82C825

ISA Docking Station Bridge

Preliminary Data Book

Revision 1.0
912-2000-019
August 7, 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.)

List of Figures

List of Tables

ISA Docking Station Bridge

1.0 Features

- Provides two modes of operation:
  - Basic Mode - provides direct support for:
    - Complete PCI-to-ISA bridge
    - Full standard 16-bit ISA interface
    - ISA master support
    - One IDE drive channel, capable of supporting two PIO mode or bus mastering mode IDE drives
    - Zero TTL solution.
  - Extended Mode - requires the addition of an external 3-to-8-line decoder. With this extension, the system can provide support for:
    - Two IDE drive cables, capable of supporting up to four drives
    - Two extra REQ#/GNT# pairs for PCI.
    - Available as a strap-selected option at hard reset time
- A further option is the external control latch, which can be used to control miscellaneous devices on the docking station. Eight or 16 control lines can be provided through one or two 74373 parts. The control latch can be used in either Basic or Extended mode.

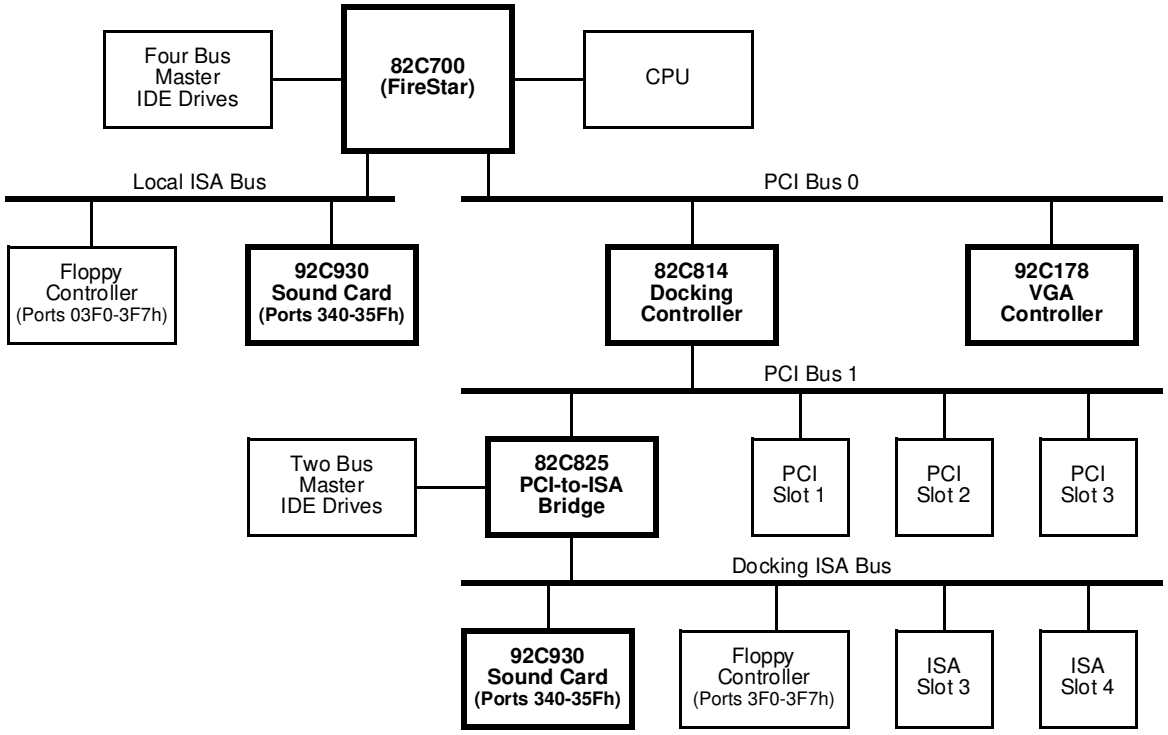
2.0 Overview

The OPTi 82C825 ISA Docking Station Bridge chip is a 160-pin device that simplifies docking station design by individually enabling and controlling each ISA device. Figure 2-1 illustrates the basic system architecture for which the 82C825 ISA Docking Station Bridge is intended.

The 82C825 is a specialized bridge solution that makes standard 8- and 16-bit ISA devices accessible across PCI buses and allows their features to be fully utilized, including DMA, ISA interrupts, and ISA bus masters, with no sideband signalling required.

While the 82C825 PCI interface is fully PCI-compliant and can reside on any PCI bus, it particularly complements the OPTi 82C814/82C824 docking station solutions to implement a secondary ISA bus on the docking station. The 82C814+82C825 configuration is referred to as FireBridge, and can be supported by either the Viper-N+ Chipset or by FireStar (single-chip solution).

Figure 2-1 Basic Docking Station System Architecture Example



3.0 Signal Definitions

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 82C825 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

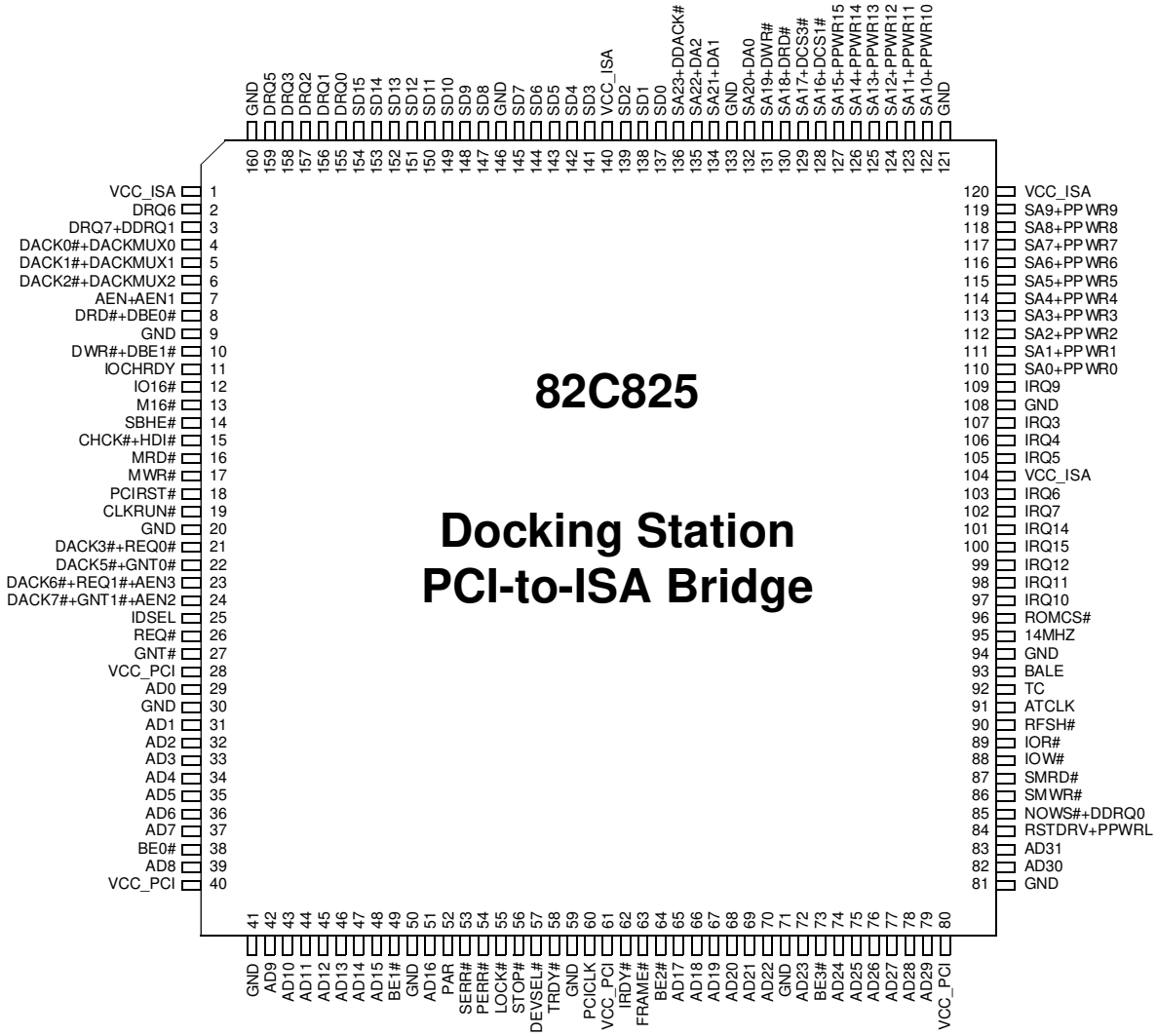


Table 3-2 Numerical Pin Cross-Reference List

Pin No.	Signal Name	Pin Type	Pwr Plane
1	VCC_ISA	P	ISA
2	DRQ6	I	ISA
3	DRQ7	I	ISA
	DDRQ1		
4	DACK0#	I/O	ISA
	DACKMUX0		
5	DACK1#	O	ISA
	DACKMUX1		
6	DACK2	O	ISA
	DACKMUX2		
7	AEN	O	ISA
	AEN1		
8	DRD#	I/O	ISA
	DBE0#		
9	GND	G	
10	DWR#	I/O	ISA
	DBE1#		
11	IOCHRDY	I	ISA
12	IO16#	I	ISA
13	M16#	I/O	ISA
14	SBHE#	I/O	ISA
15	CHCK#	I	ISA
	HDI#		
16	MRD#	I/O	ISA
17	MWR#	I/O	ISA
18	PCIRST#	I	PCI
19	CLKRUN#	I/O	PCI
20	GND	G	PCI
21	DACK3#	I/O	PCI
	REQ0#		
22	DACK5#	O	PCI
	GNT0#		
23	DACK6#	I/O	PCI
	REQ1#		
	AEN3		
24	DACK7#	O	PCI
	GNT1#		
	AEN2		
25	IDSEL	I	PCI
26	REQ#	O	PCI
27	GNT#	I	PCI
28	VCC_PCI	P	
29	AD0	I/O	PCI
30	GND	G	
31	AD1	I/O	PCI
32	AD2	I/O	PCI
33	AD3	I/O	PCI
34	AD4	I/O	PCI
35	AD5	I/O	PCI
36	AD6	I/O	PCI
37	AD7	I/O	PCI
38	BE0#	I/O	PCI
39	AD8	I/O	PCI
40	VCC_PCI	P	
41	GND	G	
42	AD9	I/O	PCI
43	AD10	I/O	PCI
44	AD11	I/O	PCI
45	AD12	I/O	PCI
46	AD13	I/O	PCI
47	AD14	I/O	PCI
48	AD15	I/O	PCI
49	BE1#	I/O	PCI
50	GND	G	
51	AD16	I/O	PCI
52	PAR	I/O	PCI
53	SERR#	OD	PCI
54	PERR#	I/O	PCI
55	LOCK#	I	PCI
56	STOP#	I/O	PCI
57	DEVSEL#	I/O	PCI
58	TRDY#	I/O	PCI
59	GND	G	
60	PCICLK	I	PCI
61	VCC_PCI	P	
62	IRDY#	I/O	PCI
63	FRAME#	I/O	PCI
64	BE2#	I/O	PCI
65	AD17	I/O	PCI
66	AD18	I/O	PCI
67	AD19	I/O	PCI
68	AD20	I/O	PCI
69	AD21	I/O	PCI
70	AD22	I/O	PCI
71	GND	G	
72	AD23	I/O	PCI
73	BE3#	I/O	PCI
74	AD24	I/O	PCI
75	AD25	I/O	PCI
76	AD26	I/O	PCI
77	AD27	I/O	PCI
78	AD28	I/O	PCI
79	AD29	I/O	PCI
80	VCC_PCI	P	
81	GND	G	
82	AD30	I/O	PCI
83	AD31	I/O	PCI
84	RSTDRV	O	ISA
	PPWRL		
85	NOWS#	I	ISA
	DDRQ0		
86	SMWR#	O	ISA
87	SMRD#	O	ISA
88	IOW#	I/O	ISA
89	IOR#	I/O	ISA
90	RFSH#	I/O	ISA
91	ATCLK	O	ISA
92	TC	O	ISA
93	BALE	O	ISA
94	GND	G	
95	14MHZ	I	ISA
96	ROMCS#	I/O	ISA
97	IRQ10	I	ISA
98	IRQ11	I	ISA
99	IRQ12	I	ISA
100	IRQ15	I	ISA
101	IRQ14	I	ISA
102	IRQ7	I	ISA
103	IRQ6	I	ISA
104	VCC_ISA	P	
105	IRQ5	I	ISA
106	IRQ4	I	ISA
107	IRQ3	I	ISA
108	GND	G	
109	IRQ9	I	ISA
110	SA0	I/O	ISA
	PPWR0		
111	SA1	I/O	ISA
	PPWR1		
112	SA2	I/O	ISA
	PPWR2		
113	SA3	I/O	ISA
	PPWR3		
114	SA4	I/O	ISA
	PPWR4		
115	SA5	I/O	ISA
	PPWR5		
116	SA6	I/O	ISA
	PPWR6		
117	SA7	I/O	ISA
	PPWR7		
118	SA8	I/O	ISA
	PPWR8		
119	SA9	I/O	ISA
	PPWR9		
120	VCC_ISA	P	
121	GND	G	
122	SA10	I/O	ISA
	PPWR10		
123	SA11	I/O	ISA
	PPWR11		
124	SA12	I/O	ISA
	PPWR12		
125	SA13	I/O	ISA
	PPWR13		
126	SA14	I/O	ISA
	PPWR14		
127	SA15	I/O	ISA
	PPWR15		
128	SA16	I/O	ISA
	DCS1#		
129	SA17	I/O	ISA
	DCS3#		
130	SA18	I/O	ISA
	DRD#		
131	SA19	I/O	ISA
	DWR#		
132	SA20	I/O	ISA
	DA0		
133	GND	G	
134	SA21	I/O	ISA
	DA1		
135	SA22	I/O	ISA
	DA2		
136	SA23	I/O	ISA
	DDACK#		
137	SD0	I/O	ISA
138	SD1	I/O	ISA
139	SD2	I/O	ISA
140	VCC_ISA	P	
141	SD3	I/O	ISA
142	SD4	I/O	ISA
143	SD5	I/O	ISA
144	SD6	I/O	ISA
145	SD7	I/O	ISA
146	GND	G	
147	SD8	I/O	ISA
148	SD9	I/O	ISA
149	SD10	I/O	ISA
150	SD11	I/O	ISA
151	SD12	I/O	ISA
152	SD13	I/O	ISA
153	SD14	I/O	ISA
154	SD15	I/O	ISA
155	DRQ0	I	ISA
156	DRQ1	I	ISA
157	DRQ2	I	ISA
158	DRQ3	I	ISA
159	DRQ5	I	ISA
160	GND	G	

Table 3-3 Alphabetical Pin Cross-Reference List

Signal Name	Pin No.
AD0	29
AD1	31
AD2	32
AD3	33
AD4	34
AD5	35
AD6	36
AD7	37
AD8	39
AD9	42
AD10	43
AD11	44
AD12	45
AD13	46
AD14	47
AD15	48
AD16	51
AD17	65
AD18	66
AD19	67
AD20	68
AD21	69
AD22	70
AD23	72
AD24	74
AD25	75
AD26	76
AD27	77
AD28	78
AD29	79
AD30	82
AD31	83
AEN+AEN1	7
ATCLK	91
BALE	93
BE0#	38
BE1#	49
BE2#	64
BE3#	73
CLKRUN#	19
CHCK#+HDI#	15
DACK0#+DACKMUX0	4
DACK1#+DACKMUX1	5
DACK2#+DACKMUX2	6
DACK3#+REQ0#	21
DACK5#+GNT0#	22
DACK6#+REQ1#+AEN3	23
DACK7#+GNT1#+AEN2	24
DEVSEL#	57
DRQ0	155
DRQ1	156
DRQ2	157
DRQ3	158
DRQ5	159
DRQ6	2
DRQ7+DDRQ1	3
DRD#+DBE0#	8
DWR#+DBE1#	10
FRAME#	63
GND	9
GND	20
GND	30
GND	41
GND	50
GND	59
GND	71
GND	81
GND	94
GND	108
GND	121
GND	133
GND	146
GND	160
GNT#	27
IDSEL	25
IO16#	12
IOCHRDY	11
IOR#	89
IOW#	88
IRDY#	62
IRQ3	107
IRQ4	106
IRQ5	105
IRQ6	103
IRQ7	102
IRQ9	109
IRQ10	97
IRQ11	98
IRQ12	99
IRQ14	101
IRQ15	100
LOCK#	55
NOWS#+DDRQ0	85
M16#	13
MRD#	16
MWR#	17
PAR	52
PCICLK	60
PCIRST#	18
PERR#	54
REQ#	26
RFSH#	90
ROMCS#	96
RSTDRV+PPWRL	84
SA0+PPWR0	110
SA1+PPWR1	111
SA2+PPWR2	112
SA3+PPWR3	113
SA4+PPWR4	114
SA5+PPWR5	115
SA6+PPWR6	116
SA7+PPWR7	117
SA8+PPWR8	118
SA9+PPWR9	119
SA10+PPWR10	122
SA11+PPWR11	123
SA12+PPWR12	124
SA13+PPWR13	125
SA14+PPWR14	126
SA15+PPWR15	127
SA16+DCS1#	128
SA17+DCS3#	129
SA18+DRD#	130
SA19+DWR#	131
SA20+DA0	132
SA21+DA1	134
SA22+DA2	135
SA23+DDACK#	136
SBHE#	14
SD0	137
SD1	138
SD2	139
SD3	141
SD4	142
SD5	143
SD6	144
SD7	145
SD8	147
SD9	148
SD10	149
SD11	150
SD12	151
SD13	152
SD14	153
SD15	154
SERR#	53
SMWR#	86
SMRD#	87
STOP#	56
TC	92
TRDY#	58
VCC_ISA	1
VCC_ISA	104
VCC_ISA	120
VCC_ISA	140
VCC_PCI	28
VCC_PCI	40
VCC_PCI	61
VCC_PCI	80
14MHZ	95

3.2 Signal Descriptions

3.2.1 PCI Interface Signals

Signal Name	Pin No.	Type	Selected By	Signal Description
AD[31:0]	83:82, 79:74, 72, 70:65, 51, 48:42, 39, 37:31, 29	I/O		**Multiplexed Address and Data Lines 31-0:** These pins are the multiplexed PCI address and data lines. During the address phase, these pins are inputs for PCI bus master cycles; otherwise they are outputs. During the data phase, these pins are inputs during PCI bus master write cycle or during CPU/VESA/DMA/ISA read from PCI bus slave; otherwise they are outputs.
BE[3:0]#	73, 64, 49, 38	I/O		**Byte Enables 3-0:** These pins are the multiplexed PCI command and byte enable lines. Normally outputs, these pins are inputs during PCI bus master cycle.
PCIRST#	18	I		**Reset:** This is RESET# from the system, used to reset PCI devices to a known state.
PCICLK	60	I		**Clock:** This signal is used to provide timing for all transactions on PCI bus.
CLKRUN#	19	I/O		**Clock Run:** The host uses this signal to indicate that it will stop the clock; the 82C825 chip uses it to prevent clock shutdown or to restart the clock. If not used, this signal must be connected to ground.
PAR	52	I/O		**Parity:** This signal is an input either during PCI bus master cycles for address and write data phases or during PCI bus slave cycle for read data phase; otherwise it is an output.
FRAME#	63	I/O		**Cycle Frame:** This pin is driven by PCI bus masters to indicate the beginning and duration of an access. Normally an input, FRAME# is driven during CPU/VESA/DMA/ISA master access to PCI bus slaves.
IRDY#	62	I/O		**Initiator Ready:** This signal is asserted by PCI bus masters to indicate the ability to complete the current data phase of the transaction. Normally an input, this pin is driven during CPU/VESA/DMA/ISA master accesses to PCI bus slaves.
TRDY#	58	I/O		**Target Ready:** This pin is asserted by the target to indicate the ability to complete the current data phase of the transaction. Normally an input, this pin is driven during PCI bus master accesses to local memory, VESA/ISA slaves and the configuration registers inside the PCIB.
STOP#	56	I/O		**Stop:** This signal is used by the target to request the master to stop the current transaction. Normally an input, this signal is driven during PCI bus master access to local memory and VESA/ISA slaves.
LOCK#	55	I		**Lock:** This signal is used to indicate an atomic operation that may require multiple transactions to complete. Since the PCIB will never assert this signal, it is always an input.

Signal Name	Pin No.	Type	Selected By	Signal Description
DEVSEL#	57	I/O		**Device Select:** This pin is an output when the PCIB decodes its address as the target of the current access via either positive or negative decoding; otherwise it is an input.
IDSEL	25	I		**ID Select:** This pin is used to access the 82C825 configuration space.
PERR#	54	I/O		**Parity Error:** Reports data parity errors during all PCI transaction except a special cycle. Normally an input, PERR# is driven when data parity errors occur either during PCI bus master write cycle or during CPU/VESA/DMA/ISA master read from a PCI slave.
SERR#	53	OD		**System Error:** Reports address parity errors, or data parity errors on the special cycle command, or any other system error where the result will be catastrophic. This pin is an open drain.
REQ#	26	O		**PCI Bus Request:** Asserted when the DMA controller needs to access memory, or when the chip needs to generate an IRQ driveback cycle.
GNT#	27	I		**PCI Bus Grant:** Acknowledge signal from the host (the 82C824 chip) that the 82C825 chip has ownership of the PCI bus.

3.2.2 ISA Interface Signals

Signal Name	Pin No.	Type	Selected By	Signal Description
SA23	136	I/O		**System Address Bus Line 23:** SA[23:0] provide the memory and I/O access on the ISA bus. The addresses are outputs when the 82C825 owns the ISA bus and are inputs when an external ISA master owns the ISA bus.
DDACK#				**Drive DMA Acknowledge**
SA[22:20]	135, 134, 132	I/O		**System Address Bus Lines 22 through 20**
DA[2:0]				**Drive Address BitS 2 through 0**
SA19	131	I/O		**System Address Bus Line 19**
DWR#				**Drive Write Control**
SA18	130	I/O		**System Address Bus Line 18**
DRD#				**Drive Read Control**
SA17	129	I/O		**System Address Bus Line 17**
DCS3#				**Drive Chip Select 3** (3x0-3x7h)
SA16	128	I/O		**System Address Bus Line 16**
DCS1#				**Drive Chip Select 1** (1x0-1x7h)
SA[15:0]	127:122, 119:110	I/O	Cycle Multiplexed	**System Address Bus Lines 15 through 0**
PPWR15-0				**Power Control Latch Information Lines 15 through 0**

Signal Name	Pin No.	Type	Selected By	Signal Description
SD[15:0]	154:147, 145:141, 139:137	I/O		**System Data Bus Lines 15-0:** SD[15:0] provides the 16-bit data path for devices residing on the ISA bus.
MRD#	16	I/O		**Memory Read**
MWR#	17	I/O		**Memory Write**
IOR#	89	I/O		**I/O Read**
IOW#	88	I/O		**I/O Write**
RSTDRV	84	O	Default	**Reset**
PPWRL			PCICFG 52h[7] = 1	**Power Control Latch Enable**
TC	92	O		**Terminal Count**
BALE	93	O		**Bus Address Latch Enable**
IOCHRDY	11	I		**I/O Channel Ready**
IO16#	12	I		**16-bit I/O Cycle**
M16#	13	I/O		**16-bit Memory Cycle**
SBHE#	14	I/O		**High Byte Enable**
CHCK#	15	I	Default	**I/O Channel Check**
HDI#			PCICFG 52h[3] = 1	**IDE Hot Drive Insertion Indication**
ATCLK	91	O		**ISA Bus Clock**
RFSH#	90	I/O		**ISA Refresh Indication**
NOWS#	85	I		**ISA No Wait State Signal**
DDRQ0			PCIIDE 43h[3:0] (any bit =1)	**Bus Mastering IDE Cable 0 DMA Request** (setting any PCIIDE 43h[3:0] bit selects DDRQ0 function)
SMRD#	87	O		**8-bit Memory Read**
SMWR#	86	O		**8-bit Memory Write**

3.2.3 DMA/Interrupt Interface and Miscellaneous Signals

Signal Name	Pin No.	Type	Selected By	Signal Description
DRQ0	155	I		**DMA Request Line**
DRQ1	156	I		**DMA Request Line**
DRQ2	157	I		**DMA Request Line**
DRQ3	158	I		**DMA Request Line**
DRQ5	159	I		**DMA Request Line**
DRQ6	2	I		**DMA Request Line**

Signal Name	Pin No.	Type	Selected By	Signal Description
DRQ7	3	I		**DMA Request Line**
DDRQ1			PCIIDE 43h[7:4] (any bit =1)	**Bus Mastering IDE Cable 0 DMA Request** (setting any PCIIDE 43h[7:4] bit selects DDRQ1 function)
DACK0#	4	I/O	DRD# Strap Option	**DMA Acknowledge Line**
DACKMUX0				**Muxed DACKs**
DACK1#	5	O	DRD# Strap Option	**DMA Acknowledge Line**
DACKMUX1				**Muxed DACKs**
DACK2#	6	O	DRD# Strap Option	**DMA Acknowledge Line**
DACKMUX2				**Muxed DACKs**
DACK3#	21	O	Default	**DMA Acknowledge Line**
REQ0#		I	DRD# Strap Option	**PCI Request 1**
DACK5#	22	O	Default	**DMA Acknowledge Line**
GNT0#			DRD# Strap Option	**PCI Grant 0**
DACK6#	23	O	Default	**DMA Acknowledge Line**
REQ1#		I	DRD# Strap Option	**PCI Request 1**
AEN3		O	PCICFG 52h[6] = 1	**AEN for Slot 3**
DACK7#	24	O	Default	**DMA Acknowledge Line**
GNT1#			DRD# Strap Option	**PCI Grant 1**
AEN2			PCICFG 52h[6] = 1	**AEN for Slot 2**
AEN	7	O		**Address Enable**
AEN1			PCICFG 52h[6] = 1	**AEN for Slot 1**
IRQ3	107	I		**Interrupt Input**
IRQ4	106	I		**Interrupt Input**
IRQ5	105	I		**Interrupt Input**
IRQ6	103	I		**Interrupt Input**
IRQ7	102	I		**Interrupt Input**
IRQ9	109	I		**Interrupt Input**
IRQ10	97	I		**Interrupt Input**
IRQ11	98	I		**Interrupt Input**
IRQ12	99	I		**Interrupt Input**
IRQ14	101	I		**Interrupt Input**

Signal Name	Pin No.	Type	Selected By	Signal Description
IRQ15	100	I		**Interrupt Input**
DRD#	8	I/O	Default	**IDE Cable 0 Chip Select, also Strap Option Sense**
DBE0#			PCICFG 52h[4] = 1	**IDE Cable 0 Buffer Enable** (4-drive configuration)
DWR#	10	I/O	Default	**IDE Cable 0 Chip Select, also Strap Option Sense**
DBE1#			PCICFG 52h[4] = 1	**IDE Cable 1 Buffer Enable** (4-drive configuration)
ROMCS#	96	I/O	Strap Option	**Expansion / CIS ROM Chip Select**
14MHZIN	95	I		**14MHz Oscillator Input** (also goes to OSC pin on ISA)

3.2.4 Power and Ground Pins

Signal Name	Pin No.	Type	Signal Description
VCC_ISA	1, 104, 120, 140	P	**Power Connection:** ISA bus power plane (3.3V or 5.0V)
VCC_PCI	28, 40, 61, 80	P	**Power Connection:** PCI bus power plane (3.3V or 5.0V)
GND	9, 20, 30, 41, 50, 59, 71, 81, 94, 108, 121, 133, 146, 160	G	**Ground Connection**

3.3 Strap-Selected Options

The 82C825 chip offers certain optional performance that must be selected before any registers can be programmed. This selection is achieved through strap-selected options, which are simply pull-up or pull-down resistors placed on normally output-only signals. At hard reset time, the chip tristates these outputs and reads in the value on the line to determine whether the option has been selected or not. Weak (about 50K ohm) internal pull-down resistors are provided so that external pull-downs may not be necessary.

The 82C825 part currently requires some features to be enabled in this way.

- A 10K ohm pull-down resistor sensed on DACK0# at reset time enables the PCI bus 3.3V interface.
- A 10K ohm pull-up resistor sensed on DRD# at reset time causes the DACK0-7# pins to be redefined for Extended mode operation (see Section 4.9, “Extended Mode Interface”). A pull-down or no resistor on DRD# leaves the DACK0-7# signal pins with their original function.
- A 10K ohm pull-up resistor sensed on ROMCS# at reset time enables the ROM interface and the associated ROMCS# pin function. A pull-down or no resistor on ROMCS# disables the ROM interface; the PCI Expansion ROM Register will always return 0 in this case. The PCICFG and PCIIDE registers that would map to the ROM also always return 0.
- A 10K ohm pull-up resistor sensed on DWR# at reset time forces the chip into Test Mode. Along with DWR# high, DRD# is sensed and used to determine the specific test to be enabled:
  - DWR# = 1, DRD# = 0 - Tristate test mode: All outputs are tristated.
  - DWR# = 1, DRD# = 1 - NAND tree test mode: All pins except ROMCS# become inputs to a NAND tree, with ROMCS# becoming the output of the tree.

DRD#, DWR#, and ROMCS# going low (active) at reset time will not cause any problems with their attached devices: the ROM does not have MRD# or MWR# to qualify the ROMCS#; the IDE drive will be in reset and must ignore the DRD# and DWR# signals.

4.0 Functional Description

The following sections discuss features of the 82C825 chip that are common to all configurations, as well as extra features that come with Extended mode. Extended mode operation requires the addition of a 74138 decoder. Selecting between Basic and Extended mode is discussed in Section 3.3, “Strap-Selected Options”.

4.1 ISA Slave Interface

ISA peripheral devices can be accessed on a remote bus as easily as on the local ISA bus. Host system PCI masters typically generate cycles in the normal ISA space, memory ranges 0-FFFFFFh (first 16MB) and I/O ranges 0-FFFFh (first 64KB), to access devices on the docking station ISA bus. The host bridge in an OPTi system, such as the 82C824 Docking Controller, claims all such cycles presented on its primary PCI bus and forwards them to the docking station. If the cycle is not positively decoded by the 82C825 ISA bridge, the OPTi host chipset recovers the cycle and presents it on its local ISA bus.

The ISA interface is fully buffered from the PCI interface by a FIFO. This FIFO is used in both directions; the logic flushes the buffer before starting a transfer in the opposite direction. ISA DMA and master accesses can be configured to delay acquisition of the PCI bus until the FIFO has been filled to a specified level.

4.2 ISA Master Interface

ISA bus devices that depend on MASTER# control can be used without compatibility problems. The chipset provides a time-out scheme to force these peripheral devices to surrender the bus on demand as required by PCI, by deasserting IOCHRDY to the device whenever it is necessary to temporarily interrupt its connection to the PCI bus.

A register that is programmable by the PCI host allows the upper bits of ISA master accesses to be specified, providing 32-bit ISA master address capability.

4.3 DMA Interface

OPTi Mobile PCI chips support a feature that allows remote PCI-based devices to contain a DMA controller (DMAC) that parallels the function of the DMAC in the main system. The scheme is referred to in this document as the Distributed DMA Protocol.

- DMA slaves are fully supported through the distributed DMA protocol. The 82C825 logic incorporates a seven-channel DMA controller based on distributed DMA protocol address decoding; it does not respond to ISA DMA register address accesses. Therefore the PCI host must remap DMA register programming to these addresses.
- Each DRQ/DACK# pair can be reprogrammed to any other DMA channel instead of its default setting.
- The DMA controller services DMA requests on a local basis: the I/O part of the cycle occurs on the ISA bus, while the memory access occurs on PCI or ISA.
- Distributed DMA support incorporates 32-bit DMA addresses and 24-bit counts as a standard feature.

For more information regarding the Distributed DMA Protocoll see Appendix B.

4.4 IRQ Interface

A complete interrupt interface is provided, with no sideband signalling required.

- The 82C825 ISA bus supports the standard set of ISA IRQ lines: IRQ3-12, 14-15. There is no interrupt controller in the 82C825; it just acts to transfer the interrupts back to the host.
- Each ISA IRQ pin can be individually reprogrammed to generate any system IRQ. SMI# and NMI generation is also possible.
- The OPTi IRQ driveback mechanism is the method used to return IRQs to the host PCI chipset.
- Various events can be programmed to generate an SMI, which is returned to the host through the IRQ driveback scheme as IRQ2.
- The ISA CHCK# signal generates IRQ13 to indicate an NMI to the host.

4.5 IDE Interface

The local bus IDE interface uses an ultra-high performance mechanism to allow sustained high transfer rates.

- PCI-based IDE is provided, incorporating bus mastering capability. The SFF (ATA-3) register set is used. One channel is directly supported in basic mode, two are supported with external TTL; each channel supports two drives. Two independent timing settings are available. A full implementation can support four IDE devices, such as CD-ROM drives and hard disk drives, having different timing requirements.
- Hot insertion of IDE devices is supported through the HDI# pin. HDI# puts the local ISA bus on hold until the drive is safely powered up, and optionally can generate an SMI to the host CPU.
- The IDE controller supports PIO modes 0-5 and DMA modes 0-2.

Support of IDE drives is described next.

4.5.1 Direct IDE Cable Support

A single IDE cable can be supported directly. The upper address lines of the SA bus are used to provide most of the control signals to the drive; these are qualified by dedicated DRD# and DWR# signals from the 82C825 chip. Table 4-1 shows the correspondence of the control lines to the SA lines.

Table 4-1 IDE Control Signals

IDE Signal	ISA Pin Name	Description
DCS1#	SA16	Drive Chip Select 1 (1x0-1x7h)
DCS3#	SA17	Drive Chip Select 3 (3x0-3x7h)
DA0	SA20	Drive Address bit 0
DA1	SA21	Drive Address bit 1
DA2	SA22	Drive Address bit 2
DDACK#	SA23	Drive DMA Acknowledge
DRD#	Dedicated	Drive Read Control
DWR#	Dedicated	Drive Write Control
DDRQ	NOWS#	Drive DMA Request

Note that NOWS# must be sacrificed to obtain a dedicated DDRQ input for bus mastering IDE drives. However, the SA[23:20] and SA[17:16] lines are time-multiplexed with the IDE control signals so no functionality is lost.

4.5.2 Buffered Dual IDE Cable Support

Two IDE cables can be supported through a buffering scheme. Once again, the upper address lines of the SA bus are used to provide control signals to the drive; these are qualified by dedicated DBE0# and DBE1# signals from the 82C825 chip. Table 4-2 shows the correspondence of the control lines to the SA lines. Note that most of the shared signals are the same as in direct support mode, except that DRD# and DWR# come from the SA lines (the 82C825 DRD# and DWR# pins from direct support mode become DBE0# and DBE1# in buffered two-cable mode).

Table 4-2 IDE Control Signals

IDE Signal	ISA Pin Name	Description
DCS1#	SA16	Drive Chip Select 1 (1x0-1x7h)
DCS3#	SA17	Drive Chip Select 3 (3x0-3x7h)
DRD#	SA18	Drive Read Control
DWR#	SA19	Drive Write Control
DA0	SA20	Drive Address bit 0
DA1	SA21	Drive Address bit 1
DA2	SA22	Drive Address bit 2
DDACK#	SA23	Drive DMA Acknowledge
DDRQ0	NOWS#	Drive DMA Request - Cable 0
DDRQ1	DRQ7	Drive DMA Request - Cable 1
DBE0#	Dedicated	Drive Buffer Enable - Cable 0
DBE1#	Dedicated	Drive Buffer Enable - Cable 1

Note that NOWS# and DRQ7 must be sacrificed to obtain dedicated DDRQ inputs for bus mastering IDE drives on both cables.

4.6 PCI Interface

The PCI interface is fully compliant with revision 2.1 of the PCI specification.

- Full concurrency is supported on PCI bus and ISA bus operations, allowing the two to remain effectively isolated when possible.
- Sustained -1-1-1 burst transfers are practical because of the on-chip FIFO.
- The CLKRUN# pin and protocol are fully supported to save power when possible.
- The 82C825 chip requires the use of a request/grant pair, so it provides extra request/grant pairs to compensate when in the Extended mode. Two extra PCI masters are supported by encoding the seven DACK# pins onto three pins, decoded through an external 74138 chip.

4.7 ISA Decoding Feature

The 82C825 chip always uses ISA positive decoding. The 82C825 logic positively decodes the remapped cycles and forwards them to its ISA bus. In this way, the remote ISA bus will see all the cycles that it would in a typical desktop system.

Because ISA does not provide any dedicated acknowledgment signal to indicate that a cycle was claimed by a local device, the 82C825 logic uses a special protocol to determine whether or not to claim the cycle from PCI. This is an OPTi-exclusive feature described below.

4.7.1 ISA Write Cycle

The 82C825 logic waits to determine whether the peripheral device has responded by asserting M16#, IO16#, or NOWS#, or by deasserting IOCHRDY. Any of these signalling events indicates that an ISA device is responding to the cycle and that the 82C825 logic can claim the cycle without waiting for its completion. Note that because of M16#/IO16#, 16-bit writes can **always** be positively decoded.

If for an 8-bit write no device positively decodes the cycle, the 82C825 generates target abort of the write cycle on the PCI side even though it completes the cycle on the ISA side. The 82C824 Docking Controller retries ISA-directed cycles on its primary PCI bus; if the 82C825 generates target abort, the 82C824 logic ignores the retry when the host tries again. In this way, the write cycle will automatically go to the local ISA bus of the OPTi host controller (the Viper-N+ or FireStar chipsets). So in the worst case, the 8-bit write will occur on both buses (but no harm is done).

4.7.2 ISA Read Cycle

The 82C825 logic waits to determine whether the peripheral device has responded by asserting M16#, IO16#, or NOWS#, or by deasserting IOCHRDY. Any of these signalling events indicates that an ISA device is responding to the cycle and the 82C825 logic can claim the cycle without waiting for its completion. Note that because of M16#/IO16#, 16-bit reads can always be positively decoded.

If for an 8-bit read no prior event has conclusively determined that an ISA device is claiming the cycle, the 82C825 waits for read data to be returned. If the data on SD[7:0] is anything other than 0FFh, the 82C825 chip claims the cycle. If the data is 0FFh, the 82C825 PCI interface aborts the cycle by generating target abort on its PCI bus. The host controller will then run the cycle on its local ISA bus (as described above). If the 0FFh value was indeed valid data, the host will get the same value from its local ISA bus (since the ISA data bus is always pulled up). Therefore, even data reads where 0FFh is the valid response will be handled properly.

This comprehensive handling of ISA by the 82C825 design ensures full compatibility for all devices on the docking station ISA bus.

4.7.3 Plug and Play Support

The 82C825 chip offers the following features to meet plug and play requirements, allowing Windows '95 and other operating systems to resolve system resource conflicts.

- Each IRQ line can be steered to any ISA IRQ.
- Each DRQ/DACK# pair can be steered to any ISA DMA channel.
- Using an optional external latch, each ISA slot can have a dedicated RSTDRV line. All slot devices are disabled at reset time and can be individually enabled and mapped into the available system resources.
- In Extended mode, each of three ISA slots can have a separate AEN line. This feature allows the I/O space in each card to be mapped separately.

IRQ and DMA steering is straightforward and is handled through the PCI configuration registers. The dedicated RSTDRV and dedicated AEN features are described below.

4.7.4 Individual RSTDRV Control

The 82C825 chip offers the dual-purpose signal RSTDRV/ PPWRL. At chip initialization, the pin acts as PPWRL, and connects to the latch enable input of an external '373 latch. The inputs to this device are any of SA[15:0]; two '373 latches can be used if needed for spare control lines up to a total of 16. The 82C825 logic drives the SA lines to a predetermined setting at reset, so the latch retains this value when PPWRL pulses.

Once the 82C825 programming begins, software must reprogram the RSTDRV signal to act solely as PPWRL. It cannot be used for both at the same time, since every PPWRL toggle would reset the ISA bus. PCICFG 52h[7] is used to enable the PPWRL function. Refer to Table 4-3.

Table 4-3 PPWRL Function Enable Bit

7	6	5	4	3	2	1	0
PCICFG 52h			**Feature Control Register**				**Default = 20h**
PPWRL function on RSTDRV: 0 = Disable (use pin only as RSTDRV) 1 = Enable PPWRL	Individual AEN interface: 0 = Disable 1 = Enable (DACK-MUX strap option required)	IDE controller interface: 0 = Disable (PCIIDE function will not respond) 1 = Enable (Default)	IDE control pin function select: 0 = DRD#, DWR# 1 = DBE0#, DBE1#	CHCK# pin active halts ISA bus: 0 = No (normal CHCK#) 1 = Yes (HDI# function)	Flash EEPROM writes: 0 = Disable (SMWR# is blocked also) 1 = Enable	RQ8# active level: 0 = Low 1 = High	DMA register access: 0 = ISA mapping 1 = Distributed DMA mapping (Default)

If PPWRL is enabled, the RSTDRV1-4 signals are derived from the SA bus in conjunction with the PPWRL-controlled latch. RSTDRV is generated separately for each slot so that the card can be kept in reset until it can be incorporated properly into the system resource map. ISA cards that conflict with other resources can be kept permanently disabled.

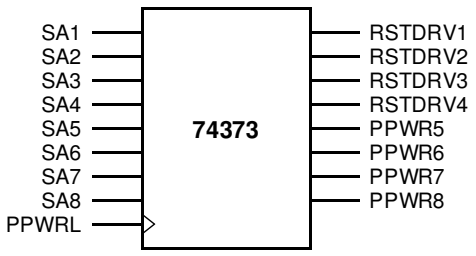
Table 4-4 RSTDRV Control

Signal	ISA Control	Derived From	Description
PPWR0	--	SA0	Reserved
PPWR1	RSTDRV1	SA1	Generate RSTDRV to ISA Slot 1
PPWR2	RSTDRV2	SA2	Generate RSTDRV to ISA Slot 2
PPWR3	RSTDRV3	SA3	Generate RSTDRV to ISA Slot 3
PPWR4	RSTDRV4	SA4	Generate RSTDRV to ISA Slot 4
PPWR5-15	--	SA5-15	User-definable controls

The same PPWRL latch that provides ISA reset lines also can provide general purpose control lines if needed. SA0-15 are latchable with PPWRL to provide power control pins PPWR0-15. Each line is individually controllable through software.

Figure 4-1 illustrates the external logic required to provide individual ISA slot reset control. PPWR5-8 (or any other PPWR signals desired) are available on the unused half of the latch. They can be used to control external devices as needed.

Figure 4-1 External Connections for Individual Slot Reset Control



4.7.5 Individual AEN Control

AEN signals can be provided separately for each of three slots in Extended mode. AEN1, 2, and 3 stay high to block ISA cards from responding to I/O cycles. PCICFG 5Ch-5Dh allow the pins to be programmed individually to respond to either all ISA addresses or only to those in certain address ranges (see Table 4-5). Seven decode ranges are provided to meet the requirements of PC96. The separate AEN signals are made available as follows:

- AEN1 comes from the original AEN pin.
- AEN2 and AEN3 are alternative signals to PCI signals REQ1# and GNT1#. If AEN2 and AEN3 are used, only one extra PCI request/grant pair, REQ0#/GNT0#, is available.
- Any additional slots use undecoded AEN, which is available from the decoder used to provide the DACK# lines.

Refer to Figure 4-2 for information on the DACK# decoder.

Table 4-5 AEN Control Registers

7	6	5	4	3	2	1	0
PCICFG 52h			**Feature Control Register**				**Default = 20h**
PPWRL function on RSTDRV: 0 = Disable (use pin only as RSTDRV) 1 = Enable PPWRL	Individual AEN interface: 0 = Disable 1 = Enable (DACK-MUX strap option required)	IDE controller interface: 0 = Disable (PCIIDE function will not respond) 1 = Enable (Default)	IDE control pin function select: 0 = DRD#, DWR# 1 = DBE0#, DBE1#	CHCK# pin active halts ISA bus: 0 = No (normal CHCK#) 1 = Yes (HDI# function)	Flash EEPROM writes: 0 = Disable (SMWR# is blocked also) 1 = Enable	RQ8# active level: 0 = Low 1 = High	DMA register access: 0 = ISA mapping 1 = Distributed DMA mapping (Default)
PCICFG 5Ch			**ISA Slot Control Register 1**				**Default = 00h**
AEN2 high for x000-x3F8h: 0 = No 1 = Yes	AEN2 enabled (low) for I/O range: 000 = 0000-FFFFh (normal AEN) 001 = 1000-1FFFh 010 = 2000-2FFFh 011 = 3000-3FFFh 100 = 4000-4FFFh 101 = 5000-5FFFh 110 = 6000-6FFFh 111 = 7000-7FFFh			AEN1 high for x000-x3F8h: 0 = No 1 = Yes	AEN1 enabled (low) for I/O range: 000 = 0000-FFFFh (normal AEN) 001 = 1000-1FFFh 010 = 2000-2FFFh 011 = 3000-3FFFh 100 = 4000-4FFFh 101 = 5000-5FFFh 110 = 6000-6FFFh 111 = 7000-7FFFh		
PCICFG 5Dh			**ISA Slot Control Register 2**				**Default = 00h**
Reserved				AEN3 high for x000-x3F8h: 0 = No 1 = Yes	AEN3 enabled (low) for I/O range: 000 = 0000-FFFFh (normal AEN) 001 = 1000-1FFFh 010 = 2000-2FFFh 011 = 3000-3FFFh 100 = 4000-4FFFh 101 = 5000-5FFFh 110 = 6000-6FFFh 111 = 7000-7FFFh		

4.7.6 Host Chipset Support

The 82C824/82C825 configuration can be supported by either the Viper-N+ Chipset or by the FireStar Single-Chip Solution. The only difference is in cycle decode time: Viper-N+ requires all local ISA cycles to pass through the docking station ISA bus first, while FireStar can positively decode cycles to known devices on its local ISA bus and effectively bypass the docking station.

4.8 ROM Interface

The 82C825 chip supports an external 256K (32Kx8) ROM that can be accessed through the ROMCS# pin. The external ROM is an optional part of the system. Providing a ROM makes it much simpler to initialize the docking station.

- Standard PCI configuration software can map the ROM into system memory space as a standard Option ROM, just as it might do for a video card Option ROM. Doing so allows the Option ROM code to configure the 82C825-specific registers, as well as docking station operations in general.
- Flash EEPROM can be used instead of a ROM. Special setup programs can then reprogram the flash EEPROM if the configuration needs change. The PPWR0 pin is assigned to controlling the programming voltage VPP to a flash EEPROM.
- Also mapped into the ROM are many PCI registers such as Subsystem Vendor ID and the CardBus CIS header. Host configuration software can depend on the values read from these registers to identify and properly configure the docking station.

The ROM interface feature is an option that is strap-selected at reset time. Table 4-6 indicates the mapping of the ROM into the PCI configuration space when the ROMCS# option has been strap-selected.

Table 4-6 ROM Mapping

Address Range	Register Name	Bytes Used For:	Mapped to the ROM on SA[17:0] at Address:
PCICFG 2Ch-2Dh	Function 0 Subsystem Vendor ID	Vendor-specific identification for Function 0 (PCI-ISA bridge)	F000:402C-402Dh
PCICFG 2Eh-2Fh	Function 0 Subsystem ID		F000:402E-402Fh
PCICFG 60h-FFh	CIS Header	Pointed at by CIS Pointer Register (PCICFG 28h)	F000:4060-40FFh
PCIIDE 2Ch-2Dh	Function 1 Subsystem Vendor ID	Vendor-specific identification for Function 1 (IDE controller)	F000:412C-412Dh
PCIIDE 2Eh-2Fh	Function 1 Subsystem ID		F000:412E-412Fh
Expansion ROM Base Register Address + 0-7FFFh	PCI Expansion ROM	Option ROM space - code can be executed to set up 82C825 chip	F000:0-7FFFh

Note from the table that the SA bus address used to access the ROM is in the F0000-F7FFFh range. Since no ISA option ROMs should occupy this space, ROM access will be exclusive.

The mapping to the 0-7FFFh addressing range of ROM space is very straightforward.

- 0-7FFFh: This space is accessible through system memory space at the register base selected by the Expansion ROM Base Address Register (PCICFG 30h or PCIIDE 30h).
- 4000-412Fh: This space is read-only through PCI configuration space, and only part of it is used. The PCICFG and PCIIDE addresses shown in the table map directly into this space: bits AD[8:2] from the PCI configuration cycle are mapped to SA[8:2] on the ISA bus. Note that 16-bit and 32-bit configuration read cycles are supported, and are broken down automatically into 8-bit reads from the ROM. This space is also accessible through system memory space at the address base selected by the Expansion ROM Base Address Register (PCICFG 30h or PCIIDE 30h). The two spaces overlap, so Expansion ROM firmware must be careful to jump around the locations accessible in PCICFG space.
- If VPP is supplied to the ROM, it can be written as well as read in system memory space. Note that only byte writes are supported when programming the flash EEPROM; PCI writes cannot be 16-bit or 32-bit.

4.9 Extended Mode Interface

The 82C825 chip provides two operating modes:

1. A full-featured PCI-to-ISA bridge mode requiring no TTL.
2. An Extended mode that provides two extra features. A strap option on DCS1# enables Extended mode.

Additional PCI Bus Masters - The 82C825 chip requires a PCI master REQ#/GNT# pair from the host system or docking bus chip. To compensate, Extended mode provides up to two additional REQ#/GNT# pairs for PCI devices.

Individual AEN Signals to Each Slot - Instead of providing two extra REQ#/GNT# pairs, just one pair can be provided. In this case, the original AEN line along with the two spare pins become AEN1, AEN2, and AEN3. PCICFG 52h[6] is used to enable this feature, which is described in Section 4.7.4, "Individual RSTDRV Control".

Figure 4-2 illustrates the external TTL required to recover the DACK# lines when Extended mode is selected.

4.9.1 Programmable Features

The following features can be enabled independently of Extended mode.

Second Bus Master IDE Interface - DBE1# and DDRQ1 become available when programmed. In this way, the 82C825 chip can support four separate IDE drives can be supported on two IDE cables.

Hot Drive Insertion - The HDI# pin allows a hard disk drive to be connected to the system while the system is running. Assertion of the HDI# pin immediately stops all activity on the ISA bus to allow safe connection of a removable hard disk drive. HDI# also generates an interrupt or an SMI on the host side. The host must clear the HDI# event before the ISA bus can be used again.

The HDI# pin defaults to the CHCK# pin function on ISA, and therefore generates NMI when asserted. PCICFG 52h[3] allows the pin to also stop the ISA bus, which is what selects the HDI# feature. PCICFG 4Dh[7:4] allows the interrupt generated to be changed from NMI to some other convenient interrupt.

Figure 4-2 Recovery of DMA Lines in Extended Mode

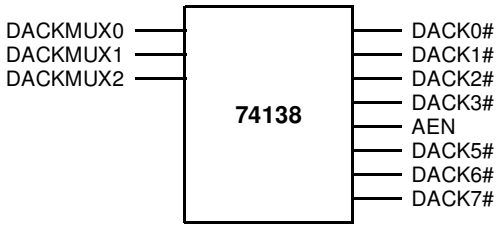


Table 4-7 HDI# Pin Programmable Features

7	6	5	4	3	2	1	0
PCICFG 52h			**Feature Control Register**				**Default = 20h**
PPWRL function on RSTDRV: 0 = Disable (use pin only as RSTDRV) 1 = Enable PPWRL	Individual AEN interface: 0 = Disable 1 = Enable (DACK-MUX strap option required)	IDE controller interface: 0 = Disable (PCIIDE function will not respond) 1 = Enable (Default)	IDE control pin function select: 0 = DRD#, DWR# 1 = DBE0#, DBE1#	CHCK# pin active halts ISA bus: 0 = No (normal CHCK#) 1 = Yes (HDI# function)	Flash EEPROM writes: 0 = Disable (SMWR# is blocked also) 1 = Enable	RQ8# active level: 0 = Low 1 = High	DMA register access: 0 = ISA mapping 1 = Distributed DMA mapping (Default)

PCICFG 4Dh			**IRQ Channel Selector Register 6**				**Default = DFh**
HDI pin (Default = NMI):				IRQ15 pin (Default = IRQ15):			
0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12	0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12
0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI
0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14	0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15

5.0 Register Set

The 82C825 register set comprises two register groups:

- Function 0 is the PCI-to-ISA Bridge function.
- Function 1 is the bus-mastering IDE function.

These functions are described in the sections below.

5.1 Function 0: PCI-to-ISA Bridge Configuration Registers

The 82C825 chip has a main set of PCI-to-ISA Bridge configuration registers that are accessed as PCI Function 0. These configuration registers are broken up into two groups:

- 82C825 Base Register Group: PCICFG 00h-3Fh
  - This register group is the standard group required for PCI peripheral device identification and configuration for the OPTi PCI-to-ISA bridge
- 82C825-Specific Register Group: PCICFG 40h-FFh
  - This register group defines many special functions that require enabling and monitoring through a dedicated register set.

The bit formats for the Base Registers are given in Table 5-1 and Table 5-2 gives the bit formats or the Specific Registers.

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

Table 5-1 82C825 Base Register Group: PCICFG 00h-3Fh

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 = 25h**
PCICFG 03h			**Device ID (RO) - Byte 1**				**Default = C8h**
PCICFG 04h			**PCI Command Register - Byte 0**				**Default = 00h**
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: 0= Disable 1 = Enable	Respond to PCI memory 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
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

Table 5-1 82C825 Base Register Group: PCICFG 00h-3Fh (cont.)

7	6	5	4	3	2	1	0
PCICFG 08h			**Revision ID Register (RO)**				**Default = 00h**
PCICFG 09h			**Programming Interface Class Code Register (RO)**				**Default = 00h**
PCICFG 0Ah			**Class Code Register (RO) - Byte 0**				**Default = 01h**
			Subclass Code bits: = 01h (PCI-to-ISA Bridge)				
PCICFG 0Bh			**Class Code Register (RO) - Byte 1**				**Default = 06h**
			Base Class Code bits: = 06h (Bridge Device)				
PCICFG 0Ch			**Cache Line Size Register**				**Default = 00h**
			Not implemented				
PCICFG 0Dh			**Latency Timer Register**				**Default = 20h**
Latency Timer: - This value indicates the time-out on the primary while the 82C825 attempts to determine whether the secondary will claim the cycle. Because the secondary bus is ISA and does not attempt retries, this register has meaning only for ISA masters. - A zero value indicates that the master can hold the primary PCI bus indefinitely. - A non zero value indicates that the master will be preempted after the programmed number of PCI clocks. Once preempted for 10 PCI-CLKs, the 82C825 will once again make a bus request using REQ# if the ISA master is still attempting a transfer.							
PCICFG 0Eh			**Header Type Register**				**Default = 80h**
Multi-function device (RO): 0 = Yes (always)			Layout type for 10-3Fh bytes bits [6:0] = 00h				
PCICFG 0Fh			**BIST Register**				**Default = 00h**
			Not implemented				
PCICFG 10h			**ISA Device Memory Base Address 0 Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
ISA Device Memory Base Address Bits [7:4]				Prefetchable: 0 = No (always)	Type: 00 = Map anywhere (always)		Address space: 0 = Memory (always)
PCICFG 11h			**ISA Device Memory Base Address 0 Register - Byte 1: Address Bits [15:8]**				**Default = 00h**
PCICFG 12h			**ISA Device Memory Base Address 0 Register - Byte 2: Address Bits [23:16]**				**Default = 00h**
PCICFG 13h			**ISA Device Memory Base Address 0 Register - Byte 3: Address Bits [31:24]**				**Default = 00h**
ISA Device Memory Base Address 0 bits [31:24] (downstream memory): - Selects the base system address that will be mapped to address 00 0000h on the ISA slots. - The value, always a multiple of 16MB, is subtracted from the PCI address.							
PCICFG 14h			**ISA Device I/O Base Address 1 Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
ISA Device I/O Base Address Bits: - Register bits [31:4] select the base system address that will be mapped to address 00 0000h on the ISA slots. - For ISA masters, this value will be added to the ISA-generated address when the address is presented on the PCI bus.				Prefetchable: 0 = No (always)	Type: 00 = Map anywhere (always)		Address space: 1 = I/O (always)
PCICFG 15h			**ISA Device I/O Base Address 1 Register - Byte 1: Address Bits [15:8]**				**Default = 00h**

Table 5-1 82C825 Base Register Group: PCICFG 00h-3Fh (cont.)

7	6	5	4	3	2	1	0
PCICFG 16h			**ISA Device I/O Base Address 1 Register - Byte 2: Address Bits [23:16]**				**Default = 00h**
PCICFG 17h			**ISA Device I/O Base Address 1 Register - Byte 3: Address Bits [31:24]**				**Default = 00h**
PCICFG 18h			**ISA Master Device Memory Base Address 2 Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
ISA Master Device Memory Base Address Bits [7:4]: - Reserved				Prefetchable: 0 = No (always)	Type: 00 = Map anywhere (always)		Address space: 0 = Memory (always)
PCICFG 19h			**ISA Master Device Memory Base Address 2 Register - Byte 1: Address Bits [15:8]**				**Default = 00h**
ISA Master Device Memory Base Address Bits [15:8]: Reserved							
PCICFG 1Ah			**ISA Master Device Memory Base Address 2 Register - Byte 2: Address Bits [23:16]**				**Default = 00h**
ISA Master Device Memory Base Address Bits [23:16]: Reserved							
PCICFG 1Bh			**ISA Master Device Memory Base Address 2 Register - Byte 3: Address Bits [31:24]**				**Default = 00h**
ISA Master Device Memory Base Address 2 bits [31:24] (upstream memory): - SelectS the address base that will be added to addresses generated by ISA masters when the address is presented on the PCI bus. The value is always a multiple of 16MB.							
PCICFG 1Ch-27h			**Base Address Registers 3-5**				**Default = 00h**
Not implemented							
PCICFG 28h			**CIS Pointer Register - Byte 0: Bits [7:0]**				**Default = xxh**
CIS Pointer Bits: - Bits [27:3]: The ISA slot CIS is found in the device-dependent configuration space starting at 80h (after the 82C825-specific space). Therefore, this pointer returns a value of 80h (bits [7:3] = 10000b).				Address space indicator: These bits are always 000, indicating a CIS in device-dependent configuration space.			
PCICFG 29h			**CIS Pointer Register - Byte 1: Bits [15:8]**				**Default = 00h**
PCICFG 2Ah			**CIS Pointer Register - Byte 2: Bits [23:16]**				**Default = 00h**
PCICFG 2Bh			**CIS Pointer Register - Byte 3: Bits [31:24]**				**Default = xxh**
ROM image: Not used				Always 0			
PCICFG 2Ch			**Subsystem Vendor Register (RO) - Byte 0: Bits [7:0]**				**Default = 00h**
Subsystem Vendor Bits: - The chipset normally responds to reads of this read-only register with 00h. But if PCICFG 30h[0] = 1, the 82C825 chip reads the data from ROM address 402Ch and returns this value as the vendor number.							
PCICFG 2Dh			**Subsystem Vendor Register (RO) - Byte 1: Bits [15:8]**				**Default = xxh**
PCICFG 2Eh			**Subsystem ID Register (RO) - Byte 0: Bits [7:0]**				**Default = 00h**
Subsystem ID Bits: - The chipset normally responds to reads of this read-only register with 00h. But if PCICFG 30h[0] = 1, the 82C825 chip reads the data from ROM address 402Eh and returns this value as the identifier.							
PCICFG 2Fh			**Subsystem ID Register (RO) - Byte 1: Bits [15:8]**				**Default = xxh**

Table 5-1 82C825 Base Register Group: PCICFG 00h-3Fh (cont.)

7	6	5	4	3	2	1	0
PCICFG 30h			**Expansion ROM Base Address Register - Byte 0: Address Bits [7:0]**				**Default = xxh**
Expansion ROM Address Bits: - Bits [31:15] indicate the address at which the flash EEPROM can be written and read. - Bits [14:1] are read-only and return zero to indicate a 32KB ROM space. - Setting bit [0] = 1 allows ROM to be read, as well as written when enabled (PCICFG 52h[2] = 1). - ROMCS# is a strap-selected option. If disabled, this register is read-only and returns 0 when read.							ROM address decode: 0 = Disable 1 = Enable
PCICFG 31h			**Expansion ROM Base Address Register - Byte 1: Address Bits [15:8]**				**Default = xxh**
PCICFG 32h			**Expansion ROM Base Address Register - Byte 2: Address Bits [23:16]**				**Default = xxh**
PCICFG 33h			**Expansion ROM Base Address Register - Byte 3: Address Bits [31:24]**				**Default = xxh**
PCICFG 34h-3Bh			**Reserved**				**Default = xxh**
PCICFG 3Ch			**Interrupt Line Register**				**Default = 00h**
Not used - always reads FFh.							
PCICFG 3Dh			**Interrupt Pin Register**				**Default = 00h**
Not used - always reads 0.							
PCICFG 3Eh			**Minimum Grant Register (RO)**				**Default = 00h**
Specifies the maximum potential length of burst sequences, in units of 250ns. The 82C825 returns a value of 05h, indicating that bursts can last up to 1.25µs (a 32-clock burst at 25MHz). Software can use this information to enable the most efficient use of 82C825 bandwidth.							
PCICFG 3Fh			**Maximum Latency Register (RO) - Byte 1**				**Default = 00h**
Specifies the maximum latency, in units of 250ns, acceptable to the 82C825 logic after requesting the PCI bus. The 82C825 returns a value of 04h, indicating that latencies of up to 1µs are acceptable. This value primarily reflects the desire for a quick response to IRQ driveback requests.							

Table 5-2 82C825-Specific Register Group: PCICFG 40h-FFh

7	6	5	4	3	2	1	0
PCICFG 40h			**Power Control Latch Register 1 (WO)**				**Default = xxh**
Pin setting: 0 = Low 1 = High	Reserved			PWRx pin to set or clear PPWR0-PPWR15: - PPWR1-4 may be used as RSTDRV1-4 - PPWR0-7 default to high at reset - PPWR8-15 default to low at reset			
PCICFG 41h			**Power Control Latch Register 2**				**Default = xxh**
Reserved							
PCICFG 42h			**Power Control Latch Register 3 (RO)**				**Default = xxh**
PPWR7 writes: 0 = Disable 1 = Enable	PPWR6 value: 0 = Low 1 = High	PPWR5 value: 0 = Low 1 = High	PPWR4 value: 0 = Low 1 = High	PPWR3 value: 0 = Low 1 = High	PPWR2 value: 0 = Low 1 = High	PPWR1 value: 0 = Low 1 = High	PPWR0 value: 0 = Low 1 = High
PCICFG 43h			**Power Control Latch Register 4 (RO)**				**Default = xxh**
PPWR15 value: 0 = Low 1 = High	PPWR14 value: 0 = Low 1 = High	PPWR13 value: 0 = Low 1 = High	PPWR12 value: 0 = Low 1 = High	PPWR11 value: 0 = Low 1 = High	PPWR10 value: 0 = Low 1 = High	PPWR9 value: 0 = Low 1 = High	PPWR8 value: 0 = Low 1 = High
PCICFG 44h			**DMA Channel Selector Register 1**				**Default = 98h**
Channel 1: 0 = Not claimed 1 = On docking ISA	DRQ1/DACK1# pin: 000 = Channel 0 001 = Channel 1 (Default) 010 = Channel 2 011 = Channel 3	 100 = Reserved 101 = Channel 5 110 = Channel 6 111 = Channel 7		Channel 0: 0 = Not claimed 1 = On docking ISA	DRQ0/DACK0# pin: 000 = Channel 0 (Default) 001 = Channel 1 010 = Channel 2 011 = Channel 3	 100 = Reserved 101 = Channel 5 110 = Channel 6 111 = Channel 7	
PCICFG 45h			**DMA Channel Selector Register 2**				**Default = BAh**
Channel 3: 0 = Not claimed 1 = On docking ISA	DRQ3/DACK3# pin: 000 = Channel 0 001 = Channel 1 010 = Channel 2 011 = Channel 3 (Default)	 100 = Reserved 101 = Channel 5 110 = Channel 6 111 = Channel 7		Channel 2: 0 = Not claimed 1 = On docking ISA	DRQ2/DACK2# pin: 000 = Channel 0 001 = Channel 1 010 = Channel 2 (Default) 011 = Channel 3	 100 = Reserved 101 = Channel 5 110 = Channel 6 111 = Channel 7	
PCICFG 46h			**DMA Channel Selector Register 3**				**Default = EDh**
Channel 6: 0 = Not claimed 1 = On docking ISA	DRQ6/DACK6# pin: 000 = Channel 0 001 = Channel 1 010 = Channel 2 011 = Channel 3	 100 = Reserved 101 = Channel 5 110 = Channel 6 (Default) 111 = Channel 7		Channel 5: 0 = Not claimed 1 = On docking ISA	DRQ5/DACK5# pin: 000 = Channel 0 001 = Channel 1 010 = Channel 2 011 = Channel 3	 100 = Reserved 101 = Channel 5 (Default) 110 = Channel 6 111 = Channel 7	
PCICFG 47h			**DMA Channel Selector Register 4**				**Default = 0Fh**
Reserved				Channel 7: 0 = Not claimed 1 = On docking ISA	DRQ7/DACK7# pin: 000 = Channel 0 001 = Channel 1 010 = Channel 2 011 = Channel 3	 100 = Reserved 101 = Channel 5 110 = Channel 6 111 = Channel 7 (Default)	

Table 5-2 82C825-Specific Register Group: PCICFG 40h-FFh (cont.)

7	6	5	4	3	2	1	0
PCICFG 48h			**IRQ Channel Selector Register 1**				**Default = 43h**
IRQ4 pin (Default = IRQ4):				IRQ3 pin (Default = IRQ3):			
0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12	0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12
0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI
0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14	0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
PCICFG 49h			**IRQ Channel Selector Register 2**				**Default = 65h**
IRQ6 pin (Default = IRQ6):				IRQ5 pin (Default = IRQ5):			
0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12	0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12
0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI
0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14	0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
PCICFG 4Ah			**IRQ Channel Selector Register 3**				**Default = 97h**
IRQ9 pin (Default = IRQ9):				IRQ7 pin (Default = IRQ7):			
0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12	0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12
0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI
0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14	0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
PCICFG 4Bh			**IRQ Channel Selector Register 4**				**Default = BAh**
IRQ11 pin (Default = IRQ11):				IRQ10 pin (Default = IRQ10):			
0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12	0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12
0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI
0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14	0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
PCICFG 4Ch			**IRQ Channel Selector Register 5**				**Default = ECh**
IRQ14 pin (Default = IRQ14):				IRQ12 pin (Default = IRQ12):			
0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12	0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12
0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI
0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14	0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
PCICFG 4Dh			**IRQ Channel Selector Register 6**				**Default = DFh**
HDI pin (Default = NMI):				IRQ15 pin (Default = IRQ15):			
0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12	0000 = IRQ0	0100 = IRQ4	1000 = IRQ8#	1100 = IRQ12
0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = NMI
0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14	0010 = SMI#	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
PCICFG 4Eh-4Fh			**Reserved**				**Default = 00h**

Table 5-2 82C825-Specific Register Group: PCICFG 40h-FFh (cont.)

7	6	5	4	3	2	1	0
PCICFG 50h			**Hot Docking Control Register**				**Default = xxh**
Reserved	HDI# pin has made low-to-high transition? 0 = No 1 = Yes Write 1 to clear	HDI# pin has made high-to-low transition? 0 = No 1 = Yes Write 1 to clear	Hot insertion attempt failed? 0 = No 1 = Yes Write 1 to clear	Hot IDE insertion failure determination: Selects number of stabilization checks to be performed before hot insertion success is determined. 00 = No retry attempts 01 = 16 attempts 10 = 256 attempts 11 = 1024 attempts		HDI# input stabilization check: Selects duration of sample period during which HDI# must remain stable to be considered a successful check. 00 = 1ms 01 = 10ms 10 = 100ms 11 = 1s	
PCICFG 51h			**ISA Control Register**				**Default = 00h**
Reserved		ISA refresh: 0 = Disable 1 = Enable	PCI SERR# generates NMI? 0 = No 1 = Yes	Reserved	ISA CHCK# NMI: 0 = Disable 1 = Enable	ATCLK source: 00 = 14MHz/2 (Default) 01 = PCICLK/3 10 = PCICLK/4 11 = PCICLK/5	
PCICFG 52h			**Feature Control Register**				**Default = 21h**
PPWRL function on RSTDRV: 0 = Disable (use pin only as RSTDRV) 1 = Enable PPWRL	Individual AEN interface: 0 = Disable 1 = Enable (DACK-MUX strap option required)	IDE controller interface: 0 = Disable (PCIIDE function will not respond) 1 = Enable (Default)	IDE control pin function select: 0 = DRD#, DWR# 1 = DBE0#, DBE1#	CHCK# pin active halts ISA bus: 0 = No (normal CHCK#) 1 = Yes (HDI# function)	Flash EEPROM writes: 0 = Disable (SMWR# is blocked also) 1 = Enable	RQ8# active level: 0 = Low 1 = High	DMA register access: 0 = ISA mapping 1 = Distributed DMA mapping (Default)
PCICFG 53h			**SMI Status Register**				**Default = 00h**
ROMCS# pin status during reset: 0 = Low 1 = High	DRD# pin status during reset: 0 = Low 1 = High	DWR# pin status during reset: 0 = Low 1 = High	Reserved			PPWR cycle status: 0 = Complete 1 = Pending	IRQ pin generated SMI? 0 = No 1 = Yes Write 1 to clear
PCICFG 54h			**IRQ Driveback Protocol Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
IRQ Driveback Protocol Address: - When the 82C825 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. - This register defaults to a value of 0h, which disables the IRQ driveback scheme.							
PCICFG 55h			**IRQ Driveback Protocol Address Register - Byte 1: Address Bits [15:8]**				**Default = xxh**
PCICFG 56h			**IRQ Driveback Protocol Address Register - Byte 2: Address Bits [23:16]**				**Default = xxh**
PCICFG 57h			**IRQ Driveback Protocol Address Register - Byte 3: Address Bits [31:24]**				**Default = xxh**

Table 5-2 82C825-Specific Register Group: PCICFG 40h-FFh (cont.)

7	6	5	4	3	2	1	0
PCICFG 58h			**DRQ Remap Base Address Register - Byte 0: Address Bits [7:0]**				**Default = 00h**
IRQ Remap Base Address: - 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. The 82C825 logic uses this base address to claim accesses to an ISA DMA controller channel.							
PCICFG 59h			**DRQ Remap Base Address Register - Byte 1: Address Bits [15:8]**				**Default = xxh**
PCICFG 5Ah			**DRQ Remap Base Address Register - Byte 2: Address Bits [23:16]**				**Default = xxh**
PCICFG 5Bh			**DRQ Remap Base Address Register - Byte 3: Address Bits [31:24]**				**Default = xxh**
PCICFG 5Ch			**ISA Slot Control Register 1**				**Default = 00h**
AEN2 high for x000-x3F8h: 0 = No 1 = Yes	AEN2 enabled (low) for I/O range: 000 = 0000-FFFFh (normal AEN) 001 = 1000-1FFFh 010 = 2000-2FFFh 011 = 3000-3FFFh 100 = 4000-4FFFh 101 = 5000-5FFFh 110 = 6000-6FFFh 111 = 7000-7FFFh			AEN1 high for x000-x3F8h: 0 = No 1 = Yes	AEN1 enabled (low) for I/O range: 000 = 0000-FFFFh (normal AEN) 001 = 1000-1FFFh 010 = 2000-2FFFh 011 = 3000-3FFFh 100 = 4000-4FFFh 101 = 5000-5FFFh 110 = 6000-6FFFh 111 = 7000-7FFFh		
PCICFG 5Dh			**ISA Slot Control Register 2**				**Default = 00h**
Reserved				AEN3 high for x000-x3F8h: 0 = No 1 = Yes	AEN3 enabled (low) for I/O range: 000 = 0000-FFFFh (normal AEN) 001 = 1000-1FFFh 010 = 2000-2FFFh 011 = 3000-3FFFh 100 = 4000-4FFFh 101 = 5000-5FFFh 110 = 6000-6FFFh 111 = 7000-7FFFh		
PCICFG 5Eh			**82C825 Write Posting Control Register**				**Default = 00h**
Reserved: Write as read.				Reserved	Retry Attempts on Posted Writes: These bits relate to the number of retries allowed to deliver posted data before SERR# is generated on the initiator side. 000 = 2^8 101 = 2^{20} 001 = 2^{10} 100 = 2^{16} 010 = 2^{12} 110 = 2^{24} 011 = 2^{14} 111 = Infinite retries		
PCICFG 5Fh			**82C825 Write Posting Status Register (RO)**				**Default = 00h**
This register returns the number of retry attempts made to write posted data to the target. More than 256 retries are indicated by FFh. Used for diagnostic purposes.							
PCICFG 60h-FFh			**Reserved**				**Default = 00h**

5.2 Function 1: Bus Master IDE Registers

The 82C825 chip incorporates the identical bus mastering IDE logic of the OPTi Viper-M Chipset. The register set is shown below. A device-specific register set is appended to this register set in order to select certain 82C825-specific functions.

Table 5-3 82C825 Bus Master IDE Registers: PCIIDE 00h-FFh

7	6	5	4	3	2	1	0
PCIIDE 00h			**Vendor Identification Register (RO) - Byte 0**				**Default = 45h**
PCIIDE 01h			**Vendor Identification Register (RO) Byte 1**				**Default = 10h**
PCIIDE 02h			**Device ID (RO) - Byte 0**				**Default = 21h**
PCIIDE 03h			**Device ID (RO) - Byte 1**				**Default = C6h**
PCIIDE 04h			**PCI Command Register - Byte 0**				**Default = 01h**
Address/data stepping: 0 = Disable (always)	PERR# generation: 0 = Disable 1 = Enable	VGA palette snoop: 0 = Disable (always)	Mem write and Invalidate (RO): 0 = Disable (always)	Special Cycle (RO): 0 = Disable (always)	Bus master by slot interfaces: 0= Disable 1 = Enable	Respond to PCI memory accesses: 0 = No (always)	Respond to PCI I/O accesses: 0 = No 1 = Yes
PCIIDE 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 (always)
PCIIDE 06h			**PCI Status Register - Byte 0**				**Default = 00h**
Fast back-to-back capability (RO): 0 = No 1 = Yes	Reserved (RO)						
PCIIDE 07h			**PCI Status Register - Byte 1**				**Default = 02h**
Parity error: 0 = No 1 = Yes Write 1 to clear	System error: 0 = No (always)	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 (always)	DEVSEL# timing (RO): 00 = Fast 01 = Medium (always) 10 = Slow 11 = Reserved		PERR# active as master: 0 = No 1 = Yes Write 1 to clear
PCIIDE 08h			**Revision ID Register (RO)**				**Default = 00h**
PCIIDE 09h			**Programming Interface Class Code Register (RO)**				**Default = 00h**
PCIIDE 0Ah			**Class Code Register (RO) - Byte 0** Subclass Code Bits: = 01h (IDE Controller)				**Default = 01h**
PCIIDE 0Bh			**Class Code Register (RO) - Byte 1** Base Class Code Bits: = 01h (Mass Storage)				**Default = 06h**
PCIIDE 0Ch			**Cache Line Size Register** Indicates the number of transfers per burst to fill/empty the FIFO.				**Default = 00h**

Table 5-3 82C825 Bus Master IDE Registers: PCIIDE 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIIDE 0Dh			**Latency Timer Register** Indicates the time-out value.				**Default = 00h**
PCIIDE 0Eh			**Header Type Register**				**Default = 80h**
Multi-function device (RO): 0 = Yes (always)			Layout type for 10-3Fh bytes bits [6:0] = 00h				
PCIIDE 0Fh			**BIST Register** Not implemented				**Default = 00h**
PCIIDE 10h			**Cable 0 IDE Command Block Base Address Register 0 - Byte 0: Address Bits [7:0]**				**Default = 00h**
PCIIDE 11h			**Cable 0 IDE Command Block Base Address Register 0 - Byte 1: Address Bits [15:8]**				**Default = xxh**
PCIIDE 12h			**Cable 0 IDE Command Block Base Address Register 0 - Byte 2: Address Bits [23:16]**				**Default = xxh**
PCIIDE 13h			**Cable 0 IDE Command Block Base Address Register 0 - Byte 3: Address Bits [31:24]**				**Default = xxh**
PCIIDE 14h			**Cable 0 IDE Control Block Base Address Register 1 - Byte 0: Address Bits [7:0]**				**Default = 00h**
PCIIDE 15h			**Cable 0 IDE Control Block Base Address Register 1 - Byte 1: Address Bits [15:8]**				**Default = xxh**
PCIIDE 16h			**Cable 0 IDE Control Block Base Address Register 1 - Byte 2: Address Bits [23:16]**				**Default = xxh**
PCIIDE 17h			**Cable 0 IDE Control Block Base Address Register 1 - Byte 3: Address Bits [31:24]**				**Default = xxh**
PCIIDE 18h			**Cable 1 IDE Command Block Base Address Register 2 - Byte 0: Address Bits [7:0]**				**Default = 00h**
PCIIDE 19h			**Cable 1 IDE Command Block Base Address Register 2 - Byte 1: Address Bits [15:8]**				**Default = xxh**
PCIIDE 1Ah			**Cable 1 IDE Command Block Base Address Register 2 - Byte 2: Address Bits [23:16]**				**Default = xxh**
PCIIDE 1Bh			**Cable 1 IDE Command Block Base Address Register 2 - Byte 3: Address Bits [31:24]**				**Default = xxh**
PCIIDE 1Ch			**Cable 1 IDE Control Block Base Address Register 3 - Byte 0: Address Bits [7:0]**				**Default = 00h**
PCIIDE 1Dh			**Cable 1 IDE Control Block Base Address Register 3 - Byte 1: Address Bits [15:8]**				**Default = xxh**
PCIIDE 1Eh			**Cable 1 IDE Control Block Base Address Register 3 - Byte 2: Address Bits [23:16]**				**Default = xxh**
PCIIDE 1Fh			**Cable 1 IDE Control Block Base Address Register 3 - Byte 3: Address Bits [31:24]**				**Default = xxh**
PCIIDE 20h			**Bus Master IDE Base Address Register 4 - Byte 0: Address Bits [7:0]**				**Default = 00h**
PCIIDE 21h			**Bus Master IDE Base Address Register 4 - Byte 1: Address Bits [15:8]**				**Default = xxh**
PCIIDE 22h			**Bus Master IDE Base Address Register 4 - Byte 2: Address Bits [23:16]**				**Default = xxh**
PCIIDE 23h			**Bus Master IDE Base Address Register 4 - Byte 3: Address Bits [31:24]**				**Default = xxh**
PCIIDE 24h-27h			**Base Address Register 5** Not implemented				**Default = xxh**
PCIIDE 28h-2Bh			**CardBus CIS Pointer Register** Not implemented				**Default = xxh**

Table 5-3 82C825 Bus Master IDE Registers: PCIIDE 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIIDE 2Ch			**Subsystem Vendor Register (RO) - Byte 0: Bits [7:0]**				**Default = 00h**
Subsystem Vendor Bits: - The chipset normally responds to reads of this read-only register with 00h. But if PCICFG 30h[0] = 1, the 82C825 chip reads the data from ROM address 402Ch and returns this value as the vendor number.							
PCIIDE 2Dh			**Subsystem Vendor Register (RO) - Byte 1: Bits [15:8]**				**Default = xxh**
PCIIDE 2Eh			**Subsystem ID Register (RO) - Byte 0: Bits [7:0]**				**Default = 00h**
Subsystem ID Bits: - The chipset normally responds to reads of this read-only register with 00h. But if PCICFG 30h[0] = 1, the 82C825 chip reads the data from ROM address 402Eh and returns this value as the identifier.							
PCIIDE 2Fh			**Subsystem ID Register (RO) - Byte 1: Bits [15:8]**				**Default = xxh**
PCIIDE 30h-33h			**Expansion ROM Base Address Register** Not used				**Default = xxh**
PCIIDE 34h-3Bh			**Reserved**				**Default = xxh**
PCIIDE 3Ch			**Interrupt Line Register** Not used - always reads FFh.				**Default = 00h**
PCIIDE 3Dh			**Interrupt Pin Register** Not used - always reads 0.				**Default = 00h**
PCIIDE 3Eh			**Minimum Grant Register (RO)** Not implemented				**Default = 00h**
PCIIDE 3Fh			**Maximum Latency Register (RO)** Not implemented				**Default = 00h**
PCIIDE 40h			**IDE Initialization Control Register**				**Default = xxh**
Reserved		FIFO mode: 0 = 16-byte 1 = 32-byte	ISA parking: 0 = When needed 1 = Always	Secondary IDE: 0 = Enable 1 = Disable	Address relocation: 0 = Fixed (always)	PIO Mode: 00 = Mode 0 (600ns) 01 = Mode 1 (380ns) 10 = Mode 2 (240ns) 11 = Mode 3 (180ns)	
PCIIDE 41h-42h			**Reserved**				**Default = xxh**
PCIIDE 43h			**IDE Enhanced Mode Register**				**Default = xxh**
Secondary Drive 1 Mode: 00 = Controlled by timing registers set 01 = PIO 4/DMA 1 10 = PIO 5/DMA 2 11 = Reserved		Secondary Drive 0 Mode: 00 = Controlled by timing registers set 01 = PIO 4/DMA 1 10 = PIO 5/DMA 2 11 = Reserved		Primary Drive 1 Mode: 00 = Controlled by timing registers set 01 = PIO 4/DMA 1 10 = PIO 5/DMA 2 11 = Reserved		Primary Drive 0 Mode: 00 = Controlled by timing registers set 01 = PIO 4/DMA 1 10 = PIO 5/DMA 2 11 = Reserved	
PCIIDE 44h			**Reserved**				**Default = xxh**

Table 5-3 82C825 Bus Master IDE Registers: PCIIDE 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIIDE 45h			**IDE Interrupt Selection Register**				**Default = xxh**
Secondary Drive 1 interrupt pin: 00 = IRQ10/PCIRQ0# 01 = IRQ11/PCIRQ1# 10 = IRQ14/PCIRQ2# 11 = IRQ15/PCIRQ3#		Secondary Drive 0 interrupt pin: 00 = IRQ10/PCIRQ0# 01 = IRQ11/PCIRQ1# 10 = IRQ14/PCIRQ2# 11 = IRQ15/PCIRQ3#		Primary Drive 1 interrupt pin: 00 = IRQ10/PCIRQ0# 01 = IRQ11/PCIRQ1# 10 = IRQ14/PCIRQ2# 11 = IRQ15/PCIRQ3#		Primary Drive 0 interrupt pin: 00 = IRQ10/PCIRQ0# 01 = IRQ11/PCIRQ1# 10 = IRQ14/PCIRQ2# 11 = IRQ15/PCIRQ3#	
Note: ISA IRQ is selected for Legacy mode; PCI IRQ is selected for Native mode (see PCIIDE 09h).							
PCIIDE 46h-FFh			**Reserved**				**Default = xxh**

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

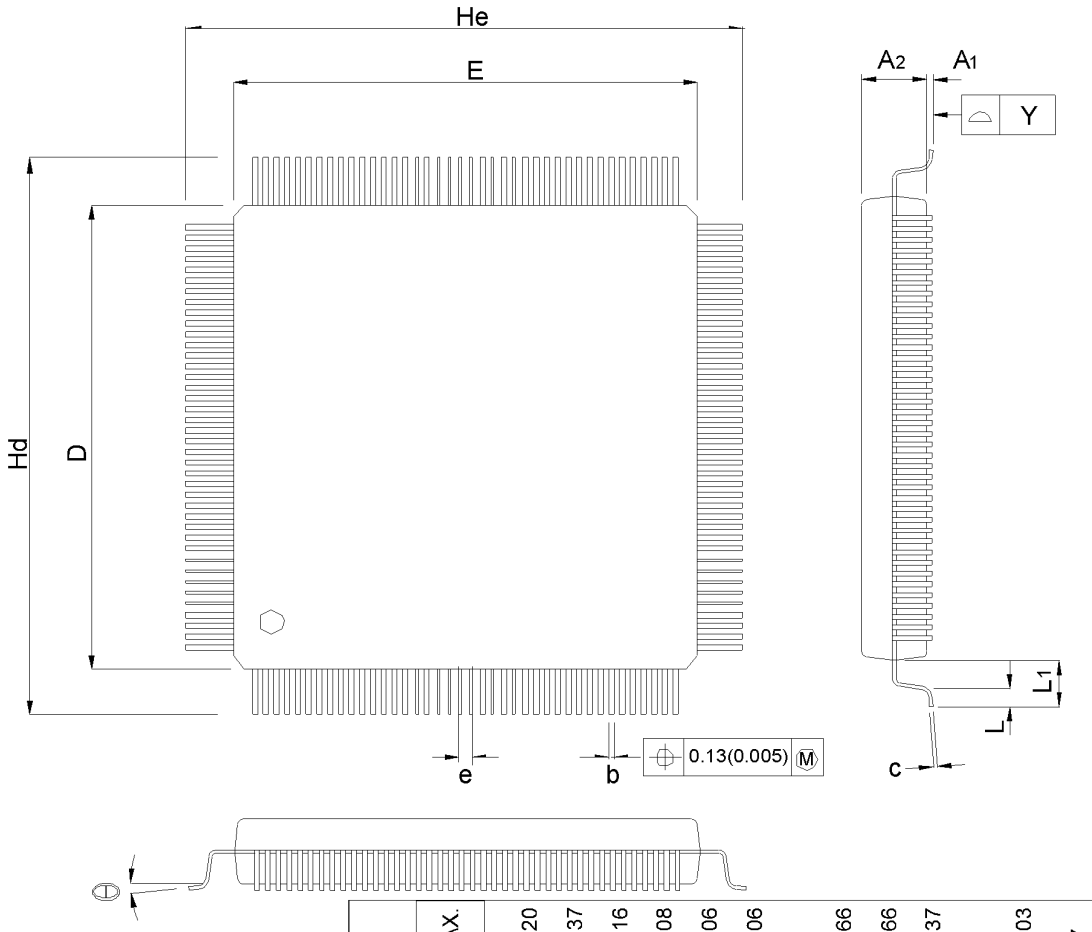
Symbol	Parameter	Min	Max	Unit	Condition
VIL	Input low Voltage	–0.5	+0.8	V	
VIH	Input high Voltage	+2.0	VCC + 0.5	V	
VOL	Output low Voltage		+0.4	V	IOL = 4.0mA
VOH	Output high Voltage	+2.4		V	IOH = –1.6mA
IIL	Input Leakage Current		+10.0	µA	VIN = VCC
IOZ	Tristate Leakage Current		+10.0	µA	
CIN	Input Capacitance		+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 (TBD)

6.4 AC Timing Diagrams (TBD)

7.0 Mechanical Package Outline

Figure 7-1 160-Pin Plastic Quad Flat Pack (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	0.20	0.30	0.40	0.008	0.012	0.016
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		0.65			0.026	
Hd	31.65	31.90	32.15	1.246	1.256	1.266
He	31.65	31.90	32.15	1.246	1.256	1.266
L	0.65	0.80	0.95	0.025	0.031	0.037
L_1		1.95			0.077	
Y			0.08			0.003
①	0		7	0		7

Dwg. No.: AS160PQFP-001
Dwg. Rev.: A0 Unit: MM / INCH

Appendix A IRQ Driveback Protocol

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

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

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

A.1 Driveback Cycle Format

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

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

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

Figure A-1 IRQ Driveback Cycle High-Priority Request

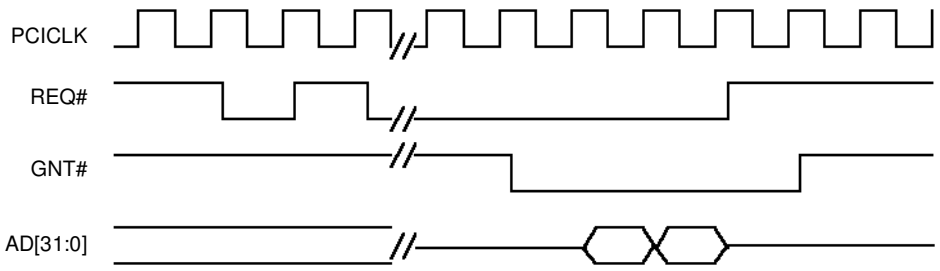


Table A-1 Information Provided on a Driveback Cycle

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

There is a convention for assignment of otherwise unusable IRQs:

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

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

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

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

A.2 Edge vs Level Mode, IRQ Polarity

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

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

A.3 Host Handling of IRQ Driveback Information

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

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

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

A.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 A-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 A-2 Dynamic Resourcing

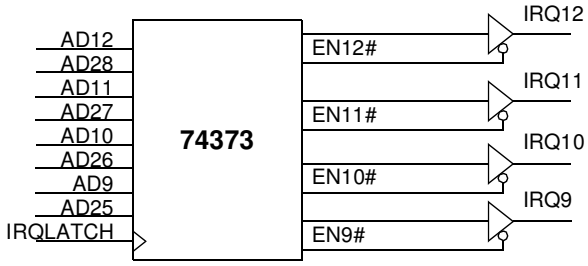
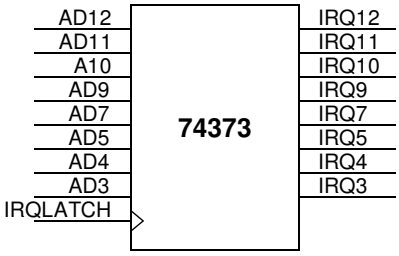


Figure A-3 Static Resourcing



Appendix B 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.

B.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.

B.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.

B.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.

B.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 B-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.							

B.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.

B.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 B-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.

B.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 B-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 B-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 B-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 B-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)			

2.3.4.1 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.

2.3.4.2 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.

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/