TSB12LV42 (DVLynx)

IEEE 1394-1995 Link-Layer Controller for Digital Video

SLLS293
December 1998

PRINTED WITH SOY INK

TEXAS INSTRUMENTS

Printed on Recycled Paper

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.

Copyright © 1998, Texas Instruments Incorporated

Contents

Section	Title	Page
1	**Introduction**	**1–1**
1.1	TSB12LV42	1–1
1.2	TSB12LV42 Features	1–2
1.3	DVLynx Pinout	1–3
1.4	Ordering Information	1–3
1.5	TSB12LV42 Terminal Functions	1–4
1.5	TSB12LV42 Terminal Functions (Continued)	1–5
2	**Architecture**	**2–1**
2.1	Bulky Data Interface	2–1
2.2	Bulky Data FIFO	2–1
2.2.1	Bulky DV Transmit FIFO (BDTX)	2–2
2.2.2	Bulky DV Receive FIFO (BDRX)	2–2
2.2.3	Bulky Asynchronous Transmit FIFO (BATX)	2–2
2.2.4	Bulky Asynchronous Receive FIFO (BARX)	2–2
2.2.5	Bulky Isochronous Transmit FIFO (BITX)	2–2
2.2.6	Bulky Isochronous Receive FIFO (BIRX)	2–2
2.3	DV Transmit and Receive Control	2–2
2.4	Microprocessor/Microcontroller Interface	2–3
2.5	Control FIFO	2–3
2.5.1	Asynchronous Control Transmit FIFO (ACTX)	2–3
2.5.2	Asynchronous Control Receive FIFO (ACRX)	2–3
2.5.3	Broadcast Write Receive FIFO (BWRX)	2–3
2.6	Physical Layer	2–3
2.7	Configuration Register (CFR)	2–3
3	**Functional Description and Data Formats**	**3–1**
3.1	Overview	3–1
3.2	DV on 1394 Overview	3–2
3.2.1	DV interface	3–2
3.2.2	DV Bandwidth on IEEE1394	3–3
3.2.3	DV Transmission over IEEE1394	3–3
3.2.4	Source Packet/DIF Block Format	3–4
3.2.5	DV Packets CIP Header Calculations	3–5
3.3	Transmit Operation	3–6
3.3.1	Transmitting Asynchronous Control Packets	3–6
3.3.2	Transmitting Asynchronous Data Packets	3–7
3.3.4	Byte Padding	3–12
3.3.5	Transmitting DV Formatted Isochronous Packets	3–12
3.4	Receive Operation	3–17
3.4.1	Receiving Asynchronous Packets	3–17

3.5 Time Stamps 3–27
3.5.1 Time Stamp Encoding/Decoding for DV Transmit and Receive 3–27
3.5.2 Time Stamp Calculation on Transmit 3–28
3.5.3 Time Stamp Determination on Receive 3–29
3.6 Asynchronous Transmit Data Formats (Host Bus to TSB12LV42) 3–29
3.6.1 Quadlet Transmit 3–29
3.6.2 Block Transmit 3–30
3.6.3 Quadlet Receive 3–31
3.6.4 Block Receive 3–33
3.7 Isochronous Transmit and Receive (Host Bus to TSB12LV42) Data Formats . . 3–35
3.7.1 Isochronous Transmit 3–35
3.7.2 Isochronous Receive Data Formats 3–35
3.8 Snoop 3–36
3.9 CycleMark 3–37
3.10 Phy Configuration 3–37
3.11 Receive Self-ID Packet 3–38

4 External Interfaces 4–1
4.1 Bulky Data Interface 4–1
4.1.1 BDIF Control Register (D8h) Configuration 4–4
4.1.2 Modes of the Bulky Data Interface (BDIF) 4–6
4.1.3 Mode A – 8 Bit Parallel I/O 4–7
4.1.4 Mode B – 8-Bit Parallel I/O with No Read Control 4–8
4.1.5 Mode C – 8 Bit Parallel Asynchronous Input/8 Bit Parallel Asynchronous Output 4–9
4.1.6 Mode D– 8 Bit Parallel Bidirectional Mode 4–10
4.1.7 Bulky Data Interface Timing 4–10
4.1.8 Bidirectional Modes 4–13
4.2 Microprocessor Interface 4–16
4.2.1 Microprocessors Supported 4–16
4.2.2 Microprocessor Interface Control 4–19
4.2.3 Handshake and Blind Access modes 4–20
4.2.4 General Read Instructions 4–20
4.2.5 General Write Instructions 4–21
4.2.6 TMS320AV7100 Mode Timing Diagrams 4–22
4.2.7 68000 Mode Timing Diagrams 4–27
4.2.8 8051 Mode Timing Diagrams 4–31
4.2.9 Blind Access Mode Specific Issues 4–35
4.2.10 Endianness 4–36
4.2.11 Use of Interrupts with DVLynx 4–38
4.3 TSB12LV42 to 1394 Phy Interface Specification 4–39
4.3.1 Introduction 4–39
4.3.2 Assumptions 4–40
4.3.3 Block Diagram 4–40
4.3.4 Operational Overview 4–40
4.3.5 Request 4–41
4.3.6 Status 4–43
4.3.7 TSB12LV42 to Phy Bus Timing 4–45

5 Detailed Operation and Programmers Reference 5–1
5.1 TSB12LV42 Configuration Register 5–1
5.2 Version Register (VERS at Addr 000h) 5–6
5.3 C Acknowledge Register (CACK at Addr 004h) 5–6
5.4 B Acknowledge Register (BACK at Addr 008h) 5–6
5.5 Link Control Register (LCTRL at Addr 00Ch) 5–7
5.6 Interrupt Register (IR at Addr 010h) 5–9
5.7 Interrupt Register Enable Register (IMR at Addr 014h) 5–11
5.8 Extended Interrupt Register (EIR at Addr 018h) 5–13
5.9 Extended Interrupt Mask Register (EIMR at Addr 01Ch) 5–15
5.10 Isochronous Receive Comparators Register 0 (IRPR0 at Addr 020h) 5–16
5.11 Isochronous Receive Comparators Register 1 (IRPR1 at Addr 024h) 5–17
5.12 Cycle Timer Register (CLKTIM at Addr 028h) 5–18
5.13 Extended Cycle Time Register (EXTTIM at Addr 02Ch) 5–18
5.14 Link Diagnostics Register (DIAG at Addr 030h) 5–19
5.15 Phy Access Register (PHYAR at Addr 034h) 5–19
5.16 Expected Response (PHYSR at Addr 038h) 5–20
5.17 Reserved Register (at Addr 03Ch – 040h) 5–20
5.18 Asynchronous Control Data Transmit FIFO Status (ACTFS at Addr 044h) 5–20
5.19 Bus Reset Data Register (BRD at Addr 048h) 5–21
5.20 Bus Reset Error Register (BRERR at Addr 04Ch) 5–22
5.21 Asynchronous Control Data Receive FIFO Status (ACRXS at Addr 050h) 5–22
5.22 Read Write Test Register (UCRWTEST at Addr 054h) 5–23
5.23 Reserved Register (at Addr 058h – 07Ch) 5–23
5.24 Asynchronous Control Data Transmit FIFO First (ACTXF at Addr 080h) 5–23
5.25 Asynchronous Control Data Transmit FIFO Continue (ACTXC at Addr084h) . . 5–23
5.26 Asynchronous Control Data Transmit FIFO First and Update (ACTXFU at Addr 088h) 5–23
5.27 Asynchronous Control Data Transmit FIFO Continue and Update (ACTXCU at Addr 08Ch) 5–24
5.28 Reserved Register (at Addr 090h – 0BCh) 5–24
5.29 Asynchronous Control Data Receive FIFO (ACRX at Addr 0C0h) 5–24
5.30 Broadcast Write Receive FIFO (BWRX at Addr 0C4h) 5–24
5.31 Reserved Register (at 0C8 – 0D4) 5–24
5.32 Bulky Data Interface Control (BIF at Addr 0D8h) 5–24
5.33 Transmit Timestamp Offset Register (XTO at Addr 0DCh) 5–25
5.34 Reserved Register (at Addr 0E0h) 5–25
5.35 Receive Timestamp Offset (RTO at Addr 0E4h) 5–25
5.36 Reserved Register (at Addr 0E8h) 5–25
5.37 Asynchronous/Isochronous Application Data Control Register (AICR at Addr 0ECh) 5–25
5.38 DV Formatter Control Register (DCR at Addr 0F0h) 5–27
5.39 Reserved Register (at Addr 0F4h) 5–29
5.40 FIFO Misc (FMISC at Addr 0F8h) 5–29
5.41 Reserved Register (at Addr 0FCh – 100h) 5–29
5.42 Bulky A Size Register (BASZ at Addr 104h) 5–30
5.43 Bulky A Avail Register (BAAVAL at Addr 108h) 5–30

5.44 Asynchronous Application Data Transmit FIFO First and Continue (BATXFC at Addr 10Ch) 5–30
5.45 Asynchronous Application Data Transmit FIFO Last and Send (BATXLS at Addr 110h) 5–30
5.46 Asynchronous Application Data Receive FIFO (BARX at Addr 114h) 5–30
5.47 Asynchronous Application Data Receive Header Register 0 (ARH0 at Addr 118h) 5–31
5.48 Asynchronous Application Data Receive Header Register 1 (ARH1 at Addr 11Ch) 5–31
5.49 Asynchronous Application Data Receive Header Register 2 (ARH2 at Addr 120h) 5–31
5.50 Asynchronous Application Data Receive Header Register 3 (ARH3 at Addr 124h) 5–31
5.51 Asynchronous Application Data Receive Trailer (ART at Addr 128h) 5–31
5.52 Bulky Isochronous Size Register (BISZ at Addr 12Ch) 5–32
5.53 Bulky Isochronous Avail Register (BIAVAL at Addr 130h) 5–32
5.54 Isochronous Transmit First and Continue (BITXFC at Addr 134h) 5–32
5.55 Isochronous Transmit Last and Send (BITXLS at Addr 138h) 5–32
5.56 Isochronous Receive FIFO (BIRX at Addr 13Ch) 5–33
5.57 Isochronous Packed Received Header (IRH at Addr 140h) 5–33
5.58 Isochronous Packet Received Trailer (IRT at Addr 144h) 5–33
5.59 Receive Packet Router (RMISC at Addr 148h) 5–34
5.60 Bulky Asynchronous Retry (BARTRY at Addr 14Ch) 5–35
5.61 Bulky DV Size Register (BDSZ at Addr 150h) 5–35
5.62 Bulky DV Avail Register (BDAVAL at Addr 154h) 5–36
5.63 Reserved Register (at Addr 158h) 5–36
5.64 Reserved Register (at Addr 15Ch) 5–36
5.65 DV Transmit FIFO First and Continue (BDTXFC at Addr 160h) 5–36
5.66 DV Transmit FIFO Last & Send (BDTXLS at Addr 164h) 5–36
5.67 DV Formatted Packet Receive FIFO (BDRX at Addr 168h) 5–36
5.68 Reserved Register (at Addr 16Ch) 5–36
5.69 Reserved Register (at Addr 170h) 5–36
5.70 Reserved Register (at Addr 174h) 5–36
5.71 DV Receive Header (DRH at Addr 178h) 5–36
5.72 DV CIP Receive Header 0 (DCIPR0 at Addr 17Ch) 5–37
5.73 DV CIP Receive Header 1 (DCIPR1 at Addr 180h) 5–37
5.74 DV Receive Trailer Register (DRT at Addr 184h) 5–37
5.75 Reserved Register (at Addr 188h – 194h) 5–38
5.76 DV Receive Cell Header Register 0 (DRX0 at Addr 198h) 5–38
5.77 DV Receive Cell Header Register 1 (DRX1 at Addr 19Ch) 5–38
5.78 Reserved Register (at Addr 1A0h) 5–38
5.79 DV Transmit Cell Header Register 0 (DTX0 at Addr 1A4h) 5–38
5.80 DV Transmit Cell Header Register 1 (DTX1 at Addr 1A8h) 5–38
5.81 Reserved Register (at Addr 1ACh) 5–39
5.82 Asynchronous Header 0 for Auto Transmit (AHEAD 0) at Addr 1B0h) 5–39
5.83 Asynchronous Header 1 for Auto Transmit (AHEAD1) at Addr 1B4h) 5–39
5.84 Asynchronous Header 2 for Auto Transmit (AHEAD2) at Addr 1B8h) 5–39

5.85 Asynchronous Header 3 for Auto Transmit (AHEAD3) at Addr 1BCh 5–39
5.86 Isochronous Header for Auto Transmit (IHEAD0 at Addr 1C0h) 5–39
5.87 Packetizer Control (PKTCTL at Addr 1C4h) 5–40
5.88 DV Transmit Header Register (DXH at Addr 1C8h) 5–41
5.89 DV CIP Transmit Header 0 (DCIPX0 at Addr 1CCh) 5–41
5.90 DV CIP Transmit Header 1 (DCIPX1 at Addr 1D0h) 5–42
5.91 Reserved Register (at Addr 1D4h) 5–42
5.92 Reserved Register(at Addr 1D8h) 5–42
5.93 Reserved Register (at Addr 1DCh) 5–42
5.94 MDAltCont (MDALT at Addr 1E0h) 5–42
5.95 Reserved Register (at Addr 1E4h) 5–42
5.96 Reserved Register (at Addr 1E8h) 5–42
5.97 Microinterface Input/Output Control Register (IOCR at Addr 1ECh) 5–42
5.98 Blind Access Status Register (BASTAT at Addr 1F0h) 5–43
5.99 Blind Access Holding Register (BAHR at Addr 1F4h) 5–44
5.100 Reserved Register (at Addr 1F8h) 5–44
5.101 Software Reset Register (SRES at Addr 1FCh) 5–44

6 Electrical Characteristics 6–1
6.1 Absolute Maximum Ratings Over Free-Air Temperature Range 6–1
6.2 Recommended Operating Conditions 6–2
6.3 Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature 6–2
6.4 DVLynx Power 6–3

7 Mechanical Information 7–1

A Receive Operation Examples A–1
A.1 Asynchronous Receive A–1
A.1.1 Receiving Asynchronous Data to the Bulky Asynchronous FIFO (Bulky Data Interface) A–1
A.1.2 Receiving Asynchronous Data to the Asynchronous Control FIFO (Microprocessor Port) A–1
A.2 Unformatted Isochronous Receive A–2
A.2.1 Receiving Isochronous Data to the Bulky Isochronous FIFO (Bulky Data Interface) A–2
A.2.2 Receiving Isochronous Data to the Bulky Isochronous FIFO (Microprocessor Interface) A–3
A.3 DV Receive A–3
A.3.1 Receiving DV Data to the Bulky DV FIFO (Bulky Data Interface) A–4
A.3.2 Receiving DV Data to the Bulky DV FIFO (Microprocessor Interface) A–4

B Transmit Operation Examples B–1
B.1 Asynchronous Transmit B–1
B.1.1 Transmitting Asynchronous Data Packets (Bulky Data Interface) B–1
B.1.2 Transmitting Asynchronous Control Packets B–2
B.2 Unformatted Isochronous Transmit B–2
B.2.1 Transmitting Isochronous Data, Headers Auto Inserted (Bulky Data Interface) B–3
B.2.2 Transmitting Fully Formatted Isochronous Data (Microprocessor Interface) B–3
B.3 DV Transmit B–4
B.3.1 Transmitting DV Data from Bulky Data Interface, Headers Auto-Inserted B–4
B.3.2 Transmitting Fully Formatted Data Fully Formatted with 1394 Isochronous, CIP, and H0 Headers (Microprocessor Interface) B–5

C Isolation Considerations for TSB12LV42 . . . . . C–1

List of Illustrations

Figure	*Title*	*Page*
2–1	Functional Block Diagram	2–1
3–1	Example of a Source Packet Transmit Event	3–2
3–2	Source Packet Transmit Event Timing	3–2
3–3	DV Packet on 1394 Bus	3–3
3–4	DIF Block H0	3–4
3–5	ID Data In DIF Block	3–5
3–6	ID Data in DIF Block	3–5
3–7	DV Source Packet Format	3–5
3–8	CIP Header Format	3–6
3–9	Transmit from the Asynchronous Control Transmit FIFO (ACTX)	3–7
3–10	Transmit Asynchronous/Isochronous Data from BATX by the Bulky Data Interface with Auto-Packetization	3–8
3–11	Transmit Asynchronous/Isochronous Data from BATX by the MP/MC Interface with Auto-Packetization	3–9
3–12	Transmit Asynchronous/Isochronous Data from BATX by the Bulky Data Interface, No Auto-Packetization	3–9
3–13	Transmit Asynchronous/Isochronous Data from BATX by the MP/MC Interface, No Auto-Packetization	3–10
3–14	Data from Bulky Data Interface, Headers/Timestamp/H0 Automatically Inserted	3–14
3–15	Data and H0 Header from Bulky Data Interface, Headers/Timestamp Automatically Inserted	3–14
3–16	Data 1394 Isochronous and CIP Headers	3–15
3–17	All Data from BDIF, including 1394 Isochronous Header	3–15
3–18	DBC Example	3–17
3–19	Receive Asynchronous/Isochronous Data to Bulky Data Interface	3–20
3–20	Receive Asynchronous/Isochronous Data to Microprocessor Interface	3–20
3–21	Header, Data, and Trailer Received at BDIF	3–24
3–22	Header, Data, and Trailer Received at MP/MC	3–24
3–23	Header, Trailer Stripped, Data only sent to BDIF	3–25
3–24	Header, Trailer Stripped, Data only set to MP/MC	3–25
3–25	DV Sub Mode Header, Trailer Saved to Registers, Data Discarded	3–26
3–26	Determination of High Add and Low Add	3–28
3–27	Time Stamp Value for LowAdd <3072	3–28
3–28	Time Stamp Value for LowAdd . 3072	3–29
3–29	Time Stamp Determination on Receive	3–29
3–30	Quadlet-Transmit Format	3–29
3–31	Block-Transmit Format	3–30
3–32	Quadlet-Receive Format for Control FIFO	3–31
3–33	Quadlet-Receive Format for Bulky Data FIFO	3–32
3–34	Block-Receive Format for Control FIFO	3–33

3–35 Block-Receive Format for Bulky Data FIFO 3–33
3–36 Isochronous-Transmit Format 3–35
3–37 Isochronous-Receive Format for Bulky Data FIFO 3–35
3–38 Snoop Format 3–36
3–39 CycleMark Format 3–37
3–40 Phy Configuration Format 3–37
3–41 Receive Self-ID Format for Broadcast Write Receive FIFO 3–38
3–42 Receive Self-ID Format for Bulky Asynchronous Receive FIFO 3–38
3–43 Phy Self-ID Packet #0 Format 3–39
3–44 Phy Self-ID Packet #1, Packet #2, and Packet #3 Format 3–40

4–1 Bulky Data Interface 4–2
4–2 BDIF Control Register 4–4
4–3 Bulky Data Interface Mode A Typical Application 4–7
4–4 Bulky Data Interface Mode B Typical Application 4–8
4–5 Bulky Data Interface Mode C Typical Application 4–9
4–6 Bulky Data Interface Mode D Typical Application 4–10
4–7 Functional Timing for Write Operations in the Unidirectional Modes 4–11
4–8 Critical Timing for Write Operations in Unidirectional Mode 4–11
4–9 Functional Timing for Write Operations in the Asynchronous Mode 4–12
4–10 Functional Timing for Read Operations in Unidirectional Mode 4–12
4–11 Critical Timing for Read Operations in Unidirectional Mode 4–13
4–12 Functional Timing for Read Operations in Asynchronous Mode 4–13
4–13 Functional Timing for Write Operations in Bidirectional Mode 4–14
4–14 Critical Timing for Write Operations in Bidirectional Mode 4–14
4–15 Functional Timing for Read Operations in Bidirectional Mode 4–15
4–16 Critical Timing for Read Operations in Bidirectional Mode 4–15
4–17 DVLynx Connections for 68000 Microcontroller 4–17
4–18 DVLynx Connections for 8051 Microcontroller 4–18
4–19 DVLynx Connections for TMS320AV7100 ARM Processor 4–18
4–20 TSB12LV42 IOCR Register 4–19
4–21 TMS320AV7100 ARM Read/Write Critial Timing 4–19
4–22 TMS320AV7100 Handshake Mode Read Timing 4–24
4–23 TMS320AV7100 Handshake Mode Write Timing 4–24
4–24 TMS320AV7100 ARM Blind Access Mode Read Timing 4–25
4–25 TMS320AV7100 ARM Blind Access Mode Write Timing 4–25
4–26 Motorola 68000 Read Timing 4–28
4–27 Motorola 68000 Write Timing 4–28
4–28 Motorola 68000 Read Critical Timing 4–29
4–29 Motorola 68000 Write Critical Timing 4–30
4–30 Intel 8051 Read Timing (Blind Access Read) 4–32
4–31 Intel 8051 Write Timing (Blind Access Write) 4–32
4–32 Intel 8051 Read Critial Timing 4–33
4–33 Intel 8051 Write Critial Timing 4–34
4–34 Big Endian Illustration chart 4–36
4–35 Little Endian Illustration chart 4–36
4–36 Little Endian Data Invariant System Design Illustration Chart 4–37
4–37 Little Endian Address Invariant System Design Illustration Chart 4–38

4–38 Interrupt Hierarchy 4–39
4–39 Functional Block Diagram of the TSB12LV42 to Phy Layer 4–40
4–40 LREQ Timing 4–45
4–41 Status-Transfer Timing 4–45
4–42 Transmit Timing 4–46
4–43 Receiver Timing 4–46

5–1 Internal Register Map 5–1

List of Tables

Table	*Title*	*Page*
3–1	DV Bandwidth on IEEE 1394	3–3
3–2	DHIM/H0INST Settings for Different Header Insertion Schemes	3–13
3–3	Asynchronous Receive Modes	3–19
3–4	Receiving Isochronous data to the BIRX FIFO	3–22
3–5	DV Receive Modes	3–23
3–6	Time Stamp Field of Source Packet	3–27
3–7	Quadlet-Transmit Format Functions	3–30
3–8	Block-Transmit Format Functions	3–31
3–9	Quadlet-Receive Format Functions	3–32
3–10	Block-Receive Format Functions	3–34
3–11	Isochronous-Transmit Functions	3–35
3–12	Isochronous-Receive Functions	3–36
3–13	Snoop Functions	3–36
3–14	CycleMark Function	3–37
3–15	Phy Configuration Functions	3–37
3–16	Receive Self-ID Function	3–38
3–17	Broadcast Write Receive FIFO Contents With Three Nodes on a Bus	3–39
3–18	Bulky Data Asynchronous Receive FIFO (BARX FIFO) Contents	3–39
3–19	Phy Self-ID Functions	3–40
4–1	Modes of the BDIF	4–6
4–2	MCSEL Settings for Various Microprocessors	4–16
4–3	TSB12LV42 MP/MC Interface Pin Function Matrix	4–16
4–4	TSB12LV42 IOCR Bit/Function Correlation Table and Power–up Default Setting	4–19
4–5	TMS320AV7100 Critical Timing Characteristics	4–22
4–6	Motorola 68000 Critical Timing Characteristics	4–27
4–7	Intel 8051 Critical Timing Characteristics	4–31
4–8	Phy Interface Control of Bus Functions	4–40
4–9	TSB12LV42 Control of Bus Functions	4–41
4–10	Request Functions	4–41
4–11	Bus-Request Functions (Length of Stream: 7 Bits)	4–41
4–12	Read-Register Request Functions (Length of Stream: 9 Bits)	4–41
4–13	Write-Register Request (Length of Stream: 17 Bits)	4–42
4–14	TSB12LV42 Request Functions	4–42
4–15	Request-Speed Functions	4–42
4–16	Status-Request Functions (Length of Stream: 16 Bits)	4–44
4–17	Speed Code for Receive	4–45

A–1 Receiving Asynchronous Data to the Bulky Asynchronous FIFO A–1
A–2 Receiving Asynchronous Data to the Asynchronous Control FIFO A–2
A–3 Receiving Isochronous Data to the Bulky Isochronous FIFO A–2
A–4 Receiving Isochronous Data to the Bulky Isochronous FIFO A–3
A–5 Receiving DV Data to the Bulky DV FIFO A–4
A–6 Receiving DV Data to the Bulky DV FIFO A–4

B–1 Transmitting Asynchronous Data Packets B–1
B–2 Transmitting Asynchronous Control Packets B–2
B–3 Transmitting Isochronous Data, Headers Auto Inserted B–3
B–4 Transmitting Fully Formatted Isochronous Data B–3
B–5 Transmitting DV Data from Bulky Data Interface, Headers Auto-Inserted B–4
B–6 Transmitting Fully Formatted Data Fully Formatted B–5

1 Introduction

1.1 TSB12LV42

- Supports Provisions of IEEE 1394-1995 Standard for High-Performance Serial Bus
- Fully Interoperable with FireWire™ Implementation of IEEE-1394 (1995)
- Interfaces Directly to Texas Instruments TSB11LV01 and TSB21LV03A Physical Layer Devices (100/200 Mbits/s)
- Single 3.3-V Supply Operation with 5-V Tolerance using 5-V Bias Terminals.
- High-Performance 100-Pin PZ (S-PQFP-G100) Package
- Multi-Microcontroller/Microprocessor Interface Supports TMS320AV7xxx, 680xx, 650x, 80x86, Z8x Processors
- 64 Quadlet (256 byte) Control FIFO Accessed through Microcontroller Interface Supports Command/Status Operations
- 8K-Byte FIFO Supports Standard-Definition Digital-Video Cassette Recorder (SD-DVCR), Asynchronous, and Isochronous Modes
- Bus Reset Functions and Automatic IEEE-1394 Self-ID Verification
- Supports IEC61883 standard formats for transmitting SD-DVCR data over 1394.

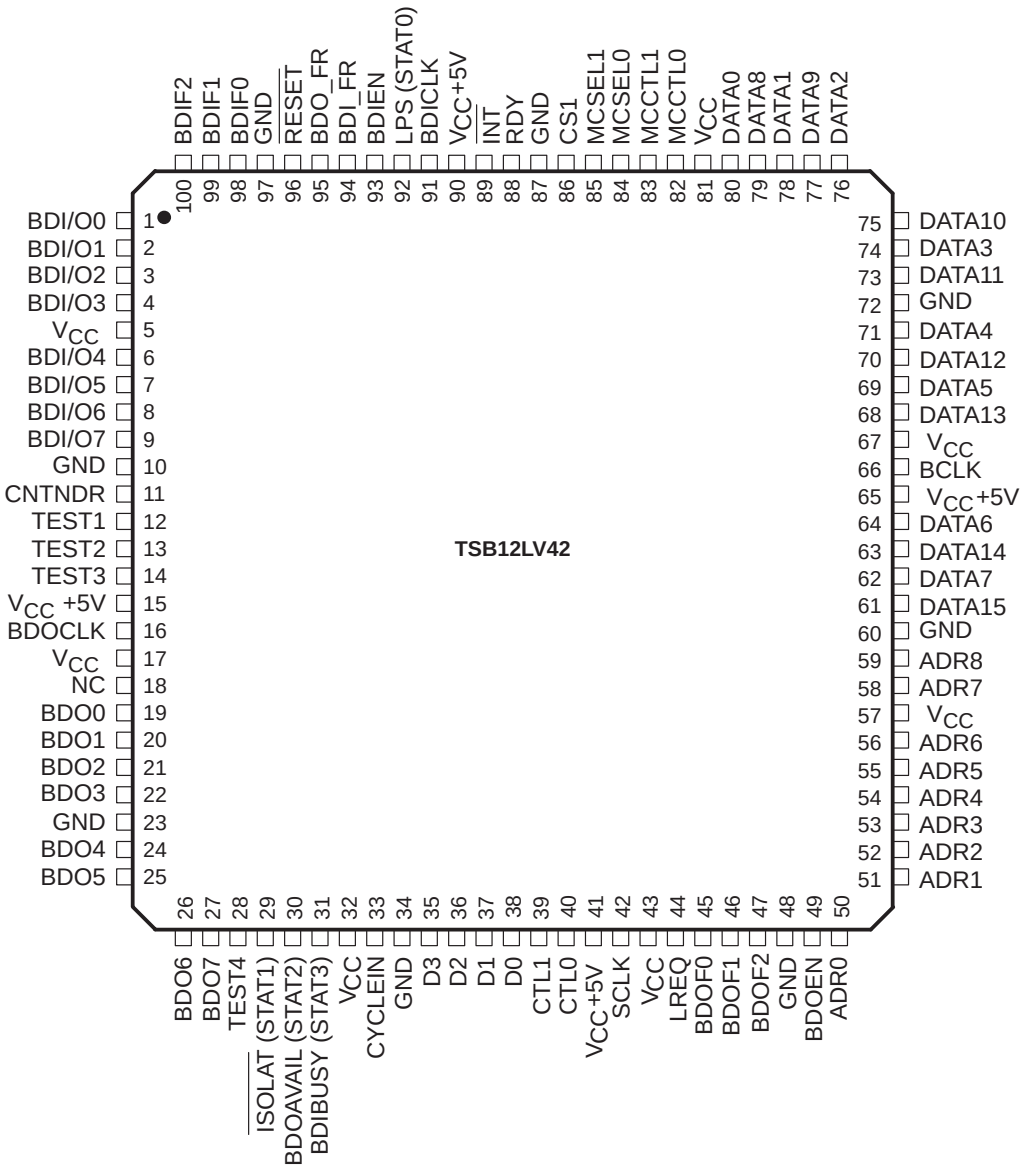
FireWire is a trademark of Apple Computer, Incorporated.

1.2 TSB12LV42 Features

- Complies with IEEE 1394-1995 Standard
- Transmits and Receives Correctly Formatted 1394 Packets
- Supports SD-DVCR (DV) Formatted Isochronous Data Transfer
- Supports Isochronous Data Transfer
- Cycle Master (CM), Isochronous Resource Manager (IRM) and Bus Manager(BM) Capable
- Generates and Checks 32-Bit CRC
- Detects Lost Cycle-Start Packets
- 8K-Byte Bulky Data Interface (BDIF) for DV, Isochronous, and Asynchronous Data Transfer
- Multimode BDIF programmable for bytewide and memory mapped modes (independent for RX and TX)
- Implements a 256-Byte Control FIFO (Control FIFO) and an 8K-Byte Bulky Data FIFO
- 8K-Byte BDIF FIFO Implements Six Independent Logical FIFOs for DV, Isochronous, and Asynchronous Data Receive and Transmit through the BDIF
- Performs Bulky Asynchronous FIFO Packet Retry for Transmit (up to 256 Retries with Intervals Up to 256 $\times$ 125 μs)
- 256-Byte Control FIFO for Control Packets
- Interfaces Directly to 100-Mbits/s and 200-Mb/s Physical Layer Devices Conforming to Annex J of 1394-1995
- Chip Control with Directly Addressable Configuration Registers (CFRs)
- Interrupt Driven to Minimize Host Polling
- Multimode 8-/16-Bit Microcontroller/Microprocessor Interface
- Supports 16-Bit Width Timestamp Offsets for DV Receive and Transmit.
- Optimized Pinout for Easy Board Layout
- Includes Texas Instruments Bus Holder Circuitry for Phy-Link Isolation
- Automatic CIP Header Insertion
- Automatic H0 DIF Block Insertion
- Automatic Empty Packet Insertion
- Supports both NTSC and PAL Formats
- Generates Output Frame Pulse

1.3 DVLynx Pinout

PZ PACKAGE
(TOP VIEW)



1.4 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	DATA RATE	PACKAGE
TSB12LV42PZ	DVLynx	3.3 V – 5 V Tolerant I/O's	Up to 200 Mbits/s	100 pin PQFP

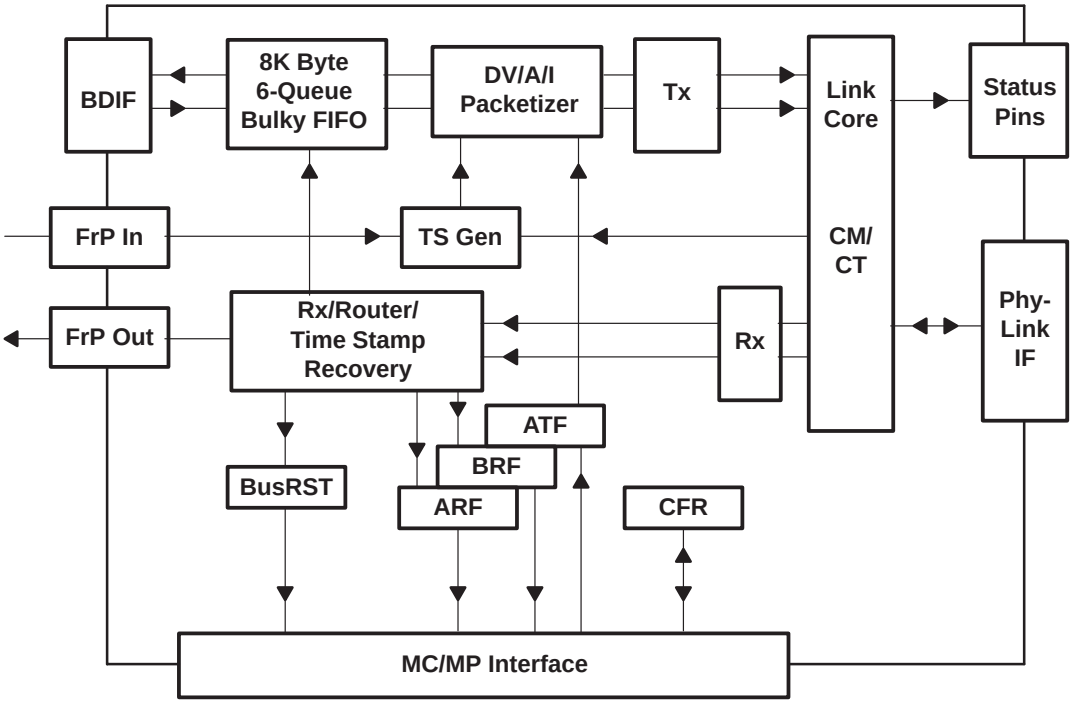
1.5 TSB12LV42 Terminal Functions

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
ADR0 – ADR8	50, 51, 52, 53 54, 55, 56, 58 59	I	MP/MC address lines. ADR0 is the MSB.
BCLK	66	I	Host bus clock
BDI_FR	94	I	Frame pulse input. The frame pulse input signal for PAL is 25 Hz and for NTSC is 29.97 Hz at 50% duty cycle.
BDIBUSY(STAT3)	31	O	BDIF is busy or status output 3. STAT3 is programmable at DIAG register (reg 30h).
BDICLK	91	I	Bulky data I/O clock. This terminal operates at up to 40 MHz.
BDIEN	93	I	BDIO bus enable. If BDIEN is low, accesses to BDIO-bus are ignored.
BDIF2 – BDIF0	100, 99, 98	I/O	BDIF format bus for BDIO port. BDIF2 is MSB.
BDIO7 – BDIO0	9, 8, 7, 6, 4, 3 2, 1	I/O	BDIF I/O data lines. BDIO7 – BDIO0 are high-speed I/O data lines for the BDIF bus. They are primarily used for audio/data/video applications. These lines can also be configured for input only. BDIO7 is the MSB.
BDO_FR	95	O	Frame pulse output.
BDO7 – BDO0	27, 26, 25, 24 22, 21, 20, 19	O	BDIF output data lines. BDO7 – BDO0 are data lines for high-speed output on the BDIF bus for audio/data/video applications. These lines are compliant with several standard interfaces. BDO7 is the MSB.
BDOAVAIL(STAT2)	30	O	Bulky data output available or status output 2. STAT2 is programmable at DIAG register (reg 30h).
BDOCLK	16	I	Bulky data output clock. The bulky data output clock operates at up to 40 MHz.
BDOEN	49	I	BDO bus enable. If BDOEN is low, accesses to BDO-bus are ignored.
BDOF2 – BDOF0	47, 46, 45	O	BDIF format bus for BDO Port. BDOF2 is the MSB.
CNTNDR	11	I/O	Contender. The CNTNDR tells the Link when the local node is a contender for IRM. This terminal can also be driven by the Link. The default status of this terminal is input.
CS1	86	I	Chip select. CS1 needs to be 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 indicate the four operations that can occur in this interface.
CYCLEIN	33	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. It is enabled by the cycle source bit and should be tied high when not used.
D0 – D3	38, 37, 36, 35	I/O	Data 0 – 3 of the Phy-Link data bus. Data is expected on D0 – D1 for 100 Mbits/s and D0 – D3 for 200 Mbits/s. Data0 is the MSB.
DATA0 – DATA15	80, 78, 76, 74 71, 69, 64, 62 79, 77, 75, 73 70, 68, 63, 61	I/O	Data 0 – 15 of MC/MP host processor. Some of the DATAx terminals have second functions depending on the status of the MCSEL terminals. DATA0 is the MSB.

1.5 TSB12LV42 Terminal Functions (Continued)

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
GND	10, 23, 34, 48 60, 72, 87, 97		Ground
$\overline{\text{INT}}$	89	O	Interrupt. $\overline{\text{INT}}$ is used to notify the host that an interrupt has occurred. This terminal can be active low or active high depending on the status of the INTPOL bit in IOCR register. It is high-impedance when no interrupt is pending.
$\overline{\text{ISOLAT}}$(STAT1)	29	I/O	Isolation or status output 1. $\overline{\text{ISOLAT}}$ is sampled during hardware reset to determine if isolation is present. When this terminal is high, this indicates **NO** isolation is being used. STAT1 is driven as an output after hardware reset and used as a status output. STAT1 is programmable in the DIAG register (reg 30h)
LPS/STAT0	92	O	Link power status or status output 0. LPS is used to drive the LPS input to the Phy. This signal indicates that the LLC is powered up and active. This output toggles at 1/32 of BCLK or SYSCLK depending on the microprocessor used. STAT0 is used to MUX out internal signals. STAT0 is programmable in the DIAG register (reg 30h).
LREQ	44	O	Link request. LREQ is a DVLynx output that makes bus request and accesses the Phy layer.
MCCTL0/1	82, 83	I	Control lines for bus access. MCCTL0 and MCCTL1 function depends on MP/MC type being used.
MCSEL0/1	84, 85	I	Select lines for MP/MC. MCSEL0 and MCSEL1 are selection lines for the MP/MC-type being used. These terminals have an impact on the function MCTRL,ADR, RDY and DATA terminals.
NC	18	O	No connect
RDY	88	O	Ready line. When high, RDY indicates the end of an MP/MC access.
$\overline{\text{RESET}}$	96	I	Reset (active low). $\overline{\text{RESET}}$ is the asynchronous power on reset to the device.
SCLK	42	I	System clock. SCLK is a 49.152-MHz clock from the Phy. This terminal generates the 24.576-MHz clock (NCLK).
TEST1	12	I	Test pin should be tied to V_{CC}.
TEST2	13	I	Test should be tied to GND.
TEST3	14	I	Test pin should be tied to V_{CC}.
TEST4	28	I	Test pin should be tied to GND.
V_{CC}	5, 17, 32, 43 57, 67, 81		3.3 V (3 V – 3.6 V) supply voltage
V_{CC}5V	15, 41, 65, 90		5-V ±5% power supplies for 5V tolerant I/O terminals. These terminals can be tied to 3.3 V if no 5 V devices interface to the TSB12LV42.

2 Architecture



Conventions:

Asynchronous packet	**= Async packet**
Isochronous packet	**= Isoc packet**
Asynchronous Transmit FIFO	**= ATF**
Broadcast Receive FIFO	**= BRF**
Asynchronous Receive FIFO	**= ARF**
Configuration registers	**= CFR**
SD-DVCR 480 byte source packet	**= DV-packet**
MP/MC	**= Microprocessor/Microcontroller**

Figure 2–1. Functional Block Diagram

2.1 Bulky Data Interface

The bulky data interface (BDIF) enables the TSB12LV42 to provide sustained data rates up to 160 Mbits/s. The bulky data FIFO supports DV compressed data as defined by the Blue Book standard for digital video, asynchronous, and isochronous data for both transmit and receive.

2.2 Bulky Data FIFO

The bulky data FIFO is where the BDIF buffers the transmit or receive data. The bulky data FIFO is partitioned into six logical divisions. Each of these logical FIFO sizes are programmable on four quadlet boundaries. These six FIFOs are:

- Bulky DV transmit FIFO (BDTX)
- Bulky DV receive FIFO (BDRX)
- Bulky asynchronous transmit FIFO (BATX)

- Bulky asynchronous receive FIFO (BARX)
- Bulky isochronous transmit FIFO (BITX)
- Bulky isochronous receive FIFO (BIRX)

The following sections give functional descriptions to these logical FIFOs. See Section 4.1, *Bulky Data Interface*, for more detail on using this FIFO to transmit/receive data.

2.2.1 Bulky DV Transmit FIFO (BDTX)

The BDTX FIFO is used to transmit DV data according to the Blue Book standard. Data is typically written to this FIFO for the BDIF or microcontroller interface in quadlets (four bytes). The TSB12LV42 can be configured to automatically insert 1394 isochronous headers, CIP (or common isochronous packet) headers, and H0 DIF blocks. The TSB12LV42 can also be configured to automatically insert empty packets to smooth out the bursty data rates. This is necessary to accommodate receiving nodes whose FIFO's are sized to receive evenly distributed data.

2.2.2 Bulky DV Receive FIFO (BDRX)

The BDRX FIFO is typically used to store DV data received from the link layer core and then forward it on to a high speed application via the BDIF. Only isochronous port 0 of the link layer core can access the BDRX FIFO.

2.2.3 Bulky Asynchronous Transmit FIFO (BATX)

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

2.2.4 Bulky 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 BDIF. The microcontroller can also read data from the BARX FIFO one quadlet at a time. This FIFO is the default location for self-IDs.

2.2.5 Bulky Isochronous Transmit FIFO (BITX)

The BITX FIFO is typically used to transmit isochronous data packets from high-speed applications. Data can be loaded into the FIFO by either the BDIF or by the microcontroller one quadlet at a time.

2.2.6 Bulky Isochronous Receive FIFO (BIRX)

The BIRX FIFO is typically used for receiving isochronous data and forwarding it to a high-speed application via the BDIF. Data can also be forwarded to the microcontroller interface one quadlet at a time. 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 isochronous header TAG value.

2.3 DV Transmit and Receive Control

The DVLynx transmit and receive circuitry controls automatic insertion of the common isochronous packet (CIP) header information as defined by the IEC 61883 standard. This circuitry also controls the automatic insertion of the H0 DIF block header as defined by the Blue Book standard for SD-DVCR. The transmit circuitry also includes automatic timestamp insertion. The TSB12LV42 has an empty packet insertion function that will automatically insert a number of empty packets within a frame. These empty packets smooth out bursty data so that it is easier to handle for the receiving node, whose FIFOs are designed for evenly distributed data.

2.4 Microprocessor/Microcontroller Interface

The microprocessor/microcontroller interface (MP/MC) is used as the host controller port. It is designed to work with several standard MP/MCs including Motorola 68000, Intel 8051, and ARM processors. The interface supports both 8-bit and 16-bit wide data busses as well as both little endian and big endian microcontrollers. This interface has two basic modes of operation: handshake Mode and blind access mode. See Section 4.2, *Microprocessor Interface*, 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 sent by the microprocessor/microcontroller. The ACTX FIFO can also be used to support asynchronous traffic at very low data rates. Asynchronous packets are written into the FIFO and transmitted using the ACRXF, ACTXC, and ACTXCU registers. See Section 3.3.1, *Transmitting Asynchronous Control Packets*, for more details concerning the ACTX.

2.5.2 Asynchronous Control Receive FIFO (ACRX)

The ACRX FIFO is typically used to receive asynchronous control packets other than self-ID packets. Regular asynchronous control packets are usually received to the ACRX FIFO. This FIFO is accessible by the microcontroller port through the ACRX register. See Section 3.4.1, *Receiving Asynchronous Packets*, for more details concerning the ACRX.

2.5.3 Broadcast Write Receive FIFO (BWRX)

The BWRX FIFO is typically used to receive asynchronous broadcast write request packets. See Section 3.4.1, *Receiving Asynchronous Packets*, for more detail concerning the BWRX.

2.6 Physical Layer

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/receiving acknowledges. The TSB12LV42 supports Texas Instruments bus-holder circuitry on the Phy-link interface terminals. By using the internal bus holders, the user avoids the need for the complex Annex J isolation method. The bus holders are enabled by connecting the $\overline{\text{ISOLAT}}$ terminal to ground.

2.7 Configuration Register (CFR)

The TSB12LV42 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. Since the microcontroller interface is either 8 or 16-bits wide, it must perform a byte stacking/unstacking operation internally on the incoming (write) or outgoing (read) microcontroller data. Chapter 5 gives a map of all the registers and detailed descriptions of all the register bits.

3 Functional Description and Data Formats

3.1 Overview

The TSB12LV42 is a 1394 interface for high-speed audio, video, and data applications at up to 200 Mb/s. For these high-speed applications a bulky data interface (BDIF) has been implemented that supports long term data rates up to 60 Mb/s. Burst data rates, however, can go up to 160 Mb/s.

The TSB12LV42 contains two FIFOs that are a 256-byte control FIFO (Control FIFO) and an 8K-byte BDIF FIFO. These two FIFOs are further subdivided into smaller logical FIFOs.

Bulky data is usually buffered in the BDIF FIFO. The BDIF FIFO supports DV, asynchronous, and isochronous formatted traffic for receive and transmit.

Asynchronous packets (for 1394 bus control/status) are usually written to or read from the Control FIFO. For lower data rates the Control FIFO can also be used to buffer asynchronous application data. Based on destination address, received asynchronous request packets may be steered into either the Control FIFO or the BDIF FIFO.

A separate self-ID-FIFO allows faster BUS setup and reduces software complexity. The 256-Byte Control FIFO (Control FIFO) is partitioned into three logical FIFOs. The size of these three logical FIFOs is programmable on quadlet boundaries. These FIFOs are called:

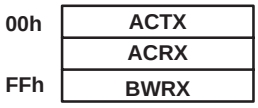
1. Asynchronous control transmit (ACTX) FIFO
2. Asynchronous control receive (ACRX) FIFO
3. Broadcast write receive (BWRX) FIFO.

The 8K-Byte BDIF FIFO is partitioned into six logical FIFOs. The size of these FIFOs is programmable on four quadlet (hexlet) boundaries. These FIFOs are called:

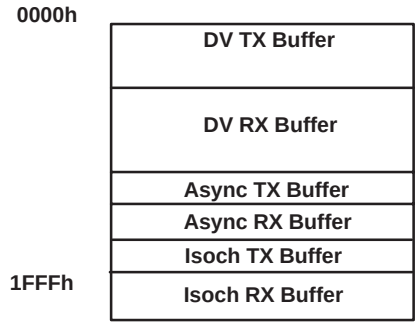
1. BDIF DV transmit (BDTX) FIFO
2. BDIF DV receive (BDRX) FIFO
3. BDIF asynchronous transmit (BATX) FIFO
4. BDIF asynchronous receive (BARX) FIFO
5. BDIF isochronous transmit (BITX) FIFO
6. BDIF isochronous receive (BIRX) FIFO

The Control and BDIF FIFOs are structured are as follows:

256-Byte Control FIFO Structure



8K-Byte BDIF FIFO Structure



The TSB12LV42 (with built-in programmable Endianness) interfaces directly to most microprocessors and microcontrollers.

3.2 DV on 1394 Overview

3.2.1 DV interface

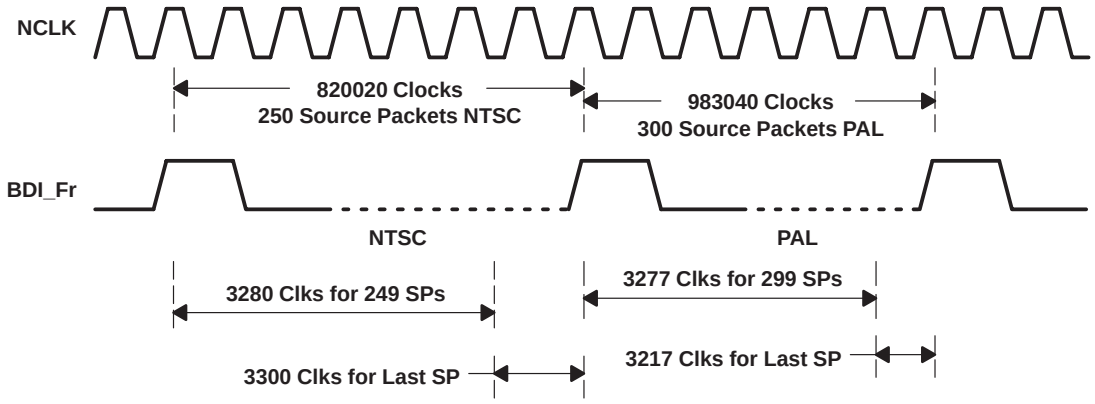


Figure 3–1. Example of a Source Packet Transmit Event

A source packet (SP) event occurs 250 times in 1 NTSC frame or 300 times in 1 PAL frame. NCLK is an internal 24.576-MHz clock derived from the 49.152-MHz SCLK. The first BDI_FR pulse starts the source packet counter used to generate the empty packet insertion algorithm. The TSB12LV42 provides automatic empty packet insertion on transmit to evenly distribute the 250/300 source packets within a frame. Turning off the appropriate bit in the MDCR Register can turn off this feature. In one NTSC frame, there are 820020.0 NCLKs and therefore 3280.08 NCLKs per source packet. That is equivalent to 249 source packets of 3280 NCLKs and one source packet of 3300 NCLKs. For NTSC a SP event is signaled every 3280 clocks for the first 249 SP event and the last SP event is signaled after 3300 clocks following the 249th event. This yields a 33.367-ms frame rate (40.69 ((3280 $\times$ 249) + 3300)). For PAL the time for 1 frame is 40.00 ms (40.69 ((3277 $\times$ 299) + 3217)).

Cycle synchronous (CS) events occurs every 125 μs (this is the isochronous cycle base rate). During each isochronous cycle a complete SP is transmitted or an empty packet (EP) is transmitted. If two contiguous CS events occur without an SP event occurring, then an empty packet is forced in the current isochronous cycle regardless if a complete source packet is available in the FIFO. If a SP event occurs between two CS events but a complete source packet is not available in the FIFO, then an empty packet is transmitted in the current isochronous cycle.

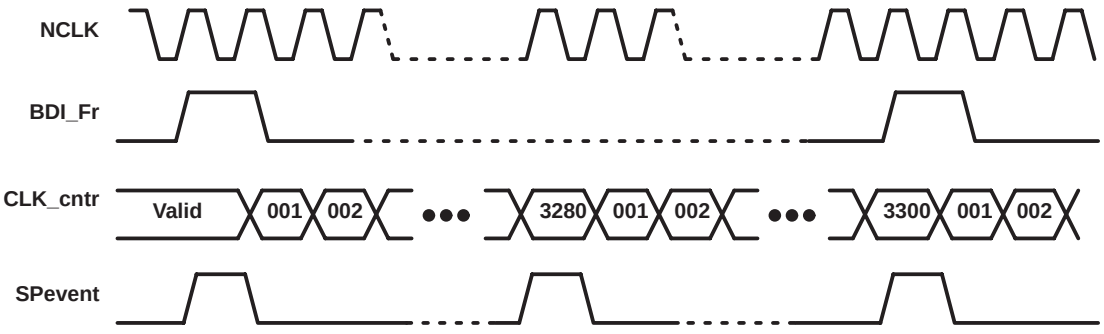


Figure 3–2. Source Packet Transmit Event Timing

NOTE: BDI_Fr has already been synchronized with NCLK.

3.2.2 DV Bandwidth on IEEE1394

Table 3–1. DV Bandwidth on IEEE 1394

MAX SP-BW† (Mbits/s)	MAX 1394-BW‡ (Mbits/s)	POSSIBLE DV BUS PACKET SIZE INCLUDING CIP, 1394-Hdr, CRCs (Bytes)
30.72	32	20/500

† SP-BW: Source package bandwidth (based on 480 byte DV).

‡ 1394-BW: Overall bandwidth of 1394 bus on physical medium (includes 4-byte 1394 packet transmit header, 4-byte packet header CRC, 8-byte CIP header, actual payload, and 4-byte payload CRC. This bandwidth should be allocated by the DV transfer initiator.

3.2.3 DV Transmission over IEEE1394

A DV-packet on the 1394 bus:

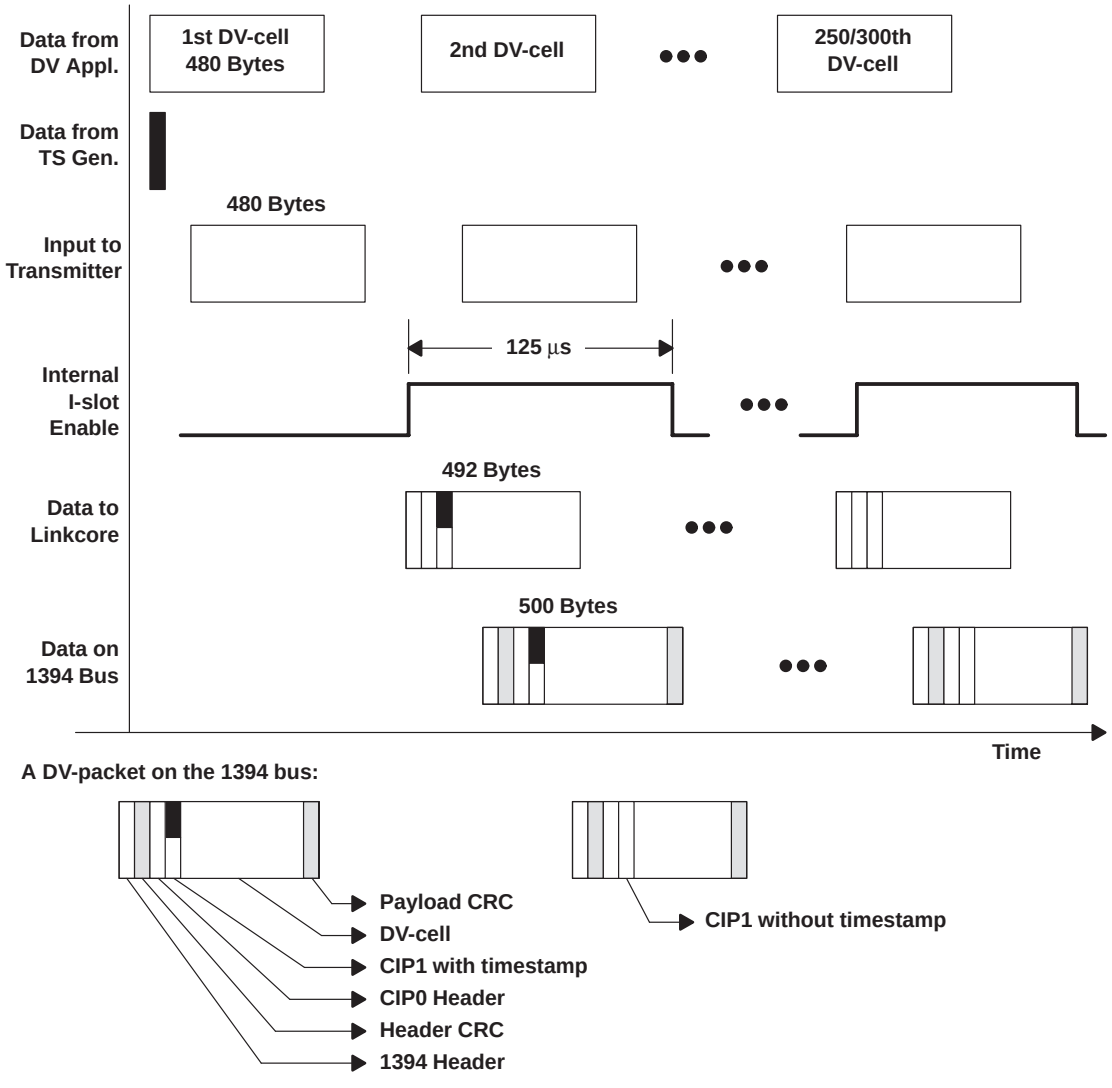
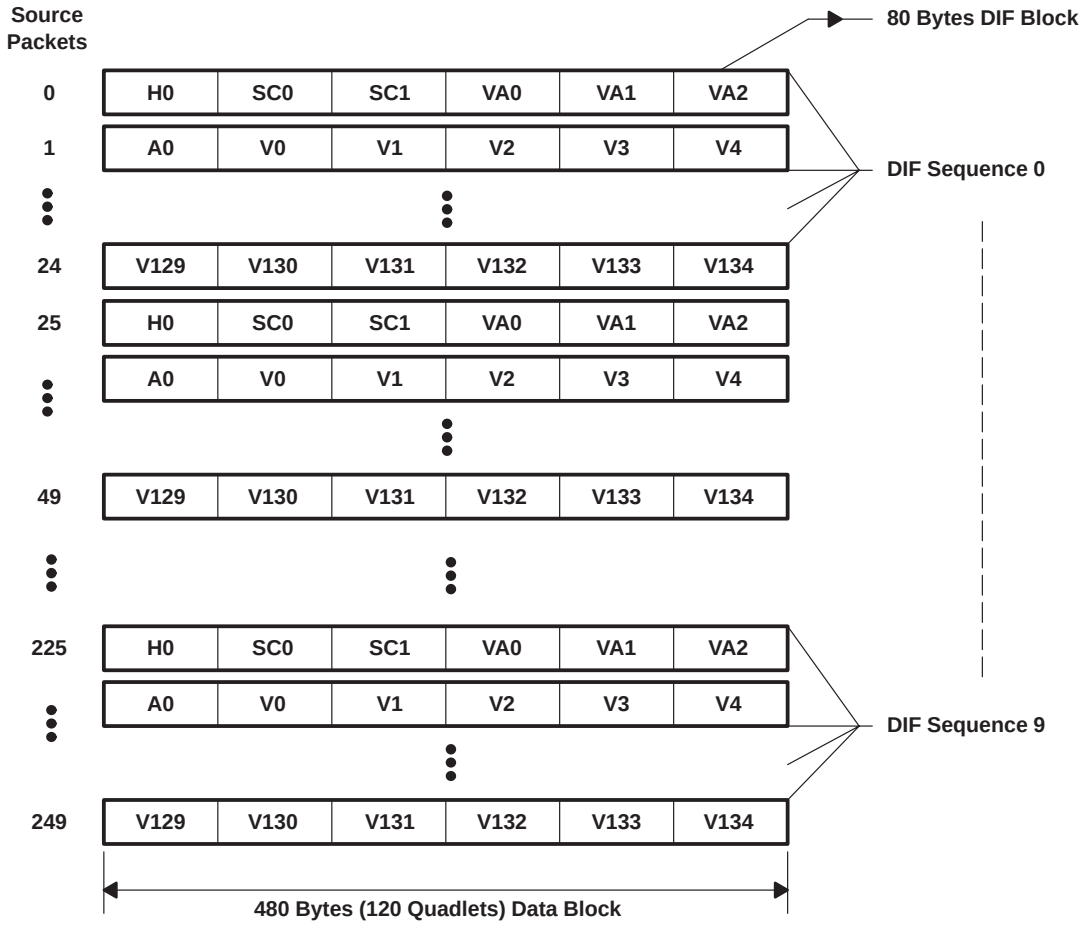


Figure 3–3. DV Packet on 1394 Bus

3.2.4 Source Packet/DIF Block Format



NOTES: A. SD-DVCR 525-50 System is identical to 525-60 system except the number of source packets is 300
B. H0 = Header DIF block
SCi = Subcode DIF block (i = 0, 1)
VAi = VAUX DIF block i (i = 0, 1, 2)
Vi = Video DIF block (i = 0 – 134)
Ai = Audio DIF block (i = 0 – 8)

Figure 3–4. DIF Block H0

The H0 DIF block is inserted into the first 80 bytes every 25th packet. The TSB12LV42 gives the system designer the option to automatically insert the 80 byte DIF block before transmit. The value of the H0 header DIF block is programmable via internal registers 1A4h and 1A8h.

Figure 3–5 shows the H0 header DIF block. Bytes 0–7 can be inserted by the link core or can be provided by the application with the data.

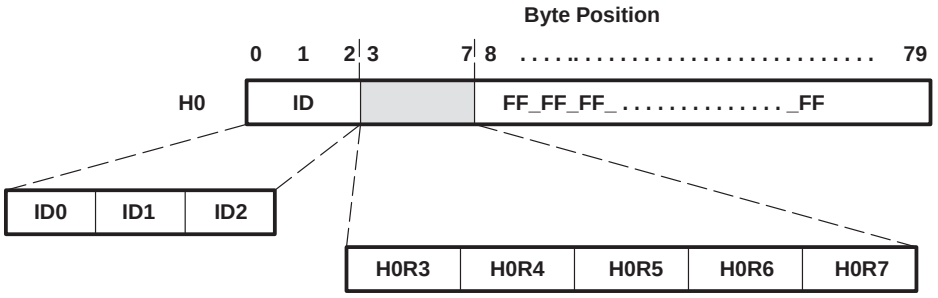


Figure 3–5. ID Data In DIF Block

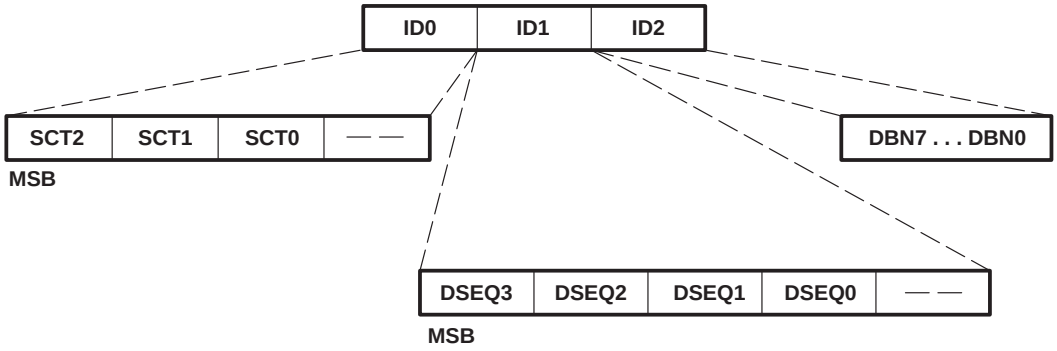
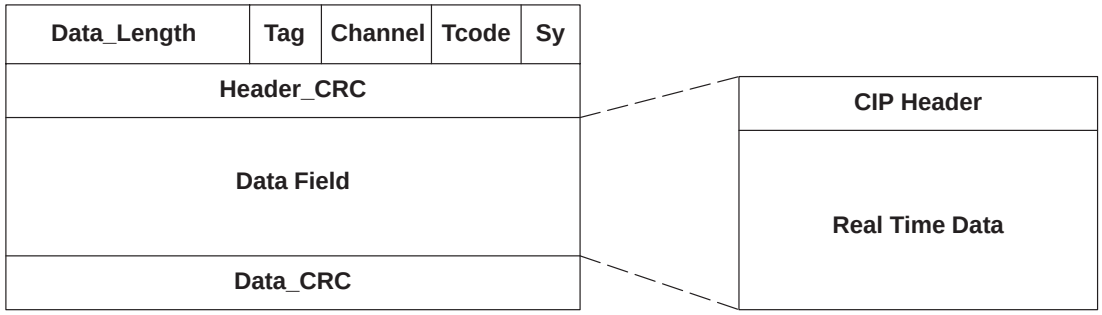


Figure 3–6. ID Data in DIF Block

3.2.5 DV Packets CIP Header Calculations

The TSB12LV42 has an option to automatically insert CIP headers and timestamps during transmission. The CIP headers, or common isochronous packet headers, follow the format of the IEC 61883-2 standard for transmitting SD-DVCR data over 1394. The values of the CIP headers are programmable in registers 1CCh and 1D0h. The TSB12LV42 also has the option to automatically calculate and insert timestamp values into the CIP1 header. The timestamp is inserted either into the first transmitted packet in the next isochronous cycle (register F0h, bit 11 INTSSP=0) or into the first full data packet of the next frame (register F0h, bit 11 INTSSP=1).



Length field is either 488 bytes (DV-Source Packet) or 8 bytes (empty packet).

Figure 3–7. DV Source Packet Format

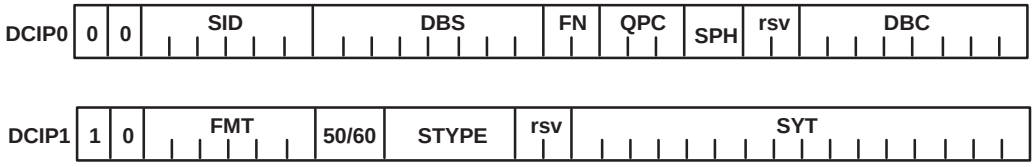


Figure 3–8. CIP Header Format

static values:

DXH: tag,chanNum, spd, sy (1394 Isochronous Header)
DCIPX0:SID, DBS, FN, QPC, SPH, DBC
DCIPX1:FMT, 50/60, STYPE

calculated values:

DXH: length (1394 Isochronous Header)
DCIPX1: SYT
DCIPX0: DBC

STYPE	50/60	50/60
	0	1
00000	525–60 System†	625–50 System†
00001	Reserved	Reserved
00010	1125–60 System	1250–50 System
00011 . . . 11111	Reserved	Reserved

† 525-60 system: The 525-line system with a frame frequency of 29.97 Hz
625-50 system: The 625-line system with a frame frequency of 25.00 Hz
SD-DVCR: Standard-definition digital-video cassette recorder

3.3 Transmit Operation

The functional description for transmission is shown in the following sections. The transmit format describes the expected organization of data presented to the TSB12LV42 at the host-bus interface.

3.3.1 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 3.6, *Asynchronous Transmit Data Formats (Host Bus to TSB12LV42)*.

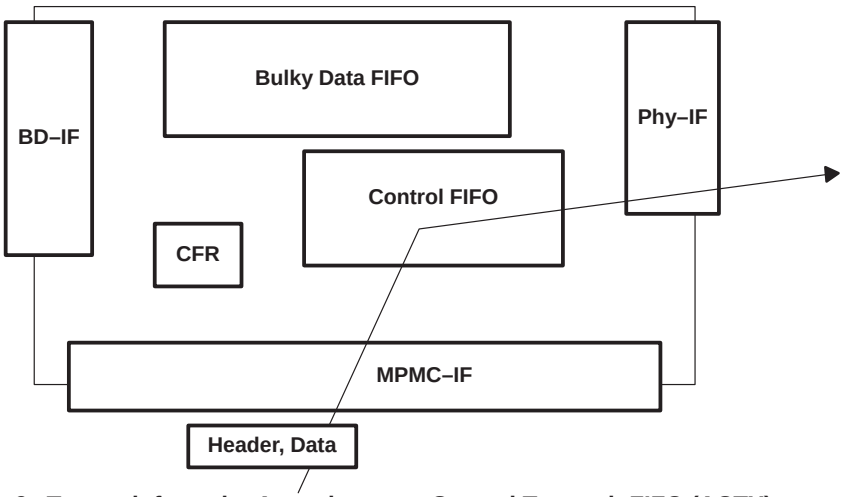


Figure 3–9. Transmit from the Asynchronous Control Transmit FIFO (ACTX)

To transmit an asynchronous packet from the ACTX:

- Register 80h (asynchronous cControl 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.

3.3.2 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 ECh (asynchronous/isochronous application data control register).

The BATX has an auto-packetization feature. This allows the user to program header registers within the DVLynx CFRs and supply raw data to the DVLynx for transmit. The DVLynx 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 3.3.4, *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 3.6, *Asynchronous Transmit Data Formats (Host Bus to TSB12LV42)*. The DVLynx 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 ECh 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 ECh in this mode: ATENABLE=1, BDAXE=1, AHIM=1) (see Figure 3–10).

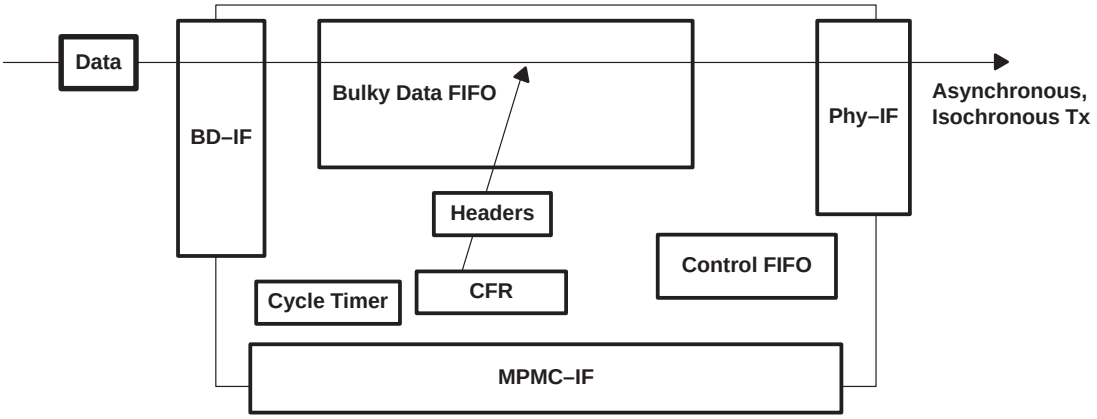


Figure 3–10. Transmit Asynchronous/Isochronous Data from BATX by the Bulky Data Interface with Auto-Packetization

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 ECh in this mode: ATENABLE=1, BDAXE=0, AHIM=1) (see Figure 3–11).

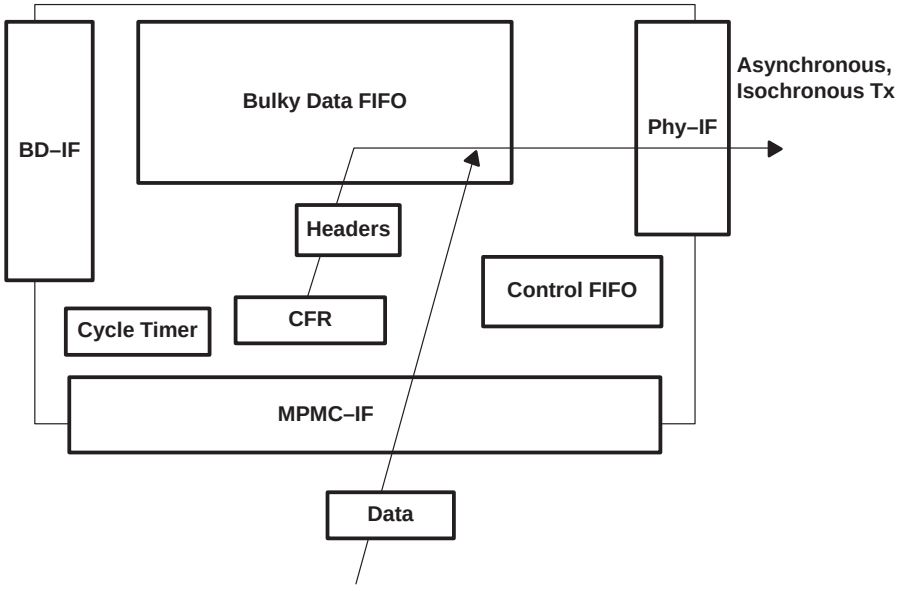


Figure 3–11. Transmit Asynchronous/Isochronous Data from BATX by the MP/MC Interface with Auto-Packetization

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 format as shown in Section 3.6, *Asynchronous Transmit Data Formats (Host Bus to TSB12LV42)*. 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 ECh in this mode: ATENABLE=1, BDAXE=1, AHIM=0) (see Figure 3–12).

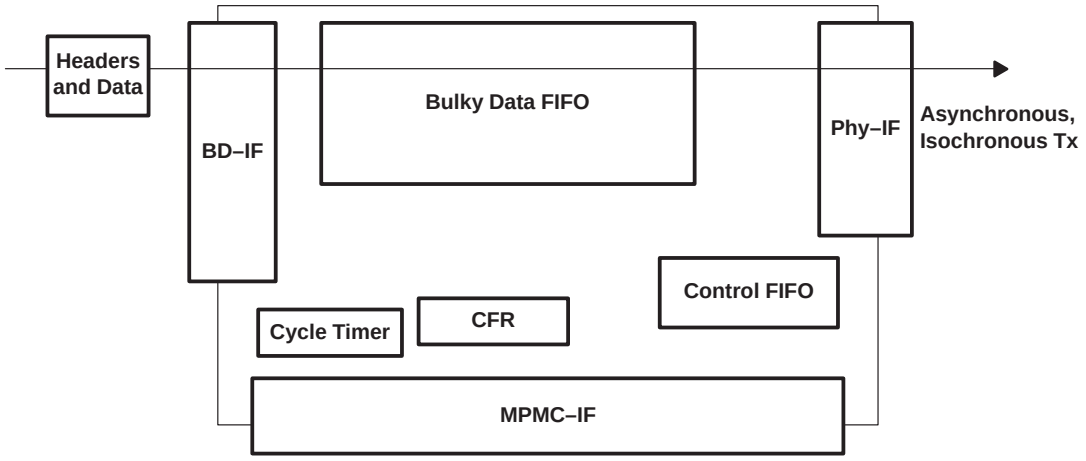


Figure 3–12. Transmit Asynchronous/Isochronous Data from BATX by the Bulky Data Interface, No Auto-Packetization

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 format as shown in Section 3.6, *Asynchronous Transmit Data Formats (Host Bus to TSB12LV42)*. 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 ECh in this mode: ATENABLE=1, BDAXE=0, AHIM=0) (see Figure 3–13).

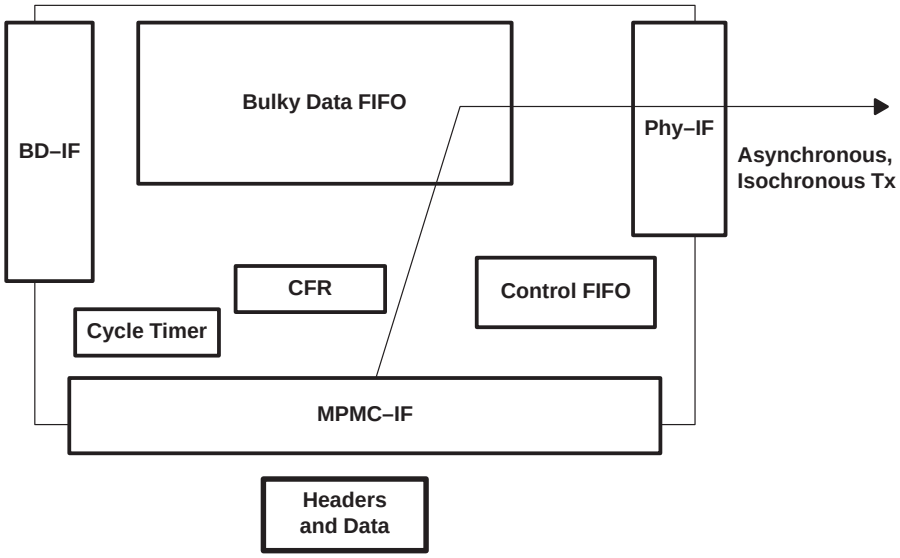


Figure 3–13. Transmit Asynchronous/Isochronous Data from BATX by the MP/MC Interface, No Auto-Packetization

General Asynchronous Transmit Notes

- Packet llush: The entire BATX FIFO can be flushed by setting the AXFLSH bit in register ECh.
- 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 DVLynx 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 08h (BATACK 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 04h (CATACK 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.

3.3.3 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 ECh (asynchronous/isochronous application data control register).

The BITX has an auto-packetization feature which allows the user to program header registers within the DVLynx CFRs. The application can then supply raw data to the DVLynx for transmit, and the DVLynx 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 3.3.4, *Byte Padding*, for more information on byte padding. If the number of bytes in the packet is different from 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). The header programmed in register 1C0h must match the format given in Section 3.7, *Isochronous Transmit and Receive Data Formats (Host Bus to TSB12LV42)*. The DVLynx 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 ECh that are 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 isochronous 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 3–10) (settings for register ECh 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 all the 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 3–11) (settings for register ECh 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 3.7, *Isochronous Transmit and Receive Data Formats (Host Bus to TSB12LV42)*. 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 ECh 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 interface 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 3.7, *Isochronous Transmit and Receive Data Formats (Host Bus to TSB12LV42)*. 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 3–13) (settings for register ECh 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 register ECh (AICR).
- 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 DVLynx 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 DVLynx arbitrates for the bus after every packet that is sent. No concatenated isochronous subactions are allowed.

3.3.4 Byte Padding

All packets on 1394 must end on a quadlet boundary (4 byte boundary). The DVLynx can insert padding bits to a data packet that is not quadlet aligned. The DVLynx 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]) indicate 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.

3.3.5 Transmitting DV Formatted Isochronous Packets

Naming convention:

480 bytes = 120 quadlets
120 quadlets = 1 source packet = 1 data block
250 source packets = 1 frame for NTSC
300 source packets = 1 frame for PAL

DV data may be transmitted via the bulky data FIFO or the microcrontoller interface. The bulky data FIFO DV transmit modes of operation are shown below. The following modes are supported; headers and data from BDIF/micro, data from BDIF/micro, and headers from CFRs. In addition, DV transmit gives the option of automatic timestamping, empty packet insertion, and inserting H0 DIF blocks. Most DV transmit functions are controlled by the DCR register at address F0h.

If the BDDXE bit (DCR register) is set to logic 1, then the BDIF has access to the BDTX-FIFO. DV source packets (SP) are usually transmitted via the bulky data interface. Here the single bytes are packetized into

quadlets (1 quadlet = 4 bytes of data = 32 bits). After four bytes or 1 quadlet is written to the BDIF, the complete quadlet is written to the bulky data DV transmit FIFO (BDTX). The first byte of a DV source packet is indicated by setting the BDIF[2..0] signals to 101b. An internal cyclic counter keeps track of DV SP boundaries. DV SP, frame size, and DIF blocks are shown in Section 3.2, *DV on 1394 Overview*.

If the BDDXE bit is set low, MC interface has access to the BDTX FIFO. Quadlets are written into the BDTX FIFO through registers 160h and 164h.

The DHIM bit (DCR Register) selects if the 1394 header registers should be interleaved automatically (DHIM=1) or if complete formatted DV source packet and 1394 headers are expected from the application data source (DHIM=0) (see Table 3–2).

The H0INST bit (DCR register) selects if the H0 DIF block is automatically inserted by DVLynx (H0INST=1) or if the H0 DIF block is supplied by the application (H0INST=0). If the H0INST bit is 1, indicating automatic insertion of the H0 DIF block, the hardware still expects 480 bytes input for each source packet. See Table 3–2.

Table 3–2. DHIM/H0INST Settings for Different Header Insertion Schemes

MODE	BDDXE	DHIM	H0INST	TIME STAMP FROM	HEADERS FROM	DATA FROM	DIF H0 FROM
1	1	1	1	Cycle timer	CFR (internal)	BDIF	CFR (internal)
2	1	1	0	Cycle timer	CFR (internal)	BDIF	BDIF
3	1	0	1	BDIF	BDIF	BDIF	CFR (internal)
4	1	0	0	BDIF	BDIF	BDIF	BDIF
5	0	1	1	Cycle timer	CFR (internal)	MP/MC	CFR (internal)
6	0	1	0	Cycle timer	CFR (internal)	MP/MC	MP/MC
7	0	0	1	MP/MC	MP/MC	MP/MC	CFR (internal)
8	0	0	0	MP/MC	MP/MC	MP/MC	MP/MC

On each isochronous transmission cycle the DV framer checks if 480 valid bytes are available in the BDTX FIFO. If so, these bytes are appended with 8 bytes of CIP header information (if DHIM=1), taken as the payload for a DV packet and sent over the 1394 bus. If 480 valid bytes are not available in the BDTX FIFO, an empty DV packet is sent. Possible 1394 bus packet sizes are 500 bytes for a complete packet and 20 bytes for an empty packet.

A 20 quadlet DIF block (H0) is automatically inserted every 26 source packets of a DV frame. A total of 250 source packets make up a DV frame for NTSC and, 120 quadlets make up a source packet. These DIF blocks are numbered in a sequence, 0 through 9. The first DIF block, DIF Sequence 0, contains header information, subcode, audio and video data. The following DIF blocks in the sequence contain similar information. However, DIF Sequence 9 also indicates the end of a frame.

Mode 1: Data from bulky data interface, headers/timestamp/H0 automatically inserted

The application provides a 480 byte data packet to the Bulky Data Interface. The DVLynx automatically formats the packet and transmits it over 1394. The DVLynx only transmits a packet if there are 480 bytes of data available in the bulky data FIFO. If 480 bytes are not available, an empty packet is sent. The DVLynx automatically includes the 1394 isochronous header, two-quadlet CIP header, timestamp (if applicable), and the H0 DIF header block (if applicable) at the beginning of each packet (see Figure 3–14).

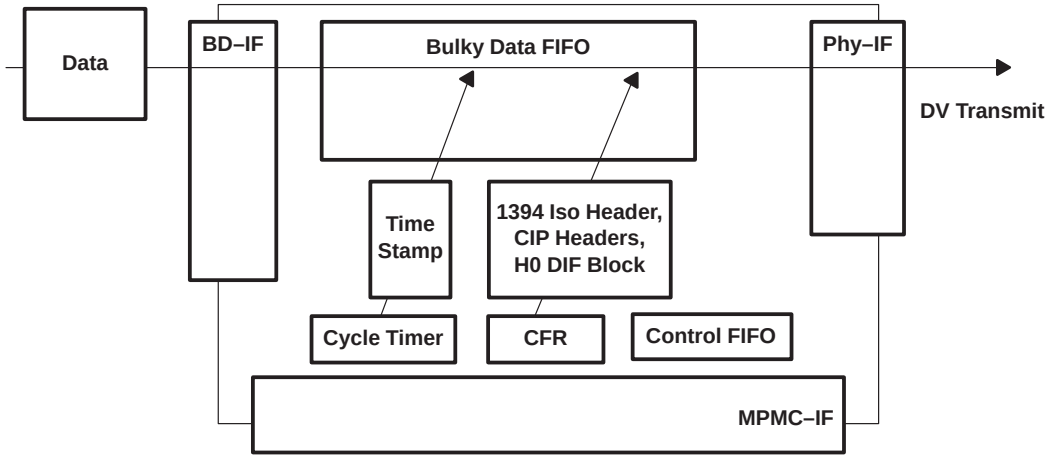


Figure 3–14. Data from Bulky Data Interface, Headers/Timestamp/H0 Automatically Inserted

Mode 2: Data and H0 header from bulky data interface, headers/timestamp automatically inserted

The application provides a 480 byte data packet to the bulky data interface, including the H0 DIF Header. The DVLynx automatically formats the packet and transmits it over 1394. The DVLynx only transmits a packet if there are 480 bytes of data available in the bulky data FIFO. If 480 bytes are not available, an empty packet is sent. The DVLynx automatically includes the 1394 isochronous header, two-quadlet CIP header, and timestamp (if applicable) at the beginning of each packet (see Figure 3–15).

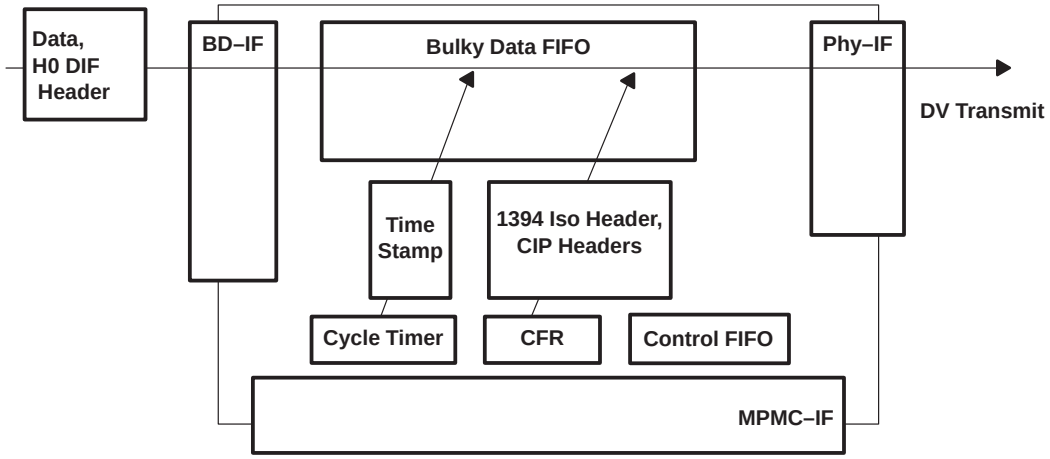


Figure 3–15. Data and H0 Header from Bulky Data Interface, Headers/Timestamp Automatically Inserted

Mode 3: Data, 1394 isochronous and CIP headers, and timestamp provided by the application through BDIF, H0 DIF header automatically inserted by CFRs

The application provides a 492-byte data packet to the bulky data interface, including the 1394 Isochronous header, two quadlet CIP headers, and timestamp. The DVLynx automatically formats the packet and transmits it over 1394. The DVLynx only transmits a packet if there are 492 bytes of data available in the bulky data FIFO. If 492 bytes are not available, an empty packet is sent. The DVLynx automatically includes H0 DIF Header when necessary (see Figure 3–16).

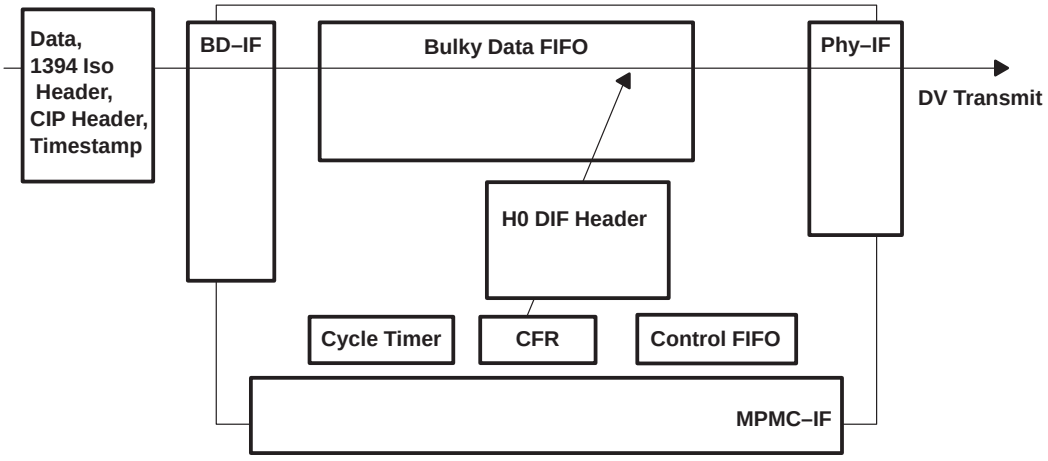


Figure 3–16. Data 1394 Isochronous and CIP Headers

Mode 4: All data from BDIF, including 1394 isochronous header, cip headers, timestamp, data, and H0 DIF header

The application provides a 492-byte data packet to the bulky data interface, including the 1394 Isochronous header, two quadlet CIP header, timestamp, and H0 DIF header when applicable. The DVLynx automatically formats the packet and transmits it over 1394. The DVLynx only transmits a packet if there are 492 bytes of data available in the bulky data FIFO. If 492 bytes are not available, an empty packet is sent. The DVLynx does not insert any headers, except for CRC checking quadlets (see Figure 3–17).

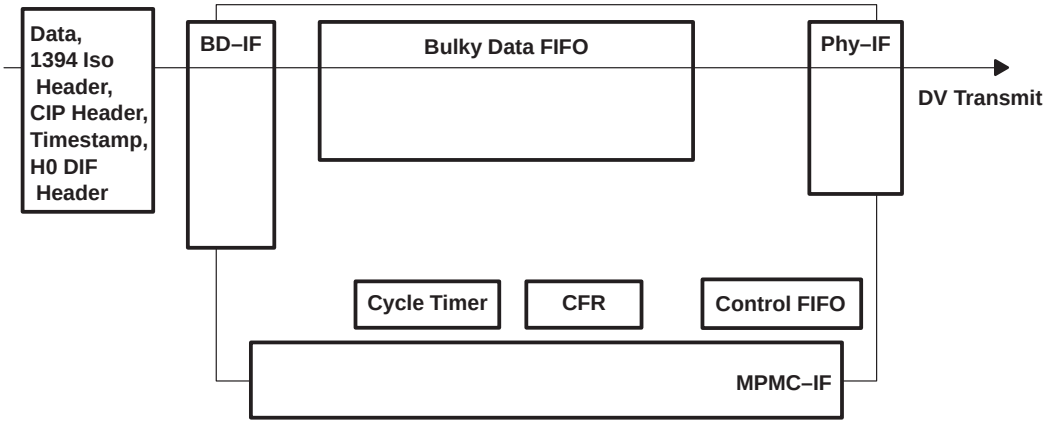


Figure 3–17. All Data from BDIF, including 1394 Isochronous Header

Mode 5: Data from bulky data interface, headers/timestamp/H0 automatically inserted

This mode is similar to Mode 1 except that the data is written into the bulky data FIFO by the microprocessor, one quadlet at a time (see Figure 3–14).

Mode 6: Data and H0 header from bulky data interface, headers/timestamp automatically inserted

This mode is similar to Mode 2 except that the data and H0 DIF header is written into the bulky data FIFO by the microprocessor one quadlet at a time (see Figure 3–15).

Mode 7: Data, 1394 isochronous and CIP headers, and timestamp provided by the application through BDIF, H0 DIF header automatically inserted by CFRs

This mode is similar to Mode 3 except that the data, 1394 and CIP headers, and timestamp are written into the bulky data FIFO by the microprocessor one quadlet at a time (see Figure 3–16).

Mode 8: Data, 1394 isochronous and CIP headers, and timestamp provided by the application through BDIF, H0 DIF header automatically inserted by CFRs

This mode is similar to Mode 3 except that the data, 1394, CIP, and H0 headers and timestamp are written into the bulky data FIFO by the microprocessor one quadlet at a time (see Figure 3–17).

3.3.5.1 Empty Packet Insertion (DCR.EPINST=1)

Most receiving 1394 interfaces have FIFOs sized to accommodate evenly distributed data. These receiving nodes can overrun if received data is bursty. To solve this problem, the DVLynx inserts empty packets evenly throughout the data stream. For DVLynx, a null packet is sent whenever 2 cycle sync events occur without a source packet between them. For NTSC, an average of 16.9 null packets are inserted per frame. For PAL, an average of 20 null packets are inserted per frame.

3.3.5.2 Sending DV Data with Timestamps

If the DHIM bit is set high and the frame pulse (BDI_FR) is detected a timestamp is calculated and inserted into the SYT field of the MDCIPX1 register. This timestamp is calculated from the value of the transmit offset register (XTO) and the cycle timer.

If the DHIM bit is low then the timestamp is expected from the A/D/V application or from the MP/MC if DV traffic is sent via the MP/MC interface. The 1394 header bytes and the CIP headers must also be provided by the application.

3.3.5.3 DV Intermediate Mode (DCR.DVSUB=1)

DV intermediate mode determines whether complete packets or empty packets are transmitted. When DVSUB=1, only empty packets are transmitted. If DVSUB=1 and in receive mode, only H0 and CIP are saved to, DRX0, DRX1, DCIPro, and DCIPR1. When DVSUB=0, normal operation of complete DV packet transmission and reception occurs. The default value for DVSUB is 0.

3.3.5.4 Transmit Thresholds

By default, the DVLynx does not transmit a packet until at least 480 bytes of data has accumulated in the bulky data FIFO. An additional transmit threshold can be added to this 480 byte requirement for either the first packet only OR for all packets transmitted. Bit 16 in register F8h (FMISC.INITXTSEL) programs whether only the first packet has this extra transmit threshold or if every transmitted packet has a threshold. The default value is 0. Bits 18 and 19 in register F8h (FMISC.XTSEL0,XTSEL1) choose the value of the threshold. These values are in addition to the 480 byte data requirement. The default value is 00 where no additional threshold is required.

3.3.5.5 Data Block Counter (DBC)

The DBC is incremented by one on each successive transmission of a source packet (SP). When an empty packet or another SP follows a SP, the DBC is incremented. If an empty packet or a SP follows an empty

packet, then the DBC is NOT incremented. The DBC counter is modulo 255 and is used to keep transmitter and receiver synchronized. Figure 3–18 shows an example of the DBC for a frame of data.

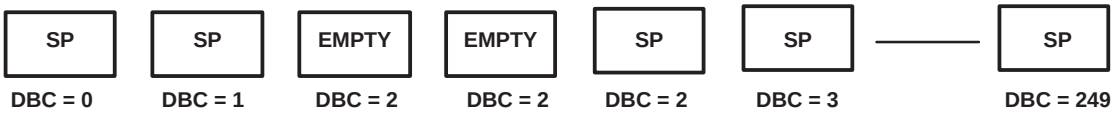


Figure 3–18. DBC Example

The DBC can be initialized to any 8 bit value by simply writing to DCIPX0 Register. The default power on value is 0.

3.3.5.6 Data Block Size (DBS)

The DBS is a fixed constant and is inserted by hardware on transmission of a source packet. The value of DBS is 78h (120) which indicates 120 quadlets per source packet.

3.4 Receive Operation

The functional description of the reception of data is shown in the following sections. The receive formats describe the data format that the TSB12LV42 presents to the host-bus interface.

3.4.1 Receiving Asynchronous Packets

Asynchronous traffic can be directed into different 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 or 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 (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 (38h). Unexpected response packets do not have tlabels and tcodes that match the programmed value in register 38h.

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	Nonbroadcast nonregister 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 nonbroadcast 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
		Nonbroadcast 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

3.4.1.1 Receiving Asynchronous Control Packets

- Reads from broadcast write receive FIFO (BWRX)

  The BWRX FIFO is mapped to register C4h (broadcast write receive FIFO). Reads 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). Reads here access the ACRX FIFO. The status of this FIFO is available in register 50h (asynchronous control data receive FIFO status).

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

When the DVLynx 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 DVLynx. 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 asynchronous 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 ECh controls this function. The first packet in a queue of asynchronous packets stored to the BARX FIFO automatically have their header and trailer quadlets stored to registers 118h–128h. The ARAV interrupt is generated to the application when this operation completes. When the header/data/trailer are received to the BARX, it has the format shown in Section 3.6, *Asynchronous Transmit Data Formats (Host Bus to TSB12LV42)*.

The four methods of receiving asynchronous data to the BARX are shown in Table 3–3. The control signals located in register ECh that are necessary for these four modes are summarized in the following text. A detailed description is also included for each mode.

Table 3–3. Asynchronous Receive Modes

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 DVLynx, 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 ECh for this mode: ARENABLE=1, ARHS=1, BDARE=1) (see Figure 3–19).

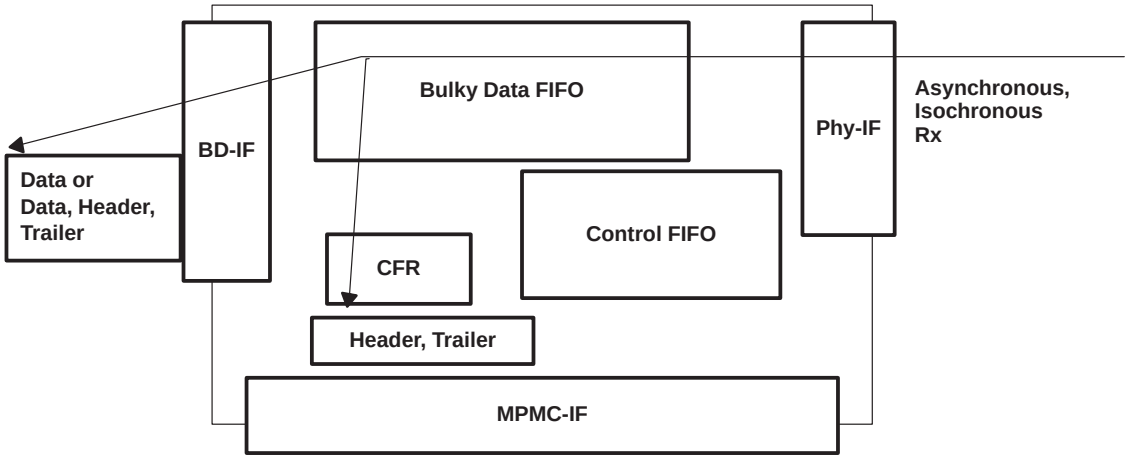


Figure 3–19. Receive Asynchronous/Isochronous Data to Bulky Data Interface

Mode 2: Receiving asynchronous data to the microprocessor interface using the BARX, headers are stripped

The DVLynx 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 register 114h (asynchronous application data receive FIFO) (settings for register ECh for this mode: ARENABLE=1, ARHS=1, BDARE=0) (see Figure 3–20).

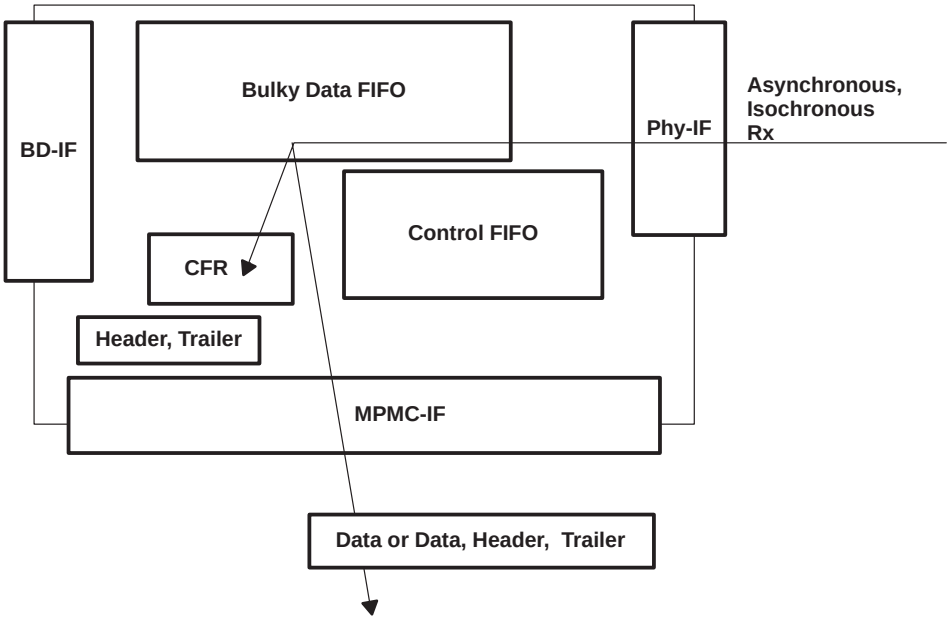


Figure 3–20. Receive Asynchronous/Isochronous Data to Microprocessor Interface

Mode 3: Receiving asynchronous header/data/trailer to the bulky data interface using the BARX

Data is received by the DVLynx, and the headers and trailer are automatically copied to registers 118 – 128h. The headers and data 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 BARX FIFO has the same format as described in Section 3.6, *Asynchronous Transmit Data Formats (Host Bus to TSB12LV42)*. The BDOAVAIL signal is activated once a full quadlet is in the FIFO (settings for register ECh for this mode: ARENABLE=1, ARHS=0, BDARE=1) (see Figure 3–19).

Mode 4: Receiving asynchronous header/data/trailer to the microprocessor interface using the BARX

The DVLynx 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 BARX FIFO has the same format as described in Section 3.6, *Asynchronous Transmit Data Formats (Host Bus to TSB12LV42)*. The microprocessor has access to the BARX through registers 114h (asynchronous application data receive FIFO) (settings for register ECh for this mode: ARENABLE=1, ARHS=0, BDARE=0) (see Figure 3–20).

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 148h, 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 0Ch, 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.

3.4.2 Receiving Isochronous Packets

When the DVLynx 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 DVLynx 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 are not 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 ECh controls this function.

The isochronous packets stored in the BIRX FIFO automatically have their header and trailer quadlets stored to registers. The IRAV interrupt is generated to the application when this operation completes. When the header/data/trailer are received in the BIRX FIFO, the FIFO has the format shown in Section 3.7, *Isochronous Transmit and Receive Data Formats (Host Bus to TSB12LV42)*.

The four methods of receiving isochronous data to the BIRX FIFO are shown in Table 3–4. The control signals located in register ECh that are necessary for these four modes are summarized in the following text. A detailed description is also included for each mode.

Table 3–4. Receiving Isochronous data to the BIRX FIFO

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 DVLynx, 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 ECh for this mode: IRENABLE=1, IRHS=1, BDIRE=1) (see Figure 3–19).

Mode 2: Receiving Isochronous data to the microprocessor interface using the BIRX, headers are stripped

The DVLynx 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 register 13Ch (isochronous receive FIFO) (settings for register ECh for this mode: IRENABLE=1, IRHS=1, BDIRE=0) (see Figure 3–20).

Mode 3: Receiving isochronous header/data/trailer to the bulky data interface using the BIRX

Data is received by the DVLynx, 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 BIRX has the format described in Section 3.7, *Isochronous Transmit and Receive Data Formats (Host Bus to TSB12LV42).* The BDOAVAIL signal is activated once a full quadlet is in the FIFO (settings for register ECh for this mode: IRENABLE=1, IRHS=0, BDIRE=1) (see Figure 3–19).

Mode 4: Receiving isochronous header/data/trailer to the microprocessor interface using the BIRX

The DVLynx 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 BIRX has the format described in Section 3.7, *Isochronous Transmit and Receive Data Formats (Host Bus to TSB12LV42).* The microprocessor has access to the BIRX through register 13Ch (Isochronous Receive FIFO) (settings for register ECh for this mode: IRENABLE=1, IRHS=0, BDIRE=0) (see Figure 3–20).

3.4.3 Receiving DV Formatted Isochronous Packets

When the DVLynx link layer controller starts receiving DV data from the 1394 bus via the physical layer, it accepts data from the bus and places it into the DV Receive FIFO (BDRX FIFO) where the BDIF or MP/MC can access it. A DV packet is stored in the BDRX FIFO only if it meets the following conditions:

- Packets are received at Port 0
- TAG field has value 01
- DREN bit in register F0 is set
- FMT field in CIP1 is 000_000

The DVLynx does not put the data into the BDRX FIFO until the beginning of a frame (first source packet of a frame) is detected. DVLynx can detect the beginning of the frame by decoding ID0, ID1, and ID2 of the H0 DIF block. See Figure 3–6. The beginning of the frame is detected when SCT2..SCT0 of ID0 field are

all 0s, when DSEQ3..DSEQ0 of ID1 field are all 0s, and DBN7..DBN0 of ID2 are all 0s. Prior to the beginning of frame detect, all received packets are discarded and nothing is saved in the BDRX FIFO.

Upon the reception of a DV packet, the 1394 packet header quadlet and the 1394 packet trailer are always copied to the DRH and DRT registers. The headers, trailer, and data are then available to the BDIF or MP/MC in a variety of modes.

The DRHS bit (DCR register) determines whether or not the 1394 headers and trailers and the CIP headers are stripped off the DV packet before it is sent to the host. If DRHS is set low, then the complete DV packet, including 1394 header and trailer and CIP headers, is sent to either the BDIF or MP/MC interface. If the DRHS bit is high, then the 1394 header and trailer and CIP header quadlets are stripped off of the DV packet before being sent to the interface. When the headers/data/trailer are received to BDRX FIFO, the FIFO has the format described in Section 3.7, *Isochronous Transmit and Receive Data Formats (Host Bus to TSB12LV42)*.

The software can check if the BDIF (application) or MP/MC should receive the DV packet. The BDDRE bit (DCR register) determines whether or not the MP/MC or BDIF has access to the received DV packet. If the BDDRE bit is low, then MP/MC has access to the BDRX FIFO via the BDRX Register. If BDDRE is high, then the packet is accessed through the BDIF.

On reception of the first source packet of a frame when a correct DV packet has been decoded, the time stamp value is extracted. If a valid time stamp exists, BDO_FR toggles to signal the host of the start of a valid frame once the local timer (CLKTIM register) equals sum of SYT and timestamp offset register RTO. DRELTIM interrupt in the extended interrupt register (EIR) is also generated to signal software of the frame beginning. If the received packet is an empty packet, it is discarded and nothing is saved in the BDRX FIFO. When a complete packet is confirmed in the BDRX FIFO, a DRAV interrupt is generated.

In auto output mode, once a complete source packet is confirmed into the BDRX FIFO, the BDOAVAIL output signal is activated and, 1 BDOCLK period later, the BDRX FIFO begins outputting data onto the bulky data interface (see Section 4.1, *Bulky Data Interface*).

When in command read mode, BDOAVAIL is activated, but data is not dumped from the BDRX FIFO until the application requests to output the data (see Section 4.1, *Bulky Data Interface*).

The modes supported for DV receive are shown in Table 3–5.

Table 3–5. DV Receive Modes

MODE	DRHS	BDDRE	DVSUB	OUTPUT INTERFACE	PACKET FORMAT
1	0	1	0	BDIF	All data including headers and trailer
2	0	0	0	MP/MC	All data including headers and trailer
3	1	1	0	BDIF	Data only (headers are stripped)
4	1	0	0	MP/MC	Data only (headers are stripped)
5	0	0	1	NA	DV intermediate mode: No packets are placed in bulky data FIFO. Only headers/trailer are copied to internal registers.

Mode 1: Header, Data, and Trailer Received at BDIF

The complete DV packet, including 1394 header and trailer and CIP headers is received to the BDIF. The headers and trailer are also automatically copied to internal registers 178h–184h upon reception. Settings for F0 in this mode: DRHS=0, BODRE=1) (see Figure 3–21).

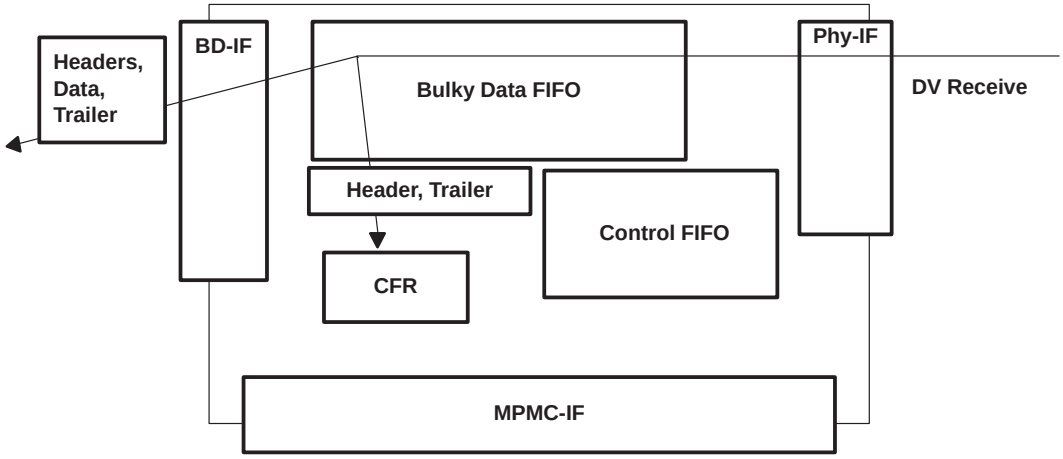


Figure 3–21. Header, Data, and Trailer Received at BDIF

Mode 2: Header, Data, and Trailer Received at BDIF

The complete DV packet, including 1394 header and trailer and CIP headers is sent to MP/MC. The MP/MC has access to the BDRX FIFO via the BDRX register (168h). Settings for F0 in this mode: DRHS=0, BDDRE=1) (see Figure 3–22).

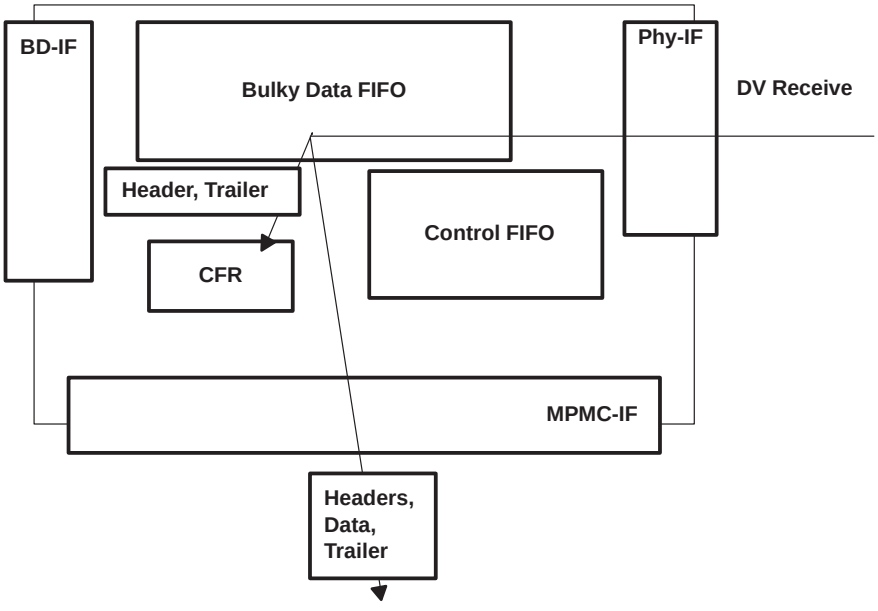


Figure 3–22. Header, Data, and Trailer Received at MP/MC

Mode 3: Header, Trailer Stripped, Data only set to BDIF

The 1394 isochronous header and CIP headers, as well as the packet trailer are stripped off the DV packet before it is received to the BDRX. The BDIF has access to the BDRX. Settings for F0 in this mode: DRHS=1, BDDRE=1 (see Figure 3–23).

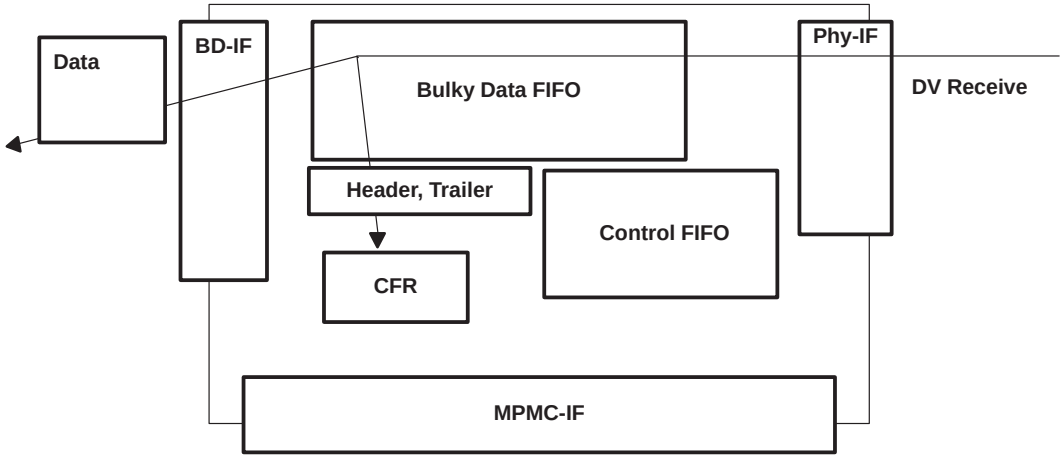


Figure 3–23. Header, Trailer Stripped, Data only sent to BDIF

Mode 4: Header, Trailer Stripped, Data only set to MP/MC

The 1394 isochronous header and CIP headers, as well as the trailer are stripped off the DV packet before it is received to the BDRX. The MP/MC has access to the BDRX. Settings for F0 in this mode: DRHS=0, BDDRE=0 (see Figure 3–24).

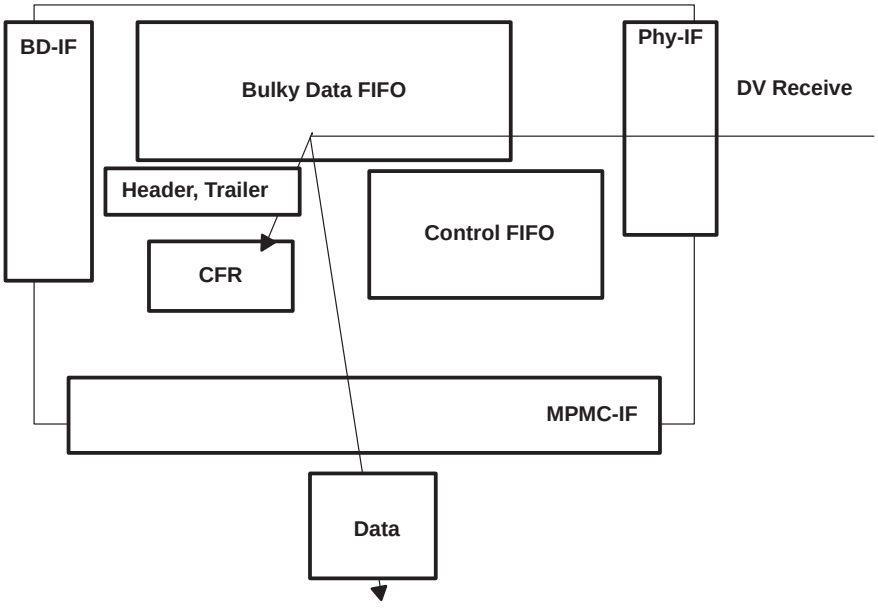


Figure 3–24. Header, Trailer Stripped, Data only set to MP/MC

Mode 5: DV Intermediate Mode Header, Trailer Saved to Registers, Data Discarded

The DCR.DVSUB bit dictates what happens on packet reception and transmission. For receive, if DVSUB is on, no packets are saved in the BDRX FIFO. Only CIP headers and the 8 most significant bytes of H0 are saved to the hardware registers (DCIP0, DCIP1, DRX0, and DRX1). When this occurs, the EIR.DRAV interrupt bit is set. The hardware registers contain only the last value of CIP and H0 received. If DCR.DVSUB is off, then all received source packets are saved in the BDRX FIFO. Whenever the mode is switched on-off/off-on the BDRX FIFO is automatically flushed. The default power is DVSUB off. Settings for F0 in this mode: DVSUB=1 (see Figure 3–25).

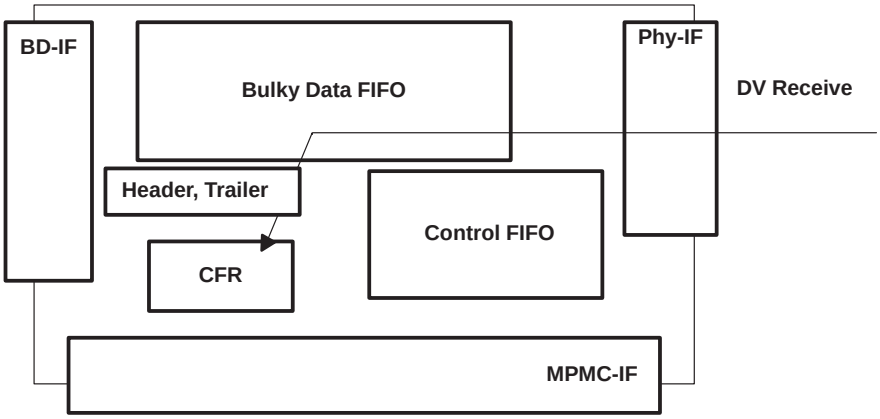


Figure 3–25. DV Sub Mode Header, Trailer Saved to Registers, Data Discarded

3.4.3.1 Release Data Control {DCR.RELDATA}

The release data control bit (DCR.RELDATA) controls the reading requirement sequence on the bulky data interface. When this bit is set to 0 (power on default) the following sequence must be followed in order to receive data at the bulky data interface.

- BDO_FR must be activated
- BDI_FR must be activated
- BDOEN must be activated

There is a delay associated with each step, but delay length is not critical. About 8–10 BDI_CLK can be used for the delay.

If this bit is set to 1 then read data would be gated out to the application on activation of BDOEN regardless of BDO_FR and BDI_FR.

3.4.3.2 DBC Error

The DCR.DBCECNT is an error threshold for data block count (DBC) errors. This value determines if the current packet is discarded (not saved in the BDRX FIFO). For the value DBCENT=0, the current packet is discarded if an error occurs with the DBC, but the BDRX FIFO is not flushed. For the value DBCECNT=1, the current packet is discarded on the first DBC error and the entire BDRX FIFO is flushed. No other packets are saved until the next detection of start frame. DBCECNT values of 2 and 3 operate similarly. The power default is 0.

3.4.3.3 CRC Error

If a data CRC error is detected, and the error was not caused by a data length mismatch, then the entire packet is saved. The DATACRC Interrupt is posted. The error code in the trailer register (DRT.ERRCODE) also indicates that an error has occurred. If a header CRC error or a data CRC error with a problem in data

length is detected, then the packet is discarded and the BDRX FIFO flushed and the appropriate interrupt (IR.DATACRC, IR.HDRERR) is generated. No other packets are saved until the next beginning of frame is detected.

3.5 Timestamps

Timestamping lets DVLynx tell the application when to read data from the receive FIFOs. When a packet is transmitted over 1394, the transmitting DVLynx includes a timestamp with the data. Upon reception, the receiving DVLynx generates a BDO_FR pulse to the application whenever the received timestamp is equal to the local cycle timer. BDO_FR signals the application to take data from the receive FIFOs. The transmitted timestamp is usually set for a few isochronous cycles in the future.

The timestamp also keeps a constant time difference between the BDO_FR signal of the receiving DVLynx and BDI_FR signal of the transmitting DVLynx. BDO_FR signals when the application should read data from the DVLynx FIFO. BDI_FR is input into DVLynx and signals the start of a frame. The time between these signals needs to remain constant to reduce any effects of jitter.

The 4 most significant bits of the timestamp designate how long the offset is in terms of the number of isochronous cycles. The 12 least significant bits designate how far the offset is into an isochronous cycle. The smallest offset is 1 isochronous cycle (4 most significant bits= 0001b) or 125 μs. The largest offset is 15 isochronous cycles (4 most significant bits = 1111b) or 1.875 ms.

The transmit timestamp offset is added to the cycle timer value of a packet being transmitted. When the packet is received, the timestamp is compared to the cycle timer of the receiving node.

The receive timestamp offset can be added to the timestamp of a received packet. This offset determines when the received packet is released to the application.

The reason there are two timestamp offsets is that a designer may only have access to one node. For a transmitting node, the transmit offset should be set to counter the effects of cable length and jitter over 1394. A receiving node should set the received offset to counter the effects or delays of the application.

3.5.1 Time Stamp Encoding/Decoding for DV Transmit and Receive

At the beginning of each frame that is being transmitted, the timestamp is inserted into the SYT field of DCIPX1 header. The range of timestamp value is limited to the SYT field size, 0–FBFFh. See Table 3–6 for ranges. When no timestamp information is present, the SYT field is filled with FFFFhs. On reception, the SYT field is decoded. If any value other than FFFFh is decoded, a timestamp is extracted.

Table 3–6. Time Stamp Field of Source Packet

SYT (BINARY)			DESCRIPTION
HIGH 4 BITS		LOWER 12 BITS	
0000 . . 1111	and	0000 0000 0000 . . 1011 1111 1111	Timestamp
1111	and	1111 1111 1111	No information
Other values			Reserved

3.5.2 Time Stamp Calculation on Transmit

The timestamp is calculated by adding an offset to the value of the cycle timer register. This offset is the XTO (transmit timestamp offset) register. The 16 bit timestamp value is placed in the SYT field of the CIP header. The least significant 12 bits after the addition of cycle timer register and XTO register are low add. The four most significant bits after the addition are high add.

The cycle timer register is made up of the cycle count (4 most significant bits) and the cycle offset (12 least significant bits). The cycle-offset portion of the cycle timer register is modulo 3072. Each time this counter wraps around it signals the beginning of a new isochronous cycle. For a cycle master device, a cyclestart packet is transmitted at the beginning of each new isochronous cycle. For a non-cycle master device, a cyclestart is decoded from a received cycle start packet.

High add specifies the offset in number of isochronous cycles, and low add specifies an offset into an isochronous cycle. If the computation results in a low add, which is less than 3072 (125 μs), then the resultant timestamp is simply high add and low add. If the computation results in a LowAdd, which is equal to or greater than 3072, then the resultant timestamp is high add + 1 and the difference between the computed low add and 3072.

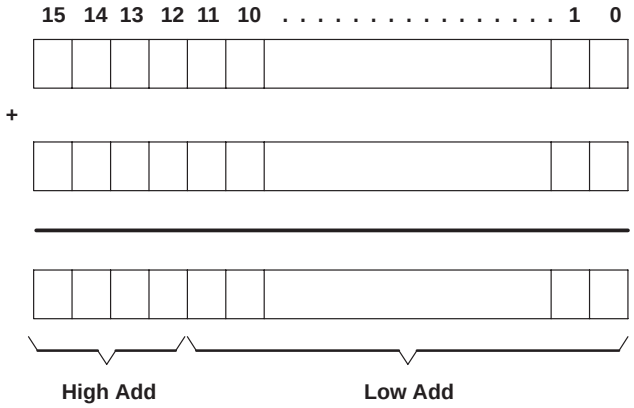


Figure 3–26. Determination of High Add and Low Add

If low add <3072, then the timestamp is simply high add and low add:

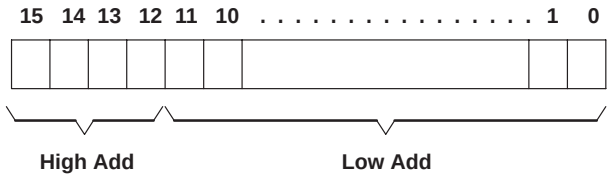


Figure 3–27. Time Stamp Value for LowAdd <3072

If low add is equal to or greater than 3072, then the resultant timestamp is high add + 1 and the difference between the computed low add and 3072.

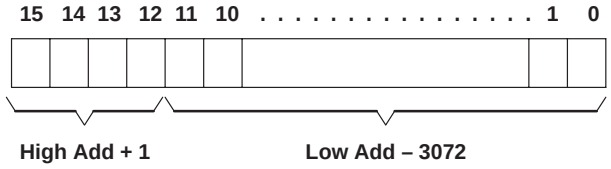


Figure 3–28. Time Stamp Value for LowAdd ≥ 3072

3.5.3 Timestamp Determination on Receive

The SyncTime is extracted from the SYT field of the CIP headers of the packet containing the timestamp. An additional offset from the receive SyncTime (RTO) located in a CFR register of the receiving node is added to the SyncTime value. Figure 3–29 shows how the timestamp is computed on receive. LowAdd is computed by adding as shown in the figure. High add is computed by adding also. The resuling timestamp is the concatenation of low add and high add. The resulting time computation is used to signal the reception of a frame at regular intervals (BDO_FR).

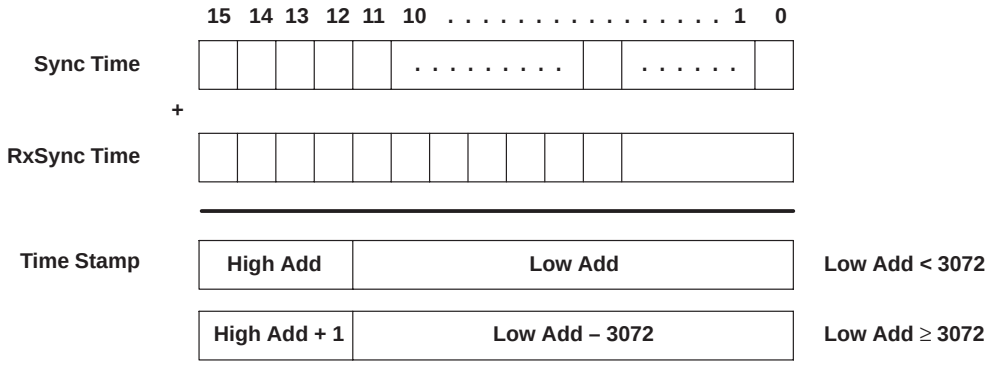


Figure 3–29. Time Stamp Determination on Receive

3.6 Asynchronous Transmit Data Formats (Host Bus to TSB12LV42)

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.

3.6.1 Quadlet Transmit

The quadlet-transmit format is shown in Figure 3–30 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 TSB12LV42 uses information in the header quadlets to form the IEEE 1394 headers for transmit.

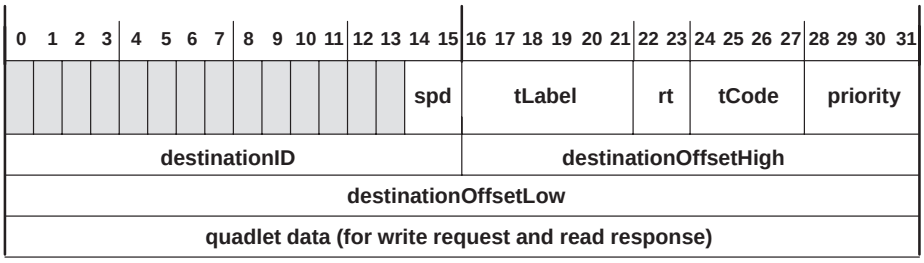


Figure 3–30. Quadlet-Transmit Format

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.

3.6.2 Block Transmit

The block-transmit format is shown in Figure 3–31 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 contain 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.

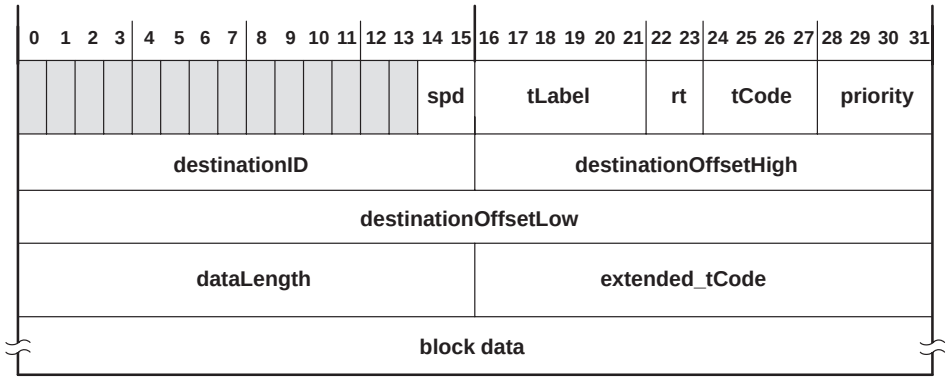


Figure 3–31. Block-Transmit Format

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.

3.6.3 Quadlet Receive

The quadlet-receive format is shown in Figure 3–32 and Figure 3–33 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 TSB12LV42). For packets received to the bulky data FIFO, the packet trailer is included as the first quadlet.

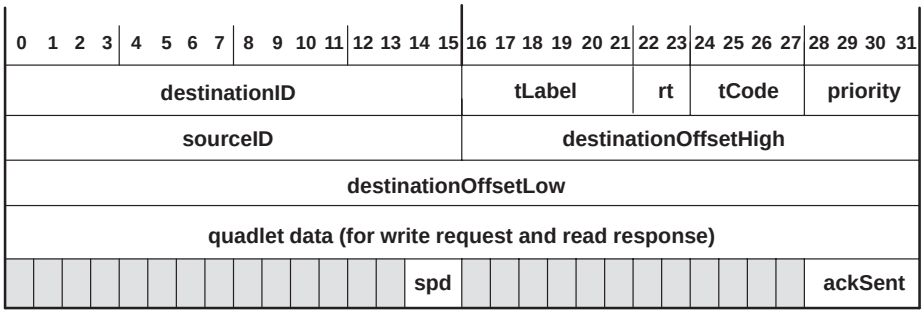


Figure 3–32. Quadlet-Receive Format for Control FIFO

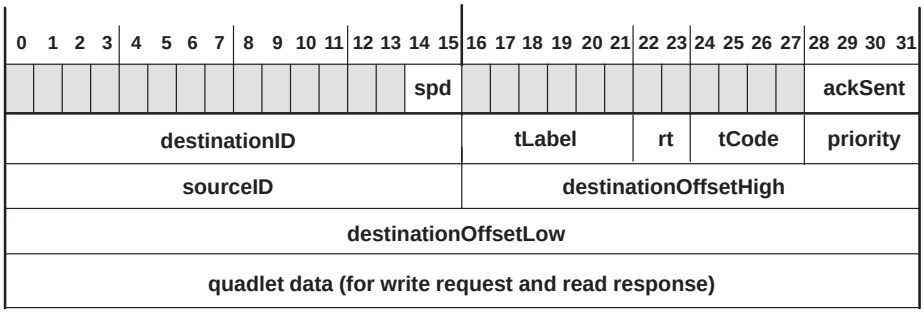


Figure 3–33. Quadlet-Receive Format for Bulky Data FIFO

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).

3.6.4 Block Receive

The block-receive format is shown in Figures 3–34 and 3–35 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.

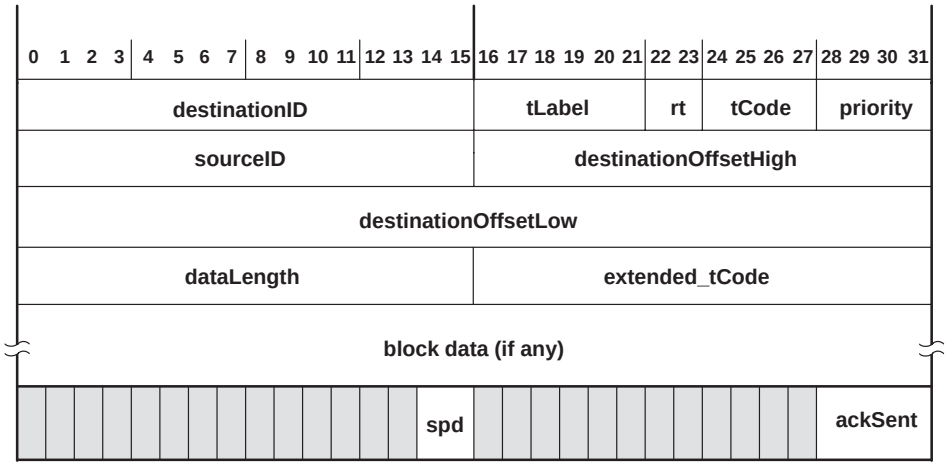


Figure 3–34. Block-Receive Format for Control FIFO

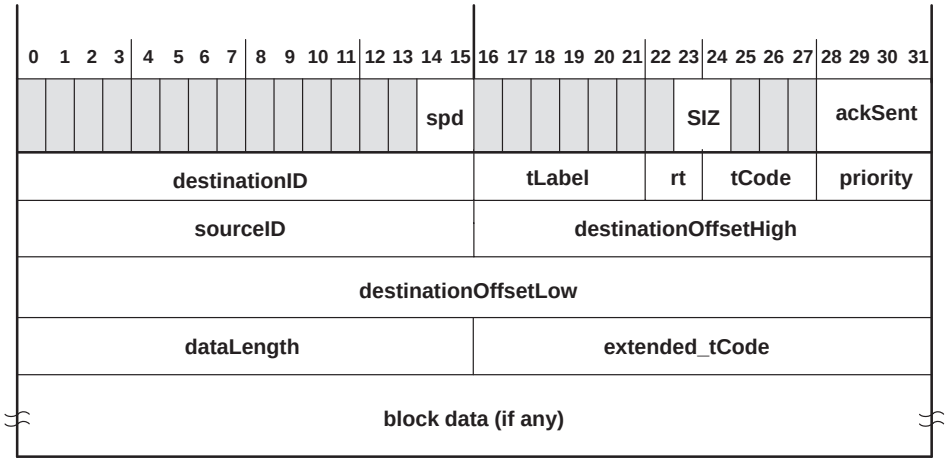


Figure 3–35. Block-Receive Format for Bulky Data FIFO

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.

3.7 Isochronous Transmit and Receive (Host Bus to TSB12LV42) Data Formats

3.7.1 Isochronous Transmit

The format of the isochronous-transmit packet is shown in Figure 3–36 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 TSB12LV42 information to build the 1394 isochronous header for transmit.

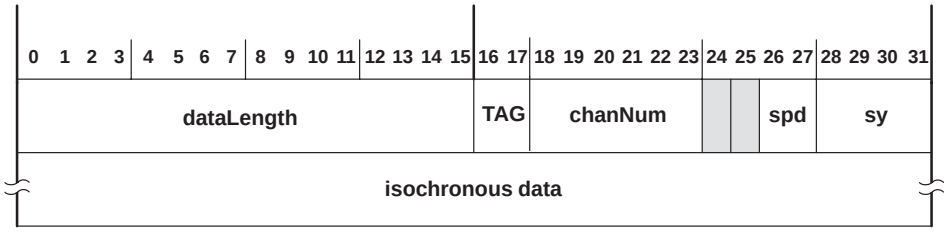


Figure 3–36. Isochronous-Transmit Format

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.

3.7.2 Isochronous Receive Data Formats

The format of the iscohronous-receive data is shown in Figure 3–37 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.

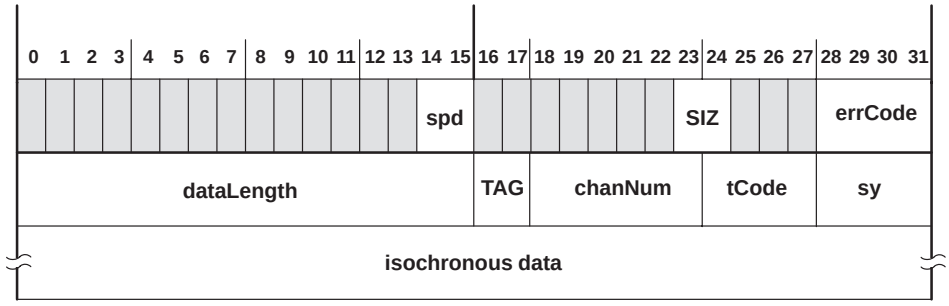


Figure 3–37. Isochronous-Receive Format for Bulky Data FIFO

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.

3.8 Snoop

The format of the snoop data is shown in Figure 3–38 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.

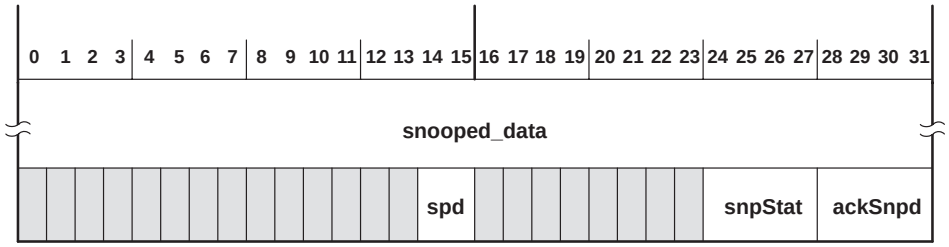


Figure 3–38. Snoop Format

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.

3.9 CycleMark

The format of the CycleMark data is shown in Figure 3–39 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.

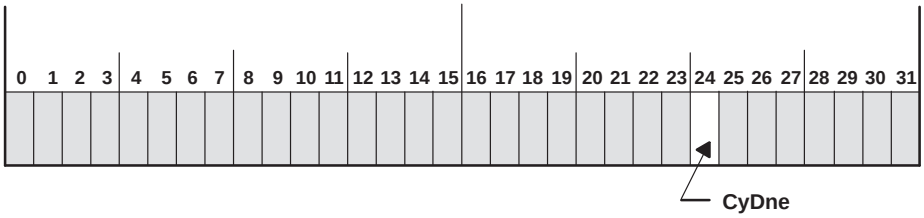


Figure 3–39. CycleMark Format

Table 3–14. CycleMark Function

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

3.10 Phy Configuration

The format of the Phy configuration packet is shown in Figure 3–40 and is described in Table 3–15. The Phy configuration packet transmit contains two quadlets. The first quadlet tells the TSB12LV42 that this quadlet is the Phy configuration packet. The Eh is then replaced with 0h before the packet is transmitted to the Phy interface.

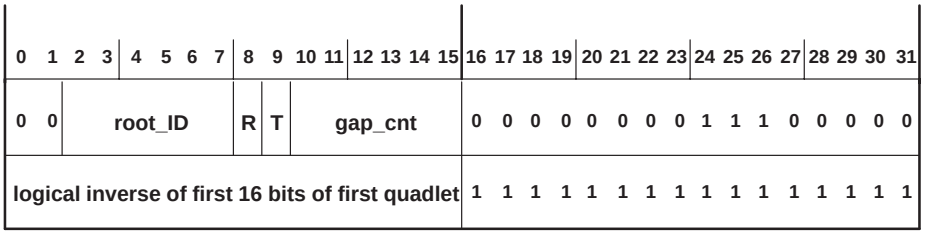


Figure 3–40. Phy Configuration Format

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.

3.11 Receive Self-ID Packet

The format of the receive self-ID packet is shown in Figure 3–41 and Figure 3–42 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.

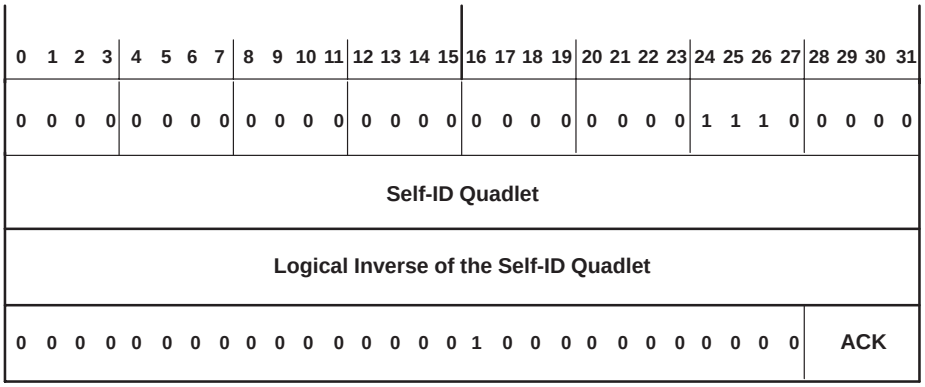


Figure 3–41. Receive Self-ID Format for Broadcast Write Receive FIFO

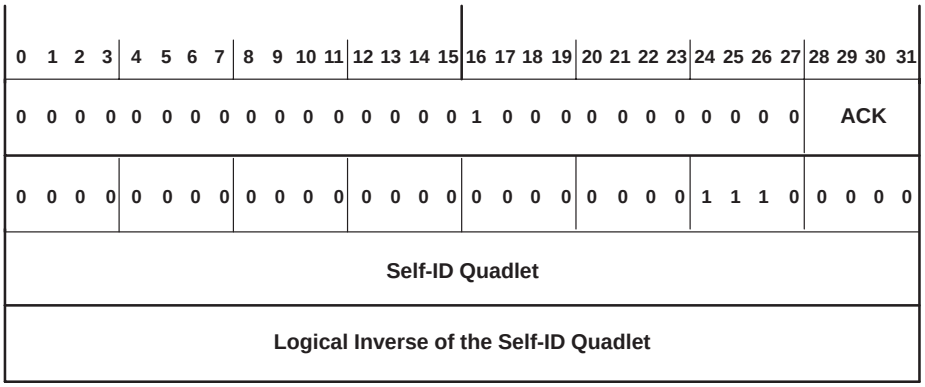


Figure 3–42. Receive Self-ID Format for Bulky Asynchronous Receive FIFO

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 200 Mbps 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 3–43 and Figure 3–44, 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 3–43 and Figure 3–44 show the formats of the cable Phy self-ID packets.

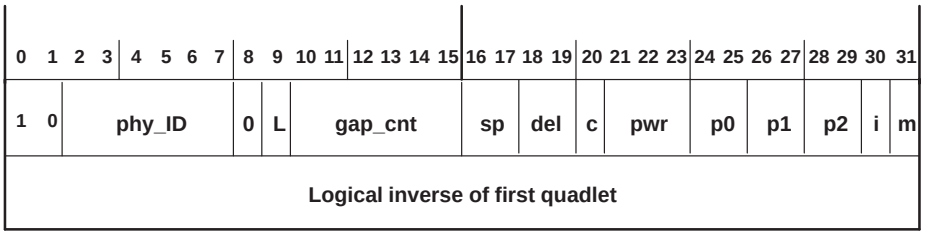
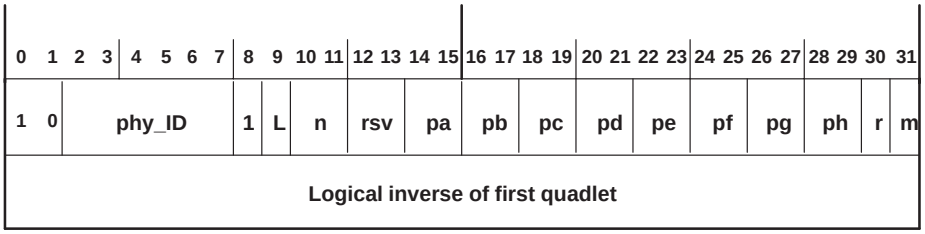


Figure 3–43. Phy Self-ID Packet #0 Format



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 3–44. 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).	
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

† 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	
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.

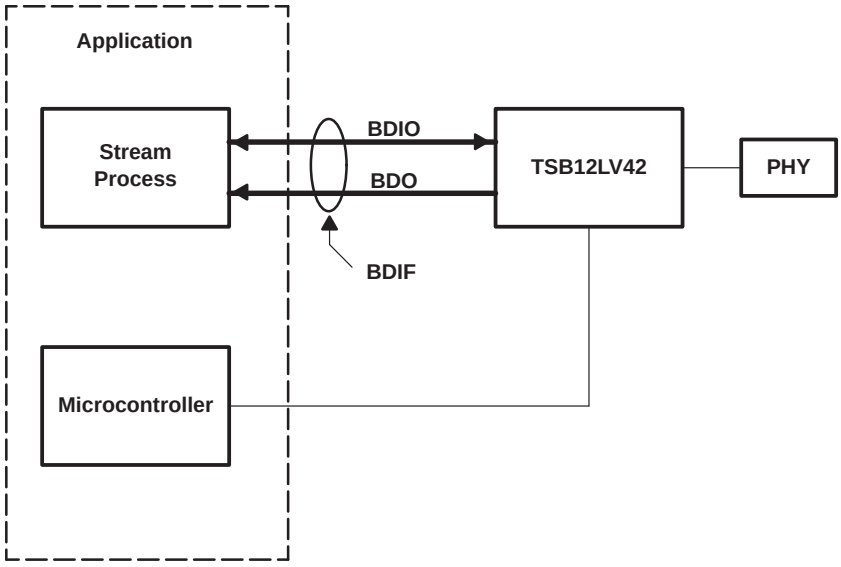
‡ The LLC and higher layers are enabled by the Link-On Phy packet.

4 External Interfaces

4.1 Bulky Data Interface

The bulky data interface or BDIF is a pair of ports supported by the TSB12LV42 Link-Layer controller. The BDIF is the physical medium by which autonomous streams of different types are piped to an application that uses the TSB12LV42.

A system diagram is shown below:



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:

1. Asynchronous
2. Isochronous
3. DV (a special Isochronous type)

A format bus bound to the port identifies these stream types. The encoding 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.

A more detailed look at the BDI's signaling is in order. Even though the two ports (BDIO and BDO) have some control signal and clock dependencies, we first look at them separately.

BDIO Port:

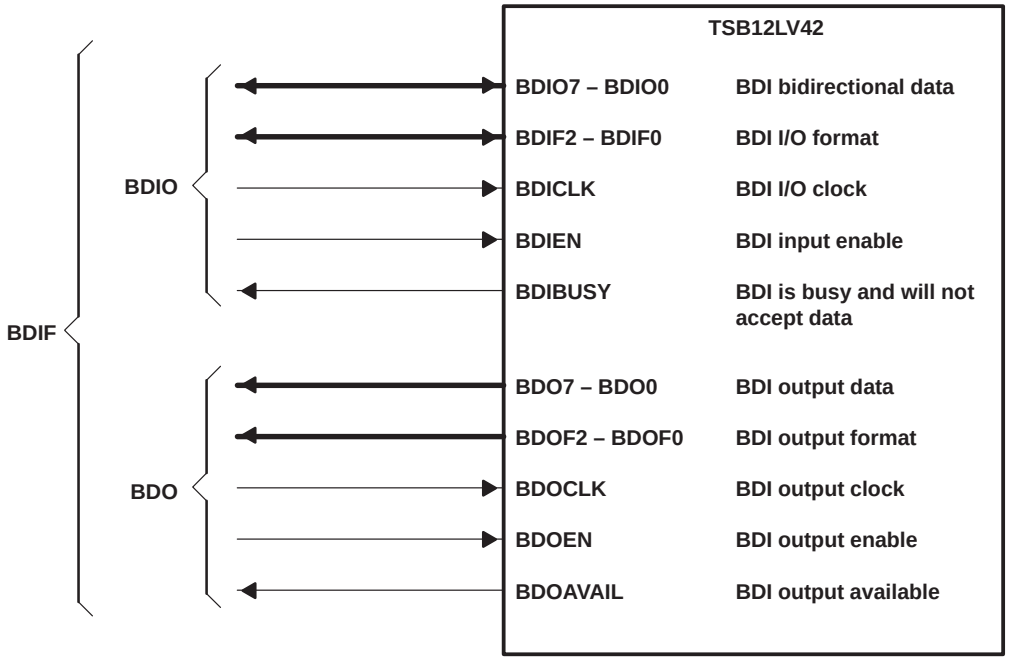


Figure 4–1. Bulky Data Interface

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

BDIO SIGNAL NAME	DRIVER TYPE	DESCRIPTION
BDIO[7:0]	I/O	Bidirectional BDI port (can be configured for input only). BDIO[7] = MSB and BDIO[0] = LSB.
BDIF[2:0]	I/O	Bidirectional BDI Format (can be configured for input only. 000 Reserved 001 A byte of an DV cell 010 A byte of an unformatted Isochronous packet 011 A byte of an Asynchronous packet 100 Idle 101 First byte of an DV cell 110 Last byte of an unformatted Isochronous packet 111 Last byte of an Asynchronous packet
BDICLK	I	BDIO data input clock
BDIEN	I	BDI Enable: Qualifies data for writes, data on BDIO, Format on BDIF. Read/Write* enable when in bidirectional mode.
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.

BDO Port:

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. BDO[7] = MSB and BDO[0] = LSB.
BDOF[2:0]	O	Unidirectional BDO Format 000 Reserved 001 A byte of an DV cell 010 A byte of an unformatted Isochronous packet 011 A byte of an Asynchronous packet 100 Idle 101 First byte of an DV cell 110 Last byte of an unformatted Isochronous packet 111 Last byte of an Asynchronous packet
BDOCLK	I	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	Signal data is available on BDO and also BDIO for reads.

4.1.1 BDIF Control Register (D8h) Configuration

The BDIF is programmed by writes to the BDIF control register. This register is located at offset D8h in the TSB12LV42 microcontroller address space. The register format and bit definitions are shown below.

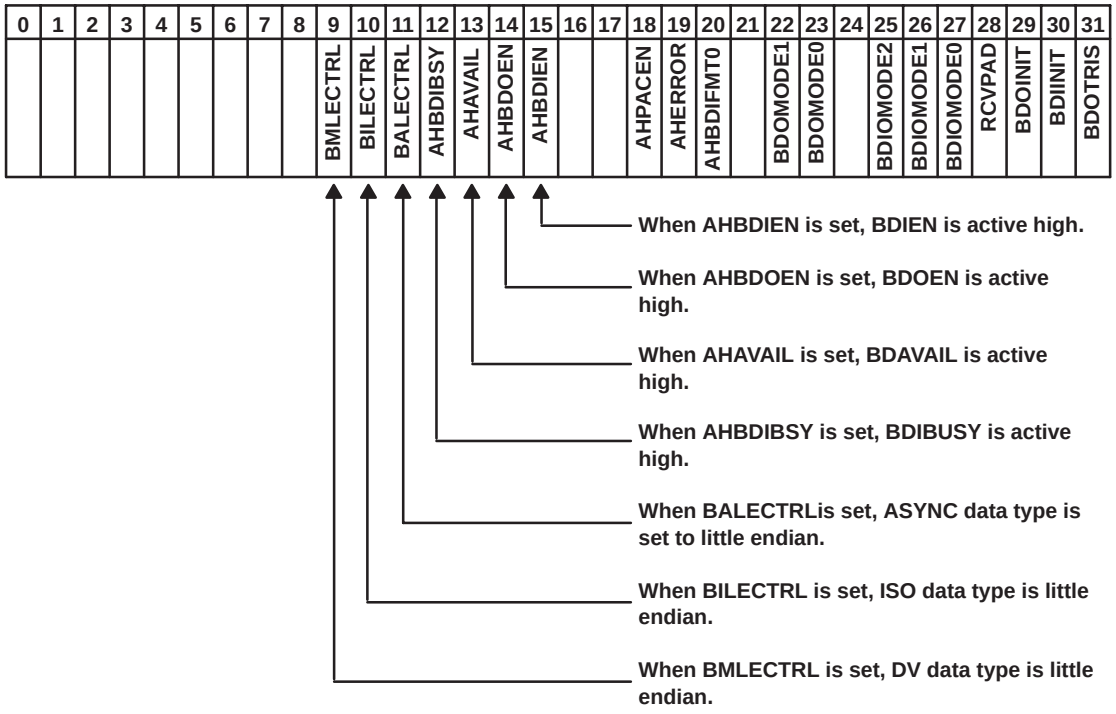


Figure 4–2. BDIF Control Register

4.1.1 BDIF Control Register (D8h) Configuration (Continued)

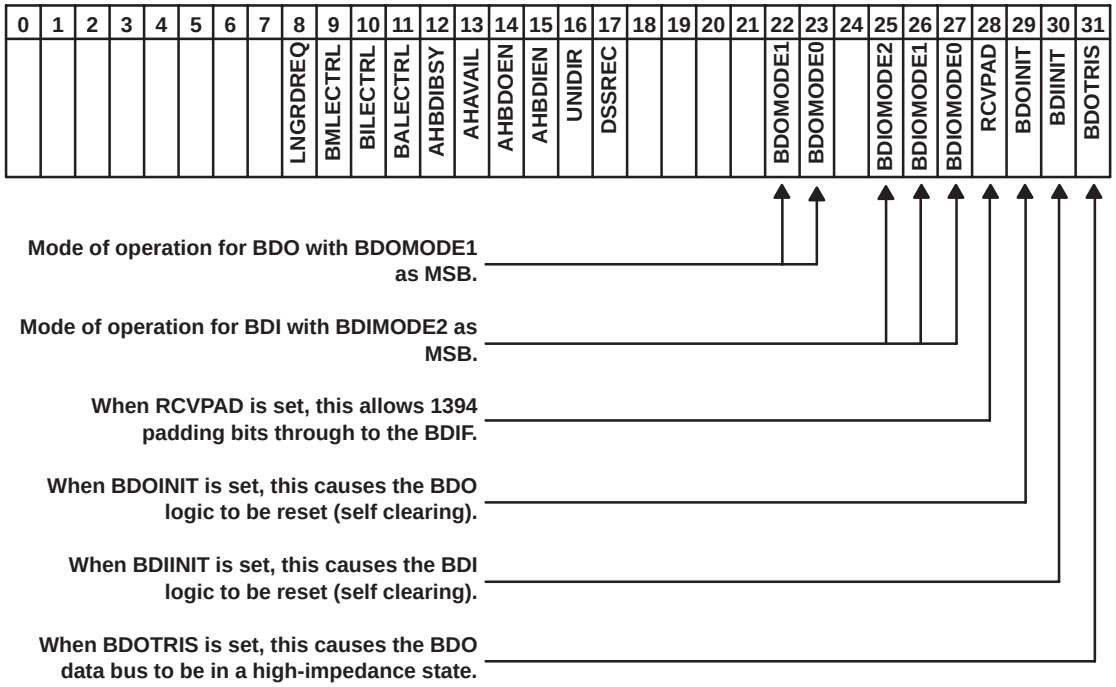


Figure 4–2. BDIF Control Register (Continued)

4.1.2 Modes of the Bulky Data Interface (BDIF)

The BDIF has four valid modes of operation. These modes are selected using the BDIMODE and BDIOMODE fields of the BDI control register. The Table 4–1 shows the basic features of each mode.

Table 4–1. Modes of the BDIF

MODE	A	B	C	D
BDIMODE	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/sec (max)	20 Write 20 Read	20 Write 20 Read	10 Write 10 Read	20.25
Control Signal Use:				
BDIEN	√	√	√	√
BDIBUSY	√	√		√
BDOEN	√		√	√
BDOAVAIL	√	√	√	√

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

4.1.3 Mode A – 8 Bit Parallel I/O

BDI MODE: A	8 Bit parallel input	8 Bit parallel output
	BDIOMODE = 000	BDOMODE = 00

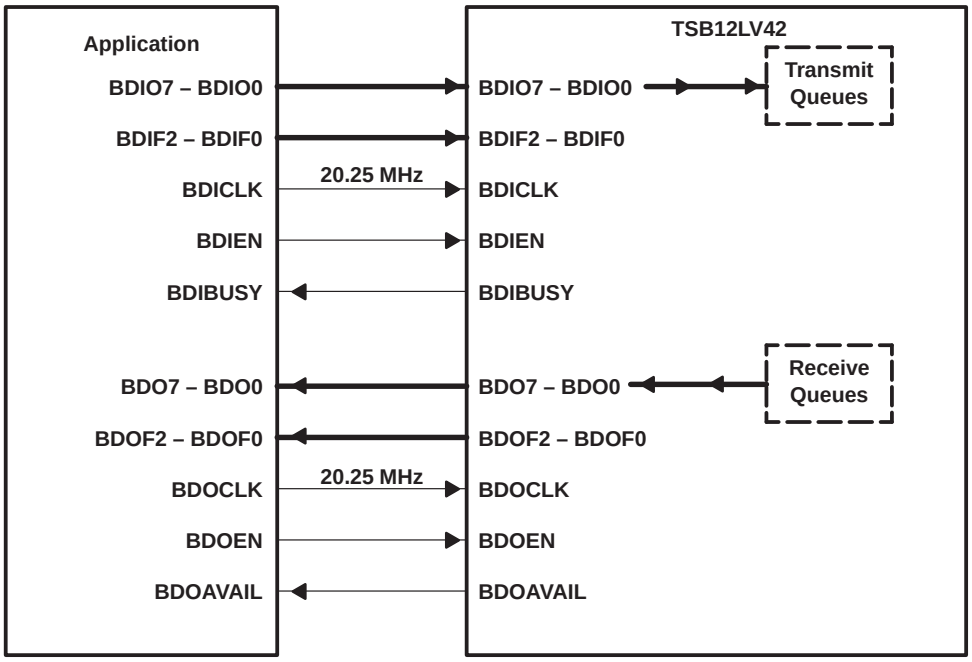


Figure 4–3. Bulky Data Interface Mode A Typical Application

In this mode, the BDIO bus is input only. The BDO bus is output only. The BDIF operates in full duplex mode. It can receive data at the BDIO port and transmit data from the BDO port simultaneously.

Both the input and output buses operate synchronously to the BDICLK and BDOCLK. The maximum throughput of both the BDIO and BDO ports is 20 Mbytes/sec. If the BDIF expects new data every clock cycle, then the data input clock (BDICLK) and data output (BDOCLK) clock run up to 20.25 MHz. The BDICLK/BDOCLK can be operated up to 40.5 MHz if data is presented at the BDIF every other clock cycle.

BDIEN qualifies data on the BDIO port for writes. BDIBUSY signals a busy condition to the application on BDIO for writes. When BDIBUSY is activated, the TSB12LV42 does not accept any more data from the application.

BDOEN qualifies data on BDO port for reads. BDOAVAIL signals the application that data is available on BDO port for reads.

4.1.4 Mode B – 8-Bit Parallel I/O with No Read Control

BDI MODE: B	8 Bit parallel input	8 Bit parallel output with no read control
	BDIOMODE = 000	BDOMODE = 01

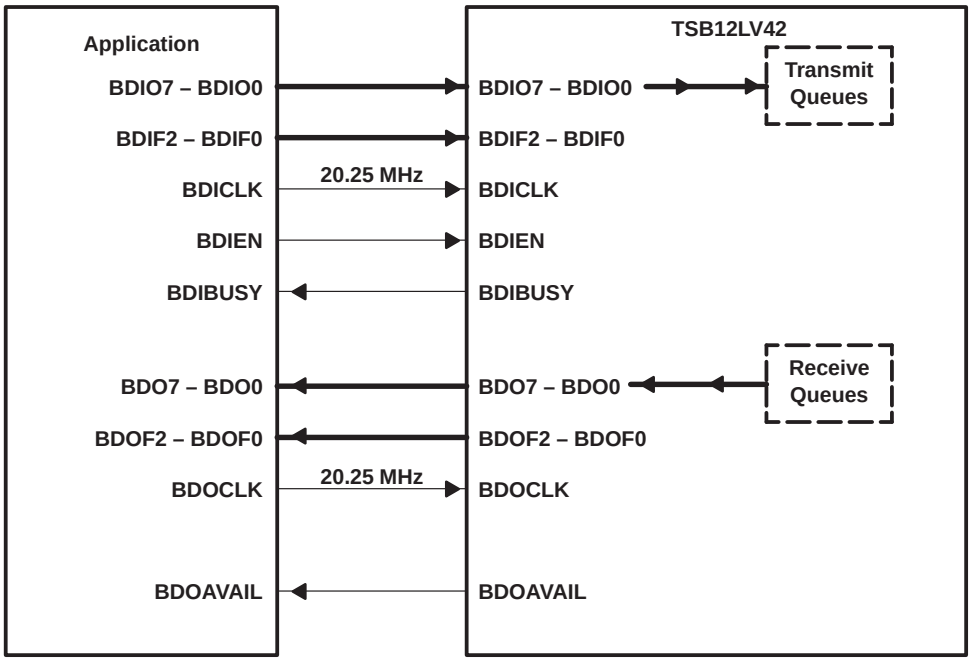


Figure 4–4. Bulky Data Interface Mode B Typical Application

The data input and output for this mode are similar to the bulky data Mode A. The BDIO port is input only. The BDO port is output only. The clock speeds, data rates, and full duplex capability are similar to Mode A.

The difference between this mode and Mode A is the absence of BDOEN, the bulky data output enable, for read operations. Since Mode B does not use BDOEN to read data out of the TSB12LV42FIFOs, data is continuously output to the host whether or not it can accept it. The main advantage of this mode is that no signal is required by the host for receive. However, if the host FIFO can not handle the data output from TSB12LV42, then data may be lost.

4.1.5 Mode C – 8 Bit Parallel Asynchronous Input/ 8 Bit Parallel Asynchronous Output

BDI MODE: C	8 Bit parallel input (Asynchronous)	8 Bit parallel output (Asynchronous)
	BDIOMODE = 101	BDOMODE = 11

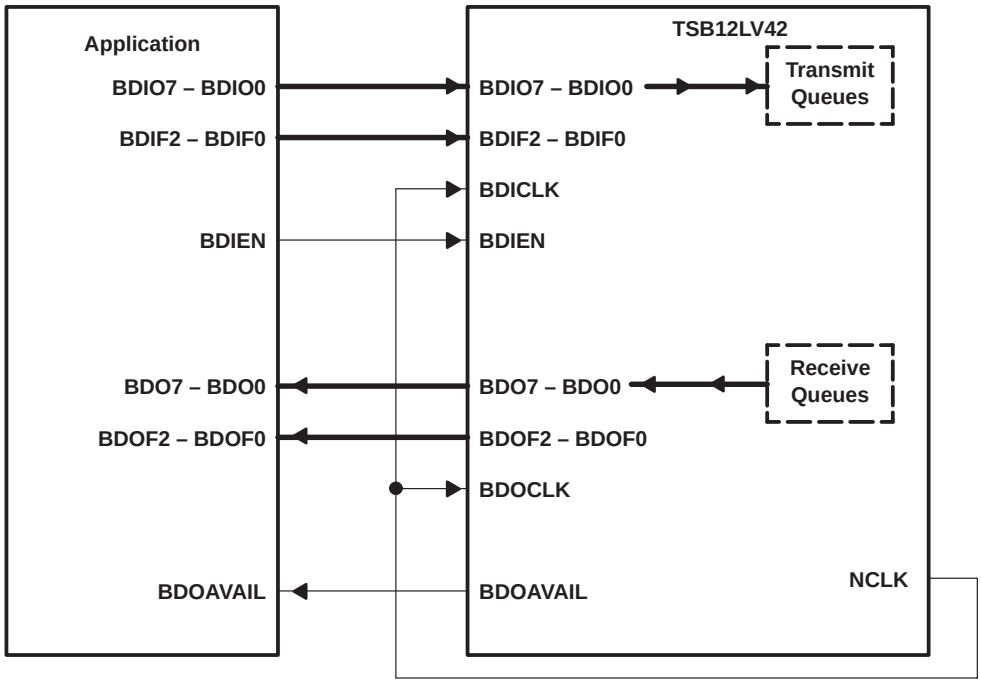


Figure 4–5. Bulky Data Interface Mode C Typical Application

This mode provides two data buses: the BDIO for transmit and BDO for receive. The data at both the BDIO and BDO ports in this mode is resynchronized internally with the clock provided at the BDICLK/BDOCLK pins. This clock can be either NCLK (supplied by TSB12LV42) or an external clock. This mode operates in full duplex.

NCLK can be used to sample both the BDIO and BDO ports. This clock rate is 24.576 MHz, or SYSCLK/2. NCLK is available at STAT0 or STAT3 and programmable at register 30h. For correct operation, NCLK must be physically attached to the BDICLK or BDOCLK pin. The BDICLK or BDOCLK can also be driven with an external clock. The maximum frequency of this external clock is 40 MHz.

This mode has a maximum data throughput capability of 10 Mbyte/s for a write and 10 Mbyte/s for a read. Read and write operations can occur every fourth NCLK clock cycle. There must be at least one inactive BDIEN/BDOEN cycle between read or write operations. The host is responsible for meeting this timing requirement.

There is no BDIBUSY signal available in this mode. This means that during a write operation, there is no way for the TSB12LV42 to signal to the application that it is busy and can not accept any more data.

NOTE:
Only DV data is supported in the asynchronous bulky data mode.

4.1.6 Mode D– 8 Bit Parallel Bidirectional Mode

BDI MODE: D	8 Bit parallel bidirectional	8 Bit parallel bidirectional
	BDIOMODE = 001	BDOMODE = 00

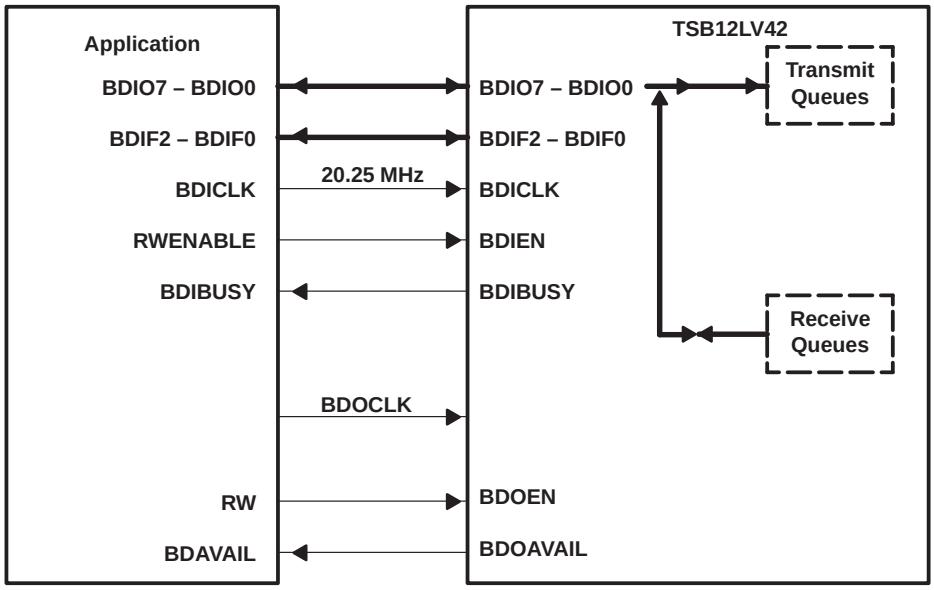


Figure 4–6. Bulky Data Interface Mode D Typical Application

This mode is the bulky data bidirectional mode. The device in this mode operates in half duplex. The read and write operations share the BDIO port.

In the bidirectional mode, BDIEN serves as the read/write enable on the BDIO port, BDOEN serves as Read/Write on the BDIO port. The BDIF[2–0] serves as the format bus for both the read and write operations.

For bidirectional mode, BDICLK and BDOCLK must have the same frequency. The maximum data throughput for the bidirectional mode is 20.25 Mbytes/sec. If the TSB12LV42 expects data on every clock cycle, then the maximum BDICLK/BDOCLK rate is 20.25 MHz. If the TSB12LV42 expects data on every other clock cycle, then the maximum BDICLK/BCOCLK rate is 40.5 MHz.

BDIBUSY signals the application when the TSB12LV42 BDIO port is busy and does not accept any more data during a write operation. BDOAVAIL signals the application when the TSB12LV42 BDIO port has data available for reading.

4.1.7 Bulky Data Interface Timing

4.1.7.1 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. The asynchronous mode (modes C) supports only DV data.

4.1.7.2 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 4–16, 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 TSB12LV42 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 DV data, as well as other consecutive bytes, to the TSB12LV42 FIFOs. The format bus can also represent asynchronous and isochronous packets. Please see Section 4.1.1, *BDIF Control Register (D8h) Configuration*, more detail. Please see Figure 4–17 for critical write timing in bulky data modes A and B.

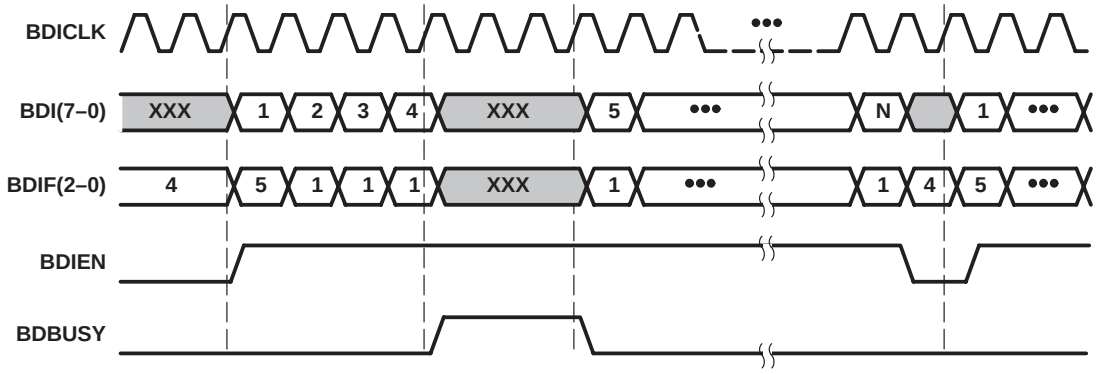
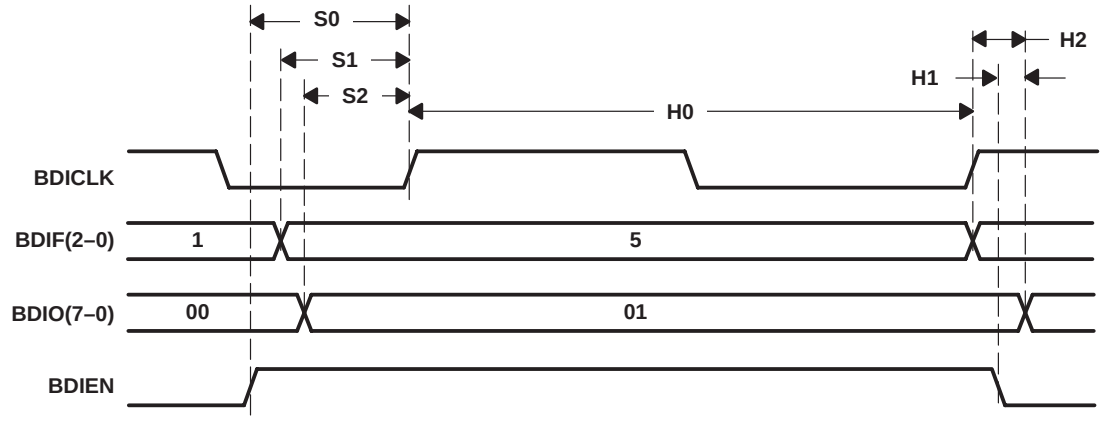


Figure 4–7. Functional Timing for Write Operations in the Unidirectional Modes



NAME	MIN (ns)	MAX (ns)	DESCRIPTION
H0	50		Clock period, 50% duty cycle
S0	10		Setup time for BDIEN relative to BDICLK
S1	10		Setup time for format bus (BDIF) relative to BDICLK
S2	10		Setup time for data bus (BDIO) relative to BDICLK
H1	1		Hold time for BDIEN deassert
H2	1		Hold time for data bus (BDIO) relative to clock edge

Figure 4–8. Critical Timing for Write Operations in Unidirectional Mode

4.1.7.3 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 one-fourth NCLK in Figure 4–9). Since the asynchronous mode only supports DV/DSS data, only the BDIF2 format signal is necessary to indicate the beginning or continuing byte of a DV cell. Please see Figure 4–9 for asynchronous mode functional timing.

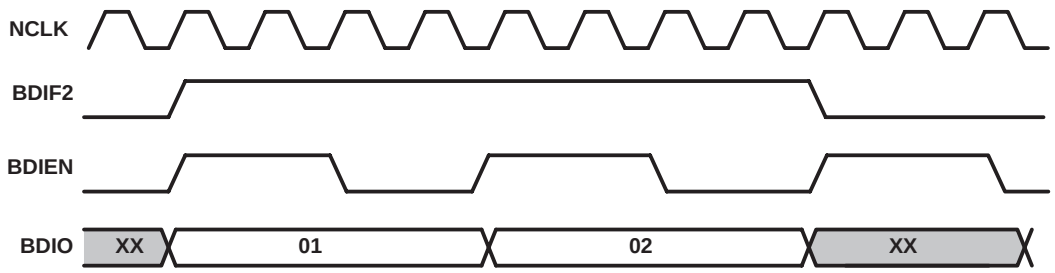
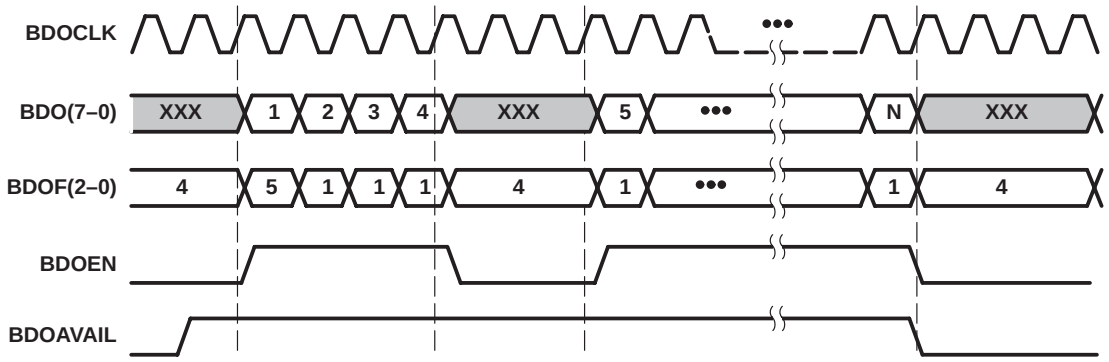


Figure 4–9. Functional Timing for Write Operations in the Asynchronous Mode

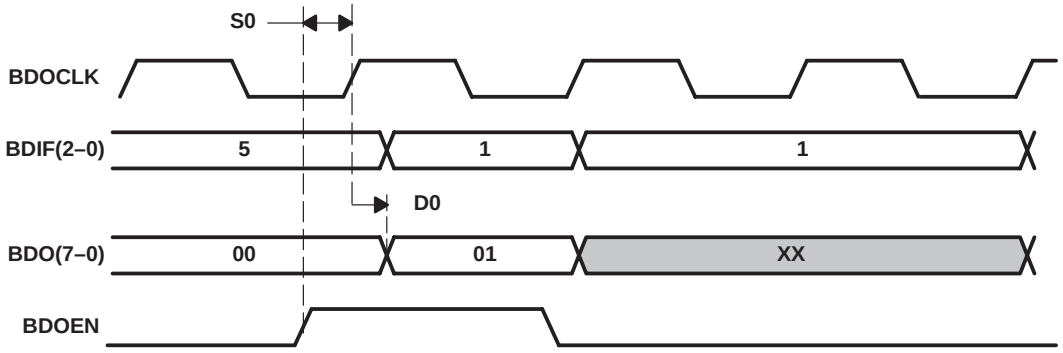
4.1.7.4 Unidirectional Read Timing

In the unidirectional modes (modes A and B), data is read out of the TSB12LV42 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, *BDIF Control Register (D8h) Configuration*. As shown in Figure 4–10, BDOEN signals the TSB12LV42 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 TSB12LV42 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 4–11 for critical read timing in bulky data Modes A and B.



NOTE A: BDOEN is not necessary in Mode B.

Figure 4–10. Functional Timing for Read Operations in Unidirectional Mode



NAME	MIN (ns)	MAX (ns)	DESCRIPTION
S0	10		Setup time for BDOEN relative to rising clock edge
D0	7		Delay time for data and format bus valid relative to clock edge

NOTES: A. BDOEN is not necessary in Mode B.
B. All MIN and MAX in nanoseconds

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

4.1.7.5 Asynchronous Read Timing

In the asynchronous mode (mode C), data is read out of the TSB12LV42 bulky data interface on the BDIO[7–0] pins. 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. Data can be read on the following BDOEN rising clock edge. See Figure 4–12 for asynchronous read functional timing.

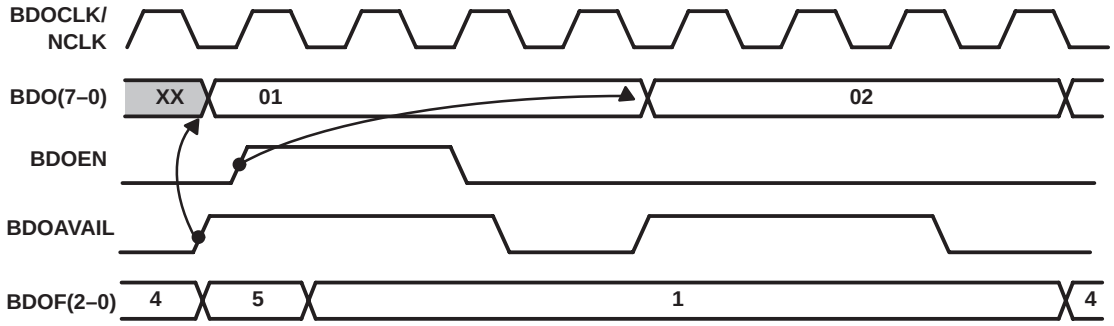


Figure 4–12. Functional Timing for Read Operations in Asynchronous Mode

4.1.8 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.

4.1.8.1 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 4–13 and 4–14.

In bidirectional mode, the BDIF[2–0] signals are the format bus. This signals the start of a DV packet (binary value of 101), a byte of an DV cell (binary value of 001), or an idle (binary value of 100). There are also format codes for isochronous and asynchronous data. See Section 4.1, *Bulky Data Interface*, 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 TSB12VL42 FIFO is full and can not accept any more data.

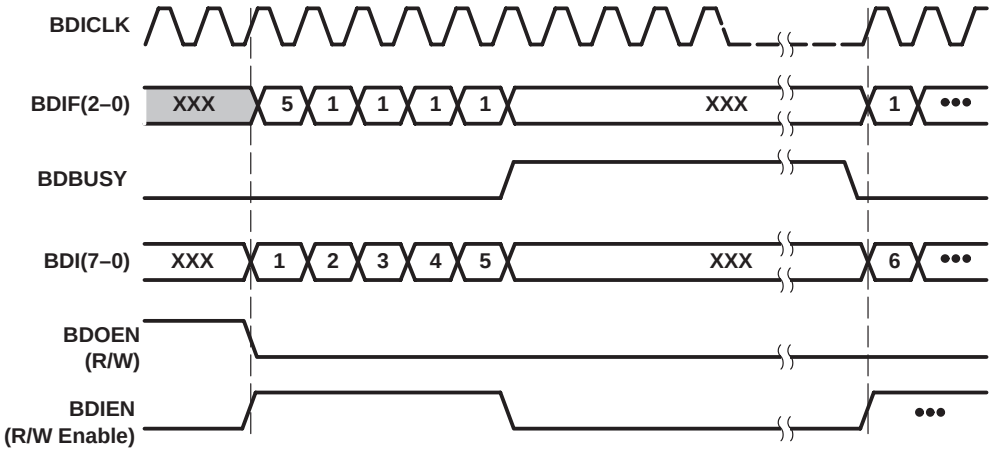
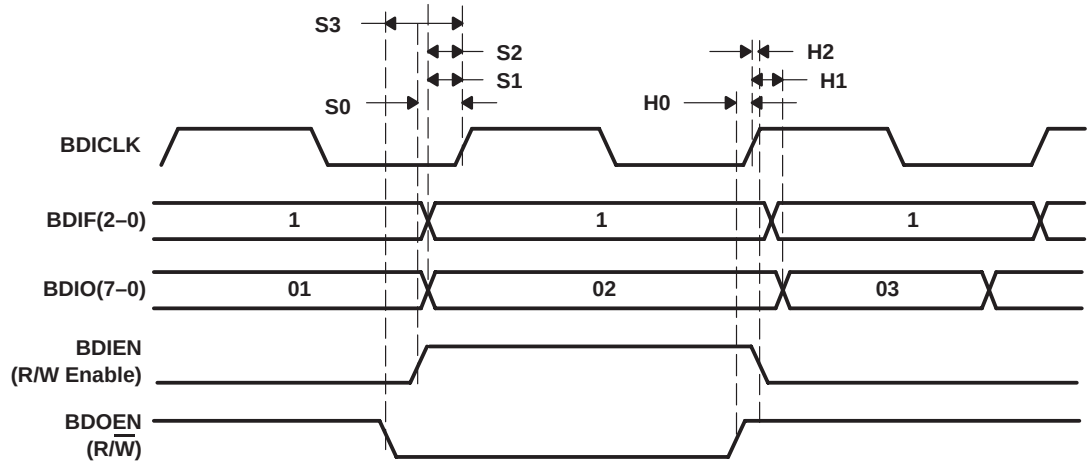


Figure 4–13. Functional Timing for Write Operations in Bidirectional Mode



NAME	MIN (ns)	MAX (ns)	DESCRIPTION
S0	10		Setup time between BDIEN (RW enable) and rising edge of clock
S1	10		Setup time between format bus (BDIF) and rising edge of clock
S2	10		Setup time between data bus (BDIO) and rising edge of clock
S3	15		Setup time for BDOEN relative to rising edge of clock
H0		0	Hold time for BDOEN after rising edge of clock
H1	1		Hold time for data bus (BDIO) to rising edge of clock
H2	1		Hold time for BDIEN after rising edge of clock

NOTE A: All MIN and MAX in nanoseconds

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

4.1.8.2 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 4–15 and 4–16 for bidirectional read timing.

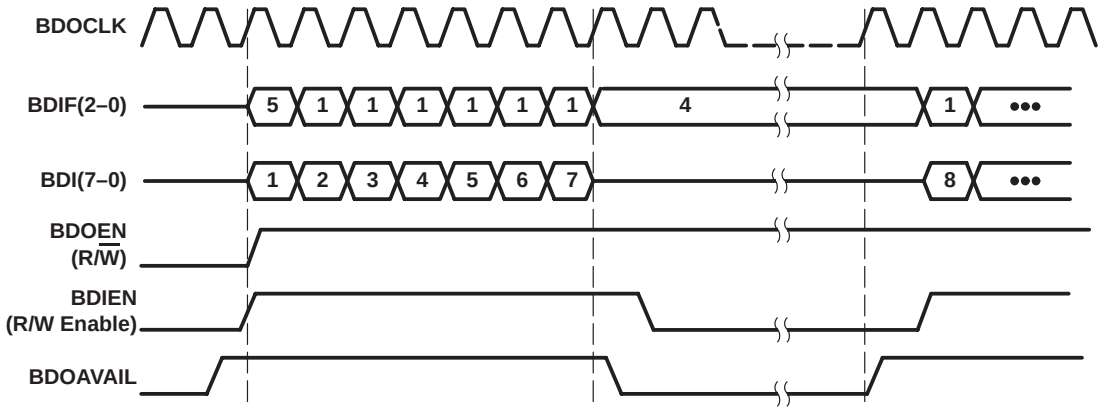
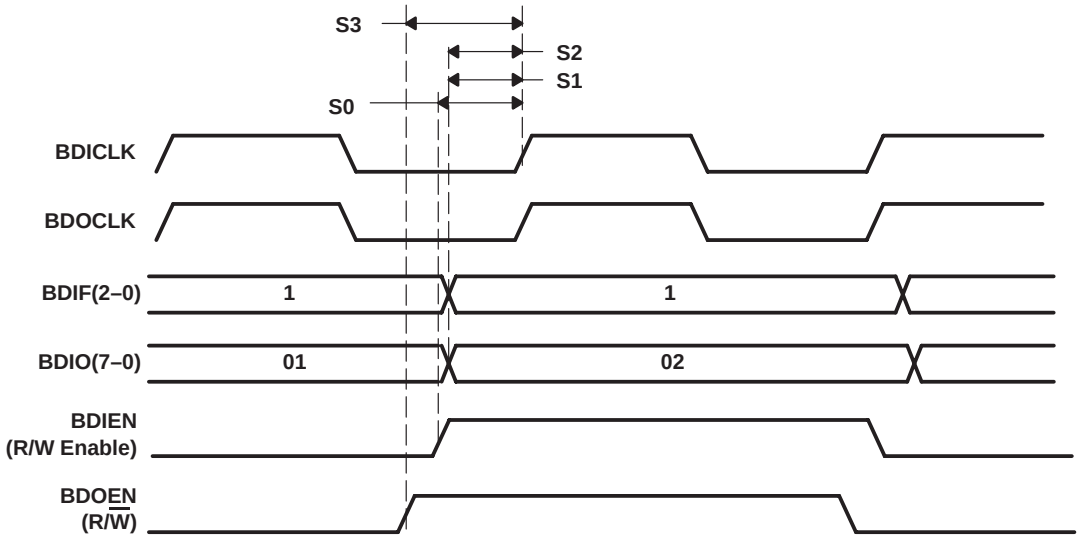


Figure 4–15. Functional Timing for Read Operations in Bidirectional Mode



NAME	MIN (ns)	MAX (ns)	DESCRIPTION
S0	10		Setup time between BDIEN (RW enable) and rising edge of clock
S1	10		Setup time between format bus (BDIF) and rising edge of clock
S2	10		Setup time between data bus (BDIO) and rising edge of clock
S3	13		Setup time for BDOEN relative to rising edge of clock

NOTE A: All MIN and MAX in nanoseconds

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

4.2 Microprocessor Interface

4.2.1 Microprocessors Supported

The TSB12LV42'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 detect which kind of microprocessor/microcontroller (MP/MC) is connected to TSB12LV42, two MP/MC select lines, MCSEL1 and MCSEL0, are driven to certain logic levels. Table 4–2 shows the MCSEL settings for each type of processor.

Table 4–2. MCSEL Settings for Various Microprocessors

PROCESSOR SELECTED	MCSEL1	MCSEL0
Reserved	1	1
8051	1	0
68000	0	1
TMS320AV7100 ARM	0	0

After the type of MP/MC has been determined, all the I/O control pins on the MP/MC Interface are mapped according to which type of MP/MC is present. The following matrix table (Table 4–3) defines the actual pin functions for different MP/MCs.

Table 4–3. TSB12LV42 MP/MC Interface Pin Function Matrix

TSB12LV42 PIN	MP/MC Type		
Name	**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	–

† For Intel 8051, AD[7:0] is connected to TSB12LV42's DATA[8:15] as a bidirectional address/data bus. Intel 8051 port2's LS address bit A0 output is connected to TSB12LV42'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.

TSB12LV42's microprocessor interface is synchronized to the BCLK in TMS320AV7100 Mode. BCLK input is from TMS320AV7100's extension bus external clock input (CLK40, 40.5 MHz). For Motorola 68000 and Intel 8051 modes the TSB12LV42's microprocessor interface is asynchronous. The internal clock used for these two modes is SCLK, which is the 49.152 MHz clock from the PHY.

All bus signal labeling on the TSB12LV42's microprocessor interface is denoted as bit0 as MSB and bit 15 as LSB.

The data path from the microprocessor interface to the host is either 16 or 8 bits wide. However, all internal configuration registers are 32 bits wide. Thus the Microprocessor Interface of the TSB12LV42 must stack the incoming/outgoing write and read 32 bit data prior to delivery to either an internal CFR or the

microprocessor data bus. The byte swapping function required for different endianness settings is performed in the stacking buffer. Section 4.2.10, *Endianness*, describes how the byte swap works for the various endianness settings.

Figures 4–17, 4–18 and 4–19 show hook up diagrams for each type of processor supported.

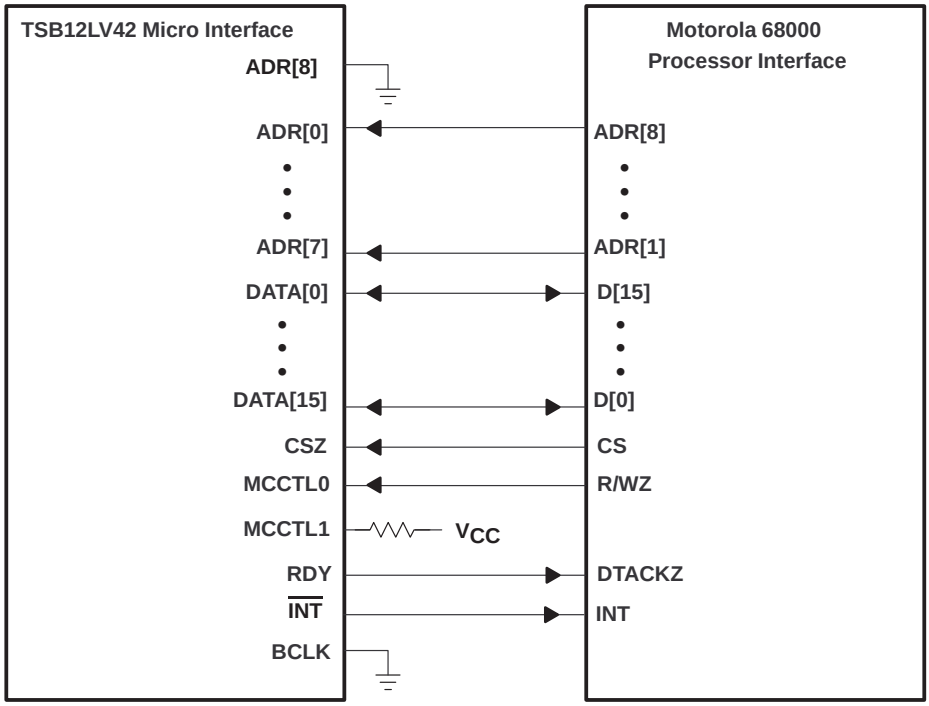


Figure 4–17. DVLynx Connections for 68000 Microcontroller

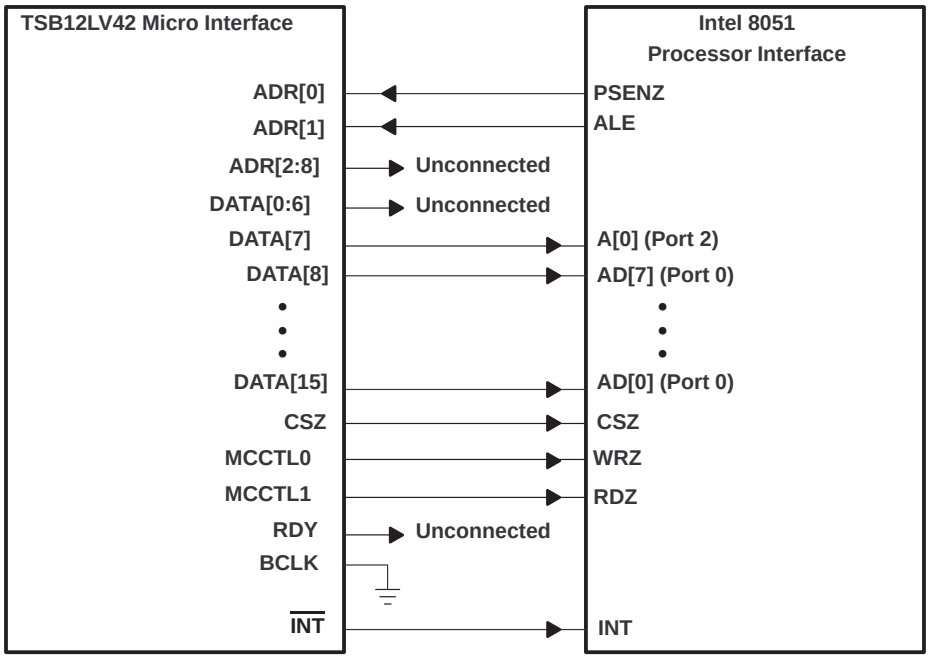
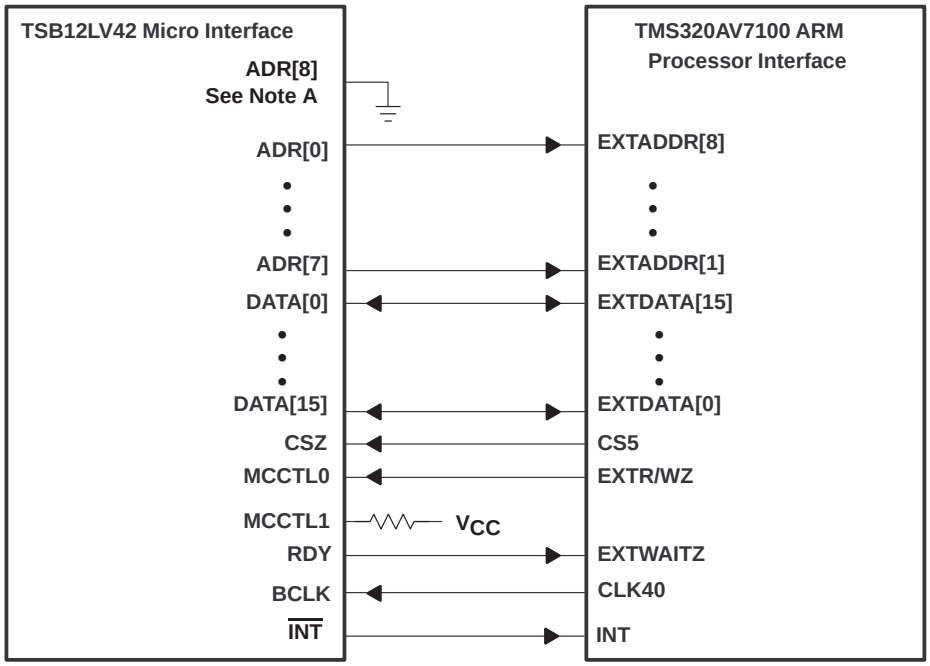


Figure 4–18. DVLynx Connections for 8051 Microcontroller



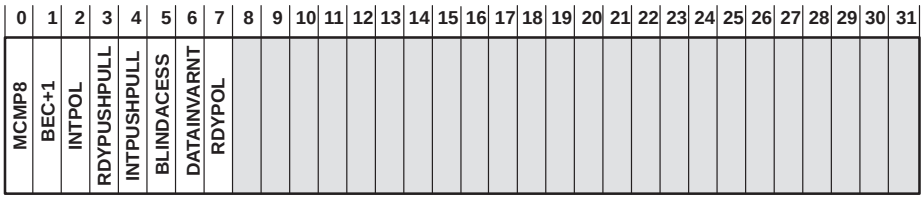
NOTE A: Connect as shown for 16-bit data. If 8-bit data bus is being used then connect ADR[8] on the TSB12LV42 to EXTADDR[0] on the TMS320AV7100.

Figure 4–19. DVLynx Connections for TMS320AV7100 ARM Processor

4.2.2 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 4–20 shows the bit map of the IOCR register. Table 4–4 shows a correlation table of the value and meaning of each bit, along with the power up default setting.

The TSB12LV42 supports both 8 bit and 16 bit data busses. It also supports both little endian and big endian microprocessors. The TSB12LV42 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 TSB12LV42 does not provide any on chip pull-up or pull-down resistor for those open-drain type outputs. The board level designer has to add it if necessary to achieve appropriate level control or sharing between devices



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

Figure 4–20. TSB12LV42 IOCR Register

Table 4–4. TSB12LV42 IOCR Bit/Function Correlation Table and Power–up Default Setting

BIT NO.	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	RdyPushPull	RDY output signal control	Active push/pull	3-state	3-state		N/A‡
4	IntPushPull	INT output signal control	Active push/pull	3-state	Active push/pull	3-state	
5	BlindAccess	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	DataInvarnt†	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 DataInvarnt bit setting has no effect.

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

Although the IOCR provides a way to setup the extension bus interface, not all settings are supported for each type of microprocessor. The following summarizes the special notes for each type of MP/MC 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) is deasserted. This ensures that the TSB12LV42'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 later writes. 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). All the other three writes could be loaded with all zeros (bits 8–31 of the IOCR register).

4.2.3 Handshake and Blind Access modes

The host microprocessor can access the TSB12LV42 in either handshake or non–handshake mode. The handshake signal between TSB12LV42 and the MP/MC is RDY. It 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, then 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. More detail on the blind access mode functionality is provided in Section 4.2.9, *Blind Access Specific Issues*. 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.

4.2.4 General Read Instructions

For the read case, the TSB12LV42'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. This occurs regardless of which byte position inside the quadlet the address accesses. For example, a read request from either address 0F3h or 0F2h returns the 32 bit value stored at address 0F0h (formatter control register). Then the microprocessor interface generates a cycle start (CSZ) to the internal logic, passes along the read address, and waits for a response. Once the internal response comes back, a cycle acknowledge (RDY) is generated to the microprocessor interface to indicate that the current read process is finished and the whole quadlet data is valid to be read from the data bus. In handshake mode, the TSB12LV42 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. For example: 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 TSB12LV42 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 TSB12LV42 microprocessor interface's 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 TSB12LV42 initiates a write cycle and any future read requests will initiate a new read cycle

4.2.5 General Write Instructions

For the write case, the TSB12LV42's 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 TSB12LV42 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 actually be started until the present write cycle is complete). 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 (preloaded) transaction if any.

The TSB12LV42 supports any byte/doublet order writes, as long as they are within the same quadlet boundary. If a user tries to do either one of the following before he fills up the complete quadlet, the current write is ignored and an 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 OCCURS!**
- 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 OCCURS!**

Summarizing the above 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 the microprocessor interface's 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 interface's byte stacking buffer.

4.2.6 TMS320AV7100 Mode Timing Diagrams

TSB12LV42 is designed to meet the read/write access timing defined by Texas Instruments' TMS320AV7100 extension bus interface. The figures below show critical and functional timing for read and write operations. This mode supports both handshake and blind access modes, thus timing diagrams are given for each.

NOTE:
The design of this device ensures the following timing parameters.

Table 4–5. TMS320AV7100 Critical Timing Characteristics

PARAMETER	MIN (ns)	MAX (ns)	DESCRIPTION
t_{su1}	12.2		Setup time, CSZ to BCLK rising edge
t_{su2}	12.1		Setup time, R/WZ to BCLK rising edge
t_{su3}	8.9		Setup time, address to BCLK rising edge
t_{su4}	1.2		Setup time, data to BCLK rising edge *
t_{h1}		0.8	Hold time, CSZ after BCLK rising edge
th2		1.0	Hold time, R/WZ after BCLK rising edge
th3		0.8	Hold time, address after BCLK rising edge
th4		0.9	Hold time, data after BCLK rising edge *
t_{d1}		17 BCLK cycles	WAITZ low ***
t_{d2}		2 BCLK cycles + 8 ns	CSZ falling edge to WAITZ
t_{d3}	2.5 BCLK cycles		CSZ rising edge to CSZ falling edge
t_{d4}		5.0	DATA valid after CSZ rising edge **
t_{d5}	5 BCLK cycles		CSZ low ****
Tck	24.5		BCLK period

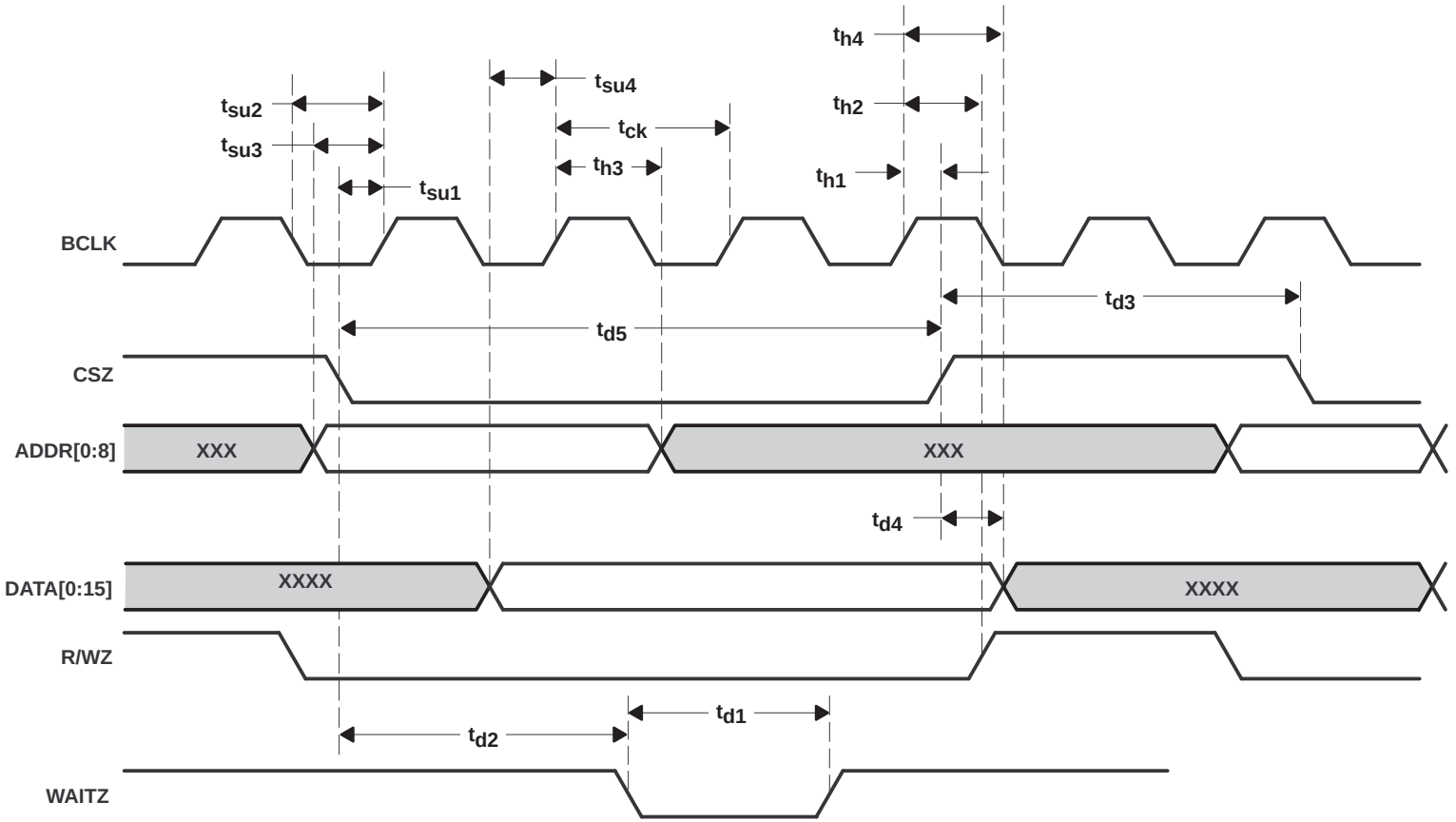


Figure 4–21. TMS320AV7100 ARM Read/Write Critial Timing

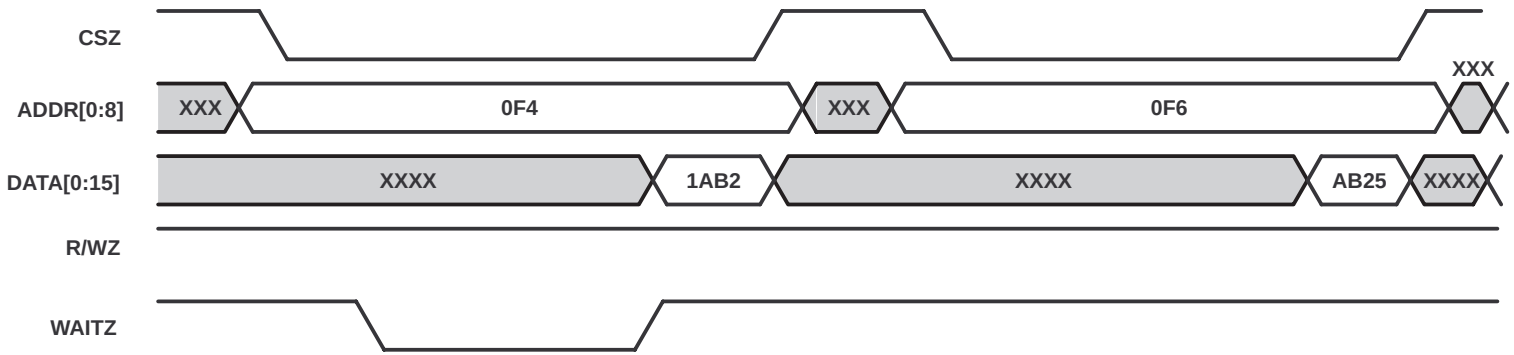


Figure 4–22. TMS320AV7100 Handshake Mode Read Timing

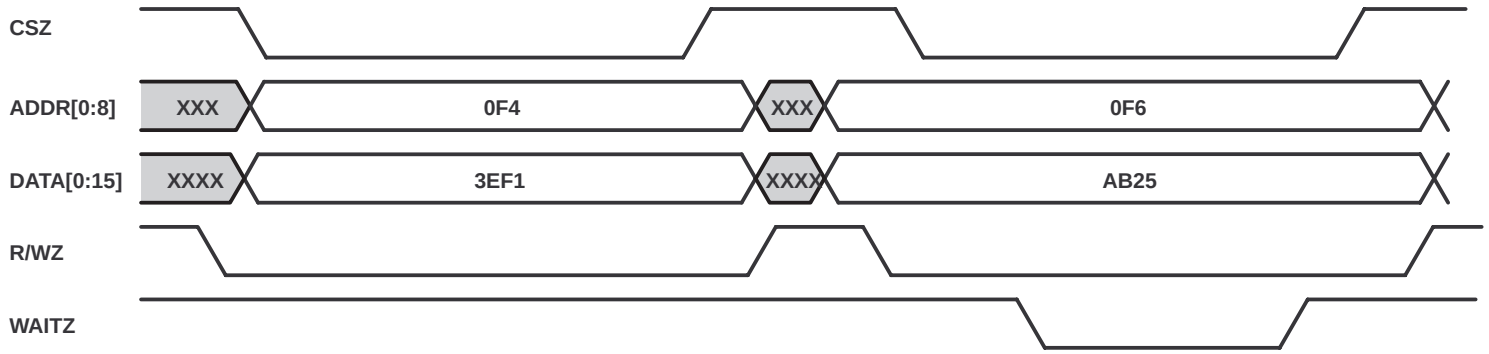


Figure 4–23. TMS320AV7100 Handshake Mode Write Timing

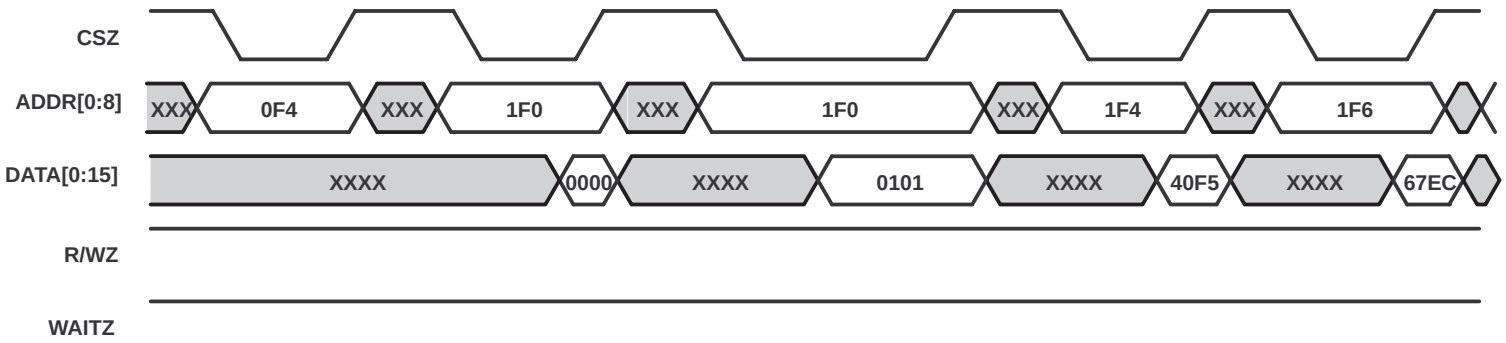


Figure 4–24. TMS320AV7100 ARM Blind Access Mode Read Timing

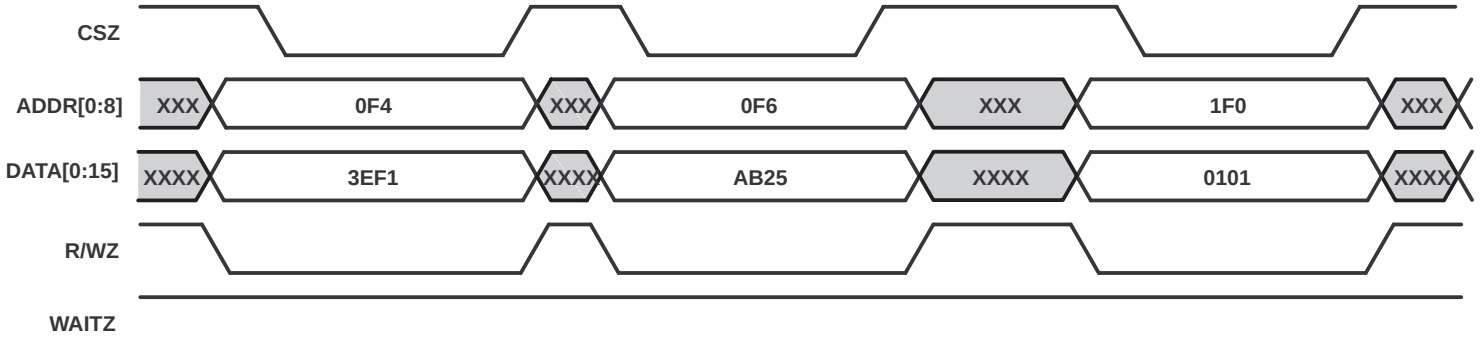


Figure 4–25. TMS320AV7100 ARM Blind Access Mode Write Timing

4.2.6.1 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 TSB12LV42 latches the data at the next rising edge of BCLK after the CSZ goes low. Once the TSB12LV42 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 TSB12LV42 has the read data available, it deasserts EXTWAITZ. The TMS320AV7100 deasserts CSZ by the next rising edge of its CLK40. The TSB12LV42 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 500 ns 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 an internal programmable wait state register. To avoid this timeout, the TSB12LV42 microprocessor interface state machine aborts the current transaction and deasserts the EXTWAITZ signal if the TSB12LV42'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 TSB12LV42 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 TSB12LV42 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 TSB12LV42 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 (3-state) in order to avoid bus contention on the EXTWAITZ pin. The TSB12LV42 does not provide any on-chip pull-up resistor for this pin, thus the user needs to add an external pull-up resistor on this pin in this case. If the TSB12LV42 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 has dedicated INT lines, therefore no sharing of this signal is needed. TSB12LV42 outputs normal totem-pole signal level for this pin. Also, the application is allowed to write a 1 to the TSB12LV42'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 TSB12LV42's 16-bit data bus contains the byte data, the lower byte (bit 8–15) is driven low. When writing to the TSB12LV42, 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 TSB12LV42. (Intel 8051 mode uses the lower 8 bits of the data bus for data exchange.)

4.2.7 68000 Mode Timing Diagrams

NOTE:
The design of this device ensures the following timing parameters.

Table 4–6. Motorola 68000 Critical Timing Characteristics

PARAMETER	MIN (ns)	MAX (ns)
t_{d1}		0
t_{d2}		7.2
t_{d3}		40
t_{d4}		130
t_{d5}		40
t_{su1}	10	
t_{su2}	0	
t_{su3}	2	
t_{h1}		40
t_{h2}		0
t_{h3}		40

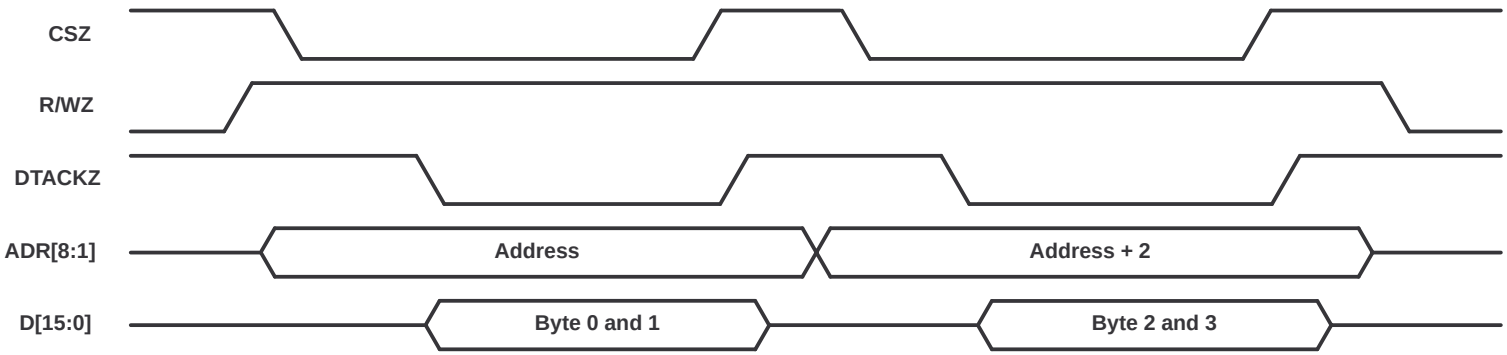


Figure 4–26. Motorola 68000 Read Timing

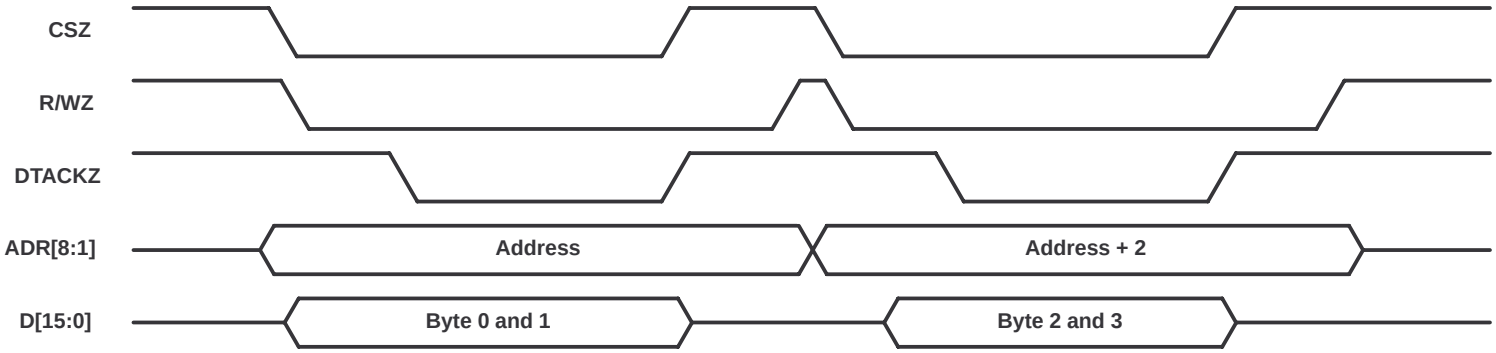


Figure 4–27. Motorola 68000 Write Timing

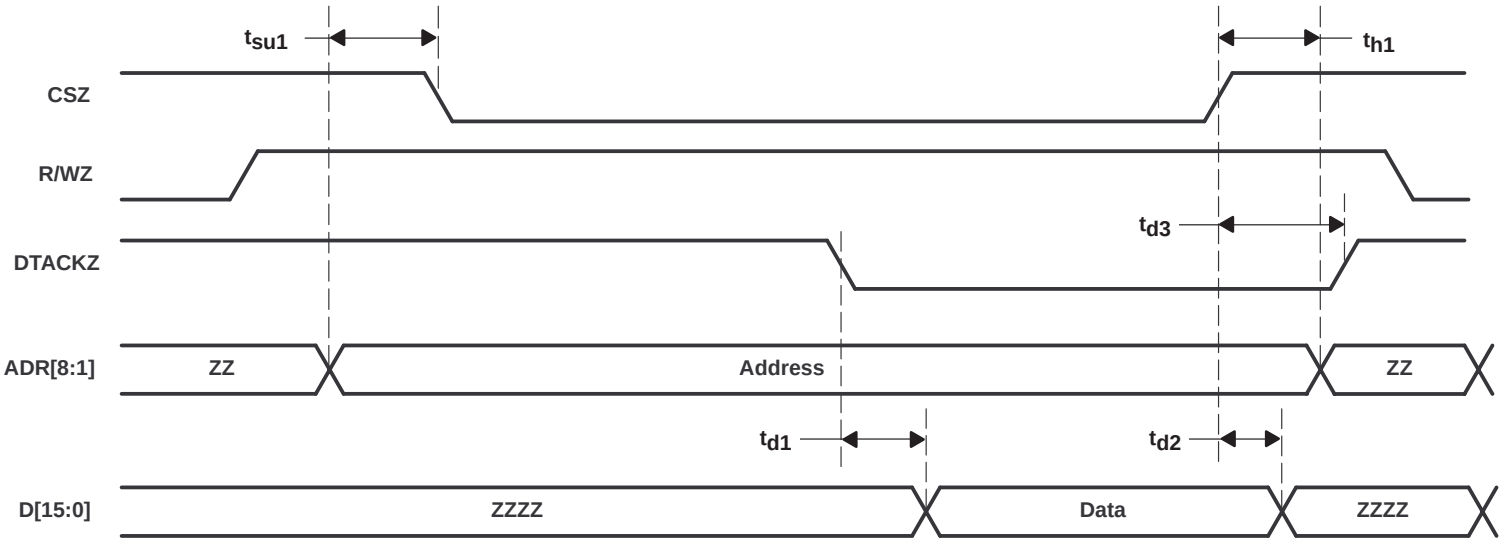


Figure 4–28. Motorola 68000 Read Critical Timing

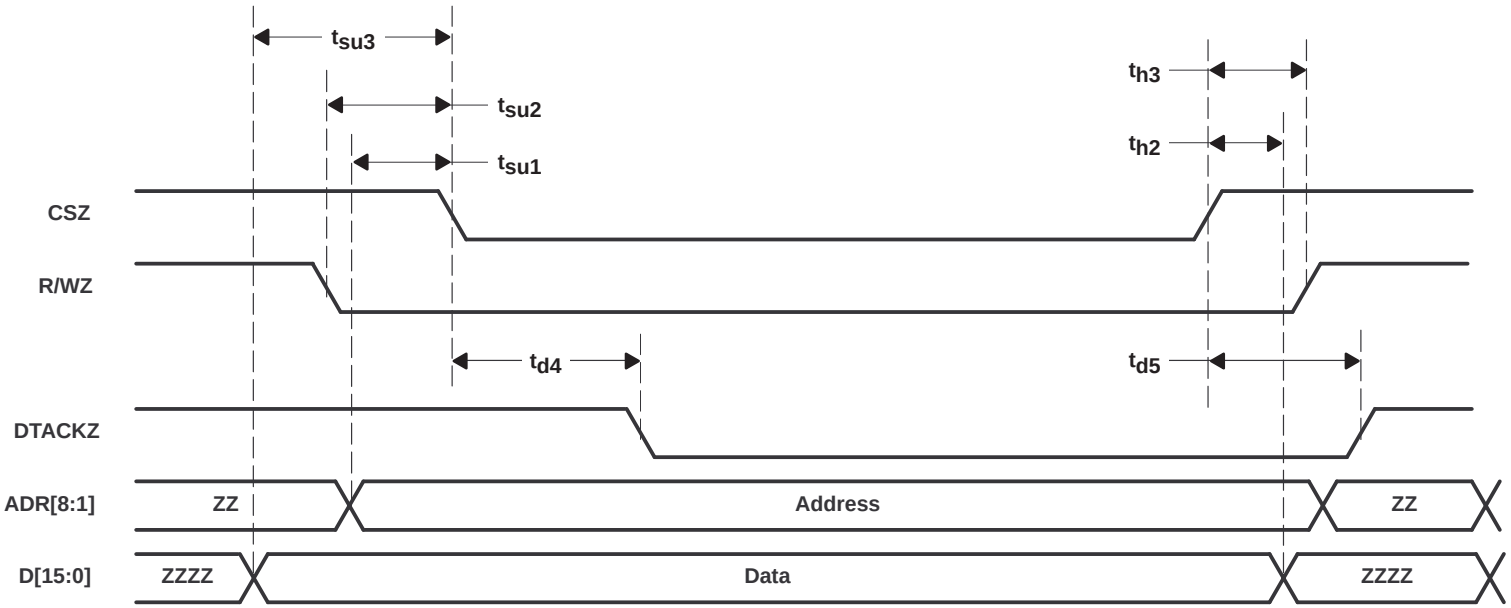


Figure 4–29. Motorola 68000 Write Critical Timing

4.2.8 8051 Mode Timing Diagrams

The Intel 8051 mode always operates in blind access mode.

The lower byte (bits 8–15) of TSB12LV42'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 TSB12LV42, the 8051 host should put the byte write data in the lower byte. The data in the upper byte has no affect on the data to be written into TSB12LV42.

NOTE:
The design of this device ensures the following timing parameters.

Table 4–7. Intel 8051 Critical Timing Characteristics

PARAMETER	MIN (ns)	MAX (ns)
t_{d1}	30	
t_{d2}	20	
t_{d3}	130	
t_{d4}	100	
t_{d5}	10	
t_{d6}	130	
t_{d7}	10	
t_{su1}	7	
t_{su2}	0	
t_{h1}		10
t_{h2}		2
t_{h3}	10	

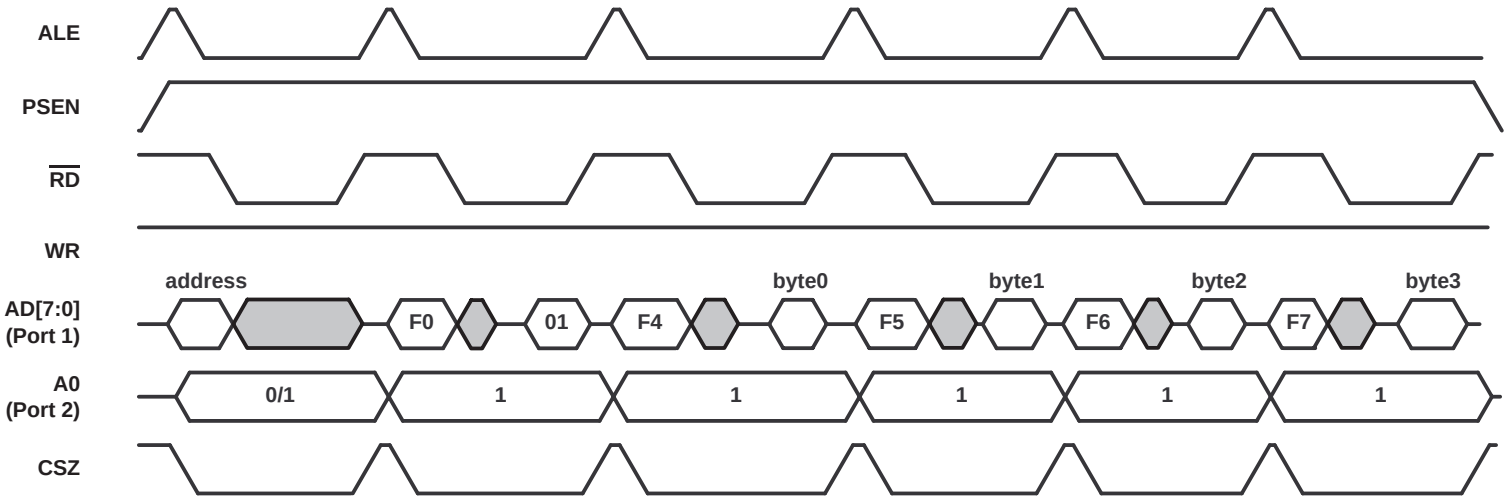


Figure 4–30. Intel 8051 Read Timing (Blind Access Read)

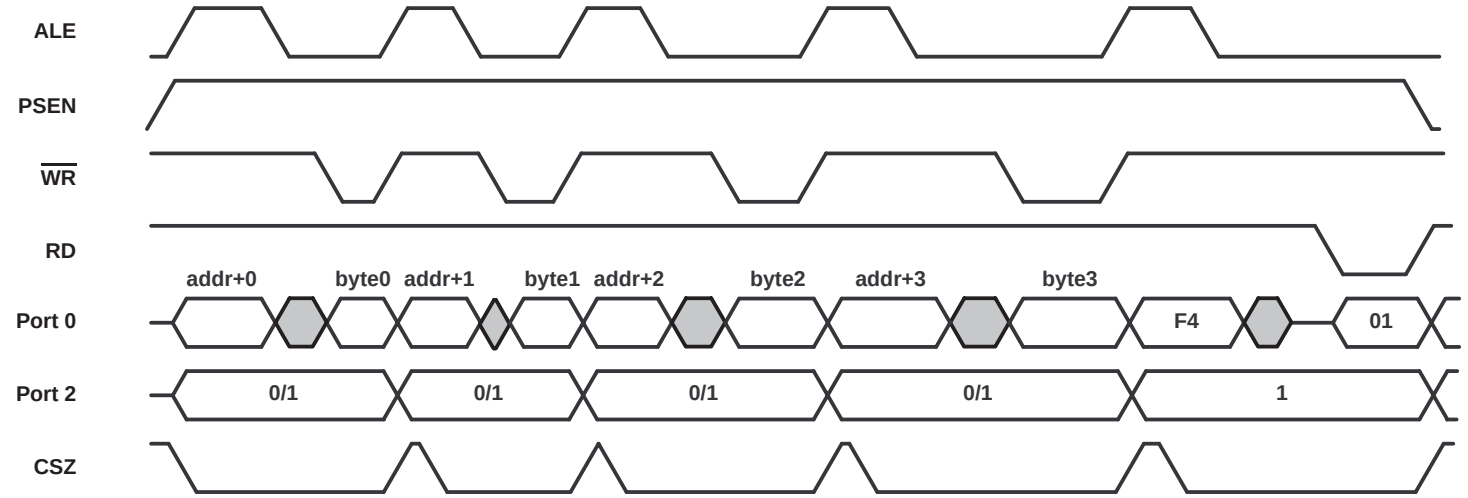


Figure 4–31. Intel 8051 Write Timing (Blind Access Write)

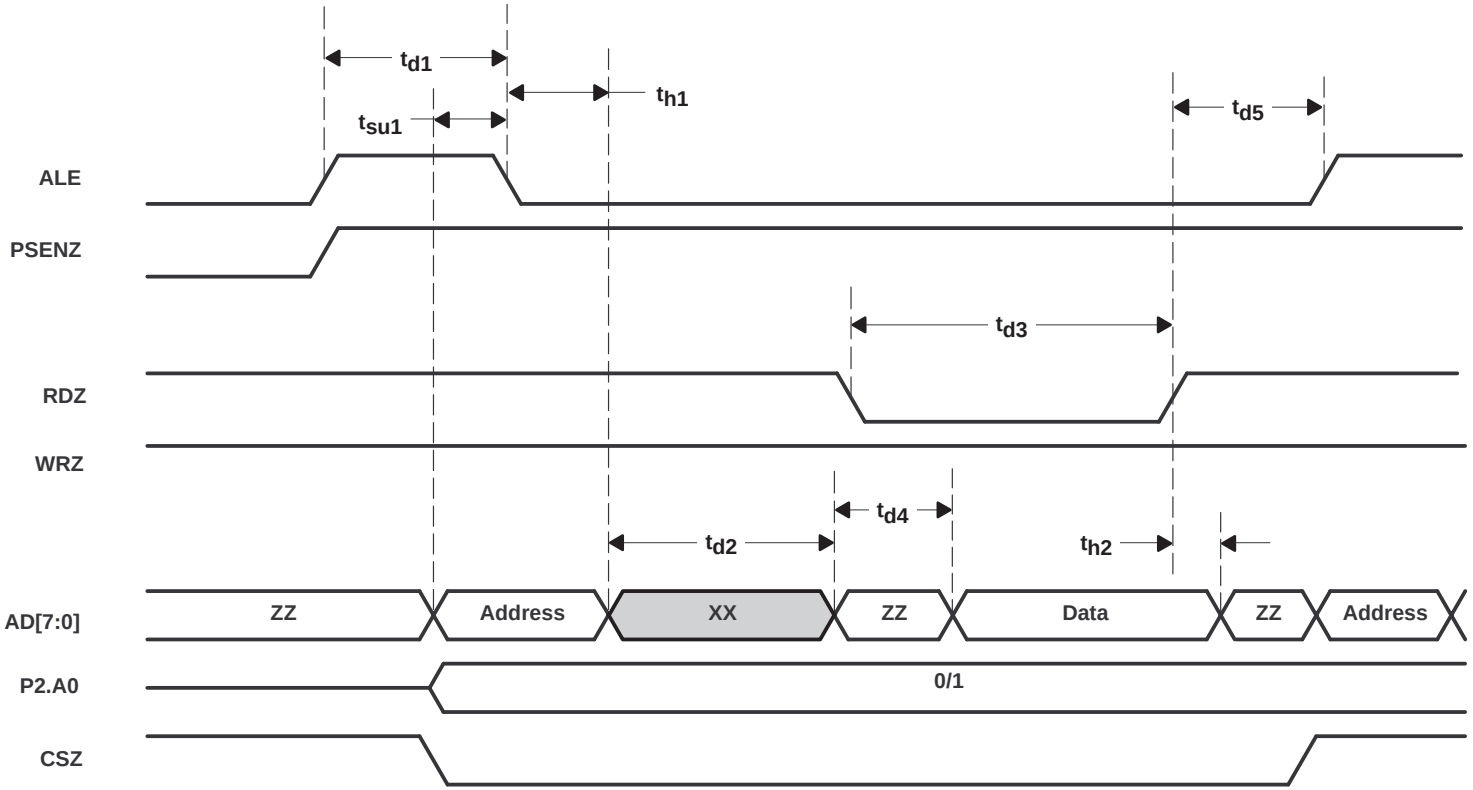


Figure 4–32. Intel 8051 Read Critial Timing

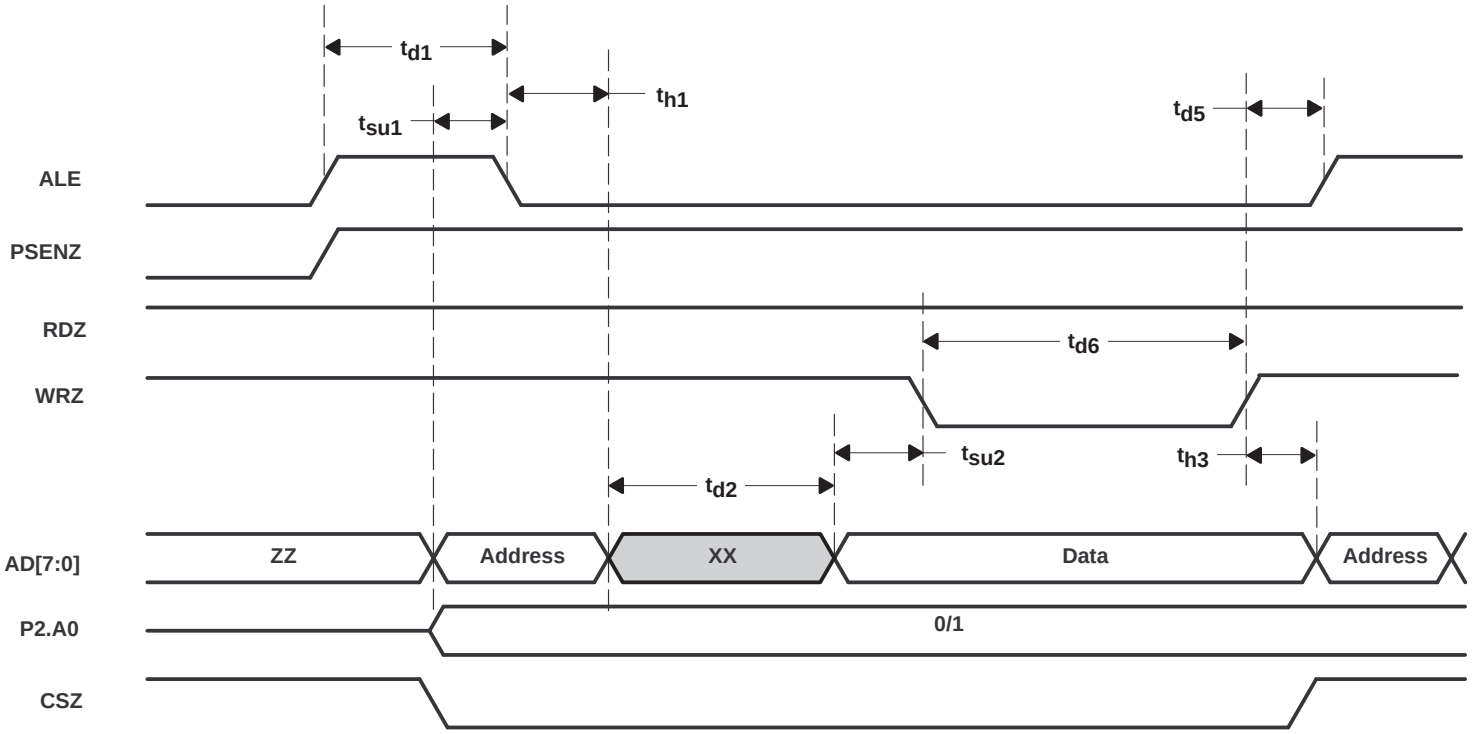


Figure 4–33. Intel 8051 Write Critial Timing

4.2.9 Blind Access Mode Specific Issues

Due to the long maximum access latency to some configuration registers (about 17 BCLK at 50 MHz), 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 an external wait/ready line handshake signal and could be unnecessarily bandwidth-limited by the long access time of the TSB12LV42's microprocessor interface. The blind access mode allows a microprocessor to perform a read or write transaction to the TSB12LV42 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.

Two registers are used for blind access mode operation, BASTAT (blind access status register, at address 1F0h) and BAHR (blind access holding register, at 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.

4.2.9.1 Blind Access Write

Blind access write still 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. 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.

4.2.9.2 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 are 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.

4.2.9.3 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 TSB12LV42'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, and 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.

4.2.9.4 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 TSB12LV42'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.

4.2.10 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 4–34 and 4–35). 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 TSB12LV42 since all its Configuration Registers are Big endian and users could potentially use a little endian type processor. The TSB12LV42 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 4–34. Big Endian Illustration chart

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

Figure 4–35. Little Endian Illustration chart

Since the TSB12LV42'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 unstacking (for reads) operation must be performed on the data bus. For little endian processors, the TSB12LV42 can perform the swapping of bytes on the data bus required to allow both the processor and the TSB12LV42 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 below.

CAUTION:

For the host processor to work correctly with the TSB12LV42, users **must** correctly connect the address and data busses of their microprocessor to the TSB12LV42's microprocessor port. Users must connect the MSB (most significant bit) of their address/data bus to the address/data MSB of the TSB12LV42. This must be done regardless of bit number labeling or which type of endianness their microprocessor uses. For little endian processor, 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).

4.2.10.1 Byte Swapping for Little Endian Systems

The BeCtl bit in the TSB12LV42'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. 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.

4.2.10.2 Data Invariant System Design

Figure 4–36 shows a little endian data invariant system design example. In this system, the byte addresses are not preserved. Byte_0 in the host microprocessor's little endian system contains aa. Byte_0 in TSB12LV42'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.

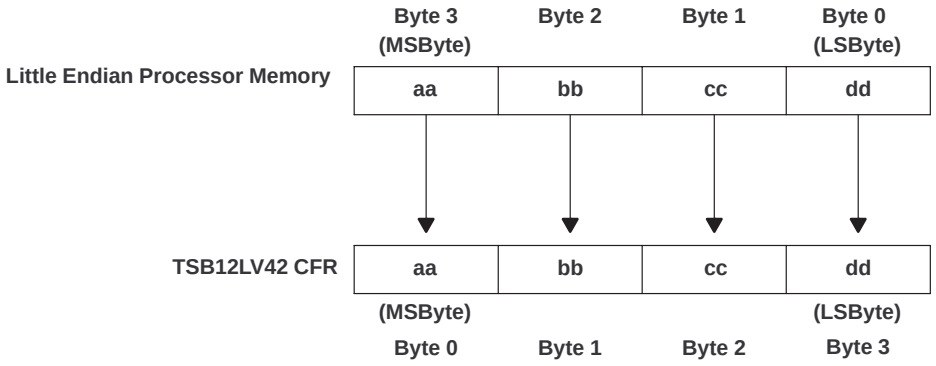


Figure 4–36. Little Endian Data Invariant System Design Illustration Chart

4.2.10.3 Address Invariant System Design

Figure 4–37 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 TSB12LV42'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.

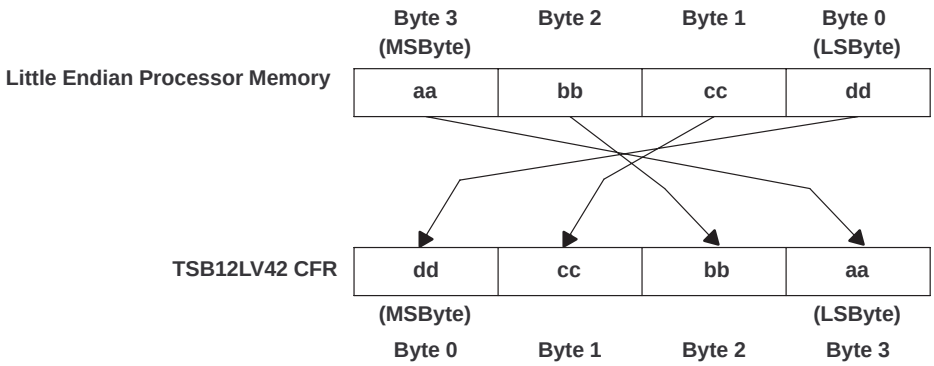


Figure 4–37. Little Endian Address Invariant System Design Illustration Chart

4.2.11 Use of Interrupts with DVLynx

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 14h and 1Ch, respectively. See Figure 4–38 for the interrupt hierarchy.

NOTE:

Even if an interrupt is masked off, its value in register 10h or 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) contains the interrupt. The IGRP0 and IGRP1 bits are available in both registers 10h and 18h. Each interrupt can be cleared by writing a 1 to the interrupt bit.

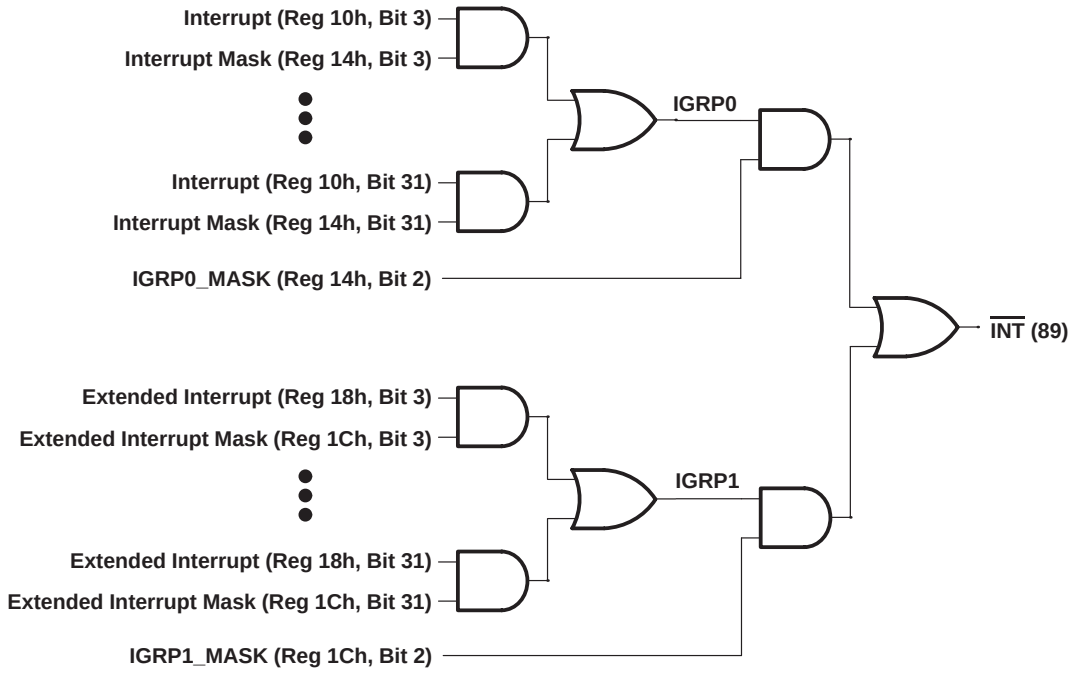


Figure 4–38. Interrupt Hierarchy

4.3 TSB12LV42 to 1394 Phy Interface Specification

4.3.1 Introduction

This chapter provides an overview of a TSB12LV42 to the phy interface. The information that follows can be used as a guide through the process of connecting the TSB12LV42 to a 1394 physical-layer device. The part numbers referenced, the TSB21LV03A and the TSB12LV42, represent the Texas Instruments implementation of the phy (TSB21LV03A) and link (TSB12LV42) 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 TSB12LV42 phy-link interface are mentioned.

4.3.2 Assumptions

The TSB12LV42 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 TSB12LV42. The TSB12LV42 can drive these terminals only after it has been given permission by the Phy layer. A dedicated request terminal (LREQ) is used by the TSB12LV42 for any activity that the designer wishes to initiate.

4.3.3 Block Diagram

The functional block diagram of the TSB12LV42 to Phy layer is shown in Figure 4–39.

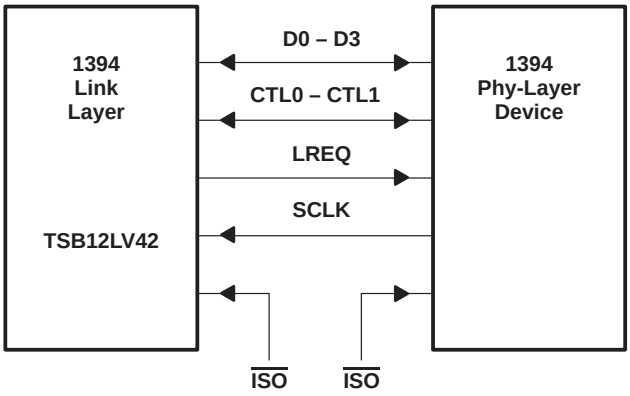


Figure 4–39. Functional Block Diagram of the TSB12LV42 to Phy Layer

4.3.4 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.

4.3.4.1 Phy Interface Has Control of the Bus

Table 4–8. 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 TSB12LV42.
10	Receive	An incoming packet is being sent from the Phy layer to the TSB12LV42.
11	Transmit	The TSB12LV42 has been given control of the bus to send an outgoing packet.

4.3.4.2 TSB12LV42 Has Control of the Bus

The TSB12LV42 has control of the bus after receiving permission from the Phy layer.

Table 4–9. TSB12LV42 Control of Bus Functions

CTL0, CTL1	NAME	DESCRIPTION OF ACTIVITY
00	Idle	The TSB12LV42 releases the bus (transmission has been completed).
01	Hold	The TSB12LV42 is holding the bus while data is being prepared for transmission, or the TSB12LV42 wants to send another packet without arbitration.
10	Transmit	An outgoing packet is being sent from the TSB12LV42 to Phy layer.
11	Reserved	None

4.3.5 Request

A serial stream of information is sent across the LREQ terminal whenever the TSB12LV42 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 4–10. Request Functions

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

4.3.5.1 LREQ Transfer

Bus Request

Table 4–11. 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 4–14 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 4–12. 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 4–14 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 4–13. 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 4–14 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 4–14. TSB12LV42 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 4–15. Request-Speed Functions

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

4.3.5.2 Bus Request

For fair or priority access, the TSB12LV42 requests control of the bus at least one clock after the TSB12LV42/Phy interface becomes idle. CTL0 – CTL1 = 00 indicates that the physical layer is in an idle state. If the TSB12LV42 senses that CTL0 – CTL1 = 10, then it knows that its request has been lost. This is true any time during or after the TSB12LV42 sends the bus request transfer. Additionally, the Phy interface ignores any fair or priority requests when it asserts the receive state while the TSB12LV42 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 TSB12LV42 can issue an isochronous bus request. When arbitration is won, the TSB12LV42 proceeds with the isochronous transfer of data. The isochronous request is cleared in the Phy interface once the TSB12LV42 sends another type of request or when the isochronous transfer has been completed.

The TakeBus request is issued when the TSB12LV42 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 TSB12LV42. The TSB12LV42 sends an acknowledgment to the sender unless the header CRC of the packet turns out to be invalid. In this case, the TSB12LV42 releases the bus immediately; it is not allowed to send another type of packet on this grant. To ensure this, the TSB12LV42 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).

4.3.5.3 Read/Write Requests

When the TSB12LV42 requests to read the specified register contents, the Phy interface sends the contents of the register to the TSB12LV42 through a status transfer. When an incoming packet is received while the Phy interface is transferring status information to the TSB12LV42, 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 TSB12LV42 is allowed to request read or write operations at any time.

See Section 4.3.6, *Status*, for a more detailed description of the status transfer.

4.3.6 Status

A status transfer is initiated by the Phy interface when it has some status information to transfer to the TSB12LV42. The transfer is initiated by asserting the following: CTL0 – CTL1 = 01 and D0 – D1 are used to transmit the status data; see Table 4–16 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.

4.3.6.1 Status Request

The definition of the bits in the status transfer is shown in Table 4–16.

Table 4–16. 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 TSB12LV42 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 TSB12LV42 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 TSB12LV42.
8–15	Data	The data bits hold the data that is to be sent to the TSB12LV42.

Normally, the Phy interface sends just the first four bits of data to the TSB12LV42. These bits are used by the TSB12LV42 state machine. However, if the TSB12LV42 initiates a read request (through a request transfer), then the Phy interface sends the entire status packet to the TSB12LV42. Additionally, the Phy interface sends the contents of the register to the TSB12LV42 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 TSB12LV42 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.

4.3.6.2 Transmit

When the TSB12LV42 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 TSB12LV42 by asserting the transmit state on the CTL terminals for at least one SCLK cycle. The TSB12LV42 takes control of the bus by asserting either hold or transmit on the CTL lines. Hold is used by the TSB12LV42 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 TSB12LV42 by asserting a data-on state on the bus. It is not necessary for the TSB12LV42 to use hold when it is ready to transmit as soon as bus ownership is granted.

When the TSB12LV42 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 TSB12LV42 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 TSB12LV42 needs to send another packet without releasing the bus. For example, the TSB12LV42 may want to send consecutive isochronous packets or it may want to attach a response to an acknowledgment. To do this, the TSB12LV42 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 TSB12LV42 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 TSB12LV42 can then proceed with the transmittal of the second packet. After all data has been transmitted and the TSB12LV42 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 to inform the network of a change in speed.

4.3.6.3 Receive

When data is received by the Phy interface from the serial bus, it transfers the data to the TSB12LV42 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 TSB12LV42 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-TSB12LV42 protocol and not included in the packets CRC calculation.

SPD = Speed code
D0 => Dn = Packet data

Table 4–17. Speed Code for Receive

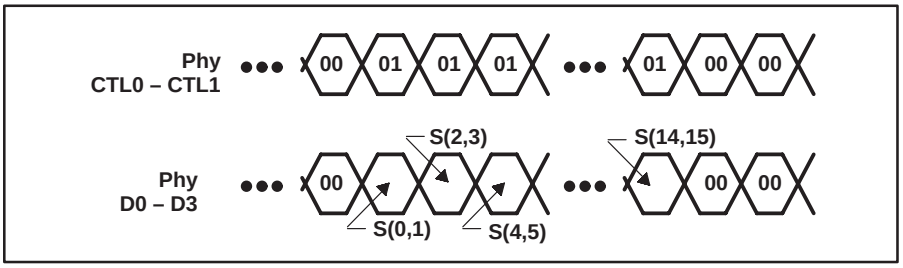
D0 – D3	DATA RATE
00xx†	100 Mbits/s
0100†	200 Mbits/s
0101	400 Mbits/s
111111111	Data-on indication

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

4.3.7 TSB12LV42 to Phy Bus Timing

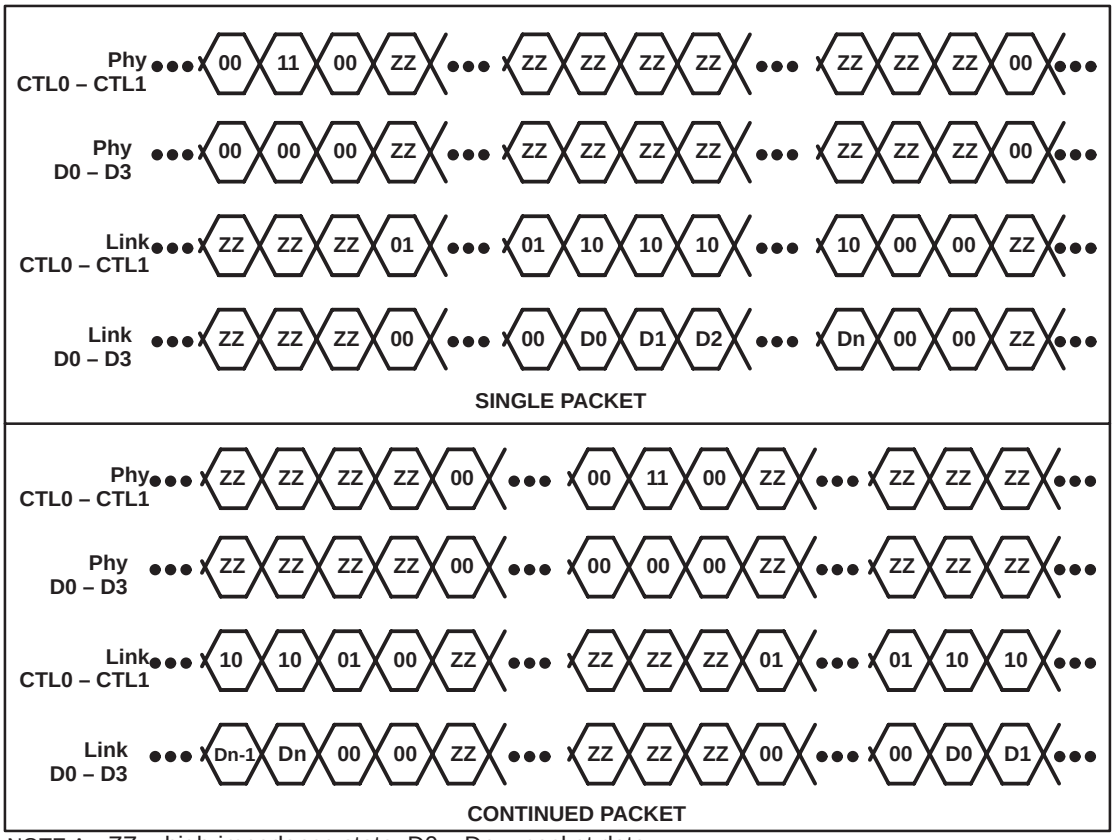


Figure 4–40. LREQ Timing



NOTE A: Each cell represents one SCLK sample time.

Figure 4–41. Status-Transfer Timing



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

Figure 4–42. Transmit Timing

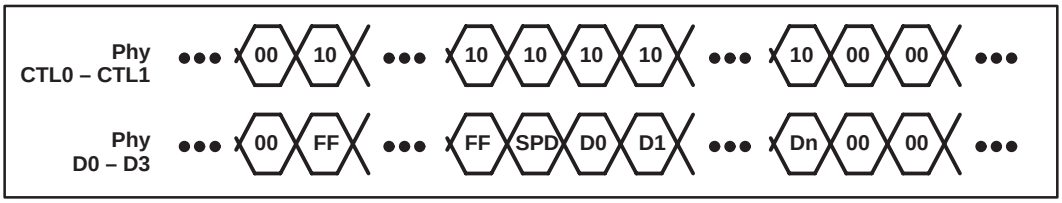
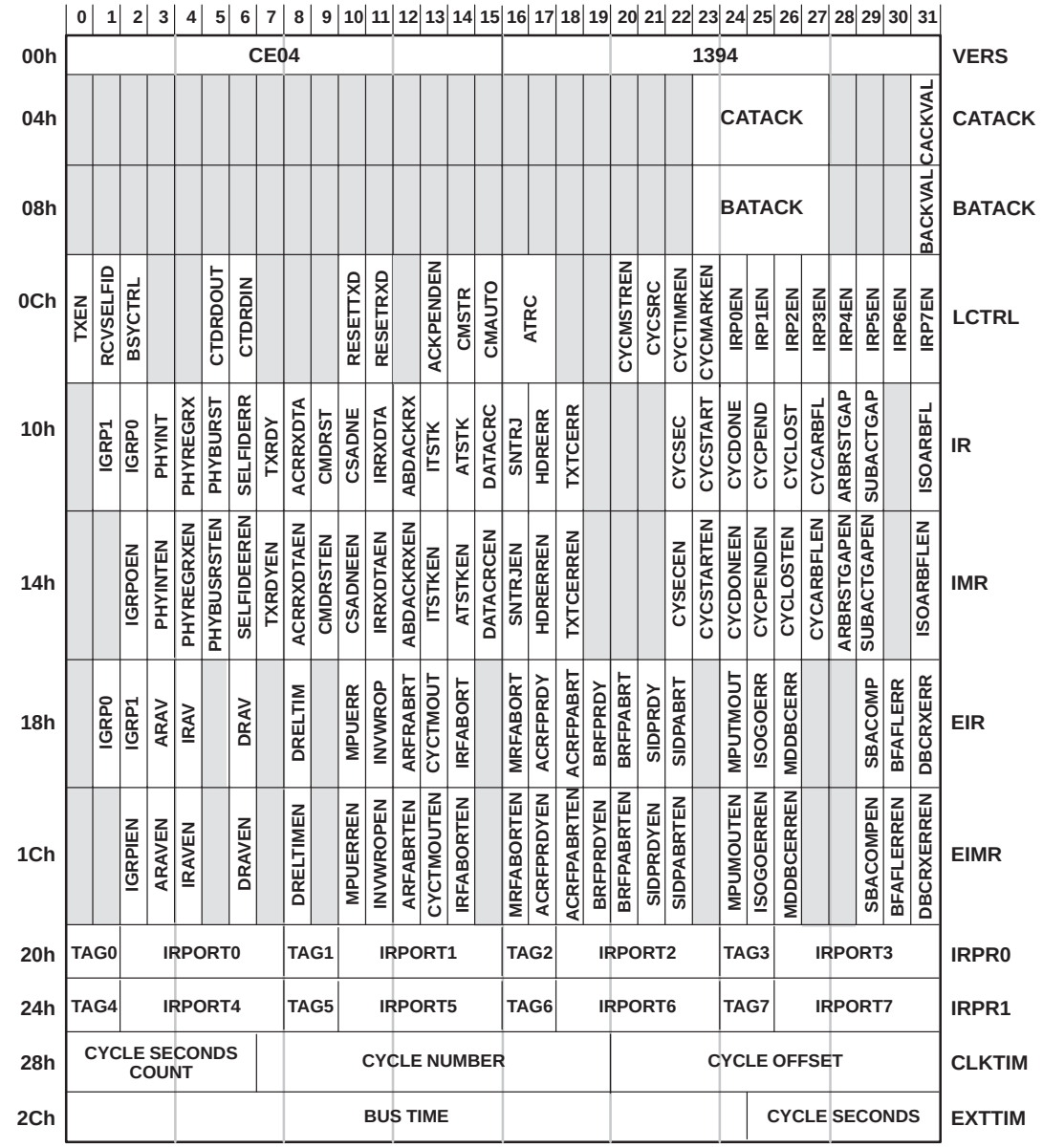


Figure 4–43. Receiver Timing

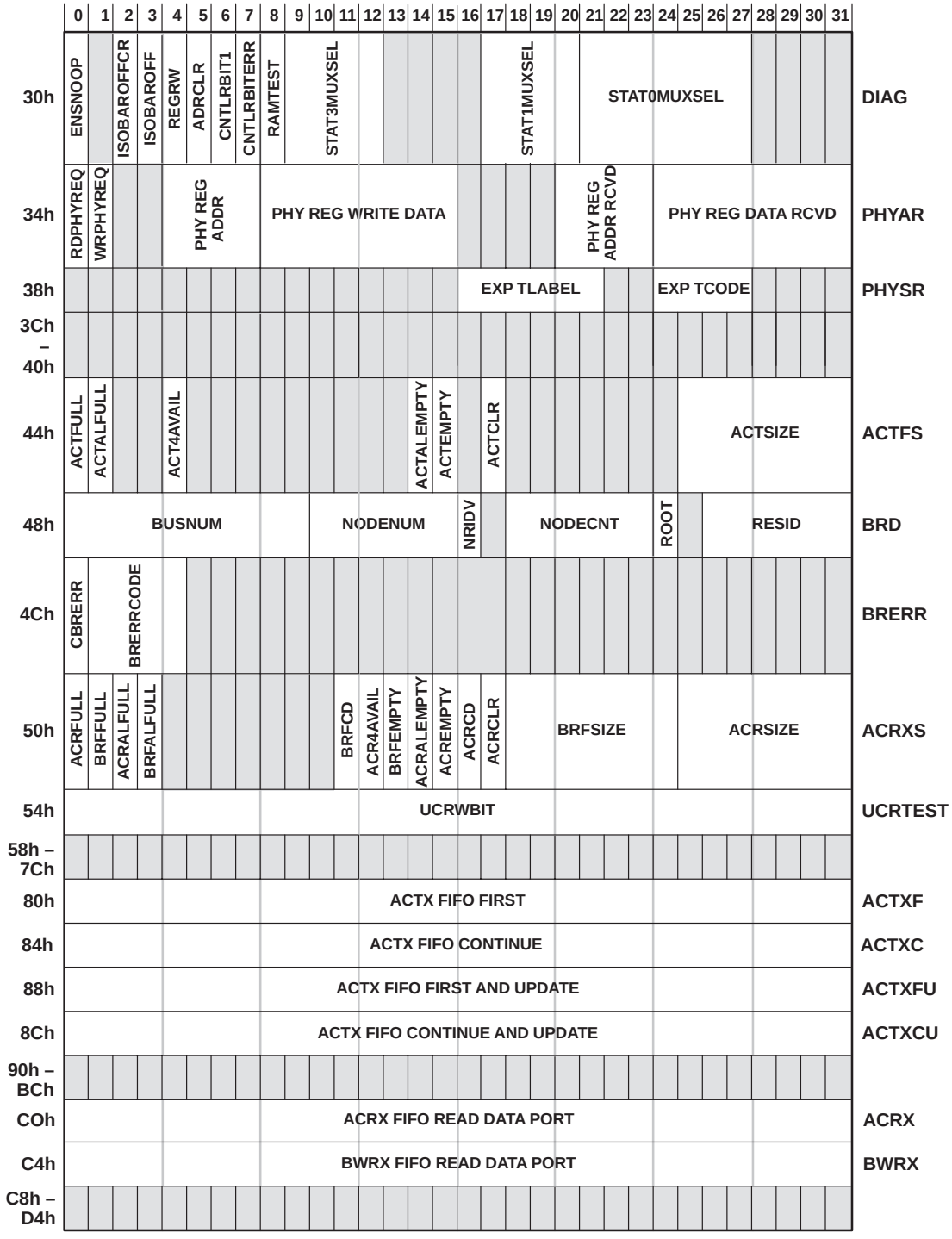
5 Detailed Operation and Programmers Reference

5.1 TSB12LV42 Configuration Register



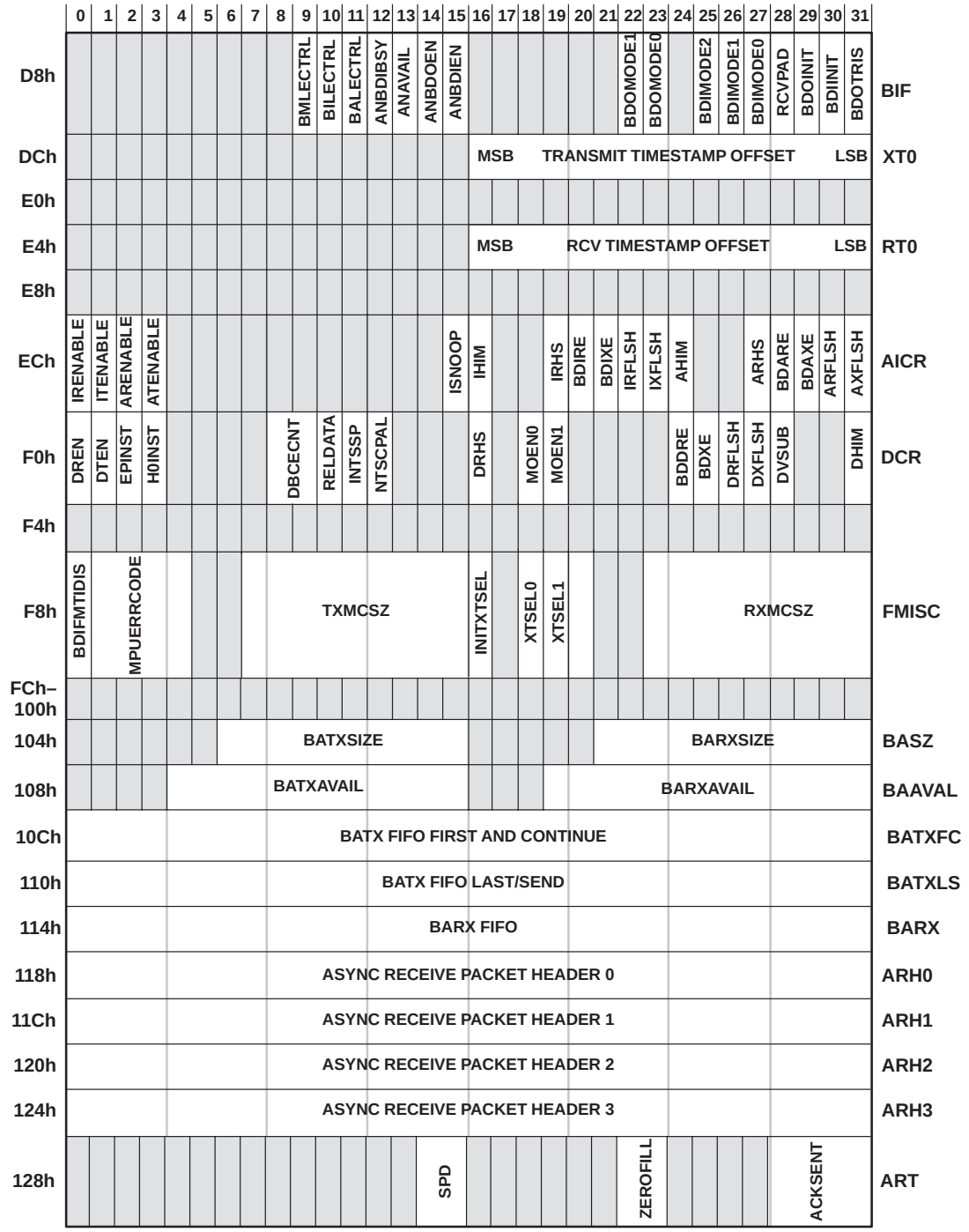
NOTE A: All gray areas (bits) are reserved bits.

Figure 5–1. Internal Register Map



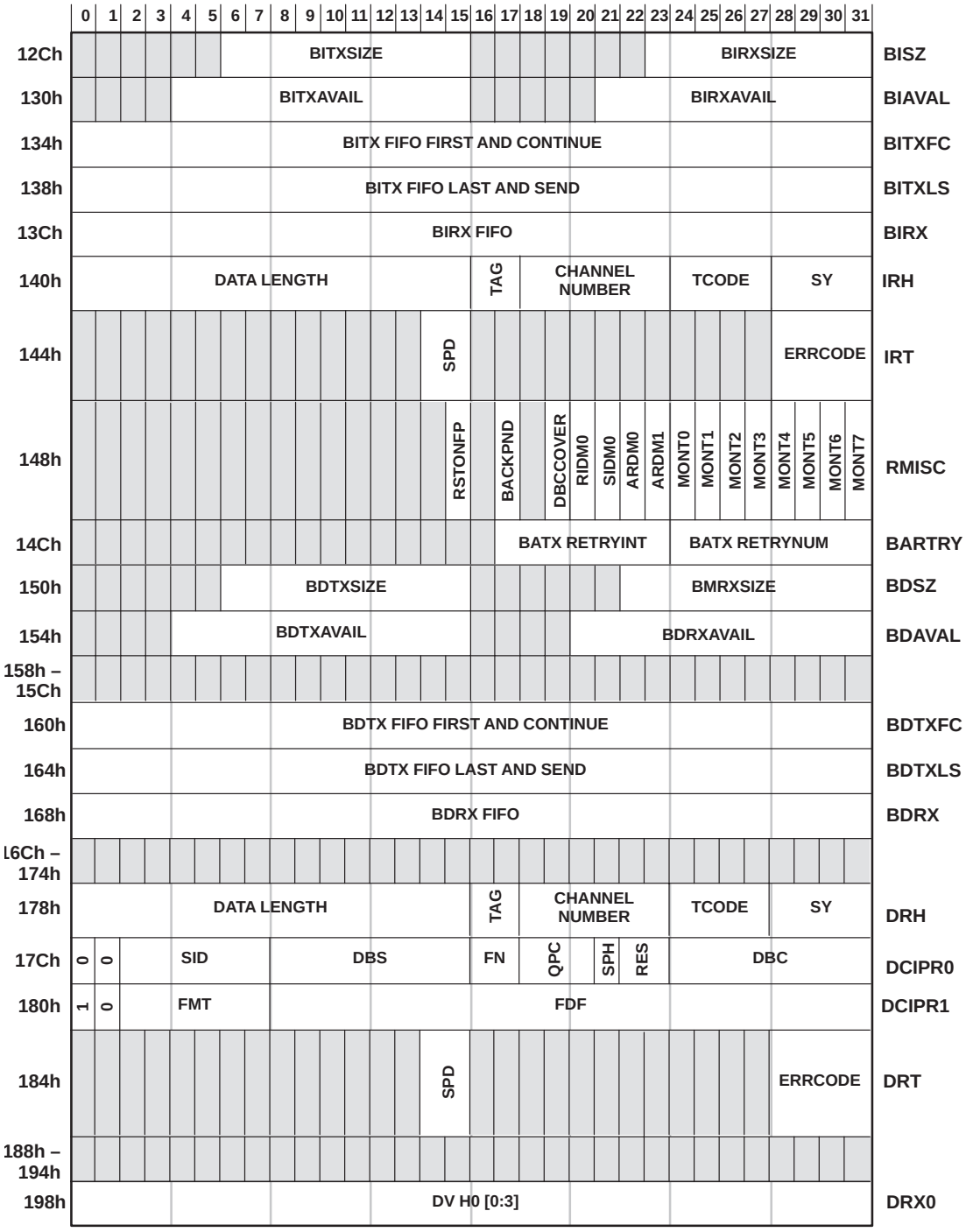
NOTE A: All gray areas (bits) are reserved bits.

Figure 5–1. Internal Register Map (Continued)



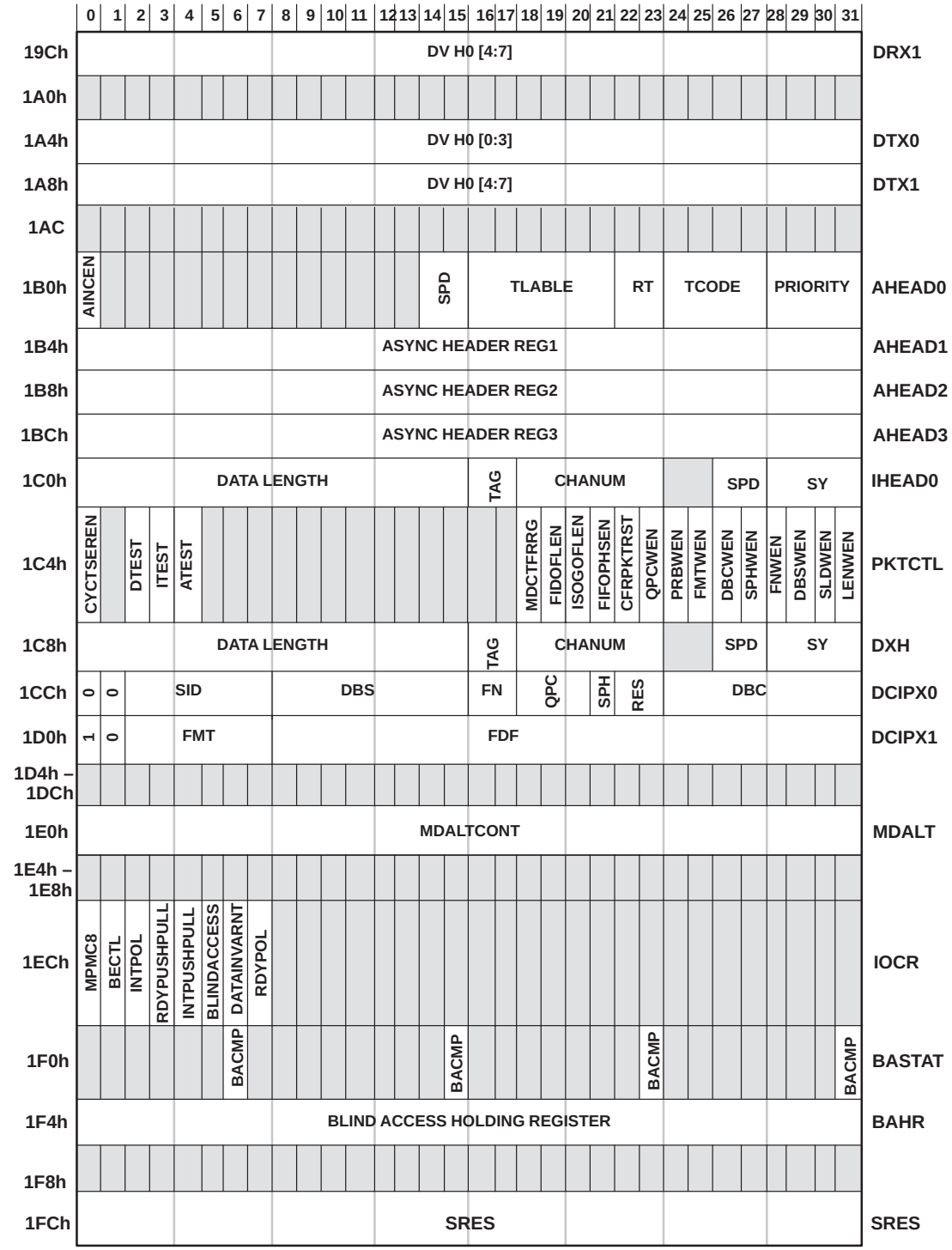
NOTE A: All gray areas (bits) are reserved bits.

Figure 5–1. Internal Register Map (Continued)



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

Figure 5–1. Internal Register Map (Continued)



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

Figure 5–1. Internal Register Map (Continued)

5.2 Version Register (VERS at Addr 000h)

This address port provides the application software with the version number and revision number of the DVLynx 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 = CE04h
16 – 31	REV	R	Revision Number. Hardwired value = 1394h

5.3 C Acknowledge Register (CACK at Addr 004h)

This register provides the application software with the last acknowledge that was received for an asynchronous packet transmitted from the asynchronous transmit control FIFO. 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 – 22	Not Used	R	
23 – 27	CATACK	R	Asynchronous control transmit acknowledge. The acknowledge value received from the link transmitter for a packet that was transmitted from the asynchronous control transmit FIFO. These bits are normally read only. For diagnostic reasons these bits can be written to by the application software by setting the REGRW bit in the link diagnostics register (DIAG at Addr 30h) to 1. The default value of CATACK is 00000b **CATACK[23] CATACK[24:27]** 0 Normal 1394 4 bit ack code. 1 0000 – No Ack received. Ack timeout 1 0001 – Ack pkt longer than 8 bits 1 0010 – Ack pkt shorter than 8 bit
28 – 30	Not Used	R	
31	CACKVAL	R	CATACK new value. 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 is normally read only. For diagnostic reasons this bit can be written to by setting the REGRW bit in the Link Diagnostics register (DIAG at Addr 30h) to a 1. The default value of CACKVAL is 0.

5.4 B Acknowledge Register (BACK at Addr 008h)

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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 22	Not Used	R	
23 – 27	BATACK	R	Asynchronous bulky transmit acknowledge. The acknowledge value received from the link transmitter for a packet that was transmitted from the asynchronous bulky transmit FIFO. These bits are normally read only. For diagnostic reasons they can be written by setting the REGRW bit in the link diagnostics register (DIAG at Addr 30h) to 1. The default value is 00000b. **BATACK[23] BATACK[24:27]** 0 Normal 1394 4 bit ack code. 1 0000 – No Ack received. Ack timeout 1 0001 – Ack pkt longer than 8 bits 1 0010 – Ack pkt shorter than 8 bit
28 – 30	Not Used	R	
31	BACKVAL	R	BATACK new value. This bit is set to 1 to indicate that BATACK is valid. This bit is normally read only. For diagnostic purposes it can be written to by setting the REGRW bit in the link diagnostics register (DIAG at Addr 30h) to 1. The default value is 0.

5.5 Link Control Register (LCTRL at Addr 00Ch)

This register provides the application software with the capability to control and configure the operation of the 1394 link layer logic. Unless otherwise specified the bits in this register are cleared to 0 on powerup 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 set to 0, the operation of the link transmitter is disabled from operating. The default value is 0.
01	RCVSELFID	R/W	Receive self-ID enable. When set to a 1 this bit enables the link layer to receive self-ID packets during 1394 bus initialization. The default value is 0.
02	BSYCTRL	R/W	Transaction layer busy override. When set to 1, the link receiver unconditionally busys off all acknowledgeable incoming packets with BUSY_X. When set to logic zero the link receiver uses the availability of the selected receiving FIFO to determine if an incoming acknowledgeable packet is to be accepted or busied off with (BUSY_X). The default value is 0
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. CTDRDOUT = 0 Link in not a contender for bus isochronous resource manager CTDRDOUT = 1 Link is a contender for bus isochronous resource manager The default value of this bit is 0.
06	CTDRDIN	R/W	Contender data direction control. This bit is used in setting the direction of the external contender terminal. The default value is 1. CTDRDIN = 1 CONTENDER terminal is in input mode. CTDRDIN = 0 CONTENDER terminal is in output mode and is driven by the value of CTDRDOUT
07 – 09	Not Used		

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
10	RESETTXD	R/W	Reset link transmitter. Writing a 1 to this bit resets all state machines in the link layer that are involved in transmitting a packet. This bit is self clearing. The default value is 0.
11	RESETRXD	R/W	Reset link receiver. Writing a 1 to this bit resets all state machines in the link layer that are involved in the reception of a packet. This bit is self clearing. The default value is 0.
12	Not Used		
13	ACKPENDEN	R/W	Acknowledge pending enabled. When set to a 1 this bit enables the link receiver to acknowledge write and lock request packets to the Control FIFO with an ack pending code of (2h). If this bit is set to 0 then the receiver uses the ack_complete code of (1h). The default value of this bit is 1.
14	CMSTR	R/W	Cycle master. The CMSTR and CMAUTO bits are used in together by the application software for enabling the cycle master auto select/deselect mode. Please see the CYCMSTREN bit description for function of the CMSTR bit. This bit is automatically cleared to 0 when a subaction is detected and the phy is no longer root. The default value is 0.
15	CMAUTO	R/W	Cycle master automatic. The CMSTR and CMAUTO bits are used in conjunction by application software for enabling the cycle master auto select/deselect mode. Please see the CYCMSTREN bit description for function of the CMAUTO bit. The default value is 0.
16 – 17	ATRC	R/W	Asynchronous transmit retry code. This code is logically ORed with the retry code field (00) in the transmit packet, and the packet is resent. 00 – retry_O (new) 01 – retry_X 10 – retry_A 11 – retry_B
18 – 19	Not Used		
20	CYCMSTREN	R/W	Cycle master enable. When this bit is set to 1, the link layer functions as cycle master for the 1394 network. If the CMAUTO bit is set to 1 and after a bus-reset, the CYCMSTREN bit is automatically set or cleared according to the following function table. **ROOT CMSTR CYCMSTREN** 0 0 0 0 1 0 1 0 0 1 1 1 If the CMAUTO bit is set to 0, then a bus-reset is allowed to clear CYCMSTREN or the application software can set or clear this bit. The default value for this bit is 0.
21	CYCSRC	R/W	Cycle reset counter. 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. The default value for this bit is 0.
22	CYCTIMREN	R/W	Cycle timer enable. The cycle timer is enabled to count when this bit is set to 1. The cycle timer is disabled when the bit is set to 0. The default value for this bit is 0

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
23	CYCMARKEN	R/W	Cycle mark enable. When this bit is set to 1, cycle marks are inserted into the bulky isochronous receive FIFO at the end of each isochronous cycle. When this bit is set to 0 then no cycle marks are inserted. The default value for this bit is 0.
24	IRP0EN (DV)	R/W	DV receive port comparator enable. when this bit is set to 1 the channel 0 DV 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. The default value for this bit is 0.
25	IRP1EN	R/W	Isochronous receive port comparator enable. when this bit is set to 1 the channel 1 isochronous packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to and expected value. The packet is received if the comparator detects a match. The default value for this bit is 0.
26	IRP2EN	R/W	Isochronous receive port comparator enable. when this bit is set to 1 the channel 2 isochronous packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to and expected value. The packet is received if the comparator detects a match. The default value for this bit is 0.
27	IRP3EN	R/W	Isochronous receive port comparator enable. when this bit is set to 1 the channel 3 isochronous packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to and expected value. The packet is received if the comparator detects a match. The default value for this bit is 0.
28	IRP4EN	R/W	Isochronous receive port comparator enable. when this bit is set to 1 the channel 4 isochronous packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to and expected value. The packet is received if the comparator detects a match. The default value for this bit is 0.
29	IRP5EN	R/W	Isochronous receive port comparator enable. when this bit is set to 1 the channel 5 isochronous packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to and expected value. The packet is received if the comparator detects a match. The default value for this bit is 0.
30	IRP6EN	R/W	Isochronous receive port comparator enable. when this bit is set to 1 the channel 6 isochronous packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to and expected value. The packet is received if the comparator detects a match. The default value for this bit is 0.
31	IRP7EN	R/W	Isochronous receive port comparator enable. when this bit is set to 1 the channel 7 isochronous packet receive comparator is enabled for comparing the channel number and/or tag field of the incoming packet to and expected value. The packet is received if the comparator detects a match. The default value for this bit is 0.

5.6 Interrupt Register (IR at Addr 010h)

The following register defines the group 0 interrupt status bits. The bits in this register can be cleared to 0 by writing a 1 to the bit. Unless otherwise specified the bits in this register are defaulted to 0 on a powerup or software reset. With the exception of bit1, 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 link diagnostics register (DIAG at Addr 30h) to 1.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	Not Used		
01	IGRP1	R	Interrupt status bit flag. When set to 1, one or more status bits in the extended interrupt status register (at Addr 14h) are set to 1.
02	IGRP0	R	Interrupt status bit flag. When set to 1, one or more interrupt status bits in this register are set to 1.
03	PHYINT	R/W	Phy interrupt. When set to 1, either an internal timeout has occurred or the cable power has dropped. The error can be decoded in register 34h
04	PHYREGRX	R/W	Phy register data received. When set to 1 a register value has been transferred to the Phy access register (at offset 34h) from the Phy interface.
05	PHYBUSRST	R/W	Phy bus reset. When set to 1, the Phy has entered the 1394 bus reset state.
06	SELFIDERR	R/W	Self-ID error. When set to 1, this interrupt 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	Asynchronous packet received. When set to a 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	Asynchronous control FIFO ack received. When this bit is set to 1, an ack has been received for a packet that was transmitted from the asynchronous control transmit FIFO.
11	IRRXDTA	R/W	Isochronous packet received. When set to a 1, the link receiver has confirmed isochronous data. This bit gets set when the last quadlet is received into the BIRX FIFO.
12	ABDACKRX	R/W	Asynchronous bulky FIFO ack received. When this bit is set to 1, an ack has been received for a packet that was transmitted from the asynchronous bulky transmit FIFO.
13	ITSTK	R/W	Isochronous Transmit FIFO stuck. When this bit is set to a 1, the link transmitter has detected an incomplete packet at the isochronous transmit-FIFO interface. To recover from this error, flush bulky isochronous transmit FIFO.
14	ATSTK	R/W	Asynchronous Transmit FIFO stuck. When this bit is set to a 1, the link transmitter has detected an incomplete packet at the asynchronous transmit-FIFO interface. When this condition occurs flush the bulky asynchronous transmit FIFO and the asynchronous control transmit FIFO.
15	DATACRC	R/W	Data CRC error. When this bit is set to 1, a data CRC error has occurred.
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 for storing it.
17	HDRERR	R/W	Header error detected. This bit is set to a 1 when the receiver detects a CRC error on a packet that may have been addressed to this node.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
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 asynchronous transmit FIFO and the asynchronous control transmit FIFO and reset the link transmitter (Reset TxD, register Ch).
19	Reserved	R/W	
20	Reserved	W	
21	Not Used		
22	CYCSEC	R/W	Cycle seconds. When set to a 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 set to a 1, the link transmitter has sent or the link receiver has received a cycle start packet.
24	CYCDONE	R/W	Cycle done. When set to a 1, a sub action 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 DVLynx is 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 DVLynx is not 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 is received. It remains set until a subaction gap is detected.
26	CYCLOST	R/W	Cycle lost. When set to a 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, cycle arbitration has failed.
28	ARBRSTGAP	R/W	Arbitration reset gap. When set to a 1, the link has detected that a arbitration reset gap has opened up on the 1394 bus.
29	SUBACTGAP	R/W	Sub action gap. When set to a 1, the link has detected that a sub action gap has opened up on the bus.
30	Not Used		
31	ISOARBFL	R/W	Isochronous arbitration failed. When set to a 1, the isochronous transmit request to send an isochronous packet failed to win bus arbitration.

5.7 Interrupt Register Enable Register (IMR at Addr 014h)

The following register defines the control bits that are used to enable the status bits in the interrupt status register (IR at Addr 10h) to generate an interrupt to the host processor. Unless otherwise specified these bits default to 0 on a powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	Not Used		
01	Reserved		
02	IGRP0EN	R/W	Interrupt master enable. This bit determines if the Interrupt register (addr 10h) is routed to the external $\overline{INT}$ terminal. Both Addr10 and Addr 18 (Extended interrupt) can be routed to the $\overline{INT}$ terminal.
03	PHYINTEN	R/W	Phy interrupt enable. When set to 1, the Phy interrupt status bit is enable for generating an interrupt.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
04	PHYREGRXEN	R/W	Phy register data received interrupt enable. When set to a 1, Phy register data received interrupt status bit is enabled for generating and interrupt to the host processor
05	PHYBUSRSTEN	R/W	Phy bus reset interrupt enable. When set to 1, the Phy bus reset interrupt status bit is enabled for generating an interrupt to the host processor
06	SELFIDERREN	R/W	Self-ID error interrupt enable. When set to a 1, the self-ID error interrupt status bit is enabled for generating an interrupt to the host processor.
07	TXRDYEN	R/W	Transmitter ready interrupt enable. When this bit is set to 1, the transmitter ready interrupt status bit is enabled for generating an interrupt to the host processor.
08	ACRRXDTAEN	R/W	Asynchronous packet received interrupt enable. When set to a 1, the asynchronous packet received interrupt status bit is enabled for generating an interrupt to the host processor.
09	CMDRSTEN	R/W	Command reset received interrupt enable. When this bit is set to 1, the command reset received interrupt status bit is enabled for generating an interrupt to the host processor.
10	CSADNEEN	R/W	Asynchronous control FIFO ack received interrupt enable. When this bit is set to 1, the asynchronous control FIFO ack received interrupt status bit is enabled for generating an interrupt to the host processor.
11	IRRXDTAEN	R/W	Isochronous packet received interrupt enable. When set to a 1, the isochronous packet received interrupt status bit is enabled for generating an interrupt to the host processor.
12	ABDACKRXEN	R/W	Asynchronous bulky FIFO ack received interrupt enable. When this bit is set to 1, the asynchronous bulky data FIFO ack received interrupt status bit is enabled to generate an interrupt to the host processor.
13	ITSTKEN	R/W	Isochronous transmit FIFO stuck interrupt enable. When this bit is set to a 1, the isochronous transmit FIFO stuck interrupt status bit is enabled for generating an interrupt.
14	ATSTKEN	R/W	Asynchronous Transmit FIFO stuck interrupt enable. When this bit is set to a 1, the Asynchronous Transmit FIFO stuck interrupt status bit is enabled for generating an interrupt to the host processor.
15	DATACRCEN	R/W	Enable data CRC interrupt. Set to 1 to enable.
16	SNTRJEN	R/W	Busy acknowledge sent by link receiver interrupt enable. When this bit is set to 1, the busy acknowledge interrupt status bit is enable for generating an interrupt to the host processor.
17	HDRERREN	R/W	Header error detected interrupt enable. When set to a 1, the header error interrupt status bit is enabled for generating an interrupt to the host processor.
18	TXTCERREN	R/W	Transmit tcode error interrupt enable. When this bit is set to 1, the transmit Tcode error interrupt status bit is enabled for generating an interrupt to the host processor.
19 – 21	Reserved		
22	CYCSECEN	R/W	Cycle seconds interrupt enable. When set to a 1, the cycle seconds interrupt status bit is enabled for generating an interrupt to the host processor.
23	CYCSTARTEN	R/W	Cycle started interrupt enable. When set to a 1, the cycle started interrupt status bit is enabled for generating an interrupt to the host processor.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
24	CYCDONEEN	R/W	Cycle done interrupt enable. When set to a 1, the cycle done interrupt status bit is enabled for generating an interrupt to the host processor.
25	CYCPENDEN	R/W	Cycle pending interrupt enable. When set to a 1, the cycle pending interrupt status bit is enabled for generating an interrupt to the host processor.
26	CYCLOSTEN	R/W	Cycle lost interrupt enable. When set to a 1, the cycle lost interrupt status bit is enabled for generating an interrupt to the host processor.
27	CYCARBFLEN	R/W	Cycle arbitration failed interrupt enable. When set to a 1, the cycle arbitration failed interrupt status bit is enabled for generating an interrupt to the host processor.
28	ARBRSTGAPEN	R/W	Arbitration reset gap interrupt enable. When set to a 1, the arbitration reset gap interrupt status bit is enabled for generating an interrupt to the host processor.
29	SUBACTGAPEN	R/W	Sub action gap interrupt enable. When set to a 1, the sub action gap interrupt status bit is enabled for generating an interrupt to the host processor.
30	Not Used		
31	ISOARBFLEN	R/W	Isochronous arbitration failed interrupt enable. When set to a 1, the isochronous arbitration failed interrupt status bit is enabled for generating an interrupt to the host processor.

5.8 Extended Interrupt Register (EIR at Addr 018h)

The following register defines the group 1 interrupt status bits. With the exception of bits 1 and 2, the bits in this register can be cleared 0 by writing a 1 to the bit. Unless otherwise specified the bits in this register are defaulted 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 link diagnostics register (DIAG at Addr 30h) to 1.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	Not Used		
01	IGRP0	R	Interrupt status bit flag. This bit is set to 1 whenever 1 or more interrupt status bits register 1Ch or register 14h are set to 1.
02	IGRP1	R	Interrupt status bit flag. When set to 1, one or more interrupt status bits in this register are set to 1.
03	ARAV	R/W	Asynchronous packet receive verification. When set to 1, a complete asynchronous packet has been received into the asynchronous bulky receive FIFO and the packet's header information has been copied into the Asynchronous Receive Packet Header registers.
04	IRAV	R/W	Isochronous packet receive verification. When set to 1, a complete isochronous packet has been received into the isochronous bulky receive FIFO and the packet's header information has been copied into the isochronous receive packet header register.
05	Reserved		
06	DRAV	R/W	DV packet receive verification. When set to 1, a complete DV packet has been received into the bulky data receive FIFO and the packet's header information has been copied into the receive packet header registers.
07	Reserved		

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
08	DRELTIM	R/W	DV release timer flag. When set to 1, the cycletimer has reached the value of the DV release time and an DVcell is now being released to the selected interface (bulky data interface or microprocessor interface). In DV mode this interrupt marks the beginning of the frame.
09	Reserved		
10	MPUERR	R/W	MPU error flag. When 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 FMISC register located at offset F8h bits 1 thru 4, contain the reason for the error.
11	INVWROP	R/W	Invalid read/write operation. When 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	Asynchronous partial packet purge. When set to 1, the receive packet routing control has detected and purged a partial packet from the asynchronous 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	CYCTMOUT	R/W	Cycle time out. When set to 1, the microprocessor interface has detected a timeout (17 BClk cycles reached) during TSB320AV7000 handshake mode.
14	IRFABORT	R/W	Isochronous partial packet purge. When 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 two quadlets available.
15	Not Used		
16	MRFABORT	R/W	DV partial packet purge. When set to 1, receive packet routing control has detected and purged a partial DV packet from the bulky DV receive FIFO. This can occur whenever the FIFO has less than two quadlets available, data has an incorrect format, or the device is not setup correctly for receive.
17	ACRFPRDY	R/W	Asynchronous packet confirm. When set to 1, the receive packet routing control has confirmed an entire packet into the asynchronous control receive FIFO
18	ACRFPABRT	R/W	Asynchronous partial packet purge. When set to 1, the receive packet routing control has detected and purged a partial packet from the asynchronous control receive FIFO
19	BRFPRDY	R/W	Packet confirm. When set to 1, the receive packet routing control has confirmed an entire packet into the broadcast control receive FIFO
20	BRFPABRT	R/W	Partial packet purge. When 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	Self-ID period end. When 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	Self-ID partial packet purge. When set to 1, the receive packet routing control has detected and purged a partial accumulation of Self-ID packets from the selected receive FIFO (asynchronous bulky receive FIFO or BROADCAST control receive FIFO).

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
23	Reserved		
24	MPUTMOUT	R/W	MPU time out. When 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	Isochronous go error. When set to 1, the packetizer has detected a premature isochronous go event
26	MDDBCERR	R/W	Data block error. When set to 1, the packetizer has detected a data block continuity error (DBC) on transmit.
27	Reserved		
28	Reserved		
29	SBACOMP	R/W	Packet transmit complete. When set to 1, the packetizer has successfully complete transmitting a packet from the bulky asynchronous transmit FIFO.
30	BFAFLERR	R/W	Asynchronous packet error. When set to a 1, the packet has failed to transmit an asynchronous packet from the bulky asynchronous transmit FIFO.
31	DBCRXERR	R/W	DBC receive error. When set to 1, the receive packet routing control has detected an error in the DV DBC count of the packet currently being received.

5.9 Extended Interrupt Mask Register (EIMR at Addr 01Ch)

This bit map defines the control bits that enable the application software to selected and enable the status bits in register (IMR at Addr 18h) to generate interrupts to the host processor. Unless otherwise specified the bits in this register are cleared to 0 on powerup or software-initiated reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	Not Used		
01	Reserved		
02	IGRP1EN	R/W	This bit determines if the external interrupt is routed to the external $\overline{\text{INT}}$ terminal. Please Both register 10h and register 18h can be routed to the external $\overline{\text{INT}}$ terminal.
03	ARAVEN	R/W	When set to 1, this bit enables ARAVEN to generate an interrupt
04	IRAVEN	R/W	When set to 1, this bit enables IRAVEN to generate an interrupt
05	Reserved		
06	DRAVEN	R/W	When set to 1, this bit enables DRAVEN to generate an interrupt
07	Reserved		
08	DRELTIMEN	R/W	When set to 1, this bit enables DRELTIMEN to generate an interrupt
09	Reserved		
10	MPUERREN	R/W	When set to 1, this bit enables MPUERREN to generate an interrupt.
11	INVWROPEN	R/W	When set to 1, this bit enables INVWROPEN to generate an interrupt.
12	ARFRABRTEN	R/W	When set to 1, this bit enables ARFRABRTEN to generate an interrupt.
13	CYCTMOUTEN	R/W	When set to 1, this bit enables CYCTMOUTEN to generate an interrupt.
14	IRFABORTEN	R/W	When set to 1, this bit enables IRFABORTEN to generate an interrupt.
15	No Used		
16	MRFABORTEN	R/W	When set to 1, this bit enables MRFABORTEN to generate an interrupt.
17	ACRFPRDYEN	R/W	When set to 1, this bit enables ACRFPRDYEN to generate an interrupt.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
18	ACRFPABRTEN	R/W	When set to 1, this bit enables ACRFPABRTEN to generate an interrupt.
19	BRFPRDYEN	R/W	When set to 1, this bit enables BRFPRDYEN to generate an interrupt.
20	BRFPABRTEN	R/W	When set to 1, this bit enables BRFPABRTEN to generate an interrupt.
21	SIDPRDYEN		When set to 1, this bit enables SIDPRDYEN to generate an interrupt.
22	SIDPABRTEN	R/W	When set to 1, this bit enables SIDPABRTEN to generate an interrupt.
23	Reserved		
24	MPUTMOUTEN	R/W	When set to 1, this bit enables MPUTMOUTEN to generate an interrupt.
25	ISOGOERREN	R/W	When set to 1, this bit enables ISOGOERREN to generate an interrupt.
26	MDDBCERREN	R/W	When set to 1, this bit enables MDDBCERREN to generate an interrupt.
27	Reserved		
28	Reserved		
29	SBACOMPEN	R/W	When set to 1, this bit enables SBACOMPEN to generate an interrupt.
30	BFAFLERREN		When set to 1, this bit enables BFAFLERREN to generate an interrupt.
31	DBCRXERREN	R/W	When set to 1, this bit enables DBCRXERREN to generate an interrupt.

5.10 Isochronous Receive Comparators Register 0 (IRPR0 at Addr 020h)

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

IRPxEN†	MONTx†	COMPARE FUNCTION
0	0	Comparator x is disable
1	0	Match IRPORTx expected value to channel number of incoming isochronous packet
0	1	Comparator disabled
1	1	Match IRPORTx and TAGx expected values to channel number and tag field of incoming isochronous packet

† x = 0,1,2,3

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 01	TAG0	R/W	DV Port 0 Tag field receive packet compare value.
02 – 07	IRPORT0	R/W	DV 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.

5.11 Isochronous Receive Comparators Register 1 (IRPR1 at Addr 024h)

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

IRPxEN†	MONTx†	COMPARE FUNCTION
0	0	Comparator x is disable
1	0	Match IRPORTx value to channel number of incoming isochronous packet
0	1	Comparator disabled
1	1	Match IRPORTx and TAGx value to channel number and tag field of incoming isochronous packet

† x = 4,5,6,7

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.

5.12 Cycle Timer Register (CLKTIM at Addr 028h)

The register provides the application software with an 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 CYCMSTREN, CYCSRC, CYCTIMEN. These bits are located in the Link Control register (LCTRL at Addr 0Ch). Unless other wise specified the cycle timer register is cleared to 0 on powerup or software-initiated reset.

CYCLE TIMER PROGRAM FUNCTION TABLE			
CYCSRC	**CYCMSTREN**	**CYCTIMEN**	
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 transitions from low to high. It must remain high for a minimum of 80ns before bringing it back low. 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 transitions from low to high. It must remain high for a minimum of 80ns before bringing it back low.

CYCLE TIMER REGISTER			
BIT NUMBER	**BIT NAME**	**DIR**	**FUNCTIONAL DESCRIPTION**
00 – 06	CYCSEC	R/W	Cycle seconds. 1 Hz cycle timer counter. This counter increments whenever the CYCNUMBER rolls over from 7999 back to 0.
07 – 19	CYCNUMBER	R/W	Cycle Number: Counts from 0 to 7999 base 10. Increments by 1 whenever the CYCOFFSET rolls over from 3071 to 0.
20 – 31	CYCOFFSET	R/W	Cycle offset. Counts from 0 to 3071 base 10. The time to count from 0 to 3071 is 125 μs.

5.13 Extended Cycle Time Register (EXTTIM at Addr 02Ch)

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 at 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	The CYCSEC field of the cycle timer register.

5.14 Link Diagnostics Register (DIAG at Addr 030h)

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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	ENSNOOP	R/W	When set to 1, 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. When set to 1, the isolation barrier is off. When set to 0, isolation barrier is on. This bit is set to 1 on powerup or software reset.
03	ISOBAROFF	R/W	When set to 1, isolation barrier function is turned off
04	REGRW	R/W	When set to 1 configures read-only register bits to function as read/write registers.
05	ADRCLR	R/W	When set to 1 causes the RAM test address generator to initialize to 0. This bit clears its self to 0.
06	CNTLRBIT1	R/W	When set to 1 the MSB of the test data that is being written into the control FIFO RAM is set to 1
07	CNTLRRBITERR	R/W	When this bit is set to 1, the control bit value from the control FIFO RAM does not match the value of CNTLRBIT1.
08	RAMTEST	R/W	When this bit is set to 1 and the REGRW bit is set to 1, the ram used in the control FIFO can be directly addressed and tested with the FIFO control logic disabled.
09 – 12	STAT3MUXSEL	R/W	STAT3 output internal signal mux select lines. **STAT3MUXSEL** **INTERNAL SIGNAL SELECTED** 4'hF NClk
13 – 16	Reserved		
17 – 20	STAT1MUXSEL	R/W	STAT1 output internal signal mux select lines. **STAT1MUXSEL** **INTERNAL SIGNAL SELECTED** 4'h0 0 4'h1 1
21 – 27	STAT0MUXSEL	R/W	STAT0 output internal signal mux select lines. **STAT0MUXSEL** **INTERNAL SIGNAL SELECTED** 7'h1E BDIBusyOut 7'h1F BDOAvailOut 7'h3B NClk
28 – 31	Reserved		

5.15 Phy Access Register (PHYAR at Addr 034h)

This register provides the application software with an interface for performing read or write accesses to the registers in the Phy layer. Unless otherwise specified the Phy Access control register is cleared to 0 on powerup or software-initiated reset. The functionality of this 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 a 1, a read request is issued to fetch the value of the Phy register specified by the register address contained in the field PHYREGADR. This request bit is cleared to 0 after the LINK sends the request to the Phy.
01	WRPHYREQ	R/W	Write Phy register request. When this bit is set to a 1, a Phy register write request is issued to the Phy that contains the register address and the data to be stored in the register. The Phy register address and the write data are specified in fields PHYREGADR and PHYREGWRDATA. The request bit is cleared to 0 after the LINK has sent the request to the Phy layer.
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 a PHY register specified by the field 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. For diagnostic purposes the host processor can write to these bits when the REGRW bit in link diagnostics register (DIAG at addr 30h) is set to 1. Otherwise these bits are read only.
24 – 31	PHYREGDA-TARCV	R	Phy register data received. These register bits buffer the data returned by the Phy in response to a Phy register read request. For diagnostic purposes the host processor can write to these bits when the REGRW bit in link diagnostics register (DIAG at addr 30h) is set to 1. Otherwise these bits are read only.

5.16 Expected Response (PHYSR at Addr 038h)

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 transaction label and tcode value that was used in previously issued request packet. 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. For diagnostic purposes this register also brings out the state machine vectors for the PHY-LINK interface state machines.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
0 – 15			
16 – 21	EXP_TLABEL	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 1 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		

5.17 Reserved Register (at Addr 03Ch – 040h)

5.18 Asynchronous Control Data Transmit FIFO Status (ACTFS at Addr 044h)

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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	ACTFULL	R	When set to 1, asynchronous control transmit FIFO is full. This means that the FIFO has no quadlets of storage left. The host processor can write and read these bits when the REGRW bit in link diagnostics register (DIAG at addr 30h) is set to 1.
01	ACTALFULL	R	When set to 1, asynchronous control transmit FIFO is almost full. This means that the FIFO has one quadlet of storage left. The host processor can write and read these bits when the REGRW bit in link diagnostics register (DIAG at addr 30h) is set to 1.
02 – 03	Not Used		
04	ACT4AVAIL	R	When set to1, the asynchronous control transmit FIFO has at least 4 empty locations available for storing data. The host processor can write and read these bits when the REGRW bit in link diagnostics register (DIAG at addr 30h) is set to 1.
05 – 13	Not Used		
14	ACTALEMPTY	R	When set to 1, the asynchronous control transmit FIFO is almost empty. This means that the FIFO is one quadlet away from being empty. The host processor can write and read these bits when the REGRW bit in link diagnostics register (DIAG at addr 30h) is set to 1.
15	ACTEMPTY	R	When set to 1, the asynchronous control transmit FIFO is empty. The host processor can write and read these bits when the REGRW bit in Link Diagnostics register (DIAG at addr 30h) is set to 1. This bit is set to 1 (default) On powerup reset, bus reset and software reset.
16	Not Used		
17	ACTCLR	R/W	When set to a 1, the asynchronous control transmit FIFO is flushed. This bit self clears to 0.
18 – 24	Not Used		
25 – 31	ACTSIZE	R/W	Asynchronous control transmit FIFO size setting in quadlets. On powerup or software reset these bits are set to 14h (default).

5.19 Bus Reset Data Register (BRD at Addr 048h)

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 powerup 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 3FFh on powerup 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 powerup/software reset occurs.
16	NRIDV	R	Node count resolver ID valid. This bit is set to a 1 when NODECNT and RESID 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, and software reset.
24	ROOT	R	The root state of the local Phy.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
25	Not used		
26 – 31	RESID	R	The ID of the resolver node. The resolver ID is set to 3Fh on bus reset, powerup reset, and software reset.

5.20 Bus Reset Error Register (BRERR at Addr 04Ch)

This register provides the application software with error status generated by bus reset controlled 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 powerup or software-initiated reset

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	CBRERR	R/W	Clear bus reset controller error. When set to a 1, the BRERRCODE is set to 0000. This bit is self clearing to 0.
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 first quadlet 0100 – phyid incremented by two 0101 – phyid incremented any 3 or more 0110 – phyid not equal in packet 0111 – Self-ID quads are not inverses of each other. 1000 – Self-ID quad is bad
05 – 31	Reserved		

5.21 Asynchronous Control Data Receive FIFO Status (ACRXS at Addr 050h)

This register provides the application software with capability to monitor the occupancy status of the asynchronous control and broadcast control receive FIFOs an 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 link diagnostics register (DIAG at addr 30h) is set to 1.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	ACRFULL	R	When set to 1, the asynchronous control receive FIFO is full. On Bus_reset, Power up reset, and software reset this bit is cleared to 0.
01	BRFFULL	R	When set to 1, the broadcast control receive FIFO is full. On Bus_reset, powerup reset, and software reset this bit is cleared to 0
02	ACRALFULL	R	When set to 1, the Asynchronous Control Receive FIFO is almost full. On Bus_reset, powerup reset, and software reset this bit is cleared to 0
03	BRFALFULL	R	When set to 1, the Broadcast Control Receive FIFO is almost full. On Bus_reset, powerup reset, and software reset this bit is cleared to 0.
04 – 10	Not Used		
11	BRFCD	R	State of the control bit for the last data quadlet read from the broadcast Receive FIFO. On Bus_reset, Power up reset, and software reset this bit is cleared to 0.
12	ACR4AVAIL	R	When set to 1, the asynchronous control receive FIFO has 4 locations available for storage. On Bus_reset, Power up reset, and software reset this bit is cleared to 0.
13	BRFEMPTY	R	When set to 1, broadcast receive FIFO is empty. On Bus_reset, Power up reset, and software reset this bit is set to 1.
14	ACRALEMPTY	R	When set to 1, asynchronous control receive FIFO is almost empty. On Bus_reset, powerup reset, and software reset this bit is set to 0

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
15	ACREMPTY	R	When set to 1, the asynchronous control receive FIFO is empty. On Bus_reset, powerup reset, and software reset this bit is set to 1.
16	ACRCD	R	State of the control bit for the last data quadlet read from the asynchronous control receive FIFO. On Bus_reset, powerup reset, and software reset this bit is set to 0.
17	ACRCLR	R/W	Control receive FIFO clear. When set to 1, the asynchronous receive control and broadcast receive control FIFOs are flushed. This bit is self clearing and is also cleared on powerup or software reset.
18 – 24	BRFSIZE	R/W	Sets the size of the broadcast receive FIFO in quadlets. On powerup or software reset these bit are set to 15h.
25 – 31	ACRSIZE	R/W	Sets the size of the asynchronous receive FIFO in quadlets. On powerup or software reset these bit are set to 15h.

5.22 Read Write Test Register (UCRWTEST at Addr 054h)

This register provides software with the capability to write a data pattern to this address and then read it back for the on-chip microinterface's diagnostic test purpose. The data pattern written does nothing to affect the internal operation of the chip. This register is cleared to all 0s on powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	UCRWBIT	R/W	Rebound test data with bit #0, 8, 16, 24 hard coded to zero. These four bits can be written to by setting the REGRW bit in the link diagnostics register (DIAG at Addr 30h) to 1.

5.23 Reserved Register (at Addr 058h – 07Ch)

5.24 Asynchronous Control Data Transmit FIFO First (ACTXF at Addr 080h)

This write only port provides application software with the capability to write the first quadlet of a packet to the asynchronous 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	Asynchronous control transmit first

5.25 Asynchronous Control Data Transmit FIFO Continue (ACTXC at Addr084h)

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

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

5.26 Asynchronous Control Data Transmit FIFO First and Update (ACTXFU at Addr 088h)

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

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

5.27 Asynchronous Control Data Transmit FIFO Continue and Update (ACTXCU at Addr 08Ch)

This write only port provides application software with the capability to write the last quadlet of a packet to the Asynchronous Control Transmit FIFO and have the entire packet confirmed for transmission.

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

5.28 Reserved Register (at Addr 090h – 0BCh)

5.29 Asynchronous Control Data Receive FIFO (ACRX at Addr 0C0h)

This register port allows read accesses to the ACRX 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 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 all 0s on powerup, bus_reset, and software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	ACRX	R	Asynchronous Control Receive FIFO read port

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

This register port allows accesses to the BWRX 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 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 all 0s on powerup, bus_reset, and software reset.

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

5.31 Reserved Register (at 0C8 – 0D4)

5.32 Bulky Data Interface Control (BIF at Addr 0D8h)

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 powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 07	Not Used		
08	Reserved	R/W	Reserved for future use.
09	BMLECTRL	R/W	DV little endian control. When set to 1, causes DV interface to operate in little endian format.
10	BILECTRL	R/W	Isochronous little endian control. When set to 1, causes isochronous interface to operate in little endian format.
11	BALECTRL	R/W	Asynchronous little endian control. When set to 1, causes asynchronous interface to operate in little endian format.
12	ANBDIBSY	R/W	Active high control for BDIBUSY terminal. When set to 1 causes signal to be active high. This bit is set to 1 on power-up or software reset.
13	ANAVAIL	R/W	Active high control for BDAVAIL terminal. When set to 1 causes signal to be active high. This bit is set to 1 on power-up or software reset.
14	ANBDOEN	R/W	Active high control for BDOEN terminal. When set to 1 causes signal to be active high. This bit is set to 1 on power-up or software reset.
15	ANBDIEN	R/W	Active high control for BDIEN terminal. When set to 1 causes signal to be active high. This bit is set to 1 on power-up or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
16 – 20	Reserved	R/W	Reserved for future use.
21	Not Used		
22	BDOMODE1	R/W	MSB of the BDOMODE select bits
23	BDOMODE0	R/W	LSB of the BDOMODE select bits
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 allows 1394 padding bits through the interface port.
29	BDOINIT	R/W	Self clearing bit. When written to with a 1 causes the BDO logic to reset.
30	BDIINIT	R/W	Self clearing bit. When written to with a 1 causes the BDI logic to reset.
31	BDOTRIS	R/W	When set to 1, causes the BDO data bus to be forced 3-state.

5.33 Transmit Timestamp Offset Register (XTO at Addr 0DCh)

This register provides the application software with the capability to program the time stamp offset for a DV transmit cell or packet. The hardware adds this offset to a sampled value of the cycle timer to determine the time stamp for the cell/packet to be transmitted. Unless otherwise specified the bits in this register are cleared to 0 on powerup or software reset. In DV mode only the 16 least significant bits are needed for the timestamp value, bits 0–15 should be set to 0. If a value greater than FBFFh is written to this register, the hardware defaults to FBFFh.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	Reserved		
16 – 31	XTO	R/W	Sign extended The 16 bit offset value.

5.34 Reserved Register (at Addr 0E0h)

5.35 Receive Timestamp Offset (RTO at Addr 0E4h)

This register provides the application software with the capability to program the time stamp offset for a DV receive cell or packet. The hardware adds this offset to the timestamp of the received cell/packet to determine the time to release the cell/packet to the application. Unless otherwise specified the bits in this register are cleared to 0 on power-up or software reset. In DV mode bits 0–15 are always zero because only the 16 least significant bits are used for timestamp. If a value greater than FBFFh is written to this register, the hardware defaults to FBFFh.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	Reserved		
16 – 31	RTO	R/W	The sign extended 16 bit offset value.

5.36 Reserved Register (at Addr 0E8h)

5.37 Asynchronous/Isochronous Application Data Control Register (AICR at Addr 0ECh)

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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	IRENABLE	R/W	Isochronous receive enable. When set to 1, bulky isochronous receive FIFO is enabled to receive data.
01	ITENABLE	R/W	Isochronous transmit enable. When set to 1, bulky isochronous transmit FIFO is enabled to transmit data.
02	ARENABLE	R/W	Asynchronous receive enable. When set to 1, bulky asynchronous receive FIFO is enabled to receive data.
03	ATENABLE	R/W	Asynchronous transmit enable. When set to 1, bulky asynchronous transmit FIFO is enabled to transmit data. This bit is cleared when 1394 bus reset occurs.
04 – 14	Not Used		
15	ISNOOP	R/W	Isochronous snoop. When set to 1, all incoming isochronous traffic is snooped and stored to the bulky isochronous receive FIFO.
16	IHIM	R/W	Isochronous header insert mode enable. When 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	Reserved		
18	Reserved		
19	IRHS	R/W	Isochronous header strip mode enable. When 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 set to 0: The MP/MC has access to the bulky data receive FIFO. Received isochronous packets are not transferred to the application thru the BDIF. When set to 1: Received isochronous packets are transferred to the application thru the BDIF. MP/MC read accesses are ignored.
21	BDIXE	R/W	Bulky data isochronous transmit FIFO data source select. When set to 0: The MP/MC has write access to the bulky isochronous transmit FIFO. Data writes from the BDIF are ignored. When set to 1: The application has write access to the bulky isochronous transmit FIFO via the BDIF. MP/MC writes are ignored.
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 a 1 flushes the isochronous transmit FIFO. This bit is self clearing.
24	AHIM	R/W	Asynchronous header insert mode enable. When set to 1, automatic header insertion and packetization of asynchronous data from the bulky asynchronous transmit FIFO is enabled. In this mode the hardware expects the application to load the asynchronous transmit FIFO with pure data that contains no header. When this bit is set to 0, the hardware expects the asynchronous transmit FIFO to contain completely formatted 1394 isochronous packets.
25	Reserved		
26	Reserved		

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

5.38 DV Formatter Control Register (DCR at Addr 0F0h)

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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	DREN	R/W	Receive enable. When set to 1 the bulky receive FIFO is enabled to receive packets.
01	DTEN	R/W	Transmit enable. When set to 1 the bulky transmit FIFO is enable for transmitting packets.
02	EPINST	R/W	This bit is valid only for DV mode. Empty packet insert. When set to 1 empty packets are evenly distributed throughout an entire frame. Power on default is 1.
03	H0INST	R/W	H0 insert. This is valid only in DV mode. When set to 1, DIF block H0 is inserted by hardware. Power on default is 1.
04 – 07	Reserved		
08–09	DBCECNT	R/W	Data block counter error threshold. This is valid in DV mode only. If the received packet DBC count is different from the expected value an interrupt is generated and the receiver continues accepting packets, if DBCECNT = 00, the power on default. If DBCECNT is other than ‘00’ and if there is a mismatch, an interrupt is generated and the entire FIFO is flushed and any and all remaining packets are rejected when the number of errors DBC count is equal to DBCECNT. The receiver waits for the next time stamped packet (start of frame) to begin again accepting packets.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
10	RELDATA	R/W	Valid only in DV mode. Release data control bit, controls the reading requirement sequence on the bulky interface. When this bit is 1 the following sequence must be followed to receive data: 1) BDO_Fr must have been activated 2) Followed by BDI_Fr activated 3) Followed by activation of BDOEN If this bit is set to 0, then on activation of BDOEN read data is gated out regardless of BDO_Fr or BDI_Fr. The default value is 0.
11	INTSSP	R/W	Insert timestamp into source packet. When set to 1, the hardware inserts the timestamp only in a full source packet at the beginning of the next frame (not an empty packet). The default value is 0.
12	NTSCPAL	R/W	Valid in DV mode. 1 for NTSC, 0 for PAL. Power on default is 1.
13 – 15	Reserved		
16	DRHS	R/W	Header strip enable. When set to 1 the isochronous, CIP0, and CIP1 headers and trailer quadlet are stripped from the receive packet and copied into buffer registers. The remaining source packet is transferred to the application
17	Reserved		
18–19	MOEN0, MOEN1	R/W	Mode enable bit 0, 1. Code as follows for the desired mode: MOEN0, MOEN1. The value is set to 1b'0 for DV
20 – 23	Reserved		
24	BDDRE	R/W	Bulky data receive FIFO Data Destination Select. When set to 0: The MP/MC has access to the bulky data receive FIFO. received DV packets are not transferred to the application through the BDIF. When set to 1: Received packets are transferred to the application through the BDIF. MP/MC read accesses are ignored.
25	BDXE	R/W	Bulky data transmit FIFO data source select. When set to 0: The MP/MC has write access to the bulky transmit FIFO. Data writes from the BDIF are ignored. When set to 1: The application has write access to the bulky transmit FIFO via the BDIF. MP/MC writes are ignored.
26	DRFLSH	W	Bulky receive FIFO flush. Setting this bit to 1 flushes the receive FIFO. This bit is self clearing.
27	DXFLSH	W	Bulky transmit FIFO flush. Setting this bit to a 1 flushes the transmit FIFO. This bit is self clearing.
28	DVSUB	R/W	DV sub mode select. For Transmit : 0 => Send a full source packet (480 bytes + CIP) 1 => Send only empty packets For Receive: 0 => Save entire source packet to FIFO 1 => Save only CIP and H0 in registers
29	Reserved		
30	Reserved		
31	DHIM	R/W	When set to 1 automatically insert the isochronous and CIP headers on transmits.

5.39 Reserved Register (at Addr 0F4h)

5.40 FIFO Misc (FMISC at Addr 0F8h)

This register provides the application software with the capability to program an alternate cell size for DV 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	BDIFMTIDIS	R/W	BDI format disable. When this bit is set to 1 the drivers to BDIF[2:0] is disabled, putting the lines into a high-impedance state. Power on default is 0.
01 – 04	MPUERRCODE	R	0h – powerup or software reset value 1h – MPU tried to pop an empty Asynchronous Receive FIFO 2h – MPU tried to pop a paused Asynchronous Receive FIFO 3h – MPU tried to push a full Asynchronous Transmit FIFO 4h – MPU tried to pop an empty Isochronous Receive FIFO 5h – MPU tried to pop a paused Isochronous Receive FIFO 6h – MPU tried to push a full Isochronous Transmit FIFO 7h – MPU tried to pop an empty DV Receive FIFO 8h – MPU tried to pop a paused DV Receive FIFO 9h – MPU tried to pop a DV Receive FIFO before time stamp release occurred. BDIF mode. Ah – MPU tried to pop a DV Receive FIFO while in BDIF mode Bh – MPU tried to push a full DV Transmit FIFO Ch – MPU tried to push a DV Transmit FIFO while in BDIF mode
05 – 06	Not Used		
07 – 15	TXMCSZ	R/W	Alternate transmit cell size. Cleared to 0 on powerup or software reset.
16	INITXTSEL	R/W	Transmit FIFO threshold select. Valid in DV mode only. When set to 0, the first DV source packet following power-on reset is transmitted only after the FIFO has been filled to (480+N) bytes of data. N is decoded in bits 18 and 19 of this register. When set to 1, every DV source packet will be transmitted only if the FIFO has accumulated (480+N) bytes of data. The value of N is decoded in bits 18 and 19 of this register. Power on default is 0.
17	Not Used		
18 – 19	XTSEL0, XTSEL1	R/W	These bits are used in conjunction with bit 16, INITXTSEL. Valid only in DV mode. These bits are used to decode an addition to the transmit threshold byte count. The decode are as follows: XTSEL0, XTSEL1 00 => 0 Bytes 01 => 128 Bytes 10 => 256 Bytes 11 => 384 Bytes The power on default is ‘00’.
20 – 22	Not Used		
23 – 31	RXMCSZ	R/W	Alternate receive cell size. Cleared to 0 on power-up or software reset.

5.41 Reserved Register (at Addr 0FCh – 100h)

5.42 Bulky A Size Register (BASZ at Addr 104h)

The register provides the application software with the capability to program the size in multiples of 4 quadlets, of the bulky asynchronous 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	BATXSIZE	R/W	Bulky asynchronous transmit FIFO size in multiples of 4 quadlets. Bit 06 is MSB.
16 –21	Not Used		
22 – 31	BARXSIZE	R/W	Bulky asynchronous receive FIFO size in multiples of 4 quadlets. Bit 22 is MSB.

5.43 Bulky A Avail Register (BAAVAL at Addr 108h)

This read-only register provides the application software with the capability to read the occupancy status in quadlets for the bulky asynchronous transmit and receive FIFOs. This register is cleared to 0 on a powerup 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 asynchronous transmit FIFO. Bit 04 is MSB. Value returned on a read performed right 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 asynchronous receive FIFO. Bit 20 is MSB. Value returned on a read performed right after a power-up or software reset is 0.

5.44 Asynchronous Application Data Transmit FIFO First and Continue (BATXFC at Addr 10Ch)

This write only port provides the application software with the capability to write the quadlets of a asynchronous transmit packet (except the last quadlet) to the bulky asynchronous transmit FIFO.

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

5.45 Asynchronous Application Data Transmit FIFO Last and Send (BATXLS at Addr 110h)

This write only port provides the application software with the capability to write the last quadlet of an asynchronous transmit packet to the bulky asynchronous 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

5.46 Asynchronous Application Data Receive FIFO (BARX at Addr 114h)

This read-only port allows the application software to read data from the bulky asynchronous 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 MSB

5.47 Asynchronous Application Data Receive Header Register 0 (ARH0 at Addr 118h)

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

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

5.48 Asynchronous Application Data Receive Header Register 1 (ARH1 at Addr 11Ch)

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

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

5.49 Asynchronous Application Data Receive Header Register 2 (ARH2 at Addr 120h)

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

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

5.50 Asynchronous Application Data Receive Header Register 3 (ARH3 at Addr 124h)

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

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

5.51 Asynchronous Application Data Receive Trailer (ART at Addr 128h)

This read-only register allows the application software to read the trailer quadlet of a received asynchronous packet after the bulky asynchronous receive FIFO control logic has copied the trailer quadlet to this register. This register is cleared to 0 on powerup 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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
24 – 27	Not Used		
28 – 31	ACKSENT	R	The 1394 ack that was sent by the link receiver after receiving the packet. 0000 – Reserved 0001 – Ack complete 0010 – Ack pending 0011 – Reserved 0100 – Ack busy_X 0101 – Ack busy_A 0110 – Ack busy_B 0111 – 1100 – reserved 1101 – Ack data error 1110 – Ack type error 1111 – Reserved

5.52 Bulky Isochronous Size Register (BISZ at Addr 12Ch)

The 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 powerup 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.

5.53 Bulky Isochronous Avail Register (BIAVAL at Addr 130h)

The 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 powerup 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

5.54 Isochronous Transmit First and Continue (BITXFC at 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

5.55 Isochronous Transmit Last and Send (BITXLS at 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

5.56 Isochronous Receive FIFO (BIRX at 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 powerup or software reset is 0.

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

5.57 Isochronous Packed Received Header (IRH at 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 powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	Data Length	R	Packet data length in bytes
16 – 17	TAG	R	Isochronous TAG field
18 – 23	Channel Number	R	Isochronous channel number
24 – 27	TCODE	R	Isochronous tCode field
28 – 31	Sy	R	Isochronous sync bits

5.58 Isochronous Packet Received Trailer (IRT at 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**.** This register is cleared to 0 on powerup 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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
24 – 27	Not Used		
28 – 31	ERRLODE	R	The 1394 ack that was sent by the link receiver after receiving the packet. 0000 – Reserved 0001 – Ack complete 0010 – Ack pending 0011 – Reserved 0100 – Ack busy_X 0101 – Ack busy_A 0110 – Ack busy_B 0111 – 1100 – reserved 1101 – Ack data error 1110 – Ack type error 1111 – Reserved

5.59 Receive Packet Router (RMISC at 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 powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 14	Not Used		
15	RSTONFP	R/W	Reset on frame pulse is valid only in DV mode. When set to 1 the DV empty packet insertion logic is reset on detection of a BDI_Fr. The power on default is 0.
16	Not Used		
17	BACKPND	R/W	Bulky Ack Pending. When this bit is set to 1, an ack pending is sent in response to an asynchronous write or lock request packet. When set to 0, an ack complete is sent. If an asynchronous packet is not received correctly, then it is acknowledged with ack data error regardless of BACKPND value. The poweron default is 1.
18	Not Used		
19	DBCCOV-ER	W	Disable Data block continuity checking on receive. When set to 1, data block continuity checking of incoming DV packets by the receive routing control logic is disabled.
20	RIDM0	R/W	Receive FIFO destination select for response packets that are matched by the expected response comparator (PHYSR at addr 38h) RIDM0 = 0 The expected response packet (tcode/tlabel in PHYSR register) is routed to BARX FIFO. RIDM0 = 1 The expected response packet is routed to the ACRX FIFO.
21	SIDM0	R/W	Self-ID Receive FIFO Destination Select. SIDM0 = 0 Route self-ID packets to broadcast receive FIFO. SIDM0 = 1 Route self-ID packets to Bulky Asynchronous Receive FIFO. Power up default = 1.
22	ARDM0	R/W	Asynchronous receive FIFO destination select bits

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
23	ARDM1		ACRX = Asynchronous control receive FIFO BWRX = Broadcast control receive FIFO BARX = Bulky data asynchronous receive FIFO **ARDM1 ARDM0 Routing Function Selected** 0 0 All non-broadcast asynchronous packets are routed to the ACRX. All broadcast asynchronous packets are routed to the BWR.
			0 1 Non-broadcast ROM/register space request asynchronous packets are routed to the ACRX. Broadcast ROM/register space request asynchronous packets are routed to BWRX. All other asynchronous packets not meeting the above decode criteria are routed to the BARX.
			1 0 All asynchronous packets are routed to the BDARF.
			1 1 Non-broadcast asynchronous request packets with addr => 48 bit destination address threshold are routed to the ACRX. Broadcast asynchronous request packets with addr => 48 bit destination address threshold are routed to the BWRX. All other asynchronous packets not meeting the above criteria are routed to the BDARF.
24	MONT0	R/W	When set to 1 enable match on tag compare for DV ISO receive comparator 0
25	MONT1	R/W	When set to 1 enable match on tag compare for ISO receive comparator 1
26	MONT2	R/W	When set to 1 enable match on tag compare for ISO receive comparator 2
27	MONT3	R/W	When set to 1 enable match on tag compare for ISO receive comparator 3
28	MONT4	R/W	When set to 1 enable match on tag compare for ISO receive comparator 4
29	MONT5	R/W	When set to 1 enable match on tag compare for ISO receive comparator 5
30	MONT6	R/W	When set to 1 enable match on tag compare for ISO receive comparator 6
31	MONT7	R/W	When set to 1 enable match on tag compare for ISO receive comparator 7

5.60 Bulky Asynchronous Retry (BARTRY at 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 asynchronous transmit FIFO. The cycle timer must be enabled for automatic retries. This register is cleared to 0 on powerup 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 retries
24 – 31	BATXRTRYNUM	R/W	Number of times to retry the asynchronous packet when the receiving node continues to ack the packet with a busy acknowledge.

5.61 Bulky DV Size Register (BDSZ at Addr 150h)

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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 05	Not Used		
06 – 15	BDTXSIZE	R/W	Bulky DV Transmit FIFO size in multiples of 4 quadlets. Bit 06 is MSB.
16 – 21	Not Used		
22 – 31	BDRXSIZE	R/W	Bulky DV Receive FIFO size in multiples of 4 quadlets. Bit 22 is MSB.

5.62 Bulky DV Avail Register (BDAVAL at Addr 154h)

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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 03	Not Used		
04 – 15	BDTXAVAIL	R	Number of empty quadlet locations available in the Bulky DV Transmit FIFO. Bit 04 is MSB
16 – 19	Not Used		
20 – 31	BDRXAVAIL	R	Number of data quadlets available in the Bulky DV Receive FIFO. Bit 20 is MSB

5.63 Reserved Register (at Addr 158h)

5.64 Reserved Register (at Addr 15Ch)

5.65 DV Transmit FIFO First and Continue (BDTXFC at Addr 160h)

This write only port provides the application software with the capability to write the quadlets of an DV transmit packet (except the last quadlet) to the bulky transmit FIFO.

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

5.66 DV Transmit FIFO Last & Send (BDTXLS at Addr 164h)

This write only port provides the application software with the capability to write the last quadlet of an transmit packet to the Bulky DV 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

5.67 DV Formatted Packet Receive FIFO (BDRX at Addr 168h)

This read-only register port allows the application software access to the Bulky 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 powerup or software reset.

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

5.68 Reserved Register (at Addr 16Ch)

5.69 Reserved Register (at Addr 170h)

5.70 Reserved Register (at Addr 174h)

5.71 DV Receive Header (DRH at Addr 178h)

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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	Data Length	R	Packet data length
16 – 17	TAG	R	Isochronous TAG field
18 – 23	Channel Number	R	Isochronous channel number
24 – 27	tcode	R	Isochronous tCode field
28 – 31	Sy	R	Isochronous sync bits

5.72 DV CIP Receive Header 0 (DCIPR0 at Addr 17Ch)

This read-only register port allows the application software to read the CIP0 header quadlet of a received DV packet after the bulky data FIFO control logic has copied the CIP0 header into register DCIPR0**.** This register is cleared to 0 on powerup or software reset.

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
16 – 17	FN	R	Fraction number
18 – 20	QPC	R	Quadlet padding count
21	SPH	R	source packet header
22 – 23	RES	R	Reserved
24 – 31	DBC	R	Data block continuity counter

5.73 DV CIP Receive Header 1 (DCIPR1 at Addr 180h)

This read-only register port allows the application software to read the CIP1 header quadlet of a received DV packet after the bulky data FIFO control logic has copied the CIP1 header into register DCIPR1**.** This register is cleared to 0 on powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	1	R	Logic 1
01	0	R	Logic 0
02 – 07	FMT	R	Format ID
08 – 31	FDF	R	Format dependent field: DV: 08 – 50/60 field 09–13 – Stype 14–15 – 00 16–31 – SYT

5.74 DV Receive Trailer Register (DRT at Addr 184h)

This read-only register port allows the application software to read the trailer quadlet of a received DV packet after the bulky receive FIFO control logic has copied the trailer quadlet to this register**.** This register is cleared to 0 on powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 13	Not Used		
14 – 15	SPD	R	Receive packet speed 00 – 100 Mbits/s 01 – 200 Mbits/s 10 – invalid 11 – invalid

16 – 27	Not Used		
28 – 31	ERRCODE	R	Receive packet error status.

5.75 Reserved Register (at Addr 188h – 194h)

5.76 DV Receive Cell Header Register 0 (DRX0 at Addr 198h)

In DV mode this register represents bytes 0 – 3 of DIF block H0. This read-only register port allows the application software to read the DV source packet H0 bytes 0 – 3 of a received DV packet after the bulky DV receive FIFO control logic has copied the bytes to this register**.** This register is cleared to 0 on powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00–07	ID0	R	Byte 0 of DIF block H0
08–15	ID1	R	Byte 1 of DIF block H0
16–23	ID2	R	Byte 2 of DIF block H0
24–31	H0R3	R	Byte 3 of DIF block H0

5.77 DV Receive Cell Header Register 1 (DRX1 at Addr 19Ch)

In DV mode this register represents bytes 4 – 7 of DIF block H0. This read-only register port allows the application software to read the DV source packet H0 bytes 4 – 7 of a received DV packet after the bulky DV receive FIFO control logic has copied the bytes to this register**.** This register is cleared to 0 on powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00–07	H0R4	R	Byte 4 of DIF block H0
08–15	H0R5	R	Byte 5 of DIF block H0
16–23	H0R6	R	Byte 6 of DIF block H0
24–31	H0R7	R	Byte 7 of DIF block H0

5.78 Reserved Register (at Addr 1A0h)

5.79 DV Transmit Cell Header Register 0 (DTX0 at Addr 1A4h)

In DV mode these registers provides the application software with the capability to program DIF block H0 bytes 0 – 3. If bit H0INST at F0h is set to 1 then on each source packet transmit that contains a H0 DIF block, the hardware is automatically inserts these registers.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00–07	ID0	R/W	Byte 0 of DIF block H0
08–15	ID1	R/W	Byte 1 of DIF block H0
16–23	ID2	R/W	Byte 2 of DIF block H0
24–31	H0R3	R/W	Byte 3 of DIF block H0

5.80 DV Transmit Cell Header Register 1 (DTX1 at Addr 1A8h)

In DV mode these registers provides the application software with the capability to program DIF block H0 bytes 4–7. If bit H0INST at hF0 is set to 1 then on each source packet transmit that contains a H0 DIF block, the hardware automatically inserts these registers.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00–07	H0R4	R/W	Byte 4 of DIF block H0
08–15	H0R5	R/W	Byte 5 of DIF block H0
16–23	H0R6	R/W	Byte 6 of DIF block H0
24–31	H0R7	R/W	Byte 7 of DIF block H0

5.81 Reserved Register (at Addr 1ACh)

5.82 Asynchronous Header 0 for Auto Transmit (AHEAD 0) at Addr 1B0h)

This register provides the application software with the capability to program the first quadlet of an asynchronous header that is used during asynchronous transmit auto packetization. Please reference Section 3.6 for more information on asynchronous header format.

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. Cleared to 0 on powerup or software reset.
1 – 13	RESERVED		Always zero on read
14 – 31	AHEAD0	R/W	Asynchronous header register 0 used for Auto-A packetization Set to 0001_0010h on powerup or software reset.

5.83 Asynchronous Header 1 for Auto Transmit (AHEAD1) at Addr 1B4h)

This register provides the application software with the capability to program the second quadlet of an asynchronous header that is used during asynchronous transmit auto packetization. Please reference Section 3.6 for more information on asynchronous header format.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	AHEAD1	R/W	Asynchronous header register 1 used for Auto-A packetization Set to 3FC1_0000h on powerup or software reset

This register provides the application software with the capability to program the third quadlet of an asynchronous header that is used during asynchronous transmit auto packetization. Please reference Section 3.6 for more information on asynchronous header format.

5.84 Asynchronous Header 2 for Auto Transmit (AHEAD2) at Addr 1B8h)

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	AHEAD2	R/W	Asynchronous header register 2 used for Auto-A packetization Set to 0000_0000h on powerup or software reset.

5.85 Asynchronous Header 3 for Auto Transmit (AHEAD3) at Addr 1BCh)

This register provides the application software with the capability to program the third quadlet of an asynchronous header that is used during asynchronous transmit auto packetization. Please reference Section 3.6 for more information on asynchronous header format.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 31	AHEAD3	R/W	Asynchronous header register 3 used for Auto-A packetization Set to 0008_0000h on powerup or software reset.

5.86 Isochronous Header for Auto Transmit (IHEAD0 at 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	Data Length	R/W	Packet data length in bytes
16 – 17	TAG	R/W	Isochronous TAG field
18 – 23	Channel Number	R/W	Isochronous channel number
24 – 25	Unused		
26 – 27	Spd	R/W	Transmit speed code. 00 = 100 mbps 01 = 200 mbps
28 – 31	Sy	R/W	Isochronous sync bits

5.87 Packetizer Control (PKTCTL at 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	CYCTSER-EN	R/W	Cycle timer serial output mode. Enabled when one. When enabled, MDAlt register used for serial cycle timer shift register. Set to 1 on powerup or software reset.
01	Unused		
02	DTEST	R/W	DV packetizer test mode. Enabled when one. When enabled, sends DV test packets. DV data is an incrementing count. One DV packet sent per isochronous cycle. Cleared to 0 on powerup or software reset.
03	ITEST	R/W	Bulky isochronous packetizer test mode. Enabled when one. When enabled, 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. Cleared to 0 on powerup or software reset.
04	ATEST	R/W	Bulky asynch packetizer test mode. Enabled when one. When enabled, sends bulky asynch test packets based on Ahead registers. Bulky ssynch data is an incrementing count. No auto retries allowed. When in bulky asynch test mode, one asynch packet is sent per isochronous cycle to prevent continuous asynch packets. Cleared to 0 on powerup or software reset.
05 – 17	Reserved		
18	MDCTFRRG	R/W	DV packetizer control taken from Mdalt when enabled with a one. 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. Cleared to 0 on powerup or software reset.
19	FIFOFLEN	R/W	Master FIFO flush enabled when one. Set to 1 on powerup or software reset.
20	ISOGO-FLEN	R/W	Enable flushing of DV FIFO when the isochronous cycle has begun and the isochronous state machine is in a non-idle state. Set to 1 on powerup or software reset.
21	FIFOPHSEN	R/W	Enable flushing of DV FIFO when the packetizer expects a timestamp from FIFO but the TimeStampValid signal is false. Set to 1 on powerup or software reset.
22	CFRPKTRST	R/W	CFR reset of the packetizer state machines. This is a self clearing bit. Cleared to 0 on powerup or software reset.
23	QPCWEN	R/W	When set to 1, Enable microprocessor writing of quadlet per cell register. Cleared to 0 on powerup or software reset.
24	PRBWEN	R/W	When set to 1, Enable microprocessor writing of PktTestMuxOutAccess register for r/w test. Cleared to 0 on powerup or software reset.
25	FMTWEN	R/W	When set to 1, Enable microprocessor writing of CIP1 Fmt field. Cleared to 0 on powerup or software reset.
26	DBCWEN	R/W	When set to 1, Enable microprocessor writing of CIP0 DBC field. Cleared to 0 on powerup or software reset.
27	SPHWEN	R/W	When set to 1, Enable microprocessor writing of CIP0 SPH field. Cleared to 0 on powerup or software reset.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
28	FNWEN	R/W	When set to 1, Enable microprocessor writing of CIP0 Fn field. Cleared to 0 on powerup or software reset.
29	DBSWEN	R/W	When set to 1, Enable microprocessor writing of CIP0 DBS field. Cleared to 0 on powerup or software reset.
30	SLDWEN	R/W	When set to 1, Enable microprocessor writing of CIP0 SID field. Cleared to 0 on powerup or software reset.
31	LENWEN	R/W	When set to 1, Enable microprocessor writing of DXH DataLength field. Cleared to 0 on powerup or software reset.

5.88 DV Transmit Header Register (DXH at Addr 1C8h)

This register provides the application software with the capability to program the 1394 header quadlet that is used during transmit auto packetization.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 15	DATA LENGTH	R/W	Packet data length, in bytes. When LENWEN at Addr 1C4h = 1, microinterface must load. When LENWEN=0, AutoLoaded by packetizer. Unaffected by BusReset. Set to 0008h on powerup or software reset.
16:17	TAG	R/W	TAG Number. Microinterface must load. Unaffected by BusReset Set to 01 on powerup or software reset.
18:23	chanNum	R/W	Isochronous channel number. Microinterface must load. Unaffected by BusReset
24:25	RESERVED		Logic 0
26:27	spd	R/W	Isochronous speed to send this packet. The microinterface must load. Unaffected by BusReset. Cleared to 00 on powerup or software reset.
28:31	sy	R/W	Isochronous synchronous bits. Resets to 0h. Microinterface must load. Unaffected by BusReset

5.89 DV CIP Transmit Header 0 (DCIPX0 at Addr 1CCh)

This register provides the application software with the capability to program the CIP0 header quadlet that is used during transmit auto packetization.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	0	R/W	Logic 0
01	0	R/W	Logic 0
02 – 07	SID	R/W	SLDWEN = 1, Microinterface must load. SLDWEN=0 (bits 23 – 30 of 1C4h), Auto-Updated with all Phy register 0 transfers.
08 – 15	DBS	R/W	DBSWEN=1 (78h), Microinterface must load. DBSWEN=0, Defaults to 120. Unaffected by BusReset
16 – 17	FN	R/W	FNWEN=1, microinterface must load. FNWEN=0, Defaults to 0 (DV). Unaffected by BusReset
18 – 20	QPC	R/W	Microinterface must load. Unaffected by BusReset. Default 0
21	SPH	R/W	SPHWEN = 1, microinterface must load. SPHWEN=0, Auto set to 1 when TS included in the Packet. For DV always 0. Unaffected by BusReset
22 – 23	RES	R/W	Logic 0
24 – 31	DBC	R/W	DBCWEN, microinterface must load. DBCWEN=0, AutoLoaded/AutoIncremented by packetizer. Unaffected by BusReset.

5.90 DV CIP Transmit Header 1 (DCIPX1 at Addr 1D0h)

This register provides the application software with the capability to program the CIP1 header quadlet that is used during transmit auto packetization. In DV mode, if automatic header insert mode is on then the transmit frame system is decoded from bit 8 (50/60) of this register.

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00	1	R/W	Default 1
01	0	R/W	Default 0
02 – 07	FMT	R/W	FMTWEN=1, microinterface must load. FMTWEN=0, Defaults to 00. Unaffected by BusReset Power on default 20h.
08 – 31	FDF	R/W	Microinterface must load. DV: 08 – 50/60 field (0 => NTSC, 1=> PAL) 09–13 – Stype 14–15 – 00 16–31 – SYT Unaffected by BusReset. Power on default h00_0000.

5.91 Reserved Register (at Addr 1D4h)

5.92 Reserved Register(at Addr 1D8h)

5.93 Reserved Register (at Addr 1DCh)

5.94 MDAltCont (MDALT at 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 powerup 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.

5.95 Reserved Register (at Addr 1E4h)

5.96 Reserved Register (at Addr 1E8h)

5.97 Microinterface Input/Output Control Register (IOCR at Addr 1ECh)

BIT NUMBER	BIT NAME		R/W	BIT VALUE SETTING MEANING		POWER UP DEFAULT SETTING		
	SYMBOL	DESCRIPTION		VALUE = 1	VALUE = 0	Simba	Mot68000	Intel8051
0	MCMP8	Microinter-face bus is 8/16 bit access	R/W	Byte access	Word access	Word access		Byte access
1	BeCtl	Big endian control	R/W	Big Endian	Little endian	Big endian		Little endian
2	IntPol	Interrupt polarity control	R/W	High true	Low true	Low true		

5.97 Microinterface Input/Output Control Register (IOCR at Addr 1ECh) (Continued)

BIT NUMBER	BIT NAME		R/W	BIT VALUE SETTING MEANING		POWER UP DEFAULT SETTING		
	SYMBOL	DESCRIPTION		VALUE = 1	VALUE = 0	Simba	Mot68000	Intel8051
3	RdyPushPull	RDY output signal control	R/W	Active push/pull	High-impedance state	High-impedance state		N/A†
4	IntPushPull	INT output signal control	R/W	Active push/pull	High-impedance state	Active push/pull	3-Sate	
5	BlindAccess	Blind access enable/disable	R	Enable blind access mode	Disable blind access mode	Enable blind access mode	Disable blind access mode	Enable blind access mode
6	DataInvarnt‡	Data invariant endianess control	R/W	Data invariant	Address invariant	Data invariant		
7	RdyPol	Rdy output polarity control	R/W	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.							

† Although there is no RDY line connection in Intel 8051 Mode, reading the IOCR.RdyPushPull still returns a value of 0 for this bit.

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

5.98 Blind Access Status Register (BASTAT at Addr 1F0h)

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

BIT NUMBER	BIT NAME	DIR	FUNCTIONAL DESCRIPTION
00 – 06	0	R	Logic 0
07	BACMP	R	When set to 1 means 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's function
16 – 22	0	R	Logic 0
23	BACMP	R	Mirror bit #7's function
24 – 30	0	R	Logic 0
31	BACMP	R	Mirror bit #7's function

5.99 Blind Access Holding Register (BAHR at 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 powerup or software reset.

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

5.100 Reserved Register (at Addr 1F8h)

5.101 Software Reset Register (SRES at Addr 1FCh)

Any write access to this register generates a device reset regardless what kind of data is written into.

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} ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1) . . . . . ±20 mA
Output clamp current, I_{OK} (TTL/LVCMOS) ($V_O < 0$ or $V_O > V_{CC}$) (see Note 2) . . . . . ±20 mA
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 applies to external input and bidirectional buffers. For 5-V tolerant terminals, use $V_I > V_{CC}5V$.
2. This applies to external output and bidirectional buffers. For 5-V tolerant terminals, use $V_O > V_{CC}5V$.

MAXIMUM DISSIPATION RATING TABLE

PACKAGE	$T_A \leq 25°C$ POWER RATING	DERATING FACTOR ABOVE $T_A = 25°C$	$T_A = 70°C$ POWER RATING
PZ	1500 mW	16.9 mW/°C	737 mW

6.2 Recommended Operating Conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	3	3.3	3.6	V
Supply voltage, $V_{CC}5V$	3	4.5	5.5	V
Input voltage, V_I	0		$V_{CC}5V$	V
Output voltage, V_O†	0		V_{CC}	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	$V_I = V_{IL}$				–20	µA
I_{IH}	High-level input current	$V_I = V_{IH}$				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 BD7 – BDI0, TDO, BDO7 – BDO0, STAT0 – STAT3, BDOF7 – BDOF0, and MCAD0 – MCAD15, RDY, INT_Z, BDO_FR, BDIF2 – BDIF0.

6.4 DVLynx Power

The maximum average power dissipated is 603 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 can be calculate using the following formulae.

Power of one 2K-byte RAM , P1 = [109.6 (rd1) + 92.4 (wr1) + 15.7 (1–rd1–wr1)] (0.000648) Watt

Power of single RAM, P2 = [37.6 (rd1) + 27.7 (wr1) + 8.8 (1–rd1–wr1)] (0.000648) Watt

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

Worst case value would be rd1=1, wr1=0 : P1 = 71.0 mW/RAM, P2 = 24.4 mW

Best case value would be rd1=0, wr1=0 : P1 = 10.2 mW/RAM, P2 = 5.7 mW

For rd1 = 0.3, wr1 = 0.3 : P1 = 43.3 mW/RAM, P2 = 15.0 mW

For an application where the reads and writes to the bulky data FIFO are equal and the control FIFO is actively reading and writing only 10% of the time each (rd1=wr1=0.1), the total maximum power consumed would be 603 mW + (43.3 + 43.3 + 43.3 +43.3) mW + 8.8 mW = 785 mW.

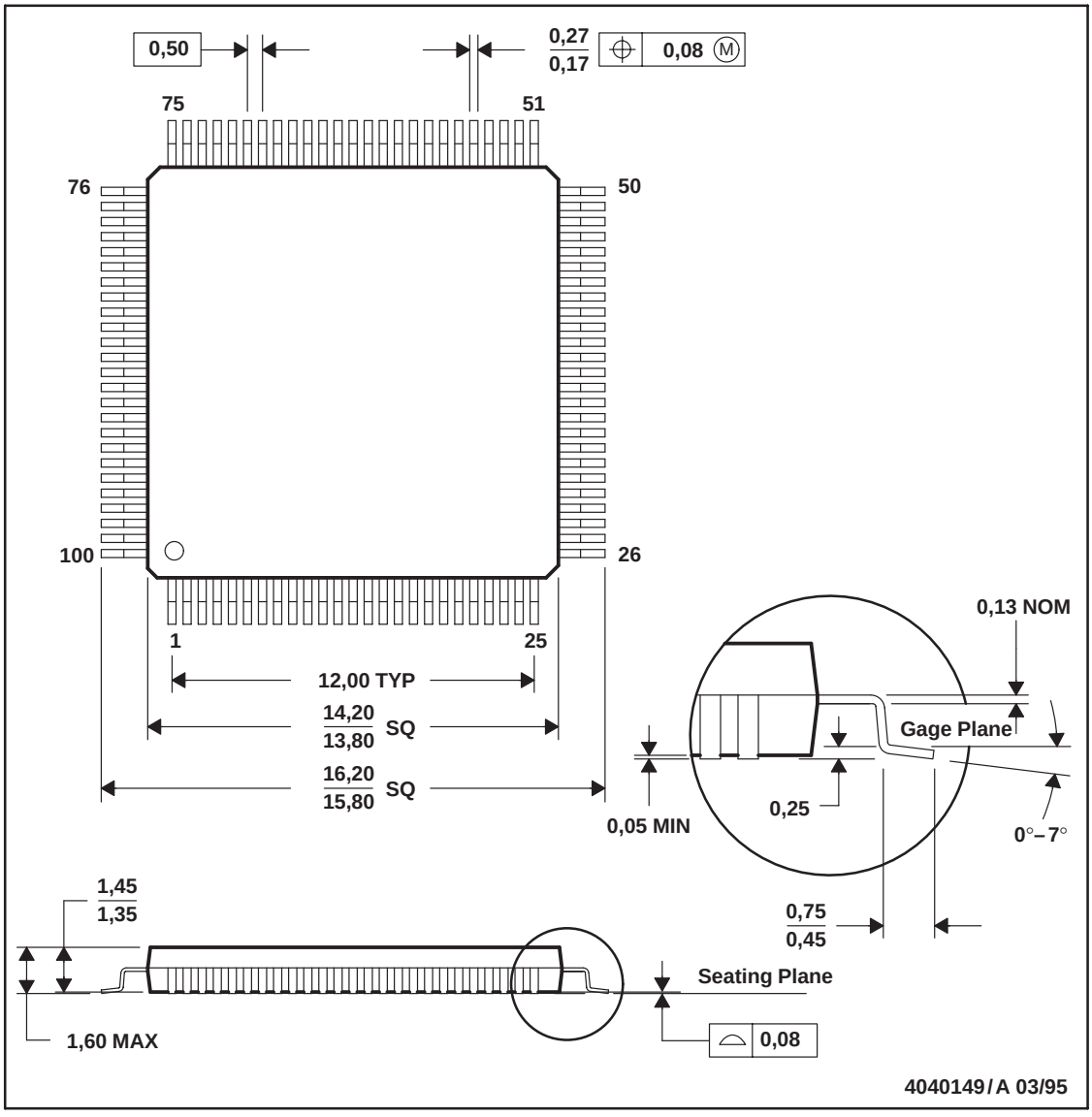
The test condition for measuring the consumed power is as follows:

The microinterface writes 50 quadlets to the bulky receive FIFO using the test modes of operation. This is followed by a read of the FIFO through the bulky port (BDO). This operation continually loops. Independently, the bulky transmit FIFO is loaded with the DV source packet data (480 bytes) through the bulky data input port (BDI). This is followed by a 1394 transmit to the Phy-Link interface. The CYCLEIN is used as the clock for each isochronous cycle and thus is going as fast as possible. The simulation is run for three isochronous cycles.

7 Mechanical Information

The TSB12LV42 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 3.4.1 of the data sheet. These examples are **not** the register settings for every system. Instead, they should be used as a guideline for configuring the DVLynx 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 DVLynx 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

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 (Microprocessor Port)

This example shows how to set up the DVLynx 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 FIFO. All non-broadcast asynchronous packets are received at the asynchronous control receive FIFO.

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 1515h	This sets the asynchronous control receive FIFO to 84 bytes (decimal) It also sets the broadcast receive FIFO to 84 bytes.
Register 148 (Receive Packet Router)	0000 0400h	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 BARX FIFO.
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 Unformatted Isochronous Receive

Receiving isochronous packets is discussed in detail in section 3.4.2 of the data sheet. These examples are **not** the register settings for every system. Instead, they should be used as a guideline for configuring the DVLynx 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 DVLynx cycle master, you must set the appropriate bits in registers Ch and 34h. For a system using TSB21LV03A PHY and DVLynx, an example of these register settings would be:

Register Ch (Link Control Register) = C407 0AC0h

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

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 0AC0h	This register sets the DVLynx to try to become cycle master AND selects ports 0 and 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 DVLynx to receive all isochronous data (including headers) 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

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 0AC0h	This register sets the DVLynx to try to become cycle master AND selects ports 0 and 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 DVLynx 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
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.3 DV Receive

Receiving DV formatted isochronous packets is discussed in detail in section 3.4.3 of the data sheet. These examples are **not** the register settings for every system. Instead, they should be used as a guideline for configuring the DVLynx to meet an individual system’s needs.

DV data can ONLY be received at port 0. Therefore register Ch must have a format similar to:

Register C (Link Control Register) = C407 0A80h.

The same requirement for a cycle master on the bus exists as for isochronous receives.

A.3.1 Receiving DV Data to the Bulky DV FIFO (Bulky Data Interface)

Table A–5. Receiving DV Data to the Bulky DV FIFO

Register Name/Number	Setting	Explanation
Register 150 (Bulky Size Register)	0000 00C0h	This sets the bulky DV receive register to 3072 bytes (decimal).
Register C (Link Control Register)	C407 0A80h	This register sets the DVLynx to try to become cycle master AND selects port 0 for receiving.
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)	4000 0000h	This selects 1 for the tag value and 0 for the channel number on receive compare.
Register 148 (Receive Packet Router)	0000 0080h	Match received data on TAG at port 0.
Register F0 (Formatter Control Register)	8008 2080h	This sets the DVLynx to receive all NTSC formatted DV data (including all headers) 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.3.2 Receiving DV Data to the Bulky DV FIFO (Microprocessor Interface)

Table A–6. Receiving DV Data to the Bulky DV FIFO

Register Name/Number	Setting	Explanation
Register 150 (Bulky Size Register)	0000 00C0h	This sets the bulky DV receive register to 3072 bytes (decimal).
Register C (Link Control Register)	C407 0A80h	This register sets the DVLynx to try to become cycle master AND selects port 0 for receiving.
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)	4000 0000h	This selects “1” for the tag value and “0” for the channel number on receive compare.
Register 148 (Receive Packet Router)	0000 0080h	Match received data on TAG at port 0.
Register F0 (Formatter Control Register)	8008 A000h	This register sets the DVLynx to receive NTSC formatted DV data at the microprocessor interface. Headers are stripped before data is placed in the FIFO.
Register 168 (Formatted Packet Receive FIFO)		This read only register allows the application software to read data out of the bulky DV 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
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.

Appendix B
Transmit Operation Examples

B.1 Asynchronous Transmit

Transmitting asynchronous packets is discussed in detail in section 3.3.1 of the data sheet. These examples are **not** the register settings for every system. Instead, they should be used as a guideline for configuring the DVLynx to meet an individual system's needs.

For all transmissions, the transmit enable (TXEN, bit 00 register C) MUST be set.

B.1.1 Transmitting Asynchronous Data Packets (Bulky Data Interface)

This example shows how to setup DVLynx registers to transmit asynchronous data packets via the bulky data interface. The DVLynx automatically inserts the headers.

The asynchronous headers for transmit are fully explained in section 3.6. The headers provided below are only examples and may not work for all systems.

Table B–1. Transmitting Asynchronous Data Packets

Register Name/Number	Setting	Explanation
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 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 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.1.2 Transmitting Asynchronous Control Packets

This example shows how to setup the DVLynx 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 3.3.2 of the data sheet. 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 receive 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
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
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 Unformatted Isochronous Transmit

Transmitting isochronous packets is discussed in detail in section 3.3.4 of the data sheet. These examples are **not** the register settings for every system. Instead, they should be used as a guideline for configuring the DVLynx 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 DVLynx cycle master, you must set the appropriate bits in registers Ch and 34h. For a system using TSB21LV03A PHY and DVLynx, 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 3.7 of the data sheet. 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

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 DVLynx 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

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 DVLynx 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 DVLynx to transmit fully formatted isochronous data from the Microprocessor Interface.
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.
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 also confirmed for transmission.
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.3 DV Transmit

Transmitting asynchronous packets is discussed in detail in section 3.3.5 of the data sheet. These examples are **not** the register settings for every system. Instead, they should be used as a guideline for configuring the DVLynx 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 DVLynx cycle master, you must set the appropriate bits in registers Ch and 34h. For a system using TSB21LV03A PHY and DVLynx, an example of these register settings would be:

Register Ch (Link Control Register) = C407 0A00h

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

B.3.1 Transmitting DV Data from bulky data interface, Headers Auto-Inserted

Table B–5. Transmitting DV Data from bulky data interface, Headers Auto-Inserted

Register Name/Number	Setting	Explanation
Register 150 (Bulky Size Register)	00C0 0000h	This sets the bulky DV transmit register to 3072 bytes (decimal).
Register C (Link Control Register)	C407 0A00	This register sets the DVLynx 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 F0 (Formatter Control Register)	7008 2041h	This sets the DVLynx to transmit data (NTSC Format) from the bulky data interface. DVLynx automatically inserts 1394 Iso header, CIP headers, and H0 headers. Empty packets are also distributed throughout the transmitted packet.
Register 1C8 (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 1CC (CIP Transmit Header 0)	0078 0000h	DV packet size is 78hex quadlets.
Register 1D0 (CIP Transmit Header 1)	8000 0000h	This value is the CIP1 header that is auto-inserted. Data Format = NTSC
Register 1A4 (DV Transmit Cell Header Register 0)		This is byte 0 – 3 of DIF block H0.
Register 1A8 (DV Transmit Cell Header Register 1)		This is byte 4 – 7 of DIF block H0.
Register DC (Transmit Timestamp Offset Register)	0000 3000h	This sets a transmit offset value of 3 isochronous cycles. This value is added to the cycle timer to make up the transmitted timestamp.
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.3.2 Transmitting Fully Formatted Data Fully Formatted with 1394 Isochronous, CIP, and H0 Headers (Microprocessor Interface)

Table B–6. Transmitting Fully Formatted Data Fully Formatted

Register Name/Number	Setting	Explanation
Register 150 (Bulky Size Register)	00C0 0000h	This sets the bulky DV receive register to 3072 bytes (decimal).
Register C (Link Control Register)	C407 0A00	This register sets the DVLynx to try to become cycle master.
Register 34 (Phy Access Register)	41C6 0000h	This tells the Phy to arbitrate for bus/cycle master.
Register F0 (Formatter Control Register)	4008 2000h	This register sets the DVLynx to transmit fully formatted DV data (NTSC format) from the Microprocessor Interface.
Register 160 (Transmit FIFO First and Continue)		Using this write only register, the application software writes all quadlets of a DV packet (expect the last) to the Bulky transmit FIFO using this register.
Register 164 (Transmit FIFO Last and Send)		Using this write only register, the application software writes the last quadlet of a DV packet to the Bulky transmit FIFO. The entire packet is confirmed for transmission.
Register DC (Transmit Timestamp Offset Register)	0000 3000h	This sets a transmit offset value of 3 isochronous cycles. This value is added to the cycle timer to make up the transmitted timestamp.
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.

Appendix C
Isolation Considerations for TSB12LV42

When using TI's Bus Holder Phy/Link Isolation solution with the TSB12LV42, 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 TSB12LV42 and physical layer device, the $\overline{\text{ISO}}$ pin of the TSB12LV42 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 3.3 V power) to the phy LPS pin. The will also ensure that a lockup condition will not occur when the link is reactivated.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)
TSB12LV42PZ	OBSOLETE	LQFP	PZ	100		TBD	Call TI	Call TI

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS) or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.