT Request Header 2).
7. Write a one to the EWReq and EPCtr bits in the PHT Control and Status Register (0x80). Assuming that the TxEn and RxEn bits in the Control 0 Register (0x08) are set, 1394 write requests are made when there is enough data (data field length amount) in the DMATF.

6.4.2 Automatic Asynchronous Read Request Packet Generation

This example retrieves a 16 Kbyte buffer from local node 0x89 at address offset 0x1234.5678.9ABC using 512-byte 1394 data fields at S100 speed. The steps are as follows:

1. Write 0x8xxx.0xxx to 0xC0 (DBUF FIFOs Level Register). This step initializes the DMARF by setting the DRRst bit (which automatically clears itself).
2. Write 0x0xxx.0020 to 0xA4 (Data Transfer Control Register). This step programs PCnt to 32). Once 32 read requests are made, the PCntR interrupt is set.
3. Write 0x003F.1FFF to 0xBC (DMA Transfer Threshold Register). In this example, the external DMA controller performs a minimum of 16 quadlet transfers or multiple of 16 for each assertion of the DINT0_DREQP output signal. Therefore DINT0_DREQP must not be asserted if there are less than 64 bytes available to be read in the DMARF (the WrLvl is programmed such that the DINT0_DREQP associated with the DMATF is never asserted).
4. Write a 1 to the DEn bit in the DMA Control and Status Register (0xB8). This step enables the DMA interface to assert the DINT0_DREQP signal.

5. Program the Split Time-out Register used for automatically generated requests. For example, write 0x0000.0800 (initial value specified by 1394-1995) to 0x84 (PHT Split Time-out Register).
6. Program the PHT Request Header Registers with the destination address, transmission speed, and 1394 data field length. Write 0xFFE9.1234 to 0x88 (PHT Request Header 0), 0x5678.9ABC to 0x8C (PHT Request Header 1), 0x01F8.0200 to 0x90 (PHT Request Header 2).
7. Write a 1 to the ERReq and EPCtr bits in the PHT Control and Status Register (0x80). Assuming that the TxEn and RxEn bits in the Control 0 Register (0x08) are set, 1394 read requests are made.

6.4.3 CIP Format Isochronous Packet Reception

This example listens to isochronous channel 5, which is a DV format stream. Do not put empty CIP packets in DMARF or put 1394 headers in DMARF. The steps for this example are:

1. Write 0x8XXX.0XXX to 0xC0 (DBUF FIFOs Level Register). This step initializes the DMARF by setting the DRRst bit (which automatically clears itself).
2. Write 0x0000.0000 to 0x94 and 0x0000.0020 to 0x98 (select channel 5 for listening).
3. Write 0x01E7.1FFF TO 0xBC (DMA Transfer Threshold Register). In this example, the external DMA controller performs a 488-byte transfer for each assertion of the DINT0_DREQP output signal. Therefore DINT0_DREQP should not be asserted if there are less than 488 bytes (one DV data block and CIP header) available to be read in the DMARF The WrLvl is programmed such that the DINT0_DREQP associated with the DMATF is never asserted.
4. Write a 1 to the DEn bit in the DMA Control and Status Register (0xB8). This step enables the DMA interface to assert the DINT0_DREQP signal.
5. Write a 1 to the ELis and CCH bits and 0b10 (strip only empty CIPs) in the PHT Control and Status Register (0x80). Assuming that the RxEn bit in the Control 0 Register (0x08) is already set.

6.4.4 Simultaneous CIP Format Isochronous Packet Reception and Transmission

This example listens to isochronous channel 5, which is a DV format stream. Do not put empty CIP packets or 1394 headers in DMARF. Transmit MPEG2 streams on channel 63 at S400 speed. This example assumes a DBUF Transmit FIFO size of 1024 bytes. The steps are as follows:

1. Write 0x8XXX.8XXX to 0xC0 (DBUF FIFOs Level Register). This step initializes the DMARF and DMATF by setting the DRRst and DXRst bits (which automatically clear themselves).
2. Write 0x0000.0000 to 0x94 and 0x0000.0020 to 0x98 (select channel 5 for listening).
3. Write 0x0188.7F40 to 0xA0 (Stream Transmit Channel Header Register). The IsoDataLen field is 392 because two source packets are transmitted when a nonempty CIP packet is transmitted. A non-empty CIP packet consists of the following:

   8-byte CIP header + 4-byte SPH time stamp + 188-byte MPEG packet + 4-byte SPH time stamp + 188-byte MPEG packet

   The Tag field is 0b01 for IEC-61883 formatted data. The ChannelNo field is 63, and the Speed field is S400.
4. Write 0x01E7.0280 to 0xBC (DMA Transfer Threshold Register). In this example, the external DMA controller performs a 488-byte read for each assertion of the DINT0_DREQP output signal. Therefore DINT0_DREQP must not be asserted if there are less than 488 bytes (one DV data block and CIP header) available to be read in the DMARF.

   The external DMA controller must not perform more than a 188-byte (one MPEG source packet) write for each assertion of the DINT0_DREQWP output signal.

   If you want to guarantee that the time stamps in consecutive nonempty CIP 1394 packets differ by at least one in the cycle count portion of the source packet header time stamp, then no more than the number of MPEG packets in a 1394 packet can be allowed in the DMATF at any one time. This condition can be met when DINT0_DREQWP is programmed to not be asserted if there are less than “eight plus the DMATF size minus the 1394 payload size (392

in this example)" empty bytes available to be written in the DMATF (0x0280 in this example).

5. Write 0x0206.C400 to 0xA8 (CIP Header Transmit 0 register). The NSP field is 2 (source packets). The DBS field is 6. The FN field is 0b11. The QPC field is 0. The SPH bit is 1. The beginning DBC is 0.
6. Write 0x2000.0000 to 0xAC (CIP Header Transmit 1 Register). FMT is MPEG2. In FDF, the time shift flag is 0.
7. Write 0x00BC.XXXX to 0xB4 (Stream Transmit Time Stamp Offset Register). The source packet size (SPS) is 188 bytes. The "XXXX" should correspond to the desired offset to be added to each time stamp, which compensates for jitter.
8. Write 0x0187.XXXX to 0xA4 (Data Transfer Control Register). This step results in a nonempty CIP packet being sent for any isochronous cycle in which two MPEG packets are in the DMATF.
9. Write a 1 to the ELis and CCH bits and 0b10 (strip only empty CIPs) and GTS, GenCIP and ETalk bits in the PHT Control and Status Register (0x80). Assume that the RxEn and TxEn bits in the Control 0 Register (0x08) are already set.
10. Write a 1 to the DEn and RActl bits in the DMA Control and Status Register (0xB8). This step enables the DMA interface to assert the DINT0_DREQP signals (DINT0_DREQP and DMAC0_DACKP are used for reads from DMARF; DINT0_DREQWP and DMAC0_DACKWP are used for writes to the DMATF).

6.4.5 Automatic Asynchronous Request Packet Processing

This example has the PHT process incoming read or write requests from node 0x25, which have an address offset below 0x0000.0010.XXXX. This procedure is primarily of use when higher layer software (for example, SBP-2) is being used in which a target node is made aware of a data buffer that has been created at this node. This node wants the target node to read or write the data buffer (for example, a data buffer associated with an SBP-2 ORB). Because this node creates the data buffer, it can program the external DMA controller (that interfaces to the core) to access the data buffer when the core asserts the appropriate DINT0_DREQP output because of an incoming request packet in the specified range.

1. Write 0x8XXX.8XXX to 0xC0 (DBUF FIFOs Level Register). This step initializes the DMARF and DMATF by setting the DRRst and DXRst bits (which automatically clear themselves).
2. Write 0x000F.000F to 0xBC (DMA Transfer Threshold Register). In this example, the external DMA controller performs at most a 16-byte read for each assertion of the DINT0_DREQP output signal. Therefore DINT0_DREQP should not be asserted if there are less than 16 bytes. In this example, the external DMA controller performs a 16-byte write for each assertion of the DINT0_DREQWP output signal. Therefore DINT0_DREQWP should not be asserted if there are less than 16 empty bytes available to be written in the DMATF.
3. Write a one to the DEn and RActl bits in the DMA Control and Status Register (0xB8). This step enables the DMA interface to assert the DINT0_DREQP signals (DINT0_DREQP and DMAC0_DACKP used for reads from DMARF, DINT0_DREQP, and DMAC0_DACKWP used for writes to the DMATF).
4. Write 0x0000.0000 to 0x94 and 0x0200.0000 to 0x98 (select node 25 from which to accept requests).
5. Write 0x0000.0010 to 0x38 (DMA Space Register). This step enables automatic processing of requests from the target node in the desired address range.
6. Write a 1 to the EnDMAS, IHdr, and possibly EHdr bits in the PHT Control and Status Register 0 (0x80).
7. Write a 1 to the TxEn and RxEn bits in the Control 0 Register (0x08). Automatic 1394 write/read responses are made to incoming requests in the specified range (assuming the application DMA controller is capable of parsing 1394 read and write request packets).

6.4.6 Asynchronous Write Request Packet Generation Writing DMA Transmit FIFO Through The Application Interface

This example transmits a 512-byte payload (using the application interface to supply the payload) to local node 0x89 at address offset 0x1234.5678.9ABC using a 512-byte 1394 data field at S100 speed. The steps are as follows:

1. Write 0x0XXX.8XXX to 0xC0 (DBUF FIFOs Level Register 0). This step initializes the DMATF by setting the DXRst bit (which automatically clears itself).

2. Write 0x0XXX.0001 to 0xA4 (Data Transfer Control Register 0). One write request is made then the PCntR interrupt is set.
3. Program the Split Time-out Register used for automatically generated requests. For example, write 0x0000.0800 (initial value specified by 1394-1995) to 0x84 (PHT Split Time-out Register 0).
4. Program the PHT Request Header Registers with the destination address, transmission speed, and 1394 data field length. Write 0xFFE9.1234 to 0x88 (PHT Request Header 0), 0x5678.9ABC to 0x8C (PHT Request Header 1), 0x01F8.0200 to 0x90 (PHT Request Header 2).
5. Write a 1 to the EWReq and EPCtr bits in the PHT Control and Status Register 0 (0x80). Assume that the TxEn and RxEn bits in the Control 0 Register (0x08) are set. A 1394 write request is made when there is enough data in the DMATF 0.
6. Write the data to 0xC4 (DBUF Tx FIFO Data Register 0).

6.4.7 Asynchronous Read Request Packet Generation Reading DMA Receive FIFO Through The Application Interface

This example retrieves a 512-byte buffer from local node 0x89 at address offset 0x1234.5678.9ABC using a 512-byte 1394 data field at S100 speed. The steps are as follows:

1. Write 0x8XXX.0XXX to 0xC0 (DBUF FIFOs Level Register 0). This step initializes the DMARF by setting the DRRst bit (which automatically clears itself).
2. Write 0x0xxx.0001 to 0xA4 (Data Transfer Control Register 0). This step programs PCnt to 1). One read request is made then the PCntR interrupt is set up for receipt of the response.
3. Program the Split Time-out Register used for automatically generated requests. For example, write 0x0000.0800 (initial value specified by 1394-1995) to 0x84 (PHT Split Time-out Register 0).
4. Program the PHT Request Header Registers with the destination address, transmission speed, and 1394 data field length. Write 0xFFE9.1234 to 0x88 (PHT Request Header 0), 0x5678.9ABC to 0x8C (PHT Request Header 1), 0x01F8.0200 to 0x90 (PHT Request Header 2).

5. Write a 1 to the ERReq and EPCtr bits in the PHT Control and Status Register 0 (0x80). Assuming that the TxEn and RxEn bits in the Control 0 Register (0x08) are set, a 1394 read request is made.
6. Read the data from 0xC8 (DBUF Rx FIFO Data Register 0) after the PCntR interrupt is detected.

6.4.8 CIP Format Isochronous Packet Transmission with SYT Time Stamps Using Programmable Frame Synchronization

This example transmits a DV stream on channel 63 at S400 speed. It assumes a DBUF Transmit FIFO size of 1024 bytes

1. Write 0x0XXX.8XXX to 0xC0 (DBUF FIFO Level Register). This step initializes the DMATF by setting the DXRst bit (which automatically clears itself).
2. Write 0xXXXX.01E7 to 0xBC (DMA Transfer Threshold Register). In this example, the core is programmed to request only one 1394 payload from the external DMA controller per isochronous cycle. It is assumed that the external DMA controller performs a 480-byte (one DV data block) write for each assertion of the DINT0_DREQWP output signal. Therefore DINT0_DREQWP must not be asserted when there are less than 488 empty bytes (DV data block plus CIP header) available to be written in the DMATF.
3. Write 0x01E8.7F40 to 0xA0 (Stream Transmit Channel Header Register). the IsoDataLen field is 488 because one source packet will be transmitted when a nonempty CIP packet is transmitted. The nonempty CIP packet consists of a 480-byte DV data block + an 8-byte CIP header. The Tag field is 0b01 for IEC-61883 formatted data. The ChannelNo field is 63, and the Speed field is S400.
4. Write 0x0178.0000 to 0xA8 (CIP Header Transmit 0 Register). The NSP field is 1 (number of source packets). The DBS field is 120 quadlets. The FN field is 0b00. The QPC field is 0. The SPH bit is 0. The beginning DBC is 0.
5. Write 0x0000.FFFF to 0xAC (CIP Header Transmit 1 Register). The FMT field is DVCR. In the FDF field, the time shift flag is 0. The SYT field is programmed to the “no information” value as per IEC-61883 (this will be replaced with the hardware time stamp in the transmitted packet, when appropriate).

6. Write 0x01E0.0000 to 0xB4 (Stream Transmit Time Stamp Offset Register). The source packet size (SPS) is 480 bytes.
7. Write 0x01E4.XXXX to 0xA4 (Data Transfer Control Register). This step results in the sending of a nonempty CIP packet for any isochronous cycle in which a source packet and CIP header (0x01E8) are in the DMATF.
8. Write 0x5F2X.XXXX to 0x84 (Split Time-out/Empty CIP Interval Register). Assuming a 525-60 system, the PPF field is 249 (0x0F9).
9. Write 0x2088.0000 to 0x80 (PHY Control and Status Register), which enables GTS, GenCIP, and ETalk. This step assumes that the RxEn and TxEn bits in the Control 0 Register (0x08) are already set.
10. Write a 1 to the DEn and RActl bits in the DMA Control and Status Register (0xB8). This step enables the DMA interface to assert the DINT0_DREQP signals (DINT0_DREQP and DMAC0_DACKP for reads from DMARF, DINT0_DREQWP and DMAC0_DACKWP for writes to the DMATF).

Customer Feedback

We would appreciate your feedback on this document. Please copy the following page, add your comments, and fax it to us at the number shown.

If appropriate, please also fax copies of any marked-up pages from this document.

Important: Please include your name, phone number, fax number, and company address so that we may contact you directly for clarification or additional information.

Thank you for your help in improving the quality of our documents.

Reader's Comments

Fax your comments to:

LSI Logic Corporation
Technical Publications
M/S E-198
Fax: 408.433.4333

Please tell us how you rate this document: *1394 Node Controller Core Technical Manual.* Place a check mark in the appropriate blank for each category.

	Excellent	Good	Average	Fair	Poor
Completeness of information	____	____	____	____	____
Clarity of information	____	____	____	____	____
Ease of finding information	____	____	____	____	____
Technical content	____	____	____	____	____
Usefulness of examples and illustrations	____	____	____	____	____
Overall manual	____	____	____	____	____

What could we do to improve this document?

If you found errors in this document, please specify the error and page number. If appropriate, please fax a marked-up copy of the page(s).

Please complete the information below so that we may contact you directly for clarification or additional information.

Name _____ Date _____

Telephone _____ Fax _____

Title _____

Department _____ Mail Stop _____

Company Name _____

Street _____

City, State, Zip _____