OPTi

FireFox

82C824
32-Bit PC Card Controller

Preliminary Data Book

Revision: 3.0
912-2000-013
August 26, 1996

Disclaimer

OPTi Inc. makes no representations or warranties with respect to the design and documentation herein described and especially disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, OPTi Inc. reserves the right to revise the design and associated documentation and to make changes from time to time in the content without obligation of OPTi Inc. to notify any person of such revisions or changes.

Note: Before designing contact OPTi for latest Product Alerts, Applications Notes, and Errata for this product line.

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

Table of Contents

Table of Contents (cont.)

Table of Contents (cont.)

List of Figures

List of Tables

List of Tables (cont.)

32-Bit PC Card Controller

1.0 Features

PC Card

- Compliant with PC Card '95 specification
- Full DMA support, any channel
- Each slot can have separate channel
- 3.3V cards supported
- Industry-standard '365 register set
- Any IRQ can be selected, not just subset

CardBus Card

- Compliant with PC Card '95 specification
- Follows “Yenta” standard
- Synchronous or asynchronous operation
- Fully buffered with 8-level FIFO
- IRQs can be selected separately for status change if desired

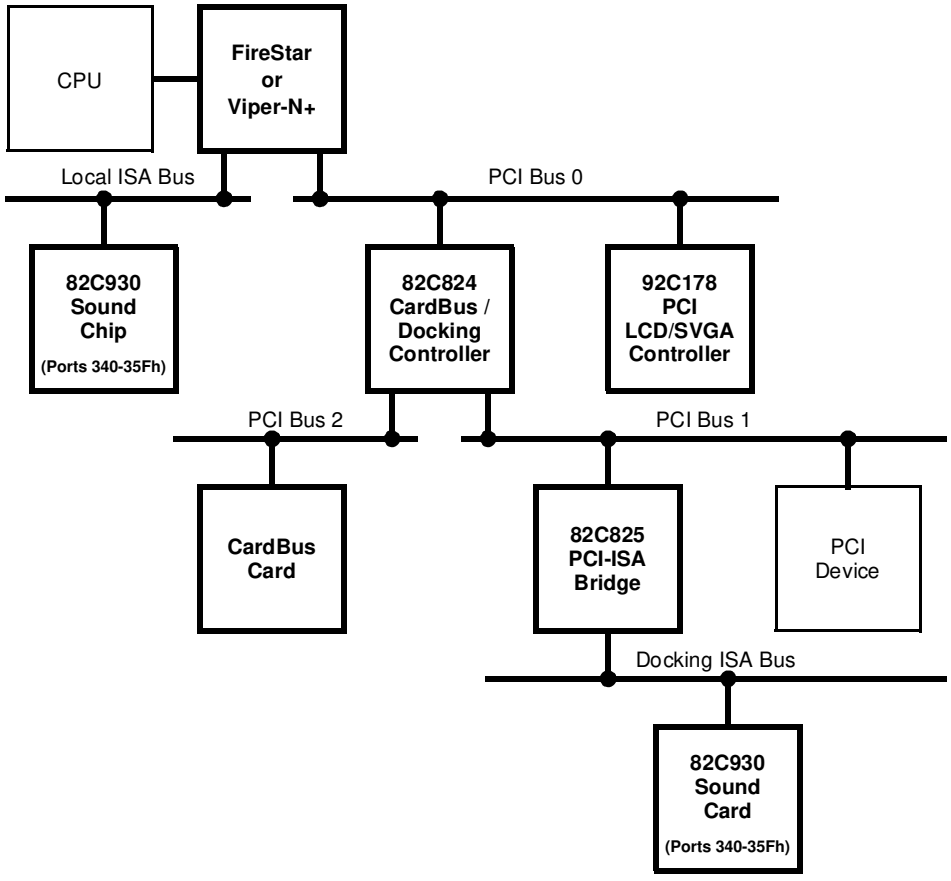
Docking Station

- Supports 3.3V or 5.0V PCI dock
- I/O buffers automatically adjust to PCI drive requirements
- Provides eight windows, selectable for memory or I/O, upstream or downstream
- Offers additional fixed window for VGA
- Supports PCIRQ[2:0]#
- Supports three bus masters
- Bridge solution increases primary PCI bus bandwidth by off-loading transactions into buffers

Packaging

- 208-pin PQFP (Plastic Quad Flat Pack)

Figure 1-1 Multiple ISA Bus Support



2.0 Overview

This document describes a completely new OPTi interface chipset that is intended to take full advantage of the CardBus interface according to the new "PC Card '95" standard. At the same time, this device extends PCI functionality to make it an essential component for any PCI-based notebook system.

CardBus is the scheme whereby the 68-pin connector used to interface with PCMCIA 2.0 cards is reconfigured before power-up to apply a complete 32-bit, 3.3V PCI interface to plug-in cards. The host interface side of the OPTi solution described herein is based on the PCI bus with special extensions; the card interface side is automatically switched between PCMCIA and CardBus, depending on the card type detected at card insertion, or can be assigned as strictly PCI.

The OPTi CardBus Controller chipset implements two identical PCI-to-PCI bridges. Both bridges have the option of interfacing directly to a secondary external PCI pinout or to a single-slot PCMCIA controller which then becomes the external pinout. The PCMCIA interface is handled in a straightforward manner using a modified Intel 82365SL PCMCIA core.

CardBus Nomenclature. The CardBus specification refers to the PCMCIA interface as “R2”, indicating a Rev. 2.1 compliant PCMCIA slot. This document uses the R2 reference when discussing a slot configured for PCMCIA operation, including the current (PC Card '95) standard. The term PCI will be used to describe the cycle that occurs with a slot configured for a CardBus card. Upon detecting card insertion, the OPTi CardBus Controller chipset checks special pins provided to identify the card and reconfigures the interface to either R2 or PCI before executing any card power-up commands.

3.0 Signal Definitions

The 82C824 chip runs CardBus cycles, PCI cycles, and PCMCIA cycles. The 82C824 chip provides a primary interface which is PCI-based. It also provides two independent attachment interfaces, each of which can be switched dynamically between CardBus and PCMCIA. One interface can also be switched to a PCI interface.

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 82C824 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

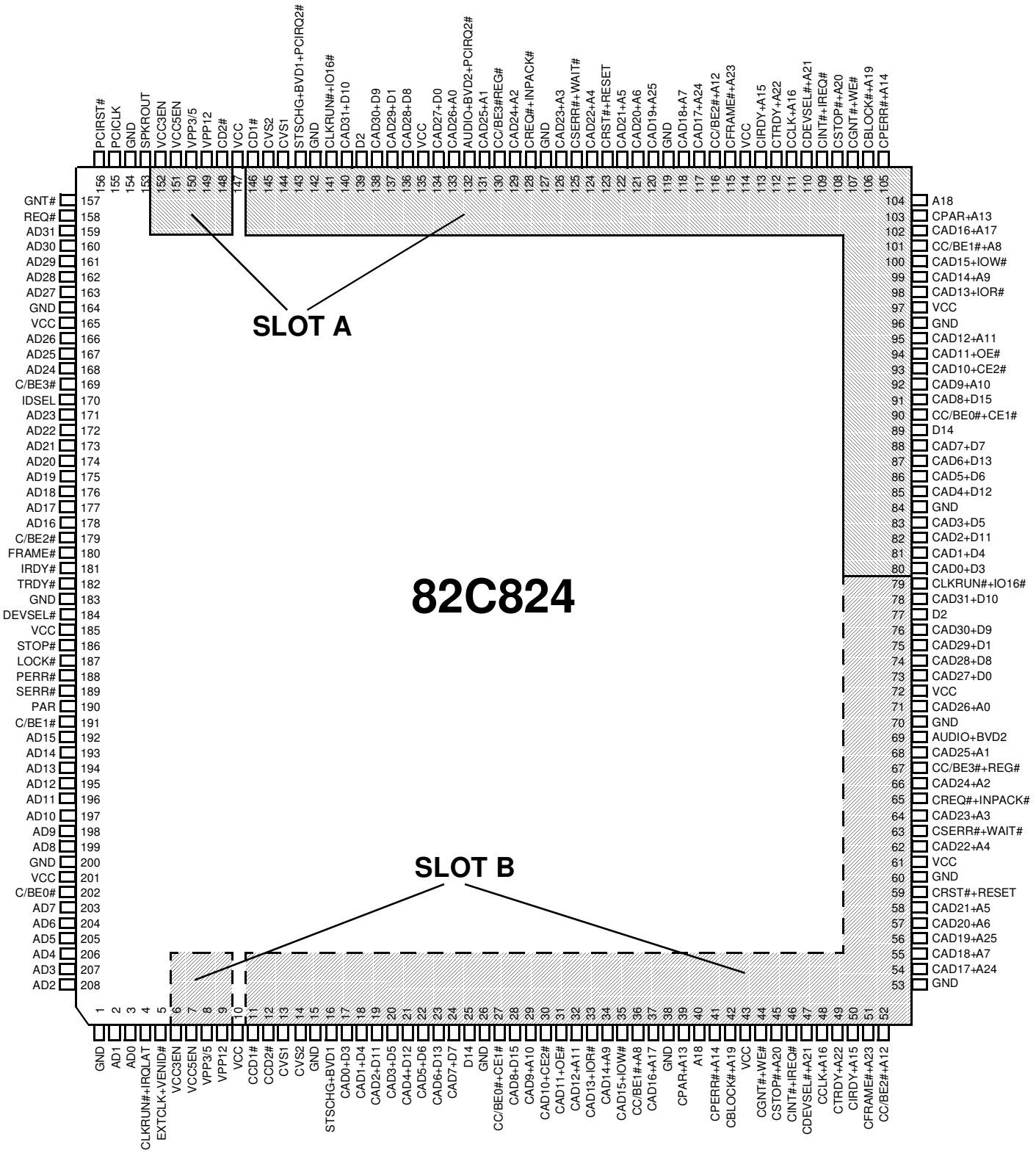


Table 3-2 Numerical Pin Cross-Reference List

Pin No.	Pin Name	Slot
1	GND	
2	AD1	
3	AD0	
4	CLKRUN#	
	IRQLAT	
5	EXTCLK	
	VENID#	
6	VCC3EN	B
7	VCC5EN	B
8	VPP3/5	B
9	VPP12	B
10	VCC	
11	CCD1#	B
12	CCD2#	B
13	CVS1	B
14	CVS2	B
15	GND	B
16	STSCHG	B
	BVD1	
17	CAD0	B
	D3	
18	CAD1	B
	D4	
19	CAD2	B
	D11	
20	CAD3	B
	D5	
21	CAD4	B
	D12	
22	CAD5	B
	D6	
23	CAD6	B
	D13	
24	CAD7	B
	D7	
25	D14	B
26	GND	B
27	CC/BE0#	B
	CE1#	
28	CAD8	B
	D15	
29	CAD9	B
	A10	
30	CAD10	B
	CE2#	
31	CAD11	B
	OE#	
32	CAD12	B
	A11	
33	CAD13	B
	IOR#	
34	CAD14	B
	A9	
35	CAD15	B
	IOW#	
36	CC/BE1#	B
	A8	
37	CAD16	B
	A17	
38	GND	B
39	CPAR	B
	A13	
40	A18	B
41	CPERR#	B
	A14	
42	CBLOCK#	B
	A19	
43	VCC	B
44	CGNT#	B
	WE#	
45	CSTOP#	B
	A20	
46	CINT#	B
	IREQ#	
47	CDEVSEL#	B
	A21	
48	CCLK	B
	A16	
49	CTRDY	B
	A22	
50	CIRDY	B
	A15	
51	CFRAME#	B
	A23	
52	CC/BE2#	B
	A12	
53	GND	B
54	CAD17	B
	A24	
55	CAD18	B
	A7	
56	CAD19	B
	A25	
57	CAD20	B
	A6	
58	CAD21	B
	A5	
59	CRST#	B
	RESET	
60	GND	B
61	VCC	B
62	CAD22	B
	A4	
63	CSERR#	B
	WAIT#	
64	CAD23	B
	A3	
65	CREQ#	B
	INPACK#	
66	CAD24	B
	A2	
67	CC/BE3#	B
	REG#	
68	CAD25	B
	A1	
69	AUDIO	B
	BVD2	
70	GND	B
71	CAD26	B
	A0	
72	VCC	B
73	CAD27	B
	D0	
74	CAD28	B
	D8	
75	CAD29	B
	D1	
76	CAD30	B
	D9	
77	D2	B
78	CAD31	B
	D10	
79	CLKRUN#	B
	IO16#	
80	CAD0	B
	D3	
81	CAD1	A
	D4	
82	CAD2	A
	D11	
83	CAD3	A
	D5	
84	GND	A
85	CAD4	A
	D12	
86	CAD5	A
	D6	
87	CAD6	A
	D13	
88	CAD7	A
	D7	
89	D14	A
90	CC/BE0#	A
	CE1#	
91	CAD8	A
	D15	
92	CAD9	A
	A10	
93	CAD10	A
	CE2#	
94	CAD11	A
	OE#	
95	CAD12	A
	A11	
96	GND	A
97	VCC	A
98	CAD13	A
	IOR#	
99	CAD14	A
	A9	
100	CAD15	A
	IOW#	
101	CC/BE1#	A
	A8	
102	CAD16	A
	A17	
103	CPAR	A
	A13	
104	A18	A
105	CPERR#	A
	A14	
106	CBLOCK#	A
	A19	
107	CGNT#	A
	WE#	
108	CSTOP#	A
	A20	
109	CINT#	A
	IREQ#	
110	CDEVSEL#	A
	A21	
111	CCLK	A
	A16	
112	CTRDY	A
	A22	
113	CIRDY	A
	A15	
114	VCC	A
115	CFRAME#	A
	A23	
116	CC/BE2#	A
	A12	
117	CAD17	A
	A24	
118	CAD18	A
	A7	
119	GND	A
120	CAD19	A
	A25	
121	CAD20	A
	A6	
122	CAD21	A
	A5	
123	CRST#	A
	RESET	
124	CAD22	A
	A4	
125	CSERR#	A
	WAIT#	
126	CAD23	A
	A3	
127	GND	A
128	CREQ#	A
	INPACK#	
129	CAD24	A
	A2	
130	CC/BE3#	A
	REG#	
131	CAD25	A
	A1	
132	AUDIO	A
	BVD2	
	PCIRQ1#	
133	CAD26	A
	A0	
134	CAD27	A
	D0	
135	VCC	A
136	CAD28	A
	D8	
137	CAD29	A
	D1	
138	CAD30	A
	D9	
139	D2	A
140	CAD31	A
	D10	
141	CLKRUN#	A
	IO16#	
142	GND	A
143	STSCHG	A
	BVD1	
	PCIRQ2#	
144	CVS1	A
145	CVS2	A
146	CD1#	A
147	VCC	
148	CD2#	A
149	VPP12	A
150	VPP3/5	A
151	VCC5EN	A
152	VCC3EN	A
153	SPKROUT	
154	GND	
155	PCICLK	
156	PCIRST#	
157	GNT#	
158	REQ#	
159	AD31	
160	AD30	
161	AD29	
162	AD28	
163	AD27	
164	GND	
165	VCC	
166	AD26	
167	AD25	
168	AD24	
169	C/BE3#	
170	IDSEL	
171	AD23	
172	AD22	
173	AD21	
174	AD20	
175	AD19	
176	AD18	
177	AD17	
178	AD16	
179	C/BE2#	
180	FRAME#	
181	IRDY#	
182	TRDY#	
183	GND	
184	DEVSEL#	
185	VCC	
186	STOP#	
187	LOCK#	
188	PERR#	
189	SERR#	
190	PAR	
191	C/BE1#	
192	AD15	
193	AD14	
194	AD13	
195	AD12	
196	AD11	
197	AD10	
198	AD9	
199	AD8	
200	GND	
201	VCC	
202	C/BE0#	
203	AD7	
204	AD6	
205	AD5	
206	AD4	
207	AD3	
208	AD2	

Table 3-3 Alphabetical Pin Cross-Reference List

Pin Name	Pin No.	Slot
A18	40	B
A18	104	A
AD0	3	
AD1	2	
AD10	197	
AD11	196	
AD12	195	
AD13	194	
AD14	193	
AD15	192	
AD16	178	
AD17	177	
AD18	176	
AD19	175	
AD2	208	
AD20	174	
AD21	173	
AD22	172	
AD23	171	
AD24	168	
AD25	167	
AD26	166	
AD27	163	
AD28	162	
AD29	161	
AD3	207	
AD30	160	
AD31	159	
AD4	206	
AD5	205	
AD6	204	
AD7	203	
AD8	199	
AD9	198	
AUDIO+BVD2	69	B
AUDIO+BVD2+ PCIRQ1#	132	A
C/BE0#	202	
C/BE1#	191	
C/BE2#	179	
C/BE3#	169	
CAD0+D3	17	B
CAD0+D3	80	A
CAD1+D4	18	B
CAD1+D4	81	A
CAD10+CE2#	30	B
CAD10+CE2#	93	A
CAD11+OE#	31	B
CAD11+OE#	94	A
CAD12+A11	32	B
CAD12+A11	95	A
CAD13+IOR#	33	B
CAD13+IOR#	98	A

Pin Name	Pin No.	Slot
CAD14+A9	34	B
CAD14+A9	99	A
CAD15+IOW#	35	B
CAD15+IOW#	100	A
CAD16+A17	37	B
CAD16+A17	102	A
CAD17+A24	54	B
CAD17+A24	117	A
CAD18+A7	55	B
CAD18+A7	118	A
CAD19+A25	56	B
CAD19+A25	120	A
CAD2+D11	19	B
CAD2+D11	82	A
CAD20+A6	57	B
CAD20+A6	121	A
CAD21+A5	58	B
CAD21+A5	122	A
CAD22+A4	62	B
CAD22+A4	124	A
CAD23+A3	64	B
CAD23+A3	126	A
CAD24+A2	66	B
CAD24+A2	129	A
CAD25+A1	68	B
CAD25+A1	131	A
CAD26+A0	71	B
CAD26+A0	133	A
CAD27+D0	73	B
CAD27+D0	134	A
CAD28+D8	74	B
CAD28+D8	136	A
CAD29+D1	75	B)
CAD29+D1	137	A
CAD3+D5	20	B
CAD3+D5	83	A
CAD30+D9	76	B
CAD30+D9	138	A
CAD31+D10	78	B
CAD31+D10	140	A
CAD4+D12	21	B
CAD4+D12	85	A
CAD5+D6	22	B
CAD5+D6	86	A
CAD6+D13	23	B
CAD6+D13	87	A
CAD7+D7	24	B
CAD7+D7	88	A
CAD8+D15	28	B
CAD8+D15	91	A
CAD9+A10	29	B
CAD9+A10	92	A
CBLOCK#+A19	42	B

Pin Name	Pin No.	Slot
CBLOCK#+A19	106	A
CC/BE0#+CE1#	27	B
CC/BE0#+CE1#	90	A
CC/BE1#+A8	36	B
CC/BE1#+A8	101	A
CC/BE2#+A12	52	B
CC/BE2#+A12	116	A
CC/BE3#+REG#	67	B
CC/BE3#+REG#	130	A
CCD1#	11	B
CCD2#	12	B
CCLK / A16	48	B
CCLK / A16	111	A
CD1#	146	A
CD2#	148	A
CDEVSEL#+A21	47	B
CDEVSEL#+A21	110	A
CFRAME#+A23	51	B
CFRAME#+A23	115	A
CGNT#+WE#	44	B
CGNT#+WE#	107	A
CINT#+IREQ#	46	B
CINT#+IREQ#	109	A
CIRDY+A15	50	B
CIRDY+A15	113	A
CLKRUN#+IO16#	79	B
CLKRUN#+IO16#	141	A
CLKRUN#+IRQLAT	4	
CPAR+A13	39	B
CPAR+A13	103	A
CPERR#+A14	41	B
CPERR#+A14	105	A
CREQ#+INPACK#	65	B
CREQ#+INPACK#	128	A
CRST#+RESET	59	B
CRST#+RESET	123	A
CSERR#+WAIT#	63	B
CSERR#+WAIT#	125	A
CSTOP#+A20	45	B
CSTOP#+A20	108	A
CTRDY+A22	49	B
CTRDY+A22	112	A
CVS1	13	B
CVS1	144	A
CVS2	14	B
CVS2	145	A
D14	25	B
D14	89	A
D2	77	B
D2	139	A
DEVSEL#	184	
EXTCLK+VENID#	5	
FRAME#	180	

Pin Name	Pin No.	Slot
GND	1	
GND	15	B
GND	26	B
GND	38	B
GND	53	B
GND	60	B
GND	70	B
GND	84	A
GND	96	A
GND	119	A
GND	127	A
GND	142	A
GND	154	
GND	164	
GND	183	
GND	200	
GNT#	157	
IDSEL	170	
IRDY#	181	
LOCK#	187	
PAR	190	
PCICLK	155	
PCIRST#	156	
PERR#	188	
REQ#	158	
SERR#	189	
SPKROUT	153	
STOP#	186	
STSCHG+BVD1	16	B
STSCHG+BVD1+ PCIRQ2#	143	A
TRDY#	182	
VCC	10	
VCC	43	B
VCC	61	B
VCC	72	B
VCC	97	A
VCC	114	A
VCC	135	A
VCC	147	
VCC	165	
VCC	185	
VCC	201	
VCC3EN	6	B
VCC3EN	152	A
VCC5EN	7	B
VCC5EN	151	A
VPP12	9	B
VPP12	149	A
VPP3/5	8	B
VPP3/5	150	A

3.2 Signal Descriptions

3.2.1 Primary PCI Interface Signals

The table below describes the function of each of the host interface PCI signals in the normal slave mode of the interface. If the 82C824 becomes the master and the host becomes the slave, the direction of signals is reversed.

Host Interface PCI Signals

Signal Name	Pin No.	Pin Type	Signal Description
AD[31:0]	159:163, 166:168, 171:178, 192:199, 203:208, 2, 3	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.
C/BE[3:0]#	169, 179, 191, 202	I/O	**Bus Command and Byte Enables 3 through 0:** These signals provide the command type information during the address phase and carry the byte enable information during the data phase.
PAR	190	I/O	**Parity:** This bit carries parity information for both the address and data phases of PCI cycles.
PCICLK	155	I	**PCI Clock:** Provides timing for all transactions on the host PCI bus; normally 33MHz. This same clock is buffered for timing the slot interface, PCI bus, or can be divided. The slot interfaces can also run from the alternative EXTCLK input.
EXTCLK	5	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.
VENID#		O	**Drive Vendor ID:** If selected by strap option, 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.
CLKRUN#	4	I/O OD	**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 82C824 uses this signal to request a restart of PCICLK in order to send a bus master request.
IRQLAT		O	**Interrupt Latch:** For use on chipsets without IRQ driveback capability, the 82C824 logic can drive this line high to drive IRQ lines using an external latch.
FRAME#	180	I/O	**Cycle Frame:** Driven by PCI bus masters to indicate the beginning and duration of an access.
IRDY#	181	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#	182	I/O	**Target Ready:** Asserted by the PCI bus target (the 82C824 when it 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 82C824, which in turn drives TRDY# to the host. The 82C824 logic drives TRDY# directly for 82C824 configuration register accesses.
STOP#	186	I/O	**Stop:** Used by the target to request that the master stop the current transaction and retry it later. The 82C824 logic uses this mechanism to back-off from a claimed cycle and generate an SMI through the IRQ driveback cycle, for example.

Host Interface PCI Signals (cont.)

Signal Name	Pin No.	Pin Type	Signal Description
LOCK#	187	I/O	**Lock:** Indicates an atomic operation that may require multiple transactions to complete. The signal can be asserted to the 82C824 by any host bus PCI master, and is driven by the 82C824 logic in response to the current slot interface bus master driving its CBLOCK# signal.
DEVSEL#	184	I/O	**Device Select:** Driven by the 82C824 logic when it decodes its address as the target of the current access via either positive or subtractive decoding.
PERR#	188	O	**Parity Error:** All devices use this signal to report data parity errors during any PCI transaction except a Special Cycle.
SERR#	189	O OD	**System Error:** The 82C824 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#	158	O	**Bus Request:** The 82C824 logic uses this signal to gain control of the PCI bus. The logic also uses this pin to generate an interrupt driveback request.
GNT#	157	I	**Bus Grant:** The system grants the bus to the 82C824 chip using this signal.
IDSEL	170	I	**ID Select:** This signal is the "chip select" for the controller configuration registers. Normally this input simply connects to one of the upper address lines to select the controller for the current cycle.
SPKROUT	153	O	**Speaker Output:** Slot interface devices can return audio information to the system on this line.

3.2.2 Slot Interface - Common PCMCIA/CardBus Interface Signals

The interface signals common to both PCMCIA and CardBus are shown below.

82C824 Common Slot Interface Signals

Signal Name	Slot A/C Pin No.	Slot B/D Pin No.	Pin Type	Signal Description
CCD1#	146	11	I/O	**Card Detect 1**
CCD2#	148	12	I/O	**Card Detect 2**
CVS1	144	13	I	**Voltage Sense 1**
CVS2	145	14	I	**Voltage Sense 2**
AUDIO	132	69	I	**Audio Input** (AUDIO - CB)
SPKR			I	**Speaker Input** (SPKR - R2 I/O card)
BVD2			I	**Battery Low Voltage Detect pin 2** (BVD2 - R1 or R2 memory card)
PCIRQ1#				**PCI Interrupt 1**
STSCHG	143	16	I	**Status Change Interrupt,** active high (STSCHG - CB)
STSCHG#			I	**Status Change Interrupt,** active low (STSCHG# - R2 I/O card)
BVD1			I	**Battery Low Voltage Detect pin 1** (BVD1 - R1 or R2 memory card)
PCIRQ2#				**PCI Interrupt 2**

82C824 Common Slot Interface Signals (cont.)

Signal Name	Slot A/C Pin No.	Slot B/D Pin No.	Pin Type	Signal Description
CINT#	109	46	I	**Interrupt Request** (CINT# - CB)
IREQ			I	**Interrupt Request** (IREQ - R2 I/O card)
RDY/BSY#			I	**Ready/Busy** (RDY/BSY# - R1 or R2 memory card)
CRST#	123	59	O	**Card Reset,** active low (CB cards)
RESET			O	**Card Reset,** active high (R2 cards)
VCC5EN	151	7	O	**5.0V VCC Enable**
VCC3EN	152	6	O	**3.3V VCC Enable**
VPP12	149	9	O	**12V VPP Enable:** These pins are also strap options for primary/secondary 82C824 selection and local/docking station 82C824 selection. Refer to the "Strap-Selected Interface Options" section for details.
VPP3/5	150	8	O	**VPP Enable as currently selected VCC**

3.2.3 Slot Interface - CardBus Configuration

The signals listed below are standard CardBus signals. The signal names indicated are valid only when the slot has been configured for CardBus use; the pins change function when a PCMCIA card or docking station is connected. If the CardBus card becomes a bus master, the 82C824 signal directions are reversed from the description below.

82C824 Slot Interface Pins - Dynamically Reconfigured for CardBus

Signal Name	Slot A/C Pin No.	Slot B/D Pin No.	Type	Signal Description
CAD[31:0]	140, 138:136, 134, 133, 131, 129, 126, 124, 122:120, 118, 117, 102, 100:98, 95:91, 88:85, 83:80	78, 76:73, 71, 68, 66, 64, 62, 58:54, 37, 35:28, 24:17	I/O	**Multiplexed Address and Data Lines 31 through 0:** These pins are the multiplexed CB address and data lines. During the address phase, these pins are outputs for CB slave cycles and inputs for CB master cycles. During the data phase, these pins are outputs during CB write cycles and inputs during CB reads.
CC/BE[3:0]#	130, 116, 101, 90	67, 52, 36, 27	I/O	**Bus Command and Byte Enables 3 through 0:** These pins are the multiplexed CB command and byte enable lines. Normally outputs, these pins are inputs during CB master cycles.
CPAR	103	39	I/O	**Parity:** This signal is an input either during CB slave cycles for address and write data phases or during CB master cycle for read data phase; otherwise it is an output.
CCLK	111	48	O	**Clock:** This signal is used to provide timing for all CB transactions. The clock is derived either directly from the host PCI interface signal PCICLK (usually 33MHz), or a divided version. It can also be derived from the external input clock to the 82C824 chip.

82C824 Slot Interface Pins - Dynamically Reconfigured for CardBus (cont.)

Signal Name	Slot A/C Pin No.	Slot B/D Pin No.	Type	Signal Description
CCLKRUN#	141	79	I	**Clock Run:** This signal is pulled low by the CardBus card needing to use the bus. If this pin is not pulled low, the slot interface bus controller is allowed to stop the PCICLK signal.
CFRAME#	115	51	I/O	**Cycle Frame:** The 82C824 drives this signal to indicate the beginning and duration of an access.
CIRDY#	113	50	I/O	**Initiator Ready:** The 82C824 drives this signal to indicate its ability to complete the current data phase of the transaction.
CTRDY#	112	49	I/O	**Target Ready:** The 82C824 monitors this input from the slot interface slave device to determine when it can complete the cycle. PCI-type devices on the slot interfaces return CTRDY# to the 82C824 which in turn drives host TRDY#.
CSTOP#	108	45	I/O	**Stop:** This signal is used by the target to request the master to stop the current transaction. The 82C824 will back-off the current cycle and retry it later.
CBLOCK#	106	42	I/O	**Bus Lock:** The 82C824 uses this signal to indicate an atomic operation that may require multiple transactions to complete.
CDEVSEL#	110	47	I/O	**Device Select:** This signal is an input from the slot interface device claiming the cycle.
CPERR#	105	41	I	**Parity Error:** All slot interface devices use this signal to report data parity errors, during any PCI transaction except a Special Cycle.
CSERR#	125	63	I	**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.
CREQ#	128	65	I	**Bus Master Request:** The CardBus card or docking station uses this signal to gain control of the PCI bus. The docking station also signals its need to run an IRQ driveback cycle by double toggling this pin.
CGNT#	107	44	O	**Bus Grant:** The 82C824 grants the bus to the requester, or acknowledges a driveback request, through this pin.

3.2.4 Differences between CardBus and Docking Station Interface

Because the signal set of a CardBus card and a docking station are mostly identical, only the pin differences are noted below.

82C824 Slot Interface Pins - Dynamically Reconfigured for Docking

Docking Name	On CardBus Pin	Slot A/C Pin No.	Slot B/D Pin No.	Signal Description
INTA#	CINT#	109	46	No changes in 82C824 logic occur. The IRQ mapping is selected at PCICFG 48h[4:0], as for CardBus.
INTB#	AUDIO	132	69	The IRQ mapping is selected at PCICFG 49h[4:0]. Setting PCICFG 51h[3] = 1 disables the INTB# function on the AUDIO pin and enables the normal digital audio function.

82C824 Slot Interface Pins - Dynamically Reconfigured for Docking (cont.)

Docking Name	On CardBus Pin	Slot A/C Pin No.	Slot B/D Pin No.	Signal Description
INTC#	CSTSCHG	143	16	Note that CSTSCHG is active high, so it changes polarity to become INTC#. The IRQ mapping is selected at PCICFG 4Ah[4:0].
CREQ0#	CREQ#	128	65	PCI Bus Request/Grant for PCI slot 0. No difference from the CardBus CREQ#/CGNT# signals except that CREQ0# must arbitrate for the bus with CREQ1# and CREQ2#.
CGNT0#	CGNT#	107	44	Rotating Priority is used; except that a "double toggle" gets top priority (for IRQ driveback).
CREQ1#	Unused (PCMCIA D2)	139	77	PCI Bus Request/Grant for PCI slot 1. See above.
CGNT1#	Unused (PCMCIA A18)	104	40	
CREQ2#	Unused (PCMCIA D14)	89	25	PCI Bus Request/Grant for PCI slot 2. See above.
CGNT2#	CCLKRUN#	141	79	The CCLKRUN# signal is not generally needed from a docking station since the unit does not run off batteries.

3.2.5 Slot Interface - PCMCIA Configuration

The signals listed next are standard PCMCIA signals. The signal names indicated are valid only when the slot has been configured for PCMCIA use; the pins change function when a CardBus card is inserted or the system is configured for docking station attachment.

82C824 Pins - Dynamically Reconfigured for PCMCIA

Signal Name	Slot A/C Pin No.	Slot B/D Pin No.	Pin Type	Signal Description
SPKR	132	69	I	**Speaker Input** (SPKR - R2 I/O card)
DREQ# alt. 2			I	**DREQ# alternative 2**
BVD2			I	**Battery Low Voltage Detect pin 2** (BVD2 - R1 or R2 memory card)
STSCHG#	143	16	I	**Status Change Interrupt**, active low (STSCHG# - R2 I/O card)
BVD1			I	**Battery Low Voltage Detect pin 1** (BVD1 - R1 or R2 memory card)
IREQ	109	46	I	**Interrupt Request** (IREQ - R2 I/O card)
RDY/BSY#			I	**Ready/Busy** (RDY/BSY# - R1 or R2 memory card)

82C824 Pins - Dynamically Reconfigured for PCMCIA (cont.)

Signal Name	Slot A/C Pin No.	Slot B/D Pin No.	Pin Type	Signal Description
A[25:0]	120, 117, 115, 112, 110, 108, 106, 104, 102, 111, 113, 105, 103, 116, 95, 92, 99, 101, 118, 121, 122, 124, 126, 129, 131, 133	56, 54, 51, 49, 47, 45, 42, 40, 37, 48, 50, 14, 39, 52, 32, 29, 34, 36, 55, 57, 38, 62, 64, 66, 68, 71	I	**Address Bus Lines 25 through 0**
D[15:0]	91, 89, 87, 85, 82, 140, 138, 136, 88, 86, 83, 81, 80, 139, 137, 134	28, 25, 23, 21, 19, 78, 76, 74, 24, 22, 20, 18, 17, 77, 75, 73	I/O	**Data Bus Lines 15 through 0**
WAIT#	125	63	I	**Wait**
IOCHRDY			I	**I/O Channel Ready**
IOIS16#	141	79	I	**16-Bit I/O Indication** (I/O card)
DREQ# alt. 3			I	**DREQ# alternative 3** (DMA I/O card)
WP			I	**Write Protect** (memory only card)
IOR#	98	33	O	**I/O Read**
IOW#	100	35	O	**I/O Write**
CE2-1#	93, 90	30, 27	O	**Upper/Lower Byte Enable**
WE#	107	44	O	**Memory Write**
TC			O	**Terminal Count** (along with IOW#)
OE#	94	31	O	**Memory Read**
TC			O	**Terminal Count** (along with IOR#)
REG#	130	67	O	**Attribute Register Space Select**
DACK			O	**DMA acknowledge**
INPACK	128	65	I	**Input Acknowledge**
DREQ# alt. 1			I	**DREQ# alternative 1**

3.3 Strap-Selected Interface Options

The 82C824 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. During actual use the resistors consume power only while programming voltage is selected to the cards, at which time the additional current draw would be 5.0V/10k ohm = 0.5mA.

The strapping possibilities are listed in Table 3-4.

Table 3-4 Strap Options for 82C824 Configurations

Strap Selection	Feature	No Strap	Pulled down by 10k ohm Resistor at Reset
VPP12_A	Legacy PCMCIA Mode	Legacy mode disabled	Legacy mode enabled: • Chip responds to I/O Port 3E0-1h access • PCICFG 04h[1:0] = 11 to enable PCI memory and I/O access • PCICFG 3Eh[7] = 1 to enable PCMCIA IREQ routing
VPP12_B	Vendor ID	EXTCLK can be used as secondary clock input	EXTCLK is reassigned as DRVVENID# to drive ID bits from an external buffer onto the PCI bus.
VPP3/5_A	IRQ Driveback Support	Host supports IRQ driveback	Host does not support IRQ driveback. CLKRUN# pin reassigned as IRQLAT pin to control the latch connecting AD bus to IRQ bus. IRQ driveback address defaults to 33333330h in this case.
VPP3/5_B	Zoomed Video Port	VPP3/5A work normally	VPP3/5A becomes ZVPENA# and VPP3/5B becomes ZVPENB#, the enable controls for the external buffers that connect between the ZVP audio/video device and the PC card A[25:4], SPKR#, INPACK# and IOIS16# signals.

3.4 Internal Pull-up Resistors

The 82C824 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 and Table 3-6.

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

Table 3-6 refers specifically to a card detected and powered-up as a 16-bit PC Card (PCMCIA card)

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.

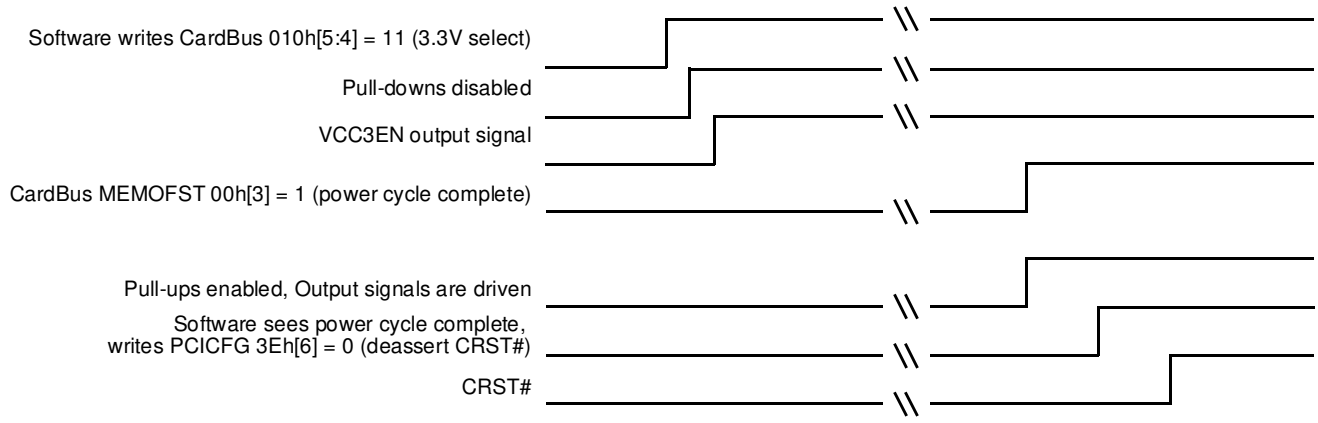
Table 3-5 Internal Keeper Resistor Scheme.

Signal Group	82C824 Action with No Attachment (Socket Empty)	82C824 Action after detecting 16-bit PC Card	82C824 Action after detecting CardBus PC Card	824 Action after detecting Docking Station	Resistor Control PCICFG
Card Detect CCD1# CCD2#	Pull up to core VCC to detect card insertion/removal	Pull up to core VCC	Pull up to core VCC	Pull up to core VCC	52h[4]
Voltage Sense CVS1 CVS2	Pull to card VCC (low w/no card inserted)	Pull to card VCC	Pull to card VCC	Pull to card VCC	52h[4]
Address/Data CAD[31:0] CC/BE[3:0]# CPAR	Pull down	Disable pull-down (card has internal pull-down)	Pull down until card is powered up	Pull down until card is powered up	52h[7]
Reset CRST#+RESET	Driven low	Driven	Driven	Driven	3Eh[6]
Frame CFRAME#+A23	Pull to card VCC (low w/no card inserted)	Disable pull-down (card has internal pull-down)	Disable pull-down (card has internal pull-up)	Pull to card VCC	52h[6]
PCI Control/Status CIRDY#+A15 CTRDY#+A22 CDEVSEL#+A21 CSTOP#+A20 CPERR#+A14 CBLOCK#+A19	Pull to card VCC (low w/no card inserted)	Disable pull-down (card has internal pull-down)	Pull to Card VCC	Pull to card VCC	52h[6]
Clock CCLK/A16	Pull down	Disable pull-down (card has internal pull-down)	Disable pull-down (clock input is always driven)	Disable pull-down (clock input is always driven)	No control
Miscellaneous Group A D02+CREQ1# D14+CREQ2# A18+CGNT1#	Pull down	Disable pull-down (card has internal pull-down)	Pull up (no connections to CardBus)	Pull to card VCC	52h[5]
Miscellaneous Group B CREQ#+INPACK#+CREQ0# CGNT#+WE#+CGNT0# CSTSCHG+STSCHG#+INTC# CAUDIO+SPKR#+INTB# CCLKRUN#+IOIS16#/CGNT2# **Open Drain** CSERR#+WAIT# CINT#+IREQ/INTA#	Pull to card VCC (low w/no card inserted)	Pull to card VCC	Pull to card VCC	Pull to card VCC	52h[5]

Table 3-6 Internal Resistor Scheme - PCMCIA Card Detected

Signal	Pull-up Scheme	Controlled by
CD1#, CD2#	Card Detect lines are pulled up to core VCC most of the time in order to detect card removal.	PCICFG 51h[4] (enabled by default)
BVD1, BVD2	Battery Voltage Detect line pull-ups are enabled only after a PCMCIA card has been installed and detected, and only if the card interface is programmed to memory-only.	82C852 register group bit 3Fh[4]
RDY/BSY#	Ready/Busy line pull-up is enabled only after a PCMCIA card has been installed and detected, and only if the card interface is programmed to memory-only.	82C852 register group bit 3Fh[4]
INPACK#	Input acknowledge line pull-up is enabled only after a PCMCIA card has been installed and detected, and only if the card interface is programmed to I/O.	82C852 register group bit 3Fh[3]
WAIT#	Wait line pull-up is enabled only after a PCMCIA card has been installed and detected	82C852 register group bit 3Fh[3]

Figure 3-2 Power-Up Timing Example



4.0 Functional Description

4.1 OPTi CardBus/Docking Controller

The OPTi CardBus Controller/Docking Station solution comprises two devices. The minimum configuration requires one chip, the 82C824 part.

- The 208-pin 82C824 CardBus Controller handles the signal multiplexing for two CardBus/R2 slots. One 82C824 device is required for support of two CardBus/R2 slots; an optional second device can be added to support two more CardBus/R2 slots. A docking station can optionally be supported on Slot A.
- The 160-pin 82C825 PCI/ISA Bridge converts CardBus PCI signals back into ISA signals. No 82C825 device 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 CardBus PCI bus interface. The 82C825 is discussed in a separate document.

The multiple interface arrangement offers the maximum in system design flexibility, allowing two- or four-slot configurations that support any combination of CardBus cards or PCMCIA cards.

4.2 Chipset Compatibility

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

4.3 Interface Overview

The OPTi 82C824 CardBus/Docking Controller uses three independent external interfaces. The terms *host interface* and *slot interface* are used throughout this document to describe these interfaces.

- The **host interface** provides industry standard PCI signals to the host system.
- Two **slot interfaces** are provided. Each slot interface can be independently configured as a PCMCIA R2-type interface (running PCMCIA cycles), or a CardBus interface (running PCI-type cycles). Optionally, the Slot A interface can be programmed to act as a docking station interface.

The interface signal groups used to integrate the OPTi CardBus/Docking Controller into the standard system are described in the following sections. Figure 4-1 illustrates the interaction of the components of the 82C824.

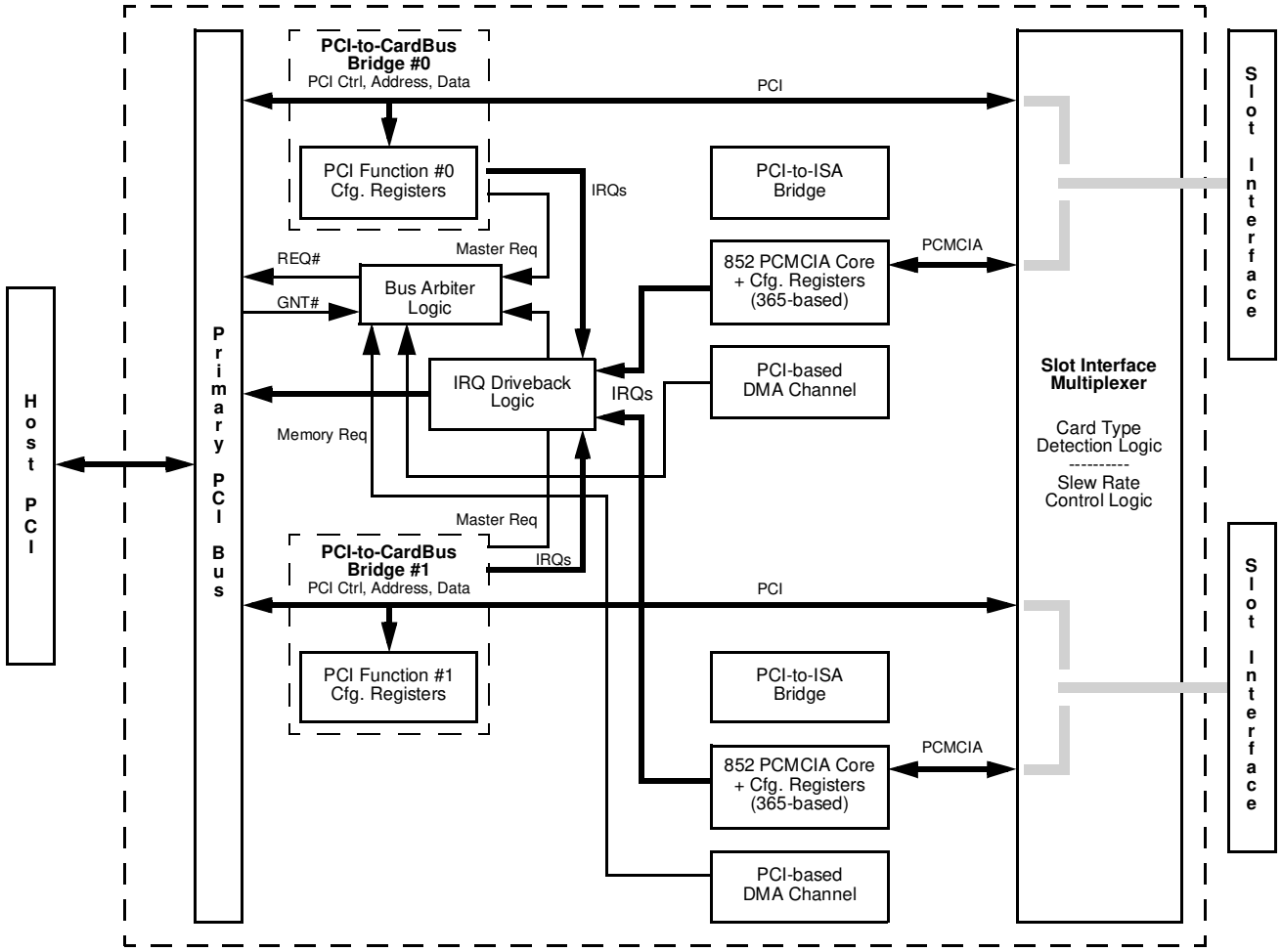
The logic implements several functional blocks that interact as indicated. The functional blocks shown in the diagram are briefly described next.

- The 82C824 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 one of the two slot interfaces. If the slot interface device is a PCMCIA card, it can act only as a slave and cannot run cycles on the primary PCI bus. (DMA cycles are not run by the card, only requested by it.)
- Part of the 82C824 logic implements two **PCI-to-CardBus bridges** controlled by independent sets of **PCI Configuration Registers.** These configuration registers are accessed from the primary PCI bus. Any bus master, including a master on the slot interface, can program these registers. The PCI Configuration Registers consist of standard CardBus registers at indexes 00h-47h and OPTi 82C824 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 first register set is accessed as PCI Function 0 of the 82C824 device, while the second register set is accessed as PCI Function 1 of the 82C824 device.
- Each of the PCI-to-PCI bridges serves to connect the primary PCI bus to an independent secondary PCI bus. It is this secondary bus that interfaces externally to either a CardBus card or to a docking station. From a system point of view, the secondary PCI bus is always present. If a PCMCIA card is installed, the secondary PCI bus signal interface will not be available externally, although the host can still access the configuration registers for that function. If a docking station is attached, the system software may see multiple devices on the bus. If a CardBus card is installed, only Device 0 responds.
- The **slot interface multiplexer** selects whether the slot interface provides an external interface to the internal secondary PCI bus, or to a PCMCIA single slot interface. This selection is controlled automatically through hardware according to the state of the CD1# and CD2# lines from each slot interface. This multiplexer circuit also sets the correct interface levels and signal slew rates for the selected option; these values are programmable.
- From a hardware and programming aspect, the PCMCIA controller sits on an internal ISA-like bus. This bus is the native 16-bit interface of the OPTi 82C852 PCMCIA Controller.
- The **PCMCIA core and configuration registers** are a superset of 82365SL-standard PCMCIA registers, accessed at 64 register indexes through an index/data port arrangement. The index/data port address is defined by the Bridge Base I/O Address defined at setup time in the PCI-to-CardBus Bridge Configuration Registers.

- The **PCI-based single channel DMA controller** derives from the industry standard 82C206 IPC, and is 8237-compatible in its functionality (but not in its address decoding). The DMA controller subsystem comprises two independent, single channel DMA controller logic modules. CardBus and docking station devices cannot utilize this subsystem and must implement their own local DMA controller channels. The DMA controller subsystem services DMA requests by becoming master of the primary PCI bus, through the bus arbiter.
- The **bus arbiter logic** takes master requests to request bus ownership for the purposes of: 1) Driving back IRQs; 2) Giving PCI master control to one of the secondary PCI buses; 3) Making a DMA memory access. Driving back IRQ status always has highest priority.
- Devices connected to the two slot interfaces can transmit interrupts to the host system through the **IRQ driveback logic**. Standard R2 PCMCIA cards and CardBus cards can generate a system interrupt and a status change interrupt, both of which must be mapped to system IRQ lines through program registers. Docking stations can generate INTA#, INTB#, and INTC# which the 82C824 logic remaps if desired. If the host system chipset does not provide the proper logic for recognition of this driveback cycle, IRQ information can be latched externally to generate discrete signals.

The logic subsystems of the 82C824 are described in detail in the following sections.

Figure 4-1 82C824 Organization



4.4 Device Type Detection Logic

The 82C824 logic includes card type detection logic and a power control state machine to determine what type of device has been attached to the slot interface. The outcome of the determination made by the power control state machine determines how the interface will act: as a 16-bit PC card interface, as a CardBus interface, or as a docking station 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.

4.5 Primary PCI Bus

The host interfaces to the 82C824 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 82C824 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]. However, even if CLKRUN# is enabled, attaching a docking station will cause CLKRUN# to always request a running primary clock because docking station CLKRUN# support is not available.

Table 4-1 Device Detection

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

Table 4-2 CLKRUN# Control Bits

7	6	5	4	3	2	1	0
PCICFG 50h			**PCI Host Feature Control Register (common to both slots)**				**Default = 00h**
ZVP mode strap selected: 0 = No 1 = Yes	Legacy mode strap-selected (RO): 0 = No 1 = Yes	Vendor ID feature strap-selected: 0 = No 1 = Yes	IRQLAT function on CLKRUN# strap-selected: 0 = No 1 = Yes	Card detect debounce: 0 = 0.25 sec 1 = 1.0 sec	CLKRUN (in host interface): 0 = Enabled per PCI 1 = Disabled, CLKRUN# tristated	SPKROUT: 0 = Tristated 1 = Driven	Reserved

4.6 PCI-to-CardBus Bridge

The PCI-to-CardBus bridge circuit of the 82C824 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 or 1 of the 82C824 chip.

4.6.1.1 Translation Between Type 0 and Type 1 Configuration Cycles

The 82C824 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 82C824 logic is different from the standard PCI-to-PCI bridge in this regard. The 82C824 allows the secondary to act as a primary. PCICFG 52h[0] is used to enable this feature. Refer to Table 4-3.

With this feature selected, master devices on the docking station interface can program the PCI configuration registers of the 82C824 (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 82C824 logic will pass this to the primary as a Type 0 configuration cycle. Since the 82C824 PCI configuration registers sit on the primary, they are also accessible this way. Thus, on the primary the 82C824 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 82C824.

4.6.2 Cycle from Host to Slot Interface

For a cycle from the host to a slot interface with a CardBus card inserted or a docking station attached, the PCI bridge resynchronizes the cycle and passes it to the external PCI-type device. Slot interface PCI-type devices can run either synchronously at 33MHz, or asynchronously (typically at 16MHz, 20MHz, or 25MHz). The bridge claims the cycle if it falls into one of the ranges programmed in the Window Registers of the PCI Configuration Register set.

A slot interface device can become a bus master by asserting REQ#. The 82C824 arbiter will assert GNT# if the bus is free. Note that the arbiter will not make the bus request on the primary side until a cycle has been started by the secondary bus master.

4.6.3 Master Cycle from Slot Interface

For a master cycle from the slot interface, the 82C824 logic presents the cycle on the host PCI bus as master.

If the cycle is directed to a device on the other slot interface, the 82C824 logic claims the cycle immediately, as a slave, since the address ranges are already programmed into the Base Address Registers for that slot.

If the cycle is not claimed by the other slot and no host device claims it, the 82C824 generates a master abort.

Table 4-3 Translation Configuration Bit

7	6	5	4	3	2	1	0
PCICFG 52h			**Slot Feature Control Register 2**				**Default = 0Fh**
Secondary PCICLK Skew: This value selects the approximate delay, in nanoseconds, that the internal secondary PCICLK must be skewed in order to compensate for external buffer delays. 0000 = No delay 0001 = 1ns 0010 = 2ns ... 1101 = 13ns 1110 = 14ns 1111=15ns				CCLKRUN# pin if dock attached: 0 = CGNT2# 1 = CCLKRUN#	Reserved		Type 1 to Type 0 conversion blocked from secondary to primary: 0 = No 1 = Yes (Default)

4.6.4 Inability to Complete a Posted Write

The 82C824 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 82C824 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 82C824 provides only one mechanism to return the error: the SERR# pin. The master must then decide how to handle the SERR# generation, either by generation of an NMI or some other means.

The 82C824 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 82C824 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 82C824-specific register PCICFG 5Fh to determine whether the retry limit was exceeded.

Target Abort or No Response - The logic generates SERR# or 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 82C824 logic terminates the cycle to the initiator normally. If PCICFG 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 82C824 logic terminates the cycle to the initiator normally and returns FFFFFFFFh as the data read. If PCICFG 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**
Slow decode for ISA windows: 0 = Disable 1 = Enable	Prefetch on upstream transactions: 0 = Disable 1 = Enable	Posted writes on upstream transactions: 0 = Disable 1 = Enable	Core voltage: 0 = 3.3V 1 = 5.0V	Retry count readback control: 0 = Write posting retries on secondary 1 = Retries on primary	Retry Limit: These bits relate to the number of times that the 82C824, as a slave, will retry accesses on the primary. If this limit is exceeded, the 82C824 generates SERR# to the host. 000=2^8 100=2^{16} 001=2^{10} 101=2^{20} 010=2^{12} 110=2^{24} 011=2^{14} 111= Infinite retries (Default)		
PCICFG 5Fh			**Retry Count Readback Register**				**Default = 00h**
This register returns the number of retry attempts either as master (on the secondary) or as slave (on the primary).							
PCICFG 3Eh			**Bridge Control Register - Byte 0**				**Default = 40h**
Route PCMCIA IREQ to PCICFG 4Ch IRQ: 0 = Yes 1 = No	Force CRST# cycling on slot interface: 0 = CRST# high 1 = Assert CRST# (Default)	Response to master abort on slot interface: 0 = Ignore 1 = Signal with target abort or SERR#	Reserved: Write as read.	Pass VGA addresses A0000-BFFFFh, 3B0-3BBh, 3C0-3DFh: 0 = No 1 = Yes	Reserved	Forwarding of SERR# from slot interface to primary PCI bus: 0 = Disable 1 = Enable	Response to parity errors on slot interface: 0 = Ignore 1 = Enable

4.7 PCI Docking Station Operation

When a docking station is attached to a slot interface, the power control state machine of the 82C824 recognizes the docking station and alters the functions of its slot interface to accommodate the change. The following discussion describes the specific differences between the slot interface in CardBus mode and in docking station mode.

1. 82C824 power control state machine recognizes a docking station through:
   - CCD2# connected to GND
   - 5.0V docking station: CCD1# connected to CVS2, CVS1 connected to GND
   - 3.3V docking station: CCD1# connected to CVS1, CVS2 connected to GND
2. 82C824 arbitration logic reassigns the following signals to accommodate three PCI masters:
   - CREQ# becomes CREQ0# and CGNT# becomes CGNT0#
   - PCMCIA D2 (unassigned for CardBus) becomes CREQ1#
   - PCMCIA A18 (unassigned for CardBus) becomes CGNT1#
   - PCMCIA D14 (unassigned for CardBus) becomes CREQ2#
   - CCLKRUN# becomes CGNT2#
3. 82C824 interrupt logic reassigns the following signals to provide three shareable interrupts:
   - CINT# is used as INTA# (no changes in 82C824 logic occur); IRQ selected at PCICFG 48h[4:0]
   - AUDIO becomes INTB#; IRQ selected at PCICFG 49h[4:0] (override with PCICFG 51h[3])
   - CSTSCHG becomes INTC# (and changes polarity); IRQ selected at PCICFG 4Ah[7:4]
   - Card insertion/removal events generate the IRQ selected at PCICFG 3Dh[3:0].
5. Docking station devices capable of following the IRQ driveback protocol (writing IRQs to specific I/O location on primary bus) have direct access to all IRQ lines.

It is assumed that when a docking station is connected, no CLKRUN# function will be required on the docking station PCI bus. Likewise, AUDIO will be either unused or can become a sideband signal if needed.

Table 4-5 PCI Docking Station Setup Related Associated Register Bits

7	6	5	4	3	2	1	0
PCICFG 48h			**Docking INTA# Interrupt Assignment Register (Slot A Only)**				**Default = 01h**
Reserved			Docking INTA# Interrupt Assignment (PCIRQ0# Default) - Interrupts from the docking INTA# 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: (FireStar only)** 00000 = Disabled 00001 = PCIRQ0# (Default) 00010 = PCIRQ1# 00011 = PCIRQ2# 00100 = PCIRQ3# 00101-01111 = Rsrvd **Edge Mode: (Viper-N+ or FireStar)** 10000 = IRQ0 10001 = IRQ1 10010 = IRQ2 10011 = IRQ3 10100 = IRQ4 10101 = IRQ5 10110 = IRQ6 10111 = IRQ7 11000 = IRQ8 11001 = IRQ9 11010 = IRQ10 11011 = IRQ11 11100 = IRQ12 11101 = IRQ13 11110 = IRQ14 11111 = IRQ15				
PCICFG 49h			**Docking INTB# Interrupt Assignment Register (Slot A Only)**				**Default = 02h**
Reserved			Docking INTB# Interrupt Assignment (PCIRQ1# Default) - Interrupts from the docking INTB# 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: (FireStar only)** 00000 = Disabled 00001 = PCIRQ0# 00010 = PCIRQ1# (Default) 00011 = PCIRQ2# 00100 = PCIRQ3# 00101-01111 = Rsrvd **Edge Mode: (Viper-N+ or FireStar)** 10000 = IRQ0 10001 = IRQ1 10010 = IRQ2 10011 = IRQ3 10100 = IRQ4 10101 = IRQ5 10110 = IRQ6 10111 = IRQ7 11000 = IRQ8 11001 = IRQ9 11010 = IRQ10 11011 = IRQ11 11100 = IRQ12 11101 = IRQ13 11110 = IRQ14 11111 = IRQ15				

Table 4-5 PCI Docking Station Setup Related Associated Register Bits (cont.)

7	6	5	4	3	2	1	0
PCICFG 4Ah			**Docking INTC# Interrupt Assignment Register (Slot A Only)**				**Default = 03h**
Reserved			Docking INTC# Interrupt Assignment (PCIRQ2# Default) - Interrupts from the docking INTC# 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: (FireStar only)** 00000 = Disabled 00001 = PCIRQ0# 00010 = PCIRQ1# 00011 = PCIRQ2#(Default) 00100 = PCIRQ3# 00101-01111 = Rsrvd **Edge Mode: (Viper-N+ or FireStar)** 10000 = IRQ0 10001 = IRQ1 10010 = IRQ2 10011 = IRQ3 10100 = IRQ4 10101 = IRQ5 10110 = IRQ6 10111 = IRQ7 11000 = IRQ8 11001 = IRQ9 11010 = IRQ10 11011 = IRQ11 11100 = IRQ12 11101 = IRQ13 11110 = IRQ14 11111 = IRQ15				
PCICFG 4Ch			**Docking Detect Interrupt Assignment Register**				**Default = 01h**
Host controller type: 0 = FireStar (burst two data phases) 1 = Viper-N+ (send single data phase on IRQ driveback)	Reserved		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. **Level Mode:** 00000 = Disabled 00001 = PCIRQ0# (Default) 00010 = PCIRQ1# 00011 = PCIRQ2# 00100 = PCIRQ3# 00101 = ACPI0 00110 = ACPI1 00111 = ACPI2 01000 = ACPI3 01001-01111 = Rsrvd **Edge Mode:** 10000 = IRQ0 10001 = IRQ1 10010 = IRQ2 10011 = IRQ3 10100 = IRQ4 10101 = IRQ5 10110 = IRQ6 10111 = IRQ7 11000 = IRQ8 11001 = IRQ9 11010 = IRQ10 11011 = IRQ11 11100 = IRQ12 11101 = IRQ13 11110 = IRQ14 11111 = IRQ15				
PCICFG 51h			**Slot Feature Control Register 1**				**Default = 00h**
Slot clock divisor: 00 = 1 (Default) 01 = 2 10 = 3 11 = 4		Slot clock source: 0 = PCICLK 1 = EXTCLK	Mode select: 0 = Automatic 1 = Force async	AUDIO pin if dock attached: 0 = INTB# 1 = AUDIO (Slot A only)	Slot threshold voltage: 0 = 3.3V 1 = 5.0V	Output Drive Select: 00 = 5.0V 16-bit PC cards, 3.3V CardBus cards 01 = 3.3V 16-bit PC cards 10 = 3.3V PCI dock 11 = 5.0V PCI dock	

4.7.1 PCI Clock Buffering

The 82C824 logic provides register settings to compensate for the delay of an external buffer (refer to Table 4-6). This buffer would be required to support multiple PCI devices from the single PCICLK generated by the 82C824. Even if only a single PCI device is used, the trace delays will probably still require some compensation.

A CardBus card will generally not have any need for this compensation. Therefore, this register value is ignored for CardBus cards.

Table 4-6 Register used to Delay PCICLK to CardBus

7	6	5	4	3	2	1	0
PCICFG 52h			**Slot Feature Control Register 2**				**Default = 0Fh**
Secondary PCICLK Skew: This value selects the approximate delay, in nanoseconds, that the internal secondary PCICLK must be skewed in order to compensate for external buffer delays. 0000 = No delay 0001 = 1ns 0010 = 2ns ... 1101 = 13ns 1110 = 14ns 1111=15ns				CCLKRUN# pin if dock attached: 0 = CGNT2# 1 = CCLKRUN# (Slot A only)	Reserved		Type 1 to Type 0 conversion blocked from secondary to primary: 0 = No 1 = Yes (Default)

4.8 PCMCIA Controller

The PCMCIA Controller core logic is similar to that of the OPTi 82C852 single slot PCMCIA controller. The PCI-to-ISA bridge translates each PCI cycle to a 16-bit ISA cycle; the 82C852 logic performs any 8- to 16-bit conversion necessary. The 82C852 controller claims cycles based on the address and command type latched in the address phase from PCI. cycles to the controller are claimed as follows.

- If the cycle is destined for a slot interface with a PCMCIA card inserted, the PCMCIA logic converts the cycle from PCI to the ISA-type cycle associated with the PCMCIA card.
- In the case of an access to the PCMCIA configuration register space, the access is not a PCI configuration cycle but instead an I/O cycle to the bridge base address previously programmed in the PCI configuration register space. The PCI-to-ISA bridge passes this access along as a normal I/O access to the PCMCIA core.
- For Yenta compliance, the PCMCIA configuration register space is also accessible in system memory space. Refer to Table 5-3.

The PCMCIA configuration registers consist of a superset of 82365SL-standard PCMCIA registers accessed at 64 register indexes. Any bus master can program the PCMCIA configuration registers.

4.9 IRQ Driveback Logic

A detailed overview of the IRQ driveback cycle is provided in Appendix B. The logic used to implement this mechanism is relatively simple. The trigger events for a driveback cycle are any transition on an IRQ line, or an SMI event as enabled by the 82C824 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 arbiter asserts REQ# on behalf of the IRQ driveback logic and toggles REQ# according to the driveback protocol discussed in Appendix B.

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 B.

4.9.1 Interrupt Sources

Interrupts come from a number of sources.

- Each CardBus/PCMCIA card can generate two IRQs: one on its CINT#/IREQ# line, and another on its CSTSCHG/STSCHG# line. The IRQ number is programmed into the PCI or PCMCIA configuration registers by Card Services.
- Slot interface devices that use the IRQ driveback scheme can generate any level of IRQ desired at any time. No programming is required to enable this feature, which can operate in parallel with the CINT#/CSTSCHG IRQ scheme if desired.
- The docking station has PCI interrupts INTA#, INTB#, and INTC#. Each of these can be directed to any interrupt.

4.9.2 Reassignment of Interrupt Indicator Bits by Host

Since not all systems need all the separate IRQ indications available, the host can assign many bits for application specific functions. The choice is made solely in the host and the interpretation of the bits is design dependent. The 82C824 logic provides only a means of indicating activity on various bits.

For example, the system design might require a Ring Indicator indication instead of IRQ15. As long as the host provides an option of differentiating the IRQ15 bits as RI bits, the system software can reassign the function of this bit position. Card and Socket Services would need to be aware that they can use IRQ15 to indicate RI.

4.9.3 Interrupt Status Return Latency

An IRQ driveback cycle has predictable latency. Since the host is required to service a driveback cycle with the highest priority, interrupt latency depends solely on the time required for the current bus master to give up the bus after the host has removed its GNT# signal. Therefore, masters on the system **must** honor a latency timer time-out after their GNT# signal has been removed. With this requirement, maximum interrupt service latency can be predicted very accurately.

A more important aspect of driveback latency is the ability of the host to inhibit activity that would be affected by IRQ status change delays. Figure 4-2 illustrates the problem. For each stage of IRQ status generation or resynchronization there is a penalty. In the case shown, the nominal latency is less than 400ns. However, even this low latency could result in false interrupt generation, as explained next.

4.9.3.1 End-of-Interrupt (EOI)

The primary concern for driveback delays is End of Interrupt (EOI) recognition at the 8259-compatible interrupt controller on the host system. At the end of interrupt service, software writes to the interrupting device (possibly across the 82C824 bridge) to command it to deassert its interrupt line. The software then generates an EOI command to the local 8259 interrupt controller, enabling it to generate another interrupt. However, there is a delay involved in passing the changed IRQ status from the interrupting device across the PCI bridge and generating the IRQ driveback cycle to the 8259 interrupt controller. Therefore, the 8259 interrupt controller could conceivably receive the EOI command while the incoming interrupt line still appears active. If the channel is programmed for level mode, the result would be a false interrupt.

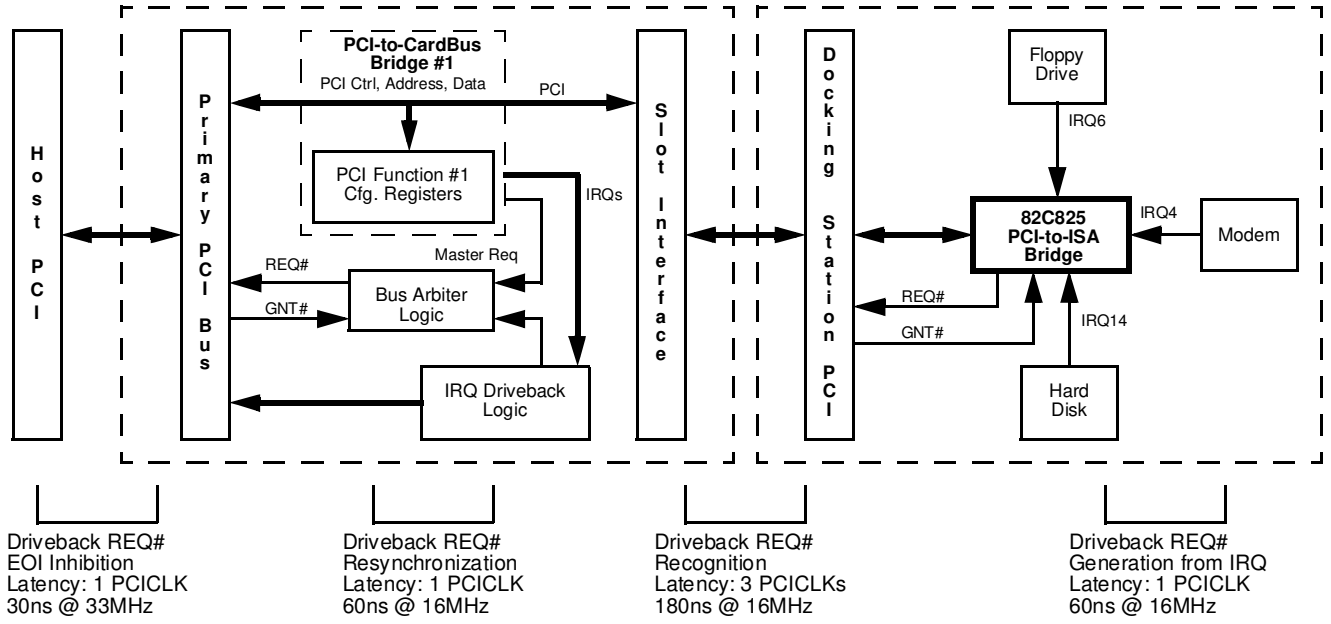
4.9.3.2 EOI Handling

The host handles this situation as follows if it has direct control of the interrupt controller, which is the case with OPTi PCI hosts. Whenever the host sees its REQ# input active, it inhibits its EOIs for a delay period.

A device across the PCI bridge, such as the docking station device on a secondary PCI bus, also uses the same driveback mechanism as the 82C824 does on the host side to generate an IRQ.

Therefore, the host device must have a programmable delay that it generates any time an EOI command is written to its 8259 interrupt controller. During this delay, IRQ writeback request activity signalled on the incoming REQ# lines must be serviced immediately, or in any case before the EOI is allowed to pass.

Figure 4-2 Worst IRQ Driveback Latency Example



4.10 DMA Controller Subsystem

The 82C824 logic incorporates two 8237-type DMA controller channels. The only sources of DMA requests for this subsystem are the two PCMCIA controller cores, each of which can be assigned to any one of the 8-bit DMA Channels 0-3 or 16-bit DMA Channels 5-7. Channel 4 cannot be selected. The desired DMA channel is selected through the PCMCIA register set, and is not affected by the DMA Channel Selector Register in the PCI configuration registers.

Appendix A describes the distributed DMA protocol used to define the operation of the DMA controller subsystem. Refer to Appendix A for complete details. Figure 4-3 illustrates the path taken by DMA requests from the PCMCIA DMA card. The DMA control/status bypass the intermediate ISA bus and go directly to the DMA controller subsystem.

The DMA controller subsystem receives the DMA requests from the 82C852 PCMCIA controllers. It services the requests by becoming master of the PCI bus, through the bus arbiter. Once it owns the host PCI bus, the DMA controller subsystem generates system memory requests on the PCI bus, and performs I/O transfers internally to the PCMCIA card. Even if a DMA device in one slot is transferring data to or from memory on a PCMCIA card in the other slot, the memory requests still go to the primary PCI bus to get to the other PCMCIA controller and card.

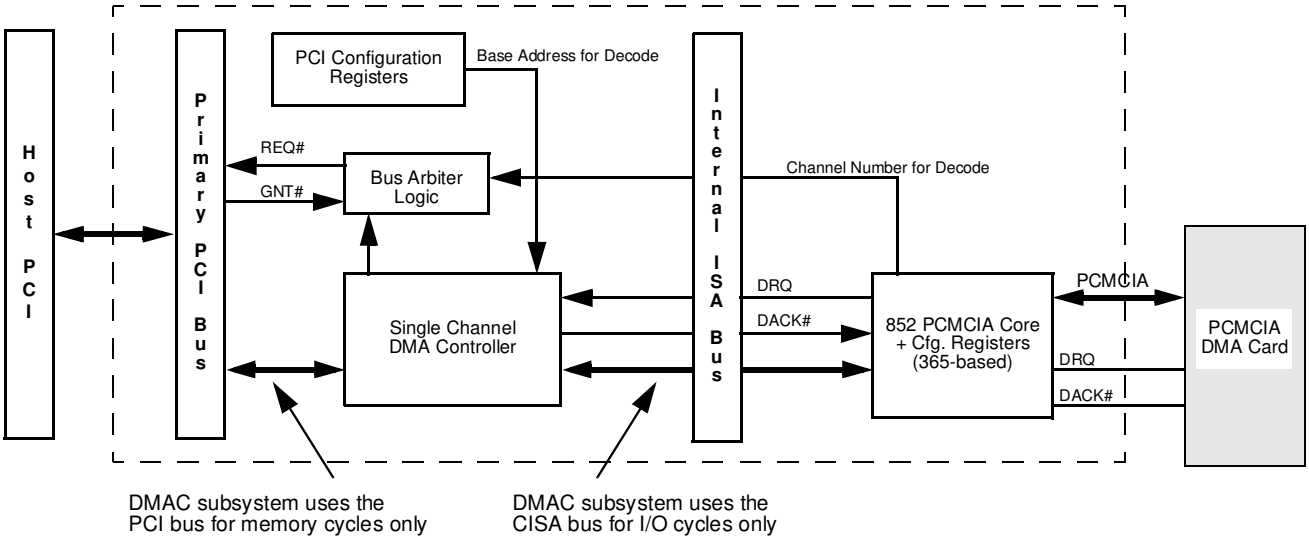
4.10.1 DMA Controller Programming Registers

The DMA controller subsystem implements two direct memory access controller channels and their associated memory mappers that are register compatible with AT-type systems. However, the distributed DMA protocol requires these registers to be remapped so that their location within the system I/O space is not necessarily AT-compatible.

The registers of this logic module in the AT-compatible host system are listed in Appendix A. Within the host chipset, these registers are accessed directly through the I/O subsystem (no index/data method is used). In the AT architecture, Channels 0-3 are in DMAC1 and are 8-bit; Channels 4-7 are in DMAC2 and are 16-bit. Table A-1 (in Appendix A) lists the register contents.

Refer to Section A.3.4 on page 67 for information regarding the 82C824 distributed DMA register mapping.

Figure 4-3 DMA Controller Subsystem Logic



4.10.2 DMA Channel Selector Register

The protocol described in Appendix A requires each DMA-aware bridge or DMAC on the PCI bus to incorporate a DMA Channel Selector Register, which indicates the DMA devices that will be handled by that device. The register is illustrated below. This register does **not** affect the claiming of cycles for the PCMCIA controllers.

The 82C824 architecture provides three DMA I/O command paths: one for its local DMAC subsystem that services the PCMCIA controllers, and two for DMA controllers on the slot interfaces (in a docking station or on a CardBus card). The DMA Channel Selector Register provides the ability to pass on DMA programming information to downstream DMACs on CardBus cards or in docking stations.

Because the PCMCIA controllers each have their own DMA channel selection bits, reading the DMA Channel Selector Register does not fully reflect all DMA channels in use; it simply returns the last value written. For example, if a docking station will use DMA Channels 2 and 7, software would set PCICFG 5Ch = 10000100b to claim Channels 2 and 7. However, if a PCMCIA card is using DMA Channel 3 as programmed through the 82C852 register set, this setting will not be reflected when reading PCICFG 5Ch. Software must also read the PCMCIA controllers to determine whether they are programmed to claim a DMA channel.

Table 4-7 DMA Channel Selection (in PCI-to-CardBus Bridge Cfg. Register Set)

7	6	5	4	3	2	1	0
PCICFG 5Ch			**DMA Channel Selector Register**				**Default = 00h**
Channel 7 (DMAC2): 0 = Not claimed 1 = On slot interface	Channel 6 (DMAC2): 0 = Not claimed 1 = On slot interface	Channel 5 (DMAC2): 0 = Not claimed 1 = On slot interface	DMAC responsibility (RO): 0 = Secondary (always) 1 = Master	Channel 3 (DMAC1): 0 = Not claimed 1 = On slot interface	Channel 2 (DMAC1): 0 = Not claimed 1 = On slot interface	Channel 1 (DMAC1): 0 = Not claimed 1 = On slot interface	Channel 0 (DMAC1): 0 = Not claimed 1 = On slot interface

4.11 Zoomed Video Port Support

The 82C824 chip incorporates support for Zoomed Video Port on 16-bit PC cards. There are several aspects to this functionality.

- The feature is enabled as a strap option, using the VPP3/5B pin.
- When the strap option is selected at reset, the VPP3/5A and VPP3/5B pins become ZVPENA# and ZVPENB#, enable controls for external buffers that connect between the ZVP audio/video device and the PC card A[25:4], SPKR#, INPACK#, and IOIS16# signals (25 pins).
- "Q" switches (Quality Semiconductor QST3584) are ideal for use as buffers, since they require no direction control signal and provide buffers as groups of five instead of the usual eight. Therefore, five "Q" switches would be needed to fully buffer two slots.
- When the ZVP feature is enabled, through bit 1Eh[5] of the PCMCIA (852) register set, the 82C824 logic tristates address outputs A[25:4] and disables (ignores) its SPKR#, INPACK#, and IOIS16# inputs. It also drives the ZVPEN# signal low for the requested slot.
- ZVP can be requested on either card slot, but not on both simultaneously. If both are selected, only the ZVPENA# signal will go active.
- Since the VPP3/5 pin is no longer provided, the power controller VPP3/5 input must be generated as an inverted version of VPP12. Therefore the 82C824 logic **must always disable** VCC (VCC5EN = 0, VCC3EN = 0) whenever the register logic tries to select both VPP12 and VPP3/5 equal to 1.

Refer to the ZVP proposal available from PCMCIA for further information on this feature.

Table 4-8 ZVP Feature Enable Register Bit

7	6	5	4	3	2	1	0
Index 1Eh, 5Eh, 9Eh, DEh / MEMOFST 81Eh		**Global Control Register**					**Default = 00h**
Reserved: Write as read.		Zoomed video port (reassigns A[25:4], IOIS16#, INPACK#, SPKR#): 0 = Disable 1 = Enable	Reserved: Write as read.		Reset change status: 0 = On status change reg read 1 = On write to bit	Reserved: Write as read.	

5.0 Register Descriptions

The 82C824 CardBus/Docking Controller provides several groups 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, 82C824-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.
- **PCI-to-CardBus Bridge 1 Register Group**
  As above, but through Function 1 of the chip.
- **PCMCIA Controller 0 Register Group**
  Accessed either through the traditional index/data mechanism at I/O Addresses 3E0-1h (or 3E2-3h), or at Offset 800h from the CardBus Control and Status Register Group Base Address in memory. Consists of 64 registers starting at Index 00h (Slot A) or 80h (Slot C). Within these 64 registers are four groups: General Purpose, I/O Mapping Window, Memory Mapping Window, and Special.
- **PCMCIA Controller 1 Register Group**
  As above, but starting at Indexes 40h (Slot B) or C0h (Slot D).

These register groups are defined in the following subsections.

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

5.1 Register State on Device Removal

PCI configuration registers retain their programmed value when the card or docking station is disconnected from the interface (CCD1# and CCD2# both high).

5.2 PCI Bridge 0 and 1 Register Groups

The 82C824 has two sets of PCI-to-PCI bridge configuration registers, one for each PCI-to-PCI bridge function of the chip. These registers are accessed as:

1) Function 0 (address bits AD[10:8] = 000 during the configuration cycle) for Bridge 0, and as
2) Function 1 (AD[10:8] = 001) for Bridge 1.

These registers are always accessible, even if the slot interface is empty or contains a PCMCIA card. They do not necessarily retain their values after a card is inserted or removed from the slot (or a docking station is attached/detached).

5.2.1 Base Register Group

The registers in Table 5-1 represent the standard group required for PCI peripheral device identification and configuration for a PCI-to-CardBus bridge.

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 = 24h**
PCICFG 03h			**Device ID (RO) - Byte 1**				**Default = C8h**
PCICFG 04h			**PCI Command Register - Byte 0**				**Default = 04h**
Address/data stepping: 0 = Disable (always)	PERR# generation: 0 = Disable 1 = Enable	VGA palette snoop: 0 = Disable 1 = Enable	Mem write and Invalidate (RO): 0 = Disable (always)	Special Cycle (RO): 0 = Disable (always)	Bus master by slot interfaces: 1 = Enable (always)	Respond to PCI mem accesses: 0 = No 1 = Yes	Respond to PCI I/O accesses: 0 = No 1 = Yes
PCICFG 05h			**PCI Command Register - Byte 1**				**Default = 00h**
Reserved: Write bits as read.						Fast back-to-back (RO): 0 = Disable (always)	SERR# generation: 0 = Disable 1 = Enable

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

7	6	5	4	3	2	1	0
PCICFG 06h			**PCI Status Register - Byte 0**				**Default = 00h**
Fast back-to-back capability (RO): 0 = No (always)	Reserved (RO)						
PCICFG 07h			**PCI Status Register - Byte 1**				**Default = 02h**
Parity error: 0 = No 1 = Yes Write 1 to clear	System error: 0 = No 1 = Yes Write 1 to clear	Received master abort: 0 = No 1 = Yes Write 1 to clear	Received target abort: 0 = No 1 = Yes Write 1 to clear	Signalled target abort: 0 = No 1 = Yes Write 1 to clear	DEVSEL# timing (RO): 00 = Fast 01 = Medium (always) 10 = Slow 11 = Reserved		PERR# active as master: 0 = No 1 = Yes Write 1 to clear
PCICFG 08h			**Revision Register (RO)**				**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 = 82h**
Multi-function device (RO): 1 = Yes (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: - The 32-bit Cardbus Base Address Register selects the starting address in memory space of the CardBus socket status and control registers. - Actual register addresses are calculated by adding the MEMOFST of the register to this base address. - 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-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 slot interface PCI bus (RO): 0 = No (always)	Reserved (RO)						
PCICFG 17h			**PCI Secondary Bus Status Register - Byte 1**				**Default = 02h**
Parity error on slot interface PCI bus: 0 = No 1 = Yes Write 1 to clear	Received system error on slot interface PCI bus: 0 = No 1 = Yes Write 1 to clear	Received master abort on slot interface PCI bus (RO): 0 = No 1 = Yes	Received target abort on slot interface PCI bus (RO): 0 = No 1 = Yes	Signalled target abort on slot interface PCI bus: 0 = No 1 = Yes Write 1 to clear	DEVSEL# timing on slot interface PCI bus (RO): 00 = Fast 01 = Medium (always) 10 = Slow 11 = Reserved		PERR# active as master on slot interface PCI bus (RO): 0 = No 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 82C824 is connected. - Defaults to 0. - The logic uses this value to determine whether type 1 configuration transactions on the slot interface should be converted to Special Cycle transactions on the host interface.							
PCICFG 19h			**CardBus Number Register**				**Default = 00h**
- Indicates the number of the PCI bus to which the slot interface of the 82C824 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 slot interface.							
PCICFG 1Ah			**Subordinate Bus Number Register**				**Default = 00h**
- Indicates the number of the highest-numbered PCI bus on the slot interface side. - Defaults to 0. - The 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 slot interface.							
PCICFG 1Bh			**CardBus Latency Timer Register**				**Default = 00h**
Indicates the time-out value for the slot 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: - The 32-bit Memory Window 0 Limit Address Register selects the end address of Memory Window 0. - Bits [11:0] are read-only and are always 0. - 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: - 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. - 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[1] (Bridge Control Register) and defaults to "enabled." - 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: - The 32-bit Memory Window 1 Limit Address Register selects the end address of Memory Window 1. - Bits [11:0] are read-only and are always 0. - 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: - 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. - The I/O windows are globally enabled by bit 04h[0] (Command Register).						RO: Always returns 0.	Decoding: 0 = 16-bit (AD[31:16] = 0) 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**
PCICFG 30h			**I/O Window 0 Limit Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
I/O Window 0 Limit Address Bits: - The 32-bit I/O Window 0 Limit Address Register selects the end address of I/O Window 0. - The minimum window size is always 4 bytes.						RO: Always returns 0.	
PCICFG 31h			**I/O Window 0 Limit Address Register - Byte 1: Address Bits [15:8]**				**Default = 00h**

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

7	6	5	4	3	2	1	0
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: - 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. - The I/O windows are globally enabled by bit 04h[0] (Command Register).						RO: Always returns 0.	Decoding: 0 = 16-bit (AD[31:16] = 0) 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: - The 32-bit I/O Window 1 Limit Address Register selects the end address of I/O Window 1. - The minimum window size is always 4 bytes.						RO: 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			**CINT# Interrupt Line Register for Status Change**				**Default = 00h**
- This register is readable and writable per the PCI specification. - The logic does not use the value written to this register.							
PCICFG 3Dh			**CSTSCHG Interrupt Pin Register for Status Change (RO)**				**Default = 01h**
- This register reflects the value written to PCICFG 4Ch. - It defaults to 01h, selecting PCIRQ0# for the status change (docking station attach/detach) interrupt. - 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**
Route PCMCIA IREQ to PCICFG 4Ch IRQ: 0 = Yes 1 = No	Force CRST# cycling on slot interface: 0 = CRST# high 1 = Assert CRST# (Default)	Response to master abort on slot interface: 0 = Ignore 1 = Signal with target abort or SERR#	Reserved: Write as read.	Pass VGA addresses A0000-BFFFFh, 3B0-3BBh, 3C0-3DFh: 0 = No 1 = Yes	Reserved	Forwarding of SERR# from slot interface to primary PCI bus: 0 = Disable 1 = Enable	Response to parity errors on slot interface: 0 = Ignore 1 = Enable
PCICFG 3Fh			**Bridge Control Register - Byte 1**				**Default = 03h**
Reserved. Write as read.					Write posting: 0 = Disable 1 = Enable	Memory Window 1 prefetch: 0 = Disable 1 = Enable (Default)	Memory Window 0 prefetch: 0 = Disable 1 = Enable (Default)

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

7	6	5	4	3	2	1	0

PCICFG 40h — **Subsystem Vendor Register - Byte 0: Bits [7:0]** — **Default = 00h**

Subsystem Vendor Bits:

- The chipset normally responds to reads of this read-only register with 00h.
- If the option is strap-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 Bits:

- The chipset normally responds to reads of this read-only register with 00h.
- If the option is strap-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 — **16-Bit PC Card Legacy Mode Address Register - Byte 0: Bits [7:0]** — **Default = 01h**

16-bit PC Card Legacy Mode Base Address Bits:

- Bits [31:1] point to the PCMCIA index and data registers at I/O address 3E0-1h for the first four slots, and 3E2-3h for the second four slots.
- This register is intended to be used only for backward-compatible operation with existing PCMCIA drivers. It is not recommended that this feature be used by new software. New code should use the PC Card Socket Status and Control Registers Base Address space to address the registers directly.

RO: Always returns 1.

PCICFG 45h — **16-Bit PC Card Legacy Mode Address Register - Byte 1: Address Bits [15:8]** — **Default = 00h**

PCICFG 46h — **16-Bit PC Card Legacy Mode Address Register - Byte 2: Address Bits [23:16]** — **Default = 00h**

PCICFG 47h — **16-Bit PC Card Legacy Mode Address Register - Byte 3: [31:24]** — **Default = 00h**

PCICFG 48h — **Docking INTA# Interrupt Assignment Register (Slot A Only)** — **Default = 01h**

Reserved

Docking INTA# Interrupt Assignment (PCIRQ0# Default) - Interrupts from the docking INTA# 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: (FireStar only)

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

Edge Mode: (Viper-N+ or FireStar)

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		

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

7	6	5	4	3	2	1	0
PCICFG 49h			**Docking INTB# Interrupt Assignment Register (Slot A Only)**				**Default = 02h**
Reserved			Docking INTB# Interrupt Assignment (PCIRQ1# Default) - Interrupts from the docking INTB# 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: (FireStar only)** 00000 = Disabled 00001 = PCIRQ0# 00010 = PCIRQ1# (Default) 00011 = PCIRQ2# 00100 = PCIRQ3# 00101-01111 = Rsrvd **Edge Mode: (Viper-N+ or FireStar)** 10000 = IRQ0 10001 = IRQ1 10010 = IRQ2 10011 = IRQ3 10100 = IRQ4 10101 = IRQ5 10110 = IRQ6 10111 = IRQ7 11000 = IRQ8 11001 = IRQ9 11010 = IRQ10 11011 = IRQ11 11100 = IRQ12 11101 = IRQ13 11110 = IRQ14 11111 = IRQ15				
PCICFG 4Ah			**Docking INTC# Interrupt Assignment Register (Slot A Only)**				**Default = 03h**
Reserved			Docking INTC# Interrupt Assignment (PCIRQ2# Default) - Interrupts from the docking INTC# 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: (FireStar only)** 00000 = Disabled 00001 = PCIRQ0# 00010 = PCIRQ1# 00011 = PCIRQ2#(Default) 00100 = PCIRQ3# 00101-01111 = Rsrvd **Edge Mode: (Viper-N+ or FireStar)** 10000 = IRQ0 10001 = IRQ1 10010 = IRQ2 10011 = IRQ3 10100 = IRQ4 10101 = IRQ5 10110 = IRQ6 10111 = IRQ7 11000 = IRQ8 11001 = IRQ9 11010 = IRQ10 11011 = IRQ11 11100 = IRQ12 11101 = IRQ13 11110 = IRQ14 11111 = IRQ15				
PCICFG 4Bh			**Reserved**				**Default = 00h**
PCICFG 4Ch			**Docking Detect Interrupt Assignment Register**				**Default = 01h**
Host controller type: 0 = FireStar (burst two data phases) 1 = Viper-N+ (send single data phase on IRQ driveback)	Reserved		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. **Level Mode:** 00000 = Disabled 00001 = PCIRQ0# (Default) 00010 = PCIRQ1# 00011 = PCIRQ2# 00100 = PCIRQ3# 00101 = ACPI0 00110 = ACPI1 00111 = ACPI2 01000 = ACPI3 01001-01111 = Rsrvd **Edge Mode:** 10000 = IRQ0 10001 = IRQ1 10010 = IRQ2 10011 = IRQ3 10100 = IRQ4 10101 = IRQ5 10110 = IRQ6 10111 = IRQ7 11000 = IRQ8 11001 = IRQ9 11010 = IRQ10 11011 = IRQ11 11100 = IRQ12 11101 = IRQ13 11110 = IRQ14 11111 = IRQ15				
PCICFG 4Dh-4Fh			**Reserved**				**Default = 00h**

5.2.2 82C824-Specific Register Group

The 82C824 defines many special functions that require enabling and monitoring through a dedicated register set. The 82C824-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).

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

7	6	5	4	3	2	1	0
PCICFG 50h		**PCI Host Feature Control Register (common to both slots)**					**Default = 01h**
ZVP mode strap selected: 0 = No 1 = Yes	Legacy mode strap-selected (RO): 0 = No 1 = Yes	Vendor ID feature strap-selected: 0 = No 1 = Yes	IRQLAT function on CLK-RUN# strap-selected: 0 = No 1 = Yes	Card detect debounce: 0 = 0.25 sec 1 = 1.0 sec	CLKRUN (in host interface): 0 = Enabled per PCI 1 = Disabled, CLKRUN# tristated	SPKROUT: 0 = Tristated 1 = Driven	Reserved
PCICFG 51h			**Slot Feature Control Register 1**				**Default = 00h**
Slot clock divisor: 00 = 1 (Default) 01 = 2 10 = 3 11 = 4		Slot clock source: 0 = PCICLK 1 = EXTCLK	Mode select: 0 = Automatic 1 = Force async	AUDIO pin if dock attached: 0 = INTB# 1 = AUDIO (Slot A only)	Slot threshold voltage: 0 = 3.3V 1 = 5.0V	Output Drive Select: 00 = 5.0V 16-bit PC cards, 3.3V CardBus cards 01 = 3.3V 16-bit PC cards 10 = 3.3V PCI dock 11 = 5.0V PCI dock	
PCICFG 52h			**Slot Feature Control Register 2**				**Default = 0Fh**
Secondary PCICLK Skew: This value selects the approximate delay, in nanoseconds, that the internal secondary PCICLK must be skewed in order to compensate for external buffer delays. 0000 = No delay 0001 = 1ns 0010 = 2ns ... 1101 = 13ns 1110 = 14ns 1111=15ns				CCLKRUN# pin if dock attached: 0 = CGNT2# 1 = CCLKRUN# (Slot A only)	Reserved		Type 1 to Type 0 conversion blocked from secondary to primary: 0 = No 1 = Yes (Default)
PCICFG 53h			**PCMCIA Controller Configuration Register**				**Default = 03h**
Reserved	Reserved	PCMCIA register offset: 00 = 00h (Slot A) 01 = 40h (Slot B) 10 = 80h (Slot C) 11 = C0h (Slot D)		Global PCMCIA read prefetch: 0 = Disable 1 = Enable	Global PCMCIA posted writes: 0 = Disable 1 = Enable	PCMCIA clock generation: 00 = PCICLK /1 01 = /2 10 = /3 11 = /4 (Default)	
PCICFG 54h			**IRQ Driveback Address Register - Byte 0: Address Bits [7:0]**				**Default = 30h**
IRQ Driveback Protocol Address Bits: - When the 82C824 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. - Bits 2:0 are reserved to be 000 and are read-only. - 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**

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

7	6	5	4	3	2	1	0
PCICFG 58h			**DRQ Remap Base Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
DRQ Remap Base Address Bits: - 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. - The 82C824 logic uses this base address two ways: 1) to claim accesses to a PCMCIA DMA controller channel; 2) 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): 0 = Not claimed 1 = On slot interface	Channel 6 (DMAC2): 0 = Not claimed 1 = On slot interface	Channel 5 (DMAC2): 0 = Not claimed 1 = On slot interface	DMAC responsibility (RO): 0 = Secondary (always) 1 = Master	Channel 3 (DMAC1): 0 = Not claimed 1 = On slot interface	Channel 2 (DMAC1): 0 = Not claimed 1 = On slot interface	Channel 1 (DMAC1): 0 = Not claimed 1 = On slot interface	Channel 0 (DMAC1): 0 = Not claimed 1 = On slot interface
PCICFG 5Dh			**SMI Status Register (for Slot A only) (Write 1 to clear bit)**				**Default = 00h**
Window 7 generated SMI: 0 = No 1 = Yes	Window 6 generated SMI: 0 = No 1 = Yes	Window 5 generated SMI: 0 = No 1 = Yes	Window 4 generated SMI: 0 = No 1 = Yes	Window 3 generated SMI: 0 = No 1 = Yes	Window 2 generated SMI: 0 = No 1 = Yes	Window 1 generated SMI: 0 = No 1 = Yes	Window 0 generated SMI: 0 = No 1 = Yes
PCICFG 5Eh			**Primary Retry Limit Register**				**Default = 07h**
Slow decode for ISA windows: 0 = Disable 1 = Enable	Prefetch on upstream transactions: 0 = Disable 1 = Enable	Posted writes on upstream transactions: 0 = Disable 1 = Enable	Core voltage (RO): 0 = 3.3V 1 = 5.0V	Retry count readback control: 0 = Write posting retries on secondary 1 = Retries on primary	Retry Limit: These bits relate to the number of times that the 82C824, as a slave, will retry accesses on the primary. If this limit is exceeded, the 82C824 generates SERR# to the host. 000=2^8 100=2^{16} 001=2^{10} 101=2^{20} 010=2^{12} 110=2^{24} 011=2^{14} 111= Infinite retries (Default)		
PCICFG 5Fh			**Retry Count Readback Register**				**Default = 00h**
This register returns the number of retry attempts either as master (on the secondary) or as slave (on the primary).							

IRQLAT function on CLKRUN# - For the purposes of generating IRQs to the host for chipsets without IRQ driveback handling capability, the CLKRUN# feature can be disabled and replaced with IRQLAT. When this feature is enabled, IRQLAT goes active on a driveback cycle to generate IRQ15-0. In this way, an external latch can be used to directly drive the IRQ lines.

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 82C824 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.

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 PCMCIA card or a CardBus card 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 bits 51h[2:0] as desired.

5.2.3 CardBus Register Group

The CardBus register group is accessible through two different means. It is always accessible as part of the PCI configuration space at the indexes shown below. 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.

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.

Table 5-3 System Memory CardBus Registers

CardBus Base Address plus:	Name
000h	CardBus Socket Event Register
004h	CardBus Socket Mask Register
008h	CardBus Socket Present State Register
00Ch	CardBus Force Event Register
010h	CardBus Control Register
014-7FFh	Reserved
800-847h	16-bit PC Card Registers (365 register set)

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

7	6	5	4	3	2	1	0
PCICFG 60h / MEMOFST 00h			**CardBus Socket Event Register - Byte 0**				**Default = 00h**
Reserved: Write as read.				Power cycle complete: 0 = No 1 = Yes Write 1 to clear	CCD2# status change: 0 = No 1 = Yes Write 1 to clear	CCD1# status change: 0 = No 1 = Yes Write 1 to clear	CSTSCHG status change: 0 = No 1 = Yes Write 1 to clear
PCICFG 61h / MEMOFST 01h			**CardBus Socket Event Register - Byte 1**				**Default = 00h**
Reserved: Write as read.							
PCICFG 62h / MEMOFST 02h			**CardBus Socket Event Register - Byte 2**				**Default = 00h**
Reserved: Write as read.							
PCICFG 63h / MEMOFST 03h			**CardBus Socket Event Register - Byte 3**				**Default = 00h**
Reserved. Write as read.							
PCICFG 64h / MEMOFST 04h			**CardBus Socket Mask Register - Byte 0**				**Default = 00h**
Reserved: Write as read.				Power cycle status change event: 0 = Mask 1 = Enable	CCD2# status change event: 0 = Mask 1 = Enable	CCD1# status change event: 0 = Mask 1 = Enable	CSTSCHG status change event: 0 = Mask 1 = Enable
PCICFG 65h / MEMOFST 05h			**CardBus Socket Mask Register - Byte 1**				**Default = 00h**
Reserved: Write as read.							
PCICFG 66h / MEMOFST 06h			**CardBus Socket Mask Register - Byte 2**				**Default = 00h**
Reserved: Write as read.							
PCICFG 67h / MEMOFST 07h			**CardBus Socket Mask Register - Byte 3**				**Default = 00h**
Reserved: Write as read.							
PCICFG 68h / MEMOFST 08h			**CardBus Socket Present State Register (RO) - Byte 0**				**Default = 00h**
Card recognized - updated only on card insertion: 0 = Yes 1 = No	Reserved: Write as read.	Card Type - updated only on card insertion: 00 = Reserved 01 = PCMCIA 10 = CardBus 11 = Docking station		Power cycle status: 0 = Not successful 1 = Successful	CCD2-1# state: 00 = Card inserted 01 = No card inserted 10 = No card inserted 11 = No card inserted		CSTSCHG state: 0 = Inactive 1 = Active (high)

Table 5-4 CardBus Register Group - PCICFG 60h-74h / MEMOFST 00h-7Fh (cont.)

7	6	5	4	3	2	1	0
PCICFG 69h / MEMOFST 09h			**CardBus Socket Present State Register - Byte 1**				**Default = 00h**
Reserved: Write as read.		Alternative Voltage Y card detected: 0 = No 1 = Yes	Alternative Voltage X card detected: 0 = No 1 = Yes	3.3V card detected: 0 = No 1 = Yes	5.0V card detected: 0 = No 1 = Yes	Bad VCC request (outside CVS1-2, CCD1-2# range): 0 = No 1 = Yes	Data lost (card removed before transaction completed): 0 = No 1 = Maybe
PCICFG 6Ah / MEMOFST 0Ah			**CardBus Socket Present State Register - Byte 2**				**Default = 00h**
Reserved: Write as read.							
PCICFG 6Bh / MEMOFST 0Bh			**CardBus Socket Present State Register - Byte 3 (bits are writeable)**				**Default = 30h**
Socket can supply Voltage Y: 0 = No 1 = Yes	Socket can supply Voltage X: 0 = No 1 = Yes	Socket can supply 3.3V: 0 = No 1 = Yes	Socket can supply 5.0V: 0 = No 1 = Yes	Reserved: Write as read.			
PCICFG 6Ch / MEMOFST 0Ch			**CardBus Force Event Register - Byte 0**				**Default = 00h**
Force card recognized bit to 1: 0 = No 1 = Yes	Reserved: Write as read.	Force card type: 00 = Reserved 01 = PCMCIA 10 = CardBus 11 = Docking station		Force power cycle event: 0 = No 1 = Yes	Force CCD2# event: 0 = No 1 = Yes	Force CCD1# event: 0 = No 1 = Yes	Force CSTSCHG event: 0 = No 1 = Yes
PCICFG 6Dh / MEMOFST 0Dh			**CardBus Force Event Register - Byte 1**				**Default = 00h**
Reserved: Write as read.	Force retest of CVS1-2, CCD1-2# pins (or force bits): 0 = No 1 = Yes	Force YV card detected bit to 1: 0 = No 1 = Yes	Force XV card detected bit to 1: 0 = No 1 = Yes	Force 3.3V card detected bit to 1: 0 = No 1 = Yes	Force 5.0V card detected bit to 1: 0 = No 1 = Yes	Force bad VCC request bit to 1: 0 = No 1 = Yes	Force data lost bit to 1: 0 = No 1 = Yes
PCICFG 6Eh / MEMOFST 0Eh			**CardBus Force Event Register - Byte 2**				**Default = 00h**
Reserved: Write as read.							
PCICFG 6Fh / MEMOFST 0Fh			**CardBus Force Event Register - Byte 3**				**Default = 00h**
Reserved: Write as read.							
PCICFG 70h / MEMOFST 10h			**CardBus Control Register - Byte 0**				**Default = 00h**
Reserved: Write as read.	Card VCC power request: 000 = Power off 001 = Reserved 010 = 5.0V 011 = 3.3V 100 = Voltage X 101 = Voltage Y 11x = Reserved			Reserved: Write as read.	Card VPP power request: 000 = Programming voltage off 001 = 12V 010 = 5.0V 011 = 3.3V 100 = Voltage X 101 = Voltage Y 11x = Reserved		
PCICFG 71h/ MEMOFST 11h			**CardBus Control Register - Byte 1**				**Default = 00h**
Reserved: Write as read.							
PCICFG 72h / MEMOFST 12h			**CardBus Control Register - Byte 2**				**Default = 00h**
Reserved: Write as read.							
PCICFG 73h / MEMOFST 13h			**CardBus Control Register - Byte 3**				**Default = 00h**
Reserved: Write as read.							
PCICFG 74h / MEMOFST 14h			**Reserved**				**Default = 00h**

5.2.3.1 Power Control

PCICFG 70h[6:4] set the external VCC5EN and VCC3EN pin levels; PCICFG 70h[2:0] used in conjunction with PCICFG 70h[6:4] set the external VPP12 and VPP3/5 pin levels. Because only these four pins are available on the 82C824 interface, the system must be designed to interpret these signals properly and select the correct voltage for the application.

Table 5-5 shows how the external control signals are interpreted by a typical power control chip, the Micrel 2560. By itself, the 2560 chip would not be sufficient to implement a system with support for the 'x.xV' selection in addition to the standard 3.3V and 5.0V selections; additional gating logic and power control devices would be necessary. Even with additional logic, the VPP selection is limited to only three of the four possibilities of 12.0V, 5.0V, 3.3V, and x.xV; system design requirements must dictate the voltages that will be made available.

In the table, 'Disabled' = high impedance; 'Ground' indicates that the voltage source is actively clamped to ground.

Table 5-5 Voltage Control Pin Interpretations using Micrel 2560 Chip

PCICFG 70h[6:4]	VCC5EN	VCC3EN	Card VCC Selected on Micrel 2560	PCICFG 70h[2:0]	VPP Requested	VPP12	VPP3/5	Card VPP Selected on Micrel 2560
00x, 101, 11x	0	0	Disabled	xxx	--	0	0	Disabled
010	1	0	5.0V	000, 101, 11x	--	0	0	Disabled
				001	12V	1	0	12.0V
				010	5.0V	0	1	5.0V
				011	3.3V	1	1	Ground
				100	x.xV	0	0	Disabled
011	0	1	3.3V	000, 101, 11x	--	0	0	Disabled
				001	12V	1	0	12.0V
				010	5.0V	1	1	Ground
				011	3.3V	0	1	3.3V
				100	x.xV	1	1	Ground
100	1	1	x.xV (Note)	000, 101, 11x	--	0	0	--
				001	12V	1	0	--
				010	5.0V	0	0	--
				011	3.3V	1	1	--
				100	x.xV	0	1	--

Note: This setting selects 3.3V on the Micrel 2560 part, but should not be used for this purpose.

5.2.4 Docking Station Window Selection Group

The remainder of the 82C824 PCI-to-CardBus configuration registers are used to select the memory or I/O address ranges that will be claimed by the bridge and passed onto the secondary PCI bus. These windows overlap in function with the predefined CardBus I/O and memory windows, but are more versatile so as to be used for docking station support. However, applications can use the docking station window selection group to access CardBus cards as well. Table 5-6 summarizes the features.

Note: Only Slot A provides the docking station registers.

5.2.4.1 Warning on Using Docking Station Windows

The docking station access windows allow far more flexibility in cycle selection, masking, etc. than do the CardBus window registers. Whenever the 82C824 chip is reset, the docking station window registers are reset to a default state that is identical to that of the CardBus windows. However, once the docking station window registers are changed from default state, the CardBus windows are no longer compatible with the CardBus standard register set requirements.

For example, assume a CardBus card is inserted and standard CardBus Card Services code is in use. If specialized software changes docking station window 4 from its default "memory" setting to make it an I/O window, the next time Card Services accesses that window it will be unable to change it back to a memory window and the application will fail.

5.2.4.2 Docking Station Window Registers

The docking station registers are listed in Table 5-6. Table 5-7 summarizes the default settings for each register and Table 5-8 gives the register's bit formats.

Table 5-6 Docking Station Access Windows

Docking Station Access Window #	Default Mask	CardBus Window Name, Bits Decoded	Memory or I/O Selectable?	Can Generate SMI#?
0	000FFFh	None, Decode A[31:12]	Memory	Yes
1	000FFFh	None, Decode A[31:12]	Memory	Yes
2	000003h	None, Decode A[15:0]	I/O	Yes
3	000003h	None, Decode A[15:0]	I/O	Yes
4	000FFFh	Memory Window 0, A[31:12]	Yes - Defaults to Memory	Yes
5	000FFFh	Memory Window 1, A[31:12]	Yes - Defaults to Memory	Yes
6	000003h	I/O Window 0, A[31:2]	Yes - Defaults to I/O	Yes
7	000003h	I/O Window 1, A[31:2]	Yes - Defaults to I/O	Yes

Table 5-7 Power-on Reset Defaults for Docking Station Window Registers

Register/0ffset	Window 0	Window 1	Window 2	Window 3	Window 4	Window 5	Window 6	Window 7
Start Address/x0h	FFFFF000h	FFFFF000h	FFFFF000h	FFFFF000h	FFFFF000h	FFFFF000h	FFFFF000h	FFFFF000h
Stop Address/x4h	00h	00h	00h	00h	00h	00h	00h	00h
Decoding Mask/x8h	0FFFh	0FFFh	03h	03h	0FFFh	0FFFh	03h	03h
Control /xBh	00h	00h	00h	00h	68h	68h	00h	00h

.

Table 5-8 Docking Station Window Registers - PCICFG 80h-FFh

7	6	5	4	3	2	1	0
PCICFG 80h			**Window 0 Start Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 0 Start Address Bits: - Register bits [31:0] indicate the start address for Memory Window 0. - Bits [11:0] are read only and always return 0 to indicate a minimum 4KB boundary.							
PCICFG 81h			**Window 0 Start Address Register - Byte 1: Address Bits [15:8]**				**Default = F0h**
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 Stop Address Bits: - Register bits [31:0] indicate the stop address for Memory Window 0. - Bits [11:0] are read only and always return 0 to indicate a minimum 4KB boundary.							
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 = FFh**
Window 0 Mask Bits: - Mask register bits [23:12] allow Window 0 to be aliased throughout the memory or I/O address space. - Setting any bit to a 1 masks out the comparison on this bit. - The register should be written to 0 to decode the entire address. - Bits [11:0] are always 1 (masked). - Mask register bits [23:0] are fixed to 000FFFh to force a 4KB boundary							
PCICFG 89h			**Window 0 Mask Register - Byte 1: Mask Bits [15:8]**				**Default = 0Fh**
PCICFG 8Ah			**Window 0 Mask Register - Byte 2: Mask Bits [23:16]**				**Default = 00h**
PCICFG 8Bh			**Window 0 Control Register**				**Default = 08h**
Window points to ISA bus: 0 = No 1 = Yes	Reads are prefetchable: 0 = No 1 = Yes Set to 0 for I/O window	Writes can be posted: 0 = No 1 = Yes Set to 0 for I/O window	Reserved	Cycle qualifier: 0 = I/O 1 = Memory	Window 0 Trap/SMI#: 0 = Disable 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: - Register bits [31:0] indicate the start address for Memory Window 1. - Bits [11:0] are read only and always return 0 to indicate a minimum 4KB boundary.							
PCICFG 91h			**Window 1 Start Address Register - Byte 1: Address Bits [15:8]**				**Default = F0h**
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**

Table 5-8 Docking Station Window Registers - PCICFG 80h-FFh (cont.)

7	6	5	4	3	2	1	0
PCICFG 94h			**Window 1 Stop Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 1 Stop Address Bits: - Register bits [31:0] indicate the stop address for Memory Window 0. - Bits [11:0] are read only and always return 0 to indicate a minimum 4KB boundary.							
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 = FFh**
Window 1 Mask Bits: - Mask register bits [23:12] allow Window 0 to be aliased throughout the memory or I/O address space. - Setting any bit to a 1 masks out the comparison on this bit. - The register should be written to 0 to decode the entire address. - Bits [11:0] are always 1 (masked). - Mask register bits [23:0] are fixed to 000FFFh to force a 4KB boundary							
PCICFG 99h			**Window 1 Mask Register - Byte 1: Mask Bits [15:8]**				**Default = 0Fh**
PCICFG 9Ah			**Window 1 Mask Register - Byte 2: Mask Bits [23:16]**				**Default = 00h**
PCICFG 9Bh			**Window 1 Control Register**				**Default = 08h**
Window points to ISA bus: 0 = No 1 = Yes	Reads are prefetchable: 0 = No 1 = Yes Set to 0 for I/O window	Writes can be posted: 0 = No 1 = Yes Set to 0 for I/O window	Reserved	Cycle qualifier: 0 = I/O 1 = Memory	Window 1 Trap/SMI#: 0 = Disable 1 = Enable	Reserved	
PCICFG 9Ch-9Fh			**Reserved**				**Default = 00h**
PCICFG A0h			**Window 2 Start Address Register - Byte 0: Address Bits [7:0]**				**Default = FCh**
Window 2 Start Address Bits: - Register bits [31:0] indicate the start address for I/O Window 2. - Bits [31:16] are read only and always return 0.						RO: Always returns 0	Decoding: 0 = 16-bit (always) AD[31:16] = 0 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 = 00h**
PCICFG A3h			**Window 2 Start Address Register - Byte 3: Address Bits [31:24]**				**Default = 00h**
PCICFG A4h			**Window 2 Stop Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 2 Stop Address Bits: - Register bits [31:0] indicate the stop address for I/O Window 2. - Bits [11:0] are read only and always return 0 to indicate a minimum 4KB boundary.							
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**

Table 5-8 Docking Station Window Registers - PCICFG 80h-FFh (cont.)

7	6	5	4	3	2	1	0
PCICFG A8h			**Window 2 Mask Register - Byte 0: Mask Bits [7:0]**				**Default = 03h**
Window 2 Mask Bits: - Mask register bits [23:0] are fixed to 000003h to force a four-byte granularity.							
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**
PCICFG ABh			**Window 2 Control Register**				**Default = 00h**
Window points to ISA bus: 0 = No 1 = Yes	Reserved	Reserved	Reserved	Cycle qualifier: 0 = I/O (always) 1 = Memory	Window 2 Trap/SMI#: 0 = Disable 1 = Enable	Reserved	
PCICFG ACh-AFh			**Reserved**				**Default = 00h**
PCICFG B0h			**Window 3 Start Address Register - Byte 0: Address Bits [7:0]**				**Default = FCh**
Window 3 Start Address Bits: - Register bits [31:0] indicate the start address for I/O Window 3. - Bits [31:16] are read only and always return 0 to indicate a minimum 4KB boundary.						RO: Always returns 0	Decoding: 0 = 16-bit (always) AD[31:16] = 0 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 = 00h**
PCICFG B3h			**Window 3 Start Address Register - Byte 3: Address Bits [31:24]**				**Default = 00h**
PCICFG B4h			**Window 3 Stop Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 3 Stop Address Bits: - Register bits [31:0] indicate the stop address for I/O Window 3. - Bits [11:0] are read only and always return 0 to indicate a minimum 4KB boundary.							
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 = 00h**
Window 3 Mask Bits: - Mask register bits [23:0] are fixed to 000003h to force a four-byte granularity.							
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**
PCICFG BBh			**Window 3 Control Register**				**Default = 00h**
Window points to ISA bus: 0 = No 1 = Yes	Reserved	Reserved	Reserved	Cycle qualifier: 0 = I/O (always) 1 = Memory	Window 3 Trap/SMI#: 0 = Disable 1 = Enable	Reserved	
PCICFG BCh-BFh			**Reserved**				**Default = 00h**

Table 5-8 Docking Station Window Registers - PCICFG 80h-FFh (cont.)

7	6	5	4	3	2	1	0
PCICFG C0h			**Window 4 Start Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window Start Address Bits: - Register bits [31:0] indicate the start address for one of the eight memory or I/O windows. - The selection between memory or I/O, as well as other feature selections, are made through the Window 4 Control Register.						RO: Always returns 0	If memory: reads 0. If I/O: Decoding 0 = 16-bit AD[31:16] = 0 1 = 32-bit
PCICFG C1h			**Window 4 Start Address Register - Byte 1: Address Bits [15:8]**				**Default = F0h**
PCICFG C2h			**Window 4 Start Address Register - Byte 2: Address Bits [23:16]**				**Default = FFh**
PCICFG C3h			**Window 4 Start Address Register - Byte 3: Address Bits [31:24]**				**Default = FFh**
PCICFG C4h			**Window 4 Stop Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 4 Address Bits: - Register bits [31:0] indicate the stop address for one of the eight memory or I/O windows.						RO: Always returns 0	
PCICFG C5h			**Window 4 Stop Address Register - Byte 1: Address Bits [15:8]**				**Default = 00h**
PCICFG C6h			**Window 4 Stop Address Register - Byte 2: Address Bits [23:16]**				**Default = 00h**
PCICFG C7h			**Window 4 Stop Address Register - Byte 3: Address Bits [31:24]**				**Default = 00h**
PCICFG C8h			**Window 4 Mask Register - Byte 0: Mask Bits [7:0]**				**Default = 00h**
Window 4 Mask Bits: - Mask register bits [23:2] allow Window 4 to be aliased throughout the memory or I/O address space. - Setting any bit to a 1 masks out the comparison on this bit. - The register should be written to 0 to decode the entire address. - Bits [1:0] are always 11 (masked).						RO: Always returns 1.	
PCICFG C9h			**Window 4 Mask Register - Byte 1: Mask Bits [15:8]**				**Default = 00h**
PCICFG CAh			**Window 4 Mask Register - Byte 2: Mask Bits [23:16]**				**Default = 00h**
PCICFG CBh				**Window 4 Control Register**			**Default = 48h**
Window points to ISA bus: 0 = No 1 = Yes	Reads are prefetchable: 0 = No 1 = Yes Set to 0 for I/O window	Writes can be posted: 0 = No 1 = Yes Set to 0 for I/O window	Reserved	Cycle qualifier: 0 = I/O 1 = Memory	Window 4 Trap/SMI#: 0 = Disable 1 = Enable	Reserved	
PCICFG CCh-CFh				**Reserved**			**Default = 00h**

Table 5-8 Docking Station Window Registers - PCICFG 80h-FFh (cont.)

7	6	5	4	3	2	1	0
PCICFG D0h			**Window 5 Start Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 5 Start Address Bits: - Register bits [31:0] indicate the start address for one of the eight memory or I/O windows. - The selection between memory or I/O, as well as other feature selections, are made through the Window 5 Control Register.						RO: Always returns 0	If memory: reads 0. If I/O: Decoding 0 = 16-bit AD[31:16] = 0 1 = 32-bit
PCICFG D1h			**Window 5 Start Address Register - Byte 1: Address Bits [15:8]**				**Default = F0h**
PCICFG D2h			**Window 5 Start Address Register - Byte 2: Address Bits [23:16]**				**Default = FFh**
PCICFG D3h			**Window 5 Start Address Register - Byte 3: Address Bits [31:24]**				**Default = FFh**
PCICFG D4h			**Window 5 Stop Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 5 Stop Address Bits: - Register bits [31:0] indicate the stop address for one of the eight memory or I/O windows.						RO: Always returns 0	
PCICFG D5h			**Window 5 Stop Address Register - Byte 1: Address Bits [15:8]**				**Default = 00h**
PCICFG D6h			**Window 5 Stop Address Register - Byte 2: Address Bits [23:16]**				**Default = 00h**
PCICFG D7h			**Window 5 Stop Address Register - Byte 3: Address Bits [31:24]**				**Default = 00h**
PCICFG D8h			**Window 5 Mask Register - Byte 0: Mask Bits [7:0]**				**Default = 00h**
Window 5 Mask Bits: - Mask register bits [23:2] allow Window 4 to be aliased throughout the memory or I/O address space. - Setting any bit to a 1 masks out the comparison on this bit. - The register should be written to 0 to decode the entire address. - Bits [1:0] are always 11 (masked).						RO: Always returns 1.	
PCICFG D9h			**Window 5 Mask Register - Byte 1: Mask Bits [15:8]**				**Default = 00h**
PCICFG DAh			**Window 5 Mask Register - Byte 2: Mask Bits [23:16]**				**Default = 00h**
PCICFG DBh				**Window 5 Control Register**			**Default = 48h**
Window points to ISA bus: 0 = No 1 = Yes	Reads are prefetchable: 0 = No 1 = Yes Set to 0 for I/O window	Writes can be posted: 0 = No 1 = Yes Set to 0 for I/O window	Reserved	Cycle qualifier: 0 = I/O 1 = Memory	Window 5 Trap/SMI#: 0 = Disable 1 = Enable	Reserved	
PCICFG DCh-DFh				**Reserved**			**Default = 00h**

Table 5-8 Docking Station Window Registers - PCICFG 80h-FFh (cont.)

7	6	5	4	3	2	1	0
PCICFG E0h			**Window 6 Start Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 6 Address Bits: - Register bits [31:0] indicate the start address for one of the eight memory or I/O windows. - The selection between memory or I/O, as well as other feature selections, are made through the Window 6 Control Register.						RO: Always returns 0	If memory: reads 0. If I/O: Decoding 0 = 16-bit AD[31:16] = 0 1 = 32-bit
PCICFG E1h			**Window 6 Start Address Register - Byte 1: Address Bits [15:8]**				**Default = F0h**
PCICFG E2h			**Window 6 Start Address Register - Byte 2: Address Bits [23:16]**				**Default = FFh**
PCICFG E3h			**Window 6 Start Address Register - Byte 3: Address Bits [31:24]**				**Default = FFh**
PCICFG E4h			**Window 6 Stop Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 6 Stop Address Bits: - Register bits [31:0] indicate the stop address for one of the eight memory or I/O windows.						RO: Always returns 0	
PCICFG E5h			**Window 6 Stop Address Register - Byte 1: Address Bits [15:8]**				**Default = 00h**
PCICFG E6h			**Window 6 Stop Address Register - Byte 2: Address Bits [23:16]**				**Default = 00h**
PCICFG E7h			**Window 6 Stop Address Register - Byte 3: Address bits [31:24]**				**Default = 00h**
PCICFG E8h			**Window 6 Mask Register - Byte 0: Mask Bits [7:0]**				**Default = 03h**
Window 6 Mask Bits: - Mask register bits [23:2] allow Window 4 to be aliased throughout the memory or I/O address space. - Setting any bit to a 1 masks out the comparison on this bit. - The register should be written to 0 to decode the entire address. - Bits [1:0] are always 11 (masked).						RO: Always returns 1.	
PCICFG E9h			**Window 6 Mask Register - Byte 1: Mask Bits [15:8]**				**Default = 00h**
PCICFG EAh			**Window 6 Mask Register - Byte 2: Mask Bits [23:16]**				**Default = 00h**
PCICFG EBh				**Window 6 Control Register**			**Default = 00h**
Window points to ISA bus: 0 = No 1 = Yes	Reads are prefetchable: 0 = No 1 = Yes Set to 0 for I/O window	Writes can be posted: 0 = No 1 = Yes Set to 0 for I/O window	Reserved	Cycle qualifier: 0 = I/O 1 = Memory	Window 6 Trap/SMI#: 0 = Disable 1 = Enable	Reserved	
PCICFG ECh-EFh				**Reserved**			**Default = 00h**

Table 5-8 Docking Station Window Registers - PCICFG 80h-FFh (cont.)

7	6	5	4	3	2	1	0
PCICFG F0h			**Window 7 Start Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 7 Address Bits: - Register bits [31:0] indicate the start address for one of the eight memory or I/O windows. - The selection between memory or I/O, as well as other feature selections, are made through the Window 7 Control Register.						RO: Always returns 0	If memory: reads 0. If I/O: Decoding 0 = 16-bit AD[31:16] = 0 1 = 32-bit
PCICFG F1h			**Window 7 Start Address Register - Byte 1: Address Bits [15:8]**				**Default = F0h**
PCICFG F2h			**Window 7 Start Address Register - Byte 2: Address Bits [23:16]**				**Default = FFh**
PCICFG F3h			**Window 7 Start Address Register - Byte 3: Address Bits [31:24]**				**Default = FFh**
PCICFG F4h			**Window 7 Stop Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
Window 7 Stop Address Bits: - Register bits [31:0] indicate the stop address for one of the eight memory or I/O windows.						RO: Always returns 0	
PCICFG F5h			**Window 7 Stop Address Register - Byte 1: Address Bits [15:8]**				**Default = 00h**
PCICFG F6h			**Window 7 Stop Address Register - Byte 2: Address Bits [23:16]**				**Default = 00h**
PCICFG F7h			**Window 7 Stop Address Register - Byte 3: Address Bits [31:24]**				**Default = 00h**
PCICFG F8h			**Window 7 Mask Register - Byte 0: Mask Bits [7:0]**				**Default = 03h**
Window 7 Mask Bits: - Mask register bits [23:2] allow Window 4 to be aliased throughout the memory or I/O address space. - Setting any bit to a 1 masks out the comparison on this bit. - The register should be written to 0 to decode the entire address. - Bits [1:0] are always 11 (masked).						RO: Always returns 1.	
PCICFG F9h			**Window 7 Mask Register - Byte 1: Mask Bits [15:8]**				**Default = 00h**
PCICFG FAh			**Window 7 Mask Register - Byte 2: Mask Bits [23:16]**				**Default = 00h**
PCICFG FBh				**Window 7 Control Register**			**Default = 00h**
Window points to ISA bus: 0 = No 1 = Yes	Reads are prefetchable: 0 = No 1 = Yes Set to 0 for I/O window	Writes can be posted: 0 = No 1 = Yes Set to 0 for I/O window	Reserved	Cycle qualifier: 0 = I/O 1 = Memory	Window 7 Trap/SMI#: 0 = Disable 1 = Enable	Reserved	
PCICFG FCh-FFh				**Reserved**			**Default = 00h**

Cycle Decoding - Windows 4-7 can select either memory or I/O decoding, and allows for a decode range anywhere from one dword to the entire address space.

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 82C824 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 82C824 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.

5.2.4.3 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 82C825 chip will claim the cycle and wait for positive decode on the ISA bus.
  - If positive decode is determined, the 82C825 logic will terminate the cycle normally.
  - If no positive decode can be achieved, the 82C825 logic will terminate the cycle with a target abort. Once this occurs, the 82C824 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 unit defined in PCICFG 5Eh[2:0].

5.2.4.4 Dual ISA Buses

Dual ISA buses are possible with the 82C824 chip used in conjunction with the 82C825 PCI-ISA Bridge chip. This feature depends on the ISA Windows feature of the 82C824 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 82C824 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 82C824 chip. Therefore, the 82C824 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 82C824 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 82C824 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.

5.3 PCMCIA Controller 0 and 1 Register Groups

The PCMCIA Controller core provides programming registers grouped as General Purpose, I/O Mapping Window, Memory Mapping Window, and Special. The PCMCIA Socket Configuration Registers are addressed for Slot A, B, C, or D. The index/data address to which the registers respond must be selected through the PCMCIA configuration register at PCICFG 53h (in the PCI configuration space). The registers can also be directly accessed in the system memory space if desired as MEMOFST from the CardBus Base Address (PCICFG 10h).

5.3.1 General Purpose Register Group

Table 5-9 gives the bit formats for the PCMCIA General Purpose Register Group. The Index corresponds to the slot (Slot A starts at Index 00h, Slot B at 40h, Slot C at 80h and Slot D at C0h) or as previously stated, these registers can be accessed directly in system memory (MEMOFST).

Table 5-9 PCMCIA General Purpose Register Group

7	6	5	4	3	2	1	0
Index 00h, 40h, 80h, C0h / MEMOFST 800h			**Identification Register (RO)**				**Default = 87h**
Interface Type - indicate supported interfaces. 00 = I/O only 01 = Memory only 10 = Memory and I/O (always) 11 = Reserved		Chip revision level (RO): 00 = 1st revision		Revision number bits [3:0] (RO): 1111 = OPTi 82C852 PCMCIA Controller 0111 = OPTi 82C824 CardBus Controller (always) 0100 = Intel 82C365SL 0010 = Cirrus 672x			
Index 01h, 41h, 81h, C1h / MEMOFST 801h			**Interface Status Register (RO)**				**Default = 00h**
Reserved	Card power: 0 = Off 1 = On	RDY/BSY# state: 0 = Busy 1 = Ready	WP state: 0 = Not write protected 1 = Write protected	CD2# state: 0 = CD2# high 1 = CD2# low	CD1# state: 0 = CD1# high 1 = CD1# low	BVD2/ SPKR state: 0 = Low 1 = High	BVD1/ STSCHG# state: 0 = Low 1 = High
Index 02h, 42h, 82h, C2h / MEMOFST 802h			**Power Control Register**				**Default = 00h**
Socket signals: 0 = Disable 1 = Enable (tristate or drive low)	Reserved: Write bit as read.	Auto card power-up on insertion (RO): 0 = Disable (always)	Card VCC control: Sets VCC5EN-VCC3EN to these bit values. Refer to Table 5-5.		Slot VCC threshold scaling: 0 = 3.3V 1 = 5.0V	Card VPP control: Sets VPPPGM-VPP3/5 to these bit values. Refer to Table 5-5.	
Index 03h, 43h, 83h, C3h / MEMOFST 803h			**Reset and General Control Register**				**Default = 00h**
Reserved: Write bit as read.	RESET signal state: 0 = Active (high) 1 = Inactive	PCMCIA card interface: 0 = Memory 1 = I/O	Reserved: Write bit as read.	IREQ routing: 0000 = None 0001 = Reserved 0010 = Reserved 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5	0110 = Reserved 0111 = IRQ7 1000 = Reserved 1001 = IRQ9 1010 = IRQ10	1011 = IRQ11 1100 = IRQ12 1101 = Reserved 1110 = IRQ14 1111 = IRQ15	
Index 04h, 44h, 84h, C4h / MEMOFST 804h			**Card Status Change Register**				**Default = 00h**
Reserved: Write bits as read.				CDx# status change or software interrupt: 0 = No 1 = Yes	RDY/BSY# has gone high: 0 = No 1 = Yes = 0 for I/O cards	BVD2 has gone low: 0 = No 1 = Yes = 0 for I/O cards	BVD1/ STSCHG# has gone low: 0 = No 1 = Yes

Table 5-9 PCMCIA General Purpose Register Group (cont.)

7	6	5	4	3	2	1	0
Index 05h, 45h, 85h, C5h / MEMOFST 805h STSCHG# Interrupt Configuration Register							**Default = 00h**
STSCHG# routing (overridden when PCICFG 3Eh[7] = 0): 0000 = None 0001 = Reserved 0010 = Reserved 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5 0110 = Reserved 0111 = IRQ7 1000 = Reserved 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = IRQ12 1101 = Reserved 1110 = IRQ14 1111 = IRQ15				STSCHG# on CD1-2# change: 0 = Disable 1 = Enable	STSCHG# on RDY/BSY# low-to-high change: 0 = Disable 1 = Enable	STSCHG# on battery warning BVD2 high-to-low change: 0 = Disable 1 = Enable	STSCHG# on battery dead BVD1 high-to-low change: 0 = Disable 1 = Enable
Index 06h, 46h, 86h, C6h / MEMOFST 806h Address Window Enable Register							**Default = 00h**
I/O Window 1: 0 = Disable 1 = Enable	I/O Window 0: 0 = Disable 1 = Enable	Reserved	Memory Window 4: 0 = Disable 1 = Enable	Memory Window 3: 0 = Disable 1 = Enable	Memory Window 2: 0 = Disable 1 = Enable	Memory Window 1: 0 = Disable 1 = Enable	Memory Window 0: 0 = Disable 1 = Enable

5.3.1.1 Power Control

Bits 02h[4:3] set the external VCC5EN and VCC3EN pin levels directly. Bits 02h[1:0] set the external VPPPGM-VPP3/5 pin levels directly. The interpretation of these signals depends on the external logic used. Socket Services must be aware of the hardware design in order to make the proper selections.

Table 5-10 shows how the external control signals are interpreted by a typical power control chip, the Micrel 2560. Using this device allows the power control to be compatible with the Intel 82365SL definition. In the table, 'Disabled' = high impedance; 'Ground' indicates that the voltage source is actively clamped to ground.

Slot VCC Threshold Scaling - The threshold level of the chip input buffers is controlled by bit 02h[2] and is independent of the voltage control pin settings. This independent selection feature allows the designer to choose a voltage control chip with different control pin selection definitions than the Micrel 2560 part. The voltage threshold should be set by software according to the card voltage being enabled.

Table 5-10 Voltage Control Pin Interpretations using Micrel 2560 Chip

VCC5EN bit 02h[4]	VCC3EN bit 02h[3]	Card VCC Selection	VPPPGM/ bit 02h[1]	VPP3/5 bit 02h[0]	Card VPP Selection
0	0	Disabled	0	0	Disabled
			0	1	Disabled
			1	0	Disabled
			1	1	Ground
0	1	3.3V	0	0	Disabled
			0	1	3.3V
			1	0	12V
			1	1	Ground
1	0	5.0V(1)	0	0	Disabled
			0	1	5.0V
			1	0	12V
			1	1	Ground
1	1	3.3V(2)	0	0	Disabled
			0	1	3.3V
			1	0	5.0V
			1	1	Ground

Notes: (1) If the VS2 (5.0VDET) pin from the card is grounded, VCC5EN-VCC3EN stay low when bits 02h[4:3] = 10. This feature prevents 5.0V from being applied to a 3.3V-only card.

(2) This setting of '11' should not be used to select 3.3V, as it may be reassigned to a lower voltage in the future.

5.3.2 I/O Mapping Window Register Group

The I/O Window Registers contain bits that maintain Cirrus 6722 compatibility. Only the window address offset is shown. See below for calculation of base index address for each of the two available windows.

Table 5-11 I/O Mapping Window Register Group - Offset +7h through +Bh

7	6	5	4	3	2	1	0
Offset +7h			**I/O Window Control Register**				**Default = 00h**
Window 1 additional wait states: 0 = None 1 = One	Window 1 zero-wait 8-bit cycles: 0 = No 1 = Yes	Window 1 size select: 0 = Use bit 4 1 = Use IOIS16#	Window 1 data size: 0 = 8 bits 1 = 16 bits	Window 0 additional wait states: 0 = None 1 = One	Window 0 zero-wait 8-bit cycles: 0 = No 1 = Yes	Window 0 size select: 0 = Use bit 0 1 = Use IOIS16#	Window 0 data size: 0 = 8 bits 1 = 16 bits
Offset +8h			**I/O Window Start Address Register - Byte 0: Address Bits IOS[7:0]**				**Default = 00h**
Offset +9h			**I/O Window Start Address Register - Byte 1: Address Bits IOS[15:8]**				**Default = 00h**
Offset +Ah			**I/O Window Stop Address Register - Byte 0: Address Bits IOST[7:0]**				**Default = 00h**
Offset +Bh			**I/O Window Stop Address Register - Byte 1: Address Bits IOST[15:8]**				**Default = 00h**

Table 5-12 Index Addresses for I/O Window Registers

I/O Window Index Address for:	I/O Window Control	I/O Window 0				I/O Window 1			
		Start Address		Stop Address		Start Address		Stop Address	
		Byte 0	Byte 1	Byte 0	Byte 1	Byte 0	Byte 1	Byte 0	Byte 1
Slot A	07h	08h	09h	0Ah	0Bh	0Ch	0Dh	0Eh	0Fh
Slot B	47h	48h	49h	4Ah	4Bh	4Ch	4Dh	4Eh	4Fh
Slot C	87h	88h	89h	8Ah	8Bh	8Ch	8Dh	8Eh	8Fh
Slot D	C7h	C8h	C9h	CAh	CBh	CCh	CDh	CEh	CFh
MEMOFST	807h	808h	809h	80Ah	80Bh	80Ch	80Dh	80Eh	80Fh

5.3.3 Memory Mapping Window Register Group

Only the window address offset is shown in Table 5-13. See Table 5-14 for the base index address for each of the five available windows.

Zero Wait States - This setting enables shorter cycles for faster PCMCIA cards.

Command Length - This value selects the command length for both 8-bit and 16-bit windows. In the 82365SL part, this value controls only 16-bit windows.

Calculation of Addresses for all Memory Windows - Add the offset shown in Table 5-14 to the Index Base for the desired slot and window listed.

Table 5-13 Memory Mapping Window Register Group - Offset +0h through +5h

7	6	5	4	3	2	1	0
Offset +0h	**Memory Window Start Address Register - Byte 0: Address Bits MS[19:12]**						**Default = 00h**
Offset +1h	**Memory Window Start Address Register - Byte 1**						**Default = 00h**
Data path: 0 = 8 bits 1 = 16 bits	Zero wait states: 0 = No 1 = Yes	Reserved: Write bits as read.		Memory Mapping Window Start Address bits MS[23:20]			
MEMOFST (see Table 5-14)	**Memory Window Start Address Register - Byte 2: Address Bits MS[31:24]**						**Default = 00h**
Offset +2h	**Memory Window Stop Address Register - Byte 0: Address Bits MST[19:12]**						**Default = 00h**
Offset +3h	**Memory Window Stop Address Register - Byte 1**						**Default = 00h**
Command Length (ATCLKs): 00 = Two 10 = Four 01 = Three 11 = Five		Reserved: Write bit as read.		Memory Mapping Window Stop Address bits MST[23:20]			
Offset +4h	**Memory Window Offset Address Register - Byte 0: Address Bits MOFST[19:12]**						**Default = 00h**
Offset +5h	**Memory Window Offset Address Register - Byte 1**						**Default = 00h**
Window write control: 0 = Enable 1 = Disable	Memory access: 0 = Common 1 = Attribute	Memory Mapping Window Offset Address bits MOFST[25:20]					

Table 5-14 Index Base Addresses for Memory Windows

Index Base For:	Start Address			Stop Address		Offset Address	
	Byte 0	Byte 1	Byte 2	Byte 0	Byte 1	Byte 0	Byte 1
Memory Window 0#							
Slot A	10h	11h	--	12h	13h	14h	15h
Slot B	50h	51h	--	52h	53h	54h	55h
Slot C	90h	91h	--	92h	93h	94h	95h
Slot D	D0h	D1h	--	D2h	D3h	D4h	D5h
MEMOFST	810h	811h	840h	812h	813h	814h	815h
Memory Window 1#							
Slot A	18h	19h	--	1Ah	1Bh	1Ch	1Dh
Slot B	58h	59h	--	5Ah	5Bh	5Ch	5Dh
Slot C	98h	99h	--	9Ah	9Bh	9Ch	9Dh
Slot D	D8h	D9h	--	DAh	DBh	DCh	DDh
MEMOFST	818h	819h	841h	81Ah	81Bh	81Ch	81Dh
Memory Window 2#							
Slot A	20h	21h	--	22h	23h	24h	25h
Slot B	60h	61h	--	62h	63h	64h	65h
Slot C	A0h	A1h	--	A2h	A3h	A4h	A5h
Slot D	E0h	E1h	--	E2h	E3h	E4h	E5h
MEMOFST	820h	821h	842h	822h	823h	824h	825h
Memory Window 3#							
Slot A	28h	29h	--	2Ah	2Bh	2Ch	2Dh
Slot B	68h	69h	--	6Ah	6Bh	6Ch	6Dh
Slot C	A8h	A9h	--	AAh	ABh	ACh	ADh
Slot D	E8h	E9h	--	EAh	EBh	ECh	EDh
MEMOFST	828h	829h	843h	82Ah	82Bh	82Ch	82Dh
Memory Window 4#							
Slot A	30h	31h	--	32h	33h	34h	35h
Slot B	70h	71h	--	72h	73h	74h	75h
Slot C	B0h	B1h	--	B2h	B3h	B4h	B5h
Slot D	F0h	F1h	--	F2h	F3h	F4h	F5h
MEMOFST	830h	831h	844h	832h	833h	834h	835h

5.3.4 Special PCMCIA Register Group

The 82C824 PCMCIA logic provides compatibility with the Intel 82365SL PCMCIA chipset. In addition, certain Cirrus 6722 PCMCIA chipset features are implemented. Since there are register conflicts between these two devices in certain locations, the 82C824 PCMCIA logic implements the register features as noted below.

5.3.4.1 DMA on the PCMCIA Interface

DMA operations are described with respect to system memory access. During a DMA write, data is transferred from a PC Card to system memory. During a DMA read, data is transferred from system memory to a PC Card. Address lines to the PC Card are ignored during DMA operations. DMA signals are defined as follows for the PCMCIA interface.

DREQ# - The DMA Request signal DREQ# is only available when a PC Card and socket are configured for DMA operations. Note that DREQ# is active low, opposite to the traditional ISA bus sense of the signal. A PC Card asserts DREQ# to indicate to the host that it is requesting service. The PC Card asserts DREQ# until the host responds by asserting DACK. A PC Card may use any one of the following three pins for DREQ#: SPKR#, INPACK# or IOIS16#. The PC Card indicates the pin used for DREQ# in the Miscellaneous Features Field of the card configuration header (CIS).

DACK - A DMA transfer is indicated when DACK is active along with either IOR# or IOW#. Note that DACK is active high, opposite to the traditional ISA bus sense of the signal. The 82C824 uses the card REG# pin to indicate a DMA operation. The card must be programmed for an I/O interface before the DMA interface can be enabled. The DACK(REG#) signal is then used to distinguish between a DMA cycle and a normal I/O cycle. For a normal I/O cycle, REG# is held low for the complete bus cycle. For a DMA transfer, REG# is held high during the entire DMA bus cycle.

TC - The 82C824 signals terminal count for DMA read operations by asserting WE# along with IOW#, and for DMA write operations by asserting OE# along with IOR#.

5.3.4.2 DMA Control Register

The DMA Control Register uses a similar format to that available in the Cirrus 6722 register set at offset 3Fh for its upper 3 bits; however, bits [4:0] are different. Bits [2:0] select the DMA channel because, unlike the Cirrus controller, the 82C824 controller generates all DMA channel requests directly (it does not depend on the host to redirect the DREQ/DACK lines). Bits [4:3] allow enabling of built-in pull-up resistors that are not available on the Cirrus part.

DREQ# Select - These bits select the pin that will be used to provide the DREQ# signal to the PCMCIA card. Most PCMCIA cards will be able to sacrifice INPACK# for the DREQ# function; the IOIS16# and SPKR# pins are offered as alternatives. The “No DMA function” disables the DMA feature altogether and eliminates the need for the bit 1Eh[6] used by the Cirrus 6722 to enable DMA operation.

CD1-2# and VS1-2 Pull-ups - The PCMCIA card detect (CD1-2#) and voltage sense (VS1-2) lines are normally pulled up internal to the 82C824 chip to avoid the need for external resistors. The control bit is provided to disable these resistors during power-down situations.

DMA Channel - These bits indicate the system DMA channel to which the DREQ will be directed. DRQ0-3 are 8-bit channels; DRQ5-7 are 16-bit channels. These bits are **not** present in the Cirrus 6722 part.

Table 5-15 DMA Control Register

7	6	5	4	3	2	1	0
Index 3Fh, 7Fh, BFh, FFh / MEMOFST 83Fh			**DMA Control Register**				**Default = 00h**
DREQ# select: 00 = No DMA function 01 = Use INPACK# 10 = Use IOIS16# 11 = Use SPKR#		Reserved(1)	BVD1-2, RDY/BSY# pull-ups: 0 = Disable (default) 1 = Enable	INPACK#, WAIT# pull-ups: 0 = Disable (default) 1 = Enable	DMA channel: 000 = DRQ0 011 = DRQ3 110 = DRQ6	 001 = DRQ1 100 = Reserved 111 = DRQ7	 010 = DRQ2 101 = DRQ5

(1) The CD1-2# and VS1-2 pull-ups are controlled globally through PCICFG 51h[4].

5.3.4.3 ATA Interface

The ATA Control Register is provided to allow a minor redefinition of the interface to accommodate ATA interface devices, normally IDE types of devices such as disk drives and Flash EEPROM cards. This register is not strictly compatible with the register at offset 26h in the Cirrus 6722 register set.

Interface Mode - Selecting ATA mode changes operation as follows:

1) Bits 3Eh[7:3] are enabled to manually control address bits A[25:21] to the card.
2) CE1# takes on the IDE function of CS1# which goes low when address bit A9 is low (address range 1F0-1F7h or 170-177h).
3) CE2# takes on the IDE function of CS3# which goes low when address bit A9 is high (address range 3F6-3F7h or 376-377h).

For proper operation of a card with this type of interface, it is also necessary to program the I/O Mapping Windows to the 1F0-1F7h (or 170-177h) range and to the 3F6-3F7h (or 376-377h) range.

5.3.4.4 Control Registers

The PCMCIA slot interface implements the VS1 and VS2 signals. The new PCMCIA specification allows VS1-2 to be used in determining whether a card can be powered up at 5.0V or not according to Table 5-17. This information pertains to the Miscellaneous Control Register at offset 16h, described in 5.3.4.5.

Table 5-16 ATA Control Register

7	6	5	4	3	2	1	0
Index 3Eh, 7Eh, BEh, FEh / MEMOFST 83Eh			**ATA Control Register**				**Default = 00h**
A25 (CSEL control)	A24 (M/S# control)	A23 (VU control)	A22 (Misc. control)	A21 (Misc. control)	Card RESET polarity: 0 = Normal 1 = Inverted	Card IREQ# polarity: 0 = Normal 1 = Inverted	Interface mode: 0 = PCMCIA 1 = ATA

Table 5-17 VS1-2 Status Indication for PCMCIA Cards.

VS2	VS1	Key on PC Card	PCMCIA Card Type Indicated
Open	Open	5.0V	5.0V R2 card
Open	Ground	Low Voltage	3.3V R2 card
Open	Ground	5.0V	3.3V or 5.0V R2 card
Ground	Open	Low Voltage	Low Voltage R2 card
Ground	Ground	Low Voltage	Low Voltage or 3.3V R2 card
Ground	Ground	5.0V	Low Voltage, 3.3V, or 5.0V R2 card

5.3.4.5 Miscellaneous Control Register

At offset 16h, the Intel 82365SL implements the Card Detect and General Control Register, while the Cirrus 6722 part implements Miscellaneous Control Register 1. The PCMCIA controller register at this offset incorporates bits from both of these registers and is therefore **not** strictly compatible with either.

TC Timing - Bit 16h[6] is provided to control the duration of Terminal Count (TC) to the PCMCIA DMA card. While the PCMCIA specification requires that TC be taken away before command, the Cirrus data book shows TC asserted even after the command edge. DMA cards designed to latch TC on the rising edge of command must set 16h[6] = 1.

I/O Command and Address Setup Time - Bits 16h[7] and 16h[3:2] allow performance that is faster than the PCMCIA specification requires for many cards. These bits should be set only for cards that are capable of using this enhanced timing.

5.3.4.6 Global Control Register

Only one bit of this Intel 82365SL register is implemented. The other bits correspond to IRQ manipulation that is unnecessary in the 82C824 chip.

Reset Change Status - Bit 1Eh[2] selects the mode used to clear status change events in the Card Status Change Register at offset 04h. In its default setting of 1Eh[2] = 0, the status change events are all cleared at once every time the register at offset 04h is read. If 1Eh[2] = 1, reading the register at offset 04h does not clear any events. To clear each event, software must write a 1 to the bit position at offset 04h that indicated status change event. Effectively, writing back the same value read will clear the status change event.

Table 5-18 Miscellaneous and Global Control Registers

7	6	5	4	3	2	1	0
Index 16h, 56h, 96h, D6h / MEMOFST 816h			**Miscellaneous Control Register**				**Default = 00h**
I/O command: 0 = Based on standard ISA timing 1 = Short (6 PCICLKs)	TC timing: 0 = PCMCIA standard 1 = Stays active past end of cmd.	Reset Cfg. Registers if CD1-2# go high (RO): 1 = Yes (always)	SPKROUT: 0 = Disable 1 = Driven if PCICFG 50h[1] = 1	Memory cycle address setup time: 00 = 3 PCICLKs 01 = 2 PCICLKs 10 = 1 PCICLK 11 = None		VS1 status (RO): 0 = Low 1 = High	VS2 status (RO): 0 = 3.3V 1 = 5.0V
Index 1Eh, 5Eh, 9Eh, DEh / MEMOFST 81Eh			**Global Control Register**				**Default = 00h**
Reserved: Write as read.		Zoomed video port (reassigns A[25:4], IOIS16#, INPACK#, SPKR#): 0 = Disable 1 = Enable	Reserved: Write as read.		Reset change status: 0 = On status change reg read 1 = On write to bit	Reserved: Write as read.	

5.4 Register Summary

Table 5-19 summarizes the locations, register names, and default values for the PCI Bridge 0 and 1 Register Groups. Note that the table lists only the PCICFG location, the CardBus Control Group can also be accessed in system memory space. Refer to Section 5.2.3 for details regarding accessing those memory locations.

Table 5-19 PCI Bridge 0 and 1 Register Groups Summary

Loc.	Register Name	Default
PCICFG 00h-4Fh: Base Register Group		
	Vendor Identification Register (RO)	
00h	Byte 0	45h
01h	Byte 1	10h
	Device ID (RO)	
02h	Byte 0	24h
03h	Byte 1	C8h
	PCI Command Register	
04h	Byte 0	04h
05h	Byte 1	00h
	PCI Status Register	
06h	Byte 0	00h
07h	Byte 1	02h
08h	Revision Register (RO)	10h
09h	Prgrm Interface Class Code Register (RO)	00h
	Class Code Register (RO)	
0Ah	Byte 0	07h
0Bh	Byte 1	06h
0Ch	Cache Line Size Register	00h
0Dh	Latency Timer Register	00h
0Eh	Header Type Register	82h
0Fh	BIST Register	00h
	CardBus Base Address Register	
10h	Byte 0: Address Bits [7:0]	00h
11h	Byte 1: Address Bits [15:8]	00h
12h	Byte 2: Address Bits [23:16]	00h
13h	Byte 3: Address Bits [31:24]	00h
14h-15h	Reserved	00h
	PCI Secondary Bus Status Register	
16h	Byte 0	00h
17h	Byte 1	02h
18h	Primary PCI Bus Number Register	00h
19h	CardBus Number Register	00h
1Ah	Subordinate Bus Number Register	00h
1Bh	CardBus Latency Timer Register	00h
	Memory Window 0 Base Address Register	
1Ch	Byte 0: Address Bits [7:0]	00h
1Dh	Byte 1: Address Bits [15:8]	F0h
1Eh	Byte 2: Address Bits [23:16]	FFh
1Fh	Byte 3: Address Bits [31:24]	FFh

Loc.	Register Name	Default
	Memory Window 0 Limit Address Register	
20h	Byte 0: Address Bits [7:0	00h
21h	Byte 1: Address Bits [15:8]	00h
22h	Byte 2: Address Bits [23:16]	00h
23h	Byte 3: Address Bits [31:24]	00h
	Memory Window 1 Base Address Register	
24h	Byte 0: Address Bits [7:0]	00h
25h	Byte 1: Address Bits [15:8]	F0h
26h	Byte 2: Address Bits [23:16]	FFh
27h	Byte 3: Address Bits [31:24]	FFh
	Memory Window 1 Limit Address Register	
28h	Byte 0: Address Bits [7:0]	00h
29h	Byte 1: Address Bits [15:8]	00h
2Ah	Byte 2: Address Bits [23:16]	00h
2Bh	Byte 3: Address Bits [31:24]	00h
	I/O Window 0 Base Address Register	
2Ch	Byte 0: Address Bits [7:0]	00h
2Dh	Byte 1: Address Bits [15:8]	F0h
2Eh	Byte 2: Address Bits [23:16]	FFh
2Fh	Byte 3: Address Bits [31:24]	FFh
	I/O Window 0 Limit Address Register	
30h	Byte 0: Address Bits [7:0]	00h
31h	Byte 1: Address Bits [15:8]	00h
32h	Byte 2: Address Bits [23:16]	00h
33h	Byte 3: Address Bits [31:24]	00h
	I/O Window 1 Base Address Register	
34h	Byte 0: Address Bits [7:0]	00h
35h	Byte 1: Address Bits [15:8]	F0h
36h	Byte 2: Address Bits [23:16]	FFh
37h	Byte 3: Address Bits [31:24]	FFh
	I/O Window 1 Limit Address Register	
38h	Byte 0: Address Bits [7:0]	00h
39h	Byte 1: Address Bits [15:8]	00h
3Ah	Byte 2: Address Bits [23:16]	00h
3Bh	Byte 3: Address Bits [31:24]	00h
3Ch	CINT# Interrupt Line Register for Status Change	00h
3Dh	CSTSCHG Interrupt Pin Register for Status Change (RO)	01h
	Bridge Control Register	
3Eh	Byte 0	40h
3Fh	Byte 1	03h
	Subsystem Vendor Register	
40h	Byte 0: Bits [7:0]	00h
41h	Byte 1: Bits [15:8]	00h

Table 5-19 PCI Bridge 0 and 1 Register Groups Summary (cont.)

Loc.	Register Name	Default
	Subsystem ID Register	
42h	Byte 0: Bits [7:0]	00h
43h	Byte 1: Bits [15:8]	00h
	16-Bit PC Card Legacy Mode Addr. Register	
44h	Byte 0: Bits [7:0]	01h
45h	Byte 1: Address Bits [15:8]	00h
46h	Byte 2: Address Bits [23:16]	00h
47h	Byte 3: [31:24]	00h
48h	Docking INTA# Interrupt Assign. (Slot A only)	01h
49h	Docking INTB# Interrupt Assign. (Slot A only)	02h
4Ah	Docking INTC# Interrupt Assign. (Slot A only)	03h
4Bh	Reserved	00h
4Ch	Docking Detect Interrupt Assignment Register	01h
4Dh-4Fh	Reserved	00h
PCICFG 50h-5Fh: Specific Register Group		
50h	PCI Host Feature Control Register	01h
51h	Slot Feature Control Register 1	00h
52h	Slot Feature Control Register 2	0Fh
53h	PCMCIA Controller Configuration Register	03h
	IRQ Driveback Address Register	
54h	Byte 0: Address Bits [7:0]	30h
55h	Byte 1: Address Bits [15:8]	33h
56h	Byte 2: Address Bits [23:16]	33h
57h	Byte 3: Address Bits [31:24]	33h
	DRQ Remap Base Address Register	
58h	Byte 0: Address Bits [7:0]	00h
59h	Byte 1: Address Bits [15:8]	00h
5Ah	Byte 2: Address Bits [23:16]	00h
5Bh	Byte 3: Address Bits [31:24]	00h
5Ch	DMA Channel Selector Register	00h
5Dh	SMI Status Register (Slot A only)	00h
5Eh	Primary Retry Limit Register	07h
5Fh	Retry Count Readback Register	00h
PCICFG 60h-74h: CardBus Register Group		
	CardBus Socket Event Register	
60h	Byte 0	00h
61h	Byte 1	00h
62h	Byte 2	00h
63h	Byte 3	00h
	CardBus Socket Mask Register	
64h	Byte 0	00h
65h	Byte 1	00h
66h	Byte 2	00h
67h	Byte 3	00h

Loc.	Register Name	Default
	CardBus Socket Present State Register	
68h	Byte 0 (RO)	00h
69h	Byte 1	00h
6Ah	Byte 2	00h
6Bh	Byte 3	30h
	CardBus Force Event Register	
6Ch	Byte 0	00h
6Dh	Byte 1	00h
6Eh	Byte 2	00h
6Fh	Byte 3	00h
	CardBus Control Register	
70h	Byte 0	00h
71h	Byte 1	00h
72h	Byte 2	00h
73h	Byte 3	00h
74h	Reserved	00h
PCICFG 80h-FFh: Docking Station Window Registers		
	Window 0 Start Address Register	
80h	Byte 0: Address Bits [7:0]	00h
81h	Byte 1: Address Bits [15:8]	F0h
82h	Byte 2: Address Bits [23:16]	FFh
83h	Byte 3: Address Bits [31:24]	FFh
	Window 0 Stop Address Register	
84h	Byte 0: Address Bits [7:0]	00h
85h	Byte 1: Address Bits [15:8]	00h
86h	Byte 2: Address bits [23:16]	00h
87h	Byte 3: Address Bits [31:24]	00h
	Window 0 Mask Register	
88h	Byte 0: Mask Bits [7:0]	FFh
89h	Byte 1: Mask Bits [15:8]	0Fh
8Ah	Byte 2: Mask Bits [23:16]	00h
8Bh	Window 0 Control Register	08h
8Ch-8Fh	Reserved	00h
	Window 1 Start Address Register	
90h	Byte 0: Address Bits [7:0]	00h
91h	Byte 1: Address Bits [15:8]	F0h
92h	Byte 2: Address Bits [23:16]	FFh
93h	Byte 3: Address Bits [31:24]	FFh
	Window 1 Stop Address Register	
94h	Byte 0: Address Bits [7:0]	00h
95h	Byte 1: Address Bits [15:8]	00h
96h	Byte 2: Address bits [23:16]	00h
97h	Byte 3: Address Bits [31:24]	00h
	Window 1 Mask Register	
98h	Byte 0: Mask Bits [7:0]	FFh
99h	Byte 1: Mask Bits [15:8]	0Fh
9Ah	Byte 2: Mask Bits [23:16]	00h
9Bh	Window 1 Control Register	08h
9Ch-9Fh	Reserved	00h

Table 5-19 PCI Bridge 0 and 1 Register Groups Summary (cont.)

Loc.	Register Name	Default
	Window 2 Start Address Register	
A0h	Byte 0: Address Bits [7:0]	FCh
A1h	Byte 1: Address Bits [15:8]	FFh
A2h	Byte 2: Address Bits [23:16]	00h
A3h	Byte 3: Address Bits [31:24]	00h
	Window 2 Stop Address Register	
A4h	Byte 0: Address Bits [7:0]	00h
A5h	Byte 1: Address Bits [15:8]	00h
A6h	Byte 2: Address Bits [23:16]	00h
A7h	Byte 3: Address Bits [31:24]	00h
	Window 2 Mask Register	
A8h	Byte 0: Mask Bits [7:0]	03h
A9h	Byte 1: Mask Bits [15:8]	00h
AAh	Byte 2: Mask Bits [23:16]	00h
ABh	Window 2 Control Register	00h
ACh-AFh	Reserved	00h
	Window 3 Start Address Register	
B0h	Byte 0: Address Bits [7:0]	FCh
B1h	Byte 1: Address Bits [15:8]	FFh
B2h	Byte 2: Address Bits [23:16]	00h
B3h	Byte 3: Address Bits [31:24]	00h
	Window 3 Stop Address Register	
B4h	Byte 0: Address Bits [7:0]	00h
B5h	Byte 1: Address Bits [15:8]	00h
B6h	Byte 2: Address Bits [23:16]	00h
B7h	Byte 3: Address Bits [31:24]	00h
B8h	Window 3 Mask Register	00h
B9h	Byte 0: Mask Bits [7:0]	00h
BAh	Byte 1: Mask Bits [15:8]	00h
	Byte 2: Mask Bits [23:16]	
BBh	Window 3 Control Register	00h
BCh-BFh	Reserved	00h
	Window 4 Start Address Register	
C0h	Byte 0: Address Bits [7:0]	00h
C1h	Byte 1: Address Bits [15:8]	F0h
C2h	Byte 2: Address Bits [23:16]	FFh
C3h	Byte 3: Address Bits [31:24]	FFh
	Window 4 Stop Address Register	
C4h	Byte 0: Address Bits [7:0]	00h
C5h	Byte 1: Address Bits [15:8]	00h
C6h	Byte 2: Address Bits [23:16]	00h
C7h	Byte 3: Address Bits [31:24]	00h
	Window 4 Mask Register	
C8h	Byte 0: Mask Bits [7:0]	00h
C9h	Byte 1: Mask Bits [15:8]	00h
CAh	Byte 2: Mask Bits [23:16]	00h
CBh	Window 4 Control Register	48h
CCh-CFh	Reserved	00h

Loc.	Register Name	Default
	Window 5 Start Address Register	
D0h	Byte 0: Address Bits [7:0]	00h
D1h	Byte 1: Address Bits [15:8]	F0h
D2h	Byte 2: Address Bits [23:16]	FFh
D3h	Byte 3: Address Bits [31:24]	FFh
	Window 5 Stop Address Register	
D4h	Byte 0: Address Bits [7:0]	00h
D5h	Byte 1: Address Bits [15:8]	00h
D6h	Byte 2: Address Bits [23:16]	00h
D7h	Byte 3: Address Bits [31:24]	00h
	Window 5 Mask Register	
D8h	Byte 0: Mask Bits [7:0]	00h
D9h	Byte 1: Mask Bits [15:8]	00h
DAh	Byte 2: Mask Bits [23:16]	00h
DBh	Window 5 Control Register	48h
DCh-DFh	Reserved	00h
	Window 6 Start Address Register	
E0h	Byte 0: Address Bits [7:0]	00h
E1h	Byte 1: Address Bits [15:8]	F0h
E2h	Byte 2: Address Bits [23:16]	FFh
E3h	Byte 3: Address Bits [31:24]	FFh
	Window 6 Stop Address Register	
E4h	Byte 0: Address Bits [7:0]	00h
E5h	Byte 1: Address Bits [15:8]	00h
E6h	Byte 2: Address Bits [23:16]	00h
E7h	Byte 3: Address bits [31:24]	00h
	Window 6 Mask Register	
E8h	Byte 0: Mask Bits [7:0]	03h
E9h	Byte 1: Mask Bits [15:8]	00h
EAh	Byte 2: Mask Bits [23:16]	00h
EBh	Window 6 Control Register	00h
ECh-EFh	Reserved	00h
	Window 7 Start Address Register	
F0h	Byte 0: Address Bits [7:0]	00h
F1h	Byte 1: Address Bits [15:8]	F0h
F2h	Byte 2: Address Bits [23:16]	FFh
F3h	Byte 3: Address Bits [31:24]	FFh
	Window 7 Stop Address Register	
F4h	Byte 0: Address Bits [7:0]	00h
F5h	Byte 1: Address Bits [15:8]	00h
F6h	Byte 2: Address Bits [23:16]	00h
F7h	Byte 3: Address Bits [31:24]	00h
	Window 7 Mask Register	
F8h	Byte 0: Mask Bits [7:0]	03h
F9h	Byte 1: Mask Bits [15:8]	00h
FAh	Byte 2: Mask Bits [23:16]	00h
FBh	Window 7 Control Register	00h
FCh-FFh	Reserved	00h

Table 5-19 summarizes the locations, register names, and default values for the PCMCIA Controller 0 and 1 Register Groups. Note that the table lists only the index location, the General Purpose Register Group can also be accessed in system memory space. Refer to Section 5.3 for details regarding accessing those memory locations.

Table 5-20 PCMCIA Controller 0 and 1 Register Groups Summary

Index Location					
Slot A	**Slot B**	**Slot C**	**Slot D**	**Register Name**	**Default**
PCMCIA General Purpose Register Group					
00h	40h	80h	C0h	Identification Register (RO)	87h
01h	41h	81h	C1h	Interface Status Register (RO)	00h
02h	42h	82h	C2h	Power Control Register	00h
03h	43h	83h	C3h	Reset and General Control Register	00h
04h	44h	84h	C4h	Card Status Change Register	00h
05h	45h	85h	C5h	STSCHG# Interrupt Configuration Register	00h
06h	46h	86h	C6h	Address Window Enable Register	00h
I/O Mapping Window Register Group					
07h	47h	87h	C7h	I/O Window Control Register	00h
				I/O Window 0 Start Address Register	
08h	48h	88h	C8h	Byte 0: Address Bits IOS[7:0]	00h
09h	49h	89h	C9h	Byte 1: Address Bits IOS[15:8]	00h
				I/O Window 0 Stop Address Register	
0Ah	4Ah	8Ah	CAh	Byte 0: Address Bits IOST[7:0]	00h
0Bh	4Bh	8Bh	CBh	Byte 1: Address Bits IOST[15:8]	00h
				I/O Window 1 Start Address Register	
0Ch	4Ch	8Ch	CCh	Byte 0: Address Bits IOS[7:0]	00h
0Dh	4Dh	8Dh	CDh	Byte 1: Address Bits IOS[15:8]	00h
				I/O Window 1 Stop Address Register	
0Eh	4Eh	8Eh	CEh	Byte 0: Address Bits IOST[7:0]	00h
0Fh	4Fh	8Fh	CFh	Byte 1: Address Bits IOST[15:8]	00h
Memory Mapping Window Register Group					
				Memory Window 0 Start Address Register	
10h	50h	90h	D0h	Byte 0: Address Bits MS[19:12]	00h
11	51h	91h	D1h	Byte 1: Address Bits MS[23:20]	00h
				Memory Window 0 Stop Address Register	
12h	52h	92h	D2h	Byte 0: Address Bits MST[19:12]	00h
13h	53h	93h	D3h	Byte 1: Address Bits MST[23:20]	00h
				Memory Window 0 Offset Address Register	
14h	54h	94h	D4h	Byte 0: Address Bits MOFST[19:12]	00h
15h	55h	95h	D5h	Byte 1: Address Bits MOFST[25:20]	00h
				Memory Window 1 Start Address Register	
18h	58h	98h	D8h	Byte 0: Address Bits MS[19:12]	00h
19h	59h	99h	D9h	Byte 1: Address Bits MS[23:20]	00h
				Memory Window 1 Stop Address Register	
1Ah	5Ah	9Ah	DAh	Byte 0: Address Bits MST[19:12]	00h
1Bh	5Bh	9Bh	DBh	Byte 1: Address Bits MST[23:20]	00h
				Memory Window 1 Offset Address Register	
1Ch	5Ch	9Ch	DCh	Byte 0: Address Bits MOFST[19:12]	00h
1Dh	5Dh	9Dh	DDh	Byte 1: Address Bits MOFST[25:20]	00h
				Memory Window 2 Start Address Register	
20h	60h	A0h	E0h	Byte 0: Address Bits MS[19:12]	00h
21h	61h	A1h	E1h	Byte 1: Address Bits MS[23:20]	00h

Table 5-20 PCMCIA Controller 0 and 1 Register Groups Summary (cont.)

Index Location				Register Name	Default
Slot A	**Slot B**	**Slot C**	**Slot D**		
				Memory Window 2 Stop Address Register	
22h	62h	A2h	E2h	Byte 0: Address Bits MST[19:12]	00h
23h	63h	A3h	E3h	Byte 1: Address Bits MST[23:20]	00h
				Memory Window 2 Offset Address Register	
24h	64h	A4h	E4h	Byte 0: Address Bits MOFST[19:12]	00h
25h	65h	A5h	E5h	Byte 1: Address Bits MOFST[25:20]	00h
				Memory Window 3 Start Address Register	
28h	68h	A8h	E8h	Byte 0: Address Bits MS[19:12]	00h
29h	69h	A9h	E9h	Byte 1: Address Bits MS[23:20]	00h
				Memory Window 3 Stop Address Register	
2Ah	6Ah	AAh	EAh	Byte 0: Address Bits MST[19:12]	00h
2Bh	6Bh	ABh	EBh	Byte 1: Address Bits MST[23:20]	00h
				Memory Window 3 Offset Address Register	
2Ch	6Ch	ACh	ECh	Byte 0: Address Bits MOFST[19:12]	00h
2Dh	6Dh	ADh	EDh	Byte 1: Address Bits MOFST[25:20]	00h
				Memory Window 4 Start Address Register	
30h	70h	B0h	F0h	Byte 0: Address Bits MS[19:12]	00h
31h	71h	B1h	F1h	Byte 1: Address Bits MS[23:20]	00h
				Memory Window 4 Stop Address Register	
32h	72h	B2h	F2h	Byte 0: Address Bits MST[19:12]	00h
33h	73h	B3h	F3h	Byte 1: Address Bits MST[23:20]	00h
				Memory Window 4 Offset Address Register	
34h	74h	B4h	F4h	Byte 0: Address Bits MOFST[19:12]	00h
35h	75h	B5h	F5h	Byte 1: Address Bits MOFST[25:20]	00h
PCMCIA Special Register Group					
16h	56h	96h	D6h	Miscellaneous Control Register	00h
1Eh	5Eh	9Eh	DEh	Global Control Register	00h
3Eh	7Eh	BEh	FEh	ATA Control Register	00h
3Fh	7Fh	BFh	FFh	DMA Control Register	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 3.3V Core 5.0V Core		 100 150	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[2:0]# setup time to PCICLK rising	12		ns	6-1
t204	CREQ[2:0]# hold time from PCICLK rising	0		ns	6-2
t205	CGNT[2:0]# valid delay from PCICLK rising	2	12	ns	6-3
t206	PCIRQ[2:0]# setup time to PCICLK rising	5		ns	6-1
t207	PCIRQ[2:0]# hold time from PCICLK rising	3		ns	6-2
t208	PCIRQ[2:0]# valid delay from PCICLK rising	2	16	ns	6-3

6.4 AC Timing Diagrams

Figure 6-1 Setup Timing Waveform

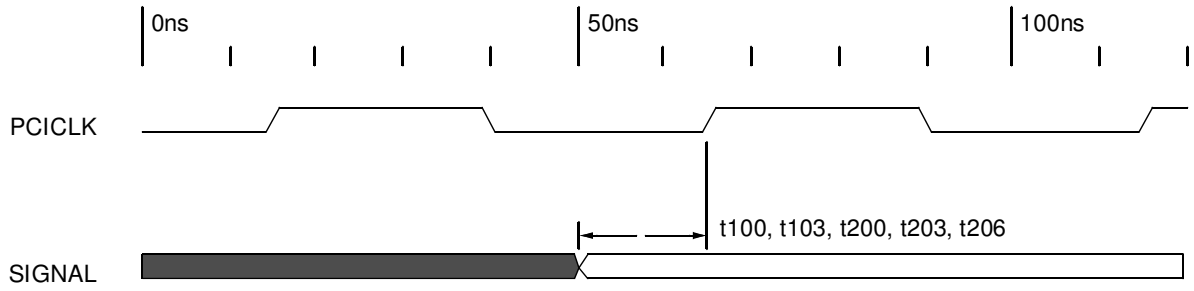


Figure 6-2 Hold Timing Waveform

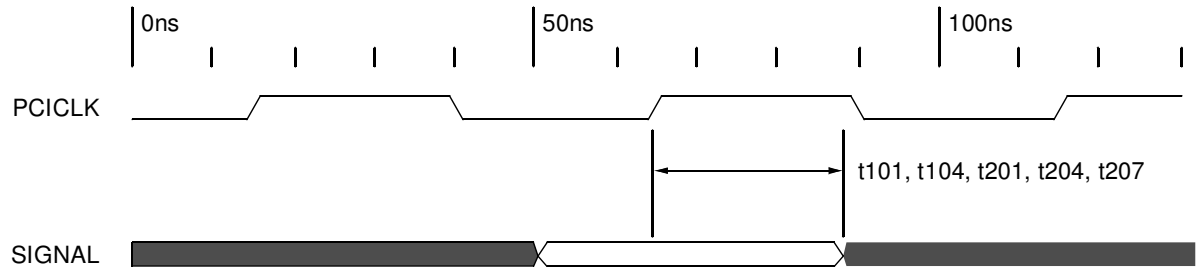
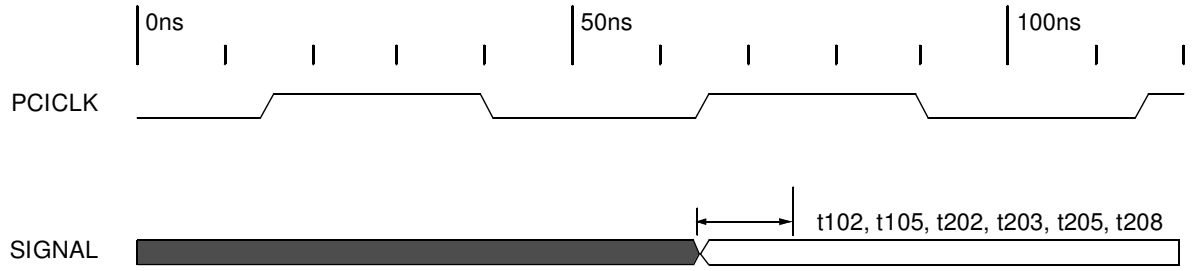
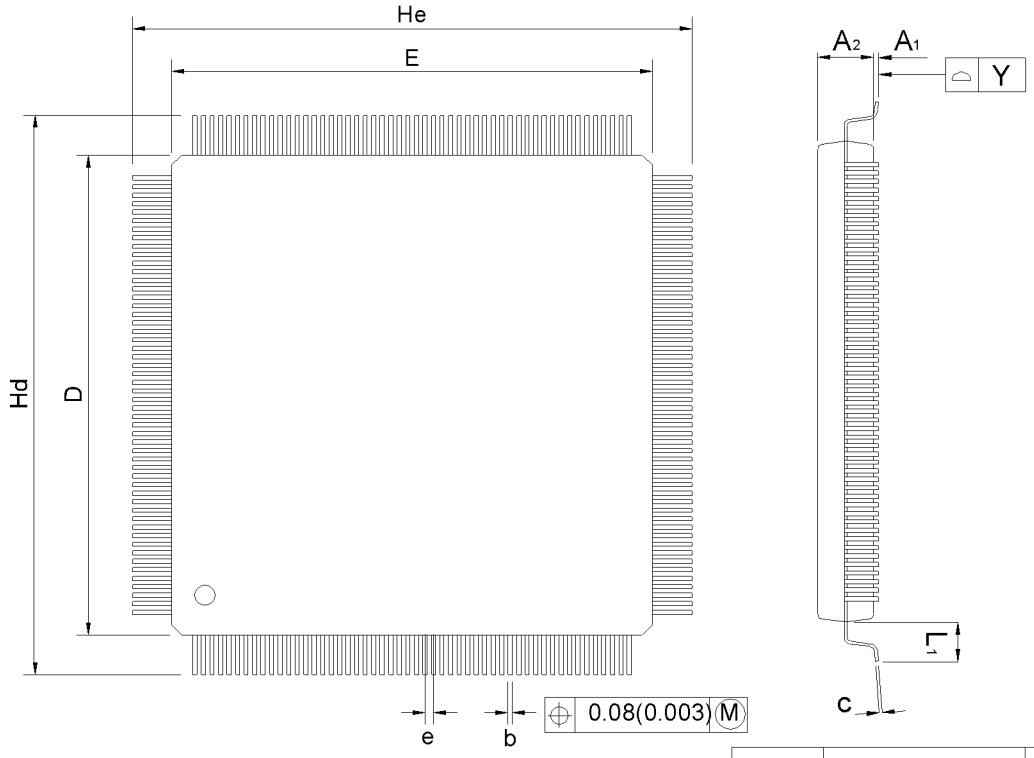


Figure 6-3 Output Delay Timing Waveform



7.0 Mechnical Package Outline

Figure 7-1 208-Pin Plastic Quad Flat Package (PQFP)



SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A_1	0.05	0.25	0.50	0.002	0.010	0.020
A_2	3.17	3.32	3.47	0.125	0.131	0.137
b	0.10	0.20	0.30	0.004	0.008	0.012
c	0.10	0.15	0.20	0.004	0.006	0.008
D	27.90	28.00	28.10	1.098	1.102	1.106
E	27.90	28.00	28.10	1.098	1.102	1.106
e		0.50			0.020	
Hd	30.35	30.60	30.85	1.195	1.205	1.215
He	30.35	30.60	30.85	1.195	1.205	1.215
L	0.50	0.60	0.75	0.020	0.024	0.030
L_1		1.30			0.051	
Y			0.08			0.003
θ	0		7	0		7

Dwg. No.:	AS208PQFP-001	
Dwg. Rev.: A1	Unit: MM / INCH	

Appendix A Distributed DMA Protocol

DMA on a PCI bus or across a PCI bridge is not currently handled by either the PCI or CardBus specifications. To fill this need, a DMA protocol has been developed. This protocol is being supported by industry leaders. The protocol provides a solid framework for compatible operation, but does not specify the exact method of implementation. Therefore, this document describes the generally agreed-to protocol and highlights its implementation in OPTi designs.

A.1 Introduction

The distributed DMA protocol allows PCI-based designs to incorporate multiple DMA controller (DMAC) channels distributed throughout the system, each of which is local to the device it will service. The PCI specification itself is not modified for DMA since only standard I/O and memory cycles are used in this scheme.

A specific protocol is needed for multiple DMA controllers on PCI. If each DMA channel had its own unique set of registers, there would be no problem; the device responsible for each channel would claim only its own accesses. Unfortunately, in the PC architecture some DMA registers are shared by groups of four channels; up to four separate devices would have to claim a single I/O read access, with disastrous results.

Therefore, the DMAC protocol specifies the means of:

- Claiming and routing I/O accesses to the correct owner of each channel
- Dividing up accesses that could be claimed by multiple devices
- Returning combined status information from multiple sources.

The means by which the distributed DMA protocol defines these responsibilities is described below.

A.2 Protocol Overview

The basic protocol simply defines new and unique I/O addresses for each register on every DMAC channel. The remapping puts all registers associated with a specific DMAC channel into a 10h byte area to make windowing requirements easier on PCI-to-PCI bridges.

When DMAC channels are present on a remote bus, the PCI controller sends DMA register I/O read and write cycles to the local PCI bus PCI-to-PCI bridges that connect the remote DMAC channels. PCI-to-PCI bridges need not be DMA-aware to pass these cycles, as long as they have an I/O mapping window programmed to claim the remapped accesses.

A.3 Distributed DMA Protocol Terminology

Devices on PCI that adhere to the distributed DMA protocol are referred to in this document using the phrases Master DMAC, DMA Channel Selector Register, Remote DMAC Channels, and DMA Remapper. These terms are described below.

A.3.1 Master DMAC

There must be one Master DMAC in the system. It is an OPTi standard 82C206-type DMAC subsystem with shadow register provisions. The Master DMAC:

- Becomes the claimer of cycles to DMAC channels that are not used by PCI peripheral devices or devices on the secondary side of PCI-to-PCI or PCI-to-ISA bridges.
- Provide all seven DMA channels: in the event that no other devices in the system support DMA, the Master DMAC must claim all cycles.
- Claims all accesses for DMA Channel 4.

The register groups for each channel in the table are assigned dynamically when the PCMCIA card is enabled for DMA through the PCMCIA register set. Only two channels are available at any one time, one for each PCMCIA card.

Table A-1 DMAC1/2 Control and Status Bits

7	6	5	4	3	2	1	0
Corresponding Port 008h/0D0h			**DMAC1/2 Status Register (RO)**				**Default = 00h**
Channel 3/7 request pending: 0 = No 1 = Yes	Channel 2/6 request pending: 0 = No 1 = Yes	Channel 1/5 request pending: 0 = No 1 = Yes	Channel 0/4 request pending: 0 = No 1 = Yes	Channel 3/7 reached terminal count: 0 = No 1 = Yes	Channel 2/6 reached terminal count: 0 = No 1 = Yes	Channel 1/5 reached terminal count: 0 = No 1 = Yes	Channel 0/4 reached terminal count: 0 = No 1 = Yes
Corresponding Port 00Bh/0D6h			**DMAC1/2 Mode Register (WO)**				**Default = 00h**
Mode select: 00 = Demand 01 = Single 10 = Block 11 = Cascade		Address count: 0 = Increment 1 = Decrement	Auto-initialize: 0 = Disable 1 = Enable	Transfer select: 00 = Reserved 01 = Memory Write 10 = Memory Read 11 = Reserved		Unused	
Corresponding Port 009h/0D2h			**DMAC1 DMA Request Register (WO)**				**Default = 00h**
Reserved: Write as 0.					Request: 0 = Clear 1 = Set	Unused	
Corresponding Port 008h/0D0h			**DMAC1/2 Command Register (WO)**				**Default = 00h**
Unused	DRQ active sense: 0 = High 1 = Low	Unused	Unused	Unused	DMAC operation: 0 = Enable 1 = Disable	Unused	Unused
Corresponding Port 00Fh/0DEh			**DMAC1/2 Mask Register**				**Default = 00h**
Reserved: Write as 0.				Unused	Unused	Unused	Channel: 0 = Unmasked 1 = Masked
Corresponding Port 00Ah/0DEh			**DMAC1/2 Set Single Mask Bit Register (WO)**				**Default = 00h**
Reserved. Write as 0.					Unused	Unused	
Corresponding Port 00Eh/0DEh			**DMAC1/2 Mask Clear Register (WO)**				**Default = 00h**
Writing any value clears all DMA channel mask bits at once.							
Corresponding Port 00Dh/0DAh			**DMAC1/2 Master Clear Register (WO)**				**Default = 00h**
Writing any value masks all DMA channels and resets all other DMAC values just like a hardware reset.							
Corresponding Port 00Ch/0D8h			**DMAC1/2 Clear Byte Pointer Flip-Flop (WO)**				**Default = 00h**
Writing any value resets the byte pointer flip-flop so that the next byte access to a word-wide DMA register is to the low byte.							

A.3.2 Remote DMAC Channels

Remote DMAC Channels can be anywhere in the system, even on the same PCI bus as the Master DMAC. Each remote DMAC channel must claim only the *remapped* cycles for which it is responsible. The only other difference between a remote DMAC channel and a channel on the master DMAC is that the master DMAC shadows writes to be able to respond to reads of shadowed information. Remote DMAC channels never respond to reads for write-only registers in the 8237 design.

A.3.3 DMA Channel Selector Register

Within the PCI Configuration Registers of PCI-based DMACs and DMA-aware PCI-to-PCI bridges are seven configuration bits to select whether each DMA channel is local or remote. For each device, the bits are programmed to select whether the DMAC claims that DMA channel or not. "Claimed" means that the channel is claimed by the device or that the device is claiming the cycle on behalf of another device downstream. For the scheme to work properly, each channel can be assigned "claimed" status in only one DMA Channel Selector Register; any channels that are unclaimed should be assigned to the Master DMAC.

The DMA Channel Selector Register layout is illustrated Table A-2.

DMAC Responsibility - This bit determines whether the concerned DMAC will be the system master. Only one master is possible in the system.

After Master and remote status has been properly assigned, the responsibility for claiming cycles can be defined as discussed next.

A.3.4 DMA Remapper

The address of each DMA controller port for each channel is normally listed as an absolute value in the AT-compatible I/O address space. The DMA remapper remaps these ports through a lookup table scheme. For the most part, the assignments are regular enough that a formula could be applied. Unfortunately, certain AT-compatible register locations (the Page Register in particular) introduce an irregularity in the remapping and require an inconsistent approach. The mapping is illustrated in Table A-1 using DMA channel 0 as an example.

From the CPU instruction set point of view, no change in addressing is required. All code can continue to issue the original AT-compatible port addresses. However, DMA programming code that is PCI-aware can directly address these ports if desired.

Note that only the EISA extensions to the Page Register and the Count Register are implemented. The remaining EISA extensions are not currently handled by this protocol.

Table A-2 DMA Channel Selector Registers

7	6	5	4	3	2	1	0
Channel 7 (DMAC2):	Channel 6 (DMAC2):	Channel 5 (DMAC2):	DMAC responsibility:	Channel 3 (DMAC1):	Channel 2 (DMAC1):	Channel 1 (DMAC1):	Channel 0 (DMAC1):
0 = Not claimed 1 = Claimed	0 = Not claimed 1 = Claimed	0 = Not claimed 1 = Claimed	0 = Secondary 1 = Master	0 = Not claimed 1 = Claimed	0 = Not claimed 1 = Claimed	0 = Not claimed 1 = Claimed	0 = Not claimed 1 = Claimed

Table A-3 DMA Remap Scheme - Generic for all DMA Channels

Register	Bits	Type	ISA I/O Address Example - Channel 0	Remapped Offset for PCI
Memory Address w/byte ptr low	A[7:0]	Read/Write	000h	b+(ch*10)+000h
Memory Address w/byte ptr high	A[15:8]	Read/Write	000h	b+(ch*10)+001h
Page Address	A[23:16]	Read/Write	087h	b+(ch*10)+002h
EISA High Byte Page Address	A[31:24]	Read/Write	487h	b+(ch*10)+003h
Count w/byte ptr low	C[7:0]	Read/Write	001h	b+(ch*10)+004h
Count w/byte ptr high	C[15:8]	Read/Write	001h	b+(ch*10)+005h
EISA High Byte Count	C[23:16]	Read/Write	401h	b+(ch*10)+006h
Reserved			007h	--
Status		Read-Only	008h	b+(ch*10)+008h
Command		Write-Only	008h	b+(ch*10)+008h
DMA Request		Write-Only	009h	b+(ch*10)+009h
Set Single Mask Bit		Write-Only	00Ah	b+(ch*10)+00Fh[0]
Mode		Write-Only	00Bh	b+(ch*10)+00Bh
Byte Pointer Flip-Flop Clear		Write-Only	00Ch	handled by DMA remapper
Master Clear		Write-Only	00Dh	b+(ch*10)+00Dh
Mask Clear		Write-Only	00Eh	b+(ch*10)+00Fh[0]
Mask		Read/Write	00Fh	b+(ch*10)+00Fh[0]

Notes:
'b' indicates base address
'ch' indicates channel number: ch=0 for channel 0, ch=1 for channel 1, ch=2 for channel 2, …, ch=7 for channel 7

Table A-4 Complete Remap Scheme, Channels 0-3

Register	Type	ISA I/O Port Address / PCI Remapped Address			
		DMA Ch 0	DMA Ch 1	DMA Ch 2	DMA Ch 3
Memory Address w/byte ptr low	Read/Write	000h/b+000h	002h/b+010h	004h/b+020h	006h/b+030h
Memory Address w/byte ptr high	Read/Write	000h/b+001h	002h/b+011h	004h/b+021h	006h/b+031h
Page Address	Read/Write	087h/b+002h	083h/b+012h	081h/b+022h	082h/b+032h
EISA High Byte Page Address	Read/Write	487h/b+003h	483h/b+013h	481h/b+023h	482h/b+033h
Count w/byte ptr low	Read/Write	001h/b+004h	003h/b+014h	005h/b+024h	007h/b+034h
Count w/byte ptr high	Read/Write	001h/b+005h	003h/b+015h	005h/b+025h	007h/b+035h
EISA High Byte Count	Read/Write	401h/b+006h	403h/b+016h	405h/b+026h	407h/b+036h
Status	Read-Only	008h/b+008h--b+018h--b+028h--b+038h (four reads)			
Command	Write-Only	008h/b+008h--b+018h--b+028h--b+038h (four writes)			
DMA Request	Write-Only	009h/b+009h	009h/b+019h	009h/b+029h	009h/b+039h
Set Single Mask Bit	Write-Only	00Ah/b+00Fh[0]	00Ah/b+01Fh[0]	00Ah/b+02Fh[0]	00Ah/b+03Fh[0]
Mode	Write-Only	00Bh/b+00Bh	00Bh/b+01Bh	00Bh/b+02Bh	00Bh/b+03Bh
Byte Pointer Flip-Flop Clear	Write-Only	00Ch/used by remapper, but no remapped I/O cycle generated			
Master Clear	Write-Only	00Dh/b+00Dh--b+01Dh--b+02Dh--b+03Dh (four writes)			
Mask Clear	Write-Only	00Eh/b+00Fh[0]--b+01Fh[0]--b+02Fh[0]--b+03Fh[0] (four writes)			
Mask	Read/Write	00Fh/b+00Fh[0]--b+01Fh[0]--b+02Fh[0]--b+03Fh[0] (four writes)			

Table A-5 Complete Remap Scheme, Channels 4-7

Register	Type	ISA I/O Port Address / PCI Remapped Address			
		DMA Ch 4	DMA Ch 5	DMA Ch 6	DMA Ch 7
Memory Address w/byte ptr low	Read/Write	0C0h/none	0C4h/b+050h	0C8h/b+060h	0CCh/b+070h
Memory Address w/byte ptr high	Read/Write	0C0h/none	0C4h/+051h	0C8h/b+061h	0CCh/b+071h
Page Address	Read/Write	08Fh/none	08Bh/b+052h	089h/b+062h	08Ah/b+072h
EISA High Byte Page Address	Read/Write	none/none	48Bh/b+053h	489h/b+063h	48Ah/b+073h
Count w/byte ptr low	Read/Write	0C2h/none	0C6h/b+054h	0CAh/b+064h	0CEh/b+074h
Count w/byte ptr high	Read/Write	0C2h/none	0C6h/b+055h	0CAh/b+065h	0CEh/b+075h
EISA High Byte Count	Read/Write	none/none	4C6h/b+056h	4CAh/b+066h	4CEh/b+076h
Status	Read-Only	0D0h/none	0D0h/b+058h--b+068h--b+078h (three reads)		
Command	Write-Only	0D0h/none	0D0h/b+058h--b+068h--b+078h (three writes)		
DMA Request	Write-Only	0D2h/none	0D2h/b+059h	0D2h/b+069h	0D2h/b+079h
Set Single Mask Bit	Write-Only	0D4h/none	0D4h/b+05Fh[0]	0D4h/b+06Fh[0]	0D4h/b+07Fh[0]
Mode	Write-Only	0D6h/none	0D6h/b+05Bh	0D6h/b+06Bh	0D6h/b+07Bh
Byte Pointer Flip-Flop Clear	Write-Only	0D8h/used by remapper, but no remapped I/O cycle generated			
Master Clear	Write-Only	0DAh/b+05Dh--b+06Dh--b+07Dh (three writes)			
Mask Clear	Write-Only	0DCh/none	0DCh/b+05Fh[0]--b+06Fh[0]--b+07Fh[0] (three writes)		
Mask	Read/Write	0DEh/b+05Fh[0]--b+06Fh[0]--b+07Fh[0] (three writes)			

A.3.4.1 Register Writes

Most, but not all, DMA I/O register writes are remapped by the DMA remapper. For all cases, the DMA remapper must generate STOP# in response to the original cycle until these remapped cycles are complete.

Mode and Request - For these write-only DMA registers, bits [1:0] indicate the channel number. Therefore, the DMA remapper need only generate a single I/O access, to the channel specified.

Command, Mask, and Master Clear - The DMA remapper remaps the access to four unique I/O locations (only three for DMAC2 accesses since DMA channel 4 is not important). Each device claims only its own access.

Single-Channel Mask and Mask Clear - These accesses simply update the Mask Register. Therefore, the DMA remapper must maintain a copy of the Mask Register internally so that it can update the mask. It then generates remapped writes to all Mask Registers.

Byte Pointer Flip-Flop Clear - The DMA remapper uses this value internally to determine the remapping for Address and Count accesses. However, it does not generate any external I/O cycles for this write.

All Other Registers - The DMA remapper remaps the I/O write according to the tables.

A.3.4.2 Register Reads

Only certain reads are remapped by the DMA remapper. Reads to other registers are reads of DMA shadow registers, which are not at industry-standard addresses and therefore are not covered by the distributed DMA protocol. Claiming DMAC register reads is straightforward. For all cases, the DMA remapper must generate STOP# in response to the original cycle until these remapped cycles are complete.

Address, Count, and Page Address Registers - All reads are remapped. The channel owner claims the remapped cycle and returns the data. PCI bridges must claim this cycle and pass it on to the secondary bus to return the data.

Mask Register - Reads are not remapped. The DMA remapper claims the cycle and returns shadowed information.

Status Register - Reads are remapped to four unique I/O locations. The DMA remapper combines the returned status information for each channel and provides it to the requester.

Write-only Registers - Reads are not remapped. The 82C206 core provides readback capability of these registers as shadowed information.

Note that there is no provision for conflicting claims by more than one device. As long as exactly one "claimed" assignment is made for each channel, there will never be a conflict.

Appendix B 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.

B.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 B-1 IRQ Driveback Cycle High-Priority Request

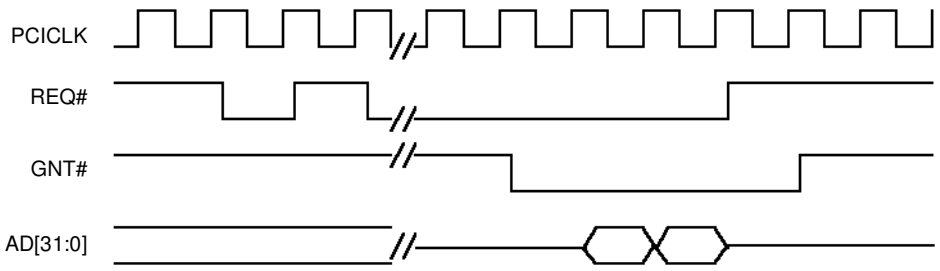


Table B-1 Information Provided on a Driveback Cycle

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 B-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 B-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#

B.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.

B.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.

B.4 External Implementation

An IRQ driveback-capable device can implement the signal IRQLATCH. IRQLATCH allows IRQs to be driven onto the ISA bus directly through external TTL. There are two possible support circuits.

Static Resourcing - Using a single 74373 latch provides direct control of up to eight IRQ lines. However, the selected IRQs are always under the control of the IRQ driveback device, even if the device is not actively using the IRQs. They cannot be dynamically reassigned to other devices. Figure B-3 shows a typical connection.

Dynamic Resourcing - Uses one 74373 latch and one 74125 tristate buffer to provide dynamic control over four specific IRQ lines; each four line group requires an additional 74373/74125 pair. Dynamic control allows the interrupt to be driven only when it has been assigned to a sub-function of the IRQ driveback device; otherwise, the output remains tristated and is open for use by other system devices. The figure below shows a typical connection.

Note that if the IRQLATCH function is selected on the primary, devices on the secondary are no longer free to generate any IRQ. They are limited to the IRQs supported through the latch.

Figure B-2 Dynamic Resourcing

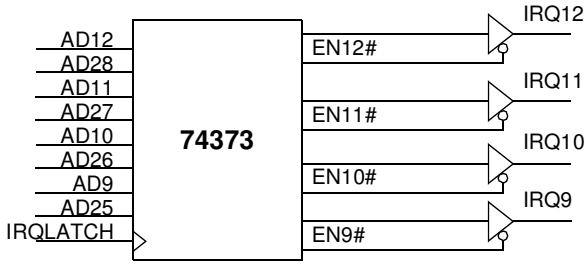
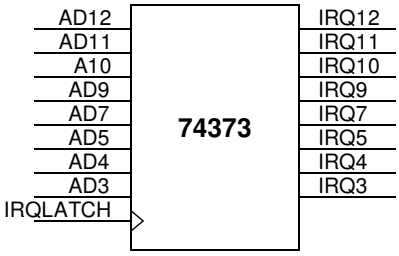


Figure B-3 Static Resourcing



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 Kingdom & Europe
OPTi Inc.
Bicester Business Center
Market Court, Victoria Road
Bicester, Oxon OX6 7QB
U.K.
tel: + 44-1-869-321-622
fax: + 44-1-869-241-448

United States
OPTi Inc.
8 Corporate Park, Ste. #300
Irvine, CA 92714-5117
tel: 714-838-0589
fax: 714-838-9753

OPTi Inc.
4400 N. Federal Highway, Ste. #120
Boca Raton, FL 33431
tel: 407-395-4555
fax: 407-395-4554

OPTi Inc.
20405 State Highway 249, Ste. #220
Houston, TX 77070
tel: 713-257-1856
fax: 713-257-1825

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

California - Southern
Jones & McGeoy Sales
5100 Campus Dr., Ste. #300
Newport Beach, CA 92660
tel: 714-724-8080
fax: 714-724-8090

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.
40 Mall Road, Suite 202
Burlington, MA 01803
tel: 617-272-5552
fax: 617-272-5515

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.
629 Aljunied Road #05-15
Cititech Industrial Building
Singapore 1438
tel: 65-741-7507
65-741-7530
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

(7/96)

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