Tsi148™ Device Errata

80A3020_ER001_08

October 14, 2009

6024 Silver Creek Valley Road San Jose, California 95138

Telephone: (408) 284-8200 • FAX: (408) 284-3572

Printed in U.S.A.

About this Document

This document describes device errata for the Tsi148. Since this information may change over time, please ensure you have the most recent version by contacting a member of the IDT technical support team.

Revision History

80A3020_ER001_08, October 2009

This version of the document was rebranded as IDT. It does not include any technical changes.

80A3020_ER001_07, March 2008

This version was been updated to reflect errata that are applicable for the production version of the device. It does not include any technical changes.

Part Numbers

Part Number	Frequency	Temperature	Package	Pin Count
Prototype				
Tsi148-133CL	133 MHz	Commercial	PBGA	456
Tsi148-133CLY	133 MHz	Commercial	PBGA	456
Tsi148-133IL	133 MHz	Industrial	PBGA	456
Tsi148-133ILY	133 MHz	Industrial	PBGA	456

Device Errata

Device Errata[a]	Applicability	
	Tsi148-133CLY	Tsi148-133ILY
VME Interface		
“[VME1] 2eSST Broadcast Erroneous DTACK* Assertion”	✓	✓
“[VME2] 2eSST Broadcast T4 Timing Parameter Specification Violation”	✓	✓
“[VME3] VMEbus Device Wants Bus Issue”	✓	✓
“[VME4] VMEbus Write During Wrap-around RMW Issue”	✓	✓
“[VME5] VMEbus RETRY* Specification Violation”	✓	✓
“[VME6] DMA Data Pattern to VME”	✓	✓
PCI Interface		
“[PCI1] Address Parity Error Detection Issue”	✓	✓
“[PCI2] PCI Fast Back-to-Back Capable Issue”	✓	✓

a. Errata numbers in brackets denote interface-specific or general device errata.

[VME1] 2eSST Broadcast Erroneous DTACK* Assertion

Device Mode

2eSST

Description

When a Tsi148 sourced 2eSST broadcast terminates with slave suspended response (i.e slave asserts RETRY*), the Tsi148 enters a DTACK* assertive state. The Tsi148 remains in the DTACK* assertive state until it successfully reruns the 2eSST broadcast that was suspended.

Impact

While in the DTACK* assertive state, the Tsi148 could interfere with other VMEbus masters by erroneously asserting DTACK*. This erroneous assertion of DTACK* is a problem even if the Tsi148 happens to be the intended slave for the access. The issue is likely to surface as data corruption. It is also possible that it could cause other boards in the system to become confused resulting in VMEbus time-outs or even board/system lock-ups.

Work around

The following work arounds are recommended:

1. Program the Tsi148 so that it never performs 2eSST broadcasts..

Normal 2eSST writes operate properly

2. Ensure that no other board in the system can ever become VMEbus master. By doing so the Tsi148 can rerun the 2eSST broadcast until completion without interruption.
3. Ensure that if other boards can become VMEbus master, they will not do so during the Tsi148's 2eSST broadcast time. The Tsi148's 2eSST broadcast time starts when the Tsi148 begins a 2eSST broadcast cycle and ends only after all of the broadcast data has been transferred over the VMEbus.
4. Ensure that no 2eSST broadcast slave in the system ever asserts a slave suspended response (that is, RETRY*). The following example application accomplishes this configuration:

All 2eSST broadcast slaves in the system are Tsi148 based, and each Tsi148's PCI/X master is able to continuously write data to the PCI/X bus as fast as the VMEbus data is received. In other words make sure the Tsi148's internal buffers are not backed up so much that it issues a slave suspended response.

[VME2] 2eSST Broadcast T4 Timing Parameter Specification Violation

Device Mode

2eSST

Description

The Tsi148's VME Slave violates timing parameter T4 by asserting RETRY* late when it signals a slave suspended response to a 2eSST broadcast. The 2eSST timing parameter T4 requires that 2eSST broadcast slaves assert RETRY* no later than 40ns after the assertion of DS0* during a slave suspended response to a 2eSST broadcast. The Tsi148 does not conform to the specification in that it asserts its RETRYOE pin as late as 60ns after it samples its DS0I_ pin asserted. When taking into account a 5ns delay each from the external buffers, the maximum time from DS0* to Tsi148 assertion of RETRY* becomes 70ns.

Impact

The VME Master (or any tracking device) may not see that the Tsi148's response is a slave suspended response. This could confuse multiple boards in the system resulting in possible time-outs or even lockups. Testing has not produced any failures, only timing violations.

Work around

The following work arounds are recommended:

1. The Tsi148 must never be a slave to any 2eSST broadcasts.
2. The Tsi148 must never asserts a slave suspended response during a 2eSST broadcast. The Tsi148's internal buffers must never get backed up, since this requires that the PCI/X master continuously write data to the PCI/X bus as fast as the data is received from the VMEbus.

[VME3] VMEbus Device Wants Bus Issue

Device Mode

All VMEbus Modes

Description

This errata is associated with the VME Device Wants Bus (DWB) functionality of the Tsi148. The DWB and the Device Has Bus (DHB) control bits are in the VME Master Control register (LCSR register group + 0x234). The DWB/DHB functions support exclusive bus ownership through VMEbus arbitration. This errata pertains to releasing bus ownership (by clearing of the DWB bit) and the Tsi148 VME Master either incorrectly initiating or interpreting a VMEbus transaction.

Impact

There is internal race condition between the DWB mechanism releasing bus ownership and the Tsi148 VME Master initiating a VMEbus transaction. The following different results are possible when this race condition occurs.

- When bus ownership is released, there is a possibility that the Tsi148 VME Master will incorrectly start a VMEbus transaction without having a properly qualified bus grant. This is caused by an internal race condition between the DWB mechanism clearing the bus grant and the Tsi148 VME Master initiating a VMEbus transaction. It is possible that the Tsi148 VME Master will collide with another VMEbus master who was given a qualified grant. The error signatures due to a collision are numerous, but the more common signatures are either a VMEbus error or a corrupt transaction.
- It is also possible that the Tsi148 VME Master will incorrectly interpret the release of bus ownership associated with a current pending transaction. The Tsi148 VME Master may incorrectly conclude the completion of a transaction when in fact some or all of the transaction never took place. The error signature for this incorrect interpretation is either corrupt or lost data.

Work around

To guarantee the proper release of bus ownership, the Tsi148 VME Master must not have any transactions pending when bus ownership is released. However, it is very difficult to guarantee this state, particularly when concurrent DMA transactions are occurring. The DMA controller will attempt to utilize the Tsi148 VME Master to satisfy all VME related DMA transactions. It may also be difficult to guarantee the blocking of VMEbus bound programmed I/O transactions originating from PCI/X when multiple threads are competing for VMEbus resources.

Although difficult, in some cases it may be possible to guarantee proper operation of the DWB/DHB functionality. The following actions must be controlled by system software in order to guarantee proper operation of DWB/DHB:

- The DMA controller must be either stopped or paused.

The timing of a successful pause could be unpredictable because a pause of a linked-list DMA process only takes place at descriptor boundaries.

- VMEbus bound posted writes must be flushed by a VMEbus bus bound read.
- No more VMEbus bound transactions can be issued until after bus ownership has been successfully released. This assumes the handling of all threads of control originating from any PCI/X bus master targeted for the VMEbus.

Alternative Design

One use of the DWB/DHB functionality is exclusive resource locking. To work around this problem, it may be possible to use a software semaphore-based solution instead of DWB/DHB. Please contact IDT support for additional work around information.

[VME4] VMEbus Write During Wrap-around RMW Issue

Device Mode

All VMEbus Modes

Description

This errata is associated with the VME Read-Modify-Write (RMW) functionality of the Tsi148. The errata only pertains to wrap-around RMW cycles where the Tsi148 VME Master performs a RMW cycle directed to itself (that is, the same Tsi148's VME Slave interface). This errata does not affect RMW cycles to targets located in a different device.

During a wrap-around RMW cycle, the read is always performed correctly and valid read data is always returned. During the write portion of the RMW, the Tsi148 VME Master may not drive the data bus. It may appear that the Tsi148 VME Master always writes all FF's because of pull-up resistors that reside on the data bus.

Impact

There are timing windows that exist in the VMEbus protocol where the Tsi148 VME Master and the VME Slave could be contending for data bus resources. If left unresolved, the contention could result in a collision between the Tsi148 and the external data bus buffers. To manage this situation, a sideband signal is sent from the VME Slave to the VME Master that notifies of such a condition. The VME Master samples this signal before driving the data bus. In the case of a wrap-around RMW cycle, the sampling of the signal is in error and the VME Master does not drive the data bus during the write portion.

Work around

There are two work arounds for this issue. The first work around is to avoid wrap-around VMEbus RMW cycles in a Tsi148 system. The RMW functionality can be enabled or disabled through a programming option in the Tsi148 VME Control register (CRG register group + 0x234). For more information, refer to the *Tsi148 User Manual*.

The second work around is to adjust the DTACK signal negation timing. The adjustment of DTACK negation timing is a programming option for the Tsi148 within the VMEbus Filter Register (CRG register group + 0x250). It is recommended that ACKD field be programmed to the slow setting (that is, bits 25:24 set to 2’b00). By setting DTACK negation timing to slow, then the exchange of information between the Tsi148 VME Master and the Tsi148 VME Slave will always occur without error and the VME Master will always drive the data bus during the write portion of the wrap-around RMW cycles.

When implementing this work around it is recommended that the adjustment to DTACK negation timing be programmed once at startup. It is possible to adjust DTACK negation timing during normal system operation (that is, as a wrapper around suspect RMW cycles), however this method has not been fully validated and could create undefined results in the system.

Adjusting DTACK negation timing could result in a decrease in performance. It is estimated that in some cases the VMEbus transfer rate degradation could be as high as 5% on all non-source synchronous protocols. Adjusting DTACK negation timing has a negligible affect on source synchronous transfer protocols.

[VME5] VMEbus RETRY* Specification Violation

Device Mode

All VMEbus Modes

Description

When a cycle is terminated with RETRY*, the Tsi148 VME Master does not drive the address and attributes busses as specified in the *American National Standard for VME64 (ANSI/VITA 1-1994)*.

The Tsi148 VME Master detects the assertion of the RETRY* signal and shortly afterwards drives the DS* signals to the de-asserted state. On the same internal clock edge, the Tsi148 VME Master tri-states the address and attributes bus. The de-assertion of DS* and the tri-stating of address and attributes always happens at a fixed time frame after the Tsi148 VME Master has detected RETRY* asserted, and has no relationship to the de-assertion of the RETRY* signal. The VME64 specification states that the tri-state of the address and attributes must occur some finite period of time after the VME master has detected the de-assertion of RETRY*.

This errata relates to Timing Note #48 as described in *American National Standard for VME64 (ANSI/VITA 1-1994)*. The following sections of the specification should be reviewed for more information about this errata:

- Table 2-24: Master, Timing RULEs and OBSERVATIONs
- Table 2-25: Slave, Timing RULEs and OBSERVATIONs
- Figure 2-30: Master And Slave Data Transfer Timing

Impact

During a normal bus cycle, VME slaves see from the assertion of DTACK* that a data transfer is about to complete. The release of the address and attributes simultaneously with the de-assertion of DS* is qualified by the assertion of DTACK*. During a RETRY* cycle, legacy VME slaves do not see the assertion of RETRY* and, therefore, do not have an indication that a data transfer is about to complete.

A race condition is created between the de-assertion of DS* and the tri-state of the address and attributes. The combination of a slow de-assertion of DS* and a fast tri-state of the address and attributes can, in some cases, be misinterpreted by legacy VME slaves as a false decode.

Work around

None

[VME6] DMA Data Pattern to VME

Description

When the DMA operates in the Data Pattern mode and a bus error (BERR) occurs due to accessing an invalid memory region, the DMA Data Pattern circuitry fails to properly terminate its operation causing the DMA engine to remain stuck in busy mode; once it is stuck the DMA cannot be freed by aborting the ongoing transaction. This problem does not happen for Data Pattern transfers whose byte count is 8 Kbytes or smaller. Note that this erratum is contained within the DMA Data Pattern circuitry. This problem does not occur when PCI to VME DMA is executed.

Impact

Attempt to write Data Pattern into an invalid memory could permanently tie up a DMA engine.

Work around

None.

[PCI1] Address Parity Error Detection Issue

Device Mode

PCI or PCI-X mode.

Description

The Tsi148 does not set the PCI Detected Parity Error (DPE) bit in the PCI Command/Status Register (bit 31 of offset 0x04) when an address parity error is detected.

Impact

The transaction with the corrupted address will be accepted by the bridge and passed along undetected.

Work around

Both the Parity Error Response (PERR) and System Error Enable (SERR) bits must be enabled in the PCI Command/Status Register (bits 6 and 8 of offset 0x04). When these bits are enabled the Tsi148 signals a system error and drive its SERR_ output when it detects an address parity error.

[PCI2] PCI Fast Back-to-Back Capable Issue

Device Mode

PCI mode

Description

When the Tsi148 is the target of the second cycle of a PCI fast back-to-back transaction the Tsi148 does not respond with DEVSEL# if another agent on the bus is the target of the first cycle. The Tsi148 PCI target needs to see an idle cycle on the first access it receives. The Tsi148 does support PCI fast back-to back accesses if it is the target of the first cycle.

Impact

The Tsi148 does not assert DEVSEL#. The initiating master will receive a master-abort response.

Work around

PCI fast back-to-back support is optional and can be enabled within the PCI master device. Designs that require support for PCI fast back-to-back transactions must guarantee that the Tsi148 is the target of the first cycle (the write). When the Tsi148 is the target of the first cycle then fast back-to-back support can be enabled within the PCI master.