# Tsi148™ VMEbus Test and Set Application Note

80A3020_AN002_03

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.

# 1. Tsi148 VMEbus Test and Set Application Note

This document details both the traditional mechanisms for implementing a VMEbus Test and Set (TAS) function, as well as additional Tsi148-specific solutions. The following topics are discussed:



## Revision History

### 80A3020_AN002_03, Formal, October 2009

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

### 80A3020_AN002_02, Formal, October 2008

General content update.

## 1.1 Overview

The traditional methods for implementing a VMEbus test-and-set (TAS) function include the following:

1. A VME RMW cycle
   - The standard way to implement a TAS is to use the PCI- to-VME bridge device's RMW cycle to set the semaphore. This method is used by a board that does not host the semaphore in its own memory but needs to access the semaphore over the VMEbus. In order to guarantee indivisibility, both the processor host bridge as well as the PCI-to-VME bridge should support the PCI Lock feature.
2. Using bus locking capabilities in the PCI-to-VME bridge device
   - Locking of the VMEbus by using the bus locking capability of the PCI-to-VME bridge device can be used to complete the TAS function. When the VMEbus is locked, the semaphore can be read, tested, and, optionally, written. The same method is used by the board that hosted the semaphore and boards that must cross the VMEbus to access the semaphore.

3. Software TAS
   — The software TAS option continuously re-reads the address to obtain the semaphore. The number of times the address must be read is a configurable value which must be large enough to ensure that no other agent trying to acquire the semaphore could have written its own node-unique locking value to the semaphore. The number of times the address must be read is very system dependent and it is difficult to guarantee the desired locked condition in all circumstances and environments.

The Tsi148, like most host bridges, does not support the PCI bus LOCK feature. Because of this, the standard method (method 1) of atomic RMW cycles to obtain the semaphore is not supported. Also, due to a current Tsi384 device errata, the Tsi148 does not support the hardware method (method 2) of locking the VMEbus by using the Tsi148 Device Wants Bus/Device Has Bus (DWB/DHB) bus locking capability. For more information, see the *Tsi148 Device Errata*.

The Tsi148 does support the software TAS but this method can introduce latency and uncertainty into the process. Also because it is a software loop, this process consumes processor time while checking for a collision.

However, in specific applications the Tsi148 can be configured to perform a wrap-around RMW cycle for TAS. The following section details the configuration required.

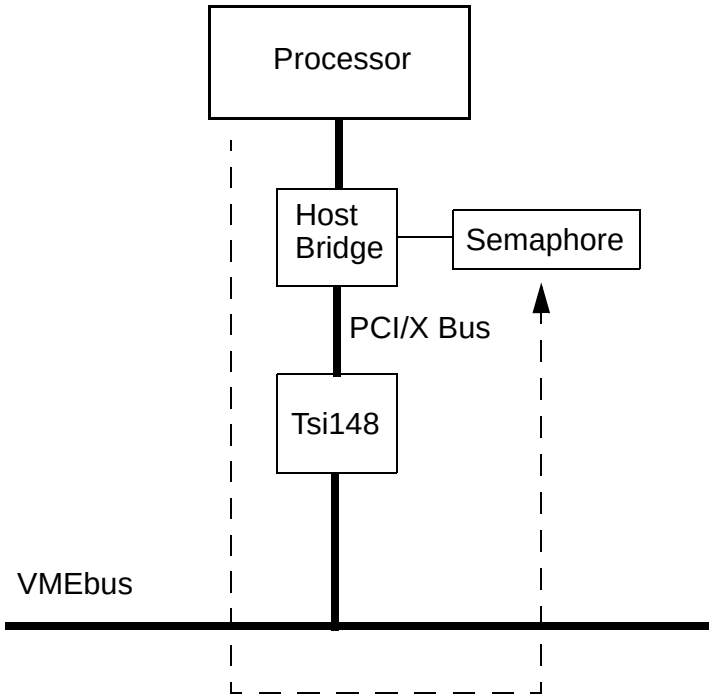
## 1.2 Tsi384-specific Wrap-around RMW Solution

The Tsi148 is capable of performing a wrap-around RMW cycle. The device-specific solution can be used to force all local bus (PCI) accesses to the semaphore to be channeled through the VMEbus and the RMW can be used for TAS.

This feature helps control accesses to the semaphore from the local bus by forcing all the processor bus accesses to memory through the VMEbus (see Figure 1). This method works only when the board hosting the semaphore is the same board who is accessing the semaphore. The Tsi148 is the only PCI-to-VME device that supports the wrap-around RMW. If the Tsi148 is used in a system that contains other boards containing different PCI-to-VME bridges, or a system where a different board hosts the semaphore, then the software TAS method (method 3) must be implemented instead.

The Tsi148's errata associated with the wrap-around RMW does not impact TAS functionality. With the errata, during a wrap-around RMW cycle, the read function is always performed correctly but during the write portion, the VMEbus Master may not drive the data bus (due to pull-up resistors that reside on the data bus, all ones (0xFFFFFFFF) would be written). However, the logic of the test and set is that a read of the cleared condition (which is 0) indicates the semaphore has been acquired. Whether the semaphore was acquired, or already in use, writing 1 does not matter because either the semaphore value was already a 1 or it should be a 1. The important part of the algorithm is that the read portion of the wrap-around RMW is correct. The release of the semaphore, setting it to 0, is not performed by means of a wrap-around RMW; it is performed by a write to the semaphore.

**Figure 1: Tsi148 Wrap-around RMW TAS Method**



*CORPORATE HEADQUARTERS*
6024 Silver Creek Valley Road
San Jose, CA 95138

*for SALES:*
800-345-7015 or 408-284-8200
fax: 408-284-2775
www.idt.com

*for Tech Support:*
email: VMEhelp@idt.com
phone: 408-360-1538
document: 80A3020_AN002_03