ERRATA
TO THE TSB12LV21 (PCILynx) DATA SHEET
(TEXAS INSTRUMENTS LITERATURE NO. SLLS253, JULY 1996)

This document is an errata to the TSB12LV21 data sheet (Literature No. SLLS244).

Errata 1

Electrical isolation as described in Appendix J of IEEE 1394-1995 is not supported by the PCILynx.

TI has an improved isolation technique that is the recommended isolation solution.

Errata 2

The 1394 transmit state machine returns a NO_ACK status to the DMA when an unformatted broadcast packet type is transmitted from the asynchronous transmit FIFO (ATF). This status causes the DMA channel to branch to the PCL address specified at offset 0×4 of the current PCL, instead of to the PCL address specified at offset 0×0.

Suggested Software Solution

Make the next PCL address at PCL offset 0×4 the same as the next address specified at PCL offset 0×0.

Errata 3

The PCI interface has data transfer problems when:

- The Enable Posted Writes is cleared to 0 and Enable Slave Burst is set to 1.
- The Enable Slave Burst bit is cleared to 0, a burst slave cycle is received by the PCILynx device, and the DMA is executing an internal master cycle (i.e. to internal registers or local bus).

Suggested Solution

Before enabling the DMA, the Enable Posted Write (bit 7) and Enable Slave Burst (bit 6) in the PCI Miscellaneous Control register (register offset 0×40) should be set to 1.

Errata 4

The input polarity control bits GPIO_RDY_POL3 and GPIO_RDY_POL2 are not functional. The input polarity for the GPIO ready inputs is always active high.

Suggested Solution

- When using the GPIO ready bits for DMA wait control, design the external logic to provide an active-high ready signal.
- When using the GPIO ready bits for a branch condition, reorganize the PCLs to use only a active-high ready signal.

Errata 5

When an isochronous transmit FIFO (ITF) underrun occurs, the ITF flush control logic does not decrement the isochronous packet counter after it has flushed the partial packet to the end-of-packet control token. As a result, the isochronous packet count does not indicate the correct number of packets remaining in the ITF. When the isochronous DMA channel tries loading another packet in the ITF, the incorrect packet count causes the link to start transmitting the packet before enough of the packet has accumulated in the ITF. This, in turn, results in another ITF underrun. As a result the DMA channel then gets stuck in an underrun flush condition every time it tries transmitting a packet. The DMA continues to execute PCLs and push data onto the ITF, even though no data is being transmitted.

Suggested Software Solution

Setup the ITF underrun counter register at offset 0×F24 to 0×FE, so that an ITF underrun causes an ITF_UNDER_FLOW interrupt to occur when the underrun counter increments to 0×FF.

Software can then recover from the underrun hang by executing the following numbered sequence. When an ITF_UNDER_FLOW interrupt occurs:

1. Disable the isochronous DMA channel(s).
2. Reset the TX_ISO_EN in the link control register (offset 0×F04).
3. Flush the ITF by setting the ITF_FLUSH bit in the FIFO control register (offset 0×A10).
4. Wait for the ITF_FLUSH bit in the FIFO control register to return to 0.
5. Re-enable the TX_ISO_EN bit in the link control register (offset 0×F04).
6. Reload the ITF underrun counter register at offset 0×F24 to 0×FE.
7. Service the interrupt (clear the PCI and link interrupt status registers).
8. Rebuild the isochronous DMA PCLs.
9. Restart the isochronous DMA channels.

Errata 6

Asynchronous or isochronous transmit FIFOs (ITFs) can hang with the link receiving a constant FIFO empty status, and the DMA receiving normal full/empty indications. Typically, the DMA fills the FIFO up to its high-water condition, then be waiting for additional space in the FIFO while no data is being transmitted.

This condition can happen whenever the link side of a transmit FIFO goes empty, then on the next clock the FIFO goes immediately not empty.

Suggested Software Solution

ALTERNATIVE 1

DMA Isochronous Transmit Channels

1. Program the PCL with an auxiliary command sequence that causes the DMA channel to wait for the ITF to go empty before it starts loading the the next packet into the ITF.
2. Restrict the packet size and ITF_TRSHLD such that $(4 \times 2) + \text{packet size} \leq \text{ITF\_TRSHLD} \times 4$.
3. Only allow one isochronous DMA channel and only one isochronous packet/isochronous period.

DMA Asynchronous Transmit Channels

1. Program the PCL to wait for asynchronous transmit status from the link before starting the transfer of the next packet into the asynchronous transmit FIFO (ATF).

   Setting the WAIT FOR STATUS bit in the PCLs at PCL offset 0×18 bit 17 to a 1 enables the wait for status behavior.
2. Restrict the packet size and ATF_TRSHLD such that $(4 \times 2) + \text{packet size} \leq \text{ATF\_TRSHLD} \times 4$.

Alternative 2

Based on a deadman timer, identify the hung channels, then:

1. Disable the DMA channel(s).

2. Reset the TX_ASYNC_EN and/or TX_ISO_EN in the link control register (offset 0×F04).
3. Flush the FIFO(s) by setting the ATF_FLUSH and/or ITF_FLUSH bit in the FIFO control register (offset 0×A10).
4. Wait for the ATF_FLUSH and/or ITF_FLUSH bit(s) to return to 0.
5. Re-enable the TX_ASYNC_EN and/or TX_ISO_EN bit(s) in the link control register (offset 0×F04).
6. Rebuild the DMA PCLs.
7. Restart the DMA channels.

Errata 7

The link interrupt status register does not capture an event when the corresponding interrupt enable bit is not set. Thus, unless the interrupt enable register is always set for the interrupt bits of interest, interrupts is missed.

Suggested Software Solution

Alternative 1 (Polling Mode)

If polling link interrupts:

1. Disable the P1394_INT interrupt in the PCI interrupt enable register (offset 0×04C) .
2. Enable the desired link interrupt events by setting the corresponding bits in the 1394 link interrupt enable register (offset 0×F18)
3. Poll the 1394 link interrupt status register for the desired interrupts
4. Reset the 1394 link interrupt status register by writing 1s to the bit position to be cleared.

This allows the link interrupt status register bits to set and inhibit the link interrupt events from generating an interrupt to the CPU.

Alternative 2 (Interrupt Service Mode)

Only enable the low frequency of occurrence link status bit and service interrupts normally, except never disable the selected bits in the 1394 link interrupt enable register.

Interrupts not enabled are not set in the 1394 link interrupt status register.

Errata 8

If the PCILynx is not the isochronous cycle master, a 1394 bus reset inhibits the reception of cycle start packets. This, in turn, prevents the transmission or reception of isochronous packets. Thus, transmission and receipt of isochronous packets does not continue after a bus reset.

The bus reset should only disable the link from transmitting and receiving asynchronous packets but should still allow the transmission and reception of cycle start and isochronous packets.

Suggested Software Solution

Set up the link layer to generate an interrupt whenever a 1394 bus reset occurs. When a bus reset interrupt occurs, the interrupt service routine should first disable all of the receive channel comparators that were programmed to accept asynchronous packets for the previous 1394 bus configuration. Comparators that had been programmed for receiving isochronous packets should remain enabled. Next, turn on the asynchronous receive enable bit in the link layer control register. This allows cycle start packets to be received and isochronous packets to be received and transmitted.

After a bus reset, asynchronous channels always require being disabled, the asynchronous transmit FIFO (ATF) flushed, the bus configuration determined, the PCLs rebuilt, and the DMA restarted.

Errata 9

Self-ID packets are sometimes received on the wrong DMA channel number. This problem occurs when a higher priority channel comparator that has not been enabled to receive Self-ID packets matches the incoming Self-ID packet and overrides the lower priority channel comparator that has been enabled to receive Self-ID packets.

Suggested Software Solution

Program all address comparators that are assigned to receive asynchronous packets to match the destination ID field of the asynchronous packet to be received according to the procedure in the following steps.

1. In specifying the destination ID field of an asynchronous packet, only use 10-bit bus numbers that have the two most significant bits set to 00 or 11. The quadlets of a Self-ID packet are set to 10 or 01 in these bit positions.

   Bus Number Examples: 0011111111, 1111111111, 1100000001

2. In each address comparator that is assigned to receive asynchronous packets, program the address comparator to match on the bus numbers that you have selected.

The bus number part of the destination ID should be programmed as shown in the following example:

CMP0_FIELD1[15:6] = bus number selected

CMP0_FIELD1_MASK[15:6] = 1111111111

Errata 10

The link receiver intermittently rejects isochronous packets if the packet transmission occurs very close to the time when the the subaction gap opens.

Suggested Software Solution

This is still being investigated

Errata 11

The serial EEPROM interface on the PCILynx reads the serial interface bits backwards into its data register. The data bit transferred first is placed in the least significant bit; according to the I^{2}C interface, the most significant data bit should be transferred first. This makes each byte appear to be loaded in bit-reversed order in the PCILynx control registers upon power up. This only affects the automatic, power up loading of certain PCILynx registers (serial EEPROM addresses 0–7). The software interface in not affected.

For more information see the I^{2}C-Bus Specification, p.5 in *I^{2}C Peripherals for Microcontrollers*, Signetics/Phillips Semiconductors.

Suggested Solution

Software should build the data values in bit-reversed order when writing data to the serial EEPROM for the hardware read locations. All other serial EEPROM addresses should be written with data built according to the I^{2}C-Bus Specification. The register load sequence will be corrected in the next revision of the PCILynx.

Errata 12

The PCILynx is unable to issue PHY register read/write access requests when the link asynchronous and isochronous transmit enable bits are both off (cleared to 0), and the cyclemaster enable bit is on (set to 1).

Suggested Solution

Ensure that the asynchronous or isochronous transmit enable bits in the link layer control register are set to 1 before issuing a PHY register read or write access request.

Errata 13

There exists a bug in the DMA which causes the command quadlet of a command/buffer address pair to be overwritten by the buffer address. This happens when the PCI bus performs a target disconnect on the first access of the command/buffer instead of bursting and a slave cycle accesses a PCILynx register or local bus resource.

Suggested Solution

Always have PCI bursting enabled.

Errata 14

The PCI slave interface does not meet the PCI 2.1 specification for maximum latency under certain conditions. When the DMA is accessing local bus resources (i.e., local RAM, ROM, AUX, or ZOOM PORT) or internal registers in the link layer or accessing the link side of the FIFO host, PCI cycles may be held longer than the required 16 PCI clocks before completing the cycle.

Suggested Solution

Do not use local bus resources (RAM, ROM, AUX, or ZOOM PORT) if your system has time-critical PCI functions sharing the PCI bus.

Errata 15

Several of the PCI-interface signals do not meet the PCI 2.1 specification for setup and hold time under worst case conditions.

Suggested Solution

Do not use the part at temperature or voltage extremes. A normal user environment should not experience malfunctions due to this specification violation.