QSpan™
(CA91C860B, CA91L860B)

PCI to Motorola Processor Bridge Manual

http://www.tundra.com

The information in this document is subject to change without notice and should not be construed as a commitment by Tundra Semiconductor Corporation. While reasonable precautions have been taken, Tundra Semiconductor Corporation assumes no responsibility for any errors that may appear in this document.

Tundra® products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Tundra product could create a situation where personal injury or death may occur. Should Buyer purchase or use Tundra products for any such unintended or unauthorized application, Buyer shall indemnify and hold Tundra and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Tundra was negligent regarding the design or manufacture of the part.

The acceptance of this document will be construed as an acceptance of the foregoing conditions.

QSpan™ PCI to Motorola Processor Bridge Manual

Document: 8091860.MD302.05

Printed in Canada

Overview

Table of Contents

For the latest QSpan documentation, including: application notes, user manual, manual addenda, reference design manual, etc., please visit our web site: **http://www.tundra.com**

List of Figures

List of Tables

Corporate Profile

Tundra Semiconductor Corporation is an international, fabless semiconductor company focused on the design, development and support of bus-bridging semiconductors for embedded datacommunications and telecommunications applications.

Our mission is to provide high-quality components that enable embedded systems designers to get to market faster with a more competitive product. Tundra takes a ‘total product solution’ approach to its bus-bridges. This approach includes the QSpan Reference Design Kit (RDK) and the QSpan Software Development Kit (SDK). In addition, customers are supported by our acclaimed applications engineering support.

Tundra offers a growing line of PCI Bus bus-bridging chips that support embedded processors, such as the QSpan, the Motorola© embedded processor-to-PCI bridge. Our components are developed through strategic partnerships with leading processor manufacturers such as Motorola.

We also have an industry-leading family of VMEbus bus-bridging components, including the Universe II, the first commercially available VME-to-PCI Bus bridge, and the SCV64, the pioneering VME64 interface device.

Tundra develops devices that bridge leading bus types to industry-leading processors in the fast-growing embedded systems market. We accomplish this in two ways. First, we form strategic relationships with leading manufacturers. Next, we utilize a Lead Customer model, in which we invite innovators in embedded systems design to help further define the functionality of a Tundra component.

Tundra Semiconductor Corporation

603 March Road, Kanata, Ontario, Canada K2K 2M5 TEL (613) 592-0714

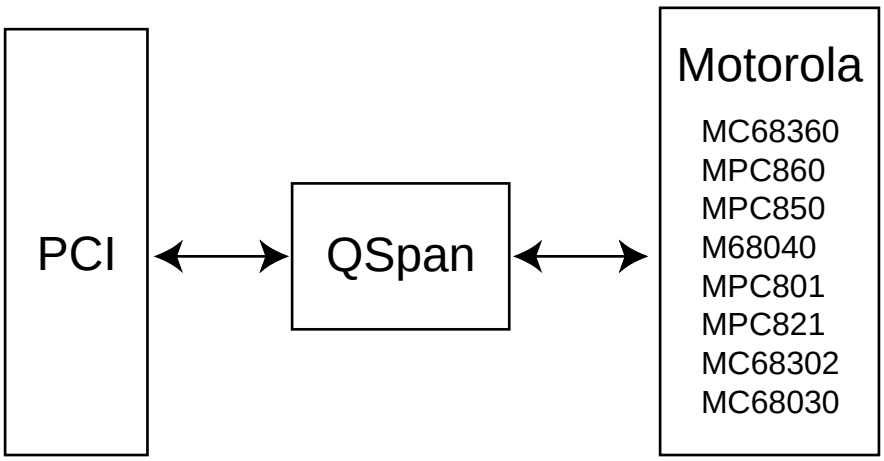
1-800-267-7231 FAX (613) 592-1320 http://www.tundra.com

1 General Information

1.1 Introduction

The QSpan™ chip is a member of Tundra Semiconductor Corporation's growing family of PCI bus-bridging devices enabling board designers to bring PCI-based embedded products to market faster, for less cost and with high performance. Developed as part of an ongoing strategic relationship with Motorola®, the QSpan is a PCI to processor bus bridge. The QSpan is designed to gluelessly bridge the QUICC™ (MC68360), the PowerQUICC™ (MPC860, MPC850, MPC801), the M68040, the M68060 and the MPC821 embedded controllers to PCI. The QSpan can also be connected to lower end communications controller and processors such as the MC68302 and MC68030, with some additional glue logic.

Figure 1.1 The QSpan Bridging PCI and Processor Buses



1.2 Product Overview

The Features

The QSpan has the following features:

- A direct connect interface to the PCI bus for Motorola's QUICC (MC68360), PowerQUICC (MPC860, MPC850,MPC801), the M68040, the M68060 and the MPC821 embedded controllers
- Support for up to 50 MHz MPC8xx bus frequencies
- Available in 208-PQFP (5 volt) and 256-PBGA (3.3 volt)
- 32-bit PCI interface compliant with PCI Revision 2.1
- Five FIFOs buffer multiple transactions in both directions
- Accepts and generates burst reads and writes on PCI and processor buses
- Separate channel supports QUICC and PowerQUICC IDMA
- Flexible address space mapping and translation between the PCI and processor buses;
- Programmable endian byte ordering
- Four user programmable images
- Serial EEPROM interface for Plug'N Play compatibility
- Interrupt handler and generator on PCI bus and processor bus
- Registers accessible from both PCI and processor buses
- PCI bus and processor bus can be operated at different clock frequencies
- IEEE 1149.1 JTAG boundary-scan support

1.3 Using This Document

This manual is organized as follows:

- Chapter 1 - General Information
- Chapter 2 - Functional Description
- Chapter 3 - Signal Description
- Chapter 4 - Signals and DC Characteristics
- Appendix A - Registers

- Appendix B - Timing
- Appendix C - Typical Applications
- Appendix D - Initialization
- Appendix E - Endian Mapping
- Appendix F - Operating and Storage Conditions
- Appendix G - Mechanical and Ordering Information
- Index
- Sales Network

Chapter 1 introduces the QSpan and provides the reader with conventions used in the manual.

Chapter 2 describes the QSpan's functionality.

Chapter 3 describes the QSpan's signals.

Chapter 4 provides the DC characteristics and pin-out.

The Appendices are reference sources necessary for the implementation of the QSpan.

The Sales Network appendix at the end of the manual lists our current team of national and international sales representatives and distributors.

1.4 Revision History

The current version of the *QSpan - PCI to Motorola Processor Bridge Manual*, document number 8091860.MD302.05, is a revision of document 8091860.MD302.04.

All of the chapters for this manual have been updated. However, the main revisions in the new manual are:

- Chapter 2 - Functional Description
- Chapter 4 - Signals and DC Characteristics
- Appendix B - Timing

1.5 Conventions

1.5.1 Signals

Signals on the QSpan are either active high or active low. Active low signals are defined as true (asserted) when they are at a logic low. Similarly, active high signals are defined as true at a logic high. Signals are considered asserted when active and negated when inactive, irrespective of voltage levels.

For voltage levels, the use of ‘0’ indicates a low voltage while a ‘1’ indicates a high voltage.

Table 1.1 Signal Conventions Used in this document

SIGNAL#	active low signals on the PCI interface
$\overline{\text{SIGNAL}}$	active low signals on the QBus interface

In addition to the difference between PCI and processor bus signals, the PowerQUICC conventions are different from both the other Motorola conventions and the PCI conventions. PowerQUICC signals are arranged in the opposite order from the other processor buses and PCI buses: i.e., the least significant signals are listed first. For example, the QUICC A[31:0] signals correspond to the PowerQUICC A[0:31] signal. This manual adopts the convention that the most significant bit (address or data) is always the largest number. PowerQUICC designers must ensure that they connect their pins accordingly: e.g., pin A[31] on the QSpan connects to pin A[31] on the QUICC bus, but connects to pin A[0] on the PowerQUICC bus. This applies to all PowerQUICC buses (D[31:0], AT[3:0], TSIZ[1:0]) not only the address bus.

1.5.2 Terminology

The term “QBus” is used as a generic term referring to the interface between the QSpan and the processor bus to which the QSpan is connected.

When discussing bus ownership, this manual uses “master” to indicate the bus owner, and “slave” or “target” to indicate the address accessed by the master. The term “master” is used for both the PCI and the QBus, while “slave” is reserved for addresses accessed by QBus masters, and “target” is reserved for addresses accessed by PCI masters.

Note also that the term “cycle” refers to a single data beat, while a “transaction” is composed of one or more cycles.

1.5.3 Symbols

Caution: This symbol alerts the reader to procedures or operating levels which may result in misuse of or damage to the QSpan.

Note: This symbol directs the reader's attention to useful information or suggestions.

The globe symbol alerts the reader that the initialization must be performed as a minimum to access the channel in question.

1.6 QSpan Technical Support and Documentation

Tundra is dedicated to providing our customers with superior technical documentation and support. The following means of support are available:

1. **The QSpan User Manual.** This is the main source of technical information. We strive hard to produce excellent documentation, and this manual contains the answers to most of our customers' questions.
2. **The QSpan Documentation Web Page**. This contains the latest manual, application notes, FAQ, articles, and any device errata and manual addenda. Please visit and bookmark http://www.tundra.com
3. **QSpan Design Notes**. This documentation informs and guides clients through specific design issues associated with the QSpan. The latest Design Notes are available on the Tundra website - www.tundra.com.

4. **The Designer's Resource Forum**. This is a public discussion forum at http://www.tundra.com which allows you to post questions and read threads pertaining specifically to the QSpan. Tundra technical support staff moderate this forum and promptly respond to customer inquiries.
5. **The Designer's Resource Center.** You can tailor how the Tundra web site is presented to you by using this web resource, available from http://www.tundra.com. There, you may also register to receive automatic e-mail notification when the addendum, manual, or any other "resource" has changed. This is the best way to ensure that you always have the latest QSpan documentation.
6. **support@tundra.com**. You may also direct questions and feedback to Tundra using this e-mail address. Please include "QSpan" in the subject header of your message.
7. **Phone support.** Tundra's technical support staff may be reached at (613) 592-0714 - Please ask for QSpan technical support.

2 Functional Description

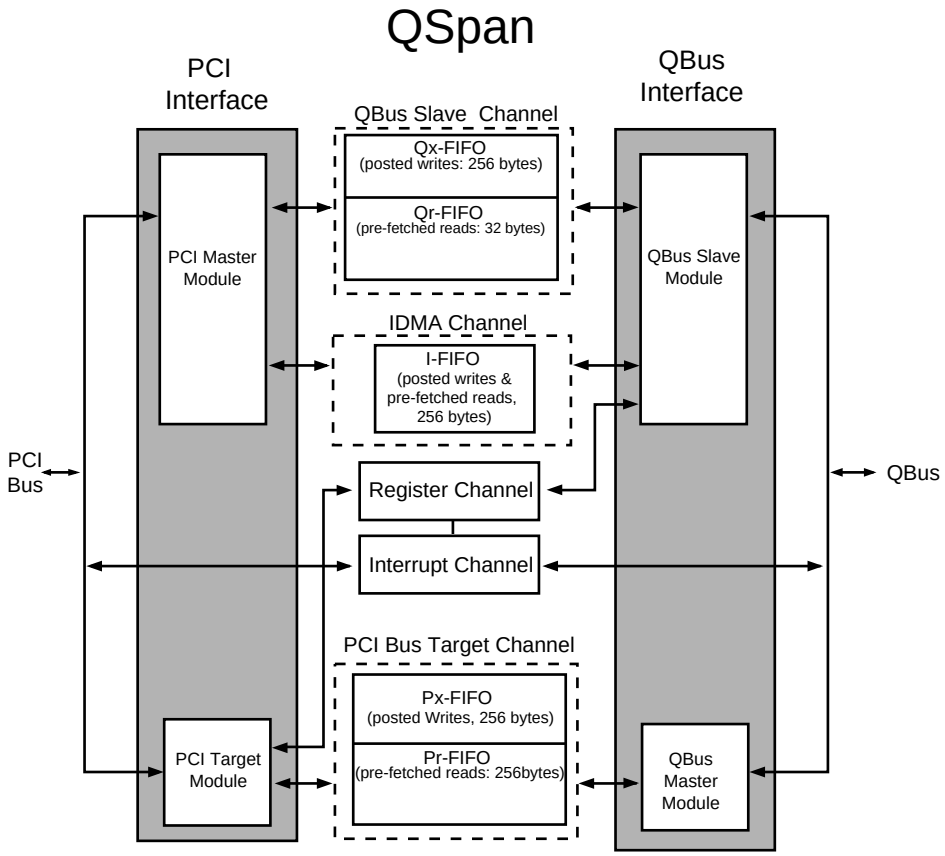
This chapter explains the operation of the QSpan. It is organized in terms of the data and control paths of the QSpan. The following topics are discussed.

2.1 Architectural Overview

The QSpan has two interfaces, a PCI bus interface and a “QBus” Interface (see Figure 2.1). The QBus is used to denote the programmable 860/360 and 040 interface on the QSpan. The PCI Interface is the QSpan’s connection to the PCI bus. The PCI Interface contains a PCI Master Module and a PCI Target Module. The QBus Interface is the interface which connects the QSpan to the processor bus. The processor interface also supports master and slave transactions. The QBus may be directly connected to a QUICC bus, a PowerQUICC bus, and/or a M68040 bus (or some other bus with glue logic). Each of the two interfaces has two functionally distinct modules: a master module, and a slave (or target) module. These modules are connected to the different functional channels operating in the QSpan.

The rest of this section describes the QSpan in terms of its channels.

Figure 2.1 QSpan Functional Block Diagram



2.1.1 The QBus Slave Channel: From Processor to PCI Bus

PCI Memory and I/O spaces are accessible through two slave images associated with the QBus Slave Channel. (Configuration space is accessible by accessing the CON_DATA register, page A-37). The QBus slave images are selected using a pair of chip-select signals on the QSpan.

The QBus Slave Channel supports posted writes, and delayed single reads and writes. Write transactions from the QBus to the PCI bus can be posted or delayed. Posted writes are queued in the Qx-FIFO with immediate data acknowledgment on the QBus. The QSpan then completes the write on the PCI bus. Delayed transactions - both reads and writes - require data acknowledgment on the PCI bus before data acknowledgment is provided on the QBus.

2.1.2 The PCI Target Channel: From PCI Bus to QBus

The QSpan provides two programmable target images on the PCI Interface. These images can be mapped anywhere in Memory or I/O space. Transactions from the PCI bus to the QBus can be executed as delayed or posted transactions.

PCI burst writes can be posted to ensure zero-wait state bursting. The 256-byte write FIFO (the Px-FIFO) allows for long PCI burst writes to be queued.

Delayed writes and reads must complete on the processor bus before data acknowledgment occurs on the on the PCI bus. Reads are always executed as delayed transactions, but the QSpan can be configured to pre-fetch read data. This supports long read bursts on the PCI bus. Pre-fetched reads are queued in a 256-byte FIFO (the Pr-FIFO).

2.1.3 The IDMA Channel

The QSpan can be programmed to operate as a QBus IDMA peripheral for high performance data transfer between the QBus and the PCI bus. For transfers going to or from PCI, software can perform bulk data movement using only the IDMA and the QSpan. The IDMA channel supports single and dual address cycles, and fast-terminations. A separate set of IDMA handshake signals are provided on the QBus. The IDMA Channel can be used by external QBus masters to read data from or write data to a PCI target in one direction at a time. The IDMA Channel contains a 256-byte FIFO (the I-FIFO) and a set of IDMA registers (see page A-30 to page A-34).

When programmed to perform writes to the PCI bus, the QSpan requests transfers from the processor IDMA on the QBus. Once the processor’s IDMA is requested for write data, it loads posted writes into the I-FIFO. When enough data is available in the I-FIFO for a burst transaction on the PCI bus, the QSpan requests the PCI bus and begins bursting data to the PCI target. This continues until the number of transfers programmed in the QSpan's IDMA_CS register completes. This technique provides high data throughput by using the IDMA to drive data transfers across the QBus.

When programmed to perform IDMA transfers from the PCI bus to the QBus, the QSpan reads data from a PCI target and loads the data into the I-FIFO. As the I-FIFO begins to fill, the QSpan requests the processor’s IDMA to transfer data from the QSpan to the destination on the QBus. The processor’s IDMA then transfers data from the QSpan's I-FIFO until the number of transfers programmed into the QSpan's IDMA registers completes or the QSpan signals to the processor’s IDMA that there is currently no further data available in the I-FIFO. Like the method used for PCI writes, this technique optimizes data transfer by taking full advantage of the processor’s IDMA. See "The IDMA Channel" on page 2-49.

2.1.4 The Register Channel

The QSpan provides 4 Kbytes of Control and Status Registers (QCSRs) that are used to program PCI settings as well as the QSpan's device specific parameters. All of the QCSR space is accessible from both the PCI bus and the QBus.

An internal arbitration mechanism is used to grant access to the QCSRs. The access mechanisms (including arbitration protocol) for the QCSRs differ depending on whether the registers are accessed from the PCI bus or the QBus.

PCI configuration cycles may be generated from the QBus by accessing QSpan registers. The cycles proceed as delayed transfers. See "The Register Channel" on page 2-58.

2.1.5 The Interrupt Channel

The QSpan can generate interrupts based on hardware or software events. Two bidirectional interrupt pins are provided: one on the PCI Interface, the other on the QBus Interface. Interrupt registers track the status of errors, and allow users to enable, clear, and map them. Interrupts can also be generated using one of the four available software interrupt sources ("doorbell" interrupts). See "The Interrupt Channel" on page 2-64.

2.1.6 The EEPROM Channel

Certain registers of the QSpan can be programmed by data in an EEPROM at system reset. This allows board designers to set unique identifiers for their cards on the PCI bus at reset, to enable the PCI Bus Expansion ROM Control Register, and set various address and parameters of images. Configuring the QSpan with the EEPROM allows the QSpan to boot-up as a Plug 'n Play compatible device. The QSpan supports reads from and writes to the EEPROM. The EEPROM itself is not included with the QSpan.

2.2 The QBus Slave Channel

The QBus is the name given to the bus interfacing with the processor bus. The QBus direct-connects to a QUICC bus, a PowerQUICC bus, or a M68040 bus. The QBus may also be direct-connected to a combination of buses, such as a QUICC bus and a PowerQUICC bus. A QBus master may access a PCI target through the QSpan by using the QBus Slave Channel or the IDMA Channel. This section is concerned with the QBus Slave Channel and is broken down as follows:

- QBus Slave Channel Architecture Overview
- Overview of Channel Description
- Address Phase
- Data Transfer
- Termination Phase

The register section on page 22-62 discusses how to generate PCI configuration cycles from the QBus. The initialization of this channel is discussed in “QBus Slave Channel Initialization” on page D-2.

2.2.1 QBus Slave Channel Architecture Overview

In this subsection, the functional architecture of the QBus Slave Channel is briefly presented. The rest of the section deals with the processes involving the channel.

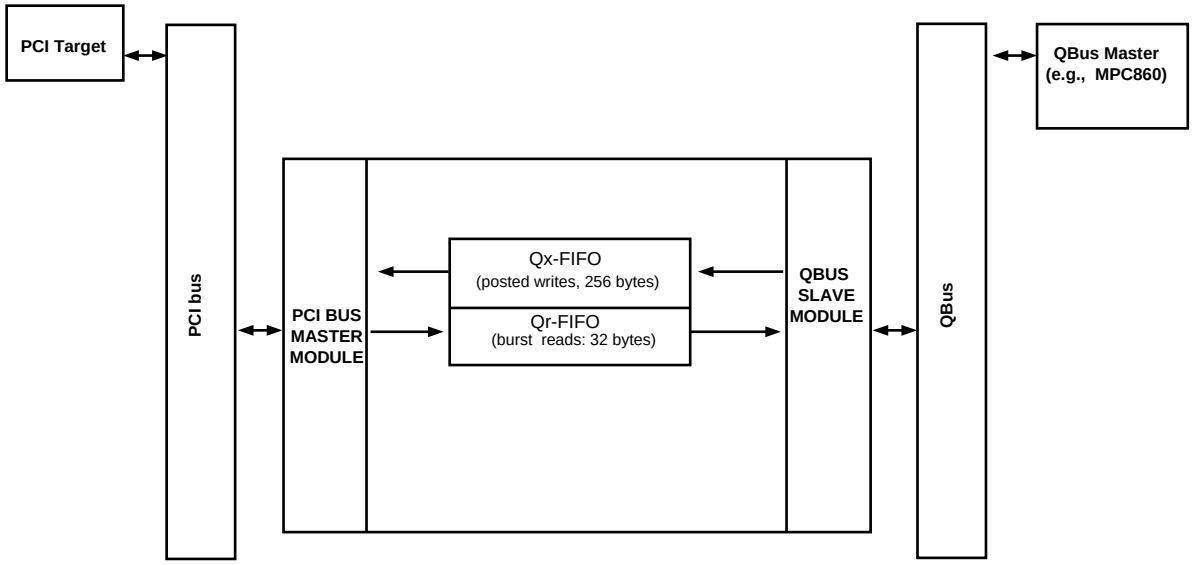


Figure 2.2 QBus Slave Channel in Context

Figure 2.2 depicts the QBus Slave Channel in relation to the QBus and the PCI bus. The QBus is shown as having a PowerQUICC processor and the PCI bus is shown with a single PCI device. The arrows represent data flow. The QBus Slave Channel has the following components:

- The QBus Slave Module
- The Qx-FIFO
- The Qr-FIFO
- The PCI Master Module

The first and last of these components are shared between the QBus Slave Channel and the IDMA channel. These components are discussed in turn.

2.2.1.1 QBus Slave Module

The QBus Slave Module is a non-multiplexed 32-bit address, 32-bit data interface. The QBus Slave Module is always capable of accepting QUICC cycles and either PowerQUICC or M68040 cycles. Whether the QBus Slave Module is in PowerQUICC or M68040 mode depends on the value of the SIZ[1] signal at reset. This reset option is presented in Table 2.1. (See also "Reset Options" on page 2-75). The MSTSLV[1:0] field in the MISC_CTL register indicates the slave (and master) mode of the QBus (page A-46). The connections required for interfacing the QSpan to a QUICC, PowerQUICC, and/or M68040 are given in “Typical Applications” on page C-1.

Table 2.1 Reset Options for QBus Slave Modes

Reset sampling of SIZ[1]	Slave Mode
0	QUICC and M68040
1	QUICC and PowerQUICC

2.2.1.2 Qx-FIFO and Qr-FIFO

The Qx-FIFO is a 256-byte buffer for posted writes from the QBus to the PCI bus (it can accommodate sixty-four 32-bit entries). The Qx-FIFO can accept data from an external QBus master while sinking data to a PCI target. The Qx-FIFO is described in greater detail in "Writes" on page 2-18.

The Qx-FIFO is on the data path for single delayed writes. A delayed write must be completed before the following write can be posted.

The Qr-FIFO is a 32-byte buffer used to store data read from PCI targets.

2.2.1.3 PCI Master Module

The PCI Master Module is a 32 bit/33MHz PCI 2.1 Specification compliant initiator interface. A list of all the PCI signals supported by the QSpan is in “PCI Bus Signals” on page 3-14. Please refer to the PCI 2.1 Specification for more information.

The QSpan requests PCI bus mastership through its PCI Master Module. The PCI Master Module is available to either the QBus Slave Channel (access from a remote QBus master) or the IDMA Channel.

2.2.2 Overview of Channel Description

The operation of the QBus Slave Channel is described below by tracing the path of a transaction from the QBus to the PCI bus. This is done by breaking transactions down into three phases:

- Address Phase
- Data Transfer
- Termination Phase

The address phase section describes transaction decoding and how address information from the QBus is passed through to a corresponding address space access on the PCI bus (e.g., how to access memory, I/O or Configuration space on the PCI bus).

The data transfer section describes endian mapping and byte lane translation through the QBus Slave Channel. This section also describes the different ways data is buffered within the QBus Slave Channel depending on the programming of the QBus slave images.

The termination phase section discusses how terminations from the PCI target are communicated back to the master on the QBus. It also describes how the QSpan PCI Master Module handles different terminations (e.g., retries or target-aborts) and the conditions that drive the terminations the QSpan issues as a QBus slave.

2.2.3 Address Phase

2.2.3.1 Transaction decoding and QBus Slave Images

The QSpan accepts a transaction through its QBus Slave Module when one of its chip selects is asserted along with the Address Strobe ($\overline{AS}$) or Transaction Start signal ($\overline{TS}$). The chip selects, ($\overline{CSREG}$, $\overline{CSPCI}$), need not be detected asserted on the same clock edge as the transaction start signal for QBus Slave Channel accesses. This allows for wait states to be inserted to perform address decoding. However, the IDMA channel requires that $\overline{CSPCI}$ be detected asserted on the same clock edge as the transaction start signal for dual address IDMA transfers.

Single address IDMA transfers do not require $\overline{CSPCI}$ *to be asserted.*

If $\overline{\text{CSREG}}$ is asserted, then the transaction is decoded as a QSpan register access (if the address 0x504 is a PCI configuration cycle, see "The Register Channel" on page 2-58). Then the QBus master asserts the PCI chip-select pin ($\overline{\text{CSPCI}}$) and the QBus Slave Module claims the cycle for the QBus Slave Channel, and one of the two QBus slave images is selected. The QBus Slave image is qualified by the IMSEL (Image Select Signal).

The type of PCI cycle that is generated by the QSpan depends both on (a) which QBus slave image is selected, and (b) what type of transaction was initiated by the external QBus master. The level of IMSEL determines which of the two QBus slave images is used. (a) If IMSEL is at a logic 0, QBus Slave Image 0 is selected (see page A-50 and page A-52); if IMSEL is at a logic 1, QBus Slave Image 1 is selected (see page A-53 and page A-55). (b) The levels of $\overline{\text{BURST}}$ and R/$\overline{\text{W}}$ determine whether the QSpan will generate a single PCI cycle or a burst, a PCI read or a write, respectively. (There is some interaction between images and hardware signals, as described on pages A-50 to A-55).

A **slave image** is a set of parameters (encoded in QSpan registers) that control transfers from the QBus to the PCI bus (similar “target” images are provided in the PCI Target Channel). Two QBus slave images of equal capability are provided so that designers can quickly access (on the basis of hardware rather than software) different PCI addresses from the QBus, or access addresses in different ways. The two slave images are completely independent from one another. For example, the designer may set-up QBus Slave Image 0 to access a hard-disk using 128 Mbytes of memory in PCI memory space. The designer could simultaneously have QBus Slave Image 1 available to access a different device, with its own memory size (e.g., 32 Mbytes). For example, the designer would be able to access the first device with posted writes (PWEN = 1), and the other with delayed writes (PWEN = 0). (The PWEN bit has no effect on reads.) For a third type of access, it would be necessary to share one of the slave images.

Table 2.2 and Table 2.3 summarize the QBus slave image control and address fields.

Table 2.2 Address Fields for QBus Slave Image

Field	Abbreviation and Register Page	Description
Block Size	BS (page A-52 and page A-55)	Amount of PCI memory accessed from QBus
PCI Address Space	PAS (page A-50 and page A-53)	Mapping to PCI memory space or I/O space
Translation Address	TA (page A-52 and page A-55)	Address bits that are substituted to generate the PCI bus address
Enable Address Translation	EN (page A-52 and page A-55)	Enables address translation using TA field

Table 2.3 Control Fields for QBus Slave Image

Field	Abbreviation and Register Page	Description
Posted Write	PWEN (page A-50 and page A-53)	Posted write enable bit

The QBus Slave Channel allows a QBus master to access a range of addresses within PCI Memory or I/O space. The PCI address space bit (PAS) of the selected image determines whether the current transfer is directed towards PCI Memory or I/O space. The range of addresses that can be accessed through a slave image is controlled by the block size field (BS). Up to 2 GBytes of PCI Memory or I/O space can be accessed from the QBus in one slave image if address translation is required. The use of the Block Size, PCI Address Space, Translation Address and Enable Address Translation fields is discussed in "Address Translation" on page 2-12 and "Address Phase on the PCI Bus" on page 2-13.

The QBus Slave Image Control registers allow the user to specify how writes are processed (page A-50 and page A-53). If the PWEN bit is “1”, then the QSpan will perform posted writes when that particular QBus slave image is accessed with a single write; otherwise writes are handled as single delayed transactions.

QBus Slave Image 0 can be programmed from an external EEPROM.

The QSpan accepts cycles from QUICC bus masters and either PowerQUICC or M68040 bus masters, as discussed in "QBus Slave Module" on page 2-7. The wait states that the QSpan inserts as a QBus slave are listed in “Wait State Insertion (QBus Slave Module)” on page B-19.

2.2.3.1.1 PowerQUICC Cycles

The QSpan behaves as a PowerQUICC slave in response to the assertion of the $\overline{TS}$ signal when it is powered-up as a PowerQUICC slave (see "QBus Slave Module" on page 2-7). When the QBus Slave Module receives $\overline{TS}$ it always responds with $\overline{DSACK1}$/$\overline{TA}$, $\overline{TEA}$, or $\overline{HALT}$/$\overline{TRETRY}$. The QSpan recognizes a transaction as intended for it, and acknowledges it accordingly, only if one of $\overline{CSREG}$ or $\overline{CSPCI}$ is sampled active in conjunction with $\overline{TS}$. The QSpan samples the address bus and other $\overline{TS}$ qualified signals on the same rising edge of QCLK in which it samples $\overline{TS}$ asserted.

If $\overline{BURST}$/$\overline{TIP}$ is asserted at the beginning of the bus cycle (along with the address) the QSpan accepts the incoming cycle as a burst. During bursts, the QSpan monitors $\overline{BDIP}$, which when negated indicates that the current data phase is second last.

When the QSpan is operating as a PowerQUICC slave for non-IDMA transfers, it functions as a 32-bit peripheral and must be addressed as such. External QBus masters must comply with the PowerQUICC timing specification.

2.2.3.1.2 QUICC Cycles

The QSpan behaves as a QUICC slave in response to the assertion of the $\overline{AS}$ signal. When it receives $\overline{AS}$ it always asserts a subset of $\overline{DSACK1}/\overline{TA}$, $\overline{DSACK0}$, $\overline{BERR}/\overline{TEA}$ and $\overline{HALT}/\overline{TRETRY}$. The QSpan recognizes a transaction as intended for it, and acknowledges it accordingly, only if one of $\overline{CSREG}$ or $\overline{CSPCI}$ is sampled active in conjunction with $\overline{AS}$. The QSpan does not require that the input signals qualified by $\overline{AS}$ be valid when $\overline{AS}$ is asserted—it requires only that they meet the set-up time before the same falling clock edge when $\overline{AS}$ is first sampled asserted. The QBus Slave Module inserts one wait state during QUICC cycles.

When the QSpan is operating as a QUICC slave, it functions as a 32-bit peripheral in synchronous mode. As a master, the QSpan also operates synchronously and therefore the BSTM bit in the QUICC should be set to a “1”. External QBus masters must comply with the QUICC timing specification.

2.2.3.1.3 M68040 Cycles

The QSpan behaves as a M68040 slave in response to the assertion of the $\overline{TS}$ signal when it is powered-up as a M68040 slave (see "QBus Master and Slave Modes" on page 2-77). When the QBus Slave Module receives $\overline{TS}$ it always responds with $\overline{DSACK1}/\overline{TA}$ or $\overline{BERR}/\overline{TEA}$. The QSpan recognizes a transaction as intended for it, and acknowledges it accordingly, only if one of $\overline{CSREG}$ or $\overline{CSPCI}$ is sampled active in conjunction with $\overline{TS}$. The QSpan samples the address bus and other $\overline{TS}$ qualified signals on the same rising edge of QCLK in which it samples $\overline{TS}$ asserted. The QBus Slave Module accepts bursting of incoming data.

When the QSpan is operating as a M68040 slave, it functions as a 32-bit peripheral in synchronous mode and must be addressed as such. External QBus masters must comply with the M68040 timing specification.

2.2.3.2 PCI Bus Request

The PCI Master Module requests the PCI bus when write data is received in the Qx-FIFO, the last data phase of a burst write has been received in the Qx-FIFO, or if there is a read request. When the QSpan requires control of the PCI bus it asserts REQ# and gains bus mastership when the PCI arbiter asserts GNT#. If the arbiter removes GNT# after the QSpan has begun its PCI transaction, the QSpan completes the current cycle and releases the PCI bus. This means that the QSpan PCI Master Module will have to re-arbitrate for the PCI bus after every cycle if its GNT# is removed. The QSpan performance as PCI master can be enhanced through bus parking, as defined in the PCI 2.1 Specification (GNT# asserted to the QSpan by the PCI bus arbiter).

The QSpan cannot be both Master and target on the PCI bus at the same time.

2.2.3.3 Address Translation

The QBus Slave Channel contains an Address Generator (see Figure 2.3 below) which is used if address translation is enabled (EN bit in Table A.42 or Table A.47). The Address Generator produces the PCI address using three inputs: the address of the QBus signal (A[31:0]), the block size of the QBus slave image (BS field of the QBSIx_AT register), and the translation address of the QBus slave image (TA field of the QBSIx_AT register). The translation address is a 16-bit number whose upper bits specify the location of the slave image on the PCI bus. The correlation between BS and the number of TA bits to use in generating the PCI address is shown in Table 2.4 on page 22-13. For example, with a 64 KByte block size, the Address Generator copies the entire translation address into the PCI address and only copies the lower 16 bits from the QBus address signals—i.e., the Address Generator translates A[31:16] and copies A[15:0] from the QBus. With a 2 GByte block size, the Address Generator copies all but bit 31 from the QBus address signal (i.e., the Address Generator translates A[31] only while copying A[30:0]), and uses the top translation address bit as bit 31 of the PCI address.

Figure 2.3 Address Generator for QBus Slave Channel Transfers

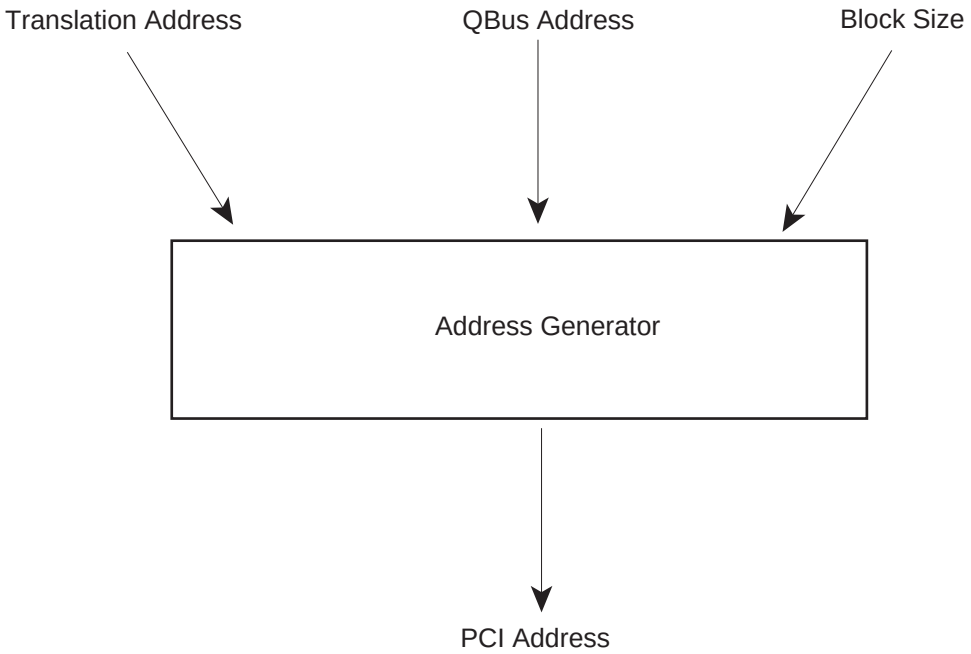


Table 2.4 Translation of QBus Address to PCI Address

BS in QBSI0_CTL or QBSI1_CTL	Block Size	Address Lines Translated	Translation Address Bits Copied
0000	64K	A31-A16	TA31-TA16
0001	128K	A31-A17	TA31-TA17
0010	256K	A31-A18	TA31-TA18
0011	512K	A31-A19	TA31-TA19
0100	1M	A31-A20	TA31-TA20
0101	2M	A31-A21	TA31-TA21
0110	4M	A31-A22	TA31-TA22
0111	8M	A31-A23	TA31-TA23
1000	16M	A31-A24	TA31-TA24
1001	32M	A31-A25	TA31-TA25
1010	64M	A31-A26	TA31-TA26
1011	128M	A31-A27	TA31-TA27
1100	256M	A31-A28	TA31-TA28
1101	512M	A31-A29	TA31-TA29
1110	1G	A31-A30	TA31-TA30
1111	2G	A31	TA31

This manual adopts the convention that the most significant bit (address or data) is always the largest number. PowerQUICC designers must ensure that they connect their pins accordingly: e.g., pin A[31] on the QSpan connects to pin A[31] on the QUICC bus, but connects to pin A[0] on the PowerQUICC bus. This applies to all PowerQUICC buses (D[31:0], AT[3:0], TSIZ[1:0]) not only the address bus.

2.2.3.4 Address Phase on the PCI Bus

The address supplied on the AD[31:0] lines on the PCI bus is the result of the address translation described above. The PCI command encoding on the C/BE#[3:0] lines is determined by the type of transaction on the QBus and the programming of the PAS bit in the QBus Slave Image Control Register (page A-50 or page A-53).

Table 2.5 lists the C/BE encoding supported by the QSpan.

Table 2.6 below shows the mapping from QBus transaction type to PCI transaction type as a function of PAS programming.

Table 2.5 Command Type Encoding for Transfer Type

C/BE# [3:0]	Command Type	QSpan Capability
0000	Interrupt Acknowledge	See "Interrupt Acknowledge Cycle" on page 2-68
0001	Special Cycle	N/A
0010	I/O Read	Target/Master
0011	I/O Write	Target/Master
0100	Reserved	N/A
0101	Reserved	N/A
0110	Memory Read	Target/Master
0111	Memory Write	Target/Master
1000	Reserved	N/A
1001	Reserved	N/A
1010	Configuration Read	Target/Master
1011	Configuration Write	Target/Master
1100	Memory Read Multiple	Master
1101	Dual Address Cycle	N/A
1110	Memory Read Line	Master for IDMA transfer
1111	Memory Write and Invalidate	Master for IDMA transfer

PCI targets are expected to assert DEVSEL# if they have decoded the access. If a target does not respond with DEVSEL# within 6 clocks, a Master-Abort is generated by the QSpan.

Table 2.6 Translation from QBus Transaction to PCI Transaction Type

QBus transaction received	PAS bit programming	PCI transaction type
Single or Burst Read	Memory	Memory Read
Single Read	I/O	I/O Read
Burst Read	I/O	None[a]
Single or Burst Write	Memory	Memory Write
Single Write	I/O	I/O Write
Burst Write	I/O	None[a]

a. In this case an error is signalled on the QBus.

2.2.4 Data Transfer

This section describes how endian mapping is executed in the QBus Slave Channel; it also discusses the data path for different transaction types.

2.2.4.1 Endian Mapping

The PCI bus and the Motorola processors have some differences in the way they order and address bytes. These differences are explained in Appendix E. The present section describes how the QSpan translates cycles from the QBus to the PCI bus.

The PCI bus is always a Little-Endian environment. The QBus may be configured as Little-Endian or Big-Endian, depending on the value of the QBus Byte Ordering Control bit (QB_BOC) in the MISC_CTL register (page A-46). The default mode for the QBus is Big-Endian. The QSpan translates byte lane ordering when the QBus is Big-Endian, while preserving the addressing of bytes. When the QBus is Little-Endian (according to QB_BOC), the QSpan preserves byte lane ordering, while translating the addressing of bytes. Note that the QB_BOC bit affects transactions in all channels.

Table 2.7 and Table 2.8 describe cycle mapping for Little-Endian and Big-Endian transfers, respectively, for transfers of all sizes (8, 16, 24, or 32 bits).

Table 2.7 Little-Endian QBus Slave Channel Cycle Mapping

QBus			PCI bus	
SIZ[1:0]	**A[1:0]**	**D[31:0]**	**BE[3:0]#**	**D[31:0]**
01	00	B3 xx xx xx	0111	B3 xx xx xx
01	01	xx B2 xx xx	1011	xx B2 xx xx
01	10	xx xx B1 xx	1101	xx xx B1 xx
01	11	xx xx xx B0	1110	xx xx xx B0
10	00	B3 B2 xx xx	0011	B3 B2 xx xx
10	01	xx B2 B1 xx	1001	xx B2 B1 xx
10	10	xx xx B1 B0	1100	xx xx B1 B0
10	11	xx xx xx B0	1110	xx xx xx B0
11	00	B3 B2 B1 xx	0001	B3 B2 B1 xx
11	01	xx B2 B1 B0	1000	xx B2 B1 B0
11	10	xx xx B1 B0	1100	xx xx B1 B0
11	11	xx xx xx B0	1110	xx xx xx B0
00	00	B3 B2 B1 B0	0000	B3 B2 B1 B0
00	01	xx B2 B1 B0	1000	xx B2 B1 B0
00	10	xx xx B1 B0	1100	xx xx B1 B0
00	11	xx xx xx B0	1110	xx xx xx B0

This manual adopts the convention that the most significant bit is always the largest number. PowerQUICC designers must ensure that they connect their pins accordingly: e.g., pin A[31] on the QSpan connects to pin A[31] on the QUICC bus, but connects to pin A[0] on the PowerQUICC bus. This applies to all PowerQUICC buses (D[31:0], AT[3:0], TSIZ[1:0]) not only the address bus.

Table 2.8 Big-Endian QBus Slave Channel Cycle Mapping

QBus			PCI bus	
SIZ[1:0]	**A[1:0]**	**D[31:0]**	**BE[3:0]#**	**D[31:0]**
01	00	B0 xx xx xx	1110	xx xx xx B0
01	01	xx B1 xx xx	1101	xx xx B1 xx
01	10	xx xx B2 xx	1011	xx B2 xx xx
01	11	xx xx xx B3	0111	B3 xx xx xx
10	00	B0 B1 xx xx	1100	xx xx B1 B0
10	01	xx B1 B2 xx	1001	xx B2 B1 xx
10	10	xx xx B2 B3	0011	B3 B2 xx xx
10	11	xx xx xx B3	0111	B3 xx xx xx
11	00	B0 B1 B2 xx	1000	xx B2 B1 B0
11	01	xx B1 B2 B3	0001	B3 B2 B1 xx
11	10	xx xx B2 B3	0011	B3 B2 xx xx
11	11	xx xx xx B3	0111	B3 xx xx xx
00	00	B0 B1 B2 B3	0000	B3 B2 B1 B0
00	01	xx B1 B2 B3	0001	B3 B2 B1 xx
00	10	xx xx B2 B3	0011	B3 B2 xx xx
00	11	xx xx xx B3	0111	B3 xx xx xx

This manual adopts the convention that the most significant bit is always the largest number. PowerQUICC designers must ensure that they connect their pins accordingly: e.g., pin A[31] on the QSpan connects to pin A[31] on the QUICC bus, but connects to pin A[0] on the PowerQUICC bus. This applies to all PowerQUICC buses (D[31:0], AT[3:0], TSIZ[1:0]) not only the address bus.

2.2.4.2 Data Path

2.2.4.2.1 Writes

If the PWEN bit is set in the QBSIx_CTL register, single write transactions from the QBus will be posted (page A-50 and page A-53). (The level of IMSEL determines which QBSIx_CTL register is used, see "Transaction decoding and QBus Slave Images" on page 2-8). If the PWEN bit is cleared (the default setting) single write transactions are treated as delayed transactions. Burst write transfers are always treated as posted writes.

Both posted and delayed writes travel through the Qx-FIFO. With posted writes data acknowledgment is provided on the QBus as soon as the write data is queued in the Qx-FIFO. Multiple posted writes can be queued up to the 256-byte capacity of the Qx-FIFO. With delayed writes, data acknowledgment is provided on the QBus only after completion on the PCI bus. This means only one delayed write at a time. Further writes are retried while a delayed write is in progress.

Posted write transfers are stored in the Qx-FIFO. Address and data are stored as separate entries in the Qx-FIFO. For example, one single cycle data beat transaction is stored as two Qx-FIFO entries: one entry for the address of the transaction, and one for the data. The address entry contains the translated PCI address space and command information mapping relevant to the particular QBus slave image that has been accessed (see "Address Phase" on page 2-8); the data entry contains the data and the byte enables. Thus, any reprogramming of QBus slave image attributes will only be reflected in Qx-FIFO entries queued after the reprogramming. Transactions queued before the reprogramming are delivered to the PCI bus with the QBus slave image attributes that were in use before the reprogramming. The QSpan never packs data in the Qx-FIFO. For example, two 16-bit data beats are not packed as a single 32-bit data entry but as four separate entries in the Qx-FIFO (address, data, address, data).

If a QBus master attempts to post a write transaction when the Qx-FIFO does not have enough space, the QBus Slave Channel retries the master. (The exact manner in which the master is retried depends on whether the master is a QUICC, PowerQUICC or M68040 device. See "Termination Phase" on page 2-21). Since single transfers require two entries in the Qx-FIFO, two entries must be available before the QBus Slave Module accepts a single write transaction. However, the 256-byte depth of the Qx-FIFO ensures very low probability of the Qx-FIFO being too full to accept write transactions.

The PCI Master Module requests the PCI bus when there is a complete transaction in the Qx-FIFO. During write transactions, the PCI Master Module uses transactions queued in the Qx-FIFO to generate transactions on the PCI bus. No address phase deletion is performed; thus, the length of a transaction on the PCI bus corresponds to the length of the queued QBus transaction.

Only PowerQUICC and M68040 masters are capable of initiating burst transactions. Incoming burst write transactions comprise five entries in the Qx-FIFO: one entry for address and command information, and four data entries. The QBus Slave Module only accepts bursts if the Qx-FIFO has enough room for the entire burst. Burst transfers are never retried while they are in progress. The PowerQUICC and M68040 only perform bursts of 16 bytes. Bursts accepted from the QBus are translated to the PCI bus as one or more burst transactions. The QSpan always bursts in linear increments.

Burst transactions are always posted (regardless of the programming of the PWEN bit in the selected QBSIx_CTL register). The QSpan accepts bursts targeted to Memory space (not to I/O or Configuration space). If the PCI address space bit (PAS) of the selected image is set to I/O space and a burst is initiated by a QBus master, then the QSpan signals a bus error. Similarly, if a burst is attempted to the QSpan registers, a bus error is signaled by the QSpan. (See Table 2.6 on page 22-14 and "Termination Phase" on page 2-21).

2.2.4.2.2 Read Transactions (Burst and Single Cycle)

During a read transaction, address, data, size and transaction code signals are latched by the QBus Slave Module. After latching the information, the QSpan retries all incoming QBus cycles (without latching them) until the read completes on the QBus. The QSpan becomes PCI bus master and performs a read transaction on the PCI bus. The burst read data is queued in the Qr-FIFO. If the PCI transaction completes normally, then the QBus master is provided with the data from the Qr-FIFO and the transaction terminates normally on the QBus (see "Termination Phase" on page 2-21).

If the QBus master attempts a burst read to the QBus Slave Module, and the slave image is programmed for PCI Memory Space, then the QSpan initiates a read cycle. If the read is attempted to a slave image programmed for PCI I/O Space, or to QSpan registers, then the QBus Slave Module terminates the access with a bus error. All reads through the QBus Slave Channel proceed as delayed transfers.

The burst read is the same as the single read, except that the PCI read is 4 beats in length.

2.2.4.2.3 Delayed Reads and PCI Transaction Ordering

In order to satisfy PCI Transaction Ordering requirements, the rules described in this section are implemented in delayed reads. Please refer to the PCI 2.1 Specification for more information. These rules affect the relation between delayed reads within the QBus Slave Channel and posted writes within the QBus Slave Channel; they also affect the relation between delayed reads in the QBus Slave Channel and posted writes in the PCI Target Channel.

The following list summarizes the sequence of QSpan events.

1. The QBus Slave module receives a read request
2. The QBus Slave Channel empties the Qx-FIFO of any writes or completes any current reads
3. The QBus Slave module latches the read request
4. The QBus Slave module retries subsequent non-register accesses
5. The PCI Master module completes the read on the PCI bus
6. The Qx-FIFO is able to accept new write accesses
7. The PCI Target module retries all non-register accesses
8. The Px-FIFO is emptied
9. The QBus Slave module allows the read to complete on the QBus;
10. If the read has not completed then the QBus Slave module allows posted writes to the Qx-FIFO and the PCI Target module allows posted writes to the Px-FIFO

2.2.4.3 PCI Target Channel Reads

The same principles apply to PCI Target Channel reads, except that the PCI Target module latches the read request even when there is data in the Px-FIFO (it still only passes the information onto the QBus master module when the Px-FIFO is empty). See "Reads and PCI Transaction Ordering" on page 2-43. The IDMA Channel is independent from the QBus Slave Channel and PCI Target Channel.

2.2.4.4 Parity Monitoring by PCI Master Module

The QSpan monitors PAR when it accepts data as a PCI master during a read and drives PAR when it provides data as a PCI master during a write. The QSpan also drives PAR during the address phase of a transaction when it is a PCI master. In both address and data phases, the PAR signal provides even parity for C/BE#[3:0] and AD[31:0].

The PERESP bit in the PCI_CS register (page A-5) determines whether or not the QSpan responds to parity errors as PCI master. Data parity errors are reported through the assertion of PERR# if the PERESP bit is set. The D_PE (Detected Parity Error) bit in the PCI_CS register is set if the QSpan encounters a parity error as a PCI bus master on any transaction, even if the QSpan is not involved in the transfer.

The DP_D (Data Parity Detected) bit in the PCI_CS register is set if parity checking is enabled through the PERESP bit and the QSpan detects a parity error while it is PCI master (i.e. it asserts PERR# during a read transaction or receives PERR# during a write). If the QSpan sets the DP_D bit while the DPD_EN (Data Parity Detected Interrupt Enable) in the INT_CTL register (page A-41) is set, then it asserts an interrupt on the QBus or PCI bus interface (see "The Interrupt Channel" on page 2-64).

The QSpan continues with the transaction independent of any parity errors reported during the transaction.

2.2.5 Termination Phase

Except during posted writes (see "Posted Writes" on page 2-23), the termination generated by the QBus Slave Module is determined by the termination on the PCI bus. For read transactions and delayed write transactions, the QBus master is retried until the PCI transaction is complete. Once the PCI transaction is complete, the QBus master receives a translated version of the PCI termination. Table 2.9 below shows how PCI terminations are translated to the QBus Slave Module in the case of delayed transactions (delayed single reads, delayed single writes, and reads)

.

Table 2.9 Translation of Cycle Termination[a] from PCI Bus to QBus

PCI Bus Termination Received	QBus Termination Issued
Master-Completion	Normal
Master-Abort	Bus Error if MA_BE_D bit is “0” Normal if MA_BE_D bit is “1” (reads return all ones, write data flushed).
Target-Retry	N/A [b]
Target-Disconnect	
Target-Abort	Bus Error

a. This table applies to delayed transfers.

b. These cycles are not translated. The QBus Slave Module retries the master during delayed transactions until one of the other terminations is received.

The QBus Slave Module retries accesses under the following conditions:

- A QBus master attempts to post another write to the QBus Slave Channel and the Qx-FIFO does not have enough room
- A QBus master attempts a burst write transfer and there is not enough room in the Qx-FIFO for the complete burst. The QBus Slave Module never retries an ongoing burst transaction.
- A delayed transfer - read or write - is in progress in the QBus Slave Channel

The QBus Slave Module generates a bus error under the following conditions:

- A QBus master attempts to burst to a slave image whose transfers have been set to I/O space
- A delayed transfer results in a Target-Abort on the PCI bus
- A delayed transfer results in a Master-Abort and the MA_BE_D bit in the MISC_CTL register is “0”.

In order to comply with the PCI 2.1 Specification the MA_BE_D should be set if the QSpan is used as a host bus bridge.

The mapping of the QBus terminations to the QUICC, PowerQUICC and M68040 buses is shown in Tables 2.10 to 2.12, respectively.

Table 2.10 QUICC Cycle Terminations of QBus Slave Module

Termination Type	$\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$/$\overline{\text{TA}}$	$\overline{\text{BERR}}$ / $\overline{\text{TEA}}$	$\overline{\text{HALT}}$ / $\overline{\text{TRETRY}}$
Normal	Asserted	Tri-stated[a]	Tri-stated
Bus Error	Asserted	Asserted	Tri-stated
Retry	Asserted	Asserted	Asserted

a. External pull-ups bring tri-stated signals to the non-asserted state.

Table 2.11 PowerQUICC Cycle Terminations of QBus Slave Module

Termination Type	$\overline{\text{DSACK1}}$ / $\overline{\text{TA}}$	$\overline{\text{BERR}}$ / $\overline{\text{TEA}}$	$\overline{\text{HALT}}$ / $\overline{\text{TRETRY}}$
Normal	Asserted	Tri-stated[a]	Tri-stated
Bus Error	Tri-stated	Asserted	Tri-stated
Retry	Tri-stated	Tri-stated	Asserted

a. External pull-ups bring tri-stated signals to the non-asserted state.

Table 2.12 M68040 Cycle Terminations of QBus Slave Module

Termination Type	$\overline{\text{DSACK1}}$ / $\overline{\text{TA}}$	$\overline{\text{BERR}}$ / $\overline{\text{TEA}}$
Normal	Asserted	Tri-stated[a]
Bus Error	Negated	Asserted
Retry	Asserted	Asserted

a. External pull-ups bring tri-stated signals to the non-asserted state.

Table 2.13 summarizes the QSpan's response to abnormal terminations on the PCI bus.

Table 2.13 QBus Slave Channel Error Responses[a]

Transfer-type	PCI Error Type	MA_BE_D in MISC_CTL	Flush FIFO[b]?	QBus Termination
read	Master-Abort	0	Yes	Bus error
		1	Yes	Normal (return all "1")
	Target-Abort	0	Yes	Bus error
		1	Yes	Normal (return all "1")
posted write	Master-Abort	0	No. Lose one entry, continue sinking data.[c]	Normal
		1	No. Lose one entry, continue sinking data.[c]	Normal
	Target-Abort	0	No. Lose one entry, continue sinking data. [c]	Normal
		1	No. Lose one entry, continue sinking data.[c]	Normal

a. Underlined cells represent differences from previous revision of the QSpan.

b. This column pertains to Qr-FIFO in the case of reads, Qx-FIFO in the case of writes.

c. If a single posted write transfer results in a Master-Abort, then this complete transaction is lost and the QSpan will continue sinking any subsequent posted write entries.
If a burst write results in a Master-Abort on the first data beat, then this data entry is lost. It is likely that the second, third and fourth entries of the burst will also result in a Master-Abort and this data will be lost as well.

2.2.5.1 Posted Writes

QBus terminations of posted writes are not influenced by the PCI bus termination. If a posted write is terminated by a Target-Abort or Master-Abort on the PCI bus, this termination is not signalled on the QBus to the QBus master. However, errors on the PCI bus are accessible to external QBus masters via error logging if error logging is enabled by the EN bit of the PB_ERRCS register. Errors can be made to cause interrupts.

The QSpan can record the address, command, data, and byte enables of a posted write transaction that results in a Master-Abort or Target-Abort. The EN bit of the PB_ERRCS register enables error recording. If enabled, the occurrence of an error is indicated by the ES bit of the PB_ERRCS register. Transfers in the QBus Slave Channel are suspended until the ES bit is cleared. If enabled, the EEPROM_CS register latches the command information of the errored transaction (CMDERR field) as well as the byte enables of the errored transaction (BE_ERR field). The address of the errored transaction is latched in the PB_AERR register. The data of the errored transaction is latched in the PB_DERR register.

If error logging is enabled and the QBus Slave Channel is errored, the Qx-FIFO is frozen until the ES bit in the PB_ERRCS register is cleared. Posted write operation continues with the next enqueued posted write once the ES bit of the PB_ERRCS is cleared. If error logging is not enabled and the QBus Slave Channel is errored while dequeuing data the errored transfer is lost and FIFO operation continues with the next enqueued transfer.

An interrupt will be generated upon the logging of an error (ES bit in PB_ERRCS) only if the PEL_EN bit in the INT_CTL register is set (page A-41). If generated, the interrupt is directed to the QBus or the PCI bus, depending on the PEL_DIR bit in the INT_DIR register (page A-43). Interrupts are described in "The Interrupt Channel" on page 2-64.

2.3 The PCI Target Channel

An external PCI bus master may access a QBus slave through the QSpan by using its PCI Target Channel. This section is composed as follows:

- PCI Target Channel Architecture Overview
- Overview of Channel Description
- Address Phase
- Data Transfer
- QBus Arbitration and Sampling
- Terminations

The initialization of this channel is discussed in “PCI Target Channel Initialization” on page D-3.

2.3.1 PCI Target Channel Architecture Overview

In this subsection, the functional architecture of the PCI Target Channel is briefly presented. The rest of the section deals with the processes involving the channel.

Figure 2.4 PCI Target Channel

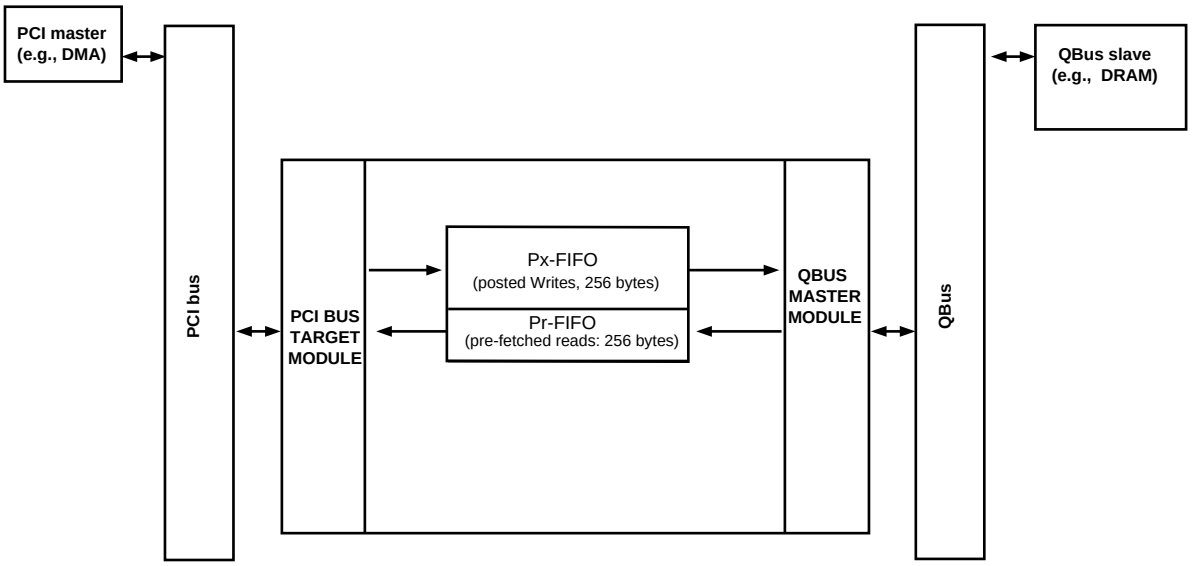


Figure 2.4 depicts the PCI Target Channel in relation to the QBus and the PCI bus. The arrows represent data flow. The QBus is shown as having one slave and the PCI bus is shown with a single PCI master. The PCI Target Channel has the following components:

- The PCI Target Module
- The Px-FIFO
- The Pr-FIFO
- The QBus Master Module.

These components are discussed in turn.

2.3.1.1 PCI Target Module

The QSpan's PCI Target Interface is a PCI 2.1 Specification compliant port. The QSpan does not implement SBO#, SDONE or PCI LOCK functionality. A list of all the PCI signals supported by the QSpan is in "PCI Bus Signals" on page 3-14.

The PCI Target Module accepts Type 0 Configuration cycles and ignores Type 1 Configuration cycles. Configuration cycles are not passed to the QBus.

2.3.1.2 Px-FIFO and Pr-FIFO

The Px-FIFO is a 256-byte buffer for posted writes from the PCI bus to the QBus. The Px-FIFO can accept data from a PCI master while writing data to a QBus slave. See "Posted Writes" on page 2-39.

The Pr-FIFO is a separate 256-byte buffer for read data from the QBus. See "Pre-fetched Read Transactions" on page 2-41.

2.3.1.3 QBus Master Module

The QBus Master Module is a non-multiplexed 32-bit address, 32-bit data interface. This module can be treated as a 32-bit, 16-bit, or 8-bit interface by programming the DSIZE field of the PCI target image (see Table 2.16 on page 22-28). The QBus Master Module is capable of generating QUICC, PowerQUICC, or M68040 cycles depending on the QBus master mode selected at reset. This reset option is determined jointly by the value of $\overline{\text{BDIP}}$ and $\overline{\text{SIZ}}$[1] at reset. Table 2.14 presents the QBus Master mode options; for completeness, it also includes the QBus Slave module options. The Master/Slave Mode (MSTSLV) field in the MISC_CTL register indicates the master and slave modes of the QBus (page A-46). The connections required for interfacing the QSpan to a QUICC, PowerQUICC, and/or M68040 are given in Appendix C.

Table 2.14 Reset Options for QBus Master and Slave Modes

Reset sampling		Master Mode	Slave Modes[a]
$\overline{\text{BDIP}}$	SIZ[1]		
0	0	QUICC	QUICC and M68040
0	1	QUICC	QUICC and PowerQUICC
1	0	M68040	QUICC and M68040
1	1	PowerQUICC	QUICC and PowerQUICC

a. This column is included because the master mode one chooses can restrict which slave one can use: the M68040 master mode is incompatible with the PowerQUICC slave mode.

The QBus Master Module supports bursts reads and burst writes in PowerQUICC mode only.

2.3.2 Overview of Channel Description

The operation of the PCI Target Channel is described below by tracing the path of a transaction from the PCI bus to the QBus. This is done by treating different aspects of a transaction in separate sections:

- **Address Phase**. This section describes how PCI bus accesses are decoded and how address information from the PCI bus is passed through to the QBus.
- **Data Transfer**. This section describes endian mapping and byte lane translation through the PCI Target Channel.
- **QBus Arbitration**. This section describes arbitration for the QBus.
- **Terminations**. This section explains how terminations from the QBus are communicated back to the master on the PCI bus. It also describes how the PCI Target Module handles different terminations (e.g., retries or target-aborts), the conditions that drive the terminations the QSpan issues as a PCI target, and error logging mechanisms for posted writes.

2.3.3 Address Phase

2.3.3.1 Transaction Decoding

All decoding by the PCI Target Module is based on the address and command information produced by a PCI bus master. The PCI Target Module claims a cycle if there is an address driven on the PCI bus that matches an image programmed into the PCI target image registers. The parameters of a target image must not overlap with the 4 Kbytes of QSpan Register Space. (Parameters for register accesses are discussed in "Register Access from the PCI Bus" on page 2-59.)

The type of cycle that is generated on the QBus is determined both by the target image that is selected and C/BE[3:0]. A **target image** is a set of parameters that determines what addresses are decoded on the PCI bus and how cycles are translated from the PCI bus to the QBus. Two target images of equal capability are provided so that PCI masters can quickly access different QBus devices, or the same device in different ways, without having to reconfigure QSpan registers. The two target images are independent from one another. For example, one target image may be set-up to access 1 Mbyte of 16-bit SRAM on the QBus using delayed writes, while the other could access 64 Mbytes of 32-bit SDRAM on the QBus with posted writes. PCI masters need not reconfigure QSpan registers to access either of these devices. For a third type of access, it would be necessary to share one of the target images.

Table 2.15 and Table 2.16 summarize the PCI target image control and address fields.

Table 2.15 Address Fields for PCI Target Image

Field	Description	Image	Register	Page
Base Address BA[31:16]	address lines compared in decoding	0	PCI_BST0 or PBTI0_ADD	A-10 or A- 20
		1	PCI_BST1 or PBTI1_ADD	A-12 or A-24
Block Size BS[3:0]	Determines the number of AD lines that are examined when decoding accesses from the PCI bus.	0	PBTI0_CTL	A-18
		1	PBTI1_CTL	A-22
PCI Address Space PAS	Memory space or I/O space	0	PCI_BST0 or PBTI0_CTL	A-10 or A-18
		1	PCI_BST1 or PBTI1_CTL	A-12 or A-22
Translation Address TA[31:16]	address bits that are substituted to generate the QBus address	0	PBTI0_ADD	A-20
		1	PBTI1_ADD	A-24

Table 2.16 Control Fields for PCI Target Image

Field	Description	Image	Register	Page
Image Enable EN	enable bit	0	PBTI0_CTL	A-18
		1	PBTI1_CTL	A-22
Posted Write Enable PWEN	determines whether writes are posted or processed as delayed transactions	0	PBTI0_CTL	A-18
		1	PBTI1_CTL	A-22
Transaction Code TC[3:0]	transaction code generated on the QBus	0	PBTI0_CTL	A-18
		1	PBTI1_CTL	A-22
Port Size DSIZE	QBus destination port size	0	PBTI0_CTL	A-18
		1	PBTI1_CTL	A-22
Pre-fetch Read Enable PREN	determines whether QSpan pre-fetches data on the QBus	0	PBTI0_CTL	A-18
		1	PBTI1_CTL	A-22
Burst Write Enable BRSTWREN	determines whether QSpan performs burst writes on the QBus.	0	PBTI0_CTL	A-18
		1	PBTI1_CTL	A-22
Invert Endianness INVEND	inverts the endian setting of the QB_BOC setting of the MISC_CTL register.	0	PBTI0_CTL	A-18
		1	PBTI1_CTL	A-22

PCI target images can be enabled or disabled by using the image enable bit. Disabling both PCI target images effectively disables the PCI Target Channel.

A PCI target image occupies a range of addresses within PCI Memory or I/O space. The PCI address space bit determines whether the target image lies in PCI Memory or I/O space. The range of addresses is specified by the base address field and the block size field. Up to 2 GBytes of PCI memory per image can reside on the QBus.

There are constraints on the possible values of the block size and the base address. The block size must be one of the 16 block sizes listed in Table 2.17 on page 22-31. The base address must be aligned to a combination of the upper address lines between AD31 and AD16. The base address must be a multiple of the block size. For example, a 128 MByte image must be aligned to a 128 MByte boundary.

Address decoding is performed by decoding a number of most significant address lines that is a function of the block size. For a 128 MByte PCI target image, the PCI Target Module only needs to decode the top five PCI address lines to know whether this image has been accessed. For a 64 KByte image, the PCI Target Module needs to decode the top 16 PCI address lines.

When one of its PCI target images is accessed, the QSpan responds with DEVSEL# within two clocks of FRAME# (making the QSpan a medium speed device, as indicated by the DEVSEL field in the PCI_CS register, page A-5).

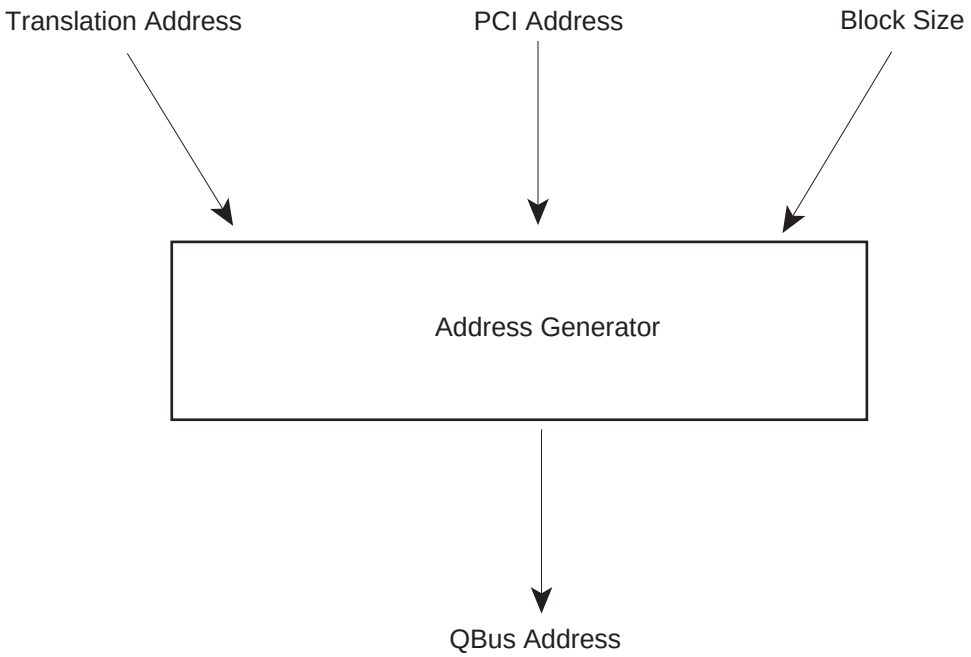
As PCI target, the QSpan responds to the following command types:

- I/O Read
- I/O Write
- Memory Read
- Memory Write
- Configuration Type 0 Read
- Configuration Type 0 Write
- Memory Read Multiple (aliased to Memory Read)
- Memory Read Line (aliased to Memory Read)
- Memory Write and Invalidate (aliased to Memory Write)

2.3.3.2 Address Translation

Figure 2.5 below illustrates the general implementation of address translation in the QSpan PCI Target Channel

Figure 2.5 Address Generator for PCI Target Channel Transfers



The Address Generator produces the QBus address using three inputs: the address generated by the PCI initiator (AD[31:0]), the block size of the PCI target image (BS field of the target image), and the translation address of the PCI target image (TA). The translation address is a 16-bit number whose upper bits specify the location of the target image on the QBus. If the translation address is programmed with the same value as that of the base address, then the PCI address is not translated but applied directly to the QBus transaction.

The most significant bits of the translation address are used by the address generator as determined by the programming of the image's block size. The number of these bits used depends on the programming of BS. The correlation between block size and the number of most significant TA bits used in generating the QBus address is shown in Table 2.17 below. For example, with a 64-KByte block size, the Address Generator copies the entire translation address into the QBus address and only copies the lower 16 bits from the PCI address signal (i.e., the Address Generator translates AD[31:16]). With a 2-GByte block size, the Address Generator copies all but bit 31 from the PCI address signal (i.e., the Address Generator translates AD[31] only), and uses the top translation address bit as bit 31 of the QBus address.

Table 2.17 Translation of PCI Bus Address to QBus Address

BS in PBTI0_CTL or PBTI1_CTL	Block Size	PCI Address Bits Translated	Translation Address Bits Copied
0000	64K	AD31-AD16	TA31-TA16
0001	128K	AD31-AD17	TA31-TA17
0010	256K	AD31-AD18	TA31-TA18
0011	512K	AD31-AD19	TA31-TA19
0100	1M	AD31-AD20	TA31-TA20
0101	2M	AD31-AD21	TA31-TA21
0110	4M	AD31-AD22	TA31-TA22
0111	8M	AD31-AD23	TA31-TA23
1000	16M	AD31-AD24	TA31-TA24
1001	32M	AD31-AD25	TA31-TA25
1010	64M	AD31-AD26	TA31-TA26
1011	128M	AD31-AD27	TA31-TA27
1100	256M	AD31-AD28	TA31-TA28
1101	512M	AD31-AD29	TA31-TA29
1110	1G	AD31-AD30	TA31-TA30
1111	2G	AD31	TA31

This manual adopts the convention that the most significant bit is always the largest number. PowerQUICC designers must ensure that they connect their pins accordingly: e.g., pin A[31] on the QSpan connects to pin A[31] on the QUICC bus, but connects to pin A[0] on the PowerQUICC bus. This applies to all PowerQUICC buses (D[31:0], AT[3:0], TSIZ[1:0]) not only the address bus.

The QSpan's chip select inputs ($\overline{CSREG}$, $\overline{CSPCI}$) must not be asserted when the QSpan is initiating a cycle on the QBus.

2.3.3.3 Transaction Codes on the QBus

The address supplied on the A[31:0] lines on the QBus is the result of the address translation described above. The QSpan also allows the user flexibly to generate various encodings on the QSpan's TC[3:0] lines. The TC[3:0] lines may be connected to the FC[3:0] lines of the QUICC bus, the AT[3:0] lines of the PowerQUICC bus, and a subset of the TT[1:0] and TM[2:0] lines of the M68040 bus. The QSpan copies the values from the TC field of the PCI target image of the current transaction to the TC[3:0] lines of the QBus. This gives the user additional control over the address information provided on the QBus.

2.3.3.4 PCI BIOS Memory Allocation

The PCI target image registers used by the QSpan to decode PCI accesses work differently depending on the EEPROM implementation. There are two possible cases:

1. The PCI_BSTx register is enabled by the EEPROM. For PCI Target Image 0, this means that bit-5 of byte 7 in the EEPROM is "1". For PCI Target Image 1, this means that bit-7 of byte 8 in the EEPROM is "1". See Table 2.43 on page 22-71.
2. The PCI_BSTx register is not enabled. For PCI Target Image 0, this means that bit-5 of byte 7 in the EEPROM is "0". For PCI Target Image 1, this means that bit-7 of byte 8 in the EEPROM is "0". See Table 2.43 on page 22-71.

Block Size and PCI Address Space

In the first case (PCI address programming from the EEPROM is enabled) the Block Size and PCI Address Space fields are set from the EEPROM, and they are read only. The PAS bit can be read from either the PCI_BSTx or the PBTIx_CTL register.

In the second case, the reset state of these two fields is zero, and they are writable from the PBTIx_CTL registers only.

Base Address

If PCI address information is loaded from an EEPROM (Case 1 above), the base address of the target image can only be set through the PCI_BSTx register (i.e., BA[31:16] of PCI_BST0 or PCI_BST1), but the base address can be read from either the PCI_BSTx register or the PBTIx_ADD register.

The PCI BIOS uses the PCI_BSTx registers to determine the address allocation for the QSpan-based board. It does this by writing all 1's to the BA field of the PCI_BSTx register and then reading back from the same location. The number of bits in the BA field of the PCI_BSTx register that are writable are determined by the target image's block size (BS[3:0] field in the PBTIx_CTL register). For example, if the PCI target image is programmed to a

block size of 128 MBytes, then the BS field in the PBTIx_CTL register has been programmed to all 1s (see Table 2.17 on page 22-31). This means that only the most significant five bits of the BA field of PCI_BSTx register are writable. When the BIOS reads back from the BA field in the PCI_BSTx register, the read returns only the most significant five bits of BA as “1”, indicating a 128 MBytes image size.

When PCI address information is not loaded from an EEPROM (Case 2 above), the base address can only be set through the PBTIx_ADD register and the PCI_BSTx registers do not support the image-sizing functionality described above.

2.3.4 Data Transfer

This section describes how endian mapping is executed in the PCI Target Channel; it also discusses the data path for different transaction types.

2.3.4.1 Endian Mapping

The PCI bus and the Motorola processors have some differences in the way they order and address bytes. These differences are explained in Appendix E. The present section describes how the QSpan translates cycles from the PCI bus to the QBus.

The PCI bus is always a Little-Endian environment. The QBus may be configured as Little-Endian or Big-Endian, depending on the value of the QBus Byte Ordering Control bit (QB_BOC) in the MISC_CTL register (page A-46). The default mode for the QBus is Big-Endian. This global ordering may be inverted on an image-by-image basis by programming the INVEND bit of the PBTIx_CTL register. The QSpan translates byte lane ordering when the QBus is Big-Endian, while preserving the addressing of bytes. When the QBus is Little-Endian (according to QB_BOC and INVEND), the QSpan preserves byte lane ordering, while translating the addressing of bytes. Note that the QB_BOC bit affects transactions in all channels whereas the INVEND bit only affects the PCI Target Channel.

Transactions from the PCI bus can be translated as 32-bit, 16-bit, or 8-bit on the QBus. The data width of the QBus transaction is controlled by the DSIZE field of the PCI bus Target Image Control register. With 16-bit peripherals they can be 16 or 8 bits wide. With 8-bit peripherals they can be 8 bits wide. Packing and unpacking of data performed by the QBus Master Module is a function of byte-enables (BE[3:0]) and the port size. The following two sections describe cycle mapping and the unpacking and packing of data.

2.3.4.1.1 Write Cycle Mapping for PCI Target Channel

This section describes write cycle mapping as a function of port size.

All tri-byte, misaligned and non-contiguous byte write operations on the PCI bus are performed as a series of 8-bit write operations on the QBus.

2.3.4.1.1.1 32-Bit QBus Port

Table 2.18 and Table 2.19 describe write transfers of various sizes to 32-bit peripherals on the QBus. Table 2.18 below describes mapping of 8, 16, and 32-bit write transfers through the PCI Target Channel with the QBus set to Little-Endian. The byte lane ordering is preserved in Little-Endian mode.

Table 2.18 Little-Endian PCI Target Write Cycle Mapping (32-Bit QBus Port)

Transfer size	PCI bus		QBus		
	BE[3:0]#	D[31:0]	SIZ[1:0]	A[1:0]	D[31:0]
8 bits	0111	B3 xx xx xx	01	00	B3 xx xx xx
8 bits	1011	xx B2 xx xx	01	01	B2 B2 xx xx
8 bits	1101	xx xx B1 xx	01	10	B1 xx B1 xx
8 bits	1110	xx xx xx B0	01	11	B0 B0 xx B0
16 bits	0011	B3 B2 xx xx	10	00	B3 B2 xx xx
16 bits	1100	xx xx B1 B0	10	10	B1 B0 B1 B0
32 bits	0000	B3 B2 B1 B0	00	00	B3 B2 B1 B0

Table 2.19 below describes mapping of 8, 16, and 32-bit write transfers through the PCI Target Channel in Big-Endian mode to 32-bit QBus peripherals. The addressing of bytes is preserved in Big-Endian mode.

Table 2.19 Big-Endian PCI Target Write Cycle Mapping (32-Bit QBus Port)

Transfer size	PCI bus		QBus		
	BE[3:0]#	D[31:0]	SIZ[1:0]	A[1:0]	D[31:0]
8 bits	1110	xx xx xx B0	01	00	B0 xx xx xx
8 bits	1101	xx xx B1 xx	01	01	B1 B1 xx xx
8 bits	1011	xx B2 xx xx	01	10	B2 xx B2 xx
8 bits	0111	B3 xx xx xx	01	11	B3 B3 xx B3
16 bits	1100	xx xx B1 B0	10	00	B0 B1 xx xx
16 bits	0011	B3 B2 xx xx	10	10	B2 B3 B2 B3
32 bits	0000	B3 B2 B1 B0	00	00	B0 B1 B2 B3

2.3.4.1.1.2 16-Bit QBus Port

16-bit QBus port transfers are explained below in terms of the 32-bit transfers described in Table 2.18 and Table 2.19. The first of these operations involves unpacking data.

- A 32-bit write operation to a QBus peripheral with a 16-bit QBus port size is performed as two 16-bit write operations would be performed to a 32-bit peripheral.
- A 16-bit write operation to a QBus peripheral with a 16-bit QBus port size is performed as a 16-bit write operation would be performed to a 32-bit peripheral.
- An 8-bit write operation to a QBus peripheral with a 16-bit QBus port size is performed as an 8-bit write operation would performed be to a 32-bit peripheral.

2.3.4.1.1.3 8-bit QBus port

8-bit QBus port transfers are explained below in terms of the 32-bit transfers described in Table 2.18 and Table 2.19. The first two operations involve unpacking data.

- A 32-bit write operation to a QBus peripheral with an 8-bit QBus port size is performed as four 8-bit write operations would be performed to a 32-bit peripheral.
- A 16-bit write operation to a QBus peripheral with an 8-bit QBus port size is performed as two 8-bit write operations would be performed to a 32-bit peripheral.
- An 8-bit write operation to a QBus peripheral with an 8-bit QBus port size is performed as an 8-bit write operation would be performed to a 32-bit peripheral.
- All tri-byte, misaligned and non-contiguous byte write operations are performed as a series of 8-bit write operations would be performed to a 32-bit peripheral.

2.3.4.1.2 Read Cycle Mapping for PCI Target Channel

This section describes cycle mapping and packing of data by the QBus Master Module.

2.3.4.1.2.1 32-bit QBus Port

Table 2.20 and Table 2.21 describe transfers of various sizes to 32-bit peripherals.

Table 2.20 below describes mapping of 8, 16, and 32-bit read transfers through the PCI Target Channel in Little-Endian mode to 32-bit QBus peripherals. The byte lane ordering is preserved in Little-Endian mode.

Table 2.20 Little-Endian PCI Target Read Cycle Mapping (32-Bit QBus Port)

Transfer size	PCI bus		QBus		
	BE[3:0]#	D[31:0]	SIZ[1:0]	A[1:0]	D[31:0]
8 bits	0111	B3 xx xx xx	01	00	B3 xx xx xx
8 bits	1011	xx B2 xx xx	01	01	xx B2 xx xx
8 bits	1101	xx xx B1 xx	01	10	xx xx B1 xx
8 bits	1110	xx xx xx B0	01	11	xx xx xx B0
16 bits	0011	B3 B2 xx xx	10	00	B3 B2 xx xx
16 bits	1100	xx xx B1 B0	10	10	xx xx B1 B0
32 bits	0000	B3 B2 B1 B0	00	00	B3 B2 B1 B0

Table 2.21 below describes mapping of 8, 16, and 32-bit read transfers through the PCI Target Channel in Big-Endian mode from 32-bit QBus peripherals. The addressing of bytes is preserved in Big-Endian mode.

Table 2.21 Big-Endian PCI Target Read Cycle Mapping (32-Bit QBus Port)

Transfer size	PCI bus		QBus		
	BE[3:0]#	D[31:0]	SIZ[1:0]	A[1:0]	D[31:0]
8 bits	1110	xx xx xx B0	01	00	B0 xx xx xx
8 bits	1101	xx xx B1 xx	01	01	xx B1 xx xx
8 bits	1011	xx B2 xx xx	01	10	xx xx B2 xx
8 bits	0111	B3 xx xx xx	01	11	xx xx xx B3
16 bits	1100	xx xx B1 B0	10	00	B0 B1 xx xx
16 bits	0011	B3 B2 xx xx	10	10	xx xx B2 B3
32 bits	0000	B3 B2 B1 B0	00	00	B0 B1 B2 B3

All tri-byte, misaligned and non-contiguous byte delayed read operations from a peripheral with a 32-bit QBus port size are performed as a series of 8-bit read operations would be from a 32-bit peripheral.

2.3.4.1.2.2 16-Bit QBus Port

Table 2.22 and Table 2.23 describe 8 and 16-bit read transfers from 16-bit QBus peripherals.

Table 2.22 below describes mapping of 8 and 16-bit read transfers through the PCI Target Channel in Little-Endian mode from 16-bit QBus peripherals. The byte lane ordering is preserved in Little-Endian mode.

Table 2.22 Little-Endian PCI Target Read Cycle Mapping (16-Bit QBus Port)

Transfer size	PCI bus		QBus		
	BE[3:0]#	D[31:0]	SIZ[1:0]	A[1:0]	D[31:0]
8 bits	0111	B3 xx xx xx	01	00	B3 xx xx xx
8 bits	1011	xx B2 xx xx	01	01	xx B2 xx xx
8 bits	1101	xx xx B1 xx	01	10	B1 xx xx xx
8 bits	1110	xx xx xx B0	01	11	xx B0 xx xx
16 bits	0011	B3 B2 xx xx	10	00	B3 B2 xx xx
16 bits	1100	xx xx B1 B0	10	10	B1 B0 xx xx

Table 2.23 below describes mapping of 8 and 16-bit read transfers through the PCI Target Channel in Big-Endian mode from 16-bit QBus peripherals. The addressing of bytes is preserved in Big-Endian mode.

Table 2.23 Big-Endian PCI Target Read Cycle Mapping (16-Bit QBus Port)

Transfer size	PCI bus		QBus		
	BE[3:0]#	D[31:0]	SIZ[1:0]	A[1:0]	D[31:0]
8 bits	1110	xx xx xx B0	01	00	B0 xx xx xx
8 bits	1101	xx xx B1 xx	01	01	xx B1 xx xx
8 bits	1011	xx B2 xx xx	01	10	B2 xx xx xx
8 bits	0111	B3 xx xx xx	01	11	xx B3 xx xx
16 bits	1100	xx xx B1 B0	10	00	B0 B1 xx xx
16 bits	0011	B3 B2 xx xx	10	10	B2 B3 xx xx

All tri-byte, misaligned and non-contiguous byte read operations from a peripheral with a 16-bit QBus port size are performed as a series of 8-bit read operations would be from a 16-bit peripheral.

- A 32-bit read operation from a peripheral with a 16-bit QBus port size is performed as two 16-bit read operations from a peripheral with a 16-bit QBus port size.

2.3.4.1.2.3 8-Bit QBus Port

Table 2.24 below describes mapping of 8-bit read transfers through the PCI Target Channel in Little-Endian mode from 8-bit QBus peripherals. The byte lane ordering is preserved in Little-Endian mode.

Table 2.24 Little-Endian PCI Target Read Cycle Mapping (8-Bit QBus Port)

Transfer size	PCI bus		QBus		
	BE[3:0]#	D[31:0]	SIZ[1:0]	A[1:0]	D[31:0]
8 bits	0111	B3 xx xx xx	01	00	B3 xx xx xx
8 bits	1011	xx B2 xx xx	01	01	B2 xx xx xx
8 bits	1101	xx xx B1 xx	01	10	B1 xx xx xx
8 bits	1110	xx xx xx B0	01	11	B0 xx xx xx

Table 2.25 below describes mapping of 8-bit read transfers through the PCI Target Channel in Big-Endian mode from 8-bit QBus peripherals. The addressing of bytes is preserved in Big-Endian mode.

Table 2.25 Big-Endian PCI Target Read Cycle Mapping (8-Bit QBus Port)

Transfer size	PCI bus		QBus		
	BE[3:0]#	D[31:0]	SIZ[1:0]	A[1:0]	D[31:0]
8 bits	1110	xx xx xx B0	01	00	B0 xx xx xx
8 bits	1101	xx xx B1 xx	01	01	B1 xx xx xx
8 bits	1011	xx B2 xx xx	01	10	B2 xx xx xx
8 bits	0111	B3 xx xx xx	01	11	B3 xx xx xx

All tri-byte, misaligned and non-contiguous byte read operations from a peripheral with an 8-bit QBus port size are performed as a series of 8-bit read operations would be from an 8-bit peripheral.

- A 32-bit read operation from a peripheral with an 8-bit QBus port size is performed as four 8-bit read operations from a peripheral with an 8-bit QBus port size.

- A 16-bit read operation from a peripheral with an 8-bit QBus port size is performed as two 8-bit read operations from a peripheral with an 8-bit QBus port size.

2.3.4.2 Data Path

This section explains how data flows between the PCI bus and the QBus through the PCI Target Channel.

The data paths for these transfers are discussed in greater detail below:

- "Posted Writes" on page 2-39 (single and burst writes)
- "Pre-fetched Read Transactions" on page 2-41
- "Single Read Transactions" on page 2-41
- "Delayed Writes" on page 2-41
- "Reads and PCI Transaction Ordering" on page 2-43.

2.3.4.2.1 Posted Writes

If the Posted Write Enable (PWEN) bit in the target image is “1”, writes to the PCI Target Module are posted in the Px-FIFO (see "Transaction Decoding" on page 2-27). If the bit is cleared (the default setting) writes are treated as delayed transactions.

Write transfers are stored in the Px-FIFO (see "Px-FIFO and Pr-FIFO" on page 2-26). Address and data are stored as separate entries in the Px-FIFO. For example, a single data transaction is stored as two entries in the Px-FIFO—one for the translated address and one for the data (see "Address Translation" on page 2-30). Thus, any reprogramming of PCI target image attributes will only be reflected in Px-FIFO entries queued after the reprogramming. Transactions queued before the re-programming are delivered to the PCI bus with the PCI target image attributes that were in use before the reprogramming.

The QSpan never packs data in the Px-FIFO. For example, two non-burst 16-bit data beats are not packed as a single 32-bit data entry but as four separate entries in the Px-FIFO (32-bit address, 16-bit data, 32-bit address, 16-bit data).

2.3.4.2.1.1 Acceptance of Burst Writes by the PCI Target Module

The PCI Target Module can accept burst write transactions from PCI bus masters. This section explains when PCI bursts are accepted, how they are stored, and how the data is transferred on the QBus.

The QSpan will not accept a PCI burst write under the following conditions.

1. If posted writes for the selected target image are disabled (page A-18 or page A-22) and a PCI master attempts to burst to the PCI Target Module, then each successive data phase is processed as a delayed single write. In this case, when the

write completes on the QBus, the QSpan issues a Target-Completion the next time the transfer is attempted by the external PCI master. Therefore, for each data phase in the burst, the external PCI master will see a series of retries and then one Target-Completion.

2. If the Px-FIFO fills while a burst is in progress, the PCI Target Module generates a Target-Disconnect.
3. If there are fewer data entries available in the Px-FIFO than specified by the cacheline (CLINE[1:0] field of the PCI_MISC0 register) when a PCI master attempts a new burst to the QSpan, the external PCI master is retried.
4. The PCI Target Module only accepts linear burst address incrementing. Any transfers requiring other addressing modes are disconnected after the first data phase.

See "Terminations driven by the PCI Target Module" on page 2-46.

The Px-FIFO stores the address and data entries of PCI bursts. For example, if a burst of four is received by the PCI Target Module, the QSpan stores the burst as five new entries of the following types: address, data, data, data, data. Because the QBus Master Module can write data at the same time as the PCI Target Module accepts data, the Px-FIFO might not contain these five entries by the end of the burst—some of the data may already have been written to the QBus before the burst completes on the PCI bus.

2.3.4.2.1.2 Bursting on the QBus

If the Burst Write Enable (BRSTWREN) bit of the selected PCI Target Image is “1”, the QSpan will burst data from the Px-FIFO onto the QBus. When enabled, all byte-lanes are assumed to be active for data written to the image. Generation of burst writes is only supported while in PowerQUICC master mode (see Table 2.14 on page 22-26).

Burst length on the QBus is controlled by the $\overline{\text{BDIP}}$ signal: by negating $\overline{\text{BDIP}}$ the QSpan signals the QBus slave that the current data beat is the second last beat of the transaction. This allows the QSpan to perform bursts of two, three or four data beats.

The PowerQUICC’s UPM does not monitor $\overline{\text{BDIP}}$ and only accepts bursts that begin at the cacheline (16-byte) boundaries. Therefore, there exists an incompatibility between the QSpan and the PowerQUICC’s memory controller for QSpan initiated burst write cycles. The QSpan should not be programmed to generate burst writes, i.e. set BRSTWREN to “0”.

2.3.4.2.2 Delayed Writes

If a write is attempted when posted writes are disabled for the PCI Target Image (PWEN=0 in the selected target image), or the address space bit is set to 1 (for I/O transfers), then write cycles are treated as delayed transactions. During a delayed write transaction the PCI initiator is retried until the transaction completes on the QBus. If the PCI transaction completes normally on the QBus, then when the PCI bus master retries the same transaction (qualified by the latched address and command information) the original PCI master is given a normal cycle termination. If the QBus transaction does not complete normally, then the appropriate termination is communicated back to the PCI master (see "Terminations" on page 2-45).

2.3.4.2.3 Single Read Transactions

When the QSpan receives a target read request, it latches the address and C/BE# information and retries the PCI master. The QSpan then becomes QBus master and initiates a read on the QBus. The external PCI master is retried until the read is completed on the QBus. When the external PCI master retries the same transaction (qualified by the latched information) it is provided with the data and the transaction terminates normally on the PCI bus (see "Terminations" on page 2-45). If the QBus transaction does not complete normally, then the appropriate termination is communicated back to the PCI bus master.

2.3.4.2.4 Pre-fetched Read Transactions

The QSpan will initiate a pre-fetch read transaction on the QBus if the following conditions are met:

1. The PREN bit in the selected target image must be “1”
2. The PRCNT[5:0] in the MISC_CTL register must be programmed
3. The PCI initiator must keep FRAME# asserted when IRDY# is asserted (i.e. PCI burst read cycle)

The QSpan will read the amount of data specified in the new PRCNT[5:0] field of the MISC_CTL register. The QSpan will retry the PCI initiator until read data is available in the Pr-FIFO. All of the data in the Pr-FIFO must be read during a single transaction on the PCI bus. Any data not read after the read will be considered stale and will be purged from the Pr-FIFO.

If the PREN bit is cleared (which is the default setting), the transfer is processed as a delayed single read (see "Single Read Transactions" on page 2-41). The QSpan will prefetch whether it is in QUICC, PowerQUICC, or M68040 Master Mode (see "QBus Master Module" on page 2-26). However, it will only prefetch with burst reads on the QBus when it is MPC860 Master Mode.

The PWEN bit and the BRSTWREN bit do not affect reads in the PCI Target Channel. If pre-fetching is enabled and the QSpan is in PowerQUICC *Master Mode, it will automatically perform burst reads when pre-fetching. This may require programming changes to the* PowerQUICC *UPM to support burst cycles.*

If a read-request is not cacheline aligned, then the QSpan will perform single beat transactions on the PowerQUICC bus until it reaches a cacheline boundary. The QBus Master module, as MPC860 master, only performs burst reads at cacheline boundaries. This makes QSpan burst reads compatible with the PowerQUICC UPM. The module requests the bus for a burst when there is enough room in the Pr-FIFO for an entire cacheline of data.

2.3.4.3 Parity Monitoring by PCI Target Module

The PCI Target Module verifies address and data parity, and may report them in hardware and/or software. Unlike the PCI Master Module, PCI Target Module parity errors cannot be mapped to interrupts (see "The Interrupt Channel" on page 2-64).

The PCI Target Module monitors parity during the address phase of transactions, and during the data phase of write transfers; i.e., the QSpan compares the PAR signal with the parity of AD[31:0] and C/BE[3:0]. The PAR signal provides even parity for C/BE#[3:0] and AD[31:0]. The QSpan drives PAR when it provides data as a target during a read.

If the PCI Target Module detects an address or data parity error, it sets the D_PE bit in the PCI_CS register (page A-5) regardless of any setting in the PCI_CS register (including the PERESP bit) and even if the QSpan is not the addressed target of the transfer. If the QSpan signals SERR#, it sets the S_SERR bit in the PCI_CS register. Notice that while there is a bit that specifically indicates an address parity error (i.e., S_SERR), there is no bit that specifically indicates a data parity error detected by the PCI Target Module (the QSpan's detection of a data parity error can be inferred from D_PE, PERESP, and SERR_EN being set while S_SERR is clear).

Address parity errors are reported if both PERESP and SERR_EN are set in the PCI_CS register (page A-5). Address parity errors are reported by the QSpan by asserting the SERR# signal and setting the S_SERR (Signalled SERR#) bit in the PCI_CS register. Assertion of SERR# can be disabled by clearing the SERR_EN bit in the PCI_CS register. An interrupt may be generated, and regardless of whether assertion of SERR# is enabled or not, the QSpan does not respond to the access with DEVSEL#. Typically, the master of the transaction terminates the cycle with a Master-Abort.

The PERESP (Parity Error Response) bit in the PCI_CS register affects how the QSpan responds to PCI parity errors. Only if the PERESP bit and the SERR_EN (SERR# Enable) bit are set does the QSpan report address parity errors by asserting SERR# and setting the S_SERR bit in the PCI_CS register. If the PERESP bit is set the QSpan reports data parity errors (during writes) by asserting PERR#.

2.3.5 Reads and PCI Transaction Ordering

The rules described in this section are implemented to satisfy PCI Transaction Ordering requirements as described in the PCI 2.1 Specification. These rules affect the relation between delayed reads within the PCI Target Channel and posted writes within the PCI Target Channel; they also affect the relation between delayed reads in the PCI Target Channel and posted writes in the QBus Slave Channel.

When a read request is latched, the PCI Target Module retries non-register accesses to the PCI Target Module until the read completes on the QBus. Once the read completes on the QBus, the QSpan ensures that all writes previously posted in the Qx-FIFO complete on the PCI bus before the read data is passed back to the PCI bus master that initiated the read transaction. During the period when the Qx-FIFO is being emptied, attempts to access the QBus Slave Channel are retried (register accesses are not affected).

The following list summarizes the sequence of events.

1. The PCI Target Module receives a read request, which it latches
2. The PCI Target Module retries all non-register accesses
3. The PCI Target Channel empties the Px-FIFO
4. The QBus Master Module completes the read on the QBus
5. The QBus Slave Module retries all non-register accesses
6. The Qx-FIFO is emptied
7. The PCI Target Module allows the read to complete on the PCI bus; the PCI Target Module allows posted writes to the Px-FIFO (even if the delayed read has not completed); the QBus Slave Module allows posted writes to the Qx-FIFO

Similar principles apply to QBus Slave Channel reads (see page 22-19). The IDMA Channel is independent from the QBus Slave Channel and PCI Target Channel.

2.3.6 QBus Arbitration and Sampling

The QBus Master Module requests the QBus when there is a read request or when there is a sufficient number of entries in the Px-FIFO (see "Acceptance of Burst Writes by the PCI Target Module" on page 2-39).

2.3.6.1 QUICC Bus Arbitration

When the QSpan requires control of the QUICC bus, it requests the bus by asserting Bus Request ($\overline{BR}$). When the QSpan samples Bus Grant ($\overline{BG}$) asserted and Bus Grant Acknowledge ($\overline{BB}$/$\overline{BGACK}$) negated, the QSpan asserts $\overline{BB}$/$\overline{BGACK}$ and negates $\overline{BR}$.

The QBus (QUICC) Master Module's default arbitration mode is asynchronous: it double-samples the $\overline{BG}$ and $\overline{BB}$/$\overline{BGACK}$ inputs using the falling and rising edge of QCLK. The default mode of operation must be altered with the QSpan (CA91C860B, CA91L860B) device. The QSpan (CA91C860B, CA91L860B) device requires that synchronous bus arbitration be utilized when interfaced with the QUICC's arbiter (i.e. S_BG and S_BB bits in the MISC_CTL register is set to "1"). However, the Arbitration Synchronous Timing Mode (ASTM) bit in the QUICC should be set for asynchronous mode of operation (i.e. ASTM bit in the MCR register be set to "0").

The QSpan is able to operate synchronously because all of our timing parameters can be met by the QUICC. However, the QUICC must be programmed for asynchronous mode because the QSpan is unable to meet the QUICC's input set-up requirements.

See Appendix B for the arbitration timing waveform.

2.3.6.2 PowerQUICC Bus Arbitration

When the QSpan requires control of the PowerQUICC bus, it arbitrates for the bus by asserting $\overline{BR}$. When the QSpan samples $\overline{BG}$ asserted and $\overline{BB}$/$\overline{BGACK}$ negated, the QSpan asserts $\overline{BB}$/$\overline{BGACK}$ and negates $\overline{BR}$.

The QSpan will always assert $\overline{BB}$ one clock after $\overline{BG}$ in accordance with MPC860 arbitration requirements.

The PowerQUICC Master Module's default arbitration mode is synchronous. This default mode may be overridden by toggling the S_BG and S_BB bits in the MISC_CTL register (page A-46). (Note that, except for termination signals with a QUICC, and for arbitration, the QBus is always synchronous.)

See Appendix B for the arbitration timing waveform.

2.3.6.3 M68040 Bus Arbitration

When the QSpan requires control of the M68040 bus, it arbitrates for the bus by asserting $\overline{BR}$. When the QSpan samples $\overline{BG}$ asserted and $\overline{BB}$/$\overline{BGACK}$ negated, the QSpan asserts $\overline{BB}$/$\overline{BGACK}$ and negates $\overline{BR}$.

The M68040 Master Module's default operation is synchronous. This default mode may be overridden by toggling the S_BG and S_BB bits in the MISC_CTL register (page A-46). (Note that, except for termination signals with a QUICC, and for arbitration, the QBus is always synchronous.)

See Appendix B for the arbitration timing waveform.

2.3.7 Terminations

This section describes cycle terminations of the QBus Master Module and their relations to terminations generated by the PCI Target module.

2.3.7.1 QBus Master Module Terminations

This section details the QBus Master Module's handling of cycle termination for the QUICC, PowerQUICC, and M68040 buses.

2.3.7.2 QUICC Cycle Terminations

When a transaction is complete, $\overline{BB}/\overline{BGACK}$ is negated by the QBus Master Module on the rising clock edge and tristated on the next falling clock edge. Termination of QUICC cycles is detailed in Table 2.26 below. The QBus Master Module samples all of the QUICC termination signals on the falling edge of QCLK. (In contrast, the QBus Master Module samples the PowerQUICC termination signals on the rising edge of QCLK). The QUICC termination inputs to the QBus Master Module can be skewed by as much as one clock period. However, they all must meet the required setup and hold time required with respect to the falling edge of QCLK. $\overline{HALT}/\overline{TRETRY}$ is ignored if asserted alone. In a Normal & Halt condition, the QBus Master Module delays the termination until $\overline{HALT}/\overline{TRETRY}$ is negated. This feature of the QUICC allows software to verify the internal state of the QUICC during an error. During QUICC Retry terminations the QBus master negates $\overline{BB}/\overline{BGACK}$, and will re-request the bus (assert $\overline{BR}$) when $\overline{HALT}/\overline{TRETRY}$ is negated.

Table 2.26 QUICC Cycle Terminations of QBus Master Module

Termination Type	$\overline{DSACK0}$, $\overline{DSACK1}/\overline{TA}$		$\overline{BERR}/\overline{TEA}$		$\overline{HALT}/\overline{TRETRY}$	
	t_0	t_1	t_0	t_1	t_0	t_1
Normal	Asserted	Asserted	Negated	Negated	Negated	Don't Care
Normal & Halt[a]	Asserted	Asserted	Negated	Negated	Asserted	Asserted
Bus Error	Don't Care Don't Care	Don't Care Don't Care	Asserted Negated	Asserted Asserted	Negated Negated	Negated Negated
Retry	Don't Care Don't Care	Don't Care Don't Care	Asserted Negated	Asserted Asserted	Don't Care Don't Care	Asserted Asserted
t_0: First sample on falling edge of QCLK t_1: Second sample on falling edge of QCLK						

a. QSpan as QBus master will sample DSACKx 2 QCLK rising edges after HALT negation.

2.3.7.3 PowerQUICC Cycle Terminations

When all desired PowerQUICC transfers are complete, $\overline{BB}/\overline{BGACK}$ is negated by the QBus Master Module on the rising clock edge and tristated on the next falling clock edge to terminate the transaction currently in progress.

Termination of cycles is detailed in Table 2.27.

Table 2.27 PowerQUICC Cycle Terminations of QBus Master Module

Termination Type	$\overline{DSACK1}/\overline{TA}$	$\overline{BERR}/\overline{TEA}$	$\overline{HALT}/\overline{TRETRY}$
Normal	Asserted	Negated	Don't Care
Bus Error	Don't Care	Asserted	Don't Care
Retry	Negated	Negated	Asserted

2.3.7.4 M68040 Cycle Terminations

When M68040 transfers complete, $\overline{BB}/\overline{BGACK}$ is negated by the QBus Master Module on the rising clock edge and tristated on the next falling clock edge.Termination of cycles is detailed in Table 2.28 below.

Table 2.28 M68040 Cycle Terminations of QBus Master Module

Termination Type	$\overline{DSACK1}/\overline{TA}$	$\overline{BERR}/\overline{TEA}$
Normal	Asserted	Negated
Bus Error	Negated	Asserted
Retry	Asserted	Asserted

2.3.7.5 Terminations driven by the PCI Target Module

This section lists the terminations generated by the PCI Target Module, and summarizes the conditions under which the various terminations are issued.

The QSpan PCI Target Module generates the following PCI terminations:

- Target-Disconnect
- Target-Retry
- Target-Abort

These terminations are discussed in turn.

During a Target-Disconnect a termination is requested by the target because it is unable to respond within the latency requirements of the PCI specification or it requires a new address phase. This termination is signalled when the target holds both TRDY# and STOP# asserted. Target-Disconnect means that the transaction is terminated after data is transferred. Target-Disconnects may be issued by the QSpan under the following conditions:

- A PCI master attempts to burst to a target image whose transfers have been set to I/O space. In this case, a target-disconnect is issued after the first data phase
- A PCI master attempts to burst to a target image with posted writes disabled and the current data phase - processed as a delayed write - has completed on the QBus. See "Acceptance of Burst Writes by the PCI Target Module" on page 2-39.
- A PCI master has attempted a burst read and the QBus Master Module has completed the single transfer
- A 128-byte boundary is reached
- The Px-FIFO fills during a burst
- A burst transfer requiring non-linear burst address incrementing is attempted.

With a Target-Retry, a termination is requested by the target because it cannot currently process the transaction. This termination is communicated by the target asserting STOP# while not asserting TRDY#. Target-Retry means that the transaction is terminated after the address phase without any data transfer. The PCI Target Module retries accesses under the following conditions:

- An external PCI bus master attempts to post a single write or the first phase of a burst write and the Px-FIFO does not have the number of data entries free that are specified by the cacheline (CLINE[1:0] in the PCI_MISC0 register)
- A delayed transaction is in progress in the PCI Target Channel
- A burst read is requested but the ensuing read has not terminated on the QBus
- A PCI bus master attempts a write through the PCI Target Channel while a read is in progress (in either the QBus Slave Channel or the PCI Target Channel) and that read has not completed on the read-destination bus (i.e., the PCI bus or the QBus, respectively). See "Reads and PCI Transaction Ordering" on page 2-43.

A Target-Abort is issued by a target for a transaction which it will never be able to respond to, or during which a fatal error occurred. This is signalled by the target asserting STOP# and negating DEVSEL#. Although there may be a fatal error for the initiating application, the transaction completes gracefully, ensuring normal PCI operation for other PCI resources.

Except during posted writes (see "Posted Writes" on page 2-48), the termination generated by the PCI Target Module is determined by the termination on the QBus. For read transactions and delayed write transactions, the master is retried until the QBus transaction is complete. Once the QBus transaction is complete, the PCI bus master receives a translated version of the QBus termination.

Table 2.29 below shows how QBus terminations are translated to the PCI bus in the case of delayed transactions. As this table shows, The PCI Target Module generates a Target-Abort when a delayed transfer results in a bus error on the QBus.

Table 2.29 Translation of Cycle Termination[a] from QBus to PCI Bus

QBus Termination Received	PCI Bus Termination Issued
Normal	Master-Completion
Bus Error	Target-Abort
Retry	None [b]

a. This table applies to read transactions and delayed write transactions.

b. These cycles are not translated. The PCI Target Module retries the PCI bus master during delayed transactions until the QBus Master Module receives a normal termination or a bus error.

How target initiated terminations are delivered on the PCI bus is discussed in “Target Initiated Termination” of the PCI 2.1 Specification.

2.3.7.6 Posted Writes

PCI bus terminations of posted writes are not influenced by QBus terminations. If a posted write leads to a bus error on the QBus, this termination is not signalled on the PCI bus to the PCI bus master. However, errors on the QBus are accessible to PCI masters via error logging registers (if error logging is enabled by the EN bit of the QB_ERRCS register). QBus errors can be configured as a source of interrupts.

If the EN bit in the QB_ERRCS register (page A-56) is set, then the QSpan records the address, transaction code, data, and size of a posted write transaction that results in a bus error. In this case, the occurrence of an error is indicated by the ES bit of the QB_ERRCS register. Transfers in the PCI Target Channel are suspended until the ES bit is cleared. The QB_ERRCS register also records the TC and SIZ information of the errored transaction. The address of the errored transaction is latched in the QB_AERR register (page A-57). The data of the errored transaction is latched in the QB_DERR register (page A-58).

If error logging is enabled and the PCI Target Channel is errored, the Px-FIFO is frozen until the ES bit in the QB_ERRCS register is cleared. Posted write operation continues with the next enqueued posted write once the ES bit of the QB_ERRCS is cleared. However, if error logging is not enabled and the PCI Target Channel is errored, the errored transfer is lost and posted write operation continues with the next enqueued transfer.

An interrupt is generated upon the logging of an error (ES bit in QB_ERRCS) only if the QEL_EN bit in INT_CTL register is set (page A-41). If generated, the interrupt is directed to the QBus or the PCI bus, depending on the QEL_DIR bit in the INT_DIR register (page A-43). Interrupts are described in "The Interrupt Channel" on page 2-64.

2.4 The IDMA Channel

The QSpan can be programmed to operate as a QBus IDMA peripheral. Although the QBus Master and Slave mode is determined at reset, this does not affect the QBus Slave Module which dynamically accepts QUICC or PowerQUICC IDMA cycles. The IMODE bit of the IDMA_CS (page A-30) is set by users in order to indicate whether the IDMA channel is to function as a QUICC or a PowerQUICC IDMA peripheral. The QSpan only supports level sensitive handshaking with the PowerQUICC.

The QSpan supports Single Address (Fly-By) and Dual Address IDMA transfers. The QSpan determines the type of IDMA transfer by detecting the state of the $\overline{\text{CSPCI}}$ pin when the IDMA cycle begins. If $\overline{\text{CSPCI}}$ is negated when $\overline{\text{AS}}$ (QUICC Applications) or $\overline{\text{TS}}$ (PowerQUICC Applications) is asserted then Single Address mode is selected. $\overline{\text{CSPCI}}$ must be detected asserted when the cycle begins in order to use Dual Address IDMA transfer mode. This requires that the address programmed in the PowerQUICC's or QUICC's buffer pointer register cause the QSpan's $\overline{\text{CSPCI}}$ chip select to be activated. The QSpan does not latch the address off the QBus during Dual Address IDMA transfers.

The IDMA Channel contains a bi-directional 256-byte (64-entry deep) FIFO called "I-FIFO". IDMA transactions are initiated on the QBus. The IDMA Channel can only access PCI Memory space—it cannot access I/O or Configuration space. The QBus Slave Module accepts IDMA read and write transfers of 16 or 32-bits. The PowerQUICC'S IDMA should be programmed for level-sensitive mode with the QSpan; edge-sensitive mode is not supported.

The rest of this section discusses the following topics concerning the IDMA Channel:

The initialization of this channel is discussed in "IDMA Channel Initialization" on page D-5.

2.4.1 PCI Read Transactions

This section describes the operation and programming of the QSpan to move data from the PCI bus to the QBus using the processor's IDMA. The IDMA registers within the QSpan need to be programmed for a read transaction as follows. These registers are described in Tables A.25-A.27.

- The direction of the transfer should be set for reads (DIR bit in the IDMA_CS register set to 0). The IDMA channel operates in one direction at a time.
- The PORT16 bit of the IDMA_CS register indicates whether IDMA transfers will be 16 or 32-bit on the QBus.
- QUICC users may indicate whether fast termination mode is to be used for dual address or single address IDMA cycle (bits QTERM and STERM, respectively, in the IDMA_CS register).
- The IWM field controls the burst read length on the PCI bus if it is set to a non-zero value. If IWM equals "0", then the QSpan PCI burst read length equals the CLINE setting.
- The IDMA_ADD register contains the absolute PCI address for an IDMA transaction. This number is always aligned to a 4-byte boundary (A1 and A0 always equal "0"). If an IDMA transfer is required to cross an A24 boundary, it must be programmed as two separate transactions.
- The CMD bit in the IDMA_CS register determines whether the read transaction on PCI proceeds as a Memory Read Line transaction or a Memory Read multiple transaction.
- The IDMA_CNT register must be programmed to indicate the amount of data to transfer.

The QUICC's IDMA count register and the QSpan's IDMA_CNT register must be programmed with the same value.

- The CLINE[1:0] field of the PCI_MISC0 register (page A-8) determines how much data is read by the PCI Master Module (either four or eight 32-bit transfers within a burst read if the IWM is set to zero).

Once all the relevant data is programmed into the IDMA register, the GO bit in the IDMA_CS register should be set to 1 to initiate the IDMA transfer. Any status bit (IRST, DONE, IPE, or IDE) affected by a previous transfer should be cleared prior to or while the GO bit is set.

The PCI Master Module performs burst reads on the PCI bus to fill the I-FIFO. When a cacheline of data is available in the I-FIFO, the QSpan asserts $\overline{\text{DREQ}}$ to request IDMA service. The processor acknowledges with $\overline{\text{DACK}}$/$\overline{\text{SDACK}}$, at which point the QSpan drives the data onto the QBus. With a 16-bit QBus port, the QSpan unpacks each 32-bit read data from the PCI into two 16-bit transfers on the QBus.

The IDMA_CNT register indicates the number of bytes to transfer in an IDMA transaction (page A-34). The QSpan does not decrease the transfer count by 4 with every 32-bit transfer on the PCI bus. (The IDMA Channel on the PCI Interface only transfers 32-bit data). The maximum amount of data that can be transferred within an IDMA transaction is 16 MBytes (i.e., 2^{22} 32-bit transfers). The QSpan may pre-fetch data up to the next cacheline boundary and discard the extra data.

When the IDMA_CNT expires, the QSpan sets the DONE bit in the IDMA_CS. The QUICC IDMA asserts the $\overline{\text{DONE}}$ signal when its transfer counter expires. If the QUICC is programmed with a larger transfer count than the QSpan, the QSpan would prematurely assert the DONE bit. The QUICC must reprogram the QSpan's IDMA_CNT register in order to complete the rest of the transaction. If the QUICC is programmed with a smaller transfer count than the QSpan, the QUICC would assert the $\overline{\text{DONE}}$ signal when its IDMA count reached zero, causing the QSpan to negate $\overline{\text{DREQ}}$. The QUICC would then have to assert the IRST_REQ bit in the IDMA_CS register to reset the QSpan's IDMA Channel.

During PowerQUICC IDMA transfers, the QSpan ignores the $\overline{\text{DONE}}$ signal.

2.4.2 PCI Write Transactions

This section describes the operation and programming of the QSpan to move data from the QBus to the PCI bus using the QUICC or PowerQUICC IDMA. IDMA registers need to be programmed for a write transaction as follows. These registers are tabulated in Tables A.25-A.27.

- The direction of the transfer must be set for writes (DIR bit in the IDMA_CS register set to "1"). The IDMA channel operates in one direction at a time.
- QUICC users may indicate whether fast termination mode is to be used for dual address or single address IDMA cycle (bits QTERM and STERM, respectively, in the IDMA_CS register).
- The PORT16 bit of the IDMA_CS indicates whether IDMA transfers will be 16-bit or 32-bit on the QBus.
- The IWM field of the IDMA_CS register determines when the QSpan will begin to burst the data onto the PCI bus. (Once the I-FIFO contents equal IWM, the QSpan burst writes up to the values of IWM - throttled only by the PCI target). The IWM must not be programmed with a value greater than the IDMA transfer byte count.

- The IDMA_ADD register (page A-33) contains the absolute PCI address for an IDMA transaction. This number is always aligned to a 4-byte boundary. An IDMA transfer wraps-around at the A24 boundary. If an IDMA transfer is required to cross an A24 boundary, it must be programmed as two separate transactions. The IWM must not be programmed with a value greater than the IDMA transfer byte count.
- The IDMA_CNT register indicates the number of bytes to transfer in an IDMA transaction (page A-34).
- The CMD bit in the IDMA_CS register determines whether the write transaction on PCI proceeds as a Memory Write Invalidate or a Memory Write transfer.
- The CLINE[1:0] field of the PCI_MISC0 register (page A-8) determines the length of the burst writes initiated by the PCI Master Module (either four or eight 32-bit transfers within a burst write if the IWM is set to zero).

Once all the relevant data is programmed in the IDMA register, the GO bit in the IDMA_CS register should be set to 1 to initiate the IDMA transfer. Any status bit (IRST, DONE, IPE, or IQE) affected by a previous transfer should be cleared prior to or while the GO bit is set.

The QSpan requests data from the IDMA by asserting $\overline{\text{DREQ}}$. By asserting $\overline{\text{DACK}}$/$\overline{\text{SDACK}}$, the IDMA acknowledges that data is being written to the I-FIFO. With a 16-bit QBus port, the QSpan packs two 16-bit transfers from the QBus into one 32-bit entry in the I-FIFO. When the I-FIFO is full the QSpan negates $\overline{\text{DREQ}}$.

The QSpan requests the PCI bus when there is as much data in the I-FIFO as is specified by the IWM field of the IDMA_CS register (if the IWM equals “0”, the QSpan requests the PCI bus when a cache line is queued in the I-FIFO). The QSpan burst-writes data to the PCI target.

The IDMA_CNT register indicates the number of bytes to transfer in an IDMA transaction (page A-34). The QSpan does not decrease the transfer count by 4 with every 32-bit transfer on the PCI bus. (The IDMA Channel on the PCI Interface transfers 32-bit data). The amount of data that can be transferred within an IDMA transaction is 16 MBytes (i.e., 2^{22} 32-bit transfers).

When the IDMA_CNT expires, the QSpan sets the DONE bit in the IDMA_CS register. The QUICC IDMA asserts the $\overline{\text{DONE}}$ signal when its transfer counter expires. If the QUICC is programmed with a larger transfer count than the QSpan, the QSpan would prematurely assert the DONE bit. This would cause the QUICC to reprogram the QSpan’s IDMA_CNT register in order to complete the rest of the transaction. If the QUICC is programmed with a smaller transfer count than the QSpan, the QUICC would assert the $\overline{\text{DONE}}$ signal when its IDMA count reached zero, causing the QSpan to negate $\overline{\text{DREQ}}$. The QUICC would then have to assert the IRST_REQ bit in the IDMA_CS register to reset the QSpan’s IDMA Channel.

During PowerQUICC IDMA transfers, the QSpan ignores the $\overline{\text{DONE}}$ signal.

2.4.3 TC[3:0] Encoding with PowerQUICC IDMA

If the PowerQUICC's I/O Port pins are being shared between multiple functions (e.g., IDMA and Ethernet) then the TC[3:0] decoding should be implemented. This allows pheripheral devices (QSpan or Ethernet devices) to determine when they are involved in a transaction.

The QSpan's TC[3:0] inputs may be used with $\overline{\text{SDACK}}$ to decode IDMA transactions. The value that is decoded is programmed by the user through the TC[3:0] field in the IDMA_CS register. The QSpan will only decode the TC[3:0] lines for IDMA transactions if the TC_EN bit in the IDMA_CS register is set to 1. If the TC_EN bit is set and an IDMA transfer is attempted where the TC[3:0] input does not match the TC[3:0] IDMA_CS field, the QSpan will not accept the IDMA transaction.

The PowerQUICC always drives the terminal count code AT[0:3] = 0xf on the last cycle of the IDMA transfer to the peripheral. Therefore, this requires the MPC860's IDMA Function Code Register (SFCR or DFCR) bits AT[1-3], and the QSpan's IDMA_CS register bits TC[3:0] to be programmed with all "1"s to make use of this feature. If the PowerQUICC is not programmed to drive AT[0:3] = 0xf and the QSpan is not set to decode TC[3:0]=0xf then the IDMA transfer will not complete successfully when this feature is enabled.

This manual adopts the convention that the most significant bit is always the largest number. PowerQUICC designers must ensure that they connect their pins accordingly: e.g., pin A[31] on the QSpan connects to pin A[31] on the QUICC bus, but connects to pin A[0] on the PowerQUICC bus. This applies to all PowerQUICC buses (D[31:0], AT[3:0], TSIZ[1:0]) not only the address bus.

2.4.4 IDMA Status Tracking

A number of bits in the IDMA_CS register record the status of the transaction.

- The IDMA ACT bit in the IDMA_CS is set by the QSpan to indicate that an IDMA transfer is in progress.
- The DONE bit indicates that the IDMA transfer is complete.
- The IPE status bit is asserted when an error is signalled on the PCI bus during an IDMA transfer.
- The IQE status bit is asserted when an error is signalled on the QBus during an IDMA transfer.

- The IRST status bit is asserted when the QSpan has reset the IDMA Channel.

Errors and resets are described in "IDMA Errors, Resets and Interrupts" on page 2-54.

2.4.5 IDMA Errors, Resets and Interrupts

This section describes how the QSpan responds to PCI bus errors and QBus errors during IDMA reads and writes; it also describes IDMA resets and interrupts. When there is an error on either bus during IDMA transfers, the QSpan will assert an IDMA error status bit. If the error is on the PCI bus, then the IPE bit in the IDMA Control and Status Register is set (page A-30). If the error is on the QBus, then the IQE bit is set. How the transfer proceeds following the error depends on whether the transfer is a read or a write, and on what bus the error occurred. Table 2.30 summarizes the sequence of events following bus errors for each of these four cases.

Table 2.30 The QSpan's Response to IDMA Errors

Transfer	PCI Bus Error	QBus Error
Read	Pre-fetching stops. IPE is asserted in the IDMA_CS register. Then the QSpan negates $\overline{\text{DREQ}}$. If enabled, an interrupt is generated on the PCI bus or the QBus.	The QSpan negates $\overline{\text{DREQ}}$. Transfers on QBus stop. IQE is asserted in the IDMA_CS register. If enabled, an interrupt is generated on the PCI bus or the QBus. Pre-fetching stops when the current IDMA PCI transfer (if any) is complete.
Write	PCI writes from I-FIFO are halted. IPE is asserted in the IDMA_CS register. If enabled, an interrupt is generated on the PCI bus or the QBus. The QSpan negates $\overline{\text{DREQ}}$.	The QSpan negates $\overline{\text{DREQ}}$. IQE is asserted in the IDMA_CS register. If enabled, an interrupt is generated on the PCI bus or the QBus. If a PCI transfer from the I-FIFO is in progress, any complete CLINE of data in the I-FIFO is transferred to the PCI target. (See below)

If a QBus bus error occurs during an IDMA write transfer, the QSpan continues to write data until the IWM level is reached. This is possible because the QSpan only initiates PCI activity once the IWM value is queued in the I-FIFO. Once the IWM amount is transferred, the QSpan responds to the bus error on the QBus as indicated in Table 2.30.

The assertion of IRST, IPE, IQE and DONE can be mapped to the interrupt pins on either bus using the QSpan's Interrupt Control Register (bits IRST_EN, IPE_EN, IQE_EN and DONE_EN, page A-41). The bus that is interrupted depends on the Interrupt Direction Register INT_DIR (page A-43). The status of the individual interrupt sources can be determined by reading the corresponding status bit (see Table 2.41 and page A-39). To clear the interrupt, the original interrupt source must be cleared. For example, to clear the IDMA Reset Interrupt, the IRST bit in the IDMA_CS register must be cleared (page A-30). The

following table is extracted from "The Interrupt Channel" on page 2-64.

Table 2.31 IDMA Interrupt Source, Enabling, Mapping, Status and Clear bits

Source	Source Bits IDMA_CS (page A-30) Write 1 to clear	Enable Bits INT_CTL (page A-41)	Mapping Bits INT_DIR (page A-43)	Interrupt Status Bits INT_STAT (page A-39)
IDMA QBus Error	IQE in IDMA_CS	IQE_EN	IQE_DIR	IQE_IS
IDMA PCI Bus Error	IPE in IDMA_CS	IPE_EN	IPE_DIR	IPE_IS
IDMA Reset	IRST in IDMA_CS	IRST_EN	IRST_DIR	IRST_IS
IDMA Done	DONE in IDMA_CS	DONE_EN	DONE_DIR	DONE_IS

An IDMA transfer that is halted due to an IDMA error (IPE or IQE asserted) will not resume once these error conditions are cleared. The IDMA channel needs to be reset by setting the IRST_REQ bit of the IDMA_CS (page A-30). The IRST status bit is set when the QSpan has reset the IDMA Channel. Then a new IDMA transfer can be programmed (either from where the error happened or the previous transfer can be attempted again). The QBus Slave Channel is not affected by IDMA Channel errors.

The IDMA Channel can be reset by setting the IRST_REQ bit in the IDMA_CS register, depending on the value of the ACT bit in the IDMA_CS register. If the ACT bit is “0”, then setting IRST_REQ to “1” has no effect. If the “ACT’ bit is “1”, then setting the IRST_REQ bit has the following effects:

1. The QSpan negates $\overline{\text{DREQ}}$, which halts transfers on the QBus.
2. If the QSpan is writing IDMA data on the PCI bus, the QSpan will terminate the transfer by negating FRAME# at the next cacheline; if the QSpan is reading data, FRAME# is negated immediately.
3. The QSpan flushes the I-FIFO (after negating FRAME#).
4. The ACT bit in the IDMA_CS register is set to “0” while the IRST bit is set to “1”. Note that the IDMA_ADD register, the IDMA_CNT register and the rest of the IDMA_CS register are not reset.
5. If enabled, an interrupt will be generated on the QBus or the PCI bus (see "The Interrupt Channel" on page 2-64).

2.4.6 IDMA Endian Issues

The PCI bus and the Motorola processors have some differences in the way they order and address bytes. These differences are explained in Appendix A. The present section describes how the QSpan translates cycles from the QBus to the PCI bus in the IDMA Channel.

The PCI bus is always a Little-Endian environment. The QBus may be configured as Little-Endian or Big-Endian, depending on the value of the QBus Byte Ordering Control bit (QB_BOC) in the MISC_CTL register (page A-46). The default mode for the QBus is Big-Endian. The QSpan translates byte lane ordering when the QBus is Big-Endian, while preserving the addressing of bytes. When the QBus is Little-Endian (according to QB_BOC), the QSpan preserves byte lane ordering, while translating the addressing of bytes. Note that the QB_BOC bit affects transactions in all channels.

Tables 2.32 to 2.35 below describe 16-bit and 32-bit cycle mapping for Little-Endian and Big-Endian modes.

Table 2.32 below describes mapping of 16-bit QBus transactions in Little-Endian mode. The byte lane ordering is preserved in Little-Endian mode. During a read transaction, a full 32-bit PCI transaction is unpacked into two 16-bit QBus transfers; during a write transaction, two 16-bit QBus transactions are packed into the I-FIFO for one PCI transfer. The table also shows the order in which the 16-bit QBus cycles appear.

Table 2.32 16-Bit Little Endian IDMA Cycle Mapping

	QBus			PCI bus	
QBus Timing	**SIZ[1:0]**	**A[1:0]**	**D[31:0]**	**BE[3:0]#**	**D[31:0]**
First 16-bit transfer:	10	00	B3 B2 xx xx	**Full 32-bit PCI bus Transfer**	
Second 16-bit transfer:	10	10	B1 B0 xx xx	0000	B3 B2 B1 B0

Table 2.33 below describes mapping of 16-bit QBus transactions in Big-Endian mode. The addressing of bytes is preserved in Big-Endian mode. During a read transaction, a full 32-bit PCI transaction is unpacked into two 16-bit QBus transfers; during a write transaction, two 16-bit QBus transactions are packed into the I-FIFO for one PCI transfer. The table also shows the order in which the 16-bit QBus cycles appear.

Table 2.33 16-Bit Big-Endian IDMA Cycle Mapping

	QBus			PCI bus	
QBus Timing	**SIZ[1:0]**	**A[1:0]**	**D[31:0]**	**BE[3:0]#**	**D[31:0]**
First 16-bit transfer:	10	00	B0 B1 xx xx	**Full 32-bit PCI bus Transfer**	
Second 16-bit transfer:	10	10	B2 B3 xx xx	0000	B3 B2 B1 B0

Table 2.34 below describes mapping of 32-bit QBus transactions in Little-Endian mode. The byte lane ordering is preserved in Little-Endian mode.

Table 2.34 32-Bit Little-Endian IDMA Cycle Mapping

QBus			PCI bus	
SIZ[1:0]	A[1:0]	D[31:0]	BE[3:0]#	D[31:0]
00	00	B3 B2 B1 B0	0000	B3 B2 B1 B0

Table 2.35 below describes mapping of 32-bit QBus transactions in Big-Endian mode. The addressing of bytes is preserved in Big-Endian mode.

Table 2.35 32-Bit Big-Endian IDMA Cycle Mapping

QBus			PCI bus	
SIZ[1:0]	A[1:0]	D[31:0]	BE[3:0]#	D[31:0]
00	00	B0 B1 B2 B3	0000	B3 B2 B1 B0

This manual adopts the convention that the most significant bit is always the largest number. PowerQUICC designers must ensure that they connect their pins accordingly: e.g., pin A[31] on the QSpan connects to pin A[31] on the QUICC bus, but connects to pin A[0] on the PowerQUICC bus. This applies to all PowerQUICC buses (D[31:0], AT[3:0], TSIZ[1:0]) not only the address bus.

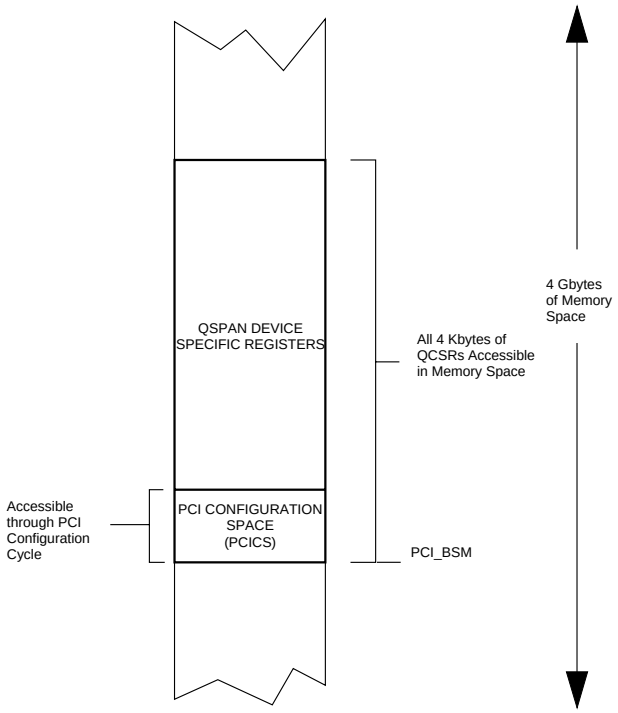
2.5 The Register Channel

The 4 Kbytes of QSpan Control and Status Registers (QCSRs) are used to program PCI settings as well as the QSpan's operating parameters (see Figure 2.6). The QCSRs are functionally divided into two groups: the PCI configuration Registers and the QSpan Device Specific Registers. QCSR space is accessible from both the PCI bus and the QBus.

Since registers of the QSpan can be accessed from either the PCI bus or the QBus, an internal arbitration occurs to indicate ownership. The access mechanisms (including arbitration protocol) for the QCSRs differ depending on whether the registers are accessed from the PCI bus or the QBus (see the next two sections). An internal pointer selects which bus can access the registers. Default ownership of the register channel is granted to the QSpan's PCI Target Module. When ownership of the register channel is granted to the QBus Slave Module, register accesses from the PCI bus are retried.

The processor on the QBus should not poll the QSpan registers within a tight loop (i.e. back-to-back accesses should not occur within 32 clock cycles), if the status bit the processor is polling is being controlled by a PCI register access. If quick back-to-back accesses were to occur, then PCI bus agents would not be allowed access to the QSpan registers. It is possible for the PCI bus to remain being retried forever or a system RETRY counter to be exceeded

Figure 2.6 .QSpan Control and Status Registers

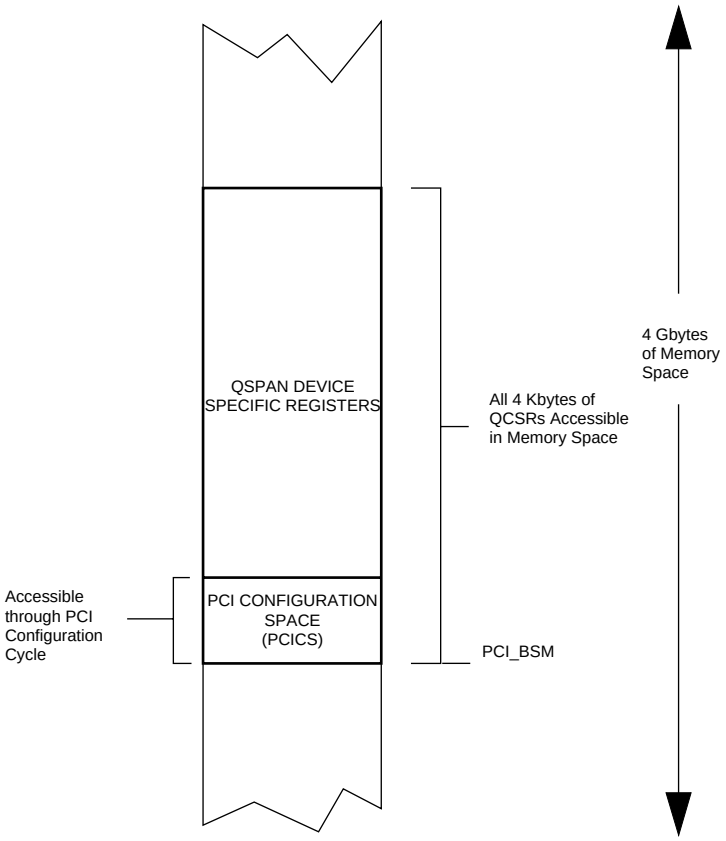


2.5.1 Register Access from the PCI Bus

The QCSRs may be accessed through configuration cycles or in Memory space (see Figure 2.7). These accesses are discussed in turn.

Default ownership of the Register Channel is granted to the PCI Target Module. If an external master on the PCI bus attempts to access the registers of the QSpan, and the ownership has been granted to the QBus Slave Module, the transfer will be retried on the PCI bus.

Figure 2.7 PCI Access to QCSR in Memory Space



Only the PCI configuration registers are accessible through PCI configuration Type 0 cycles. PCI configuration registers (in the lower 256 bytes of QCSR space) are accessible in PCI Configuration space. On its PCI Interface, the QSpan decodes AD[7:2] for accesses to its PCI configuration registers. IDSEL must be asserted for Type 0 configuration access. Configuration access from the QBus is discussed on page 22-62.

The QSpan registers may be accessed in Memory space but not in I/O space.In order to access QCSRs from PCI bus Memory space, it is necessary to set the Memory Space bit in the PCI_CS register to “1” (page A-5). (This is required for any PCI Target Module access, including register accesses.) The PCI_BSM’s SPACE bit is fixed at logic “0” so that registers can be accessed in PCI Memory space. The BA[31:12] field of the PCI_BSM register specifies the base address in memory space of the QCSRs. The QCSRs are located in Memory space as address offsets of this base address (see Figure 2.7).(On the PCI interface, the QSpan issues a Target-Disconnect after the first data phase.)

There is a direct mapping between values on the AD[11:0] lines and the register offsets. Table 2.36 illustrates PCI Memory cycle access to bits 15-08 of the PCI_CLASS register (page A-7). This table shows the AD[11:0] and C/BE#[3:0] signals required to access byte 1; it also shows how the data would be presented to the PCI master.

Table 2.36 PCI Memory Cycle Access to bits 15-08 of the PCI_CLASS register

AD[11:0] (register offset)	BE[3:0]#	AD[31:0][a]
0x008	1101	xx xx B1 xx

a. Data phase.

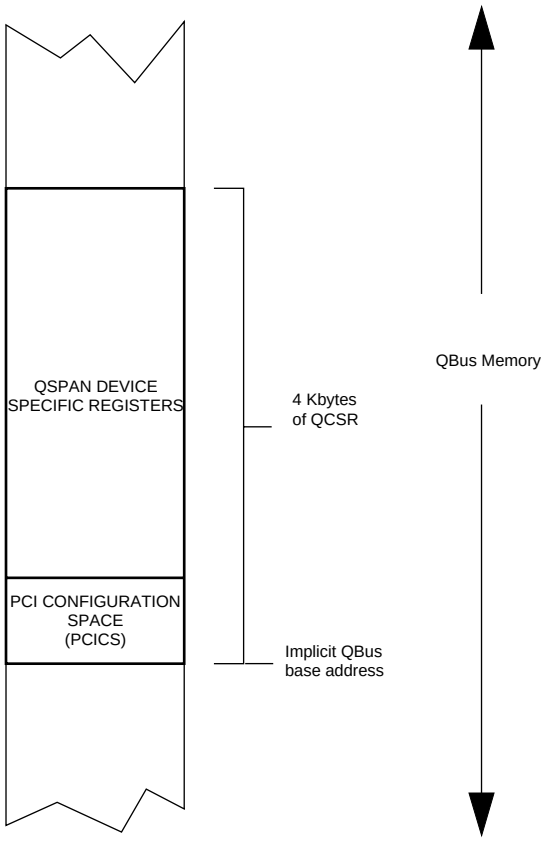
2.5.2 Register Access from the QBus

The QSpan’s registers can be selected by an external master from the QBus with the $\overline{\text{CSREG}}$ chip-select pin. Since the QSpan registers span 4K, only the lower 12 bits of the QBus address are used (see Figure 2.8). Register accesses must be performed as 32-bit data transfers.

If an external master on the QBus attempts to access the registers of the QSpan, the transfer will be retried on the QBus and the QSpan will make an internal request for register ownership by the QBus. This request will be removed if no register access is attempted within 2^{10} QCLK clock cycles. The request will also be removed if a register access completes on the QBus without a subsequent register access beginning within 32 QCLK clock cycles of the completion of the previous register access.

The QSpan registers do not support burst accesses. If a burst to register space is attempted from the QBus, a bus error will be issued.

Figure 2.8 QCSR Access from the QBus



2.5.2.1 Examples of QBus Register Accesses

Table 2.37 illustrates Big-Endian access to bits 15-08 of the PCI_CLASS register (Table A.4). This table shows the address and size signals required to access this byte; it also shows how the data would be presented to the QBus Master.

Table 2.37 Big-Endian QBus Access to bits 15-08 of the PCI_CLASS register

A[11:0]	SIZ[1:0]	A[1:0][a]	D[31:0]
0x00A	01	10	xx xx B2 xx

a. This is a subset of A[11:0].

Table 2.38 illustrates Little-Endian access to bits 15-08 of the PCI_CLASS register. There is no real difference between Big- and Little-Endian access to the QSpan's register. The only difference between Table 2.37 and Table 2.38 is the name (not the location) of the byte along the data lines.

Table 2.38 Little-Endian QBus Access to bits 15-08 of the PCI_CLASS register

A[11:0]	SIZ[1:0]	A[1:0][a]	D[31:0]
0x00A	01	10	xx xx B1 xx

a. This is a subset of A[11:0].

2.5.2.2 PCI Configuration Cycles Generated from the QBus

PCI configuration cycles of Type 0 or Type 1 may be initiated from the QBus. To accomplish this:

1. Write the target PCI address on the Configuration Address Register (CON_ADD, page A-35).
2. Access the Configuration Data register (page A-37).

The first step determines how the address of the configuration cycle is generated on the PCI bus. The second step causes the configuration cycle to occur on the PCI bus. The following subsections describe these two aspects of configuration cycles.

2.5.2.2.1 Address Phase of PCI Configuration Cycles

The type of cycle that is generated on the PCI bus (Type 0 or Type 1) is determined by the TYPE bit of the CON_ADD register (programmed in Step 1 above). This determines how the address of the configuration cycle is generated on the PCI bus. When the QSpan is being used as a host bridge, the MA_BE_D bit in the MISC_CTL register should be set to a "1" in order for the QSpan to comply with the PCI 2.1 Specification. Setting this bit to "1" allows the QSpan to signal a successful configuration cycle to the host processor, even if a Master-Abort or Target-Abort occurs on the PCI bus.

With the TYPE bit set to 1, an access of the CON_DATA register from the QBus interface performs a corresponding Configuration Type 1 cycle on the PCI bus. During the address phase of the Configuration Type 1 cycle on the PCI bus, the PCI address lines carry the values encoded in the CON_ADD register (AD[31:0] = CON_ADDR[31:0]).

With the TYPE bit set to 0, an access of the CON_DATA register from the QBus interface performs a corresponding Configuration Type 0 cycle on the PCI bus. Programming the Device Number causes one of the upper address lines, AD[31:16] to be asserted during the address phase of the Configuration Type 0 cycle; the other lines are negated. (Table 2.39 shows which PCI address line is asserted as a function of the DEV_NUM[3:0] field.) The remaining address lines during the address phase of the Configuration cycle are controlled by the Function Number and Register Number fields of the CON_ADD register:

- AD[15:11] = 00000
- AD[10:8] = FUNC_NUM[2:0]
- AD[7:2] = REG_NUM[5:0]
- AD[1:0] = 00.

Table 2.39 PCI AD[31:16] lines asserted as a function of DEV_NUM field

DEV_NUM[3:0]	AD[31:16]
0000	0000 0000 0000 0001
0001	0000 0000 0000 0010
0010	0000 0000 0000 0100
0011	0000 0000 0000 1000
0100	0000 0000 0001 0000
0101	0000 0000 0010 0000
0110	0000 0000 0100 0000
0111	0000 0000 1000 0000
1000	0000 0001 0000 0000
1001	0000 0010 0000 0000
1010	0000 0100 0000 0000
1011	0000 1000 0000 0000
1100	0001 0000 0000 0000
1101	0010 0000 0000 0000
1110	0100 0000 0000 0000
1111	1000 0000 0000 0000

2.5.2.2.2 Data Phase of PCI Configuration Cycles

PCI configuration accesses from the QBus proceed as delayed transactions. This holds for configuration reads as well as writes. When the CON_DATA register is accessed, the QBus master is retried. The QSpan then initiates a configuration cycle on the PCI bus.

The PCI byte enable driver is dependent on the attributes of the QBus cycle (i.e. QBus address and SIZ signals) during the configuration cycle. Until the configuration cycle completes on the PCI bus any further configuration cycle attempt is retried. In the case of a read (read to CON_DATA), the data from the PCI bus is stored in the CON_DATA register. After the configuration cycle completes on the PCI bus, when the QBus master attempts to access the CON_DATA register at the same address, the cycle completes successfully on the QBus. In the case of a write (write to CON_DATA) the cycle completes on the QBus after the configuration write completes on the PCI bus.

2.5.2.3 Interrupt Acknowledge Cycle

A mechanism is provided for a QBus register read to generate a PCI Interrupt Acknowledge cycle (see "Interrupt Acknowledge Cycle" on page 2-68.)

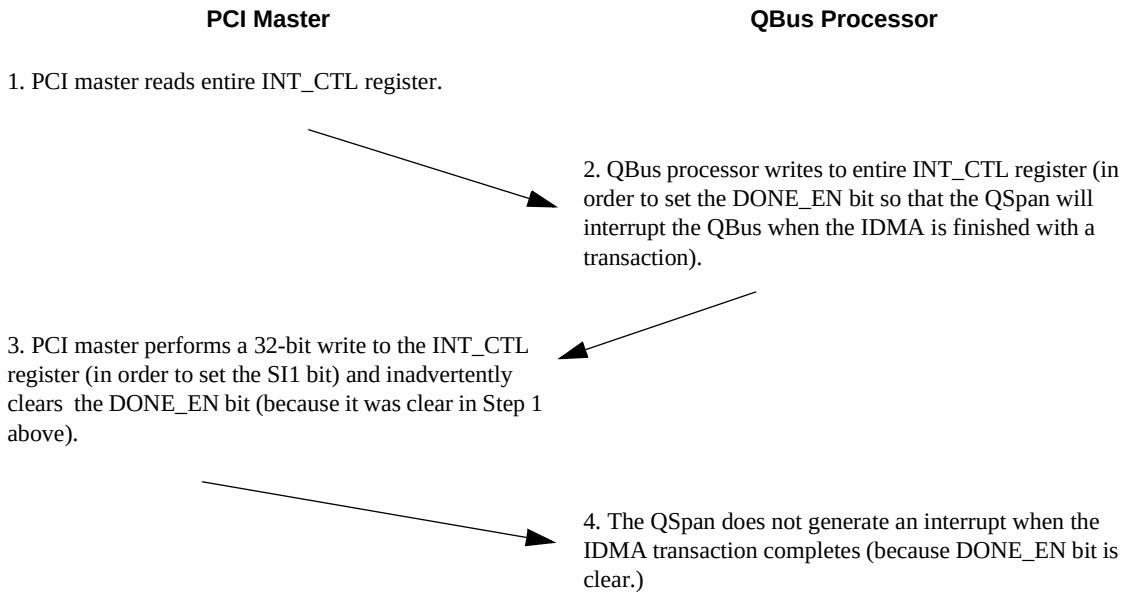
2.6 The Interrupt Channel

Through the interrupt channel, certain hardware and software events can trigger interrupts on the QBus or the PCI bus. Two bidirectional interrupt pins are provided: INT# and $\overline{\text{QINT}}$.

2.6.1 Register Access Synchronization

The fact that the Tundra QSpan allows (non-simultaneous) access to its registers from the QBus Interface and the PCI Interface presents a synchronization issue. The QSpan does not have the ability to lock out register accesses from one interface so that accesses can be performed on the other interface. Nor does the QSpan have semaphore capabilities. If a master on one interface is executing a test and set procedure (reading from a register and then setting part or all of the register) there is the possibility that between the test and the set, a master on the other interface sets the same register. The issue is most relevant to the Interrupt Control register (page A-41).

Figure 2.9 Example of a Register-Access Synchronization Problem.



Two possible solutions to this problem are:

1. Perform byte-wide writes on the PCI bus as opposed to 32-bit writes. This does not solve all the register-access synchronization problems, since the other 7 bits in the write may still change between a read and a write. Since the SI1 bit and the DONE_EN bit are separated by two bytes, it would solve the problem described in Figure .
2. The system could be designed to always set the DONE_EN bit (or whatever other bit is susceptible to this problem). However, this may lead to the generation of more interrupts than one would like to handle, and consequently impact on performance.

2.6.2 Hardware Triggered Interrupts

In order for an input to trigger an interrupt on the opposite interface, the corresponding enable bit must be set in the INT_CTL register. For example, for INT# to trigger $\overline{\text{QINT}}$, the INT_EN bit must be set. The status of the interrupt is logged in the INT_STAT register.

It is only possible to route interrupts in one direction at a time: i.e., it is not possible to allow PCI interrupt sources to be mapped to $\overline{\text{QINT}}$ while allowing $\overline{\text{QINT}}$ sources to be mapped to INT#.

Table 2.40 Mapping of Hardware-Initiated Interrupts

Input	Enable bits (INT_CTL page A-39)	Status Bits (INT_STAT page A-38)	Output
INT#	INT_EN	INT_IS	$\overline{\text{QINT}}$
PERR#	PERR_EN	PERR_IS	
SERR#	SERR_EN	SERR_IS	
$\overline{\text{QINT}}$	QINT_EN	QINT_IS	INT#

If INT_EN is set, then the assertion of INT# causes $\overline{\text{QINT}}$ to be asserted until INT# is negated and the INT_IS bit is cleared. Because INT# is level sensitive, if the INT_IS bit is cleared before INT# is negated, $\overline{\text{QINT}}$ will remain asserted.

In order to negate $\overline{\text{QINT}}$ when its assertion is due to the assertion of PERR# or SERR#, the following procedure should be followed:

1. Clear the error status bit in the source PCI device.
2. Clear the status bit in the INT_STAT register.

To negate INT# when its assertion is due to the assertion of $\overline{\text{QINT}}$, negate $\overline{\text{QINT}}$ and then clear the QINT_IS bit in the INT_STAT register.

2.6.3 Software Triggered Interrupts

The QSpan can generate interrupts based upon internal events provided that the interrupt source is enabled in the INT_CTL register (page A-41). Interrupts can be mapped to an interrupt output pin on the PCI bus (INT#) or the QBus ($\overline{\text{QINT}}$) depending on the value of the interrupt mapping bit in the INT_DIR register (page A-43). The status of the individual interrupt sources can be determined by reading the corresponding status bit (page A-39). To clear the interrupt, the original interrupt source must be cleared. For example, to clear the IDMA Reset Interrupt, the IRST bit in the IDMA_CS register must be cleared (page A-30).

Table 2.41 Interrupt Source, Enabling, Mapping, Status and Clear bits

Source	Source Bits Write 1 to clear	Enable Bits INT_CTL (page A-41)	Mapping Bits INT_DIR (page A-43)	Interrupt Status Bits INT_STAT (page A-39)
QBus Slave Channel Errors (originating on the PCI bus)				
Data Parity Error	DP_D[a] in PCI_CS (page A-5)	DPD_EN	DPD_DIR	DPD_IS
PCI Bus Target Channel Errors (originating on the QBus)				
QBus Error	ES in QB_ERRCS (page A-56)	QEL_EN	QEL_DIR	QEL_IS
IDMA Channel Events[b]				
IDMA QBus Error	IQE in IDMA_CS (page A-30)	IQE_EN	IQE_DIR	IQE_IS
IDMA PCI Bus Error	IPE in IDMA_CS (page A-30)	IPE_EN	IPE_DIR	IPE_IS
IDMA Reset	IRST in IDMA_CS (page A-30)	IRST_EN	IRST_DIR	IRST_IS
IDMA Done	DONE in IDMA_CS (page A-30)	DONE_EN	DONE_DIR	DONE_IS

a. DP_D is set if the PERR# is asserted (by the QSpan as master, or by an external target while the QSpan is master) and the PERESP bit in the PCI_CS register is set.

b. See also "IDMA Errors, Resets and Interrupts" on page 2-54.

Four software interrupt bits are provided (Software Interrupt 0 through 3). Setting a software interrupt bit (SI3, SI2, SI1, or SI0) triggers the interrupt status (see Table 2.42) and causes the QSpan to generate an interrupt on the QBus ($\overline{\text{QINT}}$) or PCI bus (INT#), depending on the relevant mapping bit. There is no enable bit for software interrupts. The interrupt line will be driven until the status bit is cleared (writing 1 to the status bit clears the interrupt).

Table 2.42 Software Interrupt Mapping, Status and Source bits

Source	Source Bits (INT_CTL page A-41) (INT_CTL2 page A-45)	Mapping Bits (INT_DIR page A-43)	Interrupt Status Bits[a] (INT_STAT page A-39)
Software Interrupt 0	SI0	SI0_DIR	SI0_IS
Software Interrupt 1	SI1	SI1_DIR	SI1_IS
Software Interrupt 2	SI2	SI2_DIR	SI2_IS
Software Interrupt 3	SI3	SI3_DIR	SI3_IS

a. Write 1 to clear

Interrupts in one channel do not affect processing in the other channel.

2.6.4 Interrupt Acknowledge Cycle

Reading the IACK_GEN register (page A-38) from the QBus causes an IACK cycle to be generated on the PCI bus. The byte lanes enabled on the PCI bus are determined by SIZ[1:0] and A[1:0] of the QBus read. The address on the QBus used to access the IACK_GEN register is passed directly over to the PCI bus during the PCI IACK cycle. However, address information is ignored during PCI IACK cycles, so this has no effect.

Reads from this register behave as delayed transfers—the QBus master is retried until the read data is latched from the PCI target. When the IACK cycle completes on the PCI bus, the data is latched into the IACK_GEN register, which is returned as read data when the QBus master attempts the cycle again.

Writing to this register from the QBus or PCI bus has no effect. Reads from the PCI bus return all zeros.

2.7 The EEPROM Channel

Certain registers of the QSpan can be programmed by data in an EEPROM at system reset. This allows board designers to set unique identifiers for their cards on the PCI bus at reset, to enable the PCI Bus Expansion ROM Control Register, and set various address and parameters of images. Configuring the QSpan with the EEPROM allows the QSpan to boot-up as a Plug ‘n Play compatible device.

The QSpan supports reads from and writes to the EEPROM.

256 bytes of data can be accessed with the EEPROM, however the QSpan only loads the first 12 bytes of data for its own programming. The additional 244 bytes are available for other purposes.

2.7.1 EEPROM Configuration and Plug ‘n Play Compatibility

There are two ways which the EEPROM may be configured to allow the QSpan to boot as a PCI Plug ‘n Play compatible device.

1. The EEPROM can be configured before it is placed on the board.
2. The EEPROM may be configured after it is installed. In this case, the first time the QSpan-based board boots, it will not be PCI Plug ‘n Play compatible. One would then need to program the EEPROM from either the QBus or the PCI bus (see "Programming the EEPROM from the QBus or PCI Bus" on page 2-73 for details).

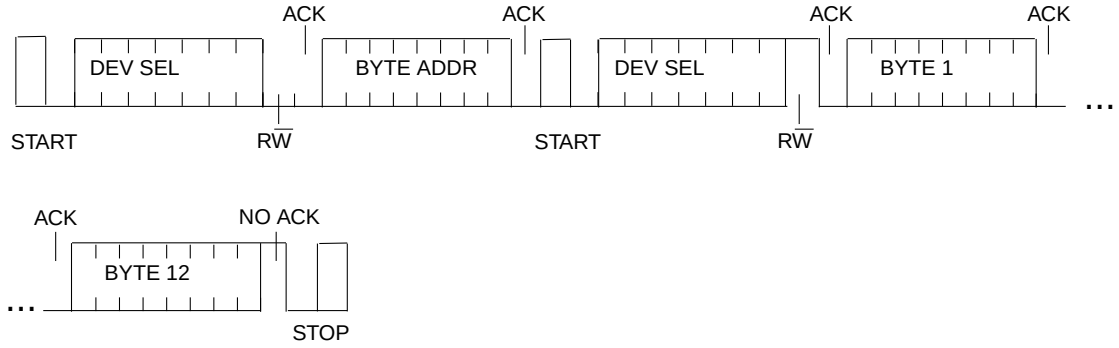
The following subsections discuss the implementation of the EEPROM.

2.7.2 EEPROM I^2C protocol

The QSpan supports the 2-wire I^2C protocol, using a clock output (SCL) and a bi-directional signal (SDA). If the SDA or ENID pin is asserted as a logic “1” during a PCI bus Reset (i.e. RST# active), then at the conclusion of this reset the QSpan reads 12 bytes of data from the EEPROM. This read is performed as a Sequential Read. After the START condition the QSpan puts out a 7-bit device select code of 1010000b. The four most significant bits of the device select code for the I^2C protocol are required to be 1010b. The following three bits are chip-enable signals that are 000. The chip-enable lines on the EEPROM should be tied low. The QSpan expects the 12 bytes of data delivered from the EEPROM starting at address zero. Figure 2.10 below shows this Sequential Read transaction.

After reset the EEPROM port does a STOP then a START before loading from the EEPROM.

Figure 2.10 Sequential Read from EEPROM



While the registers are being loaded from the EEPROM, all accesses to the QSpan by an external PCI bus master are terminated with a retry. During this period, write accesses to the EEPROM programmable registers from the QBus have no effect, and reads return all zeros.

Some EEPROM devices require a write control signal which should not be pulled inactive if the intention is to program the EEPROM device with the QSpan.

If $\overline{RESETI}$ is asserted while the QSpan is reading from the EEPROM then the EEPROM's contents may not be loaded correctly.

2.7.3 Mapping of EEPROM Bits to QSpan Registers

This section describes the mapping between EEPROM bits and the QSpan's registers. The following registers may be programmed by the EEPROM:

- PCI Configuration Subsystem ID Register (PCI_SID register)
- PCI Expansion ROM (PCI_BSROM and PBROM_CTL registers)
- PCI Bus Target Image 0 (PCI_BST0 and PBTI0_CTL registers)
- PCI Bus Target Image 1 (PCI_BST1 and PBTI1_CTL registers)
- QBus Slave Image 0 (QBSI0_CTL and QBSI0_AT registers)

Table 2.43 Destination of EEPROM Bits Read[a]

Byte	Function							
	Bit-7	**Bit-6**	**Bit-5**	**Bit-4**	**Bit-3**	**Bit-2**	**Bit-1**	**Bit-0**
0	PCI_SID							
	[31] (SID)	[30] (SID)	[29] (SID)	[28] (SID)	[27] (SID)	[26] (SID)	[25] (SID)	[24] (SID)
1	PCI_SID							
	[23] (SID)	[22] (SID)	[21] (SID)	[20] (SID)	[19] (SID)	[18] (SID)	[17] (SID)	[16] (SID)
2	PCI_SID							
	[15] (SVID)	[14] (SVID)	[13] (SVID)	[12] (SVID)	[11] (SVID)	[10] (SVID)	[9] (SVID)	[8] (SVID)
3	PCI_SID							
	[7] (SVID)	[6] (SVID)	[5] (SVID)	[4] (SVID)	[3] (SVID)	[2] (SVID)	[1] (SVID)	[0] (SVID)
4	Enables PCI_BSROM	PBROM_CTL						
		[22] (BS)	[21] (BS)	[20] (BS)	[19] (TC)	[18] (TC)	[17] (TC)	[16] (TC)
5	PBROM_CTL							
	[15] (TA)	[14] (TA)	[13] (TA)	[12] (TA)	[11] (TA)	[10] (TA)	[9] (TA)	[8] (TA)
6	PBROM_CTL							
	[7] (TA)	[6] (TA)	[5] (TA)	[4] (TA)	[3] (TA)	[2] (TA)	[1] (TA)	[0] (TA)
7	PBROM_CTL		Enables PCI_BST0	PBTI0_CTL				
	[25] (DSIZE)	[24] (DSIZE)		[27] (BS)	[26] (BS)	[25] (BS)	[24] (BS)	[6] (PAS)
8	Enables PCI_BST1	PBTI1_CTL					QBSI0_CTL	
		[27] (BS)	[26] (BS)	[25] (BS)	[24] (BS)	[6] (PAS)	[31] (PWEN)	[24] (PAS)
9	QBSI0_AT							
	[31] (TA)	[30] (TA)	[29] (TA)	[28] (TA)	[27] (TA)	[26] (TA)	[25] (TA)	[24] (TA)
10	QBSI0_AT							
	[23] (TA)	[22] (TA)	[21] (TA)	[20] (TA)	[19] (TA)	[18] (TA)	[17] (TA)	[16] (TA)
11	QBSI0_AT					Reserved (0)	Reserved (0)	Reserved (0)
	[7] (BS)	[6] (BS)	[5] (BS)	[4] (BS)	[1] (EN)			

a. The top part of each byte row indicates the register name. Bit locations within the register are in square brackets, and field names are within round brackets.

PCI Configuration Subsystem ID Register (PCI_SID register)

Bytes 0 to 3 read from the external serial EEPROM are loaded into the 32-bit PCI Configuration Subsystem ID Register (PCI_SID). Bit-7 of byte-0 is loaded into bit-31 of the PCI_SID register and bit-0 of byte-3 is loaded into bit-0 of the PCI_SID register.

PCI Expansion ROM (PCI_BSROM and PBROM_CTL registers)

If Bit-7 of byte-4 is “1” then the PCI Configuration Expansion ROM Base Address Register (PCI_BSROM) is enabled. If the register is enabled, then the QSpan supports PCI Expansion ROM image access. If the register is disabled, reads to this register and the PBROM_CTL register return all zeros.

Unlike the other non-reserved bits read from the EEPROM, bit-7 of byte-4, bit-5 of byte-7 and bit-7 of byte-8 are not written to a QSpan register; they merely determine whether certain other register bits are loaded from the EEPROM).

The next three bits of byte-4 are loaded into the BS field of the PCI Bus Expansion ROM Control Register (PBROM_CTL). The last four bits of byte-4 are loaded into the TC field of the PBROM_CTL register. The fifth and sixth bytes are loaded into the TA fields of the PBROM_CTL register, with bit-7 of byte-5 loaded into TA[15] and bit-0 of byte-6 loaded into TA[0]. Bit-7 and bit-6 of byte-7 is loaded into the DSIZE field of the PBROM_CTL register.

PCI Bus Target Image 0 (PCI_BST0 and PBTI0_CTL registers)

The rest of byte-7 pertains to the PCI Target Image 0. If bit-5 of byte-7 is set to “1”, then PCI_BST0 is enabled and the rest of byte-7 is read and applied to the PBTI0_CTL as follows: bit-4 through bit-1 determine the BS field of the PBTI0_CTL while bit-0 determines the PAS field. The functional effects of enabling the PBTI0_CTL register are discussed in "PCI BIOS Memory Allocation" on page 2-32.

PCI Bus Target Image 1 (PCI_BST1 and PBTI1_CTL registers)

Bits-7 to 2 of byte-8 perform the same function as bits-5 to 0 of byte-7, described in the previous paragraph, except that they apply to the other target image (PCI_BST1 and PBTI1_CTL). The functional effects of enabling the PCI_BST1 register are discussed in "PCI BIOS Memory Allocation" on page 2-32.

QBus Slave Image 0 (QBSI0_CTL and QBSI0_AT registers)

QBus Slave Image 0 can be programmed from the EEPROM (the other QBus Slave Image can not be programmed via the EEPROM). The PWEN and the PAS fields of the QBus Slave Image 0 Control Register (QBSI0_CTL) can be set by bit-1 and bit-0, respectively, of byte-8 of the EEPROM.

Byte-8 through 11 program the TA, BS, and EN fields of the QBus Slave Image 0 Address Translation Register (QBSI0_AT).

The last three bits of byte-11 are reserved and should be set to zero.

2.7.4 Programming the EEPROM from the QBus or PCI Bus

One may read from or write to the EEPROM from the QBus or the PCI bus through 32-bit writes to the EEPROM_CS register as described below.

2.7.4.1 Writing to the EEPROM

The EEPROM values are loaded by the QSpan when the QSpan is reset. Therefore, the user should reset the QSpan after writing to the EEPROM through the QSpan in order to load any new EEPROM data written to the lower 12 bytes.

In order to write to the EEPROM:

1. Wait for the ACT bit of the EEPROM_CS register to be “0”, allowing the QSpan a latency of 1 SCLK (PCLK/1024) to set this bit.
2. Perform a 32-bit write to the EEPROM_CS register by setting the READ bit of the EEPROM_CS to “0”, and supplying the appropriate value to the ADDR[7:0] and DATA[7:0] fields.

The address field has been modified from the previous revision of the QSpan and existing software will need to be altered.

Writes complete normally on the QBus or the PCI bus regardless of the state of the ACT bit. However, if the ACT bit is “1” the write does not change the content of the EEPROM_CS register. The master is not informed that the EEPROM_CS register access failed due to the status of the ACT bit. To ensure that the register write has been successful, the user may read from the EEPROM_CS register after the write.

The ACT bit is “1” when the QSpan is loading data from the EEPROM or is in the process of completing a write to the EEPROM caused by an access to this register. However, it takes the QSpan 1 SCLK to set the ACT bit, where 1 SCLK = PCLK/1024. Therefore, after accessing the EEPROM_CS register, the master should wait at least this amount of time before verifying the state of the ACT bit. If the PCI bus is operating at 33 MHz, this means waiting 31 us. After this period of time, the ACT bit will be valid.

2.7.4.2 Reading from the EEPROM

In order to read from the EEPROM, one should use the following procedure:

1. Wait for the ACT bit of the EEPROM_CS register to be “0”, allowing the QSpan a latency of 1 SCLK (PCLK/1024) to set this bit.
2. Write the appropriate address in the ADDR[7:0] field of the EEPROM_CS register.
3. Set the READ bit of the EEPROM_CS register to “1” (if it has not already been set). The QSpan initiates a read from the EEPROM at the address specified in the ADDR[7:0] field. The ACT bit is set (“1”) 1 SCLK after the READ bit is set, and it is cleared (“0”) when the read completes. The QSpan stores the read data in the DATA[7:0] field of the EEPROM_CS register. Only one byte may be read at a time.
4. Wait for the ACT bit of the EEPROM_CS register to be “0”.
5. Read the DATA[7:0] field of the EEPROM_CS register.

2.8 Reset Options

2.8.1 Resets

The QSpan can be reset from the QBus or the PCI bus through hardware. The QSpan uses three pins and one register for reset operation. The reset pins are listed in Table 2.44 below.

Table 2.44 Hardware Reset Mechanisms

Pin Name	Interface	Direction	Function
RST#	PCI	Input	Resets all the QSpan circuits and registers and asserts $\overline{\text{RESETO}}$. $\overline{\text{RESETO}}$ remains asserted until RST# is released. Also clears the SW_RST bit in the MISC_CTL (page A-46) register.
$\overline{\text{RESETI}}$	QBus	Input	Resets most of the QSpan circuits and registers (See Appendix A, “Registers”.)
$\overline{\text{RESETO}}$	QBus	Output	Resets devices on the QBus

The SW_RST bit in the MISC_CTL (page A-46) controls the QBus reset output ($\overline{\text{RESETO}}$). One of the uses of this mechanism is to allow the user to keep the QBus in reset while an operating system and driver are downloaded to on-board memory. When “1” is written to the SW_RST bit, the QSpan asserts $\overline{\text{RESETO}}$ and keeps it asserted until the software reset state is terminated. There are three ways to cause the QSpan to terminate the software reset state:

1. Clear the SW_RST bit by writing “0” to it. In this case, $\overline{\text{RESETO}}$ is immediately negated.
2. Assert $\overline{\text{RESETI}}$. In this case, the SW_RST bit is immediately cleared (set to “0”) and $\overline{\text{RESETO}}$ is immediately negated.
3. Assert RST#. In this case, SW_RST is immediately cleared (set to “0”), however $\overline{\text{RESETO}}$ continues to be asserted until RST# is negated.

We do not recommend $\overline{\text{RESETO}}$ being looped back to $\overline{\text{RESETI}}$

Asserting the SW_RST bit does not cause an internal reset of QSpan.

2.8.1.1 PCI Transactions During QBus Reset

Assertion of RESETI may interfere with the QSpan's response to PCI masters. If a PCI master attempts to access the QSpan while RESETI is asserted, the QSpan will not decode the incoming cycle and a Master-Abort will occur on the PCI bus. If the QSpan has already been selected by a PCI master (QSpan has asserted DEVSEL#), then the QSpan will immediately negate DEVSEL#, but TRDY# and STOP# will not be asserted by the QSpan.

2.8.1.2 IDMA Reset

IDMA reset issues are discussed in "IDMA Errors, Resets and Interrupts" on page 2-54.

2.8.1.3 Clocking and Resets

The PCLK can operate anywhere from DC to 33 MHZ, provided that the QSpan receives at least 5 PCI clocks before PCI RST# negates.The QSpan's QCLK input should also be operating for at least 5 QCLKs before the reset input (RST# or RESETI negates). Refer to the tables in "Signals and DC Characteristics" on page 4-1 for a description of the state of each QSpan pin after reset.

When a PCI reset (RST#) is asserted, the QSpan's RESETO signal will be asserted and negated as shown in Appendix C.

The QCLK and PCLK inputs are necessary for software resets. That is, these clocks are required in order for the QSpan to negate RESETO. This is due to the fact that for a software reset, the QSpan's registers must be accessed to cause RESETO to negate. If the PCLK input stopped toggling then it is impossible to write to the QSpan's registers to negate RESETO.

2.8.2 Reset Options

Four aspects of the QSpan can be determined at reset:

1. One is whether the QSpan is enabled as a PCI bus master.
2. The second is the master and slave modes of the QBus, which determines the type of cycles that the QSpan can generate as a master and accept as a slave.
3. The third is whether the QSpan loads registers from an EEPROM at reset. These options are discussed in turn.
4. The fourth option is the test mode (see "Test Mode Pins" on page 2-78).

2.8.2.1 PCI Bus Master Reset Options

If BM_EN/$\overline{\text{FIFO\_RDY}}$ is sampled at a logic high (“1”) while reset is asserted, the QSpan will set the BM bit in the PCI_CS register (page A-5). This enables the QSpan as a PCI bus master. If this pin is not connected, an internal pull-down causes the QSpan to power-up with the BM bit equaling “0”.

2.8.2.2 QBus Master and Slave Modes

The QSpan has four master and slave modes that are determined by the $\overline{\text{BDIP}}$ and the SIZ[1] signals at reset. The QBus may be in QUICC, PowerQUICC or M68040 master mode. The QBus Slave Module is always capable of accepting QUICC signals and either PowerQUICC or M68040 signals. These reset options are listed in Table 2.45 below.

Table 2.45 Reset Options for QBus Master and Slave Modes

Reset sampling		Master Mode	Slave Mode
$\overline{\text{BDIP}}$	SIZ[1]		
0	0	QUICC	QUICC and M68040
0	1	QUICC	QUICC and PowerQUICC
1	0	M68040	QUICC and M68040
1	1	PowerQUICC	QUICC and PowerQUICC

Caution: These options are reset whenever the QSpan is reset.

2.8.2.3 EEPROM Loading

If either ENID or SDA is sampled high at reset, then the QSpan will download register information from the EEPROM (see "The EEPROM Channel" on page 2-69).

2.9 Hardware Implementation Issues

This section briefly describes the following hardware topics:

- Test Mode Pins Pins
- JTAG Support
- Decoupling Capacitors

2.9.1 Test Mode Pins

The QSpan can operate in normal mode or test mode. In test mode, a NAND tree is activated and all outputs are tristated except for the SCL output pin. The output of the NAND tree is on the SCL pin.

The QSpan has two test mode input pins (TMODE[1:0]). For normal operations these inputs should be pulled down. Table 2.46 below indicates the operation modes of the QSpan as a function of the TMODE[1:0] input. At reset the TMODE[1:0] inputs are latched by the QSpan to determine the mode of operation. The QSpan remains in this mode until the TMODE[1:0] inputs have changed and a reset event has occurred.

Table 2.46 Test Mode Operation

TMODE[1:0]	Operation Mode
00	Normal Mode
01	Reserved
10	Reserved
11	NAND Tree/Tristate Outputs

2.9.2 JTAG Support

The QSpan PBGA package includes dedicated user-accessible test logic that is fully compatible with the IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture. The following pins are provided: TCK, TDI, TDO, TMS, and TRST#.

There is a internal pull-up register on TMS which will keep the QSpan's JTAG controller in a reset state without requiring the TRST# to be low.

The current revision of the Boundary Scan Description Language (BSDL) file for the QSpan (CA91L860B) device is available on the Tundra website - www.tundra.com.

2.9.3 Decoupling Capacitors

When routing the power (V_{DD}) and ground (V_{SS}) tracks to the QSpan during board layout, care should be taken to ensure that the QSpan is isolated from the power being supplied to other devices on the board by 0.1uF decoupling capacitors.

It is recommended every fourth V_{DD}/V_{SS} pair has a decoupling capacitor.

3 Description of Signals

The following detailed description of the QSpan signals is organized according to these functional groups:

- QUICC Signals
- PowerQUICC Signals
- M68040 Signals
- PCI Bus Signals
- Miscellaneous Signals

PowerQUICC signals do not necessarily operate in the same manner as QUICC signals of the same name.

The QSpan's QBus Interface defines a number of signals that may be mapped to M68040, QUICC, or PowerQUICC buses. The mapping between these signals is described in Table 3.1 below.

Table 3.1 QBus Signal Names Compared to Motorola Signals

QBus Interface	QUICC	PowerQUICC	M68040
$\overline{BB}$/$\overline{BGACK}$	$\overline{BGACK}$	$\overline{BB}$	$\overline{BB}$
$\overline{BERR}$/$\overline{TEA}$	$\overline{BERR}$	$\overline{TEA}$	$\overline{TEA}$
$\overline{BURST}$/$\overline{TIP}$	N/A	$\overline{BURST}$	$\overline{TIP}$
$\overline{DACK}$/$\overline{SDACK}$	$\overline{DACK}$	$\overline{SDACK}$	N/A
$\overline{DSACK1}$/$\overline{TA}$	$\overline{DSACK1}$	$\overline{TA}$	$\overline{TA}$
SIZ[1:0]	SIZ[1:0]	TSIZ[0:1]	SIZ[1:0]
TC[3:0]	FC[3:0]	AT[0:3]	TT[1:0] TM[2:0][a]
$\overline{HALT}$/$\overline{TRETRY}$	$\overline{HALT}$	$\overline{TRETRY}$	N/A

a. TC[3:0] may be connected to 4 out of the 5 TT[1:0] and TM[2:0] M68040 signals. The unused TC pins, if any, should be connected to pull-up resistors. See Appendix-C.

3.1 QUICC Signals

A[31:0]	Tristate bidirectional
Address Bus—Address for the current bus cycle. It is driven by the QSpan when it is the QBus master and input when QBus slave. It is qualified at the start of a transaction by $\overline{AS}$. As a **slave**, the QSpan samples A[31:0] on the same falling edge of the QCLK as $\overline{AS}$. Both A[31:0] and $\overline{AS}$ must meet the synchronous set-up and hold time parameters about the falling edge of the QCLK to ensure correct operation. As a **master**, the QSpan maintains the correct asynchronous timing relationships between A[31:0] and $\overline{AS}$. The address bus is driven valid after the rising edge of the QCLK, while the $\overline{AS}$ is driven only after the subsequent falling edge of the same clock period, ensuring the correct address before $\overline{AS}$ timing. When accesses are made to QSpan registers from the QBus, only the lower 12 bits of the address bus are used to determine the offset.	
$\overline{AS}$	**Rescinding Tristate bidirectional**
Address Strobe—indicates the beginning (and duration) of a transaction on the QBus. As an **output** $\overline{AS}$ is driven by the QSpan when the QSpan is the QBus master, and is tristated at all other times. The Address Strobe is driven low after a falling edge of the QCLK. The Address Strobe qualifies the following signals as valid when it is asserted: A[31:0], TC[3:0], SIZ[1:0], and R/$\overline{W}$. The QSpan guarantees a minimum set-up time for the qualified signals before $\overline{AS}$ is asserted (all qualified signals are driven from the rising edge of QCLK preceding the assertion of $\overline{AS}$). The QSpan rescinds $\overline{AS}$ prior to tristate As an **input**, $\overline{AS}$ is sampled on the falling edge of the QCLK. $\overline{AS}$ must meet a minimum set-up and hold time around the falling edge of the clock for correct operation. The QSpan recognizes a transaction as intended for it, and acknowledges it accordingly, only if one of $\overline{CSREG}$ or $\overline{CSPCI}$ is sampled low in conjunction with $\overline{AS}$. The QSpan does not require that the input signals qualified by the $\overline{AS}$ be valid when it is asserted.	
$\overline{BB}$/$\overline{BGACK}$	**Rescinding Tristate bidirectional**
Bus Busy—indicates ownership of the QBus. It, along with $\overline{BR}$ and $\overline{BG}$, provides the three-wire handshake for QBus arbitration. $\overline{BB}$/$\overline{BGACK}$ is intended to connect to the $\overline{BGACK}$ bus. As an **output** the QSpan asserts $\overline{BB}$/$\overline{BGACK}$ from the falling edge of QCLK (while master). The QSpan rescinds $\overline{BB}$/$\overline{BGACK}$ prior to tristate. As an **input**, the QSpan double-samples $\overline{BB}$/$\overline{BGACK}$ on the falling edge of QCLK (while master). The QSpan can also be programmed to use a synchronous mode for QBus arbitration.	
$\overline{BGACK}$	**Rescinding Tristate bidirectional**
See $\overline{BB}$/$\overline{BGACK}$	
$\overline{BDIP}$	**Input (QUICC mode) / Bidirectional (PowerQUICC)**
Burst Data In Progress—On the QUICC interface, this pin is used only to determine the QBus master mode of the QSpan is determined at reset by sensing the level of this pin. If $\overline{BDIP}$ is sampled at logic low (at power-up or reset) the QBus master module will operate as a QUICC master. If the $\overline{BDIP}$ signal is sampled at logic high (at power-up or reset), the QSpan will operate as a PowerQUICC master (See Table 2.45 on page 2-77.)	
$\overline{BERR}$/$\overline{TEA}$	**Rescinding tristate birectional pin**
Bus Error— used to indicate a bus error that occurs during a transaction. It can be used in conjunction with $\overline{HALT}$/$\overline{TRETRY}$ to indicate a busy-retry to the bus master. As a QUICC master, the QSpan samples $\overline{BERR}$/$\overline{TEA}$ on the falling edge of QCLK during cycles in which it is a QBus master. As a QUICC slave, $\overline{BERR}$/$\overline{TEA}$ is driven by the QSpan from the falling edge of QCLK. The QSpan negates $\overline{BERR}$/$\overline{TEA}$ prior to tristate.	

3.1 QUICC Signals (Continued)

$\overline{\text{BG}}$	Input
Bus Grant—indicates that the QSpan may become the next QBus master. It, along with $\overline{\text{BR}}$ and $\overline{\text{BB}}$/$\overline{\text{BGACK}}$, provides the three-wire handshake for QBus arbitration. $\overline{\text{BG}}$ is doubled-sampled on the falling edge of QCLK. The QSpan can be programmed to use a synchronous mode for QBus arbitration.	
BM_EN/$\overline{\text{FIFO\_RDY}}$	**Bidirectional**
Bus Master Enable/FIFO Ready—If this input is asserted (“1”) during a PCI Reset, the Bus Master Enable bit in the PCI_CS register will be set.	
$\overline{\text{BR}}$	Output
Bus Request —used by the QSpan to request ownership of the QBus. It, along with $\overline{\text{BG}}$ and $\overline{\text{BB}}$/$\overline{\text{BGACK}}$, provides the three-wire handshake for QBus arbitration. $\overline{\text{BR}}$ is asserted and negated from the falling edge of QCLK in QUICC mode.	
$\overline{\text{CSPCI}}$	Input
PCI Chip Select—indicates that the current transaction on the QBus is an access to the PCI Bus. During IDMA cycles, if this is sampled high, a single address transfer is indicated; if sampled low, a dual address transfer is indicated. It is sampled on the falling edge of clock.	
$\overline{\text{CSREG}}$	Input
Register Chip Select—indicates that the current transaction on the QBus is an access to the QSpan’s registers.It is sampled on the falling edge of clock.	
D[31:0]	Tristate bidirectional
Data Bus—provides the data information for the QSpan’s inputs and outputs on the QBus.\ As a QUICC slave the QSpan does not use $\overline{\text{DS}}$ to qualify data on writes. It also provides data on reads without decoding $\overline{\text{DS}}$, since DS is output only. As a QUICC master, the QSpan does use DS to qualify data on writes and to request data on reads.	
$\overline{\text{DACK}}$/$\overline{\text{SDACK}}$	Input
IDMA Acknowledge—indicates to the QSpan that the current transaction is an IDMA transaction. The timing of $\overline{\text{DACK}}$ should be the same as for $\overline{\text{AS}}$. Using the IDMA handshakes, the QSpan is capable of supporting MC68360 fast termination cycles.	
$\overline{\text{DONE}}$	Input
IDMA Done—indicates that the IDMA controller has completed the current sequence of IDMA operations, and that the QSpan should no longer use $\overline{\text{DREQ}}$ to request transactions. Setup for DONE is to falling edge of QCLK.	
$\overline{\text{DREQ}}$	Output
IDMA Request—request to the QUICC IDMA to either transfer data to QSpan IFIFO (PCI Write) or remove data from I-FIFO (PCI Read). It is asserted from the falling edge of QCLK in QUICC mode.	
$\overline{\text{DSACK0}}$	Rescinding tristate bidirectional
Data and Size Acknowledge 0—in conjunction with $\overline{\text{DSACK1}}$/$\overline{\text{TA}}$, is driven by the addressed slave to acknowledge the completion of a data transfer on the QBus $\overline{\text{DSACK0}}$ has the same timing and characteristics as $\overline{\text{DSACK1}}$/$\overline{\text{TA}}$ (see description below).	

3.1 QUICC Signals (Continued)

DSACK1/TA	Rescinding tristate bidirectional
DSACK1/TA, in conjunction with DSACK0, is driven by the addressed slave to acknowledge the completion of a data transfer on the QBus. The QSpan terminates all normal bus cycles by asserting both DSACK1/TA and DSACK0 (indicating a 32-bit port width at all times). The DSACK1/TA output is driven high (inactive) after the release of AS until the next falling edge of the clock, at which point it is tristated.	
DS	**Rescinding tristate output**
Data Strobe—used to indicate valid data on the data bus during write transactions, and to request data during read transactions. DS is driven by the QSpan when it is a QBus master, and is tristated otherwise. As a **slave** the QSpan assumes write data is valid on the rising edge of QCLK following the clock edge where AS is sampled asserted. For read transactions, the QSpan provide information independent of DS. DS is output only. As a **master** on the QBus, the QSpan asserts DS to qualify data during reads and writes. For write transactions, the DS is driven from the falling edge of QCLK one half a clock period after the data is driven onto the Data bus. For read transactions, DS is driven at the same time as AS. The QSpan negates DS prior to tristate.	
HALT/TRETRY	**Rescinding tristate bidirectional**
Halt—used for generating retries. As a QUICC slave QSpan uses HALT/TRETRY as stated in Table 2.10 on page 2-22. As a QUICC master QSpan uses HALT/TRETRY as stated in Table 2.26 on page 2-45.	
IMSEL	**Input**
Image Select—selects which QBus Slave Image to use when CSPCI is asserted. The timing requirements for IMSEL are the same as those of the address bus when the QSpan is a QBus slave.	
QCLK	**Input**
QBus Clock—All devices intended to interface with QBus side of the QSpan must be synchronized to this clock. The QCLK can operate up to 33 MHz (with a QUICC bus). During IDMA fast termination cycles the maximum QUICC QCLK frequency is 20 MHz.	
QINT	**Open drain bidirectional**
QBus Interrupt—as an output, this open drain signal is asserted by the QSpan when an interrupt event occurs,. As an input, this signal can be mapped to the PCI INT# output.	
RESETI	**Input**
QBus Reset Input—resets the QSpan from the QBus side of the QSpan. Note that RESETI does not reset PCI configuration and status registers.	
RESETO	**Open drain output**
QBus Reset Output—asserted whenever the QSpan's PCI RST# input is asserted, or the internal software reset bit is set.	
R/W	**Tristate bidirectional**
Read Write—indicates the direction of the data transfer on the Data bus. A logic high indicates a read transaction, a logic low a write. It has the same timing as the Address bus. As a **master**, the QSpan drives R/W, and tristates it otherwise. As a **slave**, the R/W pin is an input.	

3.1 QUICC Signals (Continued)

SIZ[1:0]	Tristate bidirectional
Size—indicates the number of bytes to be transferred during a bus cycle. The value of the Size bits, along with the lower two address bits and the port width, define the byte lanes that are active. Table 3.1 on page 3-9 shows the encoding for the Size bits.	
TC[3:0]	**Tristate bidirectional**
Transaction Code—provides additional information about a bus cycle when the QSpan is a QBus master. Driven by the QSpan when it is a QBus master, and tristated otherwise. As a **slave**, the QSpan samples TC[3:0] on the first falling edge of the QCLK after $\overline{AS}$ is asserted. TC[3:0] can optionally be used with $\overline{DACK}$/$\overline{SDACK}$ to decode an IDMA operation. For use in IDMA transfers, TC[3:0] should be set to all ones. The timing for the TC[3:0] outputs is the same as the timing for the address bus when the QSpan is a QBus master. The values output on the TC[3:0] bus during a transaction in which the QSpan is the bus master is determined by the value programmed in the Transaction Code field of the corresponding QSpan PCI target image. TC[3:0] is intended to connect to the QUICC's FC[3:0], but may be used for other special decoding purposes.	

3.2 PowerQUICC Signals

A[31:0]	Tristate bidirectional
Address bus—Address for the current bus cycle. It is driven by the QSpan when it is the QBus master and input as slave. It is qualified at the start of a transaction by $\overline{TS}$. As a **slave**, the QSpan samples A on the rising edge of QCLK, and is qualified by Transaction Start ($\overline{TS}$) on the same rising clock edge. As a **master**, the address is driven out following a rising edge of the QCLK. When accesses are made to QSpan registers from the QBus, only the lower 12 bits of the address bus are used to determine the register offset.	
AT[0:3]	**Tristate bidirectional**
See TC[3:0]	
$\overline{BB}$/$\overline{BGACK}$	**Rescinding tristate bidirectional**
Bus Busy—indicates ownership of the QBus. It, along with $\overline{BR}$ and $\overline{BG}$, provides the three-wire handshake for QBus arbitration. As an **output** the QSpan asserts $\overline{BB}$/$\overline{BGACK}$ from the rising edge of QCLK (while master). The QSpan drives $\overline{BB}$/$\overline{BGACK}$ to the prior to tristate. Note in the PowerQUICC mode, the QSpan guarantees $\overline{BB}$ one clock after receiving $\overline{BG}$ - in compliance with the PowerQUICC arbiter. As an **input**, the QSpan samples $\overline{BB}$/$\overline{BGACK}$ on the rising edge of QCLK. The QSpan can also be programmed to use a asynchronous mode for QBus arbitration.	
$\overline{BDIP}$	**Bidirectional**
Burst Data In Progress— As PowerQUICC master, the QSpan uses BPIP in burst writes to indicate the second last data beat of a transaction. This allows the QSpan to perform burst writes of two, three, or four beats. The QSpan does not use BDIP in the same manner for burst reads. Burst reads are always cacheline aligned and four beats in length. As PowerQUICC slave, the QSpan monitors BDIP as a signal indicating the second last data beat in the burst. This allows the QSpan to support bursts of two, three, or four data beats. The QBus master mode of the QSpan is determined at power-up and reset by sensing the level of this pin. If $\overline{BDIP}$ is sampled at logic low (at reset) the QBus master module will operate as a QUICC master. If the $\overline{BDIP}$ signal is sampled at logic high (at reset), the QSpan will operate as a PowerQUICC or M68040 master (see Table 2.45 on page 2-77.)	
$\overline{BERR}$/$\overline{TEA}$	**Rescinding tristate bidirectional**
Transfer Error Acknowledge—provides additional information about a bus cycle when the QSpan is a QBus master. Driven by the QSpan when it is a QBus master, and tristated otherwise. As an **output** $\overline{BERR}$/$\overline{TEA}$ is driven by the QSpan from the rising edge of QCLK. The QSpan negates $\overline{BERR}$/$\overline{TEA}$ prior to tristate. As an **input**, the QSpan samples $\overline{BERR}$/$\overline{TEA}$ on the rising edge of QCLK during cycles in which it is a QBus **master**.	
$\overline{BG}$	**Input**
Bus Grant—indicates that the QSpan may become the next QBus master. It, along with $\overline{BR}$ and $\overline{BB}$/$\overline{BGACK}$, provides the three-wire handshake for QBus arbitration. $\overline{BG}$ is sampled on the rising edge of QCLK. The QSpan can be programmed to use a asynchronous mode for QBus arbitration.	
$\overline{BR}$	**Output**
Bus Request —used by the QSpan to request ownership of the QBus. It, along with $\overline{BG}$ and $\overline{BB}$/$\overline{BGACK}$, provides the three-wire handshake for QBus arbitration. $\overline{BR}$ is asserted and released from the rising edge of QCLK.	

3.2 PowerQUICC Signals (Continued)

BM_EN/FIFO_RDY	Bidirectional
Bus Master Enable—If this input is asserted (“1”) during a PCI Reset, the Bus Master Enable bit in the PCI_CS register will be set. (FIFO_RDY) FIFO Ready functionality is not relevant to PowerQUICC applications.	
BURST/TIP	**Tristate bidirectional**
Burst—used to indicate a burst cycle. This signal matches the MPC860 signal of the same name.	
CSPCI	**Input**
PCI Chip Select—indicates that the current transaction on the QBus is an access to the PCI Bus. CPCI can be sampled on the same clock as TS or up to three clocks following TS assertion. During IDMA cycles, if this is sampled high, a single address transfer is indicated; otherwise, a dual address transfer is indicated.	
CSREG	**Input**
Register Chip Select—indicates that the current transaction on the QBus is an access to the QSpan’s registers. CSREG can be sampled on the same clock as TS or up to three clocks following TS assertion. This signal is sampled synchronously on the rising edge of clock after TS.	
D[31:0]	**Tristate bidirectional**
Data Bus—provides the data information for the QSpan’s inputs and outputs on the QBus.	
DACK/SDACK	**Input**
IDMA Acknowledge—indicates to the QSpan that the current transaction is an IDMA transaction.	
DONE	**Input**
IDMA Done—This signal is not used with PowerQUICC transfers.	
DREQ	**Output**
IDMA Request—request to the QUICC IDMA to either transfer data to QSpan IFIFO (PCI Write) or remove data from I-FIFO (PCI Read). It is asserted from the falling edge of QCLK in QUICC mode.	
DSACK1/TA	**Rescinding tristate bidirectional**
Transaction Acknowledge—Driven by the addressed slave to acknowledge the completion of a data transfer on the QBus. As a slave the QSpan terminates all normal bus cycles by asserting TA. The QSpan negates DSACK1/TA prior to tristate.	
HALT/TRETRY	**Rescinding tristate bidirectional**
Transfer Retry—used for generating retries. As a QUICC slave QSpan uses HALT/TRETRY as stated in Table 2.11 on page 2-22. As a QUICC master QSpan uses HALT/TRETRY as stated in Table 2.27 on page 2-46.As a **slave**, HALT/TRETRY has the same timing as DSACK1/TA. The QSpan negates HALT/TRETRY prior to tristate.	
IMSEL	**Input**
Image Select—selects which QBus Slave Image to use when CSPCI is asserted. The timing requirements for IMSEL are the same as those of the address bus when the QSpan is a QBus slave.	
QCLK	**Input**
QBus Clock—All devices intended to interface with QBus side of the QSpan must be synchronized to this clock. The maximum QCLK frequency with a PowerQUICC is 50 MHz.	

3.2 PowerQUICC Signals (Continued)

$\overline{\text{QINT}}$	**Open drain bidirectional**
QBus Interrupt—as an output, this open drain signal is asserted by the QSpan when an interrupt event occurs,. As an input, this signal can be mapped to the PCI INT# output.	
$\overline{\text{RESETI}}$	**Input**
QBus Reset Input—resets the QSpan from the QBus side of the QSpan. Note that $\overline{\text{RESETI}}$ does not reset PCI configuration and status registers.	
$\overline{\text{RESETO}}$	**Open drain output**
QBus Reset Output—asserted whenever the QSpan's PCI RST# input is asserted, or the internal software reset bit is set.	
R/$\overline{\text{W}}$	**Tristate bidirectional**
Read Write—indicates the direction of the data transfer on the Data bus. A logic high indicates a read transaction, a logic low a write. It has the same timing as the Address bus. As an active **master**, the QSpan drives R/$\overline{\text{W}}$, and tristates it otherwise. As an addressed **slave**, the R/$\overline{\text{W}}$ pin is an input.	
SIZ[1:0]	**Tristate bidirectional**
Size—indicates the number of bytes to be transferred during a bus cycle. The value of the Size bits, along with the lower two address bits and the port width, define the byte lanes that are active. Table 3.1 on page 3-9 shows the encoding for the Size bits. SIZ[1:0] is intended to connect to PowerQUICC TSIZ[0:1].	
$\overline{\text{TA}}$	**Rescinding tristate bidirectional**
See $\overline{\text{DSACK1}}$/$\overline{\text{TA}}$	
TC[3:0]	**Tristate bidirectional**
Transaction Code Bus—provides additional information about a bus cycle when the QSpan is a QBus master. Driven by the QSpan when it is a QBus master, and tristated otherwise. As a **slave**, the QSpan samples TC[3:0] on the first falling edge of the QCLK after $\overline{\text{AS}}$ is asserted. TC[3:0] can optionally be used with $\overline{\text{DACK}}$/$\overline{\text{SDACK}}$ to decode an IDMA operation. For use in IDMA transfers, TC[3:0] should be set to all ones. The timing for the TC[3:0] outputs is the same as the timing for the address bus when the QSpan is a QBus master. The values output on the TC[3:0] bus during a transaction in which the QSpan is the bus master is determined by the value programmed in the Transaction Code field of the corresponding QSpan PCI target image. TC[3:0] is intended to connect to the QUICC's FC[3:0], but may be used for other special decoding purposes.	
$\overline{\text{TEA}}$	**Rescinding tristate bidirectional**
See $\overline{\text{BERR}}$/$\overline{\text{TEA}}$	
$\overline{\text{TRETRY}}$	**Rescinding tristate bidirectional**
See $\overline{\text{HALT}}$/$\overline{\text{TRETRY}}$	

3.2 PowerQUICC Signals (Continued)

$\overline{TS}$	Rescinding Tristate bidirectional
Transfer Start—$\overline{TS}$ is a three state bi-directional signal used to indicate the beginning of a PowerQUICC bus transaction on the QBus. The $\overline{TS}$ output is driven by the QSpan when the QSpan is the QBus master, and is tri-stated at all other times. As an **output**, $\overline{TS}$ is driven low after a rising edge of the QCLK. Transfer Start indicates the following signals will be valid on the next rising edge of the QCLK: A[31:0], TC[3:0], SIZ[1:0], and R/$\overline{W}$. The QSpan rescinds $\overline{TS}$ prior to tri-state. As an **input**, $\overline{TS}$ is sampled on the rising edge of the QCLK. The QSpan samples the address bus and other $\overline{TS}$ qualified signals on the same rising edge of QCLK in which it samples $\overline{TS}$ asserted. $\overline{CPCI}$ and $\overline{CSREG}$ may have up to three wait states after $\overline{TS}$ is sampled.	

The following table applies to QUICC and PowerQUICC SIZ[1:0] signals.

Table 3.1 Size Encoding for the SIZ[1:0] Signal

SIZ[1]	SIZ[0]	QSpan Master	QSpan QUICC slave	QSpan PowerQUICC slave
0	0	4 bytes	4 bytes	4 bytes
0	1	1 byte	1 byte	1 byte
1	0	2 bytes	2 bytes	2 bytes
1	1	Reserved	3 bytes	3 bytes

3.3 M68040 Signals

A[31:0]	Tristate bidirectional
Address bus—Address for the current bus cycle. It is driven by the QSpan when the QSpan is the M68040 master and input when the QSpan is the slave. It is qualified at the start of a transaction by $\overline{TS}$. As a slave, the QSpan samples A[31:0] on the rising edge of QCLK, and is qualified by Transaction Start ($\overline{TS}$). As a master, the address is driven out following a rising edge of the QCLK. When accesses are made to QSpan registers from the QBus, only the lower 12 bits of the address bus are used to determine the register offset.	
$\overline{BB}$/$\overline{BGACK}$	**Rescinding tristate bidirectional**
Bus Busy—indicates ownership of the M68040 bus. It, along with $\overline{BR}$ and $\overline{BG}$, provides the three-wire handshake for M68040 bus arbitration. As an output the QSpan asserts $\overline{BB}$/$\overline{BGACK}$ from the rising edge of QCLK (while master). The QSpan negates $\overline{BB}$/$\overline{BGACK}$ prior to tristate. As an input, the QSpan samples $\overline{BB}$/$\overline{BGACK}$ on the rising edge of QCLK (while master). The QSpan can also be programmed to use a asynchronous mode for M68040 bus arbitration.	
$\overline{BDIP}$	**Input**
Burst Data In Progress—This signal is only used in the 68040 mode at reset. The QSpan Master/Slave mode is determined at reset by sensing the level of this pin in conjunction with SIZ[1]. See Table 2.45 on page 2-77.	
$\overline{BERR}$/$\overline{TEA}$	**Rescinding tristate bidirectional**
Transfer Error Acknowledge—provides additional information about a bus cycle when the QSpan is a M68040 bus master. Driven by the QSpan when it is a M68040 bus slave to signal on errored transaction. As an input, the QSpan samples $\overline{BERR}$/$\overline{TEA}$ during M68040-style cycles in which it is a M68040 bus master on the rising edge of QCLK. Target retries are indicated by the simultaneous assertion of $\overline{DSACK1}$/$\overline{TA}$ and $\overline{BERR}$/$\overline{TEA}$.	
$\overline{BG}$	**Input**
Bus Grant—indicates that the QSpan may become the next M68040 bus master. It, along with $\overline{BR}$ and $\overline{BB}$/$\overline{BGACK}$, provides the three-wire handshake for M68040 bus arbitration. $\overline{BG}$ is sampled on the rising edge of QCLK. The QSpan can be programmed to use a asynchronous mode for M68040 bus arbitration.	
BM_EN/$\overline{FIFO\_RDY}$	**Bidirectional**
Bus Master Enable—If this input is asserted (“1”) during a PCI Reset, the Bus Master Enable bit in the PCI_CS register will be set.	
$\overline{BR}$	**Output**
Bus Request —used by the QSpan to request ownership of the M68040 bus. It, along with $\overline{BG}$ and $\overline{BB}$/$\overline{BGACK}$, provides the three-wire handshake for M68040 bus arbitration. $\overline{BR}$ is asserted and released from the rising edge of QCLK.	
$\overline{BURST}$/$\overline{TIP}$	**Tristate bidirectional**
Transaction In Progress—Asserted for the length of a M68040 transfer. (This signal uses the same pin as the PowerQUICC $\overline{BURST}$ signal.)	
$\overline{CSPCI}$	**Input**
PCI Chip Select—indicates that the current transaction on the QBus is an access to the PCI Bus.	

3.3 M68040 Signals (Continued)

$\overline{\text{CSREG}}$	Input
Register Chip Select—indicates that the current transaction on the QBus is an access to the QSpan's registers.	
D[31:0]	**Tristate bidirectional**
Data Bus—provides the data information for the QSpan's inputs and outputs on the M68040 bus.	
$\overline{\text{DSACK1}}/\overline{\text{TA}}$	Rescinding tristate bidirectional
Transaction Acknowledge—Driven by the addressed slave to acknowledge the completion of a data transfer on the QBus. As a slave the QSpan terminates all normal bus cycles by asserting $\overline{\text{DSACK1}}/\overline{\text{TA}}$. The QSpan negates $\overline{\text{DSACK1}}/\overline{\text{TA}}$ prior to tristate. Target retries are indicated by the simultaneous assertion of $\overline{\text{DSACK1}}/\overline{\text{TA}}$ and $\overline{\text{BERR}}/\overline{\text{TEA}}$.	
IMSEL	**Input**
Image Select—selects which QBus Slave Image to use when $\overline{\text{CSPCI}}$ is asserted. The timing requirements for IMSEL are the same as those of the address bus when the QSpan is a M68040 bus slave.	
QCLK	**Input**
QBus Clock—All devices intended to interface with QBus side of the QSpan must be synchronized to this clock. If an external device, such as an ASIC, wishes to perform cycles with the QSpan, it must operate with the QBus clock and use the same timing as either an MC68360 or an MPC860 (or M68040). QCLK can operate up to 40MHz.	
$\overline{\text{QINT}}$	Open drain bidirectional
QBus Interrupt—as an output, this open drain signal is asserted by the QSpan when an interrupt event occurs,. As an input, this signal can be mapped to the PCI INT# output.	
$\overline{\text{RESETI}}$	Input
QBus Reset Input—resets the QSpan from the QBus side of the QSpan. Note that $\overline{\text{RESETI}}$ does not reset PCI configuration and status registers.	
$\overline{\text{RESETO}}$	Open drain output
QBus Reset Output—asserted whenever the QSpan's PCI RST# input is asserted, or the internal software reset bit is set.	
R/$\overline{\text{W}}$	Tristate bidirectional
Read Write—indicates the direction of the data transfer on the Data bus. A logic high indicates a read transaction, a logic low a write. It has the same timing as the Address bus. As a master, the QSpan drives R/$\overline{\text{W}}$, and tristates it otherwise. As a slave, the R/$\overline{\text{W}}$ pin is an input.	
SIZ[1:0]	**Tristate bidirectional**
Size—indicates the number of bytes to be transferred during a bus cycle. The value of the Size bits, along with the lower two address bits and the port width, define the byte lanes that are active. Table 3.1 below shows the encoding for the Size bits. SIZ[1:1] indicates a M68040 burst cycle.	
$\overline{\text{TA}}$	Rescinding tristate bidirectional
See $\overline{\text{DSACK1}}/\overline{\text{TA}}$	

3.3 M68040 Signals (Continued)

TC[3:0]	Tristate bidirectional
Transaction Code Bus—provides additional information about a bus cycle when the QSpan is a M68040 bus master. Driven by the QSpan when it is a M68040 bus master. As a slave, the QSpan samples TC[3:0] on the rising edge of QCLK, and is qualified by Transaction Start ($\overline{TS}$). The timing for the TC[3:0] outputs is the same as the timing for the address bus when the QSpan is a M68040 bus master. The values output on the TC[3:0] bus during a transaction in which the QSpan is the bus master is determined by the value programmed in the Transaction Code field of the corresponding PCI target image. TC[3:0] may be connected to a subset of the TT[1:0] and TM[2:0] M68040 pins. Unused TC[3:0] pins, if any, should be connected to pull-up resistors.	
$\overline{TEA}$	**Rescinding tristate bidirectional**
See $\overline{BERR}$/$\overline{TEA}$	
$\overline{TIP}$	**Tristate bidirectional**
See $\overline{BURST}$/$\overline{TIP}$	

Table 3.1 below describes the signal encoding for M68040 SIZ[1:0] signals. Byte lane enabling is combined with A[1:0] as described in the M68040 User's Manual.

Table 3.1 Encoding for the SIZ[1:0] Signal

SIZ[1]	SIZ[0]	QSpan as M68040 Master	QSpan as M68040 Slave
0	0	4 bytes	4 bytes
0	1	1 byte	1 byte
1	0	2 bytes	2 bytes
1	1	Reserved	Line (burst)

3.4 PCI Bus Signals

AD [31:0]	Bidirectional (t/s)
PCI Address/Data Bus – address and data are multiplexed over these pins providing a 32-bit address/data bus.	
C/BE# [3:0]	**Bidirectional (t/s)**
PCI Bus Command and Byte Enable Lines – command information during address phase and byte line enables during data phase.	
DEVSEL#	**Bidirectional (s/t/s)**
PCI Device Select – driven by the QSpan when it is accessed as PCI slave. Sampled by the QSpan when it is PCI master.	
FRAME#	**Bidirectional (s/t/s)**
Cycle Frame for PCI Bus– driven by the QSpan when it is PCI master, and is monitored by the QSpan when it is PCI target	
GNT#	**Input (in)**
PCI Grant – indicates to the QSpan that it has been granted ownership of the PCI bus. GNT# can be parked ate the QSpan to improve its PCI master performance.	
IDSEL	**Input (in)**
PCI Initialization Device Select – used as a chip select during configuration read and write transactions	
INT#	**Bidirectional (o/d)**
PCI Interrupt— indicates QSpan generating internal interrupt. As an input, this signal will cause QINT# to be asserted on the QBus bus (if enabled).The signal can be used as an input for an application where the PowerQUICC is the system host.	
IRDY#	**Bidirectional (s/t/s)**
Initiator Ready – used by the QSpan to indicate that it is ready to complete a current data phase. As PCI Target, the QSpan monitors this signal during reads to determine when the PCI master is ready to accept data.	
PAR	**Bidirectional (t/s)**
Parity – parity is even across AD [31:0] and C/BE# [3:0] (the number of 1s summed across these lines and PAR equal an even number).	
PCLK	**Input (in)**
PCI Clock - Clock input for PCI interface used to generate fixed timing parameters. PCLK can operate up to 33MHz.	
PERR#	**Bidirectional**
Parity Error – reports parity errors during all transactions. The QSpan asserts PERR# within two clocks of receiving a parity error on incoming data, and holds PERR# for at least one clock for each errored data phase.	
REQ#	**Output (t/s)**
Bus Request – used by the QSpan to indicate that it requires the use of the PCI bus.	
RST#	**Input**
PCI Reset – Asynchronous Reset for PCI Interface	
SERR#	**Bidirectional**
System Error – reports parity errors during all transactions. The QSpan asserts SERR# within two clocks of receiving a parity error on incoming data, and holds SERR# for at least one clock for each errored data phase.	
STOP#	**output (o/d)**
Stop – used by the QSpan as PCI target when it wishes to signal the PCI master to stop the current transaction. As PCI master, the QSpan terminates the transaction if it receives STOP# from the PCI target.	

3.4 PCI Bus Signals (Continued)

TRDY#	Bidirectional (s/t/s)
Target Ready – used by the QSpan as PCI target to indicate that it is ready to complete the current data phase. During a read with QSpan as PCI master, the target asserts TRDY# to indicate to the QSpan that valid data is present on the data bus.	

3.5 Miscellaneous Signals

ENID	Input
ENID—EEPROM Loading Reset Option. If ENID is sampled high after a PCI reset, then the QSpan will download register information from the EEPROM.	
SCL	**Output**
Serial Clock—EEPROM Serial clock	
SDA	**Bidirectional**
Serial Data—EEPROM Serial data line. If SDA is sampled high after a PCI reset, then the QSpan will download register information from the EEPROM.	
TMODE[1:0]	**Input**
Test Mode—Selects the QSpan test mode	
VIO	**Input**
PCI Voltage Detect - For 3.3V QSpan, VIO is connected to 3.3V or 5V to signal the QSpan's PCI buffers to operate 5V or 3.3V output mode.	

VIO is not a power pin, but is used to qualify the PCI signalling environment. If used in a mixed environment where the VIO pin is attached to the 5V power rail, the user must ensure the 3.3V power ramp occurs before the 5V power ramp and the current specification for the VIO pin is not exceeded (see Table 4.3). A current limiting register on the VIO pin may be used to meet this specification (i.e. VIO pulled to 3.3V or 5V through at lease a 470 ohm resistor). Alternatively, VIO may be driven to the appropriate logic level by a spare on-board output

3.6 JTAG Signals

TMS	Input
Test Mode Select - Used to control the state of the Test Access Port controller	
TDI	**Input**
Test Input - Used (in conjunction with TCK) to shift data and instructions into the Test Access Port (TAP) in a serial bit stream.	
TDO	**Output**
Test Output - Used (in conjunction with TCK) to shift data and instructions into the Test Access Port (TAP) in a serial bit stream.	
$\overline{\text{TRST}}$	**Input**
Test Reset - Used to force the Test Access Port (TAP) into a initialized state.	
TCK	**Input**
Test Clock - Used to clock state information and data into and out of the device during boundary scan.	

4 Signals and DC Characteristics

4.1 Terminology

The abbreviations used in this chapter are defined below.

2S	Two-state output
3S	Tristate output
B	Bidirectional
I	Input
O	Output
OD	Open Drain
PD	Internal pull-down
PU	Internal pull-up
TTL	Input with TTL threshold
TTL Sch.	TTL Schmitt trigger input

4.2 Packaging and Voltage Level Support

The QSpan is available in a 256-lead PBGA package requiring 3.3V power supply and providing 3.3V or 5V signalling. At power-up the QSpan uses VIO to qualify the PCI signalling environment as either 3.3V or 5V. This feature is only available in the 3.3 volt PBGA version of the QSpan. If your PCI design includes 5 volt signalling PCI components, then the QSpan's VIO pin should be tied to 5 volts, so that the QSpan's PCI input buffers will be 5V compliant. Please refer to Table 4.3 for more information on the current limit specification on VIO. If the VIO pin detects a 5V environment at power-up, the PCI buffers on the QSpan will support 5V signalling as outputs. The QSpan's PCI buffers are 5V tolerant.

The QSpan is also available in a 208-pin PQFP package, with 5V power and signalling only. (This package has no VIO pin.)

For detailed packaging information on the QSpan, please see Appendix-G.

VIO is not a power pin, but is used to qualify the PCI signalling environment. If used in a mixed environment where the VIO pin is attached to the 5V power rail, the user must ensure the 3.3V power ramp occurs before the 5V power ramp and the current specification for the VIO pin is not exceeded (see Table 4.3). A current limiting register on the VIO pin may be used to meet this specification (i.e. VIO pulled to 3.3V or 5V through at least a 470 ohm resistor). Alternatively, VIO may be driven to the appropriate logic level by a spare on-board output.

4.3 Signals and DC Characteristics

4.3.1 PBGA Electrical Characteristics

Table 4.1 PBGA Electrical Characteristics (non-PCI)

Symbol	Parameter	Condition	Min	Max	Units
Input low voltage					
V_{IL}	TTL			0.8	V
	CMOS			0.3 VDD	V
V_{IH}	Input high voltage				
	TTL		2.0		V
	CMOS		0.7 VDD		V
I_{IH}	Input high current	V_{IN}=VDD	-10	10	µA
	Input with pull-down[a]	V_{IN}=VDD	-10	100	µA
I_{IL}	Input low current	V_{IN}=VSS	-10	10	µA
	Input with pull-up[b]	V_{IN}=VSS	-100	10	µA
V_{OH}	Output high voltage				
	I_{OH} = +8mA	VDD = 3.0V	2.4		V
V_{OL}	Output low voltage				
	I_{OL} = -8mA	VDD = 3.0V		0.4	V
I_{OZ}	3-State Output Leakage Current	V_{OH}=VSS or VDD	-10	10	µA
I_{DD}	Quiescent Supply Current	V_{IN}=VSS or VDD		80[c]	µA
C_{IN}	Input Capacitance			10	pF

a. This pertains to pins ENID and BM_EN

b. This pertains to JTAG inputs: TDI, TMS, $\overline{TRST}$

c. Depends on customer design

Table 4.2 PBGA AC Electrical Characteristics (PCI Buffer[a])

Symbol	Parameter	Condition	Min	Max	Units
$I_{OH(AC)}$	Switching Current High	$0<V_{OUT}\leq 0.3V_{DD}$	$-12V_{IO}$		mA
		$0.3V_{DD}<V_{OUT}<0.9V_{DD}$	$-17.1(V_{DD}-V_{OUT})$		mA
		$0.7V_{DD}<V_{OUT}<V_{DD}$		Eqt'n A[b]	mA
	(Test Point)	$V_{OUT}=0.7V_{DD}$		$-32V_{DD}$	mA
$I_{OL(AC)}$	Switching Current Low	$V_{DD}>V_{OUT}\geq 0.6V_{DD}$	$16V_{DD}$		mA
		$0.6V_{DD}>V_{OUT}>0.1V_{DD}$	$26.7V_{OUT}$		mA
		$0.18V_{DD}<V_{OUT}<0$		Eqt'n B[c]	mA
	(Test Point)	$V_{OUT}=0.18V_{DD}$		$38V_{DD}$	mA
I_{CL}	Low Clamp Current	$-3<V_{IN}\leq -1$	$-25+(V_{IN}+1)/0.015$		mA
I_{CH}	High Clamp Current	$V_{DD}+4>V_{IN}\geq V_{IO}+1$	$25+(V_{IN}-V_{IO}-1)/0.015$		mA
$slew_r$	Output Rise Slew Rate	$0.2V_{DD}$ to $0.6V_{IO}$ load	1	4	V/ns
$slew_r$	Output Fall Slew Rate	$0.6V_{DD}$ to $0.2V_{IO}$ load	1	4	V/ns

a. The AC characteristics of non-PCI signals are listed in Table 4.7 on page 4-6.

b. Equation A: $I_{OH}=(98.0/V_{DD})*(V_{OUT}-V_{DD})*(V_{OUT}+0.4V_{DD})$ for $V_{DD}>V_{OUT}>0.7V_{DD}$

c. Equation B: $I_{OL}=(256/V_{DD}) * V_{OUT} * (V_{DD}-V_{OUT})$ for $Ov < V_{OUT}<0.18V_{DD}$

Table 4.3 PBGA DC Electrical Characteristics (PCI Buffer)

Symbol	Parameter	Condition	Min	Max	Units
V_{DD}	Supply Voltage		3.14	3.46	V
V_{IL}	Input Low Voltage		-0.5	$0.3 * V_{DD}$	V
V_{IH}	Input High Voltage		$0.5 * V_{DD}$	5.3[a]	V
I_{IL}	Input Low Current	$V_{IN}=V_{SS}$	-10	+10	µA
I_{IH}	Input High Input	$V_{IN}=V_{DD}$	-10	+10	µA
V_{OH}	Output High Voltage	I_{OH}=-500uA	$0.9 * V_{DD}$		V
V_{OL}	Output Low Voltage	I_{OL}=1500uA		$0.1 * V_{DD}$	V
I_{VIO_3V}	VIO Input current for 3.3V signalling	0<VIO<3.3V		7.8	mA
I_{VIO_5V}	VIO Input current for 5V signalling	0<VIO<5V		13.4	mA
C_{IN}	Input Capacitance			10	pF

a. Applies to all pins except the JTAG pins (TCK, TDI, TMS, $\overline{TRST}$). These pins are only 3.3V tolerant.

4.3.2 PQFP Electrical Characteristics

Table 4.4 PQFP Device DC Electrical Characteristics (non-PCI)[a]

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input low voltage				
	TTL			0.8	V
V_{IH}	Input high voltage				
	TTL		2.0		V
I_{IH}	Input high current	$V_{IN}=V_{DD}$	-10	10	µA
	Input with pull-down[b]	$V_{IN}=V_{DD}$	-10	180	µA
I_{IL}	Input low current	V_{IN}=VSS	-10	10	µA
V_{OH}	Output high voltage				
	I_{OH} = -12mA	V_{DD} = 4.5v	2.4		V
V_{OL}	Output low voltage				
	I_{OL} = 12mA	V_{DD} = 4.5v		0.4	V
I_{OZ}	3-State Output Leakage Current	V_{OH}=VSS or V_{DD}	-10	10	µA
I_{DD}	Quiescent Supply Current	V_{IN}=VSS or V_{DD}		100[c]	µA
C_{IN}	Input Capacitance			10	pF

a. VDD = 5V +/- 10%

b. This pertains to pins ENID and BM_EN

c. Depends on customer design

Table 4.5 PQFP Device AC Electrical Characteristics (PCI Buffer[a])

Symbol	Parameter	Condition	Min	Max	Units
$I_{OH(AC)}$	Switching Current High	$0<V_{OUT}\leq 1.4$	-44		mA
		$1.4<V_{OUT}<2.4$	$-44+(V_{OUT}-1.4)/0.024$		mA
		$3.1<V_{OUT}<V_{DD}$		Eqt'n A[b]	mA
	(Test Point)	$V_{OUT}=3.1$		-142	mA
$I_{OH(AC)}$	Switching Current Low	$V_{OUT}\geq 2.2$	95		mA
		$0.6\ V_{DD}>V_{OUT}>0.1\ V_{DD}$	$V_{OUT}/0.023$		mA
		$0.71>V_{OUT}>0$		Eqt'n B[c]	mA
	(Test Point)	$V_{OUT}=0.71$		206	mA
I_{CL}	Low Clamp Current	$-5<V_{IN}\leq -1$	$-25+(V_{IN}+1)/0.015$		mA
$slew_r$	Output Rise Slew Rate	0.4V to 2.4V load	1	5	V/ns
$slew_r$	Output Fall Slew Rate	2.4V to 0.4V load	1	5	V/ns

a. The AC characteristics of non-PCI signals are listed in Table 4.7 on page 4-6.

b. Equation A: $I_{OH}=11.9*(V_{OUT}-5.25) * (V_{OUT}+2.45)$ for $V_{DD} > V_{OUT} > 3.1$ v

c. Equation B: $I_{OL}=78.5* V_{OUT} * (4.4-V_{OUT})$ for $0v < V_{OUT} < 0.71v$

Table 4.6 PQFP Device DC Electrical Characteristics (PCI Buffer)

Symbol	Parameter	Condition	Min	Max	Units
V_{DD}	Supply Voltage		4.5	5.5	V
V_{IL}	Input low voltage		-0.5	0.8	V
V_{IH}	Input high voltage		2.0	Vdd+0.5	V
I_{IH}	Input high current	Vin=2.7V		70	µA
I_{IL}	Input low current	Vin=0.5V		-70	µA
V_{OL}	Output low voltage	Iout=3mA, 6mA		0.55	V
V_{OH}	Output high voltage	Iout=2mA	2.4		V
C_{IN}	Input Capacitance			10	pF

4.3.3 AC Signal Characteristics

Table 4.7 Pin List for QSpan Signals

Pin Name	Pin Number		Type	Input Type	Output Type	Reset State	3.3 V - 256 PBGA		5 V - 208 PQFP		Inter-face	Signal Description
	PQFP	PBGA					I_{OL} (mA)	I_{OH} (mA)	I_{OL} (mA)	I_{OH} (mA)		
A[31:0]	See Table 4.9	See Table 4.9	B	TTL	3S	Hi-Z	8	-8	12	-12	QBus	Address Lines
AD[31:0]	See Table 4.8	See Table 4.8	B	PCI[a]	3S	Hi-Z	PCI	PCI	PCI	PCI	PCI	Address/data Lines
$\overline{AS}$	126	N20	B	TTL Sch.	3S	Hi-Z	8	-8	12	-12	QBus	Address Strobe
$\overline{BB}$/$\overline{BGACK}$	33	R1	B	TTL.	3S	Hi-Z	8	-8	12	-12	QBus	Bus Busy/bus Grant Acknowledge
$\overline{BDIP}$	127	M17	B	TTL	-	Hi-Z	8	-8	12	-12	QBus	Burst Data In Progress (And Tundra QSpan Master Mode)
$\overline{BERR}$/$\overline{TEA}$	125	N19	B	TTL	3S	Hi-Z	8	-8	12	-12	QBus	Bus Error/ Transfer Error Acknowledge
$\overline{BG}$	128	M18	I	TTL	-	Hi-Z	-	-	-	-	QBus	Bus Grant
$\overline{BGACK}$	See $\overline{BB}$/$\overline{BGACK}$ above											
BM_EN/ $\overline{FIFO\_RDY}$	57	M20	B (PD)	TTL Sch.	3S	Hi-Z	-8	8	12	-12	QBus	Bus Master Enable
$\overline{BR}$	34	R2	O	TTL	2S	High	8	-8	12	-12	QBus	Bus Request
$\overline{BURST}$/$\overline{TIP}$	31	N3	B	TTL	3S	Hi-Z	8	-8	12	-12	QBus	Burst/transaction In Progress
C/BE[0]	100	W15	B	PCI	3S	Hi-Z	-	-	-	-	PCI	Command And Byte Enables
C/BE[1]	87	W12				Hi-Z						
C/BE[2]	75	Y9				Hi-Z						
C/BE[3]	62	W4				Hi-Z						
$\overline{CSPCI}$	118	R20	I	TTL	-	Hi-Z	-	-	-	-	QBus	PCI Chip Select
$\overline{CSREG}$	117	T19	I	TTL	-	Hi-Z	-	-	-	-	QBus	Qspan Register Chip Select
D[31:0]	See Table 4.10	See Table 4.10	B	TTL	3S	Hi-Z	8	-8	8	-8	QBus	Data Lines
$\overline{DACK}$/ $\overline{SDACK}$	132	K19	I	TTL	-	Hi-Z	-	-	-	-	QBus	IDMA Acknowledge
DEVSEL#	81	Y11	B	PCI	3S	Hi-Z	-	-	6	-2	PCI	Device Select
$\overline{DONE}$	133	K17	I	TTL	-	Hi-Z	-	-	-	-	QBus	IDMA Done
$\overline{DREQ}$	135	J17	O	TTL	2S	High	8	-8	12	-12	QBus	IDMA Request
$\overline{DS}$	177	A13	O	TTL	3S	Hi-Z	8	-8	12	-12	QBus	Data Strobe
$\overline{DSACK0}$	19	J4	B	TTL	3S	Hi-Z	8	-8	12	-12	QBus	Data And Size Acknowledge 0

* All outputs except for SCL are tristated when TMODE is 11. Please refer to Table 2.46 on page 2-78 for more information.

Table 4.7 Pin List for QSpan Signals (Continued)

Pin Name	Pin Number		Type	Input Type	Output Type	Reset State	3.3 V - 256 PBGA		5 V - 208 PQFP		Inter-face	Signal Description
	PQFP	PBGA					I_{OL} (mA)	I_{OH} (mA)	I_{OL} (mA)	I_{OH} (mA)		
$\overline{\text{DSACK1}}$/ $\overline{\text{TA}}$	18	H1	B	TTL	3S	Hi-Z	8	-8	12	-12	QBus	Data And Size Acknowledge 1/transfer Acknowledge
ENID	129	J18	I (PD)	TTL	-	Hi-Z	-	-	-	-		EEPROM Loading Reset Options
FRAME#	76	W10	B	PCI	3S	Hi-Z	-	-	-	-	PCI	Cycle Frame
GNT#	45	U2	I	PCI (3.3V)/ TTL (5.0V	-	Hi-Z	-	-	-	-	PCI	Grant
$\overline{\text{HALT}}$/ $\overline{\text{TRETRY}}$	44	T1	B	TTL	3S	Hi-Z	8	-8	12	-12	QBus	Halt/Tretry
IDSEL	101	Y16	I	PCI (3.3V)/ TTL (5.0V	-	Hi-Z	-	-	-	-	PCI	Initialization Device Select
IMSEL	24	L1	I	TTL	-	Hi-Z	-	-	-	-	QBus	Slave Image Select
INT#	136	T18	B	PCI (3.3V)/ TTL (5.0V	OD	Hi-Z	8	-8	12	-12	PCI	Interrupt
IRDY#	79	V10	B	PCI	3S	Hi-Z	-	-	-	-	PCI	Initiator Ready
PAR	85	Y12	B	PCI	3S	Hi-Z	-	-	-	-	PCI	Parity
PCLK	48	W11	I	PCI (3.3V)/ TTL (5.0V	-	Hi-Z	-	-	-	-	PCI	PCI Clock
PERR#	84	U11	B	PCI	3S	Hi-Z	-	-	-	-	PCI	Parity Error
QCLK	153	A10	I	TTL	-	Hi-Z	-	-	-	-	QBus	QBus clock
$\overline{\text{QINT}}$	134	J19	B	TTL	OD	Hi-Z	8	-8	8	-8	QBus	Interrupt
REQ#	35	T3	O	PCI	3S	Hi-Z	-	-	-	-	PCI	Request
$\overline{\text{RESETI}}$	29	M3	I	TTL Sch.	-	Hi-Z	-	-	-	-	QBus	Reset Input
$\overline{\text{RESETO}}$	30	N1	O	TTL	OD	Please See [2.8]	8	-8	8	-8	QBus	Reset Output
$\overline{\text{RETRY}}$	See $\overline{\text{HALT}}$/$\overline{\text{TRETRY}}$ above											
RST#	46	P4	I	PCI	-	Hi-Z	-	-	-	-	PCI	Reset
R/$\overline{\text{W}}$	17	G3	B	TTL (5.0V)	3S	Hi-Z	8	-8	12	-12	QBus	Read/Write
SCL	23	K3	O	TTL	*2S	Low	8	-8	8	-8	EEPROM	Serial Clock

* All outputs except for SCL are tristated when TMODE is 11. Please refer to Table 2.46 on page 2-78 for more information.

Table 4.7 Pin List for QSpan Signals (Continued)

Pin Name	Pin Number		Type	Input Type	Output Type	Reset State	3.3 V - 256 PBGA		5 V - 208 PQFP		Inter-face	Signal Description
	PQFP	PBGA					I_{OL} (mA)	I_{OH} (mA)	I_{OL} (mA)	I_{OH} (mA)		
SDA	22	J1	B	TTL	OD	Hi-Z	8	-8	8	-8	EEPROM	Serial Data
$\overline{SDACK}$	See $\overline{DACK}$/$\overline{SDACK}$ above											
SERR#	137	U20	B	TTL	OD	Hi-Z	8	-8	12	-12	PCI	System Error
SIZ[0]	21	J2	B	TTL	3S	Hi-Z	8	-8	8	-8	QBus	Size
SIZ[1]	20	J3				Hi-Z					QBus	Size
STOP#	83	V11	B	PCI	3S	Hi-Z	-	-	-	-	PCI	Stop
$\overline{TA}$	See $\overline{DSACK1}$/$\overline{TA}$ above											
TC[0]	121	P18	B	TTL	3S	Hi-Z	8	-8	8	-8	QBus	Transaction Code
TC[1]	122	P19				Hi-Z						
TC[2]	123	P20				Hi-Z						
TC[3]	124	N18				Hi-Z						
TCK[b]	-	J20	I	CMOS	-	Hi-Z	-	-	-	-	JTAG	JTAG Test Clock Input
TDI[b]	-	L18	I	CMOS (PU)	-	Hi-Z	-	-	-	-	JTAG	JTAG Test Data Input
TDO	-	M19	O	TTL	3S	Hi-Z	8	-8	-	-	JTAG	JTAG Test Data Output
$\overline{TEA}$	See $\overline{BERR}$/$\overline{TEA}$ above											
$\overline{TIP}$	See $\overline{BURST}$/$\overline{TIP}$ above											
TMODE[0]	28	M1	I	TTL	-	Hi-Z	-	-	-	-		Test Mode
TMODE[1]	27	L3				Hi-Z						
TMS[b]	-	K20	I	CMOS (PU)	-	Hi-Z	-	-	-	-	JTAG	JTAG Mode Select
TRDY#	80	Y10	B	PCI	3S	Hi-Z	-	-	-	-	PCI	Target Ready
$\overline{TRST}$[b]	-	K18	I	CMOS (PU)	-	Hi-Z	-	-	-	-	JTAG	JTAG Test Reset
$\overline{TS}$	32	P2	B	TTL	TS	Hi-Z	8	-8	12	-12	QBus	Transfer Start
VIO[c]		R3	I	VIO		Hi-Z					PCI	PCI Buffer Voltage Select Input
* All outputs except for SCL are tristated when TMODE is 11. Please refer to Table 2.46 on page 2-78 for more information.												

a. See Table 4.2 on page 4-3 and Table 4.5 on page 4-5 for AC characteristics of PCI signals.

b. All JTAG pins (TCK, TDI, TMS and $\overline{TRST}$) are not 5V tolerant.

c. R3 is the VIO pin. The QSpan PBGA measures the voltage on this pin to determine whether it is operating in a 3.3 volt or 5 volt signalling environment. This feature is only available in the 3.3 volt PBGA version of the QSpan.

Table 4.8 PCI Bus Address/Data Pins

Signal	PQFP	PBGA		Signal	PQFP	PBGA
AD0	116	V20		AD16	74	W9
AD1	115	U18		AD17	72	U9
AD2	113	W20		AD18	71	Y8
AD3	112	V19		AD19	70	W8
AD4	111	Y20		AD20	67	V8
AD5	109	V18		AD21	66	Y7
AD6	108	W18		AD22	65	V7
AD7	107	U14		AD23	63	U5
AD8	99	V14		AD24	61	W3
AD9	97	Y15		AD25	59	Y2
AD10	96	W14		AD26	58	Y1
AD11	95	Y14		AD27	43	W2
AD12	93	V13		AD28	42	W1
AD13	92	W13		AD29	40	U3
AD14	91	Y13		AD30	39	V2
AD15	88	V12		AD31	38	T4

Table 4.9 QBus Address Pins

Signal	PQFP	PBGA		Signal	PQFP	PBGA
A0	138	G20		A16	188	D10
A1	139	G19		A17	189	A9
A2	142	F20		A18	190	C9
A3	143	F19		A19	191	D9
A4	144	E20		A20	194	B8
A5	145	G17		A21	195	C8
A6	165	C17		A22	196	A7
A7	166	B17		A23	197	B7
A8	167	A18		A24	198	A6
A9	168	B16		A25	199	C7
A10	169	C16		A26	9	E3
A11	170	C14		A27	10	E1
A12	173	B14		A28	11	F2
A13	174	A14		A29	12	G2
A14	175	C13		A30	13	H3
A15	176	B13		A31	14	H2

Table 4.10 QBus Data Pins

Signal	PQFP	PBGA		Signal	PQFP	PBGA
D0	146	F18		D16	184	C11
D1	147	E19		D17	185	A11
D2	148	E18		D18	186	B10
D3	149	E17		D19	187	C10
D4	150	C20		D20	200	B6
D5	151	D18		D21	201	D7
D6	159	B20		D22	202	A3
D7	160	B19		D23	203	C4
D8	161	C18		D24	204	D5
D9	162	A20		D25	205	B3
D10	163	B18		D26	3	A2
D11	164	A19		D27	4	C2
D12	178	D12		D28	5	B1
D13	179	C12		D29	6	D3
D14	180	B12		D30	7	C1
D15	181	B11		D31	8	E4

Table 4.11 Pin Assignments for Power (V_{DD})

PQFP				PBGA		
2	102	192		D2	D15	T2
15	103	206		V3	F4	D20
26	106	207		U12	F17	B2
36	119			Y19	K4	
50	130			U19	L17	
51	140			G18	R4	
54	154			C19	R17	
56	155			D16	U6	
69	158			A5	U10	
78	171			D6	U15	
90	182			D11	E2	

Table 4.12 Pin Assignments for Ground (V_{SS})

PQFP					PBGA		
1	64	110	208		F1	L20	N17
16	68	114			U1	A12	U4
25	73	120			V1	A8	U8
37	77	131			V9	A1	U13
41	82	141			H19	D4	U17
47	86	152			D19	D8	
49	89	156			D1	D13	
52	94	157			P3	D17	
53	98	172			W7	H4	
55	104	183			W19	H17	
60	105	193			T17	N4	

Table 4.13 No Connect Pin Assignments[a]

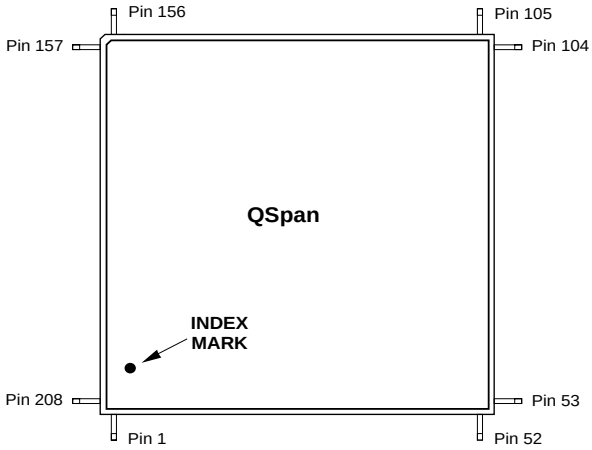
PBGA					
A4	C5	H20	N2	V4	W16
A15	C6	K1	P1	V5	W17
A16	C15	K2	P17	V6	Y3
A17	D14	L2	R18	V15	Y4
B4	F3	L4	R19	V16	Y5
B5	G1	L19	T20	V17	Y6
B9	G4	M2	U7	W5	Y17
B15	H18	M4	U16	W6	Y18
C3					

a. Route all N/C signals out to vias on your board to allow for future migration to new QSpan variants.

Table 4.14 Pin-Out of QSpan 256-PBGA

A1. VSS
A2. D[26]
A3. D[22]
A4. N/C
A5. VDD
A6. A[24]
A7. A[22]
A8. VSS
A9. A[17]
A10. QCLK
A11. D[17]
A12. VSS
A13. $\overline{DS}$
A14. A[13]
A15. N/C
A16. N/C
A17. N/C
A18. A[8]
A19. D[11]
A20. D[9]
B1. D[28]
B2. VDD
B3. D[25]
B4. N/C
B5. N/C
B6. D[20]
B7. A[23]
B8. A[20]
B9. N/C
B10. D[18]
B11. D[15]
B12. D[14]
B13. A[15]
B14. A[12]
B15. N/C
B16. A[9]
B17. A[7]
B18. D[10]
B19. D[7]
B20. D[6]
C1. D[30]
C2. D[27]
C3. N/C
C4. D[23]
C5. N/C
C6. N/C
C7. A[25]
C8. A[21]
C9. A[18]
C10. D[19]
C11. D[16]
C12. D[13]
C13. A[14]
C14. A[11]
C15. N/C
C16. A[10]
C17. A[6]
C18. D[8]
C19. VDD
C20. D[4]
D1. VSS
D2. VDD
D3. D[29]
D4. VSS
D5. D[24]
D6. VDD
D7. D[21]
D8. VSS
D9. A[19]
D10. A[16]
D11. VDD
D12. D[12]
D13. VSS
D14. N/C
D15. VDD
D16. VDD
D17. VSS
D18. D[5]
D19. VSS
D20. VDD
E1. A[27]
E2. VDD
E3. A[26]
E4. D[31]
E17. D[3]
E18. D[2]
E19. D[1]
E20. A[4]
F1. VSS
F2. A[28]
F3. N/C
F4. VDD
F17. VDD
F18. D[0]
F19. A[3]
F20. A[2]
G1. N/C
G2. A[29]
G3. R/$\overline{W}$
G4. N/C
G17. A[5]
G18. VDD
G19. A[1]
G20. A[0]
H1. $\overline{DSACK1}$/$\overline{TA}$
H2. A[31]
H3. A[30]
H4. VSS
H17. VSS
H18. N/C
H19. VSS
H20. N/C
J1. SDA
J2. SIZ[0]
J3. SIZ[1]
J4. $\overline{DSACK0}$
J17. $\overline{DREQ}$
J18. ENID
J19. $\overline{QINT}$
J20. TCK
K1. N/C
K2. N/C
K3. SCL
K4. VDD
K17. $\overline{DONE}$
K18. $\overline{TRST}$
K19. $\overline{DACK}$/$\overline{SDACK}$
K20. TMS
L1. IMSEL
L2. N/C
L3. TMODE[1]
L4. N/C
L17. VDD
L18. TDI
L19. N/C
L20. VSS
M1. TMODE[0]
M2. N/C
M3. $\overline{RESETI}$
M4. N/C
M17. $\overline{BDIP}$
M18. $\overline{BG}$
M19. TDO
M20.BM_EN/
$\overline{FIFO\_RDY}$
N1. $\overline{RESETO}$
N2. N/C
N3. $\overline{BURST}$/$\overline{TIP}$
N4. VSS
N17. VSS
N18. TC[3]
N19. $\overline{BERR}$/$\overline{TEA}$
N20. $\overline{AS}$
P1. N/C
P2. $\overline{TS}$
P3. VSS
P4. RST#
P17. N/C
P18. TC[0]
P19. TC[1]
P20. TC[2]
R1. $\overline{BB}$/$\overline{BGACK}$
R2. $\overline{BR}$
R3. VIO
R4. VDD
R17. VDD
R18. N/C
R19. N/C
R20. $\overline{CSPCI}$
T1. $\overline{HALT}$/$\overline{TRETRY}$
T2. VDD
T3. REQ#
T4. AD[31]
T17. VSS
T18. INT#
T19. $\overline{CSREG}$
T20. N/C
U1. VSS
U2. GNT#
U3. AD[29]
U4. VSS
U5. AD[23]
U6. VDD
U7. N/C
U8. VSS
U9. AD[17]
U10. VDD
U11. PERR#
U12. VDD
U13. VSS
U14. AD[7]
U15. VDD
U16. N/C
U17. VSS
U18. AD[1]
U19. VDD
U20. SERR#
V1. VSS
V2. AD[30]
V3. VDD
V4. N/C
V5. N/C
V6. N/C
V7. AD[22]
V8. AD[20]
V9. VSS
V10. IRDY#
V11. STOP#
V12. AD[15]
V13. AD[12]
V14. AD[8]
V15. N/C
V16. N/C
V17. N/C
V18. AD[5]
V19. AD[3]
V20. AD[0]
W1. AD[28]
W2. AD[27]
W3. AD[24]
W4. CBE[3]
W5. N/C
W6. N/C
W7. VSS
W8. AD[19]
W9. AD[16]
W10. FRAME#
W11. PCLK
W12. CBE[1]
W13. AD[13]
W14. AD[10]
W15. CBE[0]
W16. N/C
W17. N/C
W18. AD[6]
W19. VSS
W20. AD[2]
Y1. AD[26]
Y2. AD[25]
Y3. N/C
Y4. N/C
Y5. N/C
Y6. N/C
Y7. AD[21]
Y8. AD[18]
Y9. CBE[2]
Y10. TRDY#
Y11. DEVSEL#
Y12. PAR
Y13. AD[14]
Y14. AD[11]
Y15. AD[9]
Y16. IDSEL
Y17. N/C
Y18. N/C
Y19. VDD
Y20. AD[4]

Table 4.15 Pin-Out of QSpan 208-Pin PQFP



(Top View)

1. VSS
2. VDD
3. D[26]
4. D[27]
5. D[28]
6. D[29]
7. D[30]
8. D[31]
9. A[26]
10. A[27]
11. A[28]
12. A[29]
13. A[30]
14. A[31]
15. VDD
16. VSS
17. R/$\overline{W}$
18. $\overline{DSACK1}$/$\overline{TA}$
19. $\overline{DSACK0}$
20. SIZ[1]
21. SIZ[0]
22. SDA
23. SCL
24. IMSEL
25. VSS
26. VDD
27. TMODE[1]
28. TMODE[0]
29. $\overline{RESETI}$
30. $\overline{RESETO}$
31. $\overline{BURST}$/$\overline{TIP}$
32. $\overline{TS}$
33. $\overline{BB}$/$\overline{BGACK}$
34. $\overline{BR}$
35. REQ#
36. VDD
37. VSS
38. AD[31]
39. AD[30]
40. AD[29]
41. VSS
42. AD[28]
43. AD[27]
44. $\overline{HALT}$/ $\overline{TRETRY}$
45. GNT#
46. RST#
47. VSS
48. PCLK
49. VSS
50. VDD
51. VDD
52. VSS
53. VSS
54. VDD
55. VSS
56. VDD
57. BM_EN/ $\overline{FIFO\_RDY}$
58. AD[26]
59. AD[25]
60. VSS
61. AD[24]
62. C/BE[3]
63. AD[23]
64. VSS
65. AD[22]
66. AD[21]
67. AD[20]
68. VSS
69. VDD
70. AD[19]
71. AD[18]
72. AD[17]
73. VSS
74. AD[16]
75. C/BE[2]
76. FRAME#
77. VSS
78. VDD
79. IRDY#
80. TRDY#
81. DEVSEL#
82. VSS
83. STOP#
84. PERR#
85. PAR
86. VSS
87. C/BE[1]
88. AD[15]
89. VSS
90. VDD
91. AD[14]
92. AD[13]
93. AD[12]
94. VSS
95. AD[11]
96. AD[10]
97. AD[9]
98. VSS
99. AD[8]
100. C/BE[0]
101. IDSEL
102. VDD
103. VDD
104. VSS
105. VSS
106. VDD
107. AD[7]
108. AD[6]
109. AD[5]
110. VSS
111. AD[4]
112. AD[3]
113. AD[2]
114. VSS
115. AD[1]
116. AD[0]
117. $\overline{CSREG}$
118. $\overline{CSPCI}$
119. VDD
120. VSS
121. TC[0]
122. TC[1]
123. TC[2]
124. TC[3]
125. $\overline{BERR}$/$\overline{TEA}$
126. $\overline{AS}$
127. $\overline{BDIP}$
128. $\overline{BG}$
129. ENID
130. VDD
131. VSS
132. $\overline{DACK}$/ $\overline{SDACK}$
133. DONE
134. $\overline{QINT}$
135. $\overline{DREQ}$
136. INT#
137. SERR#
138. A[0]
139. A[1]
140. VDD
141. VSS
142. A[2]
143. A[3]
144. A[4]
145. A[5]
146. D[0]
147. D[1]
148. D[2]
149. D[3]
150. D[4]
151. D[5]
152. VSS
153. QCLK
154. VDD
155. VDD
156. VSS
157. VSS
158. VDD
159. D[6]
160. D[7]
161. D[8]
162. D[9]
163. D[10]
164. D[11]
165. A[6]
166. A[7]
167. A[8]
168. A[9]
169. A[10]
170. A[11]
171. VDD
172. VSS
173. A[12]
174. A[13]
175. A[14]
176. A[15]
177. $\overline{DS}$
178. D[12]
179. D[13]
180. D[14]
181. D[15]
182. VDD
183. VSS
184. D[16]
185. D[17]
186. D[18]
187. D[19]
188. A[16]
189. A[17]
190. A[18]
191. A[19]
192. VDD
193. VSS
194. A[20]
195. A[21]
196. A[22]
197. A[23]
198. A[24]
199. A[25]
200. D[20]
201. D[21]
202. D[22]
203. D[23]
204. D[24]
205. D[25]
206. VDD
207. VDD
208. VSS

Appendix A Registers

The 4 Kbytes of QSpan Control and Status Registers (QCSRs) facilitate host system configuration and allow the user to control QSpan operational characteristics. The QCSRs are functionally divided into two groups: the PCI Configuration Registers and the QSpan Device Specific Registers. All of the QCSR space is accessible from both the PCI bus and the QBus.

Table A.1 lists the QSpan registers by address offset. The tables following the register map (Table A.1 to Table A.51) provide detailed descriptions of each register. Registers are also described in 'The Register Channel' on page 2-58.

For a full description of the PCI Configuration registers see the PCI 2.1 Specification Revision.

Abbreviations

G_RST:	General reset: contents can be reset by either the PCI RST# or the QBus $\overline{\text{RESETI}}$
N/A:	Not applicable.
PCI_RST:	PCI Reset Only: Contents can only be reset by PCI RST#
0x:	Hexadecimal prefix (binary numbers have no prefix)

The bit combinations listed as "Reserved" must not be set to “1”. All bits listed as "Reserved" must read back a value of zero.

Table A.1 Register Map

Address Offset	Register Name	Page	R/W	Description
0x000	PCI_ID	A-4	R	PCI Configuration Space ID Register
0x004	PCI_CS	A-5	R/W	PCI Configuration Space Control and Status Register
0x008	PCI_CLASS	A-7	R	PCI Configuration Class Register
0x00C	PCI_MISC0	A-8	R/W	PCI Configuration Miscellaneous 0 Register
0x010	PCI_BSM	A-9	R/W	PCI Configuration Base Address for Memory Register
0x014	PCI Unimplemented			
0x018	PCI_BST0	A-10	R/W	PCI Configuration Base Address for Target 0 Register
0x01C	PCI_BST1	A-12	R/W	PCI Configuration Base Address for Target 1 Register
0x020	PCI Unimplemented			
0x024	PCI Unimplemented			
0x02C	PCI_SID	A-14	R	PCI Configuration Subsystem ID Register
0x030	PCI_BSROM	A-15	R/W	PCI Configuration Expansion ROM Base Address Register
0x034	PCI Reserved			
0x038	PCI Reserved			
0x03C	PCI_MISC1	A-17	R/W	PCI Configuration Miscellaneous 1 Register
0x040-0x0FF	PCI Unimplemented			
0x100	PBTI0_CTL	A-18	R/W	PCI Bus Target Image 0 Control Register
0x104	PBTI0_ADD	A-20	R/W	PCI Bus Target Image 0 Address Register
0x108-10C	QSpan Reserved			
0x110	PBTI1_CTL	A-22	R/W	PCI Bus Target Image 1 Control Register
0x114	PBTI1_ADD	A-24	R/W	PCI Bus Target Image 1 Address Register
0x118-0x138	QSpan Reserved			
0x13C	PBROM_CTL	A-26	R/W	PCI Bus Expansion ROM Control Register
0x140	PB_ERRCS	A-27	R/W	PCI Bus Error Control and Status Register
0x144	PB_AERR	A-28	R	PCI Bus Address Error Log Register
0x148	PB_DERR	A-29	R	PCI Bus Data Error Log Register
0x14C-3FC	QSpan Reserved			
0x400	IDMA_CS	A-30	R/W	IDMA Control and Status Register
0x404	IDMA_ADD	A-33	R/W	IDMA Address Register
0x408	IDMA_CNT	A-34	R/W	IDMA Transfer Count Register
0x40C-0x4FC	QSpan Reserved			
0x500	CON_ADD	A-35	R/W	Configuration Address Register
0x504	CON_DATA	A-37	R/W	Configuration Data Register
0x508	IACK_GEN	A-38	R/W	IACK Cycle Generator Register
0x50C-0x5FC	QSpan Reserved			
0x600	INT_STAT	A-39	R/W	Interrupt Status Register
0x604	INT_CTL	A-41	R/W	Interrupt Control Register
0x608	INT_DIR	A-43	R/W	Interrupt Direction Control Register

Table A.1 Register Map CONT'D

Address Offset	Register Name	Page	R/W	Description
0x60C	INT-CTL2	A-45	R/W	Interrupt Control Register
0x6010-0x7FC	QSpan Reserved			
0x800	MISC_CTL	A-46	R/W	Miscellaneous Control and Status Register
0x804	EEPROM_CS	A-49	R/W	EEPROM Control and Status Register
0x808-0xEFC	QSpan Reserved			
0xF00	QBSI0_CTL	A-50	R/W	QBus Slave Image 0 Control Register
0xF04	QBSI0_AT	A-52	R/W	QBus Slave Image 0 Address Translation Register
0xF08-0xF0C	QSpan Reserved			
0xF10	QBSI1_CTL	A-53	R/W	QBus Slave Image 1 Control Register
0xF14	QBSI1_AT	A-55	R/W	QBus Slave Image 1 Address Translation Register
0xF18-0xF7C	QSpan Reserved			
0xF80	QB_ERRCS	A-56	R/W	QBus Error Log Control and Status Register
0xF84	QB_AERR	A-57	R	QBus Address Error Log Register
0xF88	QB_DERR	A-58	R	QBus Data Error Log Register
0xF8C-0xFFC	QSpan Reserved			

Table A.2 PCI Configuration Space ID Register

Register Name: PCI_ID	Register Offset:000

Bits	Function
31-24	DID
23-16	DID
15-08	VID
07-00	VID

PCI_ID Description

Name	Type	Reset By	Reset State	Function
DID[15:0]	R	N/A	0x0860	Device ID Tundra allocated Device Identifier Device ID is 0x0860
VID[15:0]	R	N/A	0x10E3	Vendor ID PCI SIG allocated vendor Identifier

Table A.3 PCI Configuration Space Control and Status Register

Register Name: PCI_CS	Register Offset:004

Bits	Function							
31-24	D_PE	S_SERR	R_MA	R_TA	S_TA	DEVSEL		DP_D
23-16	TFBBC	UDFS	DEV66	PCI Reserved				
15-08	PCI Reserved						MFFBC	SERR_EN
07-00	WAIT	PERESP	VGAPS	MWI_EN	SC	BM	MS	IOS

PCI_CS Description

Name	Type	Reset By	Reset State	Function
D_PE	R/Write 1 to Clear	PCI_RST	0	Detected Parity Error 0 = No parity error, 1 = Parity error This bit is set by the QSpan whenever: the PCI Master Module detects a data parity error, or the PCI Target Module detects address or data parity errors.
S_SERR	R/Write 1 to Clear	PCI_RST	0	Signaled SERR# 0 = SERR# not asserted, 1 = SERR# asserted. The QSpan as PCI target sets this bit when it asserts SERR# to signal an address parity error. SERR_EN must be set before SERR# can be asserted.
R_MA	R/Write 1 to Clear	PCI_RST	0	Received Master-Abort 0 = QSpan did not generate Master-Abort, 1 = QSpan generated Master-Abort The QSpan sets this bit when a transaction it initiated had to be terminated with a Master-Abort.
R_TA	R/Write 1 to Clear	PCI_RST	0	Received Target-Abort 0 = Master did not detect Target-Abort, 1 = Master detected Target-Abort. The QSpan sets this bit when a transaction it initiated was terminated with a Target-Abort.
S_TA	R/Write 1 to Clear	PCI_RST	0	Signaled Target-Abort 0 = Target did not terminate transaction with Target-Abort, 1 = Target terminated transaction with Target-Abort.
DEVSEL[1:0]	R	N/A	01	DEVSEL Device Select Timing The QSpan is a medium speed device.
DP_D	R/Write 1 to Clear	PCI_RST	0	Data Parity Detected 0 = Master Module did not detect/generate data parity error, 1 = Master Module detected/generated data parity error. The QSpan sets this bit if the PERESP bit is set and either (a) it is the master of transaction in which it asserts PERR#, or (b) the addressed target asserts PERR#.
TFBBC	R	N/A	1	Target Fast Back-to-Back Capable QSpan can accept fast back-to-back transactions from different agents.

PCI_CS Description[CONT'D]

Name	Type	Reset By	Reset State	Function
UDFS	R	N/A	0	User Definable Features Support The QSpan does not support User Definable Features.
DEV66	R	N/A	0	Device 66 MHz Capable The QSpan is not capable of running at 66 MHz. It is a 33 MHz capable device.
MFBBC	R	N/A	0	Master Fast Back-to-Back Enable QSpan does not generate fast back-to-back transfers.
SERR_EN	R/W	PCI_RST	0	SERR# Enable 0 = Disable SERR# driver, 1 = Enable SERR# driver. Setting this and PERESP allows the QSpan to report address parity errors with SERR# as PCI target
WAIT	R	N/A	0	Wait Cycle Control 0 = No address/data stepping
PERESP	R/W	PCI_RST	0	Parity Error Response 0 = Disable, 1 = Enable Controls the QSpan response to data and address parity errors. When enabled, PERR# is asserted and the DP_D bit is set in response to data parity errors. When this bit and SERR_EN are set, the QSpan reports address parity errors on SERR#. QSpan parity generation (i.e., its assertion of PAR#) is unaffected by this bit.
VGAPS	R	N/A	0	VGA Palette Snoop 0 = Disable
MWI_EN	R	N/A	0	Memory Write and Invalidate Enable 0 = Disable The QSpan can generate a Memory Write and Invalidate command as PCI master during IDMA transfers - controlled by CMD bit IDMA_CS
SC	R	N/A	0	Special Cycles 0 = Disable The QSpan never responds to special cycles as PCI target.
BM	R/W	PCI_RST	Reset Option	Bus Master 0 = Disable, 1 = Enable Enables the QSpan to become PCI bus master. This can be set as a reset option. See “PCI Bus Master Reset Options” on page 2-77.
MS	R/W	PCI_RST	0	Memory Space 0 = Disable, 1 = Enable Enables the QSpan target to accept Memory space accesses.
IOS	R/W	PCI_RST	0	IO Space 0 = Disable, 1 = Enable Enables the QSpan target to accept I/O space accesses.

Table A.4 PCI Configuration Class Register

Register Name: PCI_CLASS	Register Offset:008

Bits	Function
31-24	BASE
23-16	SUB
15-08	PROG
07-00	RID

PCI_CLASS Description

Name	Type	Reset By	Reset State	Function
BASE[7:0]	R	N/A	0x06	Base Class Code
SUB[7:0]	R	N/A	0x80	Sub Class Code
PROG[7:0]	R	N/A	0x0	Programming Interface
RID[7:0]	R	N/A	0x02	Revision ID

Table A.5 PCI Configuration Miscellaneous 0 Register

Register Name: PCI_MISC0	Register Offset:00C

Bits	Function						
31-24	BISTC	SBIST	PCI Reserved		CCODE		
23-16	MFUNCT	LAYOUT					
15-08	LTIMER				0	0	0
07-00	0	0	0	0	CLINE	0	0

PCI_MISC0 Description

Name	Type	Reset By	Reset State	Function
BISTC	R	N/A	0	BIST Capable 0 = Device not BIST capable
SBIST	R	N/A	0	Start BIST 0 = Device not BIST capable
CCODE[3:0]	R	N/A	0	Completion Code 0 = Device not BIST capable
MFUNCT	R	N/A	0	Multifunction Device 0 = Device not Multifunction
LAYOUT[6:0]	R	N/A	0	Configuration Space Layout
LTIMER[4:0]	R/W	PCI_RST	0	Latency Timer Number of PCI bus clocks before transaction is terminated (8 clocks)
CLINE[1:0]	R/W	PCI_RST	0	CacheLine Size 00 = treated as 01, 01 = 4 x 32-bit word, 10 = 8 x 32-bit word All other combinations written to this entry will return a value of zero.

This field is optimized by optimizing the IDMA TCI read performance with long read bursts from a slow PCI target. Increasing the LTIMER[4:0] value with a slaves PCI target increases average read burst length.

CLINE[1:0] determines how the PCI Target module accepts burst write data (see page 'Acceptance of Burst Writes by the PCI Target Module' on page 2-39). It also determines how the IDMA Channel sinks data on and sources data from the PCI bus. If CLINE[1:0] is set to 00, the QSpan treats it as if it were set to 01.

Table A.6 PCI Configuration Base Address for Memory Register

Register Name: PCI_BSM	Register Offset:010

Bits	Function							
31-24	BA							
23-16	BA							
15-08	BA				0	0	0	0
07-00	0	0	0	0	0	0	0	SPACE

PCI_BSM Description

Name	Type	Reset By	Reset State	Function
BA[31:12]	R/W	NONE	0	Base Address
SPACE	R	N/A	0	PCI Bus Address Space 0 = Memory Space

This register specifies the 4 KByte aligned base address of the QSpan register space on PCI in Memory Space. The QSpan register space is only 4 KBytes, therefore the PCI address lines [11:0] are used to select the QSpan register.

A write must occur to this register before the QSpan register space can be accessed from the PCI bus. This write can be performed with a PCI configuration transaction or a QBus register access.

Table A.7 PCI Configuration Base Address for Target 0 Register

Register Name: PCI_BST0	Register Offset:018

Bits	Function							
31-24	BA							
23-16	BA							
15-08	0	0	0	0	0	0	0	0
07-00	0	0	0	0	0	0	0	PAS

PCI_BST0 Description

Name	Type	Reset By	Reset State	Function
BA[31:16]	See Below	PCI_RST	0	Base Address of PCI bus Target Image 0
PAS	R	PCI_RST	See Below	PCI Bus Address Space 0 = Memory Space, 1 = I/O Space

This register is only enabled if the state of the SDA or ENID pin is latched as a logic high during PCI reset and bit-5 of byte-7 of the EEPROM is “1”—see “PCI Bus Target Image 0 (PCI_BST0 and PBTI0_CTL registers)” on page 2-72. If this register is not enabled (i.e., there is no EEPROM or bit-5 of byte-7 of the EEPROM is “0”), the entire register is read only, and reads return all zeros.

If enabled this register specifies the Base Address and PCI Bus Address Space settings for PCI Bus Target Image 0. The Base Address is used during transaction decoding; it specifies the contiguous PCI bus address line values compared by the QSpan during PCI bus address phases. The number of address lines compared for this image is based on the value of the Block Size field in the PBTI0_CTL register (page A-18). See “Transaction Decoding” on page 2-27.

If this register is enabled, the number of writable bits in BA[31:16] is determined by the Block Size field of the PBTI0_CTL register. After power-up the serial EEPROM contents have been loaded and a PCI host may write all 1’s to the BA field of this register and the number of 1’s that are read back can be used to compute the block size of the image (Block Size = 64Kbytes * 2^N): for example, if the Block Size is 64k (BS=0000) then the BA field will be 0xFFFF; if the Block Size is 2G (BS = 1111), then the BA field will be 0x8000.

Table A.8 PCI Address Lines Compared as a Function of Block Size

BS	Block Size	Address Lines Compared
0000	64K	AD31-AD16
0001	128K	AD31-AD17
0010	256K	AD31-AD18
0011	512K	AD31-AD19
0100	1M	AD31-AD20
0101	2M	AD31-AD21
0110	4M	AD31-AD22
0111	8M	AD31-AD23
1000	16M	AD31-AD24
1001	32M	AD31-AD25
1010	64M	AD31-AD26
1011	128M	AD31-AD27
1100	256M	AD31-AD28
1101	512M	AD31-AD29
1110	1G	AD31-AD30
1111	2G	AD31

Table A.9 PCI Configuration Base Address for Target 1 Register

Register Name: PCI_BST1	Register Offset:01C

Bits	Function							
31-24	BA							
23-16	BA							
15-08	0	0	0	0	0	0	0	0
07-00	0	0	0	0	0	0	0	PAS

PCI_BST1 Description

Name	Type	Reset By	Reset State	Function
BA[31:16]	See Below	PCI_RST	0	Base Address of PCI Target Image 1
PAS	R	PCI_RST	See Below	PCI Bus Address Space of PCI Target Image 1 0 = Memory Space, 1 = I/O Space

This register is only enabled if the state of the SDA or ENID pin is latched as a logic high during PCI reset and bit-7 of byte-8 of the EEPROM is “1”. If this register is not enabled (i.e., there is no EEPROM or bit-7 of byte-8 of the EEPROM is “0”), the entire register is read only, and reads return all zeros.

If enabled this register specifies the Base Address and PAS fields for PCI Bus Target Image 1. The Base Address is used during transaction decoding; it specifies the contiguous PCI bus address line values compared by the QSpan during PCI bus address phases. The number of address lines compared for this image is based on the value of the Block Size field in the PBTI1_CTL register. See “Transaction Decoding” on page 2-27.

If this register is enabled, the number of writable bits in BA[31:16] is determined by the Block Size field of the PBTI1_CTL register (page A-22). After power-up the serial EEPROM contents have been loaded and a PCI host may write all 1’s to the BA field of this register and the number of 1’s that are read back can be used to compute the block size of the image (Block Size = 64Kbytes * 2^N): for example, if the Block Size is 64k (BS=0000) then the BA field will be 0xFFFF; if the Block Size is 2G (BS = 1111), then the BA field will be 0x8000.

Table A.10 PCI Address Lines Compared as a Function of Block Size

BS	Block Size	Address Lines Compared
0000	64K	AD31-AD16
0001	128K	AD31-AD17
0010	256K	AD31-AD18
0011	512K	AD31-AD19
0100	1M	AD31-AD20
0101	2M	AD31-AD21
0110	4M	AD31-AD22
0111	8M	AD31-AD23
1000	16M	AD31-AD24
1001	32M	AD31-AD25
1010	64M	AD31-AD26
1011	128M	AD31-AD27
1100	256M	AD31-AD28
1101	512M	AD31-AD29
1110	1G	AD31-AD30
1111	2G	AD31

Table A.11 PCI Configuration Subsystem ID Register

Register Name: PCI_SID	Register Offset:02C

Bits	Function
31-24	SID
23-16	SID
15-08	SVID
07-00	SVID

PCI_SID Description

Name	Type	Reset By	Reset State	Function
SID[15:0]	R/W	PCI_RST	See Below	Subsystem ID Values for Subsystem ID are vendor specific.
SVID[15:0]	R/W	PCI_RST	See Below	Subsystem Vendor ID Subsystem Vendor IDs are obtained from the PCI SIG and used to identify the vendor of the add-in board or subsystem.

The Subsystem ID and the Subsystem Vendor ID will be loaded from an external serial EEPROM at the conclusion of the PCI bus reset (RST#), if the state of the SDA or ENID pin is latched as a logic high during the reset. If the state of the SDA and ENID pin is latched as a logic low, the reset state of the register will be all zeros.

The PCI_SID register is initiated from the QBUS, except while the QSpan is updating its contents from the EEPROM. Writes to the PCI_SID register from the PCI bus have no affect on its contents.

Table A.12 PCI Configuration Expansion ROM Base Address Register

Register Name: PCI_BSROM	Register Offset:030

Bits	Function							
31-24	BA							
23-16	BA							
15-08	0	0	0	0	0	0	0	0
07-00	0	0	0	0	0	0	0	EN

PCI_BSROM Description

Name	Type	Reset By	Reset State	Function
BA[31:16]	See Below	PCI_RST	See Below	Expansion ROM Base Address
EN	See Below	PCI_RST	0	Enable Address Decode 0=disable, 1=enable

The number of writable bits in BA[31:16] determines the size of the external QBus Expansion ROM. (See Section 6.2.5.2 of the PCI 2.1 Specification.) The number of bits itself is determined by the Block Size field of the PCI Expansion ROM Control register (PBROM_CTL on page A-26). Thus after power-up a PCI host may write all 1’s to the BA field of this register and the number of 1s that are read-back will indicate the size of the Expansion ROM of the QSpan. See Table A.13 below.

This register is enabled if bit-7 of byte-4 of the EEPROM is latched as a logic “1” (“Destination of EEPROM Bits Read” on page 2-71). If the state of this bit is “0” or if the state of the SDA or ENID pin is latched as a logic low during PCI reset, then all bits in the entire register will be set to zero and will be read only.

The PCI Expansion ROM Base Address register can be written from either bus, if write enabled, except while the QSpan is loading data from an external serial EEPROM. Write to bits in the PCI Expansion ROM Base Address register that are not write enabled will have no effect.

Table A.13 Writable BA bits as a function of Block Size

BS	Block Size	Read/Write Bits
000	64K	BA31-BA16
001	128K	BA31-BA17
010	256K	BA31-BA18
011	512K	BA31-BA19
100	1M	BA31-BA20
101	2M	BA31-BA21
110	4M	BA31-BA22
111	8M	BA31-BA23

Table A.14 PCI Configuration Miscellaneous 1 Register

Register Name: PCI_MISC1	Register Offset:03C

Bits	Function
31-24	MAX_LAT
23-16	MIN_GNT
15-08	INT_PIN
07-00	INT_LINE

PCI_MISC1 Description

Name	Type	Reset By	Reset State	Function
MAX_LAT[7:0]	R	N/A	0	Maximum Latency The device has no special latency requirements.
MIN_GNT[7:0]	R	N/A	0	Minimum Grant The device has no major requirements.
INT_PIN[7:0]	R	N/A	0x01	Interrupt Pin The device uses INTA#
INT_LINE[7:0]	R/W	PCI_RST	0	Interrupt Line

Table A.15 PCI Bus Target Image 0 Control Register

Register Name: PBTI0_CTL	Register Offset:100

Bits	Function					
31-24	EN	QSpan Reserved		BS		
23-16	PREN	BRSTWREN	QSpan Reserved	INVEND	QSpan Reserved	
15-08	TC			DSIZE	QSpan Reserved	
07-00	PWEN	PAS	QSpan Reserved			

PBTI0_CTL Description

Name	Type	Reset By	Reset State	Function
EN	R/W	G_RST	0	Image Enable 0 = Disable, 1 = Enable
BS[3:0]	See Below	PCI_RST	See Below	Block Size (64 Kbyte*2)
PREN	R/W	G_RST	0	Prefetch Read Enable 0 = Disable, 1 = Enable
BRSTWREN	R/W	G_RST	0	Burst Write Enable 0 = Disable, 1 = Enable
INVEND	R/W	G_RST	0	Invert Endian-ness from QB_BOC Setting in MISC_CTL 0 = Use QB_BOC setting, 1 = Invert QB_BOC setting
TC[3:0]	R/W	G_RST	0	QBus Transaction Code User Defined
DSIZE	R/W	G_RST	0	QBus Destination Port Size 00=32-bit, 01=8-bit,10=16-bit, 11=reserved
PWEN	R/W	G_RST	0	Posted Write Enable 0 = Disable, 1 = Enable
PAS	See Below	PCI_RST	See Below	PCI Bus Address Space 0 = PCI Bus Memory Space, 1 = PCI Bus I/O Space

Only PCI bus Memory Space transactions can be posted.

The BS[3:0] and PAS fields can be loaded from an external serial EEPROM. See “PCI Bus Target Image 0 (PCI_BST0 and PBTI0_CTL registers)” on page 2-72. There are two cases:

1. If the fields are loaded from the EEPROM, then the EEPROM determines their reset state, and they become read only. In this case, the PAS bit has the same value as the bit of the same name in the PCI_BST0 register (page A-10).
2. If the BS[3:0] and PAS fields are not loaded from the EEPROM, their reset state is zero, and they are writable (note that in this case the PCI_BST0 register is disabled).

Table A.16 PCI Bus Target Image 0 Address Register

Register Name: PBTI0_ADD	Register Offset:104

Bits	Function
31-24	BA
23-16	BA
15-08	TA
07-00	TA

PBTI0_ADD Description

Name	Type	Reset By	Reset State	Function
BA[31:16]	See Below	PCI_RST	See below	Base Address
TA[31:16]	R/W	G_RST	0	Translation Address

The Base Address specifies the contiguous PCI bus address line values compared by the QSpan during PCI bus address phases. The number of address lines compared for this image is based on the Block Size (programmed in the PBTI0_CTL register, page A-18). The Translation Address specifies the values of the address lines substituted when generating the address for the transaction on the QBus. If no translation is to occur, the Translation Address must be programmed with the same value as that of the Base Address. See “Transaction Decoding” on page 2-27 and “Address Translation” on page 2-30 for more details.

Table A.17 PCI Address Lines Compared as a Function of Block Size

BS	Block Size	Address Lines Compared/Translated
0000	64K	AD31-AD16
0001	128K	AD31-AD17
0010	256K	AD31-AD18
0011	512K	AD31-AD19
0100	1M	AD31-AD20
0101	2M	AD31-AD21
0110	4M	AD31-AD22
0111	8M	AD31-AD23
1000	16M	AD31-AD24
1001	32M	AD31-AD25
1010	64M	AD31-AD26
1011	128M	AD31-AD27
1100	256M	AD31-AD28
1101	512M	AD31-AD29
1110	1G	AD31-AD30
1111	2G	AD31

The read/write type and reset value of the BA[31:16] field depends on whether the PCI_BST0 register (page A-10) is “enabled” (i.e., whether bit-5 of byte-7 of the EEPROM is “1”). Thus there are two cases:

1. If the EEPROM bit was indeed “1”, the BA field of PBTI0_ADD is read only and has the same value as the PCI_BST0 register from reset onwards.
2. If the EEPROM bit was “0” (or the SDA or ENID pin is “0” at PCI reset) then the entire BA field of PBTI0_ADD is readable and writable. A read from this field after reset returns all zeros. In this case, the BA field of this register is independent of the PCI_BST0 register (the PCI_BST0 register does not exist).

The presence of EEPROM does not affect TA[31:16].

Table A.18 PCI Bus Target Image 1 Control Register

Register Name: PBTI1_CTL	Register Offset:110

Bits	Function					
31-24	EN	QSpan Reserved		BS		
23-16	PREN	BRSTWREN	QSpan Reserved	INVEND	QSpan Reserved	
15-08	TC			DSIZE	QSpan Reserved	
07-00	PWEN	PAS	QSpan Reserved			

PBTI1_CTL Description

Name	Type	Reset By	Reset State	Function
EN	R/W	G_RST	0	Image Enable 0 = Disable, 1 = Enable
BS[3:0]	See Below	PCI_RST	See Below	Block Size (64 Kbyte*2)
PREN	R/W	G_RST	0	Prefetch Read Enable 0 = Disable, 1 = Enable
BRSTWREN	R/W	G_RST	0	Burst Write Enable 0 = Disable, 1 = Enable
INVEND	R/W	G_RST	0	Invert Endian-ness from QB_BOC Setting in MISC_CTL 0 = Use QB_BOC setting, 1 = Invert QB_BOC setting
TC[3:0]	R/W	G_RST	0	QBus Transaction Code User Defined
DSIZE[1:0]	R/W	G_RST	0	QBus Destination Port Size 00=32-bit, 01=8-bit,10=16-bit, 11=reserved
PWEN	R/W	G_RST	0	Posted Write Enable 0 = Disable, 1 = Enable
PAS	See Below	PCI_RST	See Below	PCI Bus Address Space 0 = PCI Bus Memory Space, 1 = PCI Bus I/O Space

Only PCI bus Memory Space transactions can be posted.

The BS[3:0] and PAS fields can be loaded from an external serial EEPROM (See “PCI Bus Target Image 1 (PCI_BST1 and PBTI1_CTL registers)” on page 2-72). There are two cases:

1. If the BS[3:0] and PAS fields are loaded from the EEPROM, then the EEPROM determines their reset state, and they become read only. In this case, the PAS bit has the same value as the bit of the same name in the PCI_BST1 register (page A-12).
2. If the fields are not loaded from the EEPROM, their reset state is zero, and they are writable (note that in this case the PCI_BST1 register is disabled).

Table A.19 PCI Bus Target Image 1 Address Register

Register Name: PBTI1_ADD	Register Offset:114

Bits	Function
31-24	BA
23-16	BA
15-08	TA
07-00	TA

PBTI1_ADD Description

Name	Type	Reset By	Reset State	Function
BA[31:16]	See Below	PCI_RST	See Below	Base Address See Note
TA[31:16]	R/W	G_RST	0	Translation Address

The Base Address specifies the contiguous PCI bus address line values compared by the QSpan during PCI bus address phases. The number of address lines compared for this image is based on the Block Size (programmed in the PBTI1_CTL register, page A-22). The Translation Address specifies the values of the address lines substituted when generating the address for the transaction on the QBus. If no translation is to occur, the Translation Address must be programmed with the same value as that of the Base Address. See “Transaction Decoding” on page 2-27 and “Address Translation” on page 2-30 for more details.

Table A.20 PCI Address Lines Compared as a Function of Block Size

BS	Block Size	Address Lines Compared/Translated
0000	64K	AD31-AD16
0001	128K	AD31-AD17
0010	256K	AD31-AD18
0011	512K	AD31-AD19
0100	1M	AD31-AD20
0101	2M	AD31-AD21
0110	4M	AD31-AD22
0111	8M	AD31-AD23
1000	16M	AD31-AD24
1001	32M	AD31-AD25
1010	64M	AD31-AD26
1011	128M	AD31-AD27
1100	256M	AD31-AD28
1101	512M	AD31-AD29
1110	1G	AD31-AD30
1111	2G	AD31

The read/write type of this register depends on whether the PCI_BST1 register (page A-12) is “enabled” (i.e., whether bit-7 of byte-8 of the EEPROM is “1”). Thus there are two cases:

1. If the EEPROM bit was indeed “1”, the BA field of PBTI1_ADD is read only and has the same value as the PCI_BST1 register from reset onwards.
2. If the EEPROM bit was “0” (or the SDA or ENID pin is “0” at PCI reset) then the entire BA field of PBTI1_ADD is readable and writable. A read from this field after reset returns a 16-bit vector of zeros. In this case, the BA field of this register is independent of the PCI_BST1 register (the PCI_BST1 register does not exist).

The presence of EEPROM does not affect TA[31:16].

Table A.21 PCI Bus Expansion ROM Control Register

Register Name: PBROM_CTL	Register Offset:13C

Bits	Function			
31-24	QSpan Reserved			DSIZE
23-16	0	BS	TC	
15-08	TA			
07-00	TA			

Table A.22 PCI Bus Error Log Control and Status Register

Register Name: PB_ERRCS	Register Offset:140

Bits	Function			
31-24	EN	QSpan Reserved		ES
23-16	QSpan Reserved			
15-08	QSpan Reserved			
07-00	CMDERR		BE_ERR	

PB_ERRCS Description

Name	Type	Reset By	Reset State	Function
EN	R/W	G_RST	0	Enable PCI Error Log 0 = disable error logging, 1 = enable error logging
ES	R/Write 1 to Clear	G_RST	0	Error Status 0 = no error currently logged, 1 = error currently logged
CMDERR[3:0]	R	N/A	0111	PCI Command Error Log
BE_ERR[3:0]	R	G_RST	0	PCI Byte Enable Error Log 0 = byte enable active, 1 = byte enable inactive

The PCI Master Module sets the ES bit if

- a posted write transaction results in a Target-Abort,
- a posted write transaction results in a Master-Abort, or
- the QSpan generates a Master-Abort on reads (error logging does not occur for reads, however).

The assertion of the ES bit can be mapped to the QSpan's interrupt pins by programming the Interrupt Control and Interrupt Direction Control registers. The mapping of interrupts can only occur if the EN bit in the PCI Bus Error Log register is set.

To disable the PCI Error Logging after it has been enabled, the ES bit must not be set. If ES is set, it can be cleared (by writing a logic 1) at the same time as a logic 0 is written to the EN bit.

The BE_ERR field only contains valid information when the ES bit is set. At all other times these fields will return all zeros when read.

Table A.23 PCI Bus Address Error Log Register

Register Name: PB_AERR	Register Offset:144

Bits	Function
31-24	PAERR
23-16	PAERR
15-08	PAERR
07-00	PAERR

PB_AERR Description

Name	Type	Reset By	Reset State	Function
PAERR[31:0]	R	G_RST	0x0000 0000	PCI Address Error Log

The QSpan as PCI master will log errors if a posted write transaction results in a target-abort, or a posted write transaction results in a master abort.

This register logs the PCI bus address information. Its contents are qualified by bit ES of the PCI Bus Error Log Control and Status Register (Table A.22, on page A-27). The PAERR field only contains valid information when ES is set. At all other times, a read of this register will return all zeros.

Table A.24 PCI Bus Data Error Log Register

Register Name: PB_DERR	Register Offset:148

Bits	Function
31-24	PDERR
23-16	PDERR
15-08	PDERR
07-00	PDERR

PB_DERR Description

Name	Type	Reset By	Reset State	Function
PDERR[31:0]	R	G_RST	0	PCI Data Error Log

The QSpan as PCI master will log errors if a posted write transaction results in a target-abort, or a posted write transaction results in a master abort.

This register logs the PCI bus data information. Its contents are qualified by bit ES of the PCI Bus Error Log Control and Status register (Table A.22, on page A-27). The PDERR field only contains valid information when the ES is set. At all other times, a read of this register will return all zeros.

Table A.25 IDMA Control and Status Register

Register Name: IDMA_CS	Register Offset:400

Bits	Function							
31-24	GO	IRST_REQ	QSpan Reserved					
23-16	ACT	IRST	DONE	IPE	IQE	CMD	QSpan Reserved	
15-08	IWM [3:0]				TC[3:0]			
07-00	TC_EN	QSpan Reserved		DIR	IMODE	QTERM	STERM	PORT16

IDMA_CS Description

Name	Type	Reset By	Reset State	Function
GO	W/Read 0 Always	G_RST	0	IDMA Go Bit 0 = no effect, 1 = enable IDMA transfers
IRST_REQ	W/Read 0 Always	G_RST	0	IDMA Reset Request 0 = no effect, 1 = request reset of IDMA control
ACT	R	G_RST	0	IDMA Active Status 0 = no IDMA transfer is active, 1 = IDMA transfer in progress
IRST	R/Write 1 to Clear	G_RST	0	IDMA Reset Status 0 = not reset, 1 = reset
DONE	R/Write 1 to Clear	G_RST	0	IDMA Done Status 0 = not done, 1 = done
IPE	R/Write 1 to Clear	G_RST	0	IDMA PCI Bus Error Status 0 = no error occurred 1 = error occurred on the PCI bus during IDMA transfer
IQE	R/Write 1 to Clear	G_RST	0	IDMA QBus Error Status 0 = no error occurred 1 = error occurred on the QBus during IDMA transfer
CMD	R/W	G_RST	0	PCI Command Option for IDMA PCI Transaction 0 = PCI Memory Write or Memory Read 1 = PCI Memory Write Invalidate or Memory Read Line
IWM [3:0]	R/W	G_RST	0000	Programmable I-FIFO Watermark 0000 = use the value programmed into the CLINE[1:0] field of the PCI_MISC0 register x = when 16x bytes have been queued in the I-FIFO, the IDMA channel will request the PCI bus. Watermark can be set to a maximum of 240 bytes.
TC [3:0]	R/W	G_RST	000	Programmable TC Encoding with PowerQuicc IDMA. Program to the value expected on the QSpan's TC[3:0] lines during an IDMA transfer. The intent of this field is to allow users to use the TC lines to encode an IDMA transaction for the QSpan (add a qualifier for SDACK assertion). Please refer to "The IDMA Channel" on page 2-49.

IDMA_CS Description CONT'D

Name	Type	Reset By	Reset State	Function
TC_EN	R/W	G_RST	0	TC Encoding Enable 0 = QSpan does not decode TC[3:0] for IDMA accesses 1 = QSpan decodes TC[3:0] for IDMA accesses[a]
DIR	R/W	G_RST	0	Direction 0 = transfer from PCI Bus to QBus, 1 = transfer from QBus to PCI bus
IMODE	R/W	G_RST	0	QSpan IDMA Mode 0 = IDMA transfer is QUICC, 1 = IDMA transfer is PowerQUICC
QTERM	R/W	G_RST	0	QSpan Termination Mode 0 = QSpan uses normal termination during QUICC dual address IDMA transfers, 1 = QSpan uses fast termination during QUICC dual address IDMA transfers
STERM	R/W	G_RST	0	Slave Termination Mode 0=external slave uses normal termination mode during QUICC single address IDMA transfers, 1=external slave uses fast termination during QUICC single address IDMA transfers
PORT16	R/W	G_RST	0	QBus PORT 16 0 = source/destination on QBus is a 32-bit port, 1 = source/destination on QBus is a 16-bit port

a. It is recommended that all ones are used in TC[3:0] to be compatible with PowerQUICC operation.

The programmable I-FIFO Watermark (IWM[3:0]) must be programmed with a value less than or equal to the value programmed in the IDMA Transfer Count Register.

The QTERM bit is only of significance if the QSpan is operating as a QUICC IDMA peripheral device during dual address IDMA transfers. During all other conditions this bit does not affect the QSpan's operation.

The STERM bit is only of significance if the QSpan is operating as a QUICC IDMA peripheral device during single address IDMA transfers. During all other conditions this bit does not affect the QSpan's operation.

The IDMA Channel can be reset from either bus while it is in progress by writing “1” to the IRST_REQ bit (see“IDMA Errors, Resets and Interrupts” on page 2-54.) If the ACT bit is “0”, then setting IRST_REQ to “1” has no effect.

The DONE bit is set by the QSpan if its transfer count expires in the IDMA Transfer Count register, or the $\overline{\text{DONE}}$ signal is asserted by the QUICC. Under either condition the QSpan's IDMA controller will return to the idle state, which is indicated by the ACT bit.

The IRST, DONE, IPE and IQE events can be mapped to the interrupt pins on either bus using the QSpan's Interrupt Control Registers (Table A.33, on page A-41).

See “The IDMA Channel” on page 2-49 for more details.

Table A.26 IDMA Address Register

Register Name: IDMA_ADD	Register Offset:404

Bits	Function		
31-24	ADDR		
23-16	ADDR		
15-08	ADDR		
07-00	ADDR	0	0

IDMA_ADD Description

Name	Type	Reset By	Reset State	Function
ADDR[31:2]	R/W	G_RST	0	PCI Bus Address for IDMA transfers

The ADDR field should be programmed with the absolute PCI address for an IDMA transaction. This number is always aligned to a 4-byte boundary. An IDMA transfer wraps-around at the A24 boundary. If an IDMA transfer is required to cross an A24 boundary, it must be programmed as two separate transactions. This field is incremented by the QSpan during a transfer—progress of the IDMA transfer on the PCI bus can be monitored by reading the IDMA_CNT register (page A-34).

If the IDMA_ADDR register is being used as a general purpose message passing register, then the IDMA DONE bit in the IDMA_CSR must be cleared (i.e. the IDMA should be returned to an inactive state) before performing a write to the IDMA_ADDR register. If the IDMA DONE bit is active then the write to the IDMA_ADDR register will not complete successfully.

Table A.27 IDMA Transfer Count Register

Register Name: IDMA_CNT	Register Offset:408

Bits	Function		
31-24	QSpan Reserved		
23-16	CNT		
15-08	CNT		
07-00	CNT	0	0

IDMA_CNT Description[a]

Name	Type	Reset By	Reset State	Function
CNT[23:2]	R/W	G_RST	0	IDMA Transfer Count Number of Bytes to Transfer (Max = 2^{22} * 4 bytes)

a. The IDMA_CNT does not decrement and cannot be programmed to be less than the specified value in the watermark field.

CNT[23:2] indicates the number of bytes to transfer in a IDMA transaction (page A-34). The QSpan decreases this transfer counter by 4 with every 32-bit transfer on the PCI bus. (The IDMA Channel on the PCI Interface transfers 32-bit data). The amount of data that can be transferred within an IDMA transaction is 16 MBytes (i.e., 2^{22} 32-bit transfers). The CNT[23:2] field must be programmed with a minimum value of 0x000010 (corresponding to 16 bytes) otherwise the IDMA channel will not start when the GO bit is set. The CNT field should initially be programmed with the same value as the processor's IDMA's Byte Count register.

Table A.28 Configuration Address Register

Register Name: CON_ADD	Register Offset:500

Bits	Function							
31-24	0	0	0	0	0	0	0	0
23-16	BUS_NUM							
15-08	0	DEV_NUM				FUNC_NUM		
07-00	REG_NUM						0	TYPE

CON_ADD Description

Name	Type	Reset By	Reset State	Function
BUS_NUM[7:0]	R/W	G_RST	0	Bus Number
DEV_NUM[3:0]	R/W	G_RST	0	Device Number
FUNC_NUM[2:0]	R/W	G_RST	0	Function Number
REG_NUM[5:0]	R/W	G_RST	0	Register Number
TYPE	R/W	G_RST	0	Configuration Cycle Type 0 = Type 0, 1 = Type 1

An access to the PCI Configuration Data Register (page A-37) from the QBus Interface performs a corresponding Configuration cycle on the PCI bus. The type of configuration cycle generated by the QSpan is a function of the TYPE bit.

With the TYPE bit set to 1, an access of the PCI Configuration Data register from the QBus interface performs a corresponding Configuration Type 1 cycle on the PCI bus. During the address phase of the Configuration Type 1 cycle on the PCI bus, the PCI address lines carry the values encoded in the Configuration Address Register (AD[31:0] = CON_ADDR[31:0]).

With the TYPE bit set to 0, an access of the PCI Configuration Data register from the QBus interface performs a corresponding Configuration Type 0 cycle on the PCI bus. Programming the Device Number causes one of the upper address lines, AD[31:16] to be asserted during the address phase of the Configuration Type 0 cycle. (Table A.29 below shows which PCI address line is asserted as a function of the DEV_NUM[3:0] field.) The remaining address lines during the address phase of the Configuration cycle are controlled by the Function Number and Register Number:

- AD[15:11] = 00000
- AD[10:8] = FUNC_NUM[2:0]
- AD[7:2] = REG_NUM[5:0]
- AD[1:0] = 00

Table A.29 PCI AD[31:16] lines asserted as a function of DEV_NUM field

DEV_NUM[3:0]	AD[31:16]
0000	0000 0000 0000 0001
0001	0000 0000 0000 0010
0010	0000 0000 0000 0100
0011	0000 0000 0000 1000
0100	0000 0000 0001 0000
0101	0000 0000 0010 0000
0110	0000 0000 0100 0000
0111	0000 0000 1000 0000
1000	0000 0001 0000 0000
1001	0000 0010 0000 0000
1010	0000 0100 0000 0000
1011	0000 1000 0000 0000
1100	0001 0000 0000 0000
1101	0010 0000 0000 0000
1110	0100 0000 0000 0000
1111	1000 0000 0000 0000

The QSpan does not have the ability to perform a Type 0 Configuration cycles to a PCI target that has its IDSEL input connected to one of the AD[15.11] signals—bit 15 of the CON_ADD register is hardcoded as 0. Therefore, for host bridging applications the hardware designer should choose to drive each PCI target's IDSEL input from one of the AD[31..16] signals.

Table A.30 Configuration Data Register

Register Name: CON_DATA	Register Offset:504

Bits	Function
31-24	CDATA
23-16	CDATA
15-08	CDATA
07-00	CDATA

CON_DATA Description

Name	Type	Reset By	Reset State	Function
CDATA[31:0]	R/W	G_RST	0	Configuration Data

A write to the PCI Configuration Data register from the PCI bus has no effect. A read from the PCI bus always returns all zeros. A write to the Configuration Data register from the QBus causes a Configuration Write Cycle to be generated on the PCI as defined by the Configuration Address register (Table A.28, on page A-35). A read of this register from the QBus causes a Configuration Read Cycle to be generated on the PCI bus. The PCI bus Configuration cycles generated by accessing the Configuration Data register are handled as delayed transfers.

The QSpan does not perform byte swapping of data in the Register Channel regardless of whether the QBus is configured as big or little endian. Bit-31 in the register is bit-31 on the QBus and PCI bus regardless of the QB_BOC bit in the MISC_CTL register (page A-46). Therefore, software on the QBus may need to swap the data when performing configuration cycles.

The QSpan will generate a bus error upon a register access to the CON_DATA register if the bus master (BM) bit in the PCI_CS register is not set.

Table A.31 IACK Cycle Generator Register

Register Name: IACK_GEN	Register Offset:508

Bits	Function
31-24	IACK_VEC
23-16	IACK_VEC
15-08	IACK_VEC
07-00	IACK_VEC

IACK_GEN Description

Name	Type	Reset By	Reset State	Function
IACK_VEC[31:0]	R	GEN_RST	0	PCI IACK Cycle Vector

This register is used to generate an Interrupt Acknowledge cycle originating on the QBus. Reading this register from the QBus causes an IACK cycle to be generated on the PCI bus. The byte lanes enabled on the PCI bus are determined by SIZ[1:0] and A[1:0] of the QBus read. The address on the QBus used to access the IACK_GEN register is passed directly over to the PCI bus during the PCI IACK cycle. However, address information is ignored during PCI IACK cycles, so this has no effect.

Reads from this register behave as delayed transfers. This means that the QBus master is retried until the read data is latched from the PCI target. When the IACK cycle completes on the PCI bus, the IACK_VEC[31:0] field is returned as read data when the QBus master returns after the retry.

Writing to this register from the QBus or PCI bus has no effect. Reads from the PCI bus return all zeros.

The QSpan does not perform byte swapping of data in the Register Channel regardless of whether the QBus is configured as big or little endian. Bit-31 in the register is bit-31 on the QBus regardless of the QB_BOC bit in the MISC_CTL register (page A-46). Therefore, software on the QBus may need to swap the data when performing IACK cycles.

Table A.32 Interrupt Status Register

Register Name: INT_STAT	Register Offset:600

Bits	Function							
31-24	PEL_IS	QEL_IS	DPD_IS	Reserved	IQE_IS	IPE_IS	IRST_IS	DONE_IS
23-16	INT_IS	PERR_IS	SERR_IS	QINT_IS	QSpan_Reserved			
15-08	QSpan Reserved							
07-00	QSpan Reserved				SI3_IS	SI2_IS	SI1_IS	SI0_IS

INT_STAT Description

Name	Type	Reset By	Reset State	Function
PEL_IS	R/Write 1 to Clear	G_RST	0	PCI Bus Error Log Interrupt Status
QEL_IS	R/Write 1 to Clear	G_RST	0	QBus Error Log Interrupt Status
DPD_IS	R/Write 1 to Clear	G_RST	0	Data Parity Detected Interrupt Status
IQE_IS	R/Write 1 to Clear	G_RST	0	IDMA QBus Error Interrupt Status
IPE_IS	R/Write 1 to Clear	G_RST	0	IDMA PCI Error Interrupt Status
IRST_IS	R/Write 1 to Clear	G_RST	0	IDMA Reset Interrupt Status
DONE_IS	R/Write 1 to Clear	G_RST	0	IDMA Done Interrupt Status
INT_IS	R/Write 1 to Clear	G_RST	0	Status of PCI interrupt input to QBus interrupt output
PERR_IS	R/Write 1 to Clear	G_RST	0	Status of PCI bus PERR# input to QBus interrupt output
SERR_IS	R/Write 1 to Clear	G_RST	0	Status of PCI bus SERR# input to QBus interrupt output
QINT_IS	R/Write 1 to Clear	G_RST	0	Status of QBus interrupt input to PCI interrupt output
SI3_IS	R/Write 1 to Clear	G_RST	0	Software Interrupt 3 Status
SI2_IS	R/Write 1 to Clear	G_RST	0	Software Interrupt 2 Status
SI1_IS	R/Write 1 to Clear	G_RST	0	Software Interrupt 1 Status
SI0_IS	R/Write 1 to Clear	G_RST	0	Software Interrupt 0 Status

Interrupt status bits are set upon the assertion of the interrupt condition. Each interrupt status bit in the Interrupt Status register will remain set until a logic 1 is written to it. Clearing of the interrupt status bit will not clear the source status bit that may have caused the interrupt to be asserted. As part of the interrupt handling routine a separate register transaction to the corresponding status register will have to occur.

The DPD_IS bit is set (if it is enabled) when DP_D bit in the PCI Configuration Control and Status Register is set.

Table A.33 Interrupt Control Register

Register Name: INT_CTL	Register Offset:604

Bits	Function							
31-24	PEL_EN	QEL_EN	DPD_EN	Reserved	IQE_EN	IPE_EN	IRST_EN	DONE_EN
23-16	INT_EN	PERR_EN	SERR_EN	QINT_EN	QSpan Reserved			
15-08	QSpan Reserved							
07-00	QSpan Reserved						SI1	SI0

INT_CTL Description

Name	Type	Reset By	Reset State	Function
PEL_EN	R/W	G_RST	0	PCI Bus Error Log Interrupt Enable 0 = disable mapping of PCI error log interrupt, 1 = enable mapping of interrupt
QEL_EN	R/W	G_RST	0	QBus Error Log Interrupt Enable 0 = disable mapping of QBus error log interrupt, 1 = enable mapping of interrupt
DPD_EN	R/W	G_RST	0	Data Parity Detected Interrupt Enable 0=disable mapping of Data Parity Detected interrupt, 1=enable mapping of interrupt
IQE_EN	R/W	G_RST	0	IDMA QBus Error Interrupt Enable 0=disable mapping of IDMA QBus error interrupt, 1=enable mapping of interrupt
IPE_EN	R/W	G_RST	0	IDMA PCI Error Interrupt Enable 0 = disable mapping of IDMA PCI error interrupt, 1 = enable mapping of interrupt
IRST_EN	R/W	G_RST	0	IDMA Reset Interrupt Enable 0 = disable mapping of IDMA reset interrupt, 1 = enable mapping of IDMA reset interrupt
DONE_EN	R/W	G_RST	0	IDMA Done Interrupt Enable 0 = disable mapping of IDMA done interrupt, 1 = enable mapping of interrupt
INT_EN	R/W	G_RST	0	Map PCI bus Interrupt Input to QBus Interrupt Output Enable 0 = disable mapping 1 = enable mapping of interrupt
PERR_EN	R/W	G_RST	0	Map Parity Error on PCI bus to QBus Interrupt Output Enable 0 = disable mapping 1 = enable mapping of interrupt
SERR_EN	R/W	G_RST	0	Map SERR# Input to QBus Interrupt Output Enable 0 = disable mapping 1 = enable mapping of interrupt

INT_CTL Description

QINT_EN	R/W	G_RST	0	Map QBus Interrupt Input to PCI Bus Interrupt Output Enable 0 = disable mapping 1 = enable mapping of interrupt
SI1	W/Read 0 Always	G_RST	0	Software Interrupt 1 0 = no effect, 1 = sets SI1_IS status bit
SI0	W/Read 0 Always	G_RST	0	Software Interrupt 0 0 = no effect, 1 = sets SI0_IS status bit

Table A.34 Interrupt Direction Register

Register Name: INT_DIR	Register Offset:608

Bits	Function							
31-24	PEL_DIR	QEL_DIR	DPD_DIR	Reserved	IQE_DIR	IPE_DIR	IRST_DIR	DONE_DIR
23-16	QSpan Reserved							
15-08	QSpan Reserved							
07-00	QSpan Reserved				SI3_DIR	SI2_DIR	SI1_DIR	SI0_DIR

INT_DIR Description

Name	Type	Reset By	Reset State	Function
PEL_DIR	R/W	G_RST	0	PCI Error Log Interrupt Direction 0 = map PCI error log interrupt to the QBus, 1 = map interrupt to the PCI bus
QEL_DIR	R/W	G_RST	0	QBus Error Log Interrupt Direction 0 = map QBus error log interrupt to the QBus, 1 = map interrupt to the PCI bus
DPD_DIR	R/W	G_RST	0	Data Parity Detected Interrupt Direction 0=map IDMA PCI error interrupt to the QBus, 1=map interrupt to the PCI bus
IQE_DIR	R/W	G_RST	0	IDMA QBus Error Interrupt Direction 0=map IDMA QBus error interrupt to the QBus, 1=map interrupt to the PCI bus
IPE_DIR	R/W	G_RST	0	IDMA PCI Error Interrupt Direction 0 = map IDMA PCI error interrupt to the QBus, 1 = map interrupt to the PCI bus
IRST_DIR	R/W	G_RST	0	IDMA Reset Interrupt Direction 0 = map IDMA reset interrupt to the QBus, 1 = map interrupt to the PCI bus
DONE_DIR	R/W	G_RST	0	IDMA Done Interrupt Direction 0 = map IDMA done interrupt to the QBus, 1 = map IDMA done interrupt to the PCI bus
SI3_DIR	R/W	G_RST	0	Software Interrupt 3 Direction 0 = map software interrupt to the QBus 1 = map interrupt to the PCI bus
SI2_DIR	R/W	G_RST	0	Software Interrupt 2 Direction 0 = map software interrupt to the QBus 1 = map interrupt to the PCI bus
SI1_DIR	R/W	G_RST	0	Software Interrupt 1 Direction 0 = map software interrupt to the QBus, 1 = map interrupt to the PCI bus

INT_DIR Description

SI0_DIR	R/W	G_RST	0	Software Interrupt 0 Direction 0 = map software interrupt to the QBus, 1 = map interrupt to the PCI bus

In order for an interrupt to be mapped to either bus, it must also have its corresponding interrupt enable set in the Interrupt Control Register (Table A.33, on page A-41).

Table A.35 Interrupt Control Register 2

Register Name: INT_CTL2	Register Offset:60C

Bits	Function			
31-24	QSpan Reserved			
23-16	QSpan Reserved			
15-08	QSpan Reserved			
07-00	QSpan Reserved	SI3	SI2	QSpan Reserved

INT_CTL2 Description

Name	Type	Reset By	Reset State	Function
SI3	W/Read 0 Always	G_RST	0	Software Interrupt 3 0 = no effect, 1 = sets SI3_IS status bit
SI2	W/Read 0 Always	G_RST	0	Software Interrupt 2 0 = no effect, 1 = sets SI2_IS status bit

Table A.36 Miscellaneous Control and Status Register

Register Name: MISC_CTL	Register Offset:800

Bits	Function						
31-24	SW_RST	QSpan Reserved					
23-16				S_BG	S_BB	0	QB_BOC
15-08	QSpan Reserved		MA_BE_D	QSpan Reserved			
07-00	PRCNT [5:0]					MSTSLV	

MISC_CTL Description

Name	Type	Reset By	Reset State	Function
SW_RST	R/W	G_RST	0	QBus Software Reset Control 0 = software reset not active, 1 = software reset active
S_BG	R/W	G_RST	See below	Synchronous Bus Grant 0 = $\overline{BG}$ input is asynchronous to QCLK 1 = $\overline{BG}$ input is synchronous to QCLK
S_BB	R/W	G_RST	See below	Synchronous Bus Grant Acknowledge 0 = $\overline{BB}$ input is asynchronous to QCLK 1 = $\overline{BB}$ input is synchronous to QCLK
QB_BOC	R/W	G_RST	0	QBus Byte Ordering Control 0 = QBus uses Big-Endian byte ordering, 1 = QBus uses PCI Little-Endian byte ordering
MA_BE_D	R/W	G_RST	0	Master Abort - Bus Error mapping disable Applies to delayed transfers. 0 = When QSpan issues a PCI master abort it maps this termination to the QBus as a bus error (behaves in same manner as previous QSpan designs) 1 = When QSpan receives a PCI master abort or target abort it maps this as a normal termination on the QBus. On writes, data is flushed from the channel, and on reads the QSpan returns all “1s”. In order to comply with the PCI 2.1 Specification, the MA_BE_D bit should be set to “1” before any PCI configuration cycles are performed, if the QSpan is being used as a host bridge.
PRCNT [5:0]	R/W	G_RST	000000	Prefetch Read Byte Count This field controls how much data the QSpan prefetches on the QBus. The number of bytes prefetched is 4 times the number programmed into this register (e.g. 001100 is 48 bytes)
MSTSLV[1:0]	R	G_RST	See Table A.37	Master/Slave Mode

The SW_RST bit allows the QBus reset output ($\overline{\text{RESETO}}$) to be controlled in software from the PCI bus. When “1” is written to this bit, the QSpan asserts $\overline{\text{RESETO}}$. There are three ways to cause the QSpan to terminate the software reset state:

1. Clear the SW_RST bit by writing “0” to it. In this case, $\overline{\text{RESETO}}$ is immediately negated.
2. Assert $\overline{\text{RESETI}}$. In this case, the SW_RST bit is immediately cleared (set to “0”) and $\overline{\text{RESETO}}$ is immediately negated.
3. Assert RST#. In this case, SW_RST is immediately cleared (set to “0”), however $\overline{\text{RESETO}}$ continues to be asserted until RST# is negated.

Unexpected results may occur if the output $\overline{\text{RESETO}}$ on the QBus Interface of the QSpan is used to generate the input $\overline{\text{RESETI}}$.

The reset state of S_BG and S_BB depends on the master mode of the QSpan (see Table A.37, on page A-48). If the Master Mode is QUICC, then the reset state of S_BG and S_BB is 0. When the QUICC’s arbiter is utilized to provide the arbitration for the QSpan (CA91C860B, CA91L860B), the QSpan must be programmed to operate in synchronous arbitration mode (i.e. S_BG and S_BB must be set to “1”). If the Master Mode is PowerQUICC or M68040, then the reset state of S_BG and S_BB is 1.

The QB_BOC bit affects the presentation of data on the QBus Interface in the cases where the data passes through the QBus Slave Channel, the PCI Target Channel, or the IDMA Channel. There are two caveats. First, the PBTIx_CTL registers each contain an INVEND bit which causes the QSpan to use the logical inversion of the QB_BOC. Second, the QB_BOC bit does not affect the presentation of data on the QBus Interface in the case where the Register Channel is accessed. That is, the QSpan does not perform byte swapping of data in the Register Channel regardless of whether the QBus is configured as big or little endian (this applies to all QBus register accesses, including the CON_DATA and IACK registers). Bit-31 in any QSpan register is presented on bit-31 of the QBus (and bit-31 of the PCI bus) regardless of the QB_BOC bit.

The MSTSLV field indicates the Master and Slave modes of the QSpan. The first bit of this field is determined by the value of $\overline{\text{BDIP}}$ at reset; the second bit is determined by the value of SIZ[1] at reset. This field is described in Table A.37.

Table A.37 Master/Slave Mode (MSTSLV field)

MSTSLV field	Master Mode	Slave Mode
00	QUICC	QUICC and M68040
01	QUICC	QUICC and PowerQUICC
10	M68040	QUICC and M68040
11	PowerQUICC	QUICC and PowerQUICC

Table A.38 EEPROM Control and Status Register

Register Name: EEPROM_CS	Register Offset:804

Bits	Function		
31-24	ADDR[7:0]		
23-16	DATA[7:0]		
15-08	QSpan Reserved		
07-00	ACT	READ	QSpan Reserved

EEPROM_CS Description

Name	Type	Reset By	Reset State	Function
ADDR[7:0]	See Below	PCI_RST	See Below	EEPROM read and write address
DATA[7:0]	See Below	PCI_RST	See Below	EEPROM read and write data
ACT	R	PCI_RST	0	EEPROM Active 0 = no, 1 = yes
READ	R/W	PCI_RST	0	EEPROM Read bit 0 = write to EEPROM, 1 = read to EEPROM

This register is provided for users to read to or write from the EEPROM through the QBus or the PCI bus. This register only accepts reads or writes when the ACT bit is “0”, therefore the ACT bit might need to be polled before a write is attempted. The ACT bit is “1” when the QSpan is loading data from the EEPROM or is in the process of completing a read or write to the EEPROM caused by an access to this register.

After an external master reads from or writes to the EEPROM_CS register, it takes the QSpan 1 SCLK to set the ACT bit, where 1 SCLK = PCLK/1024. Therefore, after accessing the EEPROM_CS register, the master should wait at least this amount of time before verifying the state of the ACT bit. If the PCI bus is operating at 33 MHz, this means waiting 31 us. After this period of time, the ACT bit will be valid. If the user fails to allow for the QSpan’s latency in setting the ACT bit, the user’s attempt to read or write to the QSpan will be unsuccessful, but there will be no hardware acknowledgment of this fact, which may be detected by a subsequent read (when ACT is valid and negated).

Writes complete on the QBus or the PCI bus regardless of the state of ACT (and hence regardless of whether the write to the DATA field was effective).

If there is no EEPROM (the SDA and ENID pins are low at PCI reset), then this register is read-only and reads return all zeros.

See “Programming the EEPROM from the QBus or PCI Bus” on page 2-73 for more details.

Table A.39 QBus Slave Image 0 Control Register

Register Name: QBSI0_CTL	Register Offset:F00

Bits	Function		
31-24	PWEN	QSpan Reserved	PAS
23-16	QSpan Reserved		
15-08	QSpan Reserved		
07-00	QSpan Reserved		

QBSI0_CTL Description

Name	Type	Reset By	Reset State	Function
PWEN	R/W	PCI_RST	See Below	Posted Write Enable 0 = Disable, 1 = Enable
PAS	R/W	PCI_RST	See Below	PCI Bus Address Space 0 = PCI Bus Memory Space, 1 = PCI Bus I/O Space

This slave image definition is used when QSpan is selected by the assertion of $\overline{\text{CSPCI}}$ and IMSEL (Image Select) is 0. Table A.40 and Table A.41 below indicate how the QSpan Slave module responds to QBus masters as a function of the PWEN and PAS bits setting. It indicates, among other things, that the QSpan does not burst to PCI I/O space and responds to such attempts by generating a bus error on the QBus. See “Transaction decoding and QBus Slave Images” on page 2-8.

Values in this register can be programmed from a serial EEPROM (see “The EEPROM Channel” on page 2-69). If they are not, their reset state is 0.

Table A.40 QSpan Response to Single Cycle Access

PWEN	PAS	Write Cycle (R/$\overline{\text{W}}$ asserted)	Read Cycle (R/$\overline{\text{W}}$ negated)
0	0	Delayed Write	Delayed Read
0	1	Delayed Write	Delayed Read
1	0	Posted Write	Delayed Read
1	1	Delayed Write	Delayed Read

Table A.41 QSpan Response to Burst Cycle Access

PWEN	PAS	Write Cycle (R/$\overline{W}$ asserted)	Read Cycle (R/$\overline{W}$ negated)
0	0	Posted Write	Delayed Burst Read
0	1	Bus Error	Bus Error
1	0	Posted Write	Delayed Burst Read
1	1	Bus Error	Bus Error

Table A.42 QBus Slave Image 0 Address Translation Register

Register Name: QBSI0_AT	Register Offset:F04

Bits	Function		
31-24	TA		
23-16	TA		
15-08	QSpan Reserved		
07-00	BS	QSpan Reserved	EN

QBSI0_AT Description

Name	Type	Reset By	Reset State	Function
TA[31:16]	R/W	PCI_RST	See Below	Translation Address
BS[3:0]	R/W	PCI_RST	See Below	Block Size (64 Kbyte*2)
EN	R/W	PCI_RST	See Below	Enable Address Translation 0 = disable, 1 = enable

The Translation Address specifies the values of the address lines substituted when generating the address for the transaction on the PCI bus. Address translation is enabled by setting the EN bit. Block Size is used to determine which address lines are translated (see Table A.43).

Values in this register can be programmed from a serial EEPROM (see “QBus Slave Image 0 (QBSI0_CTL and QBSI0_AT registers)” on page 2-72). Otherwise, their reset state is 0.

Table A.43 Address Lines Translated as a Function of Block Size

BS	Block Size	Address Lines Translated
0000	64K	A31-A16
0001	128K	A31-A17
0010	256K	A31-A18
0011	512K	A31-A19
0100	1M	A31-A20
0101	2M	A31-A21
0110	4M	A31-A22
0111	8M	A31-A23
1000	16M	A31-A24
1001	32M	A31-A25
1010	64M	A31-A26
1011	128M	A31-A27
1100	256M	A31-A28
1101	512M	A31-A29
1110	1G	A31-A30
1111	2G	A31

Table A.44 QBus Slave Image 1 Control Register

Register Name: QBSI1_CTL	Register Offset:F10

Bits	Function			
31-24	PWEN	QSpan Reserved		PAS
23-16	QSpan Reserved			
15-08	QSpan Reserved			
07-00	QSpan Reserved			

QBSI1_CTL Description

Name	Type	Reset By	Reset State	Function
PWEN	R/W	G_RST	0	Posted Write Enable 0 = Disable, 1 = Enable
PAS	R/W	G_RST	0	PCI Bus Address Space 0 = PCI Bus Memory Space, 1 = PCI Bus I/O Space

This slave image definition is used when QSpan is selected by the assertion of $\overline{\text{CSPCI}}$ and IMSEL (Image Select) is 1. Table A.45 and Table A.46 below indicate how the QSpan Slave module responds to QBus masters as a function of the PWEN and PAS bits setting. It indicates, among other things, that the QSpan does not burst to PCI I/O space and responds to such attempts by generating a bus error on the QBus. Single posted writes to I/O space are treated as delayed writes. See “Transaction decoding and QBus Slave Images” on page 2-8.

Unlike Slave Image 0, this register cannot be programmed with a serial EEPROM.

Table A.45 QSpan Response to Single Cycle Access ($\overline{\text{BURST}}$ negated)

PWEN	PAS	Write Cycle (R/$\overline{\text{W}}$ asserted)	Read Cycle (R/$\overline{\text{W}}$ negated)
0	0	Delayed Write	Delayed Read
0	1	Delayed Write	Delayed Read
1	0	Posted Write	Delayed Read
1	1	Delayed Write	Delayed Read

Table A.46 QSpan Response to Burst Cycle Access ($\overline{\text{BURST}}$ asserted)

PWEN	PAS	Write Cycle (R/$\overline{\text{W}}$ asserted)	Read Cycle (R/$\overline{\text{W}}$ negated)
0	0	Posted Write	Delayed Burst Read
0	1	Bus Error	Bus Error
1	0	Posted Write	Delayed Burst Read
1	1	Bus Error	Bus Error

Table A.47 QBus Slave Image 1 Address Translation Register

Register Name: QBSI1_AT	Register Offset:F14

Bits	Function		
31-24	TA		
23-16	TA		
15-08	QSpan Reserved		
07-00	BS	QSpan Reserved	EN

QBSI1_AT Description

Name	Type	Reset By	Reset State	Function
TA[31:16]	R/W	G_RST	0	Translation Address See Note
BS[3:0]	R/W	G_RST	0	Block Size (64 Kbyte*2)
EN	R/W	G_RST	0	Enable Address Translation 0 = Disable, 1 = Enable

The Translation Address specifies the values of the address lines substituted when generating the address for the transaction on the PCI bus. Address translation is enabled by setting the EN bit. Block Size is used to determine which address lines are translated (see Table A.48).

Unlike Slave Image 0, this register cannot be programmed with a serial EEPROM.

Table A.48 QBus Address Lines Compared as a function of Block Size

BS	Block Size	Address Lines Translated
0000	64K	A31-A16
0001	128K	A31-A17
0010	256K	A31-A18
0011	512K	A31-A19
0100	1M	A31-A20
0101	2M	A31-A21
0110	4M	A31-A22
0111	8M	A31-A23
1000	16M	A31-A24
1001	32M	A31-A25
1010	64M	A31-A26
1011	128M	A31-A27
1100	256M	A31-A28
1101	512M	A31-A29
1110	1G	A31-A30
1111	2G	A31

Table A.49 QBus Error Log Control and Status Register

Register Name: QB_ERRCS	Register Offset:F80

Bits	Function			
31-24	EN	QSpan Reserved		ES
23-16	QSpan Reserved			
15-08	QSpan Reserved			
07-00	TC_ERR		QSpan Reserved	SIZ_ERR

QB_ERRCS Description

Name	Type	Reset By	Reset State	Function
EN	R/W	G_RST	0	Enable QBus Error Log 0 = disable error logging, 1 = enable error logging
ES	R/Write 1 to Clear	G_RST	0	QBus Error Status 0 = no error currently logged, 1 = error currently logged
TC_ERR[3:0]	R	G_RST	0	QBus Transaction Code Error Log
SIZ_ERR[1:0]	R	G_RST	0	QBus SIZ Field Error Log

The QBus Master Module logs errors only when a posted write transaction from the Px-FIFO results in a bus error. The assertion of the ES bit can be mapped to the QSpan's interrupt pins by programming the Interrupt Control Register and the Interrupt Direction Register (Table A.33, on page A-41 and Table A.34, on page A-43). The mapping of interrupts can only occur if the EN bit in the QBus Error Log Control and Status Register is set.

To disable the QBus Error Logging after it has been enabled, the ES bit must not be set. If ES is set, it can be cleared (by writing a logic 1) at the same time as a logic 0 is written to the EN bit.

The TC_ERR and SIZ_ERR fields only contain valid information when the ES bit is set. At all other times these fields will return all zeros when read.

Table A.50 QBus Address Error Log

Register Name: QB_AERR	Register Offset:F84

Bits	Function
31-24	QAERR
23-16	QAERR
15-08	QAERR
07-00	QAERR

QB_AERR Description

Name	Type	Reset By	Reset State	Function
QAERR[31:0]	R	G_RST	0	QBus Address Error Log

The QBus Master Module will log errors only when a posted write transaction results in a bus error.

This register logs the QBus address information. Its contents are qualified by bit ES of the QBus Error Log Control and Status Register (Table A.49, on page A-56.) The QAERR field only contains valid information when ES is set. At all other times, a read of this register will return all zeros.

Table A.51 QBus Data Error Log

Register Name: QB_DERR	Register Offset:F88

Bits	Function
31-24	QDERR
23-16	QDERR
15-08	QDERR
07-00	QDERR

QB_DERR Description

Name	Type	Reset By	Reset State	Function
QDERR[31:0]	R	G_RST	0	QBus Data Error Log

The QBus Master Module will log errors only when a posted write transaction results in bus error.

This register logs the QBus data information. Its contents are qualified by bit ES of the QBus Error Log Control and Status register (Table A.49).

Appendix B Timing

This appendix provides timing information for the QBus interface. (PCI interface timing is not detailed since the QSpan is PCI 2.1 compliant.) Timing parameters for all processors are listed first, followed by the timing diagrams.

The timing tables are:

The sets of diagrams are:

B.1 Timing Parameters

Test conditions for timing parameters in Tables B.1 to B.5 are:

Test Conditions for 5V

Commercial (C): 0ºC to 70ºC, 5V±10%

Industrial (I): -40ºC to 85ºC, 5V±10%

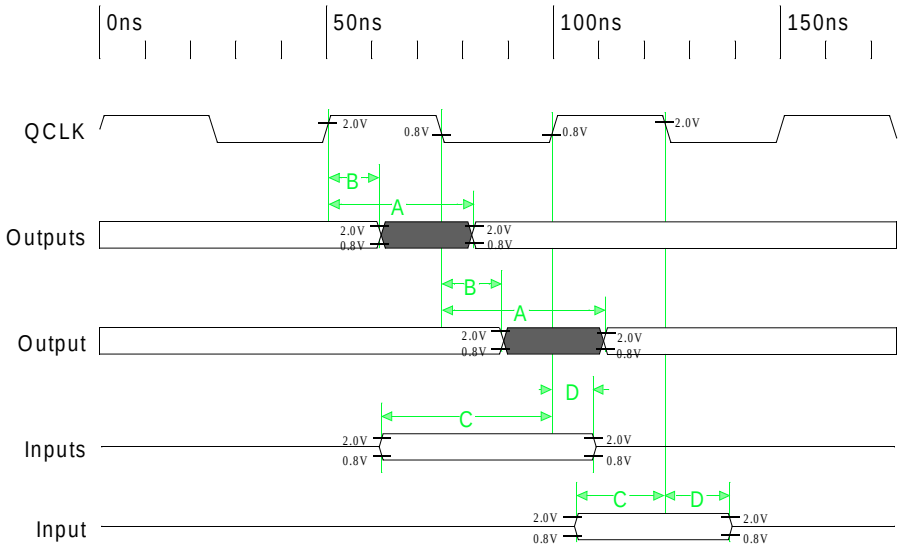
Test Conditions for 3.3V

Commercial (C): 0ºC to 70ºC, 3.3V±5%

Industrial (I): -40ºC to 85ºC, 3.3V±5%

Please refer to Figure B.1 as a reference in reading the timing diagrams in this Appendix.

Figure B.1 : Reference Voltages for AC Timing Specification



A - Max Output Delay
B - Min Output Hold
C - Min Input Setup
D - Min Input Hold

In order to condense the diagrams and tables, certain multifunctional QBus signals are presented in their bus specific forms (e.g., $\overline{DSACK1}/\overline{TA}$ is referred to as $\overline{TA}$ in the PowerQUICC and M68040 sections).

Table B.1 Timing Parameters for QUICC Interface

Timing Parameter	Description	3.3 Volt				5 Volt				Units	Note
		50CE		40CE/40IE		50CQ		40CQ/40IQ			
		Min	Max	Min	Max	Min	Max	Min	Max		
t_{200}	QCLK Frequency	-	33	-	25	-	33	-	25	MHz	
t_{201}	Period	30	-	40	-	30	-	40	-	ns	
t_{202}	Clock Pulse Width (Low)	14	-	19	-	14	-	19	-	ns	
t_{204}	Clock Pulse Width (High)	14	-	19	-	14	-	19	-	ns	
t_{205}	Clock Rise Time (tr)	-	2	-	2	-	2	-	2	ns	
t_{206}	Clock Fall Time (t_f)	-	2	-	2	-	2	-	2	ns	
t_{210a}	A asserted from QCLK (positive edge)		9.6		10.7		11.1		12.3	ns	2
t_{210b}	$\overline{BERR}$ asserted from QCLK (positive edge)		9.2		10.2		9.6		10.6	ns	2
t_{210c}	$\overline{DSACK0}$, $\overline{DSACK1}$ asserted from QCLK (positive edge)		9.4		10.4		9.3		10.4	ns	2

Table B.1 Timing Parameters for QUICC Interface[CONT'D]

Timing Parameter	Description	3.3 Volt				5 Volt				Units	Note
		50CE		40CE/40IE		50CQ		40CQ/40IQ			
		Min	Max	Min	Max	Min	Max	Min	Max		
t_{210d}	$\overline{HALT}$ asserted from QCLK (positive edge)		8.9		9.9		7.9		8.7	ns	2
t_{210e}	R/$\overline{W}$ asserted from QCLK (positive edge)		8		8.9		7.8		8.6	ns	2
t_{210f}	SIZ asserted from QCLK (positive edge)		7.9		8.8		8.3		9.2	ns	2
t_{210g}	TC asserted from QCLK (positive edge)		8.4		9.3		9.1		10.1	ns	2
t_{211a}	$\overline{BGACK}$ asserted from QCLK (positive edge)		9.2		10.2		8.4		9.3	ns	1
t_{211b}	$\overline{BR}$ asserted from QCLK (positive edge)		7.4		8.2		6.6		7.4	ns	2
t_{212a}	$\overline{DSACK0}$, $\overline{DSACK1}$ tristated from QCLK (negative edge)		9.4		10.4		9.1		10.1	ns	
t_{212b}	$\overline{BERR}$ tristated from QCLK (negative edge)		9.6		10.6		9		10	ns	
t_{212c}	$\overline{HALT}$ tristated from QCLK (negative edge)		9.1		10.1		7.7		8.5	ns	
t_{213}	D asserted (slave reads) from QCLK (negative edge)		14.8		16.4		15.3		16.9	ns	1
t_{214}	D tristated (slave reads) from QCLK (positive edge)		13.7		15.2		14.9		16.6	ns	
t_{215a}	$\overline{AS}$ asserted from QCLK (negative edge)		8		8.9		7.7		8.6	ns	1
t_{215b}	$\overline{DS}$ asserted from QCLK (negative edge)		7.8		8.7		6.5		7.2	ns	
t_{216a}	$\overline{AS}$ tristated from QCLK (positive edge)		7.9		8.8		8.1		8.9	ns	
t_{216b}	$\overline{DS}$ tristated from QCLK (positive edge)		8.1		9		7		7.8	ns	
t_{217a}	$\overline{BERR}$ setup to QCLK (negative edge)	0		0		0		0		ns	
t_{217b}	$\overline{BG}$ setup to QCLK (negative edge)	-0.9		-1.0		-0.9		-1.0		ns	
t_{217c}	$\overline{BGACK}$ setup to QCLK (negative edge)	0		0		0		0		ns	

Table B.1 Timing Parameters for QUICC Interface CONT'D

Timing Parameter	Description	3.3 Volt				5 Volt				Units	Note
		50CE		40CE/40IE		50CQ		40CQ/40IQ			
		Min	Max	Min	Max	Min	Max	Min	Max		
t_{217d}	$\overline{DSACK0}$, $\overline{DSACK1}$ setup to QCLK (negative edge)	0		0		0		0		ns	
t_{217e}	$\overline{HALT}$ setup to QCLK (negative edge)	0		0		0		0		ns	
t_{218a}	$\overline{BERR}$ hold from QCLK (negative edge)	0		0		0		0		ns	
t_{218b}	$\overline{BG}$ hold from QCLK (negative edge)	0		0		0		0		ns	
t_{218c}	$\overline{BGACK}$ hold from QCLK (negative edge)	1.4		1.5		1.8		0		ns	
t_{218d}	$\overline{DSACK0}$, $\overline{DSACK1}$ hold from QCLK (negative edge)	0		0		0		0		ns	
t_{218e}	$\overline{HALT}$ hold from QCLK (negative edge)	0		0		0		0		ns	
t_{219a}	A setup to QCLK (negative edge)	7		7.8		8.5		9.5		ns	
t_{219b}	$\overline{AS}$ setup to QCLK (negative edge)	3.4		3.8		3		3.4		ns	
t_{219c}	$\overline{CSPCI}$ setup to QCLK (negative edge)	2.8		3.1		4.8		5.3		ns	
t_{219d}	$\overline{CSREG}$ setup to QCLK (negative edge)	7		7.8		8.6		9.5		ns	
t_{219e}	IMSEL setup to QCLK (negative edge)	3		3.3		3.7		4.1		ns	
t_{219f}	R/$\overline{W}$ setup to QCLK (negative edge)	3.2		3.6		4.6		5.1		ns	
t_{219g}	SIZ setup to QCLK (negative edge)	6.6		7.3		8		8.9		ns	
t_{219h}	TC setup to QCLK (negative edge)	4.5		5		4.8		5.3		ns	
t_{220a}	A hold from QCLK (negative edge)	0.1		0.1		0		0		ns	
t_{220b}	$\overline{AS}$ hold from QCLK (negative edge)	0.5		0.5		0.2		0.2		ns	
t_{220c}	$\overline{CSPCI}$ hold from QCLK (negative edge)	0.9		1		0		0		ns	
t_{220d}	$\overline{CSREG}$ hold from QCLK (negative edge)	1.5		1.7		0.9		1		ns	
t_{220e}	IMSEL hold from QCLK (negative edge)	0		0		0		0		ns	

Table B.1 Timing Parameters for QUICC Interface[CONT'D]

Timing Parameter	Description	3.3 Volt				5 Volt				Units	Note
		50CE		40CE/40IE		50CQ		40CQ/40IQ			
		Min	Max	Min	Max	Min	Max	Min	Max		
t_{220f}	R/$\overline{W}$ hold from QCLK (negative edge)	0.9		1		0		0		ns	
t_{220g}	SIZ hold from QCLK (negative edge)	0		0		0		0		ns	
t_{220h}	TC hold from QCLK (negative edge)	0		0		0		0		ns	
t_{221}	D setup (master reads) to QCLK (negative edge)	0		0		0.4		0.4		ns	
t_{222}	D hold (master reads) from QCLK (negative edge)	2.1		2.3		1.5		1.6		ns	
t_{223}	D setup (slave writes) to QCLK (positive edge)	0		0		0		0		ns	
t_{224}	D hold (slave writes) from QCLK (positive edge)	1.8		2		1.7		1.9		ns	
t_{225}	D asserted (master writes) from QCLK (positive edge)		9.8		10.9		11.6		12.8	ns	1
t_{226}	D tristated (master writes) from QCLK (positive edge)	3.9	9.8	4.3	10.9	4.9	11.6	5.4	12.8	ns	
t_{234}	$\overline{BGACK}$ tristated from QCLK (negative edge)		9.2		10.2		8.4		9.3	ns	
t_{235a}	$\overline{BGACK}$ negated from QCLK (positive edge)	3.2	7.4	3.6	8.2	3.8	6.6	4.2	7.4	ns	2
t_{235b}	$\overline{BR}$ negated from QCLK (positive edge)	3.2	7.3	3.5	8.1	3.8	7.3	4.2	8.1	ns	1
t_{236a}	A tristated or negated from QCLK (positive edge)	3.8	9.6	4.3	10.7	4.5	11.1	5.1	12.3	ns	1
t_{236b}	$\overline{BERR}$ tristated or negated from QCLK (positive edge)	3.7	9.2	4.1	10.2	4.6	9.6	5.1	10.6	ns	1
t_{236c}	$\overline{DSACK0}$, $\overline{DSACK1}$ tristated or negated from QCLK (positive edge)	3.3	9.4	3.7	10.4	4.2	9.3	4.6	10.4	ns	1
t_{236d}	$\overline{HALT}$ tristated or negated from QCLK (positive edge)	3.5	8.9	3.9	9.9	4.3	7.9	4.8	8.7	ns	1
t_{236e}	R/$\overline{W}$ tristated or negated from QCLK (positive edge)	3.2	8	3.6	8.9	4	7.8	4.4	8.6	ns	1
t_{236f}	SIZ tristated or negated from QCLK (positive edge)	3.1	7.9	3.5	8.8	4.3	8.3	4.8	9.2	ns	1

Table B.1 Timing Parameters for QUICC Interface CONT'D

Timing Parameter	Description	3.3 Volt				5 Volt				Units	Note
		50CE		40CE/40IE		50CQ		40CQ/40IQ			
		Min	Max	Min	Max	Min	Max	Min	Max		
t_{236g}	TC tristated or negated from QCLK (positive edge)	3.3	8.4	3.7	9.3	4.5	9.1	5	10.1	ns	1
t_{250}	$\overline{DREQ}$ asserted (or negated) from QCLK (negative edge)	4	9.3	4.4	10.4	4.4	8	4.9	8.9	ns	2
t_{251}	D asserted (valid) from $\overline{DACK}$ (negative edge)		12.5		13.9		14.6		16.2	ns	1
t_{252}	D tristated from $\overline{DACK}$ (positive edge)	3.9	12.5	4.4	13.9	5	14.6	5.6	16.2	ns	
t_{253a}	$\overline{AS}$ setup to QCLK (negative edge)	3.4		3.8		3		3.4		ns	
t_{253b}	$\overline{CSPCI}$ setup to QCLK (negative edge)	2.8		3.1		4.8		5.3			
t_{254}	$\overline{DACK}$ setup to QCLK (negative edge)	2.5		2.8		3.2		3.5		ns	
t_{255}	$\overline{DSACK0}$, $\overline{DSACK1}$ setup to QCLK (negative edge)	0.6		0.6		1.1		1.2		ns	
t_{256}	$\overline{BERR}$ setup to QCLK (negative edge)	3		3.3		3.5		3.8		ns	
t_{257}	$\overline{HALT}$ setup to QCLK (negative edge)	2		2.2		3.7		4.1		ns	
t_{258}	D setup to QCLK (negative edge)	0		0		0.3		0.4		ns	
t_{259a}	$\overline{AS}$ hold from QCLK (negative edge)	0.5		0.5		0.2		0.2		ns	
t_{259b}	$\overline{CSPCI}$ hold from QCLK (negative edge)	0.9		1		0		0		ns	
t_{259c}	D hold from QCLK (negative edge)	2.1		2.4		1.1		1.2		ns	
t_{259d}	$\overline{DACK}$ hold from QCLK (negative edge)	0.6		0.7		0.3		0.3		ns	
t_{259e}	$\overline{DSACK0}$, $\overline{DSACK1}$ hold from QCLK (negative edge)	0.7		0.8		1.1		1.2		ns	
t_{259f}	$\overline{BERR}$ hold from QCLK (negative edge)	0.5		0.6		0.9		1		ns	
t_{259g}	$\overline{HALT}$ hold from QCLK (negative edge)	1.2		1.3		0.7		0.8		ns	
t_{260}	$\overline{DONE}$ setup to QCLK (negative edge)	0.1		0.1		0		0		ns	
t_{261}	$\overline{DONE}$ hold from QCLK (negative edge)	1.8		2		1.8		2		ns	

Table B.1 Timing Parameters for QUICC Interface[CONT'D]

Timing Parameter	Description	3.3 Volt				5 Volt				Units	Note
		50CE		40CE/40IE		50CQ		40CQ/40IQ			
		Min	Max	Min	Max	Min	Max	Min	Max		
t_{262a}	$\overline{AS}$ negated from QCLK (negative edge)	3.4	8	3.8	8.9	4	7.6	4.5	8.4	ns	1
t_{262b}	$\overline{DS}$ negated from QCLK (negative edge)	3.4	7.8	3.8	8.7	3.6	6.3	4.1	7.0	ns	1

1. Minimum output hold time specified for load of 10 pF. Maximum output delay specified for load of 50 pF.
2. Minimum output hold time specified for load of 10 pF. Maximum output delay specified for load of 35 pF.

During IDMA fast termination cycles the maximum QUICC QCLK frequency is 20 MHz. This applies to every variant of QSpan.

Table B.2 Timing Parameters for PowerQUICC Interface (3.3 volt, PBGA)

Timing Parameter	Description	50CE		40IE		40CE		Units	Note
		min	max	min	max	min	max		
t_{300}	QCLK Frequency		50		40		40	MHz	
t_{301}	Period	20		25		25		ns	
t_{302}	Clock Pulse Width (Low)	8		10		10		ns	
t_{304}	Clock Pulse Width (High)	8		10		10		ns	
t_{305}	Clock Rise Time (t_r)		2		2		2	ns	
t_{306}	Clock Fall Time (t_f)		2		2		2	ns	
t_{310a}	A asserted from QCLK (positive edge)		8.9		9.6		10.7	ns	1
t_{310b}	$\overline{BB}$ asserted from QCLK (positive edge)		8.2		8.8		9.8	ns	1
t_{310c}	$\overline{BURST}$ asserted from QCLK (positive edge)		7.5		8.6		9.6	ns	1
t_{310d}	R/$\overline{W}$ asserted from QCLK (positive edge)		6.9		8		8.9	ns	1
t_{310e}	SIZ asserted from QCLK (positive edge)		6.9		7.9		8.8	ns	1
t_{310f}	$\overline{TA}$ asserted from QCLK (positive edge)		8.2		9.5		10.5	ns	1
t_{310g}	TC asserted from QCLK (positive edge)		7.3		8.4		9.3	ns	1
t_{310h}	$\overline{TEA}$ asserted from QCLK (positive edge)		8.0		8.6		9.6	ns	1
t_{310i}	$\overline{TRETRY}$ asserted from QCLK (positive edge)		8		9.2		10.2	ns	1
t_{310j}	$\overline{TS}$ asserted from QCLK (positive edge)		7.3		8.5		9.4	ns	1
t_{310k}	$\overline{BDIP}$ asserted from QCLK (positive edge)		7.6		8.8		9.8	ns	1
t_{311}	$\overline{BR}$ asserted (or negated) from QCLK (positive edge)	2.8	6.4	3.2	7.4	3.6	8.2	ns	2
t_{312a}	$\overline{BB}$ tristated from QCLK (negative edge)		8		9.2		10.2	ns	
t_{312b}	$\overline{TS}$ tristated from QCLK (negative edge)		7.5		8.7		9.6	ns	
t_{313a}	$\overline{BB}$ setup to QCLK (positive edge)	4.5		5.2		5.8		ns	
t_{313b}	$\overline{BDIP}$ setup to QCLK (positive edge)	2.4		2.7		3		ns	
t_{313c}	$\overline{BG}$ setup to QCLK (positive edge)	4.7		5.4		6		ns	
t_{313d}	$\overline{BURST}$ setup to QCLK (positive edge)	4.4		5.1		5.6		ns	
t_{313e}	D setup to QCLK (positive edge)	6		6.9		7.7		ns	

Table B.2 Timing Parameters for PowerQUICC Interface (3.3 volt, PBGA)

Timing Parameter	Description	50CE		40IE		40CE		Units	Note
		min	max	min	max	min	max		
t_{313f}	IMSEL setup to QCLK (positive edge)	3.9		4.5		5		ns	
t_{313g}	$\overline{TA}$ setup to QCLK (positive edge)	5.5		6.3		7		ns	
t_{313h}	$\overline{TEA}$ setup to QCLK (positive edge)	6		6.9		7.6		ns	
t_{313i}	$\overline{TRETRY}$ setup to QCLK (positive edge)	5.3		6.1		6.8		ns	
t_{314}	D valid (slave reads) from QCLK (positive edge)		9.5		11		12.2	ns	1
t_{315a}	$\overline{A}$ hold from QCLK (positive edge)	0.7		0.8		0.8		ns	
t_{315b}	$\overline{BB}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{315c}	$\overline{BDIP}$ hold from QCLK (positive edge)	0.6		0.7		0.7		ns	
t_{315d}	$\overline{BG}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{315e}	$\overline{BURST}$ hold from QCLK (positive edge)	0.5		0.5		0.6		ns	
t_{315f}	$\overline{CSPCI}$ hold from QCLK (positive edge)	0.4		0.4		0.5		ns	
t_{315g}	$\overline{CSREG}$ hold from QCLK (positive edge)	0.5		0.6		0.6		ns	
t_{315h}	D hold from QCLK (positive edge)	1.6		1.8		2		ns	
t_{315i}	IMSEL hold from QCLK (positive edge)	0		0		0		ns	
t_{315j}	R/$\overline{W}$ hold from QCLK (positive edge)	0.6		0.6		0.7		ns	
t_{315k}	SIZ hold from QCLK (positive edge)	0.9		1		1.1		ns	
t_{315l}	$\overline{TA}$ hold from QCLK (positive edge)	0.2		0.2		0.2		ns	
t_{315m}	TC hold from QCLK (positive edge)	1.8		2		2.2		ns	
t_{315n}	$\overline{TEA}$ hold from QCLK (positive edge)	0.3		0.4		0.4		ns	
t_{315o}	$\overline{TRETRY}$ hold from QCLK (positive edge)	0.3		0.3		0.3		ns	
t_{315p}	$\overline{TS}$ hold from QCLK (positive edge)	0.6		0.7		0.8		ns	
t_{316}	D tristated from QCLK (positive edge)	3.4	12.0	3.9	13.7	4.3	15.2	ns	
t_{317}	D enabled from QCLK (positive edge)		12.0		13.7		15.2	ns	
t_{334}	D valid (master writes) from QCLK (positive edge)		8.5		9.8		10.9	ns	1
t_{335a}	$\overline{TA}$ tristated from QCLK (negative edge)		8.2		9.5		10.5	ns	

Table B.2 Timing Parameters for PowerQUICC Interface (3.3 volt, PBGA)

Timing Parameter	Description	50CE		40IE		40CE		Units	Note
		min	max	min	max	min	max		
t_{335b}	$\overline{TEA}$ tristated from QCLK (negative edge)		8.1		9.4		10.4	ns	
t_{335c}	$\overline{TRETRY}$ tristated from QCLK (negative edge)		7.8		9		10	ns	
t_{336a}	A negated from QCLK (positive edge)		8.4		9.6		10.7	ns	1
t_{336b}	$\overline{BB}$ negated from QCLK (positive edge)	2.7	6.4	3.2	7.3	3.5	8.1	ns	1
t_{336c}	$\overline{BURST}$ negated from QCLK (positive edge)		7.8		9		10	ns	1
t_{336d}	R/$\overline{W}$ negated from QCLK (positive edge)		6.9		8		8.9	ns	1
t_{336e}	SIZ negated from QCLK (positive edge)		6.9		7.9		8.8	ns	1
t_{336f}	$\overline{TA}$ negated from QCLK (positive edge)	2.8	7.3	3.3	8.4	3.7	9.3	ns	1
t_{336g}	TC negated from QCLK (positive edge)		8.2		9.5		10.5	ns	1
t_{336h}	$\overline{TEA}$ negated from QCLK (positive edge)	3.2	9.3	3.7	10.7	4.1	11.9	ns	1
t_{336i}	$\overline{TRETRY}$ negated from QCLK (positive edge)	3	8.1	3.4	9.3	3.8	10.3	ns	1
t_{336j}	$\overline{TS}$ negated from QCLK (positive edge)		6.4		7.4		8.2	ns	1
t_{336k}	$\overline{BDIP}$ negated from QCLK (positive edge)		7.8		9		10	ns	1
t_{337a}	A setup to QCLK (positive edge)	5.3		6.1		6.8		ns	
t_{337b}	$\overline{CSPCI}$ setup to QCLK (positive edge)	3.4		3.9		4.4		ns	
t_{337c}	$\overline{CSREG}$ setup to QCLK (positive edge)	5.3		6.1		6.8		ns	
t_{337d}	R/$\overline{W}$ setup to QCLK (positive edge)	3.6		4.2		4.6		ns	
t_{337e}	SIZ setup to QCLK (positive edge)	4.9		5.7		6.3		ns	
t_{337f}	TC setup to QCLK (positive edge)	5		5.8		7.1		ns	
t_{337g}	$\overline{TS}$ setup to QCLK (positive edge)	3.7		4.2		4.7		ns	
t_{350}	$\overline{DREQ}$ asserted (or negated) from QCLK (positive edge)	3.7	9.2	4.3	10.6	4.8	11.8	ns	2
t_{351}	D asserted (valid) from $\overline{SDACK}$ (negative edge)		10.8		12.5		13.9	ns	1
t_{352}	D tristated from $\overline{TA}$ (positive edge)	4.1	12.4	4.8	14.3	5.3	15.9	ns	
t_{353a}	$\overline{CSPCI}$ setup to QCLK (positive edge)	3.5		4		4.4		ns	

Table B.2 Timing Parameters for PowerQUICC Interface (3.3 volt, PBGA)

Timing Parameter	Description	50CE		40IE		40CE		Units	Note
		min	max	min	max	min	max		
t_{353b}	$\overline{TS}$ setup to QCLK (positive edge)	4.1		4.7		5.2		ns	
t_{354}	$\overline{SDACK}$ setup to QCLK (positive edge)	0		0		0		ns	
t_{355a}	$\overline{TA}$ setup to QCLK (positive edge)	3.1		3.5		3.9		ns	
t_{355b}	$\overline{TEA}$ setup to QCLK (positive edge)	3.5		4		4.5		ns	
t_{356}	$\overline{TRETRY}$ setup to QCLK (positive edge)	3.1		3.6		4		ns	
t_{357a}	$\overline{TA}$ asserted to $\overline{SDACK}$ (positive edge)	2		2		2		ns	
t_{357b}	$\overline{TEA}$ asserted to $\overline{SDACK}$ (positive edge)	2		2		2		ns	
t_{357c}	$\overline{TRETRY}$ asserted to $\overline{SDACK}$ (positive edge)	2		2		2		ns	
t_{359a}	$\overline{CSPCI}$ hold from QCLK (positive edge)	0.8		0.9		1		ns	
t_{359b}	D hold from QCLK (positive edge)	1.6		1.8		2		ns	
t_{359c}	$\overline{SDACK}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{359d}	$\overline{TA}$ hold from QCLK (positive edge)	0.7		0.8		0.8		ns	
t_{359e}	$\overline{TEA}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{359f}	$\overline{TRETRY}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{359g}	$\overline{TS}$ hold from QCLK (positive edge)	0.2		0.2		0.2		ns	
t_{360}	TC setup to QCLK (positive edge)	5.0		5.8		7.1			
t_{361}	TC hold from QCLK (positive edge)	0.4		0.4		0.4			

Table B.3 Timing Parameters for PowerQUICC Interface (5 volt, PQFP)

Timing Parameter	Description	50CQ		40IQ		40CQ		Units	Note
		min	max	min	max	min	max		
t_{300}	QCLK Frequency		50		40		40	MHz	
t_{301}	Period	20		25		25		ns	
t_{302}	Clock Pulse Width (Low)	8		10		10		ns	
t_{304}	Clock Pulse Width (High)	8		10		10		ns	
t_{305}	Clock Rise Time (t_r)		2		2		2	ns	
t_{306}	Clock Fall Time (t_f)		2		2		2	ns	
t_{310a}	A asserted from QCLK (positive edge)		9.6		11.1		12.3	ns	1
t_{310b}	$\overline{BB}$ asserted from QCLK (positive edge)		7.4		8.5		9.5	ns	1
t_{310c}	$\overline{BURST}$ asserted from QCLK (positive edge)		7.4		8.5		9.5	ns	1
t_{310d}	R/$\overline{W}$ asserted from QCLK (positive edge)		6.8		7.8		8.6	ns	1
t_{310e}	SIZ asserted from QCLK (positive edge)		7.2		8.3		9.2	ns	1
t_{310f}	$\overline{TA}$ asserted from QCLK (positive edge)		8.2		9.5		10.5	ns	1
t_{310g}	TC asserted from QCLK (positive edge)		7.9		9.1		10.1	ns	1
t_{310h}	$\overline{TEA}$ asserted from QCLK (positive edge)		8		9.2		10.2	ns	1
t_{310i}	$\overline{TRETRY}$ asserted from QCLK (positive edge)		7.5		8.7		9.6	ns	1
t_{310j}	$\overline{TS}$ asserted from QCLK (positive edge)		7.5		8.6		9.6	ns	1
t_{310k}	$\overline{BDIP}$ asserted from QCLK (positive edge)		7.3		8.5		9.4	ns	1
t_{311}	$\overline{BR}$ asserted (or negated) from QCLK (positive edge)	3.3	5.8	3.8	6.6	4.2	7.4	ns	2
t_{312a}	$\overline{BB}$ tristated from QCLK (negative edge)		7		8.1		9	ns	
t_{312b}	$\overline{TS}$ tristated from QCLK (negative edge)		7.4		8.5		9.4	ns	
t_{313a}	$\overline{BB}$ setup to QCLK (positive edge)	3.7		4.2		4.7		ns	
t_{313b}	$\overline{BDIP}$ setup to QCLK (positive edge)	1.5		1.7		1.9		ns	
t_{313c}	$\overline{BG}$ setup to QCLK (positive edge)	3.3		3.8		4.3		ns	
t_{313d}	$\overline{BURST}$ setup to QCLK (positive edge)	5		5.8		6.4		ns	
t_{313e}	D setup to QCLK (positive edge)	6		6.9		8.4		ns	

Table B.3 Timing Parameters for PowerQUICC Interface (5 volt, PQFP)

Timing Parameter	Description	50CQ		40IQ		40CQ		Units	Note
		min	max	min	max	min	max		
t_{313f}	IMSEL setup to QCLK (positive edge)	4.5		5.2		5.8		ns	
t_{313g}	$\overline{TA}$ setup to QCLK (positive edge)	6.4		7.4		8.2		ns	
t_{313h}	$\overline{TEA}$ setup to QCLK (positive edge)	6.9		8		8.9		ns	
t_{313i}	$\overline{TRETRY}$ setup to QCLK (positive edge)	5.3		6.1		6.7		ns	
t_{314}	D valid (slave reads) from QCLK (positive edge)		10.6		12.3		13.6	ns	1
t_{315a}	$\overline{A}$ hold from QCLK (positive edge)	0.6		0.6		0.7		ns	
t_{315b}	$\overline{BB}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{315c}	$\overline{BDIP}$ hold from QCLK (positive edge)	1		1.1		1.2		ns	
t_{315d}	$\overline{BG}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{315e}	$\overline{BURST}$ hold from QCLK (positive edge)	0.5		0.5		0.6		ns	
t_{315f}	$\overline{CSPCI}$ hold from QCLK (positive edge)	0.3		0.4		0.4		ns	
t_{315g}	$\overline{CSREG}$ hold from QCLK (positive edge)	0.9		1		1.1		ns	
t_{315h}	D hold from QCLK (positive edge)	1.5		1.7		1.9		ns	
t_{315i}	IMSEL hold from QCLK (positive edge)	0		0		0		ns	
t_{315j}	R/$\overline{W}$ hold from QCLK (positive edge)	0.4		0.5		0.5		ns	
t_{315k}	SIZ hold from QCLK (positive edge)	0.7		0.8		0.9		ns	
t_{315l}	$\overline{TA}$ hold from QCLK (positive edge)	0.4		0.4		0.4		ns	
t_{315m}	TC hold from QCLK (positive edge)	1.9		2.2		2.5		ns	
t_{315n}	$\overline{TEA}$ hold from QCLK (positive edge)	0.7		0.8		0.8		ns	
t_{315o}	$\overline{TRETRY}$ hold from QCLK (positive edge)	0.5		0.5		0.6		ns	
t_{315p}	$\overline{TS}$ hold from QCLK (positive edge)	1		1.1		1.2		ns	
t_{316}	D tristated from QCLK (positive edge)	4.2	12.6	4.9	14.6	5.4	16.2	ns	
t_{317}	D enabled from QCLK (positive edge)		12		13.8		15.3	ns	
t_{334}	D valid (master writes) from QCLK (positive edge)		10		11.6		12.8	ns	1
t_{335a}	$\overline{TA}$ tristated from QCLK (negative edge)		7.9		9.2		10.2	ns	

Table B.3 Timing Parameters for PowerQUICC Interface (5 volt, PQFP)

Timing Parameter	Description	50CQ		40IQ		40CQ		Units	Note
		min	max	min	max	min	max		
t_{335b}	$\overline{TEA}$ tristated from QCLK (negative edge)		8.1		9.3		10.3	ns	
t_{335c}	$\overline{TRETRY}$ tristated from QCLK (negative edge)		6.7		7.7		8.5	ns	
t_{336a}	A negated from QCLK (positive edge)		9.6		11.1		12.3	ns	1
t_{336b}	$\overline{BB}$ negated from QCLK (positive edge)		6.3		7.3		8.1	ns	1
t_{336c}	$\overline{BURST}$ negated from QCLK (positive edge)		7.3		8.4		9.3	ns	1
t_{336d}	R/$\overline{W}$ negated from QCLK (positive edge)		6.8		7.8		8.6	ns	1
t_{336e}	SIZ negated from QCLK (positive edge)		7.2		8.3		9.2	ns	1
t_{336f}	$\overline{TA}$ negated from QCLK (positive edge)	4.1	8.3	4.7	9.5	5.2	10.6	ns	1
t_{336g}	TC negated from QCLK (positive edge)		7.9		9.1		10.1	ns	1
t_{336h}	$\overline{TEA}$ negated from QCLK (positive edge)	3.9	8.7	4.5	10	5	11.1	ns	1
t_{336i}	$\overline{TRETRY}$ negated from QCLK (positive edge)	3.9	7.5	4.5	8.7	5	9.6	ns	1
t_{336j}	$\overline{TS}$ negated from QCLK (positive edge)		6.3		7.3		8.1	ns	1
t_{336k}	$\overline{BDIP}$ negated from QCLK (positive edge)		7.8		9		10	ns	1
t_{337a}	A setup to QCLK (positive edge)	6		6.9		7.6		ns	
t_{337b}	$\overline{CSPCI}$ setup to QCLK (positive edge)	4.4		5.1		5.6		ns	
t_{337c}	$\overline{CSREG}$ setup to QCLK (positive edge)	6.2		7.2		8		ns	
t_{337d}	R/$\overline{W}$ setup to QCLK (positive edge)	4.6		5.3		5.8		ns	
t_{337e}	SIZ setup to QCLK (positive edge)	5		5.8		7.3		ns	
t_{337f}	TC setup to QCLK (positive edge)	5		5.8		6.4		ns	
t_{337g}	$\overline{TS}$ setup to QCLK (positive edge)	4		4.7		5.2		ns	
t_{350}	$\overline{DREQ}$ asserted (or negated) from QCLK (positive edge)	4.6	9.1	5.3	10.4	5.9	11.6	ns	2
t_{351}	D asserted (valid) from $\overline{SDACK}$ (negative edge)		12.7		14.6		16.2	ns	1
t_{352}	D tristated from $\overline{TA}$ (positive edge)	5.1	14.4	5.9	16.6	6.5	18.5	ns	
t_{353a}	$\overline{CSPCI}$ setup to QCLK (positive edge)	4.8		5.6		6.2		ns	

Table B.3 Timing Parameters for PowerQUICC Interface (5 volt, PQFP)

Timing Parameter	Description	50CQ		40IQ		40CQ		Units	Note
		min	max	min	max	min	max		
t_{353b}	$\overline{TS}$ setup to QCLK (positive edge)	4.6		5.3		5.2		ns	
t_{354}	$\overline{SDACK}$ setup to QCLK (positive edge)	0.8		0.9		1		ns	
t_{355a}	$\overline{TA}$ setup to QCLK (positive edge)	4		4.7		5.2		ns	
t_{355b}	$\overline{TEA}$ setup to QCLK (positive edge)	4.7		5.4		6		ns	
t_{356}	$\overline{TRETRY}$ setup to QCLK (positive edge)	4.1		4.7		5.3		ns	
t_{357a}	$\overline{TA}$ asserted to $\overline{SDACK}$ (positive edge)	2		2		2		ns	
t_{357b}	$\overline{TEA}$ asserted to $\overline{SDACK}$ (positive edge)	2		2		2		ns	
t_{357c}	$\overline{TRETRY}$ asserted to $\overline{SDACK}$ (positive edge)	2		2		2		ns	
t_{359a}	$\overline{CSPCI}$ hold from QCLK (positive edge)	0.2		0.2		0.2		ns	
t_{359b}	D hold from QCLK (positive edge)	1.5		1.7		1.9		ns	
t_{359c}	$\overline{SDACK}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{359d}	$\overline{TA}$ hold from QCLK (positive edge)	0.9		1		1.1		ns	
t_{359e}	$\overline{TEA}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{359f}	$\overline{TRETRY}$ hold from QCLK (positive edge)	0		0		0		ns	
t_{359g}	$\overline{TS}$ hold from QCLK (positive edge)	0.1		0.1		0.1		ns	
t_{360}	TC setup to QCLK (positive edge)	5.0		5.8		6.4			
t_{361}	TC hold from QCLK (positive edge)	0.7		0.7		0.8			

1. Minimum output hold time specified for load of 10 pF. Maximum output delay specified for load of 50 pF.
2. Minimum output hold time specified for load of 10 pF. Maximum output delay specified for load of 35 pF.

Table B.4 Timing Parameters for M68040 Interface

Timing Parameter	Description	3.3 Volt / 40CE		5 Volt / 40CQ		Units	Note
		min	max	min	max		
t_{400}	QCLK Frequency		40		40	MHz	
t_{401}	Period	25		25		ns	
t_{402}	Clock Pulse Width (Low)	12		12		ns	3
t_{404}	Clock Pulse Width (High)	10		10		ns	3
t_{405}	Clock Rise Time (t_r)		3		3	ns	
t_{406}	Clock Fall Time (t_f)		3		3	ns	
t_{409a}	$\overline{TA}$ tristated from QCLK (negative edge)		11.4		10.9	ns	
t_{409b}	$\overline{TEA}$, tristated from QCLK (negative edge)		10.4		10.3	ns	
t_{410a}	A asserted from QCLK (positive edge)		10.7		12.3	ns	1
t_{410b}	$\overline{BB}$ asserted from QCLK (positive edge)		9.8		9.5	ns	1
t_{410c}	R/$\overline{W}$ asserted from QCLK (positive edge)		8.9		8.6	ns	1
t_{410d}	SIZ asserted from QCLK (positive edge)		8.8		9.2	ns	1
t_{410e}	$\overline{TA}$ asserted from QCLK (positive edge)		11.4		10.9	ns	1
t_{410f}	TC asserted from QCLK (positive edge)		9.3		10.1	ns	1
t_{410g}	$\overline{TEA}$ asserted from QCLK (positive edge)		11.9		11.1	ns	1
t_{410h}	$\overline{BURST}$/$\overline{TIP}$ asserted from QCLK (positive edge)		9.6		9.5	ns	1
t_{410i}	$\overline{TS}$ asserted from QCLK (positive edge)		9.4		9.6	ns	1
t_{411}	$\overline{BR}$ asserted from QCLK (positive edge)	3.6	8.2	4.2	7.4	ns	2
t_{412a}	$\overline{BB}$ tristated from QCLK (negative edge)		10.2		9.4	ns	
t_{412b}	$\overline{TS}$ tristated from QCLK (negative edge)		9.6		9.4	ns	
t_{413a}	$\overline{BB}$ setup to QCLK (positive edge)	5.8		4.7		ns	
t_{413b}	$\overline{BG}$ setup to QCLK (positive edge)	6.0		4.3		ns	
t_{413c}	D setup to QCLK (positive edge)	7.7		7.6		ns	
t_{413d}	IMSEL setup to QCLK (positive edge)	5.0		5.8		ns	
t_{413e}	$\overline{TA}$ setup to QCLK (positive edge)	7.0		8.2		ns	
t_{413f}	$\overline{TEA}$ setup to QCLK (positive edge)	7.6		8.9		ns	
t_{413g}	$\overline{BURST}$/$\overline{TIP}$ setup to QCLK (positive edge)	5.6		6.4		ns	
t_{414}	D valid from QCLK (positive edge) (master writes)	4.3	10.9	5.4	12.8	ns	1
t_{415a}	A hold from QCLK (positive edge)	0.8		0.7		ns	
t_{415b}	$\overline{BB}$ hold from QCLK (positive edge)	0		0		ns	
t_{415c}	$\overline{BG}$ hold from QCLK (positive edge)	0		0		ns	
t_{415d}	$\overline{BURST}$/$\overline{TIP}$ hold from QCLK (positive edge)	0.6		0.6		ns	
t_{415e}	$\overline{CSPCI}$ hold from QCLK (positive edge)	0.5		0.4		ns	

Table B.4 Timing Parameters for M68040 Interface CONT'D

Timing Parameter	Description	3.3 Volt / 40CE		5 Volt / 40CQ		Units	Note
		min	max	min	max		
t_{415f}	$\overline{\text{CSREG}}$ hold from QCLK (positive edge)	0.6		1.1		ns	
t_{415g}	D hold from QCLK (positive edge)	2.0		1.9		ns	
t_{415h}	IMSEL hold from QCLK (positive edge)	0.0		0.0		ns	
t_{415i}	R/$\overline{\text{W}}$ hold from QCLK (positive edge)	0.7		0.5		ns	
t_{415j}	SIZ hold from QCLK (positive edge)	1.1		0.9		ns	
t_{415k}	$\overline{\text{TA}}$ hold from QCLK (positive edge)	0.2		0.4		ns	
t_{415l}	TC hold from QCLK (positive edge)	2.2		2.5		ns	
t_{415m}	$\overline{\text{TEA}}$ hold from QCLK (positive edge)	0.4		0.8		ns	
t_{415n}	$\overline{\text{TRETRY}}$ hold from QCLK (positive edge)	0.3		0.6		ns	
t_{415o}	$\overline{\text{TS}}$ hold from QCLK (positive edge)	0.8		1.2		ns	
t_{416}	D tristated from QCLK (positive edge)	4.3	15.2	5.4	16.2	ns	
t_{417}	D valid from QCLK (positive edge) (slave reads)		12.2		13.6	ns	1
t_{418}	D hold from QCLK (positive edge)	3.9		5.2		ns	
t_{419a}	A setup to QCLK (positive edge)	6.8		7.6		ns	
t_{419b}	$\overline{\text{CSPCI}}$ setup to QCLK (positive edge)	4.4		5.6		ns	
t_{419c}	$\overline{\text{CSREG}}$ setup to QCLK (positive edge)	6.8		8.0		ns	
t_{419d}	R/$\overline{\text{W}}$ setup to QCLK (positive edge)	4.6		5.8		ns	
t_{419e}	SIZ setup to QCLK (positive edge)	6.3		6.4		ns	
t_{419f}	TC setup to QCLK (positive edge)	6.4		6.4		ns	
t_{419g}	$\overline{\text{TS}}$ setup to QCLK (positive edge)	4.7		5.2		ns	
t_{436a}	A negated from QCLK (positive edge)		10.7		12.3	ns	1
t_{436b}	$\overline{\text{BB}}$ negated from QCLK (positive edge)		7.8		8.0	ns	1
t_{436c}	R/$\overline{\text{W}}$ negated from QCLK (positive edge)		8.8		8.6	ns	1
t_{436d}	SIZ negated from QCLK (positive edge)		7.8		9.1	ns	1
t_{436e}	$\overline{\text{TA}}$ negated from QCLK (positive edge)	4.3	11.4	5.2	10.9	ns	1
t_{436f}	TC negated from QCLK (positive edge)		9.3		10.1	ns	1
t_{436g}	$\overline{\text{TEA}}$ negated from QCLK (positive edge)	4.1	11.3	4.9	11.1	ns	1
t_{436h}	$\overline{\text{BURST}}$/$\overline{\text{TIP}}$ negated from QCLK (positive edge)		9.9		9.3	ns	1
t_{436i}	$\overline{\text{TS}}$ negated from QCLK (positive edge)		7.9		8.1	ns	1

1. Minimum output hold time specified for load of 10 pF. Maximum output delay specified for load of 50 pF.
2. Minimum output hold time specified for load of 10 pF. Maximum output delay specified for load of 35 pF.
3. Measured at 1.5 volts.

Table B.5 Timing Parameters for Interrupts and Resets

Timing Parameter	Description	3.3V		5.0V		Units	Note
		Min	Max	Min	Max		
t_{001}	RST# asserted to $\overline{\text{RESETO}}$ asserted		10.8		8.5	ns	
t_{002}	RST# negated to $\overline{\text{RESETO}}$ tristated		10.8		8.5	ns	
t_{003}	$\overline{\text{RESETO}}$ asserted from PCLK (rising edge)		10.1		8.3	ns	
t_{004}	$\overline{\text{RESETO}}$ tristated from PCLK (rising edge)		10.1		8.3	ns	
t_{005}	$\overline{\text{QINT}}$ asserted from PCLK (rising edge)		10.7		8.9	ns	
t_{006}	$\overline{\text{QINT}}$ tristated from PCLK (rising edge)		10.7		8.9	ns	

Table B.6 Timing Parameters for Reset Options

Timing Parameter	Description	All Parts		Units	Note
		Min	Max		
t_{007}	SIZ[1] set-up to $\overline{\text{RESETI}}$ or RST# (rising edge)	1		ns	
t_{008}	TMODE[1] set-up to $\overline{\text{RESETI}}$ or RST# (rising edge)	2.5		ns	
t_{009}	TMODE[0] set-up to $\overline{\text{RESETI}}$ or RST# (rising edge)	2.5		ns	
t_{010}	$\overline{\text{BDIP}}$ set-up to $\overline{\text{RESETI}}$ or RST# (rising edge)	2.0		ns	
t_{011}	BM_EN set-up to $\overline{\text{RESETI}}$ or RST# (rising edge)	1.5		ns	
t_{012}	SDA set-up to $\overline{\text{RESETI}}$ or RST# (rising edge)	2.25		ns	
t_{013}	SIZ[1] hold from $\overline{\text{RESETI}}$ or RST# (rising edge)	1.75		ns	
t_{014}	TMODE[1] hold from $\overline{\text{RESETI}}$ or RST# (rising edge)	2.75		ns	
t_{015}	TMODE[0] hold from $\overline{\text{RESETI}}$ or RST# (rising edge)	2.75		ns	
t_{016}	$\overline{\text{BDIP}}$ hold from $\overline{\text{RESETI}}$ or RST# (rising edge)	2.0		ns	
t_{017}	BM_EN hold from $\overline{\text{RESETI}}$ or RST# (rising edge)	1.5		ns	
t_{018}	SDA hold from $\overline{\text{RESETI}}$ or RST# (rising edge)	2.75		ns	
t_{019a}	RST# or $\overline{\text{RESETI}}$ pulse width (asserted) @ 25MHz	40		ns	
t_{019b}	RST# or $\overline{\text{RESETI}}$ pulse width (asserted) @ 33MHz	30		ns	
t_{019c}	RST# or $\overline{\text{RESETI}}$ pulse width (asserted) @ 50MHz	20		ns	

B.2 Wait State Insertion (QBus Slave Module)

The QSpan as QBus slave inserts wait states as follows:

- One wait state for all retried cycles
- One wait state for burst writes
- One wait states for single posted writes
- Two wait states for complete delayed reads and writes
- One wait state for complete delayed burst reads
- See below for register accesses

As explained in “The Register Channel” on page 2-58, since registers of the QSpan can be accessed from either the PCI bus or the QBus, an internal arbitration occurs to indicate ownership. If an external master on the QBus attempts to access the Register Channel while it is owned by the PCI bus, the transfer will be retried on the QBus (with one wait state if the PCI Channel owns the bus) and the QSpan will make an internal request for register ownership by the QBus. Once the current PCI register access (which includes three wait states) has completed, the Register Channel is granted to the QBus. This grant will be removed if no register access is attempted within 2^{10} QCLK clock cycles. The grant will also be removed if a register access completes on the QBus without a subsequent register access beginning within 32 QCLK cycles of the completion of the previous register access.

When the QBus processor is accessing the QSpan’s registers, the maximum number of retries the QSpan will assert is two. The minimum response time for successful (non-retried) QBus accesses from $\overline{TS}$ (or $\overline{AS}$) asserted to $\overline{TA}$ (or $\overline{DSACKx}$) asserted is three QCLKs (including two wait states) for reads, and six for writes.

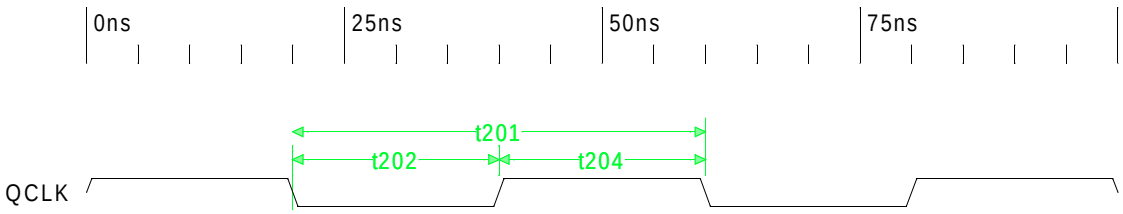
Please refer to Figure B.1 as a reference in reading the timing diagrams in this Appendix.

The following wait-state list does not apply to IDMA transfers.

B.3 Timing Diagrams

B.3.1 QBus (QUICC) Interface

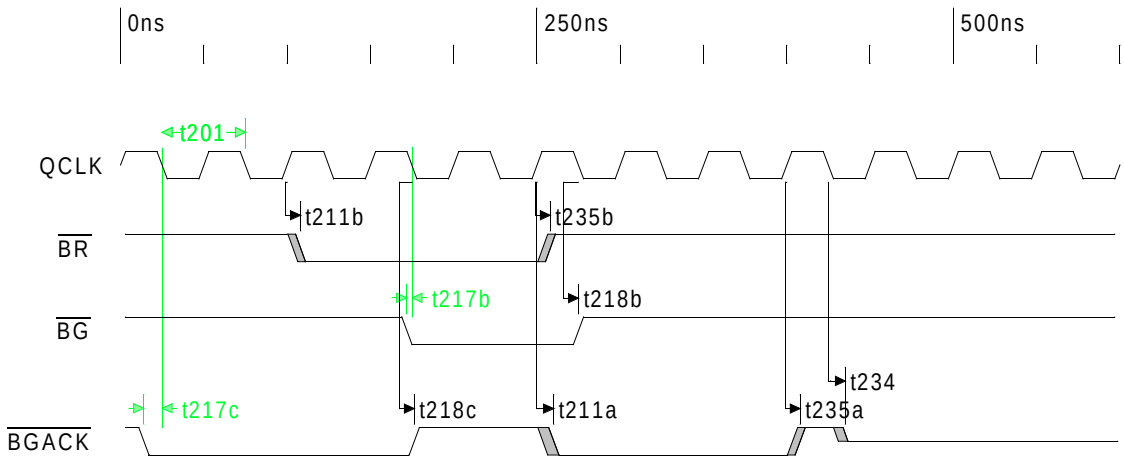
Figure B.2 : QCLK Input Timing—QUICC CLKO1



Note: The timing parameters t005 and t006 are measured between 0.8 and 2 Volts. The timing parameters t005 and t006 can be found in Table B.1.

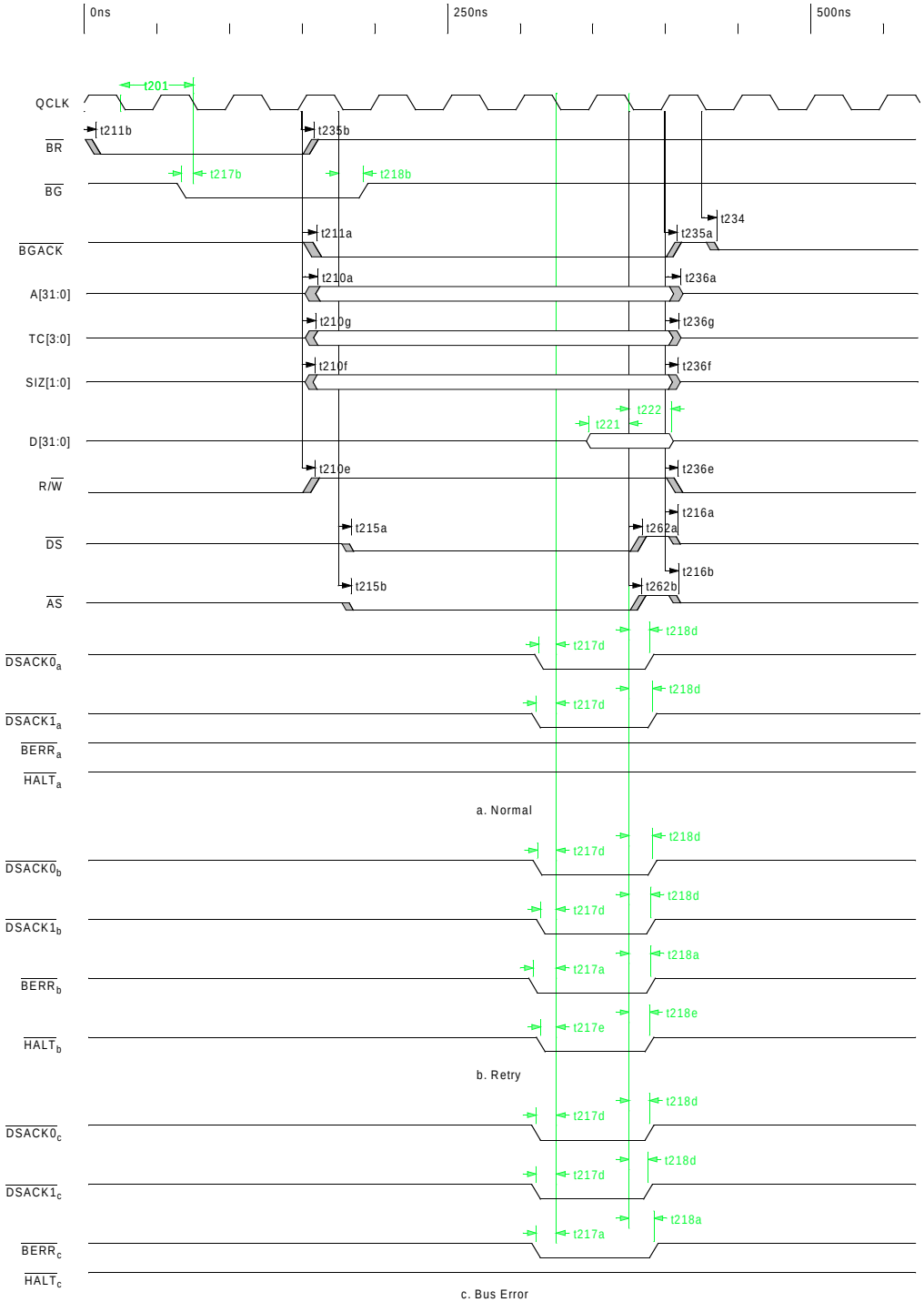
B.3.1.1 QBus (QUICC) Master Cycles

Figure B.3 : QBus (QUICC) Arbitration



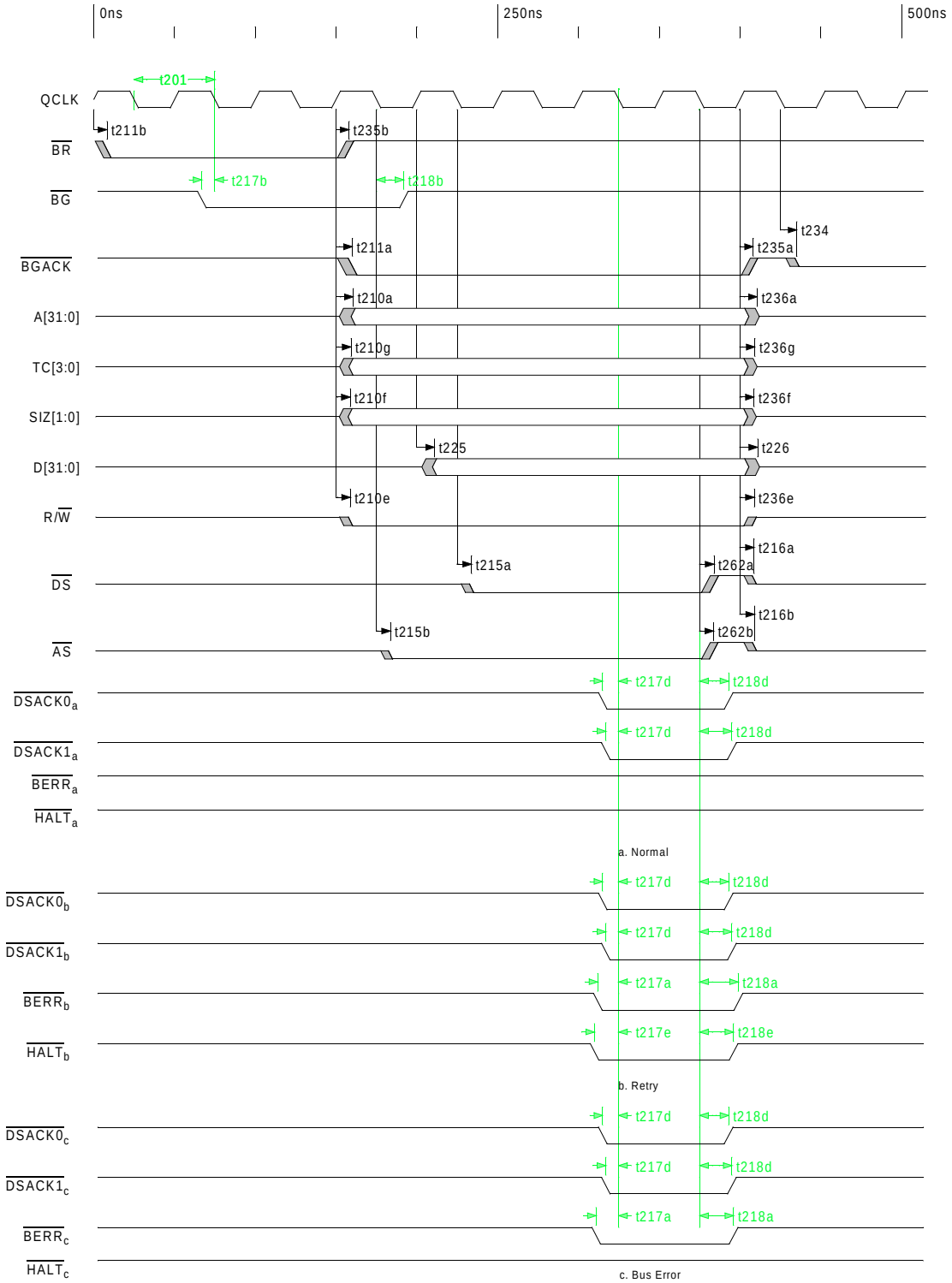
Note: This figure depicts timing in the case where the QSpan requests ownership of the QBus while another QBus master currently owns the bus ($\overline{BB}$/$\overline{BGACK}$ asserted by the other master). The QSpan obtains ownership of the bus after the other master negates $\overline{BB}$/$\overline{BGACK}$.

Figure B.4 : Delayed Single Read—QSpan as QUICC Master



Note: Wait states are not required by the QSpan.

Figure B.5 : Single Write - QSpan as QUICC Master



Note: Wait states are not required by the QSpan.

B.3.1.2 QBus (QUICC) Slave Cycles

Figure B.6 : Delayed Single Read—QSpan as QUICC Slave

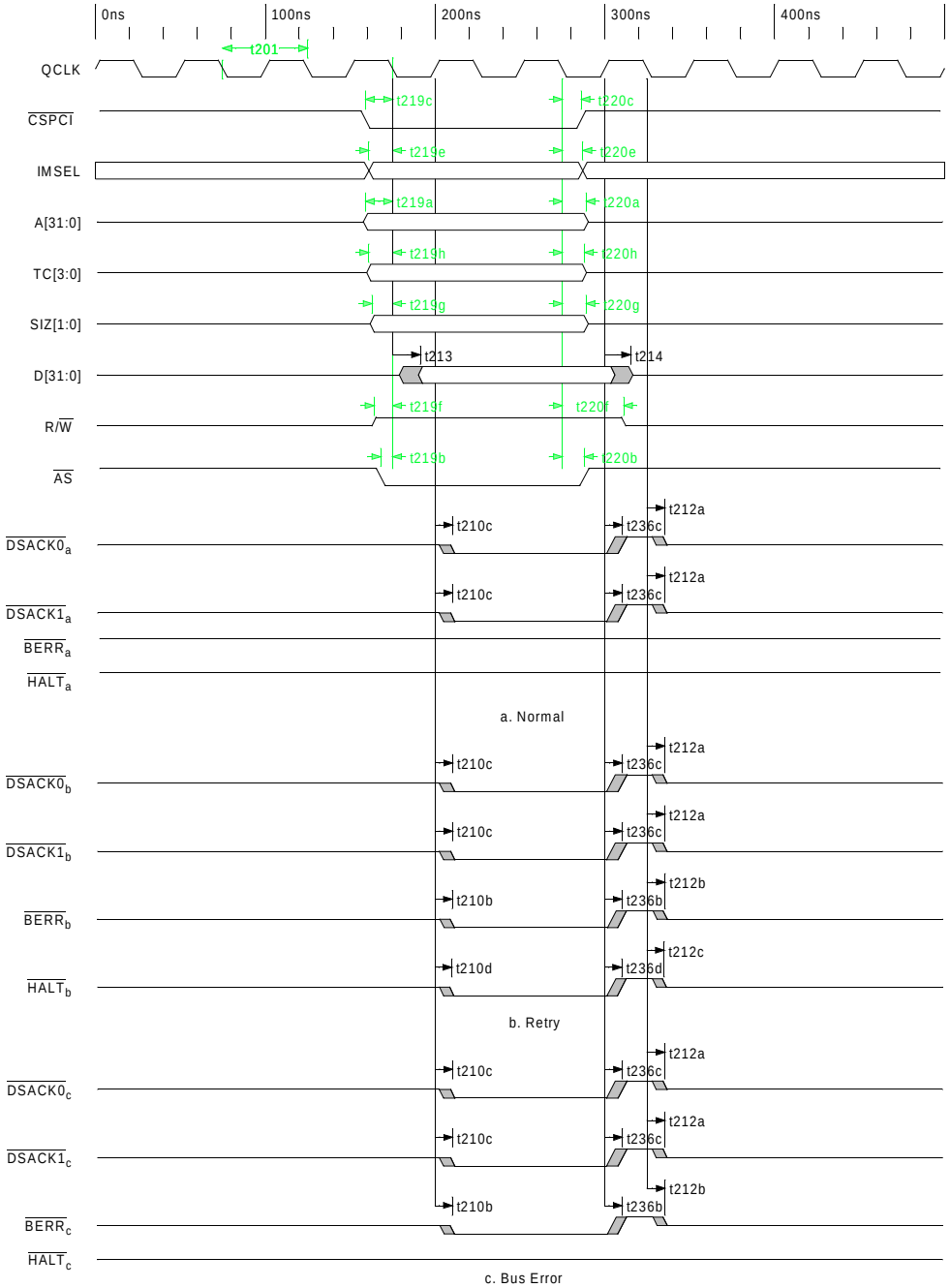


Figure B.7 : Single Write - QSpan as QUICC Slave

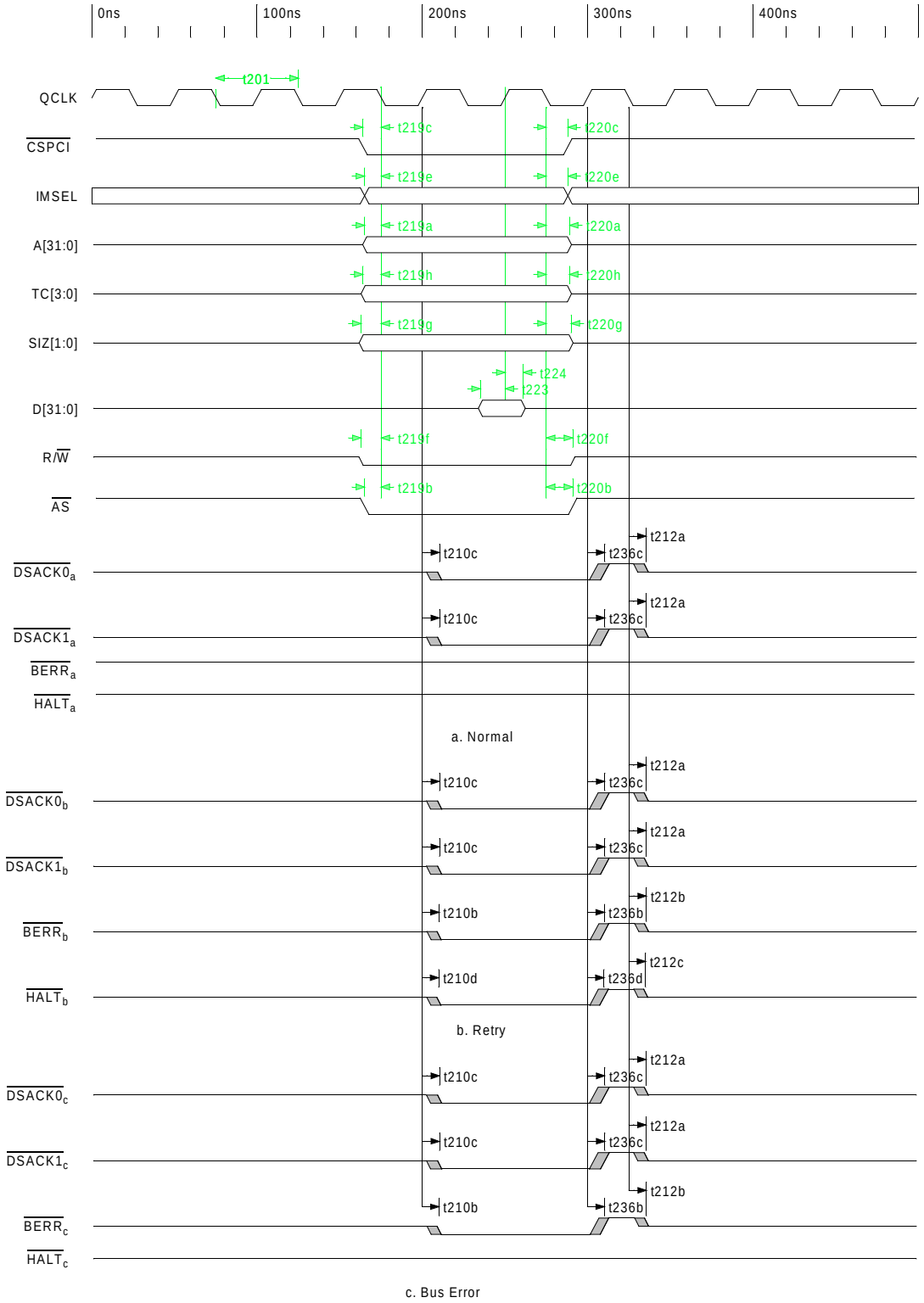
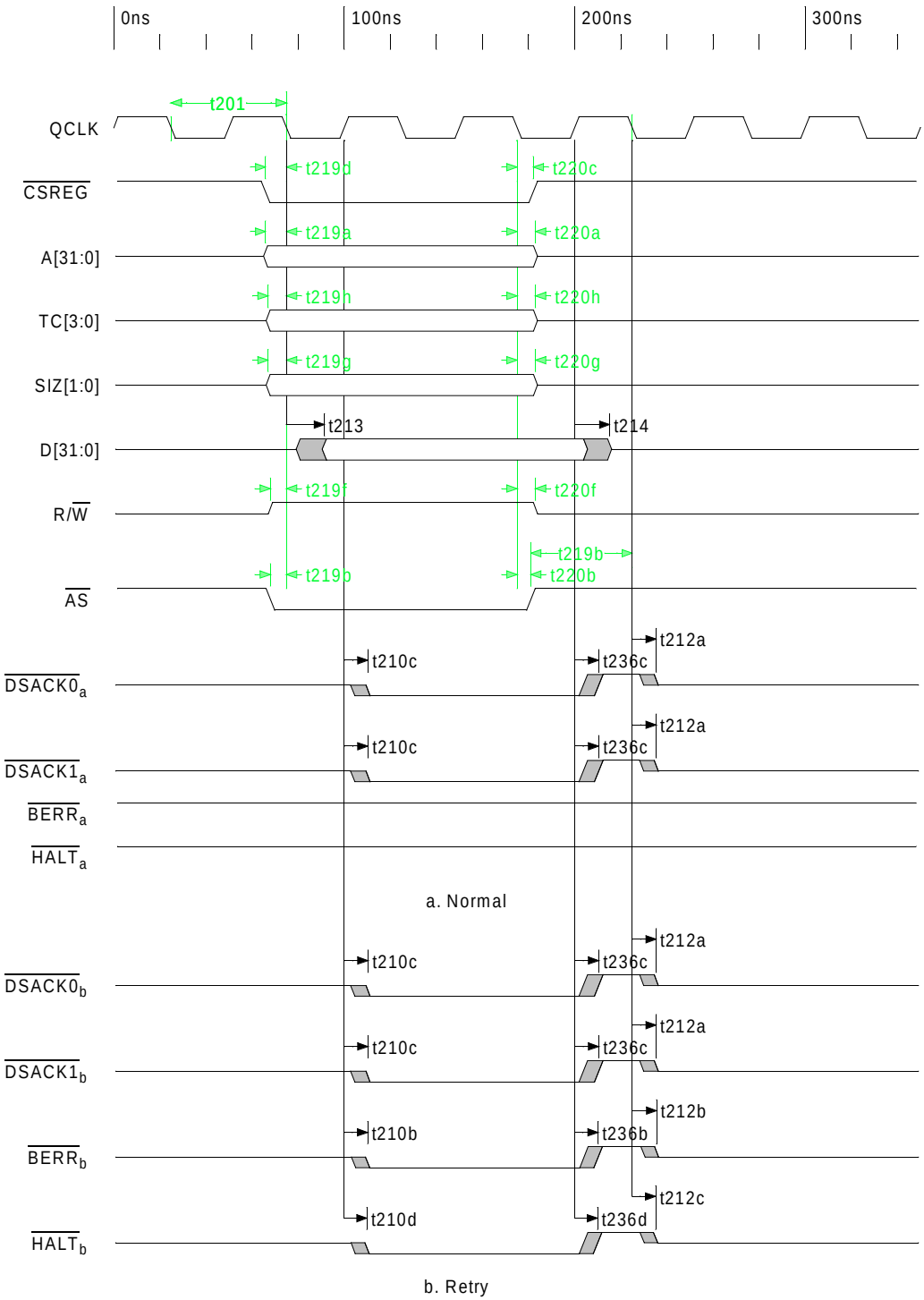
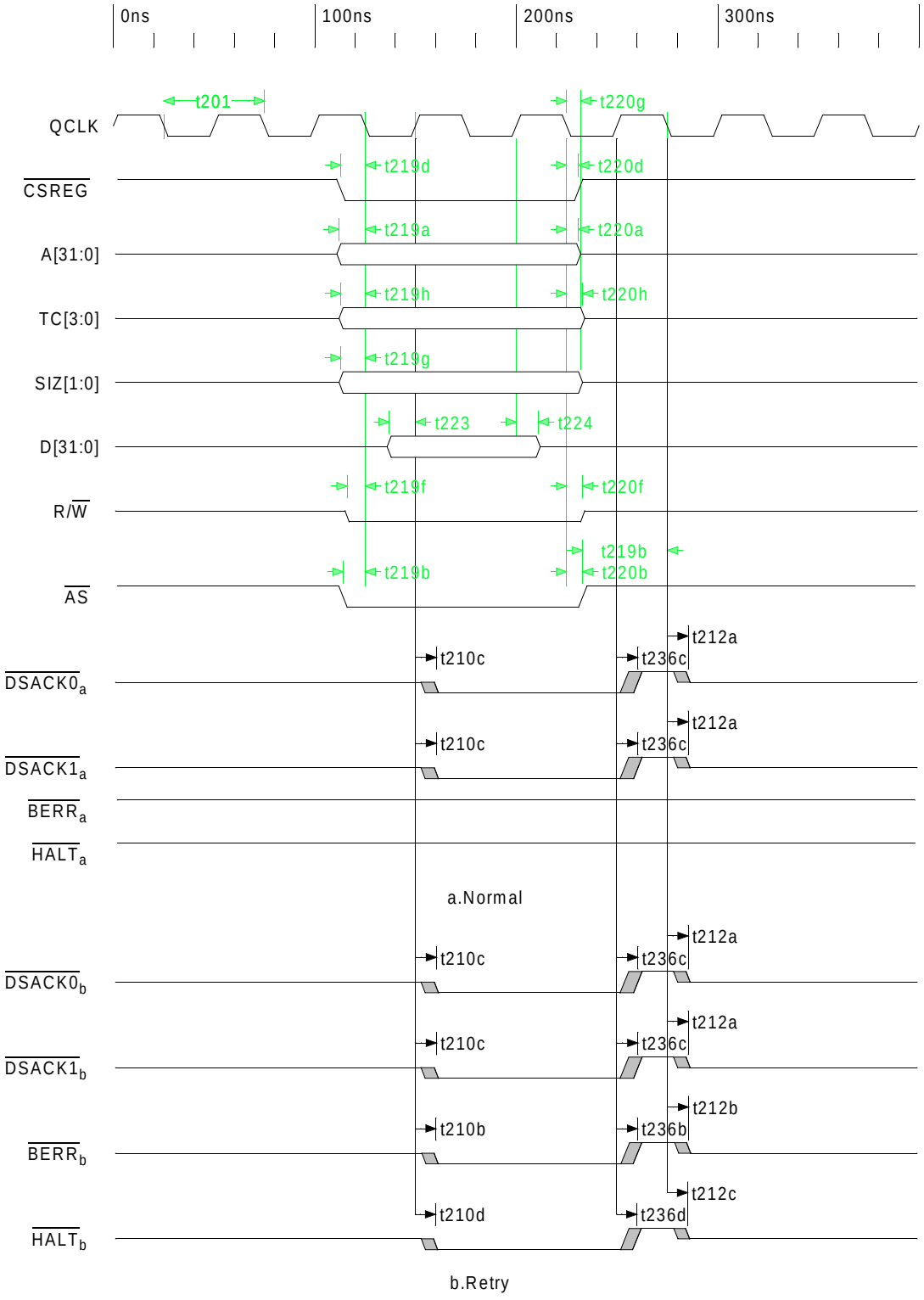


Figure B.8 : Register Read - QSpan as QUICC Slave



Note: Due to internal synchronization, the number of wait states inserted may vary between 2 and 18.

Figure B.9 : Register Write - QSpan as QUICC Slave



B.3.1.3 QBus (QUICC) IDMA Cycles

Figure B.10 : QUICC $\overline{\text{DREQ}}$ Timing

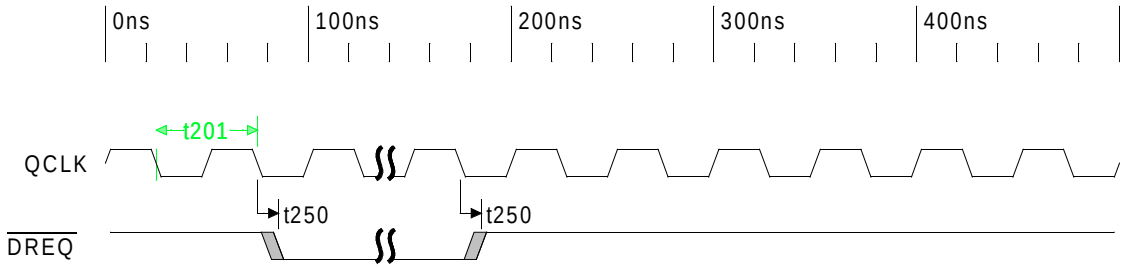


Table B.7 Direction[a] of QBus Signals During QUICC IDMA Cycles

Signal	Single Address		Dual Address	
	Standard Termination	Fast Termination	Standard Termination	Fast Termination
$\overline{\text{CSPCI}}$	I (negated)	I (negated)	I (asserted)	I (asserted)
$\overline{\text{AS}}$	I	I	I	I
D	I (write)/ O (read)	I (write)/ O (read)	I (write)/ O (read)	I (write)/ O (read)
$\overline{\text{DACK}}$	I	I	I	I
$\overline{\text{DONE}}$	I	I	I	I
$\overline{\text{DREQ}}$	O	O	O	O
$\overline{\text{DSACK0}}$ [b]	I	N/A	O	N/A
$\overline{\text{DSACK1}}$	I	N/A	O	N/A
$\overline{\text{BERR}}$	I	N/A	N/A	N/A
$\overline{\text{HALT}}$	I	N/A	N/A	N/A

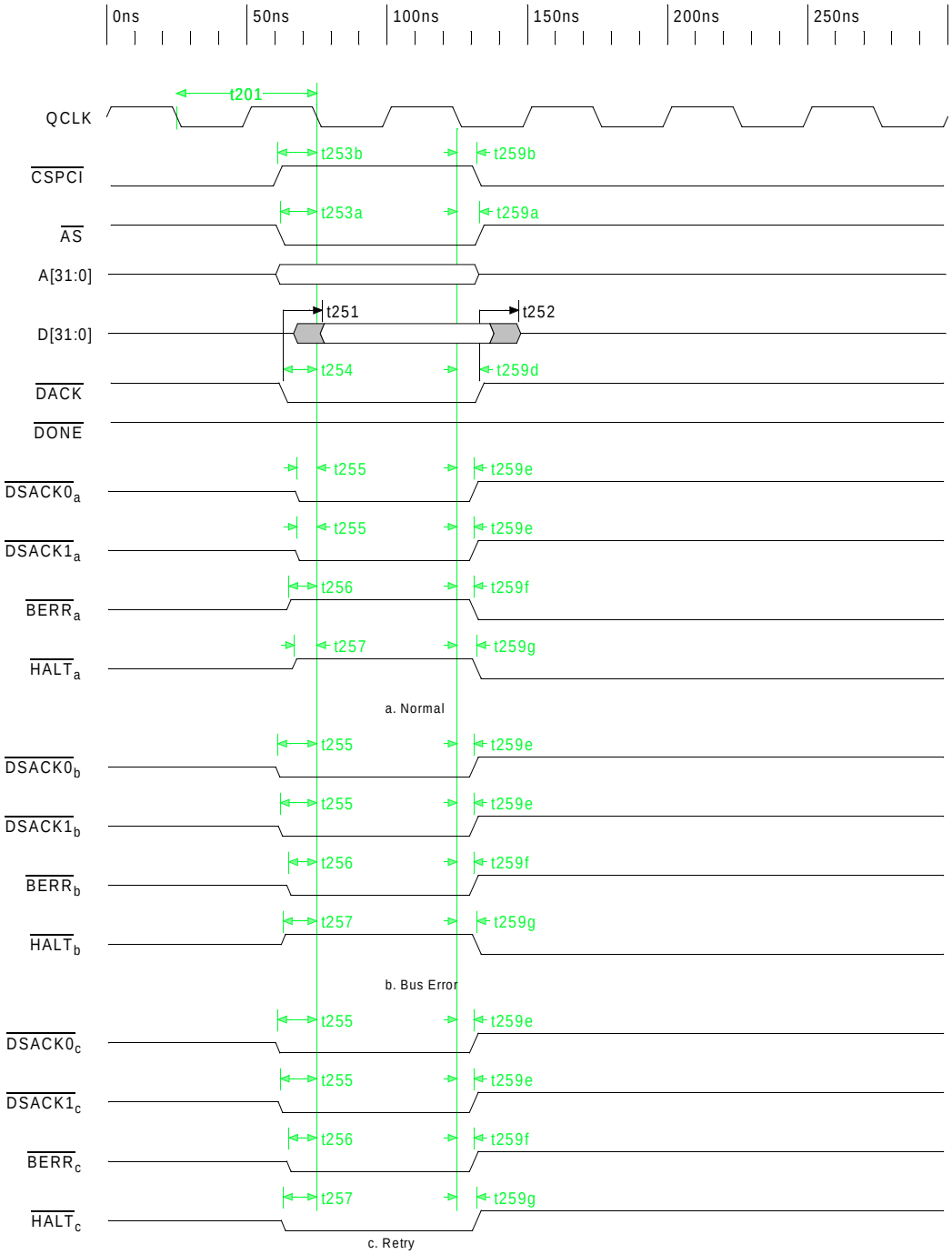
a. I: Input. O: Output. N/A: Not Applicable.

b. $\overline{\text{DSACK0}}$ is not applicable to IDMA transfers when the QBus is a 16-bit port.

Terminology

The expression “standard termination” is used in the following figures in contrast to “fast termination”. “Normal termination” is opposed to “abnormal terminations” such as bus errors and retries. Thus some standard terminations are abnormal terminations (e.g., bus errors with non-fast termination).

Figure B.11 : QUICC IDMA Read (Single Address, Standard Terminations)



Note: A[31:0] is depicted for completeness. It is not examined by the QSpan during IDMA transfers; however, it can be used to drive $\overline{\text{CSPCI}}$. If the Port16 bit of IDMA_CS is disabled, $\overline{\text{DSACK0}}$ is not asserted. $\overline{\text{CSPCI}}$ is negated during single address IDMA cycles.

Figure B.12 : QUICC IDMA Write (Single Address, Standard Terminations)

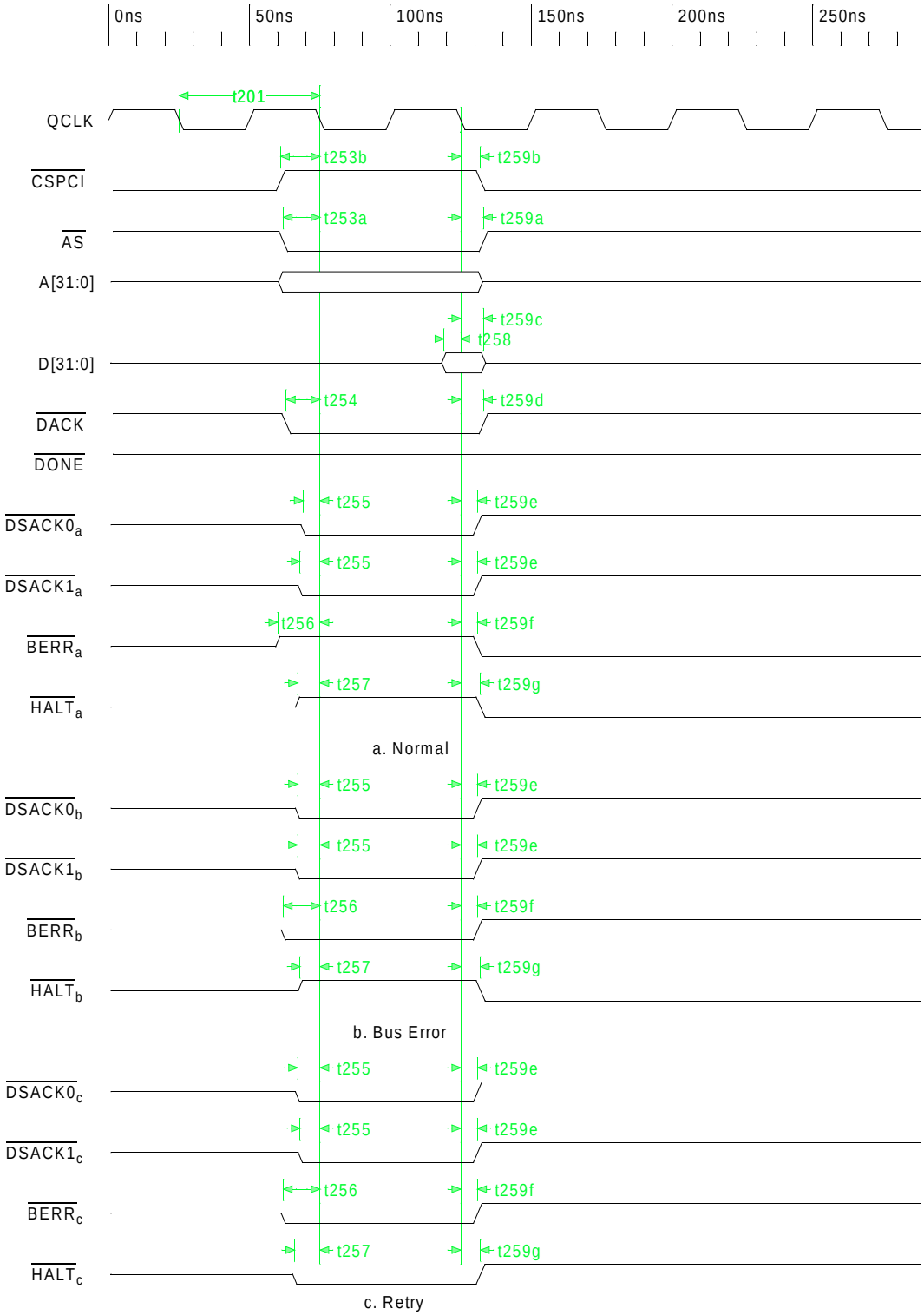
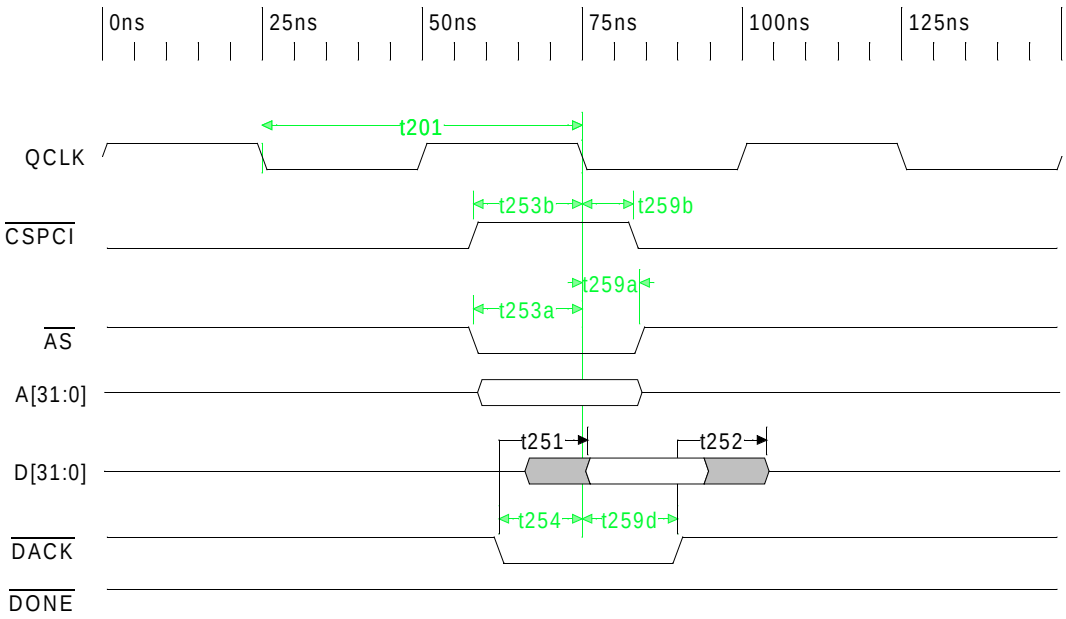


Figure B.13 : QUICC IDMA Read (Single Address, Fast Termination)



Note: During IDMA fast termination cycles the maximum QUICC QCLK frequency is 20 MHz.

Figure B.14 : QUICC IDMA Write (Single Address, Fast Termination)

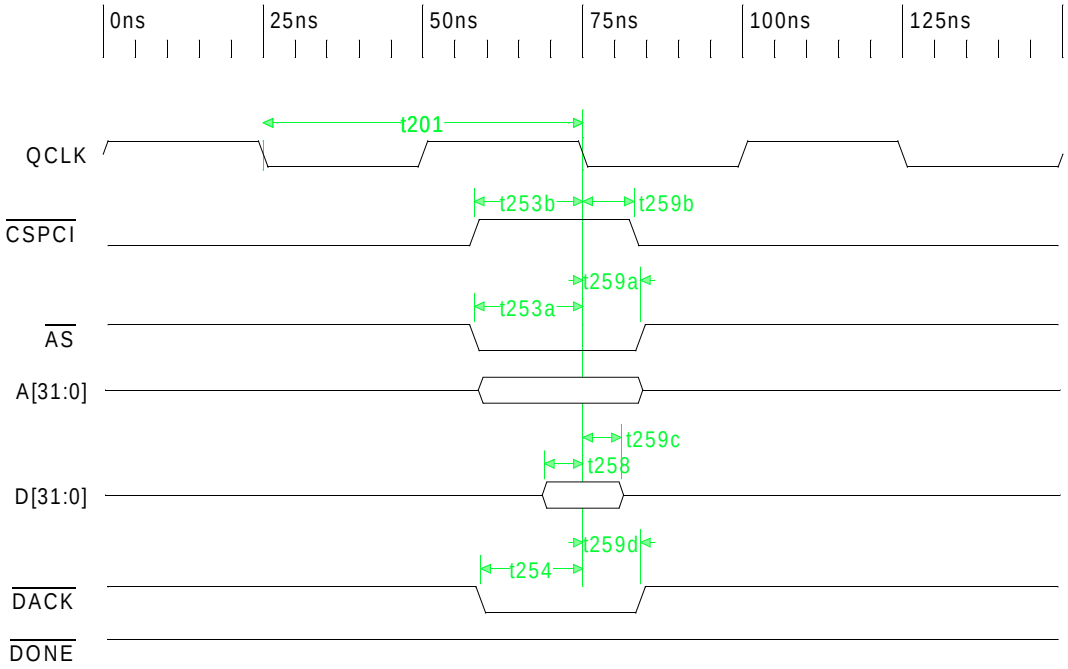
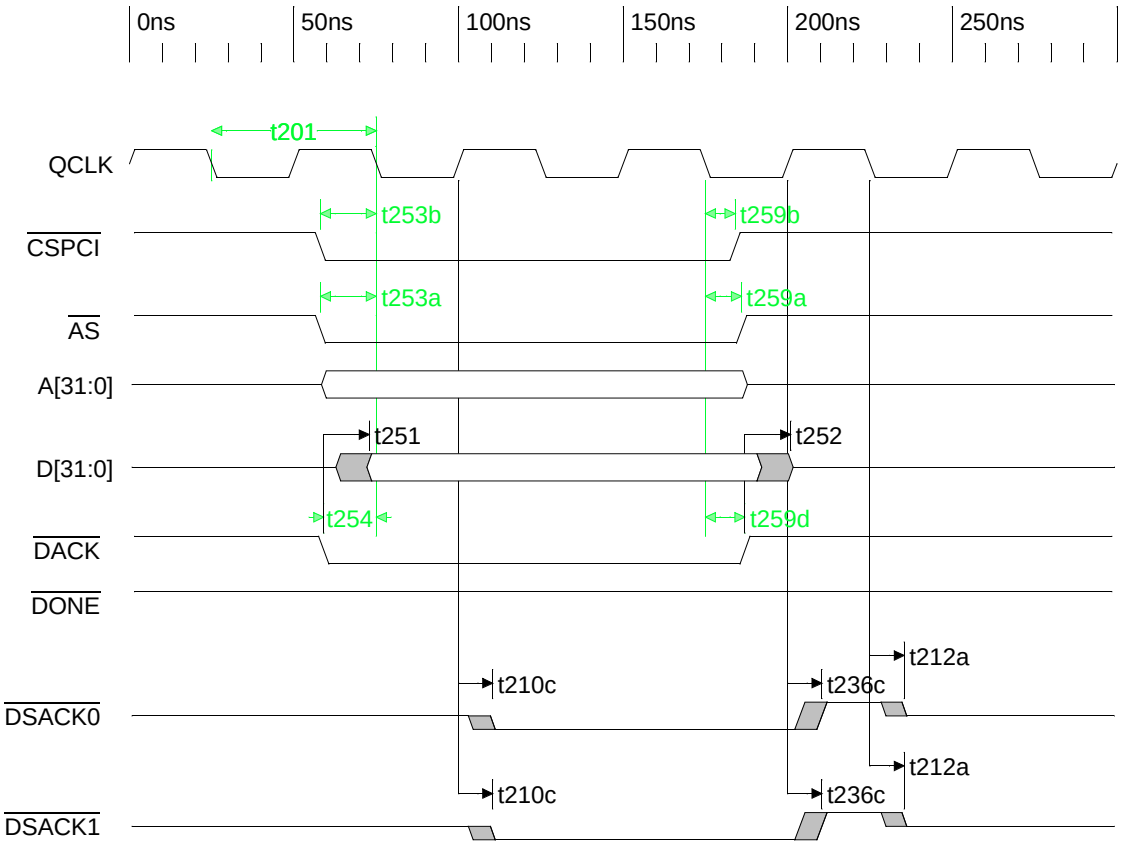


Figure B.15 : QUICC IDMA Read (Dual Address, Standard Termination)



Note: The QSpan does not issue retries or bus errors during dual address IDMA cycles.

Figure B.16 : QUICC IDMA Write (Dual Address, Standard Termination)

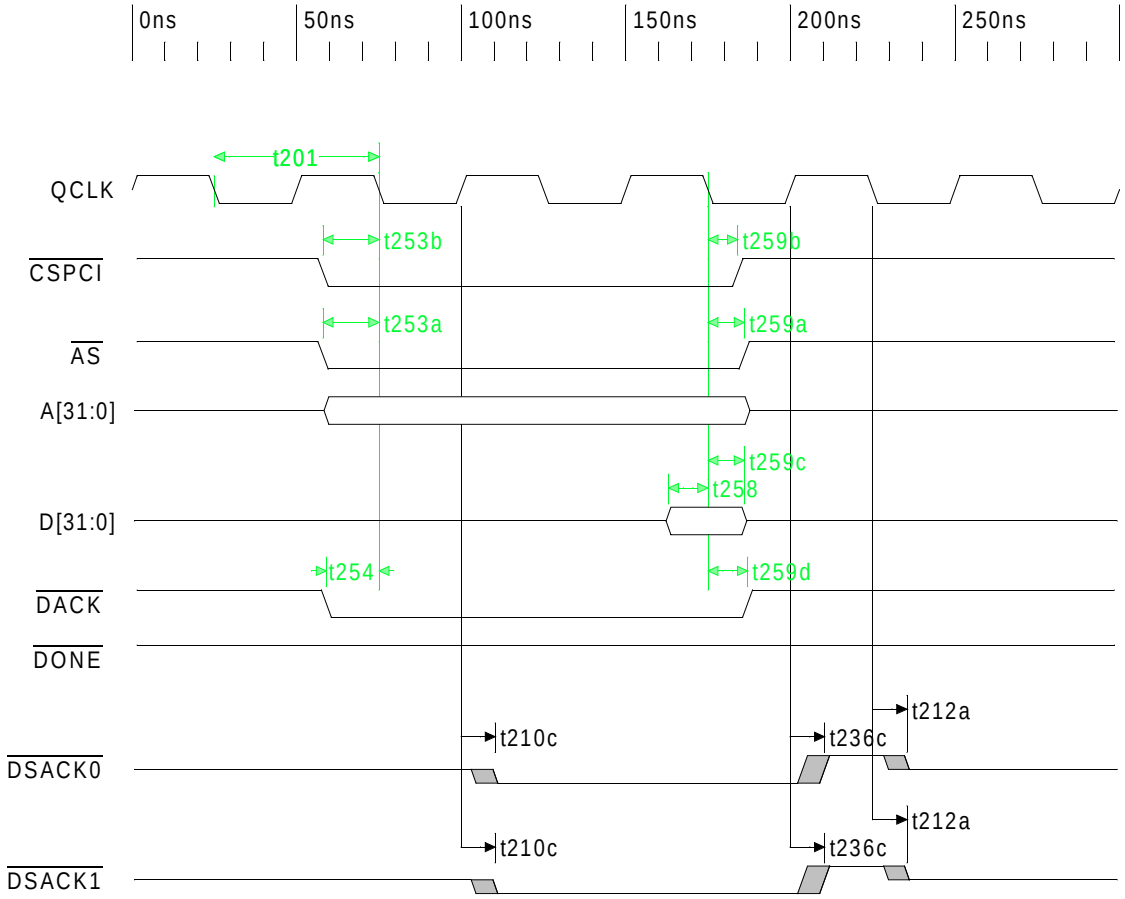
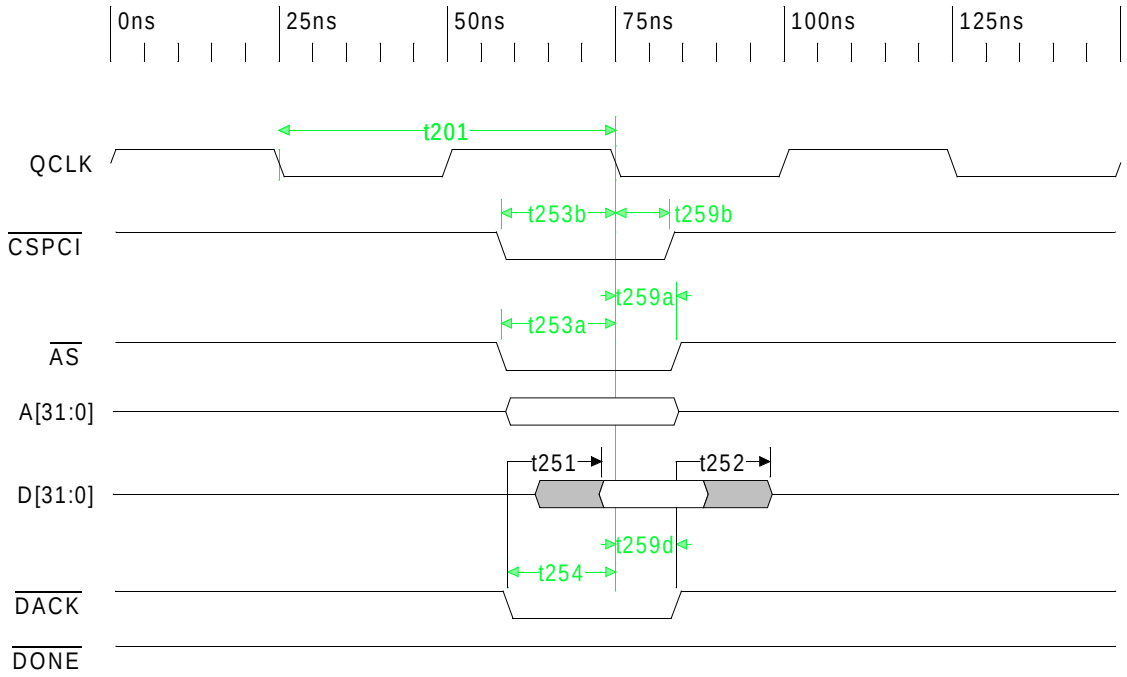
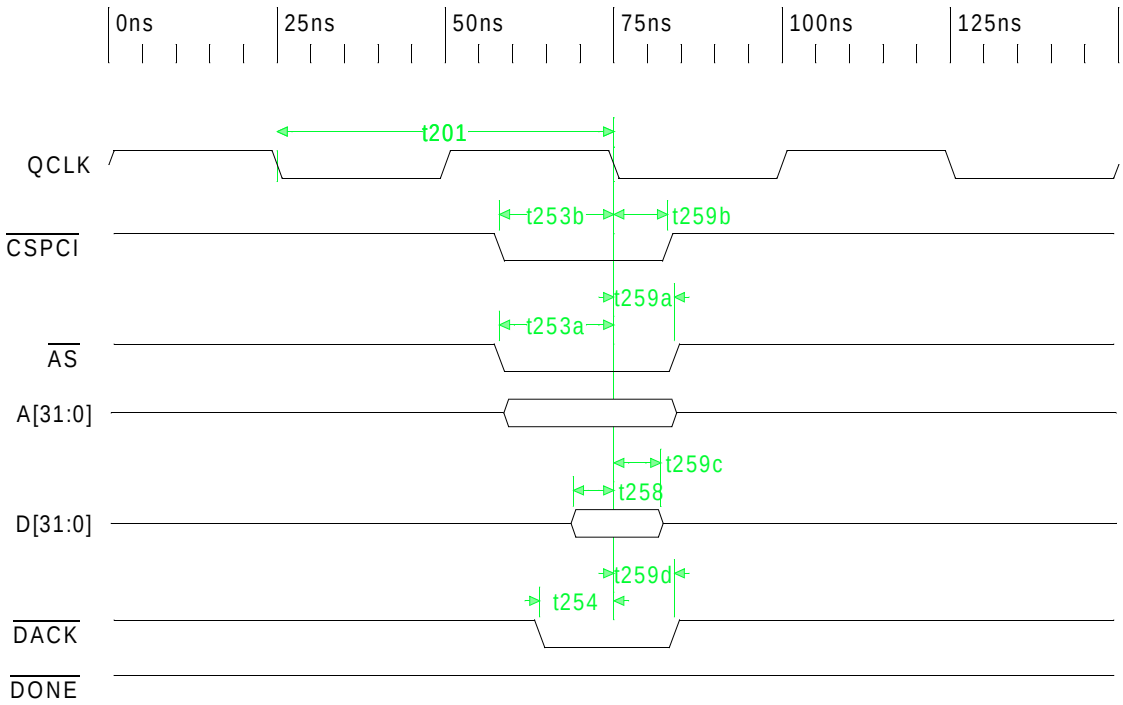


Figure B.17 : QUICC IDMA Read (Dual Address, Fast Termination)



Note: During IDMA fast termination cycles the maximum QUICC QCLK frequency is 20 MHz.

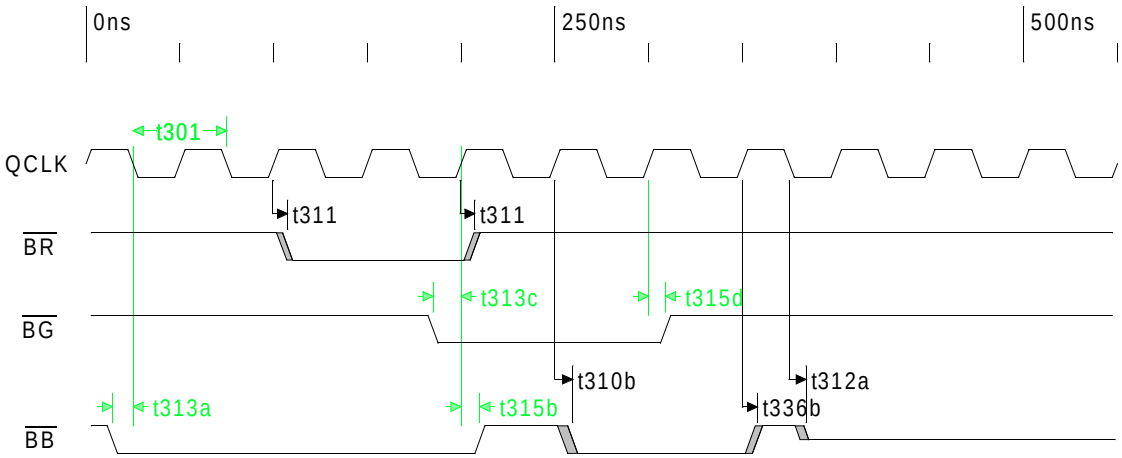
Figure B.18 : QUICC IDMA Read (Dual Address, Fast Termination)



B.3.2 QBus (PowerQUICC) Interface

B.3.2.1 QBus (PowerQUICC) Master Cycles

Figure B.19 : QBus (PowerQUICC) Arbitration



Note: This figure depicts timing in the case where the QSpan requests ownership of the QBus while another QBus master currently owns the bus ($\overline{BB}$/$\overline{BGACK}$ asserted by the other master). The QSpan obtains ownership of the bus after the other master negates $\overline{BB}$/$\overline{BGACK}$.

Figure B.20 : Single Read - QSpan as PowerQUICC Master

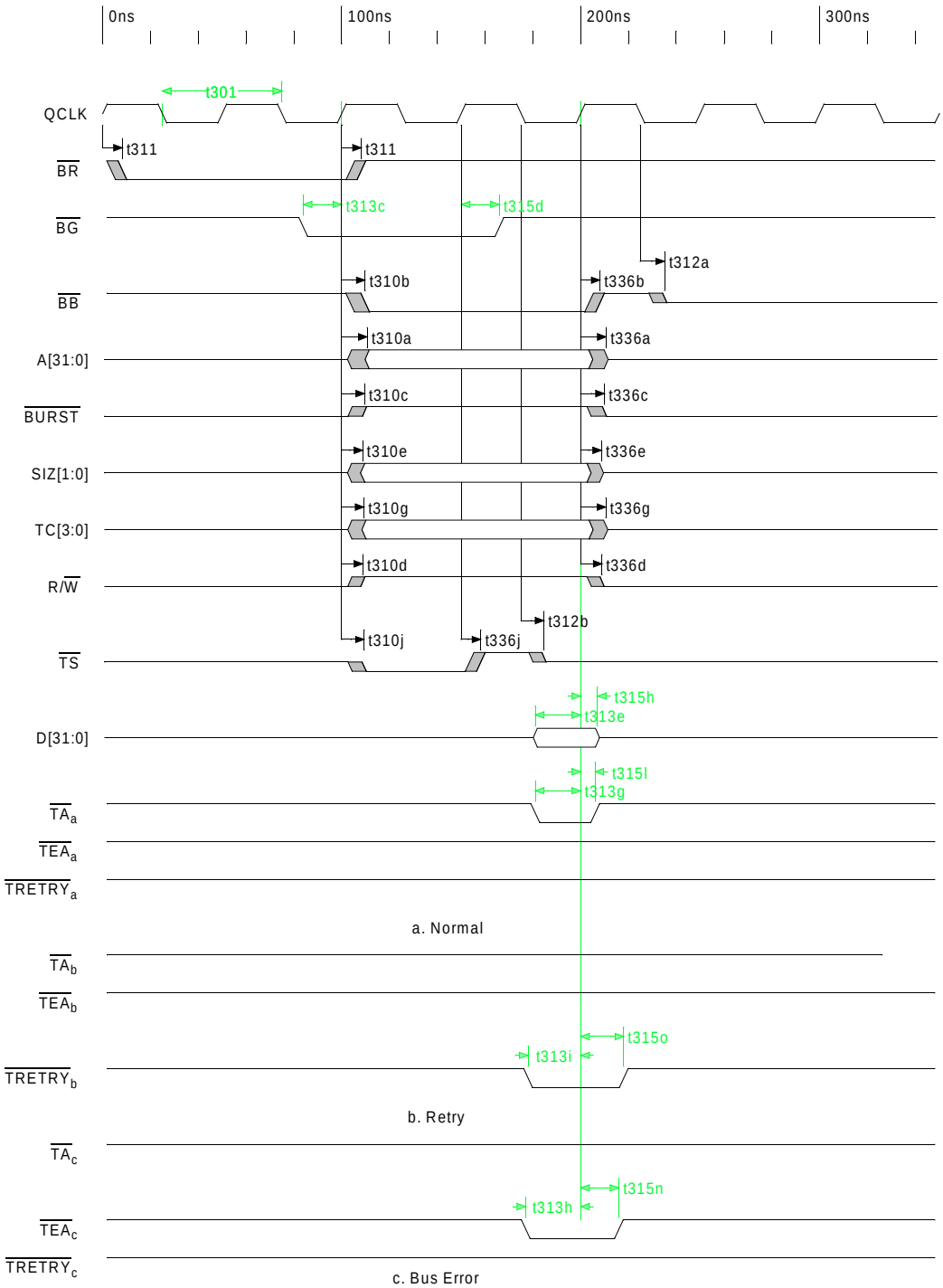


Figure B.21 : Single Write - QSpan as PowerQUICC Master

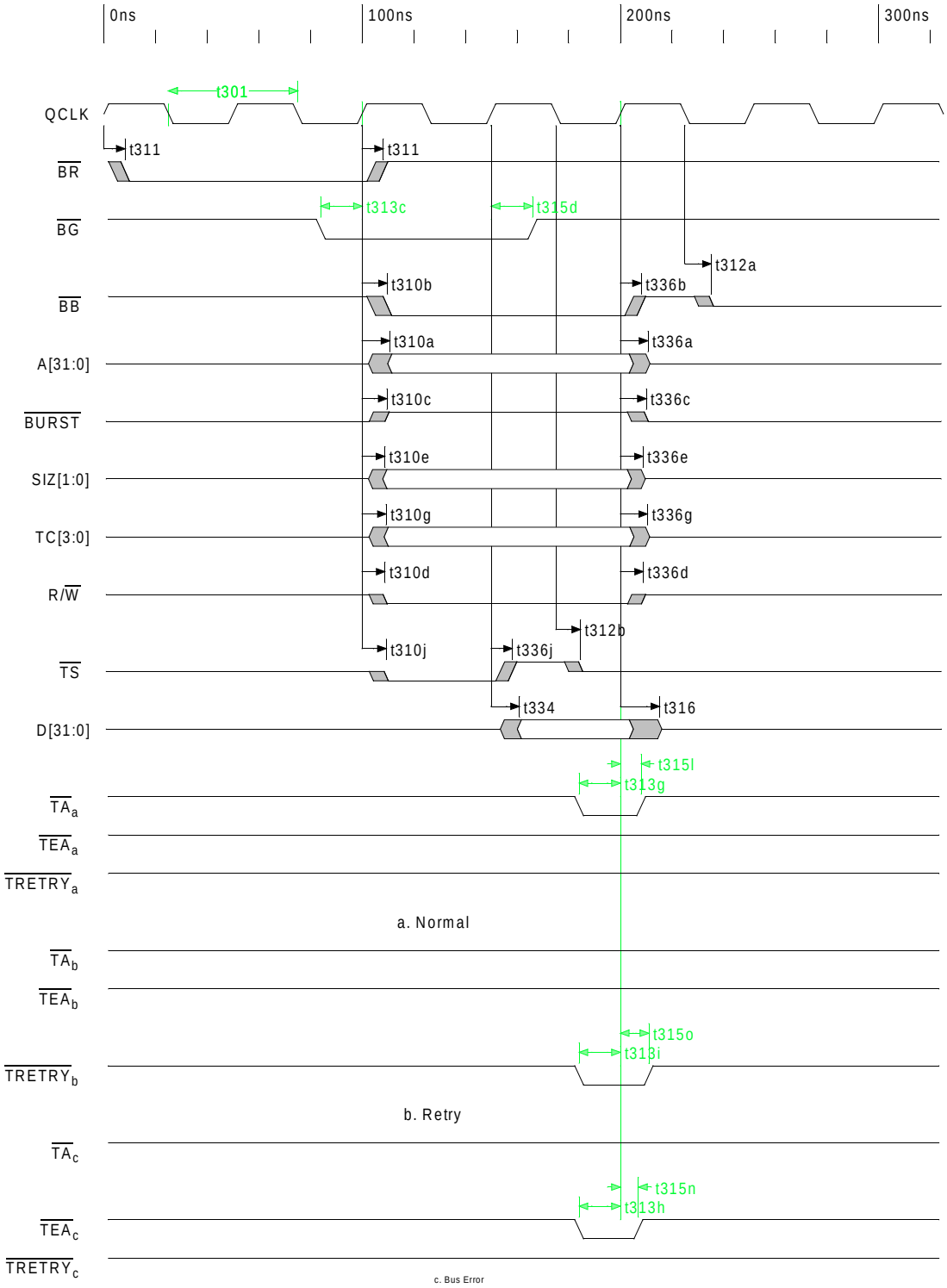


Figure B.22 : Burst Read - QSpan as PowerQUICC Master

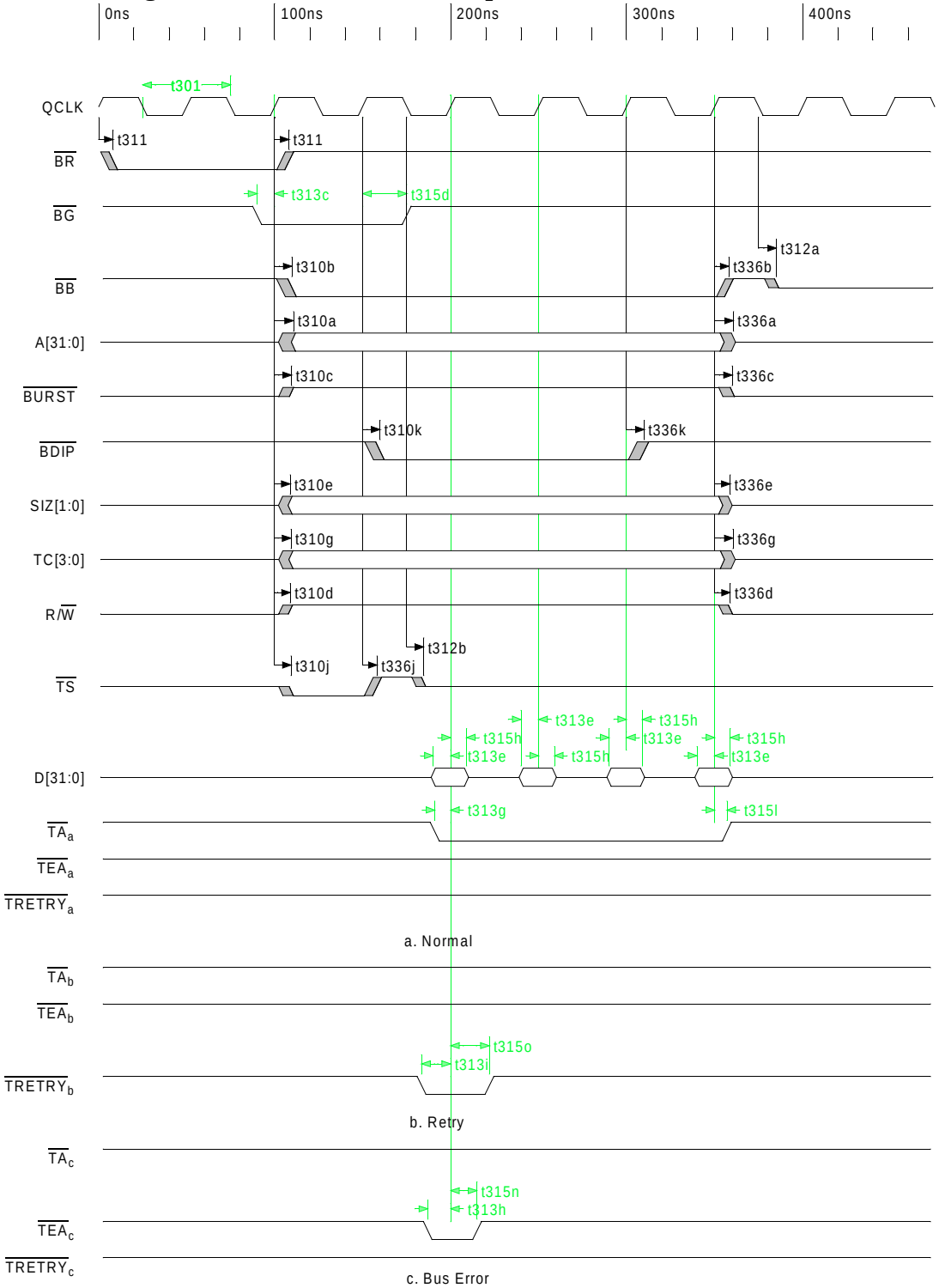
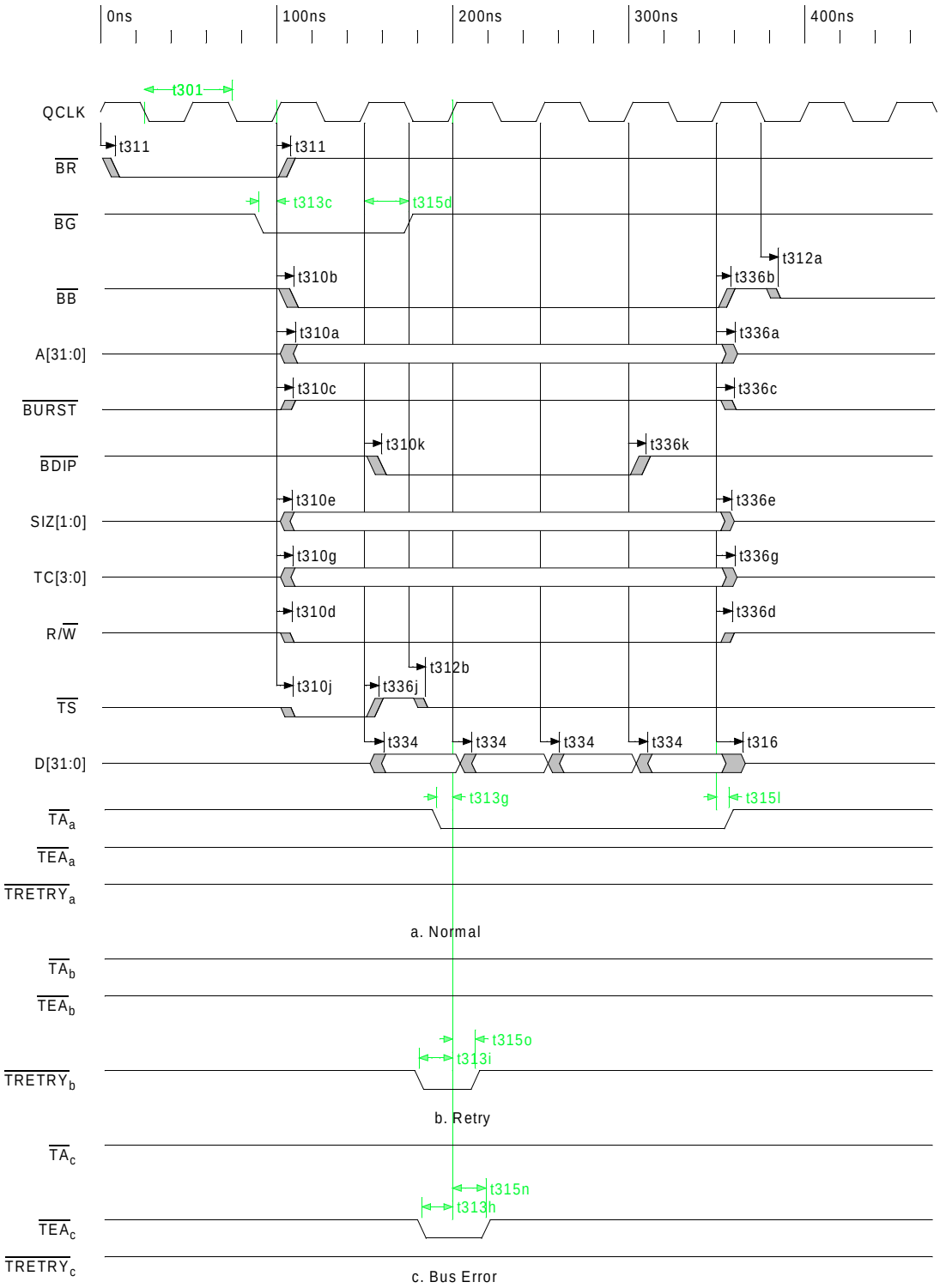


Figure B.23 : Aligned Burst Write - QSpan as PowerQUICC Master



B.3.2.2 QBus (PowerQUICC) Slave Cycle

Figure B.24 : Single Read - QSpan as PowerQUICC Slave

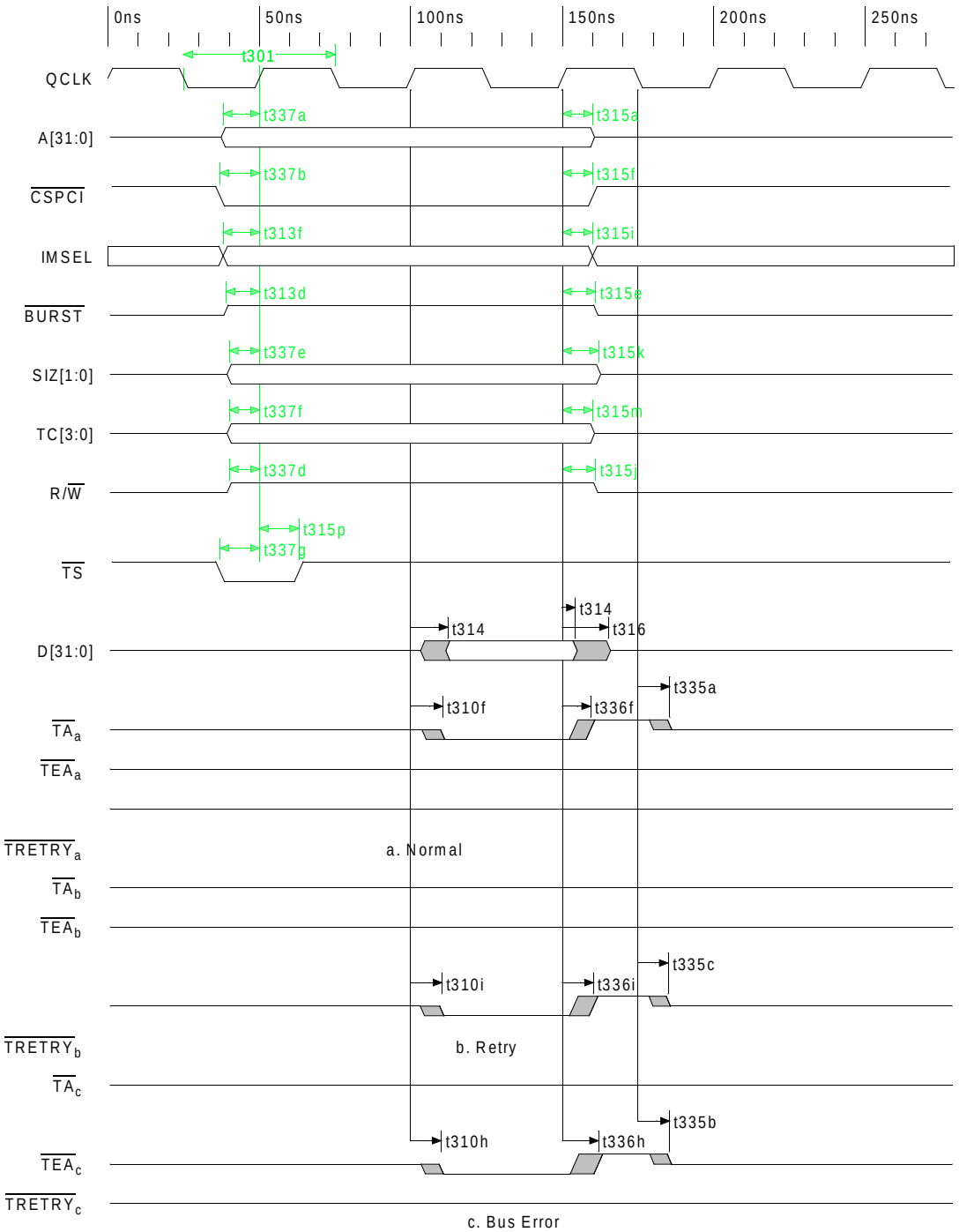


Figure B.25 : Single Write - QSpan as PowerQUICC Slave

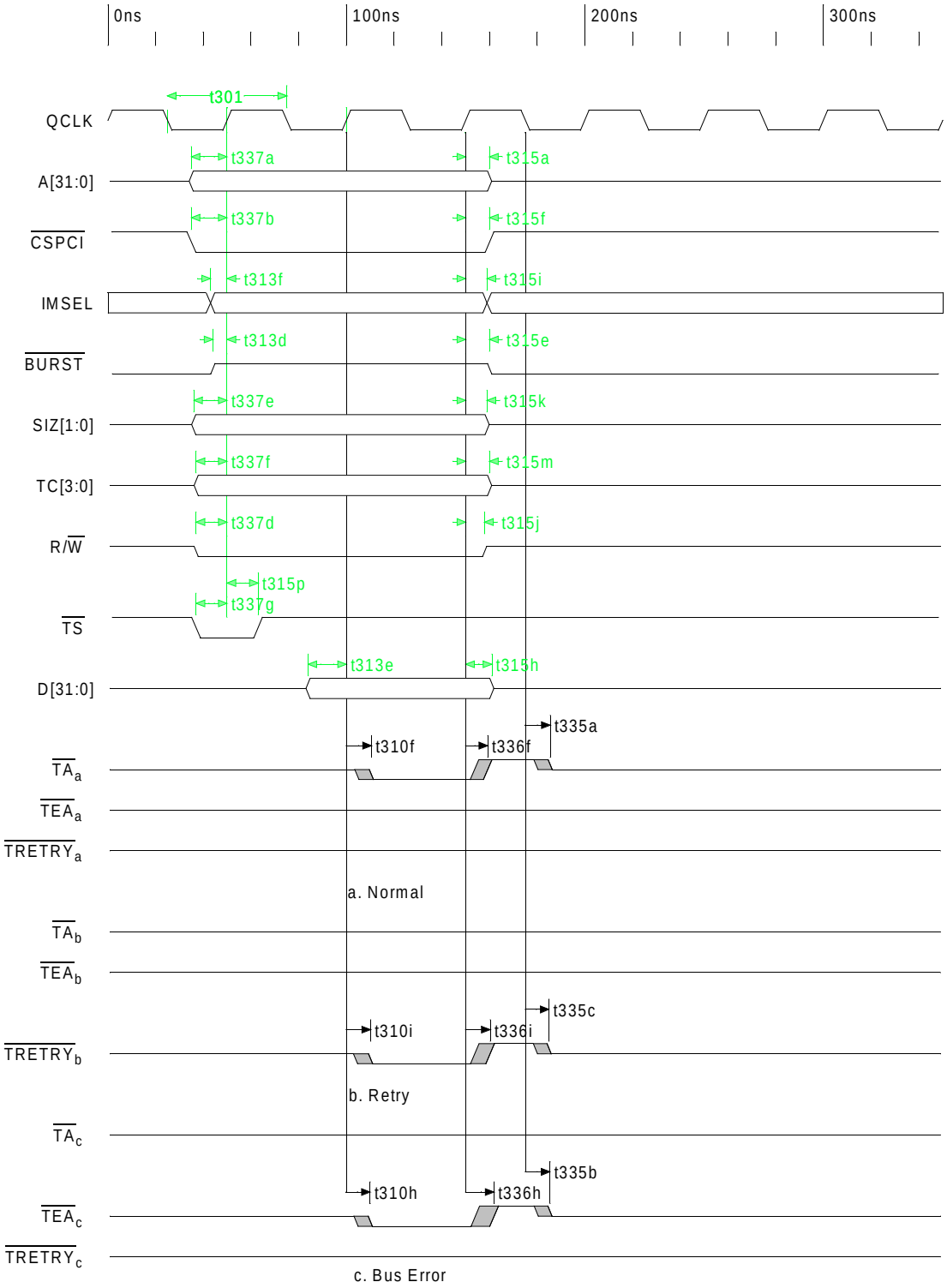


Figure B.26 : Burst Read—QSpan as PowerQUICC Slave

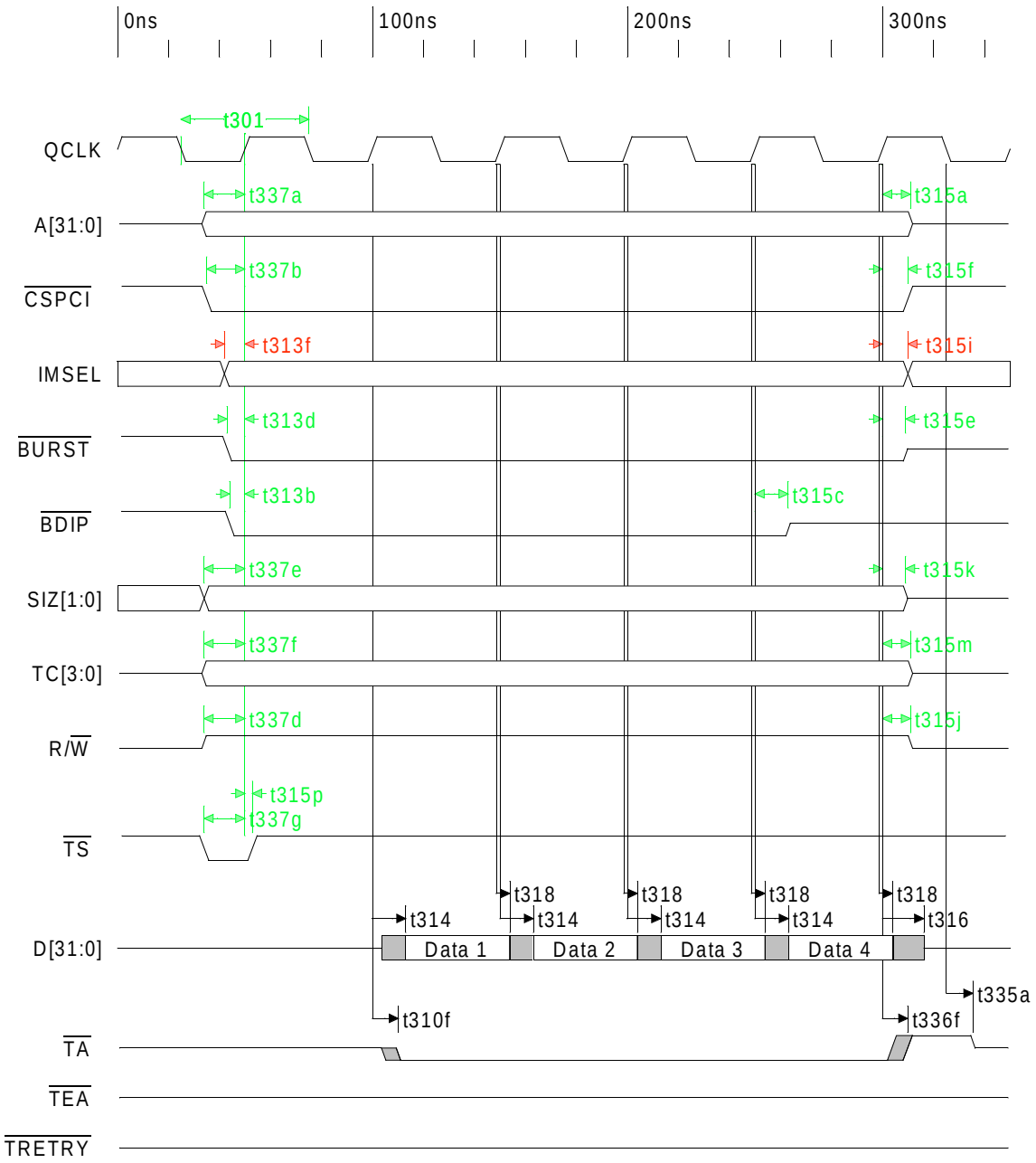
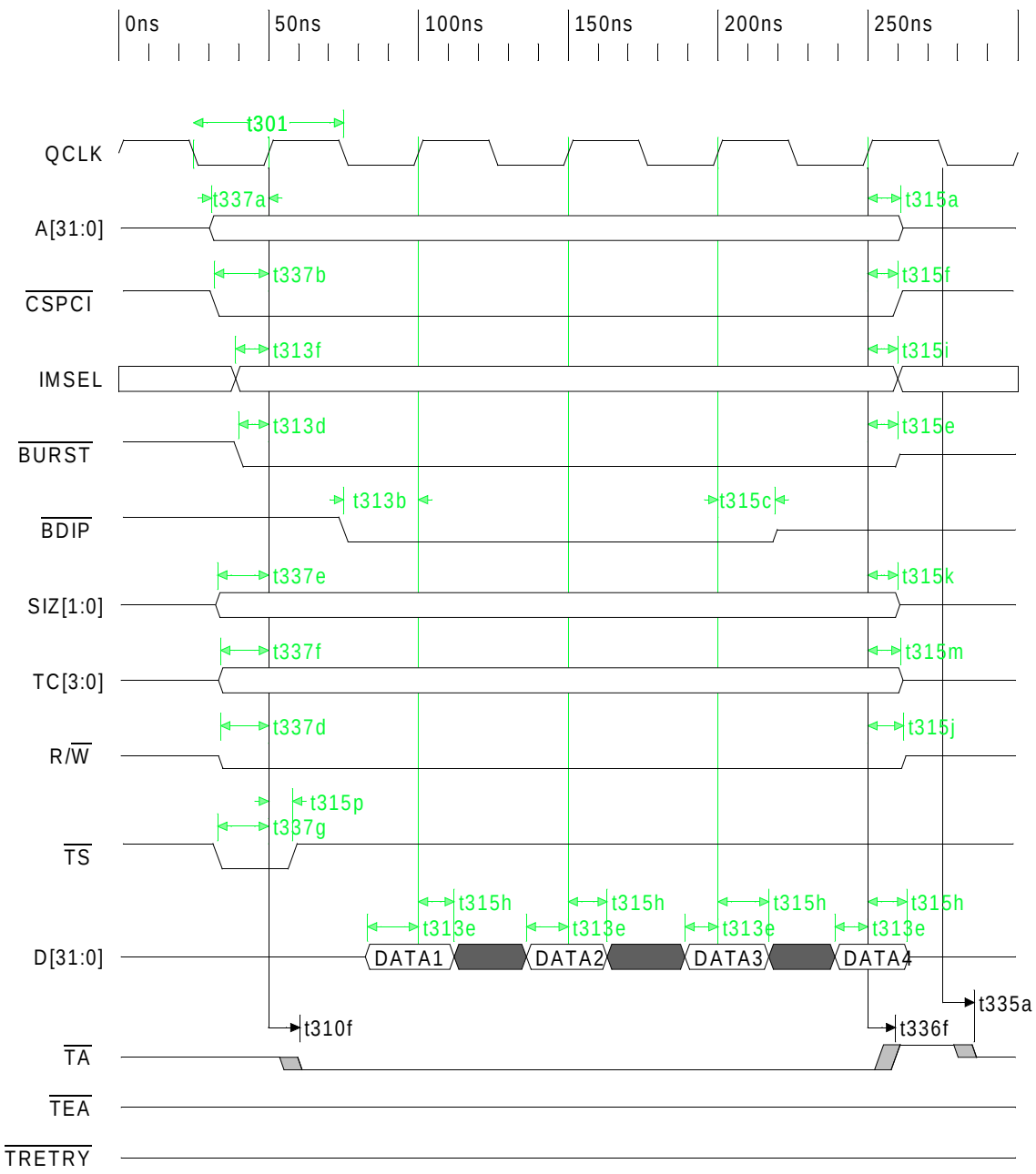


Figure B.27 : Burst Write—QSpan as PowerQUICC Slave



Note: A minimum of one wait state will be inserted if starting address is not aligned to a 16-byte boundary.

Figure B.28 : Register Read—QSpan as PowerQUICC Slave

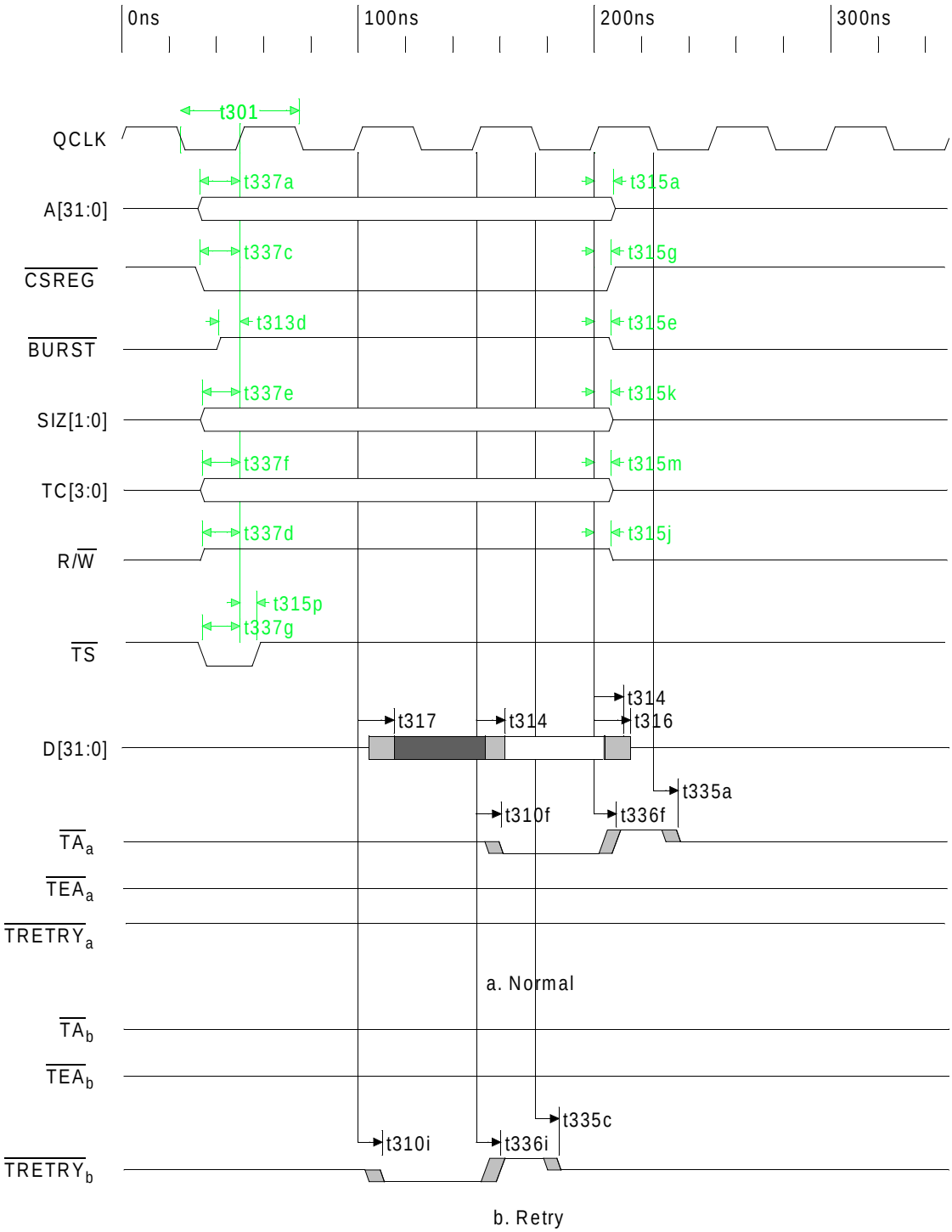
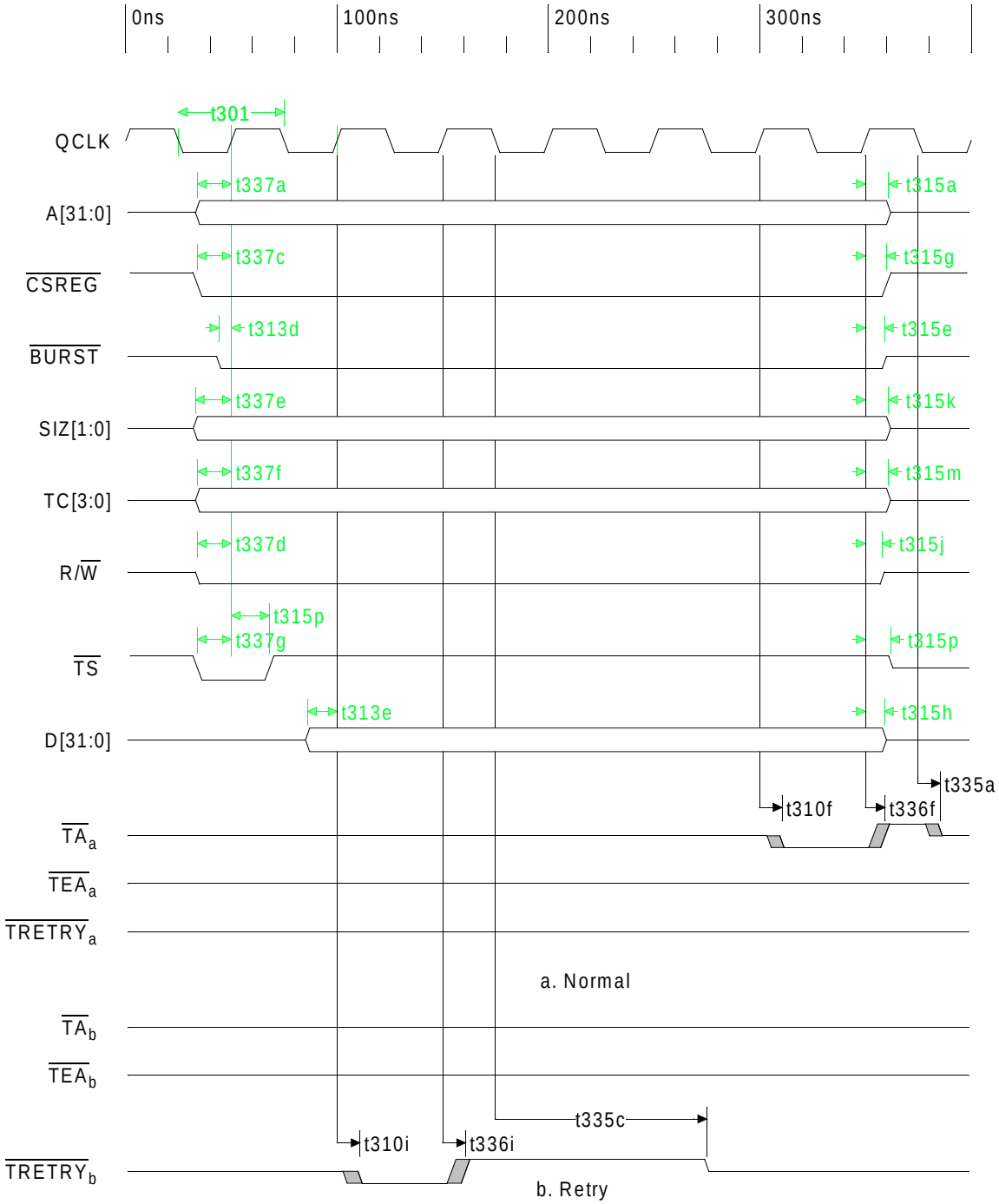


Figure B.29 : Register Write—QSpan as PowerQUICC Slave



Note: Due to internal synchronization, the number of wait states may vary.

B.3.2.3 QBus (PowerQUICC) IDMA Cycles

Figure B.30 : PowerQUICC $\overline{\text{DREQ}}$ Timing

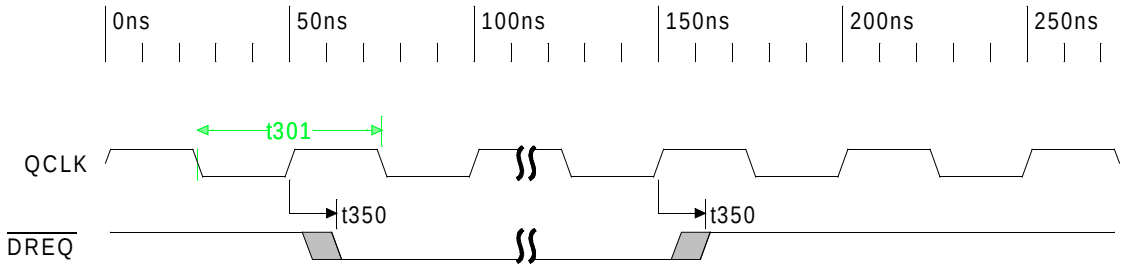


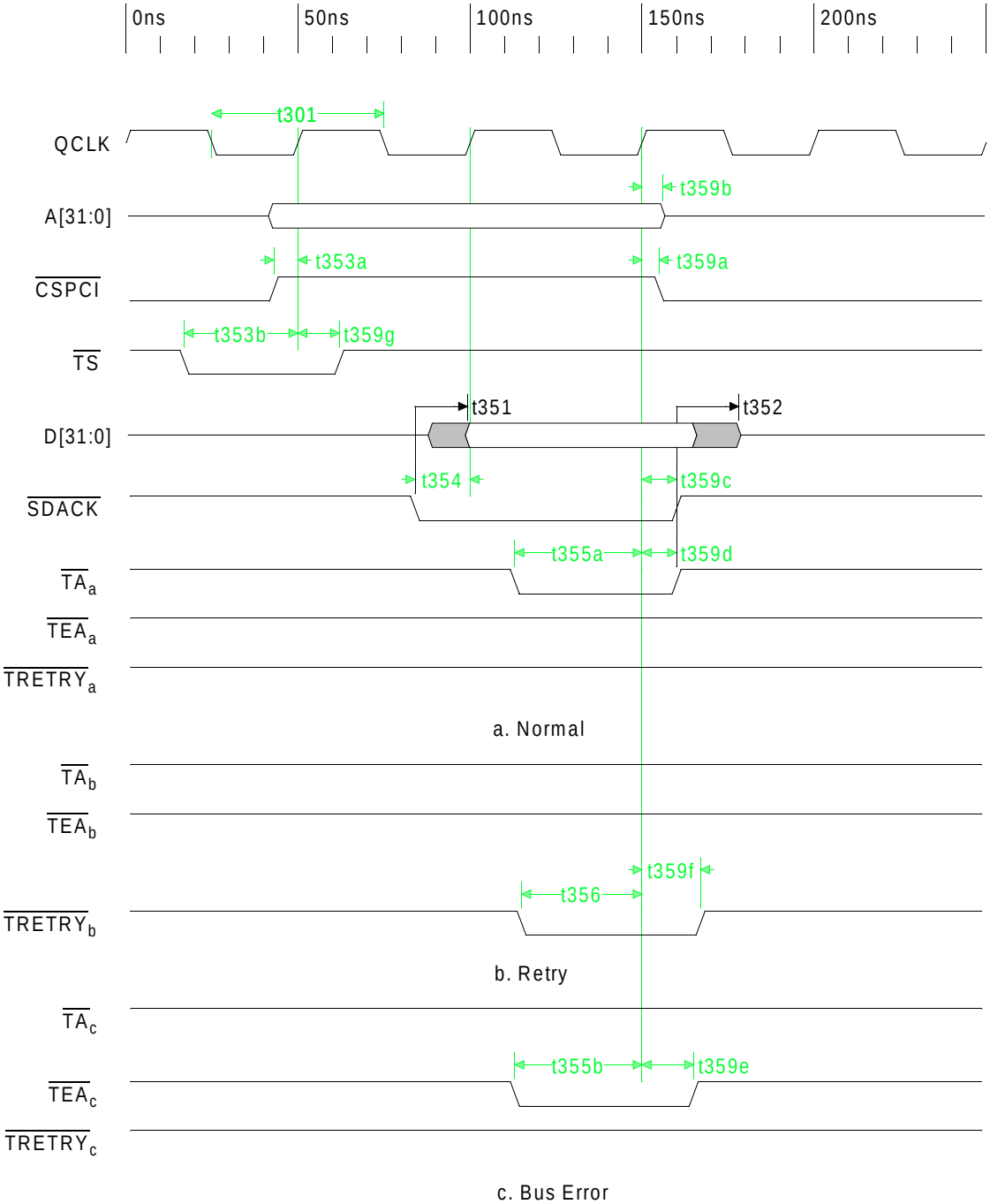
Table B.8 Direction[a] of QBus Signals During PowerQUICC IDMA Cycles

Signal	Single Address	Dual Address
$\overline{\text{CSPCI}}$	I (negated)	I (asserted)
$\overline{\text{TC}}$[3:0]	N/A	I
$\overline{\text{TS}}$	I	I
D	I (write)/ O (read)	I (write)/ O (read)
$\overline{\text{SDACK}}$	I	I
$\overline{\text{DONE}}$[b]	N/A	N/A
$\overline{\text{DREQ}}$	O	O
$\overline{\text{TA}}$	I	O
$\overline{\text{TEA}}$	I	N/A
$\overline{\text{TRETRY}}$	I	N/A

a. I: Input. O: Output. N/A: Not Applicable.

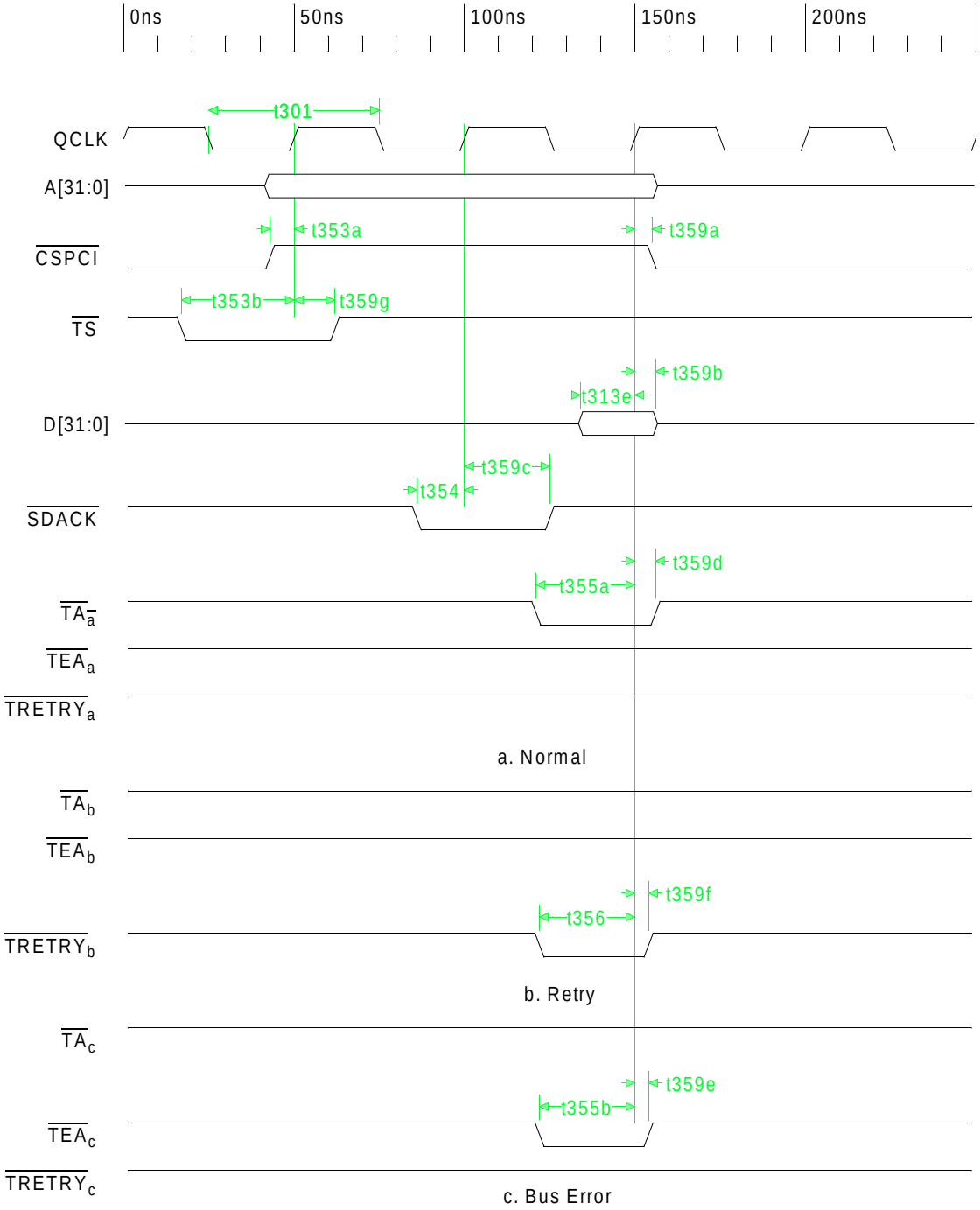
b. The QSpan ignores $\overline{\text{DONE}}$ during PowerQUICC IDMA cycles

Figure B.31 : PowerQUICC IDMA Read (Single Address)



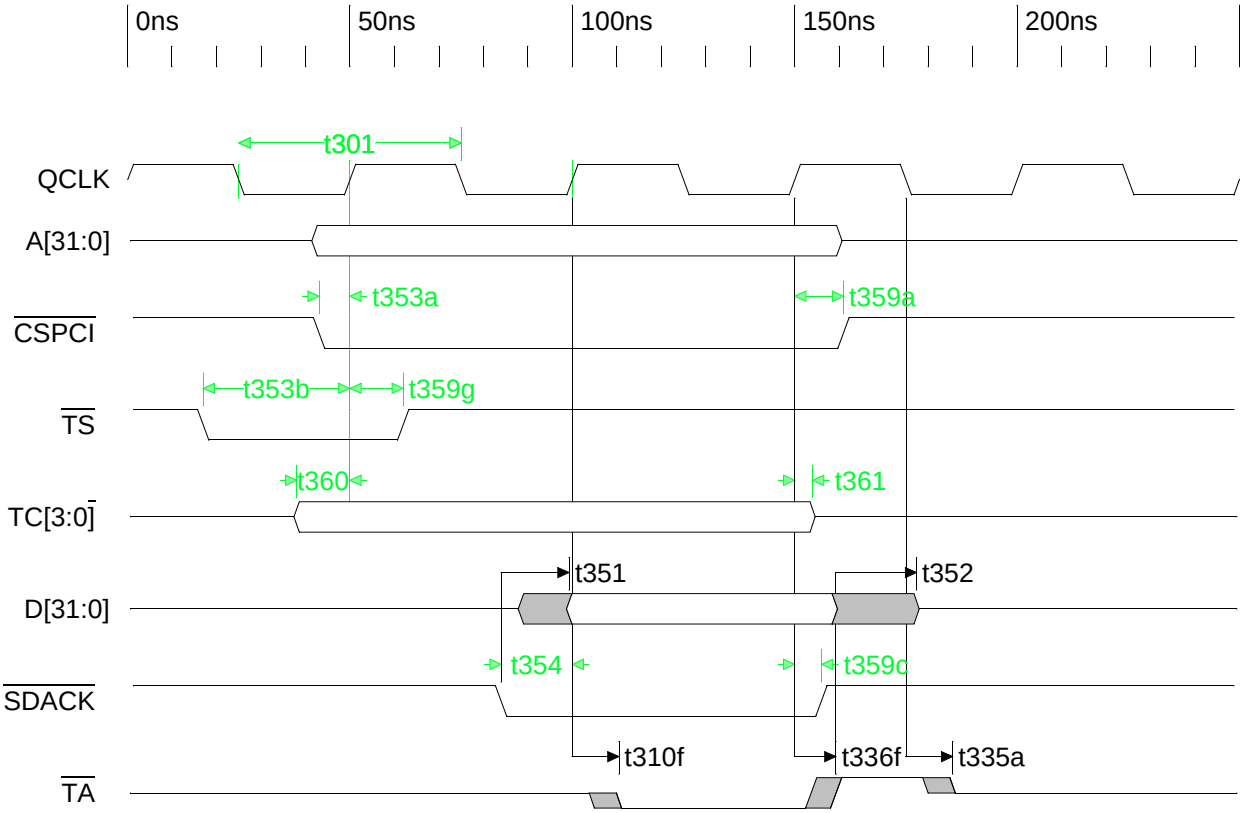
Note: A[31:0] is depicted for completeness. It is not examined by the QSpan during IDMA transfer; however, it can be used to drive $\overline{\text{CSPCI}}$.

Figure B.32 : PowerQUICC IDMA Write (Single Address)



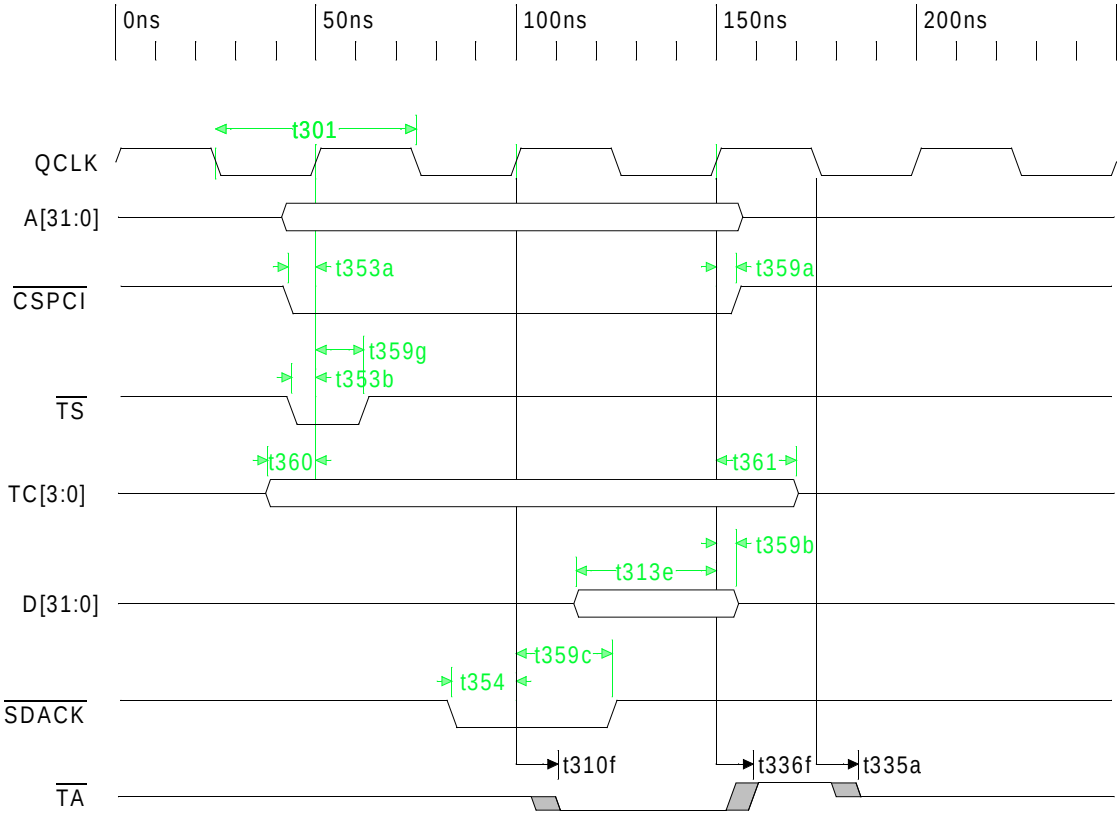
Note: A[31:0] is depicted for completeness. It is not examined by the QSpan during IDMA transfers; however, it can be used to drive $\overline{\text{CSPCI}}$.

Figure B.33 : PowerQUICC IDMA Read (Dual Address)



Note: A[31:0] is depicted for completeness. It is not examined by the QSpan during IDMA transfers; however, it can be used to drive $\overline{\text{CSPCI}}$.

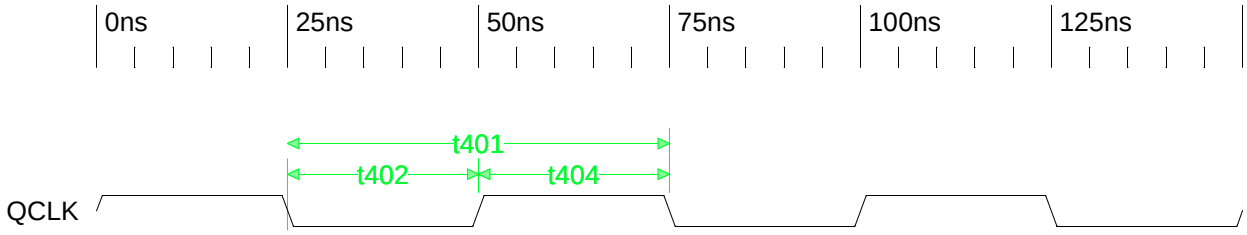
Figure B.34 : PowerQUICC IDMA Write (Dual Address)



Note: The QSpan does not issue retries or bus errors during dual address IDMA cycles

B.3.3 QBus (M68040) Interface

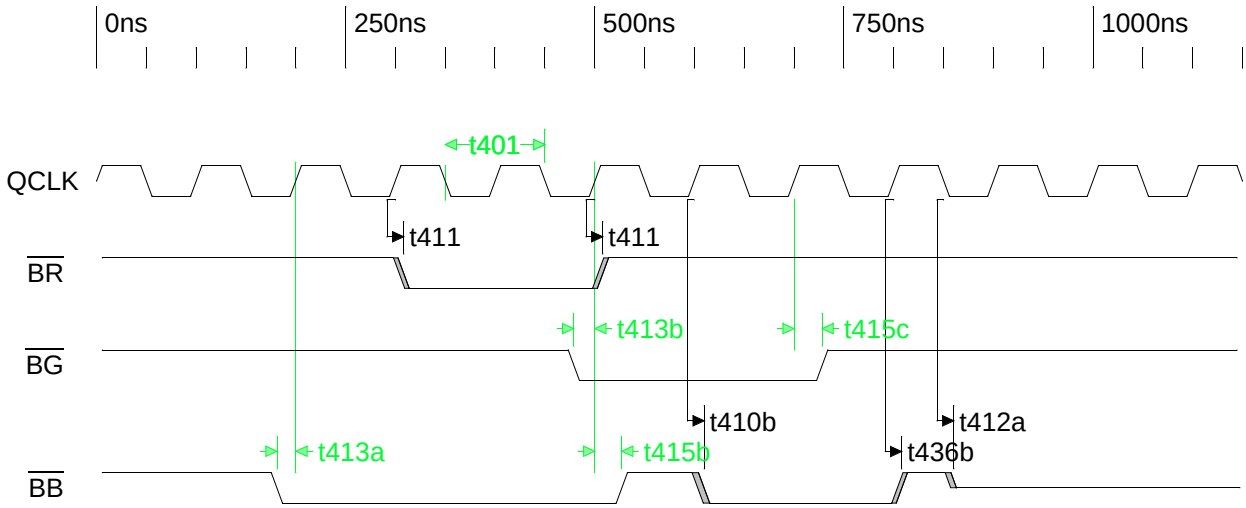
Figure B.35 : QCLK Input Timing—M68040 BCLK



Note: The timing parameters t405 and t406 are measured between 0.8 and 2 Volts. The timing parameters t405 and t406 can be found in Table B.4

B.3.3.1 QBus (M68040) Master Cycles

Figure B.36 : QBus (M68040) Arbitration



Note: This figure depicts timing in the case where the QSpan requests ownership of the QBus while another QBus master currently owns the bus ($\overline{BB}$/$\overline{BGACK}$ asserted by the other master). The QSpan obtains ownership of the bus after the other master negates $\overline{BB}$/$\overline{BGACK}$.

Figure B.37 : Single Read - QSpan as M68040 Master

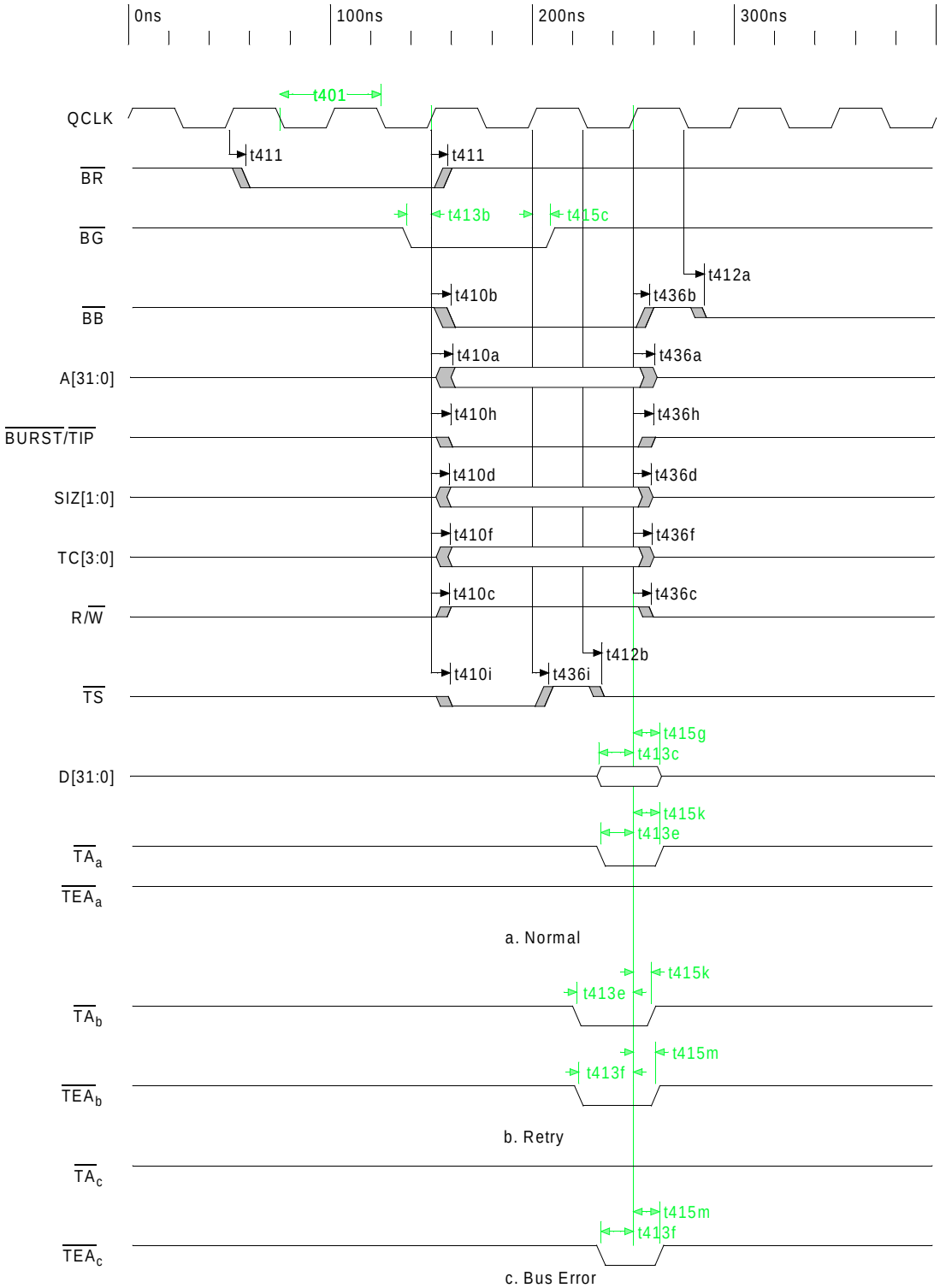
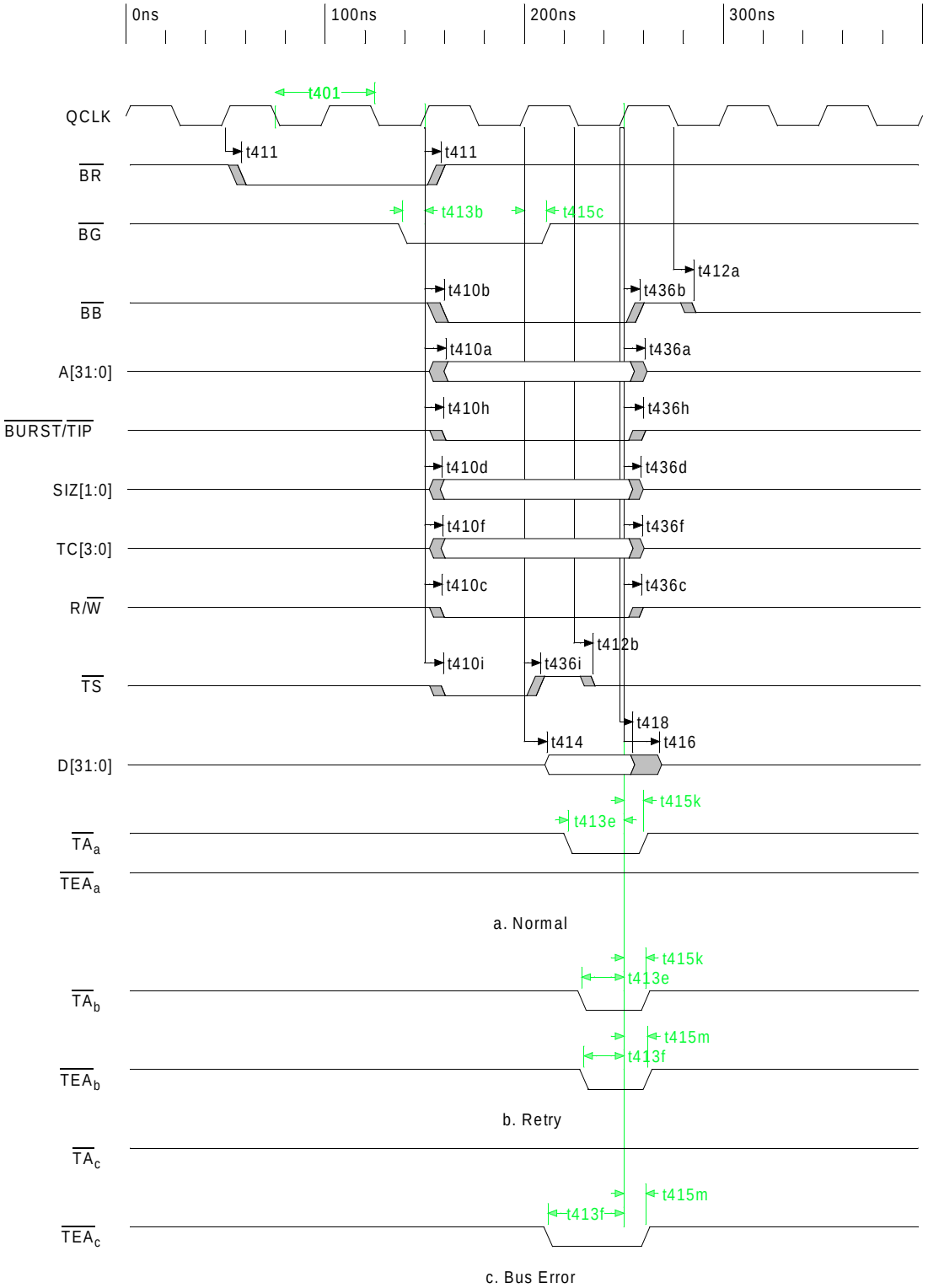


Figure B.38 : Single Write - QSpan as M68040 Master



B.3.3.2 QBus (M68040) Slave Cycles

Figure B.39 : Single Read - QSpan as M68040 Slave

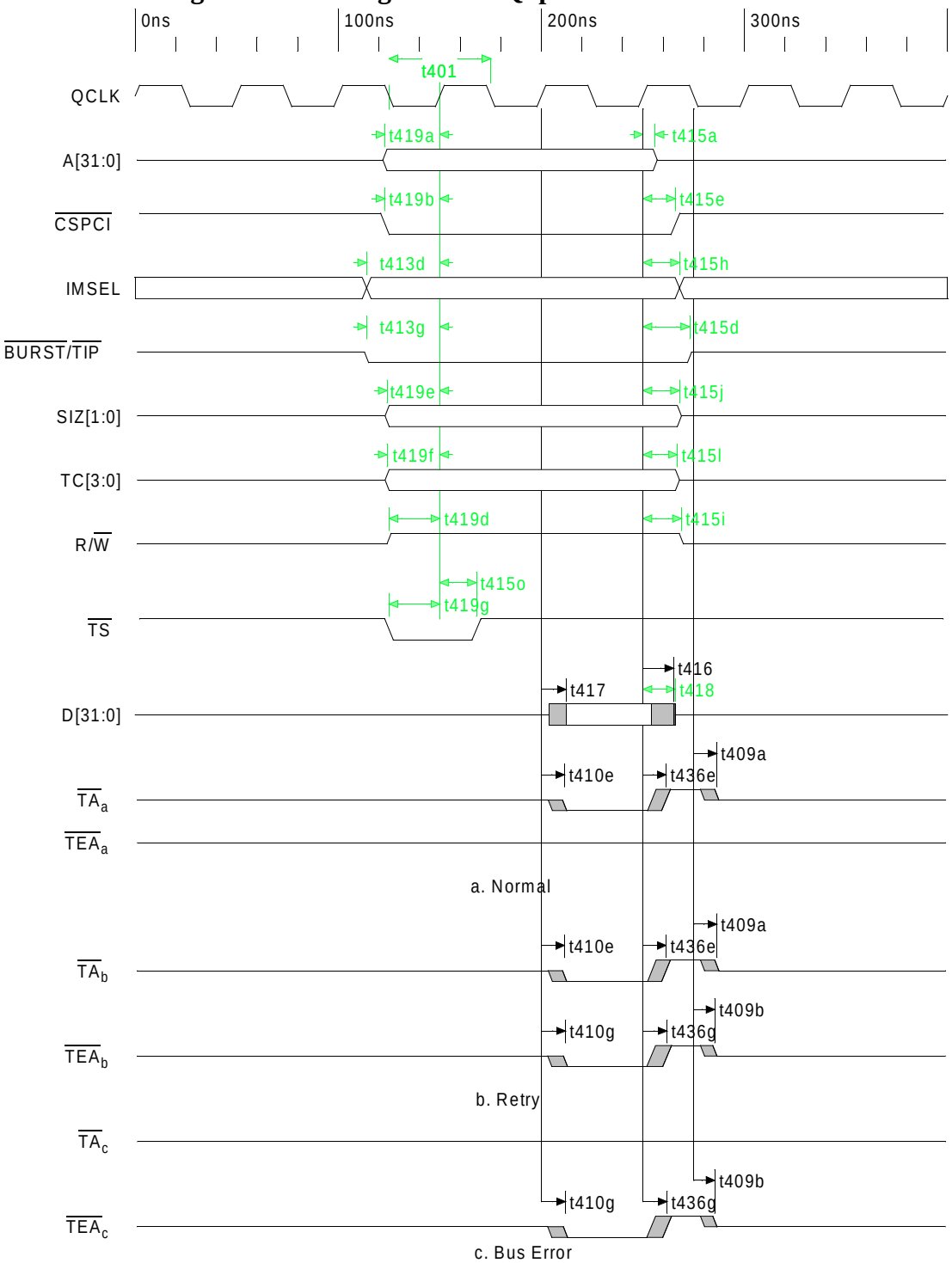


Figure B.40 : Single Write - QSpan as M68040 Slave

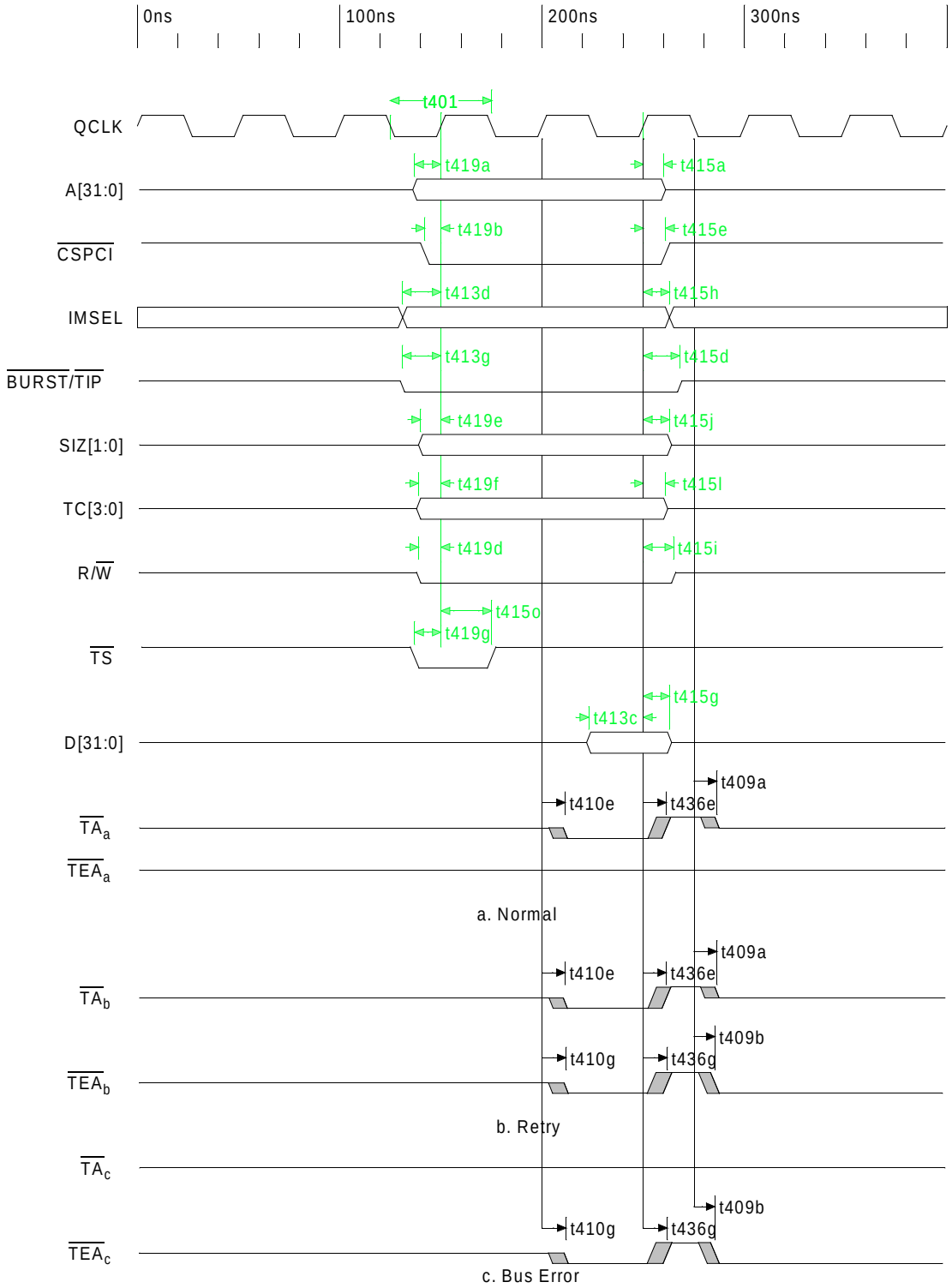
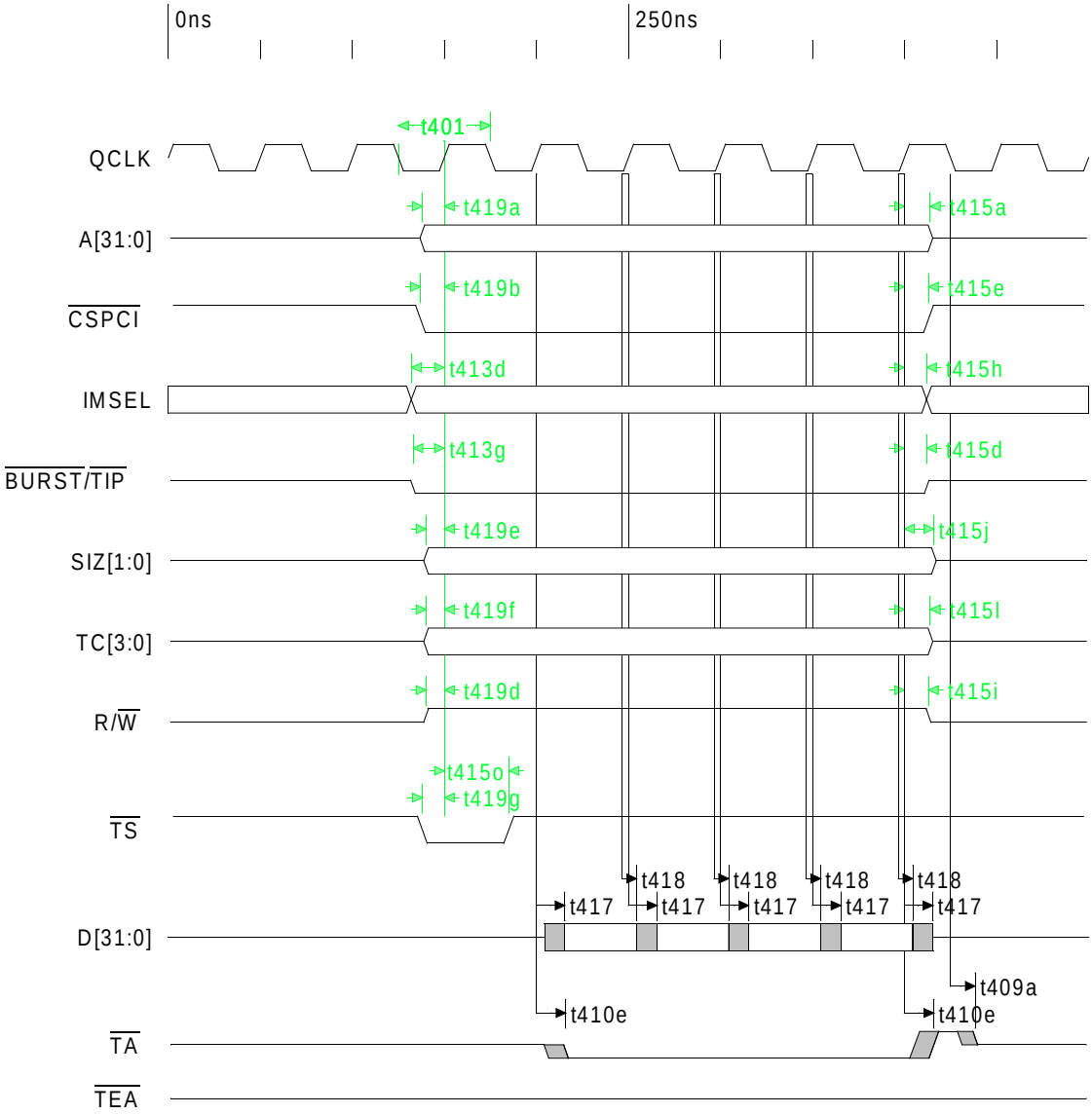
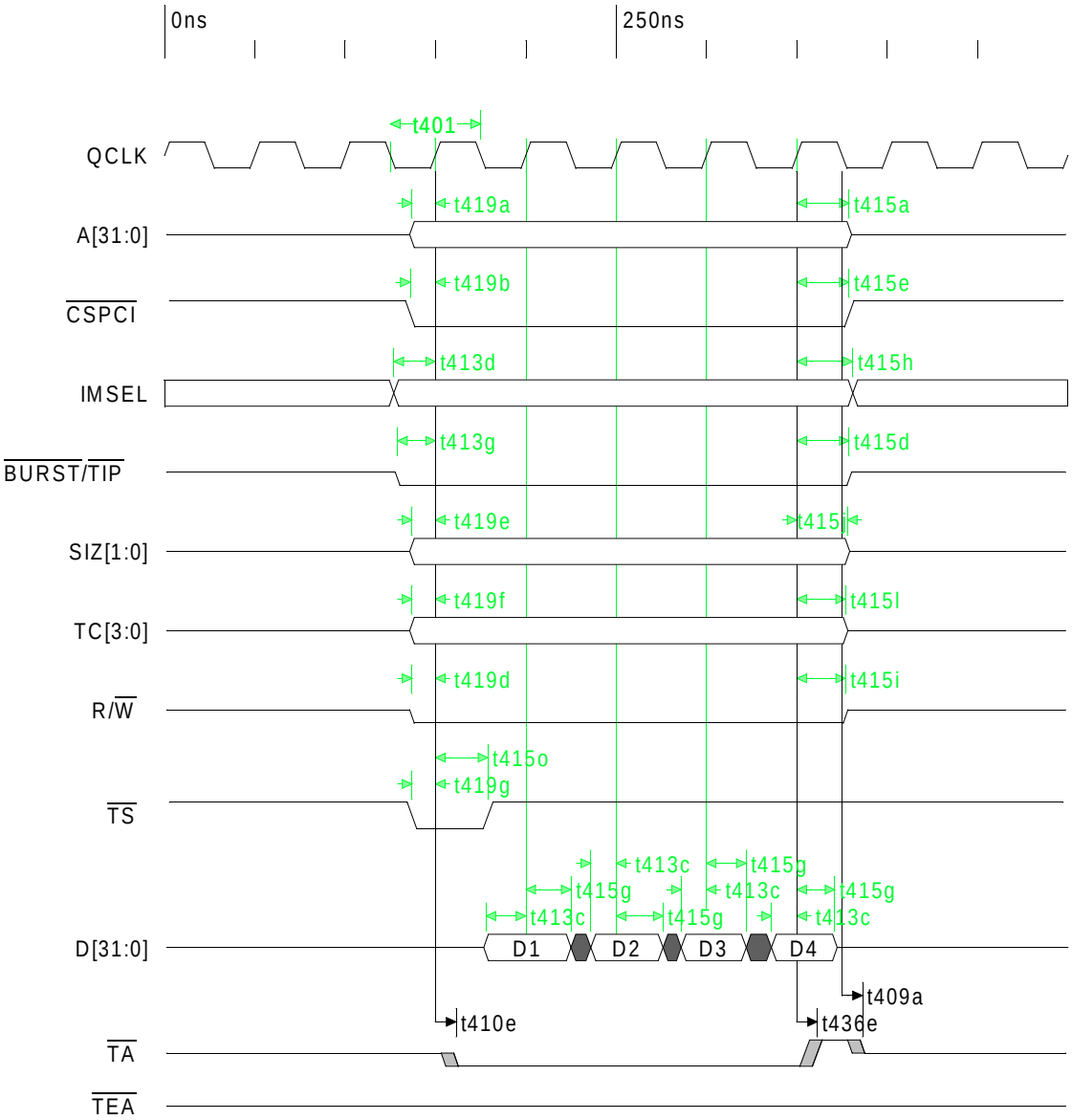


Figure B.41 : Delayed Burst Read—QSpan as M68040 Slave



Note: Wait states will be inserted if starting address is not aligned to 16-byte boundary.

Figure B.42 : Posted Burst Write—QSpan as M68040 Slave



Note: Wait states will be inserted if starting address is not aligned to 16-byte boundary.

Figure B.43 : Register Reads—QSpan as M68040 Slave

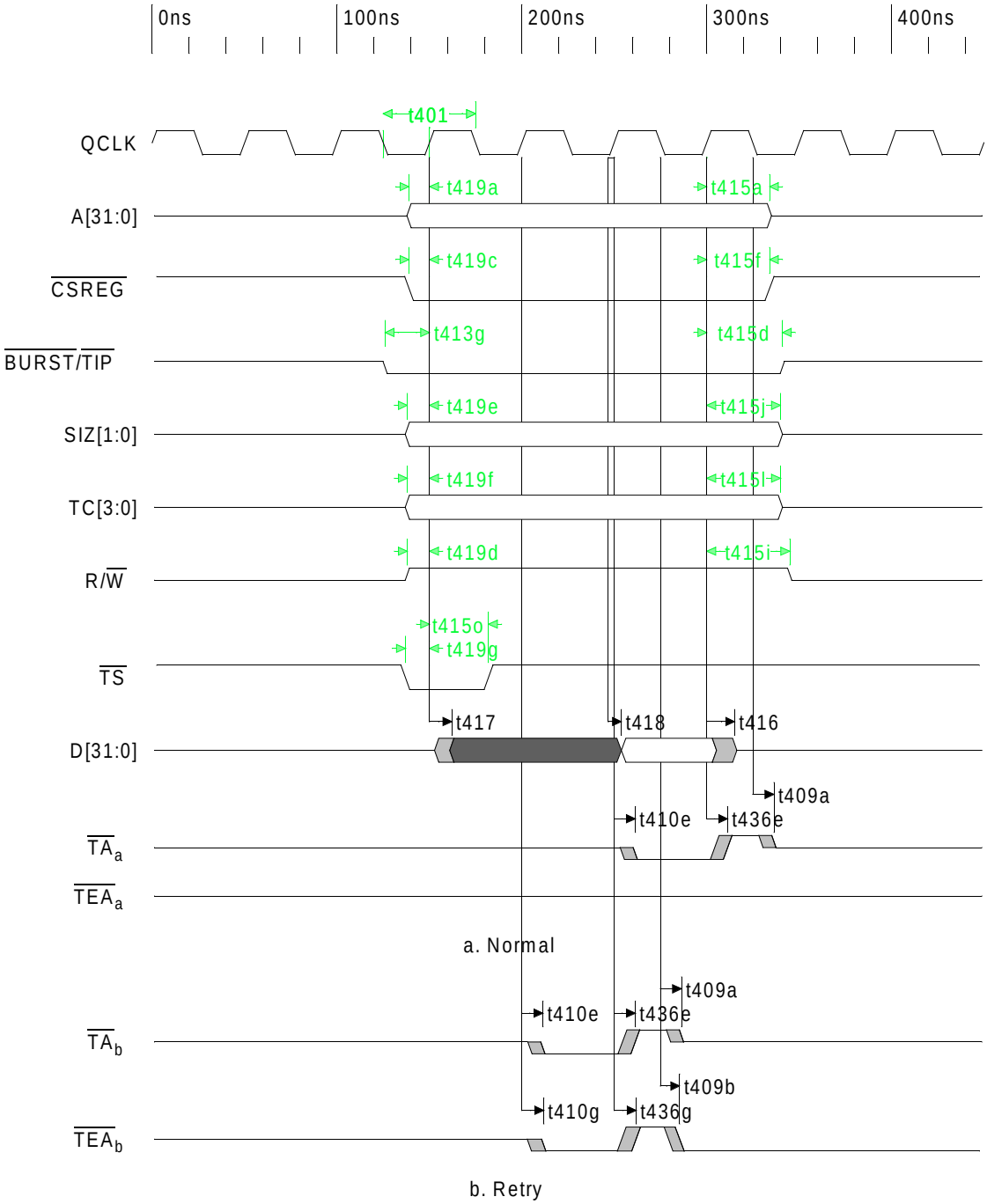
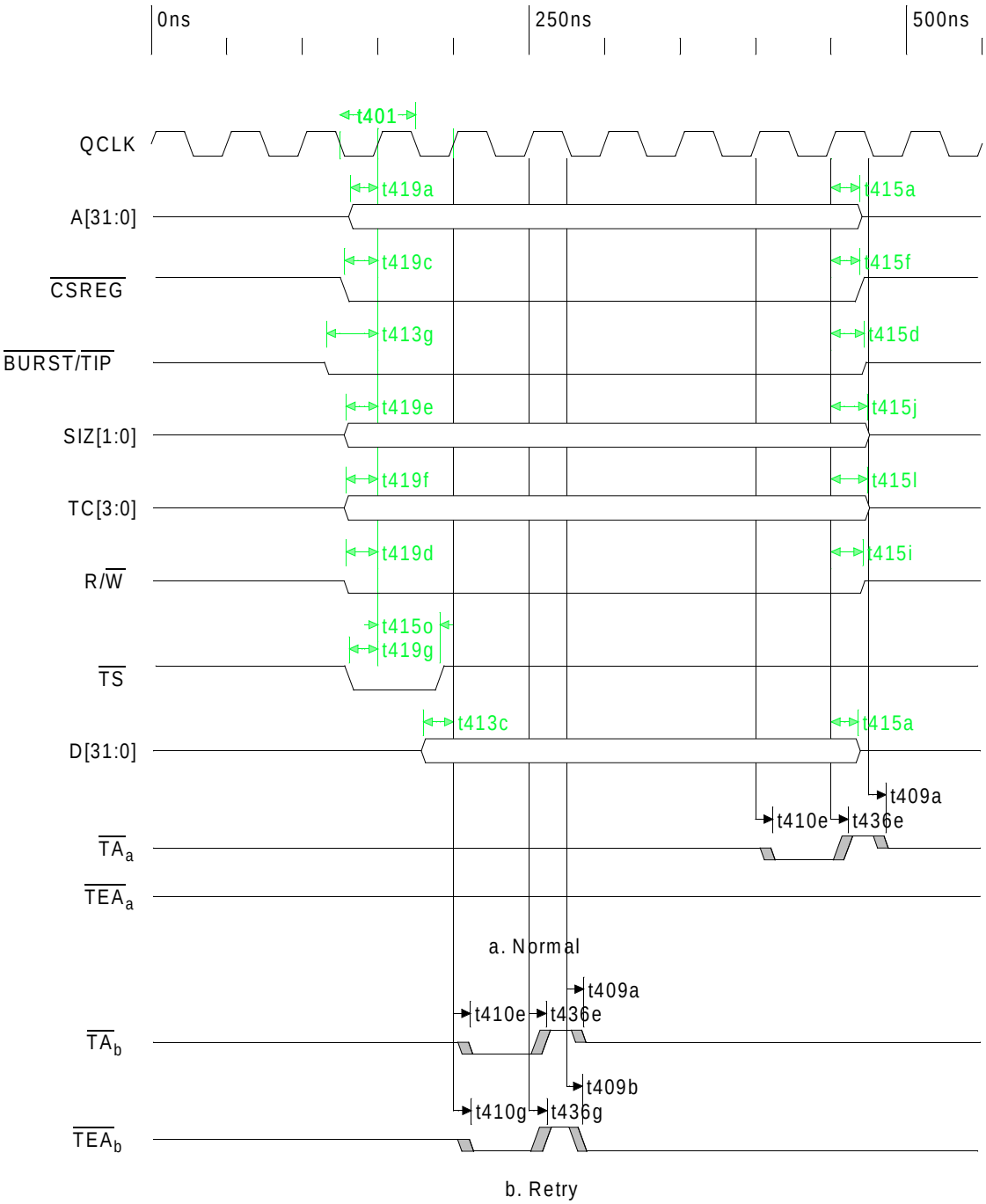


Figure B.44 : Register Write—QSpan as M68040 Slave



B.3.4 Utility Functions

Figure B.45 : Reset from PCI Interface (RST# related to $\overline{\text{RESETO}}$)

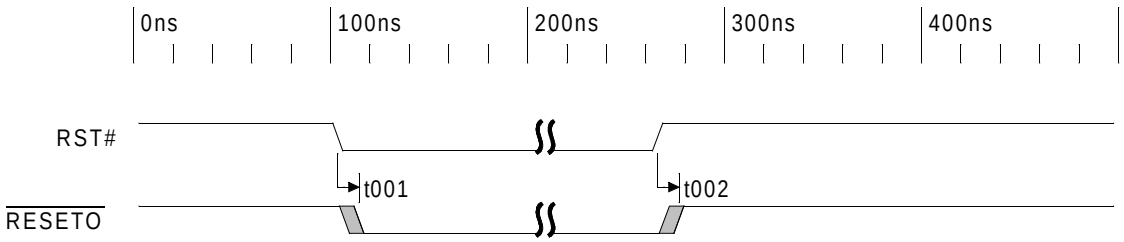


Figure B.46 : Software Reset

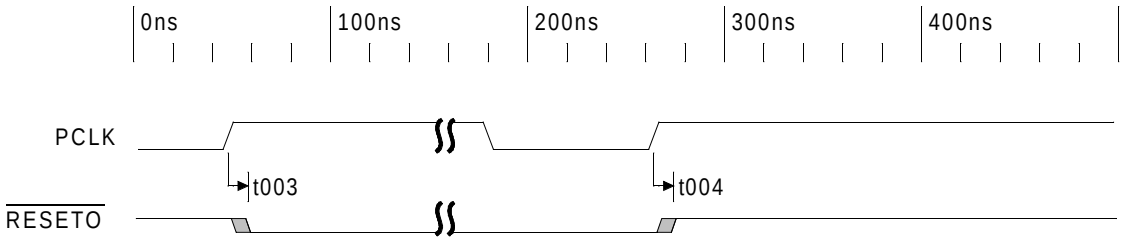


Figure B.47 : INT# Interrupt to $\overline{\text{QINT}}$ Interrupt

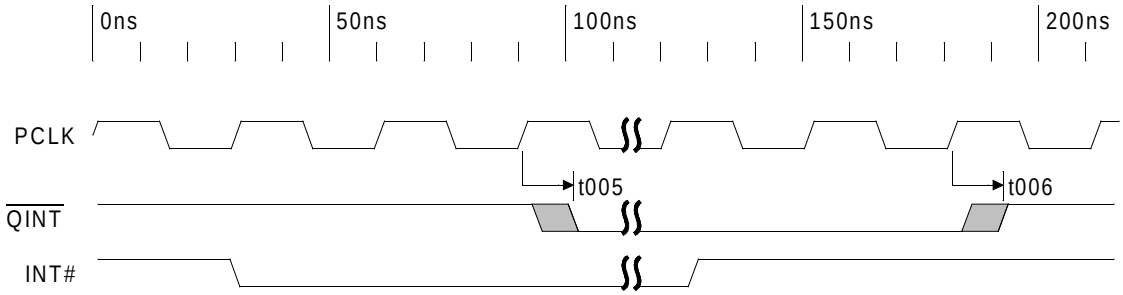


Figure B.48 : PERR# Interrupt

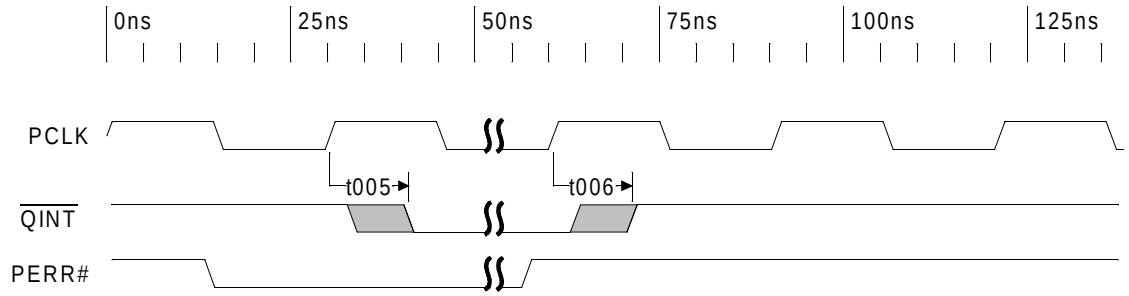


Figure B.49 : SERR# Interrupt

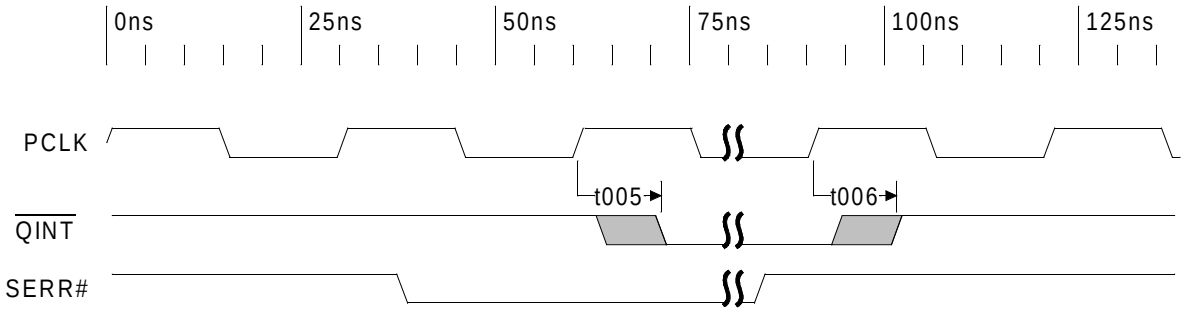
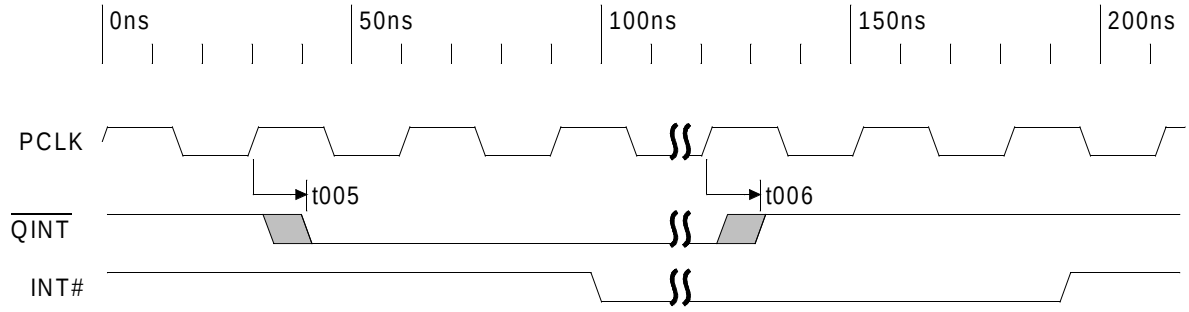
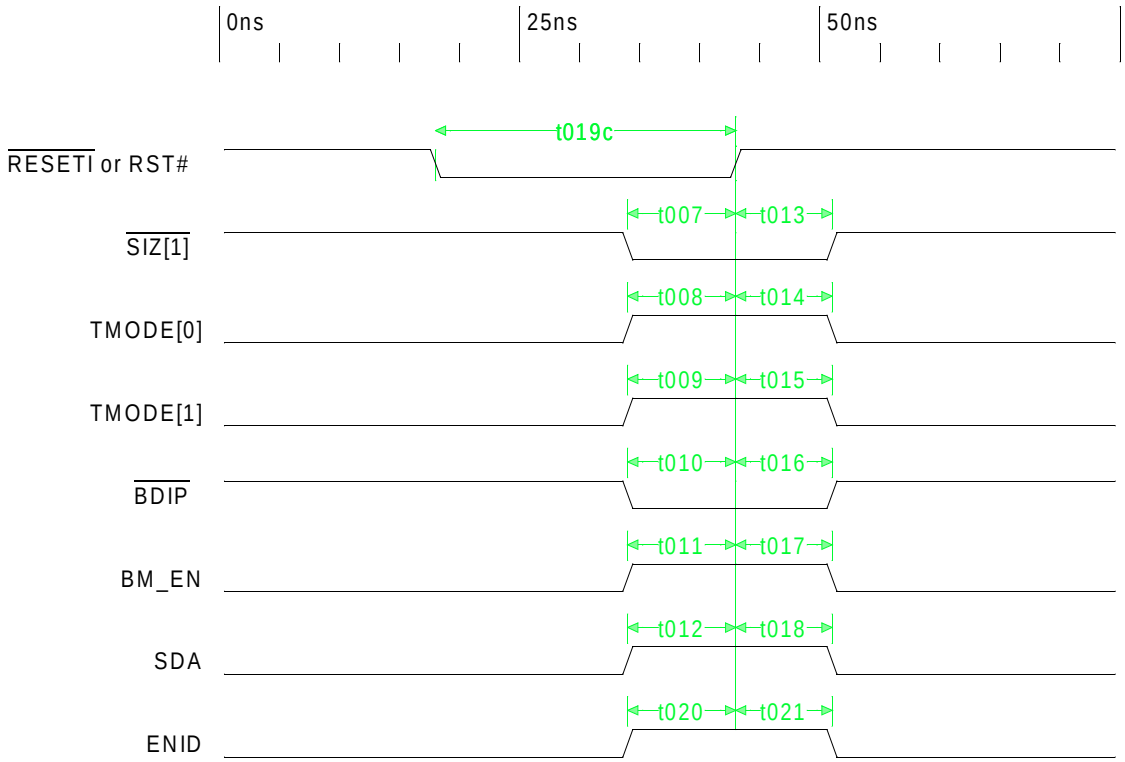


Figure B.50 : QINT Interrupt to INT# Interrupt



B.3.5 Reset Options

The parameters for the following figure are in Table B.6 on page B-18.



Appendix C Typical Applications

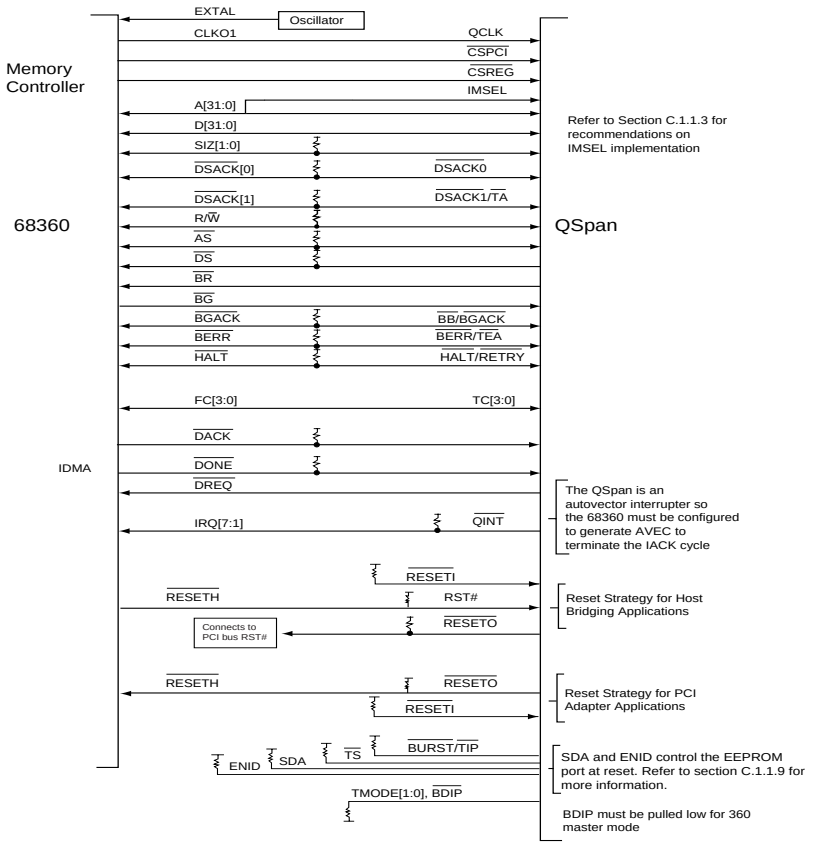
The QSpan provides a direct connection to the QUICC, PowerQUICC, and M68040 Motorola buses. No glue is required for these applications. This appendix describes how to connect the QSpan to these chips.

Tundra supports clients with Reference Design Kits (RDKs). Please check our website - www.tundra.com - for the latest downloads and information.

C.1 QUICC Interface

This section describes how the QSpan can be connected to the QUICC communications controller.

Figure C.1 : QUICC Interface



C.1.1 Hardware Interface

C.1.1.1 Clocking

The QSpan must be clocked by the CLKO1 output from the QUICC processor. All of the AC timing waveforms for the QSpan/QUICC interface are based on CLKO1. The QSpan requires both QCLK and PCLK to be operating at least five clock cycles before either reset, RST# or RESETI, negates. Based on this constraint, the QUICC's clock mode select pins, MODCK1-MODCK0, should be set to a 0b10 state at power-up.

C.1.1.2 Resets

Two reset scenarios exist depending on the use of the QSpan in your application; PCI host bridge or PCI adapter card bridge.

For a PCI Adapter card application the following reset configuration should be followed: connect the QSpan's reset output (RESETO) to the hard reset input (RESETH) on the QUICC. This enables the QSpan to reset the QUICC when PCI RST# is asserted or when the software reset bit is asserted (SW_RST bit in the MISC_CTL register, page A-46). The QSpan's reset input (RESETI) is typically unused and should be pulled high through a resistor.

For a PCI Host bridge application the following reset configuration should be followed: The QSpan's PCI RST# (global reset for the QSpan) input is connected to the hardware reset (RESETH) signal on the QUICC. The QSpan's reset output (RESETO) can be connected to the PCI RST# inputs of the other PCI devices (the QSpan's RST# input is not connected to the PCI bus RST# signal). Therefore, at power-up the QUICC's power-on-reset circuitry will assert RESETH which fully resets the QSpan device. The QSpan will, in turn, assert RESETO to reset the agents on the PCI bus when RST# is active. This reset scenario also allows the QUICC processor to reset all of the PCI agents under software control. The QUICC can write to the software reset bit in the QSpan's MISC_CTL register which will cause the QSpan to assert its RESETO signal. The QSpan's reset input (RESETI) is typically unused and should be pulled high through a resistor. Resets are described in "Reset Options" on page 2-75.

C.1.1.3 Memory Controller

The QSpan requires that two chip selects be generated in order to access the registers (CSREG) and the PCI bus (CSPCI). This can be accomplished by using two of the chip select outputs from the memory controller within the QUICC. There are two QBus slave images within the QSpan which are used to access the PCI bus. The image that is selected when the PCI chip select is asserted is dependent on the state of the Image Select signal (IMSEL). If IMSEL is low then QBus slave image 0 is selected; otherwise, QBus slave image 1 is selected. IMSEL is typically generated directly from one of the high order address lines on the QBus (i.e. dependent on the processor's memory map). An alternative method, is to use a spare I/O port pin on the QUICC processor to control the QSpan's IMSEL input pin. When the opposite QBus slave image is desired to be accessed the QUICC would first perform a write to change the state of this I/O port pin.

C.1.1.4 Qbus Direct Connects

All other QBus interface signals can be directly connected to the appropriate QUICC signal. External pull-up resistors should be connected to all bus control signals (excluding SIZ[1] and $\overline{\text{BDIP}}$ which are reset options and should be pulled to the desired state) to ensure that they are held in the inactive state. Refer to Figure C.1 to determine which signals require external pull-ups.

C.1.1.5 Interrupts

The QSpan device is now able to pass interrupts between the PCI bus and the QBus. For host bridging applications, the QSpan is able to accept INT# as an input and assert $\overline{\text{QINT}}$ as an output. The QSpan's interrupt output ($\overline{\text{QINT}}$) should be connected to one of the seven possible interrupt inputs (IRQ[7:1]) on the QUICC processor. When the QUICC is acknowledging a QSpan interrupt it must be programmed to generate the cycle termination. The QSpan is an autovector interrupter and does not have the ability to assert $\overline{\text{AVEC}}$ during the interrupt acknowledge cycle.

For PCI adapter card applications, the QSpan is able to accept interrupts from the QBus on the $\overline{\text{QINT}}$ pin and pass them through the QSpan to its PCI INT# output. Interrupts are described in “The Interrupt Channel” on page 2-64

C.1.1.6 PCI Signals

The QSpan’s PCI signals can be directly connected to the appropriate PCI signal on the motherboard or the PCI connector. Pull-up resistors may be required to be added to the PCI bus control signals depending on the application. If you are designing a local PCI bus on a motherboard then pull-up resistors will be required (refer to the PCI 2.1 Specification for additional detail). For host bridging applications, the QSpan’s IDSEL signal is not required to be implemented and therefore should be pulled low through a resistor.

The QSpan BGA device supports both 5V and 3.3V I/O signalling environments. VIO may be connected to either a 5V or a 3.3V “I/O” designated power pin on the PCI connector. Alternatively, the QSpan’s VIO pin could be directly connected to either the 5V or 3.3V supplies to select the I/O signalling environment.

VIO is not a power pin, but is used to qualify the PCI signalling environment. If used in a mixed environment where the VIO pin is attached to the 5V power rail, the user must ensure the 3.3V power ramp occurs before the 5V power ramp and the current specification for the VIO pin is not exceeded (see Table 4.3). A current limiting register on the VIO pin may be used to meet this specification (i.e. VIO pulled to 3.3V or 5V through at least a 470 ohm resistor). Alternatively, VIO may be driven to the appropriate logic level by a spare on-board output.

C.1.1.7 EEPROM Interface

A serial EEPROM is only required for applications which must support a plug and play environment. Refer to “Reset Options” on page C-4 for EEPROM reset options.

C.1.1.8 Reset Options

A number of reset options exist with the QSpan device. The following signals are sampled on the rising edge of both RST# and $\overline{\text{RESETI}}$ to determine the QSpan’s mode of operation.

- The $\overline{\text{BDIP}}$ signal must be pulled low at reset to enable the QSpan to perform as a QUICC master. The QSpan always responds to QUICC-style slave cycles independent of the state of the SIZ[1] signal at reset. (See “Reset Options” on page 2-75).
- The SDA and ENID (for new designs with the QSpan (CA91C860B, CA91L860B) signals should be pulled high if the EEPROM is being implemented. The ENID signal was added as an EEPROM reset option in the QSpan CA91C860B and CA91L860B devices to alleviate an errata in the QSpan CA91C860. The SDA signal should be pulled low if the serial EEPROM is not being implemented in this design. The ENID signal can be left open if the serial EEPROM is not being implemented as there is an internal weak pull-down resistor.
- The TMODE[1:0] signals should be pulled low through resistors if the NAND TREE test feature inside the QSpan is used during the board manufacturing process. Pulling the TMODE signals low allows an in-circuit tester to overdrive these inputs during the manufacturing testing. If an in-circuit tester will not be used to verify the board's functionality then TMODE[1:0] could be tied directly to ground which will select the normal mode of operation.
- If BM_EN/$\overline{\text{FIFO\_RDY}}$ is sampled at a logic high (“1”) while RST# is asserted, the QSpan will set the BM bit in the PCI_CS register (page A-5). This enables the QSpan as a PCI bus master. This pin can be left as a no-connect as there is an internal weak pull-down resistor.

Reset options are described in “Reset Options” on page 2-75.

C.1.1.9 Unused Inputs Requiring Pull-Ups

The $\overline{TS}$ and $\overline{BURST}$/$\overline{TIP}$ signals are unused inputs when the QSpan is being interfaced with a QUICC and therefore must be pulled high.

C.1.1.10 No Connects

The BM_EN/$\overline{FIFO\_RDY}$ can be left as a no-connect as there is an internal weak pull-down resistor. In this case, in order for the QSpan to become a PCI bus master a write to the PCI_CS register is required.

C.1.1.11 JTAG Signals

The QSpan BGA device supports JTAG. The QSpan's JTAG signals should be connected to the JTAG controller or to the JTAG signals of another device if devices are to be chained together. If JTAG will not be supported then the JTAG signals may be left open as the inputs have internal pull-up resistors.

C.1.1.12 Address Multiplexing for DRAM

Whenever DRAM is used on the QBus, external address multiplexing is required. This is described in Motorola's User Manual - MC68360 UM/AD.

C.1.2 Software Issues

When using the QUICC with the QSpan there are register bits which should be altered from the QUICC's default reset state. They should be set as follows:

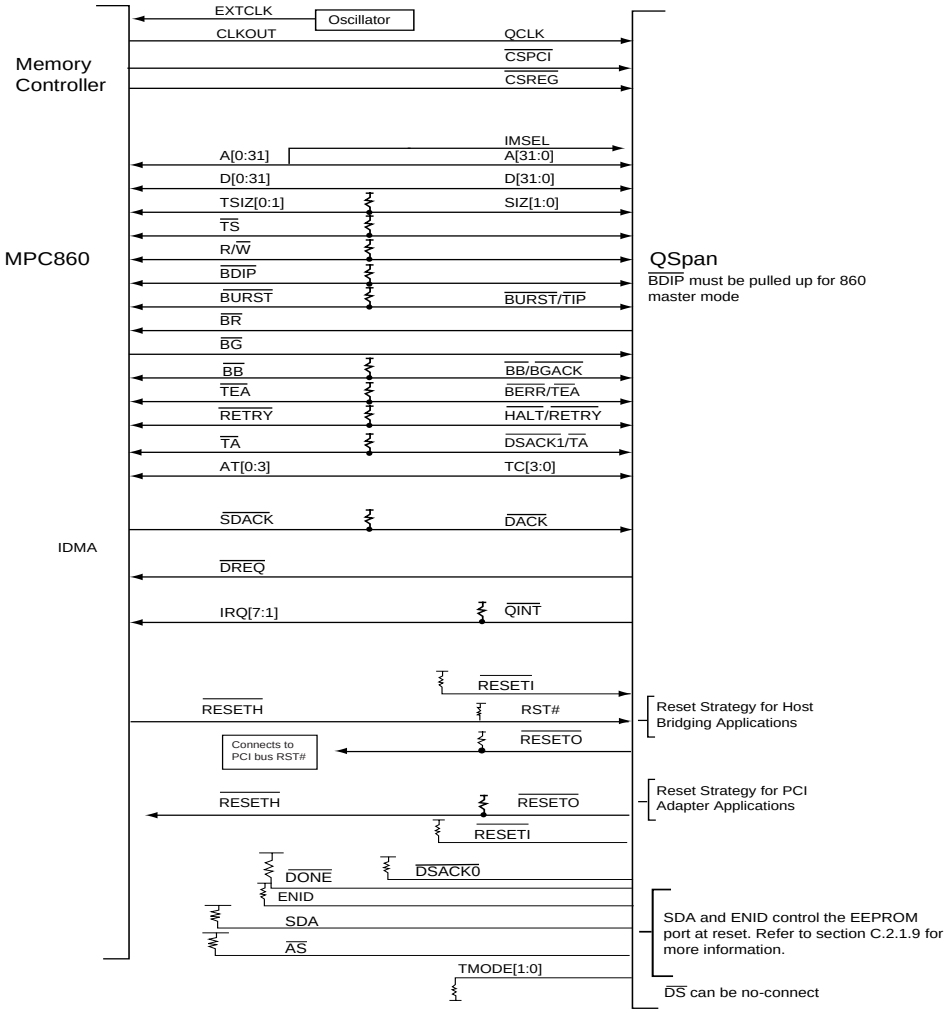
- the BSTM bit in the MCR register should be set to "1" because the QSpan is a synchronous bus master
- the ASTM bit in the MCR register should be set to "0" because the QSpan is not able to meet the QUICC's set-up requirements

The QSpan's S_BG and S_BB bits must be set to "1' when using the QUICC's internal arbiter. This is a requirement for the QSpan (CA91C860B, CA91B860B) device only. The QSpan synchronously samples the QUICC's arbitration inputs and the QUICC asynchronously samples the QSpan's arbitration inputs.

C.2 PowerQUICC Interface

This section describes how the QSpan can be connected to the PowerQUICC communications controller (and other MPC8xx devices).

Figure C.2 : PowerQUICC Interface



Tundra supports clients with Reference Design Kits (RDKs). Please check our website - www.tundra.com - for the latest downloads and information.

C.2.1 Hardware Interface

C.2.1.1 Clocking

The QSpan should be clocked by the CLKOUT output from the PowerQUICC processor. All of the AC timing waveforms for the QSpan are based on this clock output. However, it is recommended to buffer the CLKOUT signal to the QSpan with a low skew PLL based clock buffer because of the low drive strength of the CLKOUT signal.

If an external PLL clock buffer is used to generate the QSpan's QCLK input, care must be taken to ensure the QSpan's set-up and hold times are not violated while the PLL clock buffer is locking.

C.2.1.2 Resets

Two reset scenarios exist depending on the use of the QSpan in your application; PCI host bridge or PCI adapter card bridge.

For a PCI Adapter card application the following reset configuration should be followed: connect the QSpan's reset output ($\overline{\text{RESETO}}$) to the hard reset input ($\overline{\text{RESETH}}$) on the PowerQUICC. This enables the QSpan to reset the PowerQUICC when PCI RST# is asserted or when the software reset bit is asserted (SW_RST bit in the MISC_CTL register, page A-46). The QSpan's reset input ($\overline{\text{RESETI}}$) is typically unused and should be pulled high through a resistor.

For a PCI Host bridge application the following reset configuration should be followed: The QSpan's PCI RST# (global reset for the QSpan) input is connected to the hardware reset ($\overline{\text{RESETH}}$) on the PowerQUICC. The QSpan's reset output ($\overline{\text{RESETO}}$) can be connected to the PCI RST# inputs of the other PCI devices (the QSpan's RST# input is not connected to the PCI bus RST# signal). Therefore, at power-up the PowerQUICC's power-on-reset circuitry will assert $\overline{\text{RESETH}}$ which fully resets the QSpan device. The QSpan will, in turn, assert $\overline{\text{RESETO}}$ to reset the agents on the PCI bus when RST# is active. This reset scenario also allows the PowerQUICC processor to reset all of the PCI agents under software control. The PowerQUICC can write to the software reset bit in the QSpan's MISC_CTL register which will cause the QSpan to assert its $\overline{\text{RESETO}}$ signal. The QSpan's reset input ($\overline{\text{RESETI}}$) is typically unused and should be pulled high through a resistor. Resets are described in "Reset Options" on page 2-75.

C.2.1.3 Memory Controller

The QSpan requires that two chip selects be generated in order to access the registers ($\overline{\text{CSREG}}$) and the PCI bus ($\overline{\text{CSPCI}}$). This can be accomplished by using two of the chip select outputs from the memory controller within the PowerQUICC. There are two QBus slave images within the QSpan which are used to access the PCI bus. The image that is selected when the PCI chip select is asserted is dependent on the state of the Image Select signal (IMSEL). If IMSEL is low then QBus slave image 0 is selected; otherwise, QBus slave image 1 is selected. IMSEL is typically generated directly from one of the high order address lines on the QBus (i.e. dependent on the processor's memory map). An alternative method, is to use a spare I/O port pin on the PowerQUICC processor to control the QSpan's IMSEL input pin. When the opposite QBus slave image is desired to be accessed the PowerQUICC would first perform a write to change the state of this I/O port pin.

C.2.1.4 QBus Direct Connects

The bus interface signals can be directly connected together. External pull-up resistors should be connected to all bus control signals (including SIZ[1] and $\overline{\text{BDIP}}$ which are reset options and should be pulled high to support PowerQUICC mode) to ensure that they are held in the inactive state. Refer to Figure C.2 to determine which signals require external pull-ups.

C.2.1.5 Interrupts

The QSpan device is now able to pass interrupts between the PCI bus and the QBus. For host bridging applications, the QSpan is able to accept INT# as an input and assert $\overline{\text{QINT}}$ as an output. The QSpan's interrupt output ($\overline{\text{QINT}}$) should be connected to one of the seven possible interrupt inputs (IRQ[7:1]) on the PowerQUICC processor.

For PCI adapter card applications, the QSpan is able to accept interrupts from the QBus on the $\overline{\text{QINT}}$ pin and pass them through the QSpan to its PCI INT# output. Interrupts are described in "The Interrupt Channel" on page 2-64.

C.2.1.6 PCI Signals

The QSpan's PCI signals can be directly connected to the appropriate PCI signal on the motherboard or the PCI connector. Pull-up resistors may be required to be added to the PCI bus control signals depending on the application. If you are designing a local PCI bus on a motherboard then pull-up resistors will be required (refer to the PCI 2.1 Specification for additional detail). For host bridging applications, the QSpan's IDSEL signal is not required to be implemented and therefore should be pulled low through a resistor.

The QSpan BGA device supports both 5V and 3.3V I/O signalling environments. VIO may be connected to the either a 5V or a 3.3V "I/O" designated power pin on the PCI connector. Alternatively, the QSpan's VIO pin could be directly connected to either the 5V or 3.3V supplies to select the I/O signalling environment.

VIO is not a power pin, but is used to qualify the PCI signalling environment. If used in a mixed environment where the VIO pin is attached to the 5V power rail, the user must ensure the 3.3V power ramp occurs before the 5V power ramp and the current specification for the VIO pin is not exceeded (see Table 4.3). A current limiting register on the VIO pin may be used to meet this specification (i.e. VIO pulled to 3.3V or 5V through at least a 470 ohm resistor). Alternatively, VIO may be driven to the appropriate logic level by a spare on-board output.

C.2.1.7 EEPROM Interface

A serial EEPROM is only required for applications which must support a plug and play environment. Refer to “Reset Options” on page C-4 for EEPROM reset options.

C.2.1.8 Reset Options

A number of reset options exist with the QSpan device. The following signals are sampled on the rising edge of both RST# and $\overline{\text{RESETI}}$ to determine the QSpan’s mode of operation.

- The $\overline{\text{BDIP}}$ and SIZ[1] signals must be pulled high at reset to enable the QSpan to perform as a PowerQUICC master and slave. (See “Reset Options” on page 2-75).
- The SDA and ENID (for new designs with the QSpan (CA91C860B, CA91L860B)) signals should be pulled high if the EEPROM is being implemented. The ENID signal was added as an EEPROM reset option in the QSpan CA91C860B and CA91L860B devices to alleviate an errata in the QSpan CA91C860 device. The SDA signal should be pulled low if the serial EEPROM is not being implemented in this design. The ENID signal can be left open if the serial EEPROM is not being implemented as there is an internal weak pull-down resistor.
- The TMODE[1:0] signals should be pulled low through resistors if the NAND TREE test feature inside the QSpan is used during the board manufacturing process. Pulling the TMODE signals low allows an in-circuit tester to overdrive these inputs during the manufacturing testing. If an in-circuit tester will not be used to verify the board's functionality then TMODE[1:0] could be tied directly to ground which will select the normal mode of operation.
- If BM_EN/$\overline{\text{FIFO\_RDY}}$ is sampled at a logic high (“1”) while RST# is asserted, the QSpan will set the BM bit in the PCI_CS register (page A-5). This enables the QSpan as a PCI bus master. This pin can be left as a no-connect as there is an internal weak pull-down resistor.

Reset options are described in “Reset Options” on page 2-75.

C.2.1.9 Unused Inputs Requiring Pull-Ups

The $\overline{AS}$, $\overline{DSACK0}$ and $\overline{DONE}$ signals are unused inputs when the QSpan is being interfaced with a PowerQUICC and therefore must be pulled high.

C.2.1.10 No Connects

The $\overline{DS}$ output from the QSpan should be left as a no connect when the QSpan is being interfaced with a PowerQUICC.

The BM_EN/$\overline{FIFO\_RDY}$ can be left as a no-connect as there is an internal weak pull-down resistor. In this case, in order for the QSpan to become a PCI bus master a write to the PCI_CS register is required.

C.2.1.11 JTAG Signals

The QSpan BGA device supports JTAG. The QSpan's JTAG signals should be connected to the JTAG controller or to the JTAG signals of another device if devices are to be chained together. If JTAG will not be supported then the JTAG signals may be left open as the inputs have internal pull-up resistors.

C.2.1.12 Bused Signals

This manual adopts the convention that the most significant bit (address, data, size and transaction codes) is always the largest number. When interfacing the PowerQUICC to the QSpan, designers must ensure that they connect the signals accordingly (e.g., pin A[31] on the QSpan connects to pin A[0] on the PowerQUICC).

C.2.1.13 Multiplexing for DRAM

Whenever DRAM is used on the QBus, external address multiplexing is required. This is described in the MPC860 UM/AD (REV1).

C.2.2 Software Issues

When using the MPC860 with the QSpan there are a few register bits that must be altered from the MPC860's default reset state. They are described in detail below:

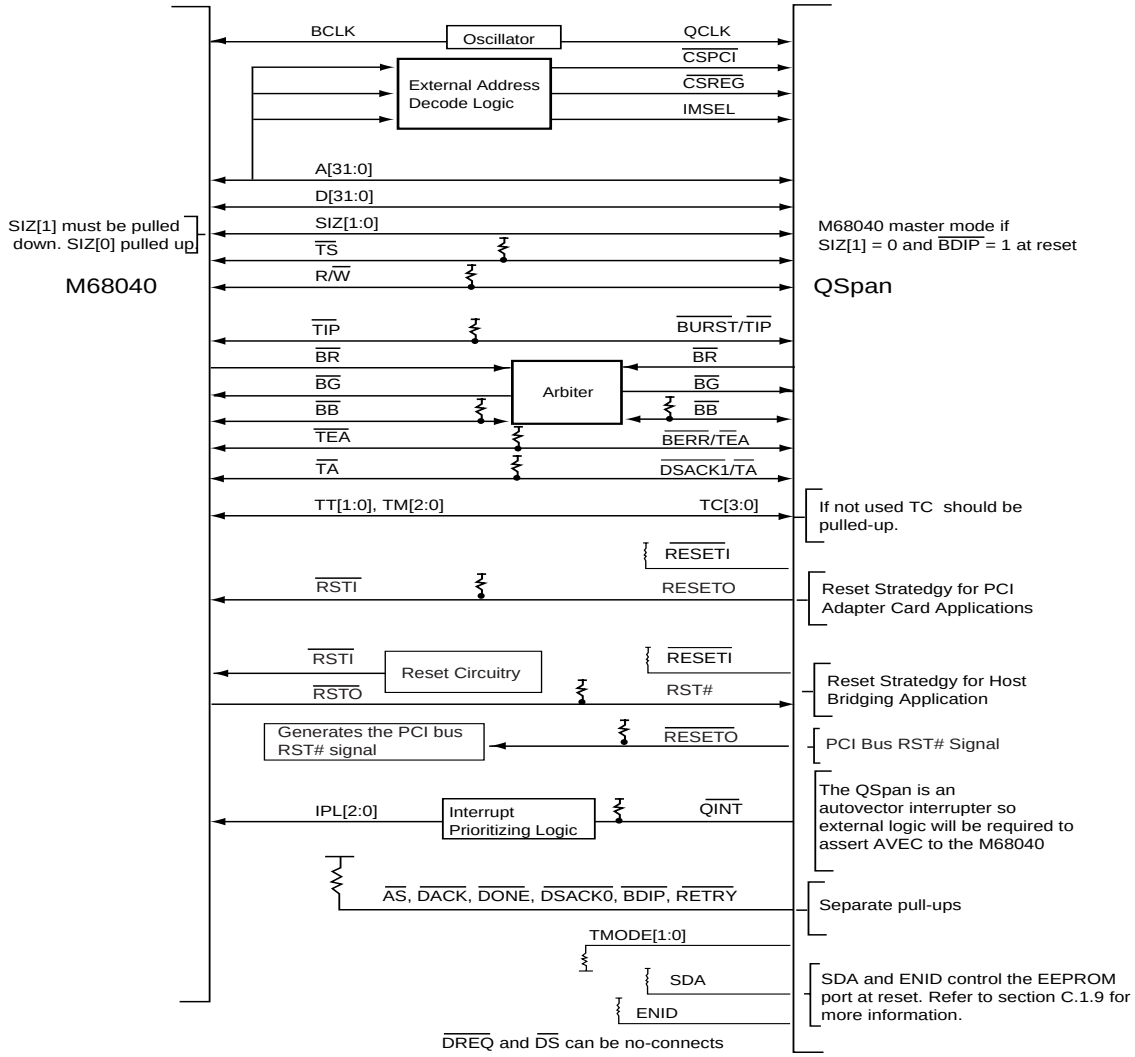
- The MLRC bits in the 860's SIUMCR register must be changed to "10" state. This configures the KR/RETRY/IRQ4/SPKROUT pin to function as a RETRY input.
- The SEME bit in the 860's SIUMCR register must be set to a "1" because the QSpan is a synchronous external master.
- The SETA bit in the 860's Option register must be set to a "1" for the two QSpan chip selects ($\overline{CSREG}$ and $\overline{CSPCI}$). The QSpan will always provide the cycle termination to the MPC860.

- The BIH bit in the 860's Option register must be set to a "0" for the QSpan's register chip select ($\overline{\text{CSREG}}$). The QSpan's PCI chip select ($\overline{\text{CSPCI}}$) pin does support burst accesses when using the MPC860's GPCM. Therefore, the BIH bit may be set to either state. The MPC860 User's manual mentions that the GPCM machine does not support burst accesses, however with the QSpan's architecture it will function correctly for the $\overline{\text{CSPCI}}$ chip select.

C.3 M68040 Interface

This section describes how the QSpan can be connected to the M68040.

Figure C.3 : M68040 Interface



C.3.1 Hardware Interface

The QSpan is compatible with all M68040 variants in large buffer mode up to 40MHz. The QSpan is compatible with all M68040 variants in small buffer mode up to 33MHz.

C.3.1.1 Clocking

The QSpan's QCLK input and the M68040's BCLK input should both be clocked from the same clock source. The AC timing waveforms for the QSpan are based on this assumption.

C.3.1.2 Resets

Two reset scenarios exist depending on the use of the QSpan in your application; PCI host bridge or PCI adapter card bridge.

For a PCI Adapter card application the following reset configuration should be followed: connect the QSpan's reset output ($\overline{\text{RESETO}}$) to the external reset logic to the reset input ($\overline{\text{RSTI}}$) on the M68040. This enables the QSpan to reset the M68040 processor when PCI RST# is asserted or when the software reset bit is asserted (SW_RST bit in the MISC_CTL register, page A-46). The QSpan's reset input ($\overline{\text{RESETI}}$) is typically unused and should be pulled high through a resistor.

For a PCI Host bridge application the following reset configuration should be followed: The QSpan's PCI RST# (global reset for the QSpan) input is connected to the reset output($\overline{\text{RSTO}}$) on the M68040. The QSpan's reset output ($\overline{\text{RESETO}}$) can be connected to the PCI RST# inputs of the other PCI devices (the QSpan's RST# input is not connected to the PCI bus RST# signal). Therefore, at power-up the M68040's power-on-reset circuitry will assert $\overline{\text{RSTO}}$ which fully resets the QSpan device. The QSpan will assert $\overline{\text{RESETO}}$ to reset the agents on the PCI bus when RST# is active. This reset scenario also allows the M68040 processor to reset all of the PCI agents under software control. The M68040 can write to the software reset bit in the QSpan's MISC_CTL register which will cause the QSpan to assert its $\overline{\text{RESETO}}$ signal. The QSpan's reset input ($\overline{\text{RESETI}}$) is typically unused and should be pulled high through a resistor. Resets are described in "Reset Options" on page 2-75.

C.3.1.3 Address Decoder

The QSpan requires two chip selects and an image select signal (IMSEL) to be generated in order to access the registers ($\overline{\text{CSREG}}$) and the PCI bus ($\overline{\text{CSPCI}}$). There is no internal memory controller within the M68040 and therefore an external address decoder must be implemented. The IMSEL signal determines which QBus slave image is accessed when $\overline{\text{CSPCI}}$ is asserted to the QSpan. If IMSEL is low then QBus slave image 0 is selected; otherwise QBus slave image 1 is selected. IMSEL is typically dependent on the processor's memory map and is generated directly from one of the high order address lines. An alternative method is to use a registered output which resides on the QBus. When the opposite slave image must be accessed the M68040 would first perform a write to change the state of this registered output.

C.3.1.4 QBus Direct Connects

All other bus interface signals can be directly connected together. External pull-up resistors should be connected to all bus control signals (excluding SIZ[1] and $\overline{\text{BDIP}}$ which are power-up options and should be pulled to the desired state) to ensure that they are held in the inactive state. Refer to Figure C.3 above to determine which signals require external pull-ups.

Depending on the version and speed of the M68040 device selected for a design an extra wait state may need to be inserted on the transfer start signal ($\overline{\text{TS}}$). This may be required because the external address decoding circuitry may not be able to generate the chip selects to the QSpan in a timely enough manner to meet the input setup requirements. An alternate solution may be to use large output buffer mode in the M68040 to eliminate the need for this wait state. The M68040's output propagation delays are much quicker in this mode and therefore there is more timing margin available for interfacing with the QSpan. However, if large output buffer mode is chosen, it must be used in an unterminated fashion as the QSpan's output drivers do not have the ability to drive a 50 ohm transmission line terminated at 2.5V.

The QSpan has four transaction code TC[3:0] signals which can be connected to any 4 of the 5 following signals on the M68040: TT[1:0] and TM[2:0].

C.3.1.5 Interrupts

The QSpan device is now able to pass interrupts between the PCI bus and the QBus. For host bridging applications, the QSpan is able to accept INT# as an input and assert $\overline{\text{QINT}}$ as an output. The QSpan's interrupt output ($\overline{\text{QINT}}$) should be connected to the interrupt prioritizing logic which is connected to the IPL[2:0] lines on the M68040 processor. When the M68040 is acknowledging a QSpan interrupt there must be external logic to terminate the cycle because the QSpan is an autovector interrupter which does not have the ability to assert $\overline{\text{AVEC}}$ or $\overline{\text{TA}}$ during the interrupt acknowledge cycle.

For PCI adapter card applications, the QSpan is able to accept interrupts from the QBus on the $\overline{\text{QINT}}$ pin and pass them through the QSpan to its PCI INT# output. Interrupts are described in “The Interrupt Channel” on page 2-64

C.3.1.6 PCI Signals

The QSpan’s PCI signals can be directly connected to the appropriate PCI signal on the motherboard or the PCI connector. Pull-up resistors may be required to be added to the PCI bus control signals depending on the application. If you are designing a local PCI bus on a motherboard then pull-up resistors will be required (refer to the PCI 2.1 Specification for additional detail). For host bridging applications, the QSpan’s IDSEL signal is not required to be implemented and therefore should be pulled low through a resistor.

The QSpan BGA device supports both 5V and 3.3V I/O signalling environments. VIO may be connected to the either a 5V or a 3.3V “I/O” designated power pin on the PCI connector. Alternatively, the QSpan’s VIO pin could be directly connected to either the 5V or 3.3V supplies to select the I/O signalling environment.

C.3.1.7 EEPROM Interface

A serial EEPROM is only required for applications which must support a plug and play environment. Refer to “Reset Options” on page 2-75 for EEPROM reset options.

C.3.1.8 Reset Options

A number of reset options exist with the QSpan device. The following signals are sampled on the rising edge of both RST# and RESETI to determine the QSpan’s mode of operation.

- The $\overline{\text{BDIP}}$ signal must be pulled high and SIZ[1] pulled low at reset to enable the QSpan to perform as a M68040 master. The SIZ[1] signal must be pulled low at reset in order for the QSpan to decode a M68040 cycle
- The SDA and ENID (for new designs with the QSpan 1.2) signals should be pulled high if the EEPROM is being implemented. The ENID signal was added as an EEPROM reset option in the QSpan 1.2 device to alleviate an errata in the QSpan 1.1. The SDA signal should be pulled low if the serial EEPROM is not being implemented in this design. The ENID signal can be left open if the serial EEPROM is not being implemented as there is an internal weak pull-down resistor.
- The TMODE[1:0] signals should be pulled low through resistors if the NAND TREE test feature inside the QSpan is used during the board manufacturing process. Pulling the TMODE signals low allows an in-circuit tester to overdrive these inputs during the manufacturing testing. If an in-circuit tester will not be used to verify the board's functionality then TMODE[1:0] could be tied directly to ground which will select the normal mode of operation.
- If BM_EN/$\overline{\text{FIFO\_RDY}}$ is sampled at a logic high (“1”) while RST# is asserted, the QSpan will set the BM bit in the PCI_CS register (page A-5). This enables the QSpan as a PCI bus master. This pin can be left as a no-connect as there is an internal weak pull-down resistor.

Reset options are described in “Reset Options” on page 2-75.

C.3.1.9 Unused Inputs Requiring Pull-Ups

The $\overline{\text{AS}}$, $\overline{\text{DSACK0}}$, $\overline{\text{HALT}}$/$\overline{\text{TRETRY}}$, $\overline{\text{DONE}}$ and $\overline{\text{DACK}}$ signals are unused inputs when the QSpan is being interfaced with a M68040 and therefore must be pulled high.

C.3.1.10 No Connects

The $\overline{DS}$ and $\overline{DREQ}$ outputs from the QSpan should be left as no connects when the QSpan is being interfaced with a M68040.

The BM_EN/FIFO_RDY can be left as a no-connect as there is an internal weak pull-down resistor. In this case, in order for the QSpan to become a PCI bus master a write to the PCI_CS register is required.

C.3.1.11 JTAG Signals

The QSpan BGA device supports JTAG. The QSpan's JTAG signals should be connected to the JTAG controller or to the JTAG signals of another device if devices are to be chained together. If JTAG will not be supported then the JTAG signals may be left open as the inputs have internal pull-up resistors.

Appendix D Initialization

This appendix specifies the hardware and software configuration required to initialize the QSpan. This is an overview. Please read the relevant sections of the QSpan PCI to Motorola Bridge Manual, cross-referenced below, to better understand the options described in this appendix.

The following topics are discussed in this appendix.

D.1 Hardware Configuration

See for configuring the QSpan in connection with Motorola hardware.

D.2 QSpan Register Configuration

This section describes how to program the registers of the QSpan to make use of its various channels and interfaces. Some of the register values can be programmed by the EEPROM..

The globe symbol alerts the reader that the initialization must be performed as a minimum to access the channel in question.

The following topics are discussed in this section:

The Miscellaneous Control Register bits affect the QSpan's mode of operation. Please see Table D.1.

Table D.1 Summary of the QSpan's Miscellaneous Control Register

Register	Field	Description
MISC_CTL	MSTSLV[1:0]	This field determines the types of cycles the QSpan Slave Module accepts, and the types of cycles the Master Module generates. It is read only and is set after a reset.
	QB_BOC	Determines whether the QSpan generates Little- or Big-endian QBus cycles. (See INVEND bit in PCI target control registers)
	S_BG	If using the PowerQUICC's arbiter, use default settings. If using the QUICC's arbiter, then these bits must be altered from their default settings (i.e. set to '1').
	S_BB	
	SW_RST	Controls the assertion of $\overline{\text{RESETO}}$ and can be used to allow boot code to be downloaded from the PCI host.
	MA_BE_D	This bit controls the QSpan's response to the Qbus processor when a Master Abort or Target Abort occurs on the PCI bus.
	PRCNT	This field controls the amount of data that is prefetched when a PCI burst read occurs to the QSpan's PCI Target Channel. (Please refer to the PCI Target Channel control registers)

D.2.1 QBus Slave Channel Initialization

To support two QBus slave images one must program both QBus slave image 0 and 1 registers. Once these registers have been programmed, the QBus processor is able to read and write data from the PCI bus.

Table D.2 QBus Slave Channel Programming Summary

Register	Field	Description
QBSI0_CTL and/or QBSI1_CTL	PWEN	Enables posted writes
	PAS	Set the PCI bus address space for either Memory or I/O space
QBSI0_AT and/or QBSI1_AT	EN	Enables address translation
	BS[3:0]	Block Size of Slave Image (affects number of address lines translated, if address translation is enabled)
	TA[31:16]	Translation Address
PCI_CS	BM	Enables the QSpan to become PCI bus master.

D.2.2 Register Access From the PCI Bus

The QSpan's PCI Configuration registers are accessible from the PCI Bus in PCI Configuration or Memory Space. The QSpan device specific registers are only accessible in Memory Space. The QSpan's PCI Configuration Registers are accessible without any software initialization requirements. To Access the QSpan device specific registers in PCI Memory Space the following bits must be configured.

Table D.3 Register Access

Register	Field	Description
PCI_CS	MS	Set this bit for the QSpan to be able to respond to PCI Memory space cycles. (This is a global bit, the PAS bit of each specific image also needs to be set.)
PCI_BSM to the QSpan	BA	Specifies the base address of the PCI memory image for access registers (size of image is 4K)

D.2.3 PCI Target Channel Initialization

There are two possible memory ranges on the PCI bus which can be used to gain access to QBus resources. These ranges and the associated registers which need to be programmed to access them are referred to as “target images”, of which there are two. Each PCI target image can have independent features as described below. To support two PCI target images one must program both PCI Target Image 0 and 1 registers.

Table D.4 PCI Target Image Programming Summary

Register	Field	Description
PCI_CS	MS	Set this bit for the QSpan to be able to respond to PCI Memory space cycles. (This is a global bit, the PAS bit of each specific image also needs to be set.)
	IOS	Set this bit for the QSpan to be able to respond to PCI I/O space cycles. (This is a global bit, the PAS bit of each specific image also needs to be set.)

Table D.4 PCI Target Image Programming Summary (Continued)

Register	Field	Description
PBTI0_CTL and/or PBTI1_CTL	EN	Enables the image
	PWEN	Posted Write Enable
	PREN	Prefetch Read Enable bit (used in conjunction with the PRCNT field in the MISC_CTL register)
	BRSTWEN	Burst Write Enable (ensure hardware is designed to support this feature)
	INVEND	Inverts the endian-ness from the state of the QB_BOC bit.
	BS[3:0]	Block Size of Target Image (affects number of address lines translated)
	PAS	QBus Address Space: set for either Memory or I/O space
	TC[3:0]	Determines how the TC[3:0] lines of the QSpan are driven.
	DSIZE[1:0]	Specifies the size of the destination port on the QBus
PBTI0_ADD[a] and/or PBTI1_ADD	BA[31:16]	Specifies the base address of the PCI memory image which the QSpan monitors. See “Address Translation” on page 2-30.
	TA[31:16]	Specifies part of the value for the translated local address. See “Address Translation” on page 2-30.

a. Also refer to PCI_BST0 and PCI_BST1 registers

Once the target images have been programmed, PCI masters can read from and write to the QBus.

These two registers can also be accessed through PCI Configuration space.

D.2.3.1 Error Logging and Interrupts

The QSpan has registers which allow the processor to recover from a failed write cycle on the QBus resulting from an entry being dequeued from the Px-FIFO. This section applies to posted writes only; delayed reads and delayed writes are treated differently. The QSpan has the ability to log the failed QBus posted write cycle and generate an interrupt back to the PCI master or the QBus.

Table D.5 QBus Error Logging Programming Summary

Register	Field	Description
QB_ERRCS	ES	Indicates if a failed cycle is currently logged
	EN	Enables error logging.
	TC_ERR[3:0]	Logs the status of the TC[3:0]of the failed cycle
	SIZ_ERR[1:0]	Logs the SIZ[1:0] field of the failed cycle
QB_AERR	QAERR[31:0]	Logs the 32-bit address of the failed transaction
QB_DERR	QDERR[31:0]	Logs the 32-bit data of the failed transaction

See “Posted Writes” on page 2-42 for an explanation of error recording in this channel. The PCI Target Channel can be programmed to cause errors to trigger an interrupt on the PCI bus or the QBus, as described in .

D.2.4 IDMA Channel Initialization

The IDMA channel has a single FIFO which can be used to perform burst writes or burst reads on the PCI Bus. During an IDMA cycle the QSpan is an IDMA peripheral while either the QUICC or the PowerQUICC is the bus master of the cycle. The QUICC (or PowerQUICC) must program three registers in order to initiate an IDMA transfer.

See “The IDMA Channel” on page 2-49 for more IDMA register programming information.

Table D.6 IDMA Channel Programming Summary

Register	Field	Description
IDMA_ADD	ADDR[31:2]	The starting address of the IDMA transaction on the PCI bus.
IDMA_CNT	CNT[18:0]	This register specifies the number of bytes which will be transferred.
IDMA_CS	(various)	Fields in this register should be set according to the transaction requirements (see “The IDMA Channel” on page 2-49.)

Interrupts can be generated based on a family of four IDMA event types: QBus error, PCI bus error, IDMA reset, or the assertion of the IDMA done bit. Their respective status bits in the IDMA_CS register will cause an interrupt if enabled in the INT_CTL register (see the tables below, and “The Interrupt Channel” on page 2-64)

.

Table D.7 IDMA Channel Interrupt Programming Summary

Register	Field	Description
PCI_CS	BM	Enables the QSpan to become the PCI bus master
INT_CTL	IQE_EN	Must be set to 1 to enable assertion of an interrupt resulting from an IDMA QBus error
	IPE_EN	Must be set to 1 to enable assertion of an interrupt resulting from an IDMA PCI bus error
	IRST_EN	Must be set to 1 to enable assertion of an interrupt resulting from an IDMA reset condition
	DONE_EN	Must be set to 1 to enable assertion of an interrupt resulting from an IDMA complete condition
INT_DIR	IQE_DIR	Determines whether a QBus error triggers a PCI bus interrupt or a QBus interrupt.
	IPE_DIR	Determines whether a PCI bus error triggers a PCI bus interrupt or a QBus interrupt.
	IRST_DIR	Determines whether an IDMA reset condition triggers a PCI bus interrupt or a QBus interrupt.
	DONE_DIR	Determines whether the assertion of the IDMA done bit in the IDMA_CS register triggers a PCI bus interrupt or a QBus interrupt.
INT_STAT	IQE_IS	Indicates an IDMA QBus error-based interrupt, a 1 must be written to clear the status bit
	IPE_IS	Indicates a PCI bus error-based interrupt, a 1 must be written to clear the status bit
	IRST_IS	Indicates an IDMA reset-based interrupt, a 1 must be written to clear the status bit
	DONE_IS	Indicates an IDMA done-based interrupt, a 1 must be written to clear the status bit

D.2.5 General Purpose Software Interrupts

See “The Interrupt Channel” on page 2-64 for a discussion of doorbell interrupts.

D.2.6 PCI Interface—Miscellaneous

This section describes the following subjects:

D.2.6.1 Generation of PCI Configuration and IACK Cycles

The QSpan has the ability to generate PCI Configuration cycles through a QBus master accessing QSpan registers. In order to generate a PCI Configuration read or write in this manner, the Configuration Address register must first be programmed. PCI Configuration read and write cycles are implemented as delayed transactions. In order to initiate a configuration write cycle on the PCI bus a write is performed to the CON_DATA Register. In order to generate a read the QBus master should perform a read of the Configuration Data Register.

See "PCI Configuration Cycles Generated from the QBus" on page 2-62, page A-35 and page A-37.

In order to generate a PCI IACK Cycle, the QBus processor should perform a read of the IACK Cycle Generator Register. Please refer to A-38 for more information.

,

Table D.8 PCI Configuration Cycle Programming Summary

Register	Field	Description
CON_ADD	BUS_NUM[7:0]	See PCI specification and QSpan register descriptions.
	DEV_NUM[3:0]	
	FUNC_NUM[2:0]	
	REG_NUM[5:0]	
	TYPE	Determines whether the Configuration cycle generated is of Type 0 or Type 1.
CON_DATA	CDATA[31:0]	When this register is written to its contents are driven onto the PCI data bus during a Configuration write cycle. The QBus master is retried until the write completes on the PCI bus (i.e., writes are delayed transactions). A read of this register initiates a PCI Configuration read cycle. The value returned from the PCI target is stored temporarily in this register while waiting for the QBus master to perform an additional read to obtain the data (i.e., reads are also implemented as delayed transactions).
MISC_CTL	MA_BE_D	This bit controls the QSpan's response to the Qbus processor when a Master Abort or Target Abort occurs on the PCI bus

Table D.8 PCI Configuration Cycle Programming Summary

Register	Field	Description
IACK_GEN	IACK_VEC	When the register is read, the QSpan will perform a PCI IACK Cycle. The QBus master is retried until the cycle completes on the PCI Bus. The value from the PCI IACK is temporarily stored in this register while waiting for the QBus master to perform an additional read to obtain the vector.

D.2.6.2 PCI Expansion ROM Implementation

An Expansion ROM can be implemented on the QBus which will contain additional information for the PCI host. In order for the PCI host to be able to read from this ROM the base address of this image must be programmed. This address can be programmed from an external serial EEPROM.

Table D.9 PCI Expansion ROM programming

Register	Field	Description
PCI_CS	MS	Set this bit before accessing the ROM.
PCI_BSROM	BA[31:16]	Defines the base address for the expansion ROM
	EN	Set this in order to use the expansion ROM image
PBROM_CTL	DSIZE[1:0]	Defines the size of the Expansion ROM (read only field, programmed by serial EEPROM
	BS[2:0]	Defines the block size of the Expansion ROM image (read only, programmed by serial EEPROM
	TC[3:0]	Defines how the QSpan drives the TC lines (read only, programmed by serial EEPROM
	TA[31:16]	Defines the Translation Address field (read only, programmed by serial EEPROM

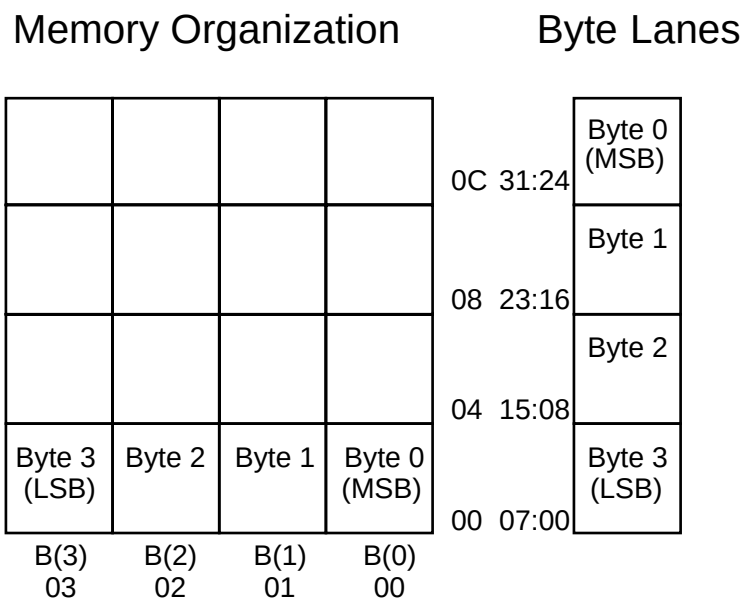
D.3 EEPROM

Many of the QSpan's operating parameters can be set by a serial EEPROM (see "The EEPROM Channel" on page 2-69.)

Appendix E Endian Mapping

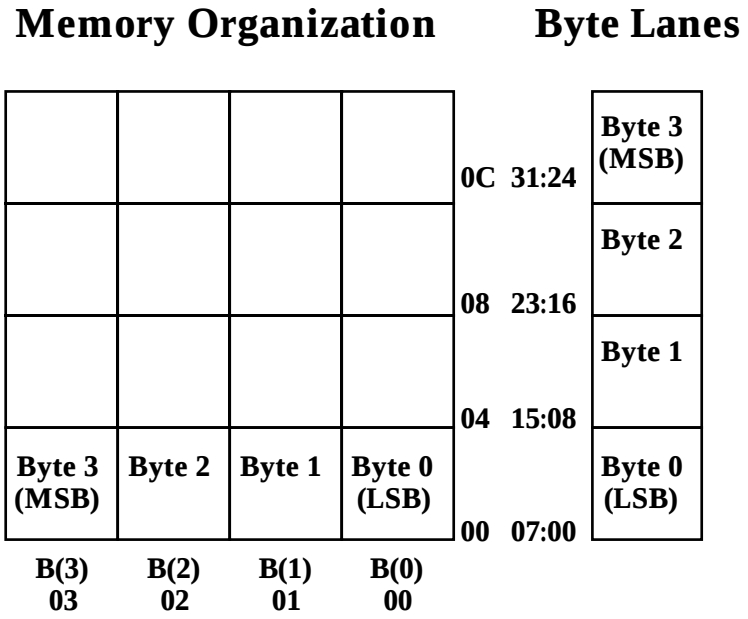
The PCI bus and the Motorola processors have some differences stemming from their separate evolutionary histories. PCI was born in the Intel world, making it Little-Endian, while the Motorola processors used Big-Endian. In a Big-Endian system, the most significant byte is located at the lowest address in memory. When the data is moved to the data bus, the least significant byte is moved to the lowest byte lane (Byte 3 in lowest byte lane) and the most significant byte is moved to the highest byte lane (Byte 0 in highest byte lane). Figure E.1 below shows a 4-byte operand being moved to the data bus in a Big-Endian system.

Figure E.1 : Big-Endian System



In a Little-Endian system, the most significant byte is located at the highest address location. When data is moved to the data bus, the least significant byte is moved to the lowest byte lane (Byte 0 in lowest byte lane) and the most significant byte is moved to the highest byte lane (Byte 3 in highest byte lane). It is important that the PCI bridge provides flexibility in how endian systems are mapped across the interface. Figure E.2 below shows a 4-byte operand being moved to the data bus in a Little-Endian system.

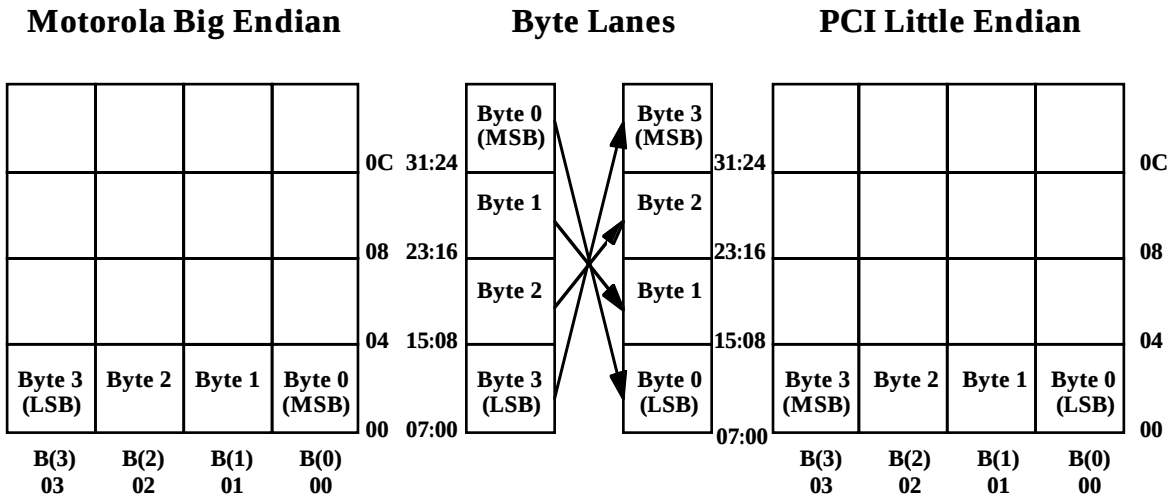
Figure E.2 : Little-Endian System



Note that many (if not all) host bus adapters (e.g. SCSI) for the PCI environment expect their descriptor blocks to be stored in main memory in Little-Endian format. This means that a PCI-to-Motorola bridge must provide a flexible endian mapping scheme to allow for PCI adapter control information to be stored in Motorola memory.

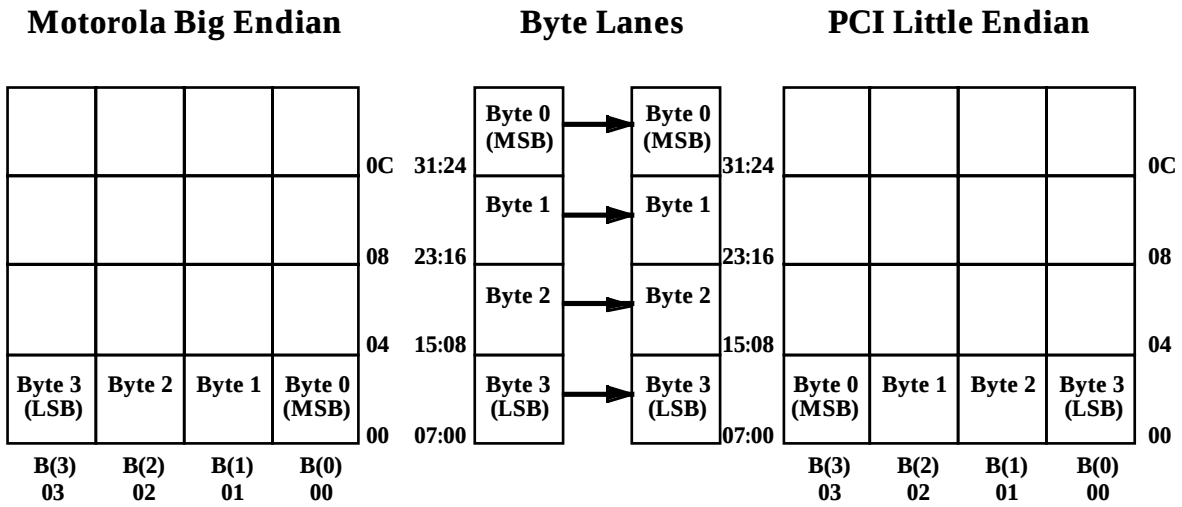
There are two approaches to endian mapping: address invariance and data invariance. With address invariance, the addressing of the bytes in memory is preserved. Figure E.3 shows that by performing byte lane swapping, the bytes appear in the same address but their relative significance is not preserved. This is fine for text information but scrambles operands.

Figure E.3 : Address Invariant Mapping



The other approach is data invariance, which preserves the relative byte significance but translates the byte addressing. Figure E.4 below shows that with data invariance, Byte 0 is still the most significant byte in the data structure but is now located at address 03 in memory rather than address 00.

Figure E.4 : Data Invariant Mapping.



By enforcing certain constraints on the system, it is possible to implement both options in a PCI-to-Motorola bridge. By assuming that all data structures are 32-bit integers, the bridge could be powered up in either of these mapping modes. In address invariant mode, byte lanes would be swapped (independent of the data path width) assuming that the bytes are part of a 32-bit word. In data invariant mode, the byte lanes would be passed straight through assuming that the bytes are again part of a 32-bit word.

Appendix F Operating and Storage Conditions

F.1 5 Volts Data

Table F.1 Power Dissipation

QCLK[a]	Minimum	Typical	Maximum
25 MHz	0.55W	1.00W	1.25W
40 MHz	0.65W	1.15W	1.50W
50 MHz	0.75W	1.25W	1.75W

a. PCI clock always runs at 33MHz

Table F.2 5.0V Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
V_{DD}	DC Supply Voltage	-0.3 to 7.0	V
V_{IN}	DC Input Voltage	-0.3 to VDD + 0.3	V
I_{IN}	DC Input Current	±10	mA
T_{STG}	Storage Temperature	-40 to +125	°C

Table F.3 5.0V Recommended Operating Conditions

Symbol	Parameter	Rating	Units
V_{DD}	DC Supply Voltage	4.5 to 5.5	V
T_C	Commercial Temperature	0 to 70	°C
T_I	Industrial Temperature	-40 to 85	°C

WARNING: Stresses beyond those listed above may cause permanent damage to the devices. These are stress ratings only, and functional operation of the devices at these or any other conditions beyond those indicated in the operational sections of this document is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

F.1.1 Thermal Characteristics

The maximum ambient temperature of the QSpan can be calculated as follows:

$Ta \leq Tj - \theta_{ja} * P$

Where,

T_a = Ambient temperature (°C)

T_j = Maximum QSpan Junction Temperature (°C) = 125°C

θ_{ja} = Ambient to Junction Thermal Impedance (°C / Watt) see Table F.4 below.

P = QSpan power consumption (Watts), see Table F.1 above.

The ambient to junction thermal impedance (θ_{ja}) is dependent on the air flow in linear feet per minute over the QSpan.

Table F.4 5.0 Volt Package Thermal Resistance

Wind Speed (LFPM)	θ_{ja} °C/W
0	30.1

F.2 3.3 Volt Data

Table F.5 Power Dissipation

QCLK[a]	Minimum	Typical	Maximum
25 MHz	0.28W	0.50W	0.63W
40 MHz	0.33W	0.58W	0.75W
50 MHz	0.38W	0.63W	0.90W

a. PCI clock always runs at 33MHz

Table F.6 3.3 Volt Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
V_{DD}	DC Supply Voltage	-0.3 to 7.0	V
V_{IN}	DC Input Voltage	-0.3 to 5.3[a] [b]	V
I_{IN}	DC Input Current	±10	mA
T_{STG}	Storage Temperature	-40 to +125	°C

a. Power available on VIO without power to VDD (VIN) can result in reliability impact.

b. The PBGA device is 5V tolerant on all pins except for JTAG inputs (TCK, TDI, TMS, $\overline{TRST}$)

Table F.7 3.3 Volt Recommended Operating Conditions

Symbol	Parameter	Rating	Units
V_{DD}	DC Supply Voltage	3.14 - 3.46	V
T_C	Commercial Temperature	0 to 70	°C
T_I	Industrial Temperature	-40 to 85	°C

F.2.1 Thermal Characteristics

The maximum ambient temperature of the QSpan can be calculated as follows:

$Ta \leq Tj - \theta_{ja} * P$

Where,

T_a = Ambient temperature (°C)

T_j = Maximum QSpan Junction Temperature (°C) = 125°C

θ_{ja} = Ambient to Junction Thermal Impedance (°C / Watt) see Table F.4 below.

P = QSpan power consumption (Watts), see Table F.1 above.

The ambient to junction thermal impedance (θ_{ja}) is dependent on the air flow in linear feet per minute over the QSpan.

Table F.8 3.3 Volt Package Thermal Resistance

Wind Speed (LFPM)	θ_{ja} °C/W
0	31.4

Appendix G Mechanical and Ordering Information

G.1 Mechanical Information

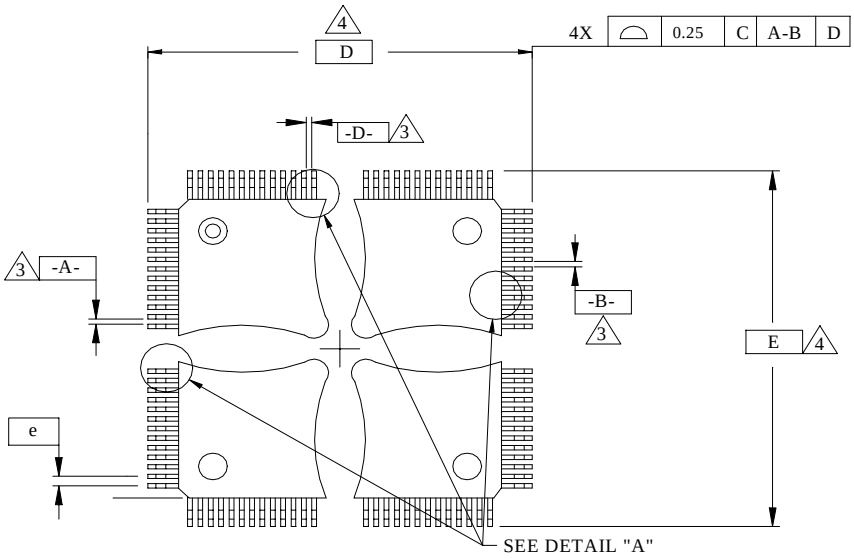
G.1.1 208 PQFP

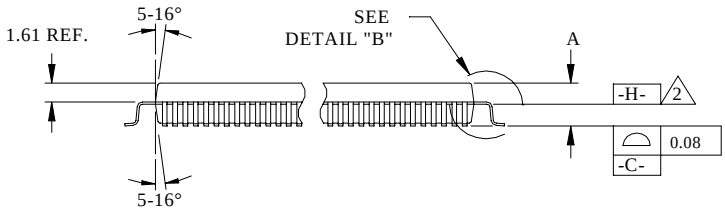
NOTES:

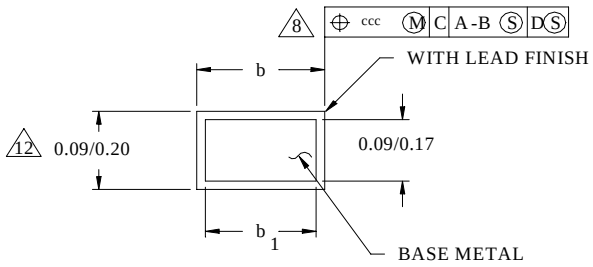
1. ALL DIMENSIONS AND TOLERANCES CONFORM TO AMSE Y14.5-1994.
2. DATUM PLANE -H- LOCATED AT MOLD PARTING LINE AND COINCIDENT WITH LEAD, WHERE LEAD EXITS PLASTIC BODY AT BOTTOM OF PARTING LINE.
3. DATUMS A-B AND -D- TO BE DETERMINED WHERE CENTERLINE BETWEEN LEADS EXITS PLASTIC BODY AT DATUM PLANE -H-.
4. TO BE DETERMINED AT SEATING PLANE -C-.
5. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION IS 0.25 MM PER SIDE. DIMENSIONS D1 AND E1 DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
6. "N" IS NUMBER OF TERMINALS.
7. PACKAGE TOP DIMENSIONS ARE SMALLER THAN BOTTOM DIMENSIONS BY 0.20 MM, AND TOP OF PACKAGE WILL NOT OVERHANG BOTTOM OF PACKAGE.
8. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 MM TOTAL IN EXCESS OF b DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CAN NOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.
9. ALL DIMENSIONS ARE IN MILLIMETERS.
10. THIS DRAWING CONFORMS TO JEDEC REGISTERED OUTLINES MS-029 VARIATION FA-1.
11. MAXIMUM ALLOWABLE DIE THICKNESS TO BE ASSEMBLED IN THIS PACKAGE FAMILY IS 0.635 MM.
12. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10 MM AND 0.25 MM FROM THE LEAD TIP.

SYMBOL	ALL DIMENSIONS IN MILLIMETRES			NOTE
	MIN.	NOM.	MAX.	
A			4.10	
A_1	0.25		0.50	
A_2	3.20	3.40	3.60	
D	30.60 BASIC			4
D_1	28.00 BASIC			5
E	30.60 BASIC			4
E_1	28.00 BASIC			5
L	0.45	0.60	0.75	
N	208			6
e	0.50 BSC.			
b	0.17		0.27	
b_1	0.17	0.20	0.23	
ccc	0.08			

Figure G.1 : 208-Pin PQFP

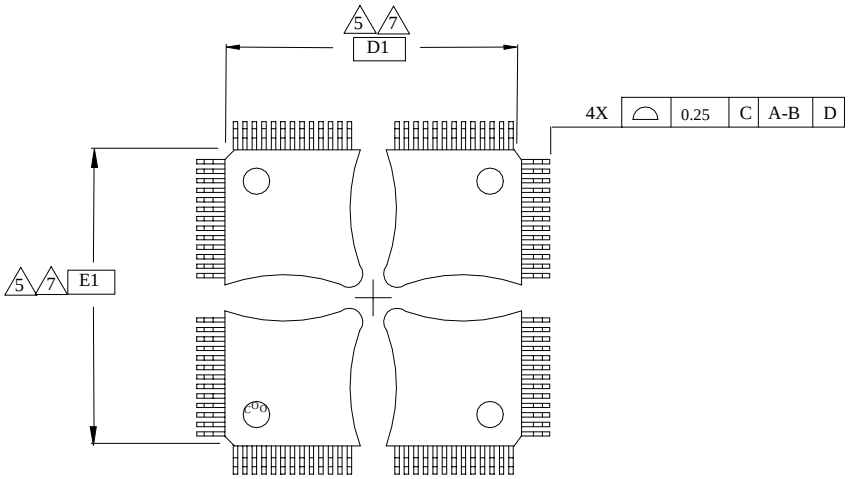




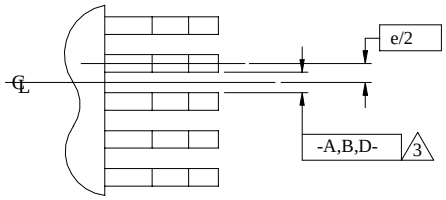


SECTION C-C

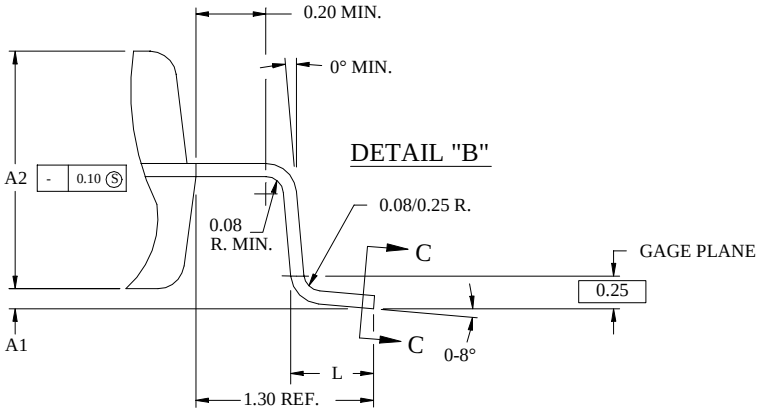
Figure G.1 208-Pin PQFP (continued)



BOTTOM VIEW

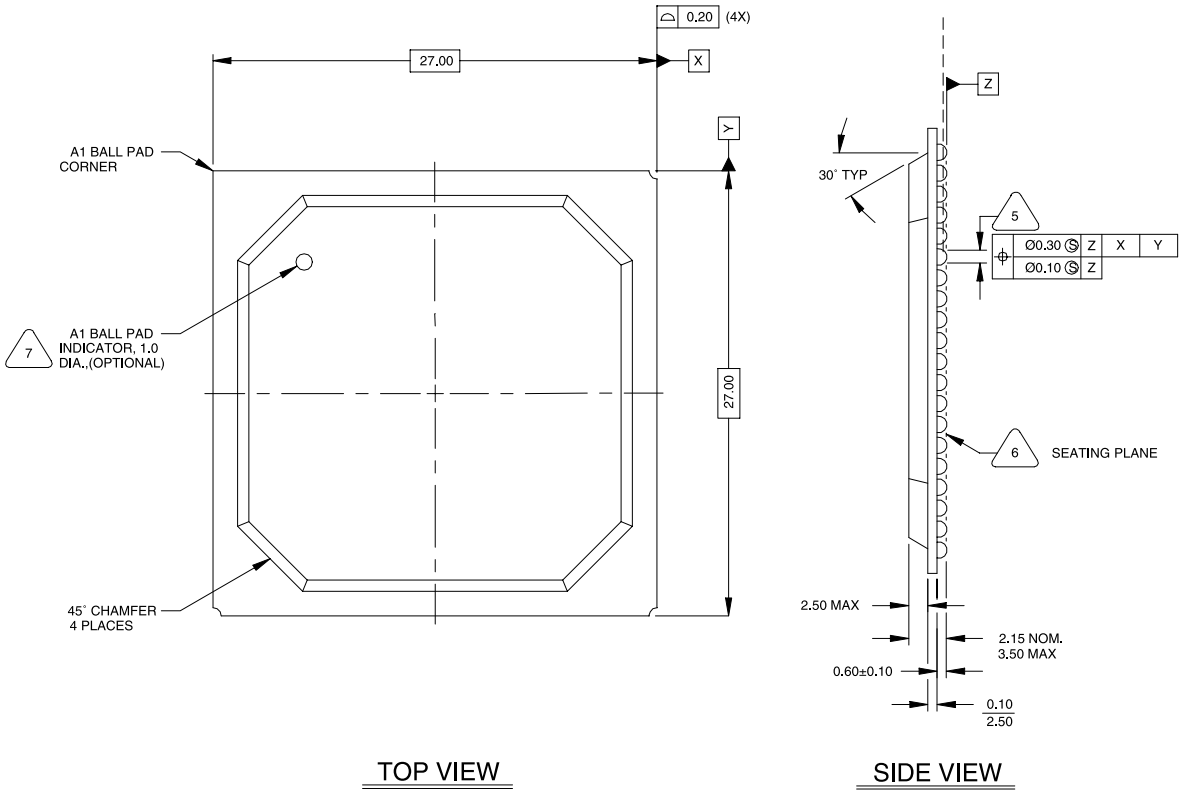


DETAIL "A"



G.1.2 256 PBGA

Figure G.2 : 256-lead PBGA



8. REFERENCE SPECIFICATIONS:
   A. THIS DRAWING CONFORMS TO THE JEDEC REGISTERED OUTLINE MO-151/B

7. A1 BALL PAD CORNER I.D. FOR PLATE MOLD: TO BE MARKED BY INK.
   AUTO MOLD: DIMPLE TO BE FORMED BY MOLD CAP.

6. PRIMARY DATUM Z AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.

5. DIMENSION IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM Z.

4. THE MAXIMUM ALLOWABLE NUMBER OF SOLDER BALLS IS 441 FOR PERIPHERIAL AND 324 FOR FULL MATRIX.

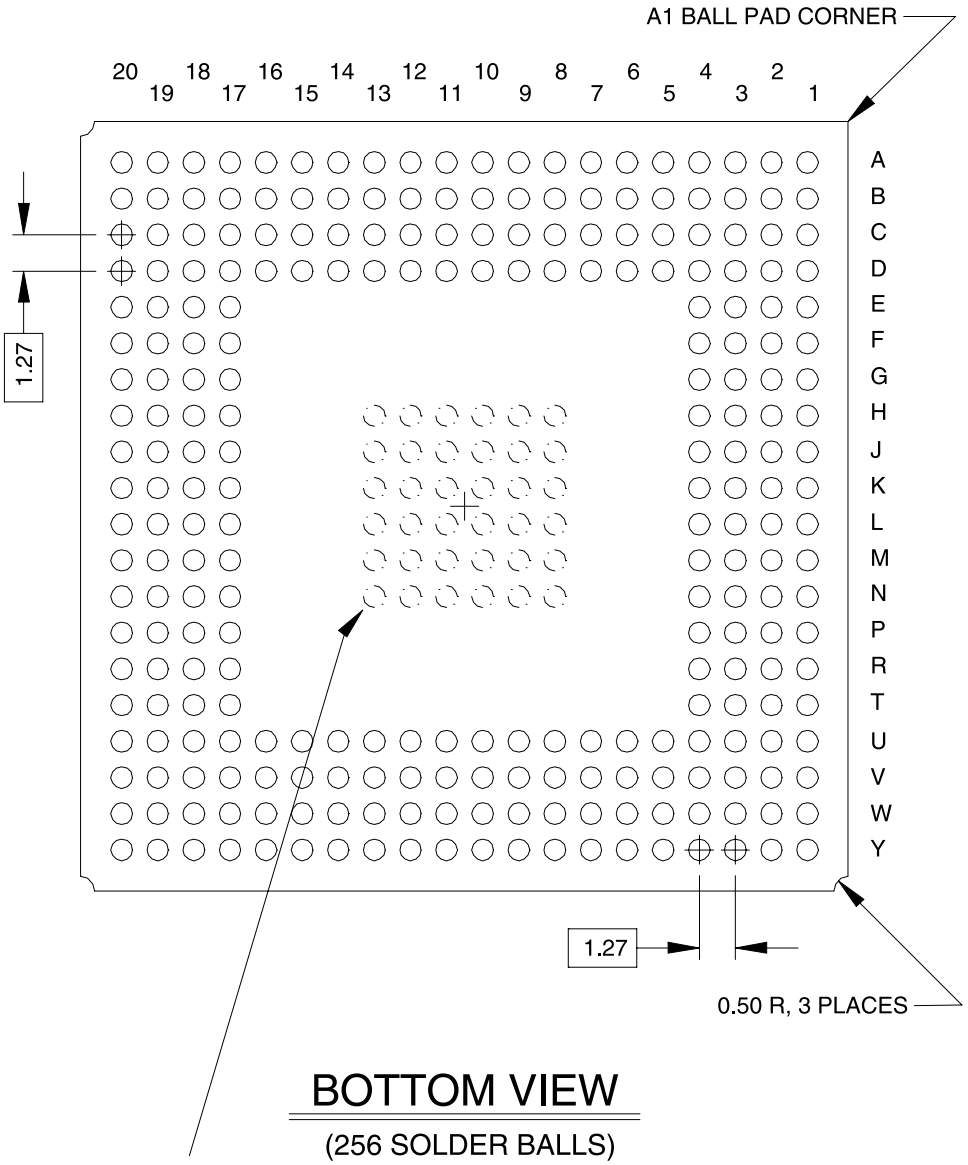
3. THE MAXIMUM SOLDER BALL MATRIX SIZE IS 21 X 21 FOR PERIPHERIAL AND 18 X 18 FOR FULL MATRIX.

2. THE BASIC SOLDER BALL GRID PITCH IS 1.27mm FOR PERIPHERIAL AND 1.50mm FOR FULL MATRIX.

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1994.

NOTES: UNLESS OTHERWISE SPECIFIED

Figure G.2 : 256-lead PBGA (continued)



Note : This substrate can accomodate up to 292 balls.
In order to make this package conform to the 256 footprint, the assembly contractor does not install the middle balls, omitting the balls in the center matrix.

G.2 Ordering Information

Tundra products are designated by a part number. When ordering, refer to products by their full part number. For detailed mechanical drawings or alternative packaging requirements, please contact Tundra directly.

Table G.1 Standard Ordering Information

Part Number	Frequency[a]	Voltage[b]	Temperature	Package
CA91C860B-40CQ	25MHz MC68360 40MHz MPC860	5	0° to 70°C	PQFP
CA91C860B-40IQ	25MHz MC68360 40MHz MPC860	5	-40° to 85° C	PQFP
CA91C860B-50CQ	33MHz MC68360 50MHz MPC860	5	0° to 70°C	PQFP
CA91L860B-40CE	25MHz MC68360 40MHz MPC860	3.3	0° to 70°C	PBGA
CA91L860B-40IE	25MHz MC68360 40MHz MPC860	3.3	-40° to 85° C	PBGA
CA91L860B-50CE	33MHz MC68360 50MHz MPC860	3.3	0° to 70°C	PBGA

a. The Tundra QSpan is compatible with all M68040 variants in large buffer mode up to 40 MHz. The Tundra QSpan is compatible with all M68040 variants in small buffer mode up to 33 MHz.

b. Note that the 3.3 volt package version supports universal PCI (3.3V / 5V tolerant inputs and 3.3V / 5V compliant output signalling).

Index

Q

T

Worldwide Sales Network

USA Sales Representatives

Eastern

D.C., Maryland, Virginia

Chesapeake Technologies

3905 National Dr., Suite 425
Burtonsville, MD 20866

Tel: 301-236-0530
Fax: 301-384-9596

Connecticut, Maine, Massachusetts, New Hampshire, New Jersey, New York, Rhode Island, Vermont

Daner Hayes Inc.

300 Eliot St., Suite 342
Ashland, MA 01721

Tel: 508-881-0400
Fax: 508-881-8227

Upstate New York, Eastern Pennsylvania, Southern New Jersey, Delaware

Supertech

103-A Kingsberry Dr.
Rochester, NY 14626

Tel: 716-720-0915
Fax: 716-720-0916

Ohio

EMA, Inc.

35 Compark Road,
Centreville, OH 45459

Tel:230-676-9147
Fax: 230-676-9147

Florida

Apollo Technical Sales Co.

1275S Patrick Dr., #M2
Satellite Beach, FL
32937-3963

Tel: 407-777-7511
Fax: 407-777-5251

Apollo Technical Sales Co.

1251 South Federal Highway,
Suite E-210
Boca Raton, FL
33432-7352

Tel: 561-750-9127
Fax: 561-750-9132

Central

Illinois, Michigan, Wisconsin

Horizon Technical Sales

4340 Cross Street,
Downer's Grove, IL 60106

Tel: 630-852-2500
Fax: 630-852-2520

Horizon Technical Sales

6011 Hilldale Dr.
Hartford, WI 53027-9541

Tel: 414-670-6776
Fax: 414-670-6778

Apollo Technical Sales Co.

43 Leeward Island
Clearwater, FL
33767

Tel: 813-445-1646
Fax: 813-468-9496

Alabama, Georgia, Mississippi, North Carolina, South Carolina, Tennessee

BITS, Inc.

2707 Artie St., Suite 10
Huntsville, AL 35805

Tel: 256-534-4020
Fax: 256-534-0410

BITS, Inc.

1 Meca Way
Norcross, GA 30093

Tel: 770-564-5599
Fax: 770-564-5588

BITS, Inc.

7706 Six Forks Rd. Suite 101
Raleigh, NC 27615

Tel: 919-676-1880
Fax: 919-676-1881

BITS, Inc.

2305 Keaton Ave., Suite E
Charlotte, NC 28269

Tel: 704-510-1064
Fax: 704-510-1065

Minnesota, Oklahoma, Texas, Louisiana, Akansas

Quad State Sales and Marketing

12160 Abrams Rd. Suite 406
Dallas, TX 75243

Tel: 972-669-8567
Fax: 972-669-8834

Quad State Sales and Marketing

8310 Capital of Texas
Highway North, Suite 365
Austin, TX 787311

Tel: 512-346-7002
Fax: 512-346-3601

Quad State Sales and Marketing

10565 Katy Freeway,
Suite 212
Houston, TX 77024

Tel: 713-467-7749
Fax: 713-467-5942

USA Sales Representatives Cont'd

Quad State Sales and Marketing

110 West Commercial,
Suite 210
Broken Arrow, OK 74013

Tel: 918-258-7723
Fax: 918-258-7653

Electromec Sales Inc.

1601 East Highway 13,
Suite 102
Burnsville, MN 55337

Tel: 612-894-8200
Fax: 612-894-9352

Arkansas, Iowa, Kansas, Louisiana, Missouri, Nebraska, North Dakota, South Dakota, Southern Illinois, Western Wisconsin

Please contact Tundra Semiconductor directly at: 1-800-267-7231Mountain

Arizona, New Mexico, Southern Nevada

S&S Technologies

4545 South Wendler Dr.
Suite 121
Tempe, AZ 85282

Tel: 602-414-1112
Fax: 602-414-1125

Colorado, Idaho, Montana, Utah, Wyoming

Intermountain Component Sales

12665 South Mountain Dr.
Drasper, UT 84020

Tel: 801-572-4010
Fax:801-572-8180

Intermountain Component Sales

3028 West 134th Place
Broomfield, CO 80020

Tel:303-469-6136
Fax:801-572-8180

Western

Northern California, Northern Nevada

Brooks Technical Group

10080 North Wolfe Road
SW3-100
Cupertino, CA 95014

Tel: 408-252-3880
Fax: 408-252-4434San Diego

Earle Associates

7585 Ronson Rd., Suite 200
San Diego, CA 92111

Tel: 619-278-5441
Fax: 619-278-5443

Southern California

First Rep

11145 Tampa Ave.,
Suite 25B
Porter Ranch, CA 91326

Tel: 818-831-3158
Fax: 818-831-9861

Oregon, Washington State

Electronic Sources, Inc.

6865 S.W. 105th, Suite B
Beaverton, OR
97008

Tel: 503-627-0838
Fax: 503-627-0238

Electronic Sources, Inc.

1603 116th Ave. NE
Suite 115
Bellevue, WA 98004

Tel: 425-451-3500
Fax: 425-451-1038

International Sales Representatives

Australia

ACD

Unit 2, 17-19 Melrich Rd.
Bayswater, Victoria
Australia 3153

Tel: 61-3-9762-7644
Fax: 61-3-9762-5446

ACD

106 Belmore Rd.
North Riverwood, NSW
Australia 2210

Tel: 61-2-534-6200
Fax: 61-2-534-4910

ACD

20D William Street,
Norwood, South Australia
Australia

Tel: 61-8-364-2844
Fax: 61-8-364-2811

ACD

Unit 1, Technology Park
Bently, West Australia
Australia

Tel: 61-9-472-3232
Fax: 61-9-470-2303

Belgium, the Netherlands

Please contact Tundra Semiconductor directly at 613-592-0714

Canada

Hewetson Repping, Inc.

161 Herzberg Rd
Kanata, ON K2K 2Y3
Tel: 613-271-9044
Fax: 613-271-9040

Hewetson Repping, Inc.

215 Traders Blvd, Unit 2
Mississauga, ON L4Z 3K5
Tel: 905-501-8800
Fax: 905-501-8818

Hewetson Repping, Inc.

100 boul. Alexis Nihon
Suite 945
St-Laurent, PQ H4M 2P5
Tel: 514-744-6511
Fax: 514-744-0904

Hewetson Repping, Inc.

103 Douglas Woods Grove SE
Calgary AL T2Z 2H6

Tel: 403-236-4622
Fax:403-236-4366

Denmark, Norway, Sweden

IE Komponenter AB

Ulvsundavagen 106C
Box 11113 161 11 Bromma,
Sweden

Tel: 46 8 804685
Fax: 46 8 262286

Finland

Integrated Electronics OyAb

Laurinmaenkuja 3
FIN-00440
Helsinki Finland

Tel: 358-0-586-1770
Fax: 358-0-586-1771

France

Compress

47 Rue de L'Estérel
539 94 633 Rungis CEDEX
FRANCE

Tel: 33 1 41 802902
Fax: 33 1 46 866763

Germany

Atlantik Electronik, GmbH

Fraunhoferstr. 11A
82152 Planegg Martinsreid,
Martinsreid, Germany

Tel: 89-89505-0
Fax: 89-89505-100

Hong Kong

Qing Cheng Enterprises Ltd.
Rm 1901
19/ F.Nan Fung Centre
264-298 Castle Peak Rd.
Tsuen Wan, Hong Kong

Tel: 852-2493-4202
Fax: 852-2413-6307

International Sales Representatives Cont'd

India

Hynetic Electronics

78/1, 3rd Cross
Gavipuram Extension,
Bangalore, 560019, India

Tel: 91-80-620852
Fax: 91-80-624073

Israel

Migvan Technologies and Engineering Ltd.

13 Hashiloach St.
PO Box 7022
Petach-Tikva 49170, Israel

Tel: 972-3-9240784
Fax: 972-3-9240787

Italy

Newtek Italia S.p.A

Via Bernina, 7
20158 Milano, Italy

Tel: 39 2 46 92 156
Fax: 39 2 46 95 197

Japan

K.K. Rocky

4-44-11, Nogata
Nakano-ku,
Tokyo, 165 Japan

Tel: 813 3228 4511
Fax: 813 3388 1391

Tokyo Electron Device Ltd.

No. 1, Higashikata-cho,
Tsuzuki-ku, Yokohama-city,
Kanagawa, 2240045
Japan

Tel: 81 45 474 7032
Fax: 81 45 474 7016

Korea

VME Tech Inc.

627-19 Shinsa-dong
Kangnam-gu,
5th Floor, Shinsa Building,
Seoul, Korea 135-120

Tel: 82-2-3443-1491
Fax: 82-2-3443-1495

I & C Microsystems Co. Ltd.

8th Floor, Bethel Building,
324-1 Yangjae-Dong Seo,
Seoul, Korea

Tel: 822-577-9131
Fax: 822-577-9130

New Zealand

ACD

University Paton NZ Ltd.
Unit 7, 110 Mays Road,
Penrose
Auckland, NZ

Tel: 649-636-5984
Fax: 649-636-5985

Singapore

BBS Electronics

12 Genting Rd.
The Blue Building
349474, Singapore

Tel: 65 748 8400
Fax: 65 748 8466

Switzerland

Egli, Fischer & Co, Ltd. Zurich

Gotthardstrasse 6
Ch-8022
Zurich, Switzerland

Tel: 41 1 209 81 11
Fax: 41 1 201 22 75

Taiwan

Data Fast Electronics

6 Fl, No 27-1, Alley 169
Kang Ning Street
Hsi Chih
Taipei, Taiwan

Tel: 886 2 6921051
Fax: 886 2 6958240

United Kingdom

Thame Components

Thame Park Road
Thame, Oxfordshire
OX9 3UQ

Tel: 44 1 844 261188
Fax: 44 1 844 261681