Universe II (CA91C142) Device Errata

Visit our Website at http://www.tundra.com

603 March Rd., Kanata, Ontario, K2K 2M5
Tel: (613) 592-0714 • 1-800-267-7231 • Fax: (613) 592-1320

© 1998 Tundra Semiconductor Corporation

Document Number : 8091142.ER600.09

Printed in Canada

Information in this document is subject to change without notice

The information in this document is subject to change without notice and should not be construed as a commitment by Tundra Semiconductor Corporation. While reasonable precautions have been taken, Tundra Semiconductor Corporation assumes no responsibility for any errors that may appear in this document.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Tundra Semiconductor Corporation.

The acceptance of this document will be construed as an acceptance of the foregoing conditions.

Universe II (CA91C142) Device Errata

The Universe II (CA91C142) is the next generation of the Universe (CA91C042) PCI-VME bridge. This document describes the known errata and presents some important design issues.

As this information may change over time, please insure that you have the most recent version of this document which is available on the web at **www. tundra.com** or directly from Tundra or one of its representatives.

Be sure to visit Tundra's *Designers' Resource Centre* at ***www.tundra.com*** for the latest product updates and technical discussions.

Universe II Device Errata

Errata #	Description
1	Disabling of PCI base address
2	SYSCLK and SYSFAIL generation during system reset
3	Invalid release of PCI LOCK# during exclusive accesses
4	DY4 Auto-ID incompatibility
5	Simultaneous, single level interrupts
6	Detect Parity Error Status Bit
7	Illegal assertion of PERR# during Target Write Retry

Universe II Design Notes

Note #	Description
1	Noise on VME Data Strobes
2	DMA operation during PCI Reads
3	Potential for DTACK*/AS* deadlock with some slave cards
4	VME Slave Channel Prefetched Reads
5	PCI Target Channel D16 Posted-Writes

UNIVERSE II DEVICE ERRATA

1. PCI BASE REGISTERS

Description

The Universe II contains two PCI base registers to allow for both greater flexibility in tightly constrained I/O space as well as the “on the fly” option to access the Universe II registers from either I/O or memory space. Both PCI base registers contained in the Universe II will accept the value of “zero” as a valid and decodable address. This differs from the PCI 2.1 specification, where a zero value being written to the PCI base register should disable the register space. Universe II will continue to decode for register accesses using the value “zero” written to the PCI_BS register as the base address for decoding.

Impact

In a typical system, the operating system might write a value of zero to one of the Base Address registers in the Universe II, with the intention of disabling it. The Universe II’s register space, will however become active. This may result in simultaneous decoding of cycles by multiple devices on the PCI bus within an address range immediately above address 0.

Solution

Software should be written such that it is aware that the Universe II register space may not be disabled, but only changed to other addresses.

2. SYSCLK AND SYSFAIL GENERATION DURING SYSTEM RESET

Description

During a system reset (SYSRST* assertion), the Universe II will stop generating SYSCLK when the Universe II is the SYSCON and BGIN3 in asserted. This violates Rule 5.3 of the VME64 specification which states that the System Clock driver must continue to provide SYSCLK regardless of the state of SYSRST* line. The Universe II will start driving SYSCLK about 8 PCI clocks after SYSRST* negation and will stop driving SYSCLK about 3 PCI clocks after SYSRST* assertion.

Also, in violation of rule 5.8 of the VME64 specification which states that SYSFAIL* must be asserted within 50 milliseconds from the falling edge of SYSRST*, the Universe II will drive SYSFAIL* only on the rising edge of SYSRST*.

Impact

In most systems, the erraneous generation of SYSCLK will have no impact. Boards that rely upon continuous SYSCLK operation during system reset will exhibit unpredictable behaviour.

During a a system fail safe routine and upon SYSRST* assertion, if an application requires to have SYSFAIL* enabled for proper diagnostic routines or power down sequences, the Universe II will not meet this requirement and could cause damage to the application's hardware or software.

Solution

The user will have to provide an external clock driver using the Universe II as the system controller if boards rely upon SYSCLK operation during SYSRST* assertion. Very few boards in production today rely upon SYSCLK for proper operation. Fewer boards still rely on it being stable during an assertion of SYSRST*. As such, this errata is unlikely to pose a problem in the majority of systems.

To meet the proper operation of SYSFAIL*, the following sequence must be implemented with the Universe II:

-Always assert SYSFAIL within 50 milliseconds during PWRRST* or SYSRST*.

-If the SYSFAIL* and VME64 Auto Id power up option of the Universe II is not selected, then SYSFAIL* must be negated once SYSRST* is negated.

-If the SYSFAIL* power up option is selected but not the VME64 Auto Id option, then SYSFAIL* must remain asserted until the SYSFAIL bit in the VCSR_CTL register or the VME64 Auto Id bit of the MISC_CTL register are written to.

-If the VME64 Auto Id option isd selected, then SYSFAIL* remains asserted until IRQ2* is enabled, regardless of the SYSFAIL* power up option.

3. INVALID RELEASE OF PCI LOCK# DURING EXCLUSIVE ACCESSES

Description

When an external VME master generates a VMEbus LOCK/ADOH to the Universe II, the cycle is processed on the PCI bus as a read cycle with the LOCK# pin asserted (Page 2-20 of the Universe II manual). If the target responds with a Master Completion Termination, then the Universe II will improperly relinquish LOCK# even though VME BBSY* is still asserted. The above situation is also true during VME RMW cycles should the PCI target respond to the initial read portion of the cycle with a Master Completion Termination. A Master Completion Termination may legally be generated by any PCI target in response to these Universe II cycles. However only a subset of PCI targets will actually do so. Whether the specific target will terminate in this manner is a function of the particular manner in which its PCI state machines have been designed.

Impact

VME RMWs and LOCK/ADOHs to PCI targets that respond with Master Completion Termination will not be guaranteed to be exclusive. The exclusivity will extend only to other VME masters. PCI bus Masters internal to the board may potentially gain access to the relevant resource when the Universe II releases its LOCK#.

Solution

If this situation is encountered, there is a solution by which the Universe II's semaphores can be used to ensure exclusive accesses. Refer to the Universe II manual for details on how to utilize the semaphores.

4. DY4 AUTO-ID INCOMPATIBILITY

Description

When using the DY4 Auto-ID mechanism in the Universe II, the propagation delay through the part from IACKIN* to IACKOUT* will be five clocks instead of four.

Impact

Existing DY4 Auto-ID algorithms may fail to correctly identify relative Universe II board positions.

Solution

Software should account for the fact that the DY4 Auto ID cycle for Universe II’s will consume an extra clock cycle.

5. SIMULTANEOUS, SINGLE LEVEL INTERRUPTS

Description

Universe II devices will attempt to claim IACK cycles that have not been granted to them by the IACK daisychain.

After asserting a VME IRQ, the Universe II will respond to any IACK cycle with a matching level, independent of whether IACKIN* has been asserted to the Universe II.

The result is that the Universe II incorrectly qualifies the assertion of VSLAVE_DIR (which enables DTACK* onto the backplane) on IACK* assertion rather than a combination of IACK* and IACKIN*. The assertion of VSLAVE_DIR will assert DTACK* high onto the VME backplane, where it will remain high until the IACK cycle is complete. If the Universe II receives an assertion of IACKIN* then it will properly drive DTACK* low to acknowledge the IACK cycle.

The result is that the Universe II incorrectly qualifies the assertion of VSLAVE_DIR on VIACK* assertion. When VSLAVE_DIR is asserted, the Universe II drives DTACK*. However, instead of driving DTACK* low as it would to properly claim the cycle, it drives it high (the qualifier for DTACK* is *correctly* associated with VIACKI*).

Impact

If a number of Universe II boards simultaneously assert the same level IRQ there is a chance that the card attempting to properly respond to the IACK cycle with DTACK* low will never be seen because all of the other Universe II cards will be driving DTACK* high. In addition, as a result of this problem, the VME64 Auto Id mechanism cannot be used if multiple Universe II components are present in the system.

Due to the asynchronous drive strength of most VME DTACK* drivers, the problem will most likely only be encountered when four or more Universe II based boards have simultaneous, same level IRQ's pending.

Solution

There are two hardware workarounds.

1. Place an open collector output between the Universe II's DTACK* line and the VMEbus, or, replace the existing DTACK* driver with an open collector driver.

The open collector will ensure that the Universe II will never be able to drive DTACK* high, thus, never overdriving proper DTACK* low assertion. The downside to this solution is that the Universe II will no longer be able to rescind DTACK* and will lose the performance enhancement this feature offers.

2. Qualify the VSLAVE_DIR driving DTACK* out to VME with VIACKI# AND VIACK# as shown in Figure 1. This will ensure that DTACK* is driven out during an IACK cycle when *both* IACKIN* and IACK* are asserted.

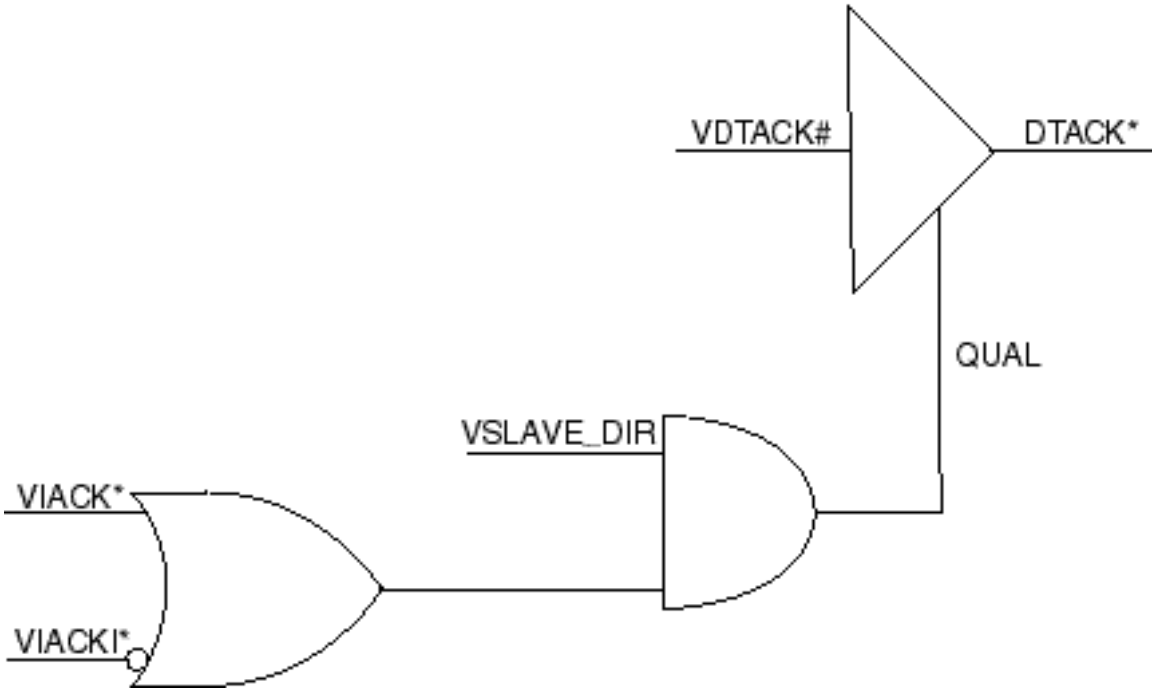

Figure 1: IRQ DTACK* Contention Solution

(Note: The workaround above will minimize the impact of errata #5, however, there remains the potential for DTACK* glitching between the time when the interrupt handler negates IACK* (which will cause the above workaround to release its blocking of VSLAVE_DIR) and when the Universe II releases its VSLAVE_DIR (see Figure 2). The result may be a brief period where the Universe II incorrectly drives DTACK* high after the interrupt handler acknowledges the DTACK* assertion by negating IACK*. Depending upon the number of Universe II's incorrectly driving DTACK* high, the result will be either i) noise introduced on the DTACK* line, or ii) an early recinsion of DTACK* as the high driving DTACK*'s overdrive the low driving one.)

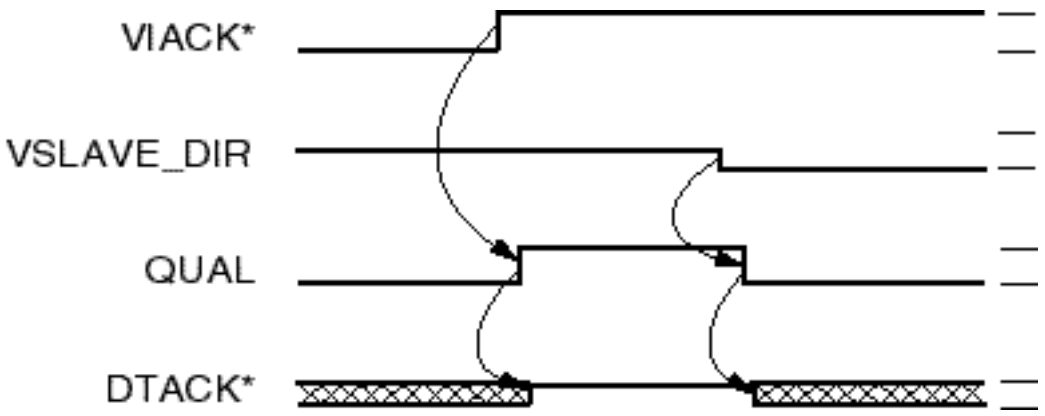

Figure 2: Potential Glitch Associated With Figure 1

6. DETECT PARITY ERROR STATUS BIT

Description

The Detected Parity Error Status bit, DP_D in the PCI_CSR register, is set whenever there is a data parity error during any write on the PCI bus and IRDY# is asserted regardless of whether the data was intended for the Universe II or not. The Detected Parity Error Status bit is also set whenever there is a data parity error during any read on the bus and TRDY# is asserted.

The Detected Parity Error Status bit should only be set when a parity error occurs and the Universe II is the target of a write or the Universe II is the master of a read.

Impact

If a parity error occurs on a system and the Universe II has been programmed to assert PERR# then the Universe will erroneously asset PERR# for cycles for which parity errors occur and the cycle was not intended for the Universe II.

Solution

There are two possible solutions to this errata:

1 . Whenever a parity error occurs, regardless of where the DP_D and DP_E bits in the PCI_CSR register should be cleared by the software.

Place external logic on the PAR and PAR64 lines which pass through the values on the PAR and PAR64 lines when the cycle is intended for the Universe II and when it is not the logic creates the correct value on PAR and PAR64 to ensure that the Universe II will not see a parity error. See Figure XXXXXXXXXXX

7. ILLEGAL ASSERTION OF PERR# DURING TARGET WRITE RETRY

Description

The Universe II illegally asserts PERR# when it is the target of a write transfer and it detects a data parity error but it cannot process the transaction so responds with retry. The Universe II should not assert PERR# as if a PCI Target ever asserts PERR#, it must also assert TRDY# which it is not when asserting STOP# and not TRDY# (Retry).

Impact

The master transferring the data may be confused ….. and …..

The asserting of PERR# may cause an unnecessary interrupt routine to be called which will result in poor bus performance.

Solution

Place external logic on the PERR between the Universe II and the PCI bus which will only pass through PERR when TRDY is asserted or IRDY is asseted.

UNIVERSE II DESIGN NOTES

1. NOISE ON VME DATA STROBES

Description

The Universe II may, under some circumstances, be sensitive to noise on the rising edge of the VME data strobes. This is particularly true in large systems with heavily loaded backplanes where a long rise time on the data strobes may be observed. In this situation, a small low-pass filter or Schmidt trigger circuit may be implemented on the inbound DS0* and DS1* to filter out noise of less than a few ns. There may be a small impact on performance from the filter design.

2. DMA OPERATION DURING PCI READS

Description

When performing DMA reads from the PCI bus, the Universe II will prefetch up to the aligned address boundary defined in the PABS field of the MAST_CTL register. This means that the Universe II may read beyond the programmed transfer length on PCI. As a result, the Universe II's DMA should only be used with memory which has no side-effects when read. The correct number of bytes, programmed in the DTBC register, will be written to VME.

Prefetching can be avoided by programming the DMA for transfers that terminate at the PABS boundary. If further data is required beyond the boundary, but before the next boundary, the DTBC register may be programmed to eight byte transfers. The DMA will fetch the full eight bytes, and nothing more. Programming the DTBC to less than eight bytes will still result in eight bytes fetched from PCI.

In addition, the starting PCI address of the transfer will always be 64 bit aligned. For example, if the designer programs the DLA register to start at an address offset of x0C, the DMA will actually start reading at address offset x08.

In any case, any extra data read from PCI will never be transferred to VME.

3. POTENTIAL FOR DTACK*/AS* DEADLOCK WITH SOME SLAVE CARDS

Description

When performing IACK cycles and coupled 8 or 16 bit cycles as a VME master, the Universe II will wait for the VME slave to release DTACK* before it removes AS*. If the slave card links release of DTACK* to the release of AS*, a deadlock condition will result where both master and slave wait for each other to end the cycle. The ANSI VME64 specification specifies no slave relationship between DTACK* and AS*, therefore no assumptions on its behaviour should be made by the slave.

The Universe II's conditioning of AS* release on DTACK during coupled cycles may be disabled by enabling the Coupled Window Timer in the LMISC register at offset x184. Enabling the timer will prevent any possible deadlock condition when operating against incompatible slave cards. The condition can not be disabled for IACK cycles.

4. VME SLAVE CHANNEL PREFETCHED READS

Description

If an external VMEbus master performs a single BLT or MBLT prefetched read with VMEbus Address bits A[10:3] equal to 0xFF, the Universe II will read the first data block on the PCI bus. Then the Universe II will continue to read data from the PCI bus until AS* is negated on the VMEbus.

The addresses that the Universe II continues to read from is the following : It maintains the same upper address bit (A[31:11]) and the lower bits (A[10:3]) of the address counter roll over to equal 0x00. The Universe II continues reading from this new address.

This does not affect data integrity but impacts performance by consuming bandwidth.

When an external VMEbus master performs a BLT, the Universe II should only prefetch on the PCI side up to a 256 byte aligned boundary. The Universe II will continue to read past this boundary.

5. PCI TARGET CHANNEL D16 POSTED-WRITES

Description

Whenever a PCI Target Image is programmed with PWEN=1, VDW=16-bits, 32-bits, or 64-bits and VCT=0 and an external PCI Master begins a 32-bit write burst with A2=0 and BE#=0011 followed by transfer with BE#=1100, the transfers do not appear on the VMEbus as 16-bit transfers, as expected, but as 8 bit transfers.

A series of 8-bit transfers is also performed if 64-bit posted write is received with BE#=11000011.

This does not affect data integrity but impacts performance for systems which regularly perform 16-bit transfers.