



# **MVME7100**

## **Product Errata**

6806800K31B

June 2014

## Copyright

© Copyright 2014 Artesyn Embedded Technologies, Inc.

All rights reserved.

## Trademarks

Artesyn Embedded Technologies, Artesyn and the Artesyn Embedded Technologies logo are trademarks and service marks of Artesyn Embedded Technologies, Inc.© 2014 Artesyn Embedded Technologies, Inc. All other product or service names are the property of their respective owners.

Intel® is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ and all other Java-based marks are trademarks or registered trademarks of Oracle America, Inc. in the U.S. and other countries.

Microsoft®, Windows® and Windows Me® are registered trademarks of Microsoft Corporation; and Windows XP™ is a trademark of Microsoft Corporation.

PICMG®, CompactPCI®, AdvancedTCA™ and the PICMG, CompactPCI and AdvancedTCA logos are registered trademarks of the PCI Industrial Computer Manufacturers Group.

UNIX® is a registered trademark of The Open Group in the United States and other countries.

## Notice

While reasonable efforts have been made to assure the accuracy of this document, Artesyn assumes no liability resulting from any omissions in this document, or from the use of the information obtained therein. Artesyn reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of Artesyn to notify any person of such revision or changes.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to an Artesyn website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Artesyn.

It is possible that this publication may contain reference to or information about Artesyn products (machines and programs), programming, or services that are not available in your country. Such references or information must not be construed to mean that Artesyn intends to announce such Artesyn products, programming, or services in your country.

## Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by Artesyn.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

## Contact Address

Artesyn Embedded Technologies  
Marketing Communications  
2900 S. Diablo Way, Suite 190  
Tempe, Arizona 85282

Artesyn Embedded Technologies  
Lilienthalstr. 17-19  
85579 Neubiberg/Munich  
Germany

---

## Abbreviations

Abbreviation	Description
PCI	Peripheral Component Interconnect
VME	Versa Module Eurocard

## Summary of Changes

Part Number	Publication Date	Description
6806800K31A	December 2009	Initial version.
6806800K31B	June 2014	Re- branded to Artesyn template.

## Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>6</b>
1.1	Purpose of this Document .....	6
<b>2</b>	<b>Known Issue(s) .....</b>	<b>7</b>
2.1	VMEbus Lock-up .....	7
2.1.1	Problem .....	7
2.1.2	Symptom .....	7
2.1.3	Additional Information.....	7
2.1.4	Workarounds .....	7

## List of Tables

Table 1	List of Workarounds .....	8
---------	---------------------------	---

# 1 Introduction

## 1.1 Purpose of this Document

This document describes a product issue that affects the following product variants:

- MVME7100
- MVME7100ET-0163
- MVME7100ET -0161
- MVME7100ET-0171
- MVME7100ET-0173

This document also provides ways to work around the issue.

## 2 Known Issue(s)

### 2.1 VMEbus Lock-up

#### 2.1.1 Problem

A lock-up occurs when a VME read cannot complete. For example, if VME board A is trying to read memory on VME board B while VME board B is streaming writes to the VMEbus a lockup can occur. If the writes fill the buffer in the Tsi148 and backup into the outbound queues of the PCI Express devices, then the read data will be blocked by the writes and the read cannot complete. The pending read on the VMEbus will prevent the stored writes from completing and there is a deadlock. If the writes do not backup in the PCI Express devices, the Tsi148 will allow the read to pass the writes and a deadlock would not occur.

#### 2.1.2 Symptom

The VMEbus will lock-up and the cycle will be terminated with a VME BERR\*.

#### 2.1.3 Additional Information

This lock-up occurs because there is a fundamental incompatibility between the PCI bus and VMEbus. The PCI specifications require bridges to flush outbound data before completing an inbound read. Because the original VMEbus specification did not define a retry function, the flush before read requirement creates a potential lock-up situation. To avoid this lock-up situation, MCG-designed host bridges included an option to disable the flush before function. Some PLX PCI-to-PCI bridges also allow the flush before read function to be disabled. The Marvell host bridges allow the flush before read function to be disabled. Current PCI Express bridges and switches do not allow the flush before read function to be disabled.

The PCI Express bridges, switches and the MC8641D processor used on the MVME7100 do not allow the flush before read function to be disabled. The packet-based PCI Express protocols place the read response packet in the outbound queue requiring any queued writes to complete before the read can complete.

The PCI specifications define a relaxed ordering feature that does allow the read completion to pass the posted writes; however, it is not guaranteed that reads will always pass writes. Also, the Tsi148 does not set the relaxed ordering bit and Tsi384 will not allow the read completion to pass the writes unless the RO bit is set. The PLX switch does not support relaxed ordering.

#### 2.1.4 Workarounds

A guaranteed way to avoid this problem in new designs which involve moving data between boards over VME is to design the application to move data between boards over VME using "writes" only.

Existing applications which are susceptible to this problem should use one of the following workarounds:

Table 1 List of Workarounds

Option	Description
1	<p>Later versions of the VME standard support a retry function. If the boards that are locking up have the Tsi148 chip, the retry function can be enabled. The retry function allows the writes to progress but the read will be retried. However, if the writes are continuous, there will be a read completion time out.</p> <p>While it is possible to create a simple test where board A writes continually to another board B while board B reads continually from board A, where one would run into a read completion timeout, it is quite likely that this will not happen in a practical application. If board A moved on to other chores even at intervals exceeding 100K words, the read completion timeouts can be avoided.</p> <p>Deadlock detection is enabled by setting the DLT field in the VME Bus Control Register.</p>
2	<p>Use the "Device Wants Bus (DWB)" feature of the Tsi148.</p> <p>In this scheme, any board that starts a large block of writes should first claim the VME bus before starting the writes, and release the bus when done.</p> <p>The DWB field is in the VME Bus Master Control Register. The Writer should use this along with the "Device Has Bus (DHB)" bit in the same register to get the bus, check that it has the bus, and release the bus when done.</p> <p>The Tsi148 has errata related to the use of DWB. The user should be aware of these errata if this option is used.</p>
3	<p>The DMA controller in the Tsi148 can be used when large blocks of data are moved. This is more efficient and avoids the lockup conditions. For this to work, DMA would have to be used instead of programmed I/O for any block size above 40 words to avoid bumping into the write post buffer limit of the TSI148.</p>
4	<p>If multiple writes are streamed to the VMEbus, a read should be done after every 32 writes. This will prevent the outbound queue in the Tsi148 from filling up.</p>