# FireBridge II

# 82C814

# Docking Station Controller

# Data Book

Revision: 1.0
912-3000-047
January 08, 1998

**OPTi Inc.**
888 Tasman Drive
Milpitas, CA 95035
Tel: (408) 486-8000
Fax: (408) 486-8001
www.opti.com

# Table of Contents

## Table of Contents (cont.)

## Table of Contents (cont.)

## Table of Contents (cont.)

# List of Figures

# List of Tables

## 1.0 Features

- Provides true hot docking and undocking
- PCI Power Management Compliant
- Supports 3.3V or 5.0V PCI dock
- Host PCI bus can be 3.3V or 5.0V
- Host and Docking PCI buses can be asynchronous
- Works in conjunction with OPTi PCI-to-ISA bridge to provide reliable ISA support on the dock
- Provides eight windows, selectable for memory or I/O
- Offers additional fixed window for VGA
- Supports INTA#, INTB#, INTC#, INTD#
- Supports four bus masters
- Generates PCI clocks for four devices
- Supports cascadeable docking with multiple 82C814 controllers
- Bridge solution increases primary PCI bus bandwidth by off-loading transactions into buffers
- Supports external bus arbiter for secondary PCI bus
- Packaged in 144-pin LQFP (Low-profile Quad Flat Pack)

## 2.0 Overview

This document describes the OPTi 82C814 Docking Station Controller, a true bridge docking solution that allows software to treat the docking station like a dynamically insertable/removable CardBus card.

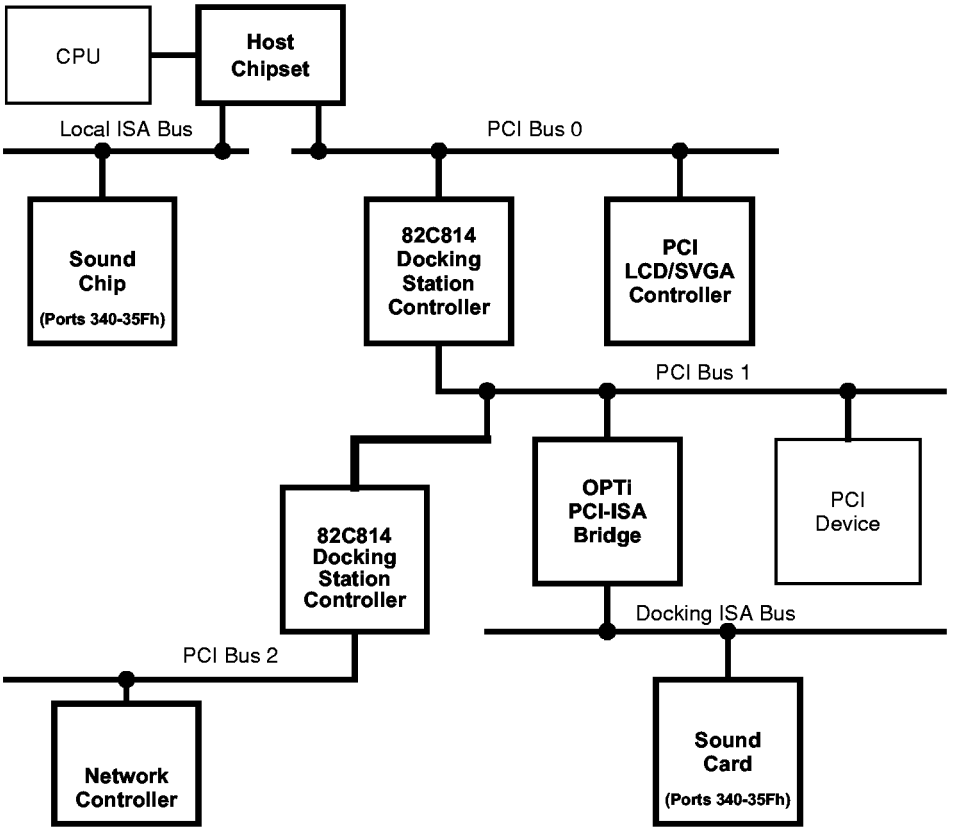
The PCI software interface conforms to the CardBus header layout, instead of the PCI-to-PCI bridge header layout, to overcome the limitations of PCI-to-PCI bridges.

The docking controller implements a true PCI-PCI bridge with full buffering and synchronous or asynchronous operation.

Figure 2-1 illustrates the flexibility of the device, including its ability to support multiple ISA buses when used with an OPTi PCI-to-ISA Bridge.

**Note:** This document describes Revision 1.0 of the 82C814 chip.

**Figure 2-1 Multiple ISA Bus Support and Cascadeable Docking**



# 3.0 Signal Definitions

The 82C814 chip provides a primary interface which is PCI-based. It also provides an independent attachment interface, which can be switched on and off dynamically.

## 3.1 Terminology/Nomenclature Conventions

The "#" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at the high voltage level.

The terms "assertion" and "negation" are used extensively. This is done to avoid confusion when working with a mixture of "active low" and "active high" signals. The term "assert", or "assertion" indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term "negate", or "negation" indicates that a signal is inactive.

The 82C814 has some pins that have multiple functions (denoted by "+" in the pin name). These functions are either:

- cycle-multiplexed (always enabled and available when a particular cycle is in progress),
- a strap option (configured at reset),
- or selected via register programming.

The tables in this section use several common abbreviations. Table 3-1 lists the mnemonics and their meanings.

**Table 3-1 Signal Definitions Legend**

| Mnemonic | Description |
|---|---|
| CMOS | CMOS-level compatible |
| Dcdr | Decoder |
| Ext | External |
| G | Ground |
| I | Input |
| I/O | Input/Output |
| Int | Internal |
| Mux | Multiplexer |
| O | Output |
| OD | Open drain (open-collector) CMOS-level compatible |
| P | Power |
| PD | Pull-down resistor |
| PU | Pull-up resistor |
| S | Schmitt-trigger TTL-level compatible |
| TTL | TTL-level compatible |

**Figure 3-1 Pin Diagram**

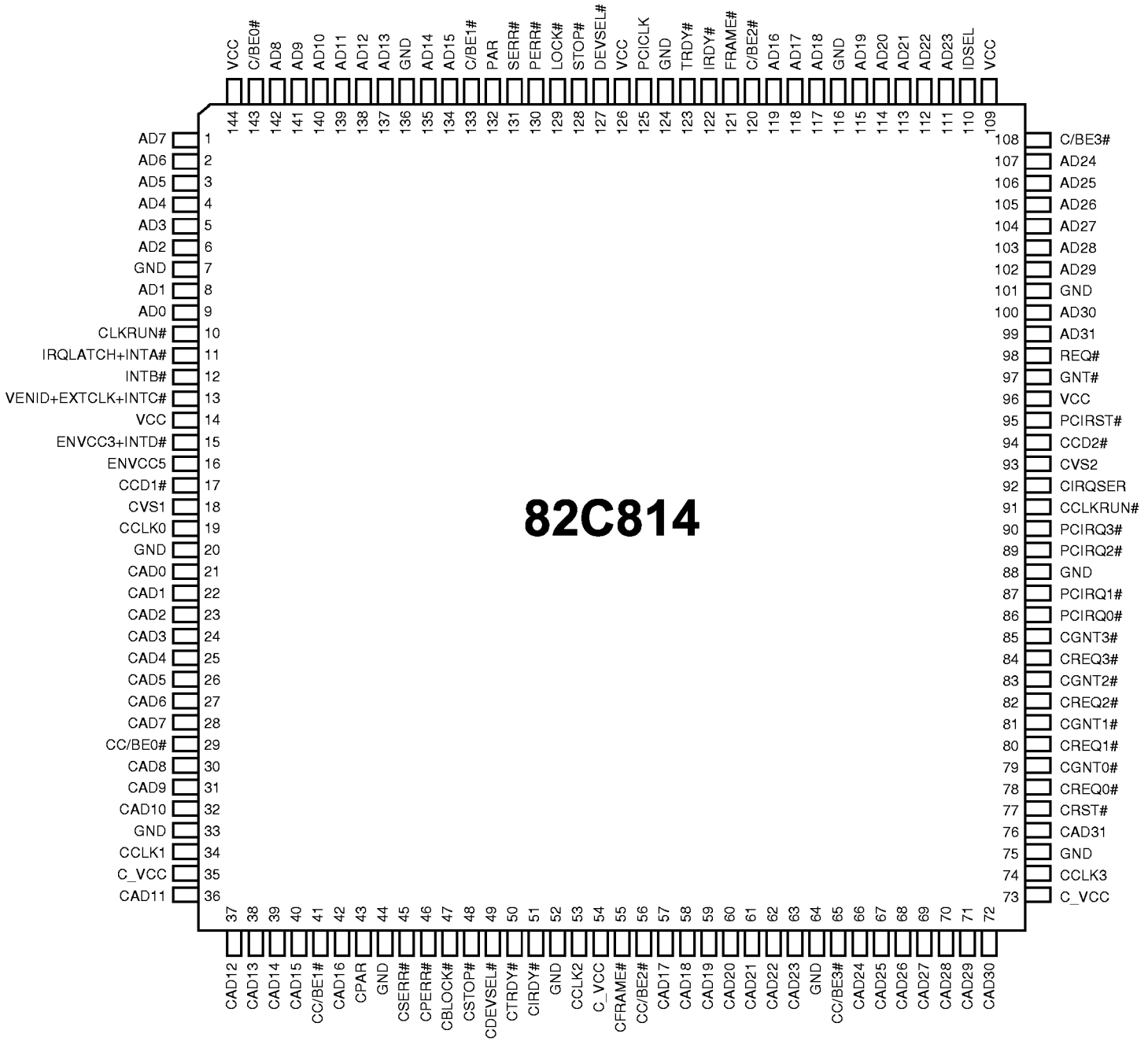


**Table 3-2 Numerical Pin Cross-Reference List**

| Pin No. | Pin Name | Pin Type |
|---|---|---|
| 1 | AD7 | I/O |
| 2 | AD6 | I/O |
| 3 | AD5 | I/O |
| 4 | AD4 | I/O |
| 5 | AD3 | I/O |
| 6 | AD2 | I/O |
| 7 | GND | G |
| 8 | AD1 | I/O |
| 9 | AD0 | I/O |
| 10 | CLKRUN# | I/O |
| 11 | IRQLATCH | I/O |
| | INTA# | I/O |
| 12 | INTB# | I/O |
| 13 | VENID | O |
| | EXTCLK | I |
| | INTC# | I/O |
| 14 | VCC | P |
| 15 | ENVCC3 | O |
| | INTD# | I/O |
| 16 | ENVCC5 | O |
| 17 | CCD1# | I |
| 18 | CVS1 | I |
| 19 | CCLK0 | O |
| 20 | GND | G |
| 21 | CAD0 | I/O |
| 22 | CAD1 | I/O |
| 23 | CAD2 | I/O |
| 24 | CAD3 | I/O |
| 25 | CAD4 | I/O |
| 26 | CAD5 | I/O |
| 27 | CAD6 | I/O |
| 28 | CAD7 | I/O |
| 29 | CC/BE0# | I/O |
| 30 | CAD8 | I/O |
| 31 | CAD9 | I/O |
| 32 | CAD10 | I/O |
| 33 | GND | G |
| 34 | CCLK1 | O |
| 35 | C_VCC | P |
| 36 | CAD11 | I/O |
| 37 | CAD12 | I/O |
| 38 | CAD13 | I/O |
| 39 | CAD14 | I/O |
| 40 | CAD15 | I/O |
| 41 | CC/BE1# | I/O |
| 42 | CAD16 | I/O |
| 43 | CPAR | I/O |
| 44 | GND | G |
| 45 | CSERR# | I/O |
| 46 | CPERR# | I/O |
| 47 | CBLOCK# | I/O |
| 48 | CSTOP# | I/O |
| 49 | CDEVSEL# | I/O |
| 50 | CTRDY# | I/O |
| 51 | CIRDY# | I/O |
| 52 | GND | G |
| 53 | CCLK2 | O |
| 54 | C_VCC | P |
| 55 | CFRAME# | I/O |
| 56 | CC/BE2# | I/O |
| 57 | CAD17 | I/O |
| 58 | CAD18 | I/O |
| 59 | CAD19 | I/O |
| 60 | CAD20 | I/O |
| 61 | CAD21 | I/O |
| 62 | CAD22 | I/O |
| 63 | CAD23 | I/O |
| 64 | GND | G |
| 65 | CC/BE3# | I/O |
| 66 | CAD24 | I/O |
| 67 | CAD25 | I/O |
| 68 | CAD26 | I/O |
| 69 | CAD27 | I/O |
| 70 | CAD28 | I/O |
| 71 | CAD29 | I/O |
| 72 | CAD30 | I/O |
| 73 | C_VCC | P |
| 74 | CCLK3 | O |
| 75 | GND | G |
| 76 | CAD31 | I/O |
| 77 | CRST# | O |
| 78 | CREQ0# | I |
| 79 | CGNT0# | O |
| 80 | CREQ1# | I |
| 81 | CGNT1# | O |
| 82 | CREQ2# | I |
| 83 | CGNT2# | O |
| 84 | CREQ3# | I |
| 85 | CGNT3# | O |
| 86 | PCIRQ0# | I |
| 87 | PCIRQ1# | I |
| 88 | GND | G |
| 89 | PCIRQ2# | I |
| 90 | PCIRQ3# | I |
| 91 | CCLKRUN# | I |
| 92 | CIRQSER | I/O |
| 93 | CVS2 | I |
| 94 | CCD2# | I |
| 95 | PCIRST# | I |
| 96 | VCC | P |
| 97 | GNT# | I |
| 98 | REQ# | O |
| 99 | AD31 | I/O |
| 100 | AD30 | I/O |
| 101 | GND | G |
| 102 | AD29 | I/O |
| 103 | AD28 | I/O |
| 104 | AD27 | I/O |
| 105 | AD26 | I/O |
| 106 | AD25 | I/O |
| 107 | AD24 | I/O |
| 108 | C/BE3# | I/O |
| 109 | VCC | P |
| 110 | IDSEL | I |
| 111 | AD23 | I/O |
| 112 | AD22 | I/O |
| 113 | AD21 | I/O |
| 114 | AD20 | I/O |
| 115 | AD19 | I/O |
| 116 | GND | G |
| 117 | AD18 | I/O |
| 118 | AD17 | I/O |
| 119 | AD16 | I/O |
| 120 | C/BE2# | I/O |
| 121 | FRAME# | I/O |
| 122 | IRDY# | I/O |
| 123 | TRDY# | I/O |
| 124 | GND | G |
| 125 | PCICLK | I |
| 126 | VCC | P |
| 127 | DEVSEL# | I/O |
| 128 | STOP# | I/O |
| 129 | LOCK# | I/O |
| 130 | PERR# | I/O |
| 131 | SERR# | O/OD |
| 132 | PAR | I/O |
| 133 | C/BE1# | I/O |
| 134 | AD15 | I/O |
| 135 | AD14 | I/O |
| 136 | GND | G |
| 137 | AD13 | I/O |
| 138 | AD12 | I/O |
| 139 | AD11 | I/O |
| 140 | AD10 | I/O |
| 141 | AD9 | I/O |
| 142 | AD8 | I/O |
| 143 | C/BE0# | I/O |
| 144 | VCC | P |

**Table 3-3 Alphabetical Pin Cross-Reference List**

| Pin No. | Pin Name | Pin Type |
|---|---|---|
| 9 | AD0 | I/O |
| 8 | AD1 | I/O |
| 6 | AD2 | I/O |
| 5 | AD3 | I/O |
| 4 | AD4 | I/O |
| 3 | AD5 | I/O |
| 2 | AD6 | I/O |
| 1 | AD7 | I/O |
| 142 | AD8 | I/O |
| 141 | AD9 | I/O |
| 140 | AD10 | I/O |
| 139 | AD11 | I/O |
| 138 | AD12 | I/O |
| 137 | AD13 | I/O |
| 135 | AD14 | I/O |
| 134 | AD15 | I/O |
| 119 | AD16 | I/O |
| 118 | AD17 | I/O |
| 117 | AD18 | I/O |
| 115 | AD19 | I/O |
| 114 | AD20 | I/O |
| 113 | AD21 | I/O |
| 112 | AD22 | I/O |
| 111 | AD23 | I/O |
| 107 | AD24 | I/O |
| 106 | AD25 | I/O |
| 105 | AD26 | I/O |
| 104 | AD27 | I/O |
| 103 | AD28 | I/O |
| 102 | AD29 | I/O |
| 100 | AD30 | I/O |
| 99 | AD31 | I/O |
| 21 | CAD0 | I/O |
| 22 | CAD1 | I/O |
| 23 | CAD2 | I/O |
| 24 | CAD3 | I/O |
| 25 | CAD4 | I/O |
| 26 | CAD5 | I/O |
| 27 | CAD6 | I/O |
| 28 | CAD7 | I/O |
| 30 | CAD8 | I/O |
| 31 | CAD9 | I/O |
| 32 | CAD10 | I/O |
| 36 | CAD11 | I/O |
| 37 | CAD12 | I/O |
| 38 | CAD13 | I/O |
| 39 | CAD14 | I/O |
| 40 | CAD15 | I/O |
| 42 | CAD16 | I/O |
| 57 | CAD17 | I/O |
| 58 | CAD18 | I/O |
| 59 | CAD19 | I/O |
| 60 | CAD20 | I/O |
| 61 | CAD21 | I/O |
| 62 | CAD22 | I/O |
| 63 | CAD23 | I/O |
| 66 | CAD24 | I/O |
| 67 | CAD25 | I/O |
| 68 | CAD26 | I/O |
| 69 | CAD27 | I/O |
| 70 | CAD28 | I/O |
| 71 | CAD29 | I/O |
| 72 | CAD30 | I/O |
| 76 | CAD31 | I/O |
| 143 | C/BE0# | I/O |
| 133 | C/BE1# | I/O |
| 120 | C/BE2# | I/O |
| 108 | C/BE3# | I/O |
| 47 | CBLOCK# | I/O |
| 29 | CC/BE0# | I/O |
| 41 | CC/BE1# | I/O |
| 56 | CC/BE2# | I/O |
| 65 | CC/BE3# | I/O |
| 17 | CCD1# | I |
| 94 | CCD2# | I |
| 19 | CCLK0 | O |
| 34 | CCLK1 | O |
| 53 | CCLK2 | O |
| 74 | CCLK3 | O |
| 91 | CCLKRUN# | I |
| 49 | CDEVSEL# | I/O |
| 55 | CFRAME# | I/O |
| 79 | CGNT0# | O |
| 81 | CGNT1# | O |
| 83 | CGNT2# | O |
| 85 | CGNT3# | O |
| 10 | CLKRUN# | I/O |
| 51 | CIRDY# | I/O |
| 92 | CIRQSER | I/O |
| 43 | CPAR | I/O |
| 46 | CPERR# | I/O |
| 78 | CREQ0# | I |
| 80 | CREQ1# | I |
| 82 | CREQ2# | I |
| 84 | CREQ3# | I |
| 77 | CRST# | O |
| 45 | CSERR# | I/O |
| 48 | CSTOP# | I/O |
| 50 | CTRDY# | I/O |
| 35 | C_VCC | P |
| 54 | C_VCC | P |
| 73 | C_VCC | P |
| 18 | CVS1 | I |
| 93 | CVS2 | I |
| 127 | DEVSEL# | I/O |
| 15 | ENVCC3+ INTD# | O |
| 16 | ENVCC5 | O |
| 121 | FRAME# | I/O |
| 7 | GND | G |
| 20 | GND | G |
| 33 | GND | G |
| 44 | GND | G |
| 52 | GND | G |
| 64 | GND | G |
| 75 | GND | G |
| 88 | GND | G |
| 101 | GND | G |
| 116 | GND | G |
| 124 | GND | G |
| 136 | GND | G |
| 97 | GNT# | I |
| 110 | IDSEL | I |
| 12 | INTB# | I/O |
| 122 | IRDY# | I/O |
| 11 | IRQLATCH+ INTA# | I/O |
| 129 | LOCK# | I/O |
| 132 | PAR | I/O |
| 125 | PCICLK | I |
| 86 | PCIRQ0# | I |
| 87 | PCIRQ1# | I |
| 89 | PCIRQ2# | I |
| 90 | PCIRQ3# | I |
| 95 | PCIRST# | I |
| 130 | PERR# | I/O |
| 98 | REQ# | O |
| 131 | SERR# | O/OD |
| 128 | STOP# | I/O |
| 123 | TRDY# | I/O |
| 14 | VCC | P |
| 96 | VCC | P |
| 109 | VCC | P |
| 126 | VCC | P |
| 144 | VCC | P |
| 13 | VENID+ EXTCLK+ INTC# | I/O |

## 3.2 Signal Descriptions

### 3.2.1 Host Interface PCI Signals

| Signal Name | Pin No. | Signal Type | Signal Description |
|---|---|---|---|
| AD[31:0] | 99, 100, 102:107, 111:115, 117:119, 134, 135, 137:142, 1:6, 8, 9 | I/O | **Address and Data Lines 31 through 0:** This bus carries the address during the address phase and the data during the data phase of a PCI cycle. During the address phase these pins are inputs only and during the data phase they are I/Os. |
| C/BE[3:0]# | 108, 120, 133, 143 | I/O | **Bus Command and Byte Enables 3 through 0:** These inputs provide the command type information during the address phase and carry the byte enable information during the data phase. |
| PAR | 132 | I/O | **Parity:** This bit carries parity information for both the address and data phases of PCI cycles. During the address or data write phase of a PCI cycle this pin is an input only. During the data read phase it acts as an output only. |
| PCICLK | 125 | I | **PCI Clock:** Provides timing for all transactions on the host PCI bus; normally 33MHz. This same clock can be used for timing the slot interfaces, or can be divided. The slot interfaces can also run from the alternative EXTCLK input. |
| VENID# | 13 | O | **Drive Vendor ID:** This pin can be used to enable an external tristate buffer to drive vendor ID bits onto the PCI bus. This feature allows system card designers to drive a unique PCI card ID for identification by software. |
| EXTCLK | | I | **External Clock:** Provides alternative clock source for transactions on the slot interface PCI bus. The frequency can be any value but is usually 20MHz or 25MHz. It should be tied low if not used. This pin is automatically sensed just after reset time to determine whether an external clock frequency is being applied. If not, the function defaults to VENID#. |
| INTC# | | I/O | See Section 3.2.4 for interrupt information. |
| CLKRUN# | 10 | I/O | **Clock Run:** Pulled low by any device needing to use the PCI bus. If no devices pull this pin low, the host PCI bus controller is allowed to stop the PCICLK signal. The interrupt logic of the 82C814 uses this signal to request a restart of PCICLK in order to send an interrupt request. |
| IRQLATCH | 11 | I/O | **Interrupt Latch:** For use on chipsets without IRQ driveback capability, the 82C814 logic can drive this line low to drive ISA IRQ lines using an external latch. This pin is also a strap option, refer to Section 5.3 |
| INTA# | | I/O | See Section 3.2.4 for interrupt information. |
| FRAME# | 121 | I/O | **Cycle Frame:** Driven by PCI bus masters to indicate the beginning and duration of an access. |
| IRDY# | 122 | I/O | **Initiator Ready:** Asserted by the PCI bus master to indicate that it is ready to complete the current data phase of the transaction. |
| TRDY# | 123 | I/O | **Target Ready:** Asserted by the PCI bus target (when the 82C814 is a slave) to indicate that it is ready to complete the current data phase of the transaction. PCI-type devices on the slot interfaces return CTRDY# to the 82C814, which in turn drives TRDY# to the host. The 82C814 logic drives TRDY# directly for 82C814 configuration register accesses. |

### 3.2.1 Host Interface PCI Signals (cont.)

| Signal Name | Pin No. | Signal Type | Signal Description |
|---|---|---|---|
| STOP# | 128 | I/O | **Stop:** Used by the target to request that the master stop the current transaction and retry it later. The 82C814 logic uses this mechanism to back-off from a claimed cycle and generate, for example, an SMI through the IRQ driveback cycle. |
| LOCK# | 129 | I/O | **Lock:** Indicates an atomic operation that may require multiple transactions to complete. The signal can be asserted to the 82C814 by any host bus PCI master, and is driven by the 82C814 logic in response to the current slot interface bus master driving its CBLOCK# signal. |
| DEVSEL# | 127 | I/O | **Device Select:** Driven by the 82C814 logic when it decodes its address as the target of the current access via either positive or subtractive decoding. |
| PERR# | 130 | I/O | **Parity Error:** All devices use this signal to report data parity errors during any PCI transaction except a Special Cycle. |
| SERR# | 131 | O/OD | **System Error:** The 82C814 logic uses this line to report address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic. This pin has an open drain output. |
| REQ# | 98 | O | **Bus Request:** The 82C814 logic uses this signal to gain control of the PCI bus. The logic also uses this pin to generate an interrupt driveback request. |
| GNT# | 97 | I | **Bus Grant:** The system grants the bus to the 82C814 chip using this signal. |
| IDSEL | 110 | I | **ID Select:** This signal is the "chip select" for the controller. This input simply connects to one of the upper address lines to select the controller for configuration cycles. |
| PCIRST# | 95 | I | **Reset:** Main chip reset input. |

### 3.2.2 Docking Control and Sense Signals

| Signal Name | Pin No. | Signal Type | Signal Description |
|---|---|---|---|
| CCD1# | 17 | I | **Connection Detect 1 and 2, Voltage Sense 1 and 2:** CCD1-2# and CVS1-2 are used to determine proper dock attachment and to sense its voltage. |
| CCD2# | 94 | I | |
| CVS1 | 18 | I | |
| CVS2 | 93 | I | |
| ENVCC5 | 16 | O | **5.0V VCC Enable:** Used to turn on power to 5.0V dock. |
| ENVCC3 | 15 | O | **3.3V VCC Enable:** Used to turn on power to 3.3V dock. |
| INTD# | | I/O | See Section 3.2.4 for interrupt information. |

### 3.2.3 PCI Docking Interface Pins

| Signal Name | Pin No. | Signal Type | Signal Description |
|---|---|---|---|
| CAD[31:0] | 76, 72:66, 63:57, 42, 40:36, 32:30, 28:21 | I/O | **Multiplexed Address and Data Lines 31 through 0:** These pins are the multiplexed PCI address and data lines. During the address phase, these pins are outputs for PCI slave cycles and inputs for PCI master cycles. During the data phase, these pins are outputs during PCI write cycles and inputs during PCI reads. |

### 3.2.3 PCI Docking Interface Pins (cont.)

| Signal Name | Pin No. | Signal Type | Signal Description |
|---|---|---|---|
| CRST# | 77 | O | **Reset:** Used to reset the docking station PCI bus. This signal defaults to "asserted" until specifically programmed to go high. |
| CC/BE[3:0]# | 65, 56, 41, 29 | I/O | **Bus Command and Byte Enables 3 through 0:** These pins are the multiplexed PCI command and byte enable lines. Normally outputs, these pins are inputs during master cycles. |
| CPAR | 43 | I/O | **Parity:** This signal is an input either during PCI slave cycles for address and write data phases or during PCI master cycle for read data phase; otherwise it is an output. |
| CCLK[3:0] | 74, 53, 34, 19 | O | **Clock 3 through 0:** These pins generate individual clocks to each PCI device on the dock. |
| CFRAME# | 55 | I/O | **Cycle Frame:** The 82C814 drives this signal to indicate the beginning and duration of an access. |
| CIRDY# | 51 | I/O | **Initiator Ready:** The 82C814 drives this signal to indicate its ability to complete the current data phase of the transaction. |
| CTRDY# | 50 | I/O | **Target Ready:** The 82C814 monitors this input from the slot interface slave device to determine when it can complete the cycle. PCI devices on the slots return CTRDY# to the 82C814 which in turn drives host TRDY#. |
| CSTOP# | 48 | I/O | **Stop:** This signal is used by the target to request the master to stop the current transaction. The 82C814 will back-off the current cycle and retry it later. |
| CBLOCK# | 47 | I/O | **Bus Lock:** The 82C814 uses this signal to indicate an atomic operation that may require multiple transactions to complete. |
| CDEVSEL# | 49 | I/O | **Device Select:** This signal is normally an input from the slot interface device claiming the cycle. The 82C814 claims the cycle ahead of time on the host side. |
| CPERR# | 46 | I/O | **Parity Error:** All slot interface devices use this signal to report data parity errors, during any PCI transaction except a Special Cycle. |
| CSERR# | 45 | I/O | **System Error:** All slot interface devices use this signal to report address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic. |
| CEXT_GNT# | 84 | I | **External Arbiter Grant Input:** This signal is asserted by an external arbiter to grant the secondary PCI bus to the 82C814. When using an external arbiter CREQ[2:0]# and CGNT[2:0]# are not functional and should be pulled high. |
| CREQ3# | | I | **Bus Master Request Line 3:** Request/grant signal pairs are provided to accommodate up to four PCI bus masters on the docking station. |
| CREQ[2:0]# | 82, 80, 78 | I | **Bus Master Request Lines 2 through 0:** Request/grant signal pairs are provided to accommodate up to four PCI bus masters on the docking station. |
| CEXT_REQ# | 85 | O | **External Arbiter Request Output:** The 82C814 asserts this signal to request the secondary PCI bus from an external arbiter. When using an external arbiter CREQ[2:0]# and CGNT[2:0]# are not functional and should be pulled high. |
| CGNT3# | | O | **Bus Grant Line 3:** Request/grant signal pairs are provided to accommodate up to four PCI bus masters on the docking station. |
| CGNT[2:0]# | 83, 81, 79 | O | **Bus Grant Lines 2 through 0:** Request/grant signal pairs are provided to accommodate up to four PCI bus masters on the docking station. |

### 3.2.3 PCI Docking Interface Pins (cont.)

| Signal Name | Pin No. | Signal Type | Signal Description |
|---|---|---|---|
| CCLKRUN# | 91 | I/O | CLKRUN signal for docking PCI devices: Pulled low by any device needing the PCI bus. If no devices pull this pin low, the 82C814 logic is allowed to stop its CCLK0-3 outputs. |

### 3.2.4 Interrupt Interface Pins

| Signal Name | Pin No. | Signal Type | Signal Description |
|---|---|---|---|
| PCIRQ0# | 86 | I | **PCI Interrupt 0:** From docking station, routed according to PCICFG 48h |
| PCIRQ1# | 87 | I | **PCI Interrupt 1:** From docking station, routed according to PCICFG 49h |
| PCIRQ2# | 89 | I | **PCI Interrupt 2:** From docking station, routed according to PCICFG 4Ah |
| PCIRQ3# | 90 | I | **PCI Interrupt 3:** From docking station, routed according to PCICFG 4Bh |
| CIRQSER | 92 | I/O | **IRQ Serial**: Single-wire Serial IRQ for docking station devices using serial IRQs |
| INTA# | 11 | I/O | **INTA#:** IRQLATCH reassigned as Primary PCI INTA#. See Table 3-4 for strap options. |
| INTB# | 12 | I/O | **INTB#:** NC pin reassigned as Primary PCI INTB#. See Table 3-4 for strap options. |
| INTC# | 13 | I/O | **INTC#:** VENID# pin reassigned as Primary PCI INTC#. See Table 3-4 for strap options. |
| INTD# | 15 | I/O | **INTD#:** ENVCC3 pin reassigned as Primary PCI INTD#. See Table 3-4 for strap options. |
| IRQSER | | | **IRQ Serial:** INTD# reassigned as IRQSER - provides serial IRQ connection to host bus core logic. Enabled in PCICFG 4Eh. |

### 3.2.5 Power and Ground Pins

| Signal Name | Pin No. | Signal Type | Signal Description |
|---|---|---|---|
| GND | 7, 20, 33, 44, 52, 64,75, 88, 101, 116, 124, 136 | G | **Ground Connection** |
| VCC | 14, 96, 109, 126, 144 | P | **Power Connection:** For Host Interface |
| C_VCC | 35, 54, 73 | P | **Power Connection:** For Docking Interface |

## 3.3 Strap-Selected Interface Options

The 82C814 CardBus Controller can be strapped to operate in one of several different modes depending on its implementation in the system.

Strap options are registered at chip reset time. The selection straps are normally 10k ohm resistors engaged full-time.

The strapping possibilities are listed in Table 3-4.

**Note:** For 5.0V core and PCI host interface designs that use host PCI interrupts INTA#-D#, it may not be possible to strap the 82C814 into 5.0V mode if there is an external pull-up on the host INTA# signal. For these designs it is necessary to program the core voltage to 5.0V by writing PCICFG 5Eh[4] = 1.

**Table 3-4 Strap Options for 82C814 Configurations**

| Strap Selection | Feature | No Strap | Pulled by 10k ohm Resistor at Reset |
|---|---|---|---|
| IRQLATCH (PCI INTA#) Pin 11 | Core Voltage Select | 3.3V Core and PCI host interface (internal pull up) | 5.0V Core and PCI host interface (external 10k ohm pull down) |
| PCI INTB# Pin 12 | PCI Interrupts | Use IRQ Driveback or IRQLATCH# (internal pull down) | Provide INTA#-D# (external 10k ohm pull up) |

## 3.4 Internal Resistors

The 82C814 slot interfaces are provided with pull-up and pull-down resistors internal to the chip. The resistors are active at the times indicated in Table 3-5.

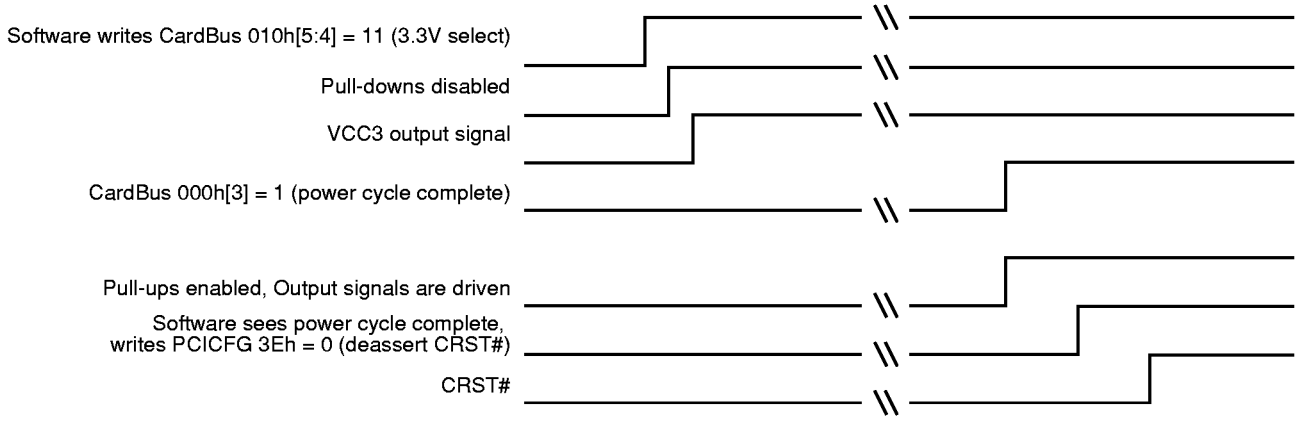
Table 3-5 refers to the chip state with no card inserted, a powered-down card inserted, or a docking station attached.

Figure 3-2 shows the functional timing relationships of software power-up and reset commands to the signals output by the power cycle state machine.

**Table 3-5 Internal Keeper Resistor Scheme**

| Signal Group | | 82C814 Action with No Attachment | 82C814 Action after Detecting Docking Station |
|---|---|---|---|
| Dock Detect: | CCD1-2# | Pull up to core VCC to detect dock insertion/removal | Pull up to core VCC |
| Address/Data: | CAD[31:0]<br>CC/BE[3:0]#<br>CPAR | Pull down | Pull down until interface is powered up |
| Reset: | CRST# | Driven low | Driven according to PCICFG 3Eh[6] |
| Frame: | CFRAME# | Pull down | None |
| PCI Control/Status: | CIRDY#<br>CTRDY#<br>CDEVSEL#<br>CSTOP#<br>CPERR#<br>CBLOCK# | Pull down | None |
| Clock: | CCLK[3:0] | Pull down | Disable pull-down (clock input is always driven) |
| Request: | CREQ[3:0]# | Pull up to card VCC | None |
| Open Drain: | CSERR# | Pull up to card VCC | None |

**Figure 3-2 Power-Up Timing**



# 4.0 Functional Description

## 4.1 OPTi Docking Station Controller Chipset

The OPTi Docking Station solution is comprised of two devices. The minimum configuration requires one chip, the 82C814 part.

- The 144-pin 82C814 Docking Controller handles the signal transfer for a complete PCI bus, including interrupts and clock generation.
- The OPTi PCI-ISA Bridge converts PCI signals back into ISA signals. No OPTi PCI-ISA Bridge is required in the system, but one can be added as an option to support ISA peripherals in an attached docking station that connects through the PCI bus interface. The OPTi PCI-ISA Bridges are discussed in a separate document.

The multiple interface arrangement offers the maximum in system design flexibility.

## 4.2 Chipset Compatibility

Because the OPTi Docking Station Controller Chipset is based on a PCI host interface, it can be used with any PCI-compliant system. ISA DMA may require special software support on non-OPTi systems.

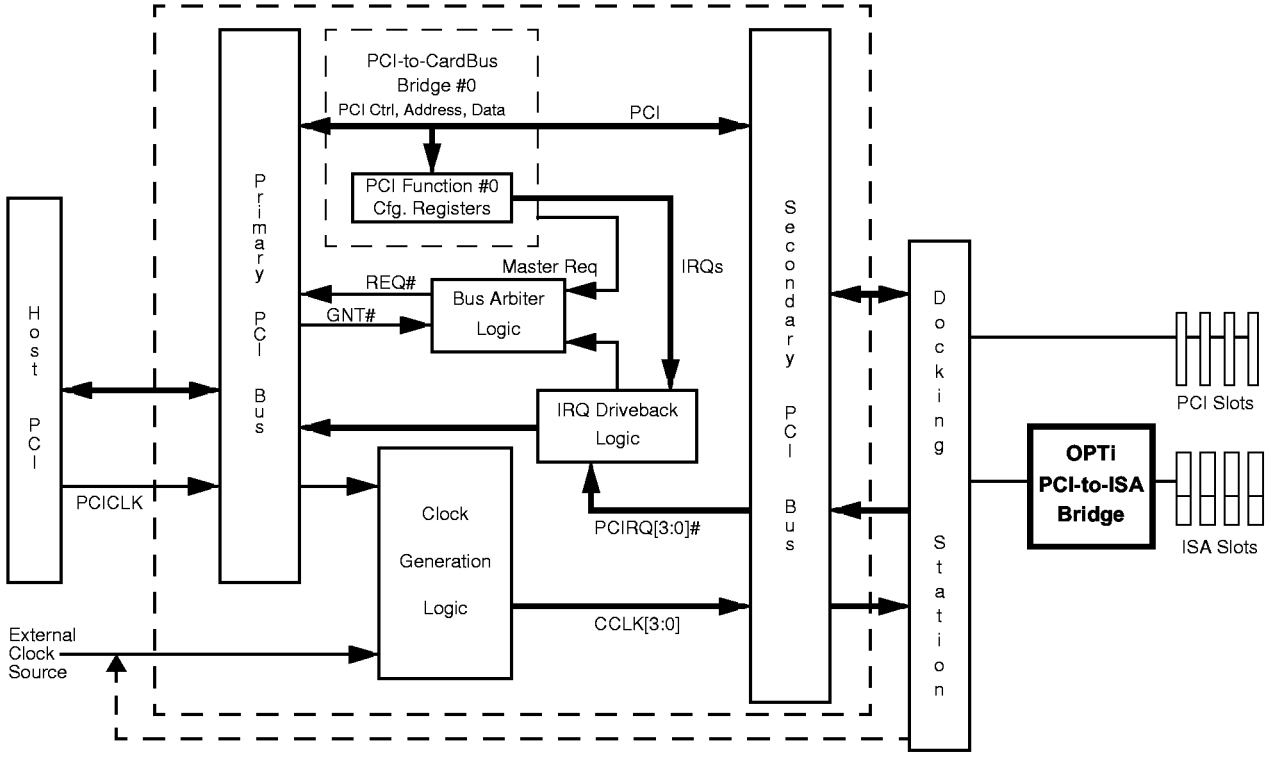
## 4.3 Interface Overview

The OPTi 82C814 Docking Station Controller Chipset uses two independent external interfaces. The terms ***host interface*** and ***docking interface*** are used throughout this document to describe these interfaces.

- The **host interface** provides industry standard PCI signals to the host system. The interface also can be programmed to return interrupt requests from the docking interfaces.
- The **docking interface** duplicates the primary PCI signal set. It is completely isolated from the primary PCI bus.

The interface signal groups used to integrate the OPTi Docking Station Controller Chipset into the standard system are described in the following sections. Figure 4-1 illustrates the interaction of the logic modules of the OPTi Docking Station Controller Chipset.

**Figure 4-1 82C814 Organization**



The logic implements several functional blocks that interact as indicated. The functional blocks shown in the diagram are briefly described below.

- The 82C814 takes its control, address, and data information from its **primary PCI bus**, which is usually controlled by the host PCI interface but can also be controlled by a master on the docking interface.
- The 82C814 logic implements a **PCI-to-PCI (Card Bus) bridge** controlled by **PCI Configuration Registers.** These configuration registers are accessed from the primary PCI bus. Any bus master, including a master on the docking interface, can program these registers. The PCI Configuration Registers consist of standard CardBus registers at indexes 00h-47h and OPTi 82C814 architecture-specific registers at indexes 48h-FFh. Settings in these registers control host interface operations, select architecture-specific settings such as interrupt routing to the host, and provide PCI status to the host on request. The register set is accessed as PCI Function 0 of the 82C814 device.
- The PCI-to-PCI bridge serves to connect the primary PCI bus to an independent secondary PCI bus. It is this secondary bus that interfaces externally to a docking station. If no dock is attached, software can still access the configuration registers for the bridge.
- The **bus arbiter logic** takes master requests for bus ownership for the purpose of giving PCI master control to one of the secondary PCI buses.
- Devices connected to the docking interface can transmit interrupts to the host system directly or through serial IRQs. Docking station PCI devices can generate INTA#, INTB#, INTC#, and INTD# which the 82C814 logic converts to an interrupt.
- **Clock generation logic** is provided to use either the primary PCICLK input for synchronous operation, or an external clock input for asynchronous operation. Four separate output clocks are provided, and can be skew-compensated to adjust for varying board trace lengths.

The logic subsystems of the 82C814 Docking Station Controller are described in detail in the following sections.

## 4.4 Device Type Detection Logic

The 82C814 logic includes attachment detection logic and a power control state machine to determine what type of dock has been attached to the docking interface.

The power control state machine follows the algorithm provided by the CardBus specification, with a slight modification for docking station detection. Table 4-1 lists the device determination rules. Although the state machine follows the rules for CardBus device detection, only docking stations are considered valid attachments.

**Table 4-1 Device Detection (CardBus Rules)**

| CCD2# | CCD1# | CVS2 | CVS1 | Key | Card Type |
|---|---|---|---|---|---|
| GND | Short to CVS1 | Open | Short to CCD1# | LV | 3.3V CardBus |
| Short to CVS2 | GND | Short to CCD2# | GND | LV | 3.3/x.xV CardBus |
| Short to CVS1 | GND | GND | Short to CCD2# | LV | 3.3/x.x/y.yV CardBus |
| Short to CVS2 | GND | Short to CCD2# | Open | LV | x.xV CardBus |
| GND | Short to CVS2 | Short to CCD1# | Open | LV | x.x/y.yV CardBus |
| Short to CVS1 | GND | Open | Short to CCD2# | LV | y.yV CardBus |
| **GND** | **Short to CVS1** | **GND** | **Short to CCD1#** | **--** | **3.3V Docking Station** |
| **GND** | **Short to CVS2** | **Short to CCD1#** | **GND** | **--** | **5.0V Docking Station** |
| GND | GND | Open | Open | 5.0V | 5.0V PCMCIA |
| GND | GND | Open | GND | LV | 3.3V PCMCIA |
| GND | GND | Open | GND | 5.0V | 3.3/5.0V PCMCIA |
| GND | GND | GND | Open | LV | x.xV PCMCIA |
| GND | GND | GND | GND | LV | x.x/3.3V PCMCIA |
| GND | GND | GND | GND | 5.0V | x.x/3.3/5.0V PCMCIA |

## 4.5 Primary PCI Bus

The host interfaces to the 82C814 chip through the primary PCI bus. This bus operates according to PCI standards, including the later addition of the CLKRUN# signal. CLKRUN# is normally controlled by the host, but at certain times can be driven low by the 82C814 chip when the chip is requesting that PCICLK be restarted or sped up. Refer to the PCI Mobile Design Guide for the requirements of CLKRUN#.

CLKRUN# is controlled by PCICFG 50h[2].

## 4.6 PCI-to-CardBus Bridge

The PCI-to-CardBus bridge circuit of the 82C814 chip recognizes the cycle being performed by the current system bus master and responds as required.

### 4.6.1 Configuration Cycle

If the access is a configuration cycle, the PCI bridge simply accesses the local PCI Configuration Register set directly. The PCI cycle controller claims all configuration accesses to PCI Function 0 of the 82C814 chip.

#### 4.6.1.1 Translation Between Type 0 and Type 1 Configuration Cycles

The 82C814 logic converts Type 1 configuration cycles on the host PCI bus to Type 1, Type 0, or a Special Cycle as is typically required of a PCI-to-PCI bridge. However, in a PCI-to-PCI bridge, Type 1 configuration cycles on the secondary PCI bus can be converted only to Type 1 or Special Cycles on the primary bus, never to Type 0.

The 82C814 logic is different from the standard PCI-to-PCI bridge in this regard. The 82C814 allows the secondary to act as a primary. PCICFG 52h[0] is used to enable this feature.

With this feature selected, master devices on the docking station interface can program the PCI configuration registers of the 82C814 (and any other PCI device on the host PCI bus). To do so, the secondary bus master must generate a Type 1 configuration cycle. The 82C814 logic will pass this to the primary as a Type 0 configuration cycle. Since the 82C814 PCI configuration registers sit on the primary, they are also accessible this way. Thus, on the primary the 82C814 acts as both initiator by generating the configuration cycle, and as target by claiming the cycle it just generated.

Note that secondary bus masters can access PCI configuration registers on any primary bus device, not just the 82C814.

**Table 4-2 CLKRUN# Control Bits**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 50h** | | | **PCI Host Feature Control Register** | | | | **Default = 01h** |
| | | | | | CLKRUN# (on host interface):<br>0 = Enabled per PCI<br>1 = Disabled, CLKRUN# tristated | | |

**Table 4-3 Translation Feature Configuration Bit**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 52h** | | | **Docking Feature Control Register 2** | | | | **Default = 0Fh** |
| | | | | | | | Type 1 to Type 0 conversion blocked from secondary to primary:<br>0 = No<br>1 = Yes (Default) |

### 4.6.2 Cycle from Host to Docking Interface

For a cycle from the host to a docking interface with a docking station attached, the PCI bridge resynchronizes the cycle and passes it to the external PCI device. Docking PCI devices can run either synchronously at the host PCI frequency, or asynchronously at any speed using an external clock. The bridge claims the cycle if it falls into one of the ranges programmed in the Window Registers of the PCI Configuration Register set.

### 4.6.3 Master Cycle from Docking Interface

For a master cycle from the docking interface, the 82C814 logic presents the cycle on the host PCI bus as master.

If the cycle is directed to a device on the other docking interface, the 82C814 logic claims the cycle immediately, as a slave, since the address ranges are already programmed into the Base Address Registers for that docking station.

If the cycle is not claimed by the other docking station and no host device claims it, the 82C814 generates a master abort.

### 4.6.4 Inability to Complete a Posted Write

The 82C814 logic provides write posting in both the downstream and upstream PCI directions. There is a special situation that arises when the target of posted write data is unable to complete the transaction. Normally, a target retry or a disconnect will result in the 82C814 logic retrying the access until it has completed the transfer of posted data.

However, after the programmed number of retries has been attempted, the logic must report the error condition back to the host. The 82C814 provides only one mechanism to return the error: the SERR# pin. The host must then decide how to handle the SERR# generation, either by generation of an NMI or some other means.

The 82C814 PCI configuration register set provides a register to program the number of retries before the logic gives up and generates SERR#, as shown in Table 4-4.

### 4.6.5 Cycle Termination by Target

The PCI-to-CardBus bridge logic responds to cycle termination by target devices in various ways for each transaction type being terminated.

#### 4.6.5.1 Posted Write Termination

Retry or Disconnect - The 82C814 logic retries the write cycle at least 256 times, and may continue trying indefinitely, according to the setting of PCICFG 5Eh[2:0]. When the logic reaches the retry limit, it generates SERR# on the master interface. No target abort will be signalled in the PCI Status Register, but software can read 82C814-Specific Register 5Fh to determine whether the retry limit was exceeded.

Target Abort or No Response - The logic generates SERR#+CSERR# on the master interface. Software reads the PCI Status Register to determine that a target abort occurred.

#### 4.6.5.2 Non-Posted Write Termination

Retry, Disconnect, or Target Abort - The logic simply conveys the target response to the initiator.

No Response - If PCICFG 3Eh[5] = 0, the 82C814 logic terminates the cycle to the initiator normally. If bit 3Eh[5] = 1, the logic generates target abort to the initiator.

#### 4.6.5.3 Read (Prefetched or Non-Prefetched) Termination

Retry, Disconnect, or Target Abort - The logic simply conveys the target response to the initiator.

No Response - If PCICFG 3Eh[5] = 0, the 82C814 logic terminates the cycle to the initiator normally and returns FFFFFFFFh as the data read. If bit 3Eh[5] = 1, the logic generates target abort to the initiator.

**Table 4-4 Write Posting Associated Registers**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 5Eh** | | | **Primary Retry Limit Register** | | | | **Default = 07h** |
| | | | | | Retry Limit:<br>These bits relate to the number of times that the 82C814, as a slave, will retry accesses on the primary. If this limit is exceeded, the 82C814 generates SERR# to the host.<br>000=$2^8$ 100=$2^{16}$<br>001=$2^{10}$ 101=$2^{20}$<br>010=$2^{12}$ 110=$2^{24}$<br>011=$2^{14}$ 111= Infinite retries (Default) | | |

**Table 4-4 Write Posting Associated Registers (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 5Fh** | | | **82C814 Retry Count Readback Register (RO)** | | | | **Default = 00h** |
| - This register returns the number of retry attempts made.<br>- More than 256 retries are indicated by FFh.<br>- Used for diagnostic purposes. Read-only.<br>- Separate counts are maintained for primary and secondary. Bit 5Eh[3] selects the count being read back. | | | | | | | |
| **PCICFG 3Eh** | | | | **Bridge Control Register - Byte 0** | | | **Default = 40h** |
| | | Response to master abort on slot interface:<br>0 = Ignore<br>1 = Signal with target abort or SERR# | | | | | |

## 4.7 PCI Docking Station Operation

OPTi docking is based on the CardBus concept: the docking station can be treated like a CardBus card being plugged into or removed from the system at any time. The docking interface is fully isolated and allows the host system to recover in case of problems on the dock.

Windows 98 and NT 5.0 fully supports 82C814 docking. When using other operating systems, BIOS support software is required. The rest of this section describes the basics of the support software needed.

### 4.7.1 Introduction

The 82C814 register set follows the Yenta standard; the registers are virtually the same whether in CardBus mode or in Docking mode. However, there are two differences from a programming point of view.

- A CardBus card can be identified as PCICFG 68h[5:4] = 10. A Docking Station is identified by PCICFG 68h[5:4] = 11.
- A CardBus card has only one interrupt, mapped to PCIRQ0#. A Docking Station has four interrupt pins, mapped through PCIRQ[3:0]#.

When a docking station is attached to the interface, the power control state machine of the 82C814 recognizes the docking station. A docking station is the only valid attachment to the 82C814 chip.

### 4.7.2 Procedure

The docking concept follows the Yenta specification. However, a more flexible set of registers is available for docking that allows eight windows instead of the four offered by Yenta. Either the Yenta window registers (PCICFG 1C-3Bh) or the docking registers (PCICFG 80-BFh) can be used. The docking window registers also allow finer control over window sizes than do the Yenta window registers.

### 4.7.3 Initial Setup

The following programming should be performed at system initialization time, and does not need to be repeated.

- **Enable Host Chipset Bus Preemption.** Write SYSCFG 1Eh[3] = 1 on the Viper-N+ and FireStar chipsets.
- **Establish Status Change Interrupt.** Write PCICFG 4Ch with the IRQ that should be generated when the dock is attached or removed. Any available IRQ can be used. On FireStar, selecting IRQ2 will generate an SMI and IRQ13 will generate an NMI. These selections are not available on Viper-N+. However, normal IRQs can be programmed on the Viper-N+ chipset to generate an SMI or NMI if desired, through the following approach:

  1. Use SYSCFG 64h and A4h to select the IRQ to use for SMI generation.
  2. Write SYSCFG 57h[6] = 1 to enable INTRGRP to generate PMI#6 when the selected IRQ goes active.
  3. Write SYSCFG 59h[5:4] = 11 to enable PMI#6 to generate SMI.

- **Establish IRQ Driveback Address.** Write PCICFG 54-57h with an I/O address to use for IRQ driveback. The default value is 33333330h, but any unused value is fine. Ideally the address should be greater than FFFFh to prevent conflicts with ISA I/O address space.

  Write the same value to the IRQ Driveback registers in the host chipset (Viper-N+ or FireStar). The registers are at the same PCI offset, but different PCI device: PCIDV1 54-57h.

- **Select PCI Bus Number of Docking Station.** PCICFG 19h selects the PCI bus number on the secondary side of the bridge. A value of 01h is typical.
- **Select Total Number of Downstream Buses.** PCICFG 1Ah selects the number of the last downstream PCI bus. A value of 01h is typical.
- **Program the Time-out Value.** PCICFG 1Bh should be set to FFh.
- **Program the Latency Timer**. PCICFG 0Dh should be set to FFh.
- **Select the Status Change Events.** PCICFG 64h[3:0] select the events that will cause a status change interrupt in the future. Typically writing PCICFG 64h = 06h is adequate. Also write PCICFG 60h = 0Fh to clear any pending events.

Table 4-5 summarizes the typical settings for system initialization.

### 4.7.4 Action Upon Attachment of Dock

At idle, with no device attached, the CD1-2# pins are pulled high internal to the 82C814 chip. CVS1-2 are driven low. All other interface lines are pulled low at this time; the docking interface itself can remain unpowered. The 82C814 monitors the CD1-2 lines to determine a docking event.

When a docking station is attached, the 82C814 sees CD1# and CD2# go low, because the docking station connector has these lines hard-wired as follows:

- CD1# is connected to CVS1 for a 3.3V docking station, or to CVS2 for a 5.0V docking station.
- CD2# is connected to ground.

The 82C814 card detection sequencer waits for the time set in PCICFG 50h[3], then performs a test on these lines to determine the type of device attached. Once the test is complete, the 82C814 generates an interrupt to the IRQ configured in PCICFG 4Ch.

**Table 4-5 Summary of Typical Settings (using IRQ5 for SMI)**

| Register | Byte 3 | Byte 2 | Byte 1 | Byte 0 |
|---|---|---|---|---|
| **82C814 Register** | | | | |
| PCICFG 4Ch | -- | -- | -- | 15h (IRQ5) |
| PCICFG 54h | 33h | 33h | 33h | 30h |
| PCICFG 0Ch | -- | -- | FFh | -- |
| PCICFG 18h | FFh | 01h | 01h | 00h |
| PCICFG 64h | -- | -- | -- | 06h |
| PCICFG 60h | -- | -- | -- | 0Fh |
| **Viper N+ Register (assuming IRQ5)** | | | | |
| PCIDV1 54h | 33h | 33h | 33h | 30h |
| SYSCFG 64h | -- | -- | -- | ****1***b(IRQ5) |
| SYSCFG 57h | -- | -- | -- | 01**0000b |
| SYSCFG 59h | -- | -- | -- | **11****b |
| SYSCFG 1Eh | -- | -- | -- | ****1***b |

* These bits should be read first, then written to the same value.

## 4.8 Status Change Service Routine

Interrupt or SMI service software should perform the following steps:

1. Read PCICFG 68h[7, 5:4] to determine whether a docking station has been recognized.

   Test: PCICFG 68h[7] = 0?
   Yes - Device recognized.
   No - Device not recognized. Go to "Retest" section.

   Test: PCICFG 68[5:4] = 11?
   Yes - Docking station recognized.
   No - Not a docking station. Exit procedure so that CardBus software can handle event.

2. Read PCICFG 68h[2:1]. The card detection sequencer drives CVS1 and CVS2 low after detection, so CD1-2# will stay low.

   Test: PCICFG 68h[2:1] = 00?
   Yes - Docking confirmed.
   No - A non zero value indicates that the connection is not valid or that an undock event has taken place.

3. Read PCICFG 60h to determine the event that caused the interrupt. Write this same value back to the register to clear these events, and cause the IRQ line that was active to go inactive. Also clear PMI event on host chipset if this was an SMI.

4. Test: Was docking confirmed in step 2?

   Yes - Go to "Docking Event" section.
   No - Force a retest by writing PCICFG 6Dh[6] = 1, and go to step 1. If this is the second time through, then proceed to "Undocking Event" section.

### 4.8.1 Docking Event

1. Read PCICFG 69h to determine the docking station voltage.

2. Power up the interface by writing PCICFG 70h[6:4] with the correct VCC value. PCICFG 70h is typically written to 20h for a 5.0V docking station.

3. Read PCICFG 68h again to check power cycling.

   Test: PCICFG 68h[3] = 1?
   Yes - Continue to next step.
   No - There is a problem. Check PCICFG 69h[1] to see if the VCC value chosen is allowable. If necessary, force a retest and then start over at step 1.

4. Select PCICLK skew through PCICFG 52h[7:4]. This value will have to be determined according to the design of the docking station. Depending on the type of PCICLK routing used on the docking station, the internal clock may need to be skewed 1-15ns.

5. Write PCICFG 3Eh[6] = 0 to deassert PCIRST# to the dock.

The Docking Station devices can now be configured in the usual manner for PCI devices.

### 4.8.2 Undocking Event

The following step should be followed if an undock event has been detected.

1. Test whether PCICFG 69h[0] = 1. If so, data may have been lost in the undocking event.

On an undock event, no other steps are necessary. The controller automatically powers down the dock, tristates the interface, and asserts the CRST# line.

### 4.8.3 Notes on Undocking

When undocking, the user can notify the system software (Windows 95) first so that the system software can turn off the 82C814 docking side to make a graceful undock. This is the safest scheme to implement but is not always practical in a real system because of cost.

If hot undocking is required without notifying the system software, shorter CD1-2# pins are required on the docking connector. The CD1-2# pins will change first. The 82C814 will complete the current cycle on the secondary, and will not attempt to start another.

The undocking event generates an interrupt to the system, so that software can check to determine if any posted write data was left in the FIFO. PCICFG 5Fh returns the number of retries attempted in flushing the FIFO, which can be used to determine whether any data was left after the hot undock.

### 4.8.4 Retest

Whenever the result of a test is ambiguous, software should force the controller to retest the detection pins. Force a retest by writing PCICFG 6Dh[6] = 1, then start the full service routine over again. If after several times through this retest sequence the status cannot be determined, assume an "undocked" state.

### 4.8.5 PCI Clock Buffering

The 82C814 logic provides register settings PCICFG 52h[7:4] to compensate for trace delays. Some compensation is generally required. Table 4-6 highlights the register used for compensating trace delays.

**Table 4-6 Register used to Delay Internal PCICLK to Compensate for Trace Delays**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 52h** | | | **Docking Feature Control Register 2** | | | | **Default = 0Fh** |
| Secondary PCICLK Skew:<br>This value selects the approximate delay, in nanoseconds, that the internal secondary PCICLK must be skewed in order to compensate for external buffer delays.<br>0000 = No delay 0001 = 1ns 0010 = 2ns<br>.....<br>1101 = 13ns 1110 = 14ns 1111 = 15ns | | | | | | | |

## 4.9 Interrupt Support

The 82C814 supports a total of three interrupt schemes from the secondary PCI bus.

1. **PCI** interrupts INTA#, INTB#, INTC#, and INTD# can be mapped internally to system PCIRQ[3:0]# lines.
2. **PCI IRQ driveback** cycles can generate any ISA interrupt. The OPTi PCI-ISA Bridge uses this scheme to generate interrupts in a parallel format back to the host controller via the 82C814 chip.
3. The **Compaq Serial IRQ** scheme uses a single wire, IRQSER, along with the PCICLK to transmit interrupts in a serial format.

The available schemes are described below.

### 4.9.1 PCI INTx# Implementation

The PCI INTA#, INTB#, INTC#, and INTD# lines can be mapped to any of the primary side PCIRQ[3:0]# lines. PCICFG 48-4Ch provide controls for this mapping.

### 4.9.2 IRQ Driveback Logic

A detailed overview of the IRQ driveback cycle is provided in Appendix A. The logic used to implement this mechanism is relatively simple. The trigger events for a driveback cycle are any transition on an interrupt line, or an SMI event as enabled by the 82C814 configuration registers. The request goes to the Request Arbiter logic, which always gives the driveback cycle top priority. Once the REQ# pin is available, the Request Arbiter asserts REQ# on behalf of the IRQ Driveback logic and toggles REQ# according to the driveback protocol discussed in Appendix A.

Once the host PCI controller returns GNT#, the driveback logic writes to the IRQ driveback address location specified in the PCI configuration registers as shown in Appendix A.

### 4.9.3 Compaq Serial IRQ Implementation

The 82C814 chip supports the Compaq standard of Serial IRQs. This one wire approach is very compact compared to the Intel two-wire approach, but if two devices on the line want to share the same interrupt, there may be brief contention since both devices drive the line low on one clock and high on the clock that immediately follows. Because of this contention, OPTi cannot guarantee against chip hardware failure if interrupts are shared in this mode.

The Compaq Serial IRQ scheme requires the register bits. shown in Table 4-7.

**Table 4-7 Compaq SIRQ Control Bits**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 4Eh** | | | **Serial IRQ Control Register 1** | | | | **Default = 00h** |
| Compaq SIRQ HALT mode request:<br>0 = Active<br>1 = Halt | Compaq SIRQ QUIET mode request:<br>0 = Continuous<br>1 = Quiet | | Compaq SIRQ data frame slots. Change only when the Serial IRQ logic is disabled or in Halt state.<br>0 = 17 slots<br>1 = 21 slots | Compaq SIRQ Start frame width in PCI clocks. Change this setting only when Serial IRQ is disabled or in Halt state.<br>00 = 4 PCI clocks<br>01 = 6 PCI clocks<br>10 = 8 PCI clocks<br>11 = Reserved | | | Compaq SIRQ (Compaq Serial IRQ scheme):<br>0 = Disable<br>1 = Enable |
| **PCICFG 4Fh** | | | **Serial IRQ Control Register 2** | | | | **Default = 00h** |
| Compaq SIRQ in HALT state (RO)?<br>0 = No<br>1 = Yes | Compaq SIRQ in QUIET state (RO)?<br>0 = No<br>1 = Yes | | | | | | |

**QUIET** - PCICFG 4Eh[6] requests the next Serial IRQ cycle to be Continuous or Quiet mode. In mobile applications, use Continuous mode only. This is to guarantee that the host gains control of the Serial IRQ for suspend and APM stop clock. In application where the PCI clock never stops, use either mode. PCICFG 4Fh[6] can be read to determine the current state of the logic.

**HALT** - PCICFG 4Eh[7] requests a temporary halt of the Serial IRQ controller as soon as the current cycle has returned to Idle state. Once in Halt state, the Serial IRQ configuration can be changed. After the logic has been put in Halt state, upon clearing this bit the logic will return to Continuous mode. PCICFG 4Fh[7] can be read to determine the current state of the logic.

#### 4.9.3.1 Operation

The Compaq Serial IRQ protocol requires one additional PCI sustained Tri-State pin, the IRQSER signal. For detailed Serial IRQ operation, refer to the "Serialized IRQ for PCI Systems" specification.

After setting PCICFG 4Eh[0] = 1 to enable Compaq Serial IRQ (CSIRQ) mode, the CSIRQ controller initiates a Continuous mode Start frame. During the Data frame, the CSIRQ logic samples the IRQSER input for the corresponding SMI, IOCHCK#, and IRQ values, and then passes the sampled values to the primary.

At the end of the Data frame, the CSIRQ controller will sample the QUIET and HALT bits to determine whether the next Compaq Serial IRQ cycle will be Continuous mode, Quiet mode, or a temporary Halt state.

- If the next cycle is sampled to be Continuous mode, IRQSER is asserted for three PCI clocks. Once the logic enters Idle state, it checks whether the PMU stop PCI clock request is pending. If so, the CSIRQ logic will stay in the Idle state until the PMU request is removed.
- If the next cycle is sampled to be Quiet mode, IRQSER is asserted for two PCI clocks. Once the logic enters Idle state, it samples the IRQSER input to begin the Quiet mode cycle. Since the 82C814 has no control of the Start frame, this mode is not recommended for mobile application.
- If the HALT bit is sampled active, then the CSIRQ logic asserts IRQSER for three PCI clocks to tell all the Serial IRQ devices that next cycle will be Continuous mode; the logic then enters Halt state. In Halt state, CSIRQ configuration can be changed. Clearing the HALT bit will immediately cause a Continuous mode Start frame to be generated.

Once enabled, the Compaq Serial IRQ logic operates all the time when docked; no clock stop synchronization is needed.

# 5.0 82C814 Register Set

The 82C814 Docking Controller chip provides a single group of programming registers, PCI-to-CardBus Bridge 0 Register Group, accessed through a PCI Configuration Cycle to Function 0 of the chip. Consists of CardBus Controller Base Register Group at PCICFG 00h-4Fh, 82C814-specific registers at 50h-5Fh, CardBus Control and Status Register Group at 60h-7Fh, and Docking Station Window Register Group at 80h-FFh. Note that the CardBus Control and Status Register Group can also be accessed in system memory space.

This register group is defined in the following subsections.

## 5.1 Register State on Device Removal

As a general rule, all PCI configuration registers default to their power-on reset value when the card or docking station is disconnected from the interface (CCD1# and CCD2# both high). However, the 82C814-specific registers at PCICFG 48h-5Fh control global configuration and remain set to their programmed values even after a device is removed.

## 5.2 Base Register Group

The registers below represent the standard group required for PCI peripheral device identification and configuration for a PCI-to-CardBus bridge.

**Note:** In the tables that follow, all bits are R/W and their default value is zero, unless otherwise specified. R/W = Read/Write, RO = Read-only, and WO = Write-only

**Table 5-1 Base Register Group - PCICFG 00h-4Fh**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 00h** | | | **Vendor Identification Register (RO) - Byte 0** | | | | **Default = 45h** |
| **PCICFG 01h** | | | **Vendor Identification Register (RO) Byte 1** | | | | **Default = 10h** |
| **PCICFG 02h** | | | **Device ID (RO) - Byte 0** | | | | **Default = 14h** |
| **PCICFG 03h** | | | **Device ID (RO) - Byte 1** | | | | **Default = C8h** |
| **PCICFG 04h** | | | **PCI Command Register - Byte 0** | | | | **Default = 04h** |
| Address/data stepping:<br>0 = Disable (always) | PERR# generation:<br>0 = Disable<br>1 = Enable | VGA palette snoop:<br>0 = Disable<br>1 = Enable | Mem write and Invalidate (RO):<br>0 = Disable (always) | Special Cycle (RO):<br>0 = Disable (always) | Bus master by docking interfaces:<br>1 = Enable (always) | Respond to PCI mem accesses:<br>0 = No<br>1 = Yes | Respond to PCI I/O accesses:<br>0 = No<br>1 = Yes |
| **PCICFG 05h** | | | **PCI Command Register - Byte 1** | | | | **Default = 00h** |
| Reserved: Write bits as read. | | | | | | Fast back-to-back (RO):<br>0 = Disable (always) | SERR# generation:<br>0 = Disable<br>1 = Enable |
| **PCICFG 06h** | | | **PCI Status Register - Byte 0** | | | | **Default = 10h** |
| Fast back-to-back capability (RO):<br>0 = No (always) | Reserved (RO) | | PCI Power Management Capability (RO)<br>1 = Yes (always) | Reserved (RO) | | | |
| **PCICFG 07h** | | | **PCI Status Register - Byte 1** | | | | **Default = 02h** |
| Parity error:<br>0 = No<br>1 = Yes<br>Write 1 to clear | System error:<br>0 = No<br>1 = Yes<br>Write 1 to clear | Received master abort:<br>0 = No<br>1 = Yes<br>Write 1 to clear | Received target abort:<br>0 = No<br>1 = Yes<br>Write 1 to clear | Signalled target abort:<br>0 = No<br>1 = Yes<br>Write 1 to clear | DEVSEL# timing (RO):<br>00 = Fast<br>01 = Medium (always)<br>10 = Slow<br>11 = Reserved | | PERR# active as master:<br>0 = No<br>1 = Yes<br>Write 1 to clear |

**Table 5-1 Base Register Group - PCICFG 00h-4Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 08h** | | | **Revision Register (RO) Revision 1.0** | | | | **Default = 10h** |
| **PCICFG 09h** | | | **Programming Interface Class Code Register (RO)** | | | | **Default = 00h** |
| **PCICFG 0Ah** | | | **Class Code Register (RO) - Byte 0** | | | | **Default = 07h** |
| Subclass Code bits: = 07h (PCI-to-Cardbus Bridge) | | | | | | | |
| **PCICFG 0Bh** | | | **Class Code Register (RO) - Byte 1** | | | | **Default = 06h** |
| Base Class Code bits: = 06h (Bus Bridge) | | | | | | | |
| **PCICFG 0Ch** | | | **Cache Line Size Register** | | | | **Default = 00h** |
| Not implemented | | | | | | | |
| **PCICFG 0Dh** | | | **Latency Timer Register** | | | | **Default = 00h** |
| Indicates the time-out value for the primary PCI interface. | | | | | | | |
| **PCICFG 0Eh** | | | **Header Type Register** | | | | **Default = 02h** |
| Multi-function device (RO): 0 = No (always) | Layout type for 10-3Fh bytes bits [6:0] = 02h (PCI-to-CardBus Header Layout) | | | | | | |
| **PCICFG 0Fh** | | | **BIST Register** | | | | **Default = 00h** |
| Not implemented | | | | | | | |
| **PCICFG 10h** | | | **CardBus Base Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| CardBus Socket Status and Control Base Address Bits:<br>- The 32-bit Cardbus Base Address Register selects the starting address in memory space of the CardBus socket status and control registers.<br>- Actual register addresses are calculated by adding the MEMOFST of the register to this base address.<br>- Bits [11:0] are read-only and are always 0, to indicate that the registers occupy 4KB of non prefetchable system memory space and starts on a 4KB boundary. | | | | | | | |
| **PCICFG 11h** | | | **CardBus Base Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 12h** | | | **CardBus Base Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG 13h** | | | **CardBus Base Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 00h** |
| **PCICFG 14h** | | | **Capabilities Pointer (RO)** | | | | **Default = F0h** |
| Indicates the offset in the PCICFG space for the location of the first item in the Capabilities Linked List. This location is PCICFG F0h. | | | | | | | |
| **PCICFG 15h** | | | **Reserved** | | | | **Default = 00h** |

**Table 5-1 Base Register Group - PCICFG 00h-4Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 16h** | | | **PCI Secondary Bus Status Register - Byte 0** | | | | **Default = 00h** |
| Fast back-to-back capability on docking interface PCI bus (RO):<br>0 = No (always) | Reserved (RO) | | | | | | |
| **PCICFG 17h** | | | **PCI Secondary Bus Status Register - Byte 1** | | | | **Default = 02h** |
| Parity error on docking interface PCI bus:<br>0 = No<br>1 = Yes<br>Write 1 to clear | Received system error on docking interface PCI bus:<br>0 = No<br>1 = Yes<br>Write 1 to clear | Received master abort on docking interface PCI bus (RO):<br>0 = No<br>1 = Yes | Received target abort on docking interface PCI bus (RO):<br>0 = No<br>1 = Yes | Signalled target abort on docking interface PCI bus:<br>0 = No<br>1 = Yes<br>Write 1 to clear | DEVSEL# timing on docking interface PCI bus (RO):<br>00 = Fast<br>01 =Medium (always)<br>10 = Slow<br>11 = Reserved | | PERR# active as master on docking interface PCI bus (RO):<br>0 = No<br>1 = Yes |

**PCICFG 18h** **Primary PCI Bus Number Register** **Default = 00h**

- Indicates the number of the PCI bus to which the host interface of the 82C814 chip is connected.
- Defaults to 0.
- The logic uses this value to determine whether Type 1 configuration transactions on the docking interface should be converted to Special Cycle transactions on the host interface.

**PCICFG 19h** **Secondary PCI Bus Number Register** **Default = 00h**

- Indicates the number of the PCI bus to which the docking interface of the 82C814 chip is connected.
- Defaults to 0.
- The logic uses this value to determine whether Type 1 configuration transactions on the host interface should be converted to Type 0 transactions on the docking interface.

**PCICFG 1Ah** **Subordinate Bus Number Register** **Default = 00h**

- Indicates the number of the highest-numbered PCI bus on the docking interface side.
- The 82C814 logic uses this value in conjunction with the Secondary Bus Number to determine when to respond to Type 1 configuration transactions on the host interface and pass them onto the docking interface.
- Defaults to 0.

**PCICFG 1Bh** **Latency Timer Register** **Default = 00h**

Indicates the time-out value for the docking interface.

**PCICFG 1Ch** **Memory Window 0 Base Address Register - Byte 0: Address Bits [7:0]** **Default = 00h**

Memory Window 0 Base Address Bits:

- The 32-bit Memory Window 0 Base Address Register selects the start address of one of two possible CardBus memory windows to the slot interface.
- Bits [11:0] are read-only and are always 0.
- The memory windows are globally enabled by bit 04h[1] (Command Register).
- Prefetching is enabled by bit 3Fh[0] (Bridge Control Register) and defaults to "enabled."
- The Limit address can be set below the Base address to individually disable a window.

**PCICFG 1Dh** **Memory Window 0 Base Address Register - Byte 1: Address Bits [15:8]** **Default = F0h**

**PCICFG 1Eh** **Memory Window 0 Base Address Register - Byte 2: Address Bits [23:16]** **Default = FFh**

**PCICFG 1Fh** **Memory Window 0 Base Address Register - Byte 3: Address Bits [31:24]** **Default = FFh**

**Table 5-1 Base Register Group - PCICFG 00h-4Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 20h** | | | **Memory Window 0 Limit Address Register - Byte 0: Address Bits [7:0** | | | | **Default = 00h** |
| Memory Window 0 Limit Address Bits:<br>- The 32-bit Memory Window 0 Limit Address Register selects the end address of Memory Window 0.<br>- Bits [11:0] are read-only and are always 0.<br>- The minimum window size is always 4KB. | | | | | | | |
| **PCICFG 21h** | | | **Memory Window 0 Limit Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 22h** | | | **Memory Window 0 Limit Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG 23h** | | | **Memory Window 0 Limit Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 00h** |
| **PCICFG 24h** | | | **Memory Window 1 Base Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| Memory Window 1 Base Address Bits:<br>- The 32-bit Memory Window 1 Base Address Register selects the start address of one of two possible CardBus memory windows to the slot interface.<br>- Bits [11:0] are read-only and are always 0.<br>- The memory windows are globally enabled by bit 04h[1] (Command Register).<br>- Prefetching is enabled by bit 3Fh[1] (Bridge Control Register) and defaults to "enabled."<br>- The Limit address can be set below the Base address to individually disable a window. | | | | | | | |
| **PCICFG 25h** | | | **Memory Window 1 Base Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = F0h** |
| **PCICFG 26h** | | | **Memory Window 1 Base Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = FFh** |
| **PCICFG 27h** | | | **Memory Window 1 Base Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = FFh** |
| **PCICFG 28h** | | | **Memory Window 1 Limit Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| Memory Window 1 Limit Address Bits:<br>- The 32-bit Memory Window 1 Limit Address Register selects the end address of Memory Window 1.<br>- Bits [11:0] are read-only and are always 0.<br>- The minimum window size is always 4KB. | | | | | | | |
| **PCICFG 29h** | | | **Memory Window 1 Limit Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 2Ah** | | | **Memory Window 1 Limit Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG 2Bh** | | | **Memory Window 1 Limit Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 00h** |
| **PCICFG 2Ch** | | | **I/O Window 0 Base Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| I/O Window 0 Base Address Bits:<br>- The 32-bit I/O Window 0 Base Address Register selects the start address of one of two possible CardBus I/O windows to the slot interface.<br>- The I/O windows are globally enabled by bit 04h[0] (Command Register). | | | | | | RO:<br>Always returns 0. | Decoding:<br>0 = 16-bit (AD[31:16] = 0)<br>1 = 32-bit |
| **PCICFG 2Dh** | | | **I/O Window 0 Base Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = F0h** |
| **PCICFG 2Eh** | | | **I/O Window 0 Base Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = FFh** |
| **PCICFG 2Fh** | | | **I/O Window 0 Base Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = FFh** |

**Table 5-1 Base Register Group - PCICFG 00h-4Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 30h** | | | **I/O Window 0 Limit Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| I/O Window 0 Limit Address Bits:<br>- The 32-bit I/O Window 0 Limit Address Register selects the end address of I/O Window 0.<br>- The minimum window size is always 4 bytes. | | | | | | RO:<br>Always returns 0. | |
| **PCICFG 31h** | | | **I/O Window 0 Limit Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 32h** | | | **I/O Window 0 Limit Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG 33h** | | | **I/O Window 0 Limit Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 00h** |
| **PCICFG 34h** | | | **I/O Window 1 Base Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| I/O Window 1 Base Address Bits:<br>- The 32-bit I/O Window 1 Base Address Register selects the start address of one of two possible CardBus I/O windows to the slot interface.<br>- The I/O windows are globally enabled by bit 04h[0] (Command Register). | | | | | | RO:<br>Always returns 0. | Decoding:<br>0 = 16-bit (AD[31:16] = 0)<br>1 = 32-bit |
| **PCICFG 35h** | | | **I/O Window 1 Base Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = F0h** |
| **PCICFG 36h** | | | **I/O Window 1 Base Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = FFh** |
| **PCICFG 37h** | | | **I/O Window 1 Base Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = FFh** |
| **PCICFG 38h** | | | **I/O Window 1 Limit Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| I/O Window 1 Limit Address Bits:<br>- The 32-bit I/O Window 1 Limit Address Register selects the end address of I/O Window 1.<br>- The minimum window size is always 4 bytes. | | | | | | RO:<br>Always returns 0. | |
| **PCICFG 39h** | | | **I/O Window 1 Limit Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 3Ah** | | | **I/O Window 1 Limit Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG 3Bh** | | | **I/O Window 1 Limit Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 00h** |
| **PCICFG 3Ch** | | | **Interrupt Line Register for Status Change** | | | | **Default = 00h** |
| - This register is readable and writable per the PCI specification.<br>- The logic does not use the value written to this register. | | | | | | | |
| **PCICFG 3Dh** | | | **Interrupt Pin Register for Status Change** | | | | **Default = 01h** |
| RO:<br>- This register reflects the value written to PCICFG 4Ch.<br>- It defaults to 01h, selecting PCIRQ0# for the status change (docking station attach/detach) interrupt.<br>- If PCICFG 4Ch is written to select an ISA interrupt or no interrupt, this register returns 00h. | | | | | | | |
| **PCICFG 3Eh** | | | **Bridge Control Register - Byte 0** | | | | **Default = 40h** |
| Reserved | Force CRST# cycling on slot interface:<br>0 = CRST# high<br>1 = Assert CRST# (Default) | Response to master abort on slot interface:<br>0 = Ignore<br>1 = Signal with target abort or SERR# | Reserved:<br>Write as read. | Pass VGA addresses A0000-BFFFFh, 3B0-3BBh, 3C0-3DFh:<br>0 = No<br>1 = Yes | Reserved | Forwarding of SERR# from slot interface to primary PCI bus:<br>0 = Disable<br>1 = Enable | Response to parity errors on slot interface:<br>0 = Ignore<br>1 = Enable |

**Table 5-1 Base Register Group - PCICFG 00h-4Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 3Fh** | | | **Bridge Control Register - Byte 1** | | | | **Default = 03h** |
| Reserved. Write as read. | | | | | Write posting:<br>0 = Disable<br>1 = Enable | Memory Window 1 prefetch:<br>0 = Disable<br>1 = Enable (Default) | Memory Window 0 prefetch:<br>0 = Disable<br>1 = Enable (Default) |
| **PCICFG 40h** | | | **Subsystem Vendor Register - Byte 0: Bits [7:0]** | | | | **Default = 00h** |
| Subsystem Vendor Bits:<br>- The chipset responds to reads of this register with the first value written. The register can be written only once, then becomes read only.<br>- If the option is selected, the EXTCLK pin can be used as DRVVENID# to enable external logic to drive this data onto the bus. In this case, the chipset claims the access but does not drive any data. | | | | | | | |
| **PCICFG 41h** | | | **Subsystem Vendor Register - Byte 1: Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 42h** | | | **Subsystem ID Register - Byte 0: Bits [7:0]:** | | | | **Default = 00h** |
| Subsystem ID<br>- The chipset responds to reads of this register with the first value written. The register can be written only once, then becomes read only.<br>- If the option is selected, the EXTCLK pin can be used as DRVVENID# to enable external logic to drive this data onto the bus. In this case, the chipset claims the access but does not drive any data. | | | | | | | |
| **PCICFG 43h** | | | **Subsystem ID Register - Byte 1: Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 44h - 47h** | | | **Reserved** | | | | **Default = 00h** |
| **PCICFG 48h** | | | **Docking PCIRQ0# Interrupt Assignment Register** | | | | **Default = 01h** |
| Reserved | | | Using OPTi IRQ driveback mechanism:<br>Docking Interrupt Assignment (PCIRQ0# Default) - Interrupts from the docking PCIRQ0# pin are mapped to this interrupt. Note that if an IRQ (an edge-mode interrupt) is selected, this IRQ must be programmed to Level mode on the host chipset.<br>**Level Mode:**<br>00000 = Disabled<br>00001 = PCIRQ0# (Default)<br>00010 = PCIRQ1#<br>00011 = PCIRQ2#<br>00100 = PCIRQ3#<br>00101-01111 = Rsrvd<br>**Edge Mode: (Viper-N+)**<br>10000 = IRQ0<br>10001 = IRQ1<br>10010 = IRQ2<br>10011 = IRQ3<br>10100 = IRQ4<br>10101 = IRQ5<br>10110 = IRQ6<br>10111 = IRQ7<br>11000 = IRQ8<br>11001 = IRQ9<br>11010 = IRQ10<br>11011 = IRQ11<br>11100 = IRQ12<br>11101 = IRQ13<br>11110 = IRQ14<br>11111 = IRQ15 | | | | |
| Reserved | | | | | Using Host PCI INTA#-D# (PCICFG 50h[6]=1)<br>000 = Disabled<br>001 = INTA# (default)<br>010 = INTB#<br>011 = INTC#<br>100 = INTD#<br>101-111 = Reserved | | |

**Table 5-1 Base Register Group - PCICFG 00h-4Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|

**PCICFG 49h — Docking PCIRQ1# Interrupt Assignment Register — Default = 02h**

Reserved

Using OPTi IRQ driveback mechanism:

Docking PCIRQ1# Interrupt Assignment (PCIRQ1# Default) - Interrupts from the docking PCIRQ1# pin are mapped to this interrupt. Note that if an IRQ (an edge-mode interrupt) is selected, this IRQ must be programmed to Level mode on the host chipset.

**Level Mode:**

| | | |
|---|---|---|
| 00000 = Disabled | 00010 = PCIRQ1# (Default) | 00100 = PCIRQ3# |
| 00001 = PCIRQ0# | 00011 = PCIRQ2# | 00101-01111 = Rsrvd |

**Edge Mode: (Viper-N+)**

| | | |
|---|---|---|
| 10000 = IRQ0 | 10110 = IRQ6 | 11011 = IRQ11 |
| 10001 = IRQ1 | 10111 = IRQ7 | 11100 = IRQ12 |
| 10010 = IRQ2 | 11000 = IRQ8 | 11101 = IRQ13 |
| 10011 = IRQ3 | 11001 = IRQ9 | 11110 = IRQ14 |
| 10100 = IRQ4 | 11010 = IRQ10 | 11111 = IRQ15 |
| 10101 = IRQ5 | | |

Reserved

Using Host PCI INTA#-D# (PCICFG 50h[6]=1)

| | |
|---|---|
| 000 = Disabled | 011 = INTC# |
| 001 = INTA# | 100 = INTD# |
| 010 = INTB# (default) | 101-111 = Reserved |

**PCICFG 4Ah — Docking PCIRQ2# Interrupt Assignment Register — Default = 03h**

Reserved

Using OPTi IRQ driveback mechanism:

Docking PCIRQ2# Interrupt Assignment (PCIRQ2# Default) - Interrupts from the docking PCIRQ2# pin are mapped to this interrupt. Note that if an IRQ (an edge-mode interrupt) is selected, this IRQ must be programmed to Level mode on the host chipset.

**Level Mode:**

| | | |
|---|---|---|
| 00000 = Disabled | 00010 = PCIRQ1# | 00100 = PCIRQ3# |
| 00001 = PCIRQ0# | 00011 = PCIRQ2# (Default) | 00101-01111 = Rsrvd |

**Edge Mode: (Viper-N+)**

| | | |
|---|---|---|
| 10000 = IRQ0 | 10110 = IRQ6 | 11011 = IRQ11 |
| 10001 = IRQ1 | 10111 = IRQ7 | 11100 = IRQ12 |
| 10010 = IRQ2 | 11000 = IRQ8 | 11101 = IRQ13 |
| 10011 = IRQ3 | 11001 = IRQ9 | 11110 = IRQ14 |
| 10100 = IRQ4 | 11010 = IRQ10 | 11111 = IRQ15 |
| 10101 = IRQ5 | | |

Reserved

Using Host PCI INTA#-D# (PCICFG 50h[6]=1)

| | |
|---|---|
| 000 = Disabled | 011 = INTC# (default) |
| 001 = INTA# | 100 = INTD# |
| 010 = INTB# | 101-111 = Reserved |

**Table 5-1 Base Register Group - PCICFG 00h-4Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 4Bh** | | | **Docking PCIRQ3# Interrupt Assignment Register** | | | | **Default = 04h** |
| Reserved | | | Using OPTi IRQ driveback mechanism:<br>Docking PCIRQ3# Interrupt Assignment (PCIRQ3# Default) - Interrupts from the docking PCIRQ3# pin are mapped to this interrupt. Note that if an IRQ (an edge-mode interrupt) is selected, this IRQ must be programmed to Level mode on the host chipset.<br>**Level Mode:**<br>00000 = Disabled<br>00001 = PCIRQ0#<br>00010 = PCIRQ1#<br>00011 = PCIRQ2#<br>00100 = PCIRQ3# (Default)<br>00101-01111 = Rsrvd<br>**Edge Mode: (Viper-N+)**<br>10000 = IRQ0<br>10001 = IRQ1<br>10010 = IRQ2<br>10011 = IRQ3<br>10100 = IRQ4<br>10101 = IRQ5<br>10110 = IRQ6<br>10111 = IRQ7<br>11000 = IRQ8<br>11001 = IRQ9<br>11010 = IRQ10<br>11011 = IRQ11<br>11100 = IRQ12<br>11101 = IRQ13<br>11110 = IRQ14<br>11111 = IRQ15 | | | | |
| Reserved | | | | | Using Host PCI INTA#-D# (PCICFG 50h[6]=1)<br>000 = Disabled<br>001 = INTA#<br>010 = INTB#<br>011 = INTC#<br>100 = INTD# (default)<br>101-111 = Reserved | | |
| **PCICFG 4Ch** | | | **Docking Detect Interrupt Assignment Register** | | | | **Default = 01h** |
| Host controller type:<br>0 = FireStar (burst two data phases)<br>1 = Viper-N+ (send single data phase on IRQ driveback) | Interrupt Pin Requested in PCICFG 3Dh<br>0 = equal to Level Mode selections in PCICFG 4Ch[4:0]<br>1 = Always 01 | Reserved | Using OPTi IRQ driveback mechanism:<br>Docking Detect Interrupt Assignment - If attachment of a docking station is detected, or if the device attached could not be determined, this interrupt will be generated. This same interrupt will be generated when the docking station is removed.<br>**Level Mode:**<br>00000 = Disabled<br>00001 = PCIRQ0# (Default)<br>00010 = PCIRQ1#<br>00011 = PCIRQ2#<br>00100 = PCIRQ3#<br>00101 = ACPI0<br>00110 = ACPI1<br>00111 = ACPI2<br>01000 = ACPI3<br>01001-01111 = Rsrvd<br>**Edge Mode: (Viper-N+)**<br>10000 = IRQ0<br>10001 = IRQ1<br>10010 = IRQ2<br>10011 = IRQ3<br>10100 = IRQ4<br>10101 = IRQ5<br>10110 = IRQ6<br>10111 = IRQ7<br>11000 = IRQ8<br>11001 = IRQ9<br>11010 = IRQ10<br>11011 = IRQ11<br>11100 = IRQ12<br>11101 = IRQ13<br>11110 = IRQ14<br>11111 = IRQ15 | | | | |
| Reserved | | | | | Using Host PCI INTA#-D# (PCICFG 50h[6]=1)<br>000 = Disabled<br>001 = INTA# (default)<br>010 = INTB#<br>011 = INTC#<br>100 = INTD#<br>101-111 = Reserved | | |
| **PCICFG 4Dh** | | | **Serial IRQ Enable Register** | | | | **Default = 00h** |
| Test Bits (for factory use only) | | | | | | Reserved | IRQSER on pin 15<br>0 = Disabled<br>1 = Enabled |
| **PCICFG 4Eh** | | | **Serial IRQ Control Register 1** | | | | **Default = 00h** |

**Table 5-1 Base Register Group - PCICFG 00h-4Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Compaq SIRQ HALT mode request:<br>0 = Active<br>1 = Halt | Compaq SIRQ QUIET mode request:<br>0 = Continuous<br>1 = Quiet | Reserved | Compaq SIRQ data frame slots. Change only when the Serial IRQ logic is disabled or in Halt state.<br>0 = 17 slots<br>1 = 21 slots | Compaq SIRQ Start frame width in PCI clocks. Change this setting only when Serial IRQ is disabled or in Halt state.<br>00 = 4 PCI clocks<br>01 = 6 PCI clocks<br>10 = 8 PCI clocks<br>11 = Reserved | | Reserved | Compaq SIRQ (Compaq Serial IRQ scheme):<br>0 = Disable<br>1 = Enable |
| **PCICFG 4Fh** | | | **Serial IRQ Control Register 2 And External Arbiter Enable** | | | | **Default = 00h** |
| Compaq SIRQ in HALT state (RO)?<br>0 = No<br>1 = Yes | Compaq SIRQ in QUIET state (RO)?<br>0 = No<br>1 = Yes | Reserved | | | | | External Arbiter on secondary PCI:<br>0 = Disable<br>1 = Enable |

## 5.3 82C814-Specific Register Group

The 82C814 defines many special functions that require enabling and monitoring through a dedicated register set. The 82C814-specific registers at PCICFG 50h-5Fh remain set to their programmed values even after a device is removed from the slot. Also, PCICFG 50h is common to both slot interfaces (i.e. changing the bit in one PCI register set changes it in the other).

The following subsections discuss some of the special functions located in the 82C814-Specific Register Group.

### 5.3.1 CLKRUN#

PCICFG 50h[2] selects whether the CLKRUN# signal to the host will toggle. Normally it will be set for automatic operation. In this mode, the 82C814 logic asserts CLKRUN# only when it wants bus ownership for master cycles, or when it has an interrupt it must send to the host. At all other times, it leaves CLKRUN# tristated and depends on the current PCI bus master to assert CLKRUN# and keep the clock running.

### 5.3.2 Slot Buffer Enable, Slew Rate, and Threshold Control

PCICFG 51h[2:0] are automatically updated by the card insertion state machine according to whether a 5.0V or 3.3V dock has been detected using CD1-2# and VS1-2. Once the card has been inserted and detected, and the interface automatically set appropriately, software can still override the automatic settings by reading and then writing PCICFG 51h[2:0] as desired.

### 5.3.3 Dual ISA Buses

Dual ISA buses are possible with the 82C814 chip used in conjunction with the OPTi PCI-ISA Bridge chips. This feature depends on the ISA Windows feature of the 82C814 chip, which allows cycles destined for the remote docking ISA bus to be claimed with positive decoding from the primary PCI bus and then retried. If the cycle turns out not to be destined for the docking ISA bus, the 82C814 chip ignores the next retry so that the cycle will be claimed using subtractive decode by the host chipset.

The FireStar chip provides an additional feature that allows positive decode of cycles to known local ISA devices. This feature would conflict with the positive decode used by the 82C814 chip. Therefore, the 82C814 chip has the option of decoding on the slow clock instead of on the medium clock. This feature is enabled by writing PCICFG 5Eh[7] = 1.

When the feature is selected, the 82C814 logic will monitor the DEVSEL# line to determine whether FireStar (or anyone else) has claimed the cycle by fast or medium decode. Only if DEVSEL# remains high through the medium decode clock will the 82C814 chip claim the cycle.

The slow decode feature works only for windows enabled as ISA windows. Other windows will continue to use a medium decode.

**Table 5-2 Specific Register Group - PCICFG 50h-5Fh**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 50h** | | | **PCI Host Feature Control Register** | | | | **Default = 00h** |
| Reserved | Primary INTA#-INTD# Select<br>0 = No (pin 12 pulled down)<br>1 = Yes (pin 12 pulled up) | Vendor ID feature selected:<br>0 = No<br>1 = Yes | IRQLATCH function<br>0 = Disable (default)<br>1 = Enable | Docking detect debounce:<br>0 = 1.0 sec<br>1 = 0.25 sec | CLKRUN# (on host interface):<br>0 = Enabled per PCI<br>1 = Disabled, CLKRUN# tristated | CRST# and ENVCC5 Control<br>0 = Normal<br>1 = Force both signals high, ignore PCICFG 3Eh[6] and MEMOFST 070h | MEMOFST 010 bits [2:0]<br>0 = Read/Write<br>1 = Read only |
| **PCICFG 51h** | | | **Docking Feature Control Register 1** | | | | **Default = 04h** |
| Dock Interface clock divisor:<br>00 = 1 (Default)<br>01 = 2<br>10 = 3<br>11 = 4 | | Dock Interface-clock source:<br>0 = PCICLK<br>1 = EXTCLK | Mode select:<br>0 = Automatic<br>1 = Force async | CCLKRUN# on dock interface<br>0 = Disabled<br>1 = Enabled | Dock Interface threshold voltage:<br>0 = 3.3V<br>1 = 5.0V | Output Drive Select:<br>00 = Reserved<br>01 = Reserved<br>10 = 3.3V PCI dock<br>11 = 5.0V PCI dock | |
| | | | | | These bits retain their previously written value regardless of dock attachment/removal. | | |
| **PCICFG 52h** | | | **Docking Feature Control Register 2** | | | | **Default = 4Fh** |
| Secondary PCICLK Skew:<br>This value selects the approximate delay, in nanoseconds, that the internal secondary PCICLK must be skewed in order to compensate for external buffer delays.<br>0000 = No delay 0001 = 1ns 0010 = 2ns<br>.....<br>1101 = 13ns 1110 = 14ns 1111 = 15ns | | | | Block Prefetch on Downstream Transactions<br>0 = No<br>1 = Yes (default)<br>Controls memory windows 0 & 1 only | Block Posted Writes on Downstream Transactions<br>0 = No<br>1 = Yes (default)<br>Controls memory windows 0 & 1 only | Enabled delayed Transactions<br>0 = Only when window selected as ISA window<br>1 = On all windows whenever retry count exceeds 50% of retry limit (PCICFG 5Eh[2:0]). (default) | Type 1 to Type 0 conversion blocked from secondary to primary:<br>0 = No<br>1 = Yes (Default) |
| **PCICFG 53h** | | | **Docking Feature Control Register 3** | | | | **Default = 00h** |
| Event signalled when PCICFG 3Eh[6] is changed | | Event signalled when CD1-2# change states | | Event signalled on read of MEMOFST 0-FFFh | | Event signalled on write of MEMOFST 0-FFFh | |
| 00 = None | | 00 = None | | 00 = None | | 00 = None | |
| 01 = SMI# | | 01 = SMI# | | 01 = SMI# | | 01 = SMI# | |
| 10 = ACPI1 (DOCK#) | | 10 = ACPI1 (DOCK#) | | 10 = ACPI1 (DOCK#) | | 10 = ACPI1 (DOCK#) | |
| 11 = ACPI2 (STSCHG#) | | 11 = ACPI2 (STSCHG#) | | 11 = ACPI2 (STSCHG#) | | 11 = ACPI2 (STSCHG#) | |
| **Note:** SMI# is available through IRQ Driveback or Serial IRQ; ACPI1 and ACPI2 are available only through IRQ Driveback | | | | | | | |

**Table 5-2 Specific Register Group - PCICFG 50h-5Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 54h** | | | **IRQ Driveback Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 30h** |
| IRQ Driveback Protocol Address Bits:<br>- When the 82C814 logic must generate an interrupt from any source, it follows the IRQ Driveback Protocol and toggles the REQ# line to the host. Once it has the bus, it writes the changed IRQ information to the 32-bit I/O address specified in this register. The host interrupt controller claims this cycle and latches the new IRQ values.<br>- Bits 2:0 are reserved to be 000 and are read-only.<br>- This register defaults to a value of 33333330h. | | | | | | | |
| **PCICFG 55h** | | | **IRQ Driveback Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 33h** |
| **PCICFG 56h** | | | **IRQ Driveback Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 33h** |
| **PCICFG 57h** | | | **IRQ Driveback Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 33h** |
| **PCICFG 58h** | | | **DRQ Remap Base Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| DRQ Remap Base Address Bits:<br>- The distributed DMA protocol requires DMA controller registers for each DMA channel to be individually mapped into I/O space outside the range claimed by ISA devices. Bits A[31:0] of this register specify that base; bits 7:0 are reserved (write 0) because the base address can fall only on 256 byte boundaries.<br>- The 82C814 logic uses this base address to forward accesses across the bridge to remote devices specified in the DMA Channel Selector Register. | | | | | | | |
| **PCICFG 59h** | | | **DRQ Remap Base Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 5Ah** | | | **DRQ Remap Base Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG 5Bh** | | | **DRQ Remap Base Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 00h** |
| **PCICFG 5Ch** | | | **DMA Channel Selector Register** | | | | **Default = 00h** |
| Channel 7 (DMAC2):<br>0 = Not claimed<br>1 = On slot interface | Channel 6 (DMAC2):<br>0 = Not claimed<br>1 = On slot interface | Channel 5 (DMAC2):<br>0 = Not claimed<br>1 = On slot interface | DMAC responsibility (RO):<br>0 = Secondary (always)<br>1 = Master | Channel 3 (DMAC1):<br>0 = Not claimed<br>1 = On slot interface | Channel 2 (DMAC1):<br>0 = Not claimed<br>1 = On slot interface | Channel 1 (DMAC1):<br>0 = Not claimed<br>1 = On slot interface | Channel 0 (DMAC1):<br>0 = Not claimed<br>1 = On slot interface |
| **PCICFG 5Dh** | | | **SMI Status Register (Write 1 to clear bit)** | | | | **Default = 00h** |
| Toggling of PCICFG 3Eh[6] Generated SMI<br>0 = No<br>1 = Yes | Dock/Undock Event Generated SMI<br>0 = No<br>1 = Yes | Read of CardBus Registers (MEMOFST 0=FFFh) Generated SMI<br>0 = No<br>1 = Yes | Write of CardBus Registers (MEMOFST 0=FFFh) Generated SMI<br>0 = No<br>1 = Yes | Docking Window 3 generated SMI:<br>0 = No<br>1 = Yes | Docking Window 2 generated SMI:<br>0 = No<br>1 = Yes | Docking Window 1 generated SMI:<br>0 = No<br>1 = Yes | Docking Window 0 generated SMI:<br>0 = No<br>1 = Yes |

**Table 5-2 Specific Register Group - PCICFG 50h-5Fh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 5Eh** | | | **Primary Retry Limit Register** | | | | **Default = 00h** |
| Slow decode for ISA windows:<br>0 = Disable<br>1 = Enable | Prefetch on upstream transactions:<br>0 = Disable<br>1 = Enable | Posted writes on upstream transactions:<br>0 = Disable<br>1 = Enable | Core voltage:<br>0 = 3.3V<br>1 = 5.0V | Retry count readback control:<br>0 = Write posting retries on secondary<br>1 = Retries on primary | Retry Limit:<br>These bits relate to the number of times that the 82C814, as a slave, will retry accesses on the primary. If this limit is exceeded, the 82C814 generates SERR# to the host.<br>000=$2^8$ 100=$2^{16}$<br>001=$2^{10}$ 101=$2^{20}$<br>010=$2^{12}$ 110=$2^{24}$<br>011=$2^{14}$ 111= Infinite retries (Default) | | |
| **PCICFG 5Fh** | | | **82C814 Retry Count Readback Register (RO)** | | | | **Default = 00h** |
| - This register returns the number of retry attempts made.<br>- More than 256 retries are indicated by FFh.<br>- Used for diagnostic purposes. Read-only.<br>- Separate counts are maintained for primary and secondary. Bit 5Eh[3] selects the count being read back. | | | | | | | |
| **Write-Only:** This register is also writable, for factory diagnostic purposes only. | | | | | | | |
| Status Change Initialization | | When PCICFG 51h[3]=1<br>0 = Automatic Mode<br>1 = Sync mode | PCI Retry Test<br>0 = Disable<br>1 = Enable | Prototype test mode:<br>0 = Disable<br>1 = Enable | Force FIFO clear | Retry test times:<br>0 = Normal<br>1 = Quick* | Power-up and detect timer:<br>0 = Normal<br>1 = Quick |
| 0 = Old scheme<br>1 = New | 0 = Original<br>1 = Variation | | | | | | |
| * Quick Mode will not function unless PCICFG 5Eh[2:0] is zero. | | | | | | | |

## 5.4 CardBus Register Group

The CardBus-style control and status register group is accessible through two different means. It is always accessible as part of the PCI configuration space at the indexes shown in Table 5-4. In addition, when the CardBus register base address at PCICFG 14h is written to any value other than zero, these same registers can be accessed through the system memory space selected (see Table 5-3).

Note that when accessing these registers in PCI memory space, they start from an offset of 00h, not 60h, from the register base address programmed.

### 5.4.1 Power Control

PCICFG 70h[6:4] set the external VCC5 and VCC3 pin levels. Because only these two pins are available on the 82C814 interface, the system must be designed to interpret these signals properly and select the correct voltage for the application.

**Table 5-3 CardBus Register Set in System Memory**

| CardBus Base Address plus: | Name |
|---|---|
| 000h | Socket Event Register |
| 004h | Socket Mask Register |
| 008h | Socket Present State Register |
| 00Ch | Force Event Register |
| 010h | Control Register |
| 014-7FFh | Reserved |

**Table 5-4 CardBus Register Group - PCICFG 60h-74h / MEMOFST 00h-7Fh**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 60h / MEMOFST 00h** | | | **Socket Event Register - Byte 0** | | | | **Default = 00h** |
| Reserved:<br>Write as read. | | | | Power cycle complete:<br>0 = No<br>1 = Yes<br>Write 1 to clear | CCD2# status change:<br>0 = No<br>1 = Yes<br>Write 1 to clear | CCD1# status change:<br>0 = No<br>1 = Yes<br>Write 1 to clear | Reserved: |
| **PCICFG 61h / MEMOFST 01h** | | | **Socket Event Register - Byte 1** | | | | **Default = 00h** |
| Reserved: Write as read. | | | | | | | |
| **PCICFG 62h / MEMOFST 02h** | | | **Socket Event Register - Byte 2** | | | | **Default = 00h** |
| Reserved: Write as read. | | | | | | | |
| **PCICFG 63h / MEMOFST 03h** | | | **Socket Event Register - Byte 3** | | | | **Default = 00h** |
| Reserved. Write as read. | | | | | | | |
| **PCICFG 64h / MEMOFST 04h** | | | **Socket Mask Register - Byte 0** | | | | **Default = 00h** |
| Reserved:<br>Write as read. | | | | Power cycle status change event:<br>0 = Mask<br>1 = Enable | CCD2# status change event:<br>0 = Mask<br>1 = Enable | CCD1# status change event:<br>0 = Mask<br>1 = Enable | Reserved |
| **PCICFG 65h / MEMOFST 05h** | | | **Socket Mask Register - Byte 1** | | | | **Default = 00h** |
| Reserved: Write as read. | | | | | | | |
| **PCICFG 66h / MEMOFST 06h** | | | **Socket Mask Register - Byte 2** | | | | **Default = 00h** |
| Reserved: Write as read. | | | | | | | |
| **PCICFG 67h / MEMOFST 07h** | | | **Socket Mask Register - Byte 3** | | | | **Default = 00h** |
| Reserved: Write as read. | | | | | | | |
| **PCICFG 68h / MEMOFST 08h** | | | **Socket Present State Register (RO) - Byte 0** | | | | **Default = 00h** |
| Dock recognized - updated only on card insertion:<br>0 = Yes<br>1 = No | PCIRQ# Status<br>0 = At least one of PCIRQ0-3# is low<br>1 = PCIRQ0-3# are all high | Device type - updated only on card insertion:<br>11 = Docking station<br>All other combinations reserved | | Power cycle status:<br>0 = Not successful<br>1 = Successful | CCD2-1# state:<br>00 = Dock attached<br>01 = No dock attached<br>10 = No dock attached<br>11 = No dock attached | | Reserved |
| **PCICFG 69h / MEMOFST 09h** | | | **Socket Present State Register - Byte 1** | | | | **Default = 00h** |
| Reserved:<br>Write as read. | | Reserved | | 3.3V dock detected:<br>0 = No<br>1 = Yes | 5.0V dock detected:<br>0 = No<br>1 = Yes | Bad VCC request (outside CVS1-2, CCD1-2# range):<br>0 = No<br>1 = Yes | Data lost (dock detached before transaction completed):<br>0 = No<br>1 = Maybe |
| **PCICFG 6Ah / MEMOFST 0Ah** | | | **Socket Present State Register - Byte 2** | | | | **Default = 00h** |
| Reserved: Write as read. | | | | | | | |
| **PCICFG 6Bh / MEMOFST 0Bh** | | | **Socket Present State Register - Byte 3 (bits are writeable)** | | | | **Default = 30h** |

**Table 5-4 CardBus Register Group - PCICFG 60h-74h / MEMOFST 00h-7Fh**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Socket can supply Voltage Y:<br>0 = No<br>1 = Yes | Socket can supply Voltage X:<br>0 = No<br>1 = Yes | Socket can supply 3.3V:<br>0 = No<br>1 = Yes | Socket can supply 5.0V:<br>0 = No<br>1 = Yes | Reserved:<br>Write as read. | | | |
| **PCICFG 6Ch / MEMOFST 0Ch** | | | **Force Event Register - Byte 0** | | | | **Default = 00h** |
| Force dock recognized bit to 1:<br>0 = No<br>1 = Yes | Reserved:<br>Write as read. | Force device type:<br>11 = Docking station<br>All other combinations reserved | | Force power cycle event:<br>0 = No<br>1 = Yes | Force CCD2# event:<br>0 = No<br>1 = Yes | Force CCD1# event:<br>0 = No<br>1 = Yes | Reserved |
| **PCICFG 6Dh / MEMOFST 0Dh** | | | **Force Event Register - Byte 1** | | | | **Default = 00h** |
| Reserved:<br>Write as read. | Force retest of CVS1-2, CCD1-2# pins (or force bits):<br>0 = No<br>1 = Yes | Reserved | | Force 3.3V dock detected bit to 1:<br>0 = No<br>1 = Yes | Force 5.0V dock detected bit to 1:<br>0 = No<br>1 = Yes | Force bad VCC request bit to 1:<br>0 = No<br>1 = Yes | Force data lost bit to 1:<br>0 = No<br>1 = Yes |
| **PCICFG 6Eh / MEMOFST 0Eh** | | | **Force Event Register - Byte 2**<br>Reserved: Write as read. | | | | **Default = 00h** |
| **PCICFG 6Fh / MEMOFST 0Fh** | | | **Force Event Register - Byte 3**<br>Reserved: Write as read. | | | | **Default = 00h** |
| **PCICFG 70h / MEMOFST 10h** | | | **Control Register - Byte 0** | | | | **Default = 00h** |
| Reserved:<br>Write as read. | Dock VCC power request:<br>000 = Power off<br>001 = Reserved<br>010 = 5.0V<br>011 = 3.3V<br>100 = Voltage X<br>101 = Voltage Y<br>11x = Reserved | | | Reserved:<br>Write as read. | Dock VPP Power Request<br>Read/write bits; do not have any function. If PCICFG 50h[0] = 1, these bits always return 0. | | |
| **PCICFG 71h / MEMOFST 11h** | | | **Control Register - Byte 1**<br>Reserved: Write as read. | | | | **Default = 00h** |
| **PCICFG 72h / MEMOFST 12h** | | | **Control Register - Byte 2**<br>Reserved: Write as read. | | | | **Default = 00h** |
| **PCICFG 73h / MEMOFST 13h** | | | **Control Register - Byte 3**<br>Reserved: Write as read. | | | | **Default = 00h** |
| **PCICFG 74h / MEMOFST 14h** | | | **Reserved** | | | | **Default = 00h** |

## 5.5 Docking Station Window Selection Group

One block of the 82C814 PCI-to-CardBus configuration registers is used to select the memory or I/O address ranges that will be claimed by the bridge and passed onto the secondary PCI bus for use with the OPTi PCI-to-ISA Bridge.

Windows 4-7, which were available in Rev 0.0 of the 82C814 chip and overlapped the CardBus windows, are no longer available. Table 5-5 summarizes the features.

The docking station access windows allow far more flexibility in cycle selection, masking, etc. than do the CardBus window registers.

### 5.5.1 Docking Station Window Registers

The docking station registers are listed in Table 5-5. Table 5-6 follows and includes the default settings for each register.

**Table 5-5 Docking Station Access Windows**

| Docking Station Access Window # | Default Mask | Bits Decoded | Memory or I/O Selectable? | Can Generate SMI#? |
|---|---|---|---|---|
| 0 | 000FFFh | A[31:2] | Yes - Defaults to Memory | Yes |
| 1 | 000FFFh | A[31:2] | Yes - Defaults to Memory | Yes |
| 2 | 000003h | A[31:2] | Yes - Defaults to I/O | Yes |
| 3 | 000003h | A[31:2] | Yes - Defaults to I/O | Yes |

#### 5.5.1.1 Cycle Decoding

Each window can select either memory or I/O decoding, and allows for a decode range anywhere from one dword to the entire address space. Upper address bits from A31 on down can be masked in the comparison, allowing any desired degree of aliasing.

#### 5.5.1.2 Cycle Trapping

Instead of passing a claimed cycle onto the intended slave PCI interface, the cycle controller can generate a STOP# or CSTOP# on the master PCI interface (primary PCI interface or slot interface) and cause the controlling device to back off. At the same time, the cycle controller generates an IRQ driveback cycle with SMI# active, therefore converting the cycle into a System Management Interrupt trap.

At this point, the master will most likely retry the cycle, at which time the 82C814 will allow it to proceed. It may or may not be able to deliver valid data. The host chipset can then run its SMM code. The SMM code can read the SMI Status Register from the 82C814 to determine the window access that caused the SMI. Once the value has been read, the host must write a 1 back to each SMI indicator bit to re-enable trapping and SMI generation on that window.

#### 5.5.1.3 ISA Window Selection

All docking station windows contain the ISA Window Selection bit. When set to 1, the window operation is modified as follows.

- When a cycle initiated on the primary is claimed through this window, the cycle will be immediately and automatically retried.
- On the docking station side, the OPTi PCI-ISA Bridge will claim the cycle and wait for positive decode on the ISA bus.
  - If positive decode is determined, the OPTi PCI-ISA Bridge logic will terminate the cycle normally.
  - If no positive decode can be achieved, the OPTi PCI-ISA Bridge logic will terminate the cycle with a Target Abort. Once this occurs, the 82C814 chip will simply ignore the next retry attempt on its primary and allow the cycle to pass to the local ISA bus of the host controller.

The retries occur up to the limit defined in PCICFG 5Eh[2:0] before SERR# is generated.

**Table 5-6 Docking Station Window Registers - PCICFG 80h-EFh**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 80h** | | | **Window 0 Start Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| Window Start Address Bits:<br>- Register bits [31:0] indicate the start address for Window 0.<br>- The selection between memory or I/O, as well as other feature selections, are made through the Window 0 Control Register. | | | | | | RO: Always returns 0 | If memory: reads 0.<br>If I/O: Decoding<br>0 = 16-bit AD[31:16] = 0<br>1 = 32-bit |
| **PCICFG 81h** | | | **Window 0 Start Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = FFh** |
| **PCICFG 82h** | | | **Window 0 Start Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = FFh** |
| **PCICFG 83h** | | | **Window 0 Start Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = FFh** |
| **PCICFG 84h** | | | **Window 0 Stop Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| Window 0 Address Bits:<br>- Register bits [31:0] indicate the stop address for one of the eight memory or I/O windows. | | | | | | RO: Always returns 0 | |

**Table 5-6 Docking Station Window Registers - PCICFG 80h-EFh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG 85h** | | | **Window 0 Stop Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 86h** | | | **Window 0 Stop Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG 87h** | | | **Window 0 Stop Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 00h** |
| **PCICFG 88h** | | | **Window 0 Mask Register - Byte 0: Mask Bits [7:0]** | | | | **Default = 03h** |
| Window 0 Mask Bits:<br>- Mask register bits [23:2] allow Window 0 to be aliased throughout the memory or I/O address space.<br>- Setting any bit to a 1 masks out the comparison on this bit.<br>- The register should be written to 0 to decode the entire address.<br>- Bits [1:0] are always 11 (masked). | | | | | | RO:<br>Always returns 1. | |
| **PCICFG 89h** | | | **Window 0 Mask Register - Byte 1: Mask Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 8Ah** | | | **Window 0 Mask Register - Byte 2: Mask Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG 8Bh** | | | **Window 0 Control Register** | | | | **Default = 48h** |
| Window points to ISA bus:<br>0 = No<br>1 = Yes | Reads are prefetchable:<br>0 = No<br>1 = Yes<br>Set to 0 for I/O window | Writes can be posted:<br>0 = No<br>1 = Yes<br>Set to 0 for I/O window | Reserved | Cycle qualifier:<br>0 = I/O<br>1 = Memory (Default) | Window 0 Trap/SMI#:<br>0 = Disable<br>1 = Enable | Reserved | |
| **PCICFG 8Ch-8Fh** | | | **Reserved** | | | | **Default = 00h** |
| **PCICFG 90h** | | | **Window 1 Start Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| Window 1 Start Address Bits:<br>- Register bits [31:0] indicate the start address for Window 1.<br>- The selection between memory or I/O, as well as other feature selections, are made through the Window 1 Control Register. | | | | | | RO:<br>Always returns 0 | If memory: reads 0.<br>If I/O: Decoding<br>0 = 16-bit AD[31:16] = 0<br>1 = 32-bit |
| **PCICFG 91h** | | | **Window 1 Start Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = FFh** |
| **PCICFG 92h** | | | **Window 1 Start Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = FFh** |
| **PCICFG 93h** | | | **Window 1 Start Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = FFh** |
| **PCICFG 94h** | | | **Window 1 Stop Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| Window 1 Stop Address Bits:<br>- Register bits [31:0] indicate the stop address for one of the eight memory or I/O windows. | | | | | | RO:<br>Always returns 0 | |
| **PCICFG 95h** | | | **Window 1 Stop Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG 96h** | | | **Window 1 Stop Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG 97h** | | | **Window 1 Stop Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 00h** |
| **PCICFG 98h** | | | **Window 1 Mask Register - Byte 0: Mask Bits [7:0]** | | | | **Default = 03h** |

**Table 5-6 Docking Station Window Registers - PCICFG 80h-EFh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Window 1 Mask Bits:<br>- Mask register bits [23:2] allow Window 0 to be aliased throughout the memory or I/O address space.<br>- Setting any bit to a 1 masks out the comparison on this bit.<br>- The register should be written to 0 to decode the entire address.<br>- Bits [1:0] are always 11 (masked). | | | | | | RO:<br>Always returns 1. | |
| **PCICFG 99h** | **Window 1 Mask Register - Byte 1: Mask Bits [15:8]** | | | | | | **Default = 00h** |
| **PCICFG 9Ah** | **Window 1 Mask Register - Byte 2: Mask Bits [23:16]** | | | | | | **Default = 00h** |
| **PCICFG 9Bh** | **Window 1 Control Register** | | | | | | **Default = 48h** |
| Window points to ISA bus:<br>0 = No<br>1 = Yes | Reads are prefetchable:<br>0 = No<br>1 = Yes<br>Set to 0 for I/O window | Writes can be posted:<br>0 = No<br>1 = Yes<br>Set to 0 for I/O window | Reserved | Cycle qualifier:<br>0 = I/O<br>1 = Memory (Default) | Window 1 Trap/SMI#:<br>0 = Disable<br>1 = Enable | Reserved | |
| **PCICFG 9Ch-9Fh** | **Reserved** | | | | | | **Default = 00h** |
| **PCICFG A0h** | **Window 2 Start Address Register - Byte 0: Address Bits [7:0]** | | | | | | **Default = 00h** |
| Window 2 Address Bits:<br>- Register bits [31:0] indicate the start address for Window 2.<br>- The selection between memory or I/O, as well as other feature selections, are made through the Window 2 Control Register. | | | | | | RO:<br>Always returns 0 | If memory: reads 0.<br>If I/O: Decoding<br>0 = 16-bit AD[31:16] = 0<br>1 = 32-bit |
| **PCICFG A1h** | **Window 2 Start Address Register - Byte 1: Address Bits [15:8]** | | | | | | **Default = FFh** |
| **PCICFG A2h** | **Window 2 Start Address Register - Byte 2: Address Bits [23:16]** | | | | | | **Default = FFh** |
| **PCICFG A3h** | **Window 2 Start Address Register - Byte 3: Address Bits [31:24]** | | | | | | **Default = FFh** |
| **PCICFG A4h** | **Window 2 Stop Address Register - Byte 0: Address Bits [7:0]** | | | | | | **Default = 00h** |
| Window 2 Stop Address Bits:<br>- Register bits [31:0] indicate the stop address for one of the eight memory or I/O windows. | | | | | | RO:<br>Always returns 0 | |
| **PCICFG A5h** | **Window 2 Stop Address Register - Byte 1: Address Bits [15:8]** | | | | | | **Default = 00h** |
| **PCICFG A6h** | **Window 2 Stop Address Register - Byte 2: Address Bits [23:16]** | | | | | | **Default = 00h** |
| **PCICFG A7h** | **Window 2 Stop Address Register - Byte 3: Address bits [31:24]** | | | | | | **Default = 00h** |
| **PCICFG A8h** | **Window 2 Mask Register - Byte 0: Mask Bits [7:0]** | | | | | | **Default = 03h** |
| Window 2 Mask Bits:<br>- Mask register bits [23:2] allow Window 0 to be aliased throughout the memory or I/O address space.<br>- Setting any bit to a 1 masks out the comparison on this bit.<br>- The register should be written to 0 to decode the entire address.<br>- Bits [1:0] are always 11 (masked). | | | | | | RO:<br>Always returns 1. | |
| **PCICFG A9h** | **Window 2 Mask Register - Byte 1: Mask Bits [15:8]** | | | | | | **Default = 00h** |
| **PCICFG AAh** | **Window 2 Mask Register - Byte 2: Mask Bits [23:16]** | | | | | | **Default = 00h** |

**Table 5-6 Docking Station Window Registers - PCICFG 80h-EFh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **PCICFG ABh** | | | **Window 2 Control Register** | | | | **Default = 00h** |
| Window points to ISA bus:<br>0 = No<br>1 = Yes | Reads are prefetchable:<br>0 = No<br>1 = Yes<br>Set to 0 for I/O window | Writes can be posted:<br>0 = No<br>1 = Yes<br>Set to 0 for I/O window | Reserved | Cycle qualifier:<br>0 = I/O (Default)<br>1 = Memory | Window 2 Trap/SMI#:<br>0 = Disable<br>1 = Enable | Reserved | |
| **PCICFG ACh-AFh** | | | **Reserved** | | | | **Default = 00h** |
| **PCICFG B0h** | | | **Window 3 Start Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| Window 3 Address Bits:<br>- Register bits [31:0] indicate the start address for Window 3.<br>- The selection between memory or I/O, as well as other feature selections, are made through the Window 3 Control Register. | | | | | | RO:<br>Always returns 0 | If memory: reads 0.<br>If I/O: Decoding<br>0 = 16-bit<br>AD[31:16] = 0<br>1 = 32-bit |
| **PCICFG B1h** | | | **Window 3 Start Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = FFh** |
| **PCICFG B2h** | | | **Window 3 Start Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = FFh** |
| **PCICFG B3h** | | | **Window 3 Start Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = FFh** |
| **PCICFG B4h** | | | **Window 3 Stop Address Register - Byte 0: Address Bits [7:0]** | | | | **Default = 00h** |
| Window 3 Stop Address Bits:<br>- Register bits [31:0] indicate the stop address for one of the eight memory or I/O windows. | | | | | | RO:<br>Always returns 0 | |
| **PCICFG B5h** | | | **Window 3 Stop Address Register - Byte 1: Address Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG B6h** | | | **Window 3 Stop Address Register - Byte 2: Address Bits [23:16]** | | | | **Default = 00h** |
| **PCICFG B7h** | | | **Window 3 Stop Address Register - Byte 3: Address Bits [31:24]** | | | | **Default = 00h** |
| **PCICFG B8h** | | | **Window 3 Mask Register - Byte 0: Mask Bits [7:0]** | | | | **Default = 03h** |
| Window 3 Mask Bits:<br>- Mask register bits [23:2] allow Window 0 to be aliased throughout the memory or I/O address space.<br>- Setting any bit to a 1 masks out the comparison on this bit.<br>- The register should be written to 0 to decode the entire address.<br>- Bits [1:0] are always 11 (masked).<br>- Bit 23 is always 0 | | | | | | RO:<br>Always returns 1. | |
| **PCICFG B9h** | | | **Window 3 Mask Register - Byte 1: Mask Bits [15:8]** | | | | **Default = 00h** |
| **PCICFG BAh** | | | **Window 3 Mask Register - Byte 2: Mask Bits [23:16]** | | | | **Default = 00h** |
| Bit 23 Reserved | Bits [22:16] | | | | | | |
| **PCICFG BBh** | | | **Window 3 Control Register** | | | | **Default = 00h** |

**Table 5-6 Docking Station Window Registers - PCICFG 80h-EFh (cont.)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Window points to ISA bus:<br>0 = No<br>1 = Yes | Reads are prefetchable:<br>0 = No<br>1 = Yes<br>Set to 0 for I/O window | Writes can be posted:<br>0 = No<br>1 = Yes<br>Set to 0 for I/O window | Reserved | Cycle qualifier:<br>0 = I/O (Default)<br>1 = Memory | Window 3 Trap/SMI#:<br>0 = Disable<br>1 = Enable | Reserved | |
| **PCICFG BCh-EFh** | **Reserved** | | | | | | **Default = 00h** |

## 5.6 PCI Power Management Register Group

the following registers comprise the PCI Power Management Register Group.

**Table 5-7 PCI Power Management Registers - PCICFG F0h-FFh**

| **PCICFG F0h** | **Capabilities ID Register (RO)** | | | **Default = 01h** |
|---|---|---|---|---|
| This register always returns 01h to identify the Linked List item as being the PCI Power Management Registers. | | | | |
| **PCICFG F1h** | **Next Item Pointer Register (RO)** | | | **Default = 00h** |
| Value of 0 indicates no additional items in Capabilities List | | | | |
| **PCICFG F2h** | **Power Management Capabilities Register - Byte 0** | | | **Default = 01h** |
| Reserved | Device Specific Individualization<br>0 = No (always) | Reserved | Returns 001b to indicate Rev 1.0 of PCI Power Management Specification | |
| **PCICFG F3h** | **Power Management Capabilities Register - Byte 1** | | | **Default = 06h** |
| Reserved | | Supports D2 Power Management State<br>1 = Yes (always) | Support D1 Power Management State<br>1 = Yes (always) | Reserved |
| **PCICFG F4h** | **Power Management Control/Status Register - Byte 0** | | | **Default = 00h** |
| Reserved | | | Power State<br>00 = State D0<br>01 = State D1<br>10 = State D2<br>11 = State D3hot | |
| **PCICFG F5h** | **Power Management Control/Status Register - Byte 1** | | | **Default = 00h** |
| PME Status<br>0 = Inactive<br>1 = Active<br>Write 1 to clear | Data Register (not Implemented) | | | PME# PCI Function<br>0 = Disable<br>1 = Enabled |

**Table 5-7 PCI Power Management Registers - PCICFG F0h-FFh (cont.)**

| **PCICFG F6h** | **PCI-to-PCI Bridge Support Extensions Register (RO)** | **Default = C0h** |
|---|---|---|
| Bus Power Clock Control<br>1 = Enabled (always) | B2/B3 Support for D3hot<br>1 =Enable (always)* | Reserved |
| * Indicates that when Power State is programmed to D3hot, secondary PCI clocks will be stopped. | | |
| **PCICFG F7h** | **Data Register (RO)** | **Default = 00h** |
| Data Register not implemented | | |
| **PCICFG F8h-FFh** | **Reserved** | **Default = 00h** |

# 6.0 Electrical Ratings

Stresses above those listed in the following tables may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification are not implied.

## 6.1 Absolute Maximum Ratings

| Symbol | Parameter | 5.0 Volt | | 3.3 Volt | | Unit |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| VCC | Supply Voltage | | +6.5 | | +4.0 | V |
| VI | Input Voltage | –0.5 | VCC + 0.5 | –0.5 | VCC + 0.5 | V |
| VO | Output Voltage | –0.5 | VCC + 0.5 | –0.5 | VCC + 0.5 | V |
| TOP | Operating Temperature | 0 | +70 | 0 | +70 | °C |
| TSTG | Storage Temperature | –40 | +125 | –40 | +125 | °C |

## 6.2 DC Characteristics: VCC = 3.3V or 5.0V □5%, TA = 0°C to +70°C

| Symbol | Parameter | Min | Max | Unit | Condition |
|---|---|---|---|---|---|
| VIL | Input low Voltage | –0.5 | +0.8 | V | |
| VIH | Input high Voltage | +2.0 | VCC + 0.5 | V | |
| VOL | Output low Voltage | | +0.4 | V | IOL = 4.0mA |
| VOH | Output high Voltage | +2.4 | | V | IOH = -1.6mA |
| IIL | Input Leakage Current | | +10.0 | µA | VIN = VCC |
| IOZ | Tristate Leakage Current | | +10.0 | µA | |
| CIN | Input Capacitance | | +10.0 | pF | |
| COUT | Output Capacitance | | +10.0 | pF | |
| ICC | Power Supply Current<br>3.3V Core<br>5.0V Core | | <br>100<br>150 | mA | Fully active |

## 6.3 AC Characteristics

| Sym | Parameter | Min | Max | Unit | Figure |
|---|---|---|---|---|---|
| **Primary PCI Bus** | | | | | |
| t100 | C/BE[3:0]#, AD[31:0], FRAME#, IRDY#, TRDY#, STOP#, DEVSEL#, LOCK#, PAR, SERR#, PERR# setup time to PCICLK rising | 7 | | ns | 6-1 |
| t101 | C/BE[3:0]#, AD[31:0], FRAME#, IRDY#, TRDY#, STOP#, DEVSEL#, LOCK#, PAR, SERR#, PERR# hold time from PCICLK rising | 0 | | ns | 6-2 |
| t102 | C/BE[3:0]#, AD[31:0], FRAME#, IRDY#, TRDY#, STOP#, DEVSEL#, LOCK#, PAR, SERR#, PERR# valid delay from PCICLK rising | 2 | 11 | ns | 6-3 |
| t103 | REQ# setup time to PCICLK rising | 12 | | ns | 6-1 |
| t104 | REQ# hold time from PCICLK rising | 0 | | ns | 6-2 |
| t105 | GNT# valid delay from PCICLK rising | 2 | 12 | ns | 6-3 |
| **Secondary PCI Bus** | | | | | |
| t200 | CC/BE[3:0]#, CAD[31:0], CFRAME#, CIRDY#, CTRDY#, CSTOP#, CDEVSEL#, CBLOCK#, CPAR, CSERR#, CPERR# setup time to PCICLK rising | 7 | | ns | 6-1 |
| t201 | CC/BE[3:0]#, CAD[31:0], CFRAME#, CIRDY#, CTRDY#, CSTOP#, CDEVSEL#, CBLOCK#, CPAR, CSERR#, CPERR# hold time from PCICLK rising | 0 | | ns | 6-2 |
| t202 | CC/BE[3:0]#, CAD[31:0], CFRAME#, CIRDY#, CTRDY#, CSTOP#, CDEVSEL#, CBLOCK#, CPAR, CSERR#, CPERR# valid delay from PCICLK rising | 2 | 11 | ns | 6-3 |
| t203 | CREQ[3:0]# setup time to PCICLK rising | 12 | | ns | 6-1 |
| t204 | CREQ[3:0]# hold time from PCICLK rising | 0 | | ns | 6-2 |
| t205 | CGNT[3:0]# valid delay from PCICLK rising | 2 | 12 | ns | 6-3 |
| t206 | PCIRQ[3:0]# setup time to PCICLK rising | 5 | | ns | 6-1 |
| t207 | PCIRQ[3:0]# hold time from PCICLK rising | 3 | | ns | 6-2 |
| t208 | PCIRQ[3:0]# valid delay from PCICLK rising | 2 | 16 | ns | 6-3 |

## 6.4 AC Timing Diagrams

**Figure 6-1 Setup Timing Waveform**

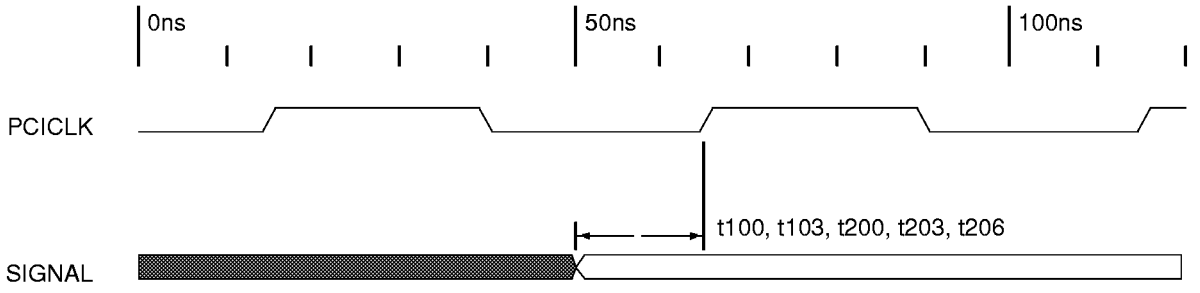



**Figure 6-2 Hold Timing Waveform**

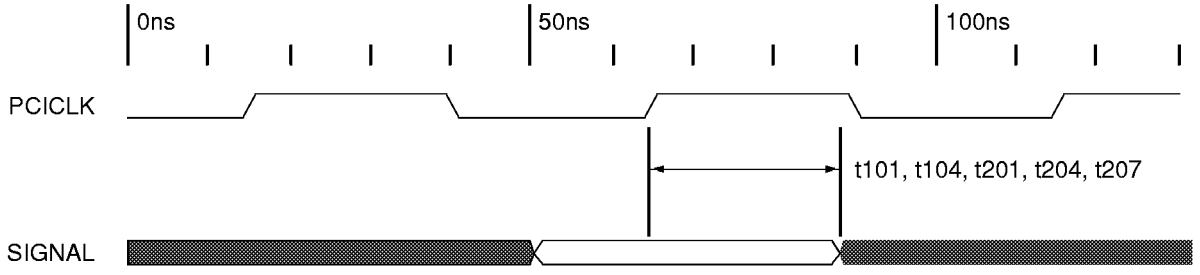
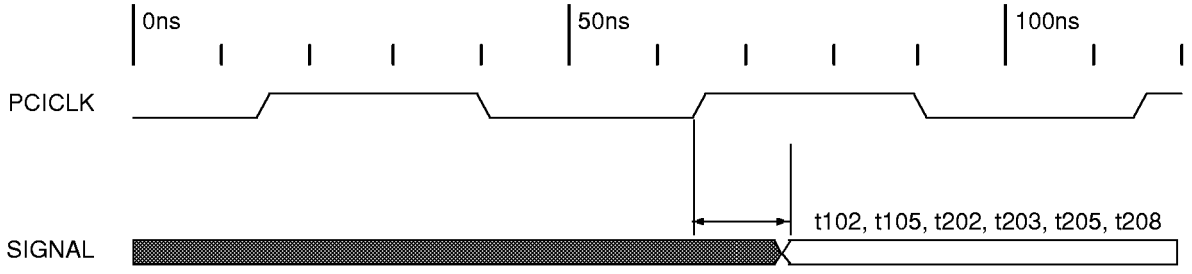
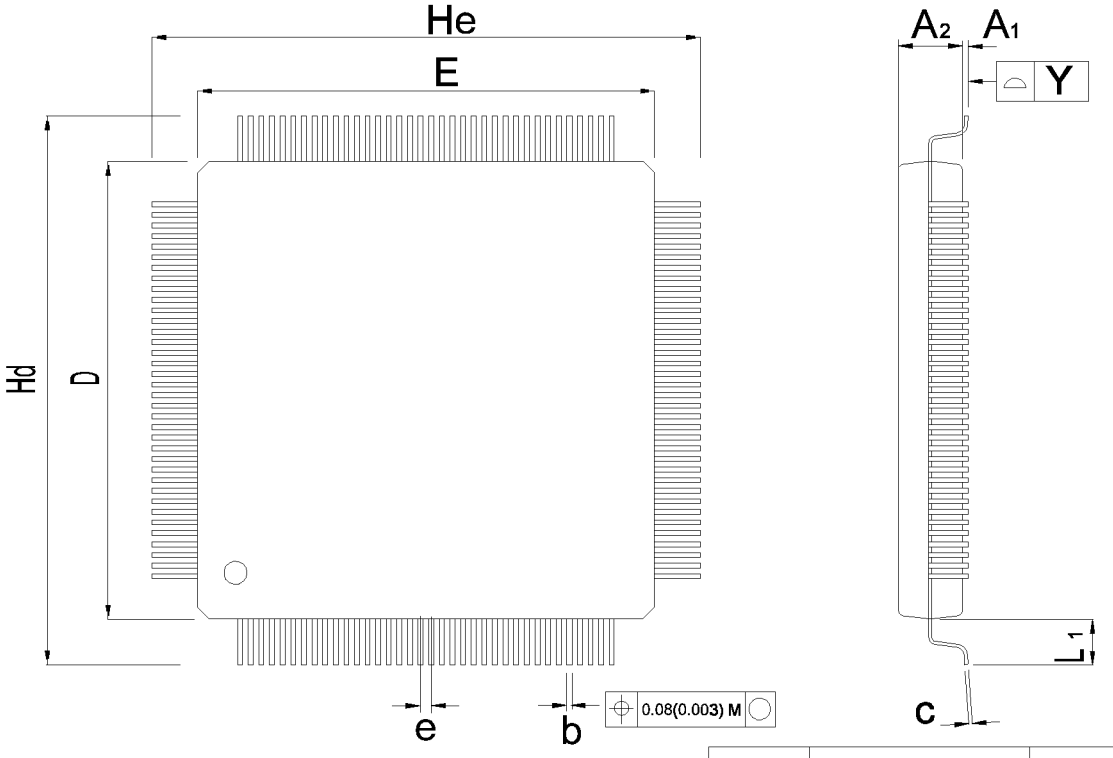



**Figure 6-3 Output Delay Timing Waveform**



## 7.0 Mechanical Package Outline

**Figure 7-1 144-Pin LQFP, Low-Profile Quad Flat Pack**




| SYMBOL | MILLIMETER | | | INCH | | |
|---|---|---|---|---|---|---|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| $A_1$ | 0.05 | 0.10 | 0.15 | 0.002 | 0.004 | 0.006 |
| $A_2$ | 1.35 | 1.40 | 1.45 | 0.053 | 0.055 | 0.057 |
| b | 0.17 | 0.22 | 0.27 | 0.007 | 0.009 | 0.011 |
| c | 0.090 | | 0.200 | 0.004 | | 0.008 |
| D | 19.90 | 20.00 | 20.10 | 0.783 | 0.787 | 0.791 |
| E | 19.90 | 20.00 | 20.10 | 0.783 | 0.787 | 0.791 |
| e | | 0.50 | | | 0.020 | |
| Hd | 21.90 | 22.00 | 22.10 | 0.862 | 0.866 | 0.870 |
| He | 21.90 | 22.00 | 22.10 | 0.862 | 0.866 | 0.870 |
| L | 0.45 | 0.60 | 0.75 | 0.018 | 0.024 | 0.030 |
| $L_1$ | | 1.00 | | | 0.039 | |
| Y | | | 0.08 | | | 0.003 |
| θ | 0 | | 7 | 0 | | 7 |

| Dwg. No.: AS144LQFP-001 | |
|---|---|
| Dwg. Rev.: 0 | Unit: MM / INCH |

# Appendix A IRQ Driveback Protocol

The OPTi PCI IRQ Driveback cycle provides a clean and simple way to convey interrupt and DMA status information to the host. The protocol is reliable and does not in any way compromise PCI compatibility.

1. Whenever a PCI peripheral device must signal an IRQ or SMI# to the system, it asserts its REQ# line to the host for one PCI clock, deasserts it for one PCI clock, then asserts it again and keeps it low until acknowledged.
2. The host recognizes this sequence as a high-priority request and immediately removes all other bus grants (GNT# lines). Once the previous bus owner is off the bus, the host acknowledges the high-priority request with GNT# as usual.
3. The peripheral device logic runs an I/O write cycle to the IRQ Driveback address specified in the PCI configuration registers, and releases REQ#.
4. The host latches the information on AD[31:0] and sets the IRQ lines appropriately.
5. An optional second burst data cycle can take place to convey additional interrupt information.

PCI-type devices on the secondary side of bridge chips can use this same protocol to convey their interrupt requests through the bridge to the host. The format of the driveback cycle request is illustrated in the figure. A second data phase is also possible.

## A.1 Driveback Cycle Format

The charts below illustrate the interrupt information indicated IRQ bits indicate whether that IRQ line is being driven high or low. The EN# bits indicate whether that IRQ is enabled to be changed or not. When the EN# bit is low, the value on the IRQ bit is valid. The device containing the central interrupt controller claims this I/O write cycle, and can then change its internal IRQ line state to match the value sent.

When a PCI device needs to generate an interrupt to the system, it runs a driveback cycle with the Enable bit low for each IRQ line under its control. For example, a device on PCI could run a driveback cycle with IRQ3 high and EN3# low to generate IRQ3 to the system. When the interrupt has been serviced and the device deasserts its interrupt, it starts another driveback cycle with IRQ3 low and EN3# low.

During both of these instances, if the device controls interrupts other than IRQ3, it must set its EN# bits low for **all** channels it controls, not just for the interrupt whose state has changed. The other IRQs must be driven with their previously used values.

**Figure A-1 IRQ Driveback Cycle High-Priority Request**

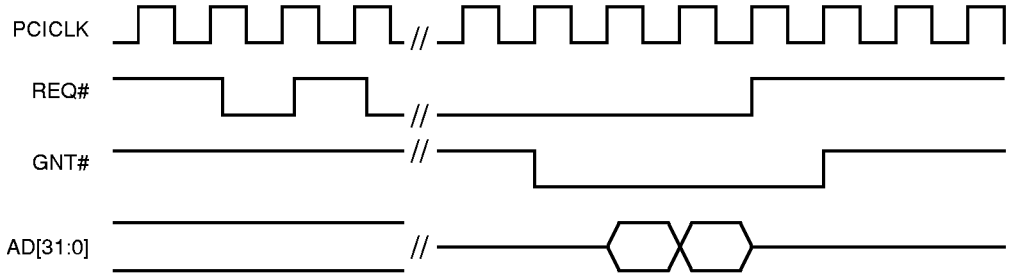



**Table A-1 Information Provided on a Driveback Cycle**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Low Word | AD15 | AD14 | AD13 | AD12 | AD11 | AD10 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 |
| | IRQ15 | IRQ14 | IRQ13 | IRQ12 | IRQ11 | IRQ10 | IRQ9 | IRQ8 | IRQ7 | IRQ6 | IRQ5 | IRQ4 | IRQ3 | IRQ2 | IRQ1 | IRQ0 |
| High Word | AD31 | AD30 | AD29 | AD28 | AD27 | AD26 | AD25 | AD24 | AD23 | AD22 | AD21 | AD20 | AD19 | AD18 | AD17 | AD16 |
| | EN15# | EN14# | EN13# | EN12# | EN11# | EN10# | EN9# | EN8# | EN7# | EN6# | EN5# | EN4# | EN3# | EN2# | EN1# | EN0# |

There is a convention for assignment of otherwise unusable IRQs:

- IRQ2 generates an SMI#. Note that the sense of IRQ2 is still active high. In this way, devices that use IRQ driveback can generate SMI# simply by routing their normal interrupt to IRQ2 without needing to change the polarity of the interrupt generation logic.
- IRQ13 generates an NMI. This feature allows PCI-to-ISA bridges such as the 82C825 chip to return the CHCK# signal from the ISA bus across the PCI bus. The sense of IRQ13 is active high.

Table A-2 illustrates the format of the optional second data phase of the IRQ driveback cycle. This phase is presently reserved for returning the PCI interrupts and ACPI Events. If the device needs to send back level-model interrupts, it bursts the information on the PCI clock following data phase one. The IRQ driveback address automatically increments to (base +4) per PCI requirements. It is also allowable for devices to drive back only phase 2, by directly accessing the (base +4) address.

**Table A-2 Information Provided on a Optional Data Phase 2 of IRQ Driveback Cycle**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Low Word | **AD15** | **AD14** | **AD13** | **AD12** | **AD11** | **AD10** | **AD9** | **AD8** | **AD7** | **AD6** | **AD5** | **AD4** | **AD3** | **AD2** | **AD1** | **AD0** |
| | Rsvd | Rsvd | Rsvd | Rsvd | Rsvd | Rsvd | Rsvd | Rsvd | ACPI3 | ACPI2 | ACPI1 | ACPI0 | PCIRQ 3 | PCIRQ 2 | PCIRQ 1 | PCIRQ 0 |
| High Word | **AD31** | **AD30** | **AD29** | **AD28** | **AD27** | **AD26** | **AD25** | **AD24** | **AD23** | **AD22** | **AD21** | **AD20** | **AD19** | **AD18** | **AD17** | **AD16** |
| | Rsvd | Rsvd | Rsvd | Rsvd | Rsvd | Rsvd | Rsvd | Rsvd | EN ACPI3# | EN ACPI2# | EN ACPI1# | EN ACPI0# | ENP3# | ENP2# | ENP1# | ENP0# |

## A.2 Edge vs Level Mode, IRQ Polarity

The IRQs driven back in data phase 1 are interpreted as edge-mode interrupts, as expected for AT compatibility. The AD[15:0] signals are interpreted as active when high (1); the Enable (EN#) signals AD[31:16] are active when low (0).

In optional data phase 2, the PCIRQ0-3 bits are interpreted as level-mode interrupts by the host hardware. As with data phase 1, the controls indicated by AD[15:0] are interpreted as active when **high**; the Enable (EN#) controls on AD[31:16] are active when **low**. Note that PCI signals INTA-D# are active low by definition.

## A.3 Host Handling of IRQ Driveback Information

The host chipset must handle the IRQ driveback information differently depending on whether the selected interrupt is sharable or not. Generally the ISA IRQ lines need no special consideration.

However, the INTA-D# lines can be shared by multiple devices on the PCI bus. Thus, one device could perform an IRQ driveback to set the INTx# line active for its purposes, while another device could follow immediately by setting the same INTx# line inactive. Therefore, the host is required to implement a counter in this case, so that it considers the line inactive only after it has received the same number of active-going drivebacks as it has inactive-going drivebacks.

A three-bit counter can be considered sufficient to handle the situation, since this would allow up to seven devices to chain to the same interrupt. It is unlikely that system requirements would exceed this number given the latency penalty incurred.

# Sales Information

## HEADQUARTERS:

**OPTi Inc.**
888 Tasman Drive
Milpitas, CA 95035
tel: 408-486-8000
fax: 408-486-8011

## SALES OFFICES:

### Japan
**OPTi Japan KK**
Murata Building 6F, 2-22-7
Ohhashi Meguro-ku
Tokyo 153, Japan
tel: 81-3-5454-0178
fax: 81-3-5454-0168

### Taiwan
**OPTi Inc.**
9F, No 303, Sec 4, Hsin Yih Road
Taipei, Taiwan, ROC
tel: 886-2-325-8520
fax: 886-2-325-6520

### United States
**OPTi Inc.**
20405 State Highway 249, Ste. #220
Houston, TX 77070
tel: 281-257-1856
fax: 281-257-1825

## REPRESENTATIVES:

## United States

### Alabama/Mississippi
**Concord Component Reps**
190 Line Quarry Rd., Ste. #102
Madison, AL 35758
tel: 205-772-8883
fax: 205-772-8262

### Florida
**Engineered Solutions Ind., Inc.**
1000 E. Atlantic Blvd., Ste. #202
Pompano Beach, FL 33060
tel: 305-784-0078
fax: 305-781-7722

### Georgia
**Concord Component Reps**
6825 Jimmy Carter Blvd., Ste. #1303
Norcross, GA 30071
tel: 770-416-9597
fax: 770-441-0790

### Illinois
**Micro-Tex, Inc.**
1870 North Roselle Rd., Ste. #107
Schaumburg, IL 60195-3100
tel: 708-885-8200
fax: 708-885-8210

### Massachusetts
**S-J Associates, Inc.**
267 Boston Road
Corporate Place, Ste. #3
N. Billerica, MA 01862
tel: 508-670-8899
fax: 508-670-8711

### Michigan
**Jay Marketing**
44752 Helm Street., Ste. A
Plymouth, MI 48170
tel: 313-459-1200
fax: 313-459-1697

### New Jersey
S-J Associates, Inc.
131-D Gaither Dr.
Mt. Laurel, NJ 08054
tel: 609-866-1234
fax: 609-866-8627

### New York
**S-J Associates, Inc.**
265 Sunrise Highway
Rockville Centre, NY 11570
tel: 516-536-4242
fax: 516-536-9638

**S-J Associates, Inc.**
735 Victor-Pittsford
Victor, NY 14564
tel: 716-924-1720

### North & South Carolina
**Concord Component Reps**
10608 Dunhill Terrace
Raleigh, NC 27615
tel: 919-846-3441
fax: 919-846-3401

### Ohio/W. Pennsylvania
**Lyons Corp.**
4812 Fredrick Rd., Ste. #101
Dayton, OH 45414
tel: 513-278-0714
fax: 513-278-3609

**Lyons Corp.**
4615 W. Streetsboro
Richfield, OH 44286
tel: 216-659-9224
fax: 216-659-9227

**Lyons Corp.**
248 N. State St.
Westerville, OH 43081
tel: 614-895-1447
fax: Same

### Texas
**Axxis Technology Marketing, Inc.**
701 Brazos, Suite 500
Austin, TX 78701
tel: 512-320-9130
fax: 512-320-5730

**Axxis Technology Marketing, Inc.**
6804 Ashmont Drive
Plano, TX 75023
tel: 214-491-3577
fax: 214-491-2508

### Virginia
**S-J Associates, Inc.**
900 S. Washington St., Ste. #307
Falls Church, VA 22046
tel: 703-533-2233
fax: 703-533-2236

### Wisconsin
**Micro-Tex, Inc.**
22660 Broadway, Ste. #4A
Waukesha, WI 53186
tel: 414-542-5352
fax: 414-542-7934

## International

### Australia
**Braemac Pty. Ltd.**
Unit 6, 111 Moore St., Leichhardt
Sydney, 2040 Australia
tel: 61-2-550-6600
fax: 61-2-550-6377

### China
**Legend Electronic Components. Ltd.**
Unit 413, Hong Kong Industrial
Technology Centre
72 Tat Chee Avenue
Kowloon Tong, Hong Kong
tel: 852-2776-7708
fax: 852-2652-2301

### France
**Tekelec Airtronic, France**
5, Rue Carle Vernet
92315 Sevres Cedex
France
tel: 33-1-46-23-24-25
fax: 33-1-45-07-21-91

### Germany
**Kamaka**
Rheinsrasse 22
76870 Kandel
Germany
tel: 49-7275-958211
fax: 49-7275-958220

### India
**Spectra Innovation**
Unit S-822 Manipal Centre
47 Dickenson Road
Bangalore 560-042
Kamataka, India
tel: 91-80-558-8323/3977
fax: 91-80-558-6872

### Israel
**Ralco Components (1994) Ltd.**
11 Benyamini St.
67443 Tel Aviv
Israel
tel: 972-3-6954126
fax: 972-3-6951743

### Korea
**Woo Young Tech Co., Ltd.**
5th Floor Koami Bldg
13-31 Yoido-Dong
Youngduengpo-Ku
Seoul, Korea 150-010
tel: 02-369-7099
fax: 02-369-7091

### Singapore
**Instep Microsolutions Pte Ltd.**
18, Tannery Lane, #05-02
Lian Tong Building
Singapore 347780
tel: 65-741-7507
fax: 65-741-1478

### South America
**Uniao Digital**
Rua Guido Caloi
Bloco B, Piso 3
Sao Paulo-SP, CEP 05802-140 Brazil
tel: 55-11-5514-3355
fax: 55-11-5514-1088

### Switzerland
**Datacomp AG**
Silbernstrasse 10
8953 Dietikon
Switzerland
tel: 41-1-740-5140
fax: 41-1-741-3423

### United Kingdom
**Spectrum**
2 Grange Mews,
Station Road
Launton, Bicester
Oxfordshire,OX6 0DX
UK
tel: 44-1869-325174
fax: 44-1869-325175

**MMD**
3 Bennet Court,
Bennet Road
Reading
Berkshire, RG2 0QX
UK
tel: 44 1734 313232
fax: 44 1734 313255



June 27, 1997

**OPTi Inc.**
888 Tasman Drive
Milpitas, CA 95035
Tel: (408) 486-8000
Fax: (408) 486-8001
www.opti.com