Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Tsi350A™ PCI-to-PCI Bridge User Manual

80D5000_MA001_08

September 5, 2009

6024 Silver Creek Valley Road, San Jose, California 95138
Telephone: (800) 345-7015 • (408) 284-8200 • FAX: (408) 284-2775
Printed in U.S.A.

About this Document

This section discusses the following topics:

Scope

The *Tsi350A PCI-to-PCI Bridge User Manual* discusses the features, capabilities, and configuration requirements for the Tsi350A. It is intended for hardware and software engineers who are designing system interconnect applications with the device.

Document Conventions

This document uses the following conventions.

Non-differential Signal Notation

Non-differential signals are either active-low or active-high. An active-low signal has an active state of logic 0 (or the lower voltage level), and is denoted by a lowercase "b". An active-high signal has an active state of logic 1 (or the higher voltage level), and is not denoted by a special character. The following table illustrates the non-differential signal naming convention.

State	Single-line signal	Multi-line signal
Active low	NAME_b	NAMEn[3]
Active high	NAME	NAME[3]

Object Size Notation

- A *byte* is an 8-bit object.
- A *word* is a 16-bit object.
- A *doubleword* (Dword) is a 32-bit object.

Numeric Notation

- Hexadecimal numbers are denoted by the prefix *0x* (for example, 0x04).

- Binary numbers are denoted by the prefix *0b* (for example, 0b010).
- Registers that have multiple iterations are denoted by {x..y} in their names; where *x* is first register and address, and *y* is the last register and address. For example, REG{0..1} indicates there are two versions of the register at different addresses: REG0 and REG1.

Symbols

This symbol indicates a basic design concept or information considered helpful.

This symbol indicates important configuration information or suggestions.

This symbol indicates procedures or operating levels that may result in misuse or damage to the device.

Document Status Information

- Advance – Contains information that is subject to change, and is available once prototypes are released to customers.
- Preliminary – Contains information about a product that is near production-ready, and is revised as required.
- Formal – Contains information about a final, customer-ready product, and is available once the product is released to production.

Revision History

80D5000_MA001_08, Formal, September 2009

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

80D5000_MA001_07, Formal, April 2008

This document supports the production version of the Tsi350A. The asynchronous mode functionality was removed from this document. Information has been removed from “Secondary Clock Outputs” on page 84 and pin 52 in “208-pin PQFP Pin List” on page 104 was changed from ASYNC_MODE to VSS.

80D5000_MA001_06, Formal, January 2008

This document supported the production version of the Tsi350A. The Tsi350A is a performance enhancement of the Tsi350 and there are no functional changes between the devices. All technical information in this document applies to both the Tsi350 and the Tsi350A.

80D5000_MA001_05, Formal, August 2007

The changes to this document were minor and include register address clarifications in the register chapter and a compliance list added to the overview chapter.

80D5000_MA001_04, Formal, March 2007

The following chapters were extensively edited:

- “Signals and Pinout” on page 93
- “Electrical Characteristics” on page 119
- “Package Information” on page 169

Contents

Figures

Tables

1. Functional Overview

This chapter describes the main features and functions of the Tsi350A. The following topics are discussed:

1.1 Overview of the Tsi350A

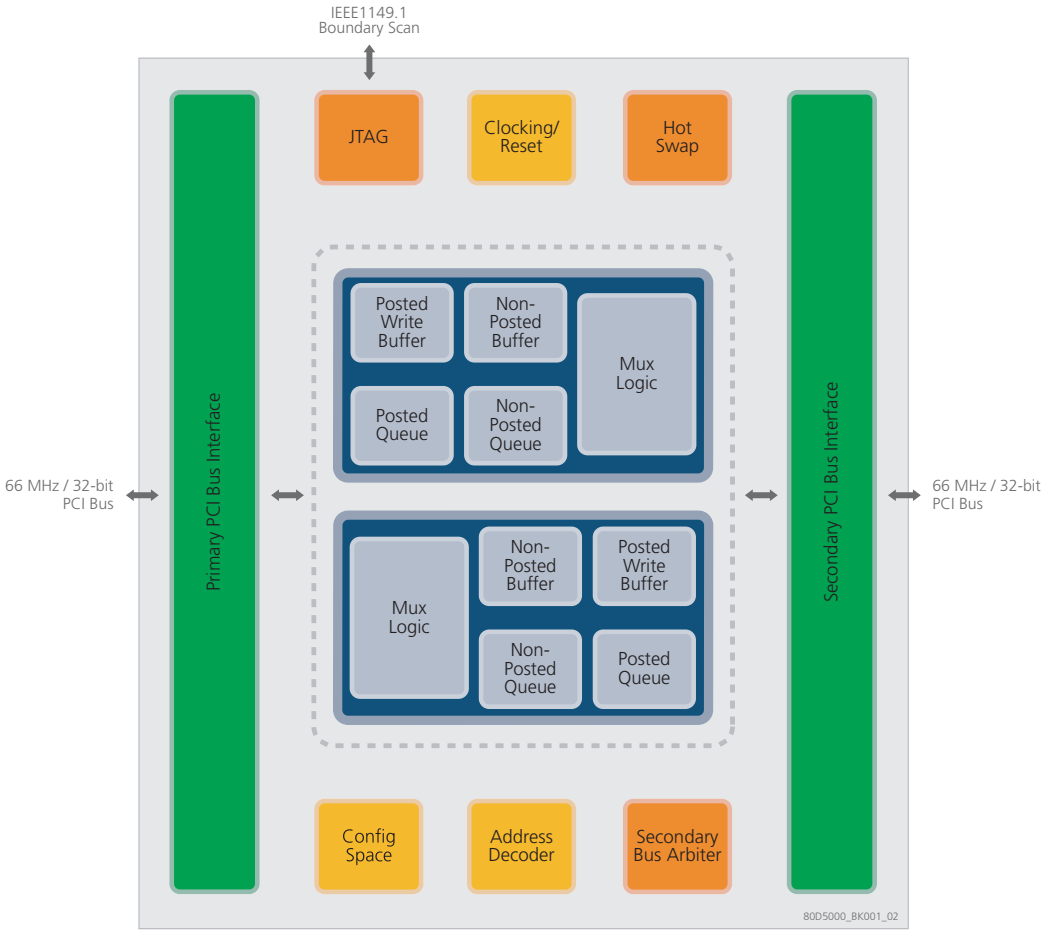
The IDT Tsi350A is a PCI-to-PCI bridge that is fully compliant with *PCI Local Bus Specification, Revision 2.3*. The Tsi350A has sufficient clock and arbitration pins to support nine PCI bus master devices directly on its secondary interface.

The Tsi350A allows the two PCI buses to operate concurrently. This means that a master and a target on the same PCI bus can communicate while the other PCI bus is busy. This traffic isolation may increase system performance in applications such as multimedia.

The Tsi350A makes it possible to extend a system's load capability limit beyond that of a single PCI bus by allowing motherboard designers to add more PCI devices or more PCI option card slots than a single PCI bus can support.

The Tsi350A has two identical PCI Interfaces that each handle PCI transactions for its respective bus, and, depending on the type of transaction, can act as either a bus master or a bus slave. These interfaces transfer data and control information flowing to and from the blocks shown in Figure 1.

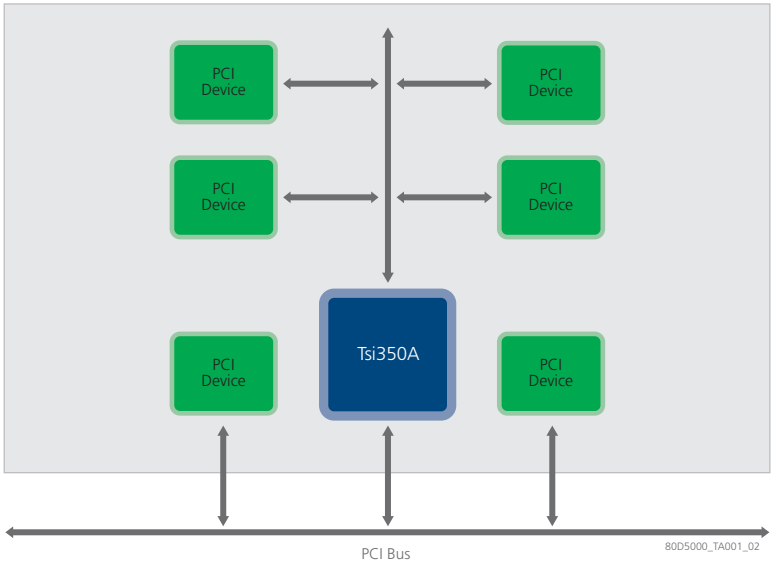
Figure 1: Block Diagram



Option card designers can use Tsi350A to implement multiple-device PCI option cards. Without a PCI-to- PCI bridge, PCI loading rules would limit option cards to one device. The PCI Local Bus Specification loading rules limit PCI option cards to a single connection per PCI signal in the option card connector. The Tsi350A overcomes this restriction by providing, on the option card, an independent PCI bus to which up to nine devices can be attached.

Figure 2 shows how the Tsi350A enables the design of a multi-component option card or expand existing PCI buses.

Figure 2: System Block Diagram



1.1.1 Features

- Industry-standard 32-bit, 66-MHz PCI bridge
- Fully *PCI Local Bus Specification, Revision 2.3* compliant
- Supports up to nine PCI bus masters on the secondary interface
- Ten independent secondary clock outputs to the secondary slots
- Primary and secondary interfaces can be operated using asynchronous clocks
- Secondary clock can either be derived from the input primary clock or supplied by an external clock source
- Secondary clocks can be masked through the GPIO interface during power up
- Supports four independent delayed transactions in each direction
- Supports up to nine secondary requests and grants
- External arbiter support on the secondary bus
- Supports CompactPCI Hot Swap functionality
- C I Power management with D3Hot support with option to disable clocks during D3Hot state
- Supports Bus Locking mechanism
- VGA/Palette memory and I/O decoding options
- Optional non-posted entry flush upon posted writes traveling the same direction
- Compatible with existing solutions from Intel, TI, PLX, and Pericom

1.1.2 Compliance

- *PCI-to-PCI Bridge Architecture Specification (Revision 1.1)*
- *PCI Local Bus Specification (Revision 2.3)*
- *PCI bus Power Management Interface Specification (Revision 1.1)*
- *Advanced Configuration Power Interface (ACPI)*
- *PICMG CompactPCI Hot-Swap Specification (Revision 2.0)*

1.2 Functional Description

This chapter outlines functionality of various interfaces and major blocks.

1.2.1 PCI Interface

Tsi350A has two PCI interfaces, one on the primary side and another on the secondary side. These interfaces transfer data/control information to and from BLU (Buffer Logic Unit).

1.2.1.1 Posted Write Buffer

This buffer is used as temporary storage for posted memory write data flowing from one interface to the other. Each posted buffer has a capacity of 128 bytes. The amount of buffer locations allotted for a transaction is dynamic. A single transaction can utilize from one memory location (4 bytes) to 32 memory locations (128 bytes) as needed.

1.2.1.2 Posted Write Queue

Posted Write Queue is used to store control information related to the posted write transaction flowing from one interface to the other. Each Posted Write Queue is a 4-entry FIFO, providing four active posted write transactions in each direction. Data related to each entry is stored in the Posted Write Buffer. Posted Write Queue will accept entry from external master only when at least one entry is free and at least one Q-word space is available in Posted Write Buffer.

1.2.1.3 Non-Posted Buffer

Non-Posted Buffer is used to hold data for read transactions. Each Non-Posted Buffer contains 128 bytes of buffer space. The amount of buffer locations allotted for a transaction is dynamic. A single transaction can utilize from one memory location (4 bytes) to 32 memory locations (128 bytes) as needed. Tsi350A will start giving data on originating bus once the first four locations of the buffer are filled or the transaction is completed on the target bus.

1.2.1.4 Non-Posted Queue

Non-Posted Queue is used to store the control information related to the all non-posted transactions. Each Non-Posted Queue is a 4-entry FIFO, providing 4-active non-posted transactions in each direction. Data related to each entry is stored in the Non-Posted Buffer. The Non-Posted Queue will accept a transaction from external master only when at least one entry is free.

1.2.1.5 Mux Logic

This module arbitrates the transactions from the Posted Queue and Non-Posted Queue.

1.2.1.6 Configuration Space

The configuration registers help the system software to configure behavior of the bridge. The bridge has Type 01 configuration header support as per the specification. All of these registers function exactly as specified in the *PCI to PCI Bridge Architecture Specification, Revision 1.2*. The configuration space can be accessed only from primary bus interface. Tsi350A implements device specific configuration registers to enable software to program the bridge features.

1.2.1.7 Address Decoding Logic

Tsi350A operates in transparent mode. In transparent mode I/O, memory, pre-fetchable memory base and limit, and optional Base Address Registers 0 and 1 define address ranges for the devices residing on secondary bus. All other addresses are assumed to reside on the primary bus. Inverse address decoding determines when to forward a transaction up-stream.

1.2.1.8 Secondary Bus Arbiter

Tsi350A is designed with an internal arbiter that can be enabled through input strap S_CFN_b. The arbiter provides bus arbitration for nine additional masters. The arbiter can be programmed to enable/disable and prioritize each request independently through software.

Tsi350A implements 2-level arbitration scheme. The requesters are divided into 2 groups, a high priority group and a low priority group. Each master can be assigned to either high or low priority group through the configuration register.

1.2.2 JTAG Controller

Tsi350A provides a JTAG test port that is compliant with IEEE Standard 1149.1, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, to facilitate card and board testing using boundary scan techniques. This function consists of five-signal test port interface with signals TCK, TDI, TDO, TMS, and TRST.

1.2.3 Hot Swap Interface

Tsi350A is designed with an interface for Hot Swap support. This allows the user to insert or extract the bridge card without powering down the system. During insertion and extraction process, the bridge indicates to system software about the Hot Swap event by driving HS_ENUM_b. It also provides a visual indication to the user through the HS_LED signal.

Hot-Swap support is implemented in the 208-pin PQFP package only. The 256-pin PBGA package does not support Hot Swap.

1.3 Architecture

Tsi350A internal architecture consists of the following major functions:

- PCI interface control logic for the primary and secondary PCI interfaces
- Data path and data path control logic
- Configuration register and configuration control logic
- Secondary bus arbiter

1.4 Data Path

The data path consists of a primary-to-secondary data path for transactions and data flowing in the downstream direction and a secondary-to-primary data path for transactions and data flowing in the upstream direction.

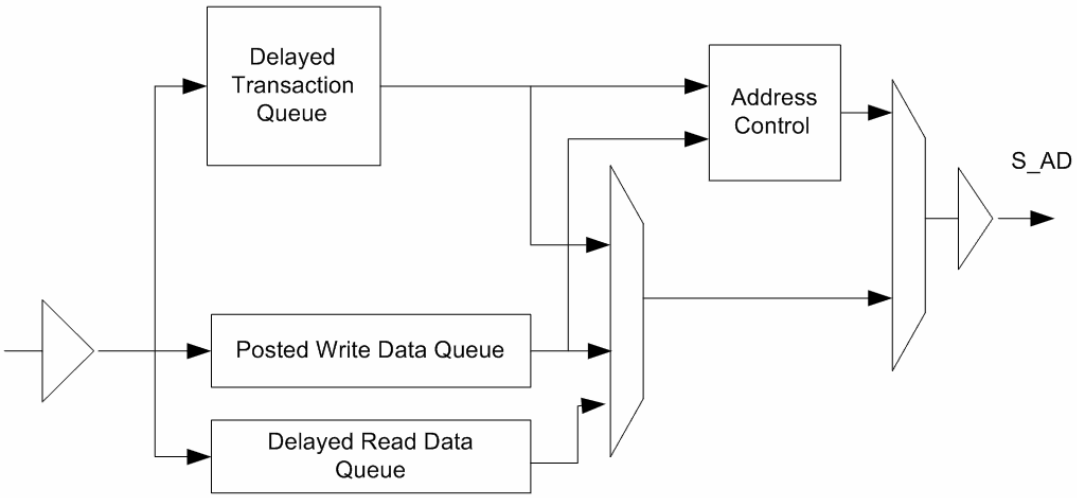
Both data paths have the following queues:

- Posted write queue
- Delayed transaction queue
- Read data queue

To prevent deadlocks and to maintain data coherency, a set of ordering rules is imposed on the forwarding of posted and delayed transactions across Tsi350A. The queue structure, along with the order in which the transactions in the queues are initiated and completed, supports these ordering requirements.

Figure 3 shows the Tsi350A data path for the downstream direction, and the following sections describe the data path queues.

Figure 3: Tsi350A Downstream Data Path



1.4.1 Posted Write Queue

The posted write queue contains the address and data of memory write transactions targeted for the opposite interface. The posted write transaction can consist of an arbitrary number of data phases, subject to the amount of space in the queue and disconnect boundaries. The posted write queue can contain multiple posted write transactions. The number of posted write transactions that can be queued at one time is dependent upon their burst size. The posted write queue consists of 128 bytes in each direction.

1.4.2 Delayed Transaction Queue

For a delayed write request transaction, the delayed transaction queue contains the address, bus command, 1 Dword of write data, byte enable bits, and parity. When the delayed write transaction is completed on the target bus, the write completion status is added to the corresponding entry. For a delayed read request transaction, the delayed transaction queue contains the address and bus command - and for non-prefetchable read transactions - the byte enable bits. When the delayed read transaction is completed on the target bus, the read completion status corresponding to that transaction is added to the delayed request entry. Read data is placed in the read data queue. The delayed transaction queue can hold up to four transactions (any combination of read and write transactions).

1.4.3 Read Data Queue

The read data queue contains read data transferred from the target during a delayed read completion. Read data travels in the opposite direction of the transaction. The primary-to secondary read data queue contains read data corresponding to a delayed read transaction residing in the secondary-to-primary delayed transaction queue. The secondary-to-primary read data queue contains read data corresponding to a delayed read transaction in the primary-to-secondary delayed transaction queue. The amount of read data per transaction depends on the amount of space in the queue and disconnect boundaries. Read data for up to four transactions, subject to the burst size of the read transactions and available queue space, can be stored. The read data queue for Tsi350A consists of 128 bytes in each direction.

2. PCI Interface

This chapter discusses the following topics:

2.1 Transaction Types

This section summarizes the PCI transactions performed by Tsi350A.

Table 1 lists the command code and name of each PCI transaction. The Master and Target columns indicate Tsi350A support for each transaction when Tsi350A initiates transactions as a master, on the primary bus and on the secondary bus, and when Tsi350A responds to transactions as a target, on the primary bus and on the secondary bus.

As indicated in Table 1, the following PCI commands are not supported by Tsi350A:

- Tsi350A never initiates a PCI transaction with a reserved command code and, as a target Tsi350A ignores reserved command codes.
- Tsi350A never initiates an interrupt acknowledge transaction and, as a target, Tsi350A ignores interrupt acknowledge transactions. Interrupt acknowledge transactions are expected to reside entirely on the primary PCI bus closest to the host bridge.
- Tsi350A does not respond to special cycle transactions. Tsi350A cannot guarantee delivery of a special cycle transaction to downstream buses because of the broadcast nature of the special cycle command and the inability to control the transaction as a target. To generate special cycle transactions on other PCI buses, either upstream or downstream, a Type 1 configuration command must be used.