Tsi620 Design Notes

April 11, 2014

GENERAL DISCLAIMER

Integrated Device Technology, Inc. ("IDT") reserves the right to make changes to its products or specifications at any time, without notice, in order to improve design or performance. IDT does not assume responsibility for use of any circuitry described herein other than the circuitry embodied in an IDT product. Disclosure of the information herein does not convey a license or any other right, by implication or otherwise, in any patent, trademark, or other intellectual property right of IDT. IDT products may contain errata which can affect product performance to a minor or immaterial degree. Current characterized errata will be made available upon request. Items identified herein as "reserved" or "undefined" are reserved for future definition. IDT does not assume responsibility for conflicts or incompatibilities arising from the future definition of such items. IDT products have not been designed, tested, or manufactured for use in, and thus are not warranted for, applications where the failure, malfunction, or any inaccuracy in the application carries a risk of death, serious bodily injury, or damage to tangible property. Code examples provided herein by IDT are for illustrative purposes only and should not be relied upon for developing applications. Any use of such code examples shall be at the user's sole risk.

About This Document

This document contains device notes for the Tsi620.

Revision History

April 11, 2014

Added “Packet discard on link partner failure”

August 2009

No technical changes were made to this version of the document.

January 2009

This document includes one new design note, “Multi-master clock generation”. This design note used to reside in the *Tsi620 Device Errata*.

October 2008

This is the first version of the *Tsi620 Design Notes*.

Design Notes

1. Transmitting on lane 0 and lane 2 in 1x mode

During 1x mode and 4x mode initialization, the Tsi620 sends idle characters when it is in the SEEK state during the system discovery. If 4x lane alignment is not achieved after discovery, the Tsi620 does not transmit on lane 2 if the 1x fail down configuration results in the 1X_MODE_LANE0 state, and it does not transmit on lane 0 if the 1x fail down results in the 1X_MODE_LANE2 state.

TIP For detailed descriptions for the states mentioned above, see the *RapidIO Interconnect Specification (Revision 1.3)*.

2. Port power-down and default configuration

Each RapidIO port has a copy of some global RapidIO registers. For the complete list of these global registers, see “Per Port Copies of Global Registers” in the *Tsi620 User Manual*.

When a RapidIO port is powered down, the port’s copy of the global RapidIO registers are returned to their default power-up settings. For example, multicast and port-write settings return to their default power-up settings after a port reset. After a port is powered down and brought back, the port’s copy of the global RapidIO registers must be reprogrammed by writing to the global RapidIO registers or to the Per Port Copies of Global Registers for that port.

3. Four 1x links to 4x port training issue

Connecting four 1x links to a 4x port is not supported and may result in false lane alignment.

The Tsi620 correctly detects alignment of idle lanes, which allows the links to establish initial connection. When data is transmitted between the links, however, the misalignment of data and CRCs prevent correct operation of the link; that is, the Tsi620 expects data to be striped across the four lanes of the link and does not receive the data in that format from the four 1x links.

In an application that requires this configuration, the Tsi620 can use two ports configured as 1x to support this mode of operation with a four 1x device on the other end of the link.

4. PCI configuration read transaction terminated with master abort

When a PCI configuration read is terminated with a master abort, a RapidIO error response is returned to the originator. Software that is enumerating the PCI bus over RapidIO must expect a RapidIO error response instead of the traditional 0xFFFFFFFF value when attempting to enumerate a device that does not exist.

5. Using the Tsi620 without the PCI Interface

When the Tsi620’s PCI Interface is not used in a customer application, it must be configured as follows:

- Pull up the AD[31:0] signals with one common 4.7kÙ resistor.
- Pull up the following signals with one common 4.7kÙ resistor:
  - PCI_CBEn[3:0]
  - PCI_DEVSELn
  - PCI_FRAMEn
  - PCI_GNTn[4:1]
  - PCI_IDSEL
  - PCI_IRDYn
  - PCI_PAR
  - PCI_PERRn

- PCI_SERRn
- PCI_STOPn
- PCI_TRDYn

- Pull down the following signals:
  - PCI_M66EN
  - PCI_PLL_BYPASS
  - PCI_ARBEN
  - PCI_HOLD_BOOT
- Pull up the PCI_RSTDIR signal with a 4.7kÙ resistor.
- Connect the following power supplies:
  - PCI_PLL_AVDD
  - PCI_PLL_AVSS
  - CLKGEN_PLL_AVDD
  - CLKGEN_PLL_AVSS
  - VDD_PCI
- Connect PCI_CLKO0 to PCI_CLK.
- Include an I^{2}C EEPROM in the design to initialize the following bits to 1 (this will power down the bridge after every reset):
  - PWDN_X4 in the SREP Digital Loopback and Clock Selection Register
  - PWDN_X4 in the RapidIO Serial Port 8 Digital Loopback and Clock Selection Register

6. Using the Tsi620 without the FPGA Interface

When the Tsi620's FPGA Interface is not used in a customer application, it must be configured as follows:

- Pull up the SP6_PWDN signal with a 4.7kÙ resistor.
- Pull down the following signals with one common 4.7kÙ resistor:
  - SP6_RXD[31:0]
  - SP6_RXCLK
  - SP6_RXCTL[3:0]
- Pull down the following signals with one common 4.7kÙ resistor:
  - SP6_TXD[31:0]
  - SP6_TXCLK
  - SP6_TXCTL[3:0]
  - SP6_PHY_DISABLE

- Pull up the SP6_RX_ERROR signal with a 4.7kÙ resistor.
- Connect the power supplies:
  - — VDD_HSTL
  - — SP6_VREF

7. Masterless I^{2}C bus busy

This design note applies only to designs that require the Tsi620 to load registers from an I^{2}C EEPROM.

Because EEPROM devices do not have reset pins, if the Tsi620 is reset the EEPROM is unaffected and can continue to drive data at the previous state. If the EEPROM continues to drive the I^{2}C data signal to 0, the Tsi620 determines the bus is busy and does not attempt to reset the EEPROM. As a result, the Tsi620 does not reload the register values after the reset is removed. Unexpected operation after a reset can result if register values cannot be loaded.

The *I^{2}C Specification (*for multiple master support) specifies that the I^{2}C bus is considered busy when the I^{2}C data signal is 0.

Hardware Work Around

To avoid this condition, design the reset of the Tsi620 and all other I^{2}C masters, so that the I^{2}C bus is always idle before asserting reset for the Tsi620 or any I^{2}C bus master.

Software Work Around

To implement a software work around, complete the following:

1. Determine that the I^{2}C bus is busy while there is no master. To do so, read the registers in Table 1; the register values must match those described in the table.

Table 1: Register Values to Diagnose Masterless I^{2}C Bus Busy

Register Name	Register Offset	Register Value Descriptions
I^{2}C Interrupt Status Register	11C	BL_OK and BL_FAIL bits are both 0
I^{2}C Event and Event Snapshot Registers	300	0x00001F00 ANDed with the register value = 0
Internal I^{2}C Status Register 1	3D0	0x0000000F ANDed with the register value = 0x0000000B
Internal I^{2}C Status Register 2	3D4	Register Value = 0x0000021
Internal I^{2}C Status Register 2, read 200 microseconds later	3D4	Register Value = 0x0000020
Internal I^{2}C Status Register 3	3D8	0x00000E00 ANDed with the register value = 0x00000600

2. Issue a reset on the I^2C bus. Driving nine I^2C clock cycles completes an interrupted transfer. An I^2C clock cycle occurs whenever the I^2C clock is driven low for at least five microseconds, and then is released to be high.

   Multiple I^2C EEPROM devices document driving nine I^2C clock cycles for reset.

   The register accesses listed in Table 2 drive I^2C clock cycles to complete the interrupted I^2C bus transfer. The sequence of writes in the table must be repeated nine times.

Table 2: Creating an I^2C Bus Reset

Register Name	Register Offset	Register Value
Internal I^2C Control Register	3C0	Write 0x00000008, wait five microseconds.
Internal I^2C Control Register	3C0	Write 0x00000000, wait five microseconds.

3. Trigger a reset of the Tsi620 to perform the register value loading. There are a number of different methods to reset the Tsi620 documented in the *Tsi620 User Manual*. It is also possible for the host processor to reset the Tsi620 by system specific means.

   ⚠ To implement the software work around, the Tsi620 must be configured to allow host processor access after reset using the power-up configuration pins (for more information, see the *Tsi620 User Manual*).

Testing

The software work around can be tested using the *JTAG Register Access Software*. This software includes scripts that recreate the Masterless I^2C Bus Busy condition.

8. BST_2_BLK mode and write coalescing

The SREP_I2R_LUT_TA_UPPER[BST_2_BLK] setting in the SREP module is used to more efficiently segment PCI transactions that have byte enables associated with every 8 bytes. The BST_2_BLK segmentation assumes that these byte enables are contiguous; that is, all bytes are enabled except for the first four and last four bytes of a transaction.

Some PCI-enabled processors perform “write coalescing” on the PCI bus, whereby two or more writes are combined into a single write. The write transaction, however, may have discontiguous byte enables which causes Bridge ISF Byte Enables Discontiguous events (I_BE_DISCONT in offset 0x780) to be detected.

Impact

There is a modest performance impact for the first two work arounds.

When BST_2_BLK segmentation is disabled, many small write transactions are created for PCI-to-RapidIO writes. This may have the effect of reducing throughput in the RapidIO Switch portion of the Tsi620.

Work Around

1. If possible, disable write coalescing in the processor.
2. Only perform transactions that will not be coalesced.
3. Disable BST_2_BLK segmentation in SREP.

9. Multi-master clock generation

Description

When the Tsi620 I^{2}C Interface is in a multi-master system, the Interface does not generate a correct clock low period. The error condition may occur when both of the following conditions are met:

- Both the external master and the I^{2}C Interface are generating the clock
- The external master pulls the clock low two reference clock cycles before the I^{2}C clock high timer expires

These conditions are possible only when an external master illegally attempts to use the bus when the Tsi620 is the bus master.

Impact

In an expected configuration, two masters are unlikely to be operating at the same time. As well, experiencing this issue requires precise timing between the two masters. Because of these factors, this issue is not likely to occur in a system.

Work around

This issue does not occur when theTsi620’s I^{2}C Interface is the only master on the bus.

10. Packet discard on link partner failure

The Tsi620 was designed to allow systems to continue operating when a link partner has been reset or otherwise failed. The device must perform two actions to allow a system to continue to operate:

- Notify the system host that a link partner has failed
- Discard packets destined for the failed link partner

Port-writes, triggered when the ERR_RATE_CNT bit in the RapidIO Port x Error Rate CSR exceeds the ERR_RFT bit threshold in the RapidIO Port x Error Rate Threshold CSR, should be used to notify the system host that a failure has occurred.

Two discard mechanisms should be used:

- Set the DROP_EN and STOP_FAIL_EN bits in the RapidIO Port x Control CSR to discard packets when ERR_RATE_CNT in the RapidIO Port x Error Rate CSR exceeds the ERR_RFT threshold in the RapidIO Port x Error Rate Threshold CSR, and the port is not in output error-stopped state. This is known as the “standard” discard mechanism.
- Enable the Dead Link Timer with the minimum time value in order to discard packets until the link has reinitialized.

Usually, systems follow a “fail stop” philosophy. Once a fault is detected on a link, all traffic destined for the link must be discarded until software recovers the link. Packet discard due to the Dead Link Timer will cease once the link has reinitialized. If the link partner has failed only temporarily, or the link has reinitialized due to a high temporary bit error rate, the standard discard mechanism will operate after the link has reinitialized. However, if the link partner was reset, the port will detect an ackID mismatch, enter output error-stopped state, and will be unable to discard packets.

Systems that must support link partner reset must set the PORT_LOCKOUT bit in the RapidIO Serial Port x Control CSR before the link reinitializes in order to ensure that packet discard continues. The implication is that software must be capable of receiving a port-write, processing the port write, and setting the PORT_LOCKOUT bit in less than 80 microseconds.

Note that the board reset controller could support holding devices in reset for a period long enough to ensure that software can set the PORT_LOCKOUT bit.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.