- **IEEE Standard for a 1394-1995 Compliant**
- **IEEE Standard 1212-1991 Compliant**
- **Compliant with IEEE Standard Control and Status Register Architecture for Microcomputer Buses.**
- **PCI Local Bus Specification Rev. 2.1 Compliant**
- **Supports IEEE 1394 Transfer Rates of 100, 200, and 400 Mbits per second**
- **3.3-Volt Core Logic while maintaining 5-Volt tolerant Inputs**
- **Performs the function of 1394 Cycle Master**
- **Provides 4KBytes of Configurable FIFO RAM**
- **Provides 5 Scatter-Gather DMA Channels**
- **Provides Software Control of Interrupt Events**
- **Implements a 32-bit PCI Address-data Path**
- **Provides 4 General Purpose Input/Outputs**
- **Supports Plug and Play Specification**
- **Generates 32-bit CRC for transmission of 1394 Packets**
- **Performs 32-bit CRC Checking on reception of 1394 Packets**
- **Provides PCI Slave Function for Read/Write Access of Internal Registers**
- **Supports Distributed DMA Transfers between 1394 and Local Bus RAM, ROM, AUX, or Zoomed Video**
- **Provides PCI Bus Master Function for supporting DMA Operations**
- **Advanced Submicron, Low-Power CMOS Technology**

1.0 Description

The TSB12LV21B (PCILynx-2) provides a high-performance IEEE 1394-1995 interface with the capability to transfer data between the 1394 PHY-link interface, the PCI bus interface, and external devices connected to the local bus interface. The 1394 PHY-link interface provides the connection to the 1394 physical layer device and is supported by the on-board link layer controller (LLC). The LLC provides the control for transmitting and receiving 1394 packet data between the FIFO and PHY-link interface at rates of 100 Mbit/s, 200 Mbit/s, and 400 Mbit/s. The link layer also provides the capability to receive status from the physical layer device and to access the physical layer control and status registers by the application software. The PCILynx-2 complies with

- PCI Local Bus Specification, Revision 2.1
- IEEE Standard for a 1394-1995 High Performance Serial Bus
- IEEE Standard 1212-1991,
- IEEE Standard Control and Status Register (CSR) Architecture for Microcomputer Buses.

An internal 4K-byte memory is provided that can be configured as multiple variable-size FIFOs and eliminates the need for external FIFOs. Separate FIFOs are user configurable to support 1394 receive, asynchronous transmit, and isochronous transmit transfer operations.

The PCI interface supports 32-bit burst transfers up to 33 MHz and is capable of operating both as a master and as a target device. Configuration registers can be loaded from an external serial EEPROM, allowing board and system designers to assign their own unique identification codes. An autoboot mode allows data-moving systems (such as docking stations) to be designed to operate on the PCI bus without the need for a host CPU.

The DMA controller uses packet control list (PCL) data structures to control the transfer of data and allow the DMA to operate without host CPU intervention. These PCLs can reside in PCI memory or in memory that is connected to a local bus port. The PCLs implement an instruction set that allows linking, conditional branching, 1394 data transfer control, auxiliary support commands, and status reporting. Five DMA channels are provided to accommodate programmable data types. PCLs can be chained together to form a channel control program

DRAFT ONLY

PRODUCT PREVIEW

that can be developed to support each DMA channel. Data can be stored in either big endian or little endian format, eliminating the need for the host CPU to perform byte swapping. Data can be transferred to either 4-byte aligned locations to provide the highest performance, or to nonaligned locations to provide the best memory use.

The RAM, ROM, AUX, ZV, and general purpose I/O (GPIO) ports collectively make up the local bus interface. These ports are mapped into the PCI address and can be accessed either through the PCI bus or through internal DMA transactions. Internal transactions do not appear on the external PCI bus, thereby conserving PCI bandwidth. DMA packet control lists or other data may be stored in external RAM or ROM attached to the local bus interface. This further reduces PCI bus use and generally improves performance. The ZV local bus port is designed to transfer data from 1394 video devices to an external device connected to the PCILynx ZV port. This interface provides a method of receiving 1394 digital camera packets directly from a ZV-compliant device attached to the local bus interface.

Built-in test registers, a dedicated test output terminal, and four GPIO terminals allow observation of internal states and provides a convenient software debug capability. Programmable interrupts are available to inform driver software of important events such as 1394 bus resets and DMA-to-PCL transfer completion.

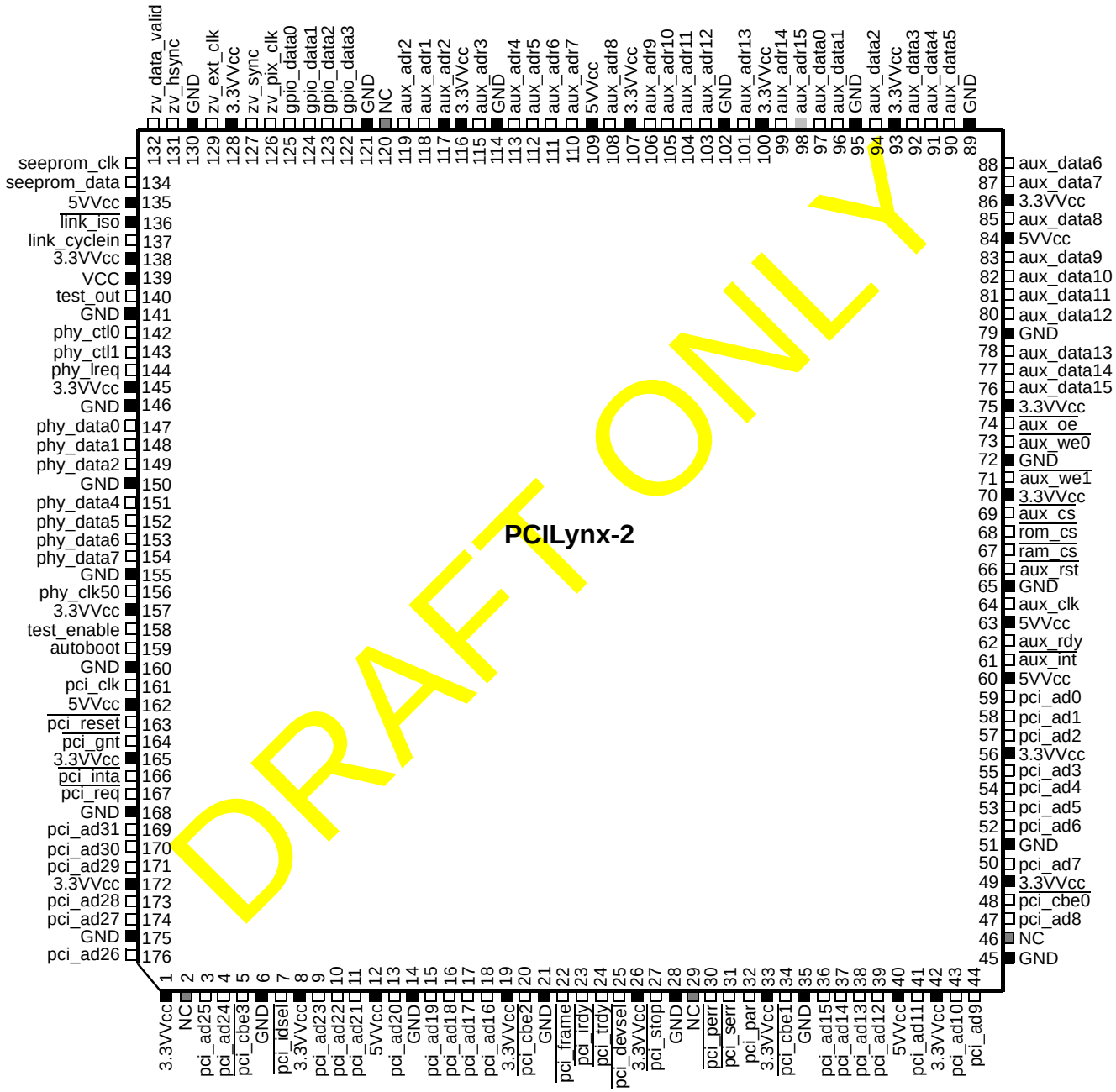
The 3.3-V internal operation provides reduced power consumption while maintaining compatibility with 5-V signaling environments. The PCI interface is compatible with both 3-V and 5-V PCI systems.

PRODUCT PREVIEW

DRAFT ONLY

2.0 Terminal Assignment

Figure 1 shows the PCILynx-2 device pinout/terminal assignments.



DRAFT ONLY

PRODUCT PREVIEW

Figure 1. PCILynx-2 Terminal Assignment/Pinout

PRODUCT PREVIEW

2.1 Pin Description Table

This sections identifies and classifies the functionality of each pin on the PCILynx-2.

Table 1. Signals Sorted by Pin Number

PIN NO.	SIGNAL NAME	PIN NO.	SIGNAL NAME	PIN NO.	SIGNAL NAME	PIN NO.	SIGNAL NAME
1	3.3VV_{CC}	45	GND	89	GND	133	seeprom_clk
2	NC	46	NC	90	aux_data5	134	seeprom_data
3	pci_ad25	47	pci_ad8	91	aux_data4	135	5VV_{CC}
4	pci_ad24	48	$\overline{\text{pci\_cbe0}}$	92	aux_data3	136	$\overline{\text{link\_iso}}$
5	$\overline{\text{pci\_cbe3}}$	49	3.3VV_{CC}	93	3.3VV_{CC}	137	link_cyclein
6	GND	50	pci_ad7	94	aux_data2	138	3.3VV_{CC}
7	pci_idsel	51	GND	95	GND	139	link_cylceout
8	3.3VV_{CC}	52	pci_ad6	96	aux_data1	140	test_out
9	pci_ad23	53	pci_ad5	97	aux_data0	141	GND
10	pci_ad22	54	pci_ad4	98	aux_adr15	142	phy_ctl0
11	pci_ad21	55	pci_ad3	99	aux_adr14	143	phy_ctl1
12	5.0VV_{CC}	56	3.3VV_{CC}	100	3.3VV_{CC}	144	phy_lreq
13	pci_ad20	57	pci_ad2	101	aux_adr13	145	3.3VV_{CC}
14	GND	58	pci_ad1	102	GND	146	phy_data0
15	pci_ad19	59	pci_ad0	103	aux_adr12	147	phy_data1
16	pci_ad18	60	5.0VV_{CC}	104	aux_adr11	148	phy_data2
17	pci_ad17	61	$\overline{\text{aux\_int}}$	105	aux_adr10	149	phy_data3
18	pci_ad16	62	$\overline{\text{aux\_rdy}}$	106	aux_adr9	150	GND
19	3.3VV_{CC}	63	5.0VV_{CC}	107	3.3VV_{CC}	151	phy_data4
20	$\overline{\text{pci\_cbe2}}$	64	aux_clk	108	aux_adr8	152	phy_data5
21	GND	65	GND	109	5.0VV_{CC}	153	phy_data6
22	$\overline{\text{pci\_frame}}$	66	$\overline{\text{aux\_rst}}$	110	aux_adr7	154	phy_data7
23	$\overline{\text{pci\_irdy}}$	67	$\overline{\text{ram\_cs}}$	111	aux_adr6	155	GND
24	$\overline{\text{pci\_trdy}}$	68	$\overline{\text{rom\_cs}}$	112	aux_adr5	156	phy_clk50
25	$\overline{\text{pci\_devsel}}$	69	$\overline{\text{aux\_cs}}$	113	aux_adr4	157	3.3VV_{CC}
26	3.3VV_{CC}	70	3.3VV_{CC}	114	GND	158	test_out
27	$\overline{\text{pci\_stop}}$	71	$\overline{\text{aux\_we1}}$	115	aux_adr3	159	auto_boot
28	GND	72	GND	116	3.3VV_{CC}	160	GND
29	NC	73	$\overline{\text{aux\_we0}}$	117	aux_adr2	161	pci_clk
30	$\overline{\text{pci\_perr}}$	74	$\overline{\text{aux\_oe}}$	118	aux_adr1	162	5.0VV_{CC}
31	$\overline{\text{pci\_serr}}$	75	3.3VV_{CC}	119	aux_adr0	163	$\overline{\text{pci\_reset}}$
32	pci_par	76	aux_data15	120	NC	164	$\overline{\text{pci\_gnt}}$
33	3.3VV_{CC}	77	aux_data14	121	GND	165	3.3VV_{CC}
34	$\overline{\text{pci\_cbe1}}$	78	aux_data13	122	gpio_data3	166	$\overline{\text{pci\_inta}}$
35	GND	79	GND	123	gpio_data2	167	$\overline{\text{pci\_req}}$
36	pci_ad15	80	aux_data12	124	gpio_data1	168	GND
37	pci_ad14	81	aux_data11	125	gpio_data0	169	pci_ad31
38	pci_ad13	82	aux_data10	126	zp_pix_clk	170	pci_ad30
39	pci_ad12	83	aux_data9	127	zv_sync	171	pci_ad29

DRAFT ONLY

40	$5.0VV_{CC}$	84	$5.0VV_{CC}$	128	$3.3VV_{CC}$	172	$3.3VV_{CC}$
41	pci_ad11	85	aux_data8	129	zv_ext_clk	173	pci_ad28
42	$3.3VV_{CC}$	86	$3.3VV_{CC}$	130	GND	174	pci_ad27
43	pci_ad10	87	aux_data7	131	zv_hsync	175	GND
44	pci_ad9	88	aux_data6	132	zv_data_valid	176	pci_ad26

DRAFT ONLY

PRODUCT PREVIEW

3.0 Terminal Functions

Power supply terminals

TERMINAL NAME	TERMINAL NO	TYPE	FUNCTION
GND	6, 14, 21, 28, 35, 45, 51, 65, 72, 79, 89, 95, 102, 114, 121, 130, 141, 150, 155, 160, 168, 175	I	Device ground terminals
3.3V VCC	1, 8, 19, 26, 33, 42, 49, 56, 70, 75, 86, 93, 100, 107, 116, 128, 138, 145, 157, 165, 172	I	3.3 V power supply terminal for core logic
5.0V VCC	12, 40, 60, 63, 84, 109, 162	I	5.0 Volt power rail for 5-V tolerant Input buffers

PCI system terminals

TERMINAL NAME	TERMINAL NO	TYPE	FUNCTION
pci_clk	161	I	System PCI bus clock. This signal ranges from 0-33 MHz and provides timing for all transactions on the PCI bus. All PCI signals are sampled at the rising edge of PCLK.
$\overline{\text{pci\_reset}}$	163	I	PCI reset. When the PCI bus reset is asserted the $\overline{\text{pci\_reset}}$ signal causes the PCILynx-2 to 3-state all output buffers and reset all internal registers. When $\overline{\text{pci\_reset}}$ is asserted, the device is completely nonfunctional. After $\overline{\text{pci\_reset}}$ is deasserted, the PCILynx-2 is in its default state.
$\overline{\text{pci\_inta}}$	166	OD	PCI system interrupt A. This is an open drain signal.

PRODUCT PREVIEW
DRAFT ONLY

PCI address and data terminals

NAME	TERMINAL NO	TYPE	FUNCTION
pci_ad31	169	I/O	Multiplexed PCI address and data signals. During the address phase of a primary bus PCI cycle, AD31:0 contain a 32-bit address or other destination information. During the data phase AD31:0 contain data
pci_ad30	170		
pci_ad29	171		
pci_ad28	173		
pci_ad27	174		
pci_ad26	176		
pci_ad25	3		
pci_ad24	4		
pci_ad23	9		
pci_ad22	10		
pci_ad21	11		
pci_ad20	13		
pci_ad19	15		
pci_ad18	16		
pci_ad17	17		
pci_ad16	18		
pci_ad15	36		
pci_ad14	37		
pci_ad13	38		
pci_ad12	39		
pci_ad11	41		
pci_ad10	43		
pci_ad9	44		
pci_ad8	47		
pci_ad7	50		
pci_ad6	52		
pci_ad5	53		
pci_ad4	54		
pci_ad3	55		
pci_ad2	57		
pci_ad1	58		
pci_ad0	59		
$\overline{\text{pci\_cbe3}}$	5	I/O	PCI Command/Byte enables.
$\overline{\text{pci\_cbe2}}$	20		
$\overline{\text{pci\_cbe1}}$	34		
$\overline{\text{pci\_cbe0}}$	48		
pci_par	32	I/O	PCI bus parity. In all PCI bus read and write cycles the PCILynx-2 calculates even parity across the AD31:0 and C/$\overline{\text{BE3:0}}$ signals. As an initiator during PCI cycles, the PCILynx-2 outputs this parity indicator with a one pci_clk delay. As a target during PCI cycles, the calculated parity is compared to the initiator's parity indicator. A miscompare can result in the assertion of a parity error ($\overline{\text{PERR}}$).

PCI interface control

NAME	TERMINAL NO	TYPE	FUNCTION
$\overline{\text{pci\_devsel}}$	24	I/O	PCI device select. The PCILynx-2 asserts this signal to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCILynx-2 monitors this signal until a target responds. If no target responds before time-out occurs, then the PCILynx-2 will terminate the cycle with an initiator abort.
$\overline{\text{pci\_frame}}$	22	I/O	PCI cycle frame. This signal is driven by the initiator of a bus cycle. $\overline{\text{FRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{FRAME}}$ is deasserted the PCI bus transaction is in the final data phase.

DRAFT ONLY
PRODUCT PREVIEW

pci_gnt	164	I	PCI bus grant. This signal is driven by the PCI bus arbiter to grant the PCILynx-2 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	7	I	Initialization device select. IDSEL selects the PCILynx-2 during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus.
pci_irdy		I/O	PCI initiator ready. 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 IRDY and TRDY are asserted. Until IRDY and TRDY are both sampled asserted, wait states are inserted.
pci_perr		I/O	PCI parity error indicator. This signal is driven by a PCI device to indicate that calculated parity does not match PAR, when PERR is enabled through bit 6 of the command register.
pci_req	167	O	PCI bus request. Asserted by the PCILynx-2 to request access to the PCI bus as an initiator.
pci_serr	31	OD	PCI system error. Output that is pulsed from the PCILynx-2, when enabled through the command register, indicating a system error has occurred. The PCILynx-2 needs not be the target of the PCI cycle in order to assert this signal. When SERR is enabled in the control register, this signal will also pulse indicating that an address parity error has occurred on a CardBus interface.
pci_stop	27	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.
pci_trdy	24	I/O	PCI target ready. TRDY indicates the primary 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 IRDY and TRDY are asserted. Until both IRDY and TRDY are asserted, wait states are inserted.

IEEE 1394 PHY/LINK interface terminals

NAME	TERMINAL NO	TYPE	FUNCTION
phy_ctl1 phy_ctl0	143 142	I/O	PHY-link bi-directional control lines.
phy_data7 phy_data6 phy_data5 phy_data4 phy_data3 phy_data2 phy_data1 phy_data0	154 153 152 151 149 148 147 146	I/O	PHY-link bi-directional data lines.
phy_clk50	156	I	50 MHz System clock from PHY chip.
phy_lreq	144	O	PHY-link request signal generated by the PCILynx-2 controller.

Auxiliary/Zoom Video Port terminals

NAME	TERMINAL NO	TYPE	FUNCTION
aux_clk	64	O	Auxiliary port clock out. This signal is output at the frequency of the PCI clock.
aux_rst	66	O	Auxiliary port reset out.
aux_int	61	I	Auxiliary port interrupt input.

PRODUCT PREVIEW
DRAFT ONLY

aux_adr15 aux_adr14 aux_adr13 aux_adr12 aux_adr11 aux_adr10 aux_adr9 aux_adr8 aux_adr7 aux_adr6 aux_adr5 aux_adr4 aux_adr3 aux_adr2 aux_adr1 aux_adr0	98 99 101 103 104 105 106 108 110 111 112 113 115 117 118 119	O	Auxiliary port address lines output to external logic.
aux_data15 aux_data14 aux_data13 aux_data12 aux_data11 aux_data10 aux_data9 aux_data8 aux_data7 aux_data6 aux_data5 aux_data4 aux_data3 aux_data2 aux_data1 aux_data0	76 77 78 80 81 82 83 85 87 88 90 91 92 94 96 97	I/O	Auxiliary port bi-directional data bus to external logic.
$\overline{\text{aux\_cs}}$	69	O	Auxiliary port chip select to external logic.
$\overline{\text{aux\_oe}}$	74	O	Auxiliary port output enable to enable external logic data on to the auxiliary data bus.
$\overline{\text{aux\_rdy}}$	62	I	Auxiliary port ready indication from external logic.
$\overline{\text{aux\_we1}}$ $\overline{\text{aux\_we0}}$	71 73	O	Auxiliary port write strobes to external logic.
ram_cs	67	O	External RAM chip select.
rom_cs	68	O	External ROM chip select.
zv_data_valid	132	O	Zoom video port data valid signal.
zv_ext_clk	129	I	Zoom video port external clock input.
zv_hsync	131	O	Zoom video port horizontal sync signal.
zv_pix_clk	126	I/O	Zoom video port pixel clock for zoomed video data.
zv_vsync	127	O	Zoom video port vertical sync signal.

DRAFT ONLY
PRODUCT PREVIEW

Miscellaneous

NAME	TERMINAL NO	TYPE	FUNCTION
autoboot	159	I	Autoboot. Selects autoboot mode. When this terminal is tied high, autoboot mode is selected.
gpio_data3 gpio_data2 gpio_data3 gpio_data2	122 123 124 125	I/O	Auxiliary port general purpose programmable I/O signals.
link_cyclein	137	I	Optional 8 kHz clock for use as the cycle clock.
link_cycleout	140	O	Cycle timer 8 kHz clock output.
$\overline{\text{link\_iso}}$	136	I	PHY-link isolation barrier mode. When this pin is low, internal bus holders are enabled and the PCILynx-2 PHY-link interface is isolated from the PHY.
seeprom_clk	133	I/O	External serial EEPROM data clock.
seeprom_data	134	I/O	External serial EEPROM read/write data line.
test_enable	158	I	Enables test_out for AND tree testing. This pin can be tied to GND if AND tree testing is not used.
test_out	140	O	Output for AND tree testing.

PRODUCT PREVIEW
DRAFT ONLY

4.0 System Block Diagram

The following figure illustrates a typical system implementation of the PCILynx-2.

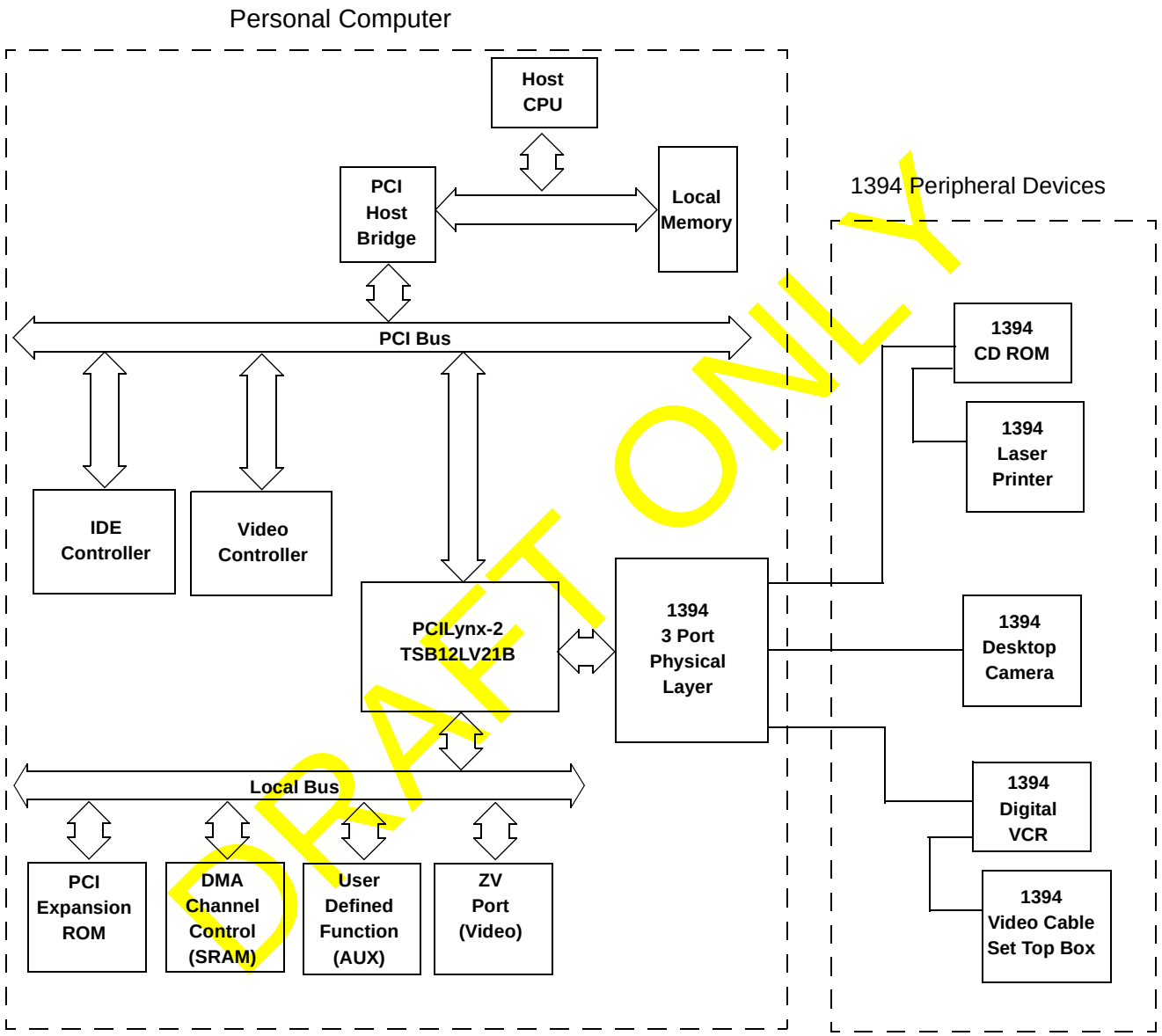


Figure 2. System Block Diagram

DRAFT ONLY
PRODUCT PREVIEW

5.0 Application Information

The following sections provides register-level configuration information for the PCILynx-2 controller.

5.1 PCI Configuration Registers

The PCILynx-2 configuration header is compliant with the PCI Specification as a type 0 header. Table 9 illustrates the PCI configuration header which includes both the predefined portion of the configuration space and the user definable registers. The registers that are labeled 'Reserved' are read only and return zero when read.

Table 2. PCI Configuration 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
Memory Base Address 0 - Internal PCILynx Registers				10h
Memory Base Address 1 - External SRAM on Local Bus				14h
Memory Base Address 2 - AUX Port on Local Bus				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
Reserved				28h
Subsystem ID		Subsystem Vendor ID		2Ch
PCI Expansion ROM Base Address				30h
Reserved				34h
Reserved				38h
Maximum Latency	Minimum Grant	Interrupt Pin	Interrupt Line	3Ch
Miscellaneous Control				40h
Serial EEPROM Control				44h
PCI Interrupt Status				48h
PCI Interrupt Enable				4Ch
PCI Test				50h
Reserved				54h
Subsystem ID (R/W)		Subsystem Vendor ID (R/W)		58h
Reserved				5C - ACh
Local Bus Control				B0h
Local Bus Address				B4h
PCI_GIO[1:0] Control A				B8h
PCI_GIO[1:0] Control B				BCh
PCI_GPIO_DATA_0000 Read Only Port				C0h
PCI_GPIO_DATA_0001 Read/Write Port				C4h
PCI_GPIO_DATA_0010 Read/Write Port				C8h
PCI_GPIO_DATA_0011 Read/Write Port				CCh
PCI_GPIO_DATA_0100 Read/Write Port				D0h

PRODUCT PREVIEW
DRAFT ONLY

PCI_GPIO_DATA_0101 Read/Write Port	D4h
PCI_GPIO_DATA_0110 Read/Write Port	D8h
PCI_GPIO_DATA_0111 Read/Write Port	DCh
PCI_GPIO_DATA_1000 Read/Write Port	E0h
PCI_GPIO_DATA_1001 Read/Write Port	E4h
PCI_GPIO_DATA_1010 Read/Write Port	E8h
PCI_GPIO_DATA_1011 Read/Write Port	ECh
PCI_GPIO_DATA_1100 Read/Write Port	F0h
PCI_GPIO_DATA_1101 Read/Write Port	F4h
PCI_GPIO_DATA_1110 Read/Write Port	F8h
PCI_GPIO_DATA_1111 Read/Write Port	FCh

DRAFT ONLY

PRODUCT PREVIEW

PRODUCT PREVIEW

5.2 PCI Subsystem Identification Register Operational Changes

The following changes were implemented to meet Microsoft PC97/PC98 requirements. This register is used for system and option card identification purposes. 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 present, then this register may be updated by writing to the PCI register at 58h. The contents of the register at 58h is reflected in PCI subsystem identification register at 2Ch.

5.2.1 PCI Register 2Ch (Subsystem Device ID/Subsystem Vendor ID)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Subsystem Device ID/Subsystem Vendor ID Register															
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	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Subsystem Device ID/Subsystem Vendor ID Register															
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: Subsystem Device ID/Subsystem Vendor ID Register
Type: Read Only
Offset: 2Ch
Default: 8000 104Ch
Description: This register is used for system and option card identification purposes. 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 present, then this register may be updated by writing to the PCI register at 58h. The contents of the register at 58h is reflected in PCI subsystem identification register at 2Ch.

This register provides application software access to the Vendor ID and Device ID numbers that are assigned to the PCILynx-2.

Table 3. Subsystem Device and Vendor ID Register

BIT	FIELD NAME	TYPE	FUNCTION
31-16	SUBDEV_ID	RU	Subsystem Device ID. This field indicates the subsystem device ID.
15-0	SUBVEN_ID	RU	Subsystem Vendor ID. This field indicates the subsystem vendor ID.

DRAFT ONLY

5.2.2 PCI Register 58h (Subsystem Device ID/Subsystem Vendor ID Update)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Subsystem Device ID/Subsystem Vendor ID Update Register															
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	Subsystem Device ID/Subsystem Vendor ID Update Register															
Type	R/W	R/W	R/W	R/W	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: Subsystem Device ID/Subsystem Vendor ID Update Register
Type: Read Only, Read/Write
Offset: 58h
Default: 0000 0000h
Description: This register is used for system and option card identification purposes. 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 present, then this register may be updated by writing to the PCI register at 58h. The contents of the register at 58h is reflected in PCI subsystem identification register at 2Ch

6.0 Serial EEPROM Implementation

This section describes the use of the serial EEPROM interface and describes the layout of the serial EEPROM.

6.1 Serial EEPROM Interface

The serial EEPROM interface provides communication between the PCILynx and an attached serial EEPROM. The serial EEPROM resides on an industry standard 2 wire serial bus at slave address 0.

At power-up, the serial EEPROM interface initializes a small number of locations in the PCI configuration registers with data from the EEPROM. While the serial EEPROM state machine is accessing the EEPROM, any incoming PCI slave access is terminated with retry status. This ensures that the system software will always read the values loaded from the serial EEPROM. A software reset will also initiate a reload of the PCI configuration register values from the serial EEPROM.

A map of the data stored in the EEPROM is given in the next section. This information is read and written by the host processor emulating the 2 wire serial bus protocol through the Serial EEPROM Control register (PCI configuration offset 44h). The 2 wire serial bus is manipulated from the host processor by setting the serial EEPROM output enable bit (EEPENA, bit 5) to a ‘1’ and then accessing the data (EEPDATA, bit 4) and clock (EEPCLK, bit 6) bits to emulate the 2 wire serial bus protocol.

The 5µs timer bit in the control register provides a timing reference for the 2 wire serial bus protocol events if a more accurate source is not available. Since this timer is based on the PCI clock, it may be longer than desired, depending on the frequency of the PCI clock. Writing a zero to the timer bit starts the timer. The timer bit will be set to ’1’ after the timer has expired. Host software may poll this bit to determine when the required time has elapsed for implementing the 2 wire serial bus protocol.

DRAFT ONLY
PRODUCT PREVIEW

6.2 Serial EEPROM Data Format

The PCILynx-2 loads certain internal configuration registers from the serial EEPROM immediately after a power reset. The first 8 bytes of the serial EEPROM address space are reserved for use by the PCILynx. These bytes are loaded into internal registers by the PCILynx as described in Table 4.

Table 4. Serial EEPROM Address Map

BYTE ADDRESS	DEFAULT VALUE	BYTE DESCRIPTION (PCI configuration register offset)
00	0x02	PCI max_lat (3Fh)
01	0x01	PCI min_gnt (3Eh)
02	0xE0	Local Bus Control Register - ROM Control (B0h)
03	0x00	PCI Subsystem Vendor ID (LSB) (2Ch)
04	0x00	PCI Subsystem Vendor ID (MSB) (2Dh)
05	0x00	PCI Subsystem ID (LSB) (2Eh)
06	0x00	PCI Subsystem ID (MSB) (2Fh)
07	-	Checksum of bytes 0-6
08	-	Reserved for future hardware use
-	-	-
0F	-	Reserved for future hardware use
10		User defined
-	-	-
FF	-	User defined

PRODUCT PREVIEW
DRAFT ONLY

The serial EEPROM should also contain, at a minimum, the Bus_Info_Block as described in the Control and Status Registers map in Table 5.

Table 5. CSR Architecture

	Offset	Value																
	00																	
	04																	
	08																	
	0C																	
bus_info_block	10		info_length				crc_length				rom_crc_length							
	14		0x31				0x33				0x39				0x34			
	18		irmc/ cmc	isc/ cmc	reserved		cyc_clk_acc				max_rec	reserved						
	1C		node_vendor_id												chip_id_hi			
	20		chip_id-lo															
root directory	24		dir_len								dir_crc							
	28		0x03				module_vendor_id (required)											
	2C	81000008	0x81				module_vendor_id textual descriptor offset (optional)											
	30		0x0D				node_capabilities (required)											
	34	8D00000E	0x8D				node_unique_id offset (required)											
	38		0xC7				module_dependent_info offset (optional)											
	3C		0x04				module_hardware_version offset (optional)											
	40		0x81				module_hardware_version textual descriptor offset (optional)											
	44		0x09				node_hardware_version offset (optional)											
	48		0x81				node_hardware_version textual descriptor offset (optional)											
	4C		leaf_len								leaf_crc							
	50																	
	54																	
	58																	
	5C																	
	60																	
	64																	
	68																	
	6C		leaf_len								leaf_crc							
	70		node_vendor_id												chip_id_hi			
	74		chip_id_lo															
	78																	
	7C																	
	80																	
	84																	
	88																	
	8C																	
	90																	

DRAFT ONLY
PRODUCT PREVIEW

Table 6 shows the registers commonly loaded via the Serial EEPROM.

Table 6. Registers Loaded from Serial EEPROM

	OFFSET	BYTE 0	BYTE 1	BYTE 2	BYTE 3
PCI Configuration Registers	00	PCI max_lat (PCI offset 3Fh) Default 0x02	PCI min_gnt (PCI offset 3Eh) Default 0x01	Local Bus Control (PCI offset B0h) Default 0xE0	SSVID LSB (PCI offset 2Ch) Default 0x00
	04	SSVID MSB (PCI offset 2Dh) Default 0x00	SSID LSB (PCI offset 2Eh) Default 0x00	SSID MSB (PCI offset 2Fh) Default 0x00	Checksum (ROM bytes 0-6)
	08	Reserved	Reserved	Reserved	Reserved
	0C	Reserved	Reserved	Reserved	Reserved
bus_info_block	10	info_length	crc_length	rom_crc_length	
	14	0x31	0x33	0x39	0x34
	18	irmc, cmc, isc, bmc	cyc_clk_acc	cyc_clk_acc	Reserved
	1C	node_vendor_id			chip_id_hi
	20	chip_id_lo			
	24				
	28				
	2C				
	30				
	34				
	38				
	3C				
	40				

Note 1: Refer to section 8.3.2.5 of IEEE Std 1394-1995

6.3 Internal Bus Holders

PCILynx-2 provides Internal Bus Holder support to phy_data0 to phy_data7, phy_ctl0, phy_ctl1, and phy_clk50.

6.4 Allow ISO packets (Tcode=A) in Async period

PCILynx-2 supports both transmission and reception of ISO packets (Tcode = A) during the Asynchronous period.

6.5 PCILynx-2 Local Bus

The PCILynx contains a 16-bit wide Local Bus that can be attached to several different types of devices to enhance its capabilities in an embedded system environment. This section describes the functionality of the Local Bus and how to implement it. The Local Bus is not recommended to be used in a PC environment.

PRODUCT PREVIEW
DRAFT ONLY

The Local Bus provides access to the following I/O ports: PCI Expansion ROM, RAM, AUX, and ZV (Zoomed Video) output. The PCI Expansion ROM, RAM, and AUX port all share the following characteristics: 64 K address space, 16-bit address bus, 8 or 16-bit read/write data bus, byte addressable/writeable, and programmable ready/wait-states. The ZV output port provides the following capabilities: horizontal and vertical sync outputs, data valid indicator, 8 or 16-bit interface, 8 bits of Y (luminance data), 8 bits of UV (chrominance data), and a programmable pixel clock output.

6.5.1 PCI Expansion ROM Interface

The PCI Expansion ROM provides the host system with the capability of reading configuration data or executable code from an attached ROM. This allows the system to boot from a 1394 device, even though the system may lack specific 1394 boot code at power-reset.

Additionally, this interface has been generalized to provide functionality beyond PCI Expansion ROM access. This interface will support PCI slave and internal DMA machine read/write access to devices such as EEPROM, FLASH, and other ROM/RAM-like devices.

ROM access is controlled by the standard PCI configuration PCI Expansion ROM base address register (PCI offset 30h) and is enabled by writing a '1' to bit 0 (ROMEN) of this register.

The ROM interface may be configured as either 8 or 16-bit wide data and can be configured to use a specified number of wait-states or an external ready signal. ROM options are configured at power-reset via the serial EEPROM. The description of the Local Bus Control Register in the PCILynx-2 Programmer's Guide **(** ?? **)** provides more information.

6.5.2 SRAM Interface

The Static RAM is accessed through PCI memory based at the address programmed in Memory Base Address Register 1 (PCI offset 14h). A few examples SRAM use are: DMA control structures, DMA data buffers, or shared memory with other functions such as a DSP.

The RAM interface may be configured for either 8 or 16-bit wide data and can be configured to use a specified number of wait-states or an external ready signal. This configuration is done through the Local Bus Control Register (PCI offset B0h). **(** ???**)** The description of the Local Bus Control Register in the PCILynx-2 Programmer's Guide **(** ?? **)** provides more information.

6.5.3 AUX Port Interface

The AUX port is a generic I/O port that can be accessed through PCI memory at the address programmed in Memory Base Address Register 2 (PCI offset 18h). This port may be used to implement a high speed data path to external dedicated resources such as compression/decompression logic, or video processor/frame buffers.

If the ZV port is enabled, the address range between F000h and FFFFh are mapped to the ZV port; otherwise, this space is available as part of the AUX port address space.

The AUX port interface may be configured for either 8 or 16-bit wide data and can be configured to use a specified number of wait-states or an external ready signal. This configuration is done through the Local Bus Control Register (PCI offset B0h).

6.5.4 ZV Port Interface

The ZV port is an output only port designed to transfer data from 1394 video devices to an external device. When correctly programmed, this interface provides a method to receive 1394 Digital Camera packets and transfer the payload data to an external ZV compliant device. The ZV port assumes quadlet data.

DRAFT ONLY
PRODUCT PREVIEW

The ZV port is accessed through the upper most 4 KByte (addresses F000h to FFFFh) of the AUX port address space if the ZV port is enabled. The ZV port is enabled when one of the six available clock sources is selected as the ZV pixel clock. If none of the six clock sources are selected, the ZV port is disabled and the AUX port claims the entire address space. When the ZV port is disabled, all ZV related outputs are 3-stated with the exception of the data bus which can still be used by the other Local Bus ports.

The ZV port is configured through the Local Bus Control Register (PCI offset B0h).

6.6 PHY-Link Interface

The PCILynx-2 communicates with the 1394 Serial Bus through a physical layer chip referred to as a PHY. The PCILynx uses the PHY-Link interface to do this. The PHY-Link interface consists of the following 12 signals: phy_ctl[0:1], phy_data[0:7], phy_clk50, and phy_lreq. This section describes how the PCILynx uses these signals to communicate with the PHY.

The control signals, phy_ctl[0:1], and the data signals, phy_data[0:7], are bi-directional signals. The PHY controls these signals and only grants control of them to the link after it requests control through the phy_lreq signal. Whenever control of the PHY-Link interface is transfered between the PHY and the Link, the side giving up control always drives the control and data pins to '0' for one clock before tristating its output buffers (an additional clock with '0' on the control and data signals is necessary for the link when it is transferring control to the PHY without a Hold request). This is necessary to ensure that the differentiator circuit can operate properly.

Data is carried between the PHY and the Link on the data bus, phy_data[0:7]. The width of the data bus depends on the transmission speed of the packet. PCILynx supports the following speeds: 100, 200, and 400 Mbits/sec. At 100 Mbits/sec, the data bus uses phy_data[0:1]. At 200 Mbits/sec, the data bus uses phy_data[0:3]. And at 400 Mbits/sec, the data bus uses phy_data[0:7]. When data bits are not being used to transmit data in the case of 100 or 200 Mbits/ sec transfers, the unused signals need to transmit '0'.

The control signals, phy_ctl[0:1], are used to indicate what kind of transaction is taking place on the PHY-Link interface. The values present on the phy_ctl[0:1] lines indicate different actions depending on which device, the PHY or the link, has control of the data bus.

The PCILynx-2 uses the phy_lreq signal to request control of the PHY-Link interface from the PHY. The request is in the form of a serial bit stream on the phy_lreq signal that includes a three bit code indicating the request type and any request type specific information. These requests on the phy_lreq signal can be 7 bits, 9 bits, or 17 bits depending on the request type.

The PHY-Link interface is clocked by the phy_clk50 signal from the PHY. All timing on the PHY-Link interface is relative to this clock. This clock is a 50 MHz clock signal that is generated by the PHY.

PRODUCT PREVIEW
DRAFT ONLY

absolute maximum ratings over operating temperature ranges (unless otherwise noted) †

Supply voltage range, V_{CC} ..... -0.5 V to 4.0 V
Supply voltage range, V_{CCP} ..... -0.5 V to 6 V
Supply voltage range, V_{CC5V} ..... -0.5 V to 6 V
Input voltage range for Universal PCI, V_I:PCI ..... -0.5 V to V_{CCP} + 0.5 V
Input voltage range for 5-V tolerant TTL/LVCMOS, V_I: ..... -0.5 V to V_{CC5V} + 0.5 V
Output voltage range for Universal PCI, V_O: ..... -0.5 V to V_{CCP} + 0.5 V
Output voltage range for 5-V tolerant TTL/LVCMOS, V_O : ..... -0.5 V to V_{CC5V} + 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

† 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 buffers, use V_I > V_{CC5V}. For Universal PCI, use V_I > V_{CCP}.
2. Applies to external output and bidirectional buffers. For 5-V tolerant buffers, use V_O > V_{CC5V}. For Universal PCI, use V_O > V_{CCP}.

DRAFT ONLY

PRODUCT PREVIEW

electrical characteristics over recommended operating conditions (unless otherwise noted)

		PINS		OPERATION	TEST CONDITIONS	MIN	MAX	UNIT
V_{OH}	High-level output voltage (see note 3)	PCI		3.3 V	I_{OH} = -0.5 mA	0.9 V_{CC}		V
				5 V	I_{OH} = -2 mA	2.4		
		TTL/LVCMOS†			I_{OH} = -18 mA	2.4		
		TTL/LVCMOS‡			I_{OH} = -14mA	2.4		
V_{OL}	Low-level output voltage	PCI		3.3 V	I_{OL} = 1.5 mA		0.1 V_{CC}	V
				5 V	I_{OL} = 6 mA		0.5	
		TTL/LVCMOS†			I_{OL} = 18 mA		0.5	
		TTL/LVCMOS‡			I_{OL} = 14 mA		0.5	
I_{IL}	Low-level input current	Input pins	Bushold		V_I = 0.8V		20	µA
			Others		V_I = GND		-1	
		I/O pins	Bushold		V_I = 0.8V		400	
			Others		V_I = GND		-20	
I_{IH}	High-level input current	Input pins	Bushold		V_I = 2.0V		-20	µA
			Others		V_I = 5.5V		20	
		I/O pins	Bushold		V_I = 2.0V		-20	
			Others		V_I = 5.5V		20	

† All PHY-link pins, aux_clk(64), $\overline{\text{aux\_we1}}$(71), and $\overline{\text{aux\_we0}}$(73).
‡ All other TTL/LVCMOS pins

NOTES:
3. V_{OH} is not tested on $\overline{\text{pci\_serr}}$(31) or $\overline{\text{pci\_inta}}$(166) due to open-drain output.

PRODUCT PREVIEW
DRAFT ONLY

recommended operating conditions (see Note 4)

			OPERATION	MIN	NOM	MAX	UNIT
V_{CC}	Core voltage	Commercial	3.3 V	3	3.3	3.6	V
V_{CCP}	I/O voltage	Commercial	5 V	3	5	5.5	V
V_{CC5V}	I/O voltage	Commercial	5 V	3	5	5.5	V
V_{IH}†	High-level Input voltage			2			V
V_{IL}†	Low-level Input voltage					0.8	V
V_I	Input voltage	Universal PCI		0		V_{CCP}	V
		5-V tolerant		0		V_{CC5V}	
V_O¶	Output voltage			0		V_{CC}	V
t_t	Input transition times (t_r and t_f)			0		6	ns
T_A	Operating ambient temperature range			0	25	70	°C
T_J#	Virtual junction temperature			0	25	115	°C

NOTES:
4. Unused or floating pins (input or I/O) must be held high or low.
† Applies for external input and bidirectional buffers without hysteresis.
¶ Applies for external output buffers.
# These junction temperatures reflect simulation conditions. Customer is responsible for verifying junction temperature.

DRAFT ONLY

PRODUCT PREVIEW