TSB12LV41A (MPEG2Lynx)

IEEE 1394-1995 Link-Layer Controller for Consumer Applications

SLLS339
MARCH 1999

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Contents

List of Illustrations

List of Illustrations (continued)

List of Illustrations (continued)

List of Tables

List of Tables (continued)

1 Introduction

1.1 Description

The Texas Instruments TSB12LV41A link-layer controller (LLC) (also called MPEG2Lynx) complies with the IEEE 1394-1995 (from here on referred to as 1394) specification for high-performance serial bus, transmits and receives correctly-formatted 1394 packets, detects lost cycle-start packets, and generates and inspects the 32-bit cyclic redundancy check (CRC). The TSB12LV41A is also capable of performing the functions of cycle master (CM), isochronous resource manager (IRM), and bus manager (BM). Support is provided for the IEC61883 standard for transmitting MPEG2 compressed video on 1394, with automatic generation of the common isochronous packet headers and timestamping as required by the IEC 61883 standard.

The TSB12LV41A provides a 1394 interface for high-performance audio, video, and data applications at up to 200 Mbits/s. It is suitable for set-top boxes, multimedia tape, disk drives, and other consumer electronic devices requiring MPEG-2 formatted isochronous data transfer according to the IEC61883 specification. The TSB12LV41A also supports non-MPEG-2/DSS isochronous and asynchronous data transfer with an auto-packetization feature.

The TSB12LV41A interfaces directly to several microprocessors and microcontrollers, including embedded ARM processor, Intel 8051 and the Motorola 68000. The microprocessor interface supports both 8-bit and 16-bit data busses. It can also automatically transform addresses to interface to either big endian or little endian type processors.

The bulky data interface (BDIF) is a 16-bit wide I/O port that enables both transmit and receive of DVB/DSS, isochronous, and asynchronous data. This port is full-duplex, meaning it is capable of transmitting and receiving 1394 packets simultaneously. A separate 8K-byte FIFO, accessible via the bulky data interface, provides logically independent FIFOs for transmit and receive of isochronous, asynchronous, and MPEG2 compressed DVB/DSS data. The 8K-byte FIFO or bulky data FIFO can perform asynchronous packet transmit retry up to 256 times with intervals up to 256×125 μs. The TSB12LV41A supports full-width time-stamped offsets for MPEG2 compressed DVB/DSS transmit and receive, and also performs age filtering functions. A 256-byte FIFO is used to transmit and receive asynchronous control packets.

1.2 Features

The TSB12LV41A supports the following features:

- Provisions of IEEE 1394-1995 Standard for High-Performance Serial Bus
- Interoperable with FireWire™ implementation of the 1394 standard
- Interfaces directly to Texas Instruments TSB11LV01 and TSB21LV03A physical layer (Phy) devices (100/200Mbits/s)
- Multimode 8-/16-bit microcontroller/microprocessor interface supports many processors
- Interrupt driven to minimize host polling
- 8K-byte FIFO supports fully bidirectional MPEG2/DSS, asynchronous, and isochronous modes for transmit and receive
- 64 quadlet (256-Byte) control FIFO accessed through microcontroller interface supports command/status operations
- Bus functions and automatic 1394 Self-ID verification
- Single 3.3-V supply operation with 5-V tolerance using 5-V bias terminals
- High-performance 100-Pin PZ (S-PQFP-G100) package

1.3 Related Documents

- IEEE STD IEEE 1394-1995 High-Performance Serial Bus
- IEC 61883 Digital Interface for Consumer Electronic Audio/Video Equipment

1.4 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	DATA RATE	PACKAGE
TSB12LV41APZ	MPEG2LYNX	3.3 V, 5-V Tolerant I/Os	Up to 200 Mbits/s	100-Pin PQFP

FireWire is a trademark of Apple Computer, Incorporated.

1.5 Terminal Assignments

PZ PACKAGE
(TOP VIEW)

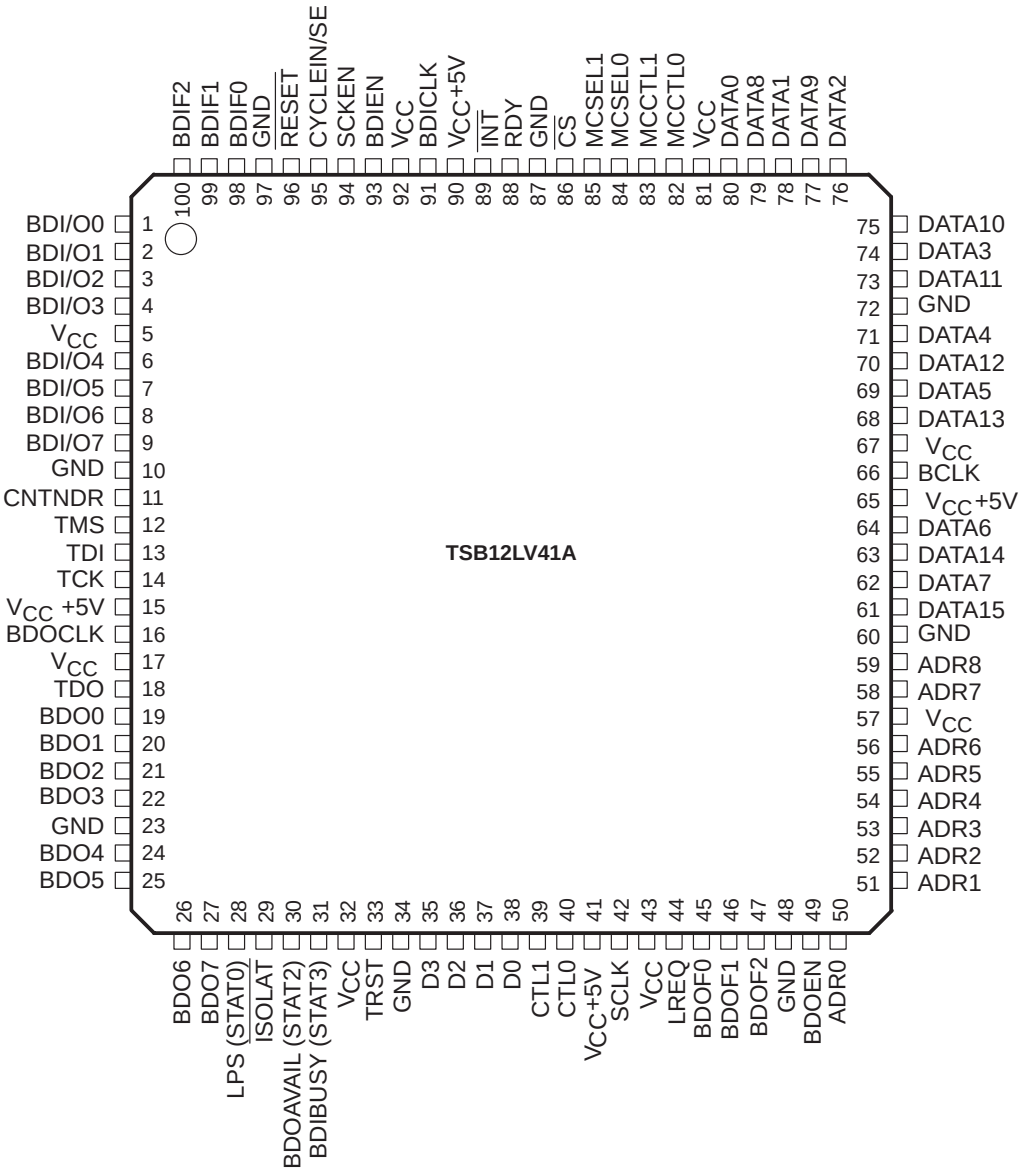


Table 1–1. Terminal Functions

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
ADR0 – ADR8	50,51,52,53,54, 55,56,58,59	I	Microprocessor interface address lines. ADR0 is the most significant bit.
BCLK	66	I	Host bus clock
BDIF2 – BDIF0	100,99,98	I/O	Indicator lines for BDIF. BDIF2 – BDIF0 are used to select the type of data to be written to or read from the BDIF. These terminals can also be used for an application-specific purpose depending on the bulky data interface mode selected.
BDI/O7 – BDI/O0	9,8,7,6,4,3,2,1	I/O	Bidirectional bulky data I/O port . BDI/O7 – BDI/O0 are data lines for high-speed I/O bus for audio/data/video applications. BDIO7 is the most significant bit and BDIO0 is the least significant bit on this bus.
BDIBUSY (STAT3)	31	O	Bulky data interface busy status. When high, BDIBUSY indicates that the FIFO being written to from the BDIF is full. This terminal can also be used to MUX out internal signals for debug purposes (STAT3).
BDICLK	91	I	Bulky data clock
BDIEN	93	I	BDI bus enable. When low, BDIEN causes accesses to BDI-bus to be ignored.
BDO7 – BDO0	27,26,25,24,22, 21,20,19	O	Output Bulky data port. BDO7 – BDO0 are data lines for the high-speed output bus and are used in audio/data/video applications. BDO7 is the most significant bit and BDO0 is the least significant bit on this bus.
$\overline{\text{BDOAVAIL}}$(STAT2)	30	O	Bulky data output available/status output 2. BDOAVAIL indicates that a complete packet (or packets) is available in the bulky data receive FIFO. This terminal can also be used to MUX out internal signals for debug purposes (STAT2).
BDOCLK	16	I	Bulky data output clock
BDOF2 – BDOF0	47,46,45	O	Indicator lines BDO. BDOF2 – BDOF0 are used to select the type of data to be read from the BDO.
BDOEN	49	I	BDO bus enable. When BDOEN is asserted low, accesses to BDO bus are ignored.
CNTNDR	11	I/O	Bus manager contender. CNTNDR indicates to the LLC when the local node is a contender for IRM. This signal can also be driven by the LLC. The default state of this signal is input.
$\overline{\text{CS}}$	86	I	Chip select. $\overline{\text{CS}}$ must be asserted low when the device is to be selected for reads and writes.
CTL0,CTL1	40,39	I/O	Control 0 and control 1 of the Phy-link control bus. CTL0 and CTL1 control the four operations that can occur in this interface.
CYCLEIN/SE	95	I	Cycle in. CYCLEIN is an optional external 8-kHz clock used as the cycle clock. It should only be used when attached to the cycle master node. CYCLEIN is enabled by the cycle source bit and should be tied high when not used. This terminal is used for testing purposes only (SE).
D0 – D3	38,37,36,35	I/O	Phy-link data. D0 – D3 is data input from the Phy-link data bus. Data is expected on D0 – D1 for 100 Mbits/s and D0 – D3 for 200 Mbits/s.

Table 1–1. Terminal Functions (Continued)

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
DATA0 – DATA15	80,78,76,74,71, 69,64,62,79,77, 75,73,70,68,63, 61	I/O	Microprocessor interface data bus. DATA0 is the most significant bit. Some of these terminals may have different functions depending on the microprocessor mode selected.
GND	10,23,34,48,60, 72,87,97		Ground reference
$\overline{\text{INT}}$	89	O	Interrupt. $\overline{\text{INT}}$ signals the host that an interrupt has occurred. This line can be active low or active high dependent on the INTPOL bit in the IOCR register. It is in a high-impedance state when no interrupt has occurred.
$\overline{\text{ISOLAT}}$	29	I/O	Isolation mode/status output 1. $\overline{\text{ISOLAT}}$ is sampled during a hardware reset to determine if isolation is present. When this input signal is high, the internal bus holders on the Phy-link interface are disabled.
LPS (STAT0)	28	O	Link power status/status output 0. This terminal is used to drive the LPS input of the Phy to indicate when the LLC is powered up and active. This output toggles at 1/32 of either the BCLK rate or the SYSCLK rate, depending on the type of microprocessor used. This terminal can also be programmed to MUX out internal signals for debug purposes (STAT0).
LREQ	44	O	Link request. LREQ is an output that makes bus request and accesses to the Phy.
MCCTL0, MCCTL1	82,83	I	Control lines for bus access function depend on MP/MC-type.
MCSEL0, MCSEL1	84,85	I	Select lines for MP/MC-type used. Has impact on function MCTRL,ADR, RDY, and DATA terminals.
RDY	88	O	Ready line. When asserted high, RDY indicates the end of an microprocessor access.
$\overline{\text{RESET}}$	96	I	Reset. $\overline{\text{RESET}}$ is the asynchronous power on reset to the TSB12LV41A and is active low.
SCKEN	94	I	Scan clock enable. This signal is for test purposes only. This signal should be tied low during normal operation.
SCLK	42	I	System clock. SCLK is a 49.152-MHz clock from the Phy.
TCK	14	I	JTAG clock. Used for test purposes only. During normal operation this terminal should be pulled up to V_{CC}.
TDI	13	I	JTAG data in. Used for test purposes only. During normal operation this terminal should be pulled down to ground.
TDO	18	O	JTAG data out. Used for test purposes only.
TRST	33	I	JTAG mode reset. Used for test purposes only. During normal operation this terminal should be pulled down to ground.
TMS	12	I	JTAG mode select. TMS is used for JTAG. During normal operation this terminal should be pulled up to V_{CC}.

Table 1–1. Terminal Functions (Continued)

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
V_{CC}	5,17,32,43,57, 67,81,92		3.3 V (3.0 V – 3.6 V) power supplies
V_{CC}+5V	15,41,65,90		5 V ± 5% power supplies for 5-V tolerant I/O terminals. These terminals can be tied to 3.3 V if no 5-V devices interface to the TSB12LV41A.

2 Architecture

The following sections give an overview of the TSB12LV41A. Figure 2–1 shows a functional block diagram of the TSB12LV41A.

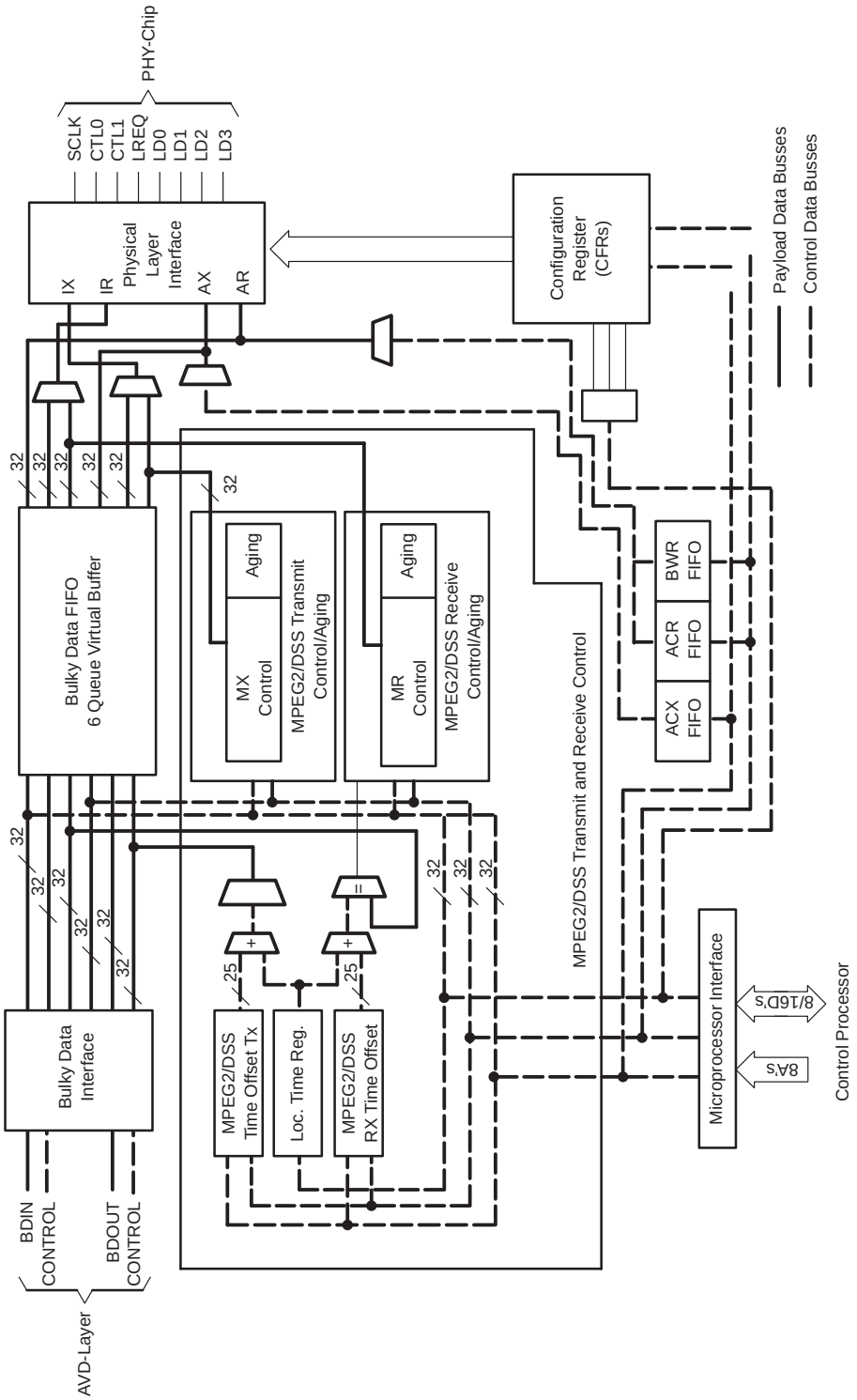


Figure 2–1. Functional Block Diagram

2.1 Bulky Data Interface

The bulky data interface (BDI) enables the TSB12LV41A to provide sustained data rates up to 160 Mbits/s. The bulky data FIFO supports MPEG2 compressed DVB/DSS, asynchronous, and isochronous packets for receive and transmit.

2.2 Bulky Data FIFO

The bulky data FIFO is where transmit and receive data is buffered via the bulky data interface (BDI). The bulky data FIFO is partitioned into six logical FIFOs. Each logical FIFO size is programmable on four quadlet boundaries. These six FIFOs are called:

- BDI MPEG2 (DVB)/DSS Transmit (BMDTX)
- BDI MPEG2 (DVB)/DSS Receive (BMDRX)
- BDI Asynchronous Transmit (BATX)
- BDI Asynchronous Receive (BARX)
- BDI Isochronous Transmit (BITX)
- BDI Isochronous Receive (BIRX)

The following sections give functional descriptions of these logical FIFOs.

2.2.1 BDI MPEG2 (DVB)/DSS Transmit FIFO (BMDTX)

The BMDTX FIFO is used to transmit either MPEG2 (DVB) or DSS data. Data is typically written to this FIFO from the BDI or microcontroller in quadlets (four bytes). See the Bulky Data Interface section (Section 4) for more detail on using this FIFO to transmit MPEG2 (DVB)/DSS data.

2.2.2 BDI MPEG2 (DVB)/DSS Receive FIFO (BMDRX)

The BMDRX FIFO is typically used to store MPEG2 (DVB)/DSS data received from the link-layer core to be forwarded to a high-speed application via the BDI. Data can be written to this FIFO by either the link layer core or the microcontroller. Note that only isochronous port 0 can access this FIFO. See the Bulky Data Interface section (Section 4) for more details.

2.2.3 BDI Asynchronous Transmit FIFO (BATX)

The BATX FIFO is typically used to transmit asynchronous data packets from high-speed applications. Data can be loaded into this FIFO with the BDI or the microcontroller.

2.2.4 BDI Asynchronous Receive FIFO (BARX)

The BARX FIFO is typically used to store received asynchronous data packets to be forwarded to a high-speed application via the BDI. Data is provided to the BDI or the microcontroller interface. This FIFO is also the default location for storing incoming Self-ID packets.

2.2.5 BDI Isochronous Transmit FIFO (BITX)

The BITX FIFO is typically used to transmit isochronous data packets from high-speed applications. Data can be loaded into this FIFO with the BDI or the microcontroller.

2.2.6 BDI Isochronous Receive FIFO (BIRX)

The BIRX FIFO is typically used for receiving isochronous data and forwarding it to a high-speed application. Data is provided to the BDI or microcontroller interface. Isochronous Ports 1 through 7 have access to this FIFO. Each port can be programmed to filter incoming packets according to the Isochronous channel and/or the isochronous header tag value.

2.3 MPEG2 (DVB)/DSS Transmit and Receive Control

The following sections give information on MPEG2 (DVB) and DSS transmit and receive control.

2.3.1 Local Time Register

This register is typically called the cycle timer. It contains the synchronized 1394 cycle timer as specified by the IEEE 1394-1995 standard. The local time register is used to timestamp packets, which determines

when to release outgoing packets to either the packetizer for transmission on 1394 or release incoming packets to the BDI. Section 5.10 describes the cycle timer register in more detail.

2.3.2 MPEG2 (DVB)/DSS Transmit and Receive Control/Aging

This circuitry controls automatic insertion of the common isochronous packet (CIP) header information as defined by the IEC61883 standard. Timestamping is also used for both transmitted and received MPEG2 (DVB)/DSS packets to determine when a packet gets released from the FIFO. The aging algorithm is used to invalidate packets based on the timestamp encapsulated in the MPEG2 (DVB)/DSS header (see Section 3.3).

2.4 Microprocessor/Microcontroller Interface

The microprocessor/microcontroller (MP/MC) interface used as the host controller port, is designed to work with several standard MP/MCs including Motorola 68000, Intel 8051, and embedded ARM processors. This interface supports both 8-bit and 16-bit wide data busses as well as both little endian and big endian microprocessors. This interface has two basic modes of operation, handshake mode and blind access mode. See the Microprocessor section (Section 4) for more details.

2.5 Control FIFO

The control FIFO is partitioned into three logical FIFOs. The size of each of these logical FIFOs is programmable on quadlet boundaries. These three FIFOs are called:

- Asynchronous Control Transmit FIFO (ACTX)
- Asynchronous Control Receive FIFO (ACRX)
- Broadcast Write Receive FIFO (BWRX)

2.5.1 Asynchronous Control Transmit FIFO (ACTX)

The ACTX FIFO is typically used to transmit small asynchronous control packets as sent by the microprocessor/microcontroller. The ACTX FIFO can also be used to support asynchronous traffic at very low data rates. Asynchronous packets are generated by using the ACTXF, ACTXC, ACTXFU, and the ACTXCU registers, all of which access the ACTX FIFO (see Section 3.1.1).

2.5.2 Asynchronous Control Receive FIFO (ACRX)

The ACRX FIFO is typically used to receive asynchronous control packets other than the Self-ID packet. Regular asynchronous control packets typically go to the ACRX FIFO. This FIFO is mapped to the ACRX register. A read from this register accesses the ACRX FIFO (see Section 3.2.1.1).

2.5.3 Broadcast Write Receive FIFO (BWRX)

The BWRX FIFO is typically used to receive asynchronous broadcast write request packets.

2.6 Physical Layer Interface

The physical layer interface provides phy-level services to the transmitter and receiver. This includes gaining access to the serial bus, sending packets, receiving packets, and sending and receiving acknowledgement packets. The TSB12LV41A supports Texas Instruments bus-holder circuity on the Phy-link interface terminals. By using the internal bus holders, the user avoids the need for the high device count 1394 Annex J method of isolation. The bus holders are enabled by connecting the $\overline{\text{ISOLAT}}$ terminal to ground.

2.7 Configuration Register (CFR)

The TSB12LV41A is configured for various modes of operation using CFRs. These registers are accessed via the host microprocessor/microcontroller. The CFR space is 512 bytes, thus the need for a 9-bit address bus. All CFRs are 32-bits wide, and since the microprocessor interface is only 8- or 16-bits wide, it must perform a byte stacking/unstacking operation of the incoming (write) or outgoing (read) microprocessor data. Section 5 gives a map of all the registers and detailed descriptions of all register bits.

3 TSB12LV41A Data Formats

The data formats for transmission and reception of data are shown in the following sections. The transmit format describes the expected organization of data presented to the TSB12LV41A at the host-bus interface. The receive formats describe the data format that the TSB12LV41A presents to the host-bus interface.

3.1 Transmit Operation

3.1.1 Transmitting Asynchronous Control Packets

Asynchronous control packets are typically transmitted by the microprocessor (host) using the asynchronous control transmit FIFO (ACTX). This FIFO is part of the 256 bytes Control FIFO. It is configurable in register 44h (Asynchronous Control Data Transmit FIFO Status.) The ACTX FIFO can also be used for asynchronous data traffic at low data rates.

For transmit the 1394 asynchronous headers and the data are loaded into the ACTX by the microprocessor. The microprocessor has access to the ACTX FIFO through registers 80h – 8Ch. The asynchronous header must fit the format described in Section 3.4.

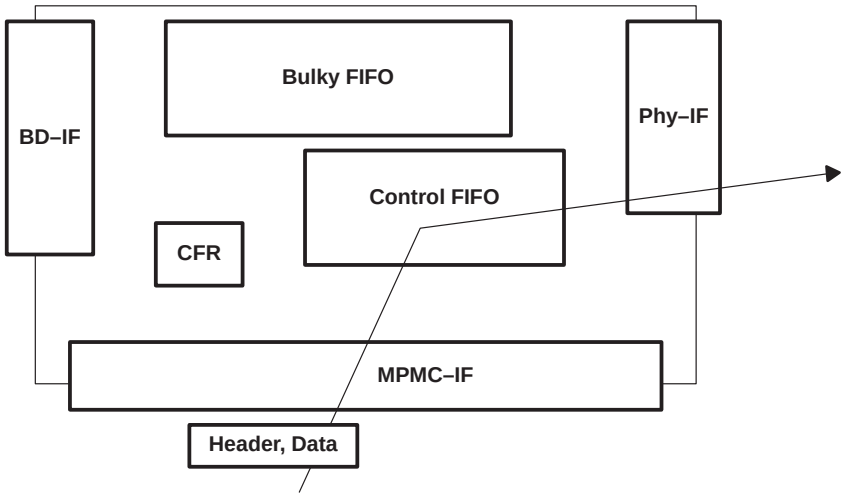


Figure 3–1. Transmit from the Asynchronous Control Transmit FIFO (ACTX)

To transmit an asynchronous packet from the ACTX:

- Register 80h (Asynchronous Control Data Transmit FIFO First): The first quadlet of an asynchronous packet is written to this register by the application software for transmit.
- Register 84h (Asynchronous Control Data Transmit FIFO Continue): All remaining quadlets of an asynchronous packet except the last are written to this register by the application software for transmit.
- Register 8Ch (Asynchronous Control Data Transmit FIFO Last and Send): The last quadlet of an asynchronous packet is written to this register by the application software. Once the last quadlet is written into the ACTX FIFO using this register, the entire packet is transmitted.

NOTE: Register 88h (Asynchronous Control Data Transmit FIFO First and Update) can be used in conjunction with register 8Ch (Asynchronous Control Data Transmit FIFO Last And Send) as an alternative method for transmitting asynchronous control packets from ACTX. The first quadlet and all continuing quadlets except the last are written to register 88h one quadlet at a time. Each quadlet is transmitted immediately. The last quadlet is written to register 8Ch and also transmitted immediately. This method of transmit should only be used in systems where the microprocessor can keep up with the 1394 bus speed.

3.1.2 Transmitting Asynchronous Data Packets

Asynchronous data packets are typically transmitted from the bulky asynchronous transmit FIFO (BATX) using either the bulky data interface (BDIF) or the microprocessor/microcontroller interface (MP/MC IF). The BATX size is configurable in multiples of four quadlets in register 104h (bulky asynchronous size register.) The number of empty quadlet locations available in the BATX is provided in register 108h (bulky asynchronous available register). The transmit operation for the BATX FIFO is configurable in register EC (asynchronous/isochronous application data control register).

The BATX has an auto-packetization feature. This allows the user to program header registers within the MPEG2Lynx CFR's and supply raw data to the MPEG2Lynx for transmit. The MPEG2Lynx automatically inserts the appropriate 1394 headers for transmit. The asynchronous packet is transmitted once the last byte is indicated on bulky data interface or microprocessor interface. If the number of quadlets in the FIFO is not a multiple of 4, then some byte padding is performed (see Section 3.1.4, *Byte Padding*). These headers for auto-packetization are available in registers 1B0 – 1BCh. Please note that the headers programmed in registers 1B0 – 1BCh for auto packetization must match the formats described in Section 3.3, *Asynchronous Data Formats*. The MPEG2Lynx uses the information from these header registers to create the 1394 asynchronous headers. Please note that automatic header insertion is only supported for write request operations (tcode 0 and 1). If the number of bytes in the transmitted packet is different from the datalength field in the header, then receiving node receives the packet with errors.

There are four methods of transmitting asynchronous data from the BATX. The control signals located in register EC that are necessary for these four modes are summarized in the following text. A detailed description is included for each mode.

MODE	ATENABLE	BDAXE	AHIM	DATA SOURCE	HEADER SOURCE
1	1	1	1	Bulky data interface	Configuration registers
2	1	0	1	Microprocessor interface	Configuration registers
3	1	1	0	Bulky data interface	Bulky data interface
4	1	0	0	Microprocessor interface	Microprocessor interface

Mode 1: Transmit Asynchronous Data from BATX Using The BDIF, Data Is Auto-Packetized

The BDIF writes data to the BATX. This data does not include any asynchronous header bytes. Registers 1B0 – 1BCh (AHEAD0 – AHEAD3) are programmed with the 1394 asynchronous header information. The packet is transmitted once the last byte is written into the BATX. The last byte is signaled by the bulky data interface format lines (BDIF[2..0]). (settings for register EC in this mode: ATENABLE=1, BDAXE=1, AHIM=1) Please reference Figure 3–2.

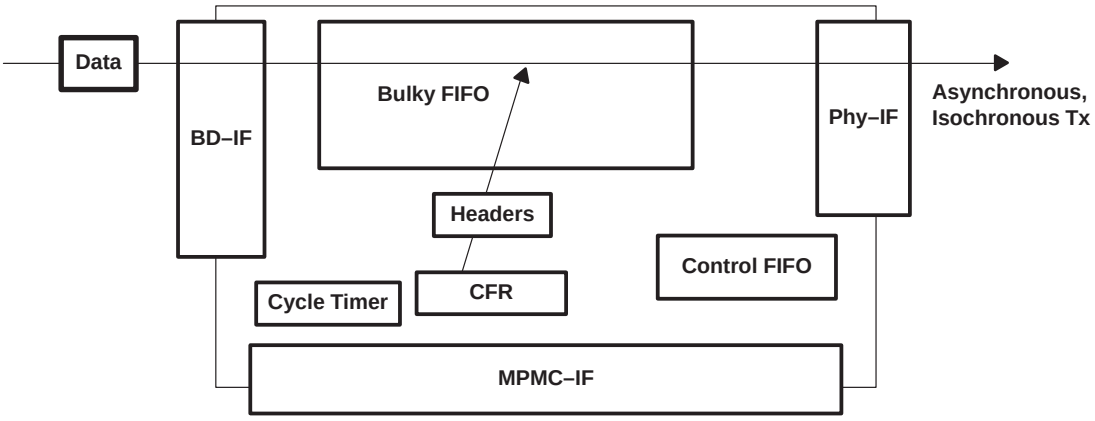


Figure 3–2. Transmit Asynchronous/Isochronous Data from BATX by the Bulky Data Interface with Auto-Packetization

Mode 2: Transmit Asynchronous Data from BATX Using the MP/MC IF, Data Is Auto-Packetized

The MP/MC IF writes data to the BATX using registers 10Ch and 110h. This data does not include any asynchronous header bytes. Registers 1B0 – 1BCh (AHEAD0 – AHEAD3) are programmed with the 1394 asynchronous header information. Register 10Ch (Asynchronous Application Data Transmit FIFO First And Continue) allows the MP/MC to write the all quadlets of the packet to be sent except for the last into the BATX. The last quadlet of the asynchronous packet is written into register 110h (Asynchronous Application Data Transmit FIFO Last and Send.) The data is transmitted once the last quadlet is written into register 110h. (Settings for register EC in this mode: ATENABLE=1, BDAXE=0, AHIM=1) (See Figure 3–3).

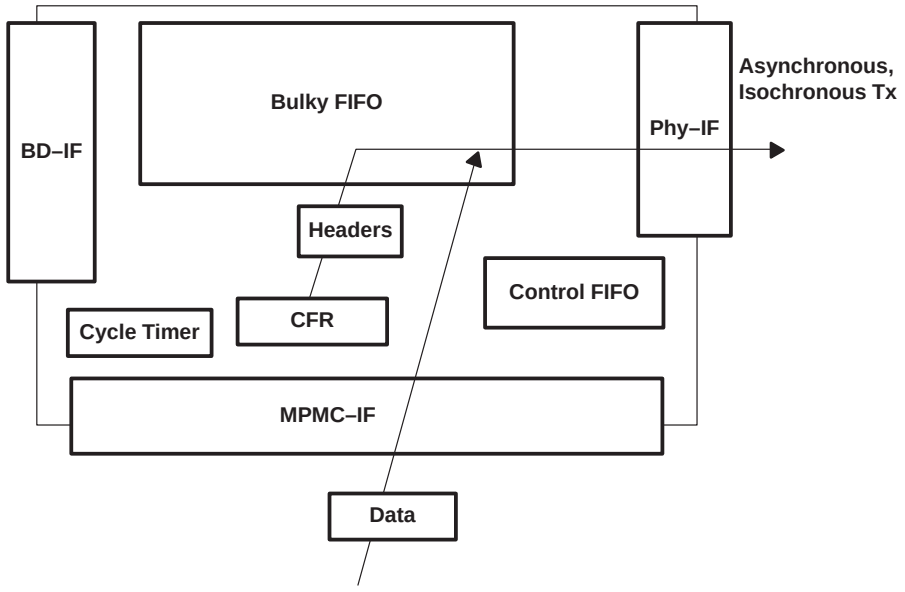


Figure 3–3. Transmit Asynchronous/Isochronous Data from BATX by the MP/MC Interface with Auto-Packetization

Mode 3: Transmit Asynchronous Data from BATX Using the BDIF, Data Is Fully Formatted from the Application

The BDIF writes data to the BATX. This data includes all asynchronous header and data bytes. The header quadlets must match the same format as shown in Section 3.3. The packet is transmitted once the last byte is written into the BATX. The last byte is signaled by the bulky data interface format lines (BDIF[2..0]). (Settings for register EC in this mode: ATENABLE=1, BDAXE=1, AHIM=0) (See Figure 3–4).

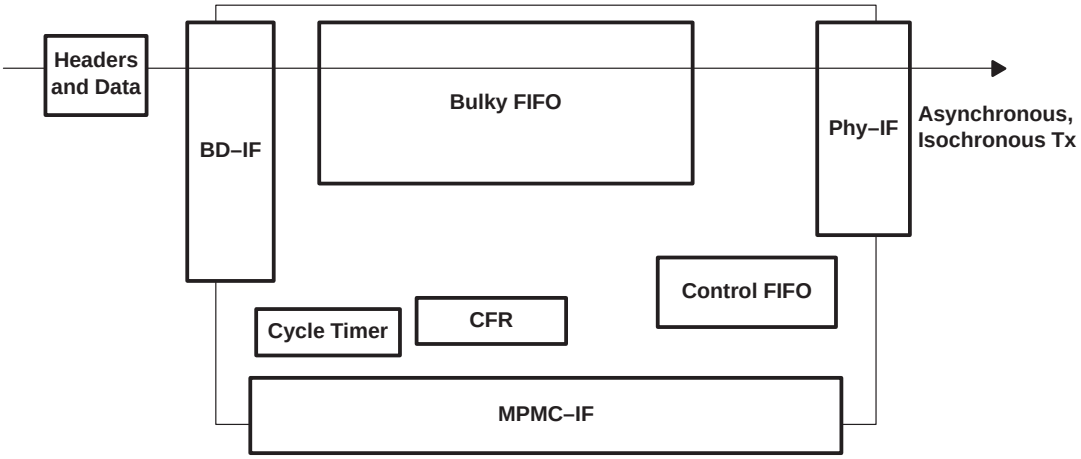


Figure 3–4. Transmit Asynchronous/Isochronous Data from BATX by the Bulky Data Interface, No Auto-Packetization

Mode 4: Transmit Asynchronous Data from BATX Using the MP/MC IF, Data Is Fully Formatted from the Application

The MP/MC IF writes data to the BATX using registers 10Ch and 110h. This data includes all asynchronous header and data bytes. The header quadlets must match the same format as shown in Section 3.3. Register 10Ch (Asynchronous Application Data Transmit FIFO First and Continue) allows the MP/MC to write the all quadlets of the packet to be sent except for the last into the BATX. The last quadlet of the asynchronous packet is written into register 110h (Asynchronous Application Data Transmit FIFO Last and Send.) The data is transmitted once the last quadlet is written into register 110h (Settings for register EC in this mode: ATENABLE=1, BDAXE=0, AHIM=0) (see Figure 3–5).

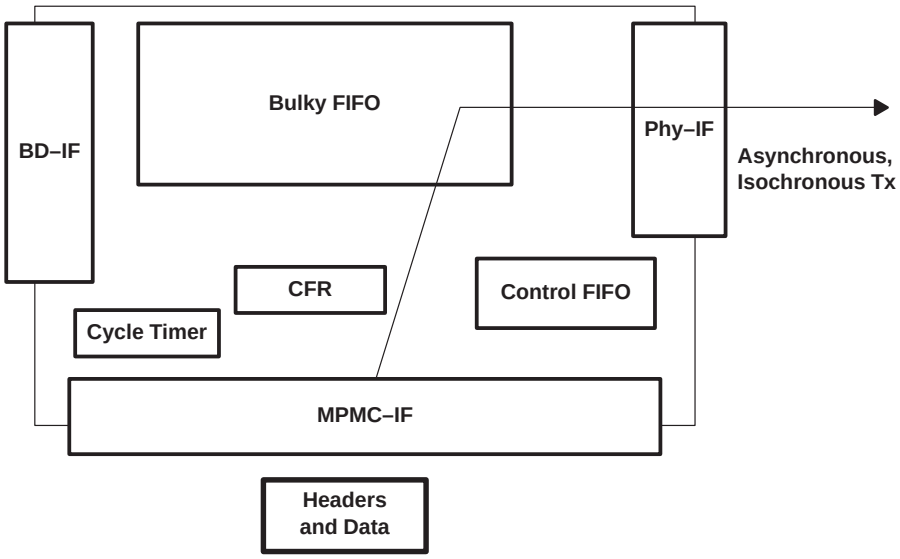


Figure 3–5. Transmit Asynchronous/Isochronous Data from BATX by the MP/MC Interface, No Auto-Packetization

General Asynchronous Transmit Notes

- Packet Flush: The entire BATX FIFO can be flushed by setting the AXFLSH bit in the AICR register (register EC).
- Packet Retries: Bulky asynchronous packets may be automatically retried up to 256 times (BATxRetryNum in register 14Ch, BARTRY) in up to 256 isochronous cycles (BATxRetryInt in register 14Ch, BARTRY). Packet retries for the asynchronous control transmit FIFO are manual.
- Retry Protocol: The MPEG2Lynx uses single phase retries only.
- Auto-packetization: For the bulky data interface, if the data from the host is a multiple of four bytes, then there is no need to indicate *last byte* of an asynchronous packet to the bulky data interface. Similarly, if data from the microprocessor interface is a multiple of four bytes, then all of the data can be written to register 10Ch only. The packet is transmitted on the bus once the number of bytes in the FIFO is equal to the data length field of the asynchronous header.
- Acknowledges received for an asynchronous packet transmitted from the bulky asynchronous transmit FIFO (BATX) are available in register 8h (B Ack register). Bit 23 indicates if the ack received was normal, BATACK[23] = 0, or if it was an error, BATACK[23] = 1. BATACK[24:27] gives the acknowledge error if one occurred.
- Acknowledges received for an asynchronous packet transmitted from the asynchronous control transmit FIFO (ACTX) are available in register 4 (C acknowledge register). Bit 23 indicates if the ack received was normal, CATACK[23] = 0, or if there was an error, CATACK[23] = 1. CATACK[24:27] gives the acknowledge error if one occurred.

3.1.3 Transmitting Isochronous Packets

Isochronous data is transmitted from the bulky isochronous transmit FIFO (BITX) using either the bulky data interface (BDIF) or the microprocessor/microcontroller interface (MP/MC IF). The BITX size is configurable in multiples of four quadlets in register 12Ch (bulky isochronous size register.) The number of empty quadlet locations available in the BITX is provided in register 130h (bulky isochronous available register). The transmit operation for the BITX FIFO is configurable in register EC (asynchronous/isochronous application data control register).

The BITX has an auto-packetization feature which allows the user to program header registers within the MPEG2Lynx CFR's. The application can then supply raw data to the MPEG2Lynx for transmit, and the MPEG2Lynx automatically packetizes the data and insert the appropriate 1394 header for transmit. The amount of data in the transmit FIFO should match the datalength field in the isochronous header. Some byte padding is performed when the data does not end on a quadlet boundary. See Section 3.4.1 for more information on byte padding. If the number of bytes in the packet is different than the datalength field of the header, then the receiving node receives the packet with errors. The 1394 isochronous header for auto-packetization is available in registers 1C0h (Isochronous Header for Auto TX). Please note that the header programmed in register 1C0h must match the format given in Section 3.5.1. The MPEG2Lynx uses the information from these registers to create the 1394 isochronous headers.

There are four methods of transmitting isochronous data from the BITX. The control signals located in register EC that necessary for these four modes are summarized in the following text. A detailed description is also included for each mode.

MODE	ITENABLE	BDIXE	IHIM	DATA SOURCE	HEADER SOURCE
1	1	1	1	Bulky data interface	Configuration registers
2	1	0	1	Microprocessor interface	Configuration registers
3	1	1	0	Bulky data interface	Bulky data interface
4	1	0	0	Microprocessor interface	Microprocessor interface

Mode 1: Transmit Isochronous Data from BITX Using the BDIF, Data Is Auto-Packetized

The BDIF writes data to the BITX. This data does not include any asynchronous header bytes. Register 1C0h (IHEAD0) is programmed with the 1394 isochronous header information. The packet is transmitted once the last byte is written into the BITX. The last byte is signaled to the BITX by the bulky data interface format lines (BDIF[2..0]). The amount of transmitted data in the FIFO should match the data length field in the isochronous header. Some byte padding is performed if the packet does not end on a quadlet boundary. (see Figure 3–2) (settings for register EC in this mode: ITENABLE=1, BDIXE=1, IHIM=1).

Mode 2: Transmit Isochronous Data from BITX Using the MP/MC IF, Data Is Auto-Packetized

The MP/MC IF writes data to the BITX using registers 134h and 138h. This data does not include the 1394 isochronous header bytes. Register 1C0h (IHEAD0) is programmed with the 1394 isochronous header information. Register 134h (Isochronous Transmit FIFO First and Continue) allows the MP/MC to write the all quadlets of the packet to be sent except for the last into the BITX. The last quadlet of the isochronous packet is written into register 138h (Isochronous Transmit FIFO Last and Send.) The data is transmitted once the last quadlet is written into register 138h. (See Figure 3–3) (settings for register EC in this mode: ITENABLE=1, BDIXE=0, IHIM=1).

Mode 3: Transmit Isochronous Data from BITX Using the BDIF, Data Is Fully Formatted from the Application

The BDIF writes data to the BITX. This data includes all isochronous header and data bytes. The 1394 isochronous header is the same format as described in Section 3.5. The packet is transmitted once the last byte is written into the BITX. The last byte is signaled by the bulky data interface format lines (BDIF[2..0]). (Settings for register EC in this mode: ITENABLE=1, BDIXE=1, IHIM=0).

Mode 4: Transmit Isochronous Data from BITX Using the MP/MC IF, Data Is Fully Formatted from the Application

The MP/MC IF writes data to the BITX using registers 134h and 138h. This data includes all isochronous header and data bytes. The isochronous header quadlet is the same format as described in Section 3.5. Register 134h (Isochronous Transmit FIFO First and Continue) allows the MP/MC to write the all quadlets of the packet to be sent except for the last into the BITX. The last quadlet of the isochronous packet is written into register 138h (Isochronous Transmit FIFO Last and Send.) The data is transmitted once the last quadlet is written into register 138h. (see Figure 3–5) (Settings for register EC in this mode: ITENABLE=1, BDIXE=0, IHIM=0).

General Isochronous Transmit Notes

- Packet Flush: The entire BITX FIFO can be flushed by setting the IXFLSH bit in the AICR register (register EC).
- Auto-packetization: For the bulky data interface, if the data from the host is a multiple of four bytes, then there is no need to indicate *last byte* of an isochronous packet to the bulky data interface. Similarly, if data from the microprocessor interface is a multiple of four bytes, then all of the data can be written to register 134h only. The packet is transmitted on the bus once the number of bytes in the FIFO is equal to the data length field of the isochronous header.
- The MPEG2Lynx can transmit more than one isochronous packet per isochronous cycle, if the application provides the 1394 isochronous header with the data and bit 24 of the isochronous header is set to 1. The MPEG2Lynx arbitrates for the bus after every packet that is sent. No concatenated isochronous subactions are allowed.

3.1.4 Notes on Byte Padding

All packets on 1394 must end on a quadlet boundary (4 byte boundary.) The MPEG2Lynx can insert *padding* bits to a data packet that is not quadlet aligned. The MPEG2Lynx only adds padding bits to the last quadlet. This allows for transmission of the data without any modification from the host. For isochronous data that is not a multiple of four bytes, or does not end on a quadlet boundary, the BDIF format lines (BDIF[2–0]) indicates the last byte of the packet. The bulky data interface logic adds padding zeroes to the data to insure that it ends on a quadlet boundary.

3.1.5 MPEG2(DVB)/DSS Transmit

MPEG2(DVB)/DSS packets are usually transmitted from the bulky MPEG FIFO (or bulky DSS FIFO) using the Bulky Data Interface, BDIF. The application transmits single bytes to the bulky data interface which are packetized into quadlets (32 bits). Once four bytes have been written to the bulky data interface, the complete quadlet is written into the Bulky FIFO. The first byte of an MPEG2(DVB)/DSS cell is denoted on the bulky data interface by the format bus. (Please see Section 4.1, *Bulky Data Interface*, for more information) An internal cyclic counter keeps track of MPEG2(DVB)/DSS cell boundaries (192 bytes for MPEG2(DVB) and 144 bytes for DSS.) The microprocessor has access to the bulky FIFOs through registers 160h and 164h (registers 16Ch and 170h for DSS).

NOTE:

Many of the MPEG2(DVB) and DSS registers are *shadow registers*. (Two addresses point to the same register). These registers include the following:

- MCR (F0h) == DCR (F4h)
- MXH (1C8h) == DXH (1D4h)
- MCIPX0 (1CCh) == DCIPX0 (1D8h)
- MCIPX1 (1D0h) == DCIPX1 (1DCh)
- MXTO (DCh) == DXTO (E0h)
- MRH (178h) == DRH (188h)
- MCIPR0 (17Ch) == DCIPR0 (18Ch)
- MCIPR1 (180h) == DCIPR1 (190h)
- MRT (184h) == DRT (194h)
- MRTO (E4h) == DRTO (E8h)

The MPEG2(DVB) transmit operation is controlled in register F0 (MPEG2 formatter control register). The DSS transmit operation is controlled in register F4 (DSS formatter control register.) Please note that the MPEG2Lynx can only support one format at a time.

The size of the MPEG2(DVB) transmit FIFO is set in register 150h. It is programmed in multiples of four quadlets. The available empty quadlet locations in the transmit FIFO are available in register 154h.

The size of the DSS transmit FIFO is set in register 158h. It is programmed in multiples of four quadlets. The available empty quadlet locations in the transmit FIFO are available in register 15Ch.

For MPEG2(DVB) formatted isochronous transmission, there are eight defined bandwidth classes. There are seven defined bandwidth classes for DSS. The bandwidth classes (or transmit classes) define the maximum bandwidth that can be used for a given MPEG2(DVB)/DSS channel. These classes are programmed in the MCR bits of the F0h register for MPEG2(DVB) and in the DCR bits of the F4h register for DSS. These bits are static and must not be changed during operation. On setup, the initiator of an MPEG2(DVB)/DSS channel needs to request a bandwidth class transport stream from the isochronous resource manager (IRM) that is higher than the peak transport stream rate.

The MPEG2Lynx transmits an MPEG2 (DVB)/DSS packet every isochronous cycle, if enough data is in the transmit FIFO (according to the selected transmit class). If there is not enough data in the FIFO to meet the transmit class requirements, then an empty packet is transmitted. An empty packet is transmitted each iso cycle until enough data is in the transmit FIFO.

The MPEG2Lynx performs automatic CIP header and timestamp insertion as defined by the IEC61883 specification. The CIP headers can be programmed in registers 1CCh and 1D0h for MPEG2(DVB) transmit. These registers are available at 1D8h and 1DCh for DSS transmit. (Please see Section 3.3 for more information on timestamps.) The default values of the CIP header registers are sufficient for transmission under normal conditions. The MPEG2Lynx also automatically inserts the 1394 isochronous header. This header can be programmed in register 1C8h for MPEG2(DVB) transmission and in register 1D4h for DSS transmission. The default value of the 1394 isochronous header registers should be sufficient for normal transmission. However, the channel number may need to be changed. The format of the 1394 isochronous header should match the format described in Section 3.5.

There are six different modes for transmitting MPEG2(DVB) or DSS data. The BDMXE/MHIM/MTXTSIN control signals can all be programmed in the F0h register. The BDDXE/DHIM/DTXTSIN control signals can all be programmed in the F4h register.

Mode	BDMXE/ BDDXE	MHIM/DHIM	MTXTSIN/ DTXTSIN	Timestamp From	1394 and CIP Headers From	Data From
1	1	1	1	Cycle timer	CFR	BDIF
2	1	1	0	BDIF	CFR	BDIF
3	1	0	X	BDIF	BDIF	BDIF
4	0	1	1	Cycle timer	CFR	MP/MC
5	0	1	0	MP/MC	CFR	MP/MC
6	0	0	X	MP/MC	MP/MC	MP/MC

Mode 1: MPEG2 (DVB)/DSS Data from Bulky Data Interface, All Headers and Timestamps Automatically Inserted

The data is written to the bulky FIFO from the bulky data interface. The 1394 header and CIP headers are automatically inserted at the beginning of each MPEG2 cell. These headers are calculated internally and are programmable in registers 1C8h – 1D0h. (Registers 1D4h – 1DCh for DSS transmits) The timestamp is also inserted automatically at the beginning of each MPEG2 cell. This timestamp is based on the cycle timer value and an offset value. The transmit offset is programmable in register DCh. (Register E0h for DSS transmit.) See the *Timestamps and Aging* Section (Section 3.3) for more detail on the calculation of timestamps.Figure 3–6 shows the data flow for this mode.

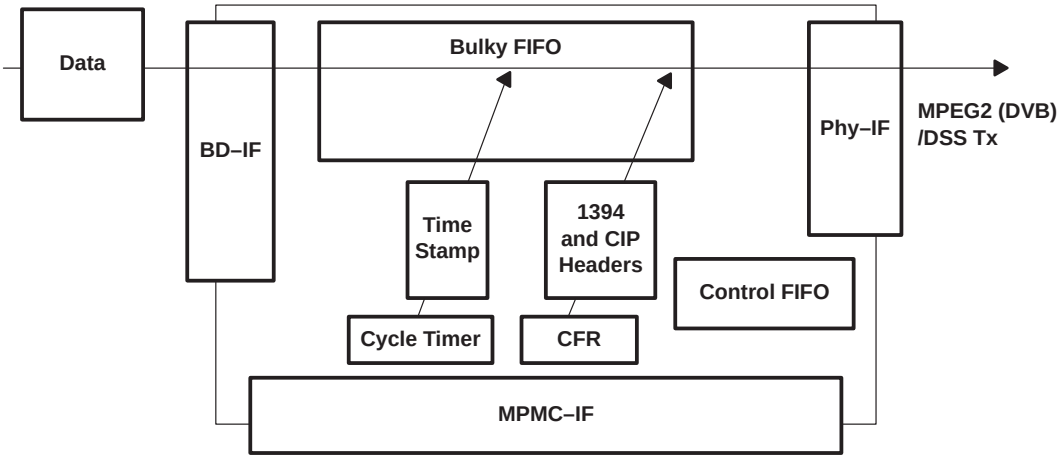


Figure 3–6. MPEG2 (DVB)/DSS Data from Bulky Data Interface, All Headers and Timestamps Automatically Inserted

Mode 2: MPEG2 (DVB)/DSS Data from Bulky Data Interface All Headers Automatically Inserted, Timestamp Supplied by Application

The data is written to the bulky FIFO from the bulky data interface. The 1394 header and CIP headers are automatically inserted at the beginning of each MPEG2 cell. These headers are calculated internally and are programmable in registers 1C8h – 1D0h. (Registers 1D4h – 1DCh for DSS transmits) The timestamp is supplied by the application with the data at the beginning of each MPEG2 cell. Figure 3–7 shows the data flow for this mode.

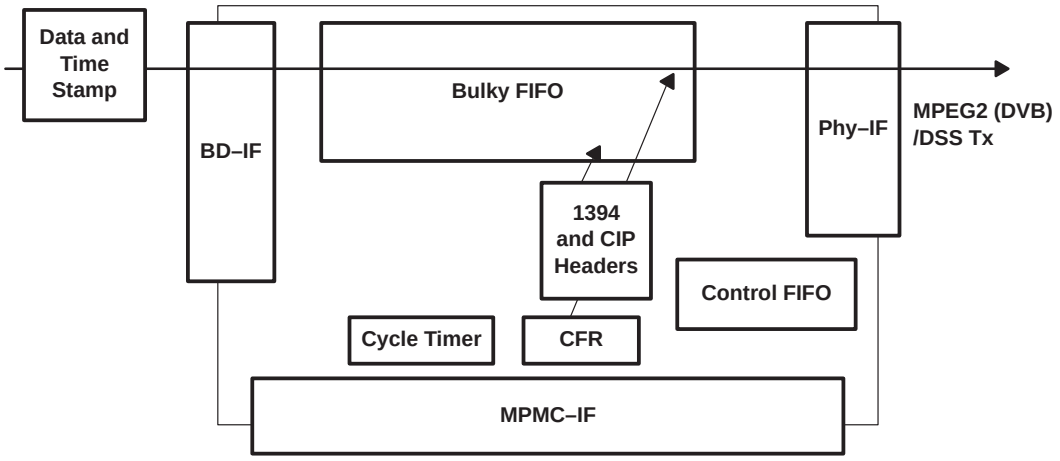


Figure 3–7. MPEG2 (DVB)/DSS Data from Bulky Data Interface All Headers Automatically Inserted, Timestamp Supplied by Application

Mode 3: Data from Bulky Data Interface, All Headers and Timestamps Supplied by Application

The data is written to the bulky FIFO from the bulky data interface. The 1394 header, CIP headers, and timestamp are supplied by the application and are included with the data at the beginning of each MPEG2 cell. Figure 3–8 shows the data flow for this mode.

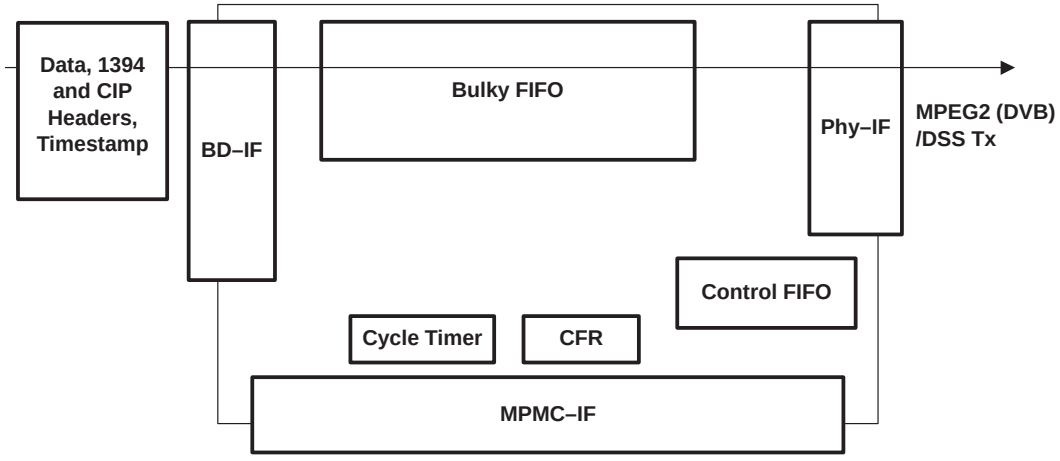


Figure 3–8. MPEG2 (DVB)/DSS Data from Bulky Data Interface, All Headers and Timestamps Supplied by Application

Mode 4: MPEG2 (DVB)/DSS Data from Microprocessor, All Headers and Timestamps Automatically Inserted

The data is written to the bulky FIFO from the microprocessor interface. The microprocessor has access to the bulky FIFO through registers 160h and 164h. (Registers 16Ch and 170h for DSS transmit). The microprocessor writes the first quadlet of the packet, as well as all consecutive quadlets except the last, to register 160h. The last quadlet is written to register 164h, and the packet is transmitted. The packet must be an appropriate size for the transmit class selected. The 1394 header and CIP headers are automatically inserted at the beginning of each MPEG2 cell. These headers are calculated internally and are programmable in registers 1C8h – 1D0h. (Registers 1D4h – 1DCh for DSS transmits) The timestamp is also inserted automatically at the beginning of each MPEG2 cell. This timestamp is based on the cycle timer value and an offset value. The transmit offset is programmable in register DCh. (Register E0h for DSS transmit.) See the *Timestamps and Aging* Section (Section 3.3) for more detail on the calculation of timestamps. Figure 3–9 shows the data flow for this mode.

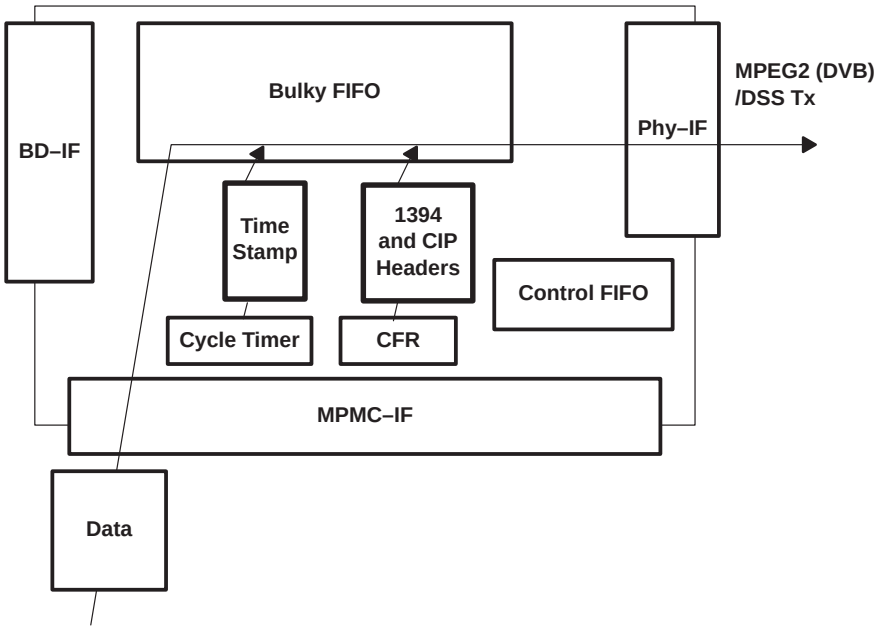


Figure 3–9. MPEG2 (DVB)/DSS Data from Microprocessor, All Headers and Timestamps Automatically Inserted

Mode 5: MPEG2 (DVB)/DSS Data from Microprocessor All Headers Automatically Inserted Timestamp Supplied by Application

The data is written to the bulky FIFO from the microprocessor Interface. The microprocessor has access to the Bulky FIFO through registers 160h and 164h. (Registers 16Ch and 170h for DSS transmit). The microprocessor writes the first quadlet of the packet, as well as all consecutive quadlets except the last, to register 160h. The last quadlet is written to register 164h, and the packet is transmitted. The packet must be an appropriate size for the transmit class selected. The 1394 header and CIP headers are automatically inserted at the beginning of each MPEG2 cell. These headers are calculated internally and are programmable in registers 1C8h – 1D0h. (Registers 1D4h – 1DCh for DSS transmits) The timestamp must be included with the data at the beginning of each MPEG2 cell by the application. Figure 3–10 shows the data flow for this mode.

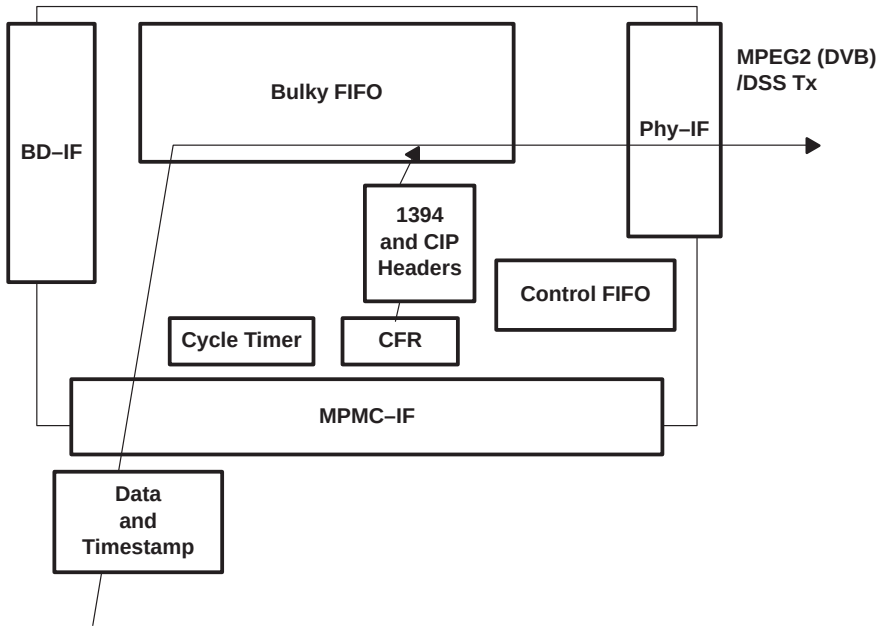


Figure 3–10. MPEG2 (DVB)/DSS Data from Microprocessor All Headers Automatically Inserted Timestamp Supplied by Application.

Mode 6: MPEG2 (DVB)/DSS Data from Microprocessor, All Headers and Timestamps Supplied by Application

The data is written to the bulky FIFO from the microprocessor interface. The microprocessor has access to the bulky FIFO through registers 160h and 164h. (Registers 16Ch and 170h for DSS transmit). The microprocessor writes the first quadlet of the packet, as well as all consecutive quadlets except the last, to register 160h. The last quadlet is written to register 164h, and the packet is transmitted. The packet must be an appropriate size for the transmit class selected. The 1394 header, the CIP headers, and the timestamp must be provided by the application with the data. Figure 3–11 shows the data flow for this mode.

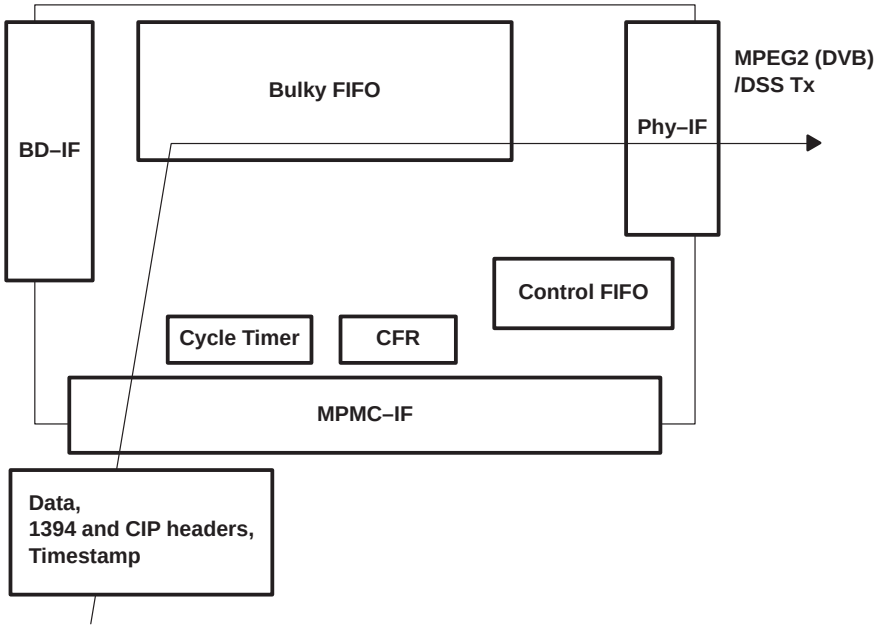


Figure 3–11. MPEG2 (DVB)/DSS Data from Microprocessor, All Headers and Timestamps Supplied by Application

3.1.5.1 MPEG-2 Bandwidth Classes on 1394

- Class 0 (0..1.5 Mbit/s)

  On each isochronous transmission cycle, the MPEG-2 framer checks if 24 valid bytes are available in the BMDTX FIFO; if so, these 24 bytes are taken as the payload for an MPEG-2 packet and sent over 1394; if not, an empty MPEG-2 packet is sent.

  The possible packet sizes are 44 bytes (1/8 cell) and 20 bytes (for an empty packet)

- Class 1 (0..3 Mbit/s)

  On each isochronous transmission cycle the MPEG-2 framer checks if 48 valid bytes are available in the BMDTX FIFO; if so, these 48 bytes are taken as the payload for an MPEG-2 packet and sent over 1394; if not, an empty MPEG-2 packet is sent.

  The possible packet sizes are 68 bytes (2/8 cell) and 20 bytes (for an empty packet)

- Class 2 (0..6 Mbit/s)

  On each isochronous transmission cycle, the MPEG-2 framer checks if 96 valid bytes are available in the BMDTX FIFO; if so, these 96 bytes are taken as the payload for an MPEG-2 packet and sent over 1394; if not, an empty MPEG-2 packet is sent.

The possible packet sizes are 116 bytes (4/8 cell) and 20 bytes (for an empty packet)

- Class 3 (0..12 Mbit/s)

  The MPEG-2 framer checks if a complete M cell (188 bytes) is available on an isochronous transmission cycle. If so, this M cell is used as payload for the M packet. If not a empty packet is sent.

  The possible packet sizes are 212 bytes (1 cell) and 20 bytes (for an empty packet).

- Class 4 (0 – 24 Mbit/s), Class 5 (0 – 36 Mbit/s), Class 6 (0 – 48 Mbit/s), Class 7 (0 – 60 Mbit/s)

  Classes 4 – 7 work like class 3; on each isochronous transmit cycle a check is done to see how many complete M cells are available for transmission. The available M cells are then taken as the payload for the M packet. If there is no complete cell available, then an empty packet is sent.

3.1.5.2 MPEG-2 CIP Header Calculations

- Static values
  - nMXH: tag,chanNum, spd, sy
  - nCIPX0:SID,FN,QPC,SPH
  - nCIPX1:FMT,FDF
- Calculated values
  - nMXH: length
  - nCIPX0: DBS, DBC
  - nCIPX1: length field

Table 3–1. Initial values for the CIP headers (to be set by SW)†

Class	SID	FN	QPC	SPH	DBC	FMT	FDF
	source ID	11	000	1	0...0	100000	0...0
1	source ID	11	000	1	0...0	100000	0...0
2	source ID	11	000	1	0...0	100000	0...0
3	source ID	11	000	1	0...0	100000	0...0
4	source ID	11	000	1	0...0	100000	0...0
5	source ID	11	000	1	0...0	100000	0...0
6	source ID	11	000	1	0...0	100000	0...0
7	source ID	11	000	1	0...0	100000	0...0

† All values listed are in binary format.

nDBC field calculation is done by adding the DBC incremental value of an M packet to be sent to the current DBC value.

nDBS field calculation is done by deciding what size of M packet can be sent over 1394 and interleaving the corresponding value to the link core.

nLength field calculation (in M-formatted, I-packet header) is done by deciding what size of M packet can be sent over 1394 and interleaving the corresponding value to the link core.

Table 3–2. DBC Incremental Numbers, DBS Value and Length Values for MPEG-2 Packets†

M Packet Size	20	44	68	116	212	404	596	788	980
DBC incremental value	0	1	2	4	8	16	24	32	40
DBS value	6	6	6	6	6	6	6	6	6
Data length	8	32	56	104	200	392	584	776	968

† All values listed are in decimal format.

3.1.5.3 MPEG2 (DVB) on 1394 Bandwidth Table

Table 3–3. MPEG2 on 1394 Bandwidth

Class MXC Value	Max. TSP BW (Mbits/s)	Max SP BW (Mbits/s)	Max 1394 BW (Mbits/s)	Possble MPEG2 (DVB)-Packet Sizes Including CIP, M Hdr, and CRCs (Bytes)
0	1.504	1.536	2.816	20, 44
1	3.008	3.072	4.352	20, 68
2	6.016	6.144	7.424	20, 116
3	12.032	12.288	13.568	20, 212
4	24.064	24.576	25.856	20, 212, 404
5	36.096	36.864	38.144	20, 212, 404, 596
6	48.128	49.152	50.432	20,212,404,596,788
7	60.160	61.440	62.720	20,212,404,596,788,980

- TSP BW

  Transport stream package bandwidth (based on 188-byte MPEG-2 cells/ BW on ADV layer)

- SP BW

  Source package bandwidth (based on 192-byte MPEG-2 source cells.)

- 1394 BW

  Overall BW of 1394 bus on physical medium (includes 4-byte M packet transmit header, 4-byte MPEG2 (DVB) packet header CRC, 8-byte CIP header, 4-byte timestamp, actual payload and 4-byte payload CRC). This is the BW that needs to be allocated by the initiator of an MPEG-2 transfer.

3.1.5.4 Simple MPEG2 (DVB) Transmission over 1394

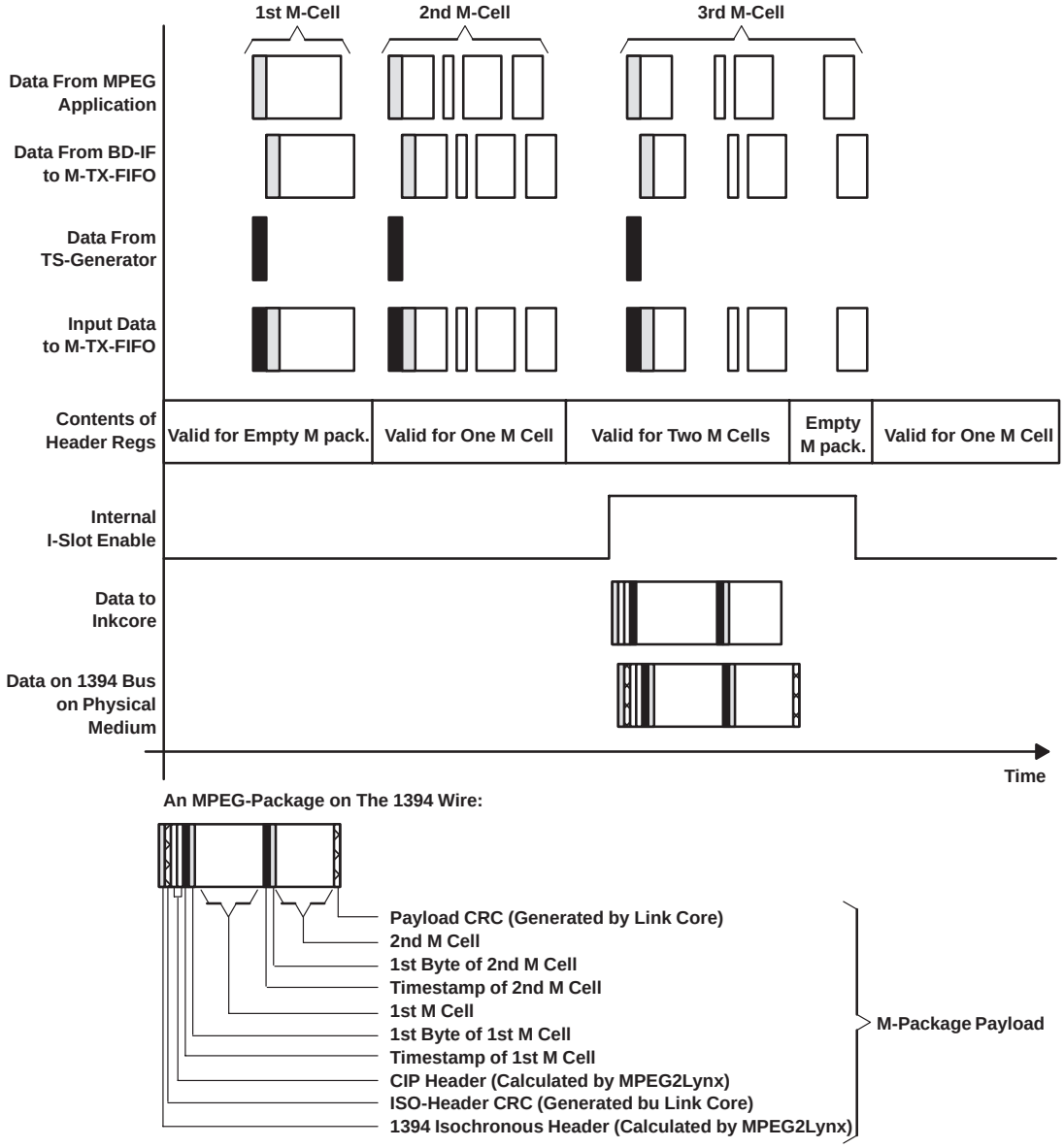


Figure 3–12. Simple MPEG-2 Transmission Over 1394

This diagram shows a simple scenario of a class 5 link (up to 36 Mbit/s). Since there were just two complete M cells available at the time of transmit, only two could be sent. The MPEG-2 data can be written to the BDIF in any manner (burst access, asynchronous, parallel).

3.1.5.5 DSS Bandwidth Classes on 1394

- Class 0 (0..1 Mbit/s)

  This mode is not supported by the TSB12LV41A.

- Class 1 (0..2 Mbit/s)

  On each isochronous transmission cycle, the DSS framer checks if 36 valid bytes are available in the BMDTX FIFO; if so, these 36 bytes are taken as the payload for a DSS packet and sent over 1394; if not, an empty DSS packet is sent instead. The possible packet sizes are 56 bytes (for a quarter cell) and 20 bytes (for an empty packet)

- Class 2 (0..4 Mbit/s)

  On each isochronous transmission cycle the DSS framer checks if 72 valid bytes are available in the BMDTX FIFO; if so, these 72 bytes are taken as the payload for an DSS packet and sent over 1394; if not, an empty DSS packet is sent instead. The possible packet sizes are 92 bytes (2 quarter cell) and 20 bytes (for an empty packet)

- Class 3 (0..8 Mbit/s)

  The DSS framer checks if a complete DSS (140 bytes) cell is available on a isochronous transmission cycle. If so, this DSS cell is used as payload for the DSS packet. If not, an empty packet is sent instead. The possible packet sizes are 164 bytes and 20 bytes.

- Class 4 (0..16 Mbit/s), Class 5 (0..24 Mbit/s), Class 6 (0..32 Mbit/s), Class 7 (0..40 Mbit/s)

  Classes 4 – 7 work like class 3; on each isochronous transmit a check is done, how many complete DSS cells are available to be sent. The available DSS cells are then taken as the payload for the DSS packet. If there is no complete cell available, then an empty packet is sent.

3.1.5.6 DSS packets CIP Header Calculations

- Static values
  - nDXH: tag,chanNum, spd, sy
  - nDCIPX0:SID,FN,QPC,SPH
  - nDCIPX1:FMT,FDF
- Calculated values
  - nDXH: Length
  - nDCIPX0: DBS, DBC
  - nDCIPX1: Length

Table 3–4. Initial Values for the DCIP Headers (to be Set By SW)†

Class	SID	FN	QPC	SPH	DBC	FMT	FDF‡
	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
1	Source ID	10	000	1	0...0	100001	x0...0
2	Source ID	10	000	1	0...0	100001	x0...0
3	Source ID	10	000	1	0...0	100001	x0...0
4	Source ID	10	000	1	0...0	100001	x0...0
5	Source ID	10	000	1	0...0	100001	x0...0
6	Source ID	10	000	1	0...0	100001	x0...0
7	Source ID	10	000	1	0...0	100001	x0...0

† All values listed are in binary format.
‡ x=0 means not time shifted; x=1 means the stream is time shifted

The nDBC field calculation is done by adding the DBC incremental value of an DSS packet to be sent to the current DBC value.

The nDBS field calculation is done by deciding what size of DSS packet can be sent over 1394 and interleaving the corresponding value to the link core.

The nLength field calculation (in DSS-formatted I-packet header) is done by deciding what size of DSS packet can be sent over 1394 and interleaving the corresponding value to the link core.

Table 3–5. DBC Incremental Numbers, DBS Value and Length Values for DSS Packets†

DSS-Packet Size	20	38	56	92	164	308	452	596	740
DBC incremental value	0	n/a	1	2	4	8	12	16	20
DBS value	9	n/a	9	9	9	9	9	9	9
Data length	8	n/a	44	80	152	296	440	584	728

† All values listed are in decimal format.

3.1.5.7 DSS on 1394 Bandwidth

Table 3–6. DSS on 1394 Bandwidth

Class DXC Value	Max. DSS130 BW (Mbits/s)	Max DSS140 BW (Mbits/s)	Max SP BW (Mbits/s)	Max 1394 BW (Mbits/s)	Possble DSS-Packet Sizes Including DCIP, D Hdr, and CRCs (Bytes)
0	n/a	n/a	n/a	n/a	n/a
1	2.08	2.24	2.304	3.584	20, 56
2	4.16	4.48	4.608	5.888	20 , 92
3	8.32	8.96	9.216	10.486	20, 164
4	16.64	17.92	18.432	19.712	20, 164, 308
5	24.96	26.88	27.648	28.928	20, 164, 308, 452
6	33.28	35.84	36.864	38.144	20, 164, 308, 452, 596
7	41.60	44.80	46.08	47.360	20, 164, 308, 452, 596, 740

- DSS130 BW

  Transport stream package bandwidth (based on 130-byte DSS cells/BW on ADV layer)

- DSS140 BW

  Transport stream package bandwidth (based on 140-byte DSS cells/BW on ADV layer)

- SP BW

  Source package bandwidth (based on 144-byte DSS source cells)

- 1394 BW

  Overall BW of 1394 bus on physical medium (includes 4-byte M-packet transmit header, 4-byte DSS-packet header CRC, 8-byte CIP header, 4-byte timestamp, actual payload and 4-byte payload CRC). This is the BW that needs to be allocated by the initiator of an DSS transfer.

3.1.5.8 Simple DSS transmission over 1394

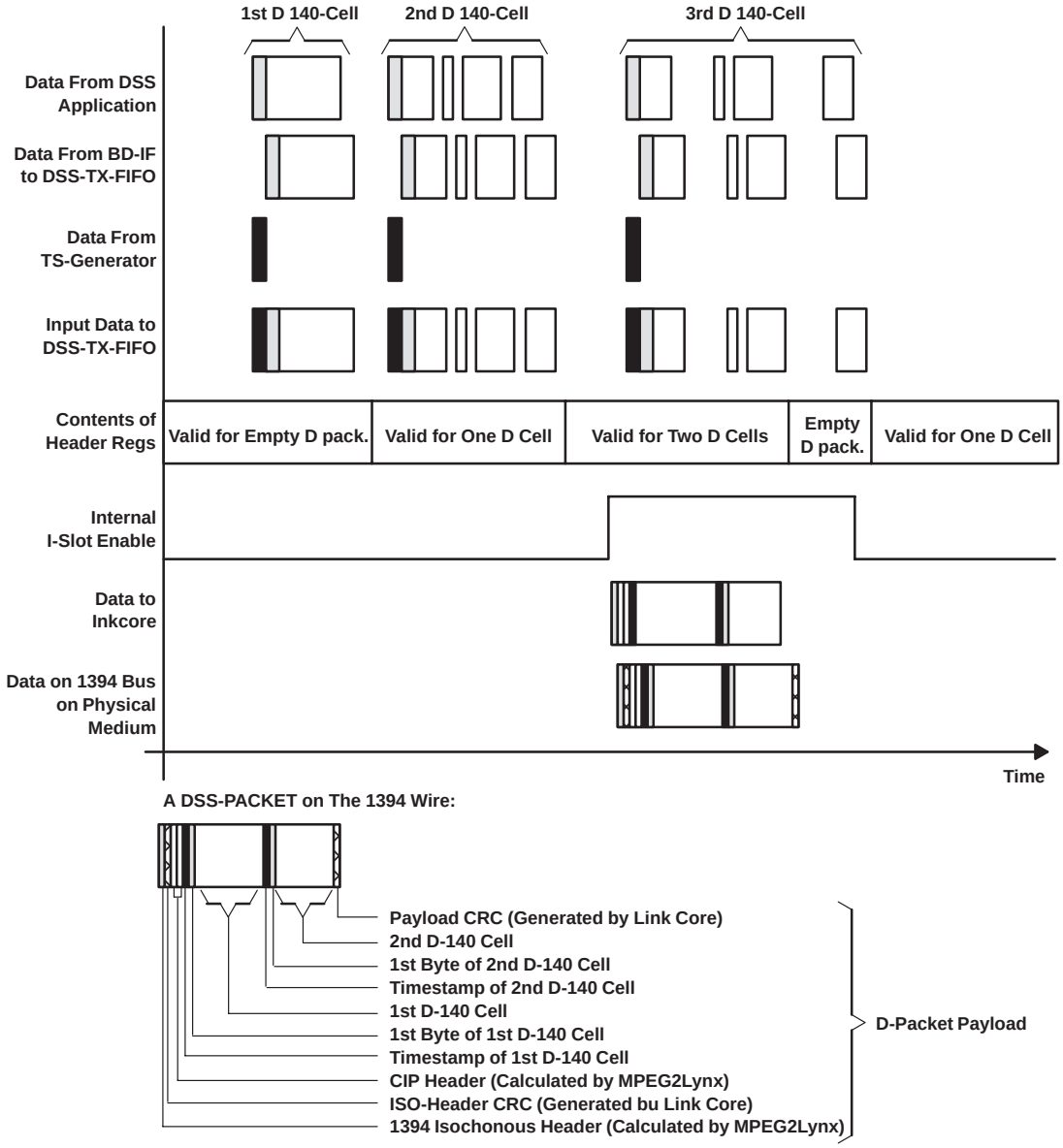


Figure 3–13. Simple DSS Transmission Over 1394

This diagram shows a simple scenario of a class 5 connection (up to 24 Mbit/s) when there were just two complete DSS cells available at the time. The DSS data can be written to the bulky data interface in any manner (burst access, asynchronous, parallel).

3.1.5.9 DSS130/DSS140 Differences

The TSB12LV41A supports the 130-byte DSS-cell-format and the 140-byte DSS-cell-format. Since all DSS-cells need to be converted to 140-byte cells before they can be transmitted, only the 130-byte cells need to be adapted. If the X130 bit (DCR register) is high, the 10 DSS header bytes in the DXX0, DXX1, and DXX2 registers are inserted on transmission. Only 130 bytes are expected on the bulky data interface.

The DSS header bytes are inserted as a 10-byte block directly after the source packet header (timestamp quadlet). The insertion of the header bytes occurs directly after the first bit/byte of a DSS cell is sent to the bulky data interface. On receive, the DRX register is always loaded with the DSS header bytes immediately after the DSS cell is to be released to the video application according to its timestamp.

If the R130 bit in the DCR register is set, only 130-byte DSS cells are given out to the bulky data interface.

The conversion from DSS130 to DSS140 cells is done in the byte merger of the bulky data interface. If the MP/MC writes and reads DSS cells by way of the MP/MC interface, full DSS140 cells need to be generated.

DSS130 packets are embedded inside the DSS140 structure as follows:

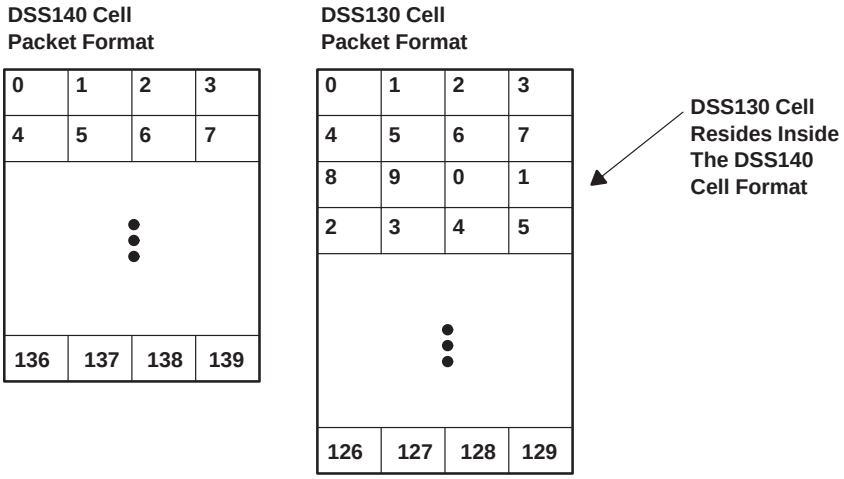


Figure 3–14. DSS130 Packets

NOTE:DSS130 format is supported by the bulky data interface only.

3.2 Receive Operation

3.2.1 Receiving Asynchronous Packets

Asynchronous traffic can be directed into either one or both asynchronous receive FIFOs depending on the type of packet received.

The BWRX (broadcast write receive FIFO) collects asynchronous Self-IDs after a reset.

Asynchronous control packets with small data payloads are meant to go to the ACRX (asynchronous control receive FIFO) while asynchronous packets with large data payload are meant to go directly to the BARX (bulky asynchronous receive FIFO). The ARDM0, ARDM1 and SIDM0 bits in the receive packet router control register (reg 148h) allow various FIFO steering configurations with large data payloads.

- SIDM0 =0 self-ID packages are kept in the BWRX FIFO (used for bus reset recovery)
- SIDM0=1 self-ID packages go directly to the BARX FIFO (used for bus manager function where numerous control packages are expected)
- RIDM0=0 expected response packet (tlabel/tcode in PHYSR register) goes to the ACRX FIFO. UnExpected Response packets (tlabel/tcode in PHYSR register) are routed to a FIFO based on ARDM1 and ARDM0.
- RIDM0=1 expected response packets (tlabel/tcode in PHYSR register) are routed to a receive FIFO based on the setting of ARDM1 and ARDM0. Unexpected response packets (tlabel/tcode in PHYSR register) go to the ACRX FIFO.

NOTE:

Expected response packets are packets whose tlabel and tcode match the programmed value in the PHYSR register (register 38). Unexpected response packets do not have tlabels and tcodes that match the programmed value in register 38.

During the reception of asynchronous packets, the destination FIFO is determined by decoding the destination address in the packet header. The type of steering that takes place is programmable, using the ARDMx bits in register 148h as shown in the following table:

ARDM1	ARDM0	ASYNCHRONOUS-TRAFFIC FLOW	DESTINATION ADDRESS
0	0	All non-broadcast asynchronous packets go to ACRX FIFO.	bus,node,00000,0000000 <= dest_addr <= bus,node,FFFFF,FFFFFFF
		All broadcast asynchronous packets go to BWRX FIFO.	bus,b_node,00000,0000000 <= dest_addr <= bus,b_node,FFFFF,FFFFFFF
0	1	Non-broadcast non-register space asynchronous packets go to BARX FIFO.	bus,node,00000,0000000 <= dest_addr <– bus,node,FFFFE,FFFFFFF
		Broadcast non-register space asynchronous packets go to BARX FIFO.	bus,b_node,00000,0000000 <= dest_addr <– bus,b_node,FFFFE,FFFFFFF
		Non-broadcast register space asynchronous packets go to ACRX FIFO.	bus,node,FFFFF,0000000 <= dest_addr <– bus,node,FFFFF,FFFFFFF
		Broadcast register space asynchronous packets go to BWRX FIFO.	bus,b_node,FFFFF,0000000 <= dest_addr <– bus,b_node,FFFFF,FFFFFFF
1	0	All non-broadcast asynchronous packets go to BARX FIFO.	bus,node,00000,0000000 <= dest_addr <– bus,node,FFFFF,FFFFFFF
		All broadcast asynchronous packets go to BARX FIFO.	bus,b_node,00000,0000000 <= dest_addr <– bus,b_node,FFFFF,FFFFFFF
1	1	Non-broadcast asynchronous packets addressed to lower half of node addressable space go to BARX FIFO.	bus,node,00000,0000000 <= dest_addr <– bus,node,7FFFF,FFFFFFF
		Broadcast asynchronous packets addressed to lower half of node addressable space go to BARX FIFO.	bus,b_node,00000,0000000 <= dest_addr <– bus,b_node,7FFFF,FFFFFFF
		Non-broadcast asynchronous packets addressed to upper half (includes private and register space) of node addressable space go to ACRX FIFO.	bus,node,80000,0000000 <= dest_addr <– bus,node,FFFFF,FFFFFFF
		Broadcast asynchronous packets addressed to upper half (includes private and register space) of node addressable space go to BWRX FIFO.	bus,b_node,80000,0000000 <= dest_addr <– bus,b_node,FFFFF,FFFFFFF

3.2.1.1 Receiving Asynchronous Control Packets

- Reads from Broadcast Write Receive FIFO (BWRX)

  The BWRX FIFO is mapped to register C4h (Broadcast Write Receive FIFO). Read accesses here access the BWRX FIFO. The Status of this FIFO is available in register 50h (Asynchronous Control Data Receive FIFO Status).

- Reads from asynchronous control receive FIFO (ACRX)

  The ACRX FIFO is mapped to register C0h (asynchronous control data receive FIFO). Read accesses here access the ACRX FIFO. The Status of this FIFO is available in register 50h (asynchronous control data receive FIFO status).

3.2.1.2 Receiving Asynchronous packets to the BARX (Bulky Asynchronous Receive) FIFO

When the MPEG2Lynx receives an asynchronous packet, the asynchronous headers and trailer quadlets are automatically copied to registers 118h – 128h. The asynchronous trailer is a quadlet inserted by the receiving MPEG2Lynx. It gives information on the packet speed, number of padding bits, and the acknowledge that was sent.

The asynchronous packet is then received into the bulky asynchronous receive FIFO (BARX). The size of the BARX can be set in register 104h (bulky isochronous size register.) This size is programmed in multiples of four quadlets. The number of quadlets that have been received to the BARX FIFO is available at register 108h. Only complete asynchronous packets can be confirmed into the BARX FIFO. If the storage space available in the BARX FIFO drops to 2 quadlets, then all incoming non-broadcast ASYNC packets are busied off. Partial packets that have accumulated in the BARX at the time that storage space runs out are purged from the FIFO.

The application has the option to receive only data to the BARX (strip headers/trailer) or to receive all data to the BARX (headers/data/trailer.) The ARHS bit in register EC (AICR) controls this function. The first packets in a queue of asynchronous packets stored to the BARX FIFO automatically have their header and trailer quadlets stored to registers. The ARAV interrupt is generated to the application when this operation completes.

There are four methods of receiving asynchronous data to the BARX. The control signals located in register EC that are necessary for these four modes are summarized in the following text. A detailed description is also included for each mode.

MODE	ARENABLE	ARHS	BDARE	OUTPUT INTERFACE	PACKET FORMAT (RECEIVED at BARX)
1	1	1	1	Bulky Data Interface	Data only. Headers are stripped.
2	1	1	0	Microprocessor Interface	Data only. Headers are stripped.
3	1	0	1	Bulky Data Interface	Header/Data/Trailer
4	1	0	0	Microprocessor Interface	Header/Data/Trailer

Mode 1: Receiving Asynchronous Data to the Bulky Data Interface Using the BARX, Headers are Stripped

Data is received by the MPEG2Lynx, and the headers and trailer are automatically copied to registers 118 – 128h. Only the data is received into the BARX FIFO. The ARAV interrupt is signaled once the headers have been copied and the data has been received to the BARX FIFO. The BDOAVAIL signal is activated once a full quadlet is in the FIFO. (Settings for register EC for this mode: ARENABLE=1, ARHS=1, BDARE=1) (see Figure 3–15).

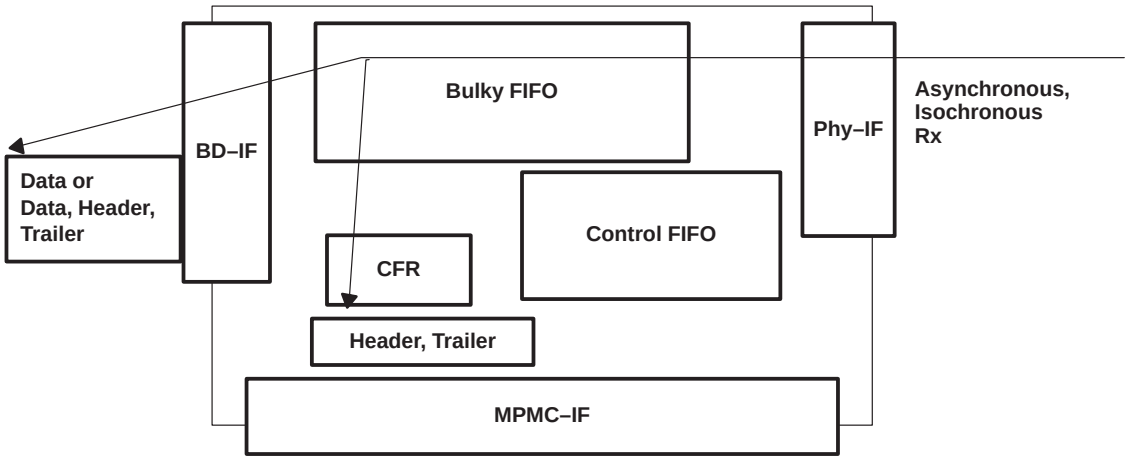


Figure 3–15. Receive Asynchronous/Isochronous Data to Bulky Data Interface

Mode 2: Receiving Asynchronous Data to the Microprocessor Interface Using the BARX, Headers are Stripped

The MPEG2Lynx receives the asynchronous packet, and the headers and trailer are automatically copied to registers 118 – 128h. Only the data is received into the BARX. The microprocessor has access to the BARX through registers 114h (asynchronous application data receive FIFO) (settings for register EC for this mode: ARENABLE=1, ARHS=1, BDARE=0) (see Figure 3–16).

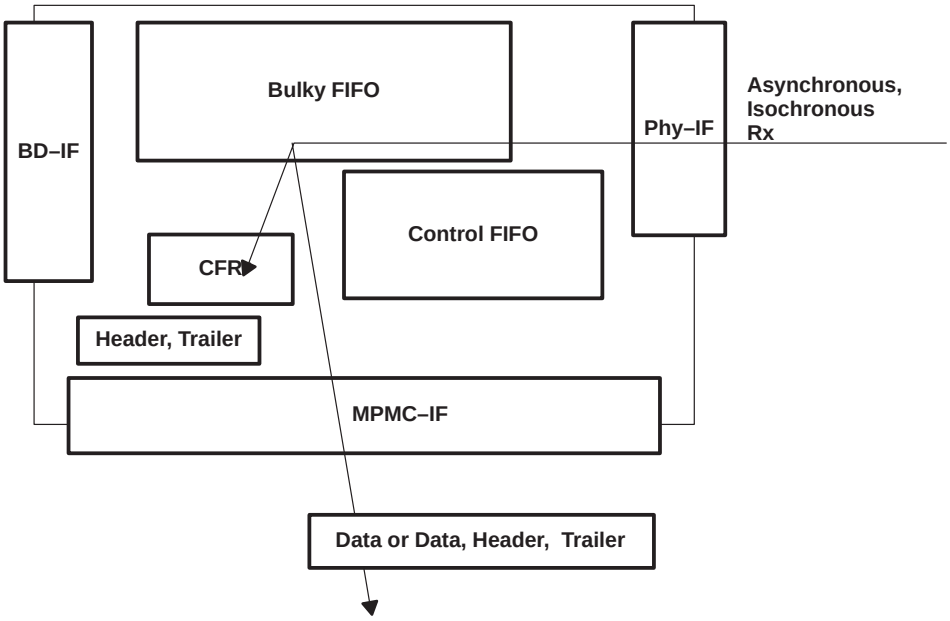


Figure 3–16. Receive Asynchronous/Isochronous Data to Microprocessor Interface

Mode 3: Receiving Asynchronous Header/Data/Trailer to the Bulky Data Interface Using the BARX

Data is received by the MPEG2Lynx, and the headers and trailer are automatically copied to registers 118 – 128h. The headers are received into the BARX FIFO. The ARAV interrupt is signaled once the headers have been copied and the data has been received to the BARX FIFO. The BDOAVAIL signal is activated once a full quadlet is in the FIFO (settings for register EC for this mode: ARENABLE=1, ARHS=0, BDARE=1) (see Figure 3–15).

Mode 4: Receiving Asynchronous Header/Data/Trailer to the Microprocessor Interface Using the BARX

The MPEG2Lynx receives the asynchronous packet, and the headers and trailer are automatically copied to registers 118 – 128h. The headers/data/trailer are received into the BARX. The microprocessor has access to the BARX through registers 114h (asynchronous application data receive FIFO) (settings for register EC for this mode: ARENABLE=1, ARHS=0, BDARE=0) (see Figure 3–16).

General Asynchronous Receive Notes

- Every correctly received asynchronous lock/write request received by the bulky asynchronous receive FIFO (BARX) is acknowledged by sending either an ack_complete (0001b) or an ack_pending (0010b). Register 148, bit 17 (BAckPendEn) programs the ack response code. For packets received to the asynchronous control receive FIFO (ACRX), the response code can be programmed in register C, bit 13 (AckPendEn).
- All correctly received read request packets are acknowledged with Ack_Pending (BusyX).
- Whenever an asynchronous packet is not received correctly to the BARX or ACRX, an Ack_Data_Error (1101b) response is sent regardless of the value of BackPendEn (or AckPendEn). This occurs anytime the data CRC check fails or there is a mismatch between the actual payload and the data length in the header.

3.2.2 Receiving Isochronous Packets

When the MPEG2Lynx receives an isochronous packet, the isochronous header and trailer quadlets are automatically copied to registers 140h and 144h (IRT and IRH). The isochronous trailer is a quadlet inserted by the receiving MPEG2Lynx at the end of a received packet. It gives information on the packet speed, number of padding bits, and whether or not the packet was received correctly.

The isochronous packet is then received into the bulky isochronous receive FIFO (BIRX). The size for the BIRX can be set in register 12Ch (bulky isochronous size register.) This size is programmed in multiples of four quadlets. The number of quadlets that have been received to the BIRX FIFO is available at register 130h. Only complete isochronous packets can be confirmed into the BIRX FIFO. If the storage space available in the BIRX FIFO drops below 2 quadlets, then all incoming isochronous packets will not be received. Partial packets that have accumulated in the BIRX at the time that storage space runs out are purged from the FIFO.

The application has the option to receive only data to the BIRX (strip header/trailer) or to receive all data to the BIRX (header /data/trailer.) The IRHS bit in register EC (AICR) controls this function.

The first packet in a queue of isochronous packets stored to the BIRX FIFO automatically have its header and trailer quadlets stored to registers. The IRAV interrupt is generated to the application when this operation completes.

There are four methods of receiving isochronous data to the BIRX FIFO. The control signals located in register EC that necessary for these four modes are summarized in the following text. A detailed description is also included for each mode.

MODE	IRENABLE	IRHS	BDIRE	OUTPUT INTERFACE	PACKET FORMAT (RECEIVED at BIRX)
1	1	1	1	Bulky Data Interface	Data only. Headers are stripped.
2	1	1	0	Microprocessor Interface	Data only. Headers are stripped.
3	1	0	1	Bulky Data Interface	Header/Data/Trailer
4	1	0	0	Microprocessor Interface	Header/Data/Trailer

Mode 1: Receiving Isochronous Data to the Bulky Data Interface Using the BIRX, Headers are Stripped

Data is received by the MPEG2Lynx, and the headers and trailer are automatically copied to registers 140h and 144h, respectively. Only the data is received into the BIRX FIFO. The IRAV interrupt is signaled once the headers have been copied and the data has been received to the BIRX FIFO. The BDOAVAIL signal is activated once a full quadlet is in the FIFO (settings for register EC for this mode: IRENABLE=1, IRHS=1, BDIRE=1) (see Figure 3–15).

Mode 2: Receiving Isochronous Data to the Microprocessor Interface Using the BIRX, Headers are Stripped

The MPEG2Lynx receives the isochronous packet, and the header and trailer are automatically copied to registers 140h and 144h, respectively. Only the data is received into the BIRX. The microprocessor has access to the BIRX through registers 13Ch (isochronous receive FIFO) (settings for register EC for this mode: IRENABLE=1, IRHS=1, BDIRE=0) (see Figure 3–16).

Mode 3: Receiving Isochronous Header/Data/Trailer to the Bulky Data Interface Using the BIRX

Data is received by the MPEG2Lynx, and the header and trailer are automatically copied to registers 140h and 144h, respectively. The header, data, and trailer are received into the BIRX FIFO. The IRAV interrupt is signaled once the header has been copied and the data has been received to the BIRX FIFO. The BDOAVAIL signal is activated once a full quadlet is in the FIFO (settings for register EC for this mode: IRENABLE=1, IRHS=0, BDIRE=1) (see Figure 3–15).

Mode 4: Receiving Isochronous Header/Data/Trailer to the Microprocessor Interface Using the BIRX

The MPEG2Lynx receives the isochronous packet, and the header and trailer are automatically copied to registers 140h and 144h, respectively. The headers/data/trailer are received into the BIRX. The microprocessor has access to the BIRX through register 13Ch (Isochronous Receive FIFO). (Settings for register EC for this mode: IRENABLE=1, IRHS=1, BDIRE=0) (see Figure 3–16).

3.2.3 MPEG2(DVB)/DSS Receive

The MPEG2Lynx has eight ports for isochronous receive. This allows the user to receive up to eight isochronous channels at one time. However, port 0 is reserved for MPEG2(DVB)/DSS receive. Port 0 can be activated for receive in register Ch, bit 24. When an MPEG2(DVB)/DSS packet is received on port 0, the packet header and trailer quadlets are automatically copied to registers 178h and 184h (registers 188h and 194h for DSS). The data (also headers and trailers if configured in this manner) are placed into the bulky receive FIFO. At this time, an MRAV (or DRAV for DSS receive) interrupt in the extended interrupt register is generated. Whenever aging is enabled, the timestamp of the received packet is then checked to determine a release time for the application. (Please see Section 3.3, *Timestamps and Aging* for more information on release time calculation.)

> **NOTE:** The MRHS bit in register F0h (DRHS in register F4h for DSS) can turn off the timestamp and aging mechanism. Even though all packets are received, the spatial relationship between packets when transmitted are not present on the receiving node. This is a good test case for receiving packets.

The data is usually transferred to the bulky data interface for receive. The format lines on the bulky data interface indicate the first quadlet of an MPEG2(DVB)/DSS cell. (Please see section 5.1, *Bulky Data Interface* for more information on BDIF receive). The microprocessor has access to the bulky receive FIFO register through 168h (register 174h for DSS).

There are four modes for MPEG2(DVB)/DSS receive. They are described in detail in the following text.

MODE	MRHS/DRHS	BDMRE/BDDRE	OUTPUT INTERFACE	PACKET FORMAT (RECEIVED at BULKY FIFO)
1	1	1	BDIF	All data, including headers and trailer
2	0	1	BDIF	Only MPEG2(DVB)/DSS cell data
3	1	0	MP/MC	All data, including headers and trailer
4	0	0	MP/MC	Only MPEG2(DVB)/DSS cell data

Mode 1: Receive MPEG2 (DVB)/DSS, Including Headers and Trailer, to Bulky Data Interface

The data is received on port 0 and all headers/trailers are copied to internal registers. The headers include the 1394 isochronous header and both CIP headers. The timestamp and aging mechanism is used to determine when this packet is output to the bulky data interface. When the timestamp value equals the isochronous cycle timer value, BDOAVAIL on the bulky data interface is activated. BDOAVAIL indicates that a packet is ready to be read from the bulky data interface. The format of this data is shown in section 4.5, *Isochronous Receive (TSB12LV41A to Host Bus)*. Figure 3–17 shows the data flow for this mode.

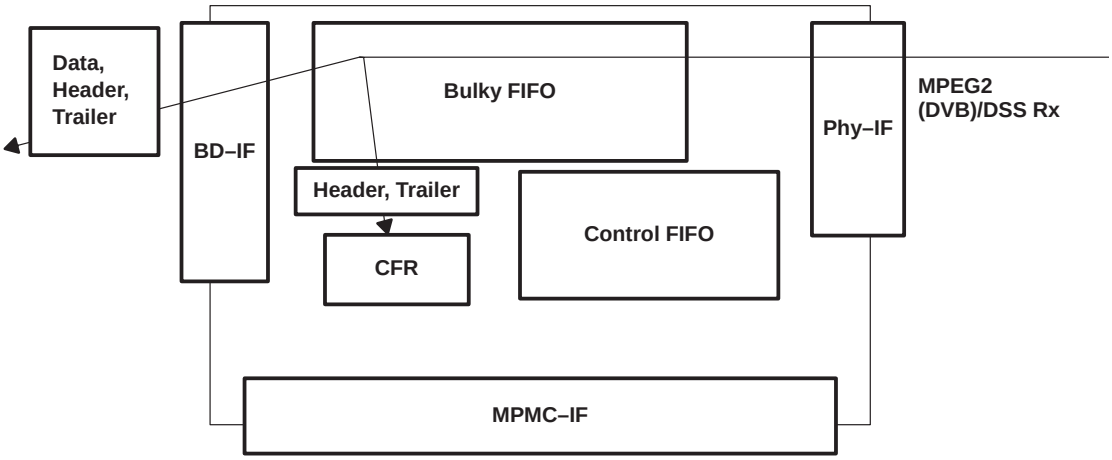


Figure 3–17. Receive All Data, Including Headers and Trailer, to Bulky Data Interface

Mode 2: Receive MPEG2(DVB)/DSS Cell Data Only to Bulky Data Interface

The data is received on port 0 and all headers/trailers are copied to internal registers. The headers include the 1394 isochronous header and both CIP headers. Only the MPEG2(DVB)/DSS cell data is written to the Bulky FIFO. The timestamp and aging mechanism is used to determine when this packet is output to the BDIF. When the timestamp value equals the isochronous cycle timer value, BDOAVAIL on the bulky data interface is activated. BDOAVAIL indicates that a packet is ready to be read from the bulky data interface. The format of this data is shown in Section 3.5, *Isochronous Receive (TSB12LV41A to Host Bus)*. Figure 3–18 shows the data flow for this mode.

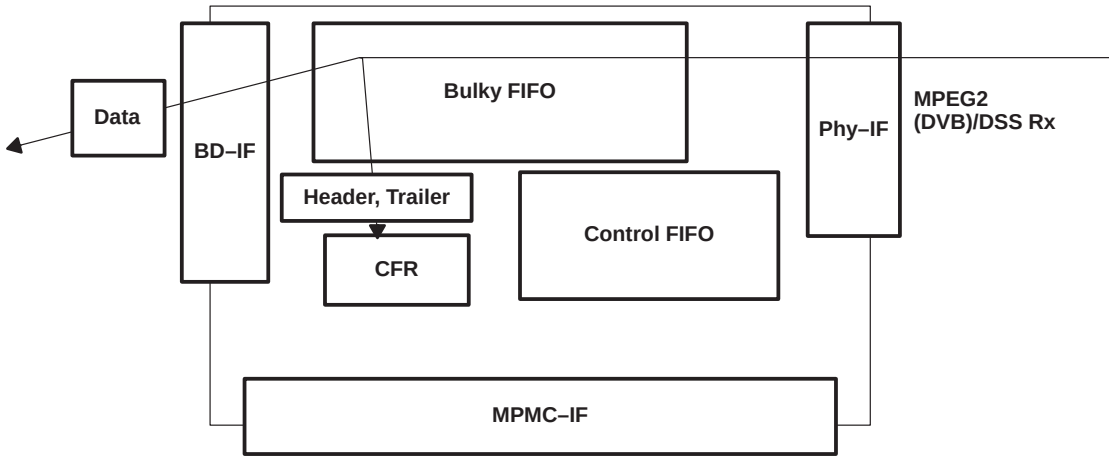


Figure 3–18. Receive MPEG2(DVB)/DSS Cell Data Only to Bulky Data Interface

Mode 3: Receive MPEG2 (DVB)/DSS, Including Headers and Trailer, to Microprocessor Interface

The data is received on port 0 and all headers/trailers are copied to internal registers. The headers include the 1394 isochronous header and both CIP headers. The headers, data, and trailer are written to the bulky FIFO. The timestamp and aging mechanism is used to determine when this packet is output to the microprocessor interface. When the timestamp value equals the isochronous cycle timer value, all data, including headers and trailers is available to the microprocessor interface through the bulky FIFO. The microprocessor has access to the bulky FIFO through register 168h (register 174h for DSS receive). The format of this data is shown in Section 3.5, *Isochronous Receive (TSB12LV41A to Host Bus)*. Figure 3–19 shows the data flow for this mode.

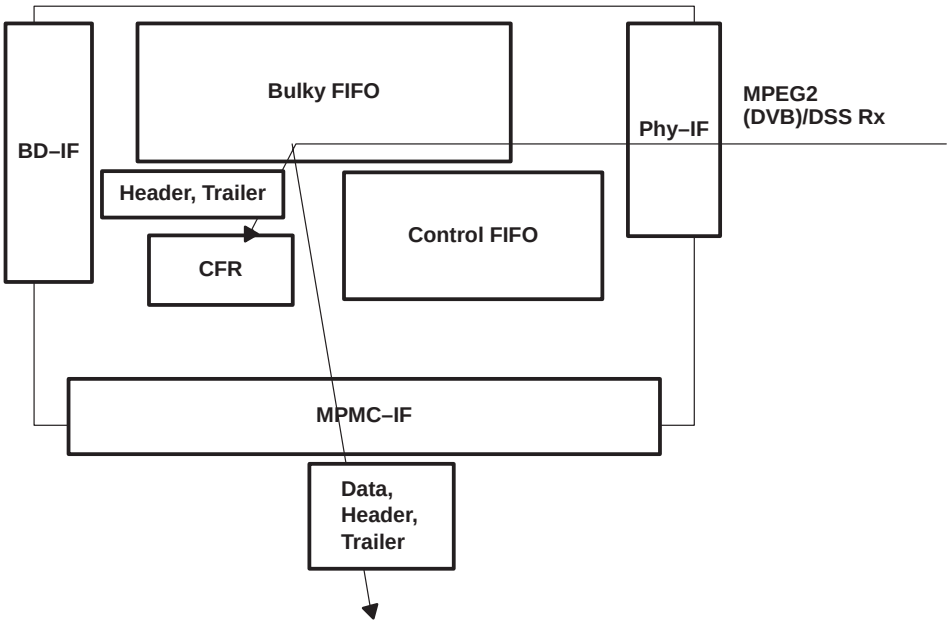


Figure 3–19. Receive All Data, Including Headers and Trailer, to Microprocessor Interface

Mode 4: Receive MPEG2(DVB)/DSS Cell Data Only to Microprocessor Interface

The data is received on port 0 and all headers/trailers are copied to internal registers. The headers include the 1394 isochronous header and both CIP headers. The timestamp and aging mechanism is used to determine when this packet is output to the microprocessor interface. When the timestamp value equals the isochronous cycle timer value, all data, including headers and trailers is available for the microprocessor interface The microprocessor has access to the bulky FIFO through register 168h (register 174h for DSS receive). The format of this data is shown in Section 3.5, *Isochronous Receive (TSB12LV41A to Host Bus)*. Figure 3–20 shows the data flow in this mode.

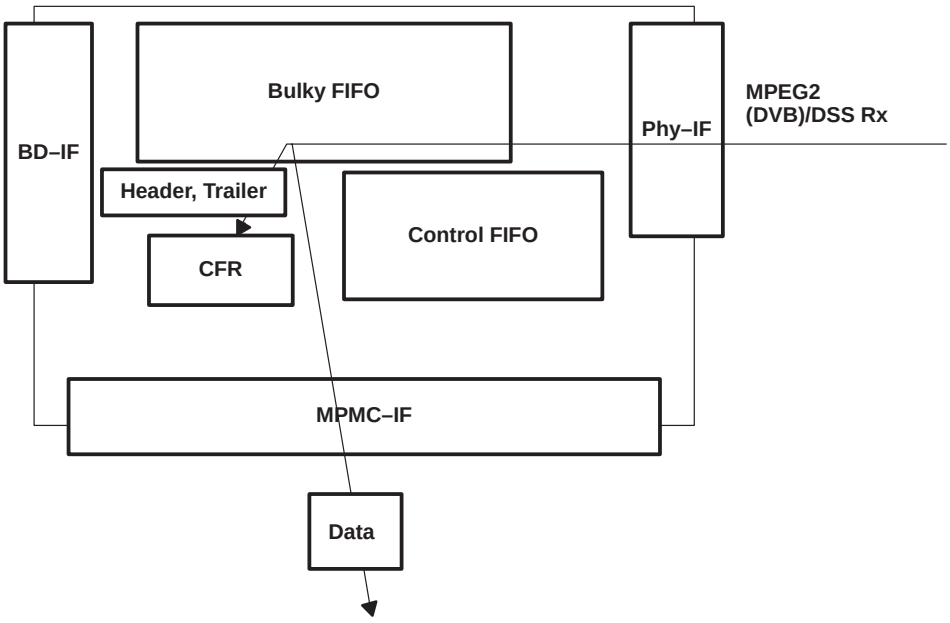


Figure 3–20. Receive MPEG2(DVB)/DSS Cell Data Only to Microprocessor Interface

3.3 Timestamps and Aging

The MPEG2Lynx uses timestamping to preserve the temporal relationship of MPEG2 (DVB)/DSS packets in a transport stream from the transmitting 1394 node to the receiving 1394 node.

The transmitting MPEG2Lynx (transmitting onto 1394) puts a timestamp on each MPEG2 (DVB)/DSS cell it sends out. The timestamp value on a transmitted MPEG packet is the sun of the current value of the isochronous cycle timer plus a user programmable transmit offset value (transmit timestamp offset). A receive offset value (receive timestamp offset) can also be added. The resulting new timestamp is the sum of the old timestamp value plus the receive offset value. This new timestamp value determines when the MPEG2 packet should be released to the application. The MPEG2Lynx releases the packet to the application whenever the timestamp value matches the current cycle timer value of the receiving node.

Old packets or packets whose timestamp has expired, remain in the FIFO and are not transmitted or released to the application. Aging allows the MPEG2Lynx to flush these old packets from the transmit or receive FIFOs. The action ensures that the FIFOs are not overflow with the old packets. Aging on the receiving node check the timestamp of the incoming MPEG2 packets. If the timestamp has a value that is less than the cycle timer (the time has already passed when the packet was supposed to be received), the FIFO logic flushes the packet. The aging function on the transmitting node checks the timestamp of a packet that is waiting to be transmitted. If the packet is too old to transmit, then the MPEG2Lynx discards the packet before transmission.

The timestamp and aging functions are controlled by register F0h (for MPEG2 (DVB)) and register F4h (for DSS). Bits 6 and 7 enable transmit and receive aging. Bit 29 enables automatic timestamp insertion on Mpeg2 (DVB) or DSS packet transmits.

Register DCh (for MPEG2 (DVB)) and register E0h (for DSS) program the transmit timestamp offset. This offset value is added to the cycle timer value to form the timestamp of a transmitted MPEG2/DSS cell. This offset should be large enough to insure that when it is received the timestamp value of the packet is greater than the current cycle timer at the receiving node. Register E4h (for MPEG2 (DVB)) and register E8h (for DSS) programs the receive timestamp offset. This offset is added to a received packet's timestamp value. This new timestamp value (old timestamp + receive offset) determines when the packet is released to the application. Inclusion of both a transmit and receive offset gives a system designer a way to insure that both receiving and transmitting nodes are able to responsibly handle timestamped data.

3.3.1 Timestamp Calculation on Transmit

The transmit timestamp is calculated by adding an offset value of the cycle timer value (register 28h). The offset is programmed in the transmit timestamp offset register (register DCh for MPEG2 (DVB) and register E0h for DSS) (see Figure 3–21). XTO (transmit timestamp offset) adds to bits 7 – 31 to the cycle timer register in normal operation. This means that XTO only adds cycle counts or cycle offsets to the cycle time value to form the timestamp. With the extended timestamp offset value, the entire 32-bit XTO transmit offset value is added to the cycle timer to form the timestamp. The extended timestamp offset value enable is available in register F0h (register F4h for DSS), bit 17.

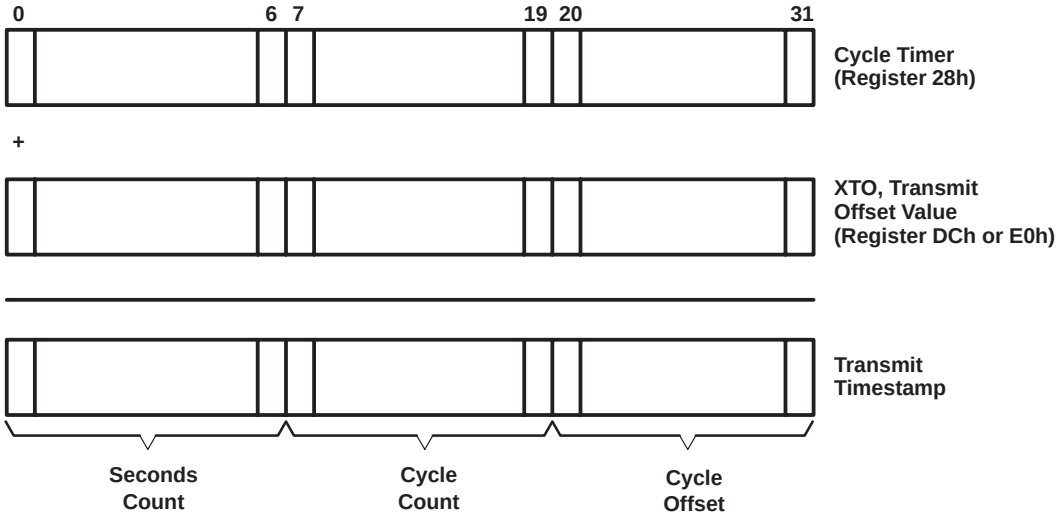


Figure 3–21. Determination of Transmit Timestamp

The transmit timestamp values are limited in order to prevent the user from creating invalid timestamps. For the transmit timestamp, the result of the addition of the cycle timer and transmit offset are limited to the following:

BIT NUMBERS	VALUE	VALID RANGE
0 – 6	Seconds count	Between 0 and 127
7 – 19	Cycle count	Between 0 and 7999
20 – 31	Cycle offset	Between 0 and 3071

3.3.1.1 Timestamp Cycle Timer (See Figure 3–21)

The cycle timer operates on an internal clock of 24.576 MHz. When the cycle offset reaches a value of 3071 (BFFh), 125 μs have elapsed (3071/24.576 MHz = 125 μs). This is exactly 1 isochronous cycle period. It is necessary to limit the cycle offset to 3071 to avoid creating a timestamp with an invalid cycle offset field (value > 3071 is invalid). When the cycle reaches 3071, it simply rolls over to 0 and starts again. If the sum of the cycle timer and XTO, the transmit offset value, results in a timestamp cycle offset greater than 3071, then the cycle count field is incremented by 1 and the resulting cycle offset value is (cycle offset – 3072).

3.3.1.2 Timestamp Cycle Count (See Figure 3–21)

Since the cycle count operates at a frequency of 8 Mhz (1/125 μs = 8 kHz), 1 second will have elapsed when the cycle count reaches 7999. Therefore, when the cycle count reaches a value 7999, there is a roll over of the seconds count. If the sum of the cycle timer and XTO, the transmit offset value, results in a timestamp cycle count greater than 7999, the the seconds count field is incremented by 1 and the resulting cycle count value is (cycle count – 8000).

If invalid values are entered into the cycle offset or cycle count fields of XTO, the transmit offset, then they are truncated to the maximum values of 3071 and 7999, respectively. There is no other adjustment to the offset fields.

3.3.2 Timestamp Determination on Receive

Receive timestamping works similar to the transmit process. When a packet with a timestamp is received, that received timestamp is captured and added with the receive timestamp offset (RTO) to form a new timestamp. This new timestamp determines when the packet is released to the application. If receive aging is activated, then the received package is purged if its value is less than that of the current cycle timer. In other words, its display window has expired.

NOTE:

There is no checking performed on the values of the receive timestamp offset register (register E4h and E8h). It is possible for a user to insert an invalid value in these registers (see the *Timestamp Cycle Timer* and *Timestamp Cycle Count* sections). Therefore, it is up to the host to **not** put invalid values into these registers.

3.4 Asynchronous Transmit (Host Bus to TSB12LV41A)

There are two basic formats for data to be transmitted and received. The first is for quadlet packets and the second is for block packets. For transmits, the FIFO address indicates the beginning, middle, and end of a packet. For receives, the data length, which is found in the header of the packet, determines the number of bytes in a block packet.

3.4.1 Quadlet Transmit

The quadlet-transmit format is shown in Figure 3–22 and is described in Table 3–7. The first quadlet contains packet control information. The second and third quadlets contain the 64-bit, quadlet-aligned address. The fourth quadlet is data and is used only for write requests and read responses. For read requests and write responses, the quadlet data field is omitted. When transmitting, the TSB12LV41A uses information in the header quadlets to form the IEEE 1394 headers for transmit.

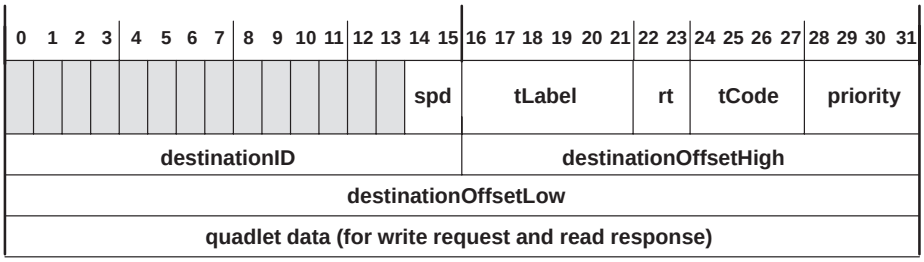


Figure 3–22. Quadlet-Transmit Format

Table 3–7. Quadlet-Transmit Format Functions

FIELD NAME	DESCRIPTION
spd	The spd field indicates the speed at which the current packet is to be sent. 00 = 100 Mbits/s, 01 = 200 Mbits/s, and 10 = 400 Mbits/s, and 11 is undefined for this implementation.
tLabel	The tLabel field is the transaction label, which is a unique tag for each outstanding transaction between two nodes. This field is used to pair up a response packet with its corresponding request packet.
rt	The rt field is the retry code for the current packet: 00 = new, 01 = retry_X, 10 = retryA, and 11 = retryB.
tCode	The tCode field is the transaction code for the current packet (see Table 6–10 of IEEE-1394 standard).
priority	The priority field contains the priority level for the current packet. For cable implementation, the value of the bits must be zero (for backplane implementation, see clause 5.4.1.3 and 5.4.2.1 of the IEEE-1394 standard).
destinationID	The destinationID field is the concatenation of the 10-bit bus number and the 6-bit node number that forms the destination node address of the current packet.
destination OffsetHigh, destination OffsetLow	The concatenation of these two fields addresses a quadlet in the destination node address space. This address must be quadlet aligned (modulo 4).
quadlet data	For write requests and read responses, the quadlet data field holds the data to be transferred. For write responses and read requests, this field is not used and should not be written into the FIFO.

3.4.2 Block Transmit

The block-transmit format is shown in Figure 3–23 and is described in Table 3–8. The first quadlet contains packet-control information. The second and third quadlets contain the 64-bit address. The first 16 bits of the fourth quadlet contains the dataLength field. This is the number of bytes of data in the packet. The remaining 16 bits represent the extended_tCode field (see Table 6–11 of the IEEE-1394 standard for more information on extended_tCodes). The block data, if any, follows the extended_tCode.

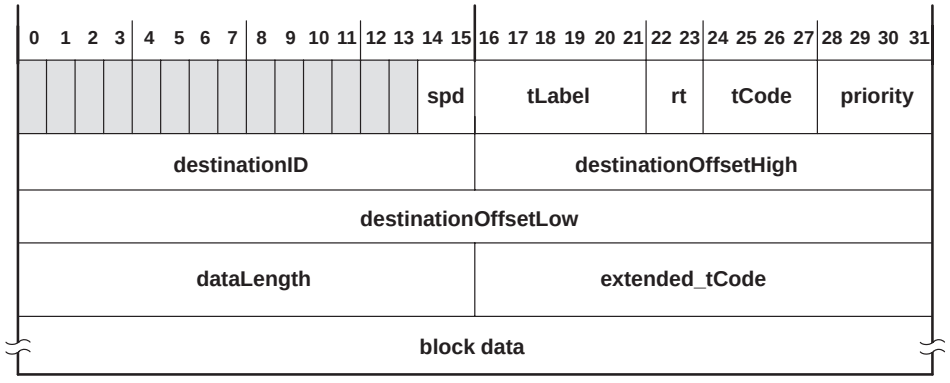


Figure 3–23. Block-Transmit Format

Table 3–8. Block-Transmit Format Functions

FIELD NAME	DESCRIPTION
spd	The spd field indicates the speed at which the current packet is to be sent. 00 = 100 Mbits/s, 01 = 200 Mbits/s, and 10 = 400 Mbits/s, and 11 is undefined for this implementation.
tLabel	The tLabel field is the transaction label, which is a unique tag for each outstanding transaction between two nodes. This field is used to pair up a response packet with its corresponding request packet.
rt	The rt field is the retry code for the current packet: 00 = new, 01 = retry_X, 10 = retryA, and 11 = retryB.
tCode	tCode is the transaction code for the current packet (see Table 6–10 of IEEE-1394 standard).
priority	The priority level for the current packet. For cable implementation, the value of the bits must be zero. For backplane implementation, see clause 5.4.1.3 and 5.4.2.1 of the IEEE-1394 standard.
destinationID	The destinationID field is the concatenation of the 10-bit bus number and the 6-bit node number that forms the node address to which the current packet is being sent.
destination OffsetHigh, destination OffsetLow	The concatenation of the destination OffsetHigh and the destination OffsetLow fields addresses a quadlet in the destination node address space. This address must be quadlet aligned (modulo 4). The upper four bits of the destination OffsetHigh field are used as the response code for lock-response packets and the remaining bits are reserved.
dataLength	The dataLength filed contains the number of bytes of data to be transmitted in the packet.
extended_tCode	The block extended_tCode to be performed on the data in the current packet (see Table 6–11 of the IEEE-1394 standard).
block data	The block data field contains the data to be sent. If dataLength is 0, no data should be written into the FIFO for this field. Regardless of the destination or source alignment of the data, the first byte of the block must appear in byte 0 of the first quadlet.

3.4.3 Quadlet Receive

The quadlet-receive format is shown in Figure 3–24 and is described in Table 3–9. The first 16 bits of the first quadlet contain the destination node and bus ID, and the remaining 16 bits contain packet-control information. The first 16 bits of the second quadlet contain the node and bus ID of the source, and the remaining 16 bits of the second and third quadlets contain the 48-bit, quadlet-aligned destination offset address. The fourth quadlet contains data that is used by write requests and read responses. For read requests and write responses, the quadlet data field is omitted. The last quadlet contains the packet trailer (packet-reception status that is added by the TSB12LV41A). For packets received to the bulky data FIFO, the packet trailer is included as the first quadlet.

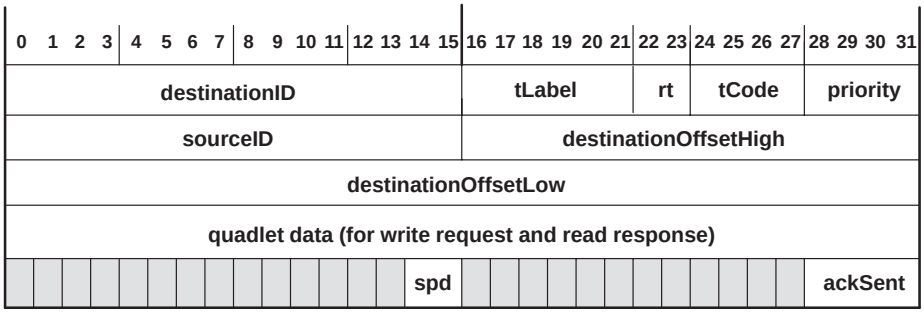


Figure 3–24. Quadlet-Receive Format for Control FIFO

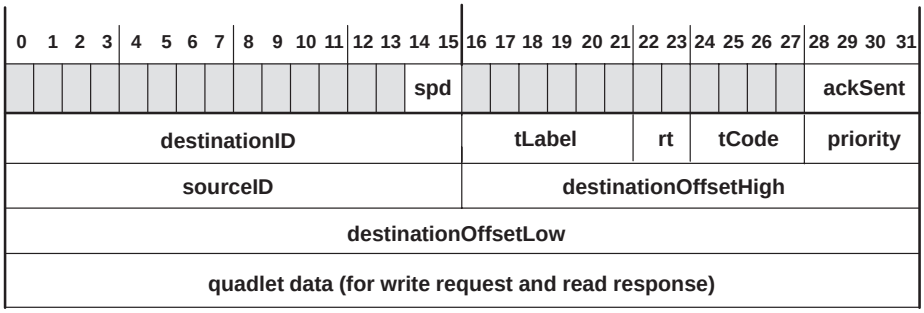


Figure 3–25. Quadlet-Receive Format for Bulky Data FIFO

Table 3–9. Quadlet-Receive Format Functions

FIELD NAME	DESCRIPTION
destinationID	The destinationID field contains the concatenation of the 10-bit bus number and the 6-bit node number that forms the node address to which the current packet is being sent.
tLabel	The tLabel field is the transaction label, which is a unique tag for each outstanding transaction between two nodes. This field is used to pair up a response packet with its corresponding request packet.
rt	The rt field is the retry code for the current packet: 00 = new, 01 = retry_X, 10 = retryA, and 11 = retryB.
tCode	The tCode field is the transaction code for the current packet (see Table 6–10 of the IEEE-1394 standard).
priority	The priority field contains the priority level for the current packet. For cable implementation, the value of the bits must be zero (for backplane implementation, see clause 5.4.1.3 and 5.4.2.1 of the IEEE-1394 standard).
sourceID	The sourceID field contains the node ID of the sender of the current packet.
destination OffsetHigh, destination OffsetLow	The concatenation of the destination OffsetHigh and the destination OffsetLow fields addresses a quadlet in the destination nodes address space. This address must be quadlet aligned (modulo 4). (The upper four bits of the destination OffsetHigh field are used as the response code for lock-response packets, and the remaining bits are reserved.)
quadlet data	For write requests and read responses, the quadlet data field holds the transferred data. For write responses and read requests, this field is not present.
spd	The spd field indicates the speed at which the current packet was sent. 00 = 100 Mbits/s, 01 = 200 Mbits/s, 10 = 400 Mbits/s, and 11 is undefined for this implementation.
ackSent	The ackSent field holds the acknowledge sent by the receiver for the current packet (see Table 6–13 in the IEEE 1394–1995 standard).

3.4.4 Block Receive

The block-receive format is shown in Figure 3–26 and is described in Table 3–10. The first 16 bits of the first quadlet contain the node and bus ID of the destination node, and the last 16 bits contain packet-control information. The first 16 bits of the second quadlet contain the node and bus ID of the source node, and the last 16 bits of the second quadlet and all of the third quadlet contain the 48-bit, quadlet-aligned destination offset address. All remaining quadlets, except for the last one, contain data that is used only for write requests and read responses. For block read requests and block write responses, the data field is omitted. The last quadlet contains the packet trailer. For packets received to the bulky asynchronous FIFOs, the packet trailer is included as the first quadlet.

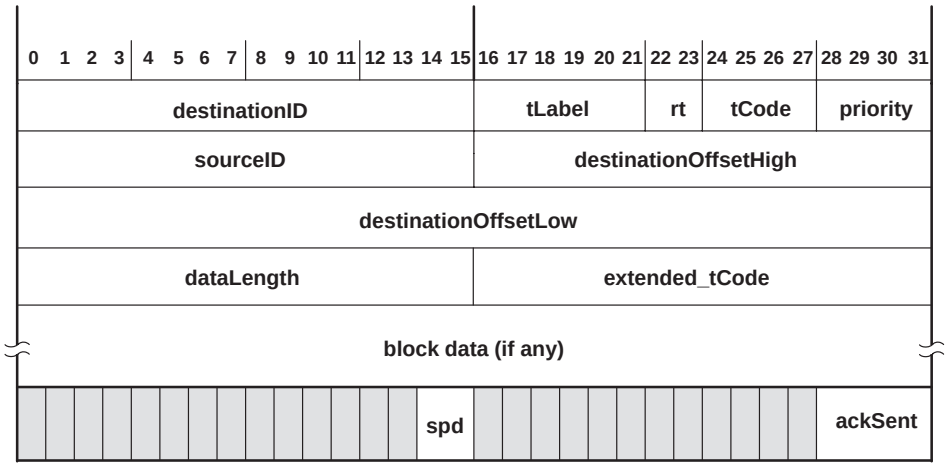


Figure 3–26. Block-Receive Format for Control FIFO

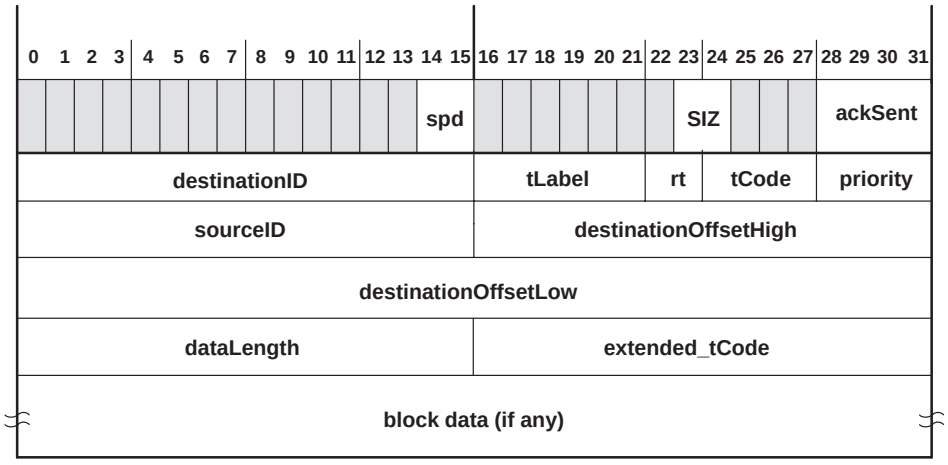


Figure 3–27. Block-Receive Format for Bulky Data FIFO

Table 3–10. Block-Receive Format Functions

FIELD NAME	DESCRIPTION
destinationID	The destinationID field is the concatenation of the 10-bit bus number and the 6-bit node number that forms the node address to which the current packet is being sent.
tLabel	The tLabel field is the transaction label, which is a unique tag for each outstanding transaction between two nodes. This field is used to pair up a response packet with its corresponding request packet.
rt	The rt field contains the retry code for the current packet: 00 = new, 01 = retry_X, 10 = retryA, and 11 = retryB.
tCode	The tCode field is the transaction code for the current packet (see Table 6–10 of the IEEE-1394 standard).
priority	The priority field contains the priority level for the current packet. For cable implementation, the value of the bits must be zero (for backplane implementation, see clause 5.4.1.3 and 5.4.2.1 of the IEEE-1394 standard).
sourceID	The sourceID field contains the node ID of the sender of the current packet.
destination OffsetHigh, destination OffsetLow	The concatenation of the destination OffsetHigh and the destination OffsetLow fields addresses a quadlet in the destination nodes address space. This address must be quadlet aligned (modulo 4). The upper four bits of the destination OffsetHigh field are used as the response code for lock-response packets and the remaining bits are reserved.
dataLength	For write request, read responses, and locks, the dataLength field indicates the number of bytes being transferred. For read requests, the dataLength field indicates the number of bytes of data to be read. A write-response packet does not use this field. Note that the number of bytes does not include the header, only the bytes of block data.
extended_tCode	The extended_tCode field contains the block extended_tCode to be performed on the data in the current packet (see Table 6–11 of the IEEE-1394 standard).
block data	The block data field contains any data being transferred for the current packet. Regardless of the destination address or memory alignment, the first byte of the data appears in byte 0 of the first quadlet of this field. The last quadlet of the field is padded with zeros out to four bytes, if necessary.
spd	The spd field indicates the speed at which the current packet was sent. 00 = 100 Mbits/s, 01 = 200 Mbits/s, 10 = 400 Mbits/s, and 11 is undefined for this implementation.
ackSent	The ackSent field holds the acknowledge sent by the receiver for the current packet.
SIZ	SIZ indicates the number of zero-filled bytes in the last quadlet of the packet.

3.5 Isochronous Transmit and Receive (Host Bus to ’12LV41A) Data Formats

3.5.1 Isochronous Transmit

The format of the isochronous-transmit packet is shown in Figure 3–28 and is described in Table 3–11. The data for each channel must be presented to the isochronous-transmit FIFO interface in this format in the order that packets are to be sent. The transmitter sends any packets available at the isochronous-transmit interface immediately following reception or transmission of the cycle-start message. The first quadlet gives the TSB12LV41A information to build the 1394 isochronous header for transmit.

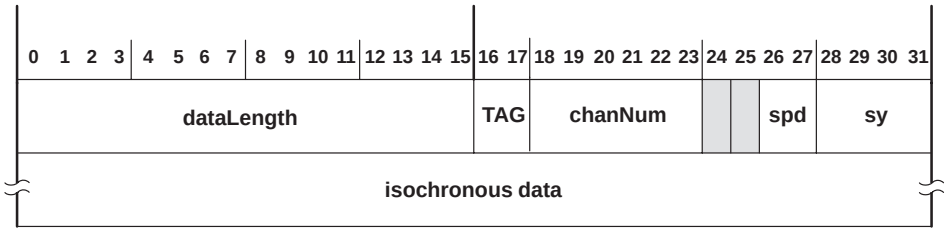


Figure 3–28. Isochronous-Transmit Format

Table 3–11. Isochronous-Transmit Functions

FIELD NAME	DESCRIPTION
dataLength	The dataLength field indicates the number of bytes in the current packet
TAG	The TAG field indicates the format of data carried by the isochronous packet (00 = formatted, 01 – 11 are reserved).
chanNum	The chanNum field carries the channel number with which the current data is associated.
spd	The spd field contains the speed at which to send the current packet.
sy	The sy field carries the transaction layer-specific synchronization bits.
isochronous data	The isochronous data field contains the data to be sent with the current packet. The first byte of data must appear in byte 0 of the first quadlet of this field. If the last quadlet does not contain four bytes of data, the unused bytes should be padded with zeros.

3.5.2 Isochronous Receive

The format of the iscohronous-receive data is shown in Figure 3–29 and is described in Table 3–12. The data length, which is found in the header of the packet, determines the number of bytes in an isochronous packet. The packet trailer is included as the first quadlet in the bulky data FIFO.

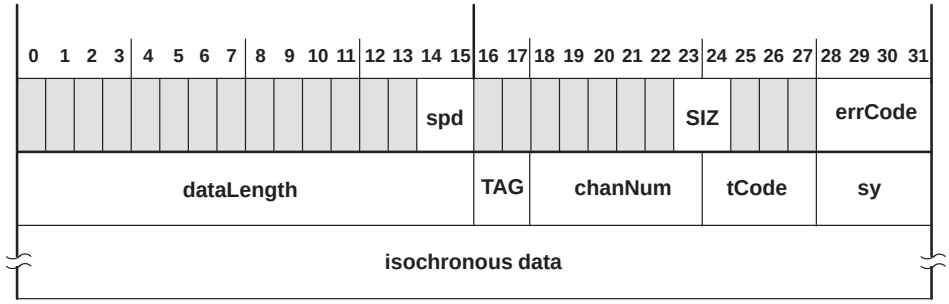


Figure 3–29. Isochronous-Receive Format for Bulky Data FIFO

Table 3–12. Isochronous-Receive Functions

FIELD NAME	DESCRIPTION
dataLength	The dataLength field indicates the number of bytes in the current packet.
TAG	The TAG field indicates the format of data carried by isochronous packet (00 = formatted, 01 – 11 are reserved).
chanNum	The chanNum field contains the channel number with which this data is associated.
tCode	The tCode field carries the transaction code for the current packet (tCode = Ah).
sy	The sy field carries the transaction layer-specific synchronization bits.
isochronous data	The isochronous data field has the data to be sent with the current packet. The first byte of data must appear in byte 0 of the first quadlet of this field. The last quadlet should be padded with zeros.
spd	The spd field indicates the speed at which the current packet was sent.
errCode	The errCode field indicates whether the current packet has been received correctly. The possibilities are Complete (0001b) and DataErr (1101b). DataErr is returned when either the data CRC check fails or there is a mismatch between the actual payload and the datalength field in the header.
SIZ	SIZ indicates the number of zero-filled bytes in the last quadlet of the packet.

3.6 Snoop

The format of the snoop data is shown in Figure 3–30 and is described in Table 3–13. The receiver module can be directed to receive any and all packets that pass by on the serial bus. In this mode, the receiver presents the data received to the receive-FIFO interface.

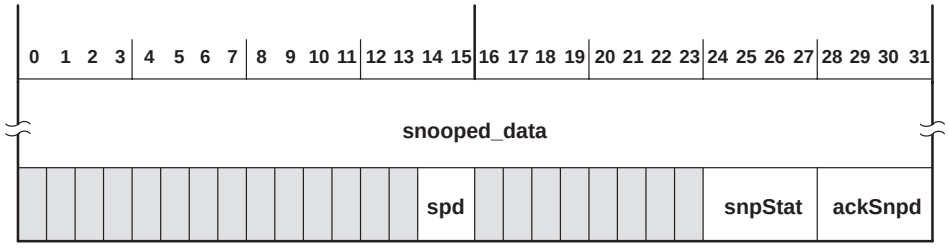


Figure 3–30. Snoop Format

Table 3–13. Snoop Functions

FIELD NAME	DESCRIPTION
snooped_data	The snooped_data field contains the entire packet received or as much as could be received.
spd	The spd field carries the speed at which the current packet was sent.
snpStat	The snpStat field indicates whether the entire packet snooped was received correctly. A value equal to the complete acknowledge code indicates complete reception. A busyA or busyB acknowledge code indicates incomplete reception.
ackSnpd	The ackSnpd field indicates the acknowledge seen on the bus after the packet is received.

3.7 Cyclemark

The format of the Cyclemark data is shown in Figure 3–31 and is described in Table 3–14. The receiver module inserts a single quadlet to mark the end of an isochronous cycle. The quadlet is inserted into the receive-FIFO.

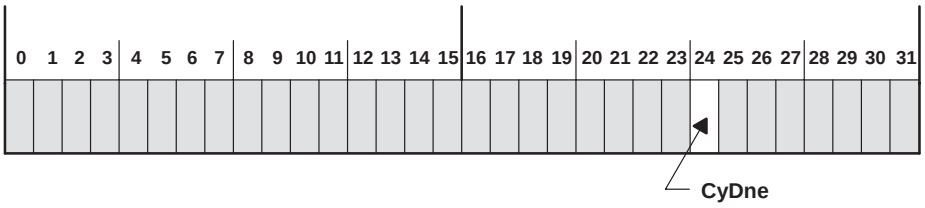


Figure 3–31. Cyclemark Format

Table 3–14. Cyclemark Functions

FIELD NAME	DESCRIPTION
CyDne	The CyDne field indicates the end of an isochronous cycle.

3.8 Phy Configuration

The format of the Phy configuration packet is shown in Figure 3–32 and is described in Table 3–15. The Phy configuration packet transmit contains two quadlets. The first quadlet tells the TSB12LV41A that this quadlet is the Phy configuration packet. The Eh is then replaced with 0h before the packet is transmitted to the Phy interface.

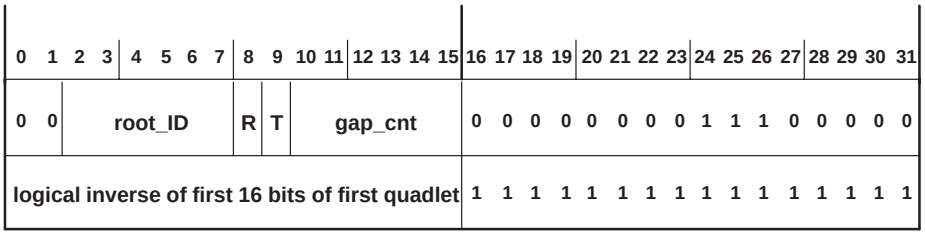


Figure 3–32. Phy Configuration Format

Table 3–15. Phy Configuration Functions

FIELD NAME	DESCRIPTION
00	The 00 field is the Phy configuration packet identifier.
root_ID	The root_ID field is the physical_ID of the node to have its force_root bit set (only meaningful when R is set).
R†	When R is set, the force-root bit of the node identified in root_ID is set and the force_root bit of all other nodes are cleared. When R is cleared, root_ID is ignored.
T†	When T is set, the PHY_CONFIGURATION.gap_count field of all the nodes is set to the value in the gap_cnt field.
gap_cnt	The gap_cnt field contains the new value for PHY_CONFIGURATION.gap_count for all nodes. This value goes into effect immediately upon receipt and remains valid after the next bus reset. After the second reset, gap_cnt is set to 63h unless a new Phy configuration packet is received.

† A Phy configuration packet with R = 0 and T = 0 is reserved and is ignored when received.

3.9 Receive Self-ID Packet

The format of the receive Self-ID packet is shown in Figure 3–33 and is described in Table 3–16. When SIDM0 (bit 21 register 148h) is set, the receive Self-ID packet is stored in broadcast write receive FIFO. Otherwise, the Self-IDs are collected in the bulky asynchronous receive FIFO.

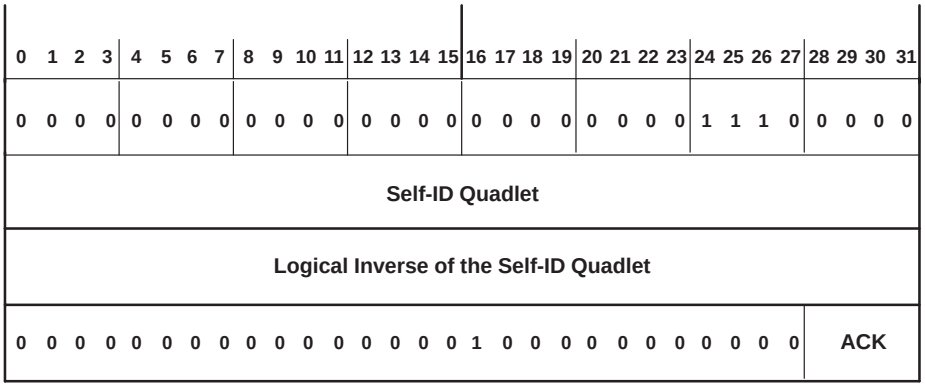


Figure 3–33. Receive Self-ID Format for Broadcast Write Receive FIFO

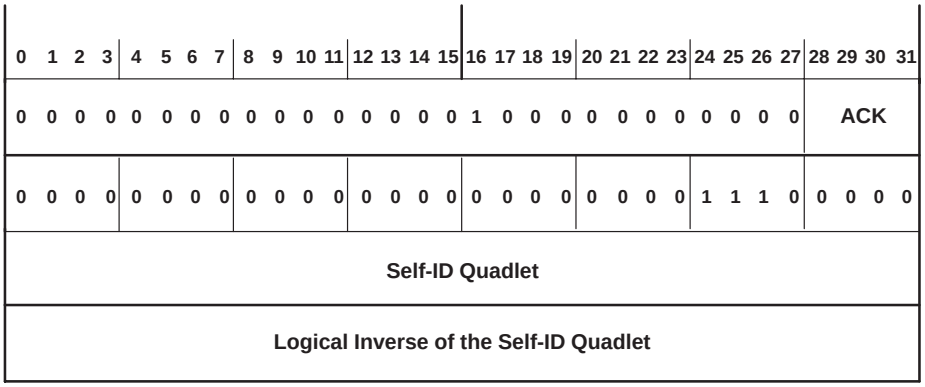


Figure 3–34. Receive Self-ID Format for Bulky Asynchronous Receive FIFO

Table 3–16. Receive Self-ID Function

FIELD NAME	DESCRIPTION
ACK	When the ACK field is set (0001b), the data in the Self-ID packet is correct. When the ACK field is set (1101b), an error occurred during transmission.

When there is only one node (i.e., one Phy/LLC pair) on the bus, following a bus reset, the FIFO contains 000_00E0h and the acknowledge quadlet only.

When there are three nodes on the bus, each with a Phy having three or less ports, following a bus reset, the FIFO of any one of the LLCs is shown in Table 3–17 for BWRX and Table 3–18 for BARX FIFO. The ACK is not written into the broadcast write receive FIFO (BWRX).

Table 3–17. Broadcast Write Receive FIFO Contents With Three Nodes on a Bus

FIFO CONTENTS	DESCRIPTION
0000_00E0h	Header for Self-ID
Self–ID1	Self_ID for Phy #1
Self–ID1 inverse	Self_ID for Phy #1 inverted
Self–ID2	Self_ID for Phy #2
Self–ID2 inverse	Self_ID for Phy #2 inverted

The first quadlet in a Self-ID packet is 0000_00E0h. The second quadlet in the Self-ID packet is described in Figure 3–35, Figure 3–36, and Table 3–19. The third quadlet is the inverse of the Self-ID quadlet.

Table 3–18. Bulky Data Asynchronous Receive FIFO (BARX FIFO) Contents

FIFO CONTENTS	DESCRIPTION
0000_800(ACK)h	Trailing acknowledge
0000_00E0h	Header for Self-ID
Self–ID1	Self_ID for Phy #1
Self–ID1 inverse	Self_ID for Phy #1 inverted
Self–ID2	Self_ID for Phy #2
Self–ID2 inverse	Self_ID for Phy #2 inverted

The format for self-IDs, stored in the BARX FIFO, is similar to the broadcast write receive FIFO, ACK codes are included as the first quadlet in the BARX FIFO.

The cable Phy sends one to four Self-ID packets at the base rate (100 Mbits/s) during the Self-ID phase of arbitration. The number of Self-ID packets sent depends on the number of ports. Figure 3–35 and Figure 3–36 show the formats of the cable Phy Self-ID packets.

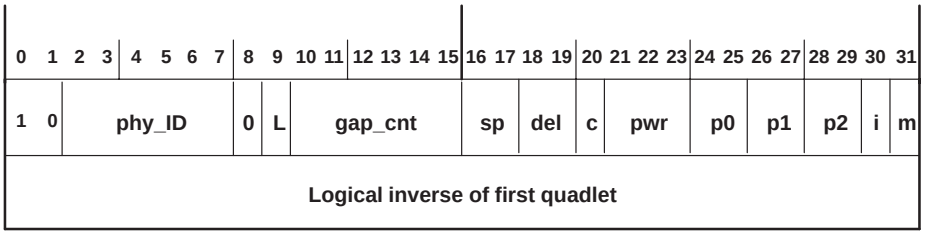
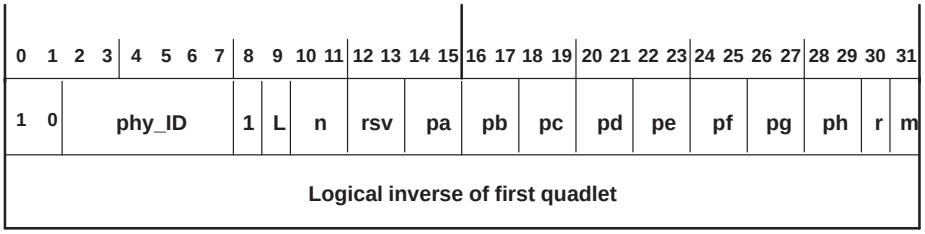


Figure 3–35. Phy Self-ID Packet #0 Format



PACKET #	n†	pa	pb	pc	pd	pe	pf	pg	ph
1	0	p3	p4	p5	p6	p7	p8	p9	p10
2	1	p11	p12	p13	p14	p15	p16	p17	p18
3	2	p19	p20	p21	p22	p23	p24	p25	p26

† For n = 3 – 7, fields pa through ph are reserved.

Figure 3–36. Phy Self-ID Packet #1, Packet #2, and Packet #3 Format

Table 3–19. Phy Self-ID Functions

FIELD NAME	DESCRIPTION	
10	The 10 field is the Self-ID packet identifier.	
c	When c is set and the link-active flag is set, this field indicates that the current node is a contender for the bus or isochronous resource manager.	
del	The del field contains the worst-case repeater-data delay time. The code is:	
	00	≤ 144 ns ≈ (14 / Base_Rate)
	01 – 11	Reserved
gap_cnt	The gap_cnt field contains the current value for the current node PHY_CONFIGURATION.gap_count field.	
i	When set, the i field indicates that the current node initiated the current bus reset (i.e., it started sending a bus reset signal before it received one†).If this function is not implemented, i is returned as 0.	
L	When L is set, the current node has an active LLC and transaction layer.	
m	When set, the m field indicates that another Self-ID packet for the current node immediately follows (i.e. when m is set and the next Self-ID packet received has a different phy_ID, then a Self-ID packet was lost).	

† There is no way to ensure that exactly one node has this bit set. More than one node can be requesting a bus reset at the same time.

Table 3–19. Phy Self-ID Functions (Continued)

FIELD NAME	DESCRIPTION	
n	The n field is the extended Self-ID packet sequence number. The code is:	
	0	Self-ID packet 1
	1	Self-ID packet 2
	2	Self-ID packet 3
phy_ID	The phy_ID field is the physical node identifier of the sender of the current packet.	
p0 – p26	The p0 – P26 field indicates the port status. The code is:	
	00	Not present on the current Phy
	01	Not connected to any other Phy
	10	Connected to the parent node
	11	Connected to the child node
pwr	The pwr field contains the bits that indicate the power consumption and source characteristics. The code is:	
	000	The node does not need power and does not repeat power.
	001	The node is self powered and provides a minimum of 15 W to the bus.
	010	The node is self powered and provides a minimum of 30 W to the bus.
	011	The node is self powered and provides a minimum of 45 W to the bus.
	100	The node can be powered from the bus and is using up to 1 W.
	101	The node is powered from the bus and is using up to 1 W. An additional 2 W is needed to enable the LLC and higher layers.‡
	110	The node is powered from the bus and is using up to 1 W. An additional 5 W is needed to enable the LLC and higher layers.‡
	111	The node is powered from the bus and is using up to 1 W. An additional 9 W is needed to enable the LLC and higher layers.‡
r	Reserved and set to all zeros.	
rsv	Reserved and set to all zeros.	
sp	The sp field contains the Phy speed capability. The code is:	
	00	98.304 Mbits/s
	01	98.304 Mbits/s and 196.608 Mbits/s
	10	98.304 Mbits/s 196.608 Mbits/s, and 393.216 Mbits/s
	11	Reserved

† There is no way to ensure that exactly one node has this bit set. More than one node can be requesting a bus reset at the same time.

‡ The LLC and higher layers are enabled by the Link-On Phy packet.

4 External Interfaces

4.1 Bulky Data Interface (BDIF)

4.1.1 Introduction

The bulky data interface or BDIF is a pair of ports supported by the TSB12LV41A Link Layer controller. The BDIF is the physical medium by which autonomous streams of different types are piped to an application that uses the TSB12LV41A. A system diagram is shown in Figure 4–1:

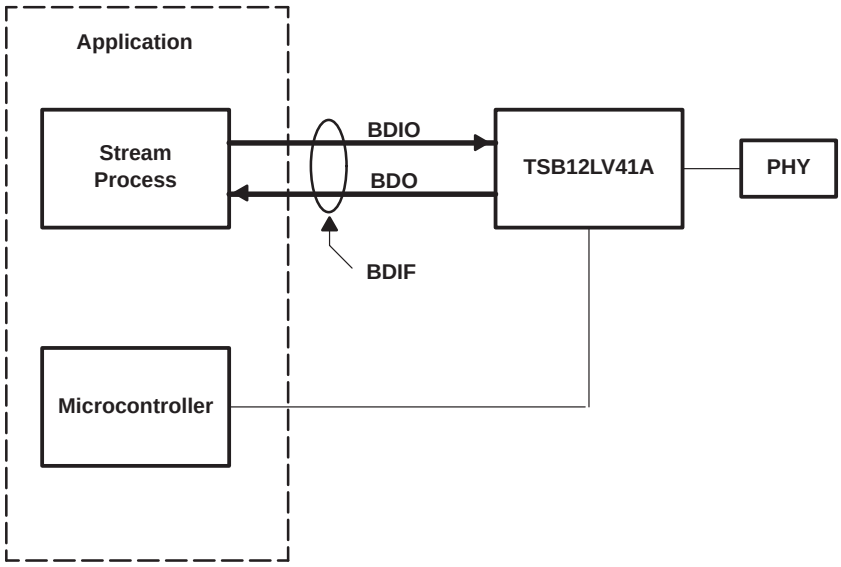


Figure 4–1. TSB12LV41A System Application Diagram

Since the BDIF has two ports, data can be full duplex. One port is bidirectional and the other is output only. Each port has its own independent clock, control signals, and modes of operation. The ports can operate in asynchronous clock domains.

The BDIF handles three stream types:

- Asynchronous
- Isochronous
- MPEG2 (DVB)/DSS (a special isochronous type)

These stream types are identified by a format bus bound to the port. The encodings on the format bus also frame packets within the stream. BDIF is the Format bus for BDIO and BDOF is the format bus for BDO

Even though the two ports (BDIO and BDO) have some control signal and clock dependencies, we first look at them separately.

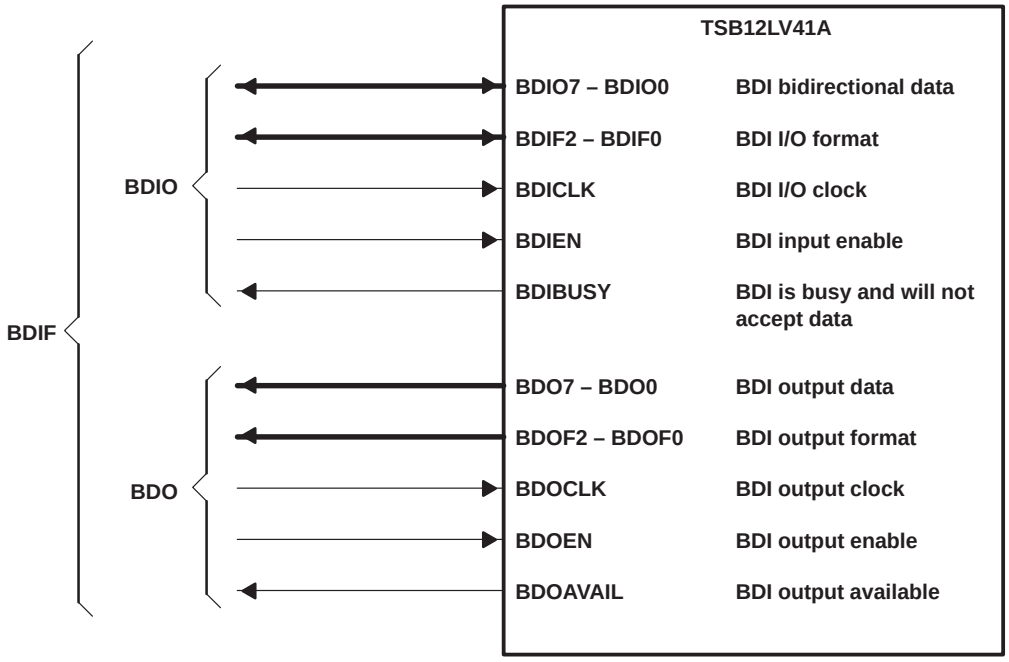


Figure 4–2. Bulky Data Interface

The signals of the bidirectional port (BDIO) have the following characteristics.

BDIO Signal Name	Driver Type	Description	
BDIO[7:0]	I/O	Bidirectional BDIF port (can be configured for input only). BDIO7 = MSB and BDIO0 = LSB	
BDIF[2:0]	I/O	Bidirectional BDIF Format (can be configured for input only.	
		000	Reserved
		001	A byte of an MPEG2 (DVB) (or DSS) cell
		010	A byte of an unformatted isochronous packet
		011	A byte of an asynchronous packet
		100	Idle
		101	First byte of an MPEG2 (DVB) (or DSS) cell
		110	Last byte of an unformatted Isochronous packet
		111	Last byte of an asynchronous packet
BDICLK	I	BDIO data input clock	
BDIEN	I	BDIF enable: Qualifies data for writes, data on BDIO, Format on BDIF. Enable data for reads , data on BDIO, Format on BDIF.	
BDIBUSY	O	Signals busy condition on BDIO for writes. This signal goes high when the FIFO being written to is full. When BDIBUSY is high, writing to the full FIFO is disabled.	

Table 4–1. BDIO Bidirectional Port Signals

The signals of the unidirectional output only port (BDO) have the following characteristics.

BDIO Signal Name	Driver Type	Description	
BDO[7:0]	O	Unidirectional BDO port	
BDOF[2:0]	O	Undirectional BDO Format. BDO7 = MSB and BDO0 = LSB	
		000	Reserved
		001	A byte of an MPEG2 (DVB) (or DSS) cell
		010	A byte of an unformatted Isochronous packet
		011	A byte of an Asynchronous packet
		100	Idle
		101	First byte of an MPEG2 (DVB) (or DSS) cell
		110	Last byte of an unformatted Isochronous packet
		111	Last byte of an Asynchronous packet
BDOCLK	I	BDIO or BDO data output clock	
BDOEN	I	BDO enable: Qualifies data on BDO for reads Read/Write* control for BDIO when it is bidirectional	
BDOAVAIL	O	Signals data is available on BDO and also BDIO for reads.	

Table 4–2. BDO Unidirectional Port Signals

NOTE:

The format codes 110 and 111 indicate the last byte of an isochronous or asynchronous packet. This is needed when the data in the FIFO is not a multiple of four bytes and padding zero bits are added to the data prior to transmit. Padding is necessary because all packets on 1394 must be on quadlet boundaries (multiples of 4 bytes). See Section 3.1.4 for more information on byte padding.

The BDIF is programmed by writes to the BDIF control register. This register is located at offset D8h in the TSB12LV41A microcontroller address space. The register format and bit definitions are shown in Figure 4–3.

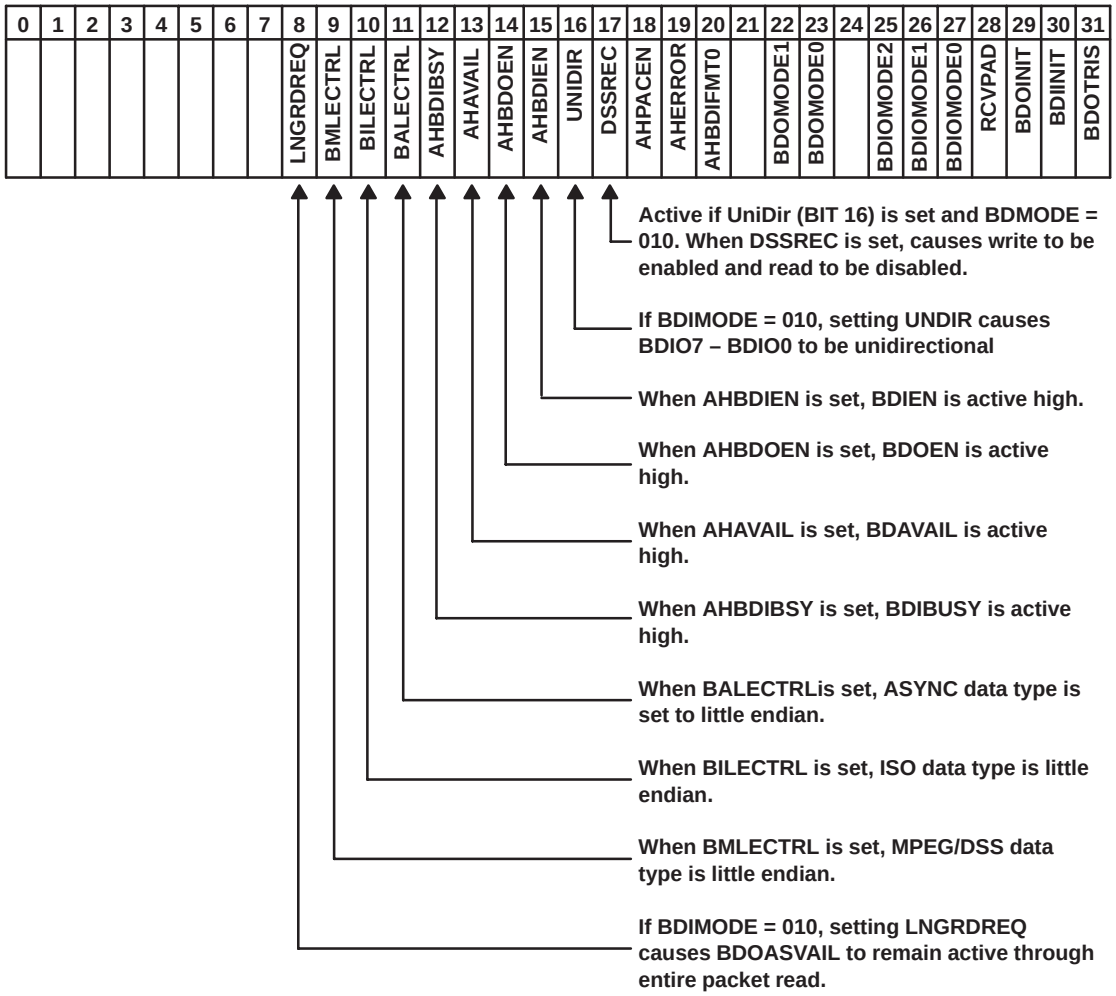


Figure 4–3. BDIF Control Register

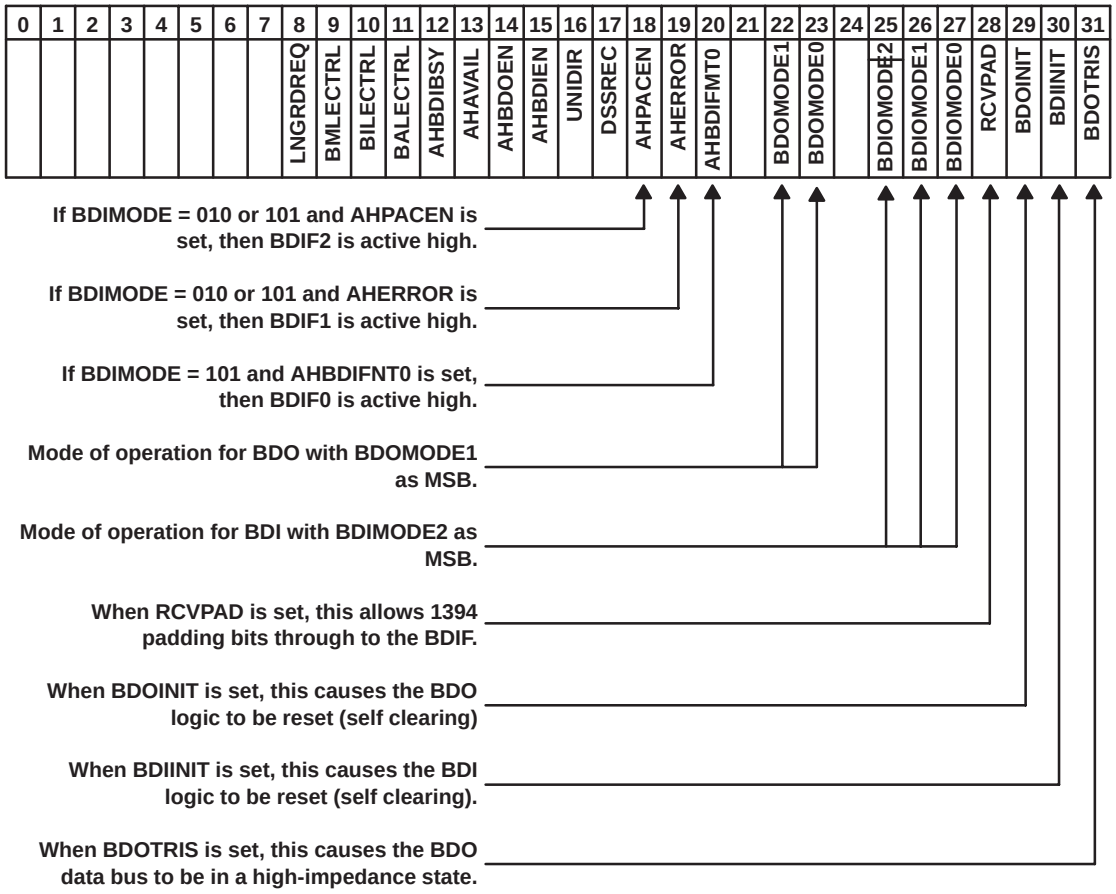


Figure 4–3. BDIF Control Register (Continued)

4.1.2 Modes of the Bulky Data Interface (BDIF)

The BDIF has four valid modes of operation. These modes are selected using the BDIOMODE and BDOMODE fields of the BDIF control register. The following table shows the basic features of each mode.

BDIF MODE	A	B	C	D
BDIOMODE	000	000	101	001
BDOMODE	00	01	11	00
Data input	BDIO	BDIO	BDIO Async	BDIO
Data output	BDO	BDO	BDO Async	BDIO
Data bus	2	2	2	1
Duplex	Full	Full	Full	Half
Data input clock (MHz)	20.25	20.25	NClk	20.25
Data output clock (MHz)	20.25	20.25	NClk	20.25
Data throughput Mbyte/s (Max)	20 Write 20 Read	20 Write 20 Read	10 Write 10 Read	20.25
Control Signal Use				
BDIEN	X	X	X	X
BDIBUSY	X	X		X
BDOEN	X		X	X
BDOAVAIL	X	X	X	X

Table 4–3. MODES of the BDIF

Detailed descriptions of each mode are contained in the paragraphs that follow.

BDIF MODE: A	8 bit Parallel Input	8 bit Parallel Output
	BDIOMODE = 000	BDOMODE = 00

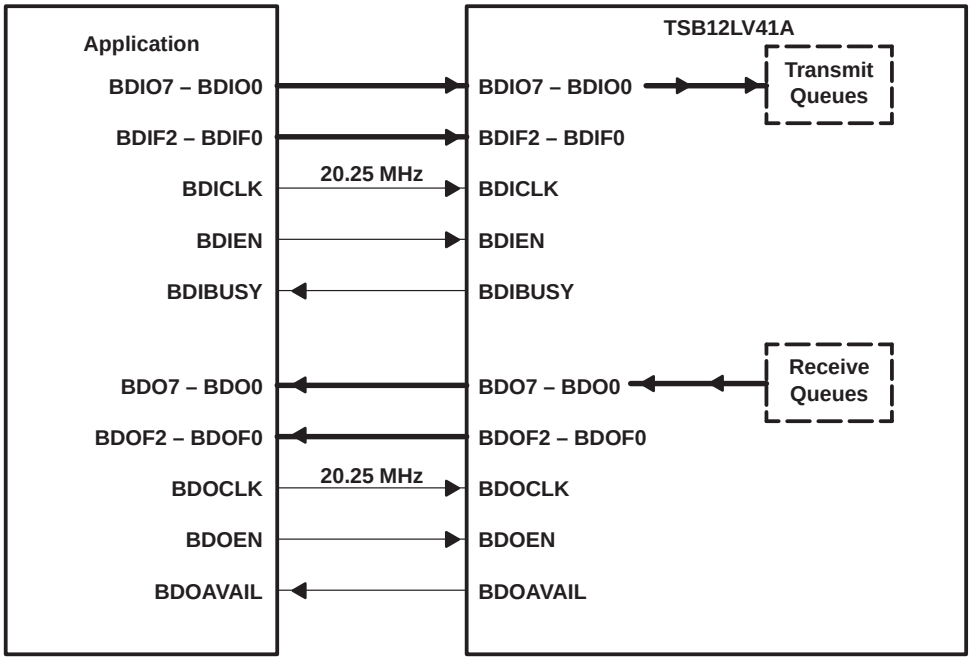


Figure 4–4. Bulky Data Interface Mode A Typical Application

In this mode, the BDIO bus is input only. The BDO bus is output only. There are 2 data busses that connect the BDIF with bulky data FIFOs. These buses go to the transmit and receive queues.

This mode is synchronous for both input and output. Both data input (BDICLK) and output (BDOCLK) clocks run at 20.25 MHz. The BDIF expects new data every clock period.

The BDIF operates in full duplex mode. In this mode, the BDIF can receive data at the BDIO port and transmit data from the BDO port simultaneously. With input/output clock speeds of 20.25 MHz, this allows a maximum throughput of 40.5 Mbyte/s.

BDIEN qualifies data on BDIO for writes. BDIBUSY signals a busy condition on BDIO for writes.

BDOEN qualifies data on BDO for reads. BDOAVAIL signals data is available on BDO for reads.

BDIF MODE: B	8 bit Parallel Input	8 bit Parallel Output with No Read Control
	BDIOMODE = 000	BDOMODE = 01

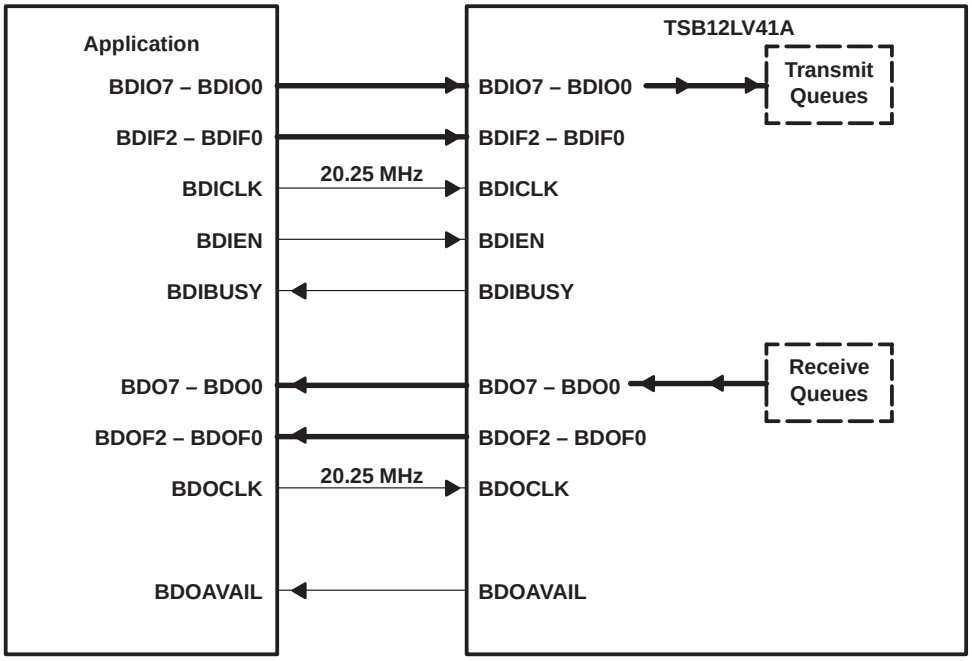


Figure 4–5. Bulky Data Interface Mode B Typical Application

The data input and output for this mode are similar to the BDIF MODE A mode. The BDIO is input only. BDO is output only. The clock speeds, data rates, and full duplex capability are the same as for BDIF MODE A. The BDIF expects new data every clock period.

The difference between this mode and BDIF MODE A is the absence of BDOEN, the bulky data output enable. Since BDIF MODE B does not use BDOEN, data is continuously output to the host whether or not it can accept the data. The main advantage of this mode is that no signal is required by the host to transmit. However, if the host FIFO is full, data may be lost.

BDIF MODE: C	8 bit Parallel Input (Asynchronous)	8 bit Parallel Output (Asynchronous)
	BDIOMODE = 101	BDOMODE = 11

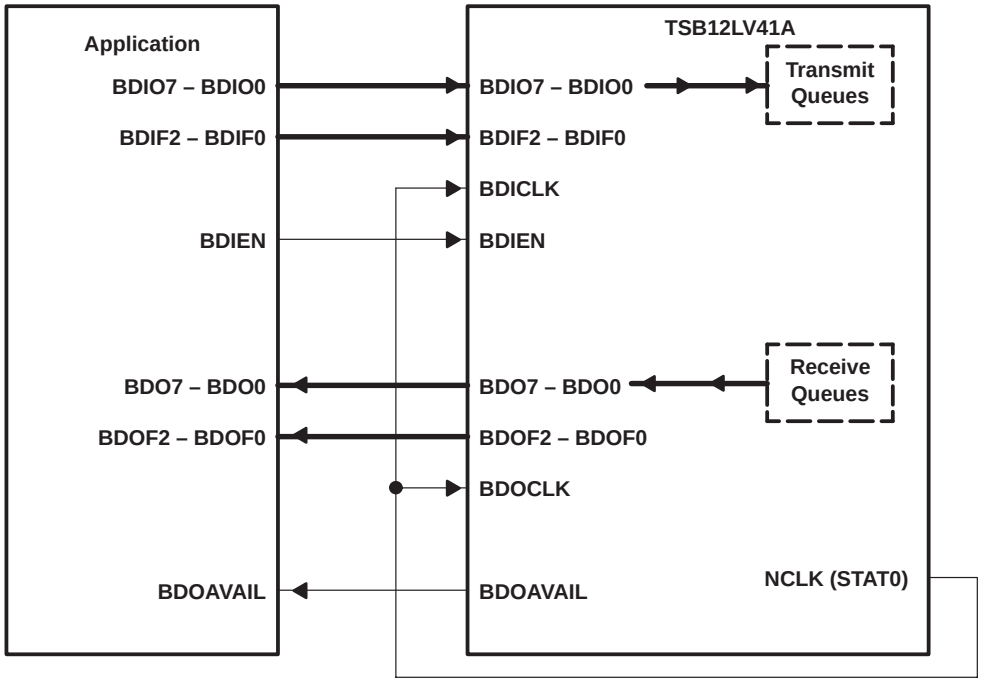


Figure 4–6. Bulky Data Interface Mode C Typical Application

The data input for this mode at the BDIO bus is asynchronous only. The BDIO is input only. The data output at BDO is also asynchronous. This mode provides 2 data buses, one for the transmit and one for the receive FIFO. This mode operates in full duplex.

Since both BDIO and BDO are asynchronous, an application typically does not provide BDICLK or BDOCLK. In this case, it is convenient to use NCLK (SYSCLK/2) to drive BDICLK and BDOCLK. NCLK is available from the STAT0 terminal (28) by programming bits 21 – 27 to 3Bh in register 30h.

This mode has a maximum data throughput capability of 10Mbyte/s for a write and 10Mbyte/s for a read.

There is no BDBUSY available in this mode. This means that there is no way for the TSB12LV41A to signal to the application that it is busy and can not accept any more data.

NOTE:
In this mode, the BDO port supports MPEG2 (DVB)/DSS format only.

BDIF MODE: D	8 bit Parallel Bidirectional	8 bit Parallel Bidirectional
	BDIOMODE = 001	BDOMODE = 00

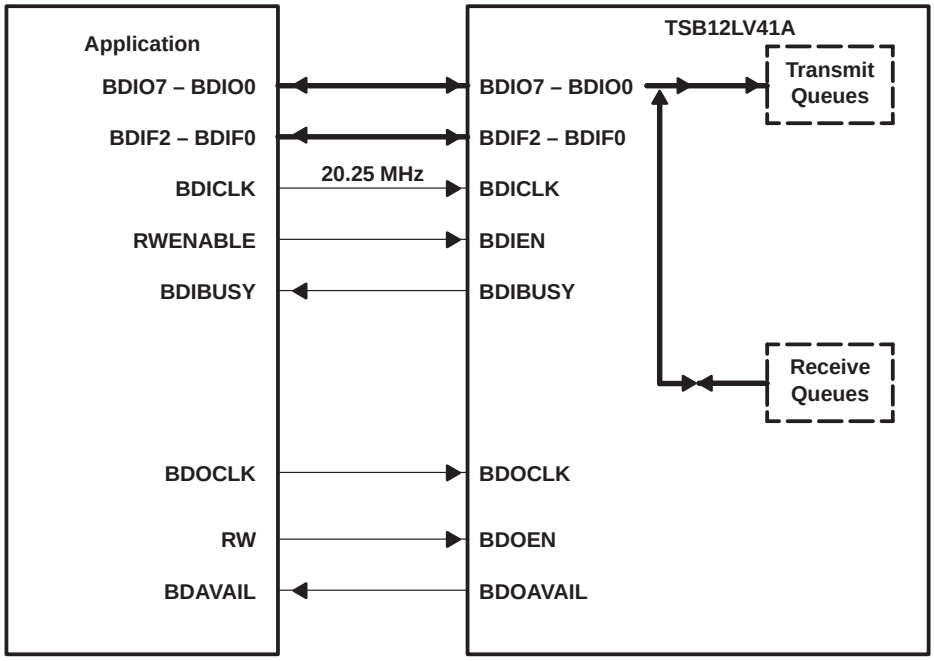


Figure 4–7. Bulky Data Interface Mode D Typical Application

This is a bidirectional mode. The data input and data output share the BDIO port. BDOEN is a Read/Write* control in bidirectional mode. The device in this mode operates in half duplex. The data input/data output clocks are both at 20.25MHz. The BDIF expects new data every clock cycle in this mode.

4.2 Bulky Data Interface Timing

4.2.1 Unidirectional and Asynchronous Modes

Writes and reads in the unidirectional modes (modes A and B) and the asynchronous mode (mode C) occur at separate ports. Reads occur at BDO[7–0] and writes occur at BDIO[7–0]. Each port has its own format bus and control signals. Please note that the asynchronous mode (modes C) supports only MPEG2(DVB)/DSS data.

4.2.1.1 Unidirectional Write Timing

In unidirectional modes (modes A and B), writes to the bulky data interface take place through the BDIO [7–0] data bus. As shown in Figure 4–8, when data is available to be written to the bulky data interface, the host activates BDIEN and simultaneously drives BDIO[7–0] and BDIF[2–0]. BDIEN allows the application to write data to the bulky data interface. If the TSB12LV41A FIFO being written to is full, the hardware activates BDIBUSY (on quadlet boundaries only) and does not accept any more data until BDIBUSY is deasserted. The format bus BDIF[2–0] signals the first byte of MPEG data, as well as other consecutive bytes, to the TSB12LV41A FIFOs. The format bus can also represent asynchronous and isochronous packets. Please reference the bulky data interface section 4.1.1 more detail. Please see Figure 4–9 for critical write timing in bulky data modes A and B.

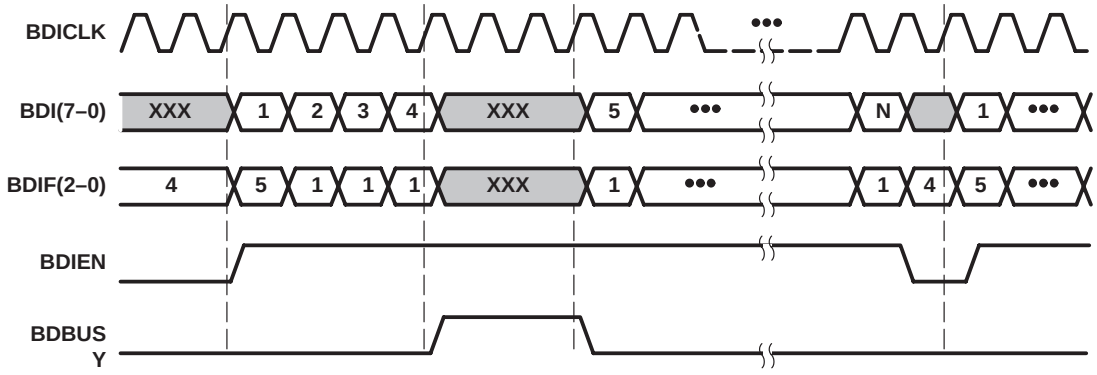
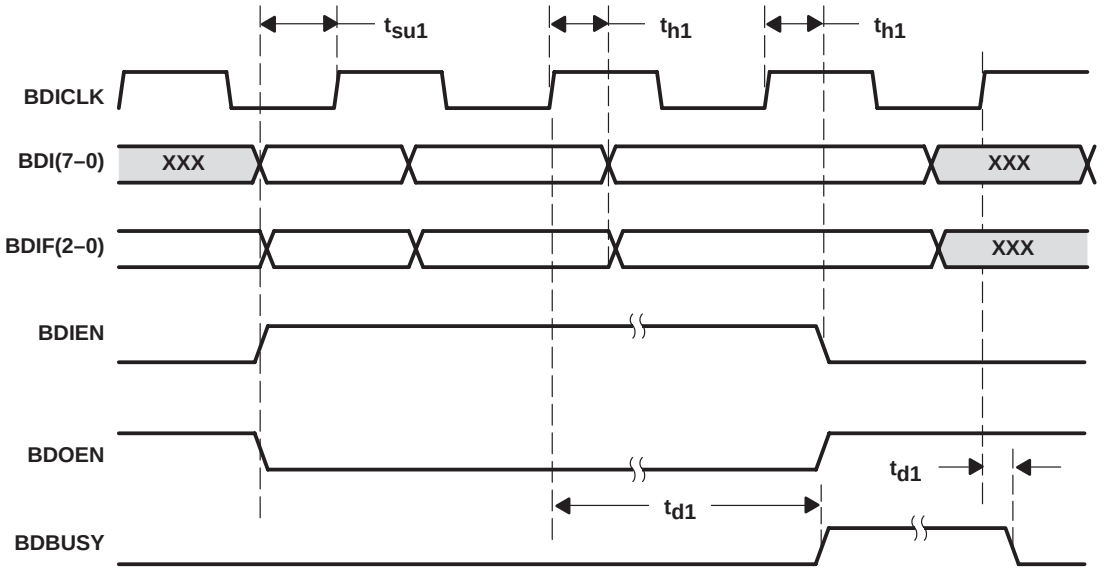


Figure 4–8. Functional Timing for Write Operations in the Unidirectional Modes



NAME	MIN (ns)	MAX (ns)	DESCRIPTION
t_{su1}	3		Setup time to rising edge of clock
t_{d1}		6	Delay time from rising edge of clock
t_{h1}	1		Hold time from rising edge of clock

Figure 4–9. Critical Timing for Write Operations in Unidirectional Mode

Asynchronous Write Timing

In the asynchronous mode (mode C), writes occur through the BDIO[7–0] data bus on every fourth clock cycle (NCLK). There has to be at least one BDIEN inactive clock cycle between two write requests. The host is responsible for meeting the necessary BDIEN timing relative to NCLK. (BDIEN is 1/4th NCLK in the following example.) Since the asynchronous mode only supports MPEG2(DVB)/DSS data, only the BDIF2 format signal is necessary to indicate the beginning or continuing byte of an MPEG2(DVB)/DSS cell. Please see Figure 4–10 for asynchronous mode functional timing.

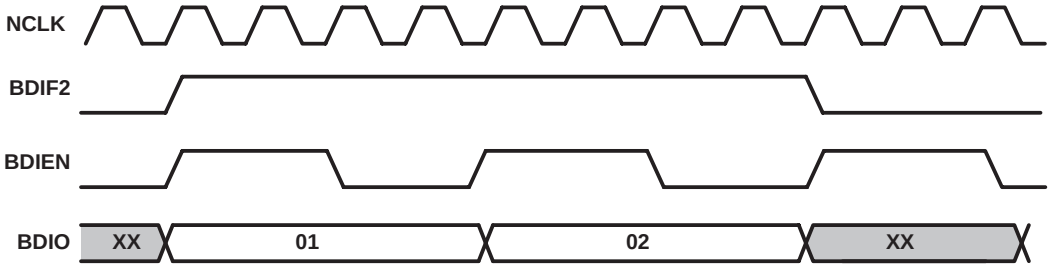
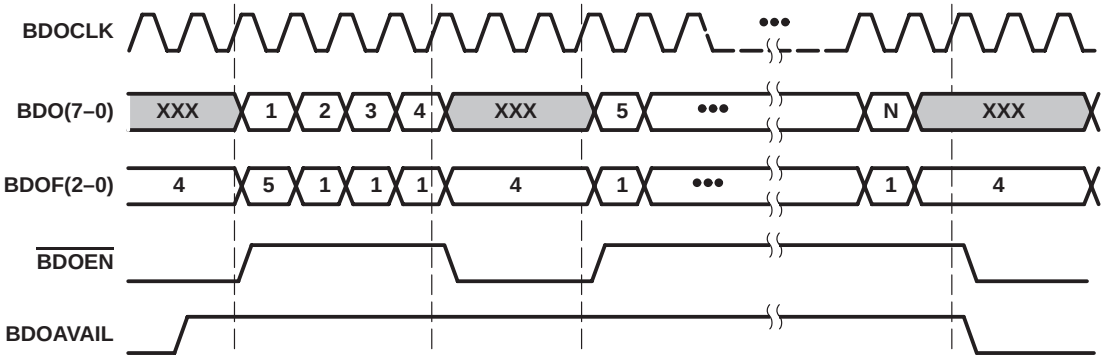


Figure 4–10. Functional Timing for Write Operations in the Asynchronous Mode

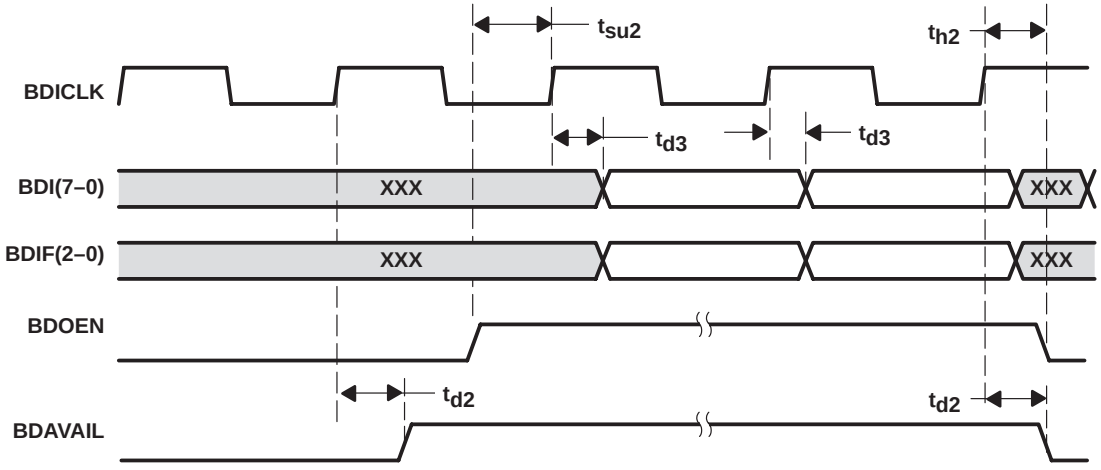
4.2.1.2 Unidirectional Read Timing

In the unidirectional modes (modes A and B), data is read out of the TSB12LV41A bulky data interface on the BDIO[7–0] pins. The data format is represented on the output format bus, BDOF[2–0]. The format is similar to BDIF[2–0] and is explained in Section 4.1.1, *Bulky Data Interface*. As shown in Figure 4–11, BDOEN signals the TSB12LV41A that the application is reading data on the BDO[7–0] data lines. For modes that do not utilize BDOEN (mode B), data is output to the application as soon as it is received in the TSB12LV41A FIFO. BDOAVAIL signals the application that it has data in the Bulky Receive FIFOs available for reading. BDOAVAIL is active whenever the FIFO has loaded the bulky data holding register with the first data quadlet that is available in the FIFO. Please see Figure 4–12 for critical read timing in bulky data modes A and B.



NOTE A: BDOEN is not necessary in Mode B.

Figure 4–11. Functional Timing for Read Operations in Unidirectional Mode



NAME	MIN (ns)	MAX (ns)	DESCRIPTION
t_{su2}	12		Setup time to rising edge of clock
t_{d2}		6	Delay time from rising edge of clock
t_{d3}		6	Delay time from rising edge of clock
t_{h2}	0		Hold time from rising edge of clock

NOTE A: BDOEN is not necessary in Mode B.

Figure 4–12. Read from Bulky Interface Unidirectional Mode

Asynchronous Read Timing

In the asynchronous mode (mode C), data is read out of the TSB12LV41A bulky data interface on the BDIO[7–0] terminals. The read operation can occur only every four NCLK cycles. There has to be at least one BDOEN inactive clock cycle between two consecutive read operations. The host should latch the data on the next rising edge of BDOEN. Please see Figure 4–13 for asynchronous read functional timing.

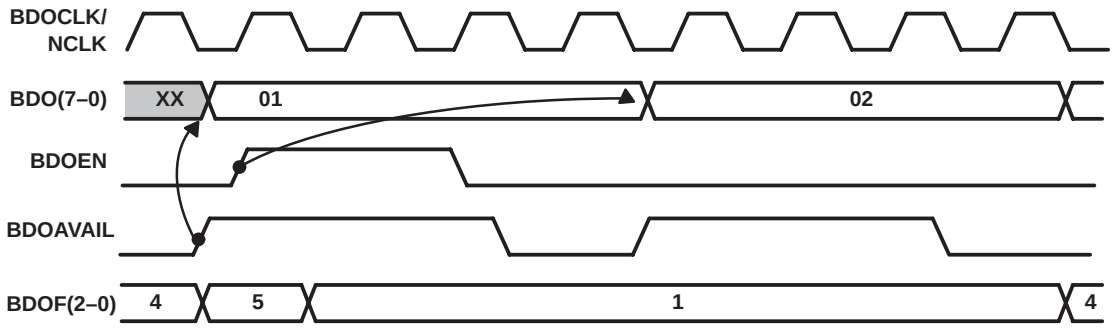


Figure 4–13. Functional Timing for Read Operations in Asynchronous Mode

4.2.2 Bidirectional Modes

Writes and reads in the bidirectional mode all occur on the BDIO[7–0] data bus and BDIF[2–0] format lines. Only bulky data mode D supports bidirectional data transfer.

4.2.2.1 Bidirectional Write Timing

Writes to the bulky data interface can occur through the BDIO port for both unidirectional and bidirectional modes. For writes to the bulky data interface in bidirectional mode, please see Figures 4–14 and 4–15.

In bidirectional mode, the BDIF[2–0] signals are the format bus. This signals the start of an MPEG2 packet (binary value of 101), a byte of an MPEG2 cell (binary value of 001), or an idle (binary value of 100). There are also format codes for isochronous and asynchronous data. (Please see the *Bulky Data Interface*, Section 4.1 for more details.) The BDIO[7–0] bus is used for data. BDIEN serves as the read/write enable. It enables the bulky data interface to accept either reads or writes from the application. The BDOEN serves as the read/write control signal. If BDIEN is active, then BDOEN high corresponds to a read and BDOEN low corresponds to a write. BDIBusy signals when the TSB12VL41 FIFO is full and can not accept any more data.

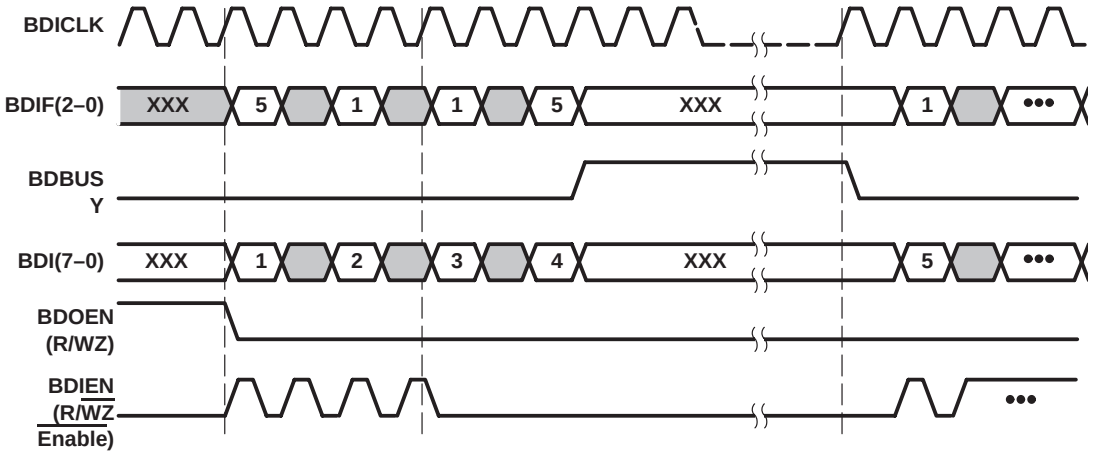
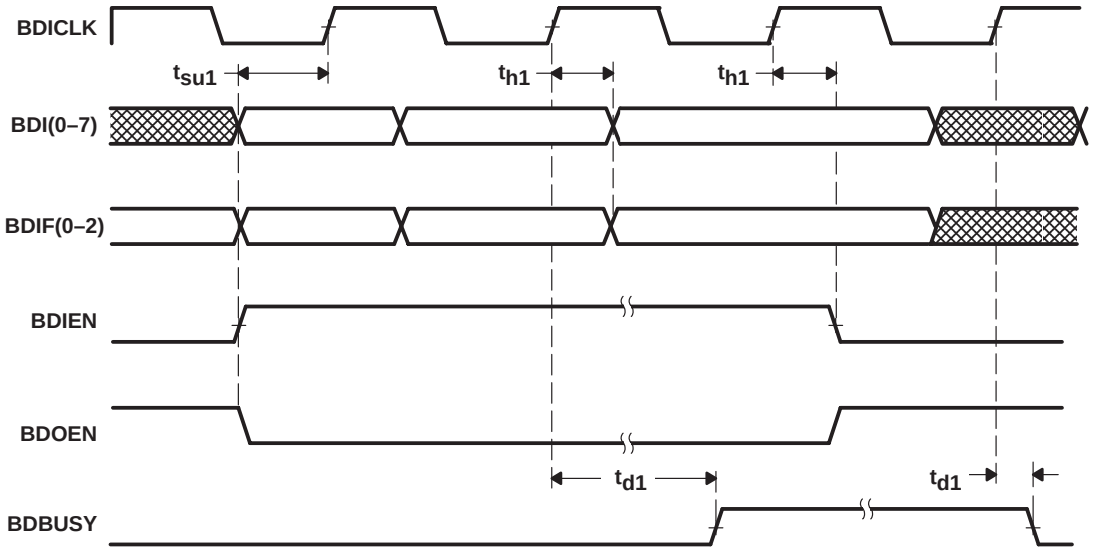


Figure 4–14. Functional Timing for Write Operations in Bidirectional Mode



NAME	MIN (ns)	MAX (ns)	DESCRIPTION
t_{su1}	3		Setup time between BDIEN (RW enable) and rising edge of clock
t_{h1}	1		Hold time from rising edge of clock
t_{d1}		6	Delay time from rising edge of clock

Figure 4–15. Critical Timing for Write Operations in Bidirectional Mode

4.2.2.2 Bidirectional Read Timing

In the bidirectional mode, the BDIF[2–0] format bus signals the bytes of data packets, similar to the bidirectional write operation. The data is presented to the application on the BDIO[7–0] data bus. BDIEN serves as a read/write enable. It enables the bulky data interface to accept either reads or writes from the application. The BDOEN serves as the read/write control signals. If BDIEN is active, then BDOEN high corresponds to a read and BDOEN low corresponds to a write. BDOAVAIL signals the application when the bulky receive FIFOs have data to read. Please refer to Figures 4–16 and 4–17 for bidirectional read timing.

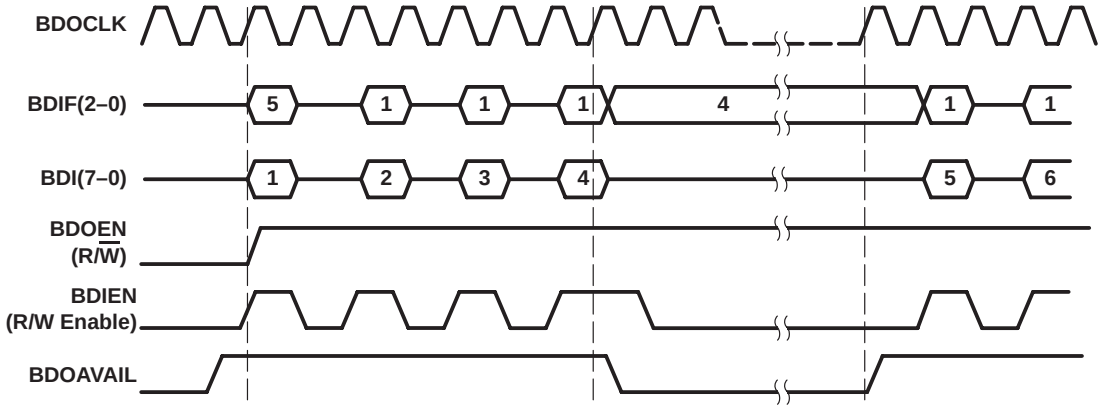
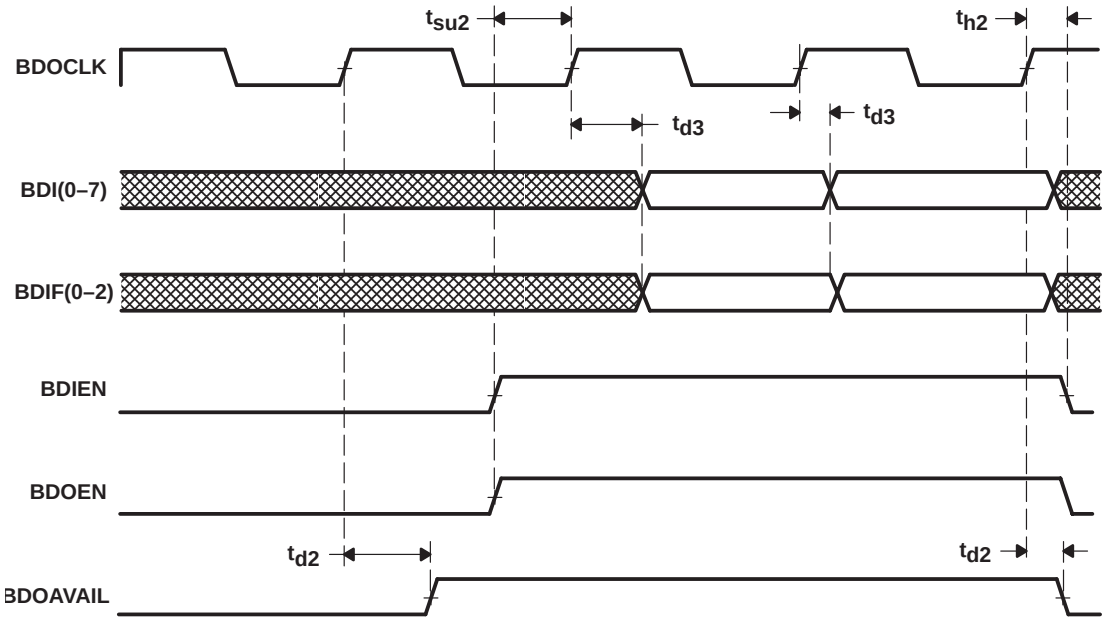


Figure 4–16. Functional Timing for Read Operations in Bidirectional Mode



NAME	MIN (ns)	MAX (ns)	DESCRIPTION
t_{su2}	12		Setup time to rising edge of clock
t_{d2}		6	Delay time from rising edge of clock
t_{d3}		6	Delay time from rising edge of clock
t_{h2}	0		Hold time from rising edge of clock

Figure 4–17. Critical Timing for Read Operations in Bidirectional Mode

4.2.2.3 Bidirectional Interleave Timing

Figure 4–18 shows the read and write interleave operation in bidirectional mode. BDICLK is used for writes to the bulky data interface and BDOCLK is used for reads form the bulky data interface.

(BDIO[7–0]_OUT and BDIO[7–0]_IN are physically the same bus. They are named so for ease of readability. The same is true for BDIF[2–0].)

BDOAVAIL active indicates that a packet is available in the receive FIFO. The application activates BDIEN (enabling read and write operations) and drives BDOEN high, indicating a read. The TSB12LV41A output the first byte of MPEG2/DSS/Asynchronous/Isochronous data. The TSB12LV41A indicates the data's format by driving values on the BDIF format bus.

Next the application performs a write operation by driving BDOEN low while keeping BDIEN active. The application simultaneously drives BDIO[7–0] with the first byte of the data packet. BDIF[2–0] is also driven by the application to indicate the data packet format.

This sequence continues with a read followed immediately by a write until the hardware indicates busy by activating BDIBUSY. This suspends write operations to the bulky data interface. However, read operations continue and take the full bandwidth of the interface. After byte 7 is read, the application suspends the read by deactivating BDIEN. At this point, both the read and write are suspended and the buses are in a high impedance state. BDIBUSY is deactivated indicating that data can continue to be written to the transmit FIFO. The application activates BDIEN and drives BDOEN low to indicate another write to the bulky data interface. Another read follows this.

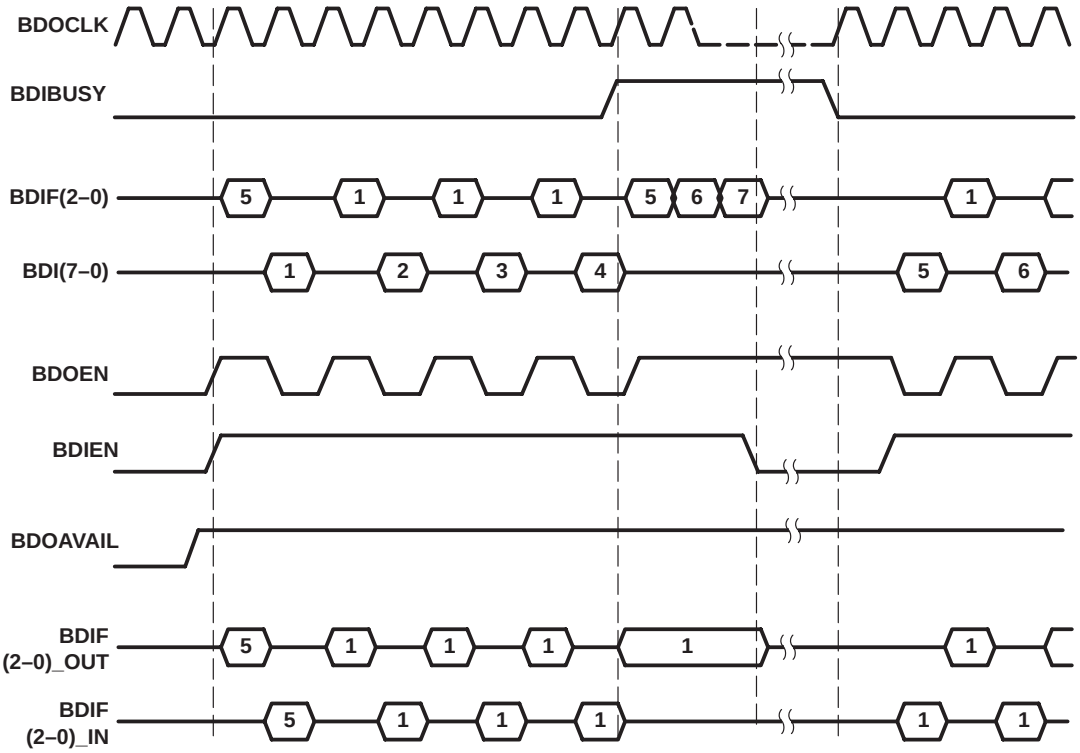


Figure 4–18. Functional Timing for Read/Write Interleave Operations in Bidirectional Mode

4.3 Microprocessor Interface

4.3.1 Microprocessors Supported

The TSB12LV41A's microprocessor interface supports the following three kinds of microprocessor/microcontrollers:

- The embedded ARM processor in Texas Instruments' TMS320AV7100 Integrated Set-Top DSP
- Motorola's 68xxx class microprocessors
- Intel's 8051 microcontroller

To be able to detect which kind of microprocessor/microcontroller (MP/MC) is connected to TSB12LV41A, two MP/MC select lines, MCSEL1 and MCSEL0, need to be driven to certain logic levels. Table 4–4 shows the MCSEL settings for each type of processor.

Processor Selected	MCSEL1	MCSEL0
Reserved	1	1
8051	1	0
68000	0	1
TMS320AV7100 ARM	0	0

Table 4–4. MCSEL Settings for Various Microprocessors

Once the type of MP/MC has been determined, all the I/O control pins related to MP/MC Interface map their functions to be applicable to the type of MP/MC detected. The following matrix table (Table 4–5) defines the actual pin functions for a particular type of MP/MCs:

TSB12LV41A Pin Name	MP/MC Type		
	Motorola 68000	TMS320AV7100	Intel 8051
ADR[0:8]	ADR[8:1]	EXTADDR[8:0]	ADR0=PSENZ, ADR1=ALE
DATA[0:15]	D[15:0]	EXTDATA[15:0]	DATA[8:15]=AD[7:0]† DATA[7]=P2.A0
CS/CSZ	CS	CSXZ	CSZ‡
MCCTL0	R/WZ	EXTR/WZ	WRZ
MCCTL1	Unused, tied high	Unused, tied high	RDZ
RDY	DTACKZ	EXTWAITZ	—

† AD[7:0] is connected to TSB12LV41A's DATA[8:15] as a bidirectional address/data bus for Intel 8051 and Intel 8051 port2's LS address bit A0 output is connected to TSB12LV41A's DATA[7] .

‡ CSZ is generated by board level glue logic, which is the binary address decoding of Intel 8051 Port2's address bus output A7 – A1.

Table 4–5. TSB12LV41A MP/MC Interface Pin Function Matrix

TSB12LV41A's Microprocessor Interface is synchronized to the BCLK in TMS320AV7100 Mode. BCLK input is from TMS320AV7100's extension bus external clock input (CLK40, 40.5MHz). For Motorola 68000 and Intel 8051 modes the TSB12LV41A's microprocessor interface is asynchronous. The internal clock used for these is derived from the 49.152-MHz SCLK from the PHY.

All bus signal labeling on the TSB12LV41A's microprocessor interface is denoted as bit0 as MSB and bit15 as LSB.

Note that the data path from the microprocessor interface to the host is 16 or 8 bits wide; however, all internal configuration registers are 32 bits wide. Thus the microprocessor interface of the TSB12LV41A must stack the incoming/outgoing write and read 32 bit data prior to delivery to either a internal CFR or the

microprocessor data bus. It is in this stacking buffer that the byte swapping function required for different endianness settings is performed. Section 4.3.10 describes how the byte swap works for the various endianness settings.

Figures 4–19, 4–20 and 4–21 show hook up diagrams for each type of processor supported.

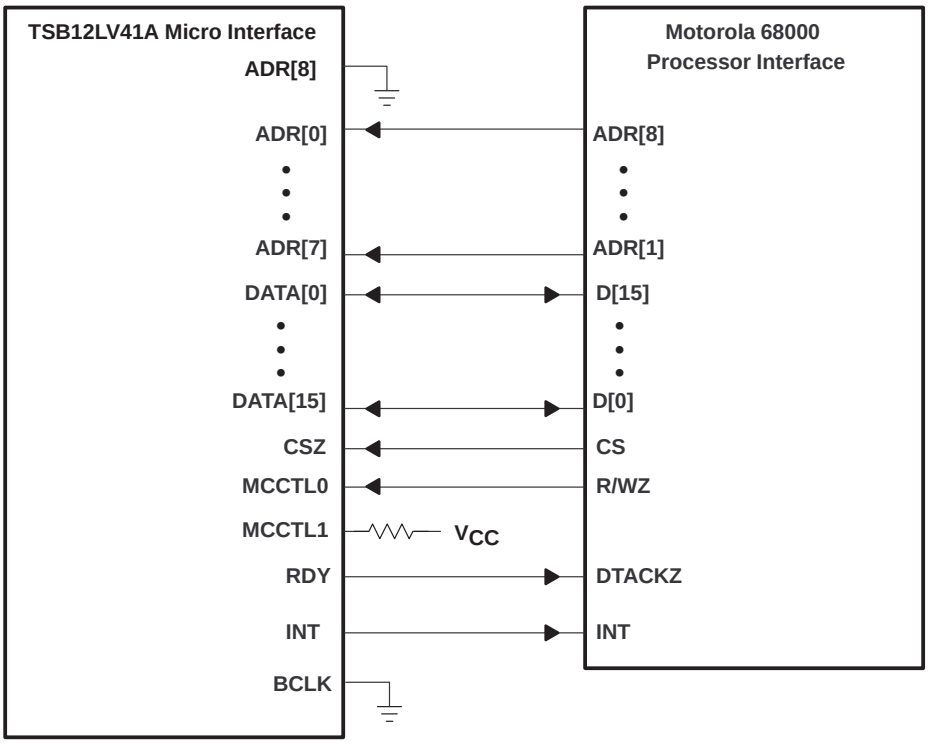


Figure 4–19. MPEG2Lynx Connections for 68000 Microcontroller

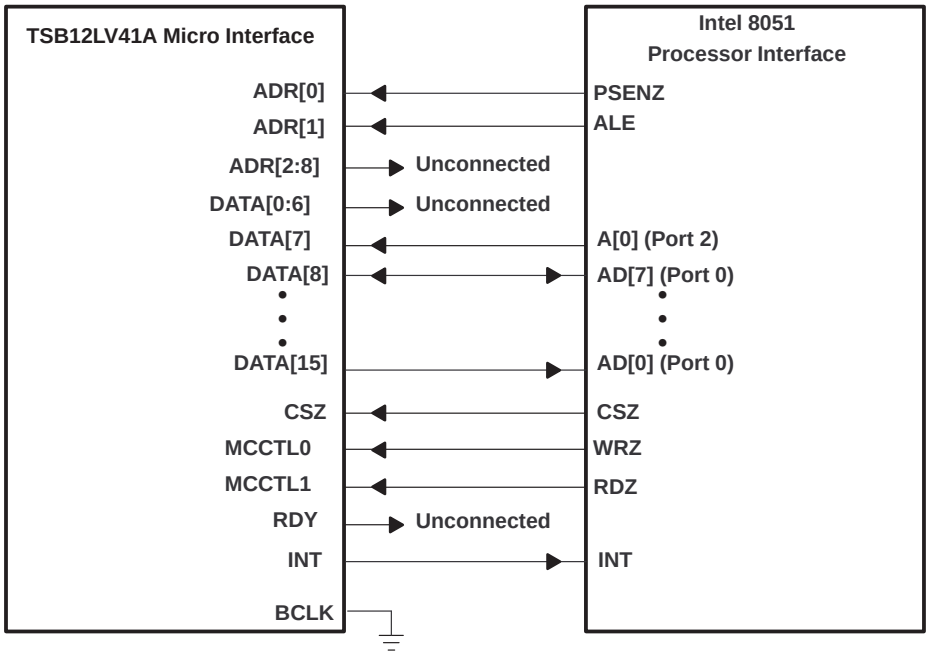
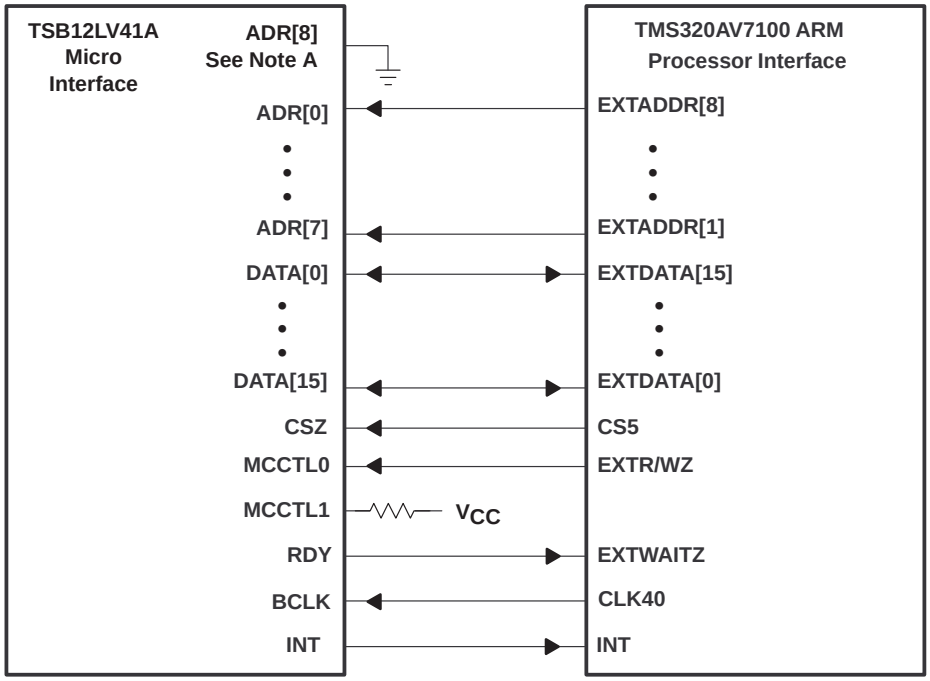


Figure 4–20. MPEG2Lynx Connections for 8051 Microcontroller



NOTE A: Connect as shown for 16-bit data. If 8-bit data bus is being used, connect ADR[8] on the TSB12LV41A to EXTADDR[0] on the TMS320AV7100.

Figure 4–21. MPEG2Lynx Connections for TMS320AV7100 ARM Processor

4.3.2 Microprocessor Interface Control

The microprocessor interface has several programmable functions such as the polarity of the RDY handshake signal and the endianness type to be used (for byte swapping). All optional functions on this interface are selected by the microprocessor via the I/O control register at offset 1ECh in the configuration register space. Figure 4–22 shows the bit map of the IOCR register. Table 4–6 shows a correlation table of the value and meaning of each bit, along with the power up default setting.

The TSB12LV41A supports both 8-bit and 16-bit data busses. It supports both little endian and big endian microprocessors. The TSB12LV41A can support either high or low true interrupt polarity. It can also support either totem-pole or open drain output for RDY and INT signals. The TSB12LV41A does not provide any on-chip pullup or pulldown resistor for those open-drain type outputs. The board level designer will have to add it if necessary to achieve appropriate level control or sharing between devices.

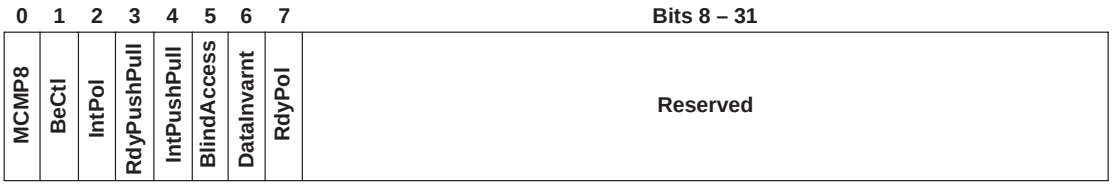


Figure 4–22. TSB12LV41A IOCR Register

Bit Number	Bit Name		Bit Value Setting Meaning		Power Up Default Setting		
	Symbol	Description	Value = 1	Value = 0	TMS320AV7100	68000	8051
0	MCMP8	Micro bus is 8/16 bit access	Byte access	Word access	Word access		Byte access
1	BeCtl	Big endian control	Big endian	Little endian	Big endian		Little endian
2	IntPol	Interrupt polarity control	High true	Low true	Low true		
3	Rdypush-pull	RDY output signal control	Active push/ pull	Three-state	Three-state		N/A†
4	Intpush-pull	INT output signal control	Active push/ pull	Three-state	Active push/pull	Three-state	
5	Blind access	Blind access enable/ disable	Enable blind access mode	Disable blind access mode (handshake)	Enable blind access mode	Disable blind access mode	Enable blind access mode
6	Data invarnt‡	Data invariant endianness control	Data invariant	Address invariant	Data invariant		
7	RdyPol	RDY output polarity control	High true	Low true	Low true		

† When the BeCtl bit is set to 1 (big endian), the Data Invarnt bit setting has no effect.
‡ Although there is no RDY line connection in Intel 8051 Mode, reading the IOCR.RdyPushPull still returns the value of ”0” for this bit.

Table 4–6. TSB12LV41A IOCR Bit/Function Correlation Table and Power-Up Default Setting

Although the IOCR provides a way to set up the extension bus interface, not all settings are supported for each type of microprocessor. The following summarizes the special notes for each type of microprocessor supported:

- TMS320AV7100

  Both byte access and word access are supported. Little endian is supported but users have to take the risk of wrong data byte swapping, since the TMS320AV7100 is big endian.

- Motorola 68000

  Only word access is supported. Blind access mode is not supported.

- Intel 8051

  Word access is not supported since it is an 8 bit processor. Uses blind access mode only.

It is strongly recommended that users program the IOCR during the first access after the external reset (RESETZ) goes away. This ensures that the TSB12LV41A's microprocessor interface is programmed correctly to work with the particular type of micro being used.

To be able to get the new IOCR setting updated, it is very important to allow 4 clocks (for 40.5 MHz TMS320AV7100 clock) idle time after the last write to fill up the whole quadlet of IOCR. This rule applies to any IOCR write, regardless if it is the first power-up write or a later writes. Note that for the 8051 to be able to load the correct setting into the IOCR, it has to do four writes to finish the whole quadlet write. Only the first write contains the actual setting information (bits 0–7 of the IOCR register). The other three writes could be loaded with all zeros (bits 8–31 of the IOCR register).

4.3.3 Handshake and Blind Access modes

The host microprocessor can access the TSB12LV41A with either handshake or non-handshake mode. The handshake signal between TSB12LV41A and the micro is called RDY, which can be interpreted as either ready or wait, based on the type of microprocessor being used. A read or write transaction from the microprocessor is always initiated by assertion of the chip select (CSZ). In handshake mode, the microprocessor drives the target address onto the address bus, asserts the chip select and read/write control line for the type of transaction desired, and waits for or provides data on the data bus. The microprocessor then holds for the RDY signal acknowledge to assert and terminate the transaction.

The non-handshake mode is the patent-pending blind access mode. In blind access mode, the microprocessor does not hold for the RDY acknowledge to terminate the transaction. Instead, the microprocessor always terminates the current transaction in a fixed number of cycles. It then polls the blind access status register (at offset address 1F0h) to determine if the current transaction is finished or not. More detail on the blind access mode functionality is provided in the *Blind Access Specific Issues* section. For both handshake and blind access mode, there are some common read/write rules that have to be followed in order to carry out a correct read or write transaction.

4.3.4 General Read Instructions

For the read case, the TSB12LV41A's microprocessor interface initiates a read process to the internal link logic after it senses a read request to a new quadlet address generated by the microprocessor, regardless of which byte position inside the quadlet the address falls. For example, a read request from either address 0F5h or 0F6h returns the 32-bit value stored at address 0F4h (DSS formatter control register). In handshake mode, the TSB12LV41A holds the microprocessor read transaction cycle for the internal response before it releases the RDY signal to indicate to the microprocessor that the current transaction is complete.

The microprocessor interface's byte stacking buffer holds the whole quadlet provided by the internal link logic. Any follow-up reads to different byte/word positions within the same quadlet boundary retrieves the data from the byte stacking buffer (rather than generate a new read transaction each time). Therefore, the first read to a new quadlet address always takes a little more time than all the other follow-up reads, since

it is this first read that handshakes internally. The read access latency to the internal link logic is a maximum of 19 clock cycles (from the falling edge of CSZ).

If the host attempts to read the same byte/word position inside the same quadlet boundary twice, the microprocessor interface initiates a new read transaction to the same quadlet again to obtain the new data. Consider the following: If the current transaction is a read request to address 054h with read to byte0 and byte1 completed, the host now leaves the current transaction to do something else (maybe access another device) then returns to access the TSB12LV41A again:

- A read to 054h byte0 or byte1 results in the new read cycle being initiated to the internal logic
- A read to 054h byte2 or byte3 gets the held data stored in TSB12LV41A microprocessor interfaces stacking buffer from the original read request
- A read to any address other than 054h results in a new read cycle to the new quadlet address
- A write to the TSB12LV41A initiates a write cycle and any future read requests initiates a new read cycle

4.3.5 General Write Instructions

For the write case, the TSB12LV41As microprocessor interface only initiates a write cycle when the byte stacking buffer is filled up. This means that the first three byte writes in byte access mode or first word write in word access mode only loads part of the quadlet into the byte stacking buffer. Once the TSB12LV41A receives the complete quadlet, it then initiates a write cycle to the internal logic and passes along the quadlet address and quadlet data. Meanwhile, an internal cycle acknowledge is sent to the microprocessor interface state machine without waiting for a response from the internal logic. This would allow the microprocessor interface to accept a new read or write transaction. A new read or write is allowed, although a new write request is only allowed to preload the first three bytes or first word (a new write transaction cannot be actually started until the present write cycle is complete). Note that a third transaction is not allowed. When the response is returned from the internal logic, the microprocessor interface sends the acknowledge to the host processor (RDY) and starts the new (pre-loaded) transaction if any.

The TSB12LV41A supports any byte/doublet order writes, as long as they are within the same quadlet boundary. If the user tries to do either one of the following before he fills up the complete quadlet, the current write is ignored and a invalid write interrupt is issued (interrupt bit INVWROP):

- Host attempts to write to a new quadlet address before completing the current write quadlet load
- Host attempts a read process before completing the current write quadlet load
  write byte_3 —> write byte_1 —> write byte_2 —> read process ==> ERROR!
- Host attempts to write to a byte address within the same quadlet twice:
  write byte_3 —> write byte_1 —> write byte_2 —> write byte_3 ==> ERROR!

To summarize, read and write rules; On read accesses, the first read to a new quadlet address is the one that handshakes with the internal link core to obtain the quadlet data. The follow up reads within the quadlet boundary only have to read data from microprocessor interfaces byte stacking buffer. On write accesses, the last write to fill up the whole write quadlet is the one that handshakes with the internal link core to transfer the whole quadlet to the register. The first three byte writes or first doublet write only loads data into the microprocessor interfaces byte stacking buffer.

4.3.6 TMS320AV7100 Mode Timing Diagrams

TSB12LV41A is designed to meet the read/write access timing defined by TI's TMS320AV7100 extension bus interface. The following figures show critical and functional timing for read and write operations. Note that this mode supports both handshake and blind access modes, thus timing diagrams are given for each.

PARAMETER		MIN	MAX	UNIT
t_{su1}	Setup time, CSZ to BCLK rising edge	12.2		ns
t_{su2}	Setup time, R/WZ to BCLK rising edge	12.1		ns
t_{su3}	Setup time, Address to BCLK rising edge	8.9		ns
t_{su4}	Setup time, Data to BCLK rising edge	1.2		ns
t_{h1}	Hold time, CSZ after BCLK rising edge		0.8	ns
t_{h2}	Hold time, R/WZ after BCLK rising edge		1	ns
t_{h3}	Hold time, Address after BCLK rising edge		0.8	ns
t_{h4}	Hold time, Data after BCLK rising edge		0.9	ns
t_{d1}	Delay time, WAITZ low		17 BCLK cycles	ns
t_{d2}	Delay time, CSZ falling edge to WAITZ		2 BCLK cycles + 8 ns	ns
t_{d3}	Delay time, CSZ rising edge to CSZ falling edge	2.5 BLCK cycles		ns
t_{d4}	Delay time, DATA valid after CSZ rising edge		5	ns
t_{d5}	Delay time, CSZ low	5 BCLK cycles		ns
t_{ck}	BCLK period	24.5		ns

Table 4–7. TMS320AV7100 Critical Timing Characteristics

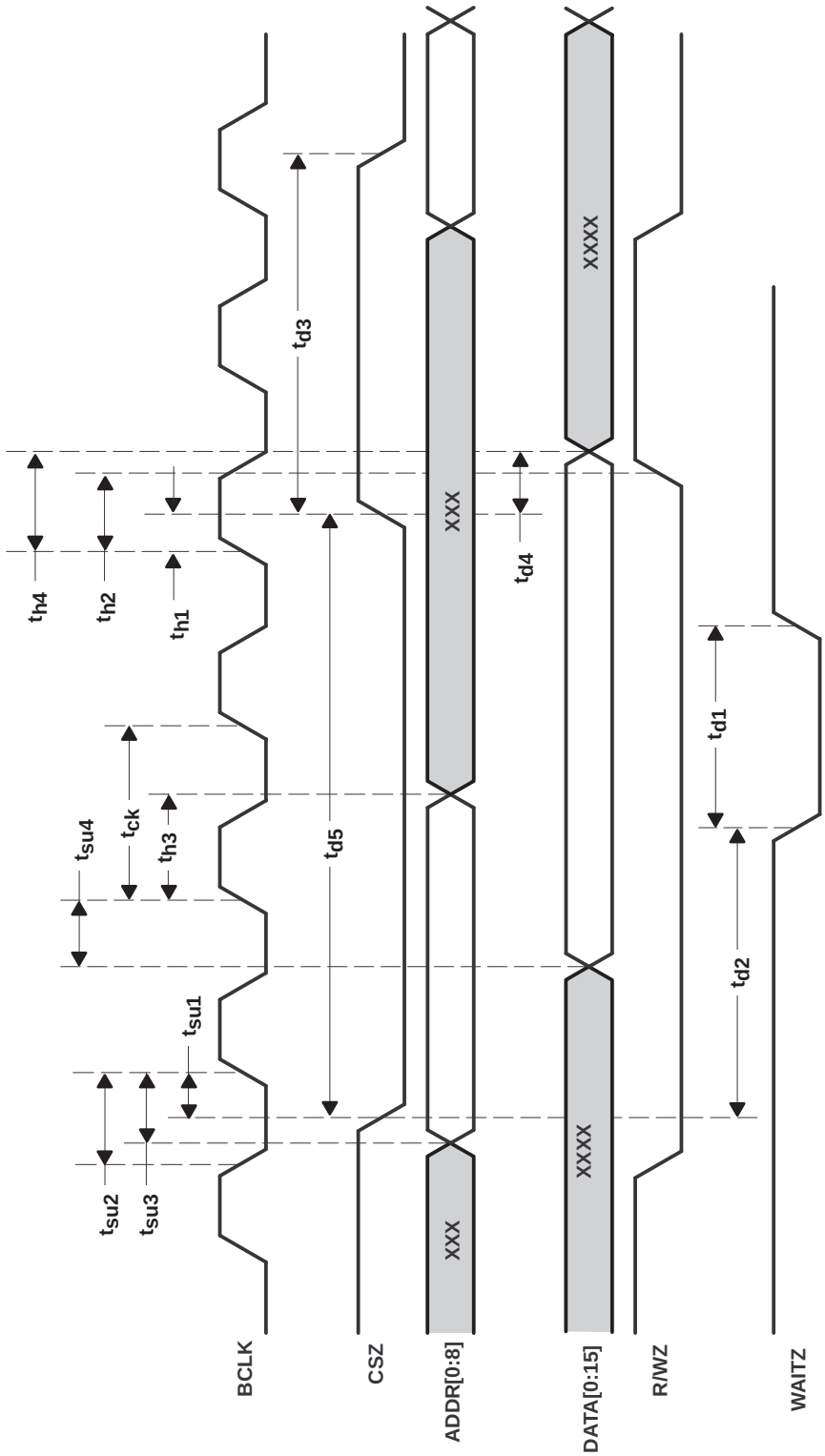


Figure 4–23. TMS320AV7100 ARM Read/Write Critial Timing

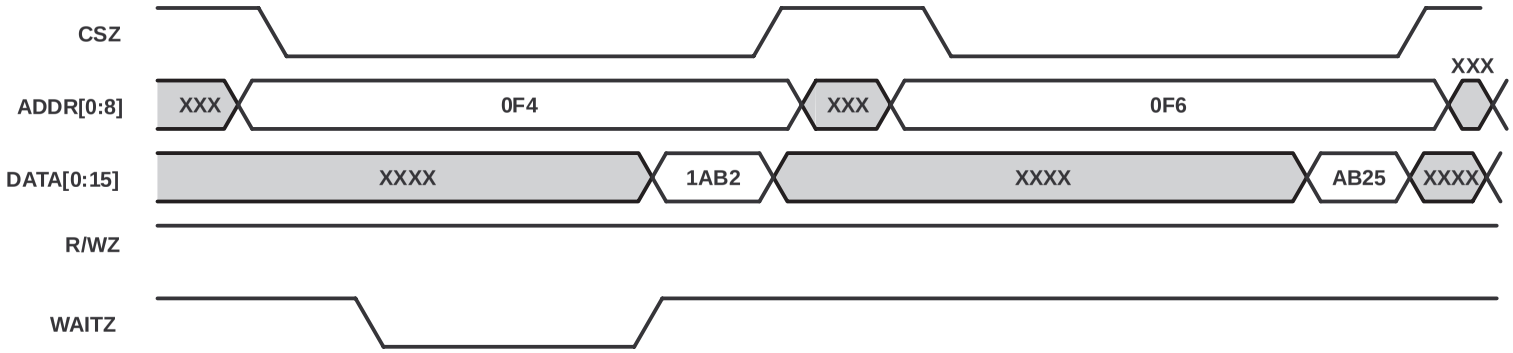


Figure 4–24. TMS320AV7100 Handshake Mode Read Timing

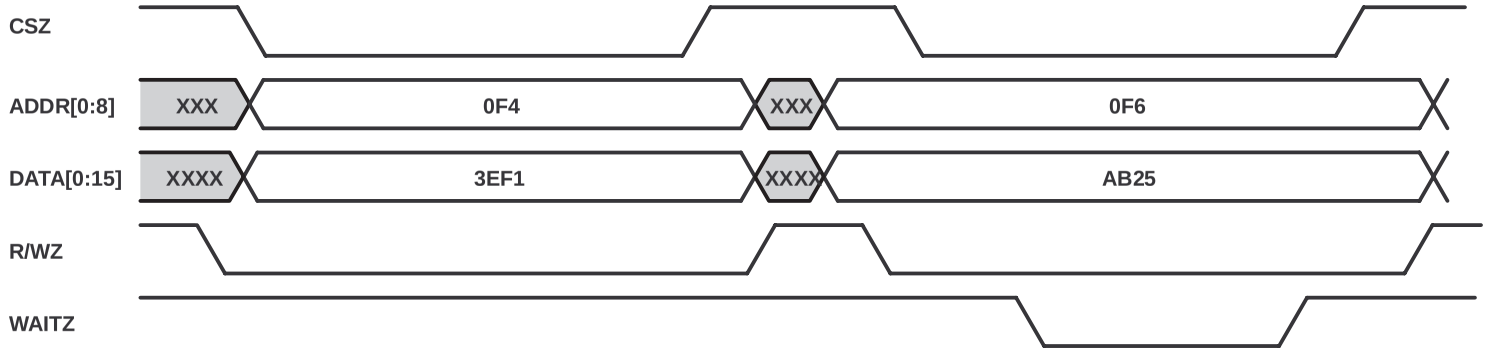


Figure 4–25. TMS320AV7100 Handshake Mode Write Timing

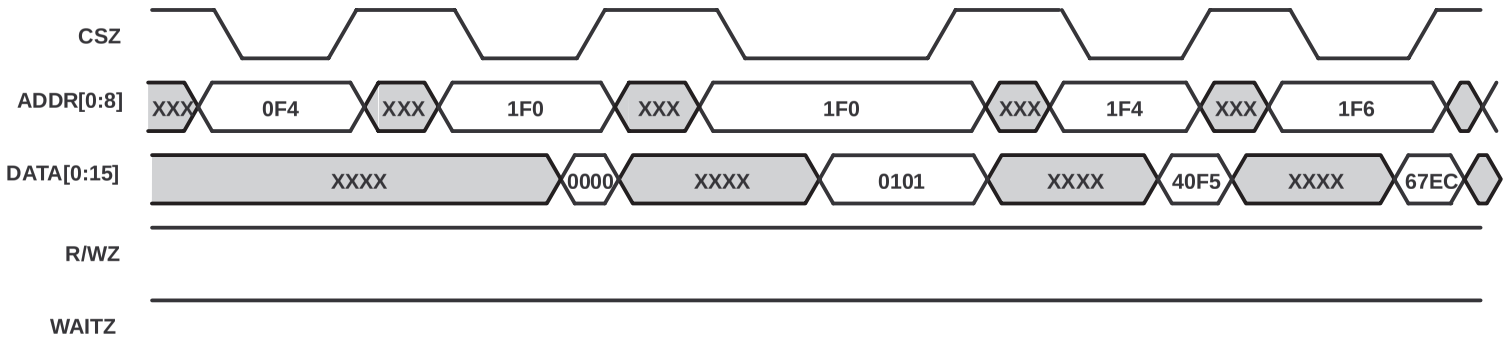


Figure 4–26. TMS320AV7100 ARM Blind Access Mode Read Timing

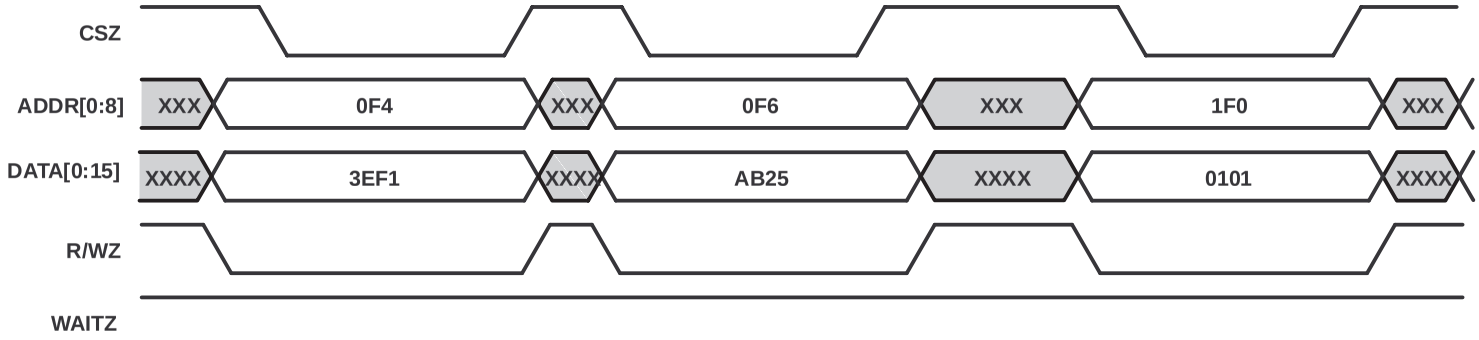


Figure 4–27. TMS320AV7100 ARM Blind Access Mode Write Timing

4.3.6.1 TMS320AV7100 Mode Specific Issues

BCLK is the input clock from the output of TMS320AV7100 chip (CLK40). Its frequency is 40.5 MHz.

For the write case, the TSB12LV41A latches the data at the next rising edge of BCLK after the CSZ goes low. Once the TSB12LV41A is ready to terminate the current cycle, it deasserts the EXTWAITZ (RDY) signal. The TMS320AV7100 samples the EXTWAITZ by the next rising edge of its CLK40 and a half cycle later, at the falling edge of CLK40, the TMS320AV7100 deasserts CSZ.

For the read case, once the TSB12LV41A has the read data available, it deasserts EXTWAITZ. The TMS320AV7100 deasserts CSZ by the next rising edge of its CLK40. The TSB12LV41A uses this CSZ rising edge to asynchronously turn off the data bus. Therefore, turning off the read cycle after EXTWAITZ roughly takes 2 CLK40 cycles.
Important Notes for TMS320AV7100 Mode:

1. 20 CLK40 Rule for EXTWAITZ signal
   The maximum allowed EXTWAITZ assertion time is 500ns for each chip select. The TMS320AV7100's extension bus interface is designed to disable the EXTWAITZ acknowledge from a peripheral if the acknowledge ever takes longer than 20 CLK40 cycles. Once the EXTWAITZ has been asserted more than 20 CLK40 cycles, the TMS320AV7100 assumes that the device generating the EXTWAITZ has failed and deasserts the CSZ at the next CLK40 cycle. Then the TMS320AV7100 ignores EXTWAITZ generated by all the devices on the bus. The TMS320AV7100 can still accept read and write transactions without using the EXTWAITZ signal by using the an internal programmable wait state register. To avoid this timeout, the TSB12LV41A microprocessor interface state machine aborts the current transaction and deassert the EXTWAITZ signal if the TSB12LV41A's internal logic cannot complete the transaction after 17 BCLK cycles. Only a software or hardware reset to the TMS320AV7100 can reactivate the EXTWAITZ input again.
2. EXTWAITZ to be recognized by TMS320AV7100 at the beginning of the transaction
   According to the TMS320AV7100 spec, the EXTWAITZ signal must assert before the programmed number of wait states expires. The TSB12LV41A is designed to always assert its wait line (RDY) on the second BCLK rising edge after it samples the falling edge of CSZ. In order for the TSB12LV41A to work with TMS320AV7100 seamlessly, it is recommended that after power-up, the host should change the wait state number to four (or greater) in the TMS320AV7100 ARM core. Otherwise, handshake mode may fail to function. (Note that blind access mode should work regardless of the TMS320AV7100 wait state setting.)
3. EXTWAITZ (RDY) Sharing Issue
   Since the TMS320AV7100 only has one EXTWAITZ input signal line, the TSB12LV41A may have to share this pin with other devices on the extension bus. Since EXTWAITZ has been defined as an open-drain type in TMS320AV7100, users must set the RdyPushPull bit in the IOCR register to zero (three-state) in order to avoid bus contention on the EXTWAITZ pin. The TSB12LV41A does not provide any on-chip pull-up resistor for this pin, thus the user also needs to add an external pullup resistor on this pin in this case. If the TSB12LV41A is the only device connected to the TMS320AV7100's EXTWAITZ input then the user can set RdyPushPull to 1.
4. INT (interrupt) Output Line Sharing Issue.
   TMS320AV7100 have dedicated INT lines, therefore no sharing of this signal is needed. TSB12LV41A outputs normal totem-pole signal level for this pin. Also, the application is allowed to write a *1* to the TSB12LV41A's interrupt register to clear the various interrupts, eliminating the need to use the TMS320AV7100's EXTACK signal (interrupt acknowledge).
5. TMS320AV7100 Byte Access mode data bus
   When using the TMS320AV7100's ARM processor in byte access mode (8 but data bus), the

upper byte (bit 0–7) of the TSB12LV41A's 16-bit data bus contains the byte data, the lower byte (bit 8–15) is driven low. When writing to the TSB12LV41A, the ARM host must put the byte write data in the upper byte. The data in lower byte has no effect to the data to be written into TSB12LV41A. (Intel 8051 mode uses the lower 8 bits of the data bus for data exchange.)

4.3.7 68000 Mode Timing Diagrams

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time,			0	ns
t_{d2}	Delay time,			130	ns
t_{su1}	Setup time,		10		ns
t_{su2}	Setup time,		0		ns
t_{su3}	Setup time,		2		ns
t_{h1}	Hold time,		0		ns
t_{h2}	Hold time,		0		ns
t_{h3}	Hold time,		0		ns
t_{h4}	Hold time,			7.2	ns
t_{h5}	Hold time,			40	ns

Table 4–8. Motorola 68000 Critical Timing Characteristics

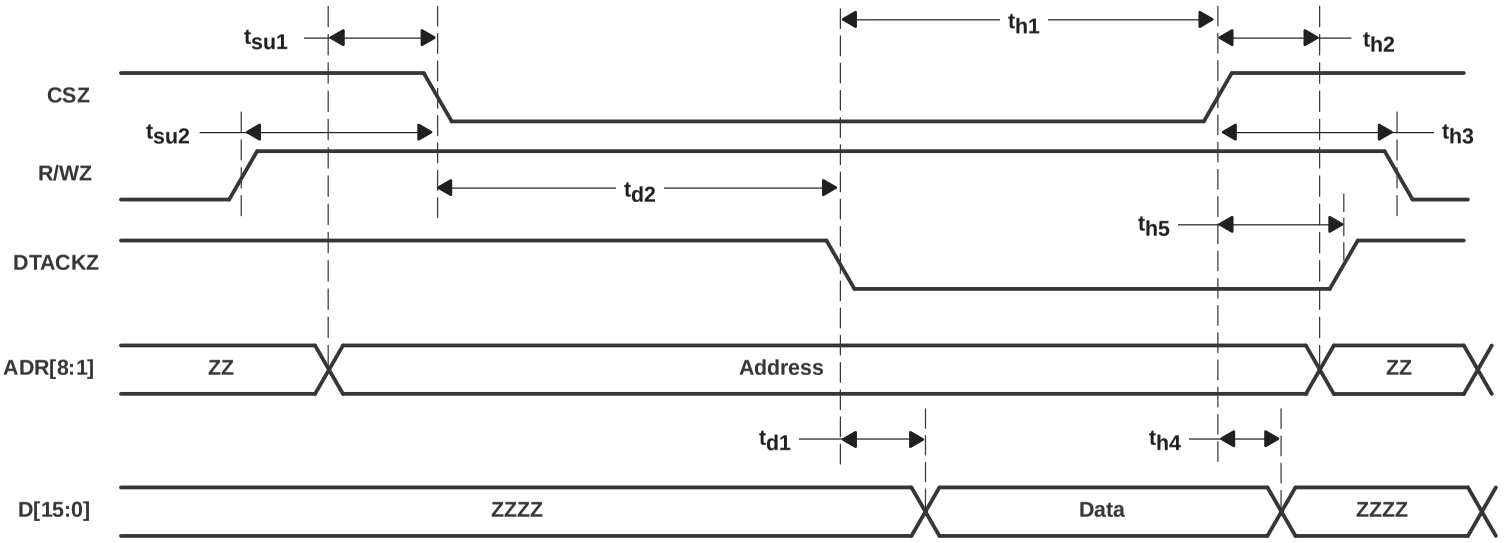


Figure 4–28. Motorola 68000 Read Critical Timing

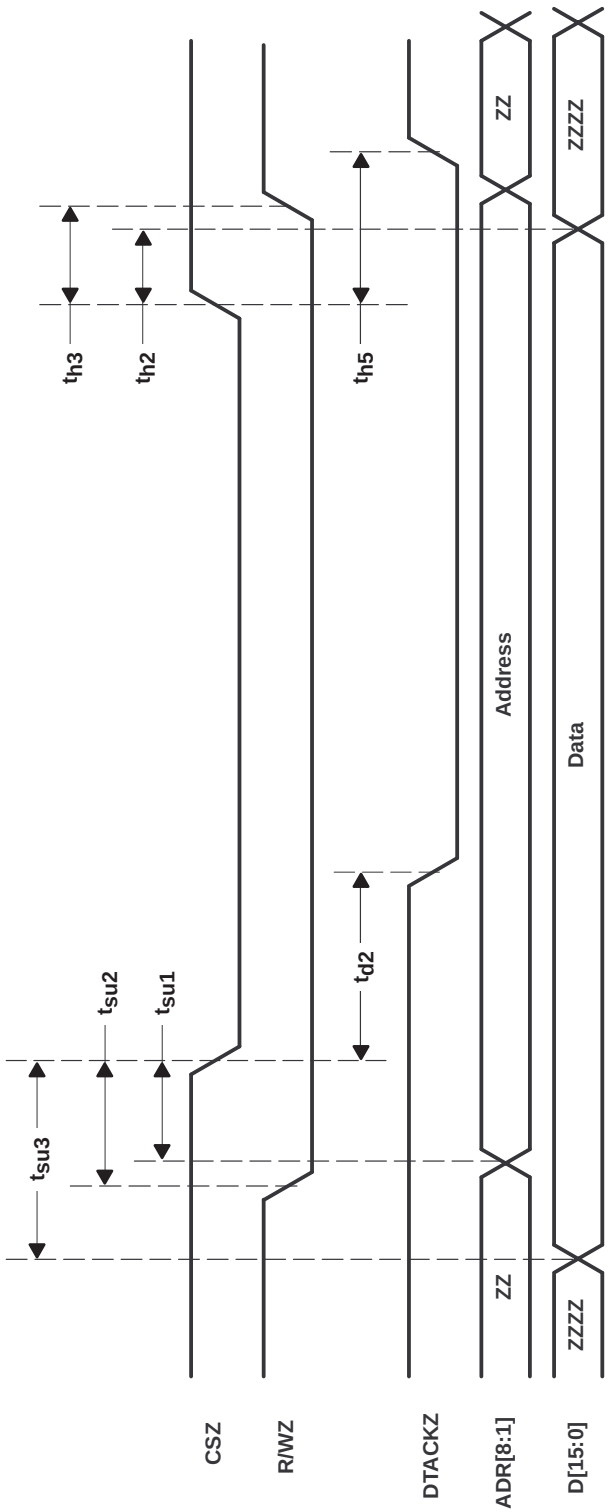


Figure 4–29. Motorola 68000 Write Critical Timing

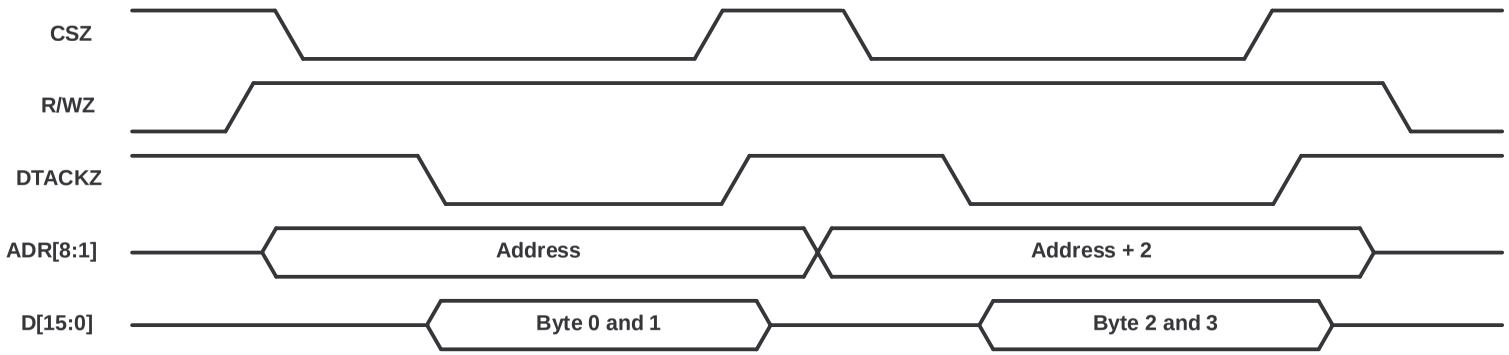


Figure 4–30. Motorola 68000 Read Timing

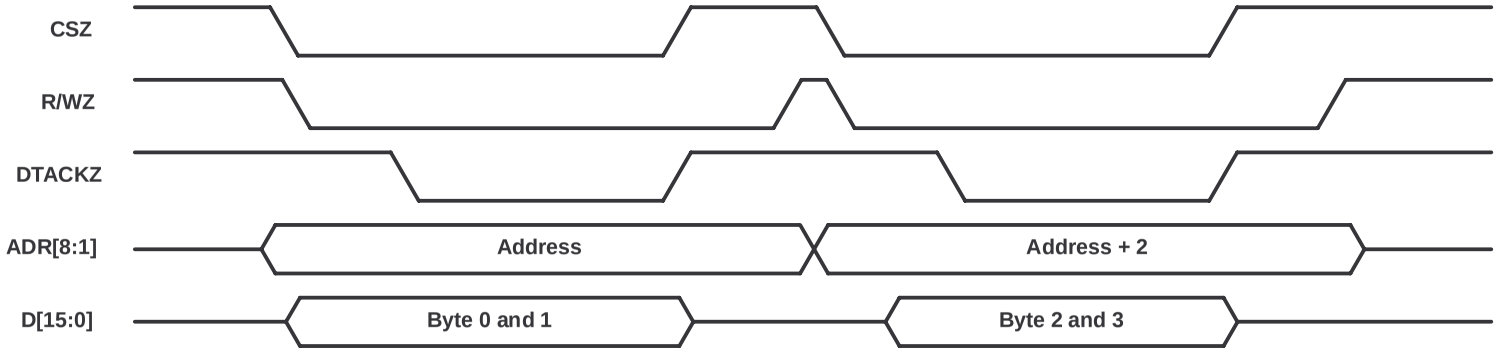


Figure 4–31. Motorola 68000 Write Timing

4.3.8 8051 Mode Timing Diagrams

Note that the Intel 8051 mode always operates in blind access mode.

The lower byte (bits 8–15) of TSB12LV41A's 16-bit data bus must be used for the byte data, the upper byte (bit 0–7) is driven low. When writing to the TSB12LV41A, the 8051 host should put the byte write data in the lower byte. The data in upper byte has no effect on the data to be written into TSB12LV41A.

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay time,		30		ns
t_{d2}	Delay time,		20		ns
t_{d3}	Delay time,		130		ns
t_{d4}	Delay time,		100		ns
t_{d5}	Delay time,		10		ns
t_{d5}	Delay time,		130		ns
t_{d6}	Delay time,		10		ns
t_{su1}	Setup time,		7		ns
t_{su2}	Setup time,		0		ns
t_{h1}	Hold time,		10		ns
t_{h2}	Hold time,		2		ns
t_{h3}	Hold time,		10		ns

Table 4–9. Intel 8051 Critical Timing Characteristics

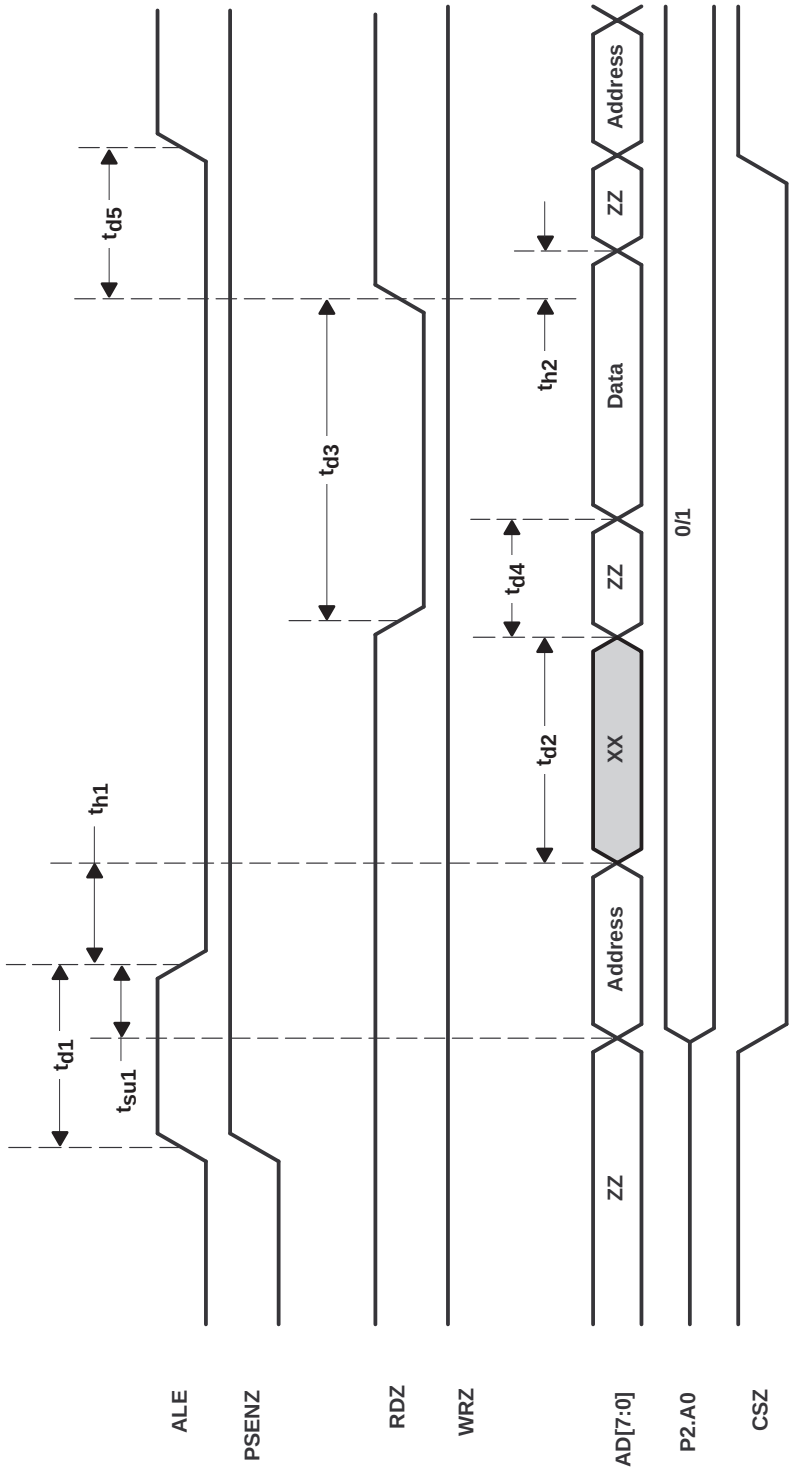


Figure 4–32. Intel 8051 Read Critial Timing

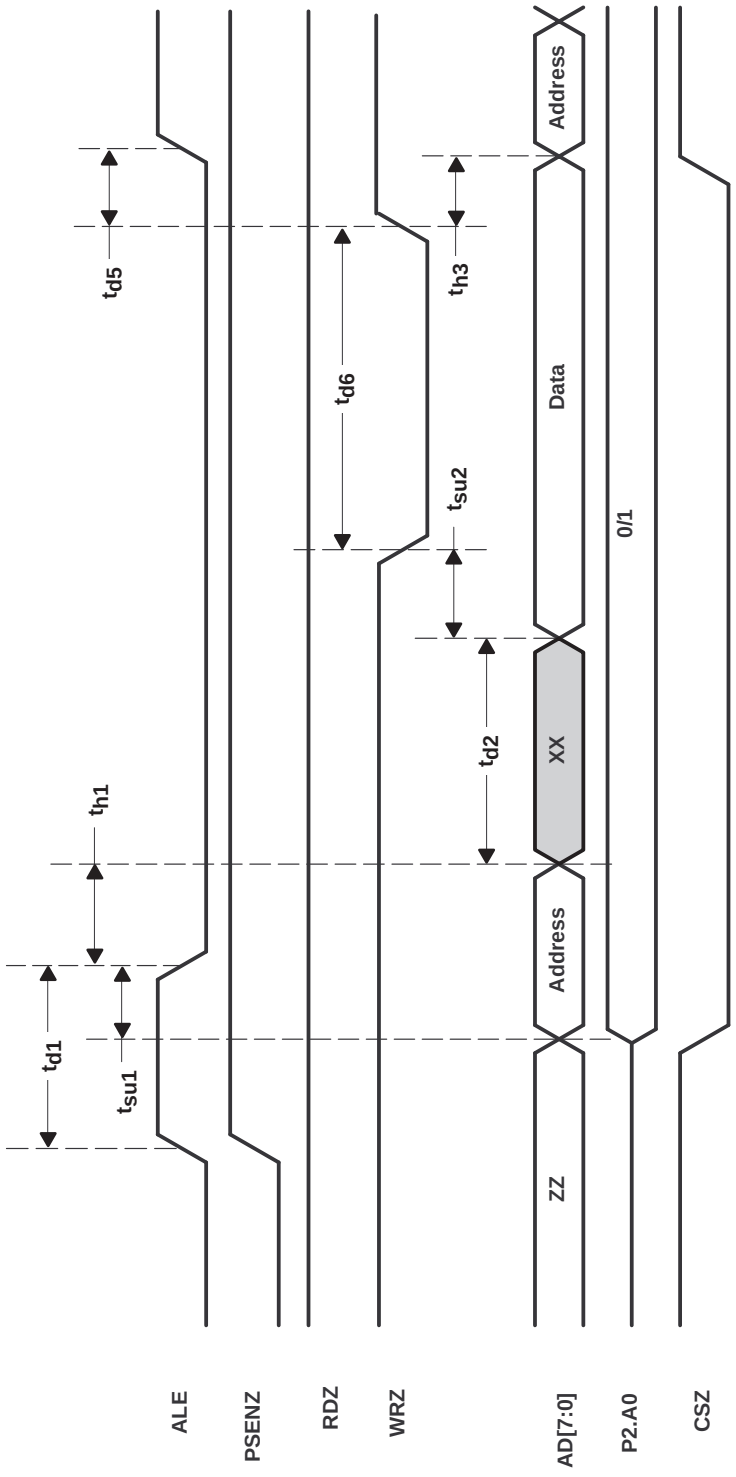


Figure 4–33. Intel 8051 Write Critial Timing

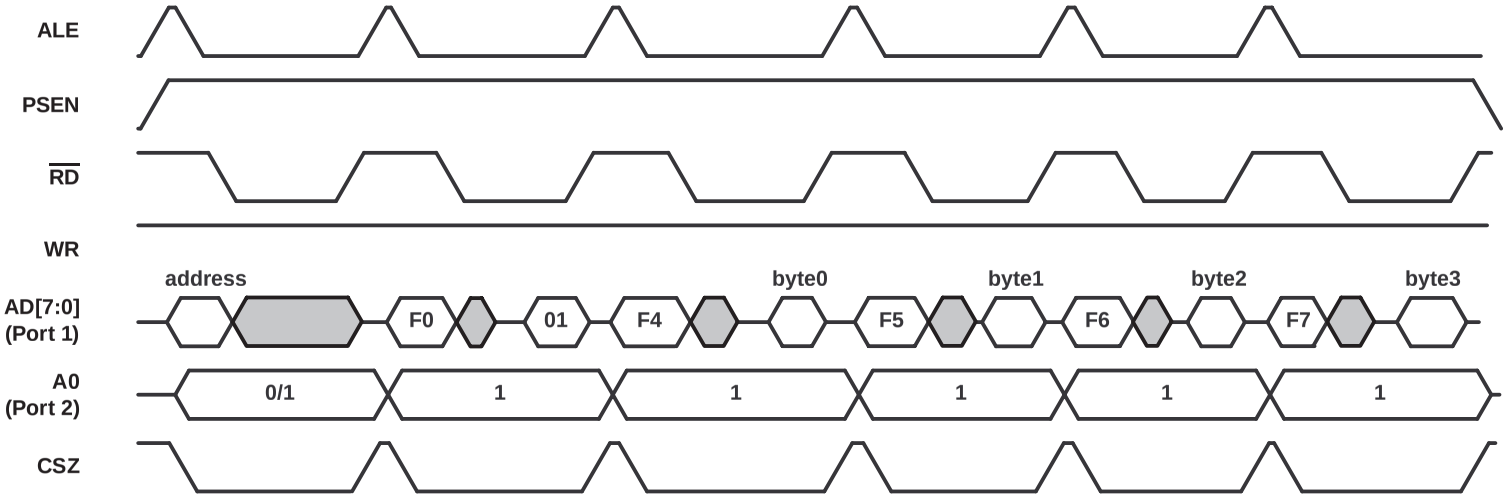


Figure 4–34. Intel 8051 Read Timing (Blind Access Read)

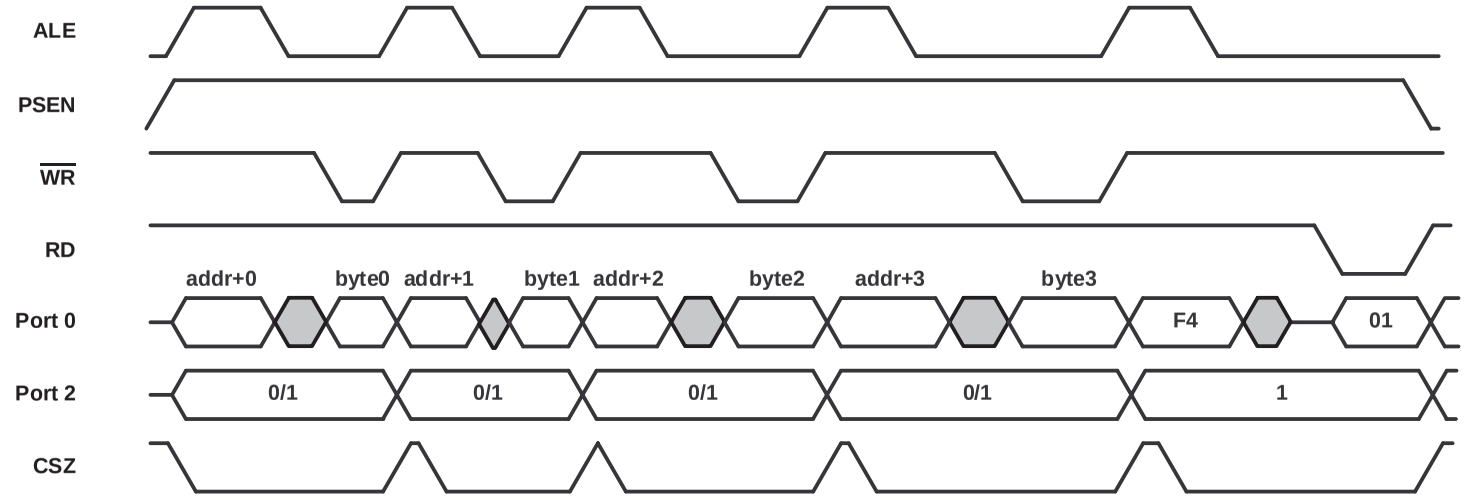


Figure 4–35. Intel 8051 Write Timing (Blind Access Write)

4.3.9 Blind Access Mode Specific Issues

Due to the long maximum access latency to some configuration registers (about 17 BCLK @ 50MHz), a patent-pending mode was developed to increase the burst speed of the microprocessor interface. This mode is called blind access mode. Users can take advantage of this mode by setting the blind access bit in the IOCR register to 1. The blind access mode is designed for faster microcontrollers like Intel's 8051 which do not have a external wait/ready line handshake signal and could be unnecessarily bandwidth limited by the long access time of the TSB12LV41A's microprocessor interface. The blind access mode allows a microprocessor to perform a read or write transaction to the TSB12LV41A without waiting for an acknowledge. The processor can then return at a later time and poll for the read or write status to determine if the transaction is complete, and if so, either start a new read/write or obtain the requested data from an internal holding register. This method allows the processor to use the CFR access latency time to perform other functions, thereby increasing processor performance. Blind access mode can be used in either TMS320AV7100 ARM mode or Intel 8051 mode.

There are two registers used for blind access mode operation, BASTAT (blind access status register, @ address 1F0h) and BAHR (blind access holding register, @ address 1F4h. The BAHR register is used as the holding register for data that is returned from a read address. The host processor can get the requested data only by reading this register. The BASTAT register can be used by the host to determine when the current read or write transaction is complete. Bits 7, 15, 23, and 31 in the BASTAT register are all the same status bit, BAcmp (blind access complete). If BAcmp is set to 1, this indicates that the current blind access transaction is finished. For a blind read, it means the data from the requested address is available in the BAHR register. For a blind write, it means the write data has been delivered to the requested address.

4.3.9.1 Blind Access Write

Blind access write follows the general write rules, except it does not use RDY signal as a handshake signal with the host processor. For consecutive writes, check the BAcmp bit in the BASTAT register and wait until the previous access is finished. Note that if the time period of consecutive writes is longer than the write latency, then the BAcmp bit does not have to be checked. The first 3 bytes (in byte access mode) or first word (in word access mode) written to the microprocessor interface is very quick, normally taking only 3 clock cycles for each write. When the write of the last byte or word to fill up the whole quadlet buffer comes in, it then initiates delivery of the whole quadlet to the address requested. By polling the BAcmp bit in BASTAT register, the host can then detect when the write is complete. Before the BAcmp bit is set, only reads from the BASTAT and BAHR registers are allowed. Also, once the whole quadlet buffer has been filled up with the data to be written, a write to the internal logic is initiated and can not be stopped except by a device reset. If the host mistakenly issues a new write or read before the BAcmp bit is set, a write/read error interrupt (INVWROP interrupt) is generated and this new write/read is aborted.

4.3.9.2 Blind Reads

Blind access read still follows the general read rules, but without the handshake RDY signal. The first read results in a dummy value on the data bus since this action only initiates the read request. The BAcmp bit in BASTAT can be checked to determine if the transaction is finished. If the time period of consecutive reads is longer than the read latency, then the BAcmp bit does not have to be checked. Before the BAcmp bit is set, only reads from the BASTAT and BAHR registers is allowed. Once the read procedure starts, it can not be stopped except by a device reset. If the host mistakenly issues a new write or read before the BAcmp bit is set, a write/read error interrupt (INVWROP interrupt) is generated and this new write/read is aborted. A follow-up read to BAHR can obtain the requested data once the BAcmp bit is set to 1.

4.3.9.3 BAcmp Bit Clear

For normal read and write cases, once the BAcmp bit has been set, reading to either BASTAT or BAHR causes the BAcmp bit to be cleared, regardless of whether the current blind access is read or write. However, because the BAcmp bit could be set at any time, even in the middle of a read to the BASTAT register, the TSB12LV41A's microprocessor interface ensures that:

- If the internal cycle acknowledge is returned before the first BCLK (or SCLK) rising edge inside the chip select window of the current read cycle to BASTAT, then the current read returns the status that BAcmp has been set to 1 and the BAcmp bit is cleared after the read.
- If the internal cycle acknowledge is returned after the first BCLK (or SCLK) rising edge, but still inside the current read cycle to BASTAT, then the current read returns the status that BAcmp has not been set. Then the BAcmp bit is set, held and is not cleared by the current read to BASTAT.
- A read to the BAHR register clears the BAcmp anyway as long as internal cycle acknowledge comes back before or at the current read BAHR cycle.

For those consecutive read/write transactions whose access time interval is longer than the read/write latency, a new blind read/write is issued without reading the BASTAT register to check the BAcmp status, the falling edge of CSZ clears the BAcmp bit. Thus, for blind write, the last write to fill up the whole quadlet buffer clears the BAcmp bit. For blind read, the first read transaction clears the BAcmp bit.

4.3.9.4 Special Notes on Blind Access

- Blind read and blind write accesses can not be nested inside each other.
- There are four registers located inside the TSB12LV41A's microprocessor interface domain: IOCR (I/O Control Register), BASTAT (blind access status register), BAHR (blind access holding register), and SRES (software reset register). Accessing them in the blind access mode does not need to go across any internal clock synchronization boundary, therefore, access to these registers is immediate and no status check to BASTAT is necessary.

4.3.10 Endianness

The term endianness refers to the way a processor stores and references bytes of data in memory. For example, consider a 32 bit processor; any 32 bit word consists of four bytes which may be stored in memory in one of two ways. Of the four bytes, either Byte 3 is considered the most significant byte and Byte 0 the least significant byte, or vice versa (see Figures 4–36 and 4–37). A little endian-type memory considers Byte 0 the least significant byte, whereas a big endian-type memory considers Byte 3 to be the least significant byte. This topic is of importance to users of the TSB12LV41A since all its configuration registers are big endian and users could potentially use a little endian-type processor. The TSB12LV41A uses the same endianness as the internal P1394 link core, which is big endian. Here we define the MSByte (most significant byte) to be Byte 0 at the left most hand side and LSByte (least significant byte) to be Byte 3 at the right most hand side.

Byte#0 (MSByte)	Byte#1	Byte#2	Byte#3 (LSByte)

Figure 4–36. Big Endian Illustration chart

Byte#3 (MSByte)	Byte#2	Byte#1	Byte#0 (LSByte)

Figure 4–37. Little Endian Illustration chart

Since the TSB12LV41A's microprocessor interface is only 8 or 16 bits wide, but the internal configuration registers are 32 bits wide, a byte stacking (for writes) and a byte un-stacking (for reads) operation must be performed on the data bus. For little endian processors the TSB12LV41A can perform the swapping of bytes on the data bus required to allow both the processor and the TSB12LV41A to interpret the data the same. There are two methods of swapping the data bytes, address invariant and data invariant. Both of these methods are described in the following.

NOTE:

For the host processor to work correctly with the TSB12LV41A, users must correctly connect the address and data busses of their microprocessor to the TSB12LV41A's microprocessor port. Users must connect the MSB (most significant bit) of their address/data bus to the address/data MSB of the TSB12LV41A. This must be done regardless of bit number labeling or which type of endianness their microprocessor uses. For processors of little endian type the correct byte swapping can be done by using the BeCtl (big endian control) and DataInvarnt (data invariant) bits in the IOCR register (at offset 1ECh).

4.3.10.1 Byte Swapping for Little Endian systems

The BeCtl bit in the TSB12LV41A's IOCR register informs the microprocessor interface of the endianness type that is being used. The DataInvarnt bit controls how the write/read data is swapped at the data bus when BeCtl is set to 0. Note that when the BeCtl bit is set to 1 the DataInvarnt bit setting has no affect and data is always interpreted in as Big Endian. The default setting for BeCtl is 1 and for DataInvarnt is 1.

4.3.10.2 Data Invariant System Design

Figure 4–38 shows a little endian data invariant system design example. In this system, the byte addresses are not preserved. Byte_0 in the host microprocessors little endian system contains aa. Byte_0 in TSB12LV41A's big endian system contains dd. In other words, a data invariant design does not preserve the addresses when mapping between endian domains. If the data represents an integer, it is interpreted the same by both systems. If the data represents a string, an array, or some other type of byte indexed structure, it is interpreted differently by both systems.

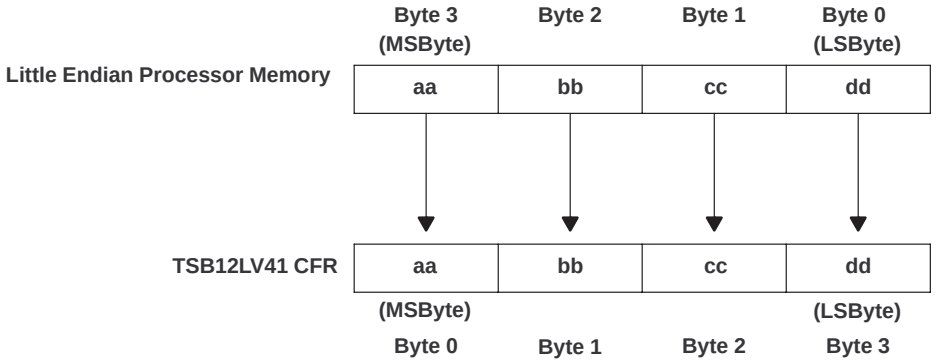


Figure 4–38. Little Endian Data Invariant System Design Illustration Chart

4.3.10.3 Address Invariant System Design

Figure 4–39 shows a little endian address invariant system design example. In this system, the byte address are preserved. Byte_0 in the host microprocessor's little endian system contains dd. Byte_0 in TSB12LV41A's big endian system also contains dd. In other words, an address Invariant design preserves the addresses when mapping between endian domains. If the data represents a string, an array, or some other type of byte indexed structure, it is interpreted the same by both systems. If the data is an integer, it is interpreted differently by both systems.

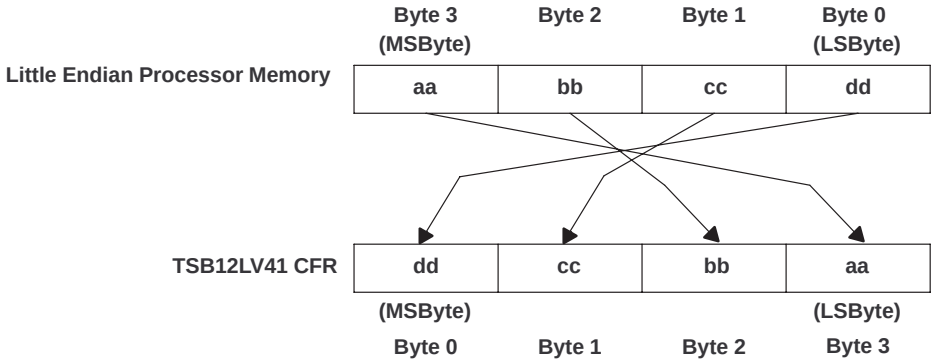


Figure 4–39. Little Endian Address Invariant System Design Illustration Chart

4.3.11 Use of Interrupts with MPEG2Lynx

The interrupts that can be routed to the external INT terminal (89) are available in registers 10h and 18h. Each interrupt has a corresponding mask bit. If the mask bit is disabled (mask bit = 0), then that interrupt is disabled. When the mask bit = 1, then the interrupt is enabled and available for output on the external INT terminal. The mask registers for the interrupt registers (10h and 18h) are 18h and 1Ch, respectively. See Figure 4–40 for the interrupt hierarchy.

NOTE:

Even if an interrupt is masked off, its value in registers 10h and 14h is still valid.

When an interrupt is signaled on the INT terminal, the host should examine the IGRP0 and IGRP1 bits to determine which register (either 10h or 18h) contain the interrupt. The IGRP0 and IGRP2 bits are available in both registers 10h and 18h. Each interrupt can be cleared by writing a 1 to the interrupt bit.

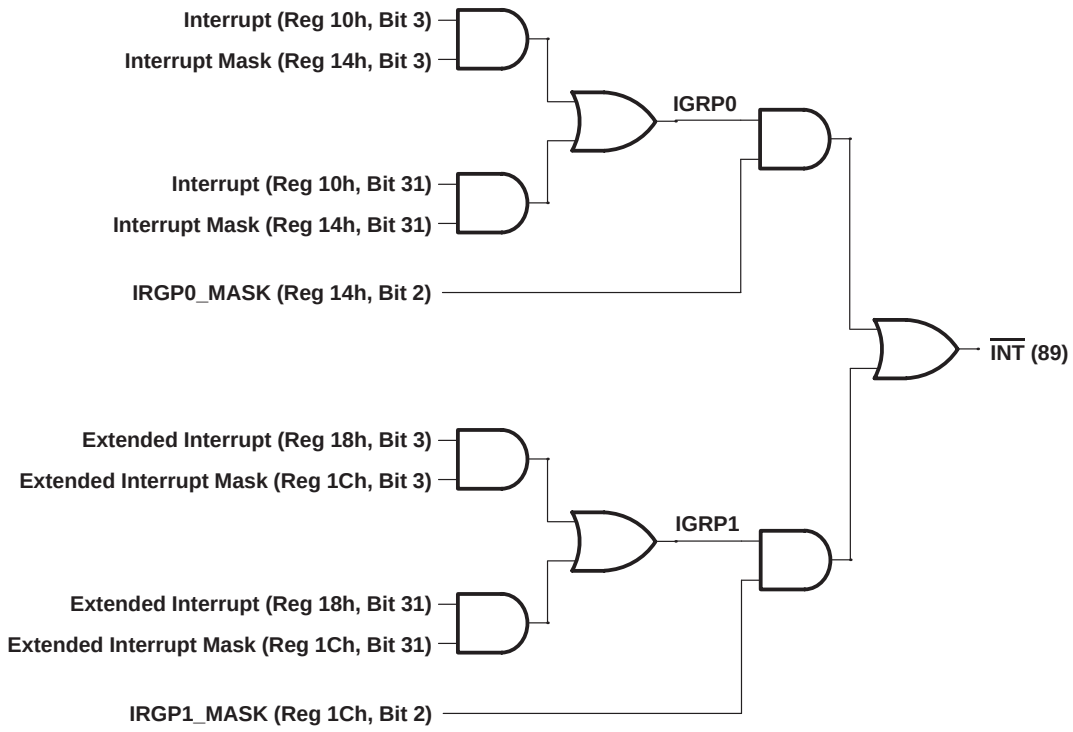


Figure 4–40. Interrupt Hierarchy

4.4 TSB12LV41A to 1394 Phy Interface Specification

4.4.1 Introduction

This chapter provides an overview of a TSB12LV41A to the phy interface. The information that follows can be used as a guide through the process of connecting the TSB12LV41A to a 1394 physical-layer device. The part numbers referenced, the TSB21LV03A and the TSB12LV41A, represent the Texas Instruments implementation of the phy (TSB21LV03A) and link (TSB12LV41A) layers of the IEEE 1394-1995 standard.

The specific details of how the TSB21LV03A device operates is not discussed in this document. Only those parts that relate to the TSB12LV41A phy-link interface are mentioned.

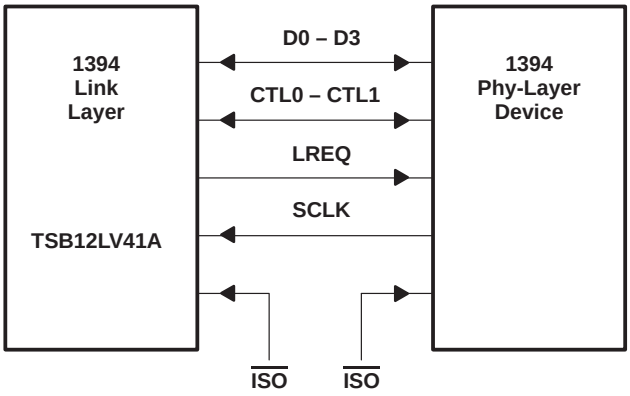
4.4.2 Assumptions

The TSB12LV41A is capable of supporting 100 Mbits/s and 200 Mbits/s phy-layer devices. For that reason, this document describes an interface to a 200-Mbits/s (actually 196.6-Mbits/s) device. To support differential-speed phy layers, adjust the width of the data bus by two terminals per 100 Mbits/s. For example, for 100- and 200-Mbits/s devices, the data bus is 2 and 4 bits wide respectively. The width of the CTL bus and the clock rate between the devices, however, does not change, regardless of the transmission speed that is used.

Finally, the 1394 phy layer has control of all bidirectional terminals that run between the phy layer and TSB12LV41A. The TSB12LV41A can drive these terminals only after it has been given permission by the phy layer. A dedicated request terminal (LREQ) is used by the TSB12LV41A for any activity that the designer wishes to initiate.

4.4.3 Block Diagram

The functional block diagram of the TSB12LV41A to phy layer is shown in Figure 4–41.



NOTE A: See Table 2–2 for signal definition.

Figure 4–41. Functional Block Diagram of the TSB12LV41A to Phy Layer

4.4.4 Operational Overview

The four operations that can occur in the phy-link interface are request, status, transmit, and receive. With the exception of the request operation, all actions are initiated by the phy layer.

The CTL0 – CTL1 bus is encoded as shown in the following sections.

4.4.4.1 Phy Interface Has Control of the Bus

Table 4–10. Phy Interface Control of Bus Functions

CTL0,CTL1	NAME	DESCRIPTION OF ACTIVITY
00	Idle	No activity is occurring (this is the default mode).
01	Status	Status information is being sent from the phy layer to the TSB12LV41A.
10	Receive	An incoming packet is being sent from the phy layer to the TSB12LV41A.
11	Transmit	The TSB12LV41A has been given control of the bus to send an outgoing packet.

4.4.4.2 TSB12LV41A Has Control of the Bus

The TSB12LV41A has control of the bus after receiving permission from the phy layer.

Table 4–11. TSB12LV41A Control of Bus Functions

CTL0, CTL1	NAME	DESCRIPTION OF ACTIVITY
00	Idle	The TSB12LV41A releases the bus (transmission has been completed).
01	Hold	The TSB12LV41A is holding the bus while data is being prepared for transmission, or the TSB12LV41A wants to send another packet without arbitration.
10	Transmit	An outgoing packet is being sent from the TSB12LV41A to phy layer.
11	Reserved	None

4.4.5 Request

A serial stream of information is sent across the LREQ terminal whenever the TSB12LV41A needs to request the bus or access a register that is located in the phy layer. The size of the stream varies depending on whether the transfer is a bus request, a read command, or a write command. Regardless of the type of transfer, a start bit of 1 is required at the beginning of the stream and a stop bit of 0 is required at the end of the stream.

Table 4–12. Request Functions

NUMBER of BITS	NAME
7	Bus request
9	Read register request
17	Write register request

4.4.5.1 LREQ Transfer

Bus Request

Table 4–13. Bus-Request Functions (Length of Stream: 7 Bits)

BIT(S)	NAME	DESCRIPTION
0	Start bit	Start bit indicates the beginning of the transfer (always set).
1–3	Request type	Request type indicates the type of bus request (see Table 7–7 for the encoding of this field).
4–5	Request speed	Request speed indicates the speed at which the phy interface sends the packet for this particular request (see Table 7–8 for the encoding of this field).
6	Stop bit	Stop bit indicates the end of the transfer (always cleared).

Read-Register Request

Table 4–14. Read-Register Request Functions (Length of Stream: 9 Bits)

BIT(S)	NAME	DESCRIPTION
0	Start bit	Start bit indicates the beginning of the transfer (always set).
1–3	Request type	Request type indicates the type of request function (see Table 7–7 for the encoding of this field).
4–7	Address	These bits contain the address of the phy register to be read.
8	Stop bit	Stop bit indicates the end of the transfer (always cleared).

Write-Register Request

Table 4–15. Write-Register Request (Length of Stream: 17 Bits)

BIT(S)	NAME	DESCRIPTION
0	Start bit	Start bit indicates the beginning of the transfer (always set).
1–3	Request type	Request type indicates the type of request (see Table 7–7 for the encoding of this field).
4–7	Address	These bits contain the address of the phy register to be written to.
8–15	Data	These bits contain the data that is to be written to the specified register address.
16	Stop bit	Stop bit indicates the end of the transfer (always cleared).

Request-Type Field for Request

Table 4–16. TSB12LV41A Request Functions

LREQ1 – LREQ3	NAME	DESCRIPTION
000	TakeBus	Immediate request. Upon detection of an idle, take control of the bus immediately (no arbitration) for asynchronous packet ACK response.
001	IsoReq	Isochronous request. IsoReq arbitrates for control of the bus after an isochronous gap.
010	PriReq	Priority request. PriReq arbitrates for control of the bus after a fair gap and ignores fair protocol.
011	FairReq	Fair request. FairReq arbitrates for control of the bus after a fair gap and uses fair protocol.
100	RdReg	Read request. RdReg returns the specified register contents through a status transfer.
101	WrReg	Write request. WrReg writes to the specified register.
110, 111	Reserved	Reserved

Request-Speed Field for Request

Table 4–17. Request-Speed Functions

LREQ4, LREQ5	DATA RATE
00	100 Mbits/s
01	200 Mbits/s
10	400 Mbits/s
11	Reserved

4.4.5.2 Bus Request

For fair or priority access, the TSB12LV41A requests control of the bus at least one clock after the TSB12LV41A/phy interface becomes idle CTL0 – CTL1 = 00 indicates that the physical layer is in an idle state. If the TSB12LV41A senses that CTL0 – CTL1 = 10, then it knows that its request has been lost. This is true any time during or after the TSB12LV41A sends the bus request transfer. Additionally, the phy interface ignores any fair or priority requests when it asserts the receive state while the TSB12LV41A is requesting the bus. The link then reissues the request one clock after the next interface idle.

The cycle master uses a normal priority request to send a cycle-start message. After receiving a cycle start, the TSB12LV41A can issue an isochronous bus request. When arbitration is won, the TSB12LV41A proceeds with the isochronous transfer of data. The isochronous request is cleared in the phy interface once the TSB12LV41A sends another type of request or when the isochronous transfer has been completed.

The TakeBus request is issued when the TSB12LV41A needs to send an acknowledgment after reception of a packet addressed to it. This request must be issued during packet reception. This is done to minimize the delay times that a phy interface would have to wait between the end of a packet reception and the transmittal of an acknowledgment. As soon as the packet ends, the phy interface immediately grants access of the bus to the TSB12LV41A. The TSB12LV41A sends an acknowledgment to the sender unless the header CRC of the packet turns out to be invalid. In this case, the TSB12LV41A releases the bus immediately; it is not allowed to send another type of packet on this grant. To ensure this, the TSB12LV41A is forced to wait 160 ns after the end of the packet is received. The phy interface then gains control of the bus and the acknowledge with the CRC error sent. The bus is then released and allowed to proceed with another request.

Although highly improbable, it is conceivable that two separate nodes believe that an incoming packet is intended for them. The nodes then issue a TakeBus request before checking the CRC of the packet. Since both phys seize control of the bus at the same time, a temporary, localized collision of the bus occurs

somewhere between the competing nodes. This collision would be interpreted by the other nodes on the network as being a ZZ line state, not a bus reset. As soon as the two nodes check the CRC, the mistaken node drops its request and the false line state is removed. The only side effect is the loss of the intended acknowledgment packet (this is handled by the higher layer protocol).

4.4.5.3 Read/Write Requests

When the TSB12LV41A requests to read the specified register contents, the phy interface sends the contents of the register to the TSB12LV41A through a status transfer. When an incoming packet is received while the phy interface is transferring status information to the TSB12LV41A, the phy interface continues to attempt to transfer the contents of the register until it is successful.

For write requests, the phy interface loads the data field into the appropriately addressed register as soon as the transfer has been completed. The TSB12LV41A is allowed to request read or write operations at any time.

See Section 4.4.6, for a more detailed description of the status transfer.

4.4.6 Status

A status transfer is initiated by the phy interface when it has some status information to transfer to the TSB12LV41A. The transfer is initiated by asserting the following: CTL0 – CTL1 = 01 and D0 – D1 are used to transmit the status data; see Table 4–18 for status-request functions. D2 – D3 are not used for status transfers.

The status transfer can be interrupted by an incoming packet from another node. When this occurs, the phy interface attempts to resend the status information after the packet has been acted upon. The phy interface continues to attempt to complete the transfer until the information has been successfully transmitted.

NOTE:
There must be at least one idle cycle between consecutive status transfers.

4.4.6.1 Status Request

The definition of the bits in the status transfer is shown in Table 4–18.

Table 4–18. Status-Request Functions (Length of Stream: 16 Bits)

BIT(s)	NAME	DESCRIPTION
0	Arbitration reset gap	The arbitration-reset gap bit indicates that the phy interface has detected that the bus has been idle for an arbitration reset gap time (this time is defined in the IEEE 1394-1995 standard). This bit is used by the TSB12LV41A in its busy/retry state machine.
1	Fair gap	The fair-gap bit indicates that the phy interface has detected that the bus has been idle for a fair-gap time (this time is defined in the IEEE 1394-1995 standard). This bit is used by the TSB12LV41A to detect the completion of an isochronous cycle.
2	Bus reset	The bus reset bit indicates that the phy interface has entered the bus reset state.
3	phy interrupt	The phy interrupt bit indicates that the phy interface is requesting an interrupt to the host.
4–7	Address	The address bits hold the address of the phy register whose contents are transferred to the TSB12LV41A.
8–15	Data	The data bits hold the data that is to be sent to the TSB12LV41A.

Normally, the phy interface sends just the first four bits of data to the TSB12LV41A. These bits are used by the TSB12LV41A state machine. However, if the TSB12LV41A initiates a read request (through a request transfer), then the phy interface sends the entire status packet to the TSB12LV41A. Additionally, the phy interface sends the contents of the register to the TSB12LV41A when it has some important information to pass on. Currently, the only condition where this occurs is after the self-identification process when the phy interface needs to inform the TSB12LV41A of its new node address (physical ID register).

There may be times when the phy interface wants to start a second status transfer. The phy interface first has to wait at least one clock cycle with the CTL lines idle before it can begin a second transfer.

4.4.6.2 Transmit

When the TSB12LV41A wants to transmit information, it first requests access to the bus through an LREQ signal. Once the phy interface receives this request, it arbitrates to gain control of the bus. When the phy interface wins ownership of the serial bus, it grants the bus to the TSB12LV41A by asserting the transmit state on the CTL terminals for at least one SCLK cycle. The TSB12LV41A takes control of the bus by asserting either hold or transmit on the CTL lines. Hold is used by the TSB12LV41A to keep control of the bus when it needs some time to prepare the data for transmission. The phy interface keeps control of the bus for the TSB12LV41A by asserting a data-on state on the bus. It is not necessary for the TSB12LV41A to use hold when it is ready to transmit as soon as bus ownership is granted.

When the TSB12LV41A is prepared to send data, it asserts transmit on the CTL lines as well as sends the first bits of the packet on the D0 – D3 lines (assuming 200 Mbits/s). The transmit state is held on the CTL terminals until the last bits of data have been sent. The TSB12LV41A then asserts idle on the CTL lines for one clock cycle after which it releases control of the interface.

However, there are times when the TSB12LV41A needs to send another packet without releasing the bus. For example, the TSB12LV41A may want to send consecutive isochronous packets or it may want to attach a response to an acknowledgment. To do this, the TSB12LV41A asserts hold instead of idle when the first packet of data has been completely transmitted. Hold, in this case, informs the phy interface that the TSB12LV41A needs to send another packet without releasing control of the bus. The phy interface then waits a set amount of time before asserting transmit. The TSB12LV41A can then proceed with the transmittal of the second packet. After all data has been transmitted and the TSB12LV41A has asserted idle on the CTL terminals, the phy interface asserts its own idle state on the CTL lines. When sending multiple packets in this fashion, it is required that all data be transmitted at the same speed. This is required because the transmission speed is set during arbitration, and since the arbitration step is skipped, there is no way of informing the network of a change in speed.

4.4.6.3 Receive

When data is received by the phy interface from the serial bus, it transfers the data to the TSB12LV41A for further processing. The phy interface asserts receive on the CTL lines and is set to 1 on each D terminal. The phy interface indicates the start of the packet by placing the speed code on the data bus (see the following note). The phy interface then proceeds with the transmittal of the packet to the TSB12LV41A on the D lines while still keeping the receive status on the CTL terminals. Once the packet has been completely transferred, the phy interface asserts idle on the CTL terminals that completes the receive operation.

NOTE:

The speed code sent is a phy-TSB12LV41A protocol and not included in the packets CRC calculation.

SPD = Speed code
D0 => Dn = Packet data

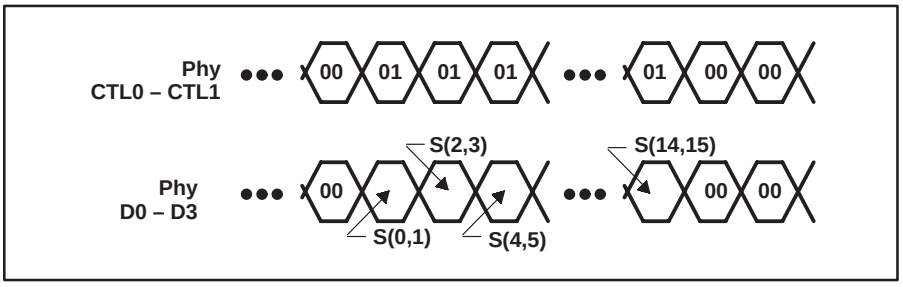
Table 4–19. Speed Code for Receive

D0 – D3	DATA RATE
00xx†	100 Mbits/s
0100†	200 Mbits/s
0101	400 Mbits/s
111111111	Data-on indication

† The x means transmitted as 0 and ignored by phy layer.

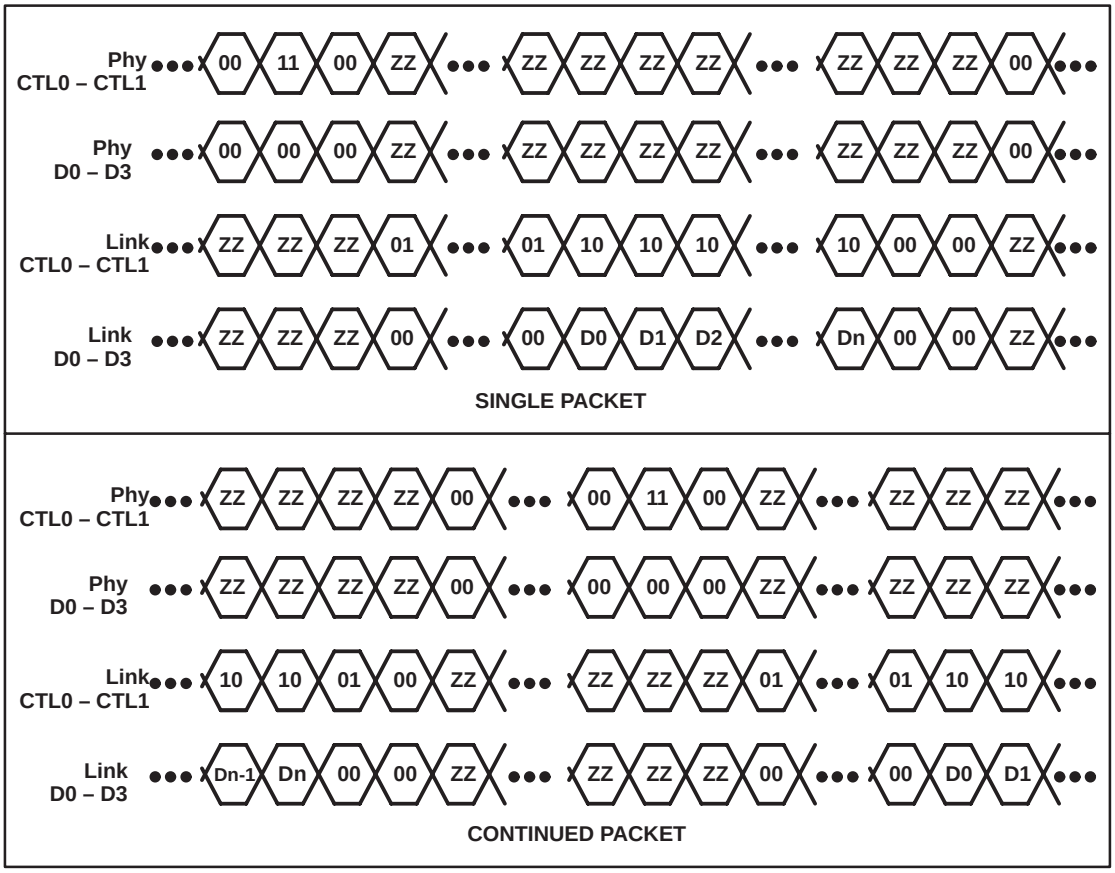
4.4.7 TSB12LV41A to Phy Bus Timing

Figure 4–42. LREQ Timing



NOTE A: Each cell represents one SCLK sample time.

Figure 4–43. Status-Transfer Timing



NOTE A: ZZ = high-impedance state, D0 – Dn = packet data

Figure 4–44. Transmit Timing

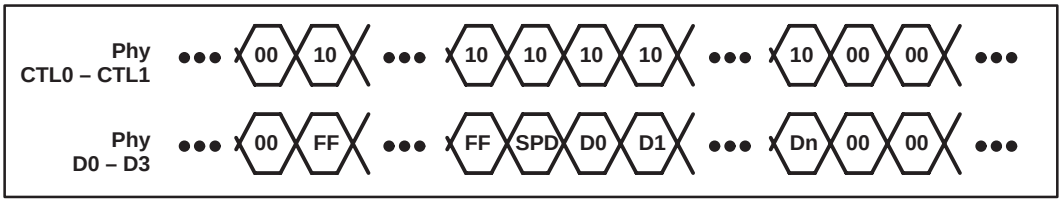


Figure 4–45. Receiver Timing

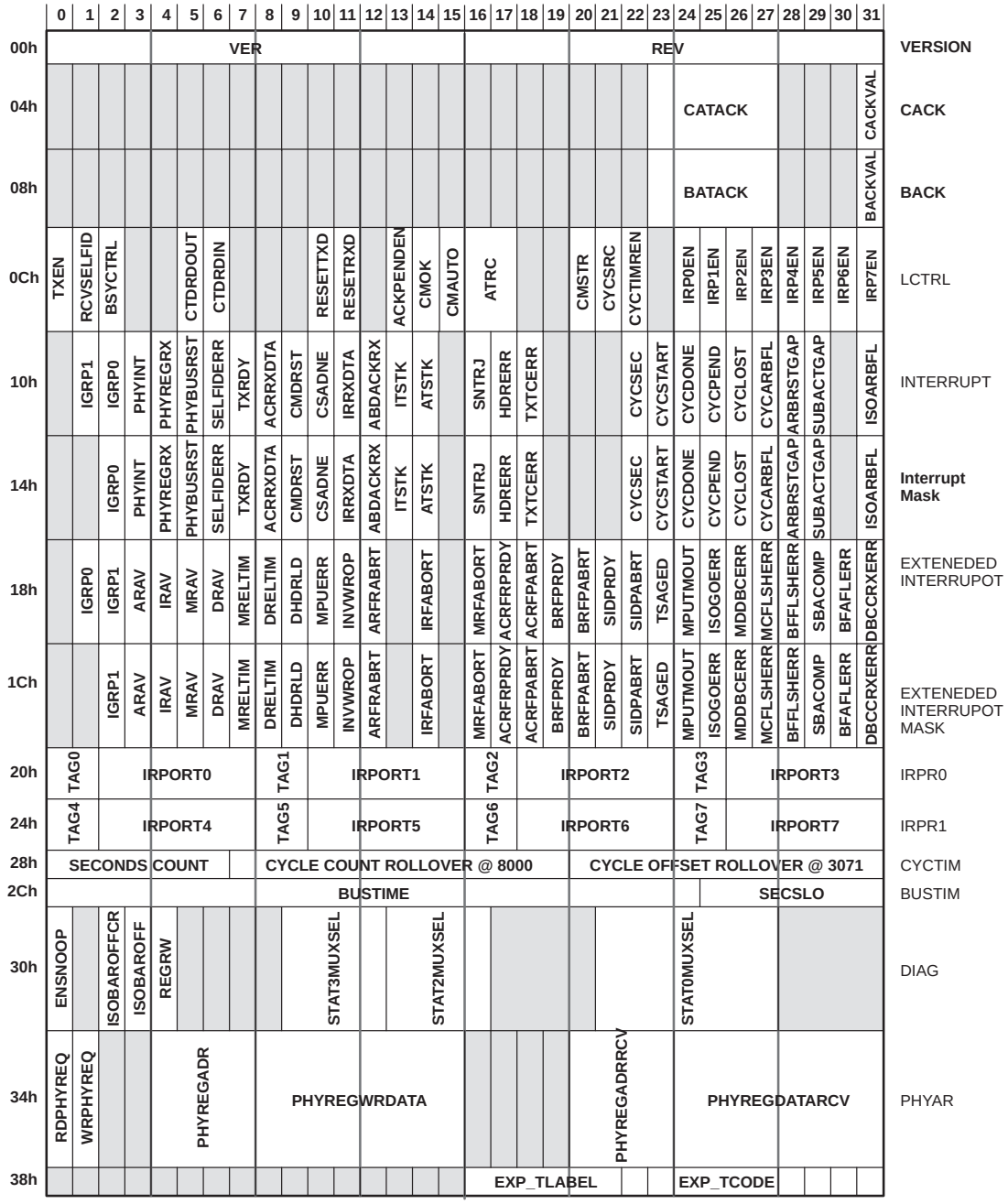
5 Detailed Operation and Programmers Reference

5.1 TSB12LV41A Configuration Register

The TSB12LV41A CFR map is shown in Figure 5–1.

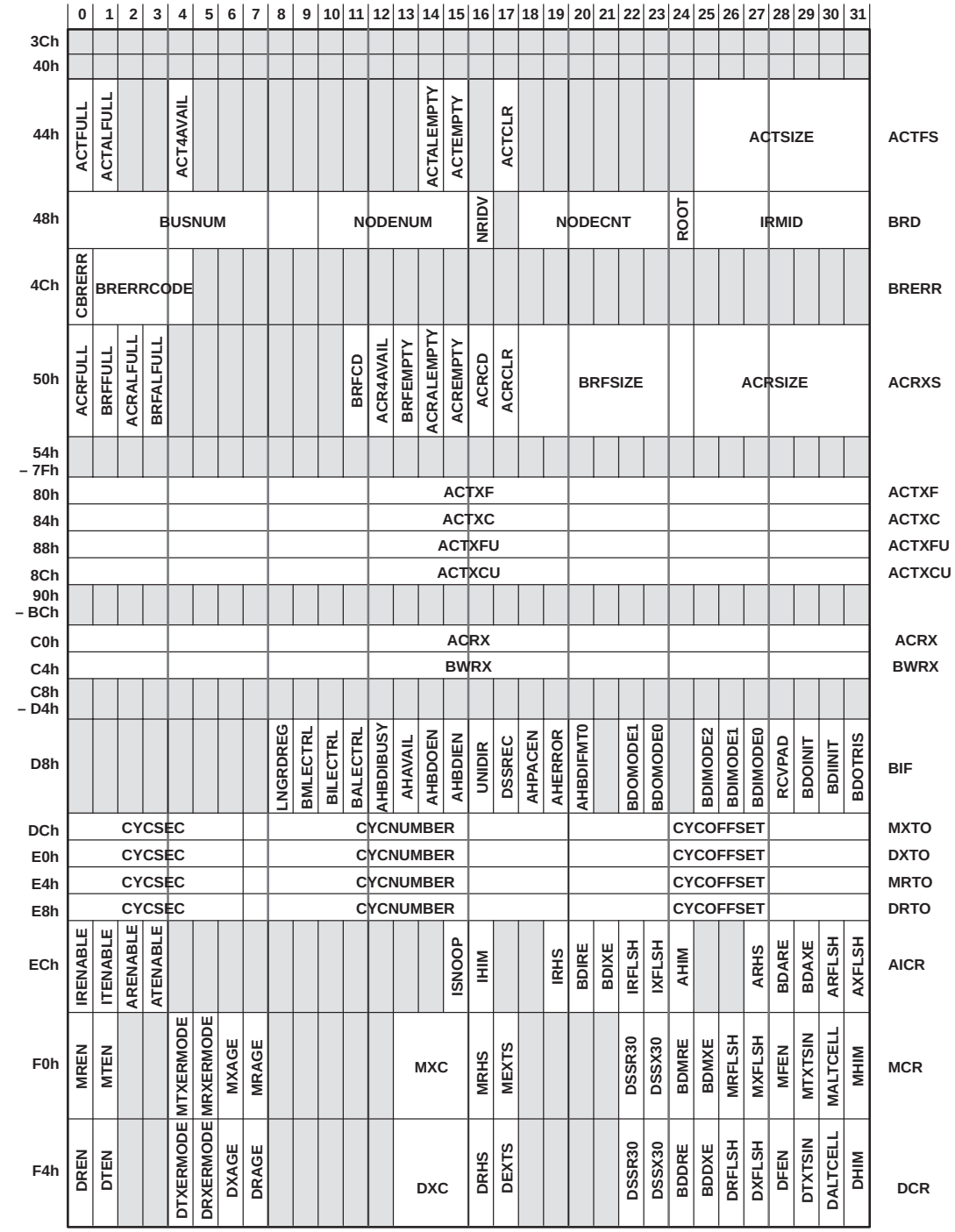
NOTE:
The following register pairs are aliased (are the same):

MXT0 (address DCh)	=	DXT0 (address E0h)
MRT0 (address E4h)	=	DRT0 (address E8h)
MCR (address F0h)	=	DCR (address F4h)
BMSZ (address 150h)	=	MDSZ (address 158h)
BMAVAL (address 154h)	=	BDAVAL (address 15Ch)
BMTXFC (address 160h)	=	BDTXFC (address 16Ch)
BMTXLS (address 164h)	=	BDTXLX (address 170h)
BMRX (address 168h)	=	BDRX (address 174h)
MRH (address 178h)	=	DRH (address 188h)
MCIPR0 (address 17Ch)	=	DCIPR0 (address 18Ch)
MCIPR1 (address 180h)	=	DCIPR1 (address 290h)
MRT (address 184h)	=	DRT (address 194h)
MXH (address 1C8h)	=	DXH (address 1D4h)
MCIPX0 (address 1CCh)	=	DCIPX0 (address 1D8h)
MCIPX1 (address 1D0h)	=	DCIPX1 (address 1DCh)



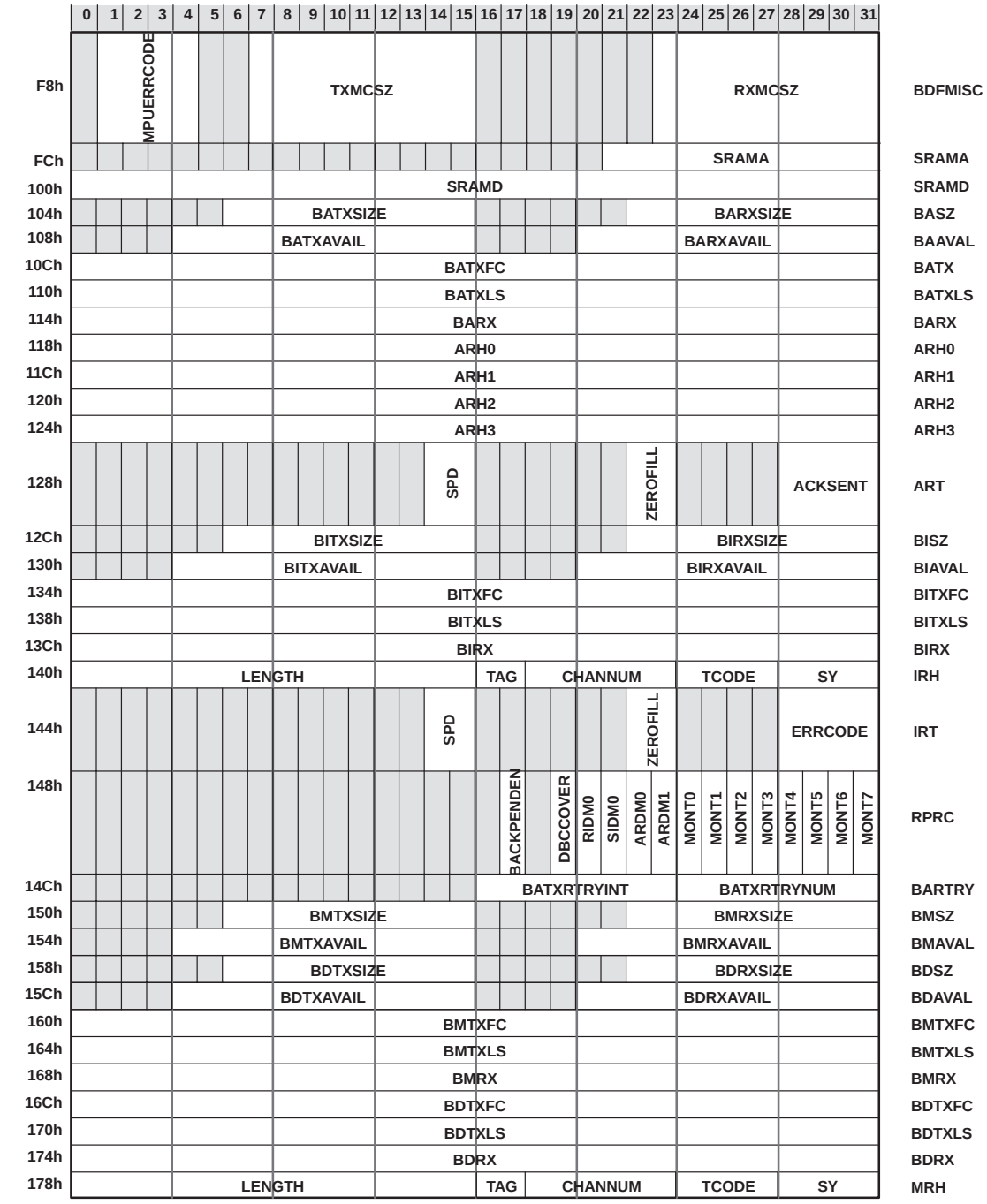
NOTE A: All gray areas (bits) are reserved bits.

Figure 5–1. Configuration Register (CFR) Map



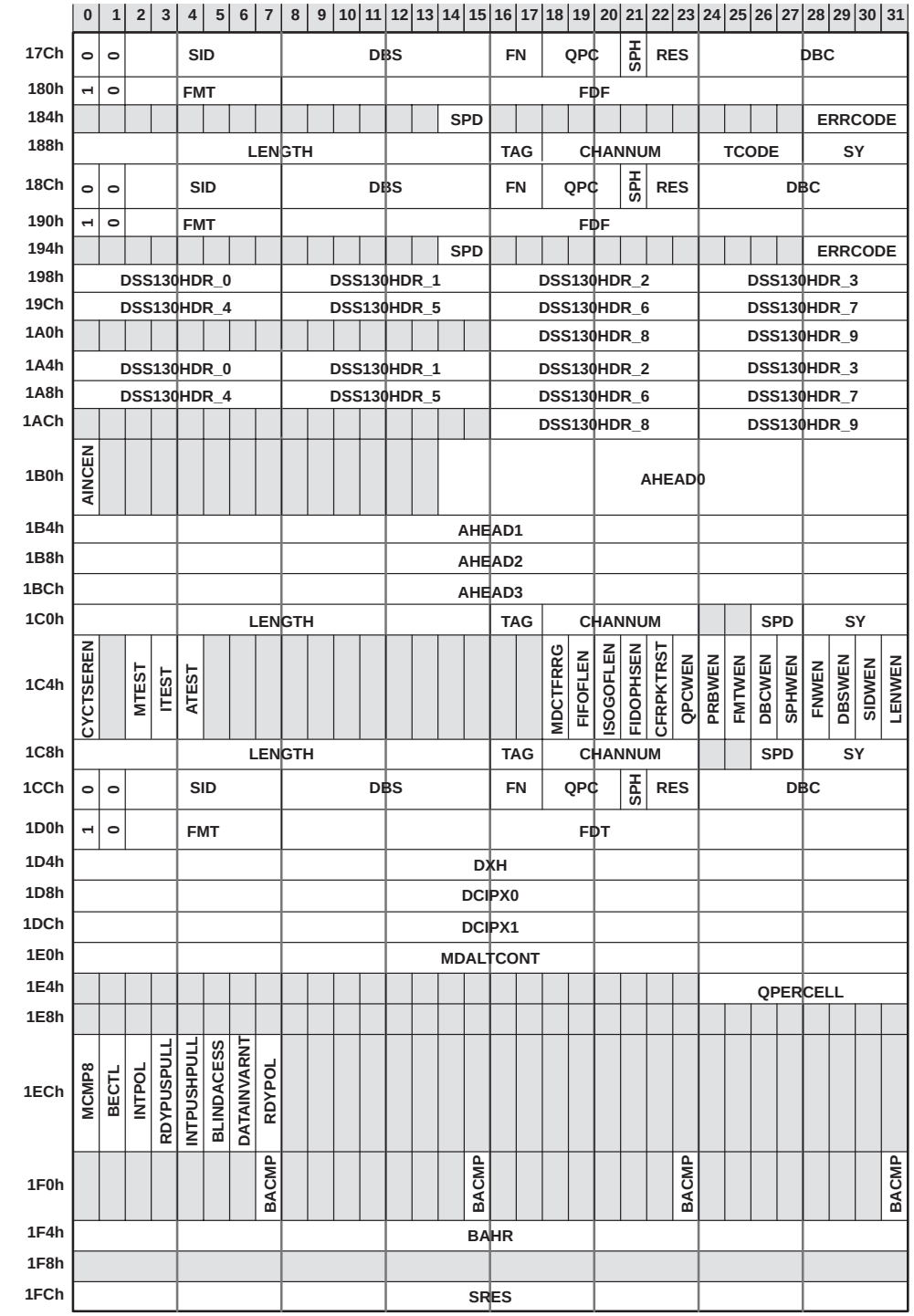
NOTE A: All gray areas (bits) are reserved bits.

Figure 5–1. Configuration Register (CFR) Map (continued)



NOTE A: All gray areas (bits) are reserved bits.

Figure 5–1. Configuration Register (CFR) Map (continued)



NOTE A: All gray areas (bits) are reserved bits.

Figure 5–1. Configuration Register (CFR) Map (continued)

5.2 Version Register (VERS @ Addr 0h)

This address port provides the application software with the version number and revision number of the MPEG2Lynx device. These numbers are hardwired in the logic of the device.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	VER	R	Version number. Hardwired value = 0071h
16 – 31	REV	R	Revision number. Hardwired value = 1539h

5.3 C Acknowledge Register (CACK @ Addr 4h)

This register provides the application software with the last acknowledge that was received for an asychronous packet transmitted from the asychronous transmit control FIFO. Unless otherwise specified, the bits in this register are cleared to 0 on power up or software initiated reset.

<table>
<tr><th>BIT NUMBER</th><th>BIT NAME</th><th>DIR</th><th colspan="2">FUNCTIONAL DESCRIPTION</th></tr>
<tr><td>00 – 22</td><td>Not used</td><td>R</td><td colspan="2"></td></tr>
<tr><td rowspan="6">23 – 27</td><td rowspan="6">CATACK</td><td rowspan="6">R</td><td colspan="2">Control transmit FIFO acknowledge. The acknowledge value received from the link transmitter for a packet that was transmitted from the asychronous control transmit FIFO. These bits can be written to by setting the REGRW bit in the diagnostic test register (DIAG @ Addr 30h) to 1.</td></tr>
<tr><td>CATACK[23]</td><td>CATACK[24:27]</td></tr>
<tr><td>0</td><td>Normal 1394 4-bit ack code.</td></tr>
<tr><td>1</td><td>0000 – No Ack received. Ack timeout</td></tr>
<tr><td>1</td><td>0001 – Ack pkt longer than 8 bits</td></tr>
<tr><td>1</td><td>0010 – Ack pkt shorter than 8 bit</td></tr>
<tr><td>28 – 30</td><td>Not used</td><td>R</td><td colspan="2"></td></tr>
<tr><td>31</td><td>CACKVAL</td><td>R</td><td colspan="2">CATACK valid. This bit is set to 1 to indicate that the value of CATACK[23:37] has been updated with a new value. This bit is cleared to 0 when the application software reads this register to obtain the value of CATACK and CACKVAL. This bit can be written to by setting the REGRW bit in the diagnostic test register (DIAG @ Addr 30h) to 1.</td></tr>
</table>

5.4 B Acknowledge Register (BACK @ Addr 8h)

This register provides the application software with the last acknowledge that was received for an asychronous packet transmitted from the asychronous transmit bulky FIFO. Unless otherwise specified, the bits in this register are cleared to 0 on power up or software initiated reset.

<table>
<tr><th>BIT NUMBER</th><th>BIT NAME</th><th>DIR</th><th colspan="2">FUNCTIONAL DESCRIPTION</th></tr>
<tr><td>00 – 22</td><td>Not used</td><td>R</td><td colspan="2"></td></tr>
<tr><td rowspan="6">23 – 27</td><td rowspan="6">BATACK</td><td rowspan="6">R</td><td colspan="2">Bulky transmit FIFO acknowledge. The acknowledge value received from the link transmitter for a packet that was transmitted from the asychronous bulky transmit FIFO. These bits can be written to by setting the REGRW bit in the diagnostic test register (DIAG @ Addr 30h) to 1.</td></tr>
<tr><td>BATACK[23]</td><td>BATACK[24:27]</td></tr>
<tr><td>0</td><td>Normal 1394 4 bit ack code.</td></tr>
<tr><td>1</td><td>0000 – No Ack received. Ack timeout</td></tr>
<tr><td>1</td><td>0001 – Ack pkt longer than 8 bits</td></tr>
<tr><td>1</td><td>0010 – Ack pkt shorter than 8 bit</td></tr>
<tr><td>28 – 30</td><td>Not used</td><td>R</td><td colspan="2"></td></tr>
<tr><td>31</td><td>BACKVAL</td><td>R</td><td colspan="2">BATACK valid. This bit is set to 1 to indicate that BATACK [23:27] has been updated with a new value. This bit is cleared to 0 when the application software reads this register to obtain the value of BATACK and BACKVAL. This bit can be written to by setting the REGRW bit in the diagnostic test register (DIAG @ Addr 30h) to 1.</td></tr>
</table>

5.5 Link Control Register (LCTRL @ Addr Ch)

This register provides the application software with the capability to control and configure the operation of the 1394 LLC logic. Unless otherwise specified, the bits in this register are cleared to 0 on power up or software initiated reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	TXEN	R/W	Transmit enable. When set to 1, this bit enables the link transmitter to begin bus arbitration and packet transmission. When this bit is set to 0, the operation of the link transmitter is disabled from operating.
01	RCVSELFID	R/W	Receive Self-ID enable. When set to 1 this bit enables the LLC to receive Self-ID packets during 1394 bus initialization.
02	BSYCTRL	R/W	Transaction layer busy override. When this bit is set to 1, the link receiver will unconditionally busy off all acknowledgeable incoming packets with BUSY_X. When this bit is set to 0, the link receiver uses the availability of the selected receiving FIFO to determine if an incoming acknowledgeable packet is to be accepted or busy off (BUSY_X).
03 – 04	Not used		
05	CTDRDOUT	R/W	Contender data out. This bit is used to set the logic value of the external terminal CONTENDER when it has been programmed to operate in output mode (see CTDRDIN, Reg C, bit 6)
			DESCRIPTION
			0 = Link is not a contender for isochronous resource manager.
			1 = Link is a contender for isochronous resource manager.
06	CTDRDIN	R/W	Contender data direction control. This bit is used in setting the direction of the external contender terminal. This terminal is set to 1 on power up or software reset.
			DESCRIPTION
			0 = CONTENDER terminal is in output mode and is driven by the value of CTDRDOUT
			1 = CONTENDER terminal is in input mode.
07 – 09	Not used		
10	RESETTXD	R/W	Reset link transmitter. Writing 1 to this bit resets all state machines in the link layer that are involved in transmitting a packet. This bit is self clearing.
11	RESETRXD	R/W	Reset link receiver. Writing 1 to this bit resets all state machines in the LLC that are involved in the reception of a packet. This bit is self clearing.
12	Not used		
13	ACKPENDEN	R/W	Ack pending enabled. When set to 1, this bit enables the link receiver to acknowledge write and lock request packets with an ack pending code of (2h). If this bit is set to 0, the receiver uses the ack_complete code of (1h). This bit is set to a 1 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
14	CMOK	R/W	When CMAUTO is set to 1 and CMOK is set to 1, the CMSTR and CYCTIMREN bits are automatically set to 1 if the root bit of the attached Phy is set to 1 following a bus reset. CMOK is automatically reset to 0 if the attached Phy is not root following a bus reset.
15	CMAUTO	R/W	When CMAUTO is set to 1 and CMOK is set to 1, the CMSTR and CYCTIMREN bits are automatically set to 1 if the root bit is 1 following a bus reset. CMAUTO is unaffected by bus reset.
16 – 17	ATRC	R/W	Asychronous transmit retry code. This code is logically ORed with the retry code field (00) in the transmit packet, and the packet is resent.
			DESCRIPTION
			00 = retry_O (new)
			01 = retry_X
18 – 19	Not used		
20	CMSTR	R/W	Cycle master. When this bit is set to 1 and the attached Phy is the root, the cyclemaster function is enabled for transmitting cycle start packets. If CMAUTO is set to 1, and if the attached Phy is root and the CMOK bit is set to 1, then this bit is set to 1 automatically and is read only. If CMAUTO is set to 0, this bit can be read and written to by the microprocessor. Also, every bus reset resets this bit and software is responsible for re-enabling it.
21	CYCSRC	R/W	When CYCSRC is set to 1, the cycle_count field of the cycletimer increments and the cycle_offset field of the cycle timer resets for each positive transition of CYCLEIN. When CYCSRC is set to 0, the cycle_count field increments when the cycle_offset field rolls over.
22	CYCTIMREN	R/W	Cycle timer enable. The cycle timer is enabled when this bit is set to 1. The cycle timer is disabled when the bit is set to 0.
23	Not used		
24	IRP0EN (MPEG/DSS)	R/W	Isochronous receive port comparator enable. When this bit is set to 1, the port 0 MPEG/DSS packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to an expected value. The packet is received if the comparator detects a match.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
25	IRP1EN	R/W	Isochronous receive port comparator enable. When this bit is set to 1, the port 1 ISO packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to an expected value. The packet is received if the comparator detects a match.
26	IRP2EN	R/W	Isochronous receive port comparator enable. When this bit is set to 1, the port 2 ISO packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to an expected value. The packet is received if the comparator detects a match.
27	IRP3EN	R/W	Isochronous receive port comparator enable. When this bit is set to 1, the port 3 ISO packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to an expected value. The packet is received if the comparator detects a match.
28	IRP4EN	R/W	Isochronous receive port comparator enable. When this bit is set to 1, the port 4 ISO packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to an expected value. The packet is received if the comparator detects a match.
29	IRP5EN	R/W	Isochronous receive port comparator enable. When this bit is set to 1, the port 5 ISO packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to an expected value. The packet is received if the comparator detects a match.
30	IRP6EN	R/W	Isochronous receive port comparator enable. When this bit is set to 1, the port 6 ISO packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to an expected value. The packet is received if the comparator detects a match.
31	IRP7EN	R/W	Isochronous receive port comparator enable. When this bit is set to 1, the port 7 ISO packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to an expected value. The packet is received if the comparator detects a match.

5.6 Interrupt Register/Interrupt Mask Register (IR @ Addr 10h/14h)

The interrupt and interrupt mask registers define the group 0 interrupt status bits. The bits in this register can be cleared to 0 by writing 1 to their corresponding bit. Unless otherwise specified, the bits in this register are cleared to 0 on a power up or software reset. With the exception of bit 2, the bits in this register can be placed in a special test mode where the software can directly write to and/or read this register. This is done by setting the REGRW bit in the diagnostic test register (DIAG @ Addr 30h) to 1. The interrupt register is at 10h and the interrupt mask register is at 14h. All the bits in the interrupt mask register are cleared to 0 on power up. A specific interrupt can be masked off when the corresponding bit in the interrupt masks register is 0.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	Not used		
01	IGRP1	R	Addr 10h. When this bit is set, one or more of the extended interrupt bits (reg 18h) is set. Addr 14h, reserved.
02	IGRP0	R	Addr 10h. When this bit is set to 1, one or more of the interrupt bits (reg 10h) is set. Addr 14h This bit determines if the interrupt register is routed to the external $\overline{\text{INT}}$ terminal. Please note that it is possible for both the interrupt register and extended interrupt register to both be routed to the external $\overline{\text{INT}}$ terminal.
03	PHYINT	R/W	Phy interrupt. When this bit is set to 1, either an internal timeout has occured of the cable power has dropped. The error can be decoded in the Phy Access Register (Addr 34h).
04	PHYREGRX	R/W	Phy register data received. When this bit is set to 1, a register value has been transferred to the Phy access register (@ offset 34h) from the Phy interface.
05	PHYBUSRST	R/W	Phy bus reset. When this bit is set to 1, the Phy has entered the 1394 bus reset state.
06	SELFIDERR	R/W	Self-ID error. When this bit is set to 1, this bit indicates that a Self-ID quadlet/packet with errors has been received.
07	TXRDY	R/W	Transmitter ready. When this bit is set to 1, the link transmitter is IDLE and ready to start transmitting.
08	ACRRXDTA	R/W	Asychronous packet received. When this bit is set to 1, the link receiver has confirmed asynchronous data.
09	CMDRST	R/W	Command reset received. When this bit is set to 1, the receiver has been sent a quadlet write addressed to the reset_start CSR register.
10	CSADNE	R/W	Asychronous control FIFO ack received. When this bit is set to 1, an ack has been received for a packet that was transmitted from the asychronous control transmit FIFO.
11	IRRXDTA	R/W	Isochronous packet received. When this bit is set to 1, the link receiver has confirmed isochronous data. This bit gets set when the last quadlet is received into the bulky isochronous receive FIFO.
12	ABDACKRX	R/W	Asychronous bulky FIFO ack received. When this bit is set to 1, an ack has been received for a packet that was transmitted from the asychronous bulky transmit FIFO.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
13	ITSTK	R/W	Isochronous transmit FIFO stuck. When this bit is set to 1, the link transmitter has detected incomplete packet at the isochronous transmit-FIFO interface. To recover from this error, flush bulky isochronous transmit to recover .
14	ATSTK	R/W	Asychronous transmit FIFO stuck. When this bit is set to 1, the link transmitter has detected imcomplete packet at the asychronous transmit-FIFO interface. When this condition occurs Flush the bulky asychronous transmit FIFO and the asychronous control transmit FIFO.
15	Not used		
16	SNTRJ	R/W	Busy acknowledge sent by link receiver. When this bit is set to 1, the link receiver was forced to busy off the incoming packet addressed to this node because the selected receiving FIFO did not have enough space available to store it.
17	HDRERR	R/W	Header error detected. This bit is set to 1 when the receiver detects a CRC error on a packet that may have been addressed to this node.
18	TXTCERR	R/W	Transmit Tcode error. When this bit is set to 1, the link transmitter detected an invalid Tcode in the packet header at the transmit-FIFO interface. To recover from this error flush the bulky asychronous transmit FIFO and the asychronous control transmit FIFO and reset the link transmitter (RESET TxD, reg Ch).
19 – 21	Not used		
22	CYCSEC	R/W	Cycle seconds. When set to to 1, the cycle seconds field in the cycle timer register has incremented. This occurs approximately every second when the cycle timer is enabled.
23	CYCSTART	R/W	Cycle started. When this bit is set to 1, the link transmitter has sent or the link receiver has received a cycle start packet.
24	CYCDONE	R/W	Cycle done. When this bit is set to 1, a subaction gap has been detected on the bus after the transmission or reception of a cycle start packet and any isochronous data packets that were transmitted or received. CYCDONE indicates that the isochronous bus period is over.
25	CYCPEND	R/W	Cycle pending. If the MPEG2Lynx is the cycle master, then CYCPEND is set to 1 when the cycle number field of its cycle timer is incremented. It remains set until a subaction gap is detected. If the MPEG2Lynx is not the cycle master, then CYCPEND is set to 1 when the cycle number field of its cycle timer is incremented or when a cycle start packet was received. It remains set until a subaction gap is detected.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
26	CYCLOST	R/W	Cycle lost. When this bit is set to 1, the cycle timer has rolled over twice with out the reception of a cycle start packet. This occurs only when this node is not the cycle master.
27	CYCARBFL	R/W	Cycle arbitration failed. When this bit is set to 1, the priority transmit request to send the cycle start packet failed to win bus arbitration.
28	ARBRSTGAP	R/W	Arbitration reset gap. When this bit is set to 1, the link has detected that an arbitration reset gap has opened up on the 1394 bus.
29	SUBACTGAP	R/W	Subaction gap. When this bit is set to 1, the link has detected that a subaction gap has opened up on the bus.
30	Not used		
31	ISOARBFL	R/W	Isochronous arbitration failed. When this bit is set to 1, the isochronous transmit request to send an isochronous packet failed to win bus arbitration.

5.7 Extended Interrupt Register/Extended Interrupt Mask Register (EIR @ Addr 18h/1Ch)

The extended interrupt register and the extended interrupt mask register define the group 1 interrupt status bits. With the exception of bit 2, the bits in this register can be cleared to 0 by writing 1 to their corresponding bit. Unless otherwise specified, the bits in this register are cleared to 0 on a power up or software reset. With the exception of bit 1, the bits defined in this register can be placed in a special test mode where the software can directly write to and/or read this register. This is done by setting the REGRW bit in the diagnostic test register (DIAG @ Addr 30h) to 1. The extended interrupt register is at 18h and the extended interrupt mask register is at 1Ch. All bits in the extended interrupt mask register are cleared to 0 on power up. A specific interrupt can be masked off when the corresponding bit in the extended interrupt mask register is cleared to 0.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	Not used		
01	IGRP0	R	Addr 10h When this bit is set to 1, one or more interrupt bits (reg 10h) or interrupt mask bits (reg 14h) are set. Addr 14h, Reserved.
02	IGRP1	R	Addr 18h When this bit is set to 1, one or more of the extended interrupt bits (reg 18h) is set. Addr 1Ch. This bit determines if the external interrupt is routed to the external $\overline{\text{INT}}$ terminal. Please note that reg 10h and reg 18h can both be routed to the external $\overline{\text{INT}}$ terminal.
03	ARAV	R/W	When this bit is set to 1, a complete asychronous packet has been received into the asychronous bulky receive FIFO and the packet header information has been copied into the asychronous receive packet header registers.
04	IRAV	R/W	When this bit is set to 1, a complete isochronous packet has been received into the isochronous bulky receive FIFO and the packet header information has been copied into the isochronous receive packet header register.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
05	MRAV	R/W	When this bit is set to 1, a complete MPEG packet has been received into the MPEG bulky receive FIFO and the packet header information has been copied into the MPEG receive packet header registers.
06	DRAV	R/W	When this bit is set to 1, a complete DSS packet has been received into the DSS bulky receive FIFO and the packet header information has been copied into the DSS receive packet header registers.
07	MRELTIM	R/W	When this bit is set to 1, the cycletimer has reached the value of the MPEG release time and an MCELL is now being released to the selected interface (bulky data interface or microprocessor interface)
08	DRELTIM	R/W	When this bit is set to 1, the cycletimer has reached the value of the DSS release time and an DSS cell is now being released to the selected interface (bulky data interface or microprocessor interface)
09	DHDRLD	R/W	When this bit is set to 1, the 10-byte DSS header has been loaded from the DTX registers in X130 mode into the MPEG/DSS transmit FIFO and the registers can now be updated with new values.
10	MPUERR	R/W	When this bit is set to 1, the microprocessor has attempted an illegal access to the FIFO. This usually occurs when the microprocessor tries to read an empty FIFO or writes to a full FIFO. The MPU error code in the BDFMISC register located @ offset F8h bits 1 – 4, contains the reason for the error.
11	INVWROP	R/W	When this bit is set to 1, the microprocessor has attempted to initiate a read or write operation before the current write operation has completed.
12	ARFRABRT	R/W	When this bit is set to 1, the receive packet routing control has detected and purged a partial packet from the asychronous bulky receive FIFO. This can occur when the link detects a partial packet, not enough storage space is available, or the packet can not be acknowledged.
13	Not used		
14	IRFABORT	R/W	When this bit is set to 1, receive packet routing control has detected and purged a partial isochronous packet from the bulky isochronous receive FIFO. This can occur when the link detects a partial packet or the receive FIFO has less than 2 quadlets available.
15	Not used		

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
16	MRFABORT	R/W	When this bit is set to 1, the receive packet routing control has detected and purged a partial MPEG packet from the bulky MPEG receive FIFO. This can occur whenever the FIFO has less than 2 quadlets available, data has an incorrect format, or the device is not configured properly for receive.
17	ACRFPRDY	R/W	When this bit is set to 1, the receive packet routing control has confirmed an entire packet into the asychronous control receive FIFO.
18	ACRFPABRT	R/W	When this bit is set to 1, the receive packet routing control has detected and purged a partial packet from the asychronous control receive FIFO.
19	BRFPRDY	R/W	When this bit is set to 1, the receive packet routing control has confirmed an entire packet into the broadcast control receive FIFO.
20	BRFPABRT	R/W	When this bit is set to 1, the receive packet routing control has detected and purged a partial packet from the broadcast control receive FIFO.
21	SIDPRDY	R/W	When this bit is set to 1, the receive packet routing control has detected the end of the Self-ID period and has confirmed the entire set of Self-ID packets into the selected receive FIFO.
22	SIDPABRT	R/W	When this bit is set to 1, the receive packet routing control has detected and purged a partial accumulation of Self-ID packets from the selected receive FIFO. (asychronous bulky receive FIFO or broadcast control receive FIFO).
23	TSAGED	R/W	When this bit is set to 1, the MPEG/DSS packet that is waiting to be transmitted has aged and has been flushed from the MPEG/DSS receive FIFO.
24	MPUTMOUT	R/W	When this bit is set to 1, the microprocessor interface tried to access a FIFO resource that was unable to respond due to a higher transfer already in progress.
25	ISOGOERR	R/W	When this bit is set to 1, the packetizer has detected a premature isochronous go event.
26	MDDBCERR	R/W	When this bit is set to 1, the packetizer has detected a data block continuity (DBC) error.
27	MCFLSHERR	R/W	When this bit is set to 1, the packetizer has flushed an MPEG/DSS cell from the MPEG/DSS transmit FIFO.
28	BFFLSHERR	R/W	When this bit is set to 1, the packetizer has flushed the entire contents of the MPEG/DSS transmit FIFO.
29	SBACOMP	R/W	When this bit is set to 1, the packetizer has successfully complete transmitting a packet from the bulky asychronous transmit FIFO.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
30	BFAFLERR	R/W	When this bit is set to 1, the transmitted asychonous packet was not successfully received by the addressed node. This occurs when a transmitted asynchronous packet was acknowledged with an ack_busy and the number of retries was already reached or an ack error code was received.
31	DBCCRXERR	R/W	When this bit is set to 1, the receive packet routing control has detected an error in the MPEG/DSS DBC count of the MPEG/DSS packet currently being received.

5.8 Isochronous Receive Comparators Register 0 (IRPR0 @ Addr 20h)

This register defines the tag and channel number values used by isochronous receive packet comparators 0 – 3 to determine if an incoming isochronous packet is to be accepted or rejected. The comparator enable bits IRP0EN – IRP3EN(LCTRL register @ Addr Ch) and match on tag enable bits MONT0 – MONT3 (RPRC register @ Addr 148h) are used in programming the filtering behavior of the comparators based on the following table. Unless otherwise specified, the bits in this this register are cleared to 0 on power up or software initiated reset.

IRPxEN†	MONTx†	COMPARE FUNCTION
0	0	Both channel and tag comparators are disabled for port x.
1	0	Match IRPORTx expected value to channel number of incoming isochronous packet. Tag comparator disabled.
1	1	Match IRPORTx and TAGx expected values to channel number and tag number of incoming isochronous packet

† x = 1, 2, or 3.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 01	TAG0	R/W	MPEG/DSS Port 0 tag field receive packet compare value
02 – 07	IRPORT0	R/W	MPEG/DSS Port 0 isochronous channel number receive packet compare value
08 – 09	TAG1	R/W	Isochronous Port 1 tag field receive packet compare value
10 – 15	IRPORT1	R/W	Isochronous Port 1 isochronous channel number receive packet compare value
16 – 17	TAG2	R/W	Isochronous Port 2 tag field receive packet compare value
17 –23	IRPORT2	R/W	Isochronous Port 2 isochronous channel number receive packet compare value
24 – 25	TAG3	R/W	Isochronous Port 3 tag field receive packet compare value
26 – 31	IRPORT3	R/W	Isochronous Port 3 isochronous channel number receive packet compare value

5.9 Isochronous Receive Comparators Register 1 (IRPR1 @ Addr 24h)

This register defines the tag and channel number values used by isochronous receive packet comparators 0 – 3 to determine if an incoming isochronous packet or MPEG/DSS packet is to be accepted or rejected. Comparator number 7 is used for receiving MPEG/DSS isochronous packet types. The comparator enable bits IRP4EN – IRP7EN (LCTRL register @ Addr Ch) and match on tag enable bits MONT4 – MONT7 (RMISC register @ Addr 148h) are used in programming the filtering behavior of the comparators based on the following table. Unless otherwise specified, the bits in this register are cleared to 0 on power up or software initiated resets.

IRPxEN†	MONTx†	COMPARE FUNCTION
0	0	Both channel and tag comparators are disabled for port x.
1	0	Match IRPORTx expected value to channel number of incoming isochronous packet. Tag comparator disabled.
1	1	Match IRPORTx and TAGx values to channel number and tag number of incoming isochronous packet.

† x = 4, 5, 6, or 7.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 01	TAG4	R/W	Isochronous Port 4 tag field receive packet compare value
02 – 07	IRPORT4	R/W	Isochronous Port 4 isochronous channel number receive packet compare value
08 – 09	TAG5	R/W	Isochronous Port 5 tag field receive packet compare value.
10 – 15	IRPORT5	R/W	Isochronous Port 5 isochronous channel number receive packet compare value
16 – 17	TAG6	R/W	Isochronous Port 6 tag field receive packet compare value.
17 –23	IRPORT6	R/W	Isochronous Port 6 isochronous channel number receive packet compare value
24 – 25	TAG7	R/W	Isochronous Port 7 tag field receive packet compare value
26 – 31	IRPORT7	R/W	Isochronous Port 7 isochronous channel number receive packet compare value

5.10 Cycle Timer Register (CYCTIM @ Addr 28h)

The register provides the application software with a read/write access interface to the cycle timer register. This register is comprised of three fields. They are: seconds_count, cycle_number_count, and cycle_offset. The operation of the timer is controlled by control bits CMSTR, CYCSRC, CYCTIMEN. These bits are located in the link control register (LCTRL @ Addr 0Ch). Unless other wise specified, the cycle timer register is cleared to 0 on power up or software initiated reset.

Table 5–1. Cycle Timer Program Function

CYCSRC	CMSTR	CYCTIMEN	DESCRIPTION
X	X	0	Cycle timer is disabled from counting.
0	0	1	This node is not the cycle master. The cycle timer is enabled to count from the internal clock source and can be initialized with the cycle_time_data extracted from a cycle start packet that is received by this node.
0	1	1	This node is the cycle master. the cycle timer is enabled to count from the internal clock source.
1	0	1	This node is not the cycle master. The timer is enabled to increment whenever the external signal CYCLEIN is pulsed high for a minimum of 80 ns. The Timer can be initialized with the cycle_time_data extracted from a cycle start packet that is received by this node.
1	1	1	This node is the cycle master. The timer is enabled to increment whenever the external signal CYCLEIN is pulsed high for a minimum of 80 ns.

Table 5–2. Cycle Time Register

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 06	CYCSEC	R/W	Cycle seconds. 1-Hz cycle timer counter
07 – 19	CYCNUMBER	R/W	8000-Hz cycle timer counter
20 – 31	CYCOFFSET	R/W	24.576-MHz cycle timer counter

5.11 Bus Time Register (BUSTIM @ Addr 2Ch)

This bit map defines the extended bus time counter register. The bus time counter (BUSTIME) is incremented whenever the cycle timer rolls over in all 32 bits. The extended cycle timer is enabled to count when the CYCTIMEN bit located in link control register (LCTRL @ Addr 0Ch) is set to 1. Unless otherwise specified, the extended cycle time register is cleared to 0 on a power up or software initiated reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 24	BUSTIME	R/W	Extended bus timer in seconds.
25 – 31	SECSLO	R	CYCSEC field of the cycle timer register.

5.12 Link Diagnostics Register (DIAG @ Addr 30h)

This register provides the application software with the capability to perform diagnostic testing. Unless otherwise specified, this register is cleared to 0 on power up or software initiated reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	ENSNOOP	R/W	When set to 1, this bit enables the 1394 Link receiver to snoop 1394 bus traffic.
01	Not used		
02	ISOBAROFFCR	R	The status value of the isolation barrier control state from the core logic (including bus headers and phy-link interface capacitors). When this bit is set to 1, the isolation barrier is off. This mode only works if there is no isolation circuitry on the phy-link interface. When this bit is set to 0, isolation barrier is on. This bit is set to 1 on power up or software reset.
03	ISOBAROFF	R/W	When this bit is set to 1, the isolation barrier function is turned off
04	REGRW	R/W	When set to 1, this bit configures read-only registers to function as read/write registers. This only applies to read-only registers that are R/W configurable.
05 – 08	Reserved		Reserved
09 – 12	STAT3MUXSEL	R/W	STAT3 output internal signal MUX select lines.
			STAT3MUXSEL / **Internal Signal Selected**
			0h / BDIBusyOut
			Fh / NClk
13 – 16	STAT2MUXSEL	R/W	STAT2 output internal signal MUX select lines.
			STAT2MUXSEL / **Internal Signal Selected**
			0h / BDOAvailOut
17 – 20	Reserved		
21 – 27	STAT0MUXSEL	R/W	STAT0 output internal signal MUX select lines.
			STAT0MUXSEL / **Internal Signal Selected**
			1Eh / BDIBusyOut
			1Fh / BDOAvailOut
			3Bh / NClk
28 – 31	Reserved		

5.13 Phy Access Register (PHYAR @ Addr 34h)

This register provides the application software with an interface for accessing the registers in the Phy. Unless otherwise specified, this register is cleared to 0 on power up or software initiated reset. The functionality of the register is defined by the following bit map.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	RDPHYREQ	R/W	Read Phy register request. When this bit is set to 1, Phy register read request is issued from the address specified PHYREGADR. This bit is cleared after the link has sent the request to the Phy.
01	WRPHYREQ	R/W	Write Phy register request. When this bit is set to 1, a Phy register write request is issued to Phy that contains the register address and write data obtained from PHYREGADR and PHYREGWRDATA. This request bit is cleared after the link has sent the request to the Phy.
02 – 03	Not used		
04 – 07	PHYREGADR	R/W	Phy register address. This field specifies the address of the Phy register that is read from or written to.
08 – 15	PHYREGWRDATA	R/W	Phy register write data. This field specifies the data that is written to the Phy register specified by PHYREGADR.
16 – 19	Not used		
20 – 23	PHYREGADRRCV	R	Phy register address received. These register bits buffer the register address returned by the Phy in response to a Phy register read request. The host processor can write to these bits when the REGRW bit in diagnostic control register (DIAG @ Addr 30h) is set to 1.
24 – 31	PHYREGDATARCV	R	Phy register data received. These register bits buffer the data returned by the Phy in response to a Phy register read request. The host processor can write to these bits when the REGRW bit in diagnostic control register (DIAG @ Addr 30h) is set to 1.

5.14 Expected Response (PHYSR @ Addr 38h)

This register provides application software with the capability to program the receive packet routing control to filter incoming response packets for one that contains a tranaction label and tcode value that was used in previously issued request packets. Once identified, the response packet can then be steered into a selected receiving FIFO. The expected response comparator logic is enabled by using a EXP_TCODE value that is expected and is not equal to Fh.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	Not used		
16 – 21	EXP_LABEL	R/W	The value of the transaction label for the expected response packet. These bits are set to 0 on power up reset, bus_reset, or software initiated reset.
22 – 23	Not used		
24 – 27	EXP_TCODE	R/W	The tcode for the expected response packet. These bits are set to all 1s on a power up reset, bus_reset, or software initiated reset. The expected response comparator is disabled when EXP_TCODE = 1111.
28 – 31	Not used		

5.15 Reserved Register (RESERVED @ Addr 3ch–40h)

This register is reserved.

5.16 Reserved Register (RESERVED @ Addr 3Ch)

This register is reserved.

5.17 Reserved Register (RESERVED @ Addr 40h)

This register is reserved.

5.18 Asynchronous Control Data Transmit FIFO Status (ACTFS @ Addr 44h)

This register provides the application software with the capability to monitor the occupancy status of the asychronous control transmit FIFO and to program its size. Unless otherwise specified, this register is cleared to 0 on power up, bus reset or software initiated reset

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	ACTFULL	R	When this bit is set to 1, asychronous control transmit FIFO is full. The host processor can write and read these bits when the REGRW bit in diagnostic control register (DIAG @ Addr 30h) is set to 1.
01	ACTALFULL	R	When this bit is set to 1, asychronous control transmit FIFO is almost full. The host processor can write and read these bits when the REGRW bit in diagnostic control register (DIAG @ Addr 30h) is set to 1.
02 – 03	Not used		
04	ACT4AVAIL	R	When this bit is set to1, the asychronous control transmit FIFO has four empty locations available for storing data. The host processor can write and read these bits when the REGRW bit in diagnostic control register (DIAG @ Addr 30h) is set to 1.
05 – 13	Not used		
14	ACTALEMPTY	R	When this bit is set to 1, the asychronous control transmit FIFO is almost empty. The host processor can write and read these bits when the REGRW bit in diagnostic control register (DIAG @ Addr 30h) is set to 1.
15	ACTEMPTY	R	When this bit is set to 1, the asychronous control transmit FIFO is empty. The host processor can write and read these bits when the REGRW bit in diagnostic control register (DIAG @ Addr 30h) is set to 1. This bit is set to 1 On power up reset, bus reset or software reset.
16	Not used		
17	ACTCLR	R/W	When this bit is set to 1, the asychronous control transmit FIFO is flushed. This bit is self clearing.
18 – 24	Not used		
25 – 31	ACTSIZE	R/W	Asychronous control transmit FIFO size setting in quadlets. On power up or software reset these bits are set to 14h (20 quadlets).

5.19 Bus Reset Data Register (BRD @ Addr 48h)

This register provides the application software with the capability to program the operation of the bus reset controller. Unless otherwise specified, this register is cleared to 0 on power up or software initiated reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 09	BUSNUM	R/W	Bus number. Set to a value that has been determined by the application software, the bus number defaults to 3FFh on power up or software initiated resets.
10 – 15	NODENUM	R/W	Node number. This value can be set by software or automatically set to the node ID value returned in a status response from the Phy, when it transmits its Self-ID packet on the 1394 bus. The node number is set to 3Fh when a bus reset status response is received by the link or a power up/software reset occurs.
16	NRIDV	R	Node count IRM ID valid. This bit is set to 1 when NODECNT and IRMID are valid.
17	Not used		
18 – 23	NODECNT	R	The number of nodes in the 1394 network. The node count is set to 1 on bus reset, power up reset, or software reset.
24	ROOT	R	The root state of the local Phy.
25 – 31	IRMID	R	The ID of the IRM node. The IRM ID is set to 3Fh on bus reset, power up reset, or software reset.

5.20 Bus Reset Error Register (BRERR @ Addr 4Ch)

This register provides the application software with error status generated by bus reset control when it detects an error condition. The internal state machine vectors of the bus reset control are also provided in this register. Unless otherwise specified, this register is cleared to 0 on power up or software initiated reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	CBRERR	R/W	Clear bus reset controller error. When this bit is set to 1, the BRERRCODE is set to 0000. This bit is self clearing.
01 – 04	BRERRCODE	R	Bus reset controller error code returned 0000 – No error occurred 0001 – Last Self-ID does not have all ports marked as child 0010 – Expected Phyid ≠ Phyid 0011 – 2nd quad of Self-ID not inverted from 1st quad 0100 – Phyid incremented by two 0101 – Phyid incremented any 3 or more 0110 – Phyid not equal in packet 0111 – Self-ID quadlets are not inverses of each other. 1000 – Self-ID quadlets is bad
05 – 15	Not used		
16 – 17	Reserved		
18 – 19	Not used		
20 – 22	Reserved		
23 – 31	Not used		

5.21 Asynchronous Control Data Receive FIFO Status (ACRXS @ Addr 50h)

This register provides the application software with capability to monitor the occupancy status of the asychronous control and broadcast control receive FIFOs and to also set their size. All of the bits that are indicated as read only, can be made read/write by the host processor. This is done by setting the REGRW bit in diagnostic control register (DIAG @ Addr 30h) to 1.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	ACRFULL	R	When this bit is set to 1, the asychronous control receive FIFO is full. On bus_reset, power up reset, or software reset, this bit is cleared to 0.
01	BRFFULL	R	When this bit is set to 1, the broadcast control receive FIFO is full. On bus_reset, power up reset, or software reset, this bit is cleared to 0
02	ACRALFULL	R	When this bit is set to 1, the asychronous control receive FIFO is almost full. On bus_reset, power up reset, or software reset, this bit is cleared to 0
03	BRFALFULL	R	When this bit is set to 1, the broadcast control receive FIFO is almost full. On bus_reset, power up reset, or software reset, this bit is cleared to 0.
04 – 10	Not used		
11	BRFCD	R	This bit indicates the state of the control bit for the first data quadlet read from the broadcast receive FIFO. On bus_reset, power up reset, or software reset, this bit is cleared to 0.
12	ACR4AVAIL	R	When this bit is set to 1, the asychronous control receive FIFO has four locations available for storage. On bus_reset, power up reset, or software reset, this bit is cleared to 0.
13	BRFEMPTY	R	When this bit is set to 1, broadcast receive FIFO is empty. On bus_reset, power up reset, or software reset, this bit is set to 1.
14	ACRALEMPTY	R	When this bit is set to 1, asychronous control receive FIFO is almost empty. On bus_reset, power up reset, or software reset, this bit is set to 0
15	ACREMPTY	R	When this bit is set to 1, the asychronous control receive FIFO is empty. On bus_reset, power up reset, or software reset, this bit is set to 1.
16	ACRCD	R	This bit indicates the state of the control bit for the first data quadlet read from the asychronous control receive FIFO. On bus_reset, power up reset, or software reset, this bit is set to 0.
17	ACRCLR	R/W	Control receive FIFO clear. When this bit is set to 1, the asychronous receive control and broadcast receive control FIFOs are flushed. This bit is self clearing and is also cleared on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
18 – 24	BRFSIZE	R/W	These bits set the size of the broadcast receive FIFO in quadlets. On power up or software reset, these bit are set to 15h (21 quadlet).
25 – 31	ACRSIZE	R/W	These bits set the size of the asychronous receive FIFO in quadlets. On power up or software reset, these bit are set to 15h (21 quadlets).

5.22 Reserved Register (RESERVED @ Addr 54h)

This register is reserved.

5.23 Reserved Register (RESERVED @ Addr 55h – 7Fh)

This register is reserved.

5.24 Asynchronous Control Data Transmit FIFO First (ACTXF @ Addr 80h)

This write-only port provides application software with the capability to write the first quadlet of a packet to the asychronous control transmit FIFO where it is marked in the FIFO as the first quadlet of the packet.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ACTXF	W	Asychronous control transmit first

5.25 Asynchronous Control Data Transmit FIFO Continue (ACTXC @ Addr 84h)

This write-only port provides application software with the capability to write the remaining quadlets of a packet except the last quadlet, to the asychronous control transmit FIFO.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ACTXC	W	Asychronous control transmit and continue

5.26 Asynchronous Control Data Transmit FIFO First & Update (ACTXFU @ Addr 88h)

This write-only port provides application software with the capability to write a quadlet to the asychronous control transmit FIFO and have it confirmed for transmission.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ACTXFU	W	Asychronous control transmit first and update

Data written to this address goes to the ACTX FIFO and is confirmed for transmission.

5.27 Asynchronous Control Data Transmit FIFO Last & Send (ACTXCU @ Addr 8ch)

This write-only port provides application software with the capability to write the last quadlet of a packet to the asychronous control transmit FIFO and have the entire packet confirmed for transmission.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ACTXCU	W	Asychronous control transmit continue and update

5.28 Reserved Register (RESERVED @ Addr 90h–BCh)

This register is reserved.

5.29 Asynchronous Control Data Receive FIFO (ACRX @ Addr C0h)

This register port allows read accesses to the ACRX FIFO. If there is more than one quadlet is in this FIFO, each read outputs the next quadlet from this FIFO. If the FIFO is empty, the last valid value is read. This FIFO is meant for low-rate asynchronous control data. However, it can also be used for application data, which is accessed through the MP/MC interface. This register is cleared to 0 on power up, bus_reset, or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ACRX	R	Asychronous control receive FIFO read port

5.30 Broadcast Write Receive FIFO (BWRX @ Addr 0C4h)

This register port allows accesses to the BWRX FIFO. If there is more than one quadlet is in this FIFO, each read outputs the next quadlet from this FIFO. If the FIFO is empty the last valid value is read. This FIFO is meant for low-rate asynchronous control data. However, it can also be used for application data, which is accessed through the MP/MC interface. This register is cleared to 0 on power up, bus_reset, or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BWRX	R	Broadcast receive FIFO read port

5.31 Reserved Register (RESERVED @ Addr C8h)

This register is reserved.

5.32 Reserved Register (RESERVED @ Addr CCh)

This register is reserved.

5.33 Reserved Register (RESERVED @ Addr D0h)

This register is reserved.

5.34 Reserved Register (RESERVED @ Addr D4h)

This register is reserved.

5.35 Bulky Data Interface Control (BIF @ Addr D8h)

This register provides the application software with the capability to program and control the functionality of the bulky data interface. Unless otherwise specified, the bits in this register are cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 07	Not used		
08	LNGRDREG	R/W	If BDIMODE = 010, this bit causes the BDOAVAIL signal to remain active through the entire packet.
09	BMLECTRL	R/W	MPEG little endian control. When set to 1, this bit causes the MPEG interface to operate in little endian format (Byte 0 = LSByte).
10	BILECTRL	R/W	Isochronous little endian control. When set to 1, this bit causes the isochronous interface to operate in little endian format (Byte 0 = LSByte).
11	BALECTRL	R/W	Asychronous little endian control. When set to 1, this bit causes the asychronous interface to operate in little endian format (Byte 0 = LSByte).
12	AHBDIBSY	R/W	Active high control for BDIBUSY terminal. When set to 1, this bit causes the signal to be active high. This bit is set to 1 on power up or software reset.
13	AHAVAIL	R/W	Active high control for BDOAVAIL terminal. When set to 1, this bit causes the signal to be active high. This bit is set to 1 on power up or software reset.
14	AHBDOEN	R/W	Active high control for BDOEN terminal. When set to 1, this bit causes the signal to be active high. This bit is set to 1 on power up or software reset.
15	AHBDIEN	R/W	Active high control for BDIEN terminal. When set to 1, this bit causes the signal to be active high. This bit is set to 1 on power up or software reset.
16	UNIDIR	R/W	If BDIMODE = 010, when set, this bit causes the interface to be unidirectional.
17	DSSREC	R/W	If UNIDIR is set to 1, BDIMODE = 010, and this bit is set to 1, it causes write functionality to be enabled and read functionality to be disabled.
18	AHPACEN	R/W	If BDIMODE is set to 010 or 101 and this bit is set to 1, it causes the BDIF2 (packet enable) terminal to be active high. This bit is set to 1 on power up or software reset.
19	AHERROR	R/W	If BDIMODE is set to 010 or 101 and this bit is set to 1, it causes the BDIF1 (packet error) terminal to be active high. This bit is set to 1 on power up or software reset.
20	AHBDIFMT0	R/W	If BDIMODE is set to 101 and this bit is set to 1, it causes the BDIF0 (valid) terminal to be active high. This bit is set to 1 on power up or software reset.
21	Not used		
22	BDOMODE1	R/W	MSB of the BDOMODE select bits
23	BDOMODE0	R/W	LSB of the BDOMODE select bits

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
24	Not used		
25	BDIMODE2	R/W	MSB of the BDIMODE select bits
26	BDIMODE1	R/W	Middle bit of the BDIMODE select bits
27	BDIMODE0	R/W	LSB of the BDIMODE select bits
28	RCVPAD	R/W	When set to 1, this bit allows 1394 padding bits through the interface port. Data must be written to the BDIF in quadlet multiples (4 bytes at a time), if a packet does not end on a quadlet boundary then padding zeros are automatically added to complete the last quadlet. When RCVPAD is set to 1, the BDIF does not strip the inserted zeroes of received packets prior to sending them to the BDIF.
29	BDOINIT	R/W	When set to 1, this bit causes the BDO logic to reset. This bit is self clearing.
30	BDIINIT	R/W	When set to 1, this bit causes the BDI logic to reset. This bit is self clearing.
31	BDOTRIS	R/W	When set to 1, this bit causes the BDO data bus to be forced high-impedance state.

5.36 MPEG2 (DVB) Transmit Timestamp Offset Register (MXTO @ Addr DCh)

This register provides the application software with the capability to program the timestamp offset for a MPEG transmit cell. The hardware adds this offset to a sampled value of the cycle timer to determine the timestamp for the MPEG cell to be transmitted. Unless otherwise specified, the bits in this register are cleared to 0 on power up or software reset. This register is the same register as E0h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 06	CYCEC	R/W	Offset value added to CYCSEC of CYCTIM register (reg 28)
07 – 19	CYCNUMBER	R/W	Offset value added to CYCNUMBER of CYCTIM register (reg 28) with a maximum value of 7999h
20 – 31	CYCOFFSET	R/W	Offset value added to CYCOFFSET of CYCTIM register (reg 28) with a maximum value of 3071h

5.37 DSS Transmit Timestamp Offset Register (DXT0 @ Addr E0h)

This register provides the application software with the capability to program the timestamp offset for a DSS transmit cell. The hardware adds this offset to a sampled value of the cycle timer to determine the timestamp for the DSS cell to be transmitted. Unless otherwise specified, this register are cleared to 0 on power up or software reset. This register is the same as DCh.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 06	CYCEC	R/W	Offset value added to CYCSEC of CYCTIM register (reg 28)
07 – 19	CYCNUMBER	R/W	Offset value added to CYCNUMBER of CYCTIM register (reg 28) with a maximum value of 7999h
20 – 31	CYCOFFSET	R/W	Offset value added to CYCOFFSET of CYCTIM register (reg 28) with a maximum value of 3071h

5.38 MPEG2 (DVB)/DSS Receive Timestamp Offset (MRTO @ Addr E4h)

This register provides the application software with the capability to program the timestamp offset for a MPEG receive cell. The hardware adds this offset to the timestamp of the received MPEG cell to determine the time to release the cell to the application. Unless otherwise specified, the bits in this register are cleared to 0 on power up or software reset. This register is the same as E8h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 06	CYCEC	R/W	Offset value added to CYCSEC of CYCTIM register (reg 28)
07 – 19	CYCNUMBER	R/W	Offset value added to CYCNUMBER of CYCTIM register (reg 28) with a maximum value of 7999h
20 – 31	CYCOFFSET	R/W	Offset value added to CYCOFFSET of CYCTIM register (reg 28) with a maximum value of 3071h

5.39 DSS Receive Timestamp Offset Register (DRT0 @ Addr E8h)

This register provides the application software with the capability to program the timestamp offset for a DSS receive cell. The hardware adds this offset to the timestamp of a received DSS cell to determine the release time of the cell to the application. Unless otherwise specified, the bits in this register are cleared to 0 on power up or software reset. This register is the same as E4h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 06	CYCEC	R/W	Offset value added to CYCSEC of CYCTIM register (reg 28)
07 – 19	CYCNUMBER	R/W	Offset value added to CYCNUMBER of CYCTIM register (reg 28) with a maximum value of 7999h
20 – 31	CYCOFFSET	R/W	Offset value added to CYCOFFSET of CYCTIM register (reg 28) with a maximum value of 3071h

5.40 Asynchronous/Isochronous Application Data Control Register (AICR @ Addr ECh)

This register provides the application software with the capability to program and control the operational behavior and data path control for the asychronous and isochronous transmit and receive FIFOs. Unless otherwise specified, all bits in the register are cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	IRENABLE	R/W	Isochronous receive enable. When this bit is set to 1, the bulky isochronous receive FIFO is enabled to receive data.
01	ITENABLE	R/W	Isochronous transmit enable. When this bit is set to 1, the bulky isochronous transmit FIFO is enabled to transmit data.
02	ARENABLE	R/W	Asychronous receive enable. When this bit is set to 1, the asychronous receive FIFO is enabled to receive data.
03	ATENABLE	R/W	Asychronous transmit enable. When this bit is set to 1, the bulky asychronous transmit FIFO is enabled to transmit data. This bit is cleared when a 1394 bus reset occurs.
04 – 14	Not used		
15	ISNOOP	R/W	Isochronous snoop. When this bit is set to 1, all incoming isochronous traffic is snooped and stored to the bulky isochronous receive FIFO.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
16	IHIM	R/W	Isochronous header insert mode enable. When this bit is set to 1, automatic header insertion and packetization of isochronous data from the isochronous transmit FIFO is enabled. In this mode the hardware expects the application to load the isochronous transmit FIFO with pure data that contains no header. When this bit is set to 0, the hardware expects the isochronous transmit FIFO to contain completely formatted 1394 isochronous packets.
17	Not used		
18	Not used		
19	IRHS	R/W	Isochronous header strip mode enable. When this bit is set to 1, the isochronous header is stripped from the packet and only data payload is delivered to the application. The isochronous header is copied to the register
20	BDIRE	R/W	Bulky data isochronous receive FIFO data destination select. When this bit is set to 0, the MP/MC has access to the bulky data receive FIFO. Received isochronous packets are not transferred to the application by way of the BDIF. When this bit is set to 1, the received isochronous packets are transferred to the application by way of the BDIF.
21	BDIXE	R/W	Bulky data isochronous transmit FIFO data source select. When this bit is set to 0, the MP/MC has write access to the bulky isochronous transmit FIFO. Data writes from the BDIF are ignored. When this bit is set to 1, the application has write access to the bulky isochronous transmit FIFO by way of the BDIF.
22	IRFLSH	W	Bulky isochronous receive FIFO flush. Setting this bit to 1 flushes the isochronous receive FIFO. This bit is self clearing
23	IXFLSH	W	Bulky isochronous transmit FIFO flush. Setting this bit to 1 flushes the isochronous transmit FIFO. This bit is self clearing.
24	AHIM	R/W	Asychronous header insert mode enable. When this bit is set to 1, automatic header insertion and packetization of asychronous data from the bulky asychronous transmit FIFO is enabled. In this mode the hardware expects the application to load the asychronous transmit FIFO with pure data that contains no header. When this bit is set to 0, the hardware expects the asychronous transmit FIFO to contain completely formatted 1394 isochronous packets.
25	Not used		
26	Not used		

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
27	ARHS	R/W	Asychronous header strip mode enable. When this bit is set to 1, the asychronous header is stripped from the packet and only data payload is delivered to the application. The asychronous header is copied to the asychronous header registers
28	BDARE	R/W	Bulky data asychronous receive FIFO data destination select. When this bit is set to 0, the MP/MC has access to the bulky data asychronous receive FIFO. Received asychronous packets are not transferred to the application by way of the BDIF. When this bit is set to 1, the received asychronous packets are transferred to the application by way of the BDIF.
29	BDAXE	R/W	Bulky data asychronous transmit FIFO data source select. When this bit is set to 0, the MP/MC has write access to the bulky asychronous transmit FIFO. Data writes from the BDIF are ignored. When this bit is set to 1, the application has write access to the bulky asychronous transmit FIFO by way of the BDIF.
30	ARFLSH	W	Bulky asychronous receive FIFO flush. Setting this bit to 1 flushes the asychronous receive FIFO. This bit is self clearing.
31	AXFLSH	W	Bulky asychronous transmit FIFO flush. Setting this bit to 1 flushes the asychronous transmit FIFO. This bit is self clearing.

5.41 MPEG2 (DVB)/DSS Formatter Control Register (MCR @ Addr F0h)

This register provides the application software with the capability to program and control the operational behavior and data path selection for the MPEG transmit and receive FIFOs. Unless otherwise specified, all bits in the register are cleared to 0 on power up or software reset. This register is the same register as F4h

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION	
00	MREN	R/W	MPEG receive enable. When this bit is set to 1, the bulky MPEG receive FIFO is enabled to receive MPEG packets.	
01	MTEN	R/W	MPEG transmit enable. When this bit is set to 1, the bulky MPEG transmit FIFO is enable for transmitting packets.	
02 – 03	Not used			
04	MTXERMODE	R/W	When this bit is set to 1, DTX0 register bit = BDIF DSSTXERROR signal.	
05	MRXERMODE	R/W	When this bit is set to 1, BDIF DSSRXERROR signal = bit 16 of register DSR0	
06	MXAGE	R/W	MPEG transmit aging enabled. When this bit is set to 1, MPEG transmit aging is enabled.	
07	MRAGE	R/W	MPEG receive aging enabled. When this bit is set to 1, MPEG receive aging is enabled.	
08 – 12	Not used			
13 – 15	MXC	R/W	MPEG transmit class: (see Table 3–3)	
			000	Source packet is divided into 8, 1/8 size blocks and transmitted in 8 seperate MPEG2 packets
			001	Source packet is divided into 4, 1/4 size blocks and transmitted in 4 MPEG2 packets
			010	Source packet is divided into 2, 1/2 size blocks and transmitted in 2 MPEG2 packets
			011	Exactly one source packet is transmitted per MPEG2 packet
			100	One or two source packets is transmitted in each MPEG2 packet
			101	One to three source packets is transmitted in each MPEG2 packet
			110	One to four source packets is transmitted in each MPEG2 packet
			111	One to five source packets is transmitted in each MPEG2 packet
16	MRHS	R/W	MPEG header strip enable. When this bit is set to 1, the Iso, CIP0, and CIP1 headers and trailer quadlet are stripped from the MPEG receive packet and copied into buffer registers. The remaining source packet is transferred to the application.	
17	MEXTS	R/W	MPEG extended timestamp enable. When this bit is set to 1, a 32-bit timestamp is used for both transmit and receive. When this bit is set to 0 a 25-bit timestamp is used for both transmit and receive.	

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
18 –21	Not used		
22	DSSR30	R/W	When this bit is set to 1, the device is ready to receive DSS130 formatted packets (DSS130 format is supported by BDIF only).
23	DSSX30	R/W	When this bit is set to 1, the device is ready to transmit DSS130 formatted packets (DSS130 format is supported by BDIF only).
24	BDMRE	R/W	Bulky data MPEG receive FIFO data destination select. When this bit is set to 0, the MP/MC has access to the bulky data MPEG receive FIFO. Received MPEG packets are not transferred to the application by way of the BDIF. When this bit is set to 1, the received MPEG packets are transferred to the application by way of the BDIF.
25	BDMXE	R/W	Bulky data MPEG transmit FIFO data source select. When this bit is set to 0, the MP/MC has write access to the bulky MPEG transmit FIFO. Data writes from the BDIF are ignored. When this bit is set to 1, the application has write access to the bulky MPEG transmit FIFO by way of the BDIF.
26	MRFLSH	W	Bulky MPEG receive FIFO flush. Setting this bit to 1 flushes the MPEG receive FIFO. This bit is self clearing.
27	MXFLSH	W	Bulky MPEG transmit FIFO flush. Setting this bit to 1 flushes the MPEG transmit FIFO. This bit is self clearing.
28	MFEN	R/W	MPEG mode enable. When this bit is set to 1, the device operates in MPEG mode. When this bit is set to 0, the device operates in DSS mode.
29	MTXTSIN	R/W	MPEG timestamp insert enable. When this bit is set to 1, the timestamp is automatically inserted on MPEG transmits.
30	MALTCELL	R/W	When this bit is set to 1, the device uses the transmit and receive alternate size defined in register BDFMISC (register F8h).
31	MHIM	R/W	When this bit is set to 1, the isochronous and CIP headers on MPEG transmits are automatically inserted.

5.42 DSS Formatter Control Register (DCR @ Addr F4h)

This register provides the application software with the capability to program and control the operational behavior and data path selection for the MPEG transmit and receive FIFOs. Unless otherwise specified, all bits in the register are cleared to 0 on power up or software reset. This register is the same as register F0h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION	
00	DREN	R/W	DSS receive enable. When this bit is set to 1, the bulky DSS receive FIFO is enabled to receive DSS packets.	
01	DTEN	R/W	DSS transmit enable. When this bit is set to 1, the bulky DSS transmit FIFO is enable for transmitting packets.	
02 – 03	Not used			
04	DTXERMODE	R/W	When this bit is set to 1, DTX0 register bit = BDIF DSSTXERROR signal.	
05	DRXERMODE	R/W	When this bit is set to 1, BDIF DSSRXERROR signal = bit 16 of register DSR0	
06	DXAGE	R/W	DSS transmit aging enabled. When this bit is set to 1, DSS transmit aging is enabled.	
07	DRAGE	R/W	DSS receive aging enabled. When this bit is set to 1, DSS receive aging is enabled.	
08 – 12	Not used			
13 – 15	DXC	R/W	DSS transmit class: (see Table 3–6)	
			001	Source packet is divided into 4, 1/4 size blocks and transmitted in 4 DSS packets
			010	Source packet is divided into 2, 1/2 size blocks and transmitted in 2 DSS packets
			011	Exactly one source packet is transmitted per DSS packet
			100	One or two source packets is transmitted in each DSS packet
			101	One to three source packets is transmitted in each DSS packet
			110	One to four source packets is transmitted in each DSS packet
			111	One to five source packets is transmitted in each DSS packet
16	DRHS	R/W	DSS header strip enable. When this bit is set to 1, the Iso, CIP0, and CIP headers and trailer quadlet are stripped from the DSS receive packet and copied into buffer registers.	
17	DEXTS	R/W	DSS extended timestamp enable. When this bit is set to 1, a 32-bit timestamp is used for both transmit and receive. When this bit is set to 0, a 25-bit timestamp is used for both transmit and receive.	
18 –21	Not used			

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
22	DSSR30	R/W	When this bit is set to 1, the device is ready to receive DSS130 formatted packets (DSS130 format is supported by BDIF only).
23	DSSX30	R/W	When this bit is set to 1, the device is ready to transmit DSS130 formatted packets (DSS130 format is supported by BDIF only).
24	BDDRE	R/W	Bulky data DSS receive FIFO data destination select. When this bit is set to 0, the MP/MC has access to the bulky data DSS receive FIFO. Received DSS packets are not transferred to the application by way of the BDIF. When this bit is set to 1, the received DSS packets are transferred to the application by way of the BDIF.
25	BDDXE	R/W	Bulky data DSS transmit FIFO data source select. When this bit is set to 0, the MP/MC has write access to the bulky DSS transmit FIFO. Data writes from the BDIF are ignored. When this bit is set to 1, the application has write access to the bulky DSS transmit FIFO by way of the BDIF.
26	DRFLSH	W	Bulky DSS receive FIFO flush. Setting this bit to 1 flushes the DSS receive FIFO. This bit is self clearing.
27	DXFLSH	W	Bulky DSS transmit FIFO flush. Setting this bit to 1 flushes the DSS transmit FIFO. This bit is self clearing.
28	DFEN	R/W	DSS mode enable. When this bit is set to 1, the device operates in MPEG mode. When this bit is set to 0, the device operates in DSS mode.
29	DTXTSIN	R/W	DSS timestamp insert enable. When this bit is set to 1, the timestamp is automatically inserted on DSS transmits.
30	DALTCELL	R/W	When this bit is set to 1, the device uses the transmit and receive alternate size defined in register BDFMISC (register F8h).
31	DHIM	R/W	When this bit is set to 1, the isochronous and CIP headers are automatically inserted on DSS transmits.

5.43 Bulky Data FIFO Miscellaneous Control and Status (BDFMISC @ Addr F8h)

This register provides the application software with the capability to program an alternate cell size for MPEG and DSS cells and to decode the error status when the microprocessor performs an illegal push or pop operation.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	RESERVED		
01 – 04	MPUERRCODE	R	0h – Power up or software reset value 1h – MPU tried to pop an empty Async RX FIFO 2h – MPU tried to pop a paused Async RX FIFO 3h – MPU tried to push a full Async TX FIFO 4h – MPU tried to pop an empty Iso RX FIFO 5h – MPU tried to pop a paused Iso RX FIFO 6h – MPU tried to push a full Iso TX FIFO 7h – MPU tried to pop an empty MPEG RX FIFO 8h – MPU tried to pop a paused MPEG RX FIFO 9h – MPU tried to pop a MPEG RX FIFO before timestamp release had occurred Ah – MPU tired to pop a MPEG RX FIFO while in BDIF mode Bh – MPU tried to push a full MPEG TX FIFO Ch – MPU tried to push a MPEG TX FIFO while in BDIF mode
05 – 06	RESERVED		
07 – 15	TXMCSZ	R/W	Alternate transmit cell size. Cleared to 0 on power up or software reset. Only MPEG2 (DVB)/DSS transmit classes 0 – 5 are supported.
16 – 22	RESERVED		
23 – 31	RXMCSZ	R/W	Alternate receive cell size. Cleared to 0 on power up

5.44 SRAM Address (SRAMA @ Addr FCh)

This register interface provides the application software with the capability to load an 11-bit starting address for directly accessing the 8K x 33 SRAM that is used in implementing the bulky data FIFOs. This address auto increments on every data read or data write from/to port (SRAMD @ Addr 100h).

This register is cleared to 0 on a power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 20	Not used		
21 – 31	SRAMA	R/W	11-bit SRAM starting address where bit 21 is the MSB

5.45 SRAM Data (SRAMD @ Addr 100h)

This register interface provides the application software with the capability to directly read or write data from/to the 8K-byte × 33 bulky FIFO SRAM. The address of the read or write access is obtained from SRAMA located in register port (SRAMA @ Addr Fch).

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	SRAMD	R/W	Read or write data. Bit 00 is the MSB The default value returned on a read is 0.

5.46 Bulky Asynchronous Size register (BASZ @ Addr 104h)

The register provides the application software with the capability to program the size, in multiples of 4 quadlets, of the bulky Async transmit and receive FIFOs.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 05	Not used		
06 – 15	BATXSIZE	R/W	Bulky asychronous transmit FIFO size in multiples of 4 quadlets. Bit 06 is MSB. Default size is 0 blocks.
16 –21	Not used		
22 – 31	BARXSIZE	R/W	Bulky asychronous receive FIFO size in multiples of 4 quadlets. Bit 22 is MSB. Default size is 125 blocks (500 quadlets).

5.47 Bulky Asynchronous Avail register (BAAVAL @ Addr 108h)

This read-only register provides the application software with the capability to read the occupancy status in quadlets for the bulky asychronous transmit and receive FIFOs. This register is cleared to 0 on a power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 03	Not used		
04 – 15	BATXAVAIL	R	Number of empty quadlet locations available in the bulky asychronous transmit FIFO. Bit 04 is MSB. Value returned on a read performed immediately after a power up or software reset is 0.
16 –19	Not used		
20 – 31	BARXAVAIL	R	Number of data quadlets available in the bulky asychronous receive FIFO. Bit 20 is MSB. Value returned on a read performed immediately after a power up or software reset is 0.

5.48 Asynchronous Application Data Transmit FIFO First and Continue (BATX @ Addr 10ch)

This write-only port provides the application software with the capability to write the quadlets of an asychronous transmit packet—except the last quadlet—to the bulky asychronous transmit FIFO.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BATXFC	W	32-bit data quadlet. Bit 00 is MSB

5.49 Asynchronous Application Data Transmit FIFO Last & Send (BATXLS @ Addr 110h)

This write-only port provides the application software with the capability to write the last quadlet of an asychronous transmit packet to the bulky asychronous transmit FIFO. This last write marks the quadlet as the last one in the packet and confirms the packet for transmission.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BATXLS	W	32-bit data quadlet. Bit 00 is MSB

5.50 Asynchronous Application Data Receive FIFO (BARX @ Addr 114h)

This write-only port allows the application software to read data from the bulky asychronous receive FIFO. If more than one quadlet is in this FIFO, each read outputs the next quadlet from this FIFO. If the FIFO is empty the last valid value is read.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BARX	R	32-bit data. Bit 00 is the MSB

5.51 Asynchronous Application Data Receive Header Register 0 (ARH0 @ Addr 118h)

This read-only register allows the application software to read the first header quadlet of a received asychronous packet header after the bulky receive FIFO control logic has copied the asychronous header into registers ARH0 to ARH3**.** This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ARH0	R	First quadlet of asychronous header. Bit 32 is MSB

5.52 Asynchronous Application Data Receive Header Register 1 (ARH1 @ Addr 11ch)

This read-only register allows the application software to read the second header quadlet of a received asychronous packet header after the bulky receive FIFO control logic has copied the asychronous header into registers ARH0 to ARH3**.** This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ARH1	R	Second quadlet of asychronous header. Bit 32 is MSB

5.53 Asynchronous Application Data Receive Header Register 2 (ARH2 @ Addr 120h)

This read-only register allows the application software to read the third header quadlet of a received asychronous packet header after the bulky receive FIFO control logic has copied the asychronous header into registers ARH0 to ARH3**.** This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ARH2	R	Third quadlet of asychronous header. Bit 32 is MSB

5.54 Asynchronous Application Data Receive Header Register 3 (ARH3 @ Addr 124h)

This read-only register allows the application software to read the fourth header quadlet of a received asychronous packet after the bulky receive FIFO control logic has copied the asychronous header into registers ARH0 to ARH3**.** This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ARH3	R	Fourth quadlet of asychronous header. Bit 32 is MSB

5.55 Asynchronous Application Data Receive Trailer (ART @ Addr 128h)

This read-only register allows the application software to read the trailer quadlet of a received asychronous packet after the bulky asychronous receive FIFO control logic has copied the trailer quadlet to this register**.** The asychrounous trailer contains the packet reception status that is added by the receiving link core. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 13	Not used		
14 – 15	SPD	R	1394 speed code of received packet 00 – 100 Mbits/s 01 – 200 Mbits/s 10 – Not valid 11 – Not valid
16 – 21	Not used		
22 – 23	ZEROFILL	R	Number of zero-fill bytes in the last quadlet of the packet data payload 00 – no zero fill bytes 01 – 1 zero fill bytes 10 – 2 zero fill bytes 11 – 3 zero fill bytes
24 – 27	Not used		
28 – 31	ACKSENT	R	The 1394 ack sent by the link receiver after receiving the packet. 0000 – Reserved 0001 – Ack complete 0010 – Ack pending 0011 – Reserved 0100 – Ack busy_X 0111 – 1100 – reserved 1101 – Ack data error 1110 – Ack type error 1111 – Reserved

5.56 Bulky Isochronous Size register (BISZ @ Addr 12Ch)

This register provides the application software with the capability to program the size in multiples of 4 quadlets, of the bulky isochronous transmit and receive FIFOs. This register is cleared to 0 on a power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 05	Not used		
06 – 15	BITXSIZE	R/W	Bulky isochronous transmit FIFO size in multiples of 4 quadlets. Bit 06 is MSB.
16 –21	Not used		
22 – 31	BIRXSIZE	R/W	Bulky isochronous receive FIFO size in multiples of 4 quadlets. Bit 22 is MSB.

5.57 Bulky Isochronous Avail register (BIAVAL @ Addr 130h)

This read-only register provides the application software with the capability to read the occupancy status in quadlets for the bulky isochronous transmit and receive FIFOs. This register is cleared to 0 on a power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 03	Not used		
04 – 15	BITXAVAIL	R	Number of empty quadlet locations available in the bulky isochronous transmit FIFO. Bit 04 is MSB
16 –19	Not used		
20 – 31	BIRXAVAIL	R	Number of data quadlets available in the bulky isochronous receive FIFO. Bit 20 is MSB

5.58 Isochronous Transmit First & Continue (BITXFC @ Addr 134h)

This write-only port provides the application software with the capability to write the quadlets of an isochronous transmit packet—except the last quadlet—to the bulky isochronous transmit FIFO.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BITXFC	W	32-bit data quadlet. Bit 00 is MSB

5.59 Isochronous Transmit Last & Send (BITXLS @ Addr 138h)

This write-only port provides the application software with the capability to write the last quadlet of an isochronous transmit packet to the bulky isochronous transmit FIFO. This last write marks the quadlet as the last one in the packet and confirms the packet for transmission.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BITXLS	W	32-bit data quadlet. Bit 00 is MSB

5.60 Isochronous Receive FIFO (BIRX @ Addr 13ch)

This read-only port allows the application software access to the bulky isochronous receive FIFO. If more than one quadlet is in this FIFO, each read outputs the next quadlet from this FIFO. If the FIFO is empty the last valid value is read. The value returned on a read immediately after power up or software reset is 0.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BIRX	R	32-bit data quadlet. Bit 00 is MSB

5.61 Isochronous Packet Received Header (IRH @ Addr 140h)

This read-only register allows the application software to read the header quadlet of a received isochronous packet header after the bulky isochronous FIFO control logic has copied the isochronous header into register IRH**.** This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	LENGTH [0:15]	R/W	Packet data length in bytes
16 – 17	TAG	R/W	Isochronous TAG field
18 – 23	CHANNUM	R/W	Isochronous channel number
24 – 27	TCODE	R/W	Isochronous tCode field
28 – 31	SY	R/W	Isochronous sync bits

5.62 Isochronous Packet received Trailer (IRT @ Addr 144h)

This read-only register allows the application software to read the trailer quadlet of a received isochronous packet after the bulky isochronous receive FIFO control logic has copied the trailer quadlet to this register**.** The isochronous packet trailer contains the packet reception status that is added by the receiving link core. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 13	Not used		
14 – 15	SPD	R	1394 speed code of received packet 00 – 100 Mbits/s 01 – 200 Mbits/s 10 – Not valid 11 – Not valid
16 – 21	Not used		
22 – 23	ZEROFILL	R	Number of zero-fill bytes in the last quadlet of the packet data payload 00 – no zero fill bytes 01 – 1 zero fill bytes 10 – 2 zero fill bytes 11 – 3 zero fill bytes
24 – 27	Not used		
28 – 31	ERRCODE	R	The 1394 ack sent by the link receiver after receiving the packet. 0000 – Reserved 0001 – Ack complete 0010 – Ack pending 0011 – Reserved 0100 – Ack busy_X 0111 – 1100 – reserved 1101 – Ack data error 1110 – Ack type error 1111 – Reserved

5.63 Receive Packet Routing Control Register (RPRC @ Addr 148h)

This register provides the application software with the capability to program and control the operation of the receive packet routing control logic. This register is cleared to 0 on power up of software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 16	Not used		
17	BACKPENDEN	R/W	Bulky ACK Pending. If set to 1, an ack pending message is sent in response to an asychronous lock/write request packet. If set to 0, an ack complete is sent in response. If an asychronous packet is not received correctly, then it is acknowledged with ack data error regardless of the BACK-PENDEN setting. BACKPENDEN defaults to 1 on power on.
18	Not used		
19	DBCCOVER	R/W	Disable data block continuity checking on receive. When set, this bit disables the data block continuity checking of incoming MPEG/DSS packets by the receive routing control logic.
20	RIDM0	R/W	Receive FIFO destination select for response packets that are matched by the expected response comparator (PHYSR @ Addr 38h) When RIDM0 = 0, the expected response packet is routed to bulky asychronous receive FIFO. When RIDM0 = 1, the expected response packet is routed to the asychronous control received FIFO.
21	SIDM0	R/W	Self-ID receive FIFO destination select. When SIDM0 = 0, the Self-ID packets are routed to broadcast receive FIFO. When SIDM0 = 1, the Self-ID packets are routed to bulky asychronous receive FIFO.
22 23	ARDM0 ARDM1	R/W R/W	Asychronous receive FIFO destination select bits (ACRX = asychronous control receive FIFO BWRX = broadcast control receive FIFO BARX = bulky data asychronous receive FIFO) When ARDM1 = 0 and ARDM0 = 0, all non-broadcast asychronous packets are routed to the ACRX. All broadcast asychronous packets are routed to the BWRX. When ARDM1 = 0 and ARDM0 = 1, the non-broadcast ROM/register space write request asychronous packets are routed to the ACRX. Broadcast ROM/register space request asychronous packets are routed to BWRX. All other asychronous packets not meeting the above decode criteria are routed to the BDARX. When ARDM1 = 1 and ARDM0 = 0, all asychronous packets are routed to the BARX. When ARDM1 = 1 and ARDM0 = 1, all non-broadcast asychronous request packets addressed to the upper half of addressable node space (dest_addr ≥ 80000,0000000). are routed to the ACRX. Broadcast asychronous request packets addressed to the upper half of addressable node space (dest_addr ≥ 80000,0000000) are routed to the BWRX. All other asychronous packets are sent to BARX.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
24	MONT0	R/W	When set, this bit enables match on tag compare for MPEG/DSS isochronous receive comparator 0
25	MONT1	R/W	When set, this bit enables match on tag compare for MPEG/DSS isochronous receive comparator 1
26	MONT2	R/W	When set, this bit enables match on tag compare for MPEG/DSS isochronous receive comparator 2
27	MONT3	R/W	When set, this bit enables match on tag compare for MPEG/DSS isochronous receive comparator 3
28	MONT4	R/W	When set, this bit enables match on tag compare for MPEG/DSS isochronous receive comparator 4
29	MONT5	R/W	When set, this bit enables match on tag compare for MPEG/DSS isochronous receive comparator 5
30	MONT6	R/W	When set, this bit enables match on tag compare for MPEG/DSS isochronous receive comparator 6
31	MONT7	R/W	When set, this bit enables match on tag compare for MPEG/DSS isochronous receive comparator 7

5.64 Bulky Asynchrounous Retry (BARTRY@ Addr 14Ch)

This register provides the application software with the capability to program the operation of the automatic retry control function for packets transmitted from the bulky asychronous transmit FIFO.The cycle timer (reg C, bit 22) must be enabled for automatic retry. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	Not used		
16 – 23	BATXRTRYINT	R/W	Number of isochronous Cycle intervals to wait between retrys.
24 – 31	BATXRTRYNUM	R/W	Number of times to retry the asychronous packet when the receiving node continues to ack the packet with a busy acknowledge.

5.65 Bulky MPEG2 (DVB) Size register (BMSZ @ Addr 150h)

The register provides the application software with the capability to program the size, in multiples of 4 quadlets, of the bulky MPEG2 (DVB) transmit and receive FIFOs. This register is cleared to 0 on a power up or software reset. This register is the same register as 158h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 05	Not used		
06 – 15	BMTXSIZE	R/W	Bulky MPEG transmit FIFO size in multiples of 4 quadlets. Bit 06 is MSB.
16 – 21	Not used		
22 – 31	BMRXSIZE	R/W	Bulky MPEG receive FIFO size in multiples of 4 quadlets. Bit 22 is MSB.

5.66 Bulky MPEG2 (DVB) Avail register (BMAVAL @ Addr 154h)

The read-only register port provides the application software with the capability to read the occupancy status, in quadlets, for the bulky MPEG transmit and receive FIFOs. When the microprocessor has access to the bulky FIFO, the available register is updated on a quadlet basis. When the bulky data interface has access to the bulky FIFO, the available register is updated on a packet basis. This register is cleared to 0 on power up or software reset. This register is the same register as 15Ch.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 03	Not used		
04 – 15	BMTXAVAIL	R	Number of empty quadlet locations available in the bulky MPEG transmit FIFO. Bit 04 is MSB
16 – 19	Not used		
20 – 31	BMRXAVAIL	R	Number of data quadlets available in the bulky MPEG receive FIFO. Bit 20 is MSB.

5.67 Bulky DSS Size register (BDSZ @ Addr 158h)

The register provides the application software with the capability to program the size, in multiples of 4 quadlets, of the bulky DSS transmit and receive FIFOs. This register is cleared to 0 on a power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 05	Not used		
06 – 15	BDTXSIZE	W	Bulky DSS transmit FIFO size in multiples of 4 quadlets. Bit 06 is MSB.
16 – 21	Not used		
22 – 31	BDRXSIZE	W	Bulky DSS receive FIFO size in multiples of 4 quadlets. Bit 22 is MSB.

5.68 Bulky DSS Avail register (BDAVAL @ Addr 15ch)

The register port provides the application software with the capability to read the occupancy status, in quadlets, for the bulky DSS transmit and receive FIFOs. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 03	Not used		
04 – 15	BDTXAVAIL	R/W	Number of empty quadlet locations available in the bulky DSS transmit FIFO. Bit 04 is MSB
16 – 19	Not used		
20 – 31	BDRXAVAIL	R/W	Number of data quadlets available in the bulky DSS receive FIFO. Bit 20 is MSB

5.69 MPEG2 (DVB) Transmit FIFO first & continue (BMTXFC @ Addr 160h)

This register provides the application software with the capability to write the quadlets of an MPEG2 (DVB) transmit packet—except the last quadlet—to the bulky MPEG2 (DVB) transmit FIFO. This register is the same register as 16Ch.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BMTXFC	W	32-bit data quadlet. Bit 00 is MSB

5.70 MPEG2 (DVB) Transmit FIFO last & send (BMTXLS @ Addr 164h)

This write-only port provides the application software with the capability to write the last quadlet of an MPEG2 (DVB) transmit packet to the bulky MPEG2 (DVB) transmit FIFO. This last write marks the quadlet as the last one in the packet and confirms the packet for transmission. This register is the same register as 170h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BMTXLS	W	last 32-bit data quadlet. Bit 00 is MSB

5.71 MPEG2 (DVB) Formatted Packet Receive FIFO (BMRX @ Addr 168h)

This read-only register port allows the application software access to the bulky MPEG2 (DVB) receive FIFO. If more than one quadlet is in this FIFO, each read outputs the next quadlet from this FIFO. If the FIFO is empty, the last valid value is read. This register is cleared to 0 on power up or software reset. This register is the same register as 174h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BMRX	R	32-bit data out. Bit 00 is MSB

5.72 DSS Transmit FIFO first & continue (BDTXFC @ Addr 16ch)

This register provides the application software with the capability to write the quadlets of an DSS transmit packet — except the last quadlet — to the bulky DSS transmit FIFO.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BDTXFC	W	32-bit data quadlet. Bit 00 is MSB

5.73 DSS Transmit FIFO last & send (BDTXLS @ Addr 170h)

This write-only port provides the application software with the capability to write the last quadlet of an DSS transmit packet to the bulky DSS transmit FIFO. This last write marks the quadlet as the last one in the packet and confirms the packet for transmission.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BDTXLS	W	Last 32-bit data quadlet. Bit 00 is MSB

5.74 DSS Formatted Packet Receive FIFO (BDRX @ Addr 174h)

This read-only register port allows the application software access to the bulky DSS receive FIFO. If more than one quadlet is in this FIFO, each read outputs the next quadlet from this FIFO. If the FIFO is empty the last valid value is read. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BDRX	R	32-bit read data. Bit 00 is MSB

5.75 MPEG2 (DVB) Receive Header (MRH @ Addr 178h)

This read-only register port allows the application software to read the isochronous header quadlet of a received MPEG2 (DVB) packet after the bulky MPEG2 (DVB) FIFO control logic has copied the isochronous header into register MRH**.** This register is cleared to 0 on power up or software reset. This register is the same register as 188h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	LENGTH [0:15]	R/W	Packet data length in bytes
16 – 17	TAG	R/W	Isochronous TAG field
18 – 23	CHANNUM	R/W	Isochronous channel number
24 – 27	TCODE	R/W	Isochronous tCode field
28 – 31	SY	R/W	Isochronous sync bits

5.76 MPEG2 (DVB) CIP Receive Header 0 (MCIPR0 @ Addr 17ch)

This read-only register port allows the application software to read the CIP0 header quadlet of a received MPEG2 (DVB) packet after the bulky MPEG2 (DVB) FIFO control logic has copied the CIP0 header into register MCIPR0**.** This register is cleared to 0 on power up or software reset. This register is the same register as 18Ch

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	0	R	Logic 0
01	0	R	Logic 0
02 – 07	SID	R	Source node ID = variable from 000000 to 111111
08 – 15	DBS	R	Data block size = 00000110
16 – 17	FN	R	Fraction number = 11
18 – 20	QPC	R	Quadlet padding count = 000
21	SPH	R	Source packet header present = 1 Source packet header not present = 0
22 – 23	RES	R	Reserved
24 – 31	DBC	R	Data block continuity counter = variable 0 to 255

5.77 MPEG2 (DVB) CIP Receive Header 1 (MCIPR1 @ Addr 180h)

This read-only register port allows the application software to read the CIP1 header quadlet of a received MPEG2 (DVB) packet after the bulky MPEG2 (DVB) FIFO control logic has copied the CIP1 header into register MCIPR1**.** This register is cleared to 0 on power up or software reset. This register is the same register as 190h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	1	R	Logic 1
01	0	R	Logic 0
02 – 07	FMT	R	Format ID = 100000
08 – 31	FDF	R	Format dependent field

5.78 MPEG2 (DVB) Receive Trailer Register (MRT @ Addr 184h)

This read-only register port allows the application software to read the trailer quadlet of a received MPEG packet after the bulky MPEG2 (DVB) receive FIFO control logic has copied the trailer quadlet to this register**.** This register is cleared to 0 on power up or software reset. This register is the same register as 194h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 13	Not used		
14 – 15	SPD	R	MPEG receive packet speed 00 – 100 Mbits/s 01 – 200 Mbits/s 10 – invalid 11 – invalid
16 – 27	Not used		
28 – 31	ERRCODE	R	MPEG receive packet error status.

5.79 DSS Formatted Packet Received Header (DRH @ Addr 188h)

This read-only register port allows the application software to read the isochronous header quadlet of a received DSS packet after the bulky DSS FIFO control logic has copied the isochronous header into register DRH**.** This register is cleared to 0 on power up or software reset. This register is the same register as 178h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	DRH	R	DSS receive packet 1394 isochronous header

5.80 DSS Formatter CIP 0 receive Register (DCIPR0 @ Addr 18ch)

This read-only register port allows the application software to read the CIP0 header quadlet of a received DSS packet after the bulky DSS FIFO control logic has copied the CIP0 header into register DCIPR0**.** This register is cleared to 0 on power up or software reset. This register is the same register as 17Ch.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	0	R	Logic 0
01	0	R	Logic 0
02 – 07	SID	R	Source node ID = variable from 000000 to 111111
08 – 15	DBS	R	Data block size = 00001001
16 – 17	FN	R	Fraction number = 10
18 – 20	QPC	R	Quadlet padding count = 000
21	SPH	R	Source packet header present = 1 Source packet header not present = 0
22 – 23	RES	R	Reserved
24 – 31	DBC	R	Data block continuity counter = variable 0 to 255

5.81 DSS Formatter CIP 1 receive Register (DCIPR1 @ Addr 190h)

This read-only register port allows the application software to read the CIP1 header quadlet of a received DSS packet after the bulky DSS FIFO control logic has copied the CIP1 header into register DCIPR1**.** This register is cleared to 0 on power up or software reset. This register is the same register as 180h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	1	R	Logic 1
01	0	R	Logic 0
02 – 07	FMT	R	Format ID = 100001
08 – 31	FDF	R	Format depended field. FDF[08] = TSF Time Shift Flag FDF[08] = 0 The stream is not time shifted FDF[08] = 1 The stream is time shifted

5.82 DSS Formatted Packet Received Trailer (DRT @ Addr 194h)

This read-only register port allows the application software to read the trailer quadlet of a received DSS packet after the bulky DSS receive FIFO control logic has copied the trailer quadlet to this register**.** This register is cleared to 0 on power up or software reset. This register is the same register as 184h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 13	Not used		
14 – 15	SPD	R	DSS receive packet speed 00 – 100 Mbits/s 01 – 200 Mbits/s 10 – invalid 11 – invalid
16 – 27	Not used		
28 – 31	ERRCODE	R	DSS receive packet error status.

5.83 DSS Receive Cell Header Register 0 (DRX0 @ Addr 198h)

This read-only register port allows the application software to read the DSS source packet header bytes 0 – 3 of a received DSS packet after the bulky DSS receive FIFO control logic has copied the bytes to this register**.** This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 07	DSS130HDR_0	R	DSS source packet header byte 0
08 – 15	DSS130HDR_1	R	DSS source packet header byte 1
16 – 23	DSS130HDR_2	R	DSS source packet header byte 2
24 – 31	DSS130HDR_3	R	DSS source packet header byte 3

5.84 DSS Receive Cell Header Register 1 (DRX1 @ Addr 19Ch)

This read-only register port allows the application software to read the DSS source packet header bytes 4 – 5 of a received DSS packet after the bulky DSS receive FIFO control logic has copied the bytes to this register**.** This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 07	DSS130HDR_4	R	DSS source packet header byte 4
08 – 15	DSS130HDR_5	R	DSS source packet header byte 5
16 – 23	DSS130HDR_6	R	DSS source packet header byte 6
24 – 31	DSS130HDR_7	R	DSS source packet header byte 7

5.85 DSS Receive Cell Header Register 2 (DRX2 @ Addr 1A0h)

This read-only register port allows the application software to read the DSS source packet header bytes 8 – 9 of a received DSS packet after the bulky DSS receive FIFO control logic has copied the bytes to this register**.** This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 07	Not used		
08 – 15	Not used		
16 – 23	DSS130HDR_8	R	DSS source packet header byte 8
24 – 31	DSS130HDR_9	R	DSS source packet header byte 9

5.86 DSS Transmit Cell Header Register 0 (DTX0 @ Addr 1A4h)

This register provides the application software with the capability to program bytes 0 – 3 of a 10-byte DSS packet header, which is automatically inserted by the FIFO control logic into the MPEG/DSS FIFO when DSS 130 transmit mode is enabled. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 07	DSS130HDR_0	R/W	DSS source packet header byte 0
08 – 15	DSS130HDR_1	R/W	DSS source packet header byte 1
16 – 23	DSS130HDR_2	R/W	DSS source packet header byte 2
24 – 31	DSS130HDR_3	R/W	DSS source packet header byte 3

5.87 DSS Transmit Cell Header Register 1 (DTX1 @ Addr 1A8h)

This register provides the application software with the capability to program bytes 4 – 7 of a 10-byte DSS packet header, which is automatically inserted by the FIFO control logic into the MPEG/DSS FIFO when DSS 130 transmit mode is enabled. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 07	DSS130HDR_4	R/W	DSS source packet header byte 4
08 – 15	DSS130HDR_5	R/W	DSS source packet header byte 5
16 – 23	DSS130HDR_6	R/W	DSS source packet header byte 6
24 – 31	DSS130HDR_7	R/W	DSS source packet header byte 7

5.88 DSS Transmit Cell Header Register 2 (DTX2 @ Addr 1ACh)

This register provides the application software with the capability to program bytes 8 – 9 of a 10-byte DSS packet header, which is automatically inserted by the FIFO control logic into the MPEG/DSS FIFO when DSS 130 transmit mode is enabled. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 07	Not used		
08 – 15	Not used		
16 – 23	DSS130HDR_8	R/W	DSS source packet header byte 8
24 – 31	DSS130HDR_9	R/W	DSS source packet header byte 9

5.89 Asychronous Header 0 for Auto Tx (AHEAD 0) @ Addr 1B0h)

This register provides the application software with the capability to program the first quadlet of an asychronous header that is used during asychronous transmit auto packetization. Please refer to Section 3.4 for more information on asynchronous header formats.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
0	AIncEn	R/W	Auto-A address increment on ack not-busy. Increments destination address by the data length. This bit is cleared to 0 on power up or software reset.
1–13	RESERVED		These bits are always read as 0s.
14 – 31	AHEAD0	R/W	Asychronous header register 0 used for Auto-A packetization. These bits are set to 00010010h on power up or software reset.

5.90 Asychronous Header 1 for Auto Tx (AHEAD(1) @ Addr 1B4h)

This register provides the application software with the capability to program the second quadlet of an asychronous header that is used during asychronous transmit auto packetization. Please refer to Section 3.4 for more information on asynchronous header formats.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	AHEAD1	R/W	Asynchronous header register 1 used for Auto-A packetization. These bits are set to 3FC10000h on power up or software reset.

5.91 Asychronous Header 2 for Auto Tx (AHEAD(2) @ Addr 1B8h)

This register provides the application software with the capability to program the third quadlet of an asychronous header that is used during asychronous transmit auto packetization. Please refer to Section 3.4 for more information on asynchronous header formats.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	AHEAD2	R/W	Asynchronous header register 2 used for Auto-A packetization. These bits are set to 00000000h on power up or software reset.

5.92 Asynchrous Header 3 for Auto Tx (AHEAD(3) @ Addr 1BCh)

This register provides the application software with the capability to program the fourth quadlet of an asychronous header that is used during asychronous transmit auto packetization. Please refer to Section 3.4 for more information on asynchronous header formats.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	AHEAD3	R/W	Asynchronous header register 3 used for Auto-A packetization. These bits are set to 00080000h on power up or software reset.

5.93 Isochronous Header for Auto Tx (IHEAD0 @ Addr 1C0h)

This register provides the application software with the capability to program the header quadlet of an isochronous header that is used during isochronous transmit auto packetization.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	LENGTH [0:15]	R/W	Packet data length in bytes. These bits are set to 0010h on power up or software reset
16 – 17	TAG	R/W	Isochronous TAG number. These bits are set to 0 on power up or software reset.
18 – 23	CHANNUM	R/W	Isochronous channel number. These bits are set to 0 on power up or software reset.
24 – 25	Not used		
26 – 27	SPD	R/W	Isochronous speed to send this packet (00 = 100 mbits/s, 01 = 200 Mbits/s). These bits are set to 01 on power up or software reset.
28 – 31	SY	R/W	Isochronous sync bits. These bits are set to 0 on power up or software reset.

5.94 Packetizer Control (PKTCTL @ Addr 1C4h)

This register provides the application software with the capability to configure and control the operation of the packetizer functionality.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	CYCTSEREN	R/W	Cycle timer serial output mode. When this bit is set to 1, the MDAlt register used for serial cycle timer shift register. Set to 1 on power up or software reset.
01	Not used		
02	MTEST	R/W	MPEG2 packetizer test mode. Enabled when one. When this bit is set to 1, it sends MPEG2/DSS test packets based on the class that is selected. MPEG2/DSS data is an incrementing count. One MPEG/DSS packet sent per isochronous cycle. This bit is cleared to 0 on power up or software reset.
03	ITEST	R/W	Bulky isochronous packetizer test mode. Whe this bit is set to 1,it sends bulky isochronous test packets based on the Ihead register. Bulky isochronous data is an incrementing count. One bulky isochronous packet sent per isochronous cycle. This bit is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
04	ATEST	R/W	Bulky asychronous packetizer test mode. When this bit is set to 1, it sends bulky asychronous test packets based on Ahead registers. Bulky asychronous data is an incrementing count. No auto retrys allowed. When in bulky asychronous test mode, one asychronous packet is sent per isochronous cycle to prevent continuous asychronous packets. This bit is cleared to 0 on power up or software reset.
05 – 17	Not used		
18	MDCTFRRG	R/W	When this bit is set to 1, the MPEG2/DSS packetizer control taken from Mdalt. When enabled: {0000,Mdalt[0:7]} = number of quadlets to transmit. Mdalt[8:23] = data length loaded into header. Mdalt[24:31] = data block increment used.In this mode, the packet transmitted is always exactly per the class. This bit is cleared to 0 on power up or software reset.
19	FIFOFLEN	R/W	When this bit is set, the Master FIFO Flush is enabled. This bit is set to 1 on power up or software reset.
20	ISOGOFLEN	R/W	Enable flushing of MPEG2 FIFO when the isochronous cycle has begun and the isochronous state machine is in a non-idle state. Set to 1 on power up or software reset.
21	FIFOPHSEN	R/W	Enable flushing of MPEG2FIFO when the packetizer expects a timestamp from FIFO but the TimeStampValid signal is false. This bit is set to 1 on power up or software reset.
22	CFRPKTRST	R/W	CFR reset of the packetizer state machines. This is a self clearing bit and is cleared to 0 on power up or software reset.
23	QPCWEN	R/W	When this bit is set to 1, enable microprocessor writing of Quadlet per Cell register. This bit is cleared to 0 on power up or software reset.
24	PRBWEN	R/W	When this bit is set to 1, enable microprocessor writing of PktTestMuxOutAccess register for R/W test. This bit is cleared to 0 on power up or software reset.
25	FMTWEN	R/W	When this bit is set to 1, enable microprocessor writing of CIP1 Fmt field. This bit is cleared to 0 on power up or software reset.
26	DBCWEN	R/W	When this bit is set to 1, enable microprocessor writing of CIP0 DBC field. This bit is cleared to 0 on power up or software reset.
27	SPHWEN	R/W	When this bit is set to 1, enable microprocessor writing of CIP0 SPH field. This bit is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
28	FNWEN	R/W	When this bit is set to 1, enable microprocessor writing of CIP0 Fn field. This bit is cleared to 0 on power up or software reset.
29	DBSWEN	R/W	When this bit is set to 1, enable microprocessor writing of CIP0 DBS field. This bit is cleared to 0 on power up or software reset.
30	SLDWEN	R/W	When this bit is set to 1, enable microprocessor writing of CIP0 SID field. Cleared to 0 on power up or software reset.
31	LENWEN	R/W	When this bit is set to 1, enable microprocessor writing of MXH DataLength field. Cleared to 0 on power up or software reset.

5.95 MPEG2 (DVB) Transmit Header Register (MXH @ Addr 1C8h)

This register provides the application software with the capability to program the 1394 header quadlet that is used during MPEG2 (DVB) transmit auto packetization. This register is the same register as 1D4h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	length[0:15]	R/W	Packet data length in bytes. When LENWEN @ Addr 1C4h=1, the microprocessor must load. When LENWEN=0, length is autoloaded by packetizer. These bits are unaffected by BusReset, but are set to 0008h on power up or software reset.
16 – 17	TAG	R/W	Isochronous tag number. The microprocessor must load. These bits are unaffected by bus reset, but are set to 01 on power up or software reset.
18 – 23	chanNum	R/W	Isochronous channel number. The microprocessor must load. These bits are unaffected by bus reset.
24 – 25	RESERVED		Logic 0
26 – 27	spd	R/W	Isochronous speed to send this packet. The microprocessor must load (00 = 100 mbits/s, 01 = 200 Mbits/s). These bits are unaffected by bus reset but are cleared to 00 on power up or software reset.
28 – 31	sy	R/W	Isochronous sync bits. Resets to 0h. The microprocessor must load. These bits are unaffected by bus reset.

5.96 MPEG2 (DVB) CIP Transmit Header 0 (MCIPX0 @ Addr 1CCh)

This register provides the application software with the capability to program the CIP0 header quadlet that is used during MPEG2 (DVB) transmit auto packetization. This register is the same register as 1D8h.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	0	R	Logic 0
01	0	R	Logic 0
02 – 07	SID	R/W	When SIDWEN=1, the microprocessor must load. When SIDWEN=0, SID is auto-updated with all Phy register 0 transfers.
08 – 15	DBS	R/W	When DBSWEN=1, the microprocessor must load. DBSWEN=0, Defaults to 6(MPEG)/9(DSS). Unaffected by bus reset
16 – 17	FN	R/W	FNWEN=1, microprocessor must load. FNWEN=0, Defaults to 3(MPEG)/2(DSS). Unaffected by bus reset
18 – 20	QPC	R/W	Microprocessor must load. These bits are unaffected by bus reset
21	SPH	R/W	When SPHWEN=1, the microprocessor must load. When SPHWEN=0, auto set to 1 when TS included in the packet This bit is unaffected by bus reset.
22 – 23	RES	R/W	Logic 0
24 – 31	DBC	R/W	When DBCWEN=1, the microprocessor must load. When DBCWEN=0, the auto loaded/auto incremented by packetizer. These bits are unaffected by bus reset.

5.97 MPEG2 (DVB) CIP Transmit Header 1 (MCIPX1 @ Addr 1D0h)

This register provides the application software with the capability to program the CIP1 header quadlet that is used during MPEG2 (DVB) transmit auto packetization. This register is the same register as 1DCh.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	1	R/W	Logic 1
01	0	R/W	Logic 0
02 – 07	FMT	R/W	When FMTWEN=1, microprocessor must load. When FMTWEN=0, defaults to 20(MPEG)/21(DSS) These bits are unaffected by bus reset
08 – 31	FDF	R/W	Microprocessor must load. These bits are unaffected by bus reset.

The CIP1 transmit header information for this packet.

5.98 DSS Formatted Isochronous Data Transmit Header (DXH @ Addr 1D4h)

DSS shadow register. See MXH @ 1C8h.

5.99 DSS Formatter CIP 0 transmit Register (DCIPX0 @ Addr 1D8h)

DSS shadow register. See MCIPX0 @ 1CCh.

5.100DSS Formatter CIP 1 transmit Register (DCIPX1 @ Addr 1DCh)

DSS shadow register. See MCIPX1 @ 1D0h.

5.101MDAltCont (MDALT @ Addr 1E0h)

This register provides the application software with the multifunction capability as defined in the functional description. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	MDALTCONT	R/W†	Provides: 1. Alternate control of the packetizer when enabled with register PKTCTL bit MDCTFRRG 2. serial cycle timer shift register when enabled with register PKTCTL bit CYCTSEREN 3. Delayed bulky Isochronous transmit enable. When enabled with register PKTCTL bits IWAITFCYC[0:1] bulky Isochronous transmit begins when the cycle timer matches the value held in MDALTCONT.

† This register is R/W only when the CYCTSERN bit = 0 in the PKCTL register (reg 1C4h).

5.102Microprocessor Control Register (MDCTL @ Addr 1E4)

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 23	Not used		
24 – 31	QPERCELL	R/W†	*QPERCELL* indicates the quadlets per cell for MPEG2/DSS packets. For MPEG2 Class 3, the default QPERCELL = 47 (188 bytes); for DSS Class 3, the default QPERCELL = 35 (140 bytes). This register allows the host to change the number of quadlets per cell from the standard value used.

† This register is R/W only when the QPCWEN bit in the PKCTL register (reg 1C4h) i set.

5.103Reserved (RSVD @ Addr 1e8h)

This register is reserved.

5.104Microprocessor Input / Output Control Register (IOCR @ Addr 1ECh)

BIT No.	BIT NAME		BIT VALUE SETTING MEANING		POWER UP DEFAULT SETTING		
	Symbol	**Description**	**Value = 1**	**Value = 0**	**TMS320AV7000**	**Mot68000**	**Intel8051**
0	**MCMP8**	Micro-processor bus is 8/16 bit access	Byte access	Word access	Word access		Byte access
1	**BeCtl**	Big endian control	Big endian†	Little endian	Big endian		Little endian
2	**IntPol**	Interrupt polarity control	High true	Low true	Low true		
3	**RDY PushPull**	*RDY* output signal control	Active push/ pull	3-state	High-impedance state		N/A‡
4	**INT PushPull**	*INT* output signal control	Active push/ pull	3-state	Active push/pull	High-impedance state	
5	**Blind Access**	Blind access enable/ disable	Enable blind access mode	Disable blind access mode	Enable blind access mode	Disable blind access mode	Enable blind access mode
6	**Data Invarnt**	Data invariant endianess control	Data invariant	Address invariant	Data invariant		
7	**RDYPol**	*RDY* output polarity control	High true	Low true	Low true		
8 – 31	Not used. All these bits (8–31) are cleared to 0 on power up or software reset.						

† When the BeCtl bit is set to “1” (Big Endian), the DataInvarnt bit setting has no effect.

‡ Although there is no RDY line connection in Intel 8051 Mode, reading the IOCR.RDYPushPull still returns the value of “0” for this bit.

5.105Blind Access Status Register (BASTAT @ Addr 1F0h)

This register provides the external microprocessor a means to check whether the current blind read/write access to the chip is complete. This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 06	0	R	Logic 0
07	BACMP	R	When this bit is set to 1, the current blind access read/write process (to all address other than 1F0h, 1F4h and 1ECh) is complete and if read, the data returned is ready in the BAHR (blind access holding register).
08 – 14	0	R	Logic 0
15	BACMP	R	Mirror bit 7 function
16 – 22	0	R	Logic 0
23	BACMP	R	Mirror bit 7 function
24 – 30	0	R	Logic 0
31	BACMP	R	Mirror bit 7 function

5.106Blind Access Holding Register (BAHR @ Addr 1F4h)

This register holds the quadlet data returned for the last blind access read process upon BACMP bit of BASTAT register is set (except read to 1F0h, 1F4h and 1ECh). This register is cleared to 0 on power up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	BAHR bits	R	BAHR Quadlet Data

5.107Reserved Register (RESERVED @ Addr 1F8h)

This register is reserved.

5.108Software Reset Register (SRES @ Addr 1FCh)

Any write access to this register generates a device reset regardless what kind of data is written. The internal reset pulse has a width of four SCLK cycle.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	SRES	W	

6 Electrical Characteristics

6.1 Absolute Maximum Ratings Over Free-Air Temperature Range (Unless Otherwise Noted)†

Supply voltage range, V_{CC} . . . . . –0.5 V to 3.6 V
Supply voltage range, $V_{CC}5V$ . . . . . –0.5 V to 5.5 V
Input voltage range, V_I . . . . . –0.5 V to $V_{CC}5V$ + 0.5 V
Output voltage range, V_O . . . . . –0.5 V to $V_{CC}5V$ + 0.5 V
Input clamp current, I_{IK} (V_I < 0 V or V_I > V_{CC}) (see Note 1) . . . . . ±20 mA
Output clamp current, I_{OK} (V_O < 0 V or V_O > V_{CC}) (see Note 2) . . . . . ±20 mA
Continuous total power dissipation, P_D . . . . . (see Dissipation Rating Table)
Operating free-air temperature range, T_A . . . . . 0°C to 70°C
Storage temperature range, T_{stg} . . . . . –65°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. This parameter applies to external input and bidirectional buffers. For 5-V tolerant terminals use $V_I > V_{CC}5V$.
2. This parameter applies to external output and bidirectional buffers. For 5-V tolerant terminals use $V_O > V_{CC}5V$.

DISSIPATION RATING TABLE

PACKAGE	$T_A \le 25°C$ POWER RATING	DERATING FACTOR ABOVE T_A = 25°C	T_A = 70°C POWER RATING
PZ	1500 mW	16.9 mW/°C	739 mW

PACKAGE THERMAL CHARACTERISTICS†

PARAMETER		TEST CONDITIONS	PZ PACKAGE MIN	PZ PACKAGE NOM	PZ PACKAGE MAX	UNIT
$R_{\theta JA}$	Junction-to-ambient thermal impedance	Board mounted, No air flow		59		°C/W
$R_{\theta JC}$	Junction-to-case thermal impedance			13		°C/W
T_J	Junction temperature				115	°C

† Thermal characteristics very depending on die and leadframe pad size as well as mold compound. These values preresent typical die and pad sizes for the respective packages. The R value decreases as the die or pad sizes increases. Thermal values represent PWB bands with minimal amounts of metal.

6.2 Recommended Operating Conditions

		MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}		3	3.3	3.6	V
Supply voltage, $V_{CC}5V$	5-V tolerant	3	4.5	5.5	V
Input voltage, V_I		0		$V_{CC}5V$	V
Output voltage, V_O†		0		$V_{CC}5V$	V
High-level input voltage, V_{IH}		2		$V_{CC}5V$	V
Low-level input voltage, V_{IL}		0		0.8	V
Input transition time, (t_r, t_f) (10% to 90%)		0		6	ns
Operating free-air temperature, T_A		0	25	70	°C
Virtual junction temperature, T_{JC}‡		0	25	115	°C

† This applies to external output buffers.
‡ The junction temperatures listed reflect simulation conditions. The absolute maximum junction temperature is 150°C. The customer is responsible for verifying the junction temperature.

6.3 Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature (Unless Otherwise Noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	TTL/LVCMOS	I_{OH} = –8 mA,†	V_{CC} – 0.6			V
			I_{OH} = –4 mA‡	V_{CC} – 0.6			
V_{OL}	Low-level output voltage	TTL/LVCMOS	I_{OL} = 8 mA,†			0.5	V
			I_{OL} = 4 mA‡			0.5	
I_{IL}	Low-level input current§	$V_I = V_{IL}$(min)				–20	µA
I_{IH}	High-level input current	$V_I = V_{IH}$(max)				20	µA
I_{OZ}	High-impedance-state output current¶	$V_O = V_{CC}$ or GND				±20	µA
$I_{CC(q)}$	Static supply current	I_O = 0			225		µA

† This test condition is for terminals D0 – D3, CTL0, CTL1, LREQ, and CONTENDER
‡ This test condition is for terminals BDI0 – BDI7, TD0, BDO0– BDO7, STAT0 – STAT3, BDOF0 – BDOF2, MCAD0 – MCAD15, RDY, INT_Z, BDIF0 – BDIF2.
§ This specification only applies when pull up and pull down terminator is turned off.
¶ Three-state output must be in high-impedance mode.

7 MPEG2Lynx Power Consumption

The TSB12LV41A has a maximum average power dissiaption of 454.62 mW + P_{FIFO}, where P_{FIFO} is the power consumed by the FIFOs.

The bulky FIFO is made up of four 2K-byte RAMs. The control FIFO is made from a single RAM. The power of the FIFOs (in Watts) can be calculated using the following fromula.

Power of single RAM:

$$P1 = [37.6\ (rd1) + 27.7\ (wr1) + 8.8\ (1 - rd1 - wr1)] \left(3.6V^2\right) (50\ MHz)$$

Power of one 2K-byte RAM:

$$P2 = [109.6\ (rd1) + 92.4\ (wr1) + 15.7\ (1 - rd1 - wr1)] \left(3.6V^2\right) (50\ MHz)$$

Where:

rd1 is a value from 0 to 1 indicating the fraction of time the FIFO is in read mode.
wr1 is a value from 0 to 1 indicating the fraction of time the FIFO is in write mode.

For a typical case (based on simuations) rd1 = 0.8 and wr1 = 0.3:

Power of single RAM:

$$P1 = 7.56\ mW$$

Power of one 2K-byte RAM:

$$P2 = 16.52\ mW$$

For applications where the rd1 = 0.8 and wr1 = .3 for both the bulky FIFO and the control FIFO the total power dissipated is:

$$Ptotal = 454.62\ mW + P1 + (4 \times P2)$$

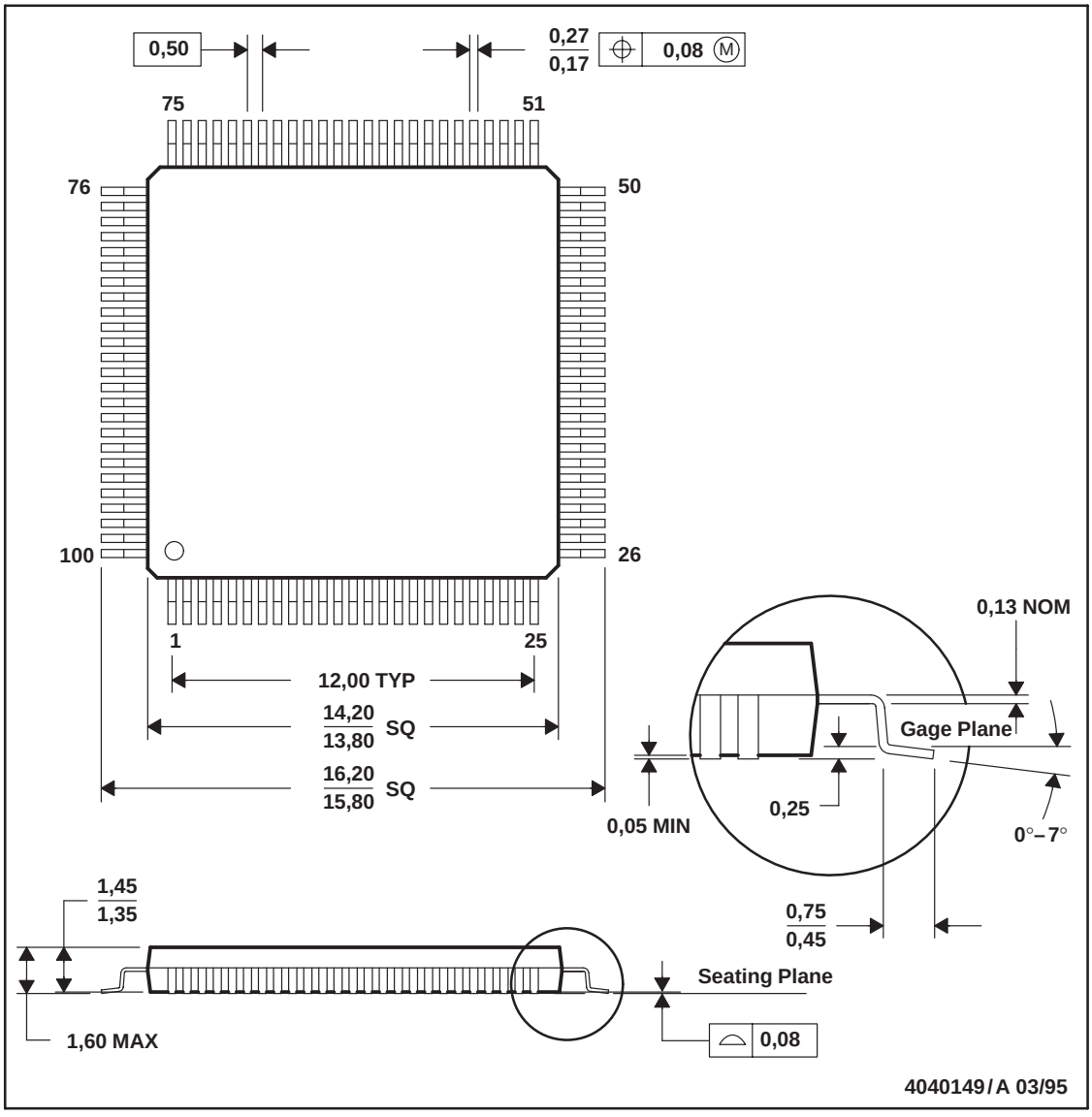
$$Ptotal = 454.62\ mW + 7.56\ mW + (4 \times 16.52\ mW) = 528.3\ mW$$

There are two test conditions for measuring the consumed power: 1) The bulky transmit FIFO is loaded with 30 MPEG2 cells (188 bytes). They are transmitted within seven isochronous cycles using bandwidth class 7 for transmit. 2) Reads and writes to the control FIFO are equal to the bulky FIFO.

8 Mechanical Information

The TSB12LV41A is packaged in a high-performance 100-pin PZ package. The following shows the mechanical dimensions of the PZ package.

PZ (S-PQFP-G100) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MO-136

Appendix A
Receive Operation Examples

A.1 Asynchronous Receive

Receiving asynchronous packets is discussed in detail in Section 3.2.1 of this data manual. These examples are **not** the de facto register settings for every system. Instead, they should be used as a guideline for configuring the MPEG2Lynx to meet an individual system's needs.

A.1.1 Receiving Asynchronous Data to the Bulky Asynchronous FIFO (Bulky Data Interface)

This example shows how to setup MPEG2Lynx registers for receiving all asynchronous data to the bulky asynchronous FIFO accessed by the bulky data interface.

Table A–1. Receiving Asynchronous Data to the Bulky Asynchronous FIFO (Bulky Data Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register 104 (Bulky *A* Size Register)	0000 00C8h	This sets the bulky asynchronous receive FIFO to 3200 bytes (decimal)
Register 148 (Receive Packet Router)	0000 0200h	This register routes all received asynchronous received packets to the bulky asynchronous receive FIFO.
Register EC (Asynchronous/Isochronous Application Data Control Register)	2000 0008h	This enables the bulky asynchronous receive FIFO to receive data. It also selects the Bulky Data Interface as the bulky asynchronous receive FIFO destination.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.

A.1.2 Receiving Asynchronous Data to the Asynchronous Control FIFO

This example shows how to setup the MPEG2Lynx registers to receive asynchronous data to the asynchronous control FIFO. The 256-byte asynchronous control FIFO is made up of three parts: the asynchronous control receive FIFO, the broadcast receive FIFO, and the asynchronous control transmit FIFO.

In this example, all broadcast asynchronous packets are received at the broadcast receive write FIFO (BWRF). All non-broadcast asynchronous packets are received at the asynchronous control receive FIFO (ACRX).

This example also sets the microprocessor port as the FIFO destination.

Table A–2. Receiving Asynchronous Data to the Asynchronous Control FIFO

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register 50 (Asynchronous Control Data Receive FIFO Status)	0000 0A95h	This sets the asynchronous control receive FIFO to 84 bytes (decimal) It also sets the broadcast receive FIFO to 84 bytes.
Register C0 (Asynchronous Control Data Receive FIFO)		This allows the application software to read the contents of ACRX one quadlet at a time.
Register C4 (Broadcast Write Receive FIFO)		This allows the application software to read the contents of BWRX one quadlet at a time.
Register 148 (Receive Packet Router)	0000 0000h	This routes all asynchronous non-broadcast packets to the asynchronous control receive FIFO. All broadcast packets are routed to the broadcast receive FIFO. Self–IDs are routed to the BWRX FIFO.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

A.2: Unformatted Isochronous Receive

Receiving isochronous packets is discussed in detail in Section 3.2.2 of this data manual. These examples are **not** the de facto register settings for every system. Instead, they should be used as a guideline for configuring the MPEG2Lynx to meet an individual system's needs.

Please note that there must be a cycle master on the bus to receive or transmit isochronous data. To make MPEG2Lynx cycle master, you must set the appropriate bits in registers Ch and 34h. For a system using TSB21LV03A PHY and MPEG2Lynx, an example of these register settings for MPEG2Lynx would be:

Register Ch (Link Control Register) = C407 0A40h

Register 34h (Phy Access Register) = 41C6 0000h

A.2.1 Receiving Isochronous Data to the Bulky Isochronous FIFO (Bulky Data Interface)

Table A–3. Receiving Isochronous Data to the Bulky Isochronous FIFO (Bulky Data Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register 12C (Bulky I Size Register)	0000 00C0h	This sets the bulky isochronous receive register to 3072 bytes (decimal).
Register C (Link Control Register)	C407 0A40h	This register sets the MPEG2Lynx to try to become cycle master AND selects port 1 for receive.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register 20 (Isochronous Receive Comparator Register 0)	0000 0000h	This selects channel 0 for the tag and channel numbers on receiver compare.
Register EC (Asynchronous/Isochronous Application Data Control Register)	8000 0800h	This sets the MPEG2Lynx to receive all isochronous data (including header and packet trailer) to the bulky data interface.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.

A.2.2 Receiving Isochronous Data to the Bulky Isochronous FIFO (Microprocessor Interface)

Table A–4. Receiving Isochronous Data to the Bulky Isochronous FIFO (Microprocessor Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register 12C (Bulky I Size Register)	0000 00C0h	This sets the bulky isochronous receive register to 3072 bytes (decimal).
Register C (Link Control Register)	C407 0A40h	This register sets the MPEG2Lynx to try to become cycle master AND selects port 1 for receive.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register 20 (Isochronous Receive Comparator Register 0)	0000 0000h	This selects 0 for the tag and 0 for the channel numbers on receiver compare.
Register EC (Asynchronous/Isochronous Application Data Control Register)	8000 1000h	This sets the MPEG2Lynx to receive only isochronous data to the microprocessor interface. Headers are stripped from the data before the data is placed in the FIFO.
Register 13C (Isochronous Receive FIFO)		This read only register allows the application software to read data out of the bulky isochronous receive FIFO one quadlet at a time.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

A.3 MPEG2/DSS Formatted Isochronous Receive

Receiving MPEG2/DSS formatted isochronous packets is discussed in detail in Section 3.2.3 of this data manual. These examples are **not** the de facto register settings for every system. Instead, they should be used as a guideline for configuring the MPEG2Lynx to meet an individual system's needs.

MPEG2/DSS data can ONLY be received at port 0. Therefore register Ch must have a format similar to:

Register C (Link Control Register) = C407 0A80h.

Please note that there are different registers settings in the MPEG2Lynx device for MPEG2 or DSS function. They access the same logic and FIFOs within the MPEG2Lynx but have different addresses.

The same requirement for a cycle master on the bus exists as for isochronous receives.

A.3.1 Receiving MPEG2 (DVB) Data to the Bulky MPEG2 FIFO (Bulky Data Interface)

Table A–5. Receiving MPEG2 (DVB) Data to the Bulky MPEG2 FIFO (Bulky Data Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register C (Link Control Register)	C407 0A80h	This register sets the MPEG2Lynx to try to become cycle master and selects port 0 for receiving.
Register 20 (Isochronous Receive Comparator Register 0)	4000 0000h	This selects 1 for the tag value and 0 for the channel number on receive compare.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.
Register F0 (MPEG2 Formatter Control Register)	8103 8088h	This register sets the MPEG2Lynx to receive MPEG2 (DVB) packets only! –MPEG2 mode enabled –Receive to bulky data FIFO –Class 3 (1 pkt. Per iso cycle) –Receive aging –Headers stripped
Register 148 (Receive Packet Router)	0000 0080h	Match received data on TAG at port 0.
Register 150 (Bulky MPEG Size Register)	0000 00C0h	This sets the bulky MPEG2 receive register to 3072 bytes (decimal).
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

A.3.2 Receiving DSS Data to the Bulky DSS FIFO (Bulky Data Interface)

Table A–6. Receiving DSS Data to the Bulky DSS FIFO (Bulky Data Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register C (Link Control Register)	C407 0A80h	This register sets the MPEG2Lynx to try to become cycle master and selects port 0 for receiving.
Register 20 (Isochronous Receive Comparator Register 0)	4000 0000h	This selects 1 for the tag value and 0 for the channel number on receive compare.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.
Register F4 (DSS Formatter Control Register)	8103 8088h	This register sets the MPEG2Lynx to receive DSS packets only! –DSS 140 mode enabled –Receive to bulky data FIFO –Class 3 (1 pkt. Per iso cycle) –Receive aging –Headers stripped
Register 148 (Receive Packet Router)	0000 0080h	Match received data on TAG at port 0.
Register 158 (Bulky DSS Size Register)	0000 00C0h	This sets the bulky DSS receive register to 3072 bytes (decimal).
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

A.3.3 Receiving MPEG2 (DVB) Data to the Bulky MPEG2 (DVB) FIFO (Microprocessor Interface)

Table A–7. Receiving MPEG2 (DVB) Data to the Bulky MPEG2 (DVB) FIFO (Microprocessor Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register C (Link Control Register)	C407 0A80h	This register sets the MPEG2Lynx to try to become cycle master and selects port 0 for receiving.
Register 20 (Isochronous Receive Comparator Register 0)	4000 0000h	This selects 1 for the tag value and 0 for the channel number on receive compare.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.
Register F0 (MPEG2 Formatter Control Register)	8103 8008h	This register sets the MPEG2Lynx to receive MPEG2 (DVB) packets only! –MPEG2 mode enabled –Microprocessor has access to bulky FIFO –Class 3 (1 pkt. Per iso cycle) –Receive aging –Headers stripped
Register 148 (Receive Packet Router)	0000 0080h	Match received data on TAG = 1 at port 0.
Register 150 (Bulky MPEG Size Register)	0000 00C0h	This sets the bulky MPEG2 receive register to 3072 bytes (decimal).
Register 168 (Formatted Packet Receive FIFO)		This read only register allows the application software to read data out of the bulky MPEG2 (DVB) receive FIFO one quadlet at a time.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

A.3.4 Receiving DSS Data to the Bulky DSS FIFO (Microprocessor Interface)

Table A–8. Receiving DSS Data to the Bulky DSS FIFO (Microprocessor Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register C (Link Control Register)	C407 0A80h	This register sets the MPEG2Lynx to try to become cycle master and selects port 0 for receiving.
Register 20 (Isochronous Receive Comparator Register 0)	4000 0000h	This selects 1 for the tag value and 0 for the channel number on receive compare.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register F4 (DSS Formatter Control Register)	8103 8208h	This register sets the MPEG2Lynx to receive DSS packets only! –DSS 130 mode enabled –Microprocessor has access to bulky FIFO –Class 3 (1 pkt. Per iso cycle) –Receive aging –Headers stripped
Register 148 (Receive Packet Router)	0000 0080h	Match received data on TAG at port 0.
Register 158 (Bulky DSS Size Register)	0000 00C0h	This sets the bulky DSS receive register to 3072 bytes (decimal).
Register 168 (Formatted Packet Receive FIFO)		This read only register allows the application software to read data out of the bulky DSS receive FIFO one quadlet at a time.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

Appendix B
Transmit Operation Examples

B.1: Asynchronous Transmit

Transmitting asynchronous packets is discussed in detail in Sections 3.1.1 and 3.1.2 of this data manual. These examples are **not** the de facto register settings for every system. Instead, they should be used as a guideline for configuring the MPEG2Lynx to meet an individual system's needs.

For all transmissions, the transmit enable (TXEN, bit 00 register C) **must** be set.

The asynchronous headers for transmit are fully explained in Section 3.4. The headers provided below (registers 1B0–1BC) are only examples and may not work for all systems.

B.1.1 Transmitting Asynchronous Data Packets (Bulky Data Interface)

This example shows how to setup MPEG2Lynx registers to transmit asynchronous data packets via the bulky data interface. The MPEG2Lynx automatically inserts the headers.

Table B–1. Transmitting Asynchronous Data Packets (Bulky Data Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.
Register EC (Asynchronous/Isochronous Application Data Control Register)	1000 0084h	This enables the bulky asynchronous transmit FIFO to receive data. It also selects the bulky data interface as the data source. It also enable asynchronous header insert mode.
Register 104 (Bulky *A* Size Register)	00C8 0000h	This sets the bulky asynchronous transmit FIFO to 3200 bytes (decimal)
Register 1B0 (Asynch Header 0 for Auto Transmit)	1001 0010h	Speed = 200Mbps tCode=1 (Write Request)
Register 1B4 (Asynch Header 1 for Auto Transmit)	FFC2 0000h	Destination ID=FFC2
Register 1B8 (Asynch Header 2 for Auto Transmit)	0000 0000h	Destination Offset = 0
Register 1BC (Asynch Header 3 for Auto Transmit)	0008 0000h	Data Length = 8 bytes
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

B.1.2. Transmitting Asynchronous Control Packets

This example shows how to setup the MPEG2Lynx registers to transmit asynchronous control data from the asynchronous control transmit FIFO (ACTX FIFO). The 256-byte asynchronous control FIFO is made up of three parts: the asynchronous control receive FIFO, the broadcast receive FIFO, and the asynchronous control transmit FIFO.

A discussion on transmitting asynchronous control packets is included in Section 3.1.1 of this data manual. Details on which registers are used for transmission are included there.

Table B–2. Transmitting Asynchronous Control Packets

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register 44 (Asynchronous Control Data Transmit FIFO)	0000 0014h	This sets the asynchronous control transmit FIFO to 80 bytes (decimal)
Register 80 (Asynchronous Control Data Transmit FIFO First)		This write–only register writes the first quadlet of a packet to the asynchronous control transmit FIFO. The application must provide all asynchronous headers (as described in Section 3.4, *Asynchronous Data Formats*) with the data.
Register 84 (Asynchronous Control Data Transmit FIFO Continue)		This write–only register writes the remaining quadlets (except for the last quadlet) to the asynchronous control transmit FIFO
Register 8C (Asynchronous Control Data Transmit FIFO Continue and Update)		This write–only register writes the last quadlet of the packet to the asynchronous control transmit FIFO and confirms the entire packet for transmission.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This register sets up the microcontroller port for Motorola 68000 mode

B.2 Unformatted Isochronous Transmit

Transmitting isochronous packets is discussed in detail in Section 3.1.2 of this data manual. These examples are **not** the de facto register settings for every system. Instead, they should be used as a guideline for configuring the MPEG2Lynx to meet an individual systems needs.

For all transmissions, the transmit enable (TXEN, bit 00 register C) MUST be set.

Please note that there must be a cycle master on the bus to receive or transmit isochronous data. To make MPEG2Lynx cycle master, you must set the appropriate bits in registers Ch and 34h. For a system using TSB21LV03A PHY and MPEG2Lynx, an example of these register settings would be:

Register Ch (Link Control Register) = C407 0A00h

Register 34h (Phy Access Register) = 41C6 0000h

A complete discussion of isochronous transmit headers is included in Section 3.4 of this data manual. The headers below are just examples and may not work for every system.

B.2.1 Transmitting Isochronous Data, Headers Auto-Inserted (Bulky Data Interface)

Table B–3. Transmitting Isochronous Data, Headers Auto-Inserted (Bulky Data Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register 12C (Bulky I Size Register)	00C0 0000h	This sets the bulky isochronous transmit register to 3072 bytes (decimal).
Register C (Link Control Register)	C407 0A00h	This register sets the MPEG2Lynx to try to become cycle master.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register EC (Asynchronous/Isochronous Application Data Control Register)	4000 8400h	This enables the bulky isochronous transmit FIFO. This also enables automatic isochronous header insert. This also grants write access to the bulky data interface.
Register 1C0 (ISO Header for Auto Transmit)	0008 0210h	This register sets the header that is inserted during automatic packetization. Data Length = 8 Tag = 0 Channel Number = 2 Speed = 200Mbps
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.

B.2.2 Transmitting Fully-Formatted Isochronous Data (Microprocessor Interface)

Table B–4. Transmitting Fully-Formatted Isochronous Data (Microprocessor Interface)

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register 12C (Bulky I Size Register)	00C0 0000h	This sets the bulky isochronous transmit register to 3072 bytes (decimal).
Register C (Link Control Register)	C407 0A00h	This register sets the MPEG2Lynx to try to become cycle master.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register EC (Asynchronous/Isochronous Application Data Control Register)	4000 0000h	This sets the MPEG2Lynx to transmit isochronous data from the microprocessor interface. The data must include the 1394 isochronous header.
Register 134 (Iso Transmit First and Continue)		The write only register allows the application to write all quadlets (except the last) to the bulky isochronous transmit FIFO via this register. The application must provide the isochronous header (as described in Section 3.5) with the data.
Register 138 (Iso Transmit Last and Send)		The write only register allows the application to write the last quadlet of an isochronous packet to this register to the bulky isochronous transmit FIFO. The entire packet is confirmed into the FIFO and transmitted.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

B.3 MPEG2/DSS Formatted Isochronous Transmit

Transmitting MPEG2/DSS packets is discussed in detail in Section 3.1.3 of this data manual. These examples are **not** the de facto register settings for every system. Instead, they should be used as a guideline for configuring the MPEG2Lynx to meet an individual system's needs.

For all transmissions, the transmit enable (TXEN, bit 00 register C) MUST be set.

Please note that there must be a cycle master on the bus to receive or transmit isochronous data. To make MPEG2Lynx cycle master, you must set the appropriate bits in registers Ch and 34h. For a system using TSB21LV03A PHY and MPEG2Lynx, an example of these register settings would be:

Register Ch (Link Control Register) = C407 0A00h

Register 34h (Phy Access Register) = 41C6 0000h

Please note that there are different registers settings in the MPEG2Lynx device for MPEG2 or DSS function. They access the same logic and FIFOs within the MPEG2Lynx but have different addresses.

B.3.1 Transmitting MPEG2 (DVB) Data from Bulky Data Interface, Headers Auto-Inserted

Table B–5. Transmitting MPEG2 (DVB) Data from Bulky Data Interface, Headers Auto-Inserted

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register C (Link Control Register)	C407 0A00	This register sets the MPEG2Lynx to try to become cycle master.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.
Register DC (Transmit Timestamp Offset Register)	0000 3000h	This sets the transmit offset value to 3 isochronous cycles. This value is added to the cycle timer to make up the transmitted timestamp.
Register F0 (MPEG2 Formatter Control Register)	4003 004Dh	This register sets the MPEG2Lynx to transmit MPEG2 (DVB) packets only! –MPEG2 transmit mode enabled –BDI/F has access Bulky Data TX FIFO –Class 3 (1 pkt. Per iso cycle) –ISO and CIP headers inserted –Timestamps inserted
Register 150 (Bulky Size Register)	00C0 0000h	This sets the Bulky MPEG2 Transmit FIFO to 3072 bytes (decimal).
Register 1C8 (MPEG2 Transmit Header Register)	0008 4010h	This register sets the 1394 isochronous header that is inserted during automatic packetization. Data Length = 8 (power up value) Tag = 1 Channel Number = 0 Speed = 200Mbps
Register 1CC (MPEG2 CIP Transmit Header 0)	0006 C400h	Header is auto–loaded for MPEG2.
Register 1D0 (CIP Transmit Header 1)	6000 0000h	Header is auto–loaded for MPEG2.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

B.3.2 Transmitting DSS Data from Bulky Data Interface, Headers Auto-Inserted

Table B–6. Transmitting DSS Data from Bulky Data Interface, Headers Auto-Inserted

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register C (Link Control Register)	C407 0A00	This register sets the MPEG2Lynx to try to become cycle master.
Register 34 (Phy Access Register)	41C6 0000h	This register tells the phy to arbitrate for bus/cycle master.
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.
Register E0 (Transmit Timestamp Offset Register)	0000 3000h	This sets the transmit offset value to 3 isochronous cycles. This value is added to the cycle timer to make up the transmitted timestamp.
Register F4 (DSS Formatter Control Register)	4003 004Dh	This register sets the MPEG2Lynx to transmit DSS packets only! –DSS 140 transmit mode enabled – BDI/F has access bulky data TX FIFO –Class 3 (1 pkt. Per iso cycle) –ISO and CIP headers inserted –Timestamps inserted
Register 158 (Bulky Size Register)	00C0 0000h	This sets the bulky DSS transmit FIFO to 3072 bytes (decimal).
Register 1D4 (DSS Transmit Header Register)	0008 4010h	This register sets the 1394 isochronous header that is inserted during automatic packetization. Data Length = 8 Tag = 1 Channel Number = 0 Speed = 200Mbps
Register 1D8 (DSS Formatter CIP Transmit Header 0)	0009 8400h	Header is auto–loaded for DSS.
Register 1DC (DSS Formatter CIP Transmit Header 1)	6100 0000h	Header is auto–loaded for DSS.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

B.3.3 Transmitting Fully-Formatted MPEG2 (DVB) Data with 1394 Isochronous Header, CIP Headers, and Timestamps (Microprocessor Interface)

Table B–7. Transmitting Fully-Formatted MPEG2 (DVB) Data

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register 150 (Bulky MPEG2 Size Register)	00C0 0000h	This sets the bulky MPEG2 receive FIFO to 3072 bytes (decimal).
Register C (Link Control Register)	C407 0A00	This register sets the MPEG2Lynx to try to become cycle master.
Register 34 (Phy Access Register)	41C6 0000h	This tells the phy to arbitrate for bus/cycle master.
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.
Register F0 (MPEG2 Formatter Control Register)	4003 0008h	This register sets the MPEG2Lynx to transmit fully-formatted class 3 MPEG2 data from the Bulky MPEG transmit FIFO via the microprocessor interface.
Register 160 (MPEG2 Transmit FIFO First and Continue)		Using this write only register, the application software writes all quadlets of a MPEG2 (DVB) packet (expect the last) to the bulky transmit FIFO using this register.
Register 164 (MPEG2 Transmit FIFO Last and Send)		Using this write only register, the application software writes the last quadlet of a MPEG2 (DVB) packet to the bulky transmit FIFO. The entire packet is confirmed for transmission.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

B.3.4 Transmitting Fully-Formatted DSS 130 Data with 1394 Isochronous and CIP Headers Included. (Microprocessor Interface)

Table B–8. Transmitting Fully-Formatted DSS 130 Data

REGISTER NAME/NUMBER	SETTING	EXPLANATION
Register C (Link Control Register)	C407 0A00	This register sets the MPEG2Lynx to try to become cycle master.
Register 34 (Phy Access Register)	41C6 0000h	This tells the phy to arbitrate for bus/cycle master.
Register D8 (Bulky Data Interface Control)	000F 3800h	This sets the bulky data interface in mode A. This also sets up the bulky data interface in big endian format.
Register E0 (Transmit Timestamp Offset Register)	0000 3000h	This sets the transmit offset value to 3 isochronous cycles. This value is added to the cycle timer to make up the transmitted timestamp.
Register F4 (DSS Formatter Control Register)	4003 010Ch	This register sets the MPEG2Lynx to transmit fully-formatted class 3 DSS 130 data from the bulky DSS transmit FIFO via the microprocessor interface. MPEG2Lynx inserts timestamps.
Register 154 (Bulky DSS Size Register)	00C0 0000h	This sets the Bulky DSS Receive FIFO to 3072 bytes (decimal).
Register 16C (Transmit FIFO First and Continue)		Using this write only register, the application software writes all quadlets of a DSS packet (expect the last) to the bulky DSS transmit FIFO using this register.
Register 170 (Transmit FIFO Last and Send)		Using this write only register, the application software writes the last quadlet of a DSS packet to the bulky transmit FIFO. The entire packet is confirmed for transmission.
Register 1A4 (DSS Transmit Cell Header Register 0)		This register programs bytes 0–3 of the DSS header whenever in DSS 130 mode.
Register 1A8 (DSS Transmit Cell Header Register 1)		This register programs bytes 4–7 of the DSS header whenever in DSS 130 mode.
Register 1AC (DSS Transmit Cell Header Register 2)		This register programs bytes 8–9 of the DSS header whenever in DSS 130 mode.
Register 1EC (Microinterface Input/Output Control Register)	4200 0000h	This sets up the microcontroller port for Motorola 68000 mode

Appendix C
Isolation Considerations for TSB12LV41A

When using TI's Bus Holder Phy/Link Isolation solution with the TSB12LV41A, there are several issues that must be considered by the designer. TI's bus holder solution for isolation requires decoupling capacitors on all signal lines between the physical layer and link layer devices. (Please reference TI's *Serial Bus Galvanic Isolation Application Report*, Literature Number SLLA011 for more information on the topic of isolation.)

To isolate the TSB12LV41A and physical layer device, the $\overline{\text{ISO}}$ pin of the TSB12LV41A is pulled low and the phy–link interface is capacitively coupled. If the phy senses that the link is powered down separately (LPS–Link Power Status goes low), then it will stop supplying SCLK to the link. This will cause a lock up condition when the link is reactivated. When in this configuration, the user must insure that the phy and link are not powered down separately.

If the system can not avoid powering down the phy and link separately, then correct operation can be achieved by using an optoisolator to connect the link power supply (the link 3.3V power) to the phy LPS pin. The will also ensure that a lockup condition will not occur when the link is reactivated.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.