Supporting Subtractive PCI-to-PCI Bridges in Windows

WinHEC 2004 Version - April 27, 2004

Abstract

This paper provides information about how the Microsoft® Windows® family of operating systems handle subtractive PCI-to-PCI bridges. It provides guidelines for device manufacturers and driver developers to understand how Windows supports PCI-to-PCI bridges.

The content of this paper assumes that reader is familiar with the basics of PCI-to-PCI bridges, including familiarity with the *PCI-to-PCI Bridge Architecture Specification, revision 1.2*, referred to in this paper as the "PCI-to-PCI bridge specification."

This information in this paper applies for the following operating systems:

Microsoft Windows 2000
Microsoft Windows XP
Microsoft Windows Server™ 2003
Microsoft Windows Vista™

The current version of this paper is maintained on the Web at: http://www.microsoft.com/whdc/system/bus/pci/default.mspx.

Contents

Disclaimer

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Introduction

A bridge that is fully compliant with the PCI-to-PCI bridge specification can support both subtractive and positive decode types. However, Microsoft® Windows® 2000, Microsoft Windows XP, and Microsoft Windows Server™ 2003 do not fully support these bridge types. Current plans are to include support for these subtractive bridges in Microsoft Windows Vista™.

How PCI-to-PCI Bridges Work

Similar to routing networking packets across a network, PCI transactions and requests are forwarded across a PCI bridge. To understand how a particular type of bridge routes requests, you must first understand resource arbitration and the address decoding method associated with a particular type of bridge. You must also understand decode windows and address decoding. These concepts are described in the following sections.

Resource Arbitration and Address Decoding

Windows assigns resources to PCI devices in the system so that all PCI devices function properly and do not attempt to use resources that are assigned to another device. This process is referred to as resource arbitration. In order to provide proper arbitration of resources, the operating system must comprehend not only the resources a PCI device claims to be using, but it must also know the method that a PCI device uses to claim a transaction that is directed at one of its resources. The process by which a device claims transactions is referred to as address decoding.

Decode Windows

In the case of a PCI-to-PCI bridge, the resources used by the bridge include ranges of addresses. The address ranges are specified as decode windows that the bridge uses to forward transactions from one side of the bridge to the other. PCI-to-PCI bridges use decode windows of two types: decode windows used to access I/O locations of devices and decode windows used to access memory locations of devices.

Address Decoding

There are three types of address decoding that a bridge can use to claim transactions for forwarding. These are described in the following list and shown in Figure 1:

- **Subtractive decode:** Address decoding in which a bridge accepts all access requests that are not positively decoded by another device (after a four-cycle timeout).
- **Positive decode:** Address decoding in which a bridge forwards requests from the primary side to the secondary side of the bridge within a window assigned by the firmware or by the operating system.
- **Negative decode:** Address decoding in which a bridge forwards access requests that are outside the bridge's assigned positive decode window from the secondary side to the primary side of the bridge. This type of decoding occurs when a device on the secondary side of the bridge attempts to access a device on the primary side of the bridge.

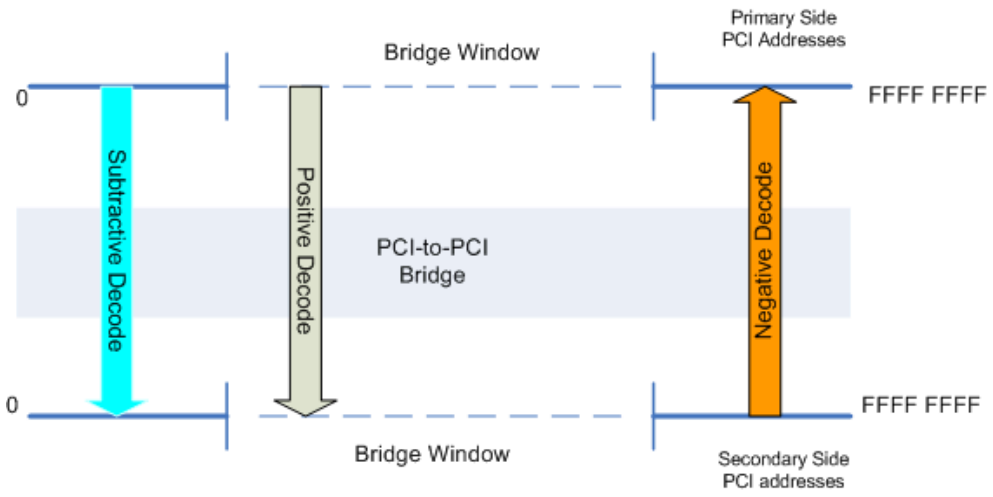

Figure 1 Address decoding types

Note that the dotted lines in the graphic represent the bridge's decode windows. For bridges that use decode windows, the firmware or operating system defines the address range for the decode windows. The same address range is used for both the primary and secondary sides of the bridge.

Rules for Forwarding Transactions Across a PCI-to-PCI Bridge

This section describes the rules outlined by the PCI-to-PCI bridge specification that bridges follow in order to forward transactions. Note that there are separate rules depending on whether a transaction originates from the primary or secondary side of the bridge. Additionally, there are also separate rules if the transaction falls within the address range specified by the bridge's decode windows.

Primary Side Bridge Rules

The primary side bridge rules are described in the following list and shown in Figure 2.

If a transaction falls outside the address ranges specified by the bridge windows, then the following rules apply:

- As shown by "1a" in Figure 2, if the bridge supports subtractive decode, it claims and forwards the transaction from the primary side to the secondary side after four bus cycles only if the transaction is not claimed by another device on the primary side.
- As shown by "1b" in Figure 2, if subtractive decode is not supported, then the bridge does not claim or forward the transaction. An unclaimed transaction is an error on the PCI bus, and this error is detected by the system chip set that uses an interrupt to alert the operating system of the error.
- As shown by "2" in Figure 2, if a transaction falls within the bridge window, the bridge uses positive decode to immediately claim the transaction and then forwards the transaction to the secondary side.

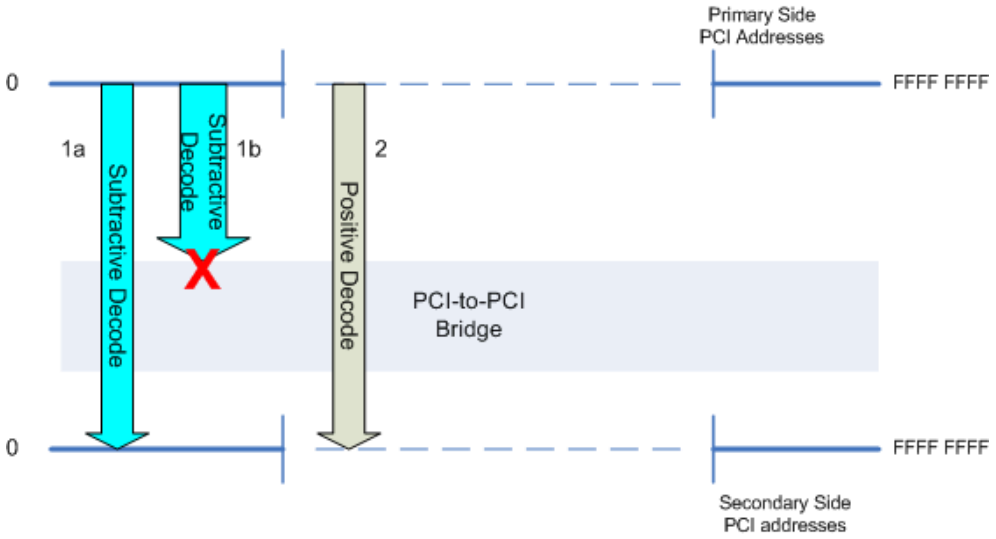

Figure 2 Primary side bridge rules

Secondary Side Bridge Rules

The secondary side bridge rules are described in the following list and shown in Figure 3:

- As shown by “1” in Figure 3, if a transaction falls within the bridge window, the bridge does not claim the transaction. This allows a peer device on the secondary side to handle the transaction.
- As shown by “2” in Figure 3, if a transaction falls outside the bridge window, the bridge forwards the transaction to the primary side.

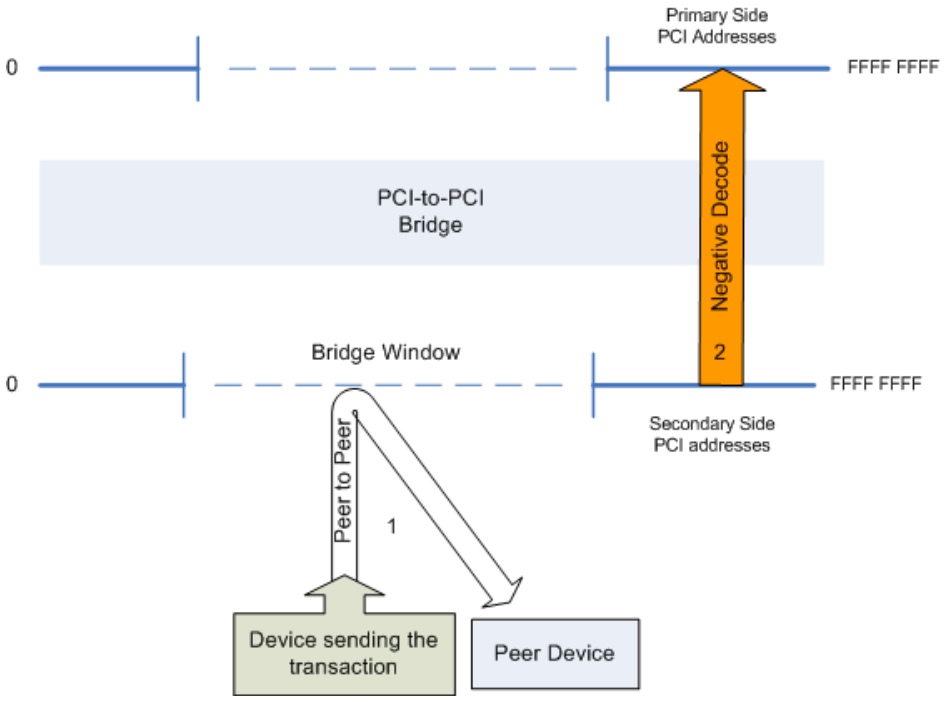

Figure 3 Secondary side bridge rules

Types of PCI-to-PCI Bridges

There are three types of PCI-to-PCI bridges that were commonly implemented on systems running Windows. These bridge types are:

- Subtractive-only PCI-to-PCI bridge
- Positive-only PCI-to-PCI bridge
- Subtractive PCI-to-PCI bridge

Subtractive-only PCI-to-PCI bridge: This bridge type is a PCI-to-PCI bridge with no bridge windows. It can only address resources that are on the secondary side of the PCI bridge by using subtractive decode. A subtractive-only PCI-to-PCI bridge does not support transactions that both originate and terminate on the secondary PCI bus. Because this bridge type does not support these transactions, peer-to-peer transactions are not supported.

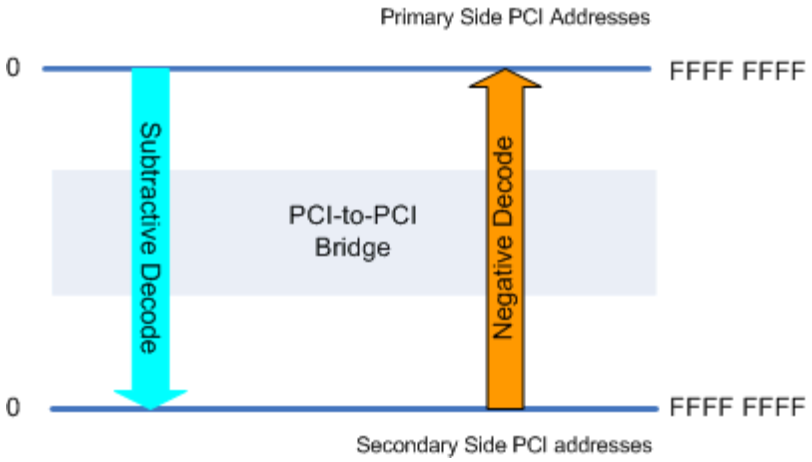

Figure 4 Subtractive-only PCI-to-PCI bridge

This bridge type is not defined in the PCI-to-PCI bridge specification. However, bridges with this behavior have determined the assumptions that led to the design and behavior of previous versions of Windows.

Positive-only PCI-to-PCI bridge: This bridge type can perform only positive decodes for transactions that originate from the primary side of the bridge. Therefore, subtractive decode is not supported, as shown in the Figure 5. Because this bridge type has a bridge window, it can perform peer-to-peer transactions. Negative decode is supported for transactions that originate from the secondary side of the bridge and fall outside the bridge windows.

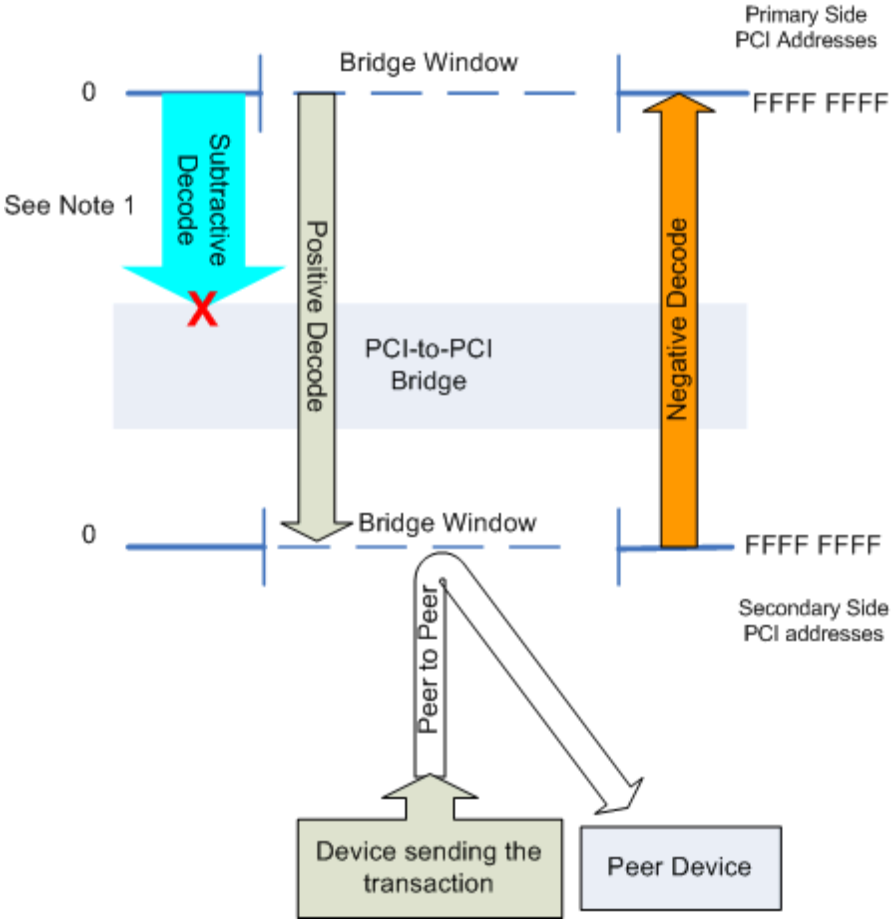

Figure 5 Positive-only PCI-to-PCI bridge

Note: A positive-only PCI-to-PCI bridge cannot address devices on the secondary side of the bridge if they require legacy resources because this bridge type does not support subtractive decodes.

Subtractive PCI-to-PCI bridge: This bridge type is a PCI-to-PCI bridge that has bridge windows and that supports both positive and subtractive decode, as shown in Figure 6. Transactions that originate on the primary side of the bridge and are within the bridge windows are decoded positively. Likewise, transactions that originate on the primary side of the bridge and are outside the bridge windows are decoded subtractively. Because this bridge type has a positive decode window, it can perform peer-to-peer transactions on the secondary side of the bridge that fall within this window. Negative decode is supported for transactions that originate from the secondary side and fall outside the bridge windows.

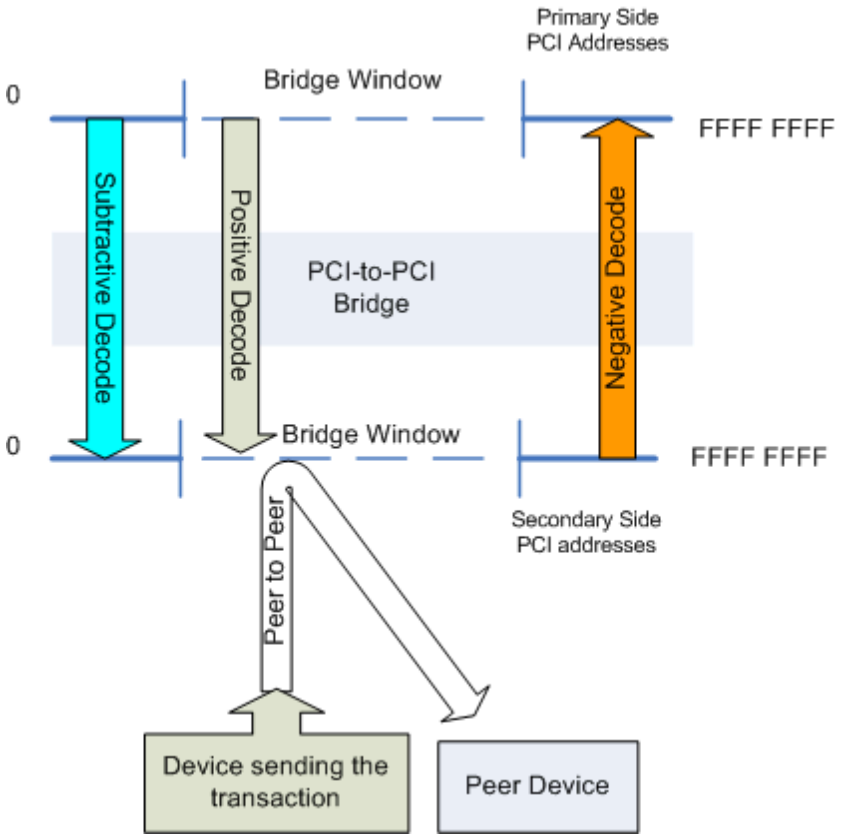

Figure 6 Subtractive PCI-to-PCI bridge

Because this bridge type supports subtractive decodes, it supports legacy devices that are on the secondary side of the bridge.

Though defined in the PCI-to-PCI bridge specification, these bridges are not fully supported on Windows 2000, Windows XP, and Windows Server 2003 operating systems.

Identifying and Supporting Bridge Types in Windows

Current versions of Windows do not support subtractive bridges that comply with the PCI-to-PCI bridge specification. Instead, subtractive bridges are supported as positive-only or subtractive-only bridges depending on the criteria listed in the following section. This behavior prevents current versions of Windows from supporting both legacy devices and peer-to-peer transactions for subtractive bridges.

When the current versions of Windows were developed, only the subtractive-only and positive-only bridge types were supported.

Because of this, current versions of Windows close the bridge windows in bridges that identify themselves as subtractive, which causes these bridges to be supported as subtractive-only bridges.

Because subtractive bridges are forced into subtractive-only mode, peer-to-peer transactions cannot be supported on the secondary side of the bridge. PCI devices that reside on the secondary side of subtractive bridges that require peer-to-peer transactions do not function properly in Windows.

Table 1 summarizes how Windows identifies and supports the three bridge types.

Table 1. How Windows Identifies and Supports Bridge Types

Bridge is identified as	Only if
Subtractive-only	Current versions of Windows identify a bridge as subtractive-only if **one** of the following is true: • The bridge's Programming Interface Register in the PCI Configuration Space is set to 01h. • The I/O Limit register is not implemented or is read-only.
Positive-only	Current versions of Windows identify a bridge as positive-only if **both** of the following are true: • The bridge's Programming Interface Register in the PCI Configuration Space is set to 00h. • The IO Limit register is sticky (retains the value written to it).
Subtractive	Current versions of Windows never identify a bridge as subtractive. All bridges are supported either as subtractive-only or positive-only bridges.

The following support for subtractive bridges and for peer-to-peer transactions on the secondary sides of these bridges is currently planned for Windows Vista:

- Windows Vista will not close the positive decode windows on subtractive bridges. (Current versions of Windows close this window.)
- Windows Vista will use the Programming Interface bit to identify subtractive bridges that comply with the PCI-to-PCI bridge specification to provide full support for all three of the bridge types.

Note: A PCI-to-PCI bridge will be supported as a subtractive-only bridge only if Windows Vista determines that the bridge doesn't correctly operate as a subtractive bridge as described in the PCI-to-PCI bridge specification.

Table 2 summarizes how Windows Vista identifies and supports the three bridge types.

Table 2. How Windows Vista Identifies and Supports Bridge Types

Bridge is identified as	Only if
Subtractive-only	Windows Vista identifies and supports a bridge as subtractive-only if **one** of the following is true: • The I/O Limit register is not implemented. • The I/O Limit register is read-only.
Positive-only	Windows Vista identifies and supports a bridge as positive-only if **both** of the following are true: • The bridge complies with the PCI-to-PCI bridge specification. • The bridge's Programming Interface Register in the PCI Configuration Space is set to 00h.
Subtractive	Windows Vista identifies and supports a bridge as subtractive if both of the following are true: • The bridge complies with the PCI-to-PCI bridge specification. • The bridge's Programming Interface Register in the PCI Configuration Space is set to 01h.

Call to Action and Resources

Call to Action

For system manufacturers:

- Windows 2000, Windows XP, or Windows Server 2003: Do not use PCI devices that perform peer-to-peer transactions on the secondary side of subtractive bridges.
- Windows Vista: Use PCI devices that perform peer-to-peer transactions behind subtractive bridges.

For device manufacturers:

- Implement PCI-to-PCI bridges that are compliant with PCI-to-PCI bridge specification.
- Windows 2000, Windows XP, and Windows Server 2003 compatibility: Do not design PCI devices that require peer-to-peer transactions on the secondary side of subtractive bridges.
- Windows Vista: Set the PCI-to-PCI bridge's Programming Interface Register in the PCI Configuration Space to 01h to allow Windows Vista to support it as a subtractive bridge.

Feedback

For questions, send mail to: **pciesup@microsoft.com**

Resources

- PCI-to-PCI Bridge Architecture specification, Revision 1.2
  http://www.pcisig.com/
- Windows Platform Development white papers and resources:
  http://www.microsoft.com/whdc/default.mspx
- Windows Driver Development Kit:
  http://www.microsoft.com/whdc/ddk/default.mspx
- Windows Logo Program for Hardware:
  http://www.microsoft.com/whdc/winlogo/default.mspx