



Host Bus Adapter Considerations with the Intel[®] 80314 I/O Processor Companion Chip

Application Note

February 2004

Order Number: 274048-002





INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® 80314 I/O Processor Companion Chip may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

AnyPoint, AppChoice, BoardWatch, BunnyPeople, CablePort, Celeron, Chips, CT Media, Dialogic, DM3, EtherExpress, ETOX, FlashFile, i386, i486, i960, iCOMP, InstantIP, Intel, Intel Centrino, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Create & Share, Intel GigaBlade, Intel InBusiness, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel Play, Intel Play logo, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel TeamStation, Intel Xeon, Intel XScale, IPLink, Itanium, MCS, MMX, MMX logo, Optimizer logo, OverDrive, Paragon, PC Dads, PC Parents, PDCharm, Pentium, Pentium II Xeon, Pentium III Xeon, Pentium 4, Performance at Your Command, RemoteExpress, SmartDie, Solutions960, Sound Mark, StorageExpress, The Computer Inside, The Journey Inside, TokenExpress, VoiceBrick, VTune, and Xircom are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2004, Intel Corporation

Contents

- 1.0 Introduction 4
- 2.0 Host Messaging with Doorbells and Mailboxes 4
 - 2.1 Doorbells 5
 - 2.2 Mailboxes 5
 - 2.3 Doorbell Messaging Register plus IRP_PIA (Recommended Method) 6

Tables

- 1 Host Messaging Mechanisms 4
- 2 PCI Interrupt Assertion Register 6

Revision History

Date	Revision	Description
January 2004	001	Initial release
February 2004	002	Removed Section 3.0.

1.0 Introduction

When using the Intel® 80314 I/O Processor Companion Chip (hereafter “80314”) on a host bus adaptor, the designer must decide how to make best use of the capabilities of the device for functions such as host communication and error recovery. This document presents relevant design considerations, such as the use of the doorbell and mailbox registers.

2.0 Host Messaging with Doorbells and Mailboxes

The 80314 Multi-Processor Interrupt Controller (MPIC) block contains both a doorbell and a mailbox mechanism to facilitate host messaging. These mechanisms provide a means for the adaptor card to make an interrupt request to the host and to pass messages. The key difference between doorbells and mailboxes is how each triggers the interrupt. For mailboxes, a write to the message register triggers the interrupt. For doorbells, however, the message register is separate from the register that triggers the interrupt.

The interrupts normally associated with doorbells and mailboxes are one of the four MPIC output interrupts. These interrupts are designed primarily for raising interrupts to the Intel XScale® processors. As a result, using one of these interrupts as a host interrupt (PCI INT) requires dedicating an FIQ/IRQ signal normally used for processor interrupts. In addition, de-assertion of the interrupt triggered by these mechanisms is a two-step process involving a read of the interrupt vector followed by an end-of-interrupt (EOI) write.

An alternate method for asserting a PCI INT using the PCI Interrupt Assertion Register (IRP_PIA) at offset 0x190 is described in [Section 2.3](#). This method is the recommended host messaging / interrupt method; it involves using the doorbell register to pass messages and the IRP_PIA to assert/de-assert the host PCI interrupt.

[Table 1](#) summarizes the methods available for host messaging and identifies recommended solutions.

Table 1. Host Messaging Mechanisms

Method	Notes
Doorbell Messaging Register plus IRP_PIA (Recommended Method)	Best for 80314-to-host communications due to the efficiency of raising PCI interrupts.
Doorbells	Best for host-to-80314 communications in which only an INT is needed (not a message) or where only a message is needed (not an INT). This method is not optimal for hosting communications with the 80314 because of how the interrupt works.
Mailboxes	Best for host-to-80314 communications in which both a message and an INT must be passed. This method is not optimal for hosting communications with the 80314 because of how the interrupt works.

The remainder of this section discusses the three host messaging / interrupt methods—doorbell ([Section 2.1](#)), mailbox ([Section 2.2](#)), and doorbell+IRP-PIA (recommended) ([Section 2.3](#)).



2.1 Doorbells

When the host needs to generate an interrupt for the companion chip, it can use one of the four doorbell (or mailbox) registers provided by the MPIC. Using the doorbells requires some initial setup and configuration, described as follows:

1. **Configure the destination:** The doorbells can be configured to assert any of the four MPIC outputs by means of the Doorbell Destination Registers at offsets 0x208, 0x214, 0x220, and 0x22c. The lower four bits of this register constitute a bitmap that indicates which output(s) are fired when the doorbell is activated, as shown in this table:

Bit 3	Bit 2	Bit 1	Bit 0
INT3/IRQ1	INT2/IRQ0	INT1/FIQ1	INT0/FIQ0

2. **Configure the Vector, Priority and Enable:** The Doorbell Vector/Priority Registers at offsets 0x204, 0x210, 0x21c, and 0x228 must be configured both to enable the doorbell and to define the priority the generated INT has when asserted. In addition, the VECTOR field must be programmed in order for the MPIC to provide this value when the core processor acknowledges the interrupt by means of a read of the CPU Interrupt Acknowledge Register.
3. When the doorbells are configured and enabled, the host simply writes to the Doorbell Activate Register at offset 0x200 to assert the interrupt. Optionally, the host can use any or all of the Doorbell Messaging Registers at offsets 0x20c, 0x218, 0x224, and 0x230 to pass 32-bit messages back and forth for context, status, commands, and so on.
4. The interrupt de-asserts by means of the firmware reading the appropriate CPU Interrupt Acknowledge Register and performing a write cycle to the appropriate CPU End of Interrupt Register.

2.2 Mailboxes

When the host needs to generate an interrupt for the companion chip, it can use one of the four mailbox (or doorbell) registers provided by the MPIC. Using the mailbox requires some initial setup and configuration, described as follows:

Note: The key difference between the mailboxes and doorbell mechanisms is how the interrupt is generated, either by means of the activate bit for the doorbell or by writing a message for the mailboxes.

1. **Configure the destination:** The mailboxes can be configured to assert any of the four MPIC outputs by using the Mailbox Destination Registers at offsets 0x288, 0x294, 0x2A8, and 0x2B8. The lower four bits of the register constitute a bitmap that indicates which output(s) are fired when the mailbox message register is written, as shown in this table:

Bit 3	Bit 2	Bit 1	Bit 0
INT3/IRQ1	INT2/IRQ0	INT1/FIQ1	INT0/FIQ0

2. **Configure the Vector, Priority and Enable:** The Mailbox Vector/Priority Registers at offsets 0x284, 0x294, 0x2a4, and 0x2b4 must be configured both to enable the mailbox and to define the priority the generated INT has when asserted. In addition, the VECTOR field must be programmed in order for the MPIC to provide this value when the core processor acknowledges the interrupt by means of a read of the CPU Interrupt Acknowledge Register. Note that the mailbox message registers do not function if the interrupt is masked in this register (unlike the doorbell messaging register).
3. Once the mailboxes are configured and enabled, the host simply writes a 32-bit message to the Mailbox Registers at offsets 0x280, 0x290, 0x2a0, and 0x2b0 in order both to post the message and to assert the interrupt.
4. The interrupt de-asserts by means of the firmware reading the appropriate CPU Interrupt Acknowledge Register and performing a write cycle to the appropriate CPU End of Interrupt Register.

2.3 Doorbell Messaging Register plus IRP_PIA (Recommended Method)

This method utilizes the PCI Interrupt Assertion Register (IRP_PIA) at offset 0x190 (Table 2) and requires some additional configuration of the existing Interrupt Control Register at offset 0x190. This register, like all other registers within the device, is accessible from firmware as well as by means of the host through PFAB_BAR0. The doorbell messaging register (discussed in Section 2.1) can be used in conjunction with this method to comprise a complete interrupt and messaging facility.

Messages are passed by means of the Doorbell Messaging Registers at offsets 0x20c, 0x218, 0x224, and 0x230 to pass 32-bit messages back and forth as described in Section 2.1. Do not configure the MPIC for doorbell-triggered interrupts. For the doorbell+IRP_PIA method, interrupts are asserted or de-asserted by writing bit[28] of the IRP_PIA register. Writing a 1 asserts PCI interrupt INTA, and clearing the bit de-asserts the interrupt. Table 2 describes the setup conditions required for this method to operate correctly.

Table 2. PCI Interrupt Assertion Register

PCI Interrupt Assertion Register Offset: 0x190		
Bit	Default	Description
31:29	–	Reserved
28	0	Writing a 1 asserts PCI interrupt INTA. Clearing the bit de-asserts the interrupt. Note the following requirements for this bit to function: <ul style="list-style-type: none"> • The PCI block must be configured with RST_DIR = 0 • IRP_CFG_CTL (0x180) INTA_TYPE field must be 01b • IRP_CFG_CTL (0x180) LOC_INT_DEST must be 01b • IRP_CFG_CTL (0x180) INTA_DIR must be 1b
27:0	–	Reserved