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

IDT
Tsi340™ PCI-to-PCI Bridge
User Manual

80E3000_MA001_05

September 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.

Contents

Figures

Tables

About this Document

This section discusses the following topics:

Scope

The *Tsi340 PCI-to-PCI Bridge User Manual* discusses the features, capabilities, and configuration requirements for the Tsi340. 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	NAME_b[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

80E3000_MA001_05, Formal, September 2009

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

80E3000_MA001_02, Advance, October 2006

This version includes numerous minor changes.

80E3000_MA001_01, Draft, July 2006

This is the first version of the *Tsi384 User Manual*.

1. Functional Overview

This chapter discusses the following topics about the Tsi340:

1.1 Overview

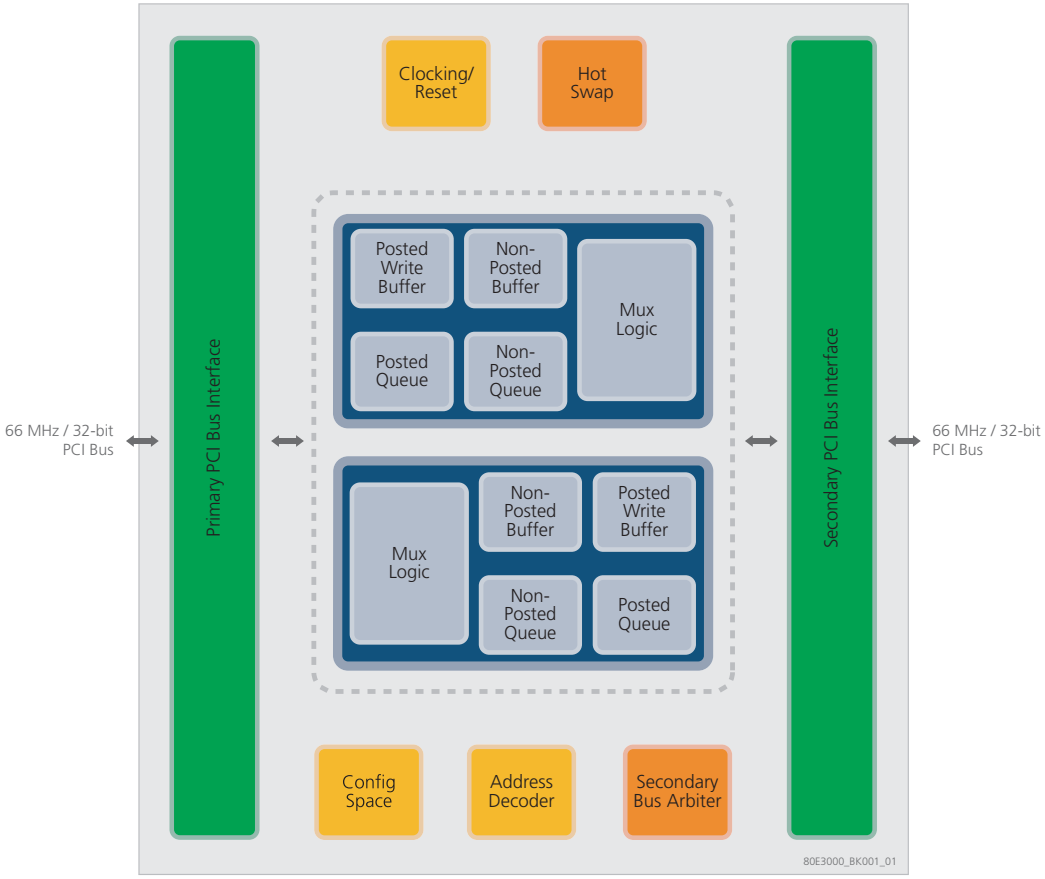
The IDT Tsi340 is a PCI-to-PCI bridge that is fully compliant with the *PCI Local Bus Specification, Revision 2.3*.

The Tsi340 has two identical PCI interfaces that support PCI transactions for each bus. The interfaces can act as either a bus master or a bus slave, depending on the type of transaction.

The Tsi340 enables 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 can increase system performance in applications such as multimedia.

The block diagram for Tsi340 is shown Figure 1 on page 16.

Figure 1: Tsi340 Block Diagram



Typical Applications

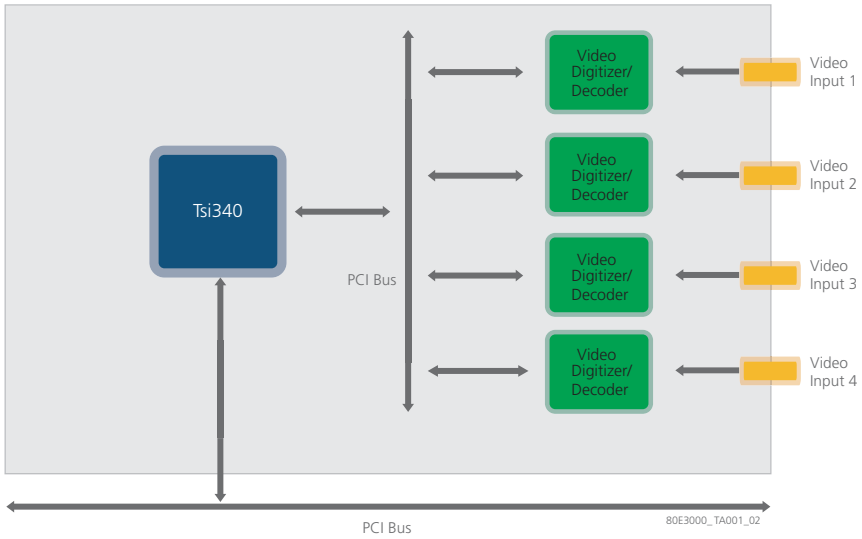
The Tsi340 is suited to applications that need to bridge from PCI to other downstream PCI devices. Its flexibility, and small footprint, make it ideal for a wide range of applications, including:

- Video capture cards
- Digital video recorders
- Industrial PC (IPC) backplanes
- Multi-function printers
- Storage host bus adapters (HBAs)
- Network interface cards (NICs)
- Firewall and security gateways
- Printers, graphics and imaging systems

Option card designers can use the Tsi340 to implement multiple-device PCI option cards. The *PCI Local Bus Specification* loading rules limit PCI option cards to a single connection per PCI signal in the option card connector. Without a PCI-to-PCI bridge, PCI loading rules limit option cards to one device.The Tsi340 overcomes this restriction by providing an independent PCI bus that can support up to four devices.

The application diagram below shows how the Tsi340 enables the design of a multi-component option card and expands the PCI architecture.

Figure 2: Application Diagram - Digital Video Recorder



1.2 Features

The following section describe the features of Tsi340:

- Industry-standard PCI-to-PCI bridge
- 66 MHz, 32-bit operation on the primary and secondary interfaces
- Up to four PCI bus masters supported on the secondary interface
- Concurrent operation of primary and secondary interfaces
- Compliant with the following specifications:
  - 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)
- Posted write buffers in both directions
- 1-KB transaction buffer (total)

- Enhanced address decoding
- Compatible with existing PCI bridging devices from PLX and Pericom
- Physical
  - 128-pin PQFP
  - RoHS compliant
  - 3.3 V I/O, 5 V tolerant
- 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)*

1.3 Functional Overview

Tsi340 has two PCI interfaces: a primary interface and a secondary interface. Each interface controls the PCI protocol for its respective bus. These interfaces transfer data/control information to and from the Buffer Logic Unit (BLU). The BLU consists of a posted write buffer, posted write queue, non-posted buffer, and non-posted queue.

1.3.1 Posted Write Buffer

The Tsi340 handles the conventional PCI transactions of Memory Write, and Memory Write and Invalidate as posted transactions.

The posted write buffer is used for temporary storage of data flowing from the primary interface to the secondary interface and from the secondary interface to the primary interface. Each posted buffer has a capacity of 256 bytes. The amount of space assigned to each transaction is dynamic. A single transaction can use sizes ranging from one memory location (4 bytes) to 64-memory location (64 bytes).

When the Tsi340 determines that a memory write transaction must be forwarded across the bridge, it first checks for empty space in the posted write buffer. If space is available, the posted write buffer accepts data until the buffer is full or the transaction is terminated. If there is no space in the posted write buffer, the transaction is terminated with retry.

1.3.2 Posted Write Queue

The posted write queue is used to store the control information related to the transaction flowing from the primary interface to secondary interface or from the secondary interface to the primary interface. Each posted write queue has a four entry FIFO, which provides four active posted write transactions in each direction. Data related to each entry is stored in the posted write buffer.

The posted write queue accepts an entry from an external master as long as at least one entry is free and at least one Dword of space is available in the posted write buffer.

1.3.3 Non-Posted Buffer

The non-posted buffer is used for storing the data related to delayed transactions. The following list of transactions use the non-posted buffer:

- Memory Read
- Memory Read Line
- Memory Read Multiple
- I/O Read
- I/O Write
- Type-1 Configuration Read
- Type-1 Configuration Write

All the non-posted transactions are processed through the non-posted queues and non-posted buffers. Each non-posted buffer has a storage capacity of up to 256 bytes for storing data related to delayed transactions.

1.3.4 Non-Posted Queue

The non-posted queue is used to store the control information related to the all non-posted transactions. Each non-posted queue has a four entry FIFO, which provides four active non-posted transactions in each direction. Data related to each entry is stored in the non-posted buffer.

The non-posted queue accepts an entry from an external master if at least one entry is available. If all four entries are full the Tsi340 retries the external master until an entry becomes available.

1.3.5 Configuration Space

Tsi340 is a PCI-to-PCI bridge, and complies with the *PCI to PCI Bridge Architecture Specification, Revision 1.1*. The Tsi340's configuration space can only be accessed from the primary interface. The Tsi340 uses additional device specific configuration registers to support optional, device specific features.

Refer to "Configuration Transactions" on page 33 for more information.

1.3.6 Address Decoding Logic

The Tsi340 is a transparent bridge. In transparent mode, the I/O, Memory, pre-fetchable memory base and limit, and optional base address registers 0 and 1 define address ranges residing on the secondary bus. All other addresses are assumed to reside on the primary bus. Inverse address decoding determines when to forward the transaction up-stream.

Refer to "Address Decoding" on page 45 for more information.

1.3.7 Secondary Bus Arbiter

The Tsi340 has an internal secondary bus arbiter. It provides bus arbitration for up to four additional masters. Each external master is assigned to either high or low priority, or may be masked off.

The internal arbiter provides a two level arbitration scheme in which arbitration is divided into the following two groups: a high priority group and low priority group. Each master can be assigned to either high priority group or low priority group through the configuration register.

Refer to "PCI Bus Arbitration" on page 59 for more information.

1.3.8 Hot Swap Interface

Tsi340 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 through the HS_LED_OUT signal.

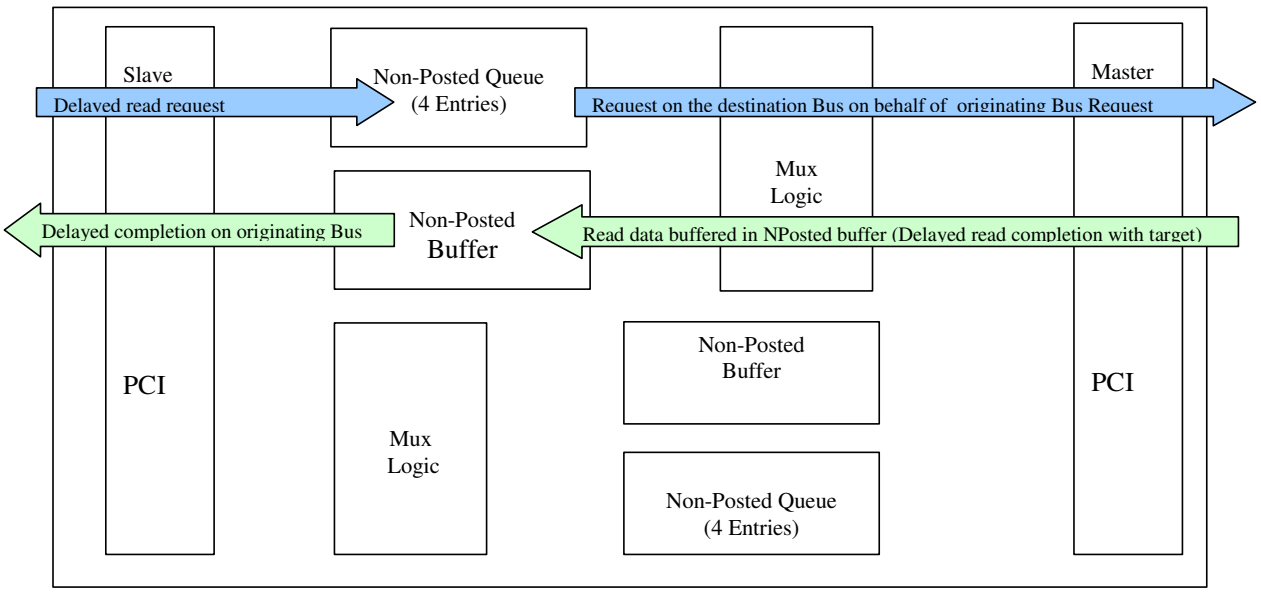
1.4 Data Flow

The following sections describe the data flow through the Tsi340 device.

1.4.1 Memory Read Transactions

The conventional PCI memory read, memory read line, and memory read multiple commands are used to transfer memory read data. Tsi340 completes all memory read transactions as delayed transactions. Figure 3 shows the Tsi340 memory read flow.

Figure 3: Memory Read Flow



The following steps detail the memory read flow through the Tsi340:

- The read request from the initiator is posted/entered into the Non-Posted Queue,
- The transaction is terminated by signaling target retry to the initiator
- When the target retry is received, the initiator is required to continue to repeat the same read transaction until at least one data transfer is completed or until a master/target abort is received
- The Tsi340 then arbitrates for the destination bus and initiates a read transaction using the exact read address and read command
  - If Tsi340 receives retry on the target bus, it continues to repeat the read transaction until at least one data transfer is completed, or until an error condition is encountered.

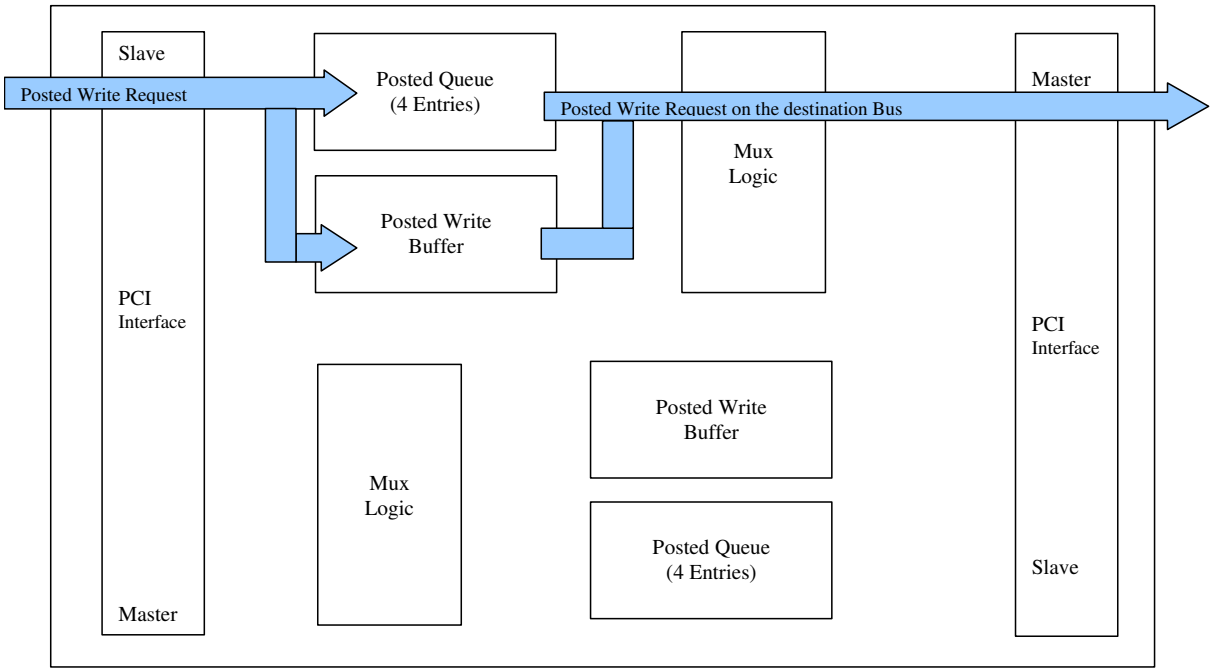
When a memory read transaction targets non-prefetchable address space the Tsi340 will pre-fetch a single DWORD of data when the memory read command is used. For all other read transactions the Tsi340 will pre-fetch data according to the pre-fetch algorithm, see “Read Transactions” on page 29.

- When the transaction is completed on the target bus, the Tsi340 transfers the data to the initiator when the initiator repeats the transaction

1.4.2 Posted Write Transaction Flow

The conventional PCI memory write and memory write and invalidate are posted transactions. Unlike non-posted transactions these transactions are first completed on the originating bus and then completed on the destination bus. Figure 4 shows the Tsi340 posted write flow.

Figure 4: Memory Write Flow



The following steps detail the posted write flow through the Tsi340:

- When Tsi340 determines that a memory write transaction is to be forwarded across the bridge, it first checks for empty space in the posted write buffer
  - If space is available, Tsi340 accepts data until the buffer is full or the transaction is terminated.
  - If there is no space in the posted write buffer, the transaction is terminated with retry.
- After buffering data into the posted buffer the Tsi340 arbitrates for the destination bus and writes the data to the destination.

2. PCI Interface

This chapter discusses the following topics about the Tsi340:

2.1 Overview

Tsi340 has two PCI interfaces: a primary interface and a secondary interface. Each interface controls the PCI protocol for its respective bus. These interfaces transfer data/control information to and from the Buffer Logic Unit (BLU). The BLU consists of a posted write buffer, posted write queue, non-posted buffer, and non-posted queue.

The following sections describe the how the Tsi340 handles PCI transactions, transaction forwarding across Tsi340, and transaction termination.

2.2 Transaction Types

This section provides a summary of PCI transactions performed by Tsi340. Table 1 lists the command code and name of each PCI transaction. The Master and Target columns indicate Tsi340 support for each transaction when Tsi340 initiates transactions as a master, on the primary bus and on the secondary bus, and when Tsi340 responds to transactions as a target, on the primary bus and on the secondary bus.

Table 1: Type of Transactions

Type of Transaction	Initiates as a Master		Responds as a Target	
	Primary	Secondary	Primary	Secondary
0000-Interrupt Acknowledge	No	No	No	No
0001-Special Cycle	Yes	Yes	No	No
0010-I/O Read	Yes	Yes	Yes	Yes
0011-I/O Write	Yes	Yes	Yes	Yes
0100-Reserved	No	No	No	No

Table 1: Type of Transactions

Type of Transaction	Initiates as a Master		Responds as a Target	
	Primary	Secondary	Primary	Secondary
0101-Reserved	No	No	No	No
0110-Memory Read	Yes	Yes	Yes	Yes
0111-Memory Write	Yes	Yes	Yes	Yes
1000-Reserved	No	No	No	No
1001-Reserved	No	No	No	No
1010-Configuration read	No	Yes	Yes	No
1011-Configuration Write	Type-1	Yes	Yes	Type-1
1100-Memory Read Multiple	Yes	Yes	Yes	Yes
1101-Dual Address Cycle	Yes	Yes	Yes	Yes
1110-Memory Read Line	Yes	Yes	Yes	Yes
1111-Memory Write and Invalidate	Yes	Yes	Yes	Yes

2.2.1 Transaction Types Not Supported

As indicated in Table 1 the following PCI commands are not supported by Tsi340:

- Tsi340 never initiates a PCI transaction with a reserved command code and, as a target, Tsi340 ignores reserved command codes.
- Tsi340 never initiates an interrupt acknowledge transaction and, as a target, Tsi340 ignores interrupt acknowledge transactions. Interrupt acknowledge transactions are expected to reside entirely on the primary PCI bus closest to the host bridge.
- Tsi340 does not respond to special cycle transactions. Tsi340 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.
- Tsi340 does not generate Type 0 configuration transactions on the primary interface, nor does it respond to Type 0 configuration transactions on the secondary PCI interface. The *PCI-to-PCI Bridge Architecture Specification* does not support configuration from the secondary bus.

2.2.2 Address Phase

A standard PCI transaction consists of one or two address phases, followed by one or more data phases. The first address phase is designated by an asserting (falling) edge on the FRAME# signal. The number of address phases depends on whether the address is 32 bits or 64 bits.

2.2.2.1 Single Address Phase

A 32-bit address uses a single address phase. This address is driven on AD[31:0], and the bus command is driven on C/BE#[3:0]. Tsi340 supports the linear increment address mode only for decoding memory address space, which is indicated when the lower two address bits are equal to 0. If either of the lower two address bits is nonzero, Tsi340 automatically disconnects the transaction after the first data transfer.

2.2.2.2 Dual Address Phase

Dual address transactions are PCI transactions that contain two address phases specifying a 64-bit address. The first address phase is denoted by the asserting edge of FRAME#. The second address phase always follows on the next clock cycle.

For a 32-bit interface, the first address phase contains the dual address command code on the C/BE#<3:0> lines, and the low 32 address bits on the AD<31:0> lines. The second address phase consists of the specific memory transaction command code on the C/BE#<3:0> lines and the high 32 address bits on the AD<31:0>lines. In this way, 64-bit addressing can be supported on 32-bit PCI buses.

The *PCI-to-PCI Bridge Architecture Specification* supports the use of dual address transactions in the prefetchable memory range only. Tsi340 supports dual address transactions in both the upstream and the downstream direction. Tsi340 supports a programmable 64-bit address range in prefetchable memory for downstream forwarding of dual address transactions. Dual address transactions falling outside the prefetchable address range are forwarded upstream, but not downstream. Prefetching and posting are performed in a manner consistent with the guidelines given in this specification for each type of memory transaction in prefetchable memory space.

Tsi340 responds only to dual address transactions that use the following transaction command codes:

- Memory Write
- Memory Write and Invalidate
- Memory Read
- Memory Read Line
- Memory Read Multiple

Use of other transaction codes may result in a master abort.

Any memory transactions addressing the first 4GB space should use a single address phase; that is, the high 32 bits of a dual address transaction should never be 0.

2.2.3 Device Select (DEVSEL#) Generation

Tsi340 always performs positive address decoding when accepting transactions on either the primary or secondary buses. Tsi340 never subtractively decodes. Medium DEVSEL# timing is used on both interfaces.