

# Contents

<i>Section</i>	<i>Title</i>	<i>Page</i>
<b>1</b>	<b>Introduction</b>	<b>1-1</b>
1.1	Description	1-1
1.2	Features	1-1
1.3	Related Documents	1-2
1.4	Ordering Information	1-2
<b>2</b>	<b>Terminal Descriptions</b>	<b>2-1</b>
<b>3</b>	<b>TSB12LV23 Controller Programming Model</b>	<b>3-1</b>
3.1	PCI/CardBus Configuration Registers	3-3
3.2	Vendor ID Register	3-3
3.3	Device ID Register	3-4
3.4	PCI Command Register	3-4
3.5	PCI Status Register	3-5
3.6	Class Code and Revision ID Register	3-6
3.7	Latency Timer and Class Cache Line Size Register	3-6
3.8	Header Type and BIST Register	3-7
3.9	OHCI Base Address Register	3-7
3.10	TI Extension Base Address Register	3-8
3.11	CIS Base Address Register	3-8
3.12	CardBus CIS Pointer Register	3-9
3.13	PCI Subsystem Identification Register	3-9
3.14	PCI Power Management Capabilities Pointer Register	3-10
3.15	Interrupt Line and Pin Registers	3-10
3.16	MIN_GNT and MAX_LAT Register	3-11
3.17	PCI OHCI Control Register	3-11
3.18	Capability ID and Next Item Pointer Registers	3-12
3.19	Power Management Capabilities Register	3-13
3.20	Power Management Control and Status Register	3-14
3.21	Power Management Extension Registers	3-14
3.22	PCI Miscellaneous Configuration Register	3-15
3.23	Link Enhancement Control Register	3-16
3.24	Subsystem Access Register	3-17
3.25	GPIO Control Register	3-18
<b>4</b>	<b>OHCI Registers</b>	<b>4-1</b>
4.1	OHCI Version Register	4-4
4.2	GUID ROM Register	4-5
4.3	Asynchronous Transmit Retries Register	4-6
4.4	CSR Data Register	4-6

4.5	CSR Compare Register .....	4-7
4.6	CSR Control Register .....	4-7
4.7	Configuration ROM Header Register .....	4-8
4.8	Bus Identification Register .....	4-8
4.9	Bus Options Register .....	4-9
4.10	GUID High Register .....	4-10
4.11	GUID Low Register .....	4-10
4.12	Configuration ROM Mapping Register .....	4-11
4.13	Posted Write Address Low Register .....	4-11
4.14	Posted Write Address High Register .....	4-12
4.15	Vendor ID Register .....	4-12
4.16	Host Controller Control Register .....	4-13
4.17	Self ID Buffer Pointer Register .....	4-14
4.18	Self ID Count Register .....	4-14
4.19	Isochronous Receive Channel Mask High Register .....	4-15
4.20	Isochronous Receive Channel Mask Low Register .....	4-16
4.21	Interrupt Event Register .....	4-18
4.22	Interrupt Mask Register .....	4-20
4.23	Isochronous Transmit Interrupt Event Register .....	4-21
4.24	Isochronous Transmit Interrupt Mask Register .....	4-22
4.25	Isochronous Receive Interrupt Event Register .....	4-22
4.26	Isochronous Receive Interrupt Mask Register .....	4-23
4.27	Fairness Control Register .....	4-23
4.28	Link Control Register .....	4-24
4.29	Node Identification Register .....	4-25
4.30	PHY Layer Control Register .....	4-26
4.31	Isochronous Cycle Timer Register .....	4-27
4.32	Asynchronous Request Filter High Register .....	4-28
4.33	Asynchronous Request Filter Low Register .....	4-30
4.34	Physical Request Filter High Register .....	4-32
4.35	Physical Request Filter Low Register .....	4-34
4.36	Physical Upper Bound Register (Optional Register) .....	4-36
4.37	Asynchronous Context Control Register .....	4-37
4.38	Asynchronous Context Command Pointer Register .....	4-38
4.39	Isochronous Transmit Context Control Register .....	4-39
4.40	Isochronous Transmit Context Command Pointer Register .....	4-40
4.41	Isochronous Receive Context Control Register .....	4-40
4.42	Isochronous Receive Context Command Pointer Register .....	4-41
4.43	Isochronous Receive Context Match Register .....	4-42
<b>5</b>	<b>GPIO Interface .....</b>	<b>5-1</b>
<b>6</b>	<b>Serial EEPROM Interface .....</b>	<b>6-1</b>
<b>7</b>	<b>Electrical Characteristics .....</b>	<b>7-1</b>
7.1	Absolute Maximum Ratings Over Operating Temperature Ranges .....	7-1

7.2	Recommended Operating Conditions .....	7-2
7.3	Electrical Characteristics Over Recommended Operating Conditions .....	7-3
7.4	Switching Characteristics for PCI Interface .....	7-3
7.5	Switching Characteristics for PHY-Link Interface .....	7-3
<b>8</b>	<b>Mechanical Information .....</b>	<b>8-1</b>

## List of Illustrations

<i>Figure</i>	<i>Title</i>	<i>Page</i>
2-1	Terminal Assignments .....	2-1
3-1	TSB12LV23 Block Diagram .....	3-2
5-1	GPIO2 and GPIO3 .....	5-1

## List of Tables

<i>Table</i>	<i>Title</i>	<i>Page</i>
2-1	Signals Sorted by Pin Number .....	2-2
2-2	Power Supply .....	2-2
2-3	PCI System .....	2-3
2-4	PCI Address and Data .....	2-3
2-5	PCI Interface Control .....	2-4
2-6	IEEE1394 PHY/Link .....	2-4
2-7	Miscellaneous .....	2-5
3-1	Bit Field Access Tag Descriptions .....	3-1
3-2	PCI Configuration Register Map .....	3-3
3-3	PCI Command Register Description .....	3-4
3-4	PCI Status Register Description .....	3-5
3-5	Class Code and Revision ID Register Description .....	3-6
3-6	Latency Timer and Class Cache Line Size Register Description .....	3-6
3-7	Header Type and BIST Register Description .....	3-7
3-8	OHCI Base Address Register Description .....	3-7
3-9	CIS Base Address Register Description .....	3-8
3-10	CardBus CIS Pointer Register Description .....	3-9
3-11	PCI Subsystem Identification Register Description .....	3-9
3-12	Interrupt Line and Pin Registers Description .....	3-10
3-13	MIN_GNT and MAX_LAT Register Description .....	3-11
3-14	PCI OHCI Control Register Description .....	3-11
3-15	Capability ID and Next Item Pointer Registers Description .....	3-12
3-16	Power Management Capabilities Register Description .....	3-13
3-17	Power Management Control and Status Register Description .....	3-14
3-18	Power Management Extension Registers Description .....	3-14

3-19	PCI Miscellaneous Configuration Register	3-15
3-20	Link Enhancement Control Register Description	3-16
3-21	Subsystem Access Register Description	3-17
3-22	General-Purpose Input/Output Control Register Description	3-18
4-1	OHCI Register Map	4-1
4-2	OHCI Version Register Description	4-4
4-3	GUID ROM Register Description	4-5
4-4	Asynchronous Transmit Retries Register Description	4-6
4-5	CSR Control Register Description	4-7
4-6	Configuration ROM Header Register Description	4-8
4-7	Bus Options Register Description	4-9
4-8	Configuration ROM Mapping Register Description	4-11
4-9	Posted Write Address Low Register Description	4-11
4-10	Posted Write Address High Register Description	4-12
4-11	Host Controller Control Register Description	4-13
4-12	Self ID Count Register Description	4-14
4-13	Isochronous Receive Channel Mask High Register Description	4-15
4-14	Isochronous Receive Channel Mask Low Register Description	4-16
4-15	Interrupt Event Register Description	4-18
4-16	Interrupt Mask Register Description	4-20
4-17	Isochronous Transmit Interrupt Event Register Description	4-21
4-18	Isochronous Receive Interrupt Event Register Description	4-22
4-19	Fairness Control Register Description	4-23
4-20	Link Control Register Description	4-24
4-21	Node Identification Register Description	4-25
4-22	PHY Control Register Description	4-26
4-23	Isochronous Cycle Timer Register Description	4-27
4-24	Asynchronous Request Filter High Register Description	4-28
4-25	Asynchronous Request Filter Low Register Description	4-30
4-26	Physical Request Filter High Register Description	4-32
4-27	Physical Request Filter Low Register Description	4-34
4-28	Asynchronous Context Control Register Description	4-37
4-29	Asynchronous Context Command Pointer Register Description	4-38
4-30	Isochronous Transmit Context Control Register Description	4-39
4-31	Isochronous Receive Context Control Register Description	4-40
4-32	Isochronous Receive Context Match Register Description	4-42
6-1	Registers and Bits Loadable through Serial EEPROM	6-1
6-2	Serial EEPROM Map	6-2

# 1 Introduction

## 1.1 Description

The Texas Instruments TSB12LV23 is a PCI-to-1394 host controller compatible with the latest *PCI Local Bus*, *PCI Bus Power Management Interface*, *IEEE 1394-1995*, and *1394 Open Host Controller Interface Specifications*. The chip provides the IEEE 1394 link function, and is compatible with serial bus data rates of 100 Mbits/s, 200 Mbits/s, and 400 Mbits/s.

As required by the *1394 Open Host Controller Interface (OHCI)* and *IEEE 1394A Specifications*, internal control registers are memory mapped and non-prefetchable. The PCI configuration header is accessed through configuration cycles specified by PCI, and provides Plug-and-Play (PnP) compatibility. Furthermore, the TSB12LV23 is compliant with the *PCI Bus Power Management Interface Specification*, per the *PC 98* requirements. TSB12LV23 supports the D0, D2, and D3 power states.

The TSB12LV23 design provides PCI bus master bursting, and is capable of transferring a cacheline of data at 132 Mbytes/s after connection to the memory controller. Since PCI latency can be large even on a PCI Revision 2.1 system, deep FIFOs are provided to buffer 1394 data.

The TSB12LV23 provides physical write posting buffers and a highly tuned physical data path for SBP-2 performance. The TSB12LV23 also provides multiple isochronous contexts, multiple cacheline burst transfers, advanced internal arbitration, and bus holding buffers on the PHY/Link interface, thus, making the TSB12LV23 the best-in-class 1394 OHCI solution.

An advanced CMOS process is used to achieve low power consumption while operating at PCI clock rates up to 33 MHz.

## 1.2 Features

The TSB12LV23 supports the following features:

- 3.3-V core logic with universal PCI interfaces compatible with 3.3-V and 5-V PCI signaling environments
- Supports serial bus data rates of 100, 200, and 400 Mbits/s
- Provides bus-hold buffers on physical interface for low-cost single capacitor isolation
- Supports physical write posting of up to three outstanding transactions
- Serial ROM interface supports 2-wire devices
- Supports external cycle timer control for customized synchronization
- Implements PCI burst transfers and deep FIFOs to tolerate large host latency
- Provides two general-purpose I/Os
- Fabricated in advanced low-power CMOS process
- Packaged in 100 LQFP (PZ)
- Supports  $\overline{\text{CLKRUN}}$
- Drop-in replacement for the TSB12LV22
- Supports PCI and CardBus applications

**1.3 Related Documents**

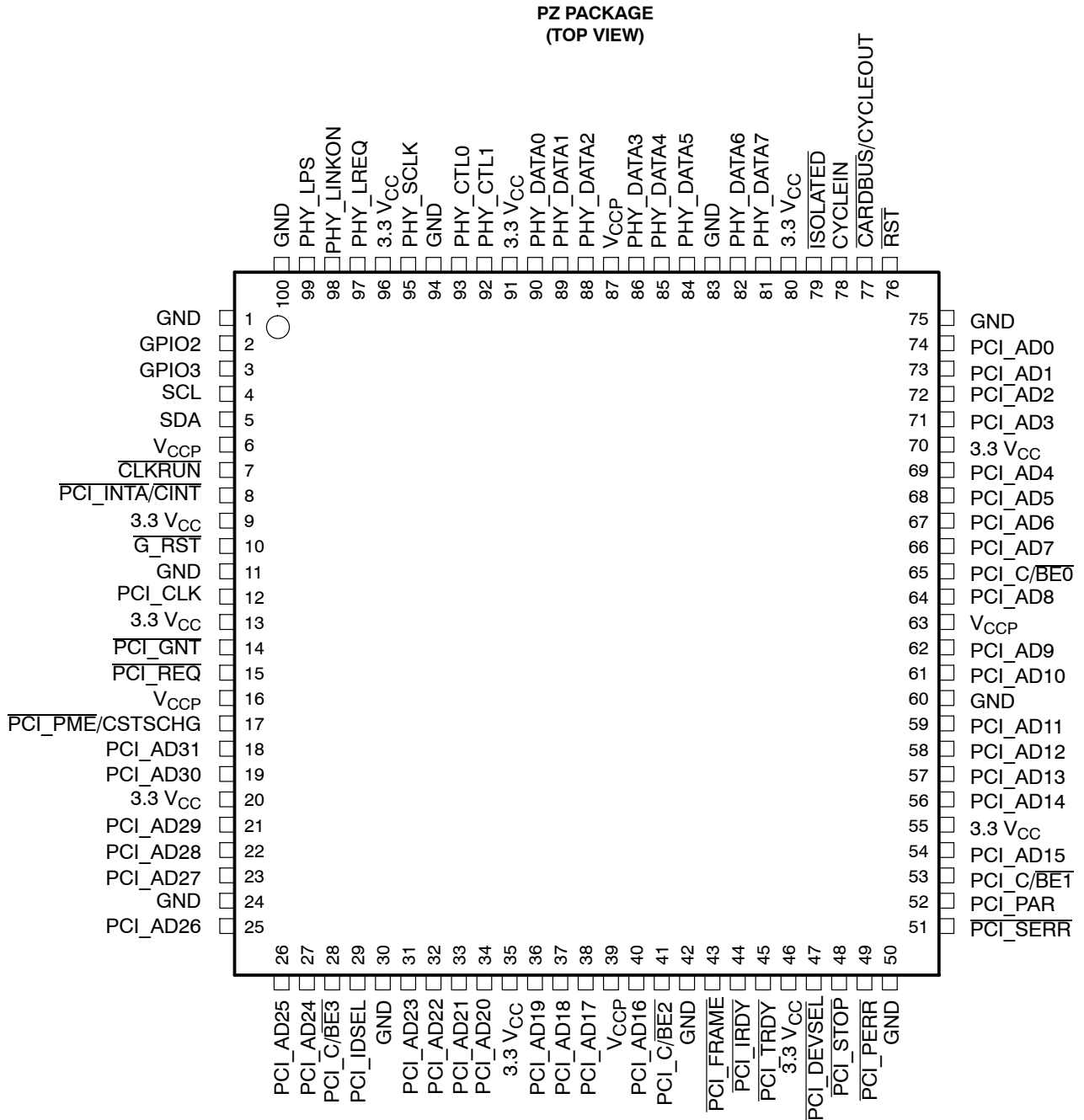
- 1394 Open Host Controller Interface Specification
- IEEE 1394-1995 and Compatible with Proposal 1394A
- PC 98
- PCI Bus Power Management Interface Specification (Revision 1.1)
- PCI Local Bus Specification (Revision 2.2)

**1.4 Ordering Information**

ORDERING NUMBER	NAME	VOLTAGE	PACKAGE
TSB12LV23	OHCI-Lynx PCI-Based IEEE 1394 Host Controller	3.3 V, 5-V Tolerant I/Os	100-pin LQFP

## 2 Terminal Descriptions

This section provides the terminal descriptions for the TSB12LV23.



**Figure 2–1. Terminal Assignments**

**Table 2–1. Signals Sorted by Pin Number**

NO.	TERMINAL NAME	NO.	TERMINAL NAME	NO.	TERMINAL NAME	NO.	TERMINAL NAME
1	GND	26	PCI_AD25	51	PCI_SERR	76	RST
2	GPIO2	27	PCI_AD24	52	PCI_PAR	77	CARDBUS/CYCLEOUT
3	GPIO3	28	PCI_C/BE3	53	PCI_C/BE1	78	CYCLEIN
4	SCL	29	PCI_IDSEL	54	PCI_AD15	79	ISOLATED
5	SDA	30	GND	55	3.3 V <sub>CC</sub>	80	3.3 V <sub>CC</sub>
6	V <sub>CCP</sub>	31	PCI_AD23	56	PCI_AD14	81	PHY_DATA7
7	CLKRUN	32	PCI_AD22	57	PCI_AD13	82	PHY_DATA6
8	PCI_INTA/CINT	33	PCI_AD21	58	PCI_AD12	83	GND
9	3.3 V <sub>CC</sub>	34	PCI_AD20	59	PCI_AD11	84	PHY_DATA5
10	G_RST	35	3.3 V <sub>CC</sub>	60	GND	85	PHY_DATA4
11	GND	36	PCI_AD19	61	PCI_AD10	86	PHY_DATA3
12	PCI_CLK	37	PCI_AD18	62	PCI_AD9	87	V <sub>CCP</sub>
13	3.3 V <sub>CC</sub>	38	PCI_AD17	63	V <sub>CCP</sub>	88	PHY_DATA2
14	PCI_GNT	39	V <sub>CCP</sub>	64	PCI_AD8	89	PHY_DATA1
15	PCI_REQ	40	PCI_AD16	65	PCI_C/BE0	90	PHY_DATA0
16	V <sub>CCP</sub>	41	PCI_C/BE2	66	PCI_AD7	91	3.3 V <sub>CC</sub>
17	PCI_PME/CSTSCHG	42	GND	67	PCI_AD6	92	PHY_CTL1
18	PCI_AD31	43	PCI_FRAME	68	PCI_AD5	93	PHY_CTL0
19	PCI_AD30	44	PCI_IRDY	69	PCI_AD4	94	GND
20	3.3 V <sub>CC</sub>	45	PCI_TRDY	70	3.3 V <sub>CC</sub>	95	PHY_SCLK
21	PCI_AD29	46	3.3 V <sub>CC</sub>	71	PCI_AD3	96	3.3 V <sub>CC</sub>
22	PCI_AD28	47	PCI_DEVSEL	72	PCI_AD2	97	PHY_LREQ
23	PCI_AD27	48	PCI_STOP	73	PCI_AD1	98	PHY_LINKON
24	GND	49	PCI_PERR	74	PCI_AD0	99	PHY_LPS
25	PCI_AD26	50	GND	75	GND	100	GND

The terminals are grouped in tables by functionality, such as PCI system function, power supply function, etc. The terminal numbers are also listed for convenient reference.

**Table 2–2. Power Supply**

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
GND	1, 11, 24, 30, 42, 50, 60, 75, 83, 94, 100	I	Device ground terminals
3.3 V <sub>CC</sub>	9, 13, 20, 35, 46, 55, 70, 80, 91, 96	I	3.3-V power supply terminals
V <sub>CCP</sub>	6, 16, 39, 63, 87	I	PCI signaling clamp voltage power input. PCI signals are clamped per the <i>PCI Local Bus Specification</i> .



**Table 2–3. PCI System**

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
PCI_CLK	12	I	PCI bus clock. Provides timing for all transactions on the PCI bus. All PCI signals are sampled at rising edge of PCLK.
$\overline{G\_RST}$	10	I	Global power reset. This reset brings all of the TSB12LV23 internal registers to their default states, including those registers not reset by RST. When asserted, the device is completely nonfunctional. When implementing wake capabilities from the 1394 host controller, it is necessary to implement two resets to the TSB12LV23. $\overline{G\_RST}$ should be a one-time power-on reset, and $\overline{RST}$ should be connected to the PCI bus RST. If wake capabilities are not required, $\overline{G\_RST}$ may be connected to the PCI bus RST (see RST, terminal 76).
PCI_INTA/CINT	8	O	Interrupt signal. This output indicates interrupts from the TSB12LV23 to the host. This terminal is implemented as open-drain.
$\overline{RST}$	76	I	PCI reset. When this bus reset is asserted, the TSB12LV23 places all output buffers in a high impedance state and resets all internal registers except device power management context- and vendor-specific bits initialized by host power-on software. When asserted, the device is completely nonfunctional. If this terminal is implemented, then it should be connected to the PCI bus $\overline{RST}$ signal. Otherwise, it should be pulled high to the link $V_{CC}$ through a 4.7-k $\Omega$ resistor, or strapped to the $\overline{G\_RST}$ terminal (see $\overline{G\_RST}$ , terminal 10).

**Table 2–4. PCI Address and Data**

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
PCI_AD31	18	I/O	PCI address/data bus. These signals make up the multiplexed PCI address and data bus on the PCI interface during the address phase of a PCI cycle, AD31–AD0 contain a 32-bit address or other destination information. During the data phase, AD31–AD0 contain data.
PCI_AD30	19		
PCI_AD29	21		
PCI_AD28	22		
PCI_AD27	23		
PCI_AD26	25		
PCI_AD25	26		
PCI_AD24	27		
PCI_AD23	31		
PCI_AD22	32		
PCI_AD21	33		
PCI_AD20	34		
PCI_AD19	36		
PCI_AD18	37		
PCI_AD17	38		
PCI_AD16	40		
PCI_AD15	54		
PCI_AD14	56		
PCI_AD13	57		
PCI_AD12	58		
PCI_AD11	59		
PCI_AD10	61		
PCI_AD9	62		
PCI_AD8	64		
PCI_AD7	66		
PCI_AD6	67		
PCI_AD5	68		
PCI_AD4	69		
PCI_AD3	71		
PCI_AD2	72		
PCI_AD	73		
PCI_AD0	74		

PCI_C/BE $\overline{0}$ PCI_C/BE $\overline{1}$ PCI_C/BE $\overline{2}$ PCI_C/BE $\overline{3}$	65 53 41 28	I/O	PCI bus commands and byte enables. The command and byte enable signals are multiplexed on the same PCI terminals. During the address phase of a bus cycle C/BE $\overline{3}$ –C/BE $\overline{0}$ defines the bus command. During the data phase, this 4-bit bus is used as byte enables.
PCI_PAR	52	I/O	PCI parity. In all PCI bus read and write cycles, the TSB12LV23 calculates even parity across the AD and C/BE $\overline{}$ buses. As an initiator during PCI cycles, the TSB12LV23 outputs this parity indicator with a one PCLK delay. As a target during PCI cycles, the calculated parity is compared to the initiator's parity indicator; a miscompare can result in a parity error assertion ( $\overline{\text{PERR}}$ ).

**Table 2–5. PCI Interface Control**

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
$\overline{\text{PCI\_DEVSEL}}$	47	I/O	PCI device select. The TSB12LV23 asserts this signal to claim a PCI cycle as the target device. As a PCI initiator, the TSB12LV23 monitors this signal until a target responds. If no target responds before time-out occurs, then the TSB12LV23 terminates the cycle with an initiator abort.
$\overline{\text{PCI\_FRAME}}$	43	I/O	PCI cycle frame. This signal is driven by the initiator of a PCI bus cycle. $\overline{\text{FRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue until while this signal is asserted. When $\overline{\text{FRAME}}$ is deasserted, the PCI bus transaction is in the final data phase.
$\overline{\text{PCI\_GNT}}$	14	I	PCI bus grant. This signal is driven by the PCI bus arbiter to grant the TSB12LV23 access to the PCI bus after the current data transaction has completed. This signal may or may not follow a PCI bus request depending upon the PCI bus parking algorithm.
PCI_IDSEL	29	I	Initialization device select. IDSEL selects the TSB12LV23 during configuration space accesses. IDSEL can be connected to 1 of the upper 24 PCI address lines on the PCI bus.
$\overline{\text{PCI\_IRDY}}$	44	I/O	PCI initiator ready. $\overline{\text{IRDY}}$ indicates the PCI bus initiator's ability to complete the current data phase of the transaction. A data phase is completed upon a rising edge of PCLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted; until which wait states are inserted.
$\overline{\text{PCI\_STOP}}$	48	I/O	PCI cycle stop signal. This signal is driven by a PCI target to request the initiator to stop the current PCI bus transaction. This signal is used for target disconnects, and is commonly asserted by target devices which do not support burst data transfers.
$\overline{\text{CLKRUN}}$	7	I/O	Clock run. This terminal provides clock control through the $\overline{\text{CLKRUN}}$ protocol. An internal pulldown resistor is implemented on this terminal. This terminal is implemented as open-drain.
$\overline{\text{PCI\_PERR}}$	49	I/O	PCI parity error indicator. This signal is driven by a PCI device to indicate that calculated parity does not match PAR when enabled through the command register.
$\overline{\text{PCI\_PME/}}\overline{\text{CSTSCHG}}$	17	O	PME or card status change. This terminal indicates wake events to the host. When in a CardBus configuration, per the $\overline{\text{CARDBUS}}$ sample, the CSTSCHG output is an active high.
$\overline{\text{PCI\_REQ}}$	15	O	PCI bus request. Asserted by the TSB12LV23 to request access to the bus as an initiator. The host arbiter asserts the $\overline{\text{GNT}}$ signal when the TSB12LV23 has been granted access to the bus.
$\overline{\text{PCI\_SERR}}$	51	O	PCI system error. Output pulsed from the TSB12LV23 when enabled indicating an address parity error has occurred. The TSB12LV23 needs not be the target of the PCI cycle to assert this signal. This terminal is implemented as open-drain.
$\overline{\text{PCI\_TRDY}}$	45	I/O	PCI target ready. $\overline{\text{TRDY}}$ indicates the PCI bus target's ability to complete the current data phase of the transaction. A data phase is completed upon a rising edge of PCLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted; until which wait states are inserted.

**Table 2–6. IEEE1394 PHY/Link**

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
PHY_CTL1 PHY_CTL0	92 93	I/O	Phy-link interface control. These bidirectional signals control passage of information between the two devices. The TSB12LV23 can only drive these terminals after the PHY has granted permission following a link request (LREQ).
PHY_DATA7 PHY_DATA6 PHY_DATA5 PHY_DATA4 PHY_DATA3 PHY_DATA2 PHY_DATA1 PHY_DATA0	81 82 84 85 86 88 89 90	I/O	Phy-link interface data. These bidirectional signals pass data between the TSB12LV23 and the PHY device. These terminals are driven by the TSB12LV23 on transmissions and are driven by the PHY on reception. Only DATA1–DATA0 are valid for 100-Mbit speeds, DATA3–DATA0 are valid for 200-Mbit speeds, and DATA7–DATA0 are valid for 400-Mbit speeds.
PHY_SCLK	95	I	System clock. This input from the PHY provides a 49.152 MHz clock signal for data synchronization.
PHY_LREQ	97	O	Link request. This signal is driven by the TSB12LV23 to initiate a request for the PHY to perform some service.
PHY_LINKON	98	I/O	LinkOn wake indication. Used and defined by 1394A and 3.3-V signaling is required.
PHY_LPS	99	I/O	Link power status. Used and defined by 1394A and 3.3-V signaling is required.

**Table 2–7. Miscellaneous**

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
SDA	5	I/O	Serial data. The TSB12LV23 determines whether a two-wire serial ROM, or no serial ROM is implemented at reset. If a two-wire serial ROM is detected, then this terminal provides the SDA serial data signaling. This terminal must be wired low to indicate no serial ROM is present.  This terminal is implemented as open-drain, and for normal operation (an EEPROM is implemented in the design), this terminal should be pulled high to the EEPROM $V_{CC}$ with a 2.7-k $\Omega$ resistor. Otherwise, it should be pulled low to ground with a 220- $\Omega$ resistor.
SCL	4	I/O	Serial clock. The TSB12LV23 determines whether a two-wire, or no serial ROM is implemented at reset. If a two-wire serial ROM is implemented, then this terminal provides the SCL serial clock signaling.  This terminal is implemented as open-drain, and for normal operation (an EEPROM is implemented in the design), this terminal should be pulled high to the EEPROM $V_{CC}$ with a 2.7-k $\Omega$ resistor. Otherwise, it should be pulled low to ground with a 220- $\Omega$ resistor.
ISOLATED	79	I	For isolated designs, the bus holders are implemented by pulling this terminal low to ground with a 220- $\Omega$ resistor. If this terminal is not implemented, it should be pulled high to the link $V_{CC}$ with a 4.7-k $\Omega$ resistor.
CYCLEIN	78	I/O	The CYCLEIN terminal allows an external 8-kHz clock to be used as a cycle timer for synchronization with other system devices.  If this terminal is not implemented, then it should be pulled high to the link $V_{CC}$ through a 4.7-k $\Omega$ resistor.
CARDBUS/ CYCLEOUT	77	I/O	This terminal is sampled when $\overline{G\_RST}$ is asserted, and it selects between PCI buffers and CardBus buffers. After reset, this terminal may also function as CYCLEOUT which provides an 8 kHz cycle timer synchronization signal.
GPIO3	3	I/O	General-purpose I/O [3]. This terminal defaults as an input and if it is not implemented, then it is recommended that it be pulled low to ground with a 220- $\Omega$ resistor.
GPIO2	2	I/O	General-purpose I/O [2]. This terminal defaults as an input and if it is not implemented, then it is recommended that it be pulled low to ground with a 220- $\Omega$ resistor.



### 3 TSB12LV23 Controller Programming Model

This section describes the internal registers used to program the TSB12LV23, including both PCI configuration registers and OHCI registers (see Section 4). All registers are detailed in the same format: a brief description for each register, followed by the register offset and a bit table describing the reset state for each register.

A bit description table, typically included, indicates bit field names, a detailed field description, and field access tags. Table 3–1 describes the field access tags.

**Table 3–1. Bit Field Access Tag Descriptions**

ACCESS TAG	NAME	MEANING
R	Read	Field may be read by software.
W	Write	Field may be written by software to any value.
S	Set	Field may be set by a write of 1. Writes of 0 have no effect.
C	Clear	Field may be cleared by a write of 1. Writes of 0 have no effect.
U	Update	Field may be autonomously updated by the TSB12LV23.

A simplified block diagram of the TSB12LV23 is provided in Figure 3–1.

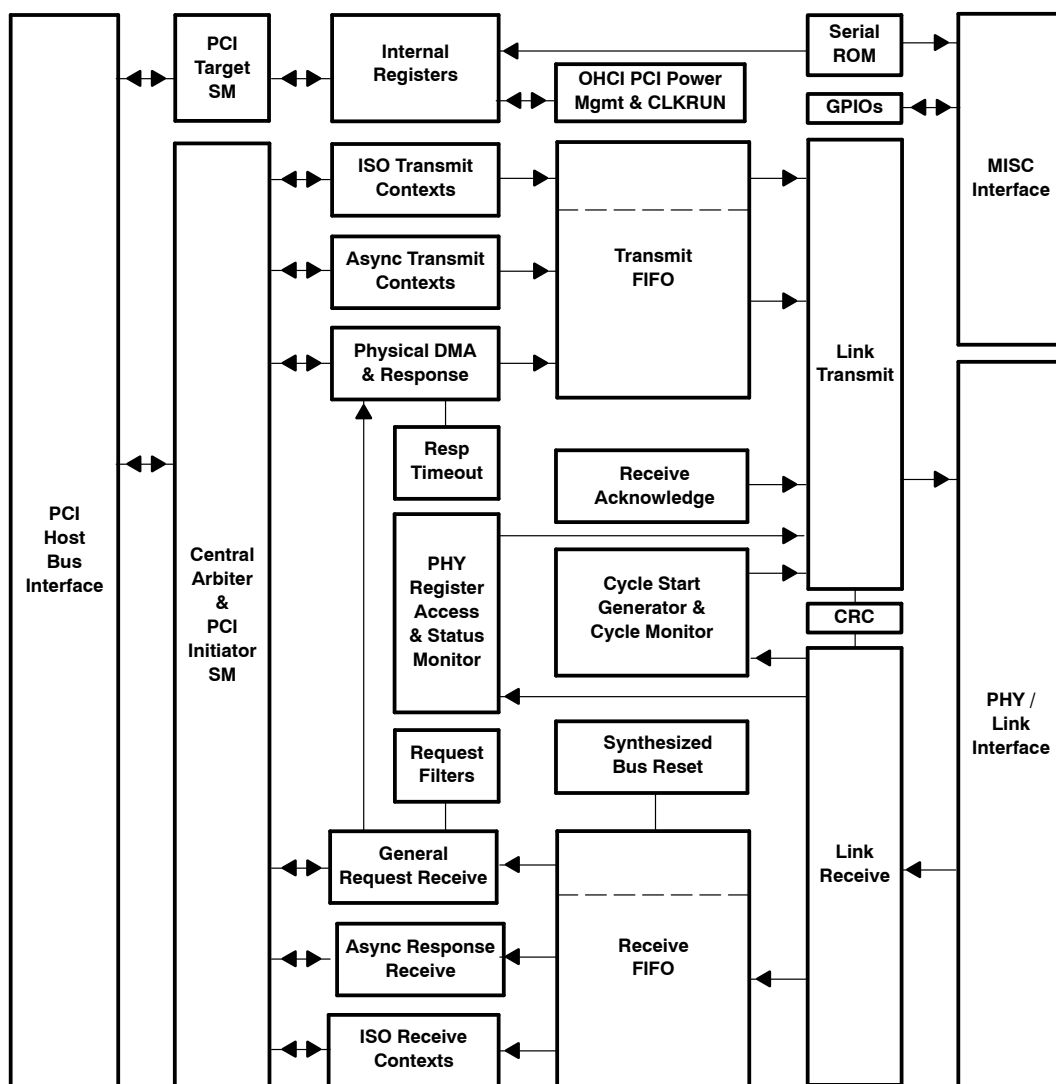


Figure 3-1. TSB12LV23 Block Diagram

### 3.1 PCI/CardBus Configuration Registers

The TSB12LV23 is a single-function PCI device that can be configured as either a PCI or CardBus device. The configuration header is compliant with the *PCI Local Bus Specification* as a standard header. Table 3–2 illustrates the PCI configuration header that includes both the predefined portion of the configuration space and the user definable registers. Most of the registers in this configuration have not changed from the TSB12LV22 design.

**Table 3–2. PCI Configuration Register Map**

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
Class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
OHCI registers base address				10h
TI extension registers base address				14h
CIS base address				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
CardBus CIS pointer				28h
Subsystem ID		Subsystem vendor ID		2Ch
Reserved				30h
Reserved			Capabilities pointer	34h
Reserved				38h
Maximum latency	Minimum grant	Interrupt pin	Interrupt line	3Ch
PCI OHCI control register				40h
Power management capabilities		Next item pointer	Capability ID	44h
PM data	PMCSR_BSE	Power management CSR		48h
Reserved				4C–ECh
PCI miscellaneous configuration register				F0h
Link_Enhancements register				F4h
Subsystem ID alias		Subsystem vendor ID alias		F8h
GPIO3	GPIO2	Reserved		FCh

### 3.2 Vendor ID Register

The vendor ID register contains a value allocated by the PCI SIG and identifies the manufacturer of the PCI device. The vendor ID assigned to Texas Instruments is 104Ch.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Vendor ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0

Register: **Vendor ID**  
 Type: Read-only  
 Offset: 00h  
 Default: 104Ch



### 3.3 Device ID Register

The device ID register contains a value assigned to the TSB12LV23 by Texas Instruments. The device identification for the TSB12LV23 is 8019.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Device ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1

Register: **Device ID**  
Type: Read-only  
Offset: 02h  
Default: 8019h

### 3.4 PCI Command Register

The command register provides control over the TSB12LV23 interface to the PCI bus. All bit functions adhere to the definitions in the *PCI Local Bus Specification*, as seen in the following bit descriptions.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI command															
Type	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W	R/W	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **PCI command**  
Type: Read/Write  
Offset: 04h  
Default: 0000h

**Table 3–3. PCI Command Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
15–10	RSVD	R	Reserved. Bits 15–10 return 0s when read.
9	FBB_ENB	R	Fast back-to-back enable. The TSB12LV23 does not generate fast back-to-back transactions, thus this bit returns 0 when read.
8	SERR_ENB	R/W	SERR enable. When this bit is set, the TSB12LV23 SERR driver is enabled. SERR can be asserted after detecting an address parity error on the PCI bus.
7	STEP_ENB	R	Address/data stepping control. The TSB12LV23 does not support address/data stepping, thus this bit is hardwired to 0.
6	PERR_ENB	R/W	Parity error enable. When this bit is set, the TSB12LV23 is enabled to drive PERR response to parity errors through the PERR signal.
5	VGA_ENB	R	VGA palette snoop enable. The TSB12LV23 does not feature VGA palette snooping. This bit returns 0 when read.
4	MWI_ENB	R/W	Memory write and invalidate enable. When this bit is set, the TSB12LV23 is enabled to generate MWI PCI bus commands. If this bit is reset, then the TSB12LV23 generates memory write commands instead.
3	SPECIAL	R	Special cycle enable. The TSB12LV23 function does not respond to special cycle transactions. This bit returns 0 when read.
2	MASTER_ENB	R/W	Bus master enable. When this bit is set, the TSB12LV23 is enabled to initiate cycles on the PCI bus.
1	MEMORY_ENB	R/W	Memory response enable. Setting this bit enables the TSB12LV23 to respond to memory cycles on the PCI bus. This bit must be set to access OHCI registers.
0	IO_ENB	R	I/O space enable. The TSB12LV23 does not implement any I/O mapped functionality; thus, this bit returns 0 when read.

### 3.5 PCI Status Register

The status register provides status over the TSB12LV23 interface to the PCI bus. All bit functions adhere to the definitions in the *PCI Local Bus Specification*, as seen in the following bit descriptions.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI status															
Type	RCU	RCU	RCU	RCU	RCU	R	R	RCU	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Register: **PCI status**  
Type: Read/Clear/Update  
Offset: 06h  
Default: 0210h

**Table 3–4. PCI Status Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
15	PAR_ERR	RCU	Detected parity error. This bit is set when a parity error is detected, either address or data parity errors.
14	SYS_ERR	RCU	Signaled system error. This bit is set when $\overline{SERR}$ is enabled and the TSB12LV23 has signaled a system error to the host.
13	MABORT	RCU	Received master abort. This bit is set when a cycle initiated by the TSB12LV23 on the PCI bus has been terminated by a master abort.
12	TABORT_REC	RCU	Received target abort. This bit is set when a cycle initiated by the TSB12LV23 on the PCI bus was terminated by a target abort.
11	TABORT_SIG	RCU	Signaled target abort. This bit is set by the TSB12LV23 when it terminates a transaction on the PCI bus with a target abort.
10–9	PCI_SPEED	R	DEVSEL timing. Bits 10–9 encode the timing of $\overline{DEVSEL}$ and are hardwired to 01b indicating that the TSB12LV23 asserts this signal at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	RCU	Data parity error detected. This bit is set when the following conditions have been met: a. PERR was asserted by any PCI device including the TSB12LV23 b. The TSB12LV23 was the bus master during the data parity error c. The parity error response bit is set in the command register (see Section 3.4)
7	FBB_CAP	R	Fast back-to-back capable. The TSB12LV23 cannot accept fast back-to-back transactions; thus, this bit is hardwired to 0.
6	UDF	R	User definable features (UDF) supported. The TSB12LV23 does not support the UDF; thus, this bit is hardwired to 0.
5	66MHZ	R	66 MHz capable. The TSB12LV23 operates at a maximum PCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4	CAPLIST	R	Capabilities list. This bit returns 1 when read, indicating that capabilities additional to standard PCI are implemented. The linked list of PCI power management capabilities is implemented in this function.
3–0	RSVD	R	Reserved. Bits 3–0 return 0s when read.

### 3.6 Class Code and Revision ID Register

The class code and revision ID register categorizes the TSB12LV23 as a serial bus controller (0Ch), controlling an IEEE1394 bus (00h), with an OHCI programming model (10h). Furthermore, the TI chip revision is indicated in the lower byte.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Class code and revision ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Class code and revision ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Class code and revision ID**  
 Type: Read-only  
 Offset: 08h  
 Default: 0C00 1000h

**Table 3–5. Class Code and Revision ID Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31–24	BASECLASS	R	Base class. This field returns 0Ch when read, which broadly classifies the function as a serial bus controller.
23–16	SUBCLASS	R	Subclass. This field returns 00h when read, which specifically classifies the function as controlling an IEEE1394 serial bus.
15–8	PGMIF	R	Programming interface. This field returns 10h when read, indicating that the programming model is compliant with the <i>1394 Open Host Controller Interface Specification</i> .
7–0	CHIPREV	R	Silicon revision. This field returns 00h when read, indicating the silicon revision of the TSB12LV23.

### 3.7 Latency Timer and Class Cache Line Size Register

The latency timer and class cache line size register is programmed by host BIOS to indicate system cache line size and the latency timer associated with the TSB12LV23.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Latency timer and class cache line size															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Latency timer and class cache line size**  
 Type: Read/Write  
 Offset: 0Ch  
 Default: 0000h

**Table 3–6. Latency Timer and Class Cache Line Size Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
15–8	LATENCY_TIMER	R/W	PCI latency timer. The value in this register specifies the latency timer for the TSB12LV23, in units of PCI clock cycles. When the TSB12LV23 is a PCI bus initiator and asserts FRAME#, the latency timer begins counting from zero. If the latency timer expires before the TSB12LV23 transaction has terminated, then the TSB12LV23 terminates the transaction when its GNT# is deasserted.
7–0	CACHELINE_SZ	R/W	Cache line size. This value is used by the TSB12LV23 during memory write and invalidate, memory read line, and memory read multiple transactions.

### 3.8 Header Type and BIST Register

The header type and BIST register indicates the TSB12LV23 PCI header type, and indicates no built-in self test.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Header type and BIST															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Header type and BIST**  
Type: Read-only  
Offset: 0Eh  
Default: 0000h

**Table 3–7. Header Type and BIST Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
15–8	BIST	R	Built-in self test. The TSB12LV23 does not include a built-in self test; thus, this field returns 00h when read.
7–0	HEADER_TYPE	R	PCI header type. The TSB12LV23 includes the standard PCI header, and this is communicated by returning 00h when this field is read.

### 3.9 OHCI Base Address Register

The OHCI base address register is programmed with a base address referencing the memory-mapped OHCI control. When BIOS writes all 1s to this register, the value read back is FFFF F800h, indicating that at least 2K bytes of memory address space are required for the OHCI registers.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OHCI base address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OHCI address															
Type	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **OHCI base address**  
Type: Read/Write  
Offset: 10h  
Default: 0000 0000h

**Table 3–8. OHCI Base Address Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–11	OHCIREG_PTR	R/W	OHCI register pointer. Specifies the upper 21 bits of the 32-bit OHCI base address register.
10–4	OHCI_SZ	R	OHCI register size. This field returns 0s when read, indicating that the OHCI registers require a 2-Kbyte region of memory.
3	OHCI_PF	R	OHCI register prefetch. This bit returns 0 when read, indicating that the OHCI registers are nonprefetchable.
2–1	OHCI_MEMTYPE	R	OHCI memory type. This field returns 0s when read, indicating that the OHCI base address register is 32 bits wide and mapping can be done anywhere in the 32-bit memory space.
0	OHCI_MEM	R	OHCI memory indicator. This bit returns 0 when read, indicating that the OHCI registers are mapped into system memory space.

### 3.10 TI Extension Base Address Register

The TI extension base address register is programmed with a base address referencing the memory-mapped TI extension registers. Refer to the *OHCI base address register* (see Section 3.9) for bit field details.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TI extension base address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TI extension base address															
Type	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **TI extension base address**  
Type: Read/Write  
Offset: 14h  
Default: 0000 0000h

### 3.11 CIS Base Address Register

If  $\overline{\text{CARDBUS}}$  is sampled high on a PCI reset, then this 32-bit register returns 0s when read. If  $\overline{\text{CARDBUS}}$  is sampled low, then this register is to be programmed with a base address referencing the memory mapped CIS. This register must be programmed with a nonzero value before the CIS may be accessed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CIS base address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CIS base address															
Type	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **CIS base address**  
Type: Read/Write  
Offset: 18h  
Default: 0000 0000h

**Table 3–9. CIS Base Address Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–11	CIS_BASE	R/W	CIS base address. Specifies the upper 21 bits of the 32-bit CIS base address. If the $\overline{\text{CARDBUS}}$ input is sampled high on a PCI reset, then this field is read-only, returning 0s when read.
10–4	CIS_SZ	R	CIS address space size. This field returns 0s when read, indicating that the CIS space requires a 2-Kbyte region of memory.
3	CIS_PF	R	CIS prefetch. This bit returns 0 when read, indicating that the CIS is nonprefetchable. Furthermore, the CIS is a byte-accessible address space, and double-word or 16-bit word access yields indeterminate results.
2–1	CIS_MEMTYPE	R	CIS memory type. This field returns 0s when read, indicating that the CIS base address register is 32 bits wide and mapping can be done anywhere in the 32-bit memory space.
0	CIS_MEM	R	CIS memory indicator. This bit returns 0 when read, indicating that the CIS is mapped into system memory space.

### 3.12 CardBus CIS Pointer Register

The **CARDBUS** input to the TSB12LV23 is sampled at PCI reset to determine the TSB12LV23 application. If **CARDBUS** is sampled high, then this register is read-only returning 0s when read. If **CARDBUS** is sampled low, then this register is the CardBus card information structure pointer.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	CardBus CIS pointer															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	CardBus CIS pointer															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

Register: **CardBus CIS pointer**  
Type: Read-only  
Offset: 28h  
Default: 0000 000xh

**Table 3–10. CardBus CIS Pointer Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–28	ROM_IMAGE	R	Since the CIS is not implemented as a ROM image, this field returns 0s when read.
27–3	CIS_OFFSET	R	This field indicates the offset into the CIS address space where the CIS begins, and bits 7–3 are loaded from the serial ROM field CIS_Offset (7–3). This implementation allows the TSB12LV23 to produce serial ROM addresses equal to the lower PCI address byte to acquire data from the serial ROM.
2–0	CIS_INDICATOR	R	This field indicates the address space where the CIS resides and returns 010b if <b>CARDBUS</b> is sampled asserted during a PCI reset. If <b>CARDBUS</b> is sampled high during a PCI reset, then this field returns 000b when read. Thus, bit 1 is implemented as the logical inverse of the <b>CARDBUS</b> input.

### 3.13 PCI Subsystem Identification Register

The PCI subsystem identification register is used for system and option card identification purposes. This register can be initialized from the serial EEPROM or programmed via the subsystem ID and subsystem vendor ID alias registers at offset 0XFC.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	PCI subsystem identification															
<b>Type</b>	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	PCI subsystem identification															
<b>Type</b>	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **PCI subsystem identification**  
Type: Read/Update  
Offset: 2Ch  
Default: 0000 0000h

**Table 3–11. PCI Subsystem Identification Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–16	OHCI_SSID	RU	Subsystem device ID. This field indicates the subsystem device ID.
15–0	OHCI_SSVID	RU	Subsystem vendor ID. This field indicates the subsystem vendor ID.

### 3.14 PCI Power Management Capabilities Pointer Register

The PCI power management capabilities pointer register provides a pointer into the PCI configuration header where the PCI power management register block resides. The TSB12LV23 configuration header double-words at offsets 44h and 48h provide the power management registers. This register is read-only and returns 44h when read.

Bit	7	6	5	4	3	2	1	0
Name	PCI power management capabilities pointer							
Type	R	R	R	R	R	R	R	R
Default	0	1	0	0	0	1	0	0

Register: **PCI power management capabilities pointer**  
Type: Read-only  
Offset: 34h  
Default: 44h

### 3.15 Interrupt Line and Pin Registers

The interrupt line and pin register is used to communicate interrupt line routing information.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Interrupt line and pin															
Type	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Register: **Interrupt line and pin**  
Type: Read/Write  
Offset: 3Ch  
Default: 0100h

**Table 3–12. Interrupt Line and Pin Registers Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
15–8	INTR_PIN	R	Interrupt pin register. This register returns 01h when read, indicating that the TSB12LV23 PCI function signals interrupts on the $\overline{INTA}$ pin.
7–0	INTR_LINE	R/W	Interrupt line register. This register is programmed by the system and indicates to software to which interrupt line the TSB12LV23 $\overline{INTA}$ is connected.

### 3.16 MIN\_GNT and MAX\_LAT Register

The MIN\_GNT and MAX\_LAT register is used to communicate to the system the desired setting of the latency timer register (see Section 3.7). If a serial ROM is detected, then the contents of this register are loaded through the serial ROM interface after a PCI reset. If no serial ROM is detected, then this register returns a default value that corresponds to the MIN\_GNT = 2, MAX\_LAT = 4.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MIN_GNT and MAX_LAT															
Type	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Register: **MIN\_GNT and MAX\_LAT**  
Type: Read/Update  
Offset: 3Eh  
Default: 0202h

**Table 3–13. MIN\_GNT and MAX\_LAT Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
15–8	MAX_LAT	RU	Maximum latency. The contents of this register may be used by host BIOS to assign an arbitration priority-level to the TSB12LV23. The default for this register indicates that the TSB12LV23 may need to access the PCI bus as often as every 0.25 $\mu$ s; thus, an extremely high priority level is requested. The contents of this field may also be loaded through the serial ROM.
7–0	MIN_GNT	RU	Minimum grant. The contents of this register may be used by host BIOS to assign a latency timer register value to the TSB12LV23. The default for this register indicates that the TSB12LV23 may need to sustain burst transfers for nearly 64 $\mu$ s; thus, requesting a large value be programmed in the TSB12LV23 latency timer register (see Section 3.7).

### 3.17 PCI OHCI Control Register

The PCI OHCI control register is defined by the *1394 Open Host Controller Interface Specification* and provides a bit for big endian PCI support.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCI OHCI control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI OHCI control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **PCI OHCI control**  
Type: Read/Write  
Offset: 40h  
Default: 0000 0000h

**Table 3–14. PCI OHCI Control Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–1	RSVD	R	Reserved. Bits 31–1 return 0s when read.
0	GLOBAL_SWAP	R/W	When this bit is set, all quadlets read from and written to the PCI interface are byte swapped (big endian).



### 3.18 Capability ID and Next Item Pointer Registers

The capability ID and next item pointer register identifies the linked list capability item and provides a pointer to the next capability item.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Capability ID and next item pointer															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register: **Capability ID and next item pointer**

Type: Read-only

Offset: 44h

Default: 0001h

**Table 3–15. Capability ID and Next Item Pointer Registers Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
15–8	NEXT_ITEM	R	Next item pointer. The TSB12LV23 supports only one additional capability that is communicated to the system through the extended capabilities list; thus, this field returns 00h when read.
7–0	CAPABILITY_ID	R	Capability identification. This field returns 01h when read, which is the unique ID assigned by the PCI SIG for PCI power management capability.

### 3.19 Power Management Capabilities Register

The power management capabilities register indicates the capabilities of the TSB12LV23 related to PCI power management.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management capabilities															
Type	RU	RU	RU	RU	RU	RU	R	R	R	R	R	R	R	R	R	R
Default	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	1

Register: **Power management capabilities**

Type: Read/Update

Offset: 46h

Default: 6411h

**Table 3–16. Power Management Capabilities Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
15	PME_D3COLD	RU	PME support from D3 <sub>COLD</sub> . When this bit is set, the TSB12LV23 generates a PME wake event from D3 <sub>COLD</sub> . This bit state is dependent upon the TSB12LV23 V <sub>AUX</sub> implementation and may be configured by host software using the PCI miscellaneous configuration register (see Section 3.22).
14–11	PME_SUPPORT	RU	PME support. This 4-bit field indicates the power states from which the TSB12LV23 may assert PME. This field returns a value of 1100b by default, indicating that PME may be asserted from the D3 <sub>HOT</sub> and D2 power states. Bit 13 may be modified by host software using the PCI miscellaneous configuration register (see Section 3.22).
10	D1_SUPPORT	RU	D2 support. This bit returns a 1 when read, indicating that the TSB12LV23 does not support the D2 power state.
9	D1_SUPPORT	R	D1 support. This bit returns a 0 when read, indicating that the TSB12LV23 does not support the D1 power state.
8	DYN_DATA	R	Dynamic data support. This bit returns a 0 when read, indicating that the TSB12LV23 does not report dynamic power consumption data.
7–6	RSVD	R	Reserved. Bits 7–6 return 0s when read.
5	DSI	R	Device specific initialization. This bit returns 0 when read, indicating that the TSB12LV23 does not require special initialization beyond the standard PCI configuration header before a generic class driver is able to use it.
4	AUX_PWR	R	Auxiliary power source. Since the TSB12LV23 does not support PME generation in the D3 <sub>COLD</sub> device state, this bit returns 0 when read.
3	PME_CLK	R	PME clock. This bit returns 0 when read, indicating that no host bus clock is required for the TSB12LV23 to generate PME.
2–0	PM_VERSION	R	Power management version. This field returns 001b when read, indicating that the TSB12LV23 is compatible with the registers described in the <i>PCI Bus Power Management Interface Specification</i> .

### 3.20 Power Management Control and Status Register

The power management control and status register implements the control and status of the PCI power management function. This register is not affected by the internally generated reset caused by the transition from the D3<sub>HOT</sub> to D0 state.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management control and status															
Type	RC	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Power management control and status**  
Type: Read/Write/Clear  
Offset: 48h  
Default: 0000h

**Table 3–17. Power Management Control and Status Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
15	PME_STS	RC	This bit is set when the TSB12LV23 would normally be asserting the $\overline{\text{PME}}$ signal, independent of the state of the PME_ENB bit. This bit is cleared by a write back of 1, and this also clears the PME signal driven by the TSB12LV23. Writing a 0 to this bit has no effect.
14–9	DYN_CTRL	R	Dynamic data control. This field returns 0s when read since the TSB12LV23 does not report dynamic data.
8	PME_ENB	R/W	PME enable. This bit enables the function to assert PME. If this bit is cleared, then assertion of PME is disabled.
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	DYN_DATA	R	Dynamic data. This bit returns 0 when read since the TSB12LV23 does not report dynamic data.
3–2	RSVD	R	Reserved. Bits 3–2 return 0s when read.
1–0	PWR_STATE	R/W	Power state. This 2-bit field is used to set the TSB12LV23 device power state and is encoded as follows: 00 = Current power state is D0 01 = Current power state is D1 10 = Current power state is D2 11 = Current power state is D3

### 3.21 Power Management Extension Registers

The power management extension register provides extended power management features not applicable to the TSB12LV23, thus it is read-only and returns 0 when read.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management extension															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Power management extension**  
Type: Read-only  
Offset: 4Ah  
Default: 0000h

**Table 3–18. Power Management Extension Registers Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
15–8	PM_DATA	R	Power management data. This field returns 00h when read since the TSB12LV23 does not report dynamic data.
7–0	PMCSR_BSE	R	Power management CSR – bridge support extensions. This field returns 00h when read since the TSB12LV23 does not provide P2P bridging.

## 3.22 PCI Miscellaneous Configuration Register

The PCI miscellaneous configuration register provides miscellaneous PCI-related configuration.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCI miscellaneous configuration															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI miscellaneous configuration															
Type	R/W	R	R/W	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

Register: **PCI miscellaneous configuration**  
Type: Read/Write  
Offset: F0h  
Default: 0000 2400h

**Table 3–19. PCI Miscellaneous Configuration Register**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–16	RSVD	R	Reserved. Bits 31–16 return 0s when read.
15	PME_D3COLD	R/W	PME support from D3 <sub>cold</sub> . This bit is used to program bit 15 (PME_D3COLD) in the power management capabilities register (see Section 3.19).
14	RSVD	R	Reserved. Bit 14 returns 0 when read.
13	PME_SUPPORT_D2	R/W	PME support. This bit is used to program bit 13 (PME_SUPPORT_D2) in the power management capabilities register (see Section 3.19). If wake from the D2 power state implemented in the TSB12LV23 is not desired, then this bit may be cleared to indicate to power management software that wake-up from D2 is not supported.
12–11	RSVD	R	Reserved. Bits 12–11 return 0s when read.
10	D2_SUPPORT	R/W	D2 support. This bit is used to program bit 10 (D2_SUPPORT) in the power management capabilities register (see Section 3.19). If the D2 power state implemented in the TSB12LV23 is not desired, then this bit may be cleared to indicate to power management software that D2 is not supported.
9–5	RSVD	R	Reserved. Bits 9–5 return 0s when read.
4	DIS_TGT_ABT	R/W	This bit defaults to 0, which provides OHCI-Lynx compatible target abort signaling. When this bit is set to 1, it enables the no-target-abort mode, in which the TSB12LV23 returns indeterminate data instead of signaling target abort. The link is divided into the PCI_CLK and SCLK domains. If software tries to access registers in the link that are not active because the SCLK is disabled, a target abort is issued by the link. On some systems this can cause a problem resulting in a fatal system error. Enabling this bit allows the link to respond to these types of requests by returning FFh. It is recommended that this bit be set to 1.
3	GP2IIC	R/W	When this bit is set to 1, the GPIO3 and GPIO2 signals are internally routed to the SCL and SDA, respectively. The GPIO3 and GPIO2 terminals are also placed in a high impedance state.
2	DISABLE_SCLKGATE	R/W	When this bit is set to 1, the internal SCLK runs identically with the chip input. This bit is a test feature only and should be cleared to 0 (all applications).
1	DISABLE_PCIGATE	R/W	When this bit is set, the internal PCI clock runs identically with the chip input. This bit is a test feature only and should be cleared to 0 (all applications).
0	KEEP_PCLK	R/W	When this bit is set to 1, the PCI clock is always kept running through the $\overline{\text{CLKRUN}}$ protocol. When this bit is cleared, the PCI clock may be stopped using $\overline{\text{CLKRUN}}$ .

### 3.23 Link Enhancement Control Register

The link enhancement control register implements TI proprietary bits that are initialized by software or by a serial EEPROM, if present. After these bits are set, their functionality is enabled only if bit 22 (aPhyEnhanceEnable) in the host controller control register (see Section 4.16) is set.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Link enhancement control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Link enhancement control															
Type	R	R	R/W	R/W	R	R	R	R	R/W	R	R	R	R	R/W	R/W	R
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Link enhancement control**  
Type: Read/Write  
Offset: F4h  
Default: 0000 1000h

**Table 3–20. Link Enhancement Control Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–14	RSVD	R	Reserved. Bits 31–14 return 0s when read.
13–12	atx_thresh	R/W	<p>This field sets the initial AT threshold value, which is used until the AT FIFO is underrun. When the TSB12LV23 retries the packet, it uses a 2-Kbyte threshold resulting in a store-and-forward operation.</p> <p>00 = Threshold ~ 2K bytes resulting in a store-and-forward operation  01 = Threshold ~ 1.7K bytes (default)  10 = Threshold ~ 1K  11 = Threshold ~ 512 bytes</p> <p>These bits fine-tune the asynchronous transmit threshold. For most applications the 1.7K threshold is optimal. Changing this value may increase or decrease the 1394 latency depending on the average PCI bus latency.</p> <p>Setting the AT threshold to 1.7K, 1K, or 512 bytes results in data being transmitted at these thresholds, or when an entire packet has been checked into the FIFO. If the packet to be transmitted is larger than the AT threshold, then the remaining data must be received before the AT FIFO is emptied; otherwise, an underrun condition will occur, resulting in a packet error at the receiving node. As a result, the link will then commence store-and-forward operation, i.e., wait until it has the complete packet in the FIFO before retransmitting it on the second attempt, to ensure delivery.</p> <p>An AT threshold of 2K results in store-and-forward operation, which means that asynchronous data will not be transmitted until an end-of-packet token is received. Restated, setting the AT threshold to 2K results in only complete packets being transmitted.</p>
11–8	RSVD	R	Reserved. Bits 11–8 return 0s when read.
7	enab_unfair	R/W	Enable asynchronous priority requests. OHCI-Lynx compatible. Setting this bit to 1 enables the link to respond to requests with priority arbitration. It is recommended that this bit be set to 1.
6	RSVD	R	This bit is not assigned in the TSB12LV23 follow-on products since this bit location loaded by the serial ROM from the Enhancements field corresponds to bit 23 (programPhyEnable) in the host controller control register (see Section 4.16).
5–3	RSVD	R	Reserved. Bits 5–3 return 0s when read.
2	enab_insert_idle	R/W	Enable insert idle. OHCI-Lynx compatible. When the PHY has control of the Ct[0:1] control lines and D[0:8] data lines and the link requests control, the PHY drives 11 on the Ct[0:1] lines. The link can then start driving these lines immediately. Setting this bit to 1 inserts an idle state, so the link waits one clock cycle before it starts driving the lines (turnaround time). It is recommended that this bit be set to 1.
1	enab_accel	R/W	Enable acceleration enhancements. OHCI-Lynx compatible. This bit notifies the PHY that the link supports the 1394a acceleration enhancements, i.e., ack-accelerated, fly-by concatenation, etc. It is recommended that this bit be set to 1.
0	RSVD	R	Reserved. Bit 0 returns 0 when read.

### 3.24 Subsystem Access Register

Write access to the subsystem access register updates the subsystem identification registers identically to OHCI-Lynx. The system ID value written to this register may also be read back from this register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Subsystem access															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Subsystem access															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Subsystem access**  
 Type: Read/Write  
 Offset: F8h  
 Default: 0000 0000h

**Table 3–21. Subsystem Access Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–16	SUBDEV_ID	R/W	Subsystem device ID. This field indicates the subsystem device ID.
15–0	SUBVEN_ID	R/W	Subsystem vendor ID. This field indicates the subsystem vendor ID.

### 3.25 GPIO Control Register

The GPIO control register has the control and status bits for the GPIO2 and GPIO3 ports.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIO control															
Type	R/W	R	R/W	R/W	R	R	R	RWU	R/W	R	R/W	R/W	R	R	R	RWU
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Register: **GPIO control**  
Type: Read/Write/Update  
Offset: FCh  
Default: 0000 1010h

**Table 3–22. General-Purpose Input/Output Control Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	INT_3EN	R/W	When this bit is set, a TSB12LV23 GPIInterrupt event occurs on a level change of the GPIO3 input. This event may generate an interrupt, with mask and event status reported through the OHCI interrupt mask (see Section 4.22) and interrupt event (see Section 4.21) registers.
30	RSVD	R	Reserved. Bit 30 returns 0 when read.
29	GPIO_INV3	R/W	GPIO3 polarity invert. When this bit is set, the polarity of GPIO3 is inverted.
28	GPIO_ENB3	R/W	GPIO3 enable control. When this bit is set, the output is enabled. Otherwise, the output is high impedance.
27–25	RSVD	R	Reserved. Bits 27–25 return 0s when read.
24	GPIO_DATA3	RWU	GPIO3 data. Reads from this bit return the logical value of the input to GPIO3. Writes to this bit update the value to drive to GPIO3 when output is enabled.
23	INT_2EN	R/W	When this bit is set, a TSB12LV23 GPIInterrupt event occurs on a level change of the GPIO3 input. This event may generate an interrupt, with mask and event status reported through the OHCI interrupt mask (see Section 4.22) and interrupt event (see Section 4.21) registers.
22	RSVD	R	Reserved. Bit 22 returns 0 when read.
21	GPIO_INV2	R/W	GPIO2 polarity invert. When this bit is set, the polarity of GPIO2 is inverted.
20	GPIO_ENB2	R/W	GPIO2 enable control. When this bit is set, the output is enabled. Otherwise, the output is high impedance.
19–17	RSVD	R	Reserved. Bits 19–17 return 0s when read.
16	GPIO_DATA2	RWU	GPIO2 data. Reads from this bit return the logical value of the input to GPIO2. Writes to this bit update the value to drive to GPIO2 when the output is enabled.
15–0	RSVD	R	Reserved. Bits 15–0 return 0s when read.

## 4 OHCI Registers

The OHCI registers defined by the *1394 Open Host Controller Interface Specification* are memory mapped into a 2-Kbyte region of memory pointed to by the OHCI base address register at offset 10h in PCI configuration space. These registers are the primary interface for controlling the TSB12LV23 IEEE1394 link function.

This section provides the register interface and bit descriptions. There are several set and clear register pairs in this programming model, which are implemented to solve various issues with typical read-modify-write control registers. There are two addresses for a set/clear register: RegisterSet and RegisterClear. Refer to Table 4–1 for an illustration. A 1 bit written to RegisterSet causes the corresponding bit in the set/clear register to be set, while a 0 bit leaves the corresponding bit unaffected. A 1 bit written to RegisterClear causes the corresponding bit in the set/clear register to be reset, while a 0 bit leaves the corresponding bit in the set/clear register unaffected.

Typically, a read from either RegisterSet or RegisterClear returns the contents of the set or clear register. However, sometimes reading the RegisterClear provides a masked version of the set or clear register. The interrupt event register is an example of this behavior.

**Table 4–1. OHCI Register Map**

DMA CONTEXT	REGISTER NAME	ABBREVIATION	OFFSET
—	OHCI version	Version	00h
	Global unique ID ROM	GUID_ROM	04h
	Asynchronous transmit retries	ATRetries	08h
	CSR data	CSRData	0Ch
	CSR compare data	CSRCompareData	10h
	CSR control	CSRControl	14h
	Configuration ROM header	ConfigROMhdr	18h
	Bus identification	BusID	1Ch
	Bus options	BusOptions	20h
	Global unique ID high	GUIDHi	24h
	Global unique ID low	GUIDLo	28h
	PCI subsystem identification	SSID	2Ch
	Reserved	—	30h
	Configuration ROM map	ConfigROMmap	34h
	Posted write address low	PostedWriteAddressLo	38h
	Posted write address high	PostedWriteAddressHi	3Ch
	Vendor identification	VendorID	40h
	Capability ID and next item pointer	CAP_ID	44h
	Power management capabilities	PM_CAP	46h
	Power management control and status	PMCSR	48h
	Power management extensions	PM_Ext	4Ah
	Reserved	—	4Ch
	Host controller control	HCCControlSet	50h
		HCCControlClr	54h
	Reserved	—	58h
	Reserved	—	5Ch



**Table 4–1. OHCI Register Map (Continued)**

DMA CONTEXT	REGISTER NAME	ABBREVIATION	OFFSET
Self ID	Reserved	—	60h
	Self ID buffer	SelfIDBuffer	64h
	Self ID count	SelfIDCount	68h
	Reserved	—	6Ch
—	Isochronous receive channel mask high	IRChannelMaskHiSet	70h
		IRChannelMaskHiClear	74h
	Isochronous receive channel mask low	IRChannelMaskLoSet	78h
		IRChannelMaskLoClear	7Ch
	Interrupt event	IntEventSet	80h
		IntEventClear	84h
	Interrupt mask	IntMaskSet	88h
		IntMaskClear	8Ch
	Isochronous transmit interrupt event	IsoXmitIntEventSet	90h
		IsoXmitIntEventClear	94h
	Isochronous transmit interrupt mask	IsoXmitIntMaskSet	98h
		IsoXmitIntMaskClear	9Ch
—	Isochronous receive interrupt event	IsoRecvIntEventSet	A0h
		IsoRecvIntEventClear	A4h
	Isochronous receive interrupt mask	IsoRecvIntMaskSet	A8h
		IsoRecvIntMaskClear	ACh
	Reserved		B0–D8h
	Fairness control	FairnessControl	DCh
	Link control	LinkControlSet	E0h
		LinkControlClear	E4h
	Node identification	NodeID	E8h
	PHY layer control	PhyControl	ECh
	Isochronous cycle timer	Isocyc timer	F0h
	Reserved		F4h
	Reserved		F8h
	Reserved		FCh
	Asynchronous request filter high	AsyncRequestFilterHiSet	100h
		AsyncRequestFilterHiClear	104h
	Asynchronous request filter low	AsyncRequestFilterLoSet	108h
		AsyncRequestFilterLoClear	10Ch
	Physical request filter high	PhysicalRequestFilterHiSet	110h
		PhysicalRequestFilterHiClear	114h
	Physical request filter low	PhysicalRequestFilterLoSet	118h
		PhysicalRequestFilterLoClear	11Ch
	Physical upper bound	PhysicalUpperBound	120h
	Reserved	—	124h–17Ch

**Table 4–1. OHCI Register Map (Continued)**

DMA CONTEXT	REGISTER NAME	ABBREVIATION	OFFSET
Asynchronous Request Transmit [ ATRQ ]	Context control	ContextControlSet	180h
		ContextControlClear	184h
	Reserved	—	188h
	Command pointer	CommandPtr	18Ch
Asynchronous Response Transmit [ ATRS ]	Reserved	—	190h–19Ch
	Context control	ContextControlSet	1A0h
		ContextControlClear	1A4h
	Reserved	—	1A8h
	Command pointer	CommandPtr	1ACh
Asynchronous Request Receive [ ARRQ ]	Reserved	—	1B0h–1BCh
	Context control	ContextControlSet	1C0h
		ContextControlClear	1C4h
	Reserved	—	1C8h
	Command pointer	CommandPtr	1CCh
Asynchronous Response Receive [ ARRS ]	Reserved	—	1D0h–1DCh
	Context control	ContextControlSet	1E0h
		ContextControlClear	1E4h
	Reserved	—	1E8h
	Command pointer	CommandPtr	1ECh
Isochronous Transmit Context n n = 0, 1, 2, 3, ... 7	Reserved	—	1F0h–1FCh
	Context control	ContextControlSet	200h + 16*n
		ContextControlClear	204h + 16*n
Isochronous Receive Context n n = 0, 1, 2, 3, ... 4	Reserved	—	208h + 16*n
	Command pointer	CommandPtr	20Ch + 16*n
	Context control	ContextControlSet	400h + 32*n
		ContextControlClear	404h + 32*n
	Reserved	—	408h + 32*n
	Command pointer	CommandPtr	40Ch + 32*n
	Context match	ContextMatch	410h + 32*n

## 4.1 OHCI Version Register

This register indicates the OHCI version support, and whether or not the serial ROM is present.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OHCI version															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	X	0	0	0	0	0	0	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OHCI version															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **OHCI version**  
 Type: Read-only  
 Offset: 00h  
 Default: 0X01 0000h

**Table 4–2. OHCI Version Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31–25	RSVD	R	Reserved. Bits 31–25 return 0s when read.
24	GUID_ROM	R	The TSB12LV23 sets this bit if the serial ROM is detected. If the serial ROM is present, then the Bus_Info_Block is automatically loaded on hardware reset.
23–16	version	R	Major version of the OHCI. The TSB12LV23 is compliant with the <i>1394 Open Host Controller Interface Specification</i> ; thus, this field reads 01h.
15–8	RSVD	R	Reserved. Bits 15–8 return 0s when read.
7–0	revision	R	Minor version of the OHCI. The TSB12LV23 is compliant with the <i>1394 Open Host Controller Interface Specification</i> ; thus, this field reads 00h.

## 4.2 GUID ROM Register

The GUID ROM register is used to access the serial ROM, and is only applicable if bit 24 (GUID\_ROM) in the OHCI version register (see Section 4.1) is set.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GUID ROM															
Type	RSU	R	R	R	R	R	RSU	R	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUID ROM															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **GUID ROM**  
 Type: Read/Set/Update  
 Offset: 04h  
 Default: 00XX 0000h

**Table 4–3. GUID ROM Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	addrReset	RSU	Software sets this bit to reset the GUID ROM address to 0. When the TSB12LV23 completes the reset, it clears this bit. The TSB12LV23 does not automatically fill bits 23–16 (rdData field) with the 0 <sup>th</sup> byte.
30–26	RSVD	R	Reserved. Bits 30–26 return 0s when read.
25	rdStart	RSU	A read of the currently addressed byte is started when this bit is set. This bit is automatically cleared when the TSB12LV23 completes the read of the currently addressed GUID ROM byte.
24	RSVD	R	Reserved. Bit 24 returns 0 when read.
23–16	rdData	RU	This field represents the data read from the GUID ROM.
15–0	RSVD	R	Reserved. Bits 15–0 return 0s when read.

### 4.3 Asynchronous Transmit Retries Register

The asynchronous transmit retries register indicates the number of times the TSB12LV23 attempts a retry for asynchronous DMA request transmit and for asynchronous physical and DMA response transmit.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Asynchronous transmit retries															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Asynchronous transmit retries															
Type	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Asynchronous transmit retries**  
Type: Read/Write  
Offset: 08h  
Default: 0000 0000h

**Table 4–4. Asynchronous Transmit Retries Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31–29	secondLimit	R	The second limit field returns 0s when read, since outbound dual-phase retry is not implemented.
28–16	cycleLimit	R	The cycle limit field returns 0s when read, since outbound dual-phase retry is not implemented.
15–12	RSVD	R	Reserved. Bits 15–12 return 0s when read.
11–8	maxPhysRespRetries	R/W	This field tells the physical response unit how many times to attempt to retry the transmit operation for the response packet when a busy acknowledge or ack_data_error is received from the target node.
7–4	maxATRespRetries	R/W	This field tells the asynchronous transmit response unit how many times to attempt to retry the transmit operation for the response packet when a busy acknowledge or ack_data_error is received from the target node.
3–0	maxATReqRetries	R/W	This field tells the asynchronous transmit DMA request unit how many times to attempt to retry the transmit operation for the response packet when a busy acknowledge or ack_data_error is received from the target node.

### 4.4 CSR Data Register

The CSR data register is used to access the bus management CSR registers from the host through compare-swap operations. This register contains the data to be stored in a CSR if the compare is successful.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSR data															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSR data															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **CSR data**  
Type: Read-only  
Offset: 0Ch  
Default: XXXX XXXXh

## 4.5 CSR Compare Register

The CSR compare register is used to access the bus management CSR registers from the host through compare-swap operations. This register contains the data to be compared with the existing value of the CSR resource.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSR compare															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSR compare															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **CSR compare**  
Type: Read-only  
Offset: 10h  
Default: XXXX XXXXh

## 4.6 CSR Control Register

The CSR control register is used to access the bus management CSR registers from the host through compare-swap operations. This register is used to control the compare-swap operation and to select the CSR resource.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSR control															
Type	RU	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSR control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X

Register: **CSR control**  
Type: Read/Write/Update  
Offset: 14h  
Default: 8000 000Xh

**Table 4–5. CSR Control Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31	csrDone	RU	This bit is set by the TSB12LV23 when a compare-swap operation is complete. It is reset whenever this register is written.
30–2	RSVD	R	Reserved. Bits 30–2 return 0s when read.
1–0	csrSel	R/W	This field selects the CSR resource as follows: 00 = BUS_MANAGER_ID 01 = BANDWIDTH_AVAILABLE 10 = CHANNELS_AVAILABLE_HI 11 = CHANNELS_AVAILABLE_LO

## 4.7 Configuration ROM Header Register

The configuration ROM header register externally maps to the first quadlet of the 1394 configuration ROM, offset 48'hFFFF\_F000\_0400.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Configuration ROM header															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Configuration ROM header															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Configuration ROM header**  
Type: Read/Write  
Offset: 18h  
Default: 0000 XXXXh

**Table 4–6. Configuration ROM Header Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–24	info_length	R/W	IEEE1394 bus management field. Must be valid when bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set.
23–16	crc_length	R/W	IEEE1394 bus management field. Must be valid when bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set.
15–0	rom_crc_value	R/W	IEEE1394 bus management field. Must be valid at any time bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set. The reset value is undefined if no serial ROM is present. If a serial ROM is present, then this field is loaded from the serial ROM.

## 4.8 Bus Identification Register

The bus identification register externally maps to the first quadlet in the Bus\_Info\_Block, and contains the constant 32'h31333934, which is the ASCII value of 1394.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Bus identification															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Bus identification															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	1	1	1	0	0	1	0	0	1	1	0	1	0	0

Register: **Bus identification**  
Type: Read-only  
Offset: 1Ch  
Default: 3133 3934h

## 4.9 Bus Options Register

The bus options register externally maps to the second quadlet of the Bus\_Info\_Block.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Bus options															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	X	X	X	X	0	0	0	0	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Bus options															
<b>Type</b>	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R	R	R	R	R	R
<b>Default</b>	1	0	1	0	0	0	0	0	X	X	0	0	0	0	1	0

Register: **Bus options**  
Type: Read/Write  
Offset: 20h  
Default: X0XX A0X2h

**Table 4–7. Bus Options Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	irmc	R/W	Isochronous resource manager capable. IEEE1394 bus management field. Must be valid when bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set.
30	cmc	R/W	Cycle master capable. IEEE1394 bus management field. Must be valid when bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set.
29	isc	R/W	Isochronous support capable. IEEE1394 bus management field. Must be valid when bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set.
28	bmc	R/W	Bus manager capable. IEEE1394 bus management field. Must be valid when bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set.
27	pmc	R/W	IEEE1394 bus management field. Must be valid when bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set.
26–24	RSVD	R	Reserved. Bits 26–24 return 0s when read.
23–16	cyc_clk_acc	R/W	Cycle master clock accuracy. (accuracy in parts per million) IEEE1394 bus management field. Must be valid when bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set.
15–12	max_rec	R/W	IEEE 1394 bus management field. Hardware initializes this field to indicate the maximum number of bytes in a block request packet that is supported by the implementation. This value, max_rec_bytes must be 512 greater, and is calculated by $2^{(\text{max\_rec} + 1)}$ . Software may change this field; however, this field must be valid at any time bit 17 (linkEnable) of the host controller control register (see Section 4.16) is set. A received block write request packet with a length greater than max_rec_bytes may generate an ack_type_error. This field is not affected by a soft reset, and defaults to value indicating 2048 bytes on a hard reset.
11–8	RSVD	R	Reserved. Bits 11–8 return 0s when read.
7–6	g	R/W	Generation counter. This field is incremented if any portion of the configuration ROM has been incremented since the prior bus reset.
5–3	RSVD	R	Reserved. Bits 5–3 return 0s when read.
2–0	Lnk_spd	R	Link speed. This field returns 010, indicating that the link speeds of 100, 200, and 400 Mb/s are supported.



## 4.10 GUID High Register

The GUID high register represents the upper quadlet in a 64-bit global unique ID (GUID) which maps to the third quadlet in the Bus\_Info\_Block. This register contains node\_vendor\_ID and chip\_ID\_hi fields. This register initializes to 0s on a hardware reset, which is an illegal GUID value. If a serial ROM is detected, then the contents of this register are loaded through the serial ROM interface after a PCI reset. At that point, the contents of this register cannot be changed. If no serial ROM is detected, then the contents of this register are loaded by the BIOS after a PCI reset. At that point, the contents of this register cannot be changed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GUID high															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUID high															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **GUID high**  
 Type: Read-only  
 Offset: 24h  
 Default: 0000 0000h

## 4.11 GUID Low Register

The GUID low register represents the lower quadlet in a 64-bit global unique ID (GUID) which maps to chip\_ID\_lo in the Bus\_Info\_Block. This register initializes to 0s on a hardware reset and behaves identical to the GUID high register (see Section 4.10).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GUID low															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUID low															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **GUID low**  
 Type: Read-only  
 Offset: 28h  
 Default: 0000 0000h

## 4.12 Configuration ROM Mapping Register

The configuration ROM mapping register contains the start address within system memory that maps to the start address of 1394 configuration ROM for this node.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Configuration ROM mapping															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Configuration ROM mapping															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Configuration ROM mapping**  
 Type: Read/Write  
 Offset: 34h  
 Default: 0000 0000h

**Table 4–8. Configuration ROM Mapping Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–10	configROMAddr	R/W	If a quadlet read request to 1394 offset 48'hFFFF_F000_0400 through offset 48'hFFFF_F000_07FF is received, then the low order 10 bits of the offset are added to this register to determine the host memory address of the read request.
9–0	RSVD	R	Reserved. Bits 9–0 return 0s when read.

## 4.13 Posted Write Address Low Register

The posted write address low register is used to communicate error information if a write request is posted and an error occurs while writing the posted data packet.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Posted write address low															
<b>Type</b>	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
<b>Default</b>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Posted write address low															
<b>Type</b>	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
<b>Default</b>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Posted write address low**  
 Type: Read/Update  
 Offset: 38h  
 Default: XXXX XXXXh

**Table 4–9. Posted Write Address Low Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–0	offsetLo	RU	The lower 32 bits of the 1394 destination offset of the write request that failed.

## 4.14 Posted Write Address High Register

The posted write address high register is used to communicate error information if a write request is posted and an error occurs while writing the posted data packet.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Posted write address high															
<b>Type</b>	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
<b>Default</b>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Posted write address high															
<b>Type</b>	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
<b>Default</b>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Posted write address high**  
 Type: Read/Update  
 Offset: 3Ch  
 Default: XXXX XXXXh

**Table 4–10. Posted Write Address High Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31–16	sourceID	RU	This field is the bus and node number of the node that issued the write request that failed.
15–0	offsetHi	RU	The upper 16 bits of the 1394 destination offset of the write request that failed.

## 4.15 Vendor ID Register

The vendor ID register holds the company ID of an organization that specifies any vendor-unique registers. The TSB12LV23 does not implement Texas Instruments unique behavior with regards to OHCI. Thus, this register is read-only and returns 0s when read.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Vendor ID															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Vendor ID															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Vendor ID**  
 Type: Read-only  
 Offset: 40h  
 Default: 0000 0000h

## 4.16 Host Controller Control Register

The host controller control set/clear register pair provides flags for controlling the TSB12LV23.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Host controller control															
<b>Type</b>	R	RSC	R	R	R	R	R	R	RC	RSC	R	R	RSC	RSC	RSC	RSCU
<b>Default</b>	0	X	0	0	0	0	0	0	0	0	0	0	0	X	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Host controller control															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Host controller control**  
 Type: Read/Set/Clear/Update  
 Offset: 50h set register  
 54h clear register  
 Default: X00X 0000h

**Table 4–11. Host Controller Control Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	RSVD	R	Reserved. Bit 31 returns 0 when read.
30	noByteSwapData	RSC	This bit is used to control whether physical accesses to locations outside the TSB12LV23 itself as well as any other DMA data accesses should be swapped.
29–24	RSVD	R	Reserved. Bits 29–24 return 0s when read.
23	programPhyEnable	RC	This bit informs upper level software that lower level software has consistently configured the P1394a enhancements in the Link and PHY. When this bit is 1, generic software such as the OHCI driver is responsible for configuring P1394a enhancements in the PHY and bit 22 (aPhyEnhanceEnable) in the TSB12LV23. When this bit is 0, the generic software may not modify the P1394a enhancements in the TSB12LV23 or PHY and cannot interpret the setting of bit 22 (aPhyEnhanceEnable). This bit is initialized from serial EEPROM.
22	aPhyEnhanceEnable	RSC	When bits 23 (programPhyEnable) and 17 (linkEnable) are 1, the OHCI driver can set this bit to use all P1394a enhancements. When bit 23 (programPhyEnable) is set to 0, the software does not change PHY enhancements or this bit.
21–20	RSVD	R	Reserved. Bits 21–20 return 0s when read.
19	LPS	RSC	This bit is used to control the link power status. Software must set this bit to 1 to permit the link-PHY communication. A 0 prevents link-PHY communication.
18	postedWriteEnable	RSC	This bit is used to enable (1) or disable (0) posted writes. Software should change this bit only when bit 17 (linkEnable) is 0.
17	linkEnable	RSC	This bit is cleared to 0 by either a hardware or software reset. Software must set this bit to 1 when the system is ready to begin operation and then force a bus reset. This bit is necessary to keep other nodes from sending transactions before the local system is ready. When this bit is cleared, the TSB12LV23 is logically and immediately disconnected from the 1394 bus, no packets are received or processed nor are packets transmitted.
16	SoftReset	RSCU	When this bit is set, all TSB12LV23 states are reset, all FIFOs are flushed, and all OHCI registers are set to their hardware reset values unless otherwise specified. PCI registers are not affected by this bit. This bit remains set while the softReset is in progress and reverts back to 0 when the reset has completed.
15–0	RSVD	R	Reserved. Bits 15–0 return 0s when read.

## 4.17 Self ID Buffer Pointer Register

The self ID buffer pointer register points to the 2-Kbyte aligned base address of the buffer in host memory where the self ID packets are stored during bus initialization. Bits 31–11 are read/write accessible.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Self ID buffer pointer															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Self ID buffer pointer															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0

Register: **Self ID buffer pointer**  
Type: Read/Write  
Offset: 64h  
Default: XXXX XX00h

## 4.18 Self ID Count Register

The self ID count register keeps a count of the number of times the bus self ID process has occurred, flags self ID packet errors, and keeps a count of the amount of self ID data in the self ID buffer.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Self ID count															
<b>Type</b>	RU	R	R	R	R	R	R	R	RU	RU	RU	RU	RU	RU	RU	RU
<b>Default</b>	X	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Self ID count															
<b>Type</b>	R	R	R	R	R	RU	RU	RU	RU	RU	RU	RU	RU	RU	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Self ID count**  
Type: Read/Update  
Offset: 68h  
Default: X0XX 0000h

**Table 4–12. Self ID Count Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	selfIDError	RU	When this bit is 1, an error was detected during the most recent self ID packet reception. The contents of the self ID buffer are undefined. This bit is cleared after a self ID reception in which no errors are detected. Note that an error can be a hardware error or a host bus write error.
30–24	RSVD	R	Reserved. Bits 30–24 return 0s when read.
23–16	selfIDGeneration	RU	The value in this field increments each time a bus reset is detected. This field rolls over to 0 after reaching 255.
15–11	RSVD	R	Reserved. Bits 15–11 return 0s when read.
10–2	selfIDSize	RU	This field indicates the number of quadlets that have been written into the self ID buffer for the current bits 23–16 (selfIDGeneration field). This includes the header quadlet and the self ID data. This field is cleared to 0 when the self ID reception begins.
1–0	RSVD	R	Reserved. Bits 1–0 return 0s when read.

## 4.19 Isochronous Receive Channel Mask High Register

The isochronous receive channel mask high set/clear register is used to enable packet receives from the upper 32 isochronous data channels. A read from either the set register or clear register returns the content of the isochronous receive channel mask high register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive channel mask high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous receive channel mask high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Isochronous receive channel mask high**

Type: Read/Set/Clear

Offset: 70h set register  
74h clear register

Default: XXXX XXXXh

**Table 4–13. Isochronous Receive Channel Mask High Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	isoChannel63	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 63.
30	isoChannel62	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 62.
29	isoChannel61	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 61.
28	isoChannel60	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 60.
27	isoChannel59	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 59.
26	isoChannel58	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 58.
25	isoChannel57	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 57.
24	isoChannel56	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 56.
23	isoChannel55	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 55.
22	isoChannel54	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 54.
21	isoChannel53	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 53.
20	isoChannel52	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 52.
19	isoChannel51	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 51.
18	isoChannel50	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 50.
17	isoChannel49	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 49.
16	isoChannel48	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 48.
15	isoChannel47	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 47.
14	isoChannel46	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 46.
13	isoChannel45	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 45.
12	isoChannel44	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 44.
11	isoChannel43	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 43.
10	isoChannel42	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 42.
9	isoChannel41	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 41.
8	isoChannel40	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 40.
7	isoChannel39	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 39.

**Table 4–13. Isochronous Receive Channel Mask High Register Description (Continued)**

BIT	FIELD NAME	TYPE	DESCRIPTION
6	isoChannel38	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 38.
5	isoChannel37	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 37.
4	isoChannel36	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 36.
3	isoChannel35	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 35.
2	isoChannel34	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 34.
1	isoChannel33	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 33.
0	isoChannel32	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 32.

## 4.20 Isochronous Receive Channel Mask Low Register

The isochronous receive channel mask low set/clear register is used to enable packet receives from the lower 32 isochronous data channels.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive channel mask low															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous receive channel mask low															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Isochronous receive channel mask low**

Type: Read/Set/Clear

Offset: 78h set register  
7Ch clear register

Default: XXXX XXXXh

**Table 4–14. Isochronous Receive Channel Mask Low Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	isoChannel31	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 31.
30	isoChannel30	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 30.
29	isoChannel29	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 29.
28	isoChannel28	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 28.
27	isoChannel27	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 27.
26	isoChannel26	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 26.
25	isoChannel25	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 25.
24	isoChannel24	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 24.
23	isoChannel23	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 23.
22	isoChannel22	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 22.
21	isoChannel21	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 21.
20	isoChannel20	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 20.
19	isoChannel19	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 19.
18	isoChannel18	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 18.
17	isoChannel17	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 17.
16	isoChannel16	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 16.

**Table 4–14. Isochronous Receive Channel Mask Low Register Description (Continued)**

<b>BIT</b>	<b>FIELD NAME</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
15	isoChannel15	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 15.
14	isoChannel14	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 14.
13	isoChannel13	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 13.
12	isoChannel12	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 12.
11	isoChannel11	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 11.
10	isoChannel10	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 10.
9	isoChannel9	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 9.
8	isoChannel8	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 8.
7	isoChannel7	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 7.
6	isoChannel6	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 6.
5	isoChannel5	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 5.
4	isoChannel4	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 4.
3	isoChannel3	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 3.
2	isoChannel2	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 2.
1	isoChannel1	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 1.
0	isoChannel0	RSC	When this bit is set, the TSB12LV23 is enabled to receive from iso channel number 0.



## 4.21 Interrupt Event Register

The interrupt event set/clear register reflects the state of the various TSB12LV23 interrupt sources. The interrupt bits are set by an asserting edge of the corresponding interrupt signal or by writing a 1 in the corresponding bit in the set register. The only mechanism to clear the bits in this register is to write a 1 to the corresponding bit in the clear register.

This register is fully compliant with OHCI and the TSB12LV23 adds OHCI 1.0 compliant vendor-specific interrupt function to bit 30. When reading the interrupt event register, the return value is the bit-wise AND function of the interrupt event and interrupt mask registers per the *1394 Open Host Controller Interface Specification*.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Interrupt event															
Type	R	RSC	R	R	R	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	R	RSCU	RSCU
Default	0	X	0	0	0	X	X	X	X	X	X	X	X	0	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Interrupt event															
Type	R	R	R	R	R	R	RSCU	RSCU	RU	RU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU
Default	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X

Register: **Interrupt event**  
Type: Read/Set/Clear/Update  
Offset: 80h set register  
84h clear register [returns the content of the interrupt event and interrupt mask registers when read]  
Default: XXXX 0XXXh

**Table 4–15. Interrupt Event Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31	RSVD	R	Reserved. Bit 31 returns 0 when read.
30	vendorSpecific	RSC	This vendor-specific interrupt event is reported when either of the general-purpose interrupts occur which are enabled via INT_EN3 and INT_EN2.
29–27	RSVD	R	Reserved. Bits 29–27 return 0s when read.
26	phyRegRcvd	RSCU	The TSB12LV23 has received a PHY register data byte which can be read from the PHY layer control register (see Section 4.30).
25	cycleTooLong	RSCU	If bit 21 (cycleMaster) of the link control register (see Section 4.28) is set, then this indicates that over 125 $\mu$ s have elapsed between the start of sending a cycle start packet and the end of a subaction gap. The link control register bit 21 (cycleMaster) is cleared by this event.
24	unrecoverableError	RSCU	This event occurs when the TSB12LV23 encounters any error that forces it to stop operations on any or all of its subunits, for example, when a DMA context sets its dead bit. While this bit is set, all normal interrupts for the context(s) that caused this interrupt are blocked from being set.
23	cycleInconsistent	RSCU	A cycle start was received that had values for cycleSeconds and cycleCount fields that are different from the values in bits 31–25 (cycleSeconds field) and bits 24–12 (cycleCount field) of the isochronous cycle timer register (see Section 4.31).
22	cycleLost	RSCU	A lost cycle is indicated when no cycle_start packet is sent/received between two successive cycleSynch events. A lost cycle can be predicted when a cycle_start packet does not immediately follow the first subaction gap after the cycleSynch event or if an arbitration reset gap is detected after a cycleSynch event without an intervening cycle start. This bit may be set either when it occurs or when logic predicts that it will occur.
21	cycle64Seconds	RSCU	Indicates that the 7 <sup>th</sup> bit of the cycle second counter has changed.
20	cycleSynch	RSCU	Indicates that a new isochronous cycle has started. This bit is set when the low order bit of the cycle count toggles.
19	phy	RSCU	Indicates the PHY requests an interrupt through a status transfer.
18	RSVD	R	Reserved. Bit 18 returns 0 when read.

**Table 4–15. Interrupt Event Register Description (Continued)**

<b>BIT</b>	<b>FIELD NAME</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
17	busReset	RSCU	Indicates that the PHY chip has entered bus reset mode.
16	selfIDcomplete	RSCU	A selfID packet stream has been received. It is generated at the end of the bus initialization process. This bit is turned off simultaneously when bit 17 (busReset) is turned on.
15–10	RSVD	R	Reserved. Bits 15–10 return 0s when read.
9	lockRespErr	RSCU	Indicates that the TSB12LV23 sent a lock response for a lock request to a serial bus register, but did not receive an ack_complete.
8	postedWriteErr	RSCU	Indicates that a host bus error occurred while the TSB12LV23 was trying to write a 1394 write request, which had already been given an ack_complete, into system memory.
7	isochRx	RU	Isochronous receive DMA interrupt. Indicates that one or more isochronous receive contexts have generated an interrupt. This is not a latched event, it is the OR'ing of all bits in the isochronous receive interrupt event and isochronous receive interrupt mask registers. The isochronous receive interrupt event register (see Section 4.25) indicates which contexts have interrupted.
6	isochTx	RU	Isochronous transmit DMA interrupt. Indicates that one or more isochronous transmit contexts have generated an interrupt. This is not a latched event, it is the OR'ing of all bits in the isochronous transmit interrupt event and isochronous transmit interrupt mask registers. The isochronous transmit interrupt event register (see Section 4.23) indicates which contexts have interrupted.
5	RSPkt	RSCU	Indicates that a packet was sent to an asynchronous receive response context buffer and the descriptor's xferStatus and resCount fields have been updated.
4	RQPkt	RSCU	Indicates that a packet was sent to an asynchronous receive request context buffer and the descriptor's xferStatus and resCount fields have been updated.
3	ARRS	RSCU	Async receive response DMA interrupt. This bit is conditionally set upon completion of an ARRS DMA context command descriptor.
2	ARRQ	RSCU	Async receive request DMA interrupt. This bit is conditionally set upon completion of an ARRQ DMA context command descriptor.
1	respTxComplete	RSCU	Asynchronous response transmit DMA interrupt. This bit is conditionally set upon completion of an ATRS DMA command.
0	reqTxComplete	RSCU	Asynchronous request transmit DMA interrupt. This bit is conditionally set upon completion of an ATRQ DMA command.

## 4.22 Interrupt Mask Register

The interrupt mask set/clear register is used to enable the various TSB12LV23 interrupt sources. Reads from either the set register or the clear register always return the contents of the interrupt mask register. In all cases except masterIntEnable (bit 31), the enables for each interrupt event align with the interrupt event register bits detailed in Table 4–15.

This register is fully compliant with OHCI and the TSB12LV23 adds an OHCI 1.0 compliant interrupt function to bit 30.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Interrupt mask															
<b>Type</b>	R	RSC	R	R	R	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	R	RSCU	RSCU
<b>Default</b>	X	X	0	0	0	X	X	X	X	X	X	X	X	0	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Interrupt mask															
<b>Type</b>	R	R	R	R	R	R	RSCU	RSCU	RU	RU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU
<b>Default</b>	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X

Register: **Interrupt mask**  
Type: Read/Set/Clear/Update  
Offset: 88h set register  
8Ch clear register  
Default: XXXX 0XXXh

**Table 4–16. Interrupt Mask Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	masterIntEnable	RSCU	Master interrupt enable. If this bit is set, then external interrupts are generated in accordance with the interrupt mask register. If this bit is cleared, then external interrupts are not generated regardless of the interrupt mask register settings.
30	VendorSpecific	RSC	When this bit is set, this vendor-specific interrupt mask enables interrupt generation when bit 30 (vendorSpecific) of the interrupt event register (see Section 4.21) is set.
29–0			See Table 4–15

## 4.23 Isochronous Transmit Interrupt Event Register

The isochronous transmit interrupt event set/clear register reflects the interrupt state of the isochronous transmit contexts. An interrupt is generated on behalf of an isochronous transmit context if an OUTPUT\_LAST command completes and its interrupt bits are set. Upon determining that the interrupt event register isoTx (bit 6) interrupt has occurred, software can check this register to determine which context(s) caused the interrupt. The interrupt bits are set by an asserting edge of the corresponding interrupt signal, or by writing a 1 in the corresponding bit in the set register. The only mechanism to clear the bits in this register is to write a 1 to the corresponding bit in the clear register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous transmit interrupt event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous transmit interrupt event															
Type	R	R	R	R	R	R	R	R	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Isochronous transmit interrupt event**  
Type: Read/Set/Clear  
Offset: 90h set register  
84h clear register [returns IsoXmitEvent and IsoXmitMask when read]  
Default: 0000 00XXh

**Table 4–17. Isochronous Transmit Interrupt Event Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–8	RSVD	R	Reserved. Bits 31–8 return 0s when read.
7	isoXmit7	RSC	Isochronous transmit channel 7 caused the interrupt event register bit 6 (isoTx) interrupt.
6	isoXmit6	RSC	Isochronous transmit channel 6 caused the interrupt event register bit 6 (isoTx) interrupt.
5	isoXmit5	RSC	Isochronous transmit channel 5 caused the interrupt event register bit 6 (isoTx) interrupt.
4	isoXmit4	RSC	Isochronous transmit channel 4 caused the interrupt event register bit 6 (isoTx) interrupt.
3	isoXmit3	RSC	Isochronous transmit channel 3 caused the interrupt event register bit 6 (isoTx) interrupt.
2	isoXmit2	RSC	Isochronous transmit channel 2 caused the interrupt event register bit 6 (isoTx) interrupt.
1	isoXmit1	RSC	Isochronous transmit channel 1 caused the interrupt event register bit 6 (isoTx) interrupt.
0	isoXmit0	RSC	Isochronous transmit channel 0 caused the interrupt event register bit 6 (isoTx) interrupt.

## 4.24 Isochronous Transmit Interrupt Mask Register

The isochronous transmit interrupt mask set/clear register is used to enable the isochTx interrupt source on a per channel basis. Reads from either the set register or the clear register always return the contents of the isochronous transmit interrupt mask register. In all cases the enables for each interrupt event align with the event register bits detailed in Table 4–17.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous transmit interrupt mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous transmit interrupt mask															
Type	R	R	R	R	R	R	R	R	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Isochronous transmit interrupt mask**

Type: Read/Set/Clear

Offset: 98h set register  
9Ch clear register

Default: 0000 00XXh

## 4.25 Isochronous Receive Interrupt Event Register

The isochronous receive interrupt event set/clear register reflects the interrupt state of the isochronous receive contexts. An interrupt is generated on behalf of an isochronous receive context if an INPUT\_\* command completes and its interrupt bits are set. Upon determining that the interrupt event register isochRx (bit 7) interrupt has occurred, software can check this register to determine which context(s) caused the interrupt. The interrupt bits are set by an asserting edge of the corresponding interrupt signal, or by writing a 1 in the corresponding bit in the set register. The only mechanism to clear the bits in this register is to write a 1 to the corresponding bit in the clear register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive interrupt event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous receive interrupt event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X

Register: **Isochronous receive interrupt event**

Type: Read/Set/Clear

Offset: A0h set register  
A4h clear register [returns the contents of isochronous receive interrupt event and isochronous receive mask registers when read]

Default: 0000 000Xh

**Table 4–18. Isochronous Receive Interrupt Event Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–4	RSVD	R	Reserved. Bits 31–4 return 0s when read.
3	isoRecv3	RSC	Isochronous receive channel 3 caused the interrupt event register bit 7 (isochRx) interrupt.
2	isoRecv2	RSC	Isochronous receive channel 2 caused the interrupt event register bit 7 (isochRx) interrupt.
1	isoRecv1	RSC	Isochronous receive channel 1 caused the interrupt event register bit 7 (isochRx) interrupt.
0	isoRecv0	RSC	Isochronous receive channel 0 caused the interrupt event register bit 7 (isochRx) interrupt.

## 4.26 Isochronous Receive Interrupt Mask Register

The isochronous receive interrupt mask register is used to enable the isochRx interrupt source on a per channel basis. Reads from either the set register or the clear register always return the contents of the isochronous receive interrupt mask register. In all cases the enables for each interrupt event align with the event register bits detailed in Table 4–18.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Isochronous receive interrupt mask															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Isochronous receive interrupt mask															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	RSC	RSC	RSC	RSC
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X

Register: **Isochronous receive interrupt mask**

Type: Read/Set/Clear

Offset: A8h set register  
ACh clear register

Default: 0000 000Xh

## 4.27 Fairness Control Register

The fairness control register provides a mechanism by which software can direct the host controller to transmit multiple asynchronous requests during a fairness interval.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Fairness control															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Fairness control															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Fairness control**

Type: Read-only

Offset: DCh

Default: 0000 0000h

**Table 4–19. Fairness Control Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–8	RSVD	R	Reserved. Bits 31–8 return 0s when read.
7–0	pri_req	R/W	This field specifies the maximum number of priority arbitration requests for asynchronous request packets that the link is permitted to make of the PHY during fairness interval.

## 4.28 Link Control Register

The link control set/clear register provides the control flags that enable and configure the link core protocol portions of the TSB12LV23. It contains controls for the receiver and cycle timer.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Link control															
Type	R	R	R	R	R	R	R	R	R	RSC	RSCU	RSC	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	X	X	X	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Link control															
Type	R	R	R	R	R	RSC	RSC	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	X	X	0	0	0	0	0	0	0	0	0

Register: **Link control**  
Type: Read/Set/Clear/Update  
Offset: E0h set register  
E4h clear register  
Default: 00X0 0X00h

**Table 4–20. Link Control Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31–23	RSVD	R	Reserved. Bits 31–23 return 0s when read.
22	cycleSource	RSC	When this bit is set, the cycle timer uses an external source (CYCLEIN) to determine when to roll over the cycle timer. When this bit is cleared, the cycle timer rolls over when the timer reaches 3072 cycles of the 24.576 MHz clock (125 $\mu$ s).
21	cycleMaster	RSCU	When this bit is set, and the PHY has notified the TSB12LV23 that it is root, the TSB12LV23 generates a cycle start packet every time the cycle timer rolls over, based on the setting of bit 22. When this bit is cleared, the OHCI-Lynx accepts received cycle start packets to maintain synchronization with the node which is sending them. This bit is automatically reset when bit 25 (cycleTooLong) of the interrupt event register (see Section 4.21) is set and cannot be set until bit 25 (cycleTooLong) is cleared.
20	CycleTimerEnable	RSC	When this bit is set, the cycle timer offset counts cycles of the 24.576 MHz clock and rolls over at the appropriate time based on the settings of the above bits. When this bit is cleared, the cycle timer offset does not count.
19–11	RSVD	R	Reserved. Bits 19–11 return 0s when read.
10	RcvPhyPkt	RSC	When this bit is set, the receiver accepts incoming PHY packets into the AR request context if the AR request context is enabled. This does not control receipt of self-identification packets.
9	RcvSelfID	RSC	When this bit is set, the receiver accepts incoming self-identification packets. Before setting this bit to 1, software must ensure that the self ID buffer pointer register contains a valid address.
8–0	RSVD	R	Reserved. Bits 8–0 return 0s when read.

## 4.29 Node Identification Register

The node identification register contains the address of the node on which the OHCI-Lynx chip resides, and indicates the valid node number status. The 16-bit combination of the busNumber field (bits 15–6) and the NodeNumber field (bits 5–0) is referred to as the node ID.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Node identification															
<b>Type</b>	RU	RU	R	R	RU	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Node identification															
<b>Type</b>	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RU	RU	RU	RU	RU	RU
<b>Default</b>	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	X

Register: **Node identification**  
 Type: Read/Write/Update  
 Offset: E8h  
 Default: 0000 FFXh

**Table 4–21. Node Identification Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	iDValid	RU	This bit indicates whether or not the TSB12LV23 has a valid node number. It is cleared when a 1394 bus reset is detected and set when the TSB12LV23 receives a new node number from the PHY.
30	root	RU	This bit is set during the bus reset process if the attached PHY is root.
29–28	RSVD	R	Reserved. Bits 29–28 return 0s when read.
27	CPS	RU	Set if the PHY is reporting that cable power status is OK.
26–16	RSVD	R	Reserved. Bits 26–16 return 0s when read.
15–6	busNumber	RWU	This number is used to identify the specific 1394 bus the TSB12LV23 belongs to when multiple 1394-compatible buses are connected via a bridge.
5–0	NodeNumber	RU	This number is the physical node number established by the PHY during self-identification. It is automatically set to the value received from the PHY after the self-identification phase. If the PHY sets the nodeNumber to 63, then software should not set ContextControl.run for either of the AT DMA contexts.



## 4.30 PHY Layer Control Register

The PHY layer control register is used to read or write a PHY register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PHY layer control															
Type	RU	R	R	R	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PHY layer control															
Type	RWU	RWU	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **PHY layer control**  
Type: Read/Write/Update  
Offset: ECh  
Default: 0000 0000h

**Table 4–22. PHY Control Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	rdDone	RU	This bit is cleared to 0 by the TSB12LV23 when either bit 15 (rdReg) or bit 14 (wrReg) is set. This bit is set when a register transfer is received from the PHY.
30–28	RSVD	R	Reserved. Bits 30–28 return 0s when read.
27–24	rdAddr	RU	This is the address of the register most recently received from the PHY.
23–16	rdData	RU	This field is the contents of a PHY register which has been read.
15	rdReg	RWU	This bit is set by software to initiate a read request to a PHY register and is cleared by hardware when the request has been sent. Bits 14 (wrReg) and 15 (rdReg) must be used exclusively.
14	wrReg	RWU	This bit is set by software to initiate a write request to a PHY register and is cleared by hardware when the request has been sent. Bits 14 (wrReg) and 15 (rdReg) must be used exclusively.
13–12	RSVD	R	Reserved. Bits 13–12 return 0s when read.
11–8	regAddr	R/W	This field is the address of the PHY register to be written or read.
7–0	wrData	R/W	This field is the data to be written to a PHY register and is ignored for reads.

## 4.31 Isochronous Cycle Timer Register

The isochronous cycle timer register indicates the current cycle number and offset. When the TSB12LV23 is cycle master, this register is transmitted with the cycle start message. When the TSB12LV23 is not cycle master, this register is loaded with the data field in an incoming cycle start. In the event that the cycle start message is not received, the fields can continue incrementing on their own (if programmed) to maintain a local time reference.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous cycle timer															
Type	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous cycle timer															
Type	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Isochronous cycle timer**  
Type: Read/Write/Update  
Offset: F0h  
Default: XXXX XXXXh

**Table 4–23. Isochronous Cycle Timer Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–25	cycleSeconds	RWU	This field counts seconds [rollovers from bits 24–12 (cycleCount field)] modulo 128.
24–12	cycleCount	RWU	This field counts cycles [rollovers from bits 11–0 (cycleOffset field)] modulo 8000.
11–0	cycleOffset	RWU	This field counts 24.576 MHz clocks modulo 3072, i.e., 125 $\mu$ s. If an external 8 kHz clock configuration is being used, then this bit must be set to 0 at each tick of the external clock.

## 4.32 Asynchronous Request Filter High Register

The asynchronous request filter high set/clear register is used to enable asynchronous receive requests on a per node basis, and handles the upper node IDs. When a packet is destined for either the physical request context or the ARRQ context, the source node ID is examined. If the bit corresponding to the node ID is not set in this register, then the packet is not acknowledged and the request is not queued. The node ID comparison is done if the source node is on the same bus as the TSB12LV23. All nonlocal bus sourced packets are not acknowledged unless bit 31 in this register is set.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Asynchronous request filter high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Asynchronous request filter high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Asynchronous request filter high**  
Type: Read/Set/Clear  
Offset: 100h set register  
104h clear register  
Default: 0000 0000h

**Table 4–24. Asynchronous Request Filter High Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31	asynReqAllBuses	RSC	If this bit is set, then all asynchronous requests received by the TSB12LV23 from non-local bus nodes are accepted.
30	asynReqResource62	RSC	If this bit is set for local bus node number 62, then asynchronous requests received by the TSB12LV23 from that node are accepted.
29	asynReqResource61	RSC	If this bit is set for local bus node number 61, then asynchronous requests received by the TSB12LV23 from that node are accepted.
28	asynReqResource60	RSC	If this bit is set for local bus node number 60, then asynchronous requests received by the TSB12LV23 from that node are accepted.
27	asynReqResource59	RSC	If this bit is set for local bus node number 59, then asynchronous requests received by the TSB12LV23 from that node are accepted.
26	asynReqResource58	RSC	If this bit is set for local bus node number 58, then asynchronous requests received by the TSB12LV23 from that node are accepted.
25	asynReqResource57	RSC	If this bit is set for local bus node number 57, then asynchronous requests received by the TSB12LV23 from that node are accepted.
24	asynReqResource56	RSC	If this bit is set for local bus node number 56, then asynchronous requests received by the TSB12LV23 from that node are accepted.
23	asynReqResource55	RSC	If this bit is set for local bus node number 55, then asynchronous requests received by the TSB12LV23 from that node are accepted.
22	asynReqResource54	RSC	If this bit is set for local bus node number 54, then asynchronous requests received by the TSB12LV23 from that node are accepted.
21	asynReqResource53	RSC	If this bit is set for local bus node number 53, then asynchronous requests received by the TSB12LV23 from that node are accepted.
20	asynReqResource52	RSC	If this bit is set for local bus node number 52, then asynchronous requests received by the TSB12LV23 from that node are accepted.
19	asynReqResource51	RSC	If this bit is set for local bus node number 51, then asynchronous requests received by the TSB12LV23 from that node are accepted.

**Table 4–24. Asynchronous Request Filter High Register Description (Continued)**

<b>BIT</b>	<b>FIELD NAME</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
18	asynReqResource50	RSC	If this bit is set for local bus node number 50, then asynchronous requests received by the TSB12LV23 from that node are accepted.
17	asynReqResource49	RSC	If this bit is set for local bus node number 49, then asynchronous requests received by the TSB12LV23 from that node are accepted.
16	asynReqResource48	RSC	If this bit is set for local bus node number 48, then asynchronous requests received by the TSB12LV23 from that node are accepted.
15	asynReqResource47	RSC	If this bit is set for local bus node number 47, then asynchronous requests received by the TSB12LV23 from that node are accepted.
14	asynReqResource46	RSC	If this bit is set for local bus node number 46, then asynchronous requests received by the TSB12LV23 from that node are accepted.
13	asynReqResource45	RSC	If this bit is set for local bus node number 45, then asynchronous requests received by the TSB12LV23 from that node are accepted.
12	asynReqResource44	RSC	If this bit is set for local bus node number 44, then asynchronous requests received by the TSB12LV23 from that node are accepted.
11	asynReqResource43	RSC	If this bit is set for local bus node number 43, then asynchronous requests received by the TSB12LV23 from that node are accepted.
10	asynReqResource42	RSC	If this bit is set for local bus node number 42, then asynchronous requests received by the TSB12LV23 from that node are accepted.
9	asynReqResource41	RSC	If this bit is set for local bus node number 41, then asynchronous requests received by the TSB12LV23 from that node are accepted.
8	asynReqResource40	RSC	If this bit is set for local bus node number 40, then asynchronous requests received by the TSB12LV23 from that node are accepted.
7	asynReqResource39	RSC	If this bit is set for local bus node number 39, then asynchronous requests received by the TSB12LV23 from that node are accepted.
6	asynReqResource38	RSC	If this bit is set for local bus node number 38, then asynchronous requests received by the TSB12LV23 from that node are accepted.
5	asynReqResource37	RSC	If this bit is set for local bus node number 37, then asynchronous requests received by the TSB12LV23 from that node are accepted.
4	asynReqResource36	RSC	If this bit is set for local bus node number 36, then asynchronous requests received by the TSB12LV23 from that node are accepted.
3	asynReqResource35	RSC	If this bit is set for local bus node number 35, then asynchronous requests received by the TSB12LV23 from that node are accepted.
2	asynReqResource34	RSC	If this bit is set for local bus node number 34, then asynchronous requests received by the TSB12LV23 from that node are accepted.
1	asynReqResource33	RSC	If this bit is set for local bus node number 33, then asynchronous requests received by the TSB12LV23 from that node are accepted.
0	asynReqResource32	RSC	If this bit is set for local bus node number 32, then asynchronous requests received by the TSB12LV23 from that node are accepted.

## 4.33 Asynchronous Request Filter Low Register

The asynchronous request filter low set/clear register is used to enable asynchronous receive requests on a per node basis, and handles the lower node IDs. Other than filtering different node IDs, this register behaves identically to the asynchronous request filter high register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Asynchronous request filter low															
<b>Type</b>	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Asynchronous request filter low															
<b>Type</b>	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Asynchronous request filter low**

Type: Read/Set/Clear

Offset: 108h set register  
10Ch clear register

Default: 0000 0000h

**Table 4–25. Asynchronous Request Filter Low Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31	asynReqResource31	RSC	If this bit is set for local bus node number 31, then asynchronous requests received by the TSB12LV23 from that node are accepted.
30	asynReqResource30	RSC	If this bit is set for local bus node number 30, then asynchronous requests received by the TSB12LV23 from that node are accepted.
29	asynReqResource29	RSC	If this bit is set for local bus node number 29, then asynchronous requests received by the TSB12LV23 from that node are accepted.
28	asynReqResource28	RSC	If this bit is set for local bus node number 28, then asynchronous requests received by the TSB12LV23 from that node are accepted.
27	asynReqResource27	RSC	If this bit is set for local bus node number 27, then asynchronous requests received by the TSB12LV23 from that node are accepted.
26	asynReqResource26	RSC	If this bit is set for local bus node number 26, then asynchronous requests received by the TSB12LV23 from that node are accepted.
25	asynReqResource25	RSC	If this bit is set for local bus node number 25, then asynchronous requests received by the TSB12LV23 from that node are accepted.
24	asynReqResource24	RSC	If this bit is set for local bus node number 24, then asynchronous requests received by the TSB12LV23 from that node are accepted.
23	asynReqResource23	RSC	If this bit is set for local bus node number 23, then asynchronous requests received by the TSB12LV23 from that node are accepted.
22	asynReqResource22	RSC	If this bit is set for local bus node number 22, then asynchronous requests received by the TSB12LV23 from that node are accepted.
21	asynReqResource21	RSC	If this bit is set for local bus node number 21, then asynchronous requests received by the TSB12LV23 from that node are accepted.
20	asynReqResource20	RSC	If this bit is set for local bus node number 20, then asynchronous requests received by the TSB12LV23 from that node are accepted.
19	asynReqResource19	RSC	If this bit is set for local bus node number 19, then asynchronous requests received by the TSB12LV23 from that node are accepted.
18	asynReqResource18	RSC	If this bit is set for local bus node number 18, then asynchronous requests received by the TSB12LV23 from that node are accepted.
17	asynReqResource17	RSC	If this bit is set for local bus node number 17, then asynchronous requests received by the TSB12LV23 from that node are accepted.

**Table 4–25. Asynchronous Request Filter Low Register Description (Continued)**

<b>BIT</b>	<b>FIELD NAME</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
16	asynReqResource16	RSC	If this bit is set for local bus node number 16, then asynchronous requests received by the TSB12LV23 from that node are accepted.
15	asynReqResource15	RSC	If this bit is set for local bus node number 15, then asynchronous requests received by the TSB12LV23 from that node are accepted.
14	asynReqResource14	RSC	If this bit is set for local bus node number 14, then asynchronous requests received by the TSB12LV23 from that node are accepted.
13	asynReqResource13	RSC	If this bit is set for local bus node number 13, then asynchronous requests received by the TSB12LV23 from that node are accepted.
12	asynReqResource12	RSC	If this bit is set for local bus node number 12, then asynchronous requests received by the TSB12LV23 from that node are accepted.
11	asynReqResource11	RSC	If this bit is set for local bus node number 11, then asynchronous requests received by the TSB12LV23 from that node are accepted.
10	asynReqResource10	RSC	If this bit is set for local bus node number 10, then asynchronous requests received by the TSB12LV23 from that node are accepted.
9	asynReqResource9	RSC	If this bit is set for local bus node number 9, then asynchronous requests received by the TSB12LV23 from that node are accepted.
8	asynReqResource8	RSC	If this bit is set for local bus node number 8, then asynchronous requests received by the TSB12LV23 from that node are accepted.
7	asynReqResource7	RSC	If this bit is set for local bus node number 7, then asynchronous requests received by the TSB12LV23 from that node are accepted.
6	asynReqResource6	RSC	If this bit is set for local bus node number 6, then asynchronous requests received by the TSB12LV23 from that node are accepted.
5	asynReqResource5	RSC	If this bit is set for local bus node number 5, then asynchronous requests received by the TSB12LV23 from that node are accepted.
4	asynReqResource4	RSC	If this bit is set for local bus node number 4, then asynchronous requests received by the TSB12LV23 from that node are accepted.
3	asynReqResource3	RSC	If this bit is set for local bus node number 3, then asynchronous requests received by the TSB12LV23 from that node are accepted.
2	asynReqResource2	RSC	If this bit is set for local bus node number 2, then asynchronous requests received by the TSB12LV23 from that node are accepted.
1	asynReqResource1	RSC	If this bit is set for local bus node number 1, then asynchronous requests received by the TSB12LV23 from that node are accepted.
0	asynReqResource0	RSC	If this bit is set for local bus node number 0, then asynchronous requests received by the TSB12LV23 from that node are accepted.

## 4.34 Physical Request Filter High Register

The physical request filter high set/clear register is used to enable physical receive requests on a per node basis and handles the upper node IDs. When a packet is destined for the physical request context and the node ID has been compared against the ARRQ registers, then the comparison is done again with this register. If the bit corresponding to the node ID is not set in this register, then the request is handled by the ARRQ context instead of the physical request context.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Physical request filter high															
<b>Type</b>	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Physical request filter high															
<b>Type</b>	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Physical request filter high**

Type: Read/Set/Clear

Offset: 110h set register  
114h clear register

Default: 0000 0000h

**Table 4–26. Physical Request Filter High Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31	physReqAllBusses	RSC	If this bit is set, then all asynchronous requests received by the TSB12LV23 from non-local bus nodes are accepted.
30	physReqResource62	RSC	If this bit is set for local bus node number 62, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
29	physReqResource61	RSC	If this bit is set for local bus node number 61, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
28	physReqResource60	RSC	If this bit is set for local bus node number 60, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
27	physReqResource59	RSC	If this bit is set for local bus node number 59, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
26	physReqResource58	RSC	If this bit is set for local bus node number 58, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
25	physReqResource57	RSC	If this bit is set for local bus node number 57, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
24	physReqResource56	RSC	If this bit is set for local bus node number 56, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
23	physReqResource55	RSC	If this bit is set for local bus node number 55, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
22	physReqResource54	RSC	If this bit is set for local bus node number 54, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
21	physReqResource53	RSC	If this bit is set for local bus node number 53, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
20	physReqResource52	RSC	If this bit is set for local bus node number 52, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
19	physReqResource51	RSC	If this bit is set for local bus node number 51, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
18	physReqResource50	RSC	If this bit is set for local bus node number 50, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.

**Table 4–26. Physical Request Filter High Register Description (Continued)**

<b>BIT</b>	<b>FIELD NAME</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
17	physReqResource49	RSC	If this bit is set for local bus node number 49, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
16	physReqResource48	RSC	If this bit is set for local bus node number 48, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
15	physReqResource47	RSC	If this bit is set for local bus node number 47, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
14	physReqResource46	RSC	If this bit is set for local bus node number 46, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
13	physReqResource45	RSC	If this bit is set for local bus node number 45, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
12	physReqResource44	RSC	If this bit is set for local bus node number 44, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
11	physReqResource43	RSC	If this bit is set for local bus node number 43, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
10	physReqResource42	RSC	If this bit is set for local bus node number 42, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
9	physReqResource41	RSC	If this bit is set for local bus node number 41, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
8	physReqResource40	RSC	If this bit is set for local bus node number 40, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
7	physReqResource39	RSC	If this bit is set for local bus node number 39, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
6	physReqResource38	RSC	If this bit is set for local bus node number 38, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
5	physReqResource37	RSC	If this bit is set for local bus node number 37, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
4	physReqResource36	RSC	If this bit is set for local bus node number 36, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
3	physReqResource35	RSC	If this bit is set for local bus node number 35, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
2	physReqResource34	RSC	If this bit is set for local bus node number 34, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
1	physReqResource33	RSC	If this bit is set for local bus node number 33, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
0	physReqResource32	RSC	If this bit is set for local bus node number 32, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.



## 4.35 Physical Request Filter Low Register

The physical request filter low set/clear register is used to enable physical receive requests on a per node basis and handles the lower node IDs. When a packet is destined for the physical request context and the node ID has been compared against the asynchronous request filter registers, then the node ID comparison is done again with this register. If the bit corresponding to the node ID is not set in this register, then the request is handled by the asynchronous request context instead of the physical request context.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Physical request filter low															
<b>Type</b>	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Physical request filter low															
<b>Type</b>	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
<b>Default</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Physical request filter low**  
Type: Read/Set/Clear  
Offset: 118h set register  
11Ch clear register  
Default: 0000 0000h

**Table 4–27. Physical Request Filter Low Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31	physReqResource31	RSC	If this bit is set for local bus node number 31, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
30	physReqResource30	RSC	If this bit is set for local bus node number 30, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
29	physReqResource29	RSC	If this bit is set for local bus node number 29, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
28	physReqResource28	RSC	If this bit is set for local bus node number 28, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
27	physReqResource27	RSC	If this bit is set for local bus node number 27, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
26	physReqResource26	RSC	If this bit is set for local bus node number 26, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
25	physReqResource25	RSC	If this bit is set for local bus node number 25, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
24	physReqResource24	RSC	If this bit is set for local bus node number 24, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
23	physReqResource23	RSC	If this bit is set for local bus node number 23, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
22	physReqResource22	RSC	If this bit is set for local bus node number 22, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
21	physReqResource21	RSC	If this bit is set for local bus node number 21, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
20	physReqResource20	RSC	If this bit is set for local bus node number 20, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
19	physReqResource19	RSC	If this bit is set for local bus node number 19, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
18	physReqResource18	RSC	If this bit is set for local bus node number 18, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.

**Table 4–27. Physical Request Filter Low Register Description (Continued)**

<b>BIT</b>	<b>FIELD NAME</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
17	physReqResource17	RSC	If this bit is set for local bus node number 17, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
16	physReqResource16	RSC	If this bit is set for local bus node number 16, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
15	physReqResource15	RSC	If this bit is set for local bus node number 15, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
14	physReqResource14	RSC	If this bit is set for local bus node number 14, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
13	physReqResource13	RSC	If this bit is set for local bus node number 13, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
12	physReqResource12	RSC	If this bit is set for local bus node number 12, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
11	physReqResource11	RSC	If this bit is set for local bus node number 11, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
10	physReqResource10	RSC	If this bit is set for local bus node number 10, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
9	physReqResource9	RSC	If this bit is set for local bus node number 9, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
8	physReqResource8	RSC	If this bit is set for local bus node number 8, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
7	physReqResource7	RSC	If this bit is set for local bus node number 7, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
6	physReqResource6	RSC	If this bit is set for local bus node number 6, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
5	physReqResource5	RSC	If this bit is set for local bus node number 5, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
4	physReqResource4	RSC	If this bit is set for local bus node number 4, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
3	physReqResource3	RSC	If this bit is set for local bus node number 3, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
2	physReqResource2	RSC	If this bit is set for local bus node number 2, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
1	physReqResource1	RSC	If this bit is set for local bus node number 1, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.
0	physReqResource0	RSC	If this bit is set for local bus node number 0, then physical requests received by the TSB12LV23 from that node are handled through the physical request context.

## 4.36 Physical Upper Bound Register (Optional Register)

The physical upper bound register is an optional register and is not implemented. It returns all 0s when read.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Physical upper bound															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Physical upper bound															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Physical upper bound**  
 Type: Read-only  
 Offset: 120h  
 Default: 0000 0000h

## 4.37 Asynchronous Context Control Register

The asynchronous context control set/clear register controls the state and indicates status of the DMA context.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Asynchronous context control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Asynchronous context control															
Type	RSCU	R	R	RSU	RU	RU	R	R	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	X	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Asynchronous context control**  
Type: Read/Set/Clear/Update  
Offset: 180h set register [ATRQ]  
184h clear register [ATRQ]  
1A0h set register [ATRS]  
1A4h clear register [ATRS]  
1C0h set register [ARRQ]  
1C4h clear register [ARRQ]  
1E0h set register [ATRS]  
1E4h clear register [ATRS]  
Default: 0000 X0XXh

**Table 4–28. Asynchronous Context Control Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31–16	RSVD	R	Reserved. Bits 31–16 return 0s when read.
15	run	RSCU	This bit is set by software to enable descriptor processing for the context and cleared by software to stop descriptor processing. The TSB12LV23 changes this bit only on a hardware or software reset.
14–13	RSVD	R	Reserved. Bits 14–13 return 0s when read.
12	wake	RSU	Software sets this bit to cause the TSB12LV23 to continue or resume descriptor processing. The TSB12LV23 clears this bit on every descriptor fetch.
11	dead	RU	The TSB12LV23 sets this bit when it encounters a fatal error and clears the bit when software resets bit 15 (run).
10	active	RU	The TSB12LV23 sets this bit to 1 when it is processing descriptors.
9–8	RSVD	R	Reserved. Bits 9–8 return 0s when read.
7–5	spd	RU	This field indicates the speed at which a packet was received or transmitted, and only contains meaningful information for receive contexts. This field is encoded as: 000 = 100 Mbits/sec, 001 = 200 Mbits/sec, 010 = 400 Mbits/sec, and all other values are reserved.
4–0	eventcode	RU	This field holds the acknowledge sent by the Link core for this packet or an internally generated error code if the packet was not transferred successfully.

## 4.38 Asynchronous Context Command Pointer Register

The asynchronous context command pointer register contains a pointer to the address of the first descriptor block that the TSB12LV23 accesses when software enables the context by setting the asynchronous context control register bit 15 (run).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Asynchronous context command pointer															
Type	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Asynchronous context command pointer															
Type	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Asynchronous context command pointer**

Type: Read/Write/Update

Offset: 19Ch [ATRQ]

1ACh [ATRS]

1CCh [ATRQ]

1ECh [ATRS]

Default: XXXX XXXXh

**Table 4–29. Asynchronous Context Command Pointer Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31–4	descriptorAddress	RWU	Contains the upper 28 bits of the address of a 16-byte aligned descriptor block.
3–0	Z	RWU	Indicates the number of contiguous descriptors at the address pointed to by the descriptor address. If Z is 0, then it indicates that the descriptorAddress field (bits 31–4) is not valid.

## 4.39 Isochronous Transmit Context Control Register

The isochronous transmit context control set/clear register controls options, state, and status for the isochronous transmit DMA contexts. The *n* value in the following register addresses indicates the context number (*n* = 0, 1, 2, 3,...).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Isochronous transmit context control															
<b>Type</b>	RSCU	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
<b>Default</b>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Isochronous transmit context control															
<b>Type</b>	RSC	R	R	RSU	RU	RU	R	R	RU	RU	RU	RU	RU	RU	RU	RU
<b>Default</b>	0	0	0	X	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Isochronous transmit context control**

Type: Read/Set/Clear/Update

Offset: 200h + (16 \* *n*) set register  
 204h + (16 \* *n*) clear register

Default: XXXX X0XXh

**Table 4–30. Isochronous Transmit Context Control Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	cycleMatchEnable	RSCU	When this bit is set to 1, processing occurs such that the packet described by the context's first descriptor block is transmitted in the cycle whose number is specified in the cycleMatch field (bits 30–16). The cycleMatch field (bits 30–16) must match the low order 2 bits of cycleSeconds and the 13-bit cycleCount field in the cycle start packet that is sent or received immediately before isochronous transmission begins. Since the isochronous transmit DMA controller may work ahead, the processing of the first descriptor block may begin slightly in advance of the actual cycle in which the first packet is transmitted. The effects of this bit, however, are impacted by the values of other bits in this register and are explained in the <i>1394 Open Host Controller Interface Specification</i> . Once the context has become active, hardware clears this bit.
30–16	cycleMatch	RSC	Contains a 15-bit value, corresponding to the low order 2 bits of the bus isochronous cycle timer register cycleSeconds field (bits 31–25) and the cycleCount field (bits 24–12). If bit 31 (cycleMatchEnable) is set, then this isochronous transmit DMA context becomes enabled for transmits when the low order 2 bits of the bus isochronous cycle timer register cycleSeconds field (bits 31–25) and the cycleCount field (bits 24–12) value equal this field's (cycleMatch) value.
15	run	RSC	This bit is set by software to enable descriptor processing for the context and cleared by software to stop descriptor processing. The TSB12LV23 changes this bit only on a hardware or software reset.
14–13	RSVD	R	Reserved. Bits 14–13 return 0s when read.
12	wake	RSU	Software sets this bit to cause the TSB12LV23 to continue or resume descriptor processing. The TSB12LV23 clears this bit on every descriptor fetch.
11	dead	RU	The TSB12LV23 sets this bit when it encounters a fatal error and clears the bit when software resets bit 15 (run).
10	active	RU	The TSB12LV23 sets this bit to 1 when it is processing descriptors.
9–8	RSVD	R	Reserved. Bits 9–8 return 0s when read.
7–5	spd	RU	This field is not meaningful for isochronous transmit contexts.
4–0	event code	RU	Following an OUTPUT_LAST* command, the error code is indicated in this field. Possible values are: ack_complete, evt_descriptor_read, evt_data_read, and evt_unknown.

## 4.40 Isochronous Transmit Context Command Pointer Register

The isochronous transmit context command pointer register contains a pointer to the address of the first descriptor block that the TSB12LV23 accesses when software enables an isochronous transmit context by setting the isochronous transmit context control register bit 15 (run). The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3,...).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous transmit context command pointer															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous transmit context command pointer															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Isochronous transmit context command pointer**  
Type: Read-only  
Offset: 20Ch + (16 \* n)  
Default: XXXX XXXXh

## 4.41 Isochronous Receive Context Control Register

The isochronous receive context control set/clear register controls options, state, and status for the isochronous receive DMA contexts. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3,...).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive context control															
Type	RSC	RSC	RSCU	RSC	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous receive context control															
Type	RSCU	R	R	RSU	RU	RU	R	R	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	X	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Isochronous receive context control**  
Type: Read/Set/Clear/Update  
Offset: 400h + (32 \* n)      set register  
404h + (32 \* n)      clear register  
Default: X000 X0XXh

**Table 4–31. Isochronous Receive Context Control Register Description**

BIT	FIELD NAME	TYPE	DESCRIPTION
31	bufferFill	RSC	When this bit is set, received packets are placed back-to-back to completely fill each receive buffer. When this bit is cleared, each received packet is placed in a single buffer. If bit 28 (multiChanMode) is set to 1, then this bit must also be set to 1. The value of this bit must not be changed while bit 10 (active) or bit 15 (run) is set.
30	isochHeader	RSC	When this bit is 1, received isochronous packets include the complete 4-byte isochronous packet header seen by the link layer. The end of the packet is marked with a xferStatus in the first doublet, and a 16-bit timeStamp indicating the time of the most recently received (or sent) cycleStart packet. When this bit is cleared, the packet header is stripped off of received isochronous packets. The packet header, if received, immediately precedes the packet payload. The value of this bit must not be changed while bit 10 (active) or bit 15 (run) is set.

**Table 4–31. Isochronous Receive Context Control Register Description (Continued)**

BIT	FIELD NAME	TYPE	DESCRIPTION
29	cycleMatchEnable	RSCU	When this bit is set, the context begins running only when the 13-bit cycleMatch field (bits 24–12) in the isochronous receive context match register matches the 13-bit cycleCount field in the cycleStart packet. The effects of this bit, however, are impacted by the values of other bits in this register. Once the context has become active, hardware clears this bit. The value of this bit must not be changed while bit 10 (active) or bit 15 (run) is set.
28	multiChanMode	RSC	When this bit is set, the corresponding isochronous receive DMA context receives packets for all isochronous channels enabled in the isochronous receive channel mask high and isochronous receive channel mask low registers. The isochronous channel number specified in the isochronous receive DMA context match register is ignored. When this bit is cleared, the isochronous receive DMA context receives packets for that single channel. Only one isochronous receive DMA context may use the isochronous receive channel mask registers. If more than one isochronous receive context control register has this bit set, then results are undefined. The value of this bit must not be changed while bit 10 (active) or bit 15 (run) is set to 1.
27–16	RSVD	R	Reserved. Bits 27–16 return 0s when read.
15	run	RSCU	This bit is set by software to enable descriptor processing for the context and cleared by software to stop descriptor processing. The TSB12LV23 changes this bit only on a hardware or software reset.
14–13	RSVD	R	Reserved. Bits 14–13 return 0s when read.
12	wake	RSU	Software sets this bit to cause the TSB12LV23 to continue or resume descriptor processing. The TSB12LV23 clears this bit on every descriptor fetch.
11	dead	RU	The TSB12LV23 sets this bit when it encounters a fatal error and clears the bit when software resets bit 15 (run).
10	active	RU	The TSB12LV23 sets this bit to 1 when it is processing descriptors.
9–8	RSVD	R	Reserved. Bits 9–8 return 0s when read.
7–5	spd	RU	This field indicates the speed at which the packet was received. 000 = 100 Mb/s/sec, 001 = 200 Mb/s/sec, and 010 = 400 Mb/s/sec. All other values are reserved.
4–0	event code	RU	Following an INPUT* command, the error code is indicated in this field.

## 4.42 Isochronous Receive Context Command Pointer Register

The isochronous receive context command pointer register contains a pointer to the address of the first descriptor block that the TSB12LV23 accesses when software enables an isochronous receive context by setting the isochronous receive context control register bit 15 (run). The *n* value in the following register addresses indicates the context number (*n* = 0, 1, 2, 3,...).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Isochronous receive context command pointer															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
<b>Bit</b>	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Isochronous receive context command pointer															
<b>Type</b>	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Default</b>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Isochronous receive context command pointer**  
Type: Read-only  
Offset: 40Ch + (32 \* *n*)  
Default: XXXX XXXXh



## 4.43 Isochronous Receive Context Match Register

The isochronous receive context match register is used to start an isochronous receive context running on a specified cycle number, to filter incoming isochronous packets based on tag values, and to wait for packets with a specified sync value. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3,...).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
<b>Name</b>	Isochronous receive context match															
<b>Type</b>	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	X	X	X	X	0	0	0	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Name</b>	Isochronous receive context match															
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Default</b>	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X

Register: **Isochronous receive context match**

Type: Read/Write

Offset: 410Ch + (32 \* n)

Default: XXXX XXXXh

**Table 4–32. Isochronous Receive Context Match Register Description**

BITS	FIELD NAME	TYPE	DESCRIPTION
31	tag3	R/W	If this bit is set, then this context matches on iso receive packets with a tag field of 11b.
30	tag2	R/W	If this bit is set, then this context matches on iso receive packets with a tag field of 10b.
29	tag1	R/W	If this bit is set, then this context matches on iso receive packets with a tag field of 01b.
28	tag0	R/W	If this bit is set, then this context matches on iso receive packets with a tag field of 00b.
27–25	RSVD	R	Reserved. Bits 27–25 return 0s when read.
24–12	cycleMatch	R/W	Contains a 15-bit value, corresponding to the low order 2 bits of cycleSeconds and the 13-bit cycleCount field in the cycleStart packet. If isochronous receive context control register bit 29 (cycleMatchEnable) is set, then this context is enabled for receives when the 2 low order bits of the bus isochronous cycle timer register cycleSeconds field (bits 31–25) and cycleCount field (bits 24–12) value equal this field's (cycleMatch) value.
11–8	sync	R/W	This field contains the 4-bit field which is compared to the sync field of each iso packet for this channel when the command descriptor's w field is set to 11b.
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	tag1SyncFilter	R/W	If this bit and bit 29 (tag1) are set, then packets with tag2b01 are accepted into the context if the two most significant bits of the packets sync field are 00b. Packets with tag values other than 01b are filtered according to tag0, tag2, and tag3 (bits 28, 30, and 31, respectively) without any additional restrictions.  If this bit is cleared, then this context matches on isochronous receive packets as specified in bits 28–31 (tag0–tag3) with no additional restrictions.
5–0	channelNumber	R/W	This 6-bit field indicates the isochronous channel number for which this isochronous receive DMA context accepts packets.

## 5 GPIO Interface

The general-purpose input/output (GPIO) interface consists of two GPIO ports. GPIO2 and GPIO3 power up as general-purpose inputs and are programmable via the GPIO control register. Figure 5–1 shows the schematic for GPIO2 and GPIO3 implementation.

GPIO0 and GPIO1 are not implemented in the TSB12LV23. The terminals for these legacy GPIOs from the TSB12LV23 have been dedicated to BMC/LINKON and LPS, respectively.

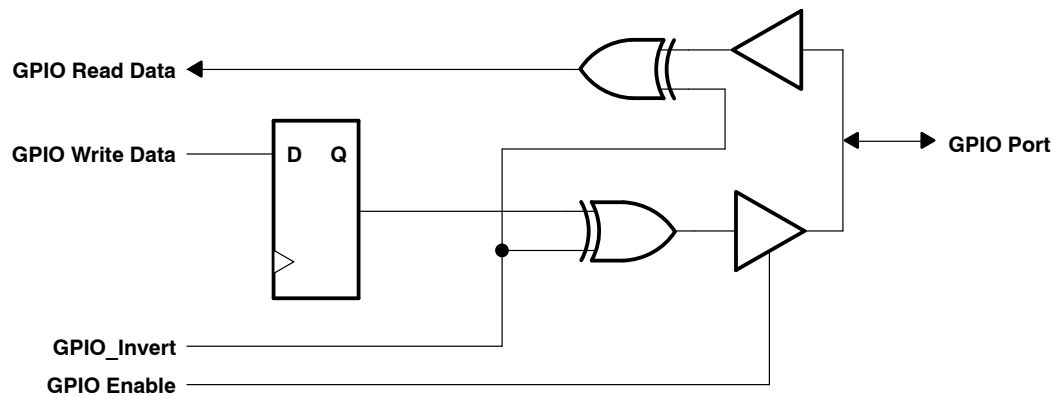


Figure 5–1. GPIO2 and GPIO3



## 6 Serial EEPROM Interface

The TSB12LV23 provides a serial bus interface to initialize the 1394 global unique ID register and a few PCI configuration registers through a serial EEPROM. The TSB12LV23 communicates with the serial EEPROM via the 2-wire serial interface.

After power-up the serial interface initializes the locations listed in Table 6–1. While the TSB12LV23 is accessing the serial ROM, all incoming PCI slave accesses are terminated with retry status. Table 6–2 shows the serial ROM memory map required for initializing the TSB12LV23 registers.

**Table 6–1. Registers and Bits Loadable through Serial EEPROM**

OFFSET	REGISTER	BITS LOADED FROM EEPROM
OHCI register (24h)	1394 GlobalUniqueIDHi	31–0
OHCI register(28h)	1394 GlobalUniqueIDLo	31–0
OHCI register (50h)	Host control register	23
PCI register (2Ch)	PCI subsystem ID	15–0
PCI register (2Dh)	PCI vendor ID	15–0
PCI register (3Eh)	PCI maximum latency, PCI minimum grant	15–0
PCI register (F4h)	Link enhancements control register	13, 12, 9, 8, 7, 2, 1
PCI register (F0h)	PCI miscellaneous register	15, 13, 10, 5–0
PCI register (40h)	PCI OHCI register	0

**Table 6–2. Serial EEPROM Map**

BYTE ADDRESS	BYTE DESCRIPTION							
00	PCI maximum latency (0h)				PCI_minimum grant (0h)			
01	PCI vendor ID							
02	PCI vendor ID (msbyte)							
03	PCI subsystem ID (lsbyte)							
04	PCI subsystem ID							
05	[7] Link_enhancement- Control.enab_unfair	[6] HCControl. ProgramPhy Enable	[5] RSVD	[4] RSVD	[3] RSVD	[2] Link_enhancement- Control.enab_ insert_idle	[1] Link_enhancement- Control.enab_accel	[0] RSVD
06	Mini ROM address							
07	1394 GlobalUniqueIDHi (lsbyte 0)							
08	1394 GlobalUniqueIDHi (byte 1)							
09	1394 GlobalUniqueIDHi (byte 2)							
0A	1394 GlobalUniqueIDHi (msbyte 3)							
0B	1394 GlobalUniqueIDLo (lsbyte 0)							
0C	1394 GlobalUniqueIDLo (byte 1)							
0D	1394 GlobalUniqueIDLo (byte 2)							
0E	1394 GlobalUniqueIDLo (msbyte 3)							
0F	Checksum							
10	[15] RSVD	[14] RSVD	[13–12] AT threshold		[11] RSVD	[10] RSVD	[9] RSVD	[8] RSVD
11	[7] RSVD	[6] RSVD	[5] RSVD	[4] Disable Target Abort	[3] GP2IIC	[2] Disable SCLK gate	[1] Disable PCI gate	[0] Keep PCI
12	[15] PME D3 Cold	[14] RSVD	[13] PME Support D2	[12] RSVD	[11] RSVD	[10] D2 support	[9] RSVD	[8] RSVD
13	[7] RSVD	[6] RSVD	[5] RSVD	[4] RSVD	[3] RSVD	[2] RSVD	[1] RSVD	[0] Global swap
14	CIS offset address							
15–1E	RSVD							
1F	RSVD							

## 7 Electrical Characteristics

### 7.1 Absolute Maximum Ratings Over Operating Temperature Ranges<sup>†</sup>

Supply voltage range, $V_{CC}$	–0.5 V to 3.6 V
Supply voltage range, $V_{CCP}$	–0.5 V to 5.5 V
Input voltage range for PCI, $V_I$	–0.5 to $V_{CCP} + 0.5$ V
Input voltage range for Miscellaneous and PHY interface, $V_I$	–0.5 to $V_{CCI} + 0.5$ V
Output voltage range for PCI, $V_O$	–0.5 to $V_{CCP} + 0.5$ V
Input voltage range for Miscellaneous and PHY interface, $V_O$	–0.5 to $V_{CCP} + 0.5$ V
Input clamp current, $I_{IK}$ ( $V_I < 0$ or $V_I > V_{CC}$ ) (see Note 1)	±20 mA
Output clamp current, $I_{OK}$ ( $V_O < 0$ or $V_O > V_{CC}$ ) (see Note 2)	±20 mA
Storage temperature range	–65°C to 150°C

<sup>†</sup> 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. Applies to external input and bidirectional buffers. For 5-V tolerant use  $V_I > V_{CC}$ . For PCI use  $V_I > V_{CCP}$ .  
 2. Applies to external output and bidirectional buffers. For 5-V tolerant use  $V_O > V_{CC}$ . For PCI use  $V_O > V_{CCP}$ .

## 7.2 Recommended Operating Conditions

			OPERATION	MIN	NOM	MAX	UNIT
V <sub>CC</sub>	Core voltage	Commercial	3.3 V	3	3.3	3.6	V
V <sub>CCP</sub>	PCI I/O clamping voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.5	5	5.5	
V <sub>IH</sub> <sup>†</sup>	High-level input voltage	PCI	3.3 V	0.475 V <sub>CCP</sub>		V <sub>CCP</sub>	V
			5 V	2		V <sub>CCP</sub>	
		PHY interface		2		V <sub>CC</sub>	
		Miscellaneous <sup>‡</sup>		2		V <sub>CC</sub>	
V <sub>IL</sub> <sup>†</sup>	Low-level input voltage	PCI	3.3 V	0		0.325 V <sub>CCP</sub>	V
			5 V	0		0.8	
		PHY interface		0		0.8	
		Miscellaneous <sup>‡</sup>		0		0.8	
V <sub>I</sub>	Input voltage	PCI	3.3 V	0		V <sub>CCP</sub>	V
		PHY interface		0		V <sub>CCP</sub>	
		Miscellaneous <sup>‡</sup>		0		V <sub>CCP</sub>	
V <sub>O</sub> <sup>§</sup>	Output voltage	PCI	3.3 V	0		V <sub>CCP</sub>	V
		PHY interface		0		V <sub>CCP</sub>	
		Miscellaneous <sup>‡</sup>		0		V <sub>CCP</sub>	
t <sub>t</sub>	Input transition time (t <sub>r</sub> and t <sub>f</sub> )	PCI		0		6	ns
T <sub>A</sub>	Operating ambient temperature			0	25	70	°C
T <sub>J</sub> <sup>¶</sup>	Virtual junction temperature			0	25	115	°C

<sup>†</sup> Applies for external inputs and bidirectional buffers without hysteresis.

<sup>‡</sup> Miscellaneous pins are: GPIO2, GPIO3, SDA, SCL, CYCLEOUT.

<sup>§</sup> Applies for external output buffers.

<sup>¶</sup> The junction temperatures reflect simulation conditions. Customer is responsible for verifying junction temperature.

### 7.3 Electrical Characteristics Over Recommended Operating Conditions (unless otherwise noted)

		OPERATION	TEST CONDITIONS	MIN	MAX	UNIT
$V_{OH}$ High-level output voltage	PCI		$I_{OH} = -0.5 \text{ mA}$	$0.9 V_{CC}$		V
			$I_{OH} = -2 \text{ mA}$	2.4		
	PHY interface		$I_{OH} = -4 \mu\text{A}$	2.8		
			$I_{OH} = -8 \text{ mA}$	$V_{CC} - 0.6$		
	Miscellaneous <sup>‡</sup>		$I_{OH} = -4 \text{ mA}$	$V_{CC} - 0.6$		
$V_{OL}^{\dagger}$ Low-level output voltage	PCI		$I_{OL} = 1.5 \text{ mA}$	$0.1 V_{CC}$		V
			$I_{OL} = 6 \text{ mA}$	0	0.55	
	PHY interface		$I_{OL} = 4 \text{ mA}$		0.4	
			$I_{OL} = 8 \text{ mA}$			
	Miscellaneous <sup>‡</sup>		$I_{OL} = 4 \text{ mA}$		0.5	
$I_{OZ}$ 3-state output high-impedance	Output pins	3.6 V	$V_O = V_{CC} \text{ or GND}$		$\pm 20$	$\mu\text{A}$
$I_{IL}$ Low-level input current	Input pins	3.6 V	$V_I = \text{GND}^{\dagger}$		$\pm 20$	$\mu\text{A}$
	I/O pins <sup>†</sup>	3.6 V	$V_I = \text{GND}^{\dagger}$	$\pm 20$		
$I_{IH}$ High-level input current	PCI <sup>†</sup>	3.6 V	$V_I = V_{CC}^{\dagger}$		$\pm 20$	$\mu\text{A}$
	Others <sup>†</sup>	3.6 V	$V_I = V_{CC}^{\dagger}$	$\pm 20$		

<sup>†</sup> For I/O pins, input leakage ( $I_{IL}$  and  $I_{IH}$ ) includes  $I_{OZ}$  of the disabled output.

<sup>‡</sup> Miscellaneous pins are: GPIO2, GPIO3, SDA, SCL, CYCLEOUT.

### 7.4 Switching Characteristics for PCI Interface<sup>§</sup>

PARAMETER		MEASURED	MIN	TYP	MAX	UNIT
$t_{su}$	Setup time before PCLK	-50% to 50%	7			ns
$t_h$	Hold time before PCLK	-50% to 50%	0			ns
$t_d$	Delay time, PHY_CLK to data valid	-50% to 50%	2		11	ns

<sup>§</sup> These parameters are ensured by design.

### 7.5 Switching Characteristics for PHY-Link Interface<sup>§</sup>

PARAMETER		MEASURED	MIN	TYP	MAX	UNIT
$t_{su}$	Setup time, Dn, CTLn, LREQ to PHY_CLK	-50% to 50%	6			ns
$t_h$	Hold time, Dn, CTLn, LREQ before PHY_CLK	-50% to 50%	1			ns
$t_d$	Delay time, PHY_CLK to Dn, CTLn	-50% to 50%	2		11	ns

<sup>§</sup> These parameters are ensured by design.



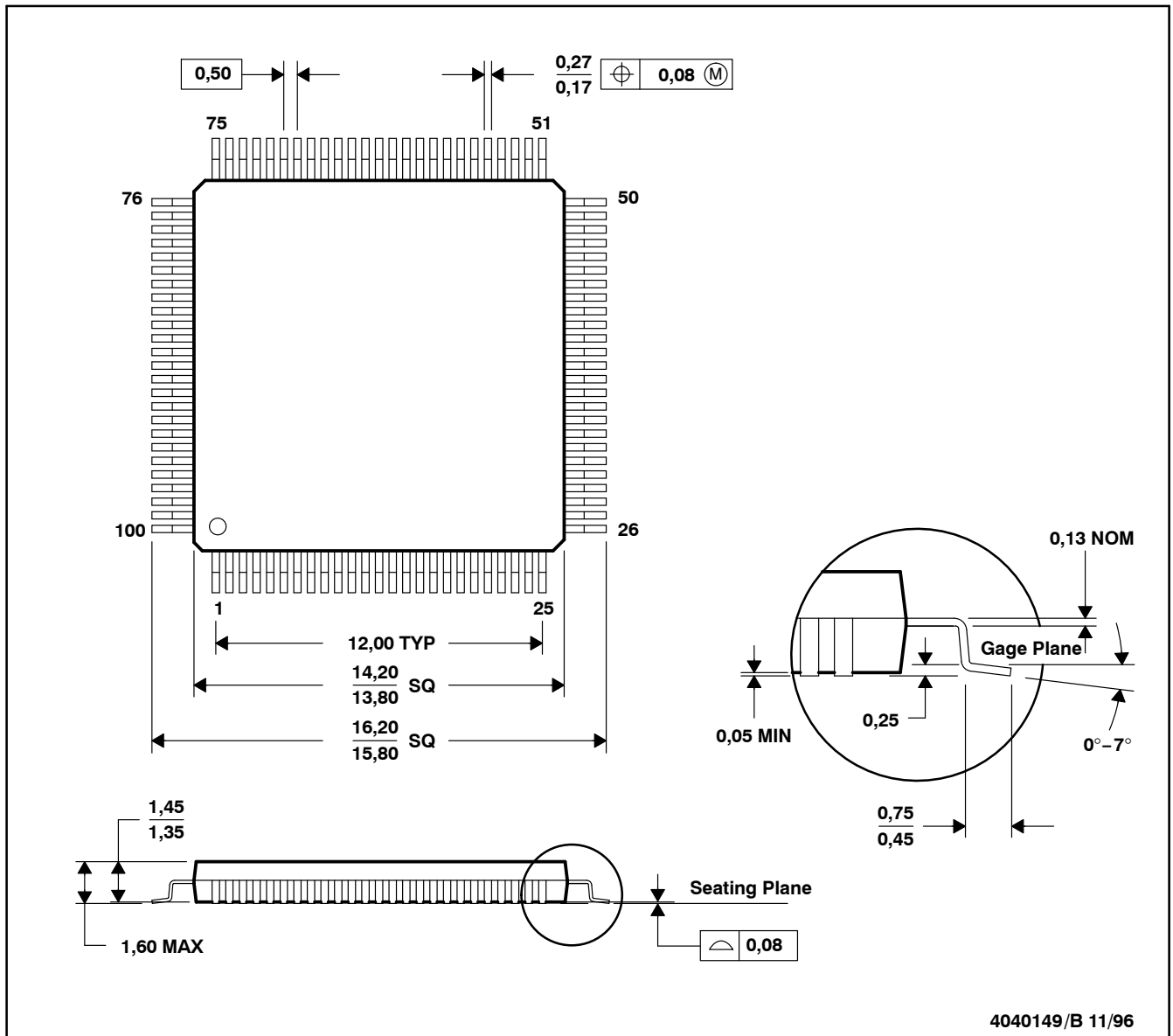


## 8 Mechanical Information

The TSB12LV23 is packaged in a 100-pin PZ package. The following shows the mechanical dimensions for the PZ package.

**PZ (S-PQFP-G100)**

**PLASTIC QUAD FLATPACK**





## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Mobile Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

TI E2E Community Home Page

[e2e.ti.com](http://e2e.ti.com)

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2012, Texas Instruments Incorporated