Viper-MAX Chipset
Scaleable MultiMedia PC Solution

1.0 Features

CPU Interface

- Fully supports all Intel® 3.3V Pentium™ processors (P54C™, P55C™, P55CT™) at 50, 60, and 66.667MHz
- Supports AMD® K86™ and Cyrix® 6x86 processors
- Supports the Cyrix 6x86 processor linear burst mode
- Chipset solution:
  - One Data Buffer Controller (82C566)
  - One System Controller (82C567)
  - One Integrated Peripherals Controller (82C568)
- Supports CPU address pipelining

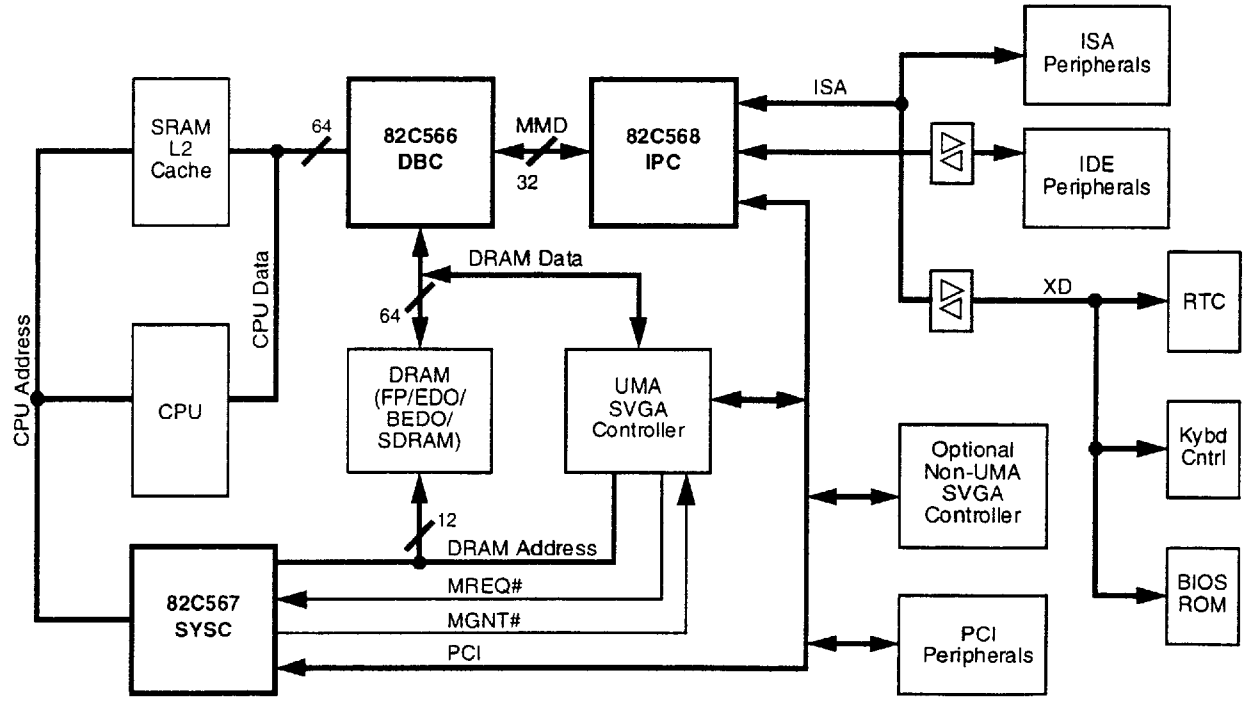
Cache Interface

- Support four types of devices:
  - Synchronous SRAM bursting at 3-1-1-1
  - Pipelined burst SRAM bursting at 3-1-1-1
  - Sony SONIC-2WP™ module bursting at 2-1-1-1
  - Asynchronous SRAM bursting at 3-2-2-2
- Supports four cache sizes:
  - 256KB, 512KB, 1MB, and 2MB
- Programmable write policy:
  - Write-back
  - Adaptive write-back
  - Write-through

DRAM Interface

- Supports both Unified Memory Architecture (UMA) and non-UMA interfaces
- Supports symmetrical and asymmetrical DRAMs
- Supports both 3.3V and 5.0V DRAM devices
- Supports 64-bit wide DRAM devices with 256KB, 512KB, 1MB, 2MB, 4MB, 8MB, and 16MB addressing
- Supports DRAM configurations up to 512MB
- Six banks of FP mode DRAMs (7-3-3-3 at 66MHz)
- Six banks of EDO DRAM support with auto detection (5-2-2-2 at 66MHz)
- Four banks of BEDO (burst EDO) (X-1-1-1 at 66MHz)
- Four banks of SDRAM (synchronous DRAM) (X-1-1-1 at 66MHz)

Figure 1-1 Viper-MAX System Block Diagram



Features (cont.)

- Deep buffering for DRAM performance
  - Six quad-word CPU-to-DRAM write posting
  - 24 double-word PCI-to-DRAM write posting
  - 24 double-word DRAM-to-PCI read prefetch
- Supports mixed DRAM memory technologies
  - FP mode/EDO/SDRAM
  - FP mode/EDO/BEDO
- Memory parity support
- Programmable drive currents for the DRAM control signals
- Hidden refresh with CAS-before-RAS refresh supported
- Self-refresh supported during Suspend mode
- Support for two programmable non-cacheable memory regions

Unified Memory Interface

- Industry Standard UMA implementation
- Compatible with all major graphics chip vendors
- Supports 0.5, 1.0, 2.0, 3.0, and 4.0MB of shared frame buffer for GUI (Graphical User Interface) within system DRAM
- Two-pin arbitration scheme with multiple request levels

PCI Interface

- PCI Specification 2.1 compliant
  - Supports delayed transactions
- X-1-1-1 PCI to memory burst transfer performance (transfer rate > 100MB/sec)
- Interfaces the CPU and standard buses to Peripheral Component Interconnect (PCI) operating in synchronous/asynchronous modes
- CPU-to-PCI deep write posting buffers (six double-words)
- PCI-to-DRAM deep write posting and read prefetch buffers (24 double-words)
- Supports five PCI masters and six ISA slots
- Supports PCI pre-snoop for PCI masters
- PCI byte/word merge support for CPU accesses to PCI bus, and support for PCI prefetch
- Several levels of concurrence
  - PCI-to-PCI / CPU-to-memory
  - PCI-to-DRAM / CPU-to-cache
  - CPU-to-PCI / GUI-to-memory

IDE Interface

- Integrated bus master IDE conforms to SFF Specification
- Two channels supported (up to four devices)
- PIO Mode transfer support (up to Mode 5)
- Enhanced ATA Specification support
- Single- and/or Multi-Word DMA Mode 2 timing
- Scatter/Gather feature
- Built-in FIFOs with data prefetch and post write support

Universal Serial Bus

- Support for Universal Serial Bus (USB) interface for serial peripherals

System I/O and Power Management

- Enhanced DMA interface
  - Type F DMA for faster device transfer
  - Distributed DMA for moving ISA function to PCI
  - Buffered DMA for efficient PCI/DRAM bandwidth
  - Two steerable DMA channels for motherboard plug and play
- Enhanced interrupt interface
  - Serial interrupt for moving ISA function to PCI
  - Two steerable interrupts for motherboard plug and play
- Includes a fully integrated 82C206 with external real-time clock (RTC) interface
- Facility to read current CMOS index
- "True" GREEN power management support, with support for STPCLK# modulation and the CPU stop clock state
- Packaged in three 208-pin PQFPs (Plastic Quad Flat Packs)

2.0 Overview

The OPTi Viper-MAX Chipset provides a highly integrated solution for fully compatible, high performance PC/AT platforms based on the Intel® 3.3V Pentium™ Processor, Cyrix® 6x86™ Processor, and AMD® K86™ Processor. As the latest member of the Desktop Viper Chipset Family, it is designed from its inception to be the highest performance Pentium chipset ever. Its feature set can be scaled to address entry level UMA-based system to high-end non-UMA work stations and servers. The deep buffers in the Viper-MAX minimize system level latencies and maximize through-puts to both DRAM and PCI subsystems.

The chipset provides 64-bit core logic, with Unified Memory Architecture (UMA), and integrated PCI support, plus sophisticated power management features. This highly integrated approach supplies the foundation for a cost effective platform without compromising performance. Its feature set furnishes an array of control and status monitoring options that are accessed through a simple and straightforward interface. All major BIOS vendors provide extensive software hooks that allow system designers to integrate their own special features with minimal effort.

2.1 82C566 Data Buffer Controller

The 82C566 performs the task of buffering the CPU to the DRAM memory data path.

- CPU to memory data buffer
- CPU to PCI local bus buffer
- Memory to PCI local bus buffer
- 208-pin PQFP

Figure 2-1 shows a logic block diagram of the 82C566.

2.2 82C567 System Controller

The 82C567 provides the control functions for the host CPU interface, the 64-bit Level-2 (L2) cache, the 64-bit DRAM bus, and the PCI interface. The 82C567 controls the data flow between the CPU bus, the DRAM bus, the local buses, and the 8/16-bit ISA bus. It interprets and translates cycles from the CPU, PCI bus master, ISA master, and DMA to the host memory, PCI bus slave, or ISA bus devices. The 82C567 also serves as the UMA (Unified Memory Architecture) and USB (Universal Serial Bus) protocol interface.

- 3.3V CPU interface
- DRAM controller
- L1/L2 cache controller
- UMA arbiter
- USB interface
- PCI interface
- Arbitration logic
- Data bus buffer control (memory data bus to and from host data bus)
- 208-pin PQFP

Figure 2-2 shows a logic block diagram of the 82C567 and Figure 2-3 shows the UMA protocol.

Figure 2-1 82C566 Logic Block Diagram

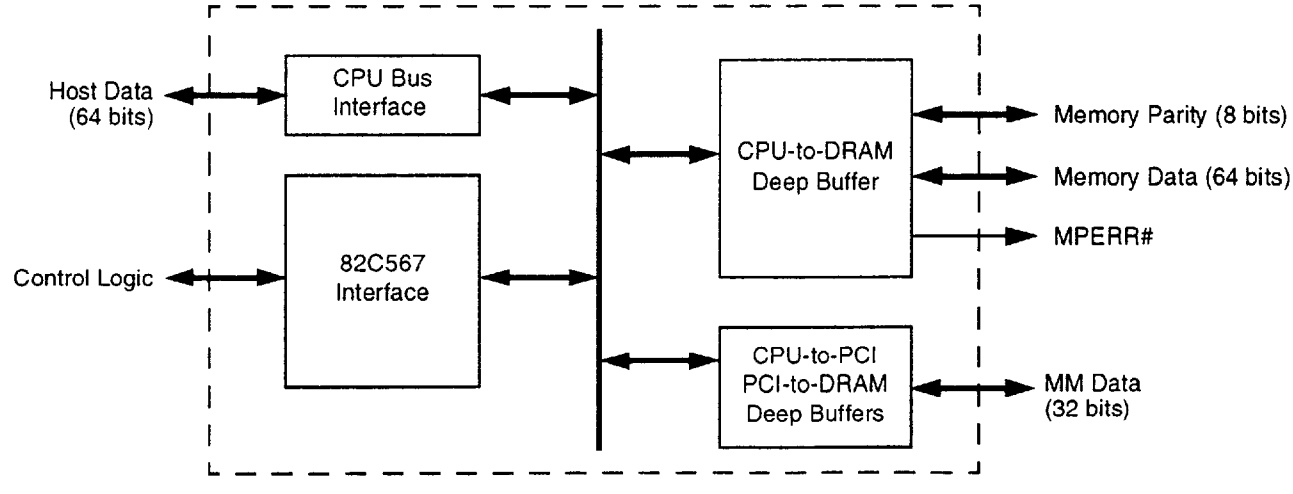


Figure 2-2 82C567 Logic Block Diagram

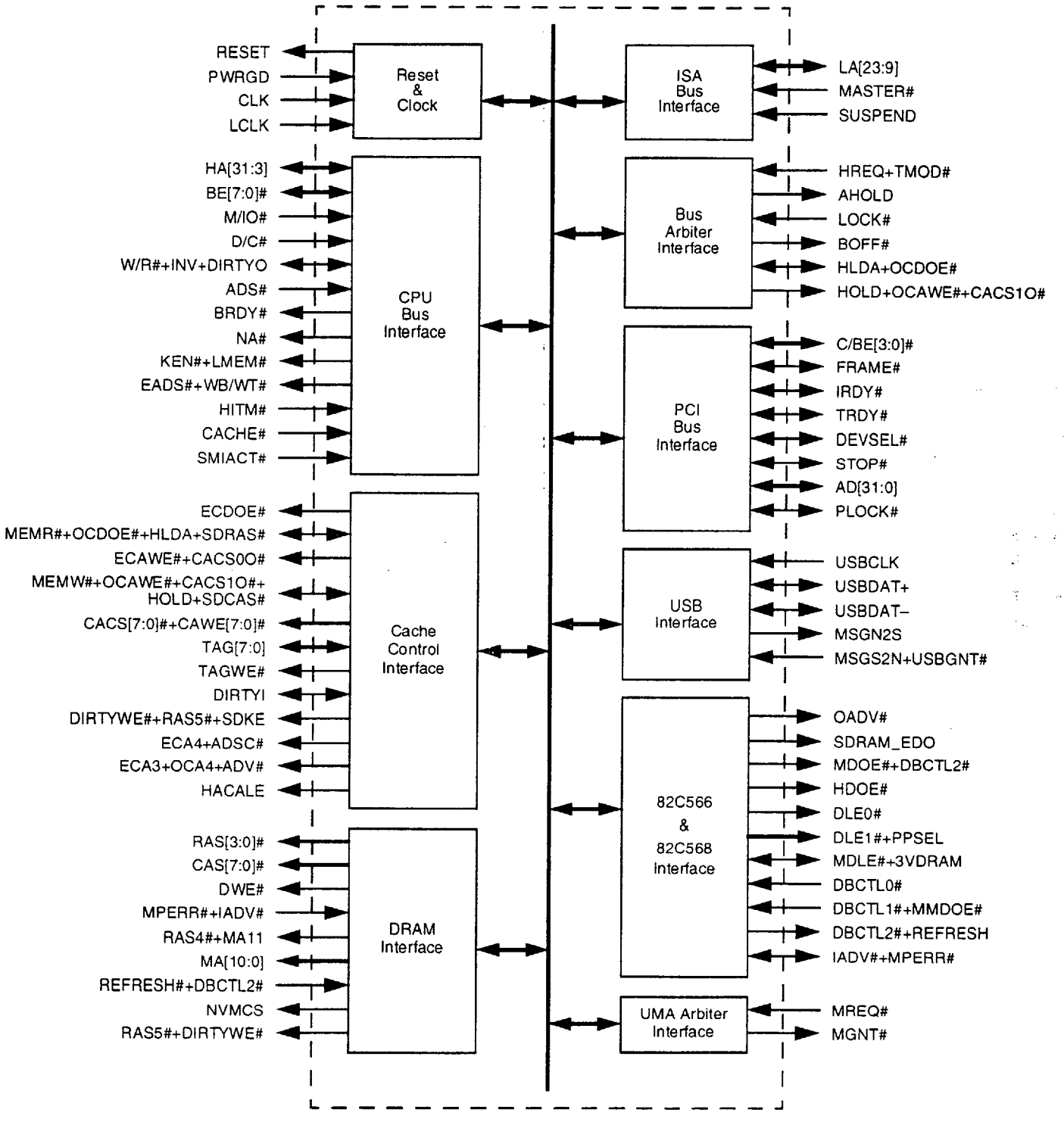
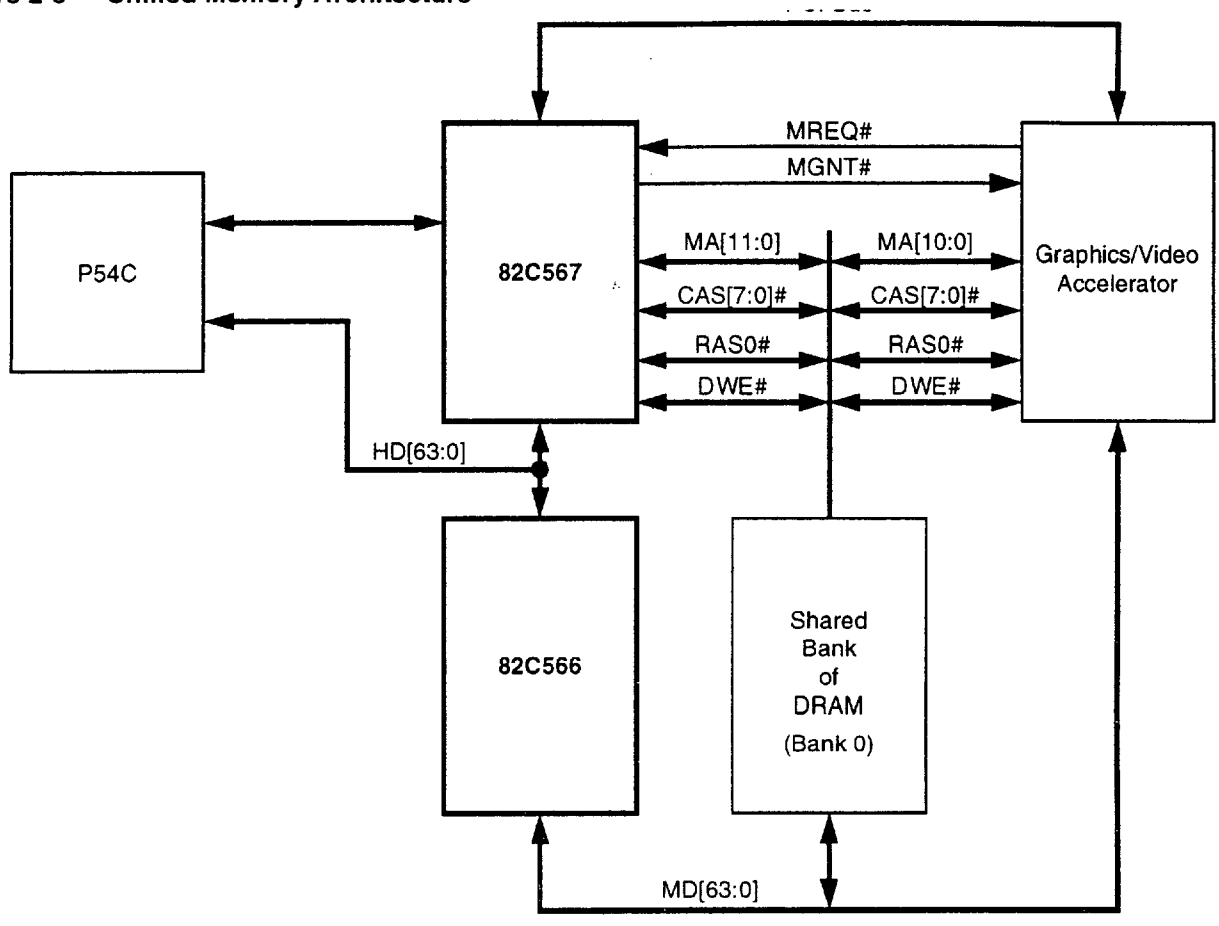


Figure 2-3 Unified Memory Architecture



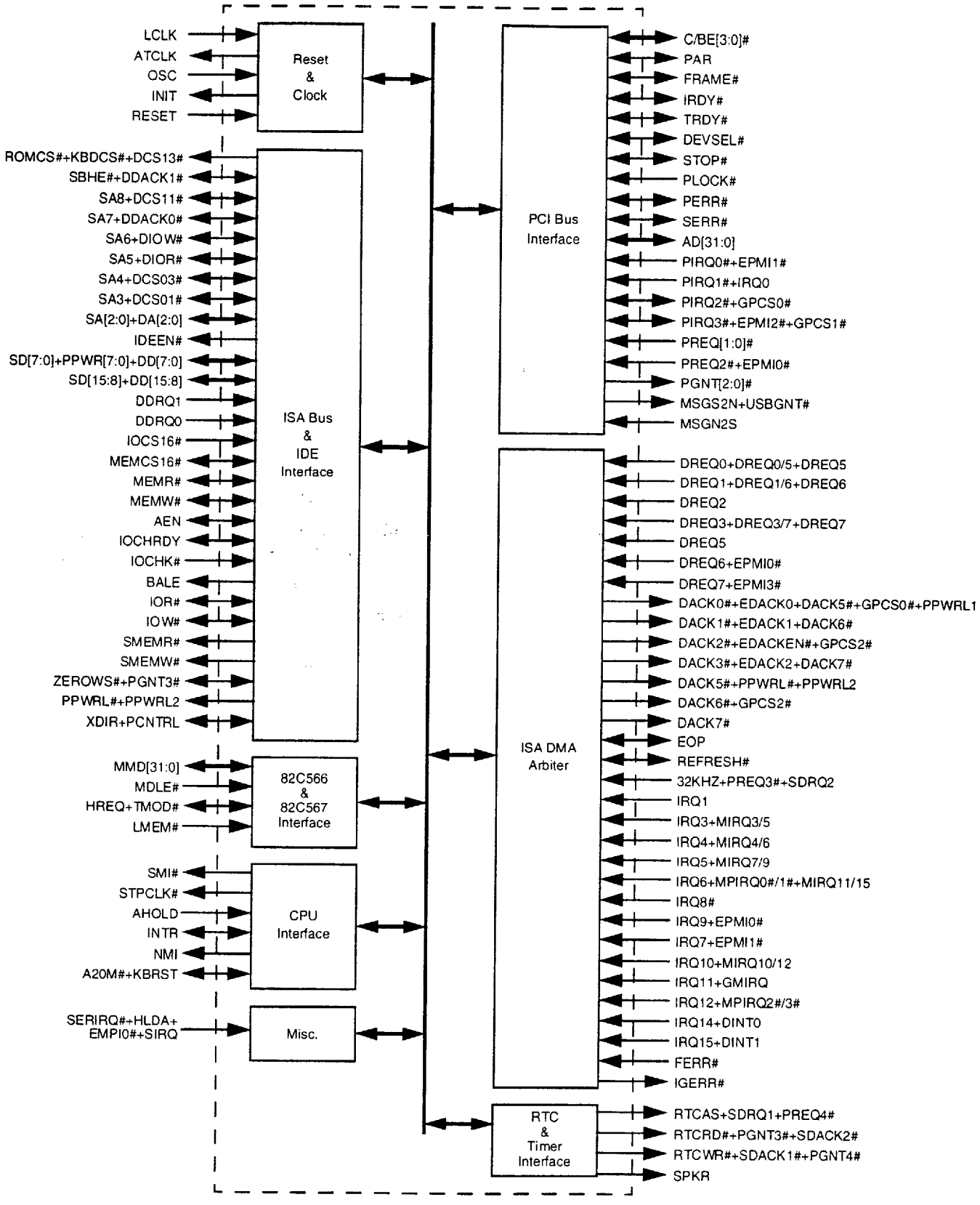
2.3 82C568 Integrated Peripherals Controller

The 82C568 contains the ISA bus controller and includes an 82C206, RTC interface, DMA controller, serial interrupt controller and distributed DMA. It also has a sophisticated system power management unit.

- ISA bus controller
- Master mode IDE
- Type F DMA support
- Integrated 82C206 IPC
- System power management functions
- PCI local bus interface
- PCI to ISA expansion bridge
- Serial interrupt controller
- Distributed DMA
- Keyboard emulation of A20M# and CPU warm reset
- Port B and Port 092h Register
- 208-pin PQFP

Figure 2-4 shows a logic block diagram of the 82C568.

Figure 2-4 82C568 Logic Block Diagram



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.

Some of the pin functions in the Viper-MAX Chipset are time-multiplexed, some have strap options, and some are selected via register programming. Included in each device's signal description is a column titled "Selected By" which explains how to implement/invoke the various functions that a pin may have. The terms PCIDV0, PCIDV1, and SYSCFG relate to registers located in the PCI Configuration Register Spaces and System Configuration Register Space of the Viper-MAX Chipset. Refer to Section 5.0, "Register Descriptions" for more details regarding these register spaces and their access mechanisms.

The tables in this section use several common abbreviations. Table 3-1 lists the mnemonics and their meanings. Note that TTL/CMOS/Schmitt-trigger levels pertain to inputs only. Outputs are driven at CMOS levels.

Table 3-1 Signal Definitions Legend

Mnemonic	Description
Analg	Analog-level compatible
CMOS	CMOS-level compatible
Dcdr	Decoder
Ext	External
G	Ground
Int	Internal
I/O	Input/Output
Mux	Multiplexer
OD	Open drain
P	Power
PD	Pull-down resistor
PU	Pull-up resistor
S	Schmitt-trigger
TTL	TTL-level compatible
VCC3	3.3V power plane
VCC5	5.0V power plane
VCC_AT	ISA bus power plane
VCC_MEM	Memory power plane

Figure 3-1 82C566 Pin Diagram

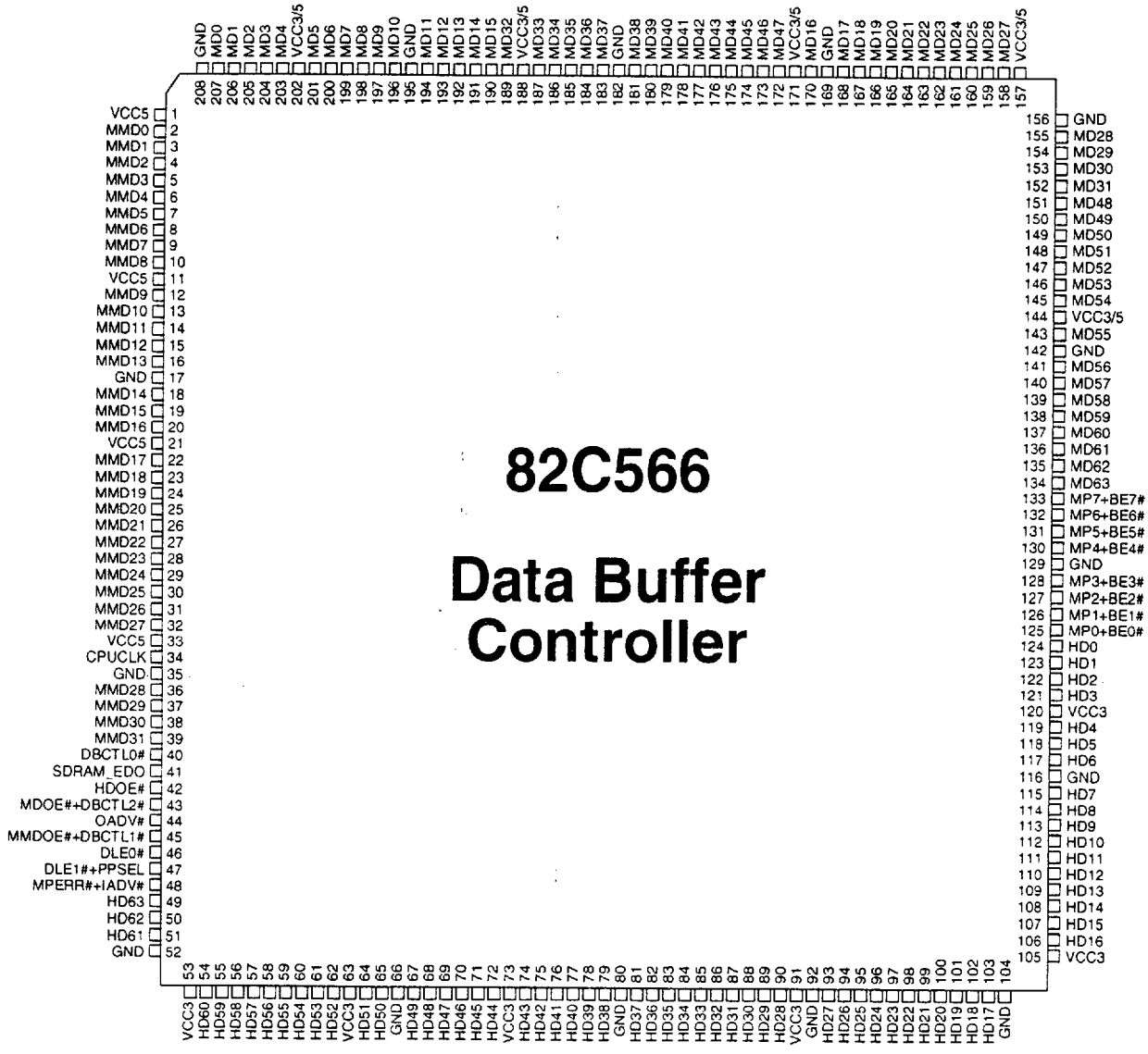


Table 3-2 82C566 Numerical Pin Cross-Reference List

Pin No.	Pin Name	Pin Type	Drive (mA)	Pwr (V)
1	VCC5	I-P		
2	MMD0	I/O-CMOS	4	5.0
3	MMD1	I/O-CMOS	4	5.0
4	MMD2	I/O-CMOS	4	5.0
5	MMD3	I/O-CMOS	4	5.0
6	MMD4	I/O-CMOS	4	5.0
7	MMD5	I/O-CMOS	4	5.0
8	MMD6	I/O-CMOS	4	5.0
9	MMD7	I/O-CMOS	4	5.0
10	MMD8	I/O-CMOS	4	5.0
11	VCC5	I-P		
12	MMD9	I/O-CMOS	4	5.0
13	MMD10	I/O-CMOS	4	5.0
14	MMD11	I/O-CMOS	4	5.0
15	MMD12	I/O-CMOS	4	5.0
16	MMD13	I/O-CMOS	4	5.0
17	GND	I-G		
18	MMD14	I/O-CMOS	4	5.0
19	MMD15	I/O-CMOS	4	5.0
20	MMD16	I/O-CMOS	4	5.0
21	VCC5	I-P		
22	MMD17	I/O-CMOS	4	5.0
23	MMD18	I/O-CMOS	4	5.0
24	MMD19	I/O-CMOS	4	5.0
25	MMD20	I/O-CMOS	4	5.0
26	MMD21	I/O-CMOS	4	5.0
27	MMD22	I/O-CMOS	4	5.0
28	MMD23	I/O-CMOS	4	5.0
29	MMD24	I/O-CMOS	4	5.0
30	MMD25	I/O-CMOS	4	5.0
31	MMD26	I/O-CMOS	4	5.0
32	MMD27	I/O-CMOS	4	5.0
33	VCC5	I-P		
34	CPUCLK	I-TTL		5.0
35	GND	I-G		
36	MMD28	I/O-CMOS	4	5.0
37	MMD29	I/O-CMOS	4	5.0
38	MMD30	I/O-CMOS	4	5.0
39	MMD31	I/O-CMOS	4	5.0
40	DBCTL0#	I-TTL		5.0
41	SDRAM_EDO	I-TTL		5.0
42	HDOE#	I-TTL		5.0
43	MDOE#	I-TTL		5.0
	DBCTL2#			
44	OADV#	I-TTL		5.0
45	MMDOE#	I-TTL		5.0
	DBCTL1#			
46	DLE0#	I-TTL		5.0
47	DLE1#	I-TTL		5.0
	PPSEL			
48	MPERR#	I/O-TTL	4	5.0
	IADV#			
49	HD63	I/O-TTL	8	3.3
50	HD62	I/O-TTL	8	3.3
51	HD61	I/O-TTL	8	3.3
52	GND	I-G		
53	VCC3	I-P		
54	HD60	I/O-TTL	8	3.3
55	HD59	I/O-TTL	8	3.3
56	HD58	I/O-TTL	8	3.3
57	HD57	I/O-TTL	8	3.3
58	HD56	I/O-TTL	8	3.3
59	HD55	I/O-TTL	8	3.3
60	HD54	I/O-TTL	8	3.3
61	HD53	I/O-TTL	8	3.3
62	HD52	I/O-TTL	8	3.3
63	VCC3	I-P		
64	HD51	I/O-TTL	8	3.3
65	HD50	I/O-TTL	8	3.3
66	GND	I-G		
67	HD49	I/O-TTL	8	3.3
68	HD48	I/O-TTL	8	3.3
69	HD47	I/O-TTL	8	3.3
70	HD46	I/O-TTL	8	3.3
71	HD45	I/O-TTL	8	3.3
72	HD44	I/O-TTL	8	3.3
73	VCC3	I-P		
74	HD43	I/O-TTL	8	3.3
75	HD42	I/O-TTL	8	3.3
76	HD41	I/O-TTL	8	3.3
77	HD40	I/O-TTL	8	3.3
78	HD39	I/O-TTL	8	3.3
79	HD38	I/O-TTL	8	3.3
80	GND	I-G		
81	HD37	I/O-TTL	8	3.3
82	HD36	I/O-TTL	8	3.3
83	HD35	I/O-TTL	8	3.3
84	HD34	I/O-TTL	8	3.3
85	HD33	I/O-TTL	8	3.3
86	HD32	I/O-TTL	8	3.3
87	HD31	I/O-TTL	8	3.3
88	HD30	I/O-TTL	8	3.3
89	HD29	I/O-TTL	8	3.3
90	HD28	I/O-TTL	8	3.3
91	VCC3	I-P		
92	GND	I-G		
93	HD27	I/O-TTL	8	3.3
94	HD26	I/O-TTL	8	3.3
95	HD25	I/O-TTL	8	3.3
96	HD24	I/O-TTL	8	3.3
97	HD23	I/O-TTL	8	3.3
98	HD22	I/O-TTL	8	3.3
99	HD21	I/O-TTL	8	3.3
100	HD20	I/O-TTL	8	3.3
101	HD19	I/O-TTL	8	3.3
102	HD18	I/O-TTL	8	3.3
103	HD17	I/O-TTL	8	3.3
104	GND	I-G		
105	VCC3	I-P		
106	HD16	I/O-TTL	8	3.3
107	HD15	I/O-TTL	8	3.3
108	HD14	I/O-TTL	8	3.3
109	HD13	I/O-TTL	8	3.3
110	HD12	I/O-TTL	8	3.3
111	HD11	I/O-TTL	8	3.3
112	HD10	I/O-TTL	8	3.3
113	HD9	I/O-TTL	8	3.3
114	HD8	I/O-TTL	8	3.3
115	HD7	I/O-TTL	8	3.3
116	GND	I-G		
117	HD6	I/O-TTL	8	3.3
118	HD5	I/O-TTL	8	3.3
119	HD4	I/O-TTL	8	3.3
120	VCC3			
121	HD3	I/O-TTL	8	3.3
122	HD2	I/O-TTL	8	3.3
123	HD1	I/O-TTL	8	3.3
124	HD0	I/O-TTL	8	3.3
125	MP0	I/O-CMOS	4	3.3/5.0
	BE0#			
126	MP1	I/O-CMOS	8	3.3/5.0
	BE1#			
127	MP2	I/O-CMOS	8	3.3/5.0
	BE2#			
128	MP3	I/O-CMOS	8	3.3/5.0
	BE3#			
129	GND	I-G		
130	MP4	I/O-CMOS	8	3.3/5.0
	BE4#			
131	MP5	I/O-CMOS	8	3.3/5.0
	BE5#			
132	MP6	I/O-CMOS	8	3.3/5.0
	BE6#			

82C566 Numerical Pin Cross-Reference List (cont.)

Pin No.	Pin Name	Pin Type	Drive (mA)	Pwr (V)
133	MP7	I/O-CMOS	8	3.3/5.0
	BE7#			
134	MD63	I/O-CMOS	8	3.3/5.0
135	MD62	I/O-CMOS	8	3.3/5.0
136	MD61	I/O-CMOS	8	3.3/5.0
137	MD60	I/O-CMOS	8	3.3/5.0
138	MD59	I/O-CMOS	8	3.3/5.0
139	MD58	I/O-CMOS	8	3.3/5.0
140	MD57	I/O-CMOS	8	3.3/5.0
141	MD56	I/O-CMOS	8	3.3/5.0
142	GND	I-G		
143	MD55	I/O-CMOS	8	3.3/5.0
144	VCC3/5	I-P		
145	MD54	I/O-CMOS	8	3.3/5.0
146	MD53	I/O-CMOS	8	3.3/5.0
147	MD52	I/O-CMOS	8	3.3/5.0
148	MD51	I/O-CMOS	8	3.3/5.0
149	MD50	I/O-CMOS	8	3.3/5.0
150	MD49	I/O-CMOS	8	3.3/5.0
151	MD48	I/O-CMOS	8	3.3/5.0
152	MD31	I/O-CMOS	8	3.3/5.0
153	MD30	I/O-CMOS	8	3.3/5.0
154	MD29	I/O-CMOS	8	3.3/5.0
155	MD28	I/O-CMOS	8	3.3/5.0
156	GND	I-G		
157	VCC3/5	I-P		
158	MD27	I/O-CMOS	8	3.3/5.0
159	MD26	I/O-CMOS	8	3.3/5.0
160	MD25	I/O-CMOS	8	3.3/5.0
161	MD24	I/O-CMOS	8	3.3/5.0
162	MD23	I/O-CMOS	8	3.3/5.0
163	MD22	I/O-CMOS	8	3.3/5.0
164	MD21	I/O-CMOS	8	3.3/5.0
165	MD20	I/O-CMOS	8	3.3/5.0
166	MD19	I/O-CMOS	8	3.3/5.0
167	MD18	I/O-CMOS	8	3.3/5.0
168	MD17	I/O-CMOS	8	3.3/5.0
169	GND	I-G		
170	MD16	I/O-CMOS	8	3.3/5.0
171	VCC3/5	I-P		
172	MD47	I/O-CMOS	8	3.3/5.0
173	MD46	I/O-CMOS	8	3.3/5.0
174	MD45	I/O-CMOS	8	3.3/5.0
175	MD44	I/O-CMOS	8	3.3/5.0
176	MD43	I/O-CMOS	8	3.3/5.0
177	MD42	I/O-CMOS	8	3.3/5.0
178	MD41	I/O-CMOS	8	3.3/5.0
179	MD40	I/O-CMOS	8	3.3/5.0
180	MD39	I/O-CMOS	8	3.3/5.0
181	MD38	I/O-CMOS	8	3.3/5.0
182	GND	I-G		
183	MD37	I/O-CMOS	8	3.3/5.0
184	MD36	I/O-CMOS	8	3.3/5.0
185	MD35	I/O-CMOS	8	3.3/5.0
186	MD34	I/O-CMOS	8	3.3/5.0
187	MD33	I/O-CMOS	8	3.3/5.0
188	VCC3/5	I-P		
189	MD32	I/O-CMOS	8	3.3/5.0
190	MD15	I/O-CMOS	8	3.3/5.0
191	MD14	I/O-CMOS	8	3.3/5.0
192	MD13	I/O-CMOS	8	3.3/5.0
193	MD12	I/O-CMOS	8	3.3/5.0
194	MD11	I/O-CMOS	8	3.3/5.0
195	GND	I-G		
196	MD10	I/O-CMOS	8	3.3/5.0
197	MD9	I/O-CMOS	8	3.3/5.0
198	MD8	I/O-CMOS	8	3.3/5.0
199	MD7	I/O-CMOS	8	3.3/5.0
200	MD6	I/O-CMOS	8	3.3/5.0
201	MD5	I/O-CMOS	8	3.3/5.0
202	VCC3/5	I-P		
203	MD4	I/O-CMOS	8	3.3/5.0
204	MD3	I/O-CMOS	8	3.3/5.0
205	MD2	I/O-CMOS	8	3.3/5.0
206	MD1	I/O-CMOS	8	3.3/5.0
207	MD0	I/O-CMOS	8	3.3/5.0
208	GND	I-G		

Table 3-3 82C566 Alphabetical Pin Cross-Reference List

Pin Name	Pin No.
CPUCLK	34
DBCTL0#	40
DLE0#	46
DLE1#+PPSEL	47
GND	17
GND	35
GND	52
GND	66
GND	80
GND	92
GND	104
GND	116
GND	129
GND	142
GND	156
GND	169
GND	182
GND	195
GND	208
HD0	124
HD1	123
HD2	122
HD3	121
HD4	119
HD5	118
HD6	117
HD7	115
HD8	114
HD9	113
HD10	112
HD11	111
HD12	110
HD13	109
HD14	108
HD15	107
HD16	106
HD17	103
HD18	102
HD19	101
HD20	100
HD21	99
HD22	98
HD23	97
HD24	96
HD25	95
HD26	94
HD27	93
HD28	90
HD29	89
HD30	88
HD31	87
HD32	86

Pin Name	Pin No.
HD33	85
HD34	84
HD35	83
HD36	82
HD37	81
HD38	79
HD39	78
HD40	77
HD41	76
HD42	75
HD43	74
HD44	72
HD45	71
HD46	70
HD47	69
HD48	68
HD49	67
HD50	65
HD51	64
HD52	62
HD53	61
HD54	60
HD55	59
HD56	58
HD57	57
HD58	56
HD59	55
HD60	54
HD61	51
HD62	50
HD63	49
HDOE#	42
MD0	207
MD1	206
MD2	205
MD3	204
MD4	203
MD5	201
MD6	200
MD7	199
MD8	198
MD9	197
MD10	196
MD11	194
MD12	193
MD13	192
MD14	191
MD15	190
MD16	170
MD17	168
MD18	167
MD19	166

Pin Name	Pin No.
MD20	165
MD21	164
MD22	163
MD23	162
MD24	161
MD25	160
MD26	159
MD27	158
MD28	155
MD29	154
MD30	153
MD31	152
MD32	189
MD33	187
MD34	186
MD35	185
MD36	184
MD37	183
MD38	181
MD39	180
MD40	179
MD41	178
MD42	177
MD43	176
MD44	175
MD45	174
MD46	173
MD47	172
MD48	151
MD49	150
MD50	149
MD51	148
MD52	147
MD53	146
MD54	145
MD55	143
MD56	141
MD57	140
MD58	139
MD59	138
MD60	137
MD61	136
MD62	135
MD63	134
MDOE#+DBCTL2#	43
MMD0	2
MMD1	3
MMD2	4
MMD3	5
MMD4	6
MMD5	7
MMD6	8

Pin Name	Pin No.
MMD7	9
MMD8	10
MMD9	12
MMD10	13
MMD11	14
MMD12	15
MMD13	16
MMD14	18
MMD15	19
MMD16	20
MMD17	22
MMD18	23
MMD19	24
MMD20	25
MMD21	26
MMD22	27
MMD23	28
MMD24	29
MMD25	30
MMD26	31
MMD27	32
MMD28	36
MMD29	37
MMD30	38
MMD31	39
MMDOE#+DBCTL1#	45
MPERR#+IADV#	48
MP0+BE0#	125
MP1+BE1#	126
MP2+BE2#	127
MP3+BE3#	128
MP4+BE4#	130
MP5+BE5#	131
MP6+BE6#	132
MP7+BE7#	133
OADV#	44
SDRAM_EDO	41
VCC3	53
VCC3	63
VCC3	73
VCC3	91
VCC3	105
VCC3	120
VCC3/5	144
VCC3/5	157
VCC3/5	171
VCC3/5	188
VCC3/5	202
VCC5	1
VCC5	11
VCC5	21
VCC5	33

3.2 82C566 Signal Descriptions

3.2.1 CPU Bus Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
HD[63:0]	49:51, 54:62, 64, 65, 67:72, 74:79, 81:90, 93:103, 106:115, 117:119, 121:124	I/O-TTL (8mA)		**Host Data Bus:** These pins are bidirectional and connected directly to the CPU data bus and L2 cache data lines.

3.2.2 82C567 Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
OADV#	44	I-TTL		**Output Pointer Advance:** This signal indicates to the 82C566 to output the next quad-word in the deep buffer. A 10 ohm series resistor is recommended on this signal.
DBCTL0#	40	I-TTL		**Data Buffer Control Line 0:** DBCTL0# is used to control the data paths when the deep buffers are enabled. A 10 ohm series resistor is recommended on this signal.
MMDOE#	45	I-TTL	Cycle Multiplexed	**MMD Output Enable**: DBCTL0#, MDOE#+DBCTL2#, MMDOE#+DBCTL1#, and HDOE# form the encoded commands sent from the 82C567 to the 82C566. These commands indicate the type of cycle currently underway and enables the 82C566 to perform the appropriate data steering, latching, and direction control. When asserted, MMDOE# enables data to be put on the MMD bus for PCI related operations.
DBCTL1#				**Data Buffer Control Line 1:** DBCTL1# is used to control the data paths when the deep buffers are enabled. A 10 ohm series resistor is recommended on this signal.
MDOE#	43	I-TTL	Cycle Multiplexed	**Memory Data Output Enable:** DBCTL0#, MDOE#+DBCTL2#, MMDOE#+DBCTL1#, and HDOE# form the encoded commands that are sent out by the 82C567. When asserted, this signal enables data to be put out on the MD bus. MDOE# is asserted for CPU writes to cache/DRAM, CPU writes to PCI, PCI reads from cache/DRAM, L2 cache write-back cycles, and PCI writes to DRAM. A 10 ohm series resistor is recommended on this signal.
DBCTL2#				**Data Buffer Control Line 2:** DBCTL2# is used to control the data paths when the deep buffers are enabled. A 10 ohm series resistor is recommended on this signal.

82C566 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
HDOE#	42	I-TTL		**Host Data Output Enable:** HDOE#, MDOE, MMDOE and DBCOE# form the encoded commands that are sent out by the 82C567. When asserted, HDOE# enables data to be put out on the HD bus. HDOE# is asserted for CPU reads from DRAM/PCI bus, PCI writes to cache, CPU linefills, Suspend mode indication, and reset state indication. A 10 ohm series resistor is recommended on this signal.
DLE1#	47	I-TTL	Cycle Multiplexed	**Data Latch Enable:** This line is connected to the DLE1# pin of the 82C567 and used to latch the HD, MD, and MMD data bus depending on which cycle is occurring. A 10 ohm series resistor is recommended on this signal.
PPSEL				**Ping-Pong Buffer Select:** This signal is connected to the 82C567 and is used to select between the ping-pong buffers. A 10 ohm series resistor is recommended on this signal.
DLE0#	46	I		**Data Latch Enable:** This line is connected to the DLE0# pin of the 82C567 and used to latch the HD, MD, and MMD data bus depending on which cycle is occurring. A 10 ohm series resistor is recommended on this signal.
SDRAM_EDO	41	I-TTL		**SDRAM and EDO Cycle:** This signal is connected to the 82C567 and is used to select the special data paths required for SDRAM and EDO memory cycles. A 10 ohm series resistor is recommended on this signal.

3.2.3 DRAM Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
CPUCLK	34	I-TTL		**Clock:** This input clock should be identical to the 82C567 CLK input and the clock signal sent to the CPU.
MD[63:0]	134:141, 143, 145:151, 172:181, 183:187, 189, 152:155, 158:168, 170, 189, 190:194, 196:201, 203:207	I/O-CMOS (8mA)		**Memory Data Bus:** These pins are connected directly to the DRAM data bus. These lines have internal pull-up resistors.

82C566 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
MP[7:0]	133:130, 128:125	I/O-CMOS (8mA)	Strap option on MMD16	**Memory Parity Bits:** These pins are connected directly to the system DRAM parity lines.
BE[7:0]#		I		**Byte Enables:** The byte enables indicate which byte lanes on the CPU data bus are carrying valid data during the current bus cycle. If memory parity is disabled, then these may be connected to BE[7:0]#
MPERR#	48	O (4mA)	PCIDV0 45h[1] = 0 and 45h[3] = 1	**Memory Parity Error Indicator:** This pin is connected to the MPERR# input of the 82C567. It indicates the detection of a parity error during a read form DRAM is qualified within the 82C567 when parity is enabled.
IADV#		I	PCIDV0 45h[1] = 1 and 45h[3] = 0	**Input Pointer Advance:** This signal is connected to the 82C567 and is used to input the next quad-word into the deep buffer.

3.2.4 82C568 Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
MMD[31:0]	39:36, 32:22, 20:18, 16:12, 10:2	I/O-CMOS (4mA)		**IPC to DBC Data Path:** These pins are connected directly to the 82C568. This private bus serves as a conduit for CPU reads/writes to/from PCI/ISA. These lines have internal pull-up resistors. During system reset, a special combination of DBCTL0#, MDOE#+DBCTL2#, MMDOE#+DBCTL1#, and HDOE# will be sent out by the 82C567 to the 82C566. During this, the 82C566 will sample the MMD[14:0], MMD16, and MMD31 lines for its strap options at the rising edge of DLE0#. Table 3-4 gives the strap options for these MMD lines.

3.2.5 Power and Ground Pins

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
GND	17, 35, 52, 66, 80, 92, 104, 116, 129, 142, 156, 169, 182, 195, 208	I-G		**Ground Connection**

82C566 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
VCC3	53, 63, 73, 91, 105, 120	I-P		**Power Connection:** 3.3V power plane
VCC3/5	144, 157, 171, 188, 202	I-P		**Power Connection:** These pins power the memory interface at either 3.3V or 5.0V.
VCC5	1, 11, 21, 33	I-P		**Power Connection:** 5.0V power plane

Table 3-4 Available Strap Options on MMD Lines

Strap	0	1 (Default Setting)	Condition(1)
MMD0	3.3V DRAM	5.0V DRAM	
MMD1	Enable test mode	Disable test mode	
MMD2	Test Mode 0 - Tristate all bidirectionals (except MMD15, end of input and bidirectional NAND chain present on MMD15)	Test Mode 1 - Tristate all bidirectionals and (COT version only) disable MD and HD internal pull-up/-down resistors	
MMD3	Disable 82C566 config_write	Enable 82C566 config_write	
MMD4	EDO/SDRAM/BEDO	FP Mode	MMD3 = 0
MMD5	Enable ping-pong buffer for PCI master read X-1-1-1	Disable ping-pong buffer for PCI master read X-1-1-1	MMD3 = 0
MMD6	Enable ping-pong buffer for PCI master write X-1-1-1	Disable ping-pong buffer for PCI master write X-1-1-1	MMD3 = 0
MMD7	BEDO	SDRAM	MMD3 = 0, MMD4 = 0, and MMD31 = 1
MMD8	Enable 6QW FIFO for CPU write to DRAM	Disable 6QW FIFO for CPU write to DRAM	MMD3 = 0
MMD9	Enable 24DW FIFO for PCI write to DRAM	Disable 24DW FIFO for PCI write to DRAM	MMD3 = 0
MMD10	Enable 24DW FIFO for PCI read from DRAM	Disable 24DW FIFO for PCI read from DRAM	MMD3 = 0
MMD11	Enable 6DW FIFO for CPU write to PCI	Disable 6DW FIFO for CPU write to PCI	MMD3 = 0
MMD12	Disable MD bus internal pull-up resistors	Enable MD bus internal pull-up resistors	MMD3 = 0
MMD13	Enable byte merge for CPU write to DRAM	Disable byte merge for CPU write to DRAM	MMD3 = 0
MMD14	Enable byte merge for CPU write to PCI	Disable byte merge for CPU write to PCI	MMD3 = 0
MMD16	Enable parity generation and parity checking	Disable parity generation and parity checking	MMD3 = 0, MMD13 = 1, and MMD14 = 1
MMD31	Enable SDRAM/BEDO	Disable SDRAM/BEDO	MMD3 = 0

(1) When strap input MMD3 = 1, strap options on MMD31, MMD16 and MMD[14:4] have yet to be defined.

Figure 3-2 82C567 Pin Diagram

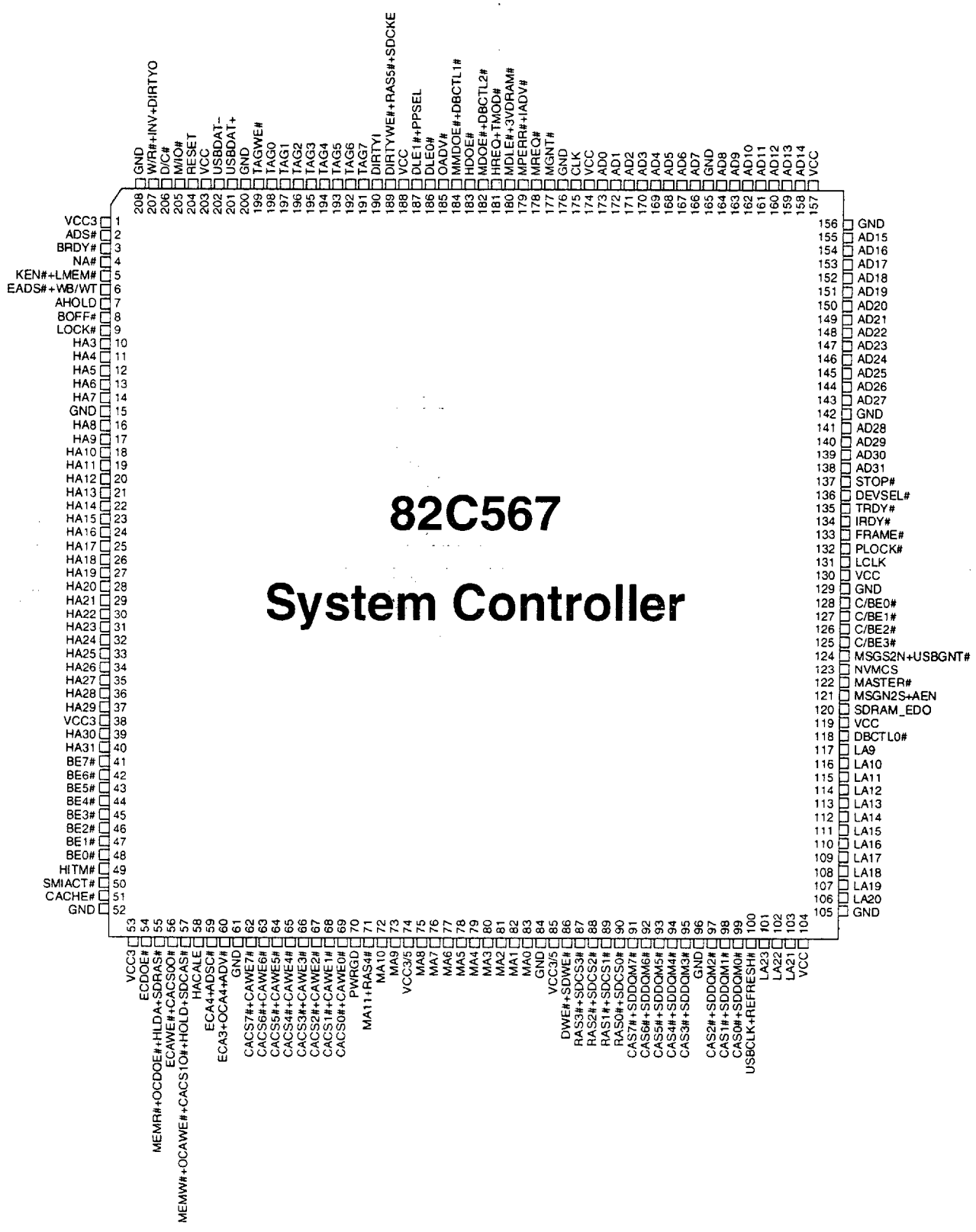


Table 3-5 82C567 Numerical Pin Cross-Reference List

Pin No.	Pin Name	Pin Type	Drive (mA)	Pwr (V)
1	VCC3	I-P		
2	ADS#	I/O-TTL	4	3.3
3	BRDY#	O	8	3.3
4	NA#	O	8	3.3
5	KEN#	O	8	3.3
	LMEM#			
6	EADS#	O	8	3.3
	WB/WT#			
7	AHOLD	O	8	3.3
8	BOFF#	O	8	3.3
9	LOCK#	I-TTL		3.3
10	HA3	I/O-TTL	4	3.3
11	HA4	I/O-TTL	4	3.3
12	HA5	I/O-TTL	4	3.3
13	HA6	I/O-TTL	4	3.3
14	HA7	I/O-TTL	4	3.3
15	GND	I-G		
16	HA8	I/O-TTL	4	3.3
17	HA9	I/O-TTL	4	3.3
18	HA10	I/O-TTL	4	3.3
19	HA11	I/O-TTL	4	3.3
20	HA12	I/O-TTL	4	3.3
21	HA13	I/O-TTL	4	3.3
22	HA14	I/O-TTL	4	3.3
23	HA15	I/O-TTL	4	3.3
24	HA16	I/O-TTL	4	3.3
25	HA17	I/O-TTL	4	3.3
26	HA18	I/O-TTL	4	3.3
27	HA19	I/O-TTL	4	3.3
28	HA20	I/O-TTL	4	3.3
29	HA21	I/O-TTL	4	3.3
30	HA22	I/O-TTL	4	3.3
31	HA23	I/O-TTL	4	3.3
32	HA24	I/O-TTL	4	3.3
33	HA25	I/O-TTL	4	3.3
34	HA26	I/O-TTL	4	3.3
35	HA27	I/O-TTL	4	3.3
36	HA28	I/O-TTL	4	3.3
37	HA29	I/O-TTL	4	3.3
38	VCC3	I-P		
39	HA30	I/O-TTL	4	3.3
40	HA31	I/O-TTL	4	3.3
41	BE7#	I-TTL		3.3
42	BE6#	I-TTL		3.3
43	BE5#	I-TTL		3.3
44	BE4#	I-TTL		3.3
45	BE3#	I-TTL		3.3
46	BE2#	I-TTL		3.3
47	BE1#	I-TTL		3.3
48	BE0#	I-TTL		3.3
49	HITM#	I-TTL		3.3
50	SMIACT#	I-TTL		3.3
51	CACHE#	I-TTL		3.3
52	GND	I-G		
53	VCC3	I-P		
54	ECDOE#	O	8	3.3
55	MEMR#	I		3.3
	OCDOE#	O	8	
	HLDA	I-TTL		
	SDRAS#	O	8	
56	ECAWE#	O	8	3.3
	CACS0O#			
57	MEMW#	I		3.3
	OCAWE#	O	8	
	CACS1O#			
	HOLD			
	SDCAS#			
58	HACALE	O	8	3.3
59	ECA4	O	8	3.3
	ADSC#			
60	ECA3	O	8	3.3
	OCA4			
	ADV#			
61	GND	I-G		
62	CACS7#	O	4	3.3
	CAWE7#			
63	CACS6#	O	4	3.3
	CAWE6#			
64	CACS5#	O	4	3.3
	CAWE5#			
65	CACS4#	O	4	3.3
	CAWE4#			
66	CACS3#	O	4	3.3
	CAWE3#			
67	CACS2#	O	4	3.3
	CAWE2#			
68	CACS1#	O	4	3.3
	CAWE1#			
69	CACS0#	O	4	3.3
	CAWE0#			
70	PWRGD	I-S		5.0
71	MA11	O	4*	5.0
	RAS4#			
72	MA10	O	4*	5.0
73	MA9	O	4*	5.0
74	VCC3/5	I-P		
75	MA8	O	4*	5.0
76	MA7	O	4*	5.0
77	MA6	O	4*	5.0
78	MA5	O	4*	5.0
79	MA4	O	4*	5.0
80	MA3	O	4*	5.0
81	MA2	O	4*	5.0
82	MA1	O	4*	5.0
83	MA0	O	4*	5.0
84	GND	I-G		
85	VCC3/5	I-P		
86	DWE#	O	16	5.0
	SDWE#			
87	RAS3#	O	4*	5.0
	SDCS3#			
88	RAS2#	O	4*	5.0
	SDCS2#			
89	RAS1#	O	4*	5.0
	SDCS1#			
90	RAS0#	O	4*	5.0
	SDCS0#			
91	CAS7#	O	8	5.0
	SDDQM7#			
92	CAS6#	O	8	5.0
	SDDQM6#			
93	CAS5#	O	8	5.0
	SDDQM5#			
94	CAS4#	O	8	5.0
	SDDQM4#			
95	CAS3#	O	8	5.0
	SDDQM3#			
96	GND	I-G		
97	CAS2#	O	8	5.0
	SDDQM2#			
98	CAS1#	O	8	5.0
	SDDQM1#			
99	CAS0#	O	8	5.0
	SDDQM0#			
100	USBCLK	I-TTL		5.0
	REFRESH#			
101	LA23	I/O-CMOS	8	5.0
102	LA22	I/O-CMOS	8	5.0
103	LA21	I/O-CMOS	8	5.0
104	VCC	I-P		
105	GND	I-G		
106	LA20	I/O-CMOS	8	5.0
107	LA19	I/O-CMOS	8	5.0
108	LA18	I/O-CMOS	8	5.0

82C567 Numerical Pin Cross-Reference List (cont.)

Pin No.	Pin Name	Pin Type	Drive (mA)	Pwr (V)
109	LA17	I/O-CMOS	8	5.0
110	LA16	I/O-CMOS	8	5.0
111	LA15	I/O-CMOS	8	5.0
112	LA14	I/O-CMOS	8	5.0
113	LA13	I/O-CMOS	8	5.0
114	LA12	I/O-CMOS	8	5.0
115	LA11	I/O-CMOS	8	5.0
116	LA10	I/O-CMOS	8	5.0
117	LA9	I/O-CMOS	8	5.0
118	DBCTL0#	O	8	5.0
119	VCC	I-P		
120	SDRAM_EDO	O	8	5.0
121	MSGN2S	I/O-TTL	4	5.0
	AEN	I		
122	MASTER#	I-CMOS		5.0
123	NVMCS	O	4	5.0
124	MSGS2N	I-TTL		5.0
	USBGNT#			
125	C/BE3#	I/O-TTL	PCI	5.0
126	C/BE2#	I/O-TTL	PCI	5.0
127	C/BE1#	I/O-TTL	PCI	5.0
128	C/BE0#	I/O-TTL	PCI	5.0
129	GND	I-G		
130	VCC	I-P		
131	LCLK	I-TTL		5.0
132	PLOCK#	I/O-TTL	PCI	5.0
133	FRAME#	I/O-TTL	PCI	5.0
134	IRDY#	I/O-TTL	PCI	5.0
135	TRDY#	I/O-TTL	PCI	5.0
136	DEVSEL#	I/O-TTL	PCI	5.0
137	STOP#	I/O-TTL	PCI	5.0
138	AD31	I/O-TTL	PCI	5.0
139	AD30	I/O-TTL	PCI	5.0
140	AD29	I/O-TTL	PCI	5.0
141	AD28	I/O-TTL	PCI	5.0
142	GND	I-G		
143	AD27	I/O-TTL	PCI	5.0
144	AD26	I/O-TTL	PCI	5.0
145	AD25	I/O-TTL	PCI	5.0
146	AD24	I/O-TTL	PCI	5.0
147	AD23	I/O-TTL	PCI	5.0
148	AD22	I/O-TTL	PCI	5.0
149	AD21	I/O-TTL	PCI	5.0
150	AD20	I/O-TTL	PCI	5.0
151	AD19	I/O-TTL	PCI	5.0
152	AD18	I/O-TTL	PCI	5.0
153	AD17	I/O-TTL	PCI	5.0
154	AD16	I/O-TTL	PCI	5.0
155	AD15	I/O-TTL	PCI	5.0
156	GND	I-G		
157	VCC	I-P		
158	AD14	I/O-TTL	PCI	5.0
159	AD13	I/O-TTL	PCI	5.0
160	AD12	I/O-TTL	PCI	5.0
161	AD11	I/O-TTL	PCI	5.0
162	AD10	I/O-TTL	PCI	5.0
163	AD9	I/O-TTL	PCI	5.0
164	AD8	I/O-TTL	PCI	5.0
165	GND	I-G		
166	AD7	I/O-TTL	PCI	5.0
167	AD6	I/O-TTL	PCI	5.0
168	AD5	I/O-TTL	PCI	5.0
169	AD4	I/O-TTL	PCI	5.0
170	AD3	I/O-TTL	PCI	5.0
171	AD2	I/O-TTL	PCI	5.0
172	AD1	I/O-TTL	PCI	5.0
173	AD0	I/O-TTL	PCI	5.0
174	VCC	I-P		
175	CLK	I-TTL		5.0
176	GND	I-G		
177	MGNT#	O	4	5.0
178	MREQ#	I-TTL		5.0
179	MPERR#	I/O-TTL	4	5.0
	IADV#	O	4	
180	MDLE#	O	4	5.0
	3VDRAM#	I-TTL		
181	HREQ	I-TTL		5.0
	TMOD#			
182	MDOE#	O	8	5.0
	DBCTL2#			
183	HDOE#	O	8	5.0
184	MMDOE#	O	8	5.0
	DBCTL1#			
185	OADV#	O	8	5.0
186	DLE0#	O	8	5.0
187	DLE1#	O	8	5.0
	PPSEL			
188	VCC	I-P		5.0
189	DIRTYWE#	O	8	5.0
	RAS5#			
	SDCKE			
190	DIRTYI	I/O-TTL	4	5.0
191	TAG7	I/O-TTL	4	5.0
192	TAG6	I/O-TTL	4	5.0
193	TAG5	I/O-TTL	4	5.0
194	TAG4	I/O-TTL	4	5.0
195	TAG3	I/O-TTL	4	5.0
196	TAG2	I/O-TTL	4	5.0
197	TAG1	I/O-TTL	4	5.0
198	TAG0	I/O-TTL	4	5.0
199	TAGWE#	O	8	5.0
200	GND	I-G		
201	USBDAT+	I/O-Analg		5.0
202	USBDAT–	I/O-Analg		5.0
203	VCC	I-P		
204	RESET	O	8	3.3
205	M/IO#	I-TTL		3.3
206	D/C#	I-TTL		3.3
207	W/R#	I-TTL		3.3
	INV			
	DIRTYO	I/O-TTL	4	
208	GND	I-G		

*The default drive is 4mA, however, by setting SYSCFG 18h[4] = 1 it can be increased to 16mA.

Table 3-6 82C567 Alphabetical Pin Cross-Reference List

Pin Name	Pin No.
AD0	173
AD1	172
AD2	171
AD3	170
AD4	169
AD5	168
AD6	167
AD7	166
AD8	164
AD9	163
AD10	162
AD11	161
AD12	160
AD13	159
AD14	158
AD15	155
AD16	154
AD17	153
AD18	152
AD19	151
AD20	150
AD21	149
AD22	148
AD23	147
AD24	146
AD25	145
AD26	144
AD27	143
AD28	141
AD29	140
AD30	139
AD31	138
ADS#	2
AHOLD	7
BE0#	48
BE1#	47
BE2#	46
BE3#	45
BE4#	44
BE5#	43
BE6#	42
BE7#	41
BOFF#	8
BRDY#	3

Pin Name	Pin No.
C/BE0#	128
C/BE1#	127
C/BE2#	126
C/BE3#	125
CACHE#	51
CACS0#+CAWE0#	69
CACS1#+CAWE1#	68
CACS2#+CAWE2#	67
CACS3#+CAWE3#	66
CACS4#+CAWE4#	65
CACS5#+CAWE5#	64
CACS6#+CAWE6#	63
CACS7#+CAWE7#	62
CAS0#+SDDQM0#	99
CAS1#+SDDQM1#	98
CAS2#+SDDQM2#	97
CAS3#+SDDQM3#	95
CAS4#+SDDQM4#	94
CAS5#+SDDQM5#	93
CAS6#+SDDQM6#	92
CAS7#+SDDQM7#	91
CLK	175
D/C#	206
DBCTL0#	118
DEVSEL#	136
DIRTYI	190
DIRTYWE#+RAS5#+ SDCKE	189
DLE0#	186
DLE1#+PPSEL	187
DWE#+SDWE#	86
EADS#+WB/WT#	6
ECA3+OCA4+ADV#	60
ECA4+ADSC#	59
ECAWE#+CACS0O#	56
ECDOE#	54
FRAME#	133
GND	15
GND	52
GND	61
GND	84
GND	96
GND	105
GND	129

Pin Name	Pin No.
GND	142
GND	156
GND	165
GND	176
GND	200
GND	208
HA3	10
HA4	11
HA5	12
HA6	13
HA7	14
HA8	16
HA9	17
HA10	18
HA11	19
HA12	20
HA13	21
HA14	22
HA15	23
HA16	24
HA17	25
HA18	26
HA19	27
HA20	28
HA21	29
HA22	30
HA23	31
HA24	32
HA25	33
HA26	34
HA27	35
HA28	36
HA29	37
HA30	39
HA31	40
HACALE	58
HDOE#	183
HITM#	49
HREQ+TMOD#	181
IRDY#	134
KEN#+LMEM#	5
LA9	117
LA10	116
LA11	115

Pin Name	Pin No.
LA12	114
LA13	113
LA14	112
LA15	111
LA16	110
LA17	109
LA18	108
LA19	107
LA20	106
LA21	103
LA22	102
LA23	101
LCLK	131
LOCK#	9
M/IO#	205
MA0	83
MA1	82
MA2	81
MA3	80
MA4	79
MA5	78
MA6	77
MA7	76
MA8	75
MA9	73
MA10	72
MA11+RAS4#	71
MASTER#	122
MDLE#+3VDRAM#	180
MDOE#+DBCTL2#	182
MEMR#+OCAWE#+ CACS1O#+HOLD+ SDCAS#	57
MEMW#+OCDOE#+ HLDA+SDRAS#	55
MGNT#	177
MMDOE#+DBCTL1#	184
MPERR#+IADV#	179
MREQ#	178
MSGN2S+AEN	121
MSGS2N+USBGNT#	124
NA#	4
NVMCS	123
OADV#	185
PLOCK#	132

Pin Name	Pin No.
PWRGD	70
RAS0#+SDCS0#	90
RAS1#+SDCS1#	89
RAS2#+SDCS2#	88
RAS3#+SDCS3#	87
RESET	204
SDRAM_EDO	120
SMIACT#	50
STOP#	137
TAG0	198
TAG1	197
TAG2	196
TAG3	195
TAG4	194
TAG5	193
TAG6	192
TAG7	191
TAGWE#	199
TRDY#	135
USBCLK+REFRESH#	100
USBDAT+	201
USBDAT–	202
VCC	104
VCC	119
VCC	130
VCC	157
VCC	174
VCC	188
VCC	203
VCC3	1
VCC3	38
VCC3	53
VCC3/5	74
VCC3/5	85
W/R#+INV+DIRTYO	207

3.3 82C567 Signal Descriptions

3.3.1 Reset and Clock Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
RESET	204	O (8mA)		**System Reset:** When asserted, this signal resets the CPU. When the PWRGD signal makes a low-to-high transition, RESET is asserted and is guaranteed to be active for 1ms such that CLK and VCC are stable.
PWRGD	70	I-S		**Power Good:** This input reflects the "wired-OR" status of the external reset switch and the power good status from the power supply.
CLK	175	I-TTL		**Clock:** This input is used as the master single frequency clock. This signal has to be identical to the clock signal sent to the CPU.
LCLK	131	I-TTL		**Local Bus Clock:** This clock is used by the PCI bus state machine within the 82C567. The same clock or another identical signal is used by the local bus devices. For a synchronous PCI implementation, this signal can be skewed from the CLK input by a margin of ±2ns.
USBCLK	100	I-TTL	SYSCFG 2Eh[6] = 0	**Universal Serial Bus Clock:** This clock is used by the PCI-USB module within the 82C567.
REFRESH#			SYSCFG 2Eh[6] = 1	**Refresh:** Driven by the 82C568 (or an ISA master), this input to the 82C567 starts a refresh cycle on local DRAM. For future compatibility, place a zero ohm resistor in this connection.

3.3.2 CPU Bus Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
HA[31:3]	40:39, 37:16, 14:10	I/O-TTL (4mA)		**Host Address Bus Lines 31 through 3:** HA[31:3] are the address lines of the CPU bus. HA[31:3] are connected to the CPU A[31:3] lines. Along with the byte enable signals, the HA[31:3] lines define the physical area of memory or I/O being accessed. During CPU cycles, the HA[31:3] lines are inputs to the 82C567. They are used for address decoding and second level cache tag lookup sequences. During inquire cycles, the HA[31:5] are outputs from the 82C567 to the CPU to snoop the first level cache tags. They also are outputs from the 82C567 to the L2 cache. HA[31:3] have internal pull-downs, however, external pull-ups should be used on HA3 and HA4.
BE[7:0]#	41:48	I-TTL		**Byte Enables 7 through 0:** The byte enables indicate which byte lanes on the CPU data bus are carrying valid data during the current bus cycle. They are inputs to the 82C567 for CPU cycles and outputs for master or DMA cycles. In the case of cacheable reads, all eight bytes of data are driven to the CPU, regardless of the state of the byte enables. The byte enable signals indicate the type of special cycle when M/IO# = D/C# = 0 and W/R# = 1. BE[7:0]# have internal pull-downs that are activated when HLDA is active.

82C567 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
M/IO#	205	I-TTL		**Memory/Input-Output:** M/IO#, D/C#, and W/R# define CPU bus cycles. Interrupt acknowledge cycles are forwarded to the PCI bus as PCI interrupt acknowledge cycles. All I/O cycles and any memory cycles that are not directed to memory controlled by the DRAM interface of the 82C567 are forwarded to PCI.
D/C#	206	I-TTL		**Data/Control**: D/C#, M/IO#, and W/R# define CPU bus cycles. (See M/IO# definition above.)
W/R#	207	I-TTL	Cycle Multiplexed	**Write/Read:** W/R#, D/C#, and M/IO# define CPU bus cycles. (See M/IO# definition above.)
INV		I-TTL		**Invalidate:** Pin 207 also serves as an output signal and is used as INV for L1 cache during an inquire cycle.
DIRTYO		I/O-TTL (4mA)		**Dirty Output:** Pin 207 also serves as an output signal and is used as DIRTYO for L2 cache during an inquire cycle. If a combined Tag/Dirty RAM implementation is being used, then the W/R# pin does not serve as a DIRTYO pin.
ADS#	2	I/O-TTL (4mA)		**Address Strobe:** The CPU asserts ADS# to indicate that a new bus cycle is beginning. ADS# is driven active in the same clock as the address, byte enables, and cycle definition signals. ADS# has an internal pull-up resistor that is disabled when the system is in the Suspend mode.
BRDY#	3	O (8mA)		**Burst Ready:** BRDY# indicates that the system has responded in one of three ways: 1) Valid data has been placed on the CPU data bus in response to a read, 2) CPU write data has been accepted by the system, or 3) the system has responded to a special cycle.
NA#	4	O (8mA)		**Next Address:** This signal is connected to the CPU's NA# pin to request pipelined addressing for local memory cycle. The 82C567 asserts NA# for one clock when the system is ready to accept a new address from the CPU, even if all data transfers for the current cycle have not completed. The 3.3V Pentium processor and the M1 processor support pipelined memory accesses, however, the K5 processor does not support this feature.
KEN#	5	O (8mA)	Cycle Multiplexed	**Cache Enable:** This pin is connected to the KEN# input of the CPU and is used to determine whether the current cycle is cacheable.
LMEM#		O (8mA)		**Local Memory Accessed:** During master cycles, the 82C567 asserts this signal to inform the 82C568 that local system memory needs to be accessed. The 82C568 is then responsible for providing the data path to the corresponding master.

82C567 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
EADS#	6	O (8mA)	Cycle Multiplexed	**External Address Strobe:** This output indicates that a valid address has been driven onto the CPU address bus by an external device. This address will be used to perform an internal cache inquiry cycle when the CPU samples EADS# active.
WB/WT#		O (8mA)		**Write-Back/Write-Through**: Pin 6 is also used to control write-back or write-through policy for the primary cache during CPU cycles.
HITM#	49	I-TTL		**Hit Modified:** Indicates that the CPU has had a hit on a modified line in its internal cache during an inquire cycle. It is used to prepare for write-back.
CACHE#	51	I-TTL		**Cacheability:** This input is connected to the CACHE# pin of the CPU. It goes active during a CPU initiated cycle to indicate when, an internal cacheable read cycle or a burst write-back cycle, occurs.
SMIACT#	50	I-TTL		**System Management Interrupt Active:** The CPU asserts SMIACT# in response to the SMI# signal to indicate that it is operating in System Management Mode (SMM).

3.3.3 Cache Control Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
ECDOE#	54	O (8mA)		**Even Bank Cache Output Enable:** This signal is connected to the output enables of the SRAMs in the even bank of the L2 cache to enable data read.
ECAWE#	56	O (8mA)	SYSCFG 11h[3] = 0	**Even Bank Cache Write Enable:** For asynchronous L2 cache operations, this pin becomes ECAWE# and is connected to the write enables of the SRAMs in the even bank of the L2 cache to enable data update.
CACS0O#		O (8mA)	SYSCFG 11h[3] = 1	**Bank 0 Synchronous SRAM Chip Select**: For synchronous L2 cache operation, this pin provides the chip select for the first bank (synchronous L2 cache is always non-interleaved).
CACS[7:0]#	62:69	O (4mA)	SYSCFG 11h[3] = 0	**Cache Chip Selects 7 through 0:** For asynchronous L2 cache operations these pins become chip selects and are connected to the chip selects of the SRAMs in the L2 cache in both banks to enable data read/write operations.
CAWE[7:0]#		O (4mA)	SYSCFG 11h[3] = 1	**Cache Write Enables 7 through 0:** For synchronous L2 cache operation these pins become cache write enables for the SRAMs.
TAG[7:0]	191:198	I/O-TTL (4mA)		**Tag RAM Data Bits 7 through 0:** Normally input signals, they become outputs whenever TAGWE# is activated to write new Tags to the Tag RAM. If using a combined Tag/Dirty RAM implementation and a 7-bit Tag is used, then TAG0 functions as the Dirty I/O bit. If using the Sony cache module, then TAG1 and TAG2 are connected to the START# output from the module and TAG3 is connected to the BOFF# output from the module. The remaining TAG bits are unused.

82C567 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
TAGWE#	199	O (8mA)		**Tag RAM Write Enable:** This control strobe is used to update the Tag RAM with the valid Tag of the new cache line that replaces the current one during external cache read miss cycles. If using a combined Tag/Dirty RAM implementation, this signal functions as both the TAGWE# and DIRTYWE#.
MEMR#	55	I	SYSCFG 21h[1] = 1 and 0Eh[6] = 1	**Memory Read Command:** This input is connected to pin 78 (MEMR#) of the 82C568 and is used to monitor ISA memory read operations. If this pin is configured as an input and not connected to MEMR#, a 10K pull-up is recommended on this pin.
OCDOE#		O (8mA)	SYSCFG 21h[1] = 1 and 0Eh[6] = 0	**Odd Bank Cache Output Enable:** This signal is connected to output enables of the SRAMs in the odd bank of the L2 cache to enable data read.
HLDA		I-TTL	SYSCFG 21h[1] = 0 and 0Eh[6] = 0	**CPU Hold Acknowledge:** This input is connected to the HLDA pin of the CPU. HLDA indicates, in response to a HOLD, when the CPU has relinquished bus control to another bus master.
SDRAS#		O (8mA)	If SYSCFG 29h[3:0] (any bit) = 1	**SDRAM Row Address Strobe:** This output is part of the SDRAM command combination. This pin should be connected to the SDRAM through a damping resistor.
MEMW#	57	I	SYSCFG 21h[1] = 1 and 0Eh[6] = 1	**Memory Write Command:** This input is connected to pin 79 (MEMW#) pin of the 82C568 and is used to monitor ISA memory write operations. If this pin is configured as an input and not connected to MEMW#, a 10K pull-up is recommended on this pin.
OCAWE#		O (8mA)	SYSCFG 11h[3] = 0	**Odd Bank Cache Write Enable:** For asynchronous L2 cache operations this pin becomes OCAWE# and is connected to the write enables of the SRAMs in the odd bank of the L2 cache to enable data update.
CACS1O#		O (8mA)	SYSCFG 11h[3] = 1	**Bank 1 Synchronous SRAM Chip Select:** For synchronous L2 cache operation, this pin provides the chip select for the second bank (synchronous L2 cache is always non-interleaved).
HOLD		O (8mA)	SYSCFG 21h[1] = 0	**CPU Hold Request:** This output is connected to the HOLD input of the CPU. HOLD requests that the CPU allow another bus master complete control of its buses. In response to HOLD going active, the CPU will float most of its output and bidirectional pins and then assert HLDA.
SDCAS#		O (8mA)	If SYSCFG 29h[3:0] (any bit) = 1	**SDRAM Column Address Strobe:** This output is part of the SDRAM command combination. This pin should be connected to the SDRAM through a damping resistor.
DIRTYI	190	I/O-TTL (4mA)		**Dirty Bit:** This input signal represents the dirty bit of the Tag RAM and is used to indicate whether a corresponding cache line has been overwritten. If using a combined Tag/Dirty implementation, this pin becomes bidirectional. If using a 7-bit Tag in a combined Tag/Dirty RAM implementation, then this pin is not used.

82C567 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
DIRTYWE#	189	O (8mA)	SYSCFG 19h[7] = 0 and 2Eh[7] = 0	**Dirty RAM Write Enable:** This control strobe is used to update the dirty bit RAM when a cache write hit occurs. A cache write hit will set the dirty bit for the currently accessed cache line. If using a combined Tag/Dirty implementation, this signal is not used to update the Dirty RAM.
RAS5#		O (8mA)	SYSCFG and 2Eh[7] = 0	**Row Address Strobe Bit 5:** Each RAS# signal corresponds to a unique DRAM bank. Depending on the kind of DRAM modules being used, this signal may or may not need to be buffered externally. This signal, however, should be connected to the corresponding DRAM RAS# line through a damping resistor.
SDCKE		O (8mA)	SYSCFG 19h[7] = X and 2Eh[7] = 1	**SDRAM Clock Enable:** This output is used to control the internal clocking of SDRAM and is also used for power-down mode when SDRAM is idle. During normal operation of SDRAM, SDCKE will be high.
ECA4	59	O (8mA)	SYSCFG 11h[3] = 0	**Even Cache Address 4:** For an asynchronous L2 cache, if a single bank is used, this pin is mapped from HA4 and connected to the second LSB of the cache SRAMs address inputs. For a double bank configuration, it is connected to the LSB of the cache SRAMs address input in the even bank.
ADSC#		O (8mA)	SYSCFG 11h[3] = 1	**Controller Address Strobe**: For a synchronous L2 cache, this pin is connected to the ADSC# input of the synchronous SRAMs.
ECA3	60	O (8mA)	SYSCFG 11h[3] = 0 and SYSCFG 08h[7] = 1 (single bank, non-interleaved)	**Even Cache Address 3:** For asynchronous L2 cache operations in a single bank configuration, this pin takes on the functionality of ECA3 and is mapped from HA3 and connected to the cache SRAMs LSB address input.
OCA4		O (8mA)	SYSCFG 11h[3] = 0 and SYSCFG 08h[7] = 0 (double bank, interleaved)	**Odd Cache Address 4:** For asynchronous L2 cache operations in a double bank configuration, this pin takes on the functionality of OCA4 and is mapped from HA4 and connected to the LSB address input of the SRAMs in the odd bank.
ADV#		O (8mA)	SYSCFG 11h[3] = 1	**Advance Output:** For synchronous L2 cache operation, this pin becomes the advance output and is connected to the ADV# input of the synchronous SRAMs.
HACALE	58	O (8mA)		**Cache Address Latch Enable:** It is used to latch the CPU address and generate latched cache addresses for the L2 cache.

82C567 Signal Descriptions (cont.)

3.3.4 DRAM Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
RAS[3:0]#	87:90	O (4mA)	Cycle Multiplexed	**Row Address Strobe Bits 3 through 0:** Each RAS# signal corresponds to a unique DRAM bank. Depending on the kind of DRAM modules being used, these signals may or may not need to be buffered externally. These signals, however, should be connected to the corresponding DRAM RAS# lines through a damping resistor. The default drive current on these lines is 4mA, however, by setting SYSCFG 18h[4] = 1, it can be increased to 16mA. RAS4# is pin-wise programmable with MA11 and RAS5# is pin-wise programmable with DIRTYWE#. RAS0# is used to share the bank with the GUI in a Unified Memory Architecture system design.
SDCS[3:0]#		O (4mA)		**SDRAM Chip Selects 3 through 0:** Each SDCS# output corresponds to a unique SDRAM Bank. When active, the SDRAM will accept the command from the chipset. These outputs must be connected to the SDRAM banks through a damping resistor.
CAS[7:0]#	91:95, 97:99	O (8mA)	Cycle Multiplexed	**Column Address Strobe Bits 7 through 0:** The CAS[7:0]# outputs correspond to the eight bytes for each DRAM bank. Each DRAM bank has a 64-bit data bus. These signals are typically connected directly to the DRAMs CAS# inputs through a damping resistor.
SDDQM[7:0]#		O (8mA)		**Synchronous DRAM Data Mask Control Bits 7 through 0:** During SDRAM read cycles, these outputs control whether the DRAM output buffers are driven on the MD bus or not. During SDRAM write cycles, these outputs control whether MD data will be written into the memory device or not.
DWE#	86	O (16mA)	Cycle Multiplexed	**DRAM Write Enable:** This signal is typically buffered externally before connection to the WE# input of the DRAMs. The default drive current on this line is 4mA, however, by setting SYSCFG 18h[4] = 1, it can be increased to 16mA.
SDWE#		O (16mA)		**SDRAM Write Enable:** This output is the write enable signal for SDRAM.
MPERR#	179	I/O (4mA)	SYSCFG 08h[4] = 1 and PCIDV0 45h[3] = 1	**Memory Parity:** This signal is an input to the 82C567 from the 82C566. The 82C567 generates PEN# internally if the corresponding register is programmed to enable parity. The 82C567 qualifies the MPERR# signal with the internally generated PEN#.
IADV#		O (4mA)	SYSCFG 08h[4] = 0 and PCIDV0 45h[3] = 0	**Input Pointer Advance:** This signal is connected to the 82C566 and is used to input the next quad-word into the deep buffer.

82C567 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
MA11	71	O (4mA)	SYSCFG 19h[3] = 0	**Memory Address Bus Bit 11:** A part of the multiplexed row/column address lines to the DRAMs. Depending on the kind of DRAM modules being used, this signal may or may not need to be buffered externally. The default drive current on the MA[11:0] lines is 4mA, however, by setting SYSCFG 18h[4] = 1, it can be increased to 16mA. As MA11, 8Mx36 and 16Mx36 SIMMs will be supported.
RAS4#		O (4mA)	SYSCFG 19h[3] = 1	**Row Address Strobe Bit 4:** Each RAS# signal corresponds to a unique DRAM bank. Depending on the kind of DRAM modules being used, this signal may or may not need to be buffered externally. This signal, however, should be connected to the corresponding DRAM RAS# line through a damping resistor. As RAS4#, SIMM sizes above 4Mx36 will not be supported and a maximum of 192MB of DRAM will be supported.
MA[10:0]	72, 73, 75:83	O (4mA)		**Memory Address Bus Bits 10 through 0:** Multiplexed row/column address lines to the DRAMs. Depending on the kind of DRAM modules being used, these signals may or may not need to be buffered externally. The default drive current on the MA[11:0] lines is 4mA, however, by setting SYSCFG 18h[4] = 1, it can be increased to 16mA.
NVMCS	123	O (4mA)		**NVRAM Chip Select:** If the current cycle has been decoded as an access to NVRAM, then this pin is used to issue the chip select signal. NVRAM is used for storing the system configuration information and is required for "plug and play" support. The NVRAM must sit on the XD bus. During power-on reset, if TMOD# is sampled low, the NVMCS pin will be floated. This pin requires an external pull-up.

3.3.5 UMA Arbiter Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
MREQ#	178	I-TTL		**Memory Request:** This input is connected to the MREQ# output of the GUI device for a unified memory scheme. This signal indicates to the 82C567 that the GUI wants control of the memory bus.
MGNT#	177	O (4mA)		**Memory Grant:** This output is connected to the MGNT# input of the GUI device for a unified memory scheme. This signal indicates to the GUI that is has been granted control of the memory bus.

82C567 Signal Descriptions (cont.)

3.3.6 ISA Bus Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
LA[23:9]	101:103, 106:117	I/O-CMOS (8mA)		**System Address Bus:** LA[23:9] and SA[8:0] on the 82C568 provide the memory and I/O access on the ISA bus. The addresses are outputs when the 82C567 owns the ISA bus and are inputs when an external ISA master owns the bus. LA[23:9] have internal pull-ups which are disabled when in the Suspend mode.
MASTER#	122	I-CMOS		**Master:** An ISA bus master asserts MASTER# to indicate that it has control of the ISA bus. Before the ISA master can assert MASTER#, it must first sample DACK# active. Once MASTER# is asserted, the ISA master has control of the ISA bus until it negates MASTER#.

3.3.7 Bus Arbiter Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
HREQ	181	I-TTL		**Hold Request:** Master or DMA cycle request from the 82C568. An external pull-up is required for normal operation.
TMOD#		I-TTL	Strap Option at RESET	**Strap Signal for ATE (Automatic Test Equipment) Test Mode Operation:** During power-up reset, this is the strap pin to enter the test mode operation. If TMOD ="high" during power-up reset, it means normal operation. If TMOD ="low", the system enters test mode.
AHOLD	7	O (8mA)		**Address Hold:** This signal is used to tristate the CPU address bus for internal cache snooping.
LOCK#	9	I-TTL		**CPU Bus Lock:** The processor asserts LOCK# to indicate the current bus cycle is locked. It is used to generate PLOCK# for the PCI bus. LOCK# has an internal pull-down resistor that is engaged when HLDA is active.
BOFF#	8	O (8mA)		**Back-off:** This pin is connected to the BOFF# input of the CPU. This signal is asserted by the 82C567 during PCI/retry cycles.

3.3.8 PCI Bus Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
C/BE[3:0]#	125:128	I/O-TTL (PCI)		**PCI Bus Command and Byte Enables 3 through 0:** C/BE[3:0]# are driven by the current bus master (CPU or PCI) during the address phase of a PCI cycle to define the PCI command, and during the data phase as the PCI byte enables. The PCI commands indicate the current cycle type, and the PCI byte enables indicate which byte lanes carry meaningful data. C/BE[3:0]# are outputs from the 82C567 during CPU cycles that are directed to the PCI bus. C/BE[3:0]# are inputs during PCI master cycles.

82C567 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
FRAME#	133	I/O-TTL (PCI)		**Cycle Frame:** Every CPU cycle is translated by the 82C567 to a PCI cycle if it is not a local memory cycle. FRAME# is asserted by the bus master, 82C567 (CPU) or PCI to indicate the beginning and the duration of an access. FRAME# is an input when the 82C567 acts as a slave.
IRDY#	134	I/O-TTL (PCI)		**Initiator Ready:** The assertion of IRDY# indicates the current bus master's ability to complete the current data phase. IRDY# works in conjunction with TRDY# to indicate when data has been transferred. A data phase is completed on each clock that TRDY# and IRDY# are both sampled asserted. Wait states are inserted until both IRDY# and TRDY# are asserted together. IRDY# is an output from the 82C567 during CPU cycles to the PCI bus. IRDY# is an input when the 82C567 acts as a slave.
TRDY#	135	I/O-TTL (PCI)		**Target Ready:** TRDY# indicates the target device's ability to complete the current data phase of the transaction. It is used in conjunction with IRDY#. A data phase is completed on each clock that TRDY# and IRDY# are both sampled asserted. Wait states are inserted on the bus until both IRDY# and TRDY# are asserted together. TRDY# is an output from the 82C567 when the 82C567 is the PCI slave. TRDY# is an input when the 82C567 is a master.
DEVSEL#	136	I/O-TTL (PCI)		**Device Select:** When asserted, DEVSEL# indicates that the driving device has decoded its address as the target of the current access. DEVSEL# is an output of the 82C567 when 82C567 is a PCI slave. During CPU-to-PCI cycles, DEVSEL# is an input. It is used to determine if any device has responded to the current bus cycle, and to detect a target abort cycle. Master abort termination results if no decode agent exists in the system, and no one asserts DEVSEL# within a fixed number of clocks.
STOP#	137	I/O-TTL (PCI)		**Stop:** STOP# indicates that the current target is requesting the master to stop the current transaction. This signal is used in conjunction with DEVSEL# to indicate disconnect, target abort, and retry cycles. When the 82C567 is acting as a master on the PCI bus, if STOP# is sampled active on a rising edge of LCLK, FRAME# is negated within a maximum of three clock cycles. STOP# may be asserted by the 82C567. Once asserted, STOP# remains asserted until FRAME# is negated.
AD[31:0]	138:141, 143:155, 158:164, 166:173	I/O-TTL (PCI)		**PCI Address and Data:** AD[31:0] are bidirectional address and data lines of the PCI bus. The AD[31:0] signals sample or drive the address and data on the PCI bus. During power-up reset, the 82C567 will drive the AD lines by default. This bus also serves as a conduit for receiving address information during ISA master cycles. The 82C568 conveys the SA[8:0] information to the 82C567 on the AD lines.
PLOCK#	132	I/O-TTL (PCI)		**PCI Lock:** PLOCK# is used to indicate an atomic operation that may require multiple transactions to complete. When PLOCK# is asserted, non-exclusive transactions may proceed to an address that is not currently locked.

82C567 Signal Descriptions (cont.)

3.3.9 Universal Serial Bus Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
USBDAT+	201	I/O-Analog		**Differential Serial Data**
USBDAT–	202	I/O-Analog		**Differential Serial Data**
MSGN2S	121	I/O-TTL (4mA)	SYSCFG 2Eh[5] = 1	**82C567 to 82C568 Message:** This pin is used to communicate between the 82C567 and 82C568. This pin should be connected directly to the MSGN2S pin of the 82C568. It is recommended to put a pull-down 4.7K in on this pin in Silicon Revision 1.0.
AEN		I	SYSCFG 2Eh[5] = 0	**Address Enable:** This input is connected to the AEN pin of the 82C568 to monitor ISA bus activity.
MSGS2N	124	I	TBD	**82C568 to 82C567 Message:** This pin is used to communicate between the 82C568 and 82C567. This pin should be connected directly to the MSGS2N pin of the 82C568.
USBGNT#		I	TBD	**Universal Serial Bus Grant:** This input comes from the 82C568 and informs the USB device in the 82C567 that the PCI bus is granted to it. This pin should be connected directly to the USBGNT# pin of the 82C568.

3.3.10 82C566/82C568 Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
OADV#	185	O (8mA)		**Output Pointer Advance:** This signal indicates to the 82C566 to output the next quad-word in the deep buffer. A 10 ohm series resistor is recommended on this signal.
DBCTL0#	118	O (8mA)		**Data Buffer Control Line 0:** DBCTL0# is used to control the data paths when the deep buffers are enabled. A 10 ohm series resistor is recommended on this signal.
MMDOE#	184	O (8mA)	Cycle Multiplexed	**MMD Output Enable**: DBCTL0#, MDOE#+DBCTL2#, MMDOE#+DBCTL1#, and HDOE# form the encoded commands that are sent out to the 82C566. These commands inform the 82C566 about the current cycle type and enable it to perform the appropriate data steering, latching and direction controls. A 10 ohm series resistor is recommended on this signal.
DBCTL1#				**Data Buffer Control Line 1:** DBCTL1# is used to control the data paths when the deep buffers are enabled. A 10 ohm series resistor is recommended on this signal.

82C567 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
MDOE#	182	O (8mA)	Cycle Multiplexed	**Memory Data Output Enable:** DBCTL0#, MDOE#+DBCTL2#, MMDOE#+DBCTL1#, and HDOE# form the encoded commands that are sent out to the 82C566. When asserted, this signal enables data to be put out on the MD bus. MDOE# is asserted for CPU writes to cache/DRAM, CPU writes to PCI, PCI reads from cache/DRAM, L2 cache write-back cycles, and PCI writes to DRAM. A 10 ohm series resistor is recommended on this signal.
DBCTL2#				**Data Buffer Control Line 2:** DBCTL2# is used to control the data paths when the deep buffers are enabled. A 10 ohm series resistor is recommended on this signal.
HDOE#	183	O (8mA)		**Host Data Output Enable:** DBCTL0#, MDOE#+DBCTL2#, MMDOE#+DBCTL1#, and HDOE# form the encoded commands that are sent out to the 82C566. When asserted, this signal enables data to be put out on the HD bus. HDOE# is asserted for CPU reads from DRAM/PCI bus, PCI writes to cache, CPU linefills, Suspend mode indication, and reset state indication. A 10 ohm series resistor is recommended on this signal.
DLE1#	187	O (8mA)	Cycle Multiplexed	**Data Latch Enable 1:** This line is connected to the 82C566 DLE1# pin and is used to latch the HD and MD data bus depending on which cycle is occurring. A 10 ohm series resistor is recommended on this signal.
PPSEL				**Ping-Pong Buffer Select:** This signal is connected to the 82C566 and is used to select between the ping-pong buffers. A 10 ohm series resistor is recommended on this signal.
DLE0#	186	O (8mA)		**Data Latch Enable 0:** This line is connected to the 82C566 DLE0# pin and is used to latch the HD and MD data bus depending on which cycle is occurring. A 10 ohm series resistor is recommended on this signal.
MDLE#	180	O (4mA)		**Memory Data Latch Enable:** This signal is connected to the MDLE# pin of the 82C568 and controls the data flow from PCI AD[31:0] bus to the high 32-bit memory data bus, MMD[31:0], and vice versa. It is used to latch the data during CPU writes to PCI and PCI writes to DRAM and L2 cache.
3VDRAM#		I-TTL	Strap Option at RESET	**Strap option for 3.3V DRAM:** At power-up reset, this pin functions as a strapping option for 3.3V or 5.0V DRAM operation. If 3.3V DRAM operation, an external pull-down is required. If 5.0V DRAM operation, an external pull-up is required. This pin should be pulled up externally for the Viper-MAX Chipset.
SDRAM_EDO	120	O (8mA)		**SDRAM and EDO Cycle:** This signal is connected to the 82C566 and is used to select the special data paths required for SDRAM and EDO memory cycles. A 10 ohm series resistor is recommended on this signal.

82C567 Signal Descriptions (cont.)

3.3.11 Power and Ground Pins

Signal Name	Pin No.	Signal Type	Selected By	Signal Description
GND	15, 52, 61, 84, 96, 105, 129, 142, 156, 165, 176, 200, 208	I-G		**Ground Connection**
VCC	104, 119, 130, 157, 174, 188, 203	I-P		**Power Connection:** 5.0V power plane
VCC3	1, 38, 53	I-P		**Power Connection:** 3.3V power plane
VCC3/5	74, 85	I-P		**Power Connection:** These pins power the memory interface at either 3.3V or 5.0V.

Figure 3-3 82C568 Pin Diagram

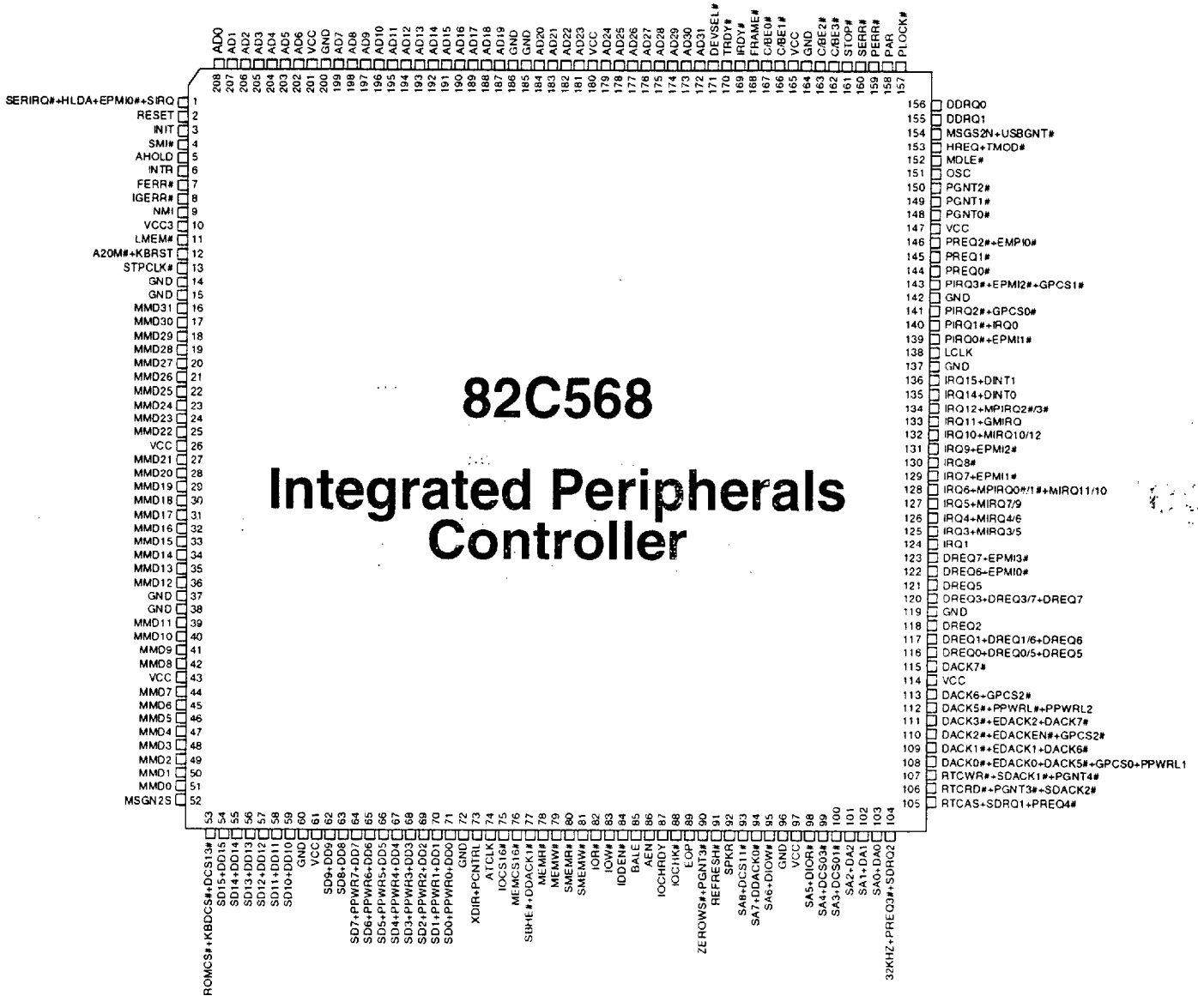


Table 3-7 82C568 Numerical Pin Cross-Reference List

Pin No.	Pin Name	Pin Type	Drive (mA)	Pwr (V)
1	SERIRQ#	I		5.0
	HLDA			
	EPMI0#			
	SIRQ			
2	RESET	I-S		3.3
3	INIT	O	4	3.3
4	SMI#	O	4	3.3
5	AHOLD	I-TTL		3.3
6	INTR	I/O-TTL	4	3.3
7	FERR#	I-TTL		3.3
8	IGERR#	O	4	3.3
9	NMI	O	4	3.3
10	VCC3	I-P		
11	LMEM#	I-TTL		3.3
12	A20M#	O	4	3.3
	KBRST	I-TTL		
13	STPCLK#	O	4	3.3
14	GND	I-G		
15	GND	I-G		
16	MMD31	I/O-TTL	4	5.0
17	MMD30	I/O-TTL	4	5.0
18	MMD29	I/O-TTL	4	5.0
19	MMD28	I/O-TTL	4	5.0
20	MMD27	I/O-TTL	4	5.0
21	MMD26	I/O-TTL	4	5.0
22	MMD25	I/O-TTL	4	5.0
23	MMD24	I/O-TTL	4	5.0
24	MMD23	I/O-TTL	4	5.0
25	MMD22	I/O-TTL	4	5.0
26	VCC	I-P		
27	MMD21	I/O-TTL	4	5.0
28	MMD20	I/O-TTL	4	5.0
29	MMD19	I/O-TTL	4	5.0
30	MMD18	I/O-TTL	4	5.0
31	MMD17	I/O-TTL	4	5.0
32	MMD16	I/O-TTL	4	5.0
33	MMD15	I/O-TTL	4	5.0
34	MMD14	I/O-TTL	4	5.0
35	MMD13	I/O-TTL	4	5.0
36	MMD12	I/O-TTL	4	5.0
37	GND	I-G		
38	GND	I-G		
39	MMD11	I/O-TTL	4	5.0
40	MMD10	I/O-TTL	4	5.0
41	MMD9	I/O-TTL	4	5.0
42	MMD8	I/O-TTL	4	5.0
43	VCC	I-P		
44	MMD7	I/O-TTL	4	5.0
45	MMD6	I/O-TTL	4	5.0
46	MMD5	I/O-TTL	4	5.0
47	MMD4	I/O-TTL	4	5.0
48	MMD3	I/O-TTL	4	5.0
49	MMD2	I/O-TTL	4	5.0
50	MMD1	I/O-TTL	4	5.0
51	MMD0	I/O-TTL	4	5.0
52	MSGN2S	I		5.0
53	ROMCS#	O	4	5.0
	KBDCS#			
	DCS13#			
54	SD15	I/O-TTL	8	5.0
	DD15			
55	SD14	I/O-TTL	8	5.0
	DD14			
56	SD13	I/O-TTL	8	5.0
	DD13			
57	SD12	I/O-TTL	8	5.0
	DD12			
58	SD11	I/O-TTL	8	5.0
	DD11			
59	SD10	I/O-TTL	8	5.0
	DD10			
60	GND	I-G		
61	VCC	I-P		
62	SD9	I/O-TTL	8	5.0
	DD9			
63	SD8	I/O-TTL	8	5.0
	DD8			
64	SD7	I/O-TTL	8	5.0
	PPWR7			
	DD7			
65	SD6	I/O-TTL	8	5.0
	PPWR6			
	DD6			
66	SD5	I/O-TTL	8	5.0
	PPWR5			
	DD5			
67	SD4	I/O-TTL	8	5.0
	PPWR4			
	DD4			
68	SD3	I/O-TTL	8	5.0
	PPWR3			
	DD3			
69	SD2	I/O-TTL	8	5.0
	PPWR2			
	DD2			
70	SD1	I/O-TTL	8	5.0
	PPWR1			
	DD1			
71	SD0	I/O-TTL	8	5.0
	PPWR0			
	DD0			
72	GND	I-G		
73	XDIR	I/O-TTL	4	5.0
	PCNTRL			
74	ATCLK	O	8	5.0
75	IOCS16#	I-TTL		5.0
76	MEMCS16#	I/O-TTL	8	5.0
77	SBHE#	I/O	8	5.0
	DDACK1#	O		
78	MEMR#	I/O-TTL	8	5.0
79	MEMW#	I/O-TTL	8	5.0
80	SMEMR#	O	8	5.0
81	SMEMW#	O	8	5.0
82	IOR#	I/O-S	8	5.0
83	IOW#	I/O-S	8	5.0
84	IDEEN#	O	8	5.0
85	BALE	O	8	5.0
86	AEN	I/O-TTL	8	5.0
87	IOCHRDY	I/O-TTL	8	5.0
88	IOCHK#	I-TTL		5.0
89	EOP	I/O-TTL	8	5.0
90	ZEROWS#	I/O-TTL	PCI	5.0
	PGNT3#	O		
91	REFRESH#	I/O-TTL	8	5.0
92	SPKR	O	16	5.0
93	SA8	I/O-TTL	8	5.0
	DCS11#	O		
94	SA7	I/O-TTL	8	5.0
	DDACK0#	O		
95	SA6	I/O-TTL	8	5.0
	DIOW#	O		
96	GND	I-G		
97	VCC	I-P		
98	SA5	I/O-TTL	8	5.0
	DIOR#	O		
99	SA4	I/O-TTL	8	5.0
	DCS03#	O		
100	SA3	I/O-TTL	8	5.0
	DCS01#	O		
101	SA2	I/O-TTL	8	5.0
	DA2			
102	SA1	I/O-TTL	8	5.0
	DA1			
103	SA0	I/O-TTL	8	5.0
	DA0			
104	32KHZ	I-S		5.0
	PREQ3#			
	SDRQ2	I		
105	RTCAS	O	4	5.0
	SDRQ1	I		
	PREQ4#	I		
106	RTCRD#	O	4	5.0
	PGNT3#			
	SDACK2#			
107	RTCWR#	O	4	5.0
	SDACK1#		6	
	PGNT4#		4	

82C568 Numerical Pin Cross-Reference List (cont.)

Pin No.	Pin Name	Pin Type	Drive (mA)	Pwr (V)
108	DACK0#	O	6	5.0
	EDACK0			
	DACK5#			
	GPCS0#		4	
	PPWRL1		4	
109	DACK1#	O	6	5.0
	EDACK1			
	DACK6#			
110	DACK2#	O	6	5.0
	EDACKEN#			
	GPCS2#			
111	DACK3#	O	6	5.0
	EDACK2			
	DACK7#			
112	DACK5#	O	6	5.0
	PPWRL#		4	
	PPWRL2		4	
113	DACK6#	O	6	5.0
	GPCS2#			
114	VCC	I-P		
115	DACK7#	O	6	5.0
116	DREQ0	I-TTL		5.0
	DREQ0/5			
	DREQ5			
117	DREQ1	I-TTL		5.0
	DREQ1/6			
	DREQ6			
118	DREQ2	I-TTL		5.0
119	GND	I-G		
120	DREQ3	I-TTL		5.0
	DREQ3/7			
	DREQ7			
121	DREQ5	I-TTL		5.0
122	DREQ6	I-TTL		5.0
	EPMI0#			
123	DREQ7	I-TTL		5.0
	EPMI3#			
124	IRQ1	I-TTL		5.0
125	IRQ3	I-TTL		5.0
	MIRQ3/5			
126	IRQ4	I-TTL		5.0
	MIRQ4/6			
127	IRQ5	I-TTL		5.0
	MIRQ7/9			
128	IRQ6	I-TTL		5.0
	MPIRQ0#/1#			
	MIRQ11/15			
129	IRQ7	I-TTL		5.0
	EPMI1#			
130	IRQ8#	I-TTL		5.0
131	IRQ9	I-TTL		5.0
	EPMI2#			

Pin No.	Pin Name	Pin Type	Drive (mA)	Pwr (V)
132	IRQ10	I-TTL		5.0
	MIRQ10/12			
133	IRQ11	I-TTL		5.0
	GMIRQ			
134	IRQ12	I-TTL		5.0
	MPIRQ2#/3#			
135	IRQ14	I-TTL		5.0
	DINT0			
136	IRQ15	I-TTL		5.0
	DINT1			
137	GND	I-G		
138	LCLK	I-TTL		5.0
139	PIRQ0#	I-TTL		5.0
	EPMI1#			
140	PIRQ1#	I-TTL		5.0
	IRQ0			
141	PIRQ2#	I-TTL		5.0
	GPCS0#	O	4	
142	GND	I-G		5.0
143	PIRQ3#	I-TTL		5.0
	EPMI2#			
	GPCS1#	O	4	
144	PREQ0#	I-TTL		5.0
145	PREQ1#	I-TTL		5.0
146	PREQ2#	I-TTL		5.0
	EPMI0#			
147	VCC	I-P		
148	PGNT0#	O	PCI	5.0
149	PGNT1#	O	PCI	5.0
150	PGNT2#	O	PCI	5.0
151	OSC	I-TTL		5.0
152	MDLE#	I-TTL		5.0
153	HREQ	O	4	5.0
	TMOD#	I-TTL		
154	MSGS2N	O	4	5.0
	USBGNT#			
155	DDRQ1	I-TTL		5.0
156	DDRQ0	I-TTL		5.0
157	PLOCK#	I-TTL		5.0
158	PAR	I/O-TTL	PCI	5.0
159	PERR#	I/O-TTL	PCI	5.0
160	SERR#	I/O-TTL	PCI	5.0
161	STOP#	I/O-TTL	PCI	5.0
162	C/BE3#	I/O-TTL	PCI	5.0
163	C/BE2#	I/O-TTL	PCI	5.0
164	GND	I-G		
165	VCC	I-P		
166	C/BE1#	I/O-TTL	PCI	5.0
167	C/BE0#	I/O-TTL	PCI	5.0
168	FRAME#	I/O-TTL	PCI	5.0
169	IRDY#	I/O-TTL	PCI	5.0
170	TRDY#	I/O-TTL	PCI	5.0

Pin No.	Pin Name	Pin Type	Drive (mA)	Pwr (V)
171	DEVSEL#	I/O-TTL	PCI	5.0
172	AD31	I/O-TTL	PCI	5.0
173	AD30	I/O-TTL	PCI	5.0
174	AD29	I/O-TTL	PCI	5.0
175	AD28	I/O-TTL	PCI	5.0
176	AD27	I/O-TTL	PCI	5.0
177	AD26	I/O-TTL	PCI	5.0
178	AD25	I/O-TTL	PCI	5.0
179	AD24	I/O-TTL	PCI	5.0
180	VCC	I-P		
181	AD23	I/O-TTL	PCI	5.0
182	AD22	I/O-TTL	PCI	5.0
183	AD21	I/O-TTL	PCI	5.0
184	AD20	I/O-TTL	PCI	5.0
185	GND	I-G		
186	GND	I-G		
187	AD19	I/O-TTL	PCI	5.0
188	AD18	I/O-TTL	PCI	5.0
189	AD17	I/O-TTL	PCI	5.0
190	AD16	I/O-TTL	PCI	5.0
191	AD15	I/O-TTL	PCI	5.0
192	AD14	I/O-TTL	PCI	5.0
193	AD13	I/O-TTL	PCI	5.0
194	AD12	I/O-TTL	PCI	5.0
195	AD11	I/O-TTL	PCI	5.0
196	AD10	I/O-TTL	PCI	5.0
197	AD9	I/O-TTL	PCI	5.0
198	AD8	I/O-TTL	PCI	5.0
199	AD7	I/O-TTL	PCI	5.0
200	GND	I-G		
201	VCC	I-G		
202	AD6	I/O-TTL	PCI	5.0
203	AD5	I/O-TTL	PCI	5.0
204	AD4	I/O-TTL	PCI	5.0
205	AD3	I/O-TTL	PCI	5.0
206	AD2	I/O-TTL	PCI	5.0
207	AD1	I/O-TTL	PCI	5.0
208	AD0	I/O-TTL	PCI	5.0

Table 3-8 82C568 Alphabetical Pin Cross-Reference List

Pin Name	Pin No.
AD0	208
AD1	207
AD2	206
AD3	205
AD4	204
AD5	203
AD6	202
AD7	199
AD8	198
AD9	197
AD10	196
AD11	195
AD12	194
AD13	193
AD14	192
AD15	191
AD16	190
AD17	189
AD18	188
AD19	187
AD20	184
AD21	183
AD22	182
AD23	181
AD24	179
AD25	178
AD26	177
AD27	176
AD28	175
AD29	174
AD30	173
AD31	172
AEN	86
AHOLD	5
ATCLK	74
A20M#+KBRST	12
BALE	85
C/BE0#	167
C/BE1#	166
C/BE2#	163
C/BE3#	162
DACK0#+EDACK0+ DACK5#+GPCS0#+ PPWRL1	108
DACK1#+EDACK1+ DACK6#	109
DACK2#+EDACKEN# +GPCS2#	110
DACK3#+EDACK2+ DACK7#	111
DACK5#+PPWRL#+ PPWRL2	112
DACK6#+GPCS2#	113
DACK7#	115
DDRQ0	156
DDRQ1	155
DEVSEL#	171
DREQ0+DREQ0/5+ DREQ5	116
DREQ1+DREQ1/6+ DREQ6	117
DREQ2	118
DREQ3+DREQ3/7+ DREQ7	120
DREQ5	121
DREQ6+EPMI0#	122
DREQ7+EPMI3#	123
EOP	89
FERR#	7
FRAME#	168
GND	14
GND	15
GND	37
GND	38
GND	60
GND	72
GND	96
GND	119
GND	137
GND	142
GND	164
GND	185
GND	186
GND	200
HREQ+TMOD#	153
IDEEN#	84
IGERR#	8
INIT	3
INTR	6
IOCHK#	88
IOCHRDY	87
IOCS16#	75
IOR#	82
IOW#	83
IRDY#	169
IRQ1	124
IRQ3+MIRQ3/5	125
IRQ4+MIRQ4/6	126
IRQ5+MIRQ7/9	127
IRQ6+MPIRQ0#/1#+ MIRQ11/15	128
IRQ7+EPMI1#	129
IRQ8#	130
IRQ9+EPMI2#	131
IRQ10+MIRQ10/12	132
IRQ11+GMIRQ	133
IRQ12+MPIRQ2#/3#	134
IRQ14+DINT0	135
IRQ15+DINT1	136
LCLK	138
LMEM#	11
MMD0	51
MMD1	50
MMD2	49
MMD3	48
MMD4	47
MMD5	46
MMD6	45
MMD7	44
MMD8	42
MMD9	41
MMD10	40
MMD11	39
MMD12	36
MMD13	35
MMD14	34
MMD15	33
MMD16	32
MMD17	31
MMD18	30
MMD19	29
MMD20	28
MMD21	27
MMD22	25
MMD23	24
MMD24	23
MMD25	22
MMD26	21
MMD27	20
MMD28	19
MMD29	18
MMD30	17
MMD31	16
MDLE#	152
MEMCS16#	76
MEMR#	78
MEMW#	79
MSGN2S	52
MSGS2N+USBGNT#	154
NMI	9
OSC	151
PAR	158
PERR#	159
PGNT0#	148
PGNT1#	149
PGNT2#	150
PIRQ0#+EPMI1#	139
PIRQ1#+IRQ0	140
PIRQ2#+GPCS0#	141
PIRQ3#+EPMI2#+ GPCS1#	143
PLOCK#	157
PREQ0#	144
PREQ1#	145
PREQ2#+EPMI0#	146
REFRESH#	91
RESET	2
ROMCS#+KBDCS#+ DCS13#	53
RTCAS+SDRQ1+ PREQ4#	105
RTCRD#+PGNT3#+ SDACK2#	106
RTCWR#+SDACK1#+ PGNT4#	107
SA0+DA0	103
SA1+DA1	102
SA2+DA2	101
SA3+DCS01#	100
SA4+DCS03#	99
SA5+DIOR#	98
SA6+DIOW#	95
SA7+DDACK0#	94
SA8+DCS11#	93
SBHE#+DDACK1#	77
SD0+PPWR0+DD0	71
SD1+PPWR1+DD1	70
SD2+PPWR2+DD2	69
SD3+PPWR3+DD3	68
SD4+PPWR4+DD4	67
SD5+PPWR5+DD5	66
SD6+PPWR6+DD6	65
SD7+PPWR7+DD7	64
SD8+DD8	63
SD9+DD9	62
SD10+DD10	59
SD11+DD11	58
SD12+DD12	57
SD13+DD13	56
SD14+DD14	55
SD15+DD15	54
SERIRQ#+HLDA+ EPMI0#+SIRQ	1
SERR#	160
SMEMR#	80
SMEMW#	81
SMI#	4
SPKR	92
STOP#	161
STPCLK#	13
TRDY#	170
VCC	26
VCC	43
VCC	61
VCC	97
VCC	114
VCC	147
VCC	165
VCC	180
VCC	201
VCC3	10
XDIR+PCNTRL	73
ZEROWS#+PGNT3#	90
32KHZ+PREQ3+ SDRQ2	104

3.4 82C568 Signal Descriptions

3.4.1 Reset and Clock Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
LCLK	138	I-TTL		**Local Bus Clock:** This is the same CLK signal that is also fed into the 82C567. It is used by the PCI bus state machine within the 82C568 and the 82C567. It is also used by the 82C568 to derive the AT clock signal. Another identical clock signal is used to clock the PCI and local bus devices.
ATCLK	74	O (8mA)		**AT Bus Clock:** This clock input to the ISA bus is programmable to be LCLK, LCLK/2, LCLK/3, and LCLK/4.
OSC	151	I-TTL		**Timer Oscillator:** This is the main clock used by the internal 8254 timers. It is connected to the 14.31818MHz oscillator.
INIT	3	O (4mA)		**CPU Initialize:** A shutdown cycle will trigger INIT, or a low-to-high transition of I/O Port 92h bit 0 will trigger INIT. If keyboard emulation is enabled (default), an INIT will be generated when a Port 64h write cycle with data FEh is decoded. If keyboard emulation has been disabled, then this signal will be triggered when it sees the KBRST from the keyboard.
RESET	2	I-S		**CPU Reset:** An output from the 82C567 in response to a PWRGD input.

3.4.2 ISA Bus and IDE Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
ROMCS#	53	O (4mA)	Cycle Multiplexed	**BIOS ROM Chip Select:** ROMCS# goes active on both reads and writes to the ROM area to support flash ROM.
KBDCS#		O (4mA)		**Keyboard Chip Select:** This output decodes accesses to the keyboard controller.
DCS13#		O (4mA)	IDEEN# will drive the 74F244 active to allow the signal to pass to the IDE interface.	**Secondary Drive Chip Select 3:** When configured as DCS13#, this pin functions as the chip select signal for the secondary IDE. After the 82C567 translates the CPU cycle to a PCI cycle, the 82C568 decodes the address on the AD lines and asserts this signal to select the command block register for the primary IDE. This signal should be buffered and this buffered output will be valid only when IDEEN# is asserted by the 82C568.

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
SBHE#	77	I/O-TTL (8mA)	Cycle Multiplexed	**System Byte High Enable:** When asserted, SBHE# indicates that a byte is being transferred on the upper byte (SD[15:8]) of the data bus. SBHE# is negated during refresh cycles. SBHE# is an output when the 82C568 owns the ISA bus.
DDACK1#		O (8mA)	IDEEN# will drive the 74F244 active to allow the signal to pass to the IDE interface.	**Secondary Drive DMA Acknowledge:** When configured as DDACK1#, this pin functions as the secondary drive DMA acknowledge signal. This signal should be buffered and this buffered output will be valid only when IDEEN# is asserted by the 82C568.
SA8	93	I/O-TTL (8mA)	Cycle Multiplexed	**System Address Bus Line 8:** The SA[8:0] and LA[23:9] signals on the 82C567 provide the address for memory and I/O accesses on the ISA bus. The addresses (SA[8:0]) are outputs when the 82C568 owns the ISA bus and are inputs when an external ISA master owns the ISA bus.
DCS11#		O (8mA)	IDEEN# will drive the 74F244 active to allow the signal to pass to the IDE interface.	**Secondary Drive Chip Select 1:** When configured as DCS11#, this pin functions as the chip select signal for the secondary IDE. After the 82C567 translates the CPU cycle to a PCI cycle, the 82C568 decodes the address on the AD lines and asserts this signal to select the command block register for the primary IDE. This signal should be buffered and this buffered output will be valid only when IDEEN# is asserted by the 82C568.
SA7	94	I/O-TTL (8mA)	Cycle Multiplexed	**System Address Bus Line 7:** Refer to the signal description for SA8.
DDACK0#		O (8mA)	IDEEN# will drive the 74F244 active to allow the signal to pass to the IDE interface.	**Primary Drive DMA Acknowledge:** When configured as DDACK0#, this pin functions as the primary drive DMA acknowledge signal. This signal should be buffered and this buffered output will be valid only when IDEEN# is asserted by the 82C568.
SA6	95	I/O-TTL (8mA)	Cycle Multiplexed	**System Address Bus Line 6:** Refer to the signal description for SA8.
DIOW#		O (8mA)	IDEEN# will drive the 74F244 active to allow the signal to pass to the IDE interface.	**IDE I/O Write Strobe:** When configured as DIOW#, this pin functions as the write strobe signal for the primary and secondary IDE drives. The rising edge of DIOW# samples data from the IDE data bus (DA[15:0]) into a register or the data port of the drive. This signal should be buffered and this buffered output will be valid only when IDEEN# is asserted by the 82C568.

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
SA5	98	I/O-TTL (8mA)	Cycle Multiplexed	**System Address Bus Line 5:** Refer to the signal description for SA8.
DIOR#		O (8mA)	IDEEN# will drive the 74F244 active to allow the signal to pass to the IDE interface.	**IDE I/O Read Strobe:** When configured as DIOR#, this pin functions as the read strobe signal for the primary and secondary IDE drives. The low level of DIOR# enables data from a register or the data port of the drive on the data bus (DD[15:0]). This signal should be buffered with a 74F244. This buffered output will be valid only when IDEEN# is asserted by the 82C568.
SA4	99	I/O-TTL (8mA)	Cycle Multiplexed	**System Address Bus Line 4:** Refer to the signal description for SA8.
DCS03#		O (8mA)	IDEEN# will drive the 74F244 active to allow the signal to pass to the IDE interface.	**Primary Drive Chip Select 3:** When configured as DCS03#, this pin functions as the chip select signal for the primary IDE. After the 82C567 translates the CPU cycle to a PCI cycle, the 82C568 decodes the address on the AD lines and asserts this signal to select the command block register for the primary IDE. This signal should be buffered and this buffered output will be valid only when IDEEN# is asserted by the 82C568.
SA3	100	I/O-TTL (8mA)	Cycle Multiplexed	**System Address Bus Line 3:** Refer to the signal description for SA8.
DCS01#		O (8mA)	IDEEN# will drive the 74F244 active to allow the signal to pass to the IDE interface.	**Primary Drive Chip Select 1:** When configured as DCS01#, this pin functions as the chip select signal for the primary IDE. After the 82C567 translates the CPU cycle to a PCI cycle, the 82C568 decodes the address on the AD lines and asserts this signal to select the command block register for the primary IDE. This signal should be buffered and this buffered output will be valid only when IDEEN# is asserted by the 82C568.
SA[2:0]	101:103	I/O-TTL (8mA)	Cycle Multiplexed	**System Address Bus Lines 2 through 0:** Refer to the signal description for SA8.
DA[2:0]		I/O-TTL (8mA)		**IDE Drive Address Lines 2 through 0:** This 3-bit binary coded address is asserted by the 82C568 to access a register or a data port in the primary/secondary IDE drive. It is recommended that these signals be buffered and driven to the IDE drives.
IDEEN#	84	O (8mA)		**IDE Enable:** This signal is asserted by the 82C568 for all accesses to and from the IDE drives. Most of the IDE control signals need to be buffered via a 74F244 and driver to the IDE drives. IDEEN# should be connected to the output enable of the 244's. This will prevent the IDE drives from responding to any other cycle. A 4.7K pull-up resistor must be provided on this signal.

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
SD[7:0]	64:71	I/O-TTL (8mA)	Cycle Multiplexed	**System Data Bus Lines 7 through 0:** SD[7:0] along with SD[15:8] provide the 16-bit data path for devices residing on the ISA bus. These lines should be pulled up externally.
PPWR[7:0]		I/O-TTL (8mA)		**Peripheral Power Control Lines 7 through 0:** The peripheral power control signals are latched externally with the PPWRL# signal. These lines should be pulled up externally.
DD[7:0]		I/O-TTL (8mA)		**Disk Data Lines 7 through 0:** DD[7:0] along with DD[15:8] provide the 16-bit data path for the IDE disk drives.
SD[15:8]	54:59, 62, 63	I/O-TTL (8mA)	Cycle Multiplexed	**System Data Bus Lines 15 through 8:** SD[15:8] are used along with SD[7:0] to provide the 16-bit data path for devices residing on the ISA bus.
DD[15:8]		I/O-TTL (8mA)		**Disk Data Lines 15 through 8:** DD[15:8] along with DD[7:0] provide the 16-bit data path for the IDE disk drives.
DDRQ1	155	I-TTL		**Secondary IDE Drive Data Request:** The secondary master IDE drive asserts this signal to the 82C568 when it needs control of the bus. It is recommended that this line be pulled low externally.
DDRQ0	156	I-TTL		**Primary IDE Drive Data Request:** The primary master IDE drive asserts this signal to the 82C568 when it needs control of the bus. It is recommended that this line be pulled low externally.
IOCS16#	75	I-TTL		**16-Bit I/O Chip Select:** This signal is driven by I/O devices on the ISA bus to indicate that they support 16-bit I/O bus cycles.
MEMCS16#	76	I/O-TTL (8mA)		**16-Bit Memory Chip Select:** ISA slaves that are 16-bit memory devices drive this signal low. MEMCS16# is an input when the 82C568 owns the ISA bus.
MEMR#	78	I/O-TTL (8mA)		**Memory Read:** MEMR# is the command to a memory slave that it may drive data onto the ISA data bus. MEMR# is an output when the 82C568 is a master on the ISA bus. MEMR# is an input when an ISA master, other than 82C568, owns the ISA bus.
MEMW#	79	I/O-TTL (8mA)		**Memory Write:** MEMW# is the command to a memory slave that it may latch data from the ISA data bus. MEMW# is an output when the 82C568 owns the ISA bus. MEMW# is an input when an ISA master, other than the 82C568, owns the ISA bus.
AEN	86	I/O-TTL (8mA)		**Address Enable:** AEN is asserted during DMA cycles to prevent I/O slaves from misinterpreting DMA cycles as valid I/O cycles. When asserted, AEN indicates to an I/O resource on the ISA bus that a DMA transfer is occurring. This signal is also asserted during refresh cycles.
IOCHRDY	87	I/O-TTL (8mA)		**I/O Channel Ready:** Resources on the ISA bus negate IOCHRDY to indicate that wait states are required to complete the cycle. If the local bus IDE is being used, the DCHRDY signal from the IDE drives could be connected to this signal directly or through a buffer.

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
IOCHK#	88	I-TTL		**I/O Channel Check:** When asserted, this signal indicates that a parity or an uncorrectable error has occurred for a device or memory on the ISA bus.
BALE	85	O (8mA)		**Bus Address Latch Enable:** BALE is an active high signal asserted by the 82C568 to indicate that the address, AEN, and SBHE# signal lines are valid. BALE remains asserted throughout ISA master and DMA cycles.
IOR#	82	I/O-S (8mA)		**I/O Read:** IOR# is the command to an ISA I/O slave device that the slave may drive data on to the ISA data bus (SD[15:0]). The I/O slave device must hold the data valid until after IOR# is negated. IOR# is an output when the 82C568 owns the ISA bus. IOR# is an input when an external ISA master owns the ISA bus.
IOW#	83	I/O-S (8mA)		**I/O Write:** IOW# is the command to an ISA I/O slave device that the slave may latch data from the ISA data bus (SD[15:0]). IOW# is an output when the 82C568 owns the ISA bus. IOW# is an input when an external ISA master owns the ISA bus.
SMEMR#	80	O (8mA)		**System Memory Read:** The 82C568 asserts SMEMR# to request a memory slave to provide data. If the access is below the 1MB range (00000000h-000FFFFFh) during DMA compatible, 82C568 master, or ISA master cycles, the 82C568 asserts SMEMR#.
SMEMW#	81	O (8mA)		**System Memory Write:** The 82C568 asserts SMEMW# to request a memory slave to accept data from the data lines. If the access is below the 1MB range (00000000h-000FFFFFh) during DMA compatible, 82C568 master, or ISA master cycles, the 82C568 asserts SMEMW#.
ZEROWS#	90	I/O-TTL (PCI)	PCIDV1 51h[6] = 0	**Zero Wait States:** An ISA slave asserts ZEROWS# after its address and command signals have been decoded to indicate that the current cycle does not require any wait states.
PGNT3#		O (PCI)	PCIDV1 51h[6] = 1	**PCI Grant 3**: When this pin functions as PGNT3# and it is asserted, it indicates that the device on PCI Slot 4 has been granted use of the PCI bus.
XDIR	73	I/O-TTL (4mA)		**X Bus Direction:** This signal is connected directly to the direction control of a 74F245 that buffers the utility data bus.
PCNTRL		I/O-TTL (4mA)		**Power Control:** During power-on reset, this pin is a strap option to decide on what kind of power management scheme is required. If sampled high on reset, the PPWRL# pin functions as a power latch control strobe and GPCS0#+PPWRL1 takes on its programmed functionality. If sampled low at reset, and PPWRL# functions as PPWRL2 and GPCS0#+PPWRL1 as PPWRL1. In this case, the external power control latch cannot be used.

82C568 Signal Descriptions (cont.)

3.4.3 82C567 and 82C566 Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
MMD[31:0]	16:25, 27:36, 39:42, 44:51	I/O-TTL (4mA)		**IPC to DBC Data Path:** These pins are connected directly to the 82C566. This private bus serves as a conduit for CPU reads/writes to/ from PCI/ISA. These lines have internal pull-up resistors.
MDLE#	152	I-TTL		**Memory Data Latch Enable:** This input is connected to the MDLE# pin of the 82C567 to control the data flow from the PCI AD[31:0] bus to the high 32-bit memory data bus, MMD[31:0], and vice versa. It is used to latch the data during CPU writes to PCI and PCI writes to DRAM and L2 cache.
HREQ	153	O (4mA)		**Hold Request:** This output signal is connected to the HREQ pin of the 82C567 and indicates that there is a master or DMA cycle request from the 82C568.
TMOD#		I-TTL	Strap Option at RESET	**Test Mode Operation:** During power-up, this pin is used to enter test mode operation if this is low.
LMEM#	11	I-TTL		**Local Memory Accessed Indication:** This signal serves as a local device memory accessed indication during local bus master cycles.
MSGN2S	52	I	PCIDV1 5Fh[7] = 1	**82C567 to 82C568 Message:** This pin is used to communicate between the 82C567 and 82C568. This pin should be connected directly to the MSGN2S pin of the 82C567.
MSGS2N	154	O (4mA)	PCIDV1 5Fh[6:5] = 10	**82C568 to 82C567 Message:** This pin is used to communicate between the 82C568 and 82C567. This pin should be connected directly to the MSGS2N pin of the 82C567.
USBGNT#		O (4mA)	PCIDV1 5Fh[6:5] = 11	**Universal Serial Bus Grant:** This pin is used to inform the USB device in the 82C567 that the PCI bus is granted to it. This pin should be connected directly to the USBGNT# pin of the 82C567.

3.4.4 PCI Bus Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
C/BE[3:0]#	162, 163, 166, 167	I/O-TTL (PCI)		**PCI Bus Command and Byte Enables:** During the address phase of a transaction, C/BE[3:0]# defines the PCI command. During the data phase, C/BE[3:0]# are used as the PCI byte enables. The PCI commands indicate the current cycle type, and the PCI byte enables indicate which byte lines carry meaningful data. The 82C568 drives C/ BE[3:0]# as an initiator of a PCI bus cycle and monitors C/BE[3:0]# as a target.
PAR	158	I/O-TTL (PCI)		**Calculated Parity Signal:** PAR is "even" parity and is calculated on 36 bits, AD[31:0] plus C/BE[3:0]#. PAR is generated for address and data phases and is only guaranteed to be valid on the PCI clock after the corresponding address or data phase.

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
FRAME#	168	I/O-TTL (PCI)		**Cycle Frame:** FRAME# is driven by the current bus master to indicate the beginning of a PCI cycle and is maintained asserted for the entire duration of an access. FRAME# is asserted to indicate that a bus transaction is beginning. FRAME# is an input to the 82C568 when the 82C568 is the target. FRAME# is an output when the 82C568 is the initiator.
IRDY#	169	I/O-TTL (PCI)		**Initiator Ready:** IRDY# indicates the ability of the 82C568, as an initiator, to complete the current data phase of the transaction. It is used in conjunction with TRDY#. A data phase is completed on each clock that both IRDY# and TRDY# are sampled asserted. IRDY# is an input to the 82C568 when the 82C568 is the target and an output when the 82C568 is an initiator.
TRDY#	170	I/O-TTL (PCI)		**Target Ready:** TRDY# indicates the ability of the 82C568 to complete the current data phase of the transaction. It is used in conjunction with IRDY#. A data phase is completed on each clock that TRDY# and IRDY# are both sampled asserted. TRDY# is an input to the 82C568 when the 82C568 is the initiator and an output when the 82C568 is a target.
DEVSEL#	171	I/O-TTL (PCI)		**Device Select:** The 82C568 asserts DEVSEL# to claim a PCI transaction. As an output, the 82C568 asserts DEVSEL# when it samples configuration cycles to the 82C568 configuration registers. As an input, DEVSEL# indicates the response to a transactions. If no slave claims the cycle within four PCICLKs after the assertion of FRAME#, the 82C568 asserts DEVSEL# to claim the cycle and initiates an ISA cycle.
STOP#	161	I/O-TTL (PCI)		**Stop:** STOP# indicates that the 82C568, as a target, is requesting a master to stop the current transaction. As a master, STOP# causes the 82C568 to stop the current transaction. STOP# is an output when the 82C568 is a target and an input when the 82C568 is an initiator
PLOCK#	157	I-TTL		**PCI Lock:** PLOCK# is used to indicate an atomic operation that may require multiple transactions to complete. When PLOCK# is asserted, non-exclusive transactions may proceed to an address that is not currently locked.
PERR#	159	I/O-TTL (PCI)		**Parity Error:** PERR# may be pulsed by any agent that detects a parity error during an address phase, or by the master or the selected target during any data phase in which the AD[31:0] lines are inputs.
SERR#	160	I/O-TTL (PCI)		**System Error:** SERR# can be pulsed active by any PCI device that detects a system error condition.
AD[31:0]	172:179, 181:184, 187:199, 202:208	I/O-TTL (PCI)		**PCI Address and Data:** AD[31:0] are bidirectional address and data lines for the PCI bus. The AD[31:0] signals sample or drive the address and data on the PCI bus. This bus also serves as a conduit for transferring address information to the 82C567 during ISA master cycles. It conveys the SA[8:0] information to the 82C567 on these lines.

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
PIRQ0#	139	I-TTL	PCIDV1 45h[1:0] = 00	**PCI Interrupt Request 0:** An active low assertion indicates that the respective interrupt is active.
EPMI1#		I-TTL	PCIDV1 45h[1:0] = 01	**External Power Management Input 1:** If configured to be an EPMI1# input, this signal should have an external pull-up.
PIRQ1#	140	I-TTL	PCIDV1 45h[3:2] = 00	**PCI Interrupt Request 1:** An active low assertion indicates that the respective interrupt is active. If IRQ6 has been programmed to take on the MPIRQ0#/1# functionality, this pin should be pulled up.
IRQ0		I-TTL	PCIDV1 45h[3:2] = 01	**Interrupt Request 0:** If IRQ6 has been programmed to take on the MPIRQ0#/1# functionality, this pin should be pulled up.
PIRQ2#	141	I-TTL	PCIDV1 51h[3] = 0	**PCI Interrupt 2:** An active low assertion indicates that the respective interrupt is active.
GPCS0#		O (4mA)	PCIDV1 51h[3] = 1	**General Purpose Chip Select 0**
PIRQ3#	143	I-TTL	PCIDV1 51h[4] = 0 and 45h[5] = 0	**PCI Interrupt 3:** An active low assertion indicates that the respective interrupt is active.
EPMI2#		I-TTL	PCIDV1 51h[4] = 0 and 45h[5] = 1	**External Power Management Input 2:** If configured to be an EPMI2# input, this signal should have an external pull-up.
GPCS1#		O (4mA)	PCIDV1 51h[4] = 1 and 45h[5] = X	**General Purpose Chip Select 1**
PREQ0#	144	I-TTL		**PCI Request 0:** An active low assertion indicates that the device on PCI Slot 1 desires the use of the PCI bus. This signal should be pulled up externally.
PREQ1#	145	I-TTL		**PCI Request 1:** An active low assertion indicates that the device on PCI Slot 2 desires the use of the PCI bus. This signal should be pulled up externally.
PREQ2#	146	I-TTL	PCIDV1 53h[6] = 0	**PCI Request 2:** An active low assertion indicates that the device on PCI Slot 3 desires the use of the PCI bus. This signal should be pulled up externally.
EPMI0#		I-TTL	PCIDV1 53h[6] = 1	**External Power Management Input 0**
PGNT[2:0]#	148:150	O (PCI)		**PCI Grants 2 through 0:** An active low assertion indicates that one of the initiators on PCI Slot 1, 2, or 3 has been granted use of the PCI bus.

82C568 Signal Descriptions (cont.)

3.4.5 CPU Interface Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
SMI#	4	O (4mA)		**System Management Interrupt:** This signal is used to request System Management Mode (SMM) operation from the CPU.
STPCLK#	13	O (4mA)		**Stop Clock:** This signal is connected to the STPCLK# input of the CPU. It causes the CPU to get into the STPGNT# state. If an M1 processor is being used, this signal can be connected to the SUSP# signal.
AHOLD	5	I-TTL		**Address Hold:** This input is connected to the AHOLD pin of the 82C567 and is used to monitor bus arbitration.
INTR	6	I/O-TTL (4mA)		**Interrupt Request:** INTR is driven by the 82C568 to signal the CPU that an interrupt request is pending and needs to be serviced. The interrupt controller must be programmed following a reset to ensure that INTR is at a known state.
NMI	9	O (4mA)		**Non-Maskable Interrupt:** This signal is activated when a parity error from a local memory read is detected or when the IOCHK# signal from the ISA bus is asserted and the corresponding control bit in Port B is also enabled. The 82C568 also generates an NMI when either PERR# or SERR# is asserted.
A20M#	12	O (4mA)	PCIDV1 41h[4] = 0	**Address 20 Mask:** This signal is derived from the keyboard GATEA20 emulation and Port 92h bit 1. If keyboard emulation has been enabled, this pin takes on the A20M# functionality. It outputs A20M# whenever it decodes a Port 92h fast GATEA20 or a keyboard GATEA20.
KBRST		I-TTL	PCIDV1 41h[4] = 1	**Keyboard Reset:** This input monitors the keyboard reset signal. If keyboard emulation has been disabled, this pin takes on the KBRST functionality. The 82C568 takes the KBRST from the keyboard and responds by generating an INIT to the CPU.

3.4.6 ISA DMA Arbiter Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
DREQ0	116	I-TTL	PCIDV1 48h[3:2] = 00	**DMA Request 0:** The DREQ is used to request DMA service from the DMA controller of the 82C568.
DREQ0/5		I-TTL	PCIDV1 48h[3:2] = 01	**Multiplexed DMA Request 0/5:** If configured as a multiplexed input, an external multiplexer is required.
DREQ5		I-TTL	PCIDV1 48h[3:2] = 10	**DMA Request 5**

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
DREQ1	117	I-TTL	PCIDV1 48h[5:4] = 00	**DMA Request 1:** The DREQ is used to request DMA service from the DMA controller of the 82C568.
DREQ1/6		I-TTL	PCIDV1 48h[5:4] = 01	**Multiplexed DMA Request 1/6:** If configured as a multiplexed input, an external multiplexer is required.
DREQ6		I-TTL	PCIDV1 48h[5:4] = 10	**DMA Request 6**
DREQ2	118	I-TTL		**DMA Request 2:** The DREQ is used to request DMA service from the DMA controller of the 82C568.
DREQ3	120	I-TTL	PCIDV1 48h[7:6] = 00	**DMA Request 3:** The DREQ is used to request DMA service from the DMA controller of the 82C568.
DREQ3/7		I-TTL	PCIDV1 48h[7:6] = 01	**Multiplexed DMA Request 3/7:** If configured as a multiplexed input, an external multiplexer is required.
DREQ7		I-TTL	PCIDV1 48h[7:6] = 10	**DMA Request 7**
DREQ5	121	I-TTL		**DMA Request 5:** The DREQ is used to request DMA service from the DMA controller of the 82C568.
DREQ6	122	I-TTL	PCIDV1 42h[0] = 0	**DMA Request 6:** The DREQ is used to request DMA service from the DMA controller of the 82C568.
EPMI0#		I-TTL	PCIDV1 42h[0] = 1	**External Power Management Input 0:** If configured to be an EPMI0# input, this signal should have an external pull-up.
DREQ7	123	I-TTL	PCIDV1 49h[1:0] = 00	**DMA Request 7:** The DREQ is used to request DMA service from the DMA controller of the 82C568.
EPMI3#		I-TTL	PCIDV1 49h[1:0] = 01	**External Power Management Input 3:** If configured to be an EPMI3# input, this signal should have an external pull-up.
DACK0#	108	O (6mA)	PCIDV1 44h[1:0] = 01 and 44h[3:2] = 0X	**DMA Acknowledge 0**
EDACK0		O (6mA)	PCIDV1 44h[1:0] = 10 and 44h[3:2] = 0X	**Encoded DACK 0:** If configured as an encoded DMA acknowledge signal, an external decoder is required.
DACK5#		O (6mA)	PCIDV1 44h[3:2] = 10	**DMA Acknowledge 5**
GPCS0#		O (4mA)	PCIDV1 44h[3:2] = 11, 44h[1:0] = 01, and SYSCFG FDh[1] = 0	**General Purpose Chip Select 0**
PPWRL1		O (4mA)	PCIDV1 44h[3:2] = 11, 44h[1:0] = 01, and SYSCFG FDh[1] = 1	**Peripheral Power Latch 1:** This signal can be used to put the clock synthesizer into the Doze mode.

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
DACK1#	109	O (6mA)	PCIDV1 44h[1:0] = 01 and 44h[5:4] = 0X	**DMA Acknowledge 1**
EDACK1		O (6mA)	PCIDV1 44h[1:0] = 10 and 44h[5:4] = 0X	**Encoded DACK 1:** If configured as an encoded DMA acknowledge signal, an external decoder is required.
DACK6#		O (6mA)	PCIDV1 44h[5:4] = 10	**DMA Acknowledge 6**
DACK2#	110	O (6mA)	PCIDV1 44h[1:0] = 0X and 51h[5] = 0	**DMA Acknowledge 2**
EDACKEN#		O (6mA)	PCIDV1 44h[1:0] = 1X and 51h[5] = 0	**Encoded DACK Enable:** If configured as an encoded DMA acknowledge signal, an external decoder is required.
GPCS2#		O (6mA)	PCIDV1 51h[5] = 1	**General Purpose Chip Select 2**
DACK3#	111	O (6mA)	PCIDV1 44h[1:0] = 01 and 44h[7:6] = 0X	**DMA Acknowledge 3**
EDACK2		O (6mA)	PCIDV1 44h[1:0] = 10 and 44h[7:6] = 0X	**Encoded DACK2:** If configured as an encoded DMA acknowledge signal, an external decoder is required.
DACK7#		O (6mA)	PCIDV1 44h[7:6] = 10	**DMA Acknowledge 7**
DACK5#	112	O (6mA)	PCIDV1 4Fh[7] = 0	**DMA Acknowledge 5**
PPWRL#		O (4mA)	PCIDV1 4Fh[7] = 1 and at reset, XDIR (pin 73) = 1	**Peripheral Power Latch Control Signal:** This signal is used to control the external latching of the peripheral power control signals PPWR[7:0]. This signal is pulsed after reset to preset the external latch.
PPWRL2		O (4mA)	PCIDV1 4Fh[7] = 1 and at reset, XDIR (pin 73) = 0	**Peripheral Power Latch 2:** This signal can be used to put the clock synthesizer into the power-down mode.

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
DACK6#	113	O (6mA)	PCIDV1 4Fh[1] = 0 and 44h[1:0] = 01	**DMA Acknowledge 6**
GPCS2#		O (6mA)	PCIDV1 4Fh[1] = 1	**General Purpose Chip Select 2**
DACK7#	115	O (6mA)		**DMA Acknowledge 7**
EOP	89	I/O-TTL (8mA)		**End of Process:** EOP is bidirectional, acting in one of two modes, and is directly connected to the TC line of the ISA bus. DMA slaves assert EOP to the 82C568 to terminate DMA cycles. The 82C568 asserts EOP to DMA slaves as a terminal count indicator.
REFRESH#	91	I/O-TTL (8mA)		**Refresh:** As an output, this signal is used to inform the 82C567 to refresh the local DRAM. When another bus master has control of the bus, this pin is an input to the 82C568. This signal is generated once in 15µs by the 82C568 or an ISA master.
32KHZ	104	I-S	PCIDV1 51h[7] = 0 and 5Eh[6] = 0	**32KHz Clock:** This signal is the 32KHz clock input.
PREQ3#		I-S	PCIDV1 51h[7] = 1 and 5Eh[6] = 0	**PCI Request 3:** An active low assertion indicates that the device on PCI Slot 4 desires the use of the PCI bus. This signal should be pulled up externally.
SDACK2#		I	PCIDV1 51h[7] = X and 5Eh[6] = 1	**Steerable DACK for Channel 2**

82C568 Signal Descriptions (cont.)

3.4.7 Interrupt Control Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
IRQ1	124	I-TTL		**Interrupt Request 1:** This IRQ1 signal provides the keyboard controller with a mechanism for asynchronously interrupting the CPU.
IRQ3	125	I-TTL	PCIDV1 53h[7] = 0	**Interrupt Request 3:** The IRQ3 signal provides serial port 2 with a mechanism for asynchronously interrupting the CPU.
MIRQ3/5		I-TTL	PCIDV1 53h[7] = 1	**Multiplexed IRQ3/5**
IRQ4	126	I-TTL	PCIDV1 49h[2] = 0	**Interrupt Request 4:** The IRQ4 signal provides serial port 1 with a mechanism for asynchronously interrupting the CPU. If configured as a multiplexed input, an external multiplexer is required.
MIRQ4/6		I-TTL	PCIDV1 49h[2] = 1	**Multiplexed IRQ4/6**: These interrupts are used as mechanisms for asynchronously interrupting the CPU. IRQ4 relates to serial port 1 and IRQ6 relates to the floppy disk controller.
IRQ5	127	I-TTL	PCIDV1 53h[7] = 0	**Interrupt Request 5:** The IRQ5 signal provides parallel port 2 with a mechanism for asynchronously interrupting the CPU.
MIRQ7/9		I-TTL	PCIDV1 53h[7] = 1	**Multiplexed IRQ7/9**
IRQ6	128	I-TTL	PCIDV1 49h[3] = 0 and 53h[7] = 0	**Interrupt Request 6:** The IRQ6 signal provides the floppy disk controller with a mechanism for asynchronously interrupting the CPU.
MPIRQ0#/1#		I-TTL	PCIDV1 49h[3] = 1	**Multiplexed PCI Interrupt 0/1:** If configured as a multiplexed input, an external multiplexer is required.
MIRQ11/15		I-TTL	PCIDV1 49h[3] = 0 and 53h[7] = 1	**Multiplexed IRQ11/15:** If configured as a multiplexed input, an external multiplexer is required.
IRQ8#	130	I-TTL		**Interrupt Request 8:** This IRQ8 signal provides the real-time clock with a mechanism for asynchronously interrupting the CPU.
IRQ9	131	I-TTL	PCIDV1 53h[7] = 0	**Interrupt Request 9:** This pin is used to provide interrupt request 9 to the CPU.
EPMI2#		I-TTL	PCIDV1 53h[7] = 1	**External PMI 2:** If configured as EPMI0#, this signal should be pulled up externally.
IRQ7	129	I-TTL	PCIDV1 53h[7] = 0	**Interrupt Request 7:** This IRQ7 signal provides parallel port 1 with a mechanism for asynchronously interrupting the CPU.
EPMI1#		I-TTL	PCIDV1 53h[7] = 1	**External PMI 1:** If configured as EPMI1#, this signal should be pulled up externally.
IRQ10	132	I-TTL	PCIDV1 49h[4] = 0	**Interrupt Request 10**
MIRQ10/12		I-TTL	PCIDV1 49h[4] = 1	**Multiplexed IRQ10/12:** If configured as a multiplexed input, an external multiplexer is required.

82C568 Signal Descriptions (cont.)

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
IRQ11	133	I-TTL	PCIDV1 53h[7] = 0	**Interrupt Request 11:** This pin used to provide interrupt request 11 to the CPU.
GMIRQ		I-TTL	PCIDV1 53h[7] = 1	**General Purpose Motherboard Interrupt Request:** Pin 133 can also function as a general purpose IRQ line on the motherboard. Through register settings, this line can be mapped onto any of the IRQ lines.
IRQ12	134	I-TTL	PCIDV1 49h[6:5] = 0X	**Interrupt Request 12:** This pin can be programmed to be interrupt request 12 for a mouse device.
MPIRQ2#/3#		I-TTL	PCIDV1 49h[6:5] = 10	**Multiplexed PCI Interrupt 2/3:** If configured as a multiplexed input, an external multiplexer is required.
IRQ14	135	I-TTL	Cycle Multiplexed	**Interrupt Request 14:** This IRQ14 signal provides the expansion slot with a mechanism for asynchronously interrupting the CPU.
DINT0		I-TTL		**Disk Interrupt 0:** If the local bus IDE is being used, then the DINT0 output from the primary drive should be directly wired to this input. If the local bus IDE is enabled, this interrupt will not be available for use on the ISA bus.
IRQ15	136	I-TTL	Cycle Multiplexed	**Interrupt Request 15:** This IRQ15 signal provides the expansion slot with a mechanism for asynchronously interrupting the CPU.
DINT1		I-TTL		**Disk Interrupt 1:** If the local bus IDE is being used, then the DINT1 output from the secondary drive should be directly wired to this input. If the local bus IDE is enabled, this interrupt will not be available for use on the ISA bus.
FERR#	7	I-TTL		**Floating Point Coprocessor Error:** This input causes two operations to occur. IRQ13 is triggered and IGERR# is enabled. An I/O write to Port F0h will set IGERR# low when FERR# is low.
IGERR#	8	O (4mA)		**Ignore Coprocessor Error:** Normally high, IGERR# will go low after FERR# goes low and an I/O write to Port 0F0h occurs. When FERR# goes high, IGERR# is driven high.

82C568 Signal Descriptions (cont.)

3.4.8 RTC, Timer, and Steerable DRQ/DACK# Signals

Signal Name	Pin No.	Signal Type (Drive)	Selected By	Signal Description
RTCAS	105	O (4mA)	PCIDV1 5Eh[7] = 0 and 60h[4] = 0	**RTC Address Strobe:** This output is connected to the address strobe of the external real-time clock.
SDRQ1		I-TTL	PCIDV1 5Eh[7] = 1 and 60h[4] = 0	**Steerable DRQ for Channel 1**
PREQ4#		I	PCIDV1 5Eh[7] = X and 60h[4] = 1	**PCI Request 4:** An active low assertion indicates that the device on PCI Slot 5 desires the use of the PCI bus. This signal should be pulled up externally.
RTCRD#	106	O (4mA)	PCIDV1 59h[0] = 0 and 5Eh[6] = 0	**RTC Read:** This pin is used to drive the read signal to the external real-time clock.
PGNT3#		O (PCI)	PCIDV1 59h[0] = 1 and 5Eh[6] = 0	**PCI Grant 3**: When this pin functions as PGNT3# and it is asserted, it indicates that the device on PCI Slot 4 has been granted use of the PCI bus.
SDACK2#		O (6mA)	PCIDV1 59h[0] = X and 5Eh[6] = 1	**Steerable DACK for Channel 2**
RTCWR#	107	O (4mA)	PCIDV1 5Eh[7] = 0 and 60h[4] = 0	**RTC Write:** This pin is used to drive the write signal to the external real-time clock.
SDACK1#		O (6mA)	PCIDV1 5Eh[7] = 1 and 60h[4] = 0	**Steerable DACK for Channel 1**
PGNT4#		O (4mA)	PCIDV1 5Eh[7] = X and 60h[4] = 1	**PCI Grant 4:** When this pin functions as PGNT4# and it is asserted, it indicates that the device on PCI Slot 5 has been granted use of the PCI bus.
SPKR	92	O (16mA)		**Speaker Data:** This pin is used to drive the system board speaker. This signal is a function of the internal Timer-0 Counter-2 count and bit 1 of Port 61h.

82C568 Signal Descriptions (cont.)

3.4.9 Serial Interrupt Interface Signals

Signal Name	Pin No.	Signal Type	Selected By	Signal Description
SERIRQ#	1	I	PCIDV1 55h[4] = 1, 59h[3] = 0, and 5Fh[4] = 0	**Serial Interrupt Request:** The SERIRQ# signal is a wired OR signal that passes the state of one or more device's IRQ(s) to the host interrupt controller. This input is used to support ISA standard IRQs within PCI-based systems.
HLDA		I	PCIDV1 55h[4] = 0, 59h[3] = 0, and 5Fh[4] = 0	**CPU Hold Acknowledge:** This input is connected to the HLDA pin of the CPU. HLDA indicates, in response to a HOLD, when the CPU has relinquished bus control to another bus master.
EPMI0#		I	PCIVD1 55h[4] = 0, 59h[3] = 1, and 5Fh[4] = 0	**External Power Management Input 0**
SIRQ		I	PCIVD1 55h[4] = 0, 59h[3] = 0, and 5Fh[4] = 1	**Steerable Interrupt Request**

3.4.10 Power and Ground Pins

Signal Name	Pin No.	Signal Type	Selected By	Signal Description
GND	14, 15, 37, 38, 60, 72, 96, 119, 137, 142, 164, 185, 186, 200	I-G		**Ground Connection**
VCC3	10	I-P		**Power Connection:** 3.3V power plane
VCC	26, 43, 61, 97, 114, 147, 165, 180, 201	I-P		**Power Connection:** 5.0V power plane

4.0 Functional Description

4.1 Reset Logic

The PWRGD input to the 82C567 is used to generate the CPU and the system reset (CPURST). PWRGD is a "cold reset" which is generated when either PWRGD goes low (from the power supply, indicating a low power condition) or the system reset button is activated. When PWRGD goes low, CPURST will go active and will remain active for at least 1ms after PWRGD goes high.

The INIT signal is used to initialize the 3.3V CPU during warm resets. INIT is generated for the following cases:

- When a shutdown condition is decoded from the CPU bus definition signals, the 82C568 will assert INIT for 15 T-states.
- Keyboard reset to I/O Port 064h.
- Fast reset to I/O Port 092h.

4.2 System Clocks

4.2.1 CPU and 82C567 Clocks

The 82C567 uses one high frequency clock input (CLK) and one PCI clock input (LCLK). The clock signals that go to the CPU and the CLK inputs to the 82C567 are required to be in the same phase and have minimum skew between them. The skew between the CLK input to the 82C567 and the CLK input to the CPU should not exceed 2ns. The 82C567 CLK is a single phase clock which is used to sample all host CPU synchronous signals and for clocking the internal state machines of the 82C567.

Table 4-1 shows the register bit that is used for setting the synchronization between the PCI bus clock (LCLK) and the CPU clock (CLK).

Figure 4-1 shows a typical CPU and 82C567 clock distribution circuit and the relationship between the CPUCLK and CLK signals.

4.2.2 PCI Bus Clocks

The 82C567 and 82C568 require LCLK for the PCI interface. The phase and frequency of the LCLK input to the 82C567 and 82C568, and the LCLK input to the PCI bus is required to be the same and the maximum skew should not exceed 2ns. Figure 4-2 and Figure 4-3 show possible clock generation and distribution schemes for LCLK.

Table 4-1 Synchronization Control between LCLK and CLK

7	6	5	4	3	2	1	0
SYSCFG 16h			**Dirty/Tag RAM Control Register**				**Default = 00h**
DIRTYI pin selection:(1) 0 = Input only 1 = I/O	Reserved: Must be written to 0.	Tag RAM size selection:(2) 0 = 8-bit 1 = 7-bit	Single write hit leadoff cycle in a combined Dirty/Tag implementation:(3) 0 = 5 cycles 1 = 4 cycles	Pre-snoop control: 0 = Pre-snoop for starting address 0 only 1 = Pre-snoop for all addresses except those on the line boundary	Synchronization between the PCI bus clock (LCLK) and the CPU clock (CLK):(4) 0 = LCLK async to CLK 1 = LCLK sync to CLK (skew not to exceed -2ns to 15ns)	Reserved: Must be written to 0.	HDOE# timing control: 0 = Negated normally 1 = Negated one clock before the cycle finishes

(1) If using a x1 SRAM for the Dirty RAM in which there is a separate DirtyIn and a separate DirtyOut bit, then the DIRTYI pin becomes an input only. If using a standard x8 or x9 SRAM, where there is no separate pin for input and output, then the DIRTYI pin becomes an I/O pin.

(2) If a 7-bit Tag is being used and a combined Tag/Dirty RAM is being used, then TAG0 functions as the DIRTYIO signal. In this case, the DIRTYI pin is unused.

(3) If bit 4 is set 1, SYSCFG 22h[0] should be set to 1.

(4) It should be noted that LCLK could be async to CLK also. This bit therefore implies that the PCI clock is either sync to the CPU clock with a skew not to exceed -2ns to 15ns, or that the PCI clock is async to the CPU clock.

Figure 4-1 CPU and 82C567 Clock Distribution

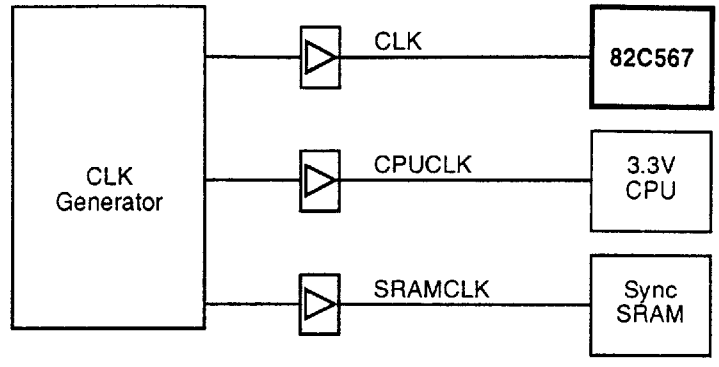


Figure 4-2 Clock Distribution Method for PCI Bus Connectors - Async PCI Bus

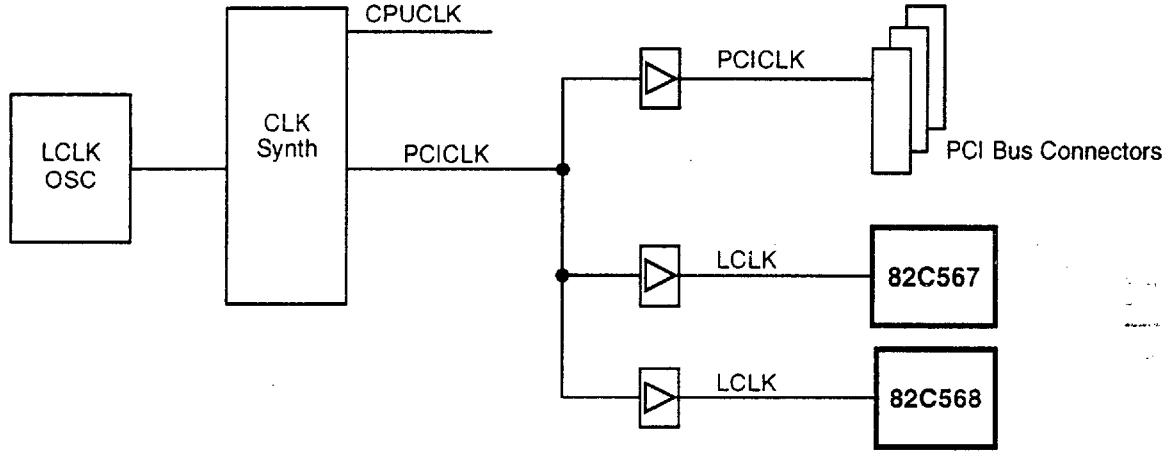
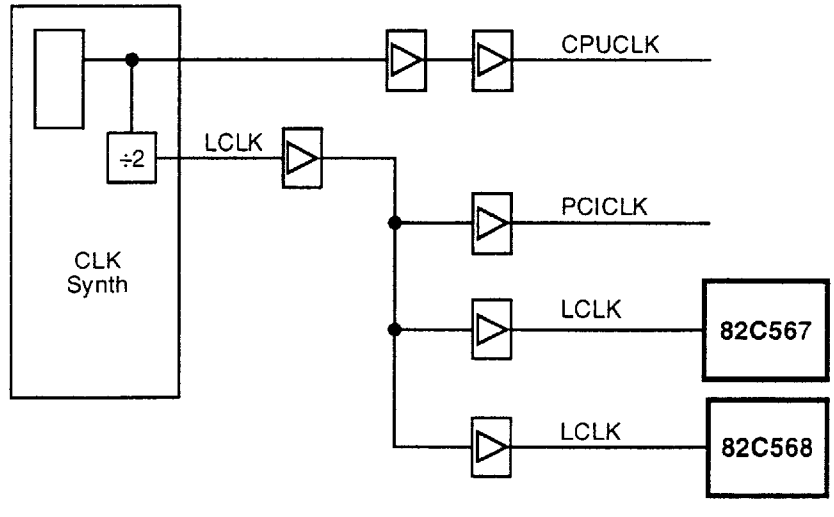


Figure 4-3 Clock Distribution Method for PCI Bus Connectors - Sync PCI Bus



4.2.3 ISA Bus Clock

The 82C568 generates the ISA bus clock (ATCLK) from an internal division of LCLK. The ATCLK frequency is programmable (via PCIDV1 47h[5:4], see Table 4-2) and can be set to any of four clock division options: LCLK÷1, LCLK÷2, LCLK÷3, LCLK÷4. This allows the system designer to tailor the ISA bus clock frequency to support a wide range of system designs and performance platforms.

Table 4-2 ATCLK Frequency Control

7	6	5	4	3	2	1	0
PCIDV1 47h			**Cycle Control Register 1 - Byte 1**				**Default = 00h**
Write protect ISA bus ROM (ROMCS# for writes): 0 = Enable 1 = Disable	Refresh select: 0 = Normal 1 = Hidden	ATCLK frequency select: 00 = LCLK÷4 10 = LCLK÷2 01 = LCLK÷3 11 = LCLK		CPU master to PCI slave write (turnaround between address and data phases): 0 = 1 LCLK 1 = 0 LCLK	PCI master to PCI master preemption timer (preempt after unserviced request pending for X LCLKs): 000 = No preemption 100 = 36 LCLKs 001 = 260 LCLKs 101 = 20 LCLKs 010 = 132 LCLKs 110 = 12 LCLKs 011 = 68 LCLKs 111 = 5 LCLKs		

4.3 Cache Subsystem

The integrated cache controller, which uses a direct-mapped, bank-interleaved scheme (for an asynchronous SRAM implementation) dramatically boosts the overall performance of the local memory subsystem by caching writes as well as reads (write-back mode). Cache memory can be configured as one or two banks of asynchronous SRAMs and sizes of 256KB, 512KB, 1MB, and 2MB are supported. In addition, the cache controller also supports 256KB, 512KB, 1MB, and 2MB of synchronous SRAM in a single/double bank configuration. Two programmable non-cacheable regions are provided. The cache controller operates in a non-pipelined or a pipelined mode, with a fixed 32-byte line size (optimized to match a CPU burst linefill) in order to simplify the motherboard design without increasing cost or degrading system performance. The secondary cache operates independently and in addition to the CPU's internal cache.

The cache controller of the 82C567 has a built-in tag comparator which improves system performance while reducing component count on the system board. The controller features a 64-bit wide data bus with 32-byte CPU burst support. The cache controller supports write-back, adaptive write-back, and write-through schemes.

The cache controller uses a 32-byte secondary cache line size. It supports read and write bursting in 3-2-2-2 bursts for the asynchronous SRAM and 3-1-1-1 burst read/write for synchronous SRAMs. 2-1-1-1 burst read/write cycles are supported for synchronous SRAMs at 50MHz. In this case, the ADSC# output of the processor needs to be connected to the ADSC# input of the synchronous SRAM. The 8-bit tag has a "dirty" bit option for the write-back cache. The cache controller uses standard single bank SRAMs or dual bank SRAMs with interleaving (only in the case for asynchronous SRAM) for optimum cache performance.

4.3.1 CPU Burst Mode Control

The Viper-MAX Chipset fully supports the 64-bit wide data path for the CPU burst read and burst write cycles. The cache and DRAM controllers in the 82C567 ensure that data is burst into the CPU whenever the CPU requests a burst linefill or a burst write to the system memory.

The 82C567 contains separate burst counters to support DRAM and external cache burst cycles. The DRAM controller performs a burst for the L2 cache read miss linefill cycle (DRAM to L2 cache and CPU) and the cache controller burst supports the CPU burst linefill (3.3V Pentium and K5 burst linefill and the Cyrix M1 linear burst linefill) for the L2 cache hit cycle (L2 cache to the 3.3V Pentium CPU). Depending on the kind of processor being used, either the 3.3V Pentium quad word burst address sequencing or the Cyrix M1 quad word linear burst address sequencing is used for all system memory burst cycles.

4.3.1.1 Cyrix Linear Burst Mode Support

The Viper-MAX Chipset supports the Cyrix linear burst mode. SYSCFG 17h[0] in the 82C567 determines which burst mode is to be implemented, the Intel 3.3V Pentium CPU burst mode or the Cyrix linear burst mode. No additional hardware is required for supporting either of these modes.

When using a synchronous SRAM solution, care must taken that the synchronous SRAM burst protocol complements the processor's burst protocol.

Table 4-3 shows the burst mode sequence for both of these processors and Table 4-4 highlights the register bits that need to be programmed upon system burst mode selection.

Table 4-3 Burst Modes

1st Address	2nd Address	3rd Address	4th Address
Cyrix Linear Burst Mode			
0	8	10	18
8	10	18	0
10	18	0	8
18	0	8	10
Intel Burst Mode			
0	8	10	18
8	0	18	10
10	18	0	8
18	10	8	0

Table 4-4 Burst Mode Control Register Bit

7	6	5	4	3	2	1	0
SYSCFG 17h			**PCI Cycle Control Register 2**				**Default = 00h**
Reserved: Must be written to 0.	Generate NA# for PCI slave access in async LCLK mode: 0 = No 1 = Yes This bit will be overridden if bit 7 is set.	Sync two bank select: 0 = Reserved 1 = Set this bit to 1 when two banks of sync SRAM are installed	Reserved: Must be written to 0.	Reserved: Must be written to 0.	Pipelining during byte merge: 0 = Disable 1 = Enable	Sync SRAM type (if SYSCFG 11h[3] = 1): 0 = Standard 1 = Pipelined	Burst type: 0 = Intel burst protocol 1 = Cyrix linear burst protocol
SYSCFG 0Dh			**Clock Control Register**				**Default = 00h**
Reserved: Must be written to 0.			Enable A0000h-BFFFFh as system memory: 0 = No 1 = Yes	Add one more wait state during PCI master cycle with Intel-type address toggling(1): 0 = No 1 = Yes	Give the 82C567 control of the PCI bus on STOP# generation after HITM# is active: 0 = No 1 = Yes(2)	CPU clock is slowed down to below 33MHz: 0 = No 1 = Yes	

(1) If the PCI master does its address toggling in the style of the Intel 486 burst, rather than a linear burst mode style, then one wait state needs to be added.

(2) The 82C567 has control over the PCI bus until the write-back is completed. If PCI master pre-snoop has been enabled (SYSCFG 0Fh[7] = 1), 0Dh[1] should be set to 1.

4.3.2 Cache Cycle Types

Some cache terminology and cycle definitions that are chipset specific:

The cache hit/miss status is generated by comparing the high-order address bits (for the memory cycle in progress) with the stored tag bits from previous cache entries. When a match is detected and the location is cacheable, a cache hit cycle takes place. If the comparator does not detect a match or a non-cacheable location is accessed (based on the internal non-cacheable region registers), then the current cycle is a cache miss.

A cache hit/miss decision is always made at the end of the first T2 for a non-pipeline cycle and at the end of the first T2P for a pipeline cycle, so the SRAM read/write cycle will begin after the first T2 or T2P. The cacheable decision is based on the DRAM bank decodes and the chipset's configuration registers for non-system memory areas and non-cacheable area definitions. If the access falls outside the system memory area, it is always non-cacheable.

The dirty bit is a mechanism for monitoring coherency between the cache and system memory. Each tag entry has a corresponding dirty bit to indicate whether the data in the represented cache line has been modified since it was loaded from system memory. This allows the 82C567 to determine whether the data in the system memory is "stale" and needs to be updated before a new memory location is allowed to overwrite the currently indexed cache entry. The Viper-MAX supports several Tag/Dirty schemes and those are described in Section 4.3.4.7 on page 71.

A linefill cycle occurs for a cache read miss cycle. It is a data read of the new address location from the system memory and a corresponding write to the cache. The tag data will also be updated with the new address.

A castout cycle occurs for a cache read miss cycle, but only if the cache line that is being replaced is "dirty". In this cycle, the dirty cache line is read from the cache and written to the system memory. The upper address bits for this cycle are provided by the tag data bits.

A write-back cycle consists of performing a castout cycle followed by a linefill cycle. The write-back cycle causes an entire cache line (32 bytes) to be written back to memory followed by a line burst from the new memory location into the cache and to the CPU simultaneously. The advantages of performing fast write cycles to the cache (for a write hit) typically outweigh the cycle overhead incurred by the write-back scheme.

4.3.3 Single and Double Bank Support

The 82C567 supports one or two banks of SRAM. SYSCFG 08h[7] controls this feature (as shown in Table 4-5).

Table 4-5 SRAM Bank Support Register Bit

7	6	5	4	3	2	1	0
SYSCFG 08h			**CPU Cache Control Register**				**Default = 00h**
L2 cache single/double bank select: 0 = Double bank (If async SRAM, then the banks are interleaved. If sync SRAM, they are not interleaved.) 1 = Single bank (non-interleaved)	Snoop filtering for bus masters:(1) 0 = Disable 1 = Enable	CPU HITM# pin sample timing: 0 = Delay 1 CLK (HITM# sampled on 3rd rising edge of LCLK after EADS# assertion) 1 = No delay (HITM# sampled on 2nd rising edge of LCLK after EADS# assertion)	Parity checking: 0 = Disable 1 = Enable	Tag/Dirty RAM implementation: 0 = Tag and Dirty are on separate chip (i.e., a separate x1 or x8 SRAM for the Dirty RAM) 1 = Tag and Dirty are on the same chip (i.e., could be either a x9 or x8 Tag/Dirty RAM)	CPU address pipelining: 0 = Disable 1 = Enable	L1 cache write-back and write-through control: 0 = Write-through only 1 = Write-back enabled	BIOS area cacheability in L1 cache: Determines if system BIOS area E0000h-FFFFFh (if SYSCFG 04h[2] = 1) or F0000h-FFFFFh (if SYSCFG 04h[2] = 0), and video BIOS area C0000h-C7FFFh is cacheable in L1 or not. 0 = Cacheable 1 = Non-Cacheable

(1) For a master request if the subsequent read/write is within the same cache line, CPU 'Inquire' cycles are not done until there is a cache line miss (i.e., line comparator not activated for accesses within the same cache line).

4.3.4 Cache Operation

The following discussion pertains to asynchronous SRAMs, but is valid for the synchronous SRAM as well, except that the synchronous SRAM supports 3-1-1-1 cycles and 2-1-1-1 cycles at 50MHz instead of 3-2-2-2 cycles.

4.3.4.1 L2 Cache Read Hit

On an L1 read miss and an L2 read hit, the secondary cache provides data to the CPU. The 82C567 follows either the 3.3V Pentium CPU's burst protocol or the M1's linear burst mode protocol to fill the processor's internal cache line.

The cache controller will sample CACHE# from the CPU at the end of T1 and perform a burst read if CACHE# is sampled active. The first cache read hit for a cycle is always one wait state. If a read cycle can be converted to a burst, the read cycle is extended for the additional three words continuing at one wait state per cycle. To achieve the burst at this rate, the hit or miss decision must be made before BRDY# is returned to the CPU at the end of the second T2. The cache hit comparator in the 82C567 compares the data from the tag RAM with the higher address bits from the CPU bus. The output of this comparator generates the BRDY# signal to the 3.3V Pentium CPU. The tag comparator's output is sampled at the end of the first T2, and BRDY# is generated one clock later for cache hits, resulting in a leadoff of three cycles. BRDY# will go inactive to add wait states depending on the wait states programmed. Refer to Table 4-7 for the tag compare table.

If two SRAM banks are used, address bit A4 from the CPU will be the least significant address bit that goes to the data SRAMs. The data output for each SRAM bank is controlled by a separate output enable for each SRAM bank (OCDOE# and ECDOE#). The OCDOE#/ECDOE# generation for the leadoff cycle is based on address bit A3 from the CPU. The two signals OCDOE# and ECDOE# will interleave the data read from the two cache banks in a burst cycle. If one SRAM bank is used, address bit A3 from the CPU will be the least significant address bit that goes to the SRAMs and the output enable ECDOE# will be active for the complete cycle.

Figures 4-4 through 4-6 show various L2 cache read hit cycles.

Figure 4-4 L2 Cache Read Hit Cycle - Async SRAMs (Double Bank)

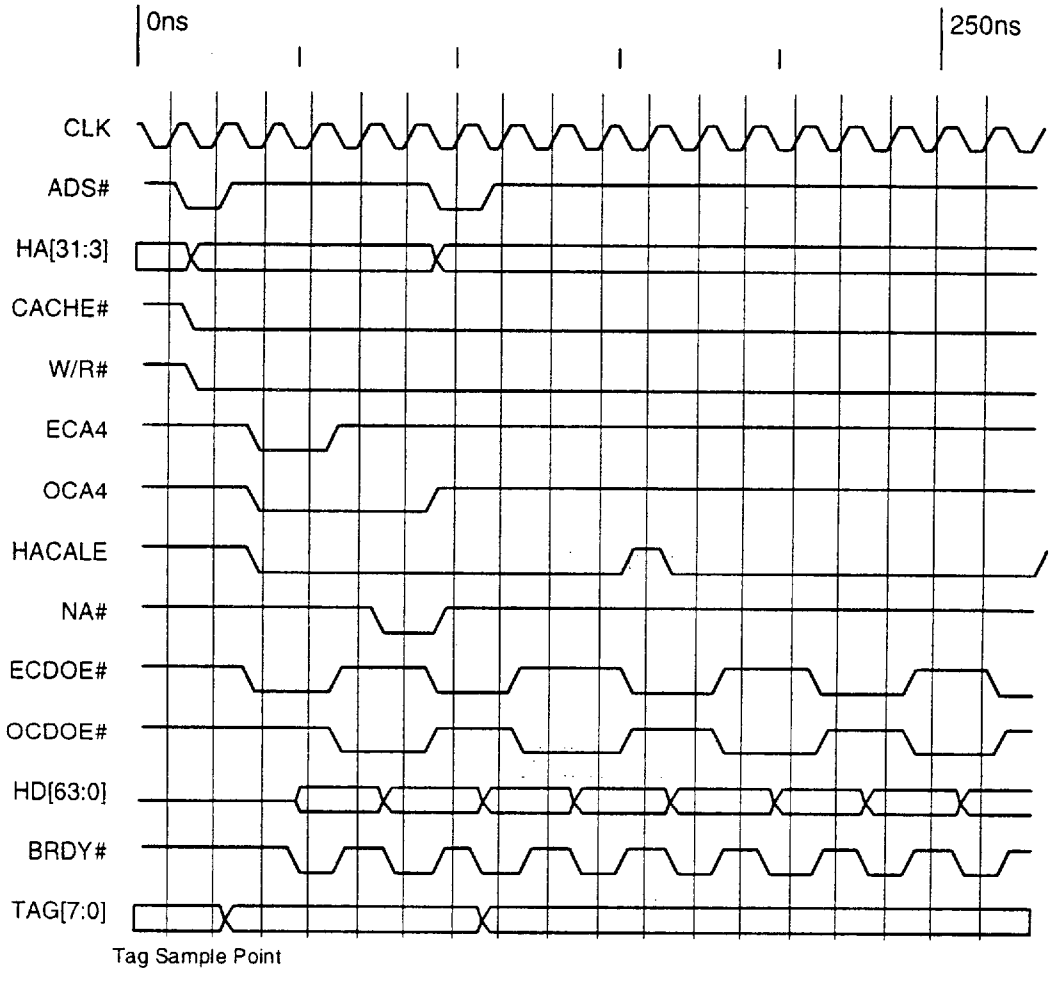


Figure 4-5 L2 Cache Read Hit Cycle - Sync SRAMs

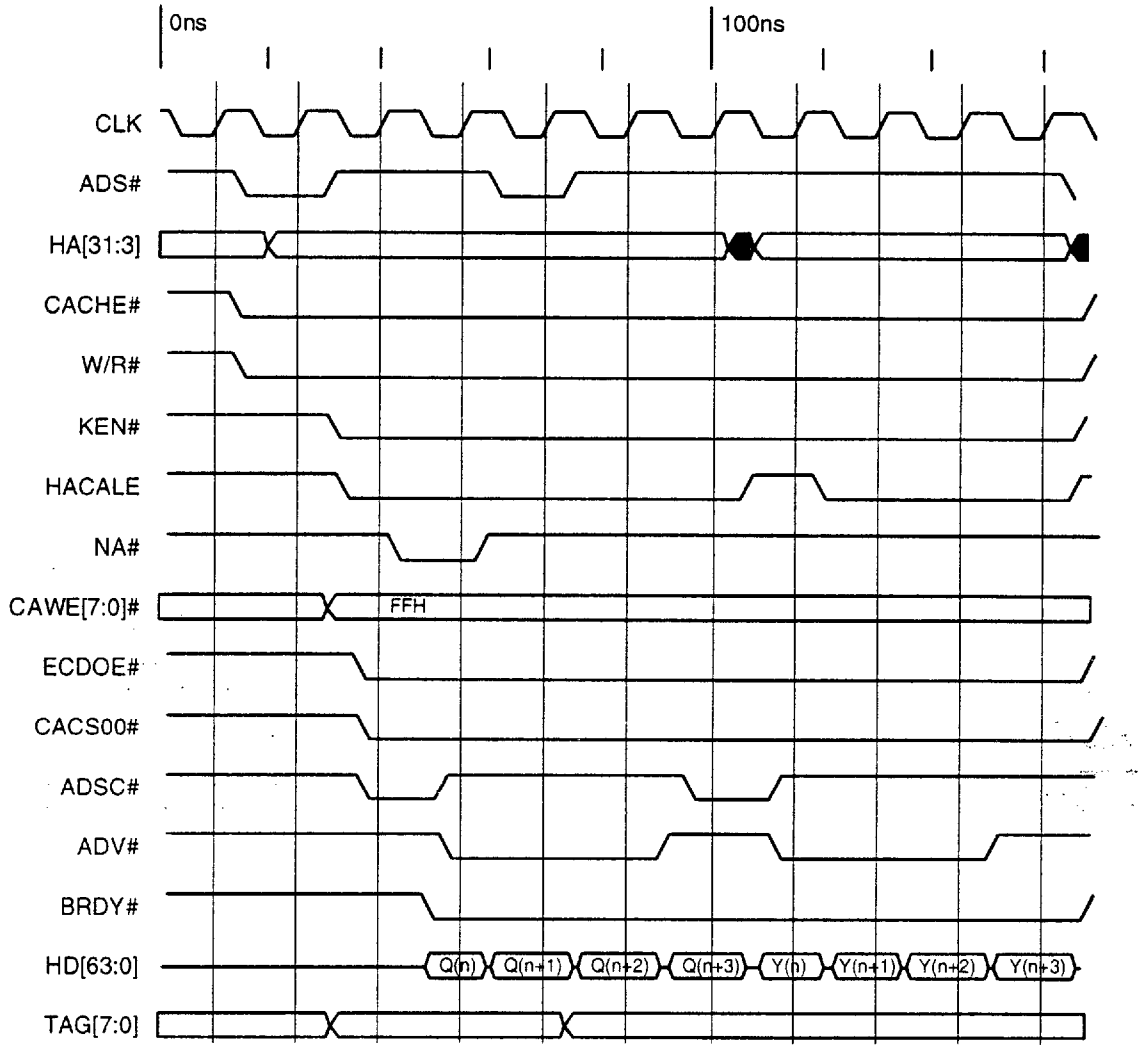
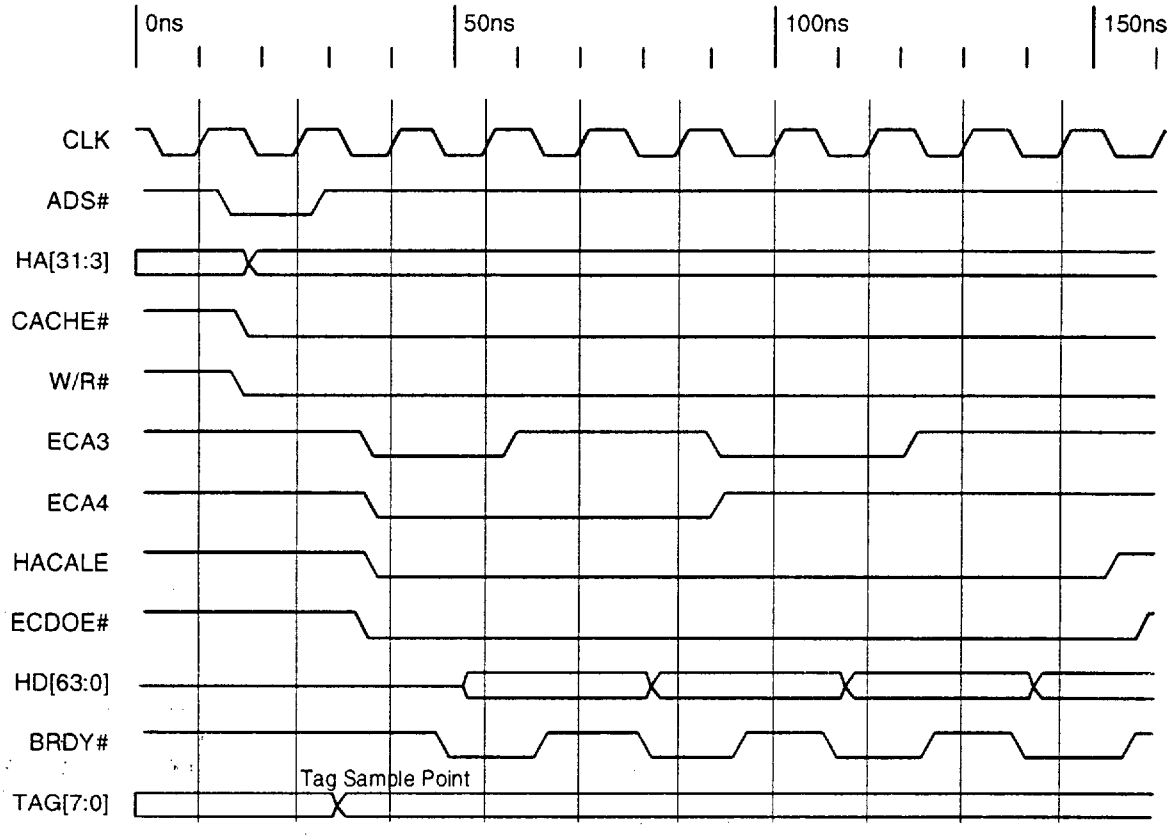


Figure 4-6 L2 Cache Read Hit Cycle Async SRAMs (Single Bank)



4.3.4.2 L2 Cache Write Hit Cycle

Write-through Mode: In this mode, data is always written to the L2 cache and to the system memory. The dirty bit is not used. When the write to the system memory is completed, BRDY# is returned to the CPU.

Write-back Mode: For a write hit case, the data is written only to the L2 cache (the system memory is not updated) and the dirty bit is always made dirty. The cache controller will sample CACHE# from the CPU at the end of T1 and execute a burst write if CACHE# is sampled active, otherwise the cycle will end in a single write. In this mode, the write cycle is completed in a 3-2-2-2 burst. For synchronous SRAMs, the cycle can be completed in a 2-1-1-1 burst if operating at 50MHz. The write enable signals OCAWE# and ECAWE# to the SRAM odd and even banks respectively, are based on address bit A3 from the CPU and will interleave writes to the two banks.

For writes, only the byte requested by the CPU can be written to the cache. This is done by using the BEx# from the CPU to control the SRAM chip selects.

Refer to Figures 4-7 through 4-9 show various write hit burst cycles.

Figure 4-7 Write Hit Burst Cycle for Write-Back Mode - Async SRAM (Double Bank)

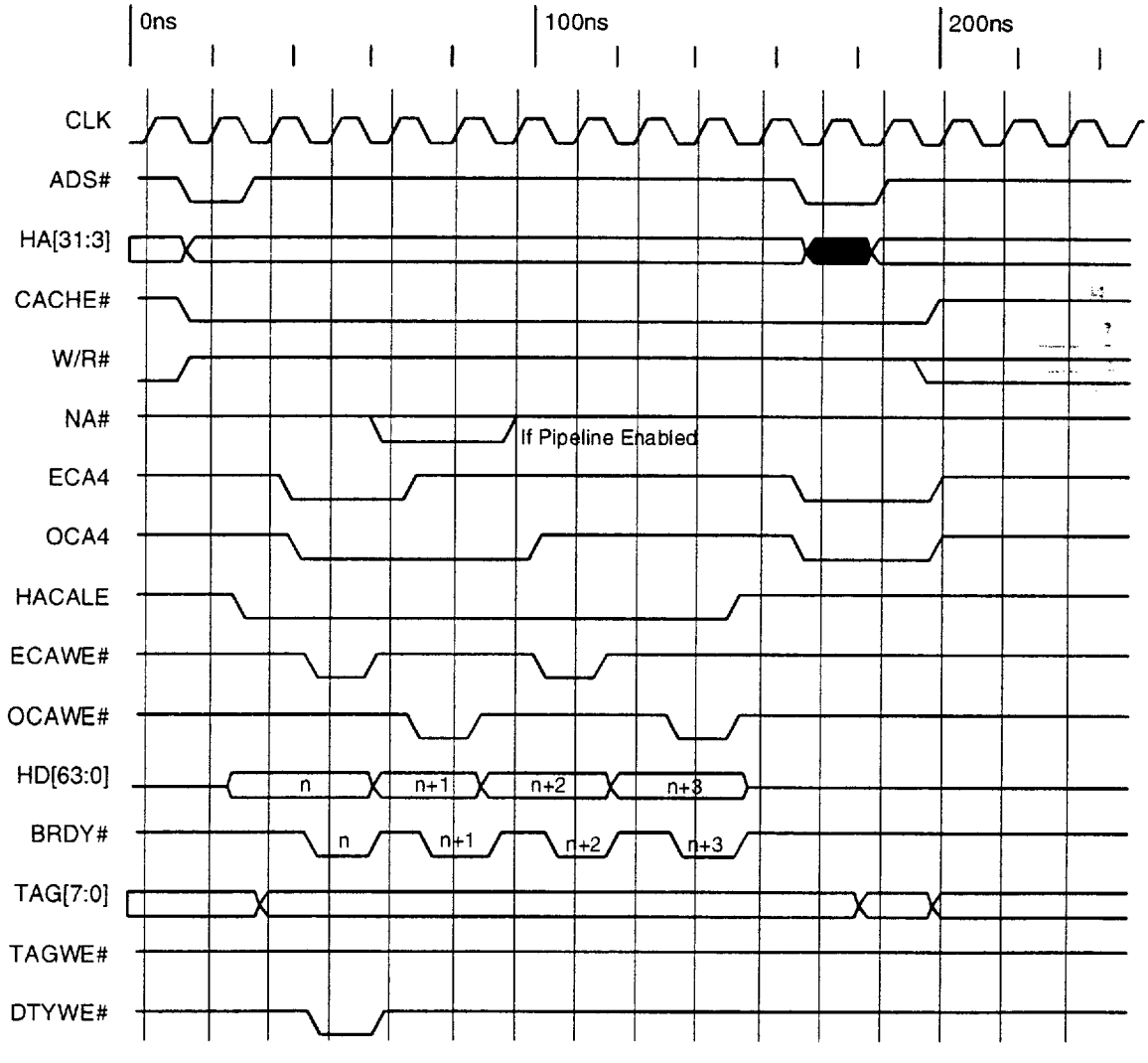


Figure 4-8 Write Hit Burst Cycle for Write-Back Mode (Single Bank) - Async SRAM

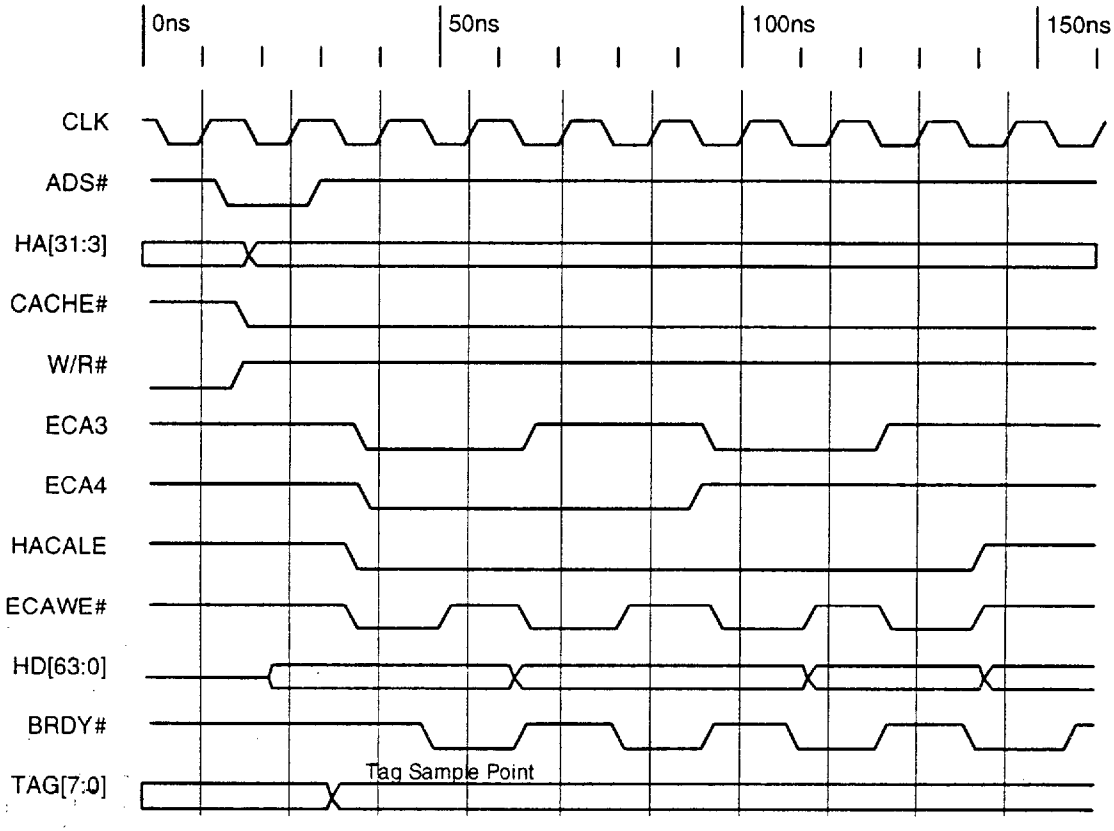
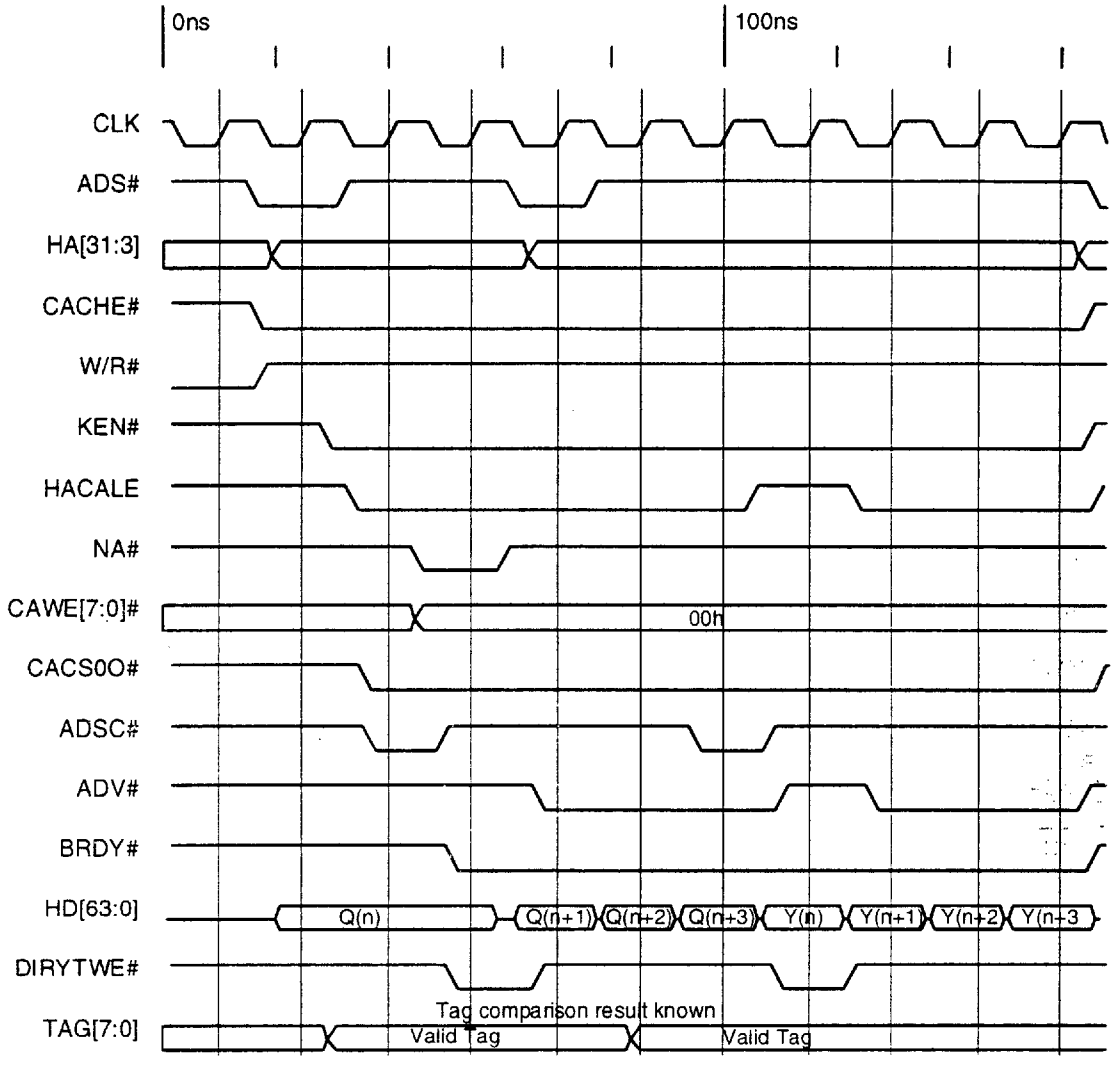


Figure 4-9 Write Hit Burst Cycle for Write-Back Mode - Sync SRAM



4.3.4.3 L2 Cache Read Miss

Write-back Mode: There are two cache read miss cases depending on the status of the dirty bit.

CASE 1: Read miss of a "clean" cache line.

In this case, only a linefill cycle is executed. The L2 cache line that is to be replaced with a new line from the DRAM will just be overwritten. The linefill cycle is done by reading the new data from the system memory first and then the data is simultaneously written to both the CPU and the secondary cache.

The sequence for CASE 1 linefill is: System memory read ⇨ write to the L2 cache + CPU read.

The cache controller will update the tag data bits and the dirty bit in the background during the linefill cycle. At the end of T1, if the CACHE# signal from the CPU is negated, a linefill cycle will not be executed. Instead, only the eight bytes requested by the CPU will be read from the system memory. The tag and the dirty bit will not be updated.

CASE 2: Read miss with cache line dirty.

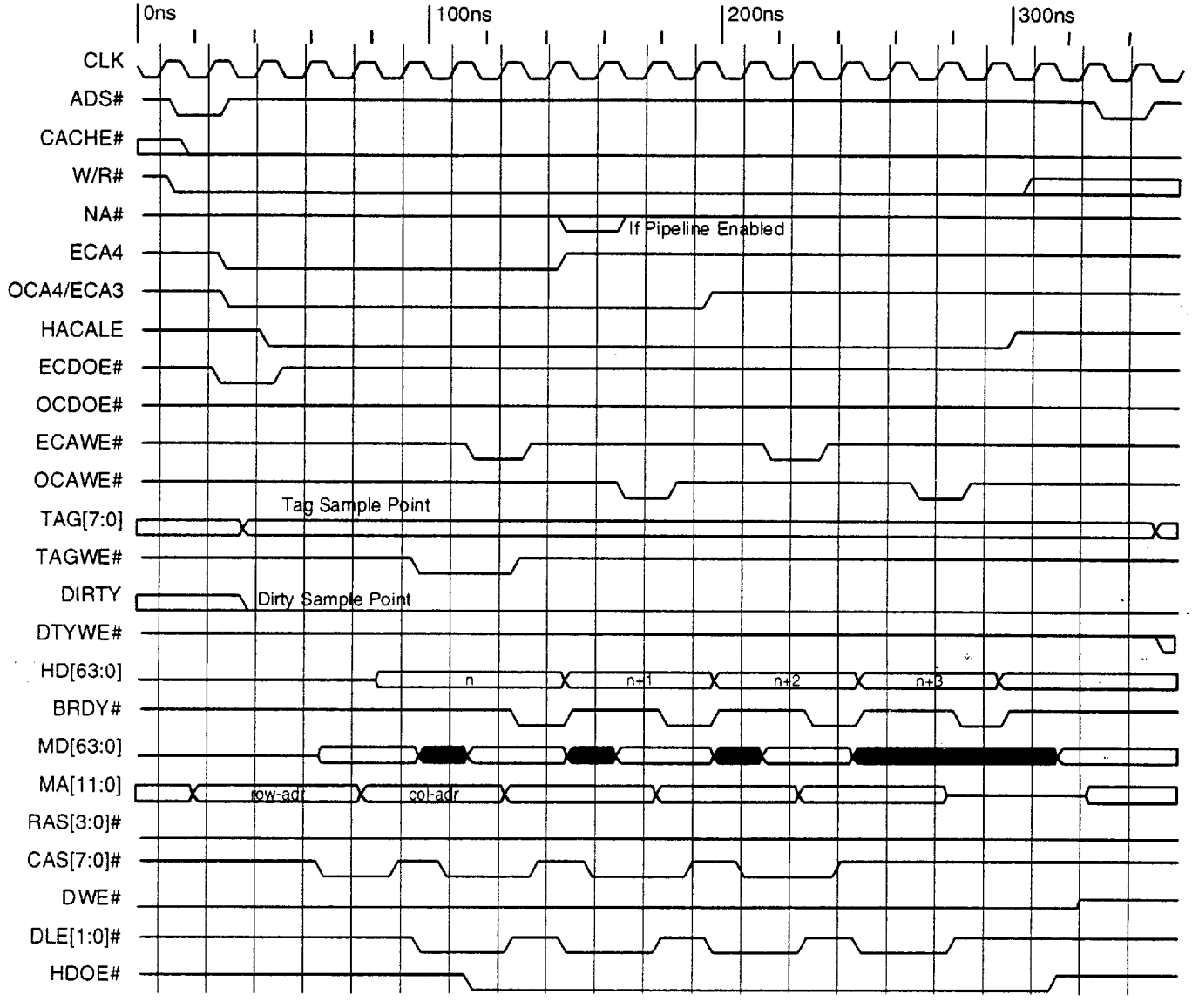
The cache line for this case has been modified and only the L1 and L2 cache have the updated copy of the data. Before this line is overwritten in the cache, the modified line must first be written to the system memory by performing a castout cycle. After the completion of the castout cycle, a linefill cycle is executed. The linefill cycle is performed by reading the new data from the system memory and then simultaneously writing this data to the CPU and the secondary cache.

The sequence for CASE 2 is: Read the dirty line from L2 cache ⇨ write to the system memory ⇨ new line read from system memory ⇨ write to the L2 cache + CPU read.

The cache controller will update the tag data bits and the dirty bit in the background during the castout cycle. If the CACHE# signal from the CPU is inactive, then the eight bytes requested by the CPU will be read from the system memory. The tag and the dirty bit are not updated.

Figures 4-10 through 4-12 show various L2 cache read miss cycles.

Figure 4-10 L2 Cache Read Miss Clean Burst of 8-3-3-3 (Linefill Cycle) - Async SRAMs



Note: This diagram is also for "DRAM Read Page Hit Cycle".

Figure 4-11 L2 Cache Read Miss Clean Burst of 8-3-3-3 (Linefill Cycle) - Sync SRAMs

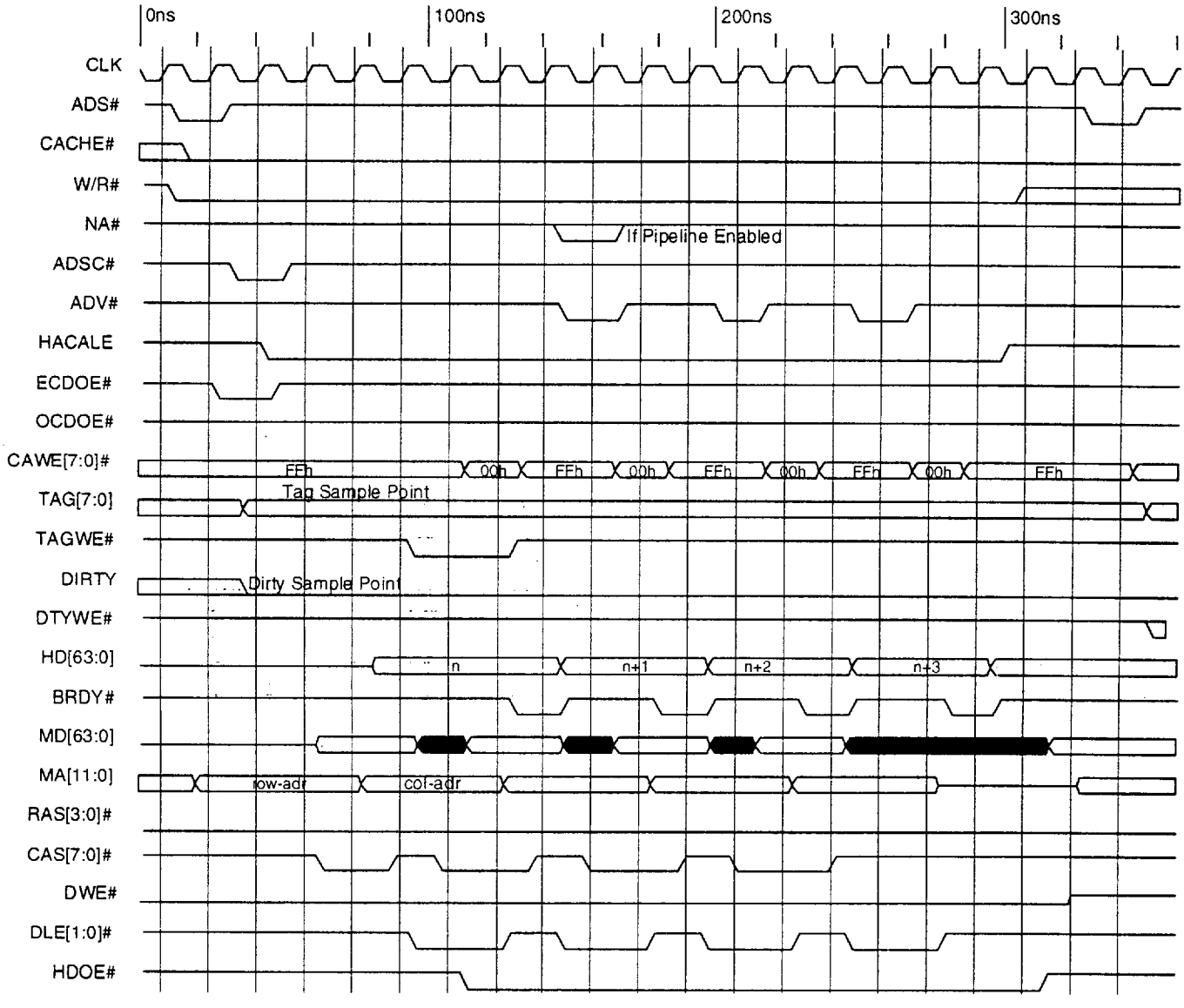
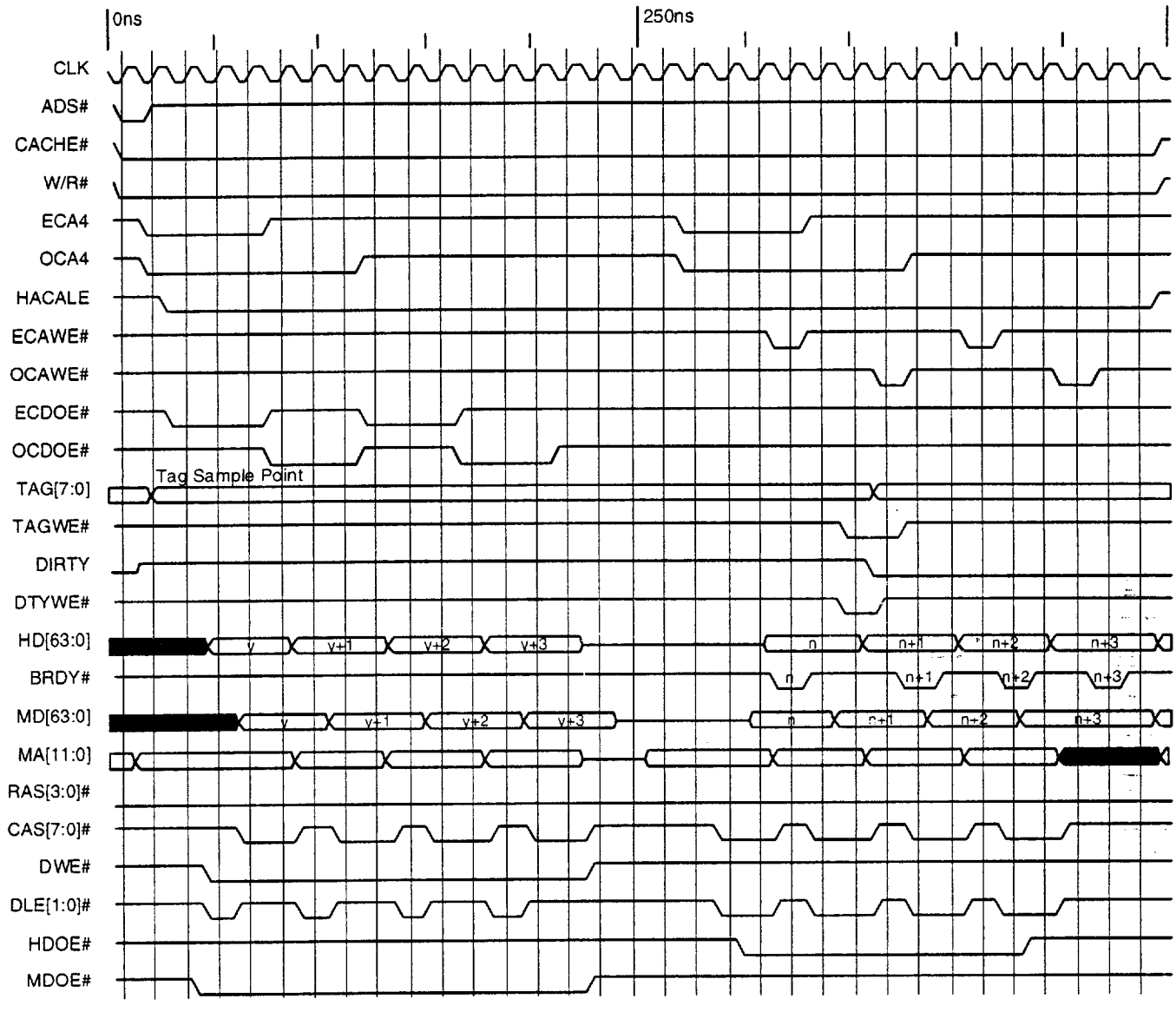


Figure 4-12 L2 Cache Read Miss Dirty Cycle - Async SRAMs



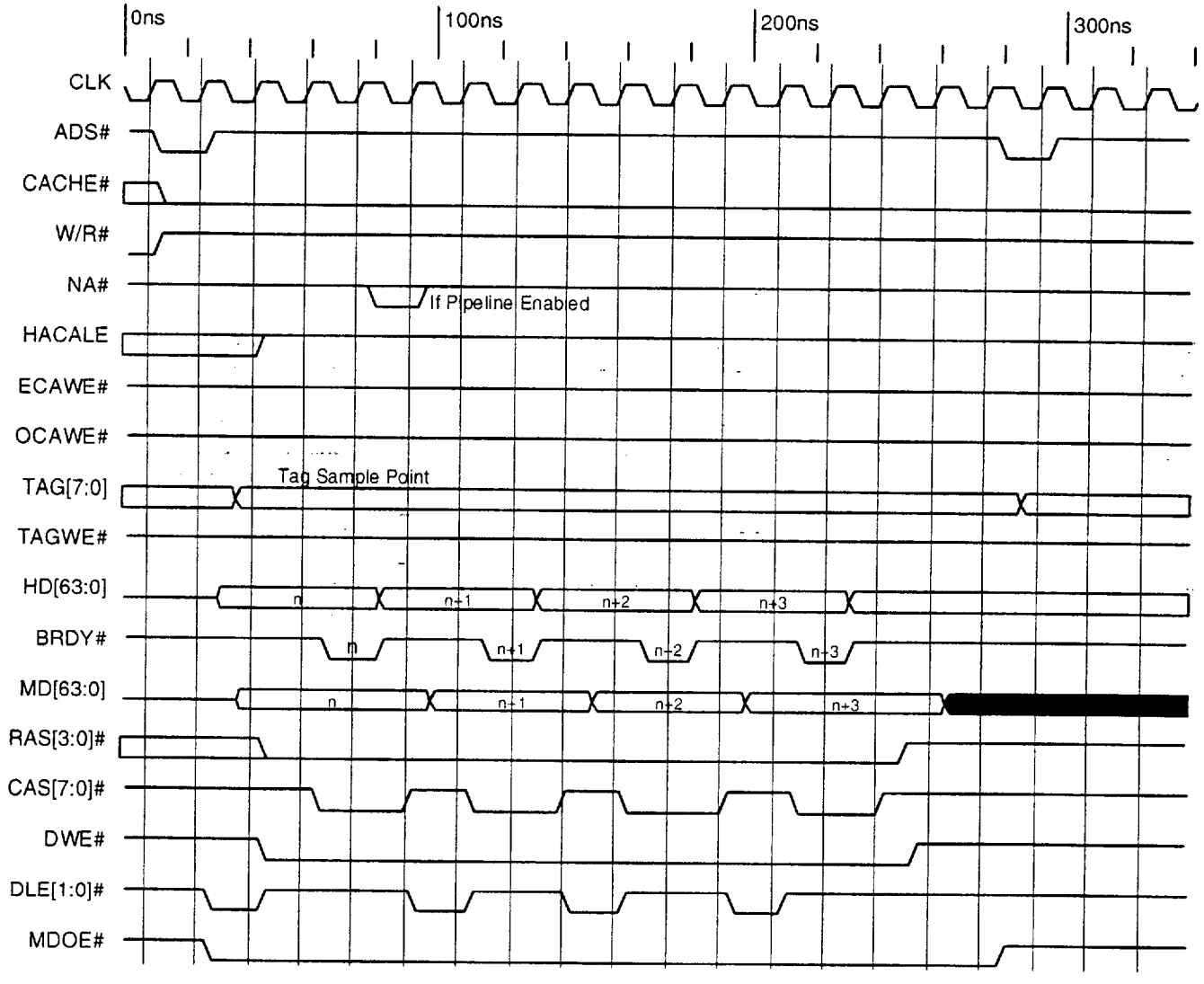
4.3.4.4 L2 Cache Write Miss

Write-back or Write-through Cases: The data is not written to the SRAM and the tag data remains unchanged. The data is written only to the system memory.

If the write buffer and DRAM posted write is enabled then is available, it is stored there and the cycles are posted writes to the DRAM. If the target is on the PCI or ISA bus, the cache controller will not be active.

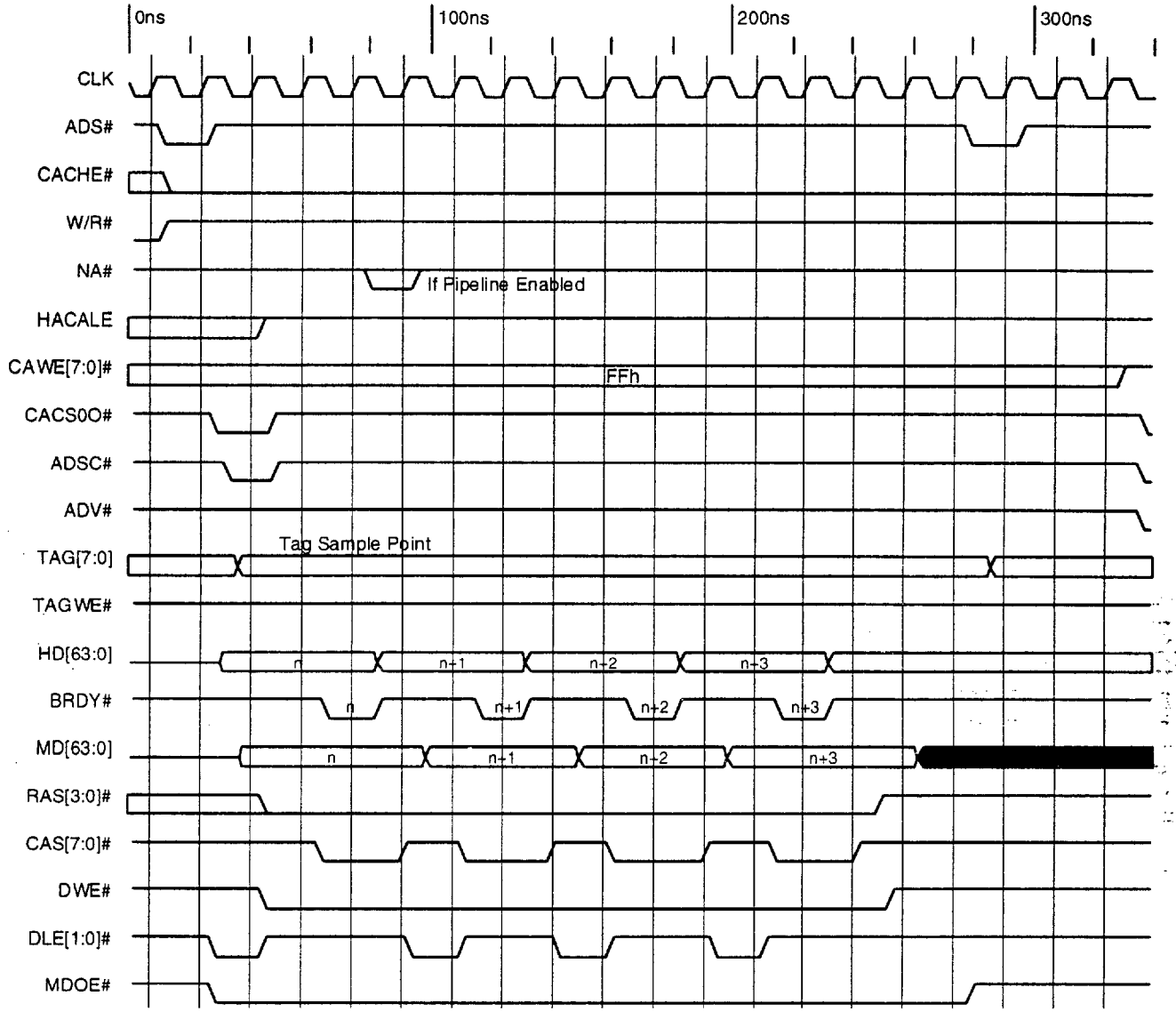
Figures 4-13 and 4-14 show L2 cache write miss cycles.

Figure 4-13 L2 Cache Write Miss Burst Cycle (4-3-3-3) - Async SRAMs



Note: This diagram is also for "DRAM Write Page Hit Cycle".

Figure 4-14 L2 Cache Write Miss Burst Cycle - Sync SRAMs



4.3.4.5 Write Policies

Any of the following three write policies supported by the Viper-MAX Chipset can be chosen: write-back, write-through, and adaptive write-back, by programming SYSCFG 02h[5:4] and SYSCFG 08h[1] (as shown in Table 4-6).

Depending on the state of these bits and the type of DRAM cycle that would be required to complete the write hit cycle, the cache controller decides whether to update the DRAM memory, however, the cache is always updated. The adaptive write-back policy tries to reduce the disadvantages of both the write-through and the write-back schemes to a minimum. The best case cache write burst timing (for an asycnhronous cache) is 3-2-2-2, and the best case DRAM page hit write burst timing is 4-3-3-3. The adaptive write-back scheme converts a write hit cycle to a write through cycle only if the address location being written to corresponds to a page hit. In this manner, this scheme incurs a four CLK penalty for a burst write cycle but it saves a 13 CLK penalty (for a castout cycle) that would have occurred later due to a read miss access. There are two adaptive write-back modes.

Write-Through on Page Hit and RAS# Active (AWB Mode 1)

In this mode, the data is written through to the DRAM on a write hit if the address being written to causes a page hit and the corresponding RAS# signal is active. The data will not be written through if, either the RAS# is inactive or if it is a page miss. In this case, the write hit cycle completes in the same manner as in a write-back scheme.

Table 4-6 Register Bits Associated with Write Policies

7	6	5	4	3	2	1	0
SYSCFG 02h			**Cache Control Register 1**				**Default = 00h**
L2 cache size selection: **If SYSCFG 0Fh[0] = 0** 00 = Reserved 01 = Reserved 10 = 256K 11 = 512K **If SYSCFG 0Fh[0] = 1** 00 = 1MB 01 = 2MB 10 = Reserved 11 = Reserved		L2 cache write policy: 00 = L2 cache write-through 01 = Adaptive Write-back Mode 1 10 = Adaptive Write-back Mode 2 11 = L2 cache write-back		L2 cache operating mode select: 00 = Disable 01 = Test Mode 1; External Tag Write (Tag data write-through SYSCFG 07h) 10 = Test Mode 2; External Tag Read (Tag data read from SYSCFG 07h) 11 = Enable L2 cache		DRAM posted write: 0 = Disable 1 = Enable	CAS precharge time: 0 = 2 CLKs 1 = 1 CLK
SYSCFG 08h			**CPU Cache Control Register**				**Default = 00h**
L2 cache single/double bank select: 0 = Double bank (If async SRAM, then the banks are interleaved. If sync SRAM, they are not interleaved.) 1 = Single bank (non-interleaved)	Snoop filtering for bus masters:(1) 0 = Disable 1 = Enable	CPU HITM# pin sample timing: 0 = Delay 1 CLK (HITM# sampled on 3rd rising edge of LCLK after EADS# assertion) 1 = No delay (HITM# sampled on 2nd rising edge of LCLK after EADS# assertion)	Parity checking: 0 = Disable 1 = Enable	Tag/Dirty RAM implementation: 0 = Tag and Dirty are on separate chip (i.e., a separate x1 or x8 SRAM for the Dirty RAM) 1 = Tag and Dirty are on the same chip (i.e., could be either a x9 or x8 Tag/Dirty RAM)	CPU address pipelining: 0 = Disable 1 = Enable	L1 cache write-back and write-through control: 0 = Write-through only 1 = Write-back enabled	BIOS area cacheability in L1 cache: Determines if system BIOS area E0000h-FFFFFh (if SYSCFG 04h[2] = 1) or F0000h-FFFFFh (if SYSCFG 04h[2] = 0), and video BIOS area C0000h-C7FFFh is cacheable in L1 or not. 0 = Cacheable 1 = Non-Cacheable

(1) For a master request if the subsequent read/write is within the same cache line, CPU 'Inquire' cycles are not done until there is a cache line miss (i.e., line comparator not activated for accesses within the same cache line).

Write-Through on Page Hit (AWB Mode 2)
In this mode, data is written through to the DRAM on a write hit if the address being written to causes a page hit. RAS# being active/inactive does not come into consideration when making this decision.

4.3.4.6 Tag Compare Table

The upper address bits used to compare for a L2 cache hit status will depend on the total L2 cache size. Table 4-7 shows the address bits from the CPU bus and the tag data bit used in the tag comparator of the 82C567. Figure 4-15 shows the block diagram of the L2 cache tag structure.

4.3.4.7 Tag and Dirty RAM implementations

There are various tag/dirty RAM implementations supported by the Viper-MAX Chipset.

Separate Tag/Dirty RAM Implementation
If a 32Kx1 part is used for the dirty RAM, there has to be a separate dirty input bit and a separate dirty output bit. In this implementation, the TAGWE# signal from the 82C567 is used to update the tag RAM and the DIRYTWE# signal from the 82C567 is used to update the dirty RAM. Only this implementation can provide a 3-2-2-2 write burst cycle at 66MHz. This scheme is shown in Figure 4-16.

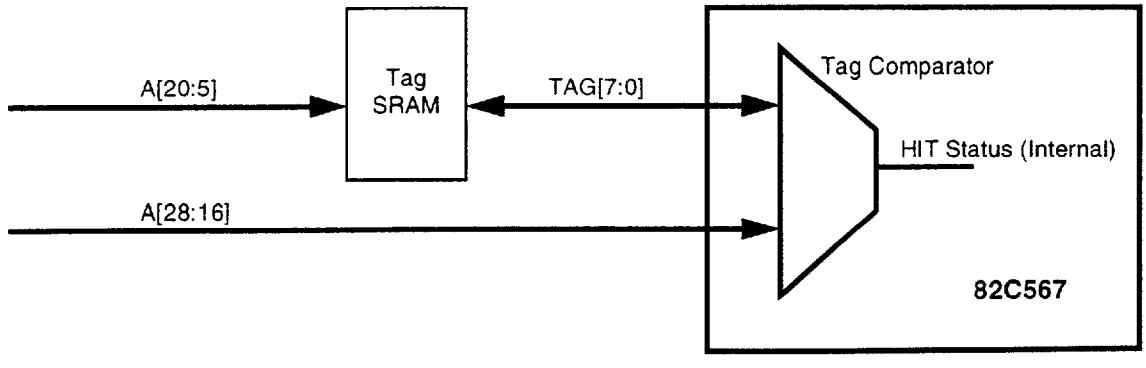
Combined Tag/Dirty RAM Implementation
There are various ways of achieving a combined tag/dirty RAM implementation. In all these implementations, the best write burst performance obtainable is a 4-2-2-2 cycle.

A 32Kx9 SRAM can be used to implement eight tag bits and one dirty bit. In this case, the TAGWE# signal from the 82C567 is used to update both the tag and dirty information. The OE# signal of the 32Kx9 SRAM can be connected to the DIRYTWE# signal from the 82C567 or it can be tied to GND. The DIRTYI signal of the 82C567 becomes a bidirectional signal and it now serves as the dirty I/O bit. This scheme is shown in Figure 4-17.

Table 4-7 Tag Compare Table

Tag Data	L2 Cache Size			
	256KB	512KB	1MB	2MB
TAG0	A24	A24	A24	A24
TAG1	A25	A25	A25	A25
TAG2	A18	A26	A26	A26
TAG3	A19	A19	A27	A27
TAG4	A20	A20	A20	A28
TAG5	A21	A21	A21	A21
TAG6	A22	A22	A22	A22
TAG7	A23	A23	A23	A23
Dirty Bit	Dirty	Dirty	Dirty	Dirty

Figure 4-15 82C567 Internal Tag Comparator Block Diagram



A 32Kx8 SRAM can be used, wherein seven bits are used for the tag RAM and one bit is used for the dirty RAM. In this case, the TAGWE# signal from the 82C567 is used to update both the tag and dirty information. The OE# of the 32Kx8 SRAM can be connected to the DIRYTWE# signal from the 82C567 or it can be tied to GND. TAG[7:1] convey the tag information and TAG0 becomes the dirty I/O bit. In this scheme, the amount of main memory that can be cached reduces by half as compared to an 8-bit tag implementation. This scheme is shown in Figure 4-18.

A 32Kx8 SRAM can be used to implement the eight tag bits and another 32Kx8 SRAM used to implement the single dirty I/O bit. This scheme is identical to the 32Kx9 implementation and is shown in Figure 4-19.

Table 4-8 Tag/Dirty RAM Control Register Bits

7	6	5	4	3	2	1	0
SYSCFG 16h			**Dirty/Tag RAM Control Register**				**Default = 00h**
DIRTYI pin selection:(1) 0 = Input only 1 = I/O	Reserved: Must be written to 0.	Tag RAM size selection:(2) 0 = 8-bit 1 = 7-bit	Single write hit leadoff cycle in a combined Dirty/Tag implementation:(3) 0 = 5 cycles 1 = 4 cycles	Pre-snoop control: 0 = Pre-snoop for starting address 0 only 1 = Pre-snoop for all addresses except those on the line boundary	Synchronization between the PCI bus clock (LCLK) and the CPU clock (CLK):(4) 0 = LCLK async to CLK 1 = LCLK sync to CLK (skew not to exceed -2ns to 15ns)	Reserved: Must be written to 0.	HDOE# timing control: 0 = Negated normally 1 = Negated one clock before the cycle finishes

(1) If using a x1 SRAM for the Dirty RAM in which there is a separate DirtyIn and a separate DirtyOut bit, then the DIRTYI pin becomes an input only. If using a standard x8 or x9 SRAM, where there is no separate pin for input and output, then the DIRTYI pin becomes an I/O pin.

(2) If a 7-bit Tag is being used and a combined Tag/Dirty RAM is being used, then TAG0 functions as the DIRTYIO signal. In this case, the DIRTYI pin is unused.

(3) If bit 4 is set 1, SYSCFG 22h[0] should be set to 1.

(4) It should be noted that LCLK could be async to CLK also. This bit therefore implies that the PCI clock is either sync to the CPU clock with a skew not to exceed -2ns to 15ns, or that the PCI clock is async to the CPU clock.

SYSCFG 08h			**CPU Cache Control Register**				**Default = 00h**
L2 cache single/double bank select: 0 = Double bank (If async SRAM, then the banks are interleaved. If sync SRAM, they are not interleaved.) 1 = Single bank (non-interleaved)	Snoop filtering for bus masters:(1) 0 = Disable 1 = Enable	CPU HITM# pin sample timing: 0 = Delay 1 CLK (HITM# sampled on 3rd rising edge of LCLK after EADS# assertion) 1 = No delay (HITM# sampled on 2nd rising edge of LCLK after EADS# assertion)	Parity checking: 0 = Disable 1 = Enable	Tag/Dirty RAM implementation: 0 = Tag and Dirty are on separate chip (i.e., a separate x1 or x8 SRAM for the Dirty RAM) 1 = Tag and Dirty are on the same chip (i.e., could be either a x9 or x8 Tag/Dirty RAM)	CPU address pipelining: 0 = Disable 1 = Enable	L1 cache write-back and write-through control: 0 = Write-through only 1 = Write-back enabled	BIOS area cacheability in L1 cache: Determines if system BIOS area E0000h-FFFFFh (if SYSCFG 04h[2] = 1) or F0000h-FFFFFh (if SYSCFG 04h[2] = 0), and video BIOS area C0000h-C7FFFh is cacheable in L1 or not. 0 = Cacheable 1 = Non-Cacheable

(1) For a master request if the subsequent read/write is within the same cache line, CPU 'Inquire' cycles are not done until there is a cache line miss (i.e., line comparator not activated for accesses within the same cache line).

Figure 4-16 Separate 32Kx8 and 32Kx1 Split Tag/Dirty RAM Implementation

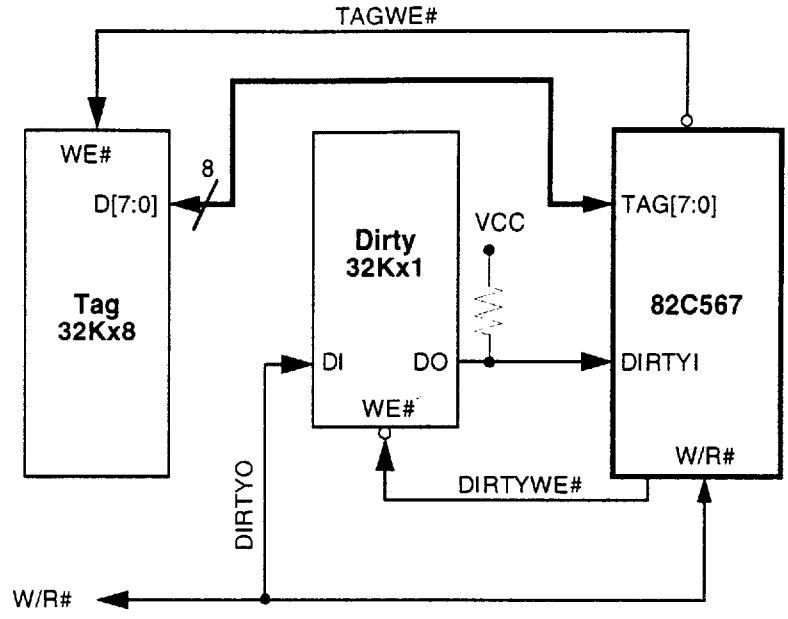


Figure 4-17 32Kx9 Combined Tag/Dirty RAM Implementation

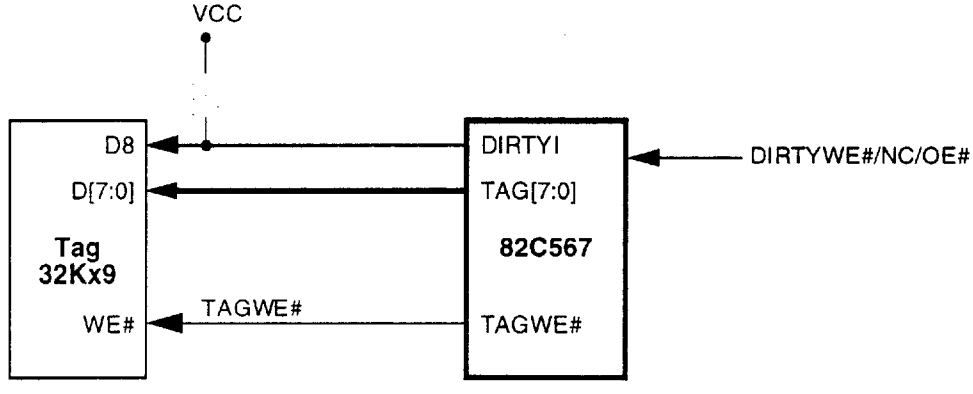


Figure 4-18 32Kx8 Combined Tag/Dirty RAM Implementation

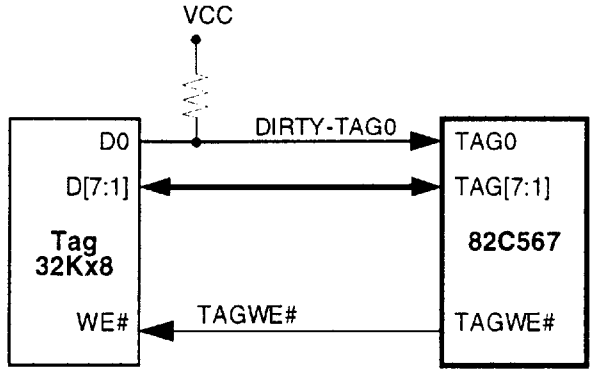
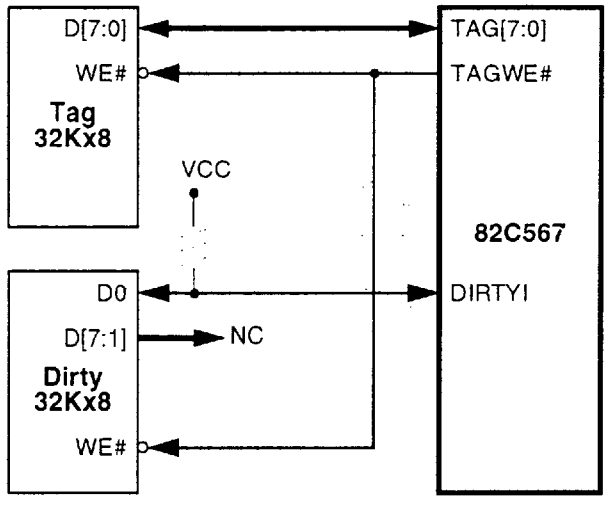


Figure 4-19 32Kx8 and 32Kx8 Combined Tag/Dirty RAM Implementation (Separate Devices)



4.3.4.8 Cache Initialization

On power-up, the tag RAM will contain random data and the L2 cache will contain no valid data. Therefore, the cache must be initialized before it is enabled.

Initializing Procedure 1: The cache is initialized by configuring the cache controller to the write-through mode. This will cause all the cache read miss cycles to fill the cache with valid data. This can be done by reading a block of system memory that is greater than or equal to the size of the cache. Once the cache is initialized, it is always valid. After this is done, the L2 cache can be set up for write-back operation by initializing the dirty bits. This is done by first enabling the cache controller to the write-back mode. Then, by reading a block of system memory that is greater than or equal to *twice* the size of the cache, the dirty bits will be cleared and the L2 cache will be valid.

Initializing Procedure 2: This procedure uses the cache controller in Test Mode 1 and Test Mode 2 as defined in SYSCFG 02h[3:2] and 07h[7:0]. (Refer to Table 4-4.)

The upper bits of an address is written to SYSCFG 07h. The cache controller is now set to Test Mode 2. Writing a block equal to the size of the cache to the system memory will write the contents of SYSCFG 07h to the tag. The cache controller is now configured in the write-through mode and reading a block of system memory equal to the size of the cache will make the data in the cache valid. Next, by reading a block of system memory which is greater than or equal to twice the size of the cache, the dirty bits will be cleared and the L2 cache will be valid.

Disabling the Cache: Disabling of a write-back cache cannot be done by just turning off the cache enable register bit in the 82C567. There may still be valid data in the cache that has not been written to the system memory. Disabling write-back cache without flushing this valid data usually causes a system crash.

This situation can be avoided by first reading a cacheable memory block *twice* the size of the cache. "Twice the size" of the cache is required to make sure every location gets a read miss, which will cause a castout cycle if the cache line is dirty. The cache can then be disabled. ***Note: No writes should occur during this process.***

4.3.4.9 Write Back Cache with DMA/ISA Master/PCI Master Operation

The L1 and the L2 cache contain the only valid copy (modified) of the data. The 82C567 will execute an inquire cycle to the L1 cache for all master accesses to the system memory area. This will increase the bus master cycle time for every access to the system memory which will also decrease the bus master performance. The Viper-MAX Chipset provides the option of a snoop-line comparator (snoop filtering) to increase the performance of a bus master with the L1 cache.

L1 Cache Inquire Cycle: This cycle begins with the CPU relinquishing the bus with the assertion of HLDA. On sampling HLDA active, the 82C567 will assert AHOLD. The address will flow from the master to the CPU bus and the 82C567 will assert EADS# for one CPU clock. If the CPU does not respond with the assertion of HITM#, the 82C567 will complete the cycle from the L2 cache or the system memory. If HITM# was asserted, the 82C567 will expect a castout cycle from the L1 cache.

Table 4-9 Test Mode Selection/Control Bits

7	6	5	4	3	2	1	0
SYSCFG 02h			**Cache Control Register 1**				**Default = 00h**
L2 cache size selection: **If SYSCFG 0Fh[0] = 0** 00 = Reserved 01 = Reserved 10 = 256K 11 = 512K **If SYSCFG 0Fh[0] = 1** 00 = 1MB 01 = 2MB 10 = Reserved 11 = Reserved		L2 cache write policy: 00 = L2 cache write-through 01 = Adaptive Write-back Mode 1 10 = Adaptive Write-back Mode 2 11 = L2 cache write-back		L2 cache operating mode select: 00 = Disable 01 = Test Mode 1; External Tag Write (Tag data write-through SYSCFG 07h) 10 = Test Mode 2; External Tag Read (Tag data read from SYSCFG 07h) 11 = Enable L2 cache		DRAM posted write: 0 = Disable 1 = Enable	CAS precharge time: 0 = 2 CLKs 1 = 1 CLK
SYSCFG 07h			**Tag Test Register**				**Default = 00h**
Data from this register is written to the tag, if in Test Mode 1 (refer to SYSCFG 02h). Data from the tag is read into this register, if in Test Mode 2 (refer to SYSCFG 02h).							

DMA/Master Read Cycle: Table 4-10 shows the action taken by the 82C567 based on the L1 and L2 cache status for bus master reads from the system memory area. The L1 cache castout cycle will be completed in the burst order provided by the CPU and will be written to the L2 cache or the system memory based on the L2 cache status. The required bytes are then read back for the completion of the master read cycle. A read hit in the L1 cache will always invalidate the L1 cache line. Refer to Figures 4-20 and 4-21.

DMA/Master Write Cycle: Table 4-11 shows the action taken by the 82C567 based on the L1 and L2 cache status for bus master writes to the system memory area. A master write to the L2 cache will always be in the write-through mode. The L1 cache castout cycle will be completed in the CPU burst sequence and the data will be written to the L2 cache or to the system memory based on the L2 cache status. Data from the master is always written to the system DRAM memory and is written to the L2 cache only if it is a L2 cache hit. Refer to Figure 4-22.

Table 4-10 DMA/Master Read Cycle Summary

DMA/Master Read Cycle					
L1 Cache	L2 Cache	Data Source	Type of Cycle for L1 Cache	Type of Cycle for L2 Cache	Type of Cycle for DRAM
Hit	Hit	L2 Cache	Invalidate	Read the Bytes Requested	No Change
hitM	Hit	L1 Cache	Castout, invalidate	Write CPU Data, Read Back the Bytes Requested	No Change
Hit	Miss	DRAM	Invalidate	No Change	Read the Bytes Requested
hitM	Miss	L1 Cache	Castout, invalidate	No Change	Write CPU Data, Read Back the Bytes Requested
Miss	Hit	L2 Cache	No Change	Read the Bytes Requested	No Change
Miss	Miss	DRAM	No Change	No Change	Read

Note: hitM - L1 cache modified

Table 4-11 DMA/Master Write Cycle Summary

DMA/Master Write Cycle					
L1 Cache	L2 Cache	Data Destination	Type of Cycle for L1 Cache	Type of Cycle for L2 Cache	Type of Cycle for DRAM
Hit	Hit	DRAM, sec	Invalidate	Write Master Data	Write Master Data
hitM	Hit	DRAM, sec	Castout, Invalidate	Write CPU Data, Write Master Data	Write Master Data
Hit	Miss	DRAM	Invalidate	No Change	Write Master Data
hitM	Miss	DRAM	Castout, Invalidate	No Change	Write CPU Data, Write Master Data
Miss	Hit	DRAM, sec	No Change	Write Master Data	Write Master Data
Miss	Miss	DRAM	No Change	No Change	Write Master Data

Figure 4-20 ISA DMA/Master Read (L1 cache with non-modified line)

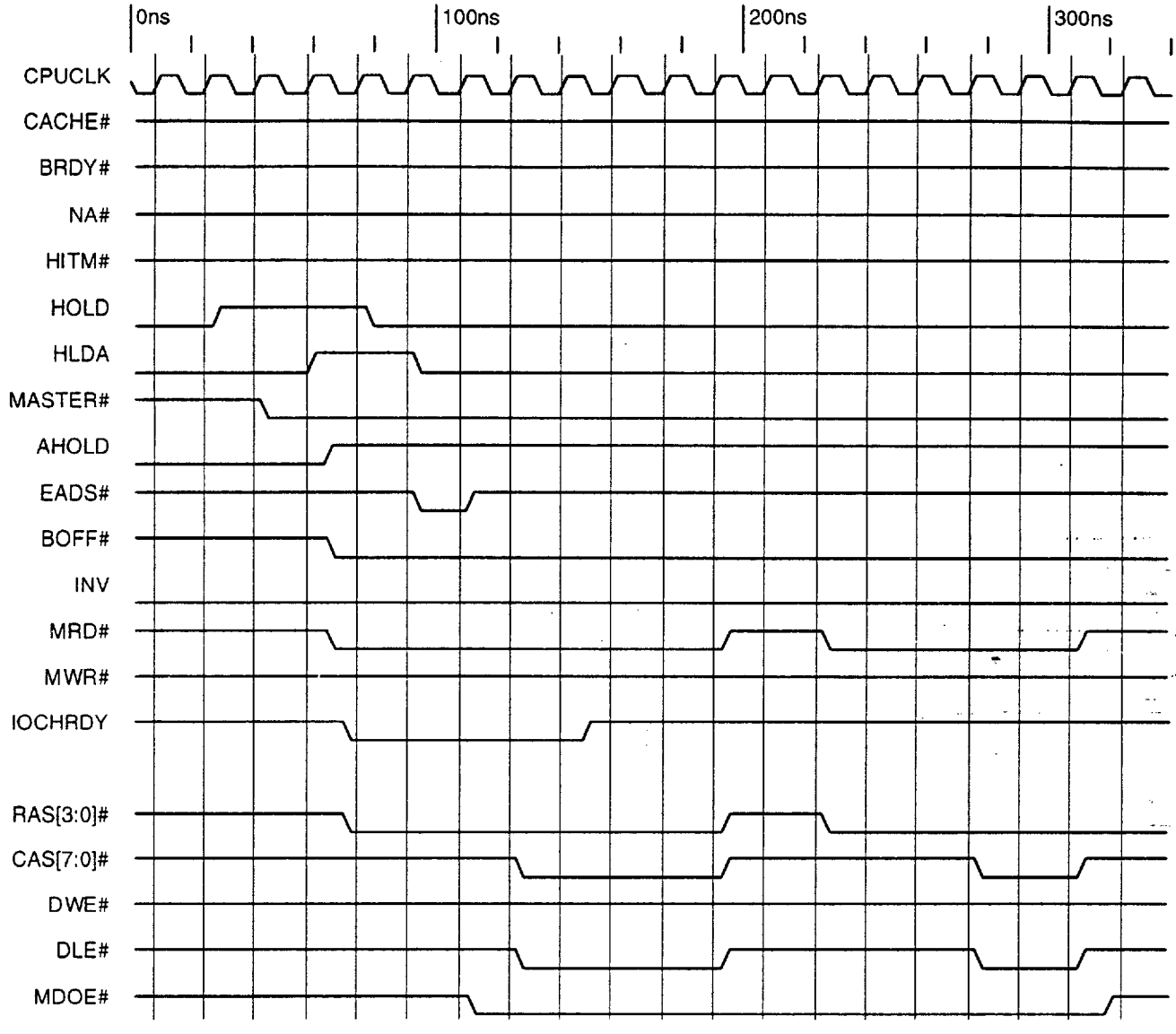


Figure 4-21 ISA DMA/Master Read (L1 cache with modified line)

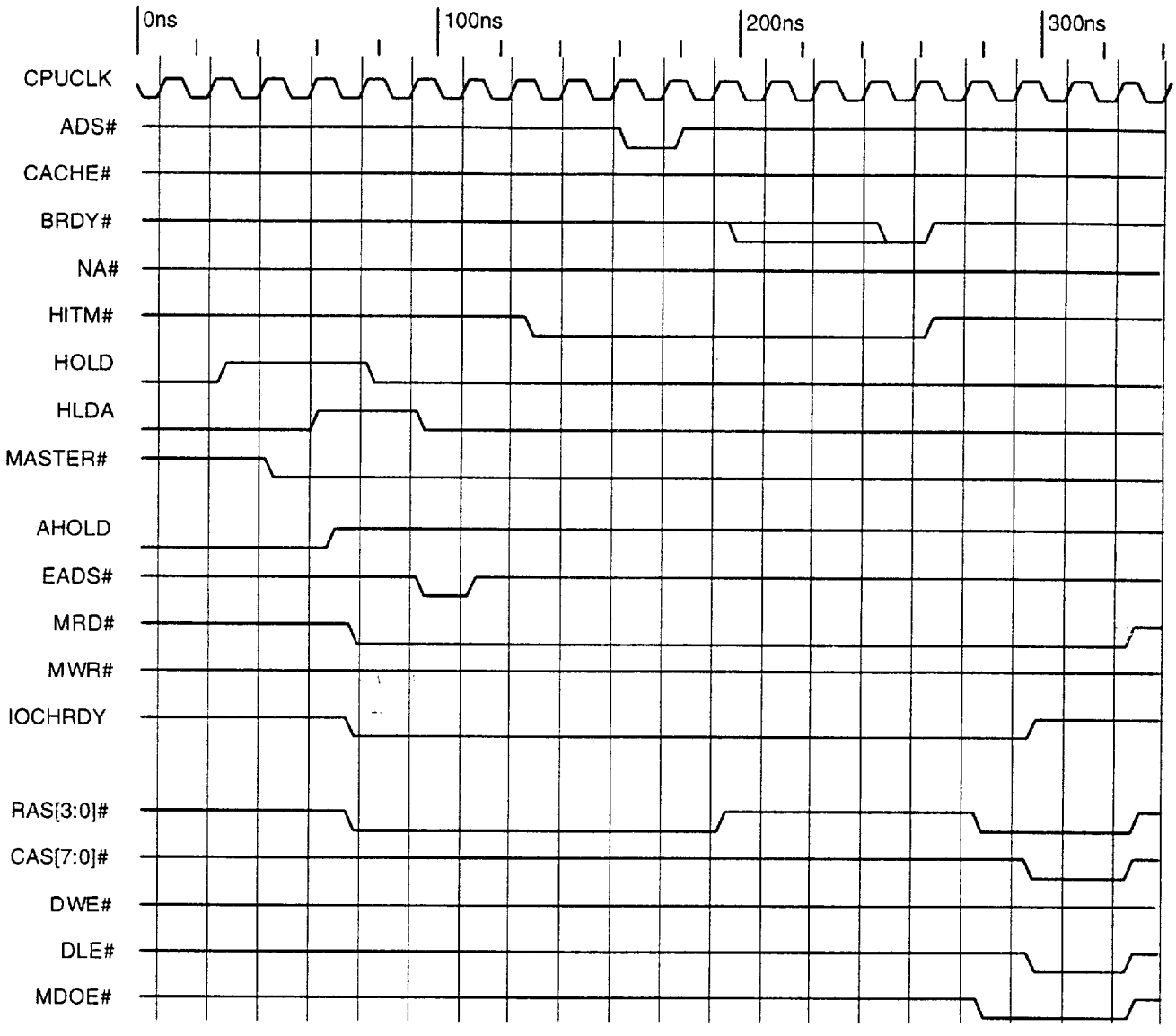
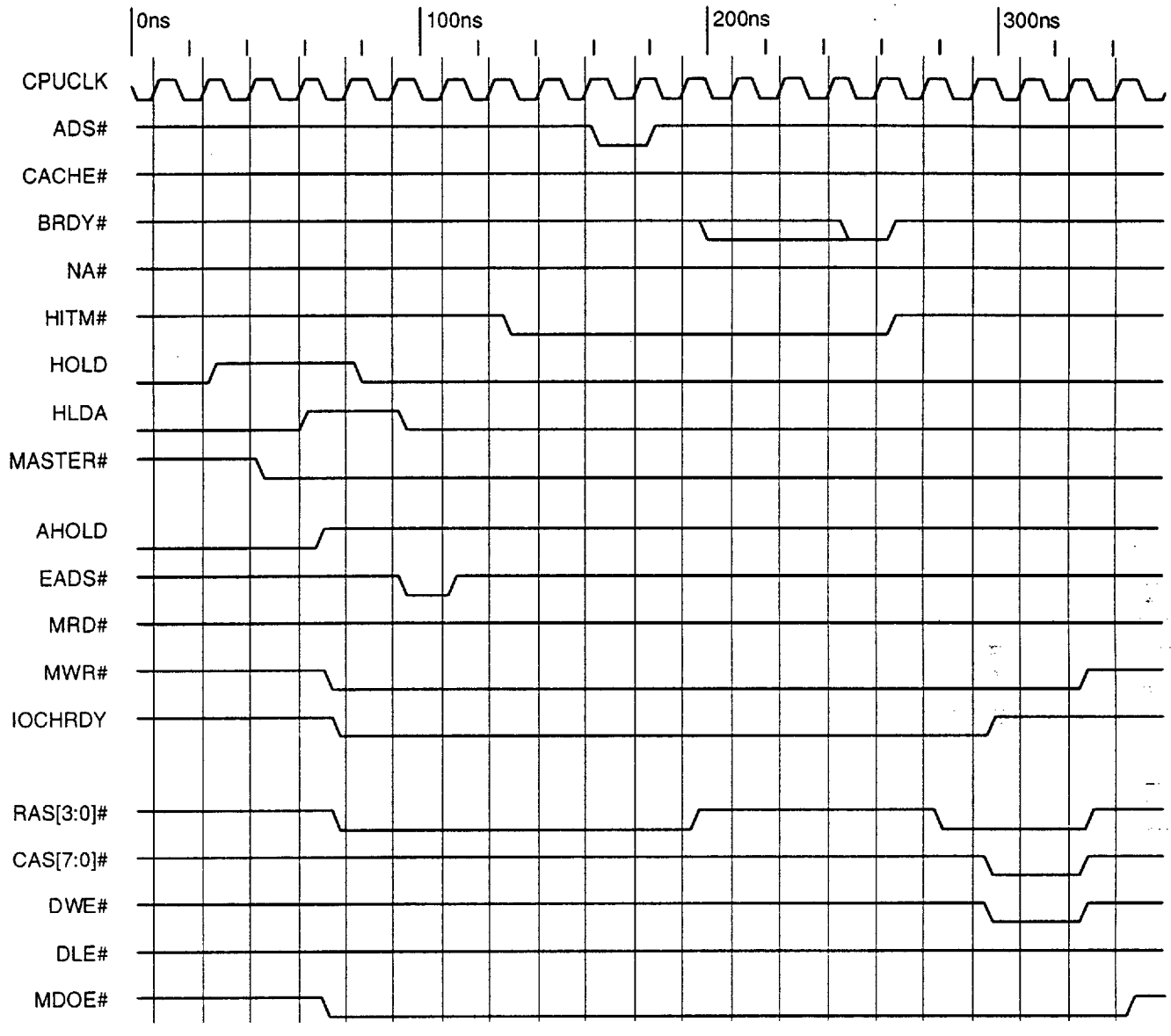


Figure 4-22 ISA DMA/Master Write (L1 cache with modified line)



4.3.5 Shadow ROM and BIOS Cacheability

When using the Viper-MAX Chipset, the procedures listed below should be followed for proper setup and configuration of shadow RAM utilities.

1. Enable ROMCS# generation for the segment to be shadowed. Although the F0000h-FFFFFh segment defaults to ROMCS# generation, the C, D, and E0000h ROM segments must have ROMCS# generation enabled by setting the appropriate bits in PCIDV1 4Ah and 4Bh.
2. Enable ROM contents to be copied into DRAM. To do this, the appropriate bits in SYSCFG 04h, 05h, and 06h should be set. These bits must be set so that reads from these segments will be executed out of ROM but will be written to DRAM.
3. Enable shadow RAM areas to permit DRAM read/write accesses. At this point, the ROMCS# generation bits that were previously necessary to access the original ROM code, must be disabled.

4. Write protect shadow RAM areas. To do this, the appropriate bits in SYSCFG 04h, 05h, and 06h should be set. These bits must be set so that reads from these segments will be executed out of DRAM, but writes will be directed to the ROM.

5. Cache shadow RAM areas in L2/L1 caches (optional). Caching of the individual code segments can be accomplished by setting the appropriate bits in SYSCFG 06h. Although write protection control for the L2 cache is provided, the L1 cache does not have a write protection mechanism and the ROM code may be overwritten or modified if stored in the L1 cache.

4.3.5.1 Cacheability and Write Protection

The Viper-MAX Chipset allows certain ROM areas to be cacheable. C0000h-C7FFFh, E0000h-EFFFFh, and F0000h-FFFFFh have separate cache-related controls. See Table 4-12.

Both system DRAM and shadow RAM are cacheable in both the primary (L1) and/or secondary (L2) cache. Of these two areas, only the shadow RAM areas (system BIOS, video BIOS and DRAM) have the capability of being write-protected (Non-shadowed BIOS ROM areas are implicitly write-protected). Since the possibility exists that write-protected shadow RAM can be cached, there also exists the possibility that this data might be modified inside the cache and subsequently executed. To prevent this from occurring, an explicit control mechanism must be used that prevents the unexpected from happening. There are three methods for controlling write protection in the Viper-MAX Chipset. (See Table 4-15 for a summary of these methods.)

Table 4-12 Cacheability Area Control Bits

7	6	5	4	3	2	1	0
SYSCFG 04h			**Shadow RAM Control Register 1**				**Default = 00h**
CC000h-CFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		C8000h-CBFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		Sync SRAM pipelined read cycle 1-1-1-1 enable:(1) 0 = Implies leadoff T-state for read pipelined cycle = 2(2) 1 = Enables leadoff T-state for read pipelined cycle = 1(3)	E0000h-EFFFFh range selection: Determines whether this region will be treated like the F0000 BIOS area or whether it will always be non-cacheable. 0 = E0000h-EFFFFh area will always be non-cacheable 1 = E0000h-EFFFFh area will be treated like the F0000h BIOS area. If this bit is set, then SYSCFG 06h[3:2] and [1:0] Should be set identically.	C0000h-C7FFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	

(1) If SYSCFG 11h[3] = 1 (i.e., sync SRAM chosen) and if SYSCFG 03h[3:2] = 11, then this register setting comes into play.
(2) It will be a 3-1-1-1 cycle followed by a 2-1-1-1 cycle, or a 3-1-1-1 cycle for successive piped cycles.
(3) It will be a 3-1-1-1 cycle followed by a 1-1-1-1 cycle for successive piped cycles. This is valid only for a single bank case.

Table 4-12 Cacheability Area Control Bits (cont.)

7	6	5	4	3	2	1	0
SYSCFG 05h			**Shadow RAM Control Register 2**				**Default = 00h**
DC000h-DFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PC I/ write to DRAM 11 = Read/write DRAM		D8000h-DBFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		D4000h-D7FFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		D0000h-D3FFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	
SYSCFG 06h			**Shadow RAM Control Register 3**				**Default = 00h**
DRAM hole in system memory from 80000h-9FFFFh:(1) 0 = No hole in memory 1 = Enable hole in memory	Wait state addition for PCI master snooping: 0 = Do not add a wait state for the cycle access finish to do the snooping 1 = Add a wait state for the cycle access to finish and then do the snooping	C0000h-C7FFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM If SYSCFG 04h[2] = 1, then the E0000h-EFFFFh read/write control should have the same setting as this.		E0000h-EFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	
(1) This setting gives the user the option to have some other device in the address range 80000h-9FFFFh instead of system memory. When bit 7 is set, the 82C567 will not start the system DRAM controller for accesses to this particular address range.							
PCIDV1 4Ah			**ROMCS# Range Control Register - Low Byte**				**Default = 00h**
ROMCS# for F8000h-FFFFFh: 0 = Enable 1 = Disable	ROMCS# for F0000h-F7FFFh: 0 = Enable 1 = Disable	ROMCS# for E8000h-EFFFFh: 0 = Disable 1 = Enable	ROMCS# for E0000h-E7FFFh: 0 = Disable 1 = Enable	ROMCS# for D8000h-DFFFFh: 0 = Disable 1 = Enable	ROMCS# for D0000h-D7FFFh: 0 = Disable 1 = Enable	ROMCS# for C8000h-CFFFFh: 0 = Disable 1 = Enable	ROMCS# for C0000h-C7FFFh: 0 = Disable 1 = Enable
PCIDV1 4Bh			**ROMCS# Range Control Register - High Byte**				**Default = 00h**
ROMCS# for FFFF8000h-FFFFFFFFh segment: 0 = Enable 1 = Disable	ROMCS# for FFFF0000h-FFFF7FFFh segment: 0 = Enable 1 = Disable	ROMCS# for FFFE8000h-FFFEFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFE0000h-FFFE7FFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFD8000h-FFFDFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFD0000h-FFFD7FFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFC8000h-FFFCFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFC0000h-FFFC7FFFh segment: 0 = Disable 1 = Enable

METHOD 1: In this method, the write protected areas are **not** cached in the L1 or the L2 cache. This is implemented by driving KEN# high for the first word with BRDY#, which will cause the CPU to not cache the data in its L1 cache and not do burst cycles. Data in the L2 cache is also not updated, so all reads and writes to this area will go directly to or from the system memory or to/from system BIOS/video BIOS (if they are not shadowed). Table 4-13 shows the associated shadow control register bits.

Table 4-13 Shadow Control Register Bits

7	6	5	4	3	2	1	0
SYSCFG 04h				**Shadow RAM Control Register 1**			**Default = 00h**
CC000h-CFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		C8000h-CBFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		Sync SRAM pipelined read cycle 1-1-1-1 enable:(1) 0 = Implies leadoff T-state for read pipe-lined cycle = 2(2) 1 = Enables leadoff T-state for read pipe-lined cycle = 1(3)	E0000h-EFFFFh range selection: Determines whether this region will be treated like the F0000 BIOS area or whether it will always be non-cacheable. 0 = E0000h-EFFFFh area will always be non-cacheable 1 = E0000h-EFFFFh area will be treated like the F0000h BIOS area. If this bit is set, then SYSCFG 06h[3:2] and [1:0] Should be set identically.	C0000h-C7FFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	
(1) If SYSCFG 11h[3] = 1 (i.e., sync SRAM chosen) and if SYSCFG 03h[3:2] = 11, then this register setting comes into play.							
(2) It will be a 3-1-1-1 cycle followed by a 2-1-1-1 cycle, or a 3-1-1-1 cycle for successive piped cycles.							
(3) It will be a 3-1-1-1 cycle followed by a 1-1-1-1 cycle for successive piped cycles. This is valid only for a single bank case.							
SYSCFG 05h				**Shadow RAM Control Register 2**			**Default = 00h**
DC000h-DFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PC I/ write to DRAM 11 = Read/write DRAM		D8000h-DBFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		D4000h-D7FFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		D0000h-D3FFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	

Table 4-13 Shadow Control Register Bits (cont.)

7	6	5	4	3	2	1	0
SYSCFG 06h			**Shadow RAM Control Register 3**				**Default = 00h**
DRAM hole in system memory from 80000h-9FFFFh:(1) 0 = No hole in memory 1 = Enable hole in memory	Wait state addition for PCI master snooping: 0 = Do not add a wait state for the cycle access finish to do the snooping 1 = Add a wait state for the cycle access to finish and then do the snooping	C0000h-C7FFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM If SYSCFG 04h[2] = 1, then the E0000h-EFFFFh read/write control should have the same setting as this.		E0000h-EFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	
(1) This setting gives the user the option to have some other device in the address range 80000h-9FFFFh instead of system memory. When bit 7 is set, the 82C567 will not start the system DRAM controller for accesses to this particular address range.							
SYSCFG 0Eh			**PCI Master Burst Control Register 1**				**Default = 00h**
Reserved: Must be written to 0.	Pin 55 and 57 functionality: **0 = Pin 55 is OCDOE# or HLDA or SDRAS# and Pin 57 is OCAWE#, CAS1O#, HOLD, or SDCAS#** **1 = Pin 55 is MEMR# and Pin 57 is MEMW#**	Reserved: Must be written to 0.	Reserved: Must be written to 0.	Parity check during master cycles (if SYSCFG 08h[4] = 1): 0 = Enable 1 = Disable	Fast NA# generation: 0 = Disable 1 = Enable for every single transfer cycle	Write protection for L1 BIOS: 0 = No 1 = Yes	PCI line comparator (if SYSCFG 08h[6] = 1): 0 = Use line comparator in PCI master 1 = Generate inquire cycle for every new FRAME#

METHOD 2: In this method, the write protected areas can be cached in the L2 cache but not in the L1 cache. This is implemented by driving KEN# high for the first word with BRDY#, which will cause the CPU to not cache the data in the L1 cache or do a burst cycle. This data can then be stored in the L2 cache, but only subsequent read requests by the CPU are serviced (discarding all writes), thus effectively write-protecting the data in the L2 cache. Read miss cycles are serviced by first performing a linefill burst from the DRAM into the L2 cache and then performing a normal non-cacheable (and non-burst) cycle to the CPU. In this method, writes to the system memory and to the L2 cache are write protected.

METHOD 3: This method is implemented by driving EADS#/WT# high during the read cycle. Data read from write protected areas are stored in both the L1 and L2 caches. Accesses from the CPU that are L2 cache read hits are serviced in burst mode and L2 cache read miss cycles are serviced by first performing a linefill burst read to the L2 cache from the write protected area and then performing a normal burst cycle to the CPU. Write cycles from the CPU to these areas are write-through and are discarded by the cache controller of the 82C567. ***However, L1 cache writes occur internally to the CPU in this mode and are therefore not write protected.*** Table 4-14 shows the register bit associated with this function.

Table 4-14 SYSCFG 08h[0]

7	6	5	4	3	2	1	0
SYSCFG 08h			**CPU Cache Control Register**				**Default = 00h**
L2 cache single/double bank select: 0 = Double bank (If async SRAM, then the banks are interleaved. If sync SRAM, they are not interleaved.) 1 = Single bank (non-interleaved)	Snoop filtering for bus masters:(1) 0 = Disable 1 = Enable	CPU HITM# pin sample timing: 0 = Delay 1 CLK (HITM# sampled on 3rd rising edge of LCLK after EADS# assertion) 1 = No delay (HITM# sampled on 2nd rising edge of LCLK after EADS# assertion)	Parity checking: 0 = Disable 1 = Enable	Tag/Dirty RAM implementation: 0 = Tag and Dirty are on separate chip (i.e., a separate x1 or x8 SRAM for the Dirty RAM) 1 = Tag and Dirty are on the same chip (i.e., could be either a x9 or x8 Tag/Dirty RAM)	CPU address pipelining: 0 = Disable 1 = Enable	L1 cache write-back and write-through control: 0 = Write-through only 1 = Write-back enabled	BIOS area cacheability in L1 cache: Determines if system BIOS area E0000h-FFFFFh (if SYSCFG 04h[2] = 1) or F0000h-FFFFFh (if SYSCFG 04h[2] = 0), and video BIOS area C0000h-C7FFFh is cacheable in L1 or not. 0 = Cacheable 1 = Non-Cacheable

(1) For a master request if the subsequent read/write is within the same cache line, CPU 'Inquire' cycles are not done until there is a cache line miss (i.e., line comparator not activated for accesses within the same cache line).

Table 4-15 Cacheability Methods

Method	System DRAM		System BIOS		Video BIOS		Write Enabled Shadow RAM		Write Protected Shadow RAM	
	Read	Write	Read	Write	Read	Write	Read	Write	Read	Write
1	L1,L2	L1,L2	Single	None	Single	None	L1,L2	L1,L2	Single	None
2	L1,L2	L1,L2	L2	None	L2	None	L1,L2	L1,L2	L2	None
3	L1,L2	L1,L2	L1,L2	L1	L1,L2	L1	L1,L2	L1,L2	L1,L2	L1

Note: **L1** = accessible to primary cache, **L2** = accessible to secondary cache, **none** = no cycle performed (or discard), **int** = internal cycle to CPU, **WT** = write-through cycle, **single** = single word (non-burst) cycle, **burst** = burst cycle

4.3.6 SRAM Support

The Viper-MAX Chipset supports many varieties of asynchronous and synchronous SRAMs. Table 4-16 shows which signals of the 82C567 change functionality to the various SRAM implementations.

In addition to the standard synchronous SRAMs, the Viper-MAX Chipset supports pipelined synchronous SRAMs as well as the Intel standard BSRAM and the Sony Sonic-2WP (cache module). Table 4-23 at the end of this section gives a SRAM cycle comparison chart.

4.3.6.1 SRAM Requirements

The data RAMs are quad-word interleaved for the two bank configuration, which requires 64-bit wide SRAM. This allows systems based on the Viper-MAX Chipset to perform a full 3-2-2-2 burst for reads and writes.

Tables 4-17 and 4-18 give configuration parameters, while Tables 4-19 and 4-20 outline the read/write cycle lengths and their speed requirements.

4.3.6.2 Pipelined Synchronous SRAM Support

Pipelined synchronous SRAMs are cheaper than their counterpart BiCMOS synchronous SRAMs (standard synchronous SRAMs). The timing requirement of the ADV# pin assertion is different for these SRAMs, and this is enabled by setting SYSCFG 17h[1] = 1 (i.e., enabling pipelined synchronous SRAM).

In a two bank synchronous SRAM implementation, there could be data contention when switching between banks. To avoid this, Intel has proposed a BSRAM standard. This standard requires the insertion of one "idle" cycle when switching between banks. The BSRAMs that support a one clock disable and a two clock enable timing, meet this standard. The Viper-MAX Chipset supports this standard. To enable this feature, SYSCFG 17h[5] should be set to 1. Table 4-21 lists the registers provided for SRAM support.

Table 4-16 82C567 Pin Functionality for SRAM Implementations

Pin No.	Async SRAM	Sync SRAM
62:69	CACS[7:0]#	CAWE[7:0]#
56	ECAWE#	CACS0O#
57	OCAWE#	CACS1O#
59	ECA4	ADSC#
60	OCA4+ECA3	ADV#
54	ECDOE#	ECDOE#
55	OCDOE#	OCDOE#

Table 4-17 Data SRAM Asynchronous Configurations

Cache Size	Data SRAMs		Tag SRAMs				Cacheable Range
	Qty	Type	Qty	Tag Address Field	Qty	Tag Dirty Bit Field	
256K Bytes	8	32Kx8	1	8Kx8	1	8Kx1	64MB
512K Bytes	16	32Kx8	1	16Kx8	1	16Kx1	128MB
1M Bytes	8	128Kx8	1	32Kx8	1	32Kx1	128MB
2M Bytes	16	128Kx8	1	64Kx8	1	64Kx1	128MB

Table 4-18 Data SRAM Synchronous Configurations

Cache Size	Qty	Size
256K Bytes	4	32Kx18
512K Bytes	4	64Kx18

Table 4-19 SRAM Cycle Lengths

Speed	Async SRAMs		Sync SRAMs	
	Burst Cycle	Leadoff Reduction if Pipelined	Standard	Pipelined Burst*
Read Burst Cycles				
50MHz	3-2-2-2	1 clock	2-1-1-1	2-1-1-1 1-1-1-1
60MHz	3-2-2-2	1 clock	3-1-1-1	3-1-1-1 1-1-1-1
66MHz	3-2-2-2	1clock	3-1-1-1	3-1-1-1 1-1-1-1
Write Burst Cycles				
50MHz	3-2-2-2	1 cycle	2-1-1-1	2-1-1-1 1-1-1-1
60MHz	4-2-2-2	1 cycle	3-1-1-1	3-1-1-1 1-1-1-1
66MHz	4-2-2-2	1 cycle	3-1-1-1	3-1-1-1 1-1-1-1

*This timing is for a single-bank. Leadoff cycle will be increased by one clock for a double-bank solution when it is a pipelined cycle due to the turn-around time of two banks and to prevent data contention between the banks.

Table 4-20 Tag and Data SRAM Speed Requirements

Parameter	Description	50MHz	60MHz	66MHz
Async SRAM Data				
tAA	Address Access Time	25ns	15ns	15ns
tOE	OE# Access Time	12ns	8ns	8ns
tWP	Write Pulse Width	25ns	14.5ns	14.5ns
SRAM Tag for Async Cache System				
tAA	Address Access Time	20ns	15ns	12ns
Sync SRAM Data				
tCD	Clock Access Time	12ns (2-1-1-1)/ 12ns (3-1-1-1)	9ns	9ns
SRAM Tag for Sync Cache System				
tAA	Address Access Time	10ns (2-1-1-1)/ 20ns (3-1-1-1)	15ns	12ns

Table 4-21 Register Bits Associated with SRAM Support

7	6	5	4	3	2	1	0
SYSCFG 02h		**Cache Control Register 1**					**Default = 00h**
L2 cache size selection: **If SYSCFG 0Fh[0] = 0** 00 = Reserved 01 = Reserved 10 = 256K 11 = 512K **If SYSCFG 0Fh[0] = 1** 00 = 1MB 01 = 2MB 10 = Reserved 11 = Reserved		L2 cache write policy: 00 = L2 cache write-through 01 = Adaptive Write-back Mode 1 10 = Adaptive Write-back Mode 2 11 = L2 cache write-back		L2 cache operating mode select: 00 = Disable 01 = Test Mode 1; External Tag Write (Tag data write-through SYSCFG 07h) 10 = Test Mode 2; External Tag Read (Tag data read from SYSCFG 07h) 11 = Enable L2 cache		DRAM posted write: 0 = Disable 1 = Enable	CAS precharge time: 0 = 2 CLKs 1 = 1 CLK
SYSCFG 03h		**Cache Control Register 2**					**Default = 00h**
Timing for burst writes to L2 cache: 00 = X-4-4-4 10 = X-2-2-2 01 = X-3-3-3 11 = X-1-1-1		Leadoff cycle time for writes to L2 cache: 00 = 5-X-X-X 10 = 3-X-X-X 01 = 4-X-X-X 11 = 2-X-X-X(1)		Timing for burst reads to L2 cache: 00 = X-4-4-4 10 = X-2-2-2 01 = X-3-3-3 11 = X-1-1-1		Leadoff cycle time for reads to L2 cache: 00 = 5-X-X-X 10 = 3-X-X-X 01 = 4-X-X-X 11 = 2-X-X-X(1)	
(1) Sync SRAM double bank implementation does not support this timing.							
SYSCFG 04h		**Shadow RAM Control Register 1**					**Default = 00h**
CC000h-CFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		C8000h-CBFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM		Sync SRAM pipelined read cycle 1-1-1-1 enable:(1) 0 = Implies leadoff T-state for read pipelined cycle = 2(2) 1 = Enables leadoff T-state for read pipelined cycle = 1(3)	E0000h-EFFFFh range selection: Determines whether this region will be treated like the F0000 BIOS area or whether it will always be non-cacheable. 0 = E0000h-EFFFFh area will always be non-cacheable 1 = E0000h-EFFFFh area will be treated like the F0000h BIOS area. If this bit is set, then SYSCFG 06h[3:2] and [1:0] Should be set identically.	C0000h-C7FFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	
(1) If SYSCFG 11h[3] = 1 (i.e., sync SRAM chosen) and if SYSCFG 03h[3:2] = 11, then this register setting comes into play. (2) It will be a 3-1-1-1 cycle followed by a 2-1-1-1 cycle, or a 3-1-1-1 cycle for successive piped cycles. (3) It will be a 3-1-1-1 cycle followed by a 1-1-1-1 cycle for successive piped cycles. This is valid only for a single bank case.							

Table 4-21 Register Bits Associated with SRAM Support (cont.)

7	6	5	4	3	2	1	0
SYSCFG 0Fh			**PCI Master Burst Control Register 2**				**Default = 00h**
PCI pre-snoop: 0 = Disable 1 = Enable(1)	Insert wait states for ISA master access: 0 = No 1 = Yes	Reserved: Must be written to 0.	Resynchronize PCI master accesses to system DRAM: 0 = No 1 = Yes(2)	New mode of single cycle NA#: 0 = Disable 1 = Enable	CPU to L2 cache hit cycles, ASDC# generation from chipset: 0 = Enable 1 = Disable(3)	Write pulse duration control for operation with async SRAM: This bit is used when the write cycle takes the form of 3-X-X-X. 0 = 1 CPUCLK 1 = CPUCLK/2 plus the delay of an internal delay line	Cache size selection: This bit along with SYSCFG 02h[1:0] defines the L2 cache size. 0 = < 1MB 1 = ≥1MB

(1) The 82C567 generates a pre-snoop cycle to the CPU assuming that the PCI master will do a burst.

(2) If bit 4 = 1 in sync SRAM mode, PCI master access to system memory will force the master to wait for the current cycle to finish and the CPU-PCI clock to become sync. This is a conservative mode.

(3) SYSCFG 0Fh[2] needs to be set if pipelined sync SRAMs are being used.

SYSCFG 11h			**Miscellaneous Control Register 2**				**Default = 00h**
Reserved: Must be written to 0.		Cache inactive during Idle state control: This bit controls the chip selects of the SRAMs. 0 = SRAM active always 1 = SRAM inactive during Idle state	Next address (NA#) mode control:(1) 0 = Normal NA# timing used with async SRAMs 1 = New NA# timing for sync SRAMs - used only when CPU operating at 50MHz	SRAM type: 0 = Async SRAM 1 = Sync SRAM	Page miss posted write: 0 = Enable 1 = Disable	ISA/DMA IOCHRDY control: 0 = Old mode, no IOCHRDY during line hit 1 = Drive IOCHRDY low until cycle is finished	Delay start: 0 = Old mode, do not delay internal master cycle cycles after an inquire cycle 1 = Delay internal master cycles by one LCLK after inquire cycle

(1) If the CPU is used at a 50MHz operating frequency, then a 2-1-1-1 cycle on read/write hits to the sync SRAM can be obtained. To obtain this performance, the ADS# output of the CPU needs to be connected to the ADSP# input of the sync SRAM directly and bit 4 needs to be set. By setting bit 4, generation of the NA# signal from the chipset to the CPU is controlled.

SYSCFG 17h			**PCI Cycle Control Register 2**				**Default = 00h**
Reserved: Must be written to 0.	Generate NA# for PCI slave access in async LCLK mode: 0 = No 1 = Yes This bit will be overridden if bit 7 is set.	Sync two bank select: 0 = Reserved 1 = Set this bit to 1 when two banks of sync SRAM are installed	Reserved: Must be written to 0.	Reserved: Must be written to 0.	Pipelining during byte merge: 0 = Disable 1 = Enable	Sync SRAM type (if SYSCFG 11h[3] = 1): 0 = Standard 1 = Pipelined	Burst type: 0 = Intel burst protocol 1 = Cyrix linear burst protocol

4.3.6.3 Sony SONIC-2WP (Cache Module) Support

The Sony SONIC-2WP is a single chip, write-back cache subsystem that integrates 256Kbytes of cache memory, tag RAM and all other associated control logic. The integrated 256Kbyte cache is direct-mapped and it supports 3-1-1-1 burst cycles, and operates at 3.3V. If this chip is used, SYSCFG 00h[5] should be set to 1. This causes a few changes in the signal functions of the 82C567. The TAG1 and the TAG2 signals are connected to the START# signal from the Sony cache module.This signal is asserted by the Sony cache module when a CPU cycle translates to a read miss, write miss, or a write-through cycle. The assertion of this signal by the cache module causes the 82C567 to take control of the KEN# and BRDY# signals which it shares with the cache module. The TAG3 signal is connected to the BOFF# signal from the Sony cache module. The remainder of the TAG lines should be unconnected. All the other cache control signals of the 82C567 are not required and should be no connects. The ADS# input of the 82C567 should be connected to the SADS# output from the cache module. One note of caution, CPU pipelining must be disabled if using this cache module.

Table 4-22 Cache Module Register Support

7	6	5	4	3	2	1	0
SYSCFG 00h			**Byte Merge/Prefetch & Sony Cache Module Control Register(1)**				**Default = 00h**
Enable pipelining of single CPU cycles to memory: 0 = Disable 1 = Enable	Video memory byte/word read prefetch enable: This setting enables/disables the prefetching of bytes/words/ from PCI video memory by the CPU. 0 = Disable 1 = Enable	Sony SONIC-2WP support enable:(2) 0 = No Sony SONIC-2WP installed 1= Sony SONIC-2WP installed	Byte/word merge support: 0 = Disable 1 = Enable	Byte/word merging with CPU pipelining (NA# generation) support: 0 = Disable 1 = Enable	Time-out counter for byte/word merge: This setting determines the maximum time difference between two consecutive PCI byte/word writes to allow merging. 00 = 4 CPU CLKs 01 = 8 CPU CLKs 10 = 12 CPU CLKs 11 = 16 CPU CLKs		Enable internal hold requests to be blocked while performing byte merge: 0 = Disable 1 = Enable

(1) SYSCFG 13h[7] must be set to 1 in order for this register to be mapped correctly (full memory decode).

(2) If bit 5 is set, ensure that the L2 cache has been disabled (i.e., set SYSCFG 02h[3:2] = 00).

Table 4-23 SRAM Comparisons

Cycles	Async	Sync	Pipelined Sync	Pipelined BSRAM	Sony Cache Module
Read hit	3-2-2-2	3-1-1-1	3-1-1-1	3-1-1-1	3-1-1-1
CPU piped RH	2-2-2-2	1-1-1-1	1-1-1-1	1-1-1-1	3-1-1-1(1)
2 BKs piped RH	2-2-2-2	1-1-1-1(2)	2-1-1-1(2)	2-1-1-1	3-1-1-1(1)
Write hit	3-2-2-2	3-1-1-1	3-1-1-1	3-1-1-1	3-1-1-1
Write-back	N	N	N+4	N+4	N+BOFF
PCI read	x-2-2-2	x-2-2-2	x-3-3-3	x-3-3-3	x-2-2-2(3)
PCI write	x-2-2-2	x-2-2-2	x-2-2-2	x-2-2-2	x-2-2-2(3)
Cost	Lowest	High	Low	Low	High

1) No CPU pipelined for Sony Cache Module.
2) Data bus conflict for sync. SRAM, minimum data bus conflict for pipelined SRAM with 82C567 OE# control.
3) L2 needs "castout" dirty line before master access.

4.4 DRAM Controller

The DRAM controller within the Viper-MAX Chipset uses a 64-bit wide DRAM data bus interface. It also uses the page mode technique for faster data access from the DRAMs.

Page mode is always used in the Viper-MAX Chipset for CPU accesses, both for bursts and between bursts. Page mode is performed by keeping RAS active while reading or writing multiple words within a DRAM page by changing only the column address and toggling CAS with the new column address. The DRAM page size is fixed at 4KB.

4.4.1 DRAM Buffering

Deep buffering is one of the major performance enhancements in the Viper-MAX Chipset. It incorporates deep buffers in the CPU-to-DRAM and PCI-to-DRAM data paths, which enables read prefetching and write posting in the data paths.

- Deep buffering for DRAM performance
  - Six quad-word CPU-to-DRAM write posting
  - 24 double-word PCI-to-DRAM write posting
  - 24 double-word DRAM-to-PCI read prefetch

Table 4-24 lists the registers/bits that are associated with DRAM Buffering.

4.4.1.1 CPU-to-DRAM Deep Buffer

A six quad-word deep FIFO is built in the 82C566 and is controlled by the 82C567 in the CPU-to-DRAM path. These deep buffers are used by the 82C567 to buffer CPU-to-DRAM data when the GUI has ownership of the memory bus. Once the GUI relinquishes the bus, the buffered data is dispatched into the DRAM. This way the system level latencies caused by shared resources are minimized.

4.4.1.2 PCI-to-DRAM Deep Buffer

A 24 double-word buffer has been designed into the PCI-to-DRAM path. During PCI master write bursts, the master posts data into this buffer. Once GUI accesses and CPU-to-DRAM cycles are completed, the posted data will be written back in to the DRAM. This avoids any stalling in the PCI and full bus bandwidth is utilized.

Table 4-24 DRAM Buffering Related Registers/Bits

7	6	5	4	3	2	1	0
PCIDV0 44h			**82C566 Control Register 1**				**Default = 00h**
6DW FIFO for CPU write to PCI: 0 = Disable 1 = Enable	24DW FIFO for PCI read from DRAM: 0 = Disable 1 = Enable	24DW FIFO for PCI write to DRAM: 0 = Disable 1 = Enable	6QW FIFO for CPU write to DRAM: 0 = Disable 1 = Enable	Memory read accesses in the 82C566 if PCIDV0 44h[0] = 1 and 47h[7] = 1: 0 = SDRAM 1 = BEDO	82C566 ping-pong buffer used for PCI master write X-1-1-1: 0 = Disable 1 = Enable	82C566 ping-pong buffer used for PCI master read X-1-1-1: 0 = Disable 1 = Enable	Memory read accesses in the 82C566: 0 = FP Mode 1 = EDO/ SDRAM/ BEDO
PCIDV0 45h			**82C566 Control Register 2**				**Default = 00h**
Reserved: Must be written to 0.				Memory parity generation and checking if PCIDV0 45h[1] = 0: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	Byte merge for CPU write to DRAM: 0 = Disable 1 = Enable	MD bus internal pull-up resistors: 0 = Enable 1 = Disable
PCIDV0 47h			**82C566 Control Register 4**				**Default = 00h**
SDRAM/BEDO memory read accesses in 82C566: 0 = Disable 1 = Enable	CPU-to-PCI FIFO clearing when combination changed: 0 = Do not clear 1 = Clear	PCI-to-DRAM FIFO clearing when combination changed:(1) 0 = Do not clear 1 = Clear	CPU-to-DRAM FIFO clearing when combination changed: 0 = Do not clear 1 = Clear	82C566 register is writable: 0 = Enable 1 = Disable (cnfg-writes blocked within 82C566)	Reserved: Must be written to 0.		

(1) Bit 5 must be set to 1 whenever PCI to DRAM FIFO is turned on in the system.

Table 4-24 DRAM Buffering Related Registers/Bits (cont.)

7	6	5	4	3	2	1	0
SYSCFG 15h			**PCI Cycle Control Register 1**				**Default = 00h**
CPU master to PCI memory slave write IRDY# control: 00 = 3 LCLKs after data 01 = 2 LCLKs after data 10 = 1 LCLK after data 11 = 0 LCLK after data		CPU master to PCI slave write posting, bursting control: 00 = No posting, no bursting 01 = Posting only, no bursting 10 = Posting, with conservative bursting 11 = Posting, with aggressive bursting		Master retry timer: Selects the delay before retry is attempted. 00 = 10 PCICLKs 01 = 18 PCICLKs 10 = 34 PCICLKs 11 = 66 PCICLKs		Reserved: Must be written to 0.	PCI FRAME# generation control: 0 = Conservative mode in CPU pipelined cycle 1 = Aggressive mode
SYSCFG 20h			**DRAM Burst Control Register**				**Default = 00h**
Reserved: Must be written to 1.	DRAM post write during HITM# cycle during PCI master access: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	PCI master parity: 0 = Disable 1 = Enable	DRAM write burst cycle control during PCI master cycles: 00 = Reserved 01 = X-3-3-3 10 = X-2-2-2 11 = X-1-1-1		DRAM read burst cycle control during PCI master cycles: 00 = Reserved 01 = X-3-3-3 10 = X-2-2-2 11 = X-1-1-1	
SYSCFG 2Ah			**PCI-to-DRAM Deep Buffer Size Register**				**Default = 00h**
Reserved: Must be written to 0.	Time-out selection when there is a GUI request during PCI master read cycles: 00 = Always FP mode, grant DRAM bus to GUI ASAP 01 = Select SDRAM or EDC time-out depending on current bank information 1X = Select FP mode, SDRAM, or EDO depending on current bank information		PCI TRDY# wait state control with PCI-DRAM deep buffer: 0 = Zero wait state (X-1-1-1) 1 = One wait state (X-2-2-2)	Write burst with PCI-DRAM deep buffer: 0 = Disable 1 = Enable	Read burst with PCI-DRAM deep buffer: 0 = Disable 1 = Enable	PCI-to-DRAM deep buffer size: 00 = 16 dword 01 = 24 dword 10 = Reserved 11 = Reserved	
SYSCFG 2Bh			**EDO/SDRAM Time-Out Register**				**Default = 00h**
SDRAM time-out count when there is a GUI request: The register value plus 9 is the number of CPU clocks delaying the GUI request to stop the DRAM controller.				EDO time-out count when there is a GUI request: The register value plus 6 is the number of CPU clocks delaying the GUI request to stop the DRAM controller.			
SYSCFG 2Ch			**CPU-to-DRAM Buffer Control Register**				**Default = 00h**
Reserved: Must be written to 0.					BOFF# assertion for DRAM read cycles: 0 = Disable 1 = Enable	Data merge when CPU has ownership of DRAM bus: 0 = Disable 1 = Enable	Buffer write data while buffer is flushing: 0 = Disable 1 = Enable

Table 4-24 DRAM Buffering Related Registers/Bits (cont.)

7	6	5	4	3	2	1	0
SYSCFG 2Eh			**PCI Master - GUI Retry Control Register**				**Default = 00h**
Pin 189 functionality: 0 = DIRTYWE# or RAS5# 1 = SDCKE	Pin 100 functionality: 0 = USBCLK 1 = REFRESH#	Pin 121 functionality and MSGN2S/ MSGS2N bus enabling: 0 = AEN 1 = MSGN2S	Reserved: Must be written to 0.	CPU-to-PCI FIFO control module: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	PCI Master HITM# cycle, if GUI high priority request jumps in before first BRDY#: 0 = Retry all PCI cycles 1 = Retry only PCI master read	PCI master requests retried during GUI cycles: 0 = All PCI master requests are retried 1 = PCI master reads are retried, writes are accepted

4.4.2 EDO Support

The Viper-MAX Chipset provides the capability to use EDO DRAMs in a system. EDO devices are very similar to devices that incorporate fast page mode accesses. However, the use of EDO DRAMs boosts the system performance considerably over conventional fast page mode DRAMs. This boost in performance stems from the different way in which the memory bus is controlled. In conventional fast page mode DRAMs, the memory bus is turned on by the falling edge of CAS# and is turned off (High-Z) when CAS# returns to high. The fast page mode DRAMs only guarantee data to be valid for 5ns (which is typically too brief for systems operating at full speed). To compensate, CAS# must be held low for an extended period until the data can be read from the bus. In contrast, EDO devices turn on the memory bus when CAS# falls low but do not turn off the bus when CAS# returns to high. Instead, the data remains valid until the next falling edge of CAS#. Because the data remains valid until the falling edge of the next CAS#, the transfer of memory data to the latch in the memory controller can be overlapped with the next column precharge. This extra time that the data remains valid resolves the system problem described above. The extended data time allows the system to run with a minimum CAS# low time. This increases the system performance by decreasing the page access cycle time. Control of the memory bus can be obtained by the OE# and CAS# signals.

The Viper-MAX Chipset allows the user to populate the system with up to six banks of EDO DRAMs. Individual bits in SYSCFG 1Ch need to be set to a "1" for each bank that uses EDO DRAMs. Timing can be programmed to achieve a 6-2-2-2 read cycle at 50MHz when EDO DRAMs are used with the Viper-MAX Chipset.

The Viper-MAX Chipset also provides the flexibility to mix and match EDO DRAM SIMMs and conventional fast page mode DRAMs among the different banks. As an example, EDO DRAMs in Banks 0 and 2 could be used and fast page mode DRAMs in the other banks. There are no restrictions in terms of which bank(s) can contain EDO SIMMs and which can contain fast page mode DRAMs. However, care must be taken to ensure that the SIMMs that make-up the 64-bit data path to DRAM are all EDO DRAMs or all fast page mode DRAMs.

In a system, all banks that have been populated with EDO DRAMs will have the same DRAM timings. Likewise, all banks that are populated by fast page mode DRAMs will have the same DRAM timings.

Table 4-25 EDO Associated Register Bits

7	6	5	4	3	2	1	0
SYSCFG 1Ch			**EDO DRAM Control Register**				**Default = 00h**
EDO DRAM usage: Each bit is set to a 1 if the user is using EDO DRAMs in each of the available six banks. Bit 2 corresponds to Bank 0 and bit 7 corresponds to Bank 5, yielding a total of six banks that the user can populate. 0 = Standard page mode DRAM 1 = EDO DRAM						Viper-MAX Chipset operating at a frequency of 50MHz:(1) 0 = No 1 = Yes	CAS pulse width during DRAM accesses: 0 = CAS pulse width determined by SYSCFG 01h[3] 1 = CAS pulse width is one CPUCLK(2)

(1) Bit 1 is set by the user when the chipset is operating at a frequency of 50MHz. The setting of this bit could potentially improve DRAM access times even if the user is not using EDO DRAMs.

(2) The width of the pulse is one CPU clock if the Viper-MAX Chipset is operating at 50MHz (selected by the setting of bit 1) if it is interfaced to EDO DRAMs (selected by bits [7:2]).

SYSCFG 1Dh			**Miscellaneous Control Register 3**			**Default = 00h**
Reserved: Must be written to 0.	DWE# timing selection:(1) 0 = Normal 1 = Removed one CLK earlier	DRAM read leadoff cycle: 0 = Normal 1 = Reduced by one CLK	DMA accesses from system memory: 0 = Enable 1 = Disable	Reserved: Must be written to 1 if 08h[4] = 1 (if parity is enabled).	Accesses to B0000h-BFFFFh during SMM mode: 0 = Accesses go to main memory 1 = Accesses go to PCI bus	Accesses to A0000h-AFFFFh during SMM mode: 0 = Accesses go to main memory 1 = Accesses go to PCI bus

(1) When using a buffered DWE# solution and the DRAM load is substantial, bit 5 may have to be set if the system begins to malfunction.

Table 4-25 EDO Associated Register Bits (cont.)

7	6	5	4	3	2	1	0
SYSCFG 1Fh			**EDO Timing Control Register**				**Default = 00h**
0 = Normal 1 = Generate conflict during EDO detection (bit 6 set) if necessary	0 = Normal (fast page mode) 1 = Detect EDO	NA# generation: 0 = Aggressive (enabled with EDO or 50MHz operation and X-2-2-2 timing selection 1 = Normal	DRAM read cycle leadoff reduced by 1 clock to support 5-2-2-2 at 50MHz: 0 = No (Normal) 1 = Yes	Reserved: Must be written to 0.	Chip selects and write enables for async SRAM: 0 = 8 CS# and 1 WE# 1 = 1 CS# and 8 WE# (CS# is OCAWE# in this mode)(1) (In sync SRAM mode, ADSC# and ADV# are swapped.)	Block AHOLD during Hidden Refresh: 0 = Enable (Normal) 1 = Disable	0D0000-0DFFFFh is cacheable in L1 and L2:(2) 0 = No 1 = Yes

(1) This is only good for single bank cache. ECAWE# pin will become CS#.

(2) Before turning on bit 0, 0D0000-0DFFFFh needs to be readable/writable and shadowed. When cached into L1, it will be in write-back mode if SYSCFG 08h[1] is on. There is no write protection in this region if bit 0 is set.

SYSCFG 26h			**UMA Control Register**		**Default = 00h**
ISA master to DRAM cycle CAS width: 0 = Controlled by ISA R/W command pulse width 1 = 2 LCLKs	ISA SA address latch: 0 = SA latch is always transparent (pass-through) 1 = SA latch is on for retry only. (When the first CPU/ISA cycle is retried, the SA address will be latched.)	GUI memory size: 00 = 1MB 01 = 2MB 10 = 3MB 11= 4MB For 0.5MB size, set these bits to 00 and SYSCFG 25h[2] = 1.	5-2-2-2 EDO DRAM read timing at 66MHz: 0 = Disable 1 = Enable	00 = Normal 01 = For low priority GUI request, 82C567 will wait for two more CLKs 11 = GUI is always at high priority	UMA support: 0 = Disable 1 = Enable

SYSCFG 2Bh	**EDO/SDRAM Time-Out Register**	**Default = 00h**
SDRAM time-out count when there is a GUI request: The register value plus 9 is the number of CPU clocks delaying the GUI request to stop the DRAM controller.	EDO time-out count when there is a GUI request: The register value plus 6 is the number of CPU clocks delaying the GUI request to stop the DRAM controller.	

SYSCFG 2Dh		**Bank-wise EDO Timing Selection Register**	**Default = 00h**
Split buffer concurrency: 0 = Disable 1 = Enable	Predictive reading: 0 = Disable (normal) 1 = Enable	Bank-wise selection for 5-X-X-X at 66MHz or 4-X-X-X at 50MHz EDO DRAM read cycle: 0 = Default setting 1 = 5-X-X-X/4-X-X-X enabled	

Table 4-25 EDO Associated Register Bits (cont.)

7	6	5	4	3	2	1	0
PCIDV0 44h			**82C566 Control Register 1**				**Default = 00h**
6DW FIFO for CPU write to PCI: 0 = Disable 1 = Enable	24DW FIFO for PCI read from DRAM: 0 = Disable 1 = Enable	24DW FIFO for PCI write to DRAM: 0 = Disable 1 = Enable	6QW FIFO for CPU write to DRAM: 0 = Disable 1 = Enable	Memory read accesses in the 82C566 if PCIDV0 44h[0] = 1 and 47h[7] = 1: 0 = SDRAM 1 = BEDO	82C566 ping-pong buffer used for PCI master write X-1-1-1: 0 = Disable 1 = Enable	82C566 ping-pong buffer used for PCI master read X-1-1-1: 0 = Disable 1 = Enable	Memory read accesses in the 82C566: 0 = FP Mode 1 = EDO/ SDRAM/ BEDO

4.4.3 Burst EDO Support

The Viper-MAX Chipset provides supports the next generation of EDO DRAM, Burst EDO (BEDO). To read data out of an EDO DRAM, the chipset needs to provide an address for every location accessed in a burst of consecutive location accesses. With BEDO, chipset needs to provide only the starting address of the burst and the BEDO chips will increment it internally. In other words, BEDO DRAMs contain a pipeline stage with a 2-bit counter. This way, the BEDO can supply data on every clock achieving an X-1-1-1 burst on the bus.

The Viper-MAX Chipset can support four banks of BEDO in the system. The 82C567 supplies the row address with RAS# and the first column address of the burst transfer. The BEDO latches this address at the following edge of the first CAS# and starts the burst read cycle. On each subsequent CAS# assertion, data is output on the MD bus and the address is automatically incremented internally.

Since CAS# will have less than 1/2 CLK pulse width to support X-1-1-1 transfers, the memory routing and loading on the system motherboard must be carefully designed to achieve this timing.

BEDO allows a system design to have reasonable bandwidth at a relatively low cost compared to other fast memory technologies. BEDO also supports both interleaved and linear bursts which is programmable by Viper-MAX chipset depending on the CPU type.

The Viper-MAX assumes the following characteristics of BEDO:

- Total number of banks supported: Four
- Read latency from first CAS#: Two
- Burst length: Two
- Burst sequence: Interleaved (Default)
- Linear
- Burst advance control: CAS#
- Supply voltage: 3.3V (5.0V tolerant)

4.4.4 SDRAM Support

The Viper-MAX Chipset provides the capability to use SDRAM DIMM modules in a system design. Up to four banks of SDRAM banks are supported. The SDRAM devices accept all its input command signals at the rising edge of system clock. The clocking allows data pipelining within the SDRAM device and data output in a continuous stream on every clock. Because of this, an SDRAM-based design boosts the memory performance considerably over FP mode/EDO DRAMs.

4.4.4.1 SDRAM Commands

With SDRAM, external control signals are latched with the rising edge of clock pulses and specific high and low combinations are recognized as commands. The SDCS# (SDCS[3:0]#), SDRAS#, SDCAS#, SDWE#, and MA address lines define the inputs commands which become active on the positive transition of SDCKE.

The Viper-MAX Chipset issues the following SDRAM COMMANDs:

- BANK ACTIVE
- BANK PRECHARGE
- PRECHAGE ALL
- WRITE
- READ
- MODE REGISTER SET
- AUTO REFRESH

The Viper-MAX Chipset does not support the following commands:

- READ WITH AUTOPRECHAGE
- WRITE WITH AUTOPRECHAGE
- CLOCK SUSPEND MODE
- SELF REFRESH

SDRAM READ and WRITE Commands

To avoid bus contention on the memory bus, the Viper-MAX ensures a dead cycle between write data and read data commands. During read cycles, SDDQM[7:0]# are used as the standard output enable function. During write cycles, these outputs act as a mask for input data buffer of the SDRAM.

4.4.4.2 SDRAM Initialization

The Viper-MAX Chipset allows SDRAM devices to stabilize and will not toggle any input command signal for 100ms. The first command will be the PRECHARGE ALL banks. After precharging, the mode register will be programmed based on the register setting. The following fields will be affected by this programming:

- Burst length: 1, 2, 4, 8 or full page
- Wrap Type: Sequential or interleaved
- CAS# latency: 1, 2 or 3

SDRAM refreshing is done in the similar fashion of FP mode and EDO DRAMs.

Table 4-26 shows the register bits associated for configuring SDRAM in a Viper-MAX based system.

4.4.4.3 Unbuffered DIMMs

The Viper-MAX Chipset supports up to four banks of unbuffered SDRAM DIMMs connectors on a system motherboard. The maximum clock frequency is 66MHz. Clock should be connected to each of the DIMM connector.

MA[10:0] are connected as the row address of the DIMM. MA11 is used the bank select pin. SDCS# pins are used as the bank select outputs.

Table 4-26 SDRAM Configuration Registers

7	6	5	4	3	2	1	0
SYSCFG 28h			**SDRAM Burst and Latency Control Register**				**Default = 00h**
Reserved: Must be written to 0.	SDRAM CAS# latency: 000 = Reserved; 001 = 1; 010 = 2; 011 = 3; 100 = Reserved; 101 = Reserved; 110 = Reserved; 111 = Reserved			Reserved: Must be written to 0.	Burst length control: 000 = 1; 001 = 2; 010 = 4; 011 = 8; 100 = Reserved; 101 = Reserved; 110 = Reserved; 111 = Full page		
SYSCFG 29h			**SDRAM Selection Register**				**Default = 00h**
When set to 1, at least one CLK will be forced between the current command and next cycle.(1)	Controls latency between bank active and command at 50MHz: 0 = 2 CLK latency; 1 = 1 CLK latency	SDRAM pre-charge control: 00 = 2 CLKs; 10 = 3 CLKs; 01 = 4 CLKs; 11 = Reserved		SDRAM enable on each bank: 0 = Disable; 1 = Enable			

(1) Otherwise, if next cycle is a page-and/or bank-miss, it could start right after the current command has been issued.

4.4.5 DRAM Configuration/Programming Parameters

The Viper-MAX DRAM controller provides a flexible full decode mode (SYSCFG 13h[7] = 1) for configuring the size and arrangement of each DRAM bank. There are various parameters that can be obtained in the DRAM state machine - number of banks, DRAM configurations, timing parameters and drive strengths.

4.4.5.1 Number of DRAM banks

The Viper-MAX Chipset supports up to six banks of DRAM. The default condition is four banks of DRAM supporting up to 512Mbytes of system memory. MA11 is multiplexed with RAS4# and DIRYTWE# is multiplexed with RAS5#.

If DIRYTWE# is used as RAS5#, the following issues arise.

- If a separate tag/dirty RAM implementation is used, the L2 cache write-back functionality is lost.
- If a combined tag/dirty RAM implementation is used, the L2 cache can still be used in the write-back mode.
- If RAS5# is used and MA11 is not used as RAS4#, the maximum amount of memory supported is 512Mbytes, with not more than three banks populated with 128Mbytes.

If both RAS4# and RAS5# are being used (i.e., six banks of DRAM), then the maximum amount of memory supported is 192Mbytes.

If MA11 is used as RAS4#, then the maximum memory size supported decreases to 192Mbytes.

RAS4# and RAS5# are selected through SYSCFG 19h[7] (RAS5#), and 19h[3] (RAS4#).

Table 4-27 Full Memory Decode Mode and RAS Selection Bits

7	6	5	4	3	2	1	0
SYSCFG 13h			**Memory Decode Control Register 1**				**Default = 00h**
Memory decode select: This bit must be set to 1 for full decode (maximum flexibility in choosing different DRAM configurations)	Full decode for logical Bank 1 (RAS1#) if SYSCFG 13h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36			SMRAM: 0 = Disable 1 = Enable	Full decode for logical Bank 0 (RAS0#) if SYSCFG 13h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36		
SYSCFG 19h			**Memory Decode Control Register 3**				**Default = 00h**
Pin 189 functionality:(1) 0 = DIRTYWE# 1 = RAS5#	Full decode for logical bank 5 (RAS5#) if SYSCFG 13h[7] is set and 19h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36			Pin 71 functionality:(2) 0 = MA11 1 = RAS4#	Full decode for logical bank 4 (RAS4#) if SYSCFG 13h[7] is set and 19h[3] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = Undefined 111 = Undefined		

(1) If six DRAM banks have been chosen, the DIRTYWE# line will become RAS5# if bit 7 = 1. If six banks of DRAM are chosen, then a combined Dirty/Tag SRAM solution must be implemented or else it will not have a Dirty RAM.

(2) If five DRAM banks have been chosen, the MA11 line will become RAS4# if bit 3 = 1. If bit 3 is set to 1, none of the DRAM banks will support the 8Mx36 or 16Mx36 options.

4.4.5.2 DRAM Size and Type

The DRAM configuration is selected through groups of three bits in SYSCFG 13h, 14h, and 19h, There is no required ordering for these selections: Any desired bank can be occupied or not. For example, if in the course of testing system DRAM the BIOS POST code finds Bank 3 (RAS3#) defective, it should simply set that bank to "disabled." The DRAM controller will automatically map around it and provide a contiguous memory map to the system.

Table 4-28 DRAM Configuration Related Register Bits

7	6	5	4	3	2	1	0
SYSCFG 13h			**Memory Decode Control Register 1**				**Default = 00h**
Memory decode select: This bit must be set to 1 for full decode (maximum flexibility in choosing different DRAM configurations)	Full decode for logical Bank 1 (RAS1#) if SYSCFG 13h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36			SMRAM: 0 = Disable 1 = Enable	Full decode for logical Bank 0 (RAS0#) if SYSCFG 13h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36		
SYSCFG 14h			**Memory Decode Control Register 2**				**Default = 00h**
82C566 mode: 0 = Normal mode 1 = Clocked mode (Must = 1 for EDO timing)	Full decode for logical Bank 3 (RAS3#) if SYSCFG 13h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36			SMRAM control: Inactive SMIACT#: 0 = Disable SMRAM 1 = Enable SMRAM(1) Active SMIACT#: 0 = Enable SMRAM for both Code and Data(1) 1 = Enable SMRAM for Code only (1)	Full decode for logical Bank 2 (RAS2#) if SYSCFG 13h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36		
(1) If SYSCFG 13h[3] is set.							
SYSCFG 19h			**Memory Decode Control Register 3**				**Default = 00h**
Pin 189 functionality:(1) 0 = DIRTYWE# 1 = RAS5#	Full decode for logical bank 5 (RAS5#) if SYSCFG 13h[7] is set and 19h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36			Pin 71 functionality:(2) 0 = MA11 1 = RAS4#	Full decode for logical bank 4 (RAS4#) if SYSCFG 13h[7] is set and 19h[3] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = Undefined 111 = Undefined		

(1) If six DRAM banks have been chosen, the DIRTYWE# line will become RAS5# if bit 7 = 1. If six banks of DRAM are chosen, then a combined Dirty/Tag SRAM solution must be implemented or else it will not have a Dirty RAM.

(2) If five DRAM banks have been chosen, the MA11 line will become RAS4# if bit 3 = 1. If bit 3 is set to 1, none of the DRAM banks will support the 8Mx36 or 16Mx36 options.

4.4.5.3 DRAM Address Muxing

Table 4-29 shows the DRAM address (MA) muxing. Note that the column address is the same for all configurations since this is the speed path. A3 and A4 must go through an internal burst counter, for generation of the MA address to the DRAMs. The table shows MA line to address bit mapping for each DRAM size configuration.

4.4.5.4 Timing Parameters

The timing constraints to achieve optimum performance at 66MHz are met without making the system design overly critical. Timing variations that are required for different system speeds are handled by a selection of timing modes that vary the wait states used. Table 4-30 summarizes these timing modes.

Table 4-29 DRAM Row/Column MA to Address Bit Map

	256KB		512KB		1MB		2MB		4MB		8MB		16MB	
Addr.	Col	Row	Col	Row	Col	Row	Col	Row	Col	Row	Col	Row	Col	Row
MA0	A3	A12	A3	A12	A3	A12	A3	A12	A3	A12	A3	A12	A3	A12
MA1	A4	A13	A4	A13	A4	A13	A4	A13	A4	A13	A4	A13	A4	A13
MA2	A5	A14	A5	A14	A5	A14	A5	A14	A5	A14	A5	A14	A5	A14
MA3	A6	A15	A6	A15	A6	A15	A6	A15	A6	A15	A6	A15	A6	A15
MA4	A7	A16	A7	A16	A7	A16	A7	A16	A7	A16	A7	A16	A7	A16
MA5	A8	A17	A8	A17	A8	A17	A8	A17	A8	A17	A8	A17	A8	A17
MA6	A9	A18	A9	A18	A9	A18	A9	A18	A9	A18	A9	A18	A9	A18
MA7	A10	A19	A10	A19	A10	A19	A10	A19	A10	A19	A10	A19	A10	A19
MA8	A11	A20	A11	A20	A11	A20	A11	A20	A11	A20	A11	A20	A11	A20
MA9	-	-	-	A21	A22	A21	A22	A21	A22	A21	A22	A21	A22	A21
MA10	-	-	-	-	-	-	-	A23	A24	A23	A24	A23	A24	A23
MA11	-	-	-	-	-	-	-	-	-	-	-	A25	A26	A25

Table 4-30 DRAM Programmable Control

DRAM Timing Being Controlled	Variation in CLK
RAS address hold time	1 to 2
CAS pulse width for reads	1 to 2
CAS pulse width for writes	2 to 3
Address setup time to CAS for write page hit	1 to 2
CAS precharge time	1 to 2
RAS precharge time	3 to 6
RAS pulse width for refresh	4 to 7

4.4.5.5 Drive Strengths

Programmable current drive for the MA[11:0], RAS[5:0]# and the DWE# lines is provided. If SYSCFG 18h[4] = 0, the current drive on these lines is 4mA. In this case, two F244 buffers will be required to drive each pair of DRAM banks. If SYSCFG 18h[4] = 1, then the current drive on these lines is increased to 16mA and it is possible to drive the first pair of DRAM banks without any buffers. Refer to Table 4-31.

4.4.6 DRAM Cycles

The fastest possible burst read is 6-2-2-2 which means the first quad-word is received in six clocks and the next three quad-words are received after two clocks each. For a cache based system, it would mean the bursting to the cache and CPU for read miss cycles or write miss cycles. Table 4-32 summarizes the DRAM cycle lengths and the following subsections describe the read/write cycle operations.

Table 4-31 Drive Strength Control Bit

7	6	5	4	3	2	1	0
SYSCFG 18h			**Tristate Control Register**				**Default = 00h**
Reserved: Must be written to 0.	Drive strength on RAS lines: 0 = 4mA 1 = 16mA	CAS lines voltage selection: 0 = 5.0V 1 = 3.3V	Drive strength on memory address lines and write enable line: 0 = 4mA 1 = 16mA	Tristate CPU interface during Suspend and during CPU power-off: 0 = Disable 1 = Enable	Tristate PCI interface during Suspend and during PCI power-off: 0 = Disable 1 = Enable	Tristate cache interface during Suspend and during cache power-off: 0 = Disable 1 = Enable	Pull-up/-down resistors active during Suspend and power-off: 0 = Disable 1 = Enable

Table 4-32 DRAM Cycle Lengths

CPU Bus Speed	Page Hit Leadoff	Page Miss RAS High Leadoff	Page Miss RAS Active Leadoff	CPU Pipeline Reduces Lead-off Cycle by:	Burst Cycle Length	Continued Burst if Pipelined
Read Burst Cycle						
50MHz FP-DRAM	5 cycles	8 cycles	8+precharge	1 clocks	-2-2-2	-2-2-2
50MHz EDO-DRAM	5 cycles	8 cycles	8+precharge	1 clocks	-2-2-2	-2-2-2
60/66MHz FP-DRAM	5 cycles	8 cycles	8+precharge	1 clocks	-3-3-3	-3-3-3
60/66MHz EDO-DRAM	5 cycles	8 cycles	8+precharge	1 clocks	-2-2-2	-2-2-2
Write Burst Cycle						
50MHz FP/EDO-DRAM		4-7-3-3	4-(7+pre)-3-3	1 clock	-3-3-3	-3-3-3
60/66MHz FP/EDO-DRAM		4-7-3-3	4-(7+pre)-3-3	1 clock	-3-3-3	-3-3-3
50/60/66MHz FP/EDO-DRAM Single Write	3	4	4			3

Single writes can be pipelined. Single reads are not pipelined.

4.4.6.1 DRAM Read Cycle

The DRAM read cycle begins with the DRAM controller detecting a page hit or a page miss cycle at the end of the first T2. Based on the status of the current open page and the active RASx#, a page hit, a page miss with RAS inactive, or a page miss with RAS active cycle is executed.

Page Miss with RASx# High Cycle: The row address is generated from the CPU address bus. (Refer back to Table 4-29 for the row/column address mux map.) After RASx# goes active, the row address is changed on the next clock edge (programmable to be two CLKs) to the column address. The CASx# will be active two CLKs after the column address is generated. (Refer to Figure 4-23.)

Page Miss with RASx# Low Cycle: RAS is first precharged for the programmed number of CLKs and then driven active, after which it will be the same as a page miss with RASx# high cycle.

Page Hit Cycle: The 82C567 generates the column address from the CPU address bus and CASx# is driven active for two clocks. Data flow from the CPU data bus to the memory data bus and vice versa is controlled by the DBCOE#, MMDOE#, MDOE#, and HDOE# signals from the 82C567 to the 82C566. Data from the DRAM is latched by the 82C566 on the rising edge of each DLE (for CPU reads from DRAM, the DLE[1:0]# signals are identical to the CAS signal). The latched data is valid on the CPU data bus until the next rising edge of CASx#. During this time, the next read is started, CASx# signals are precharged for one or two clocks (programmable via SYSCFG 02h[1]), and the next data from the DRAM is accessed and latched. The 82C566 latches the data from the DRAM and holds the data for the CPU while the DRAM controller begins the read for the next word in the burst cycle. The burst read from the DRAM is in effect pipelined into the CPU data bus by the Viper-MAX Chipset. This scheme reduces the constraints on the board layout so that routing for the CPU data bus, MD data bus, and CASx# signal lines are less critical and performance can be maintained.

Page Hit Cycle (Extended): Wait states can be added if slower DRAMs are used. In this mode, data from the DRAM is latched by the 82C566 at the end of each CAS cycle similar to the default mode. The only difference between the two modes is that the CAS low time on reads is increased by one T-state. This eases up on the page mode cycle time and CAS access time parameters.

The DRAM read cycle uses a CAS signal that is active for multiples of T-state boundaries rather than half T-state boundaries. This allows additional address decode setup time and MA bus setup time at the start of the cycle, making the fastest burst cycle 7-2-2-2.

4.4.6.2 DRAM Write Cycle

Posted writes to DRAM improves the write cycle timing relative to the CPU and allows the Viper-MAX Chipset to perform an independent write burst cycle to DRAM without holding the CPU. The Viper-MAX Chipset maintains a one quad-word deep data buffer for DRAM writes so that the CPU write cycle is completed without waiting for the external DRAM cycle. For a burst write cycle, the leadoff cycle time is reduced to four clocks even if the cycle is a non-page hit cycle. For a page hit cycle, the burst write can be completed in 4-3-3-3 with posted write enabled. The posted write buffer in the 82C566 is controlled by the DLE[1:0]# signals from the 82C567. Effectively, the rising edge of these signals will latch the high 32-bit and the low 32-bit new data respectively, from the CPU bus to the posted write buffer.

Single level posted write cycles are employed to achieve a 4-3-3-3 burst at 66MHz. The data from the CPU is latched in the write buffer of the 82C566 until CAS goes active one T-state after the first T2 (on a page hit). This provides a fast write mechanism and two wait state writes are maintained for the leadoff cycle within a page (even at 66MHz). The CAS pulse width can be extended by one more T-state to ease the timing constraints on the CAS pulse width requirement for speeds above 66MHz.

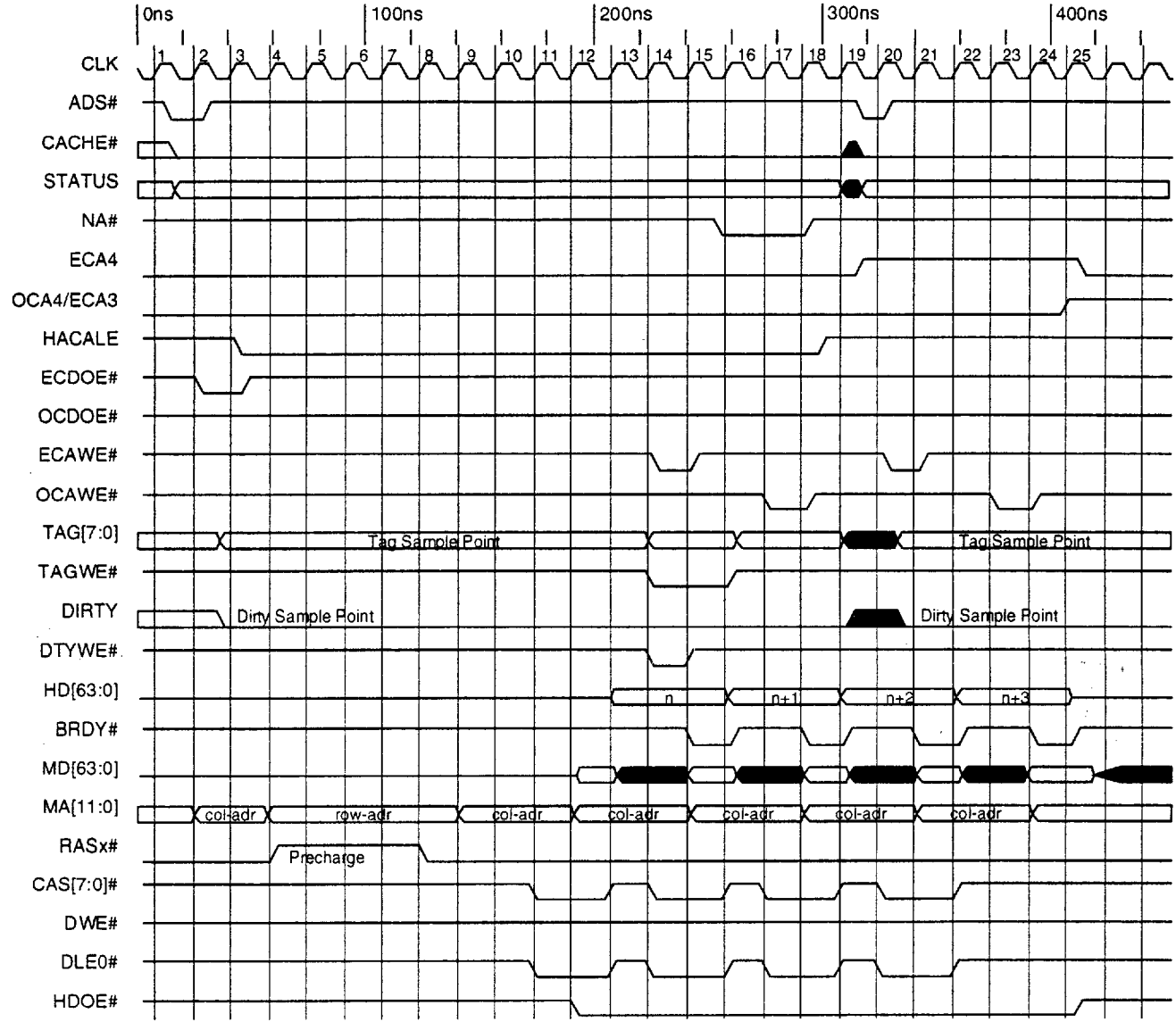
Table 4-33 DRAM Operation Programming Bits

7	6	5	4	3	2	1	0
SYSCFG 01h			**DRAM Control Register 1**				**Default = 00h**
Row address hold after RAS# active: 0 = 2 CLKs 1 = 1 CLK	RAS# active/ inactive on entering master mode: 0 = Normal page mode, RAS# active when starting master cycle 1 = RAS# inactive when starting a master cycle	RAS pulse width used during refresh: 00 = 7 CLKs 01 = 6 CLKs 10 = 5 CLKs 11 = 4 CLKs		CAS pulse width during reads: 0 = 3 CLKs 1 = 2 CLKs	CAS pulse width during writes: 0 = 3 CLKs 1 = 2 CLKs	RAS precharge time: 00 = 6 CLKs 01 = 5 CLKs 10 = 4 CLKs 11 = 3 CLKs	

Table 4-33 DRAM Operation Programming Bits (cont.)

7	6	5	4	3	2	1	0
SYSCFG 02h			**Cache Control Register 1**				**Default = 00h**
L2 cache size selection: **If SYSCFG 0Fh[0] = 0** 00 = Reserved 01 = Reserved 10 = 256K 11 = 512K **If SYSCFG 0Fh[0] = 1** 00 = 1MB 01 = 2MB 10 = Reserved 11 = Reserved		L2 cache write policy: 00 = L2 cache write-through 01 = Adaptive Write-back Mode 1 10 = Adaptive Write-back Mode 2 11 = L2 cache write-back		L2 cache operating mode select: 00 = Disable 01 = Test Mode 1; External Tag Write (Tag data write-through SYSCFG 07h) 10 = Test Mode 2; External Tag Read (Tag data read from SYSCFG 07h) 11 = Enable L2 cache		DRAM posted write: 0 = Disable 1 = Enable	CAS precharge time: 0 = 2 CLKs 1 = 1 CLK
SYSCFG 0Ch			**DRAM Hole Higher Address**				**Default = 00h**
Reserved: Must be written to 0.	Fast BRDY# generation for DRAM write page hits. BRDY# for DRAM writes generated on: 0 = 4th CLK 1 = 3rd CLK	HACALE cycle: 0 = Normal timing 1 = HACALE one-half a clock cycle early	Cache WE# pulse width: 0 = Normal (i.e., ~15ns) 1 = Wider (i.e., ~17.5ns)	DRAM Hole B starting address: These bits are used in conjunction with the bits in SYSCFG 0Bh to specify the starting address of DRAM Hole B. These bits, BST[9:8], map onto HA[28:27].		DRAM Hole A starting address: These bits are used in conjunction with the bits in SYSCFG 0Ah to specify the starting address of DRAM Hole A. These bits, AST[9:8], map onto HA[28:27].	
SYSCFG 24h			**Asymmetric DRAM Configuration Register**				**Default = 00h**
Logical Bank 3 DRAM type: 00 = Sym DRAM 01 = Asym DRAM - x8 type 10 = Asym DRAM - x9 type 11= Asym DRAM - x10 type		Logical Bank 2 DRAM type: 00 = Sym DRAM 01 = Asym DRAM - x8 type 10 = Asym DRAM - x9 type 11= Asym DRAM - x10 type		Logical Bank 1 DRAM type: 00 = Sym DRAM 01 = Asym DRAM - x8 type 10 = Asym DRAM - x9 type 11= Asym DRAM - x10 type		Logical Bank 0 DRAM type: 00 = Sym DRAM 01 = Asym DRAM - x8 type 10 = Asym DRAM - x9 type 11= Asym DRAM - x10 type	
SYSCFG 2Fh			**CAS Address Setup Time Control Register**				**Default = 00h**
Column address to CAS delay for page miss cycles: 0 = Default 1 = 1 CLK	Reserved: Must be written to 0.		Generation of NA# during CPU accesses to non-shared DRAM bank when GUI has the memory bus: 0 = Enable 1 = Disable	Reserved: Must be written to 0.			

Figure 4-23 DRAM Read Page Miss with RAS Active Read Cycle



Note: For RAS inactive cycle, clocks 4 through 7 will not exist.

Figure 4-24 DRAM Page Miss with RAS Active Write Cycle

Note: For RAS inactive cycle, clocks 4 through 7 will not exist.

4.4.6.3 DRAM Parity Generation/Detection Logic

During local DRAM write cycles, the 82C566 generates a parity bit for each byte written by the processor. Parity bits are stored in the local DRAM along with each data byte. During a DRAM read, the parity bit is checked for each data byte. If the logic detects incorrect parity, the 82C566 will assert the MPERR# signal to the 82C568. If SYSCFG 08h[4] = 1 (i.e., parity has been enabled), then the 82C567 keeps the PEN# signal to the 82C568 asserted. When the 82C568 senses that MPERR# has been asserted by the 82C566 and if PEN# is also asserted, then it will assert a NMI interrupt to the CPU.

Table 4-34 DRAM Parity Associated Register Bit

7	6	5	4	3	2	1	0
SYSCFG 08h			**CPU Cache Control Register**				**Default = 00h**
L2 cache single/double bank select: 0 = Double bank (If async SRAM, then the banks are interleaved. If sync SRAM, they are not interleaved.) 1 = Single bank (non-interleaved)	Snoop filtering for bus masters:(1) 0 = Disable 1 = Enable	CPU HITM# pin sample timing: 0 = Delay 1 CLK (HITM# sampled on 3rd rising edge of LCLK after EADS# assertion) 1 = No delay (HITM# sampled on 2nd rising edge of LCLK after EADS# assertion)	Parity checking: 0 = Disable 1 = Enable	Tag/Dirty RAM implementation: 0 = Tag and Dirty are on separate chip (i.e., a separate x1 or x8 SRAM for the Dirty RAM) 1 = Tag and Dirty are on the same chip (i.e., could be either a x9 or x8 Tag/Dirty RAM)	CPU address pipelining: 0 = Disable 1 = Enable	L1 cache write-back and write-through control: 0 = Write-through only 1 = Write-back enabled	BIOS area cacheability in L1 cache: Determines if system BIOS area E0000h-FFFFFh (if SYSCFG 04h[2] = 1) or F0000h-FFFFFh (if SYSCFG 04h[2] = 0), and video BIOS area C0000h-C7FFFh is cacheable in L1 or not. 0 = Cacheable 1 = Non-Cacheable

(1) For a master request if the subsequent read/write is within the same cache line, CPU 'Inquire' cycles are not done until there is a cache line miss (i.e., line comparator not activated for accesses within the same cache line).

4.4.6.4 DRAM Refresh Logic

The Viper-MAX Chipset supports the following types of refresh schemes:

- Normal refresh
- Hidden refresh
- Self refresh

During normal refresh, the CPU bus is put on HOLD and the DRAM bus is refreshed. This is the default condition at power-up.

In hidden refresh, once the REFRESH# input is received from the 82C568, the DRAM will be refreshed in the background while the CPU is accessing the internal cache. Hidden refresh is performed independently of the CPU and does not suffer from the performance restriction of losing processor bandwidth by forcing the CPU into its hold state. Since hidden refresh delivers higher system performance, it is recommended over normal refresh as long as the CPU does not try to access local memory or the ISA bus during a hidden refresh cycle, refresh will be transparent to the CPU. The CPU can continue to execute from its internal and secondary caches as well as execute internal instructions during hidden refresh without any loss in performance due to refresh arbitration. If a local memory or ISA bus access is required during a hidden refresh cycle, wait states will be added to the CPU cycle until the resource becomes available. Hidden refresh also separates refreshing of the ISA bus and local DRAM.

In self refresh mode, the external REFRESH# input from the 82C568 is not used. The 82C567 generates an internal refresh input from the system frequency and does a refresh in the background when the DRAM bus is available. Table 4-35 shows the refresh logic associated register bits.

The DRAM controller arbitrates between CPU DRAM accesses and DRAM refresh cycles, while the ISA bus controller arbitrates between CPU accesses to the ISA bus, DMA and ISA refresh. The ISA bus controller (the 82C568) asserts the RFSH# and MEMR# commands and outputs the refresh address during ISA bus refresh cycles.

The 82C567 implements refresh cycles to the local DRAM using CAS-before-RAS timing. The CAS-before-RAS refresh uses less power than RAS-only refresh which is important when dealing with large memory arrays. CAS-before-RAS refresh is used for both normal and hidden refresh to DRAM memory.

The periodic refresh request signal output, from the 82C568 that occurs every 15µs, originates from the counter/timer of the integrated 82C206. Requests for refresh cycles are generated by two sources: the counter/timer of the integrated 82C206 or 16-bit ISA masters that activate refresh when they have bus ownership. These ISA masters must supply refresh cycles because the refresh controller cannot preempt the bus master to perform the necessary refresh cycles. 16-bit ISA masters that hold the bus longer than 15µs must supply refresh cycles.

Table 4-35 Refresh Logic Register Bits

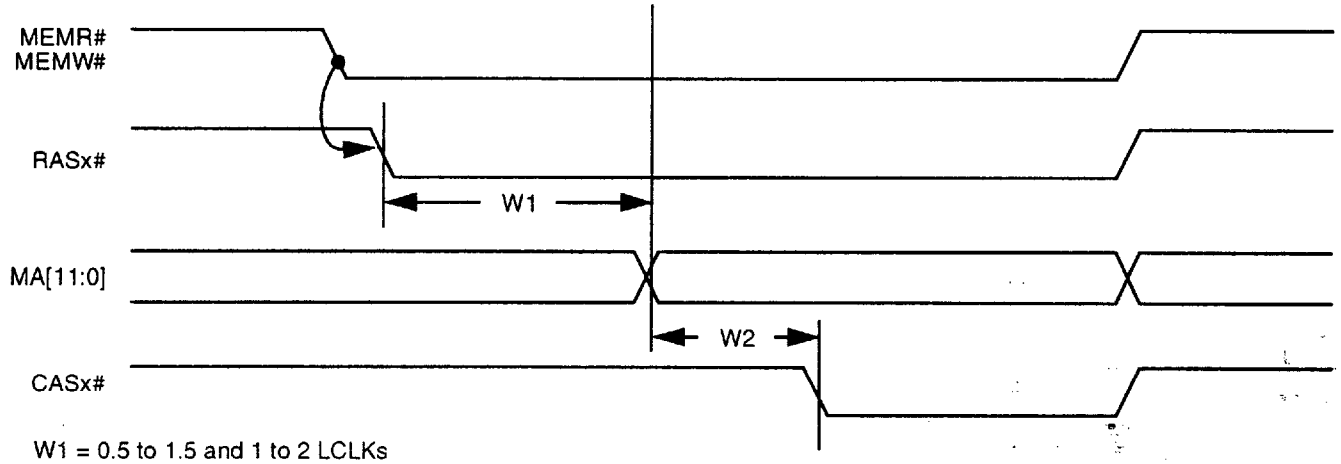
7	6	5	4	3	2	1	0
SYSCFG 12h			**Refresh Control Register**				**Default = 00h**
REFRESH# pulse source: 0 = From 82C568 or ISA master is source of the REFRESH# input 1 = From 32kHz clock	Reserved: Must be written to 0.	Suspend mode refresh: 00 = From CLK state machine 01 = Self-refresh based on 32KHz only 10 = Normal refresh based on 32KHz only 11 = Undefined		Slow refresh: Refresh on: 00 = Every REFRESH#/32KHz falling edge 01 = Alternate REFRESH#/32KHz falling edge 10 = One in four REFRESH#/32KHz falling edge 11 = Every REFRESH#/32KHz toggle		LA[23:17] enable from 8Fh during refresh: 0 = Disable 1 = Enable	Reserved: Must be written to 0.
SYSCFG 27h			**Self Refresh Timing Register**				**Default = 00h**
Reserved: Must be written to 0.			Generate AHOLD at 2nd T2 on CPU single write hit not dirty cycle: 0 = Disable 1 = Enable	Fast NA# with L2 cache: 0 = Disable 1 = Enable	Self refresh: 000 = Disable, use external refresh pin 001 = Reserved 010 = Reserved 011 = Reserved 100 = 66MHz external CPU clock 101 = 60MHz external CPU clock 110 = 50MHz external CPU clock 111 = 40MHz external CPU clock		

4.4.7 DRAM DMA/Master Cycles

For DMA and master cycles, the DRAM controller operates such that the MEMR# and MEMW# signals generate RASx# synchronously. The generation of the DRAM column address is then synchronized with LCLK. The synchronization can be programmed to be 0.5 to 1.5 LCLKs and 1.0 to 2.0 LCLKs. The generation of CASx# is always one LCLK after the generation of the column address. The cycles can thus be completed without adding wait states. For cases when the CPU write-back cache is enabled, wait states need to be added to the DMA/master cycles. This is because the CPU can request a primary cache castout (always a burst write to the DRAMs) and only after the castout is completed can the requested data from the DRAM be fetched.

Note: ISA masters which ignore IOCHRDY may not work when CPU write-back is enabled.

Figure 4-25 ISA Master Synchronization



W1 = 0.5 to 1.5 and 1 to 2 LCLKs
W2 = 1 LCLK

4.4.8 DRAM Hole Control

The Viper-MAX Chipset allows system "holes" in DRAM, to which accesses can go to the PCI bus. DRAM holes can be set through SYSCFG 09h[7:0], 0Ch[3:0], 0Ah[7:0], 0Bh[7:0], and 06h[7]. Table 4-36 shows these register bits.

Table 4-36 DRAM Hole Control Related Registers

7	6	5	4	3	2	1	0
SYSCFG 06h			**Shadow RAM Control Register 3**				**Default = 00h**
DRAM hole in system memory from 80000h-9FFFFh:(1) 0 = No hole in memory 1 = Enable hole in memory	Wait state addition for PCI master snooping: 0 = Do not add a wait state for the cycle access finish to do the snooping 1 = Add a wait state for the cycle access to finish and then do the snooping	C0000h-C7FFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM If SYSCFG 04h[2] = 1, then the E0000h-EFFFFh read/write control should have the same setting as this.		E0000h-EFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	

(1) This setting gives the user the option to have some other device in the address range 80000h-9FFFFh instead of system memory. When bit 7 is set, the 82C567 will not start the system DRAM controller for accesses to this particular address range.

SYSCFG 09h		**System Memory Function Register**		**Default = 00h**
DRAM Hole B size: 00 = 512KB 10 = 2MB 01 = 1MB 11 = 4MB Address for this hole is specified in SYSCFG 0Bh[7:0] and 0Ch[3:2]		DRAM Hole B control mode: 00 = Disable 01 = WT for L1 and L2 10 = Non-cacheable for L1 and L2 11 = Enable hole in DRAM	DRAM Hole A size: 00 = 512KB 10 = 2MB 01 = 1MB 11 = 4MB Address for this hole is specified in SYSCFG 0Ah[7:0] and 0Ch[1:0]	DRAM Hole A control mode: 00 = Disable 01 = WT for L1 and L2 10 = Non-cacheable for L1 and L2 11 = Enable hole in DRAM

SYSCFG 0Ah	**DRAM Hole A Address Decode Register 1**	**Default = 00h**
	DRAM Hole A starting address: These bits along with SYSCFG 0Ch[1:0] are used to specify the starting address of DRAM Hole A. These bits, AST[7:0], map onto HA[26:19] lines.	

SYSCFG 0Bh	**DRAM Hole B Address Decode Register 2**	**Default = 00h**
	DRAM Hole B starting address: These bits along with SYSCFG 0Ch[3:2] are used to specify the starting address of DRAM Hole A. These bits, BST[7:0], map onto HA[26:19] lines.	

4.5 PCI Bus Interface

The Viper-MAX Chipset supports up to five PCI bus masters. Both synchronous and asynchronous modes of operation of the PCI bus, with respect to the CPU, are supported. The Viper-MAX Chipset supports a 32-bit PCI implementation and supports PCI bus operating frequencies up to 33MHz. The PCI local bus controller is present in the 82C567 and the PCI data bus buffering is done within the 82C566. The 82C568 also functions as the PCI-to-ISA expansion bridge and performs the required data path conversion between the 32-bit PCI bus and the 8/16-bit ISA bus.

4.5.1 PCI Master Cycles

A PCI master is always allowed to access the system memory and system I/O spaces. Refer to Table 4-37.

4.5.1.1 System Memory Access

The PCI master asserts FRAME# and puts out the address on the AD[31:0] bus. The 82C567 decodes that address and asserts LMEM# to the 82C568 if the access is to system memory. The 82C568 then provides the data path to the PCI master to access system memory. If the access is to the system memory space, then the 82C567 acts as the PCI slave and it generates the appropriate control signals to snoop the L1 cache for every access, or for every access to a new line (if the line comparator is enabled). The 82C566 performs the data steering and latching based on the control information received from the 82C567 over the MDOE#, HDOE#, MMDOE#, DBCOE#, and the DLE[1:0]# lines.

Table 4-10 and Table 4-11 (on page 76) describe the sequence of events that take place during a master read/write cycle from/to system memory. Listed below is the data flow path for all such accesses by a PCI master.

Table 4-37 PCI Master Access Bits

7	6	5	4	3	2	1	0
PCIDV0 04h			**Command Register - Low Byte**				**Default = 07h**
Address/data stepping (RO): 0 = Disable (always)	PERR# output pin: 0 = Disable (always)	Reserved: Must be written to 0.	Memory write and invalidate cycle generation (RO): Must = 0 (always) No memory write and invalidate cycles will be generated by the 82C567.	Special cycles (RO): Must = 0 (always) The 82C567 does not respond to the PCI special cycle.	Bus master operations (RO): Must = 1 (always) This allows the 82C567 to perform bus master operations at any time. (Default = 1)	Memory access (RO): Must = 1 (always) The 82C567 allows a PCI bus master access to memory at anytime. (Default = 1)	I/O access (RO): Must = 1 (always) The 82C567 allows a PCI bus master I/O access at any time. (Default = 1)
PCIDV1 04h			**Command Register - Low Byte**				**Default = 07h**
Address/data stepping (RO): 0 = Disable (always)	PERR# output pin: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	Memory write and invalidate cycle generation (RO): Must = 0 (always) No memory write and invalidate cycles will be generated by the 82C568.	Special cycles: 0 = Disable 1 = Enable The 82C568 responds to Stop Grant special cycle.	Bus master operations: 0 = Disable 1 = Enable PCI cycle generation during DMA/ISA master may be disabled by this bit. (Default = 1)	Memory access (RO): Must = 1 (always) The 82C568 allows a PCI bus master access to memory at anytime. (Default = 1)	I/O access (RO): Must = 1 (always) The 82C568 allows a PCI bus master I/O access at any time. (Default = 1)

4.5.1.2 PCI Data Movement through 82C566 Buffers

Figures 4-26 and 4-27 are block diagrams that highlight PCI-to-DRAM and DRAM-to-PCI data movement through the data buffers of the 82C566.

Figure 4-26 4-Level Deep Ping-Pong PCI Buffer During PCI Master Write Cycles

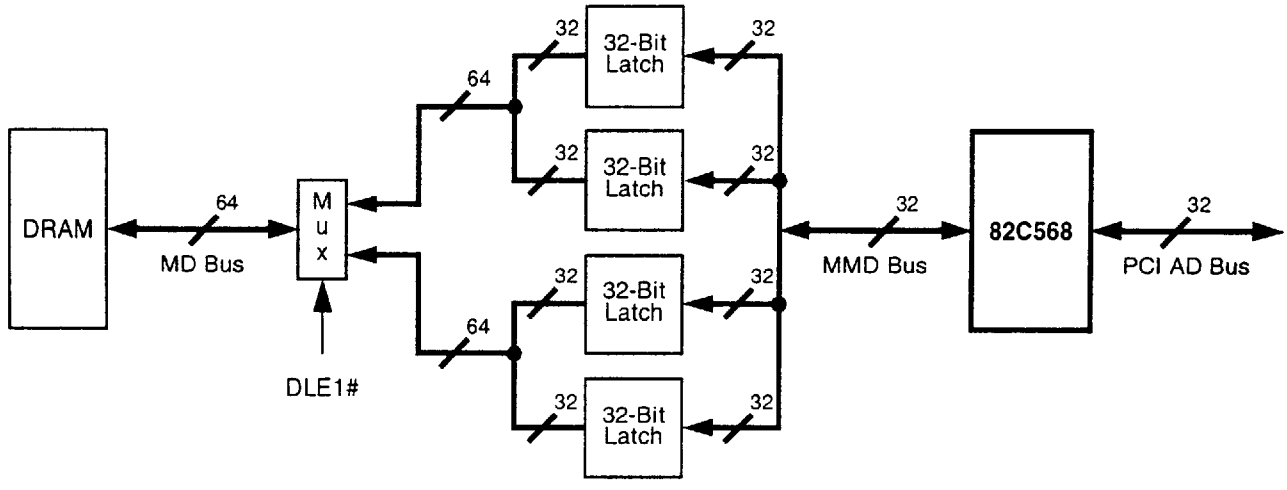
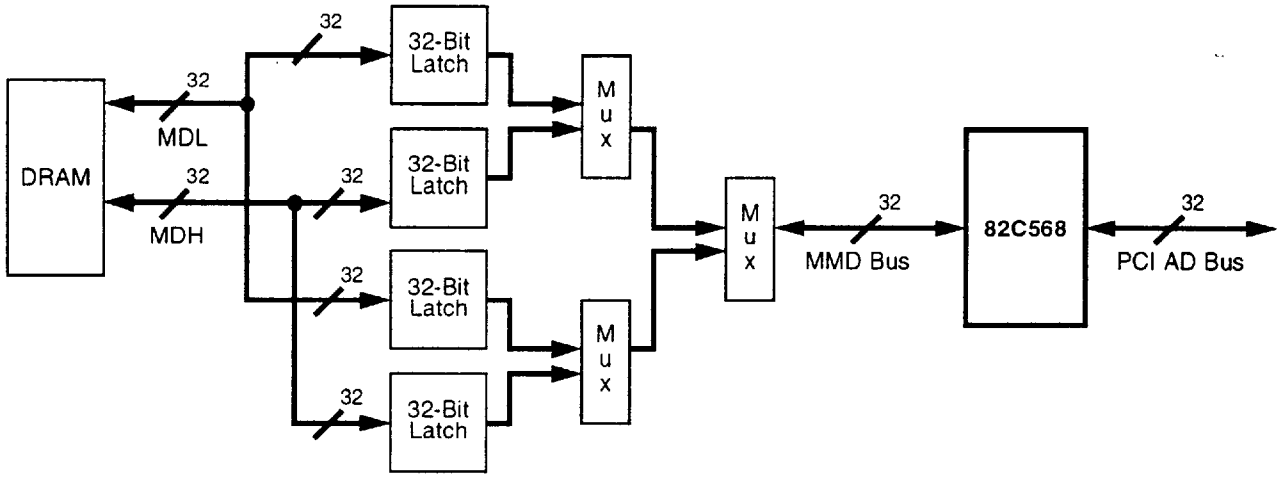


Figure 4-27 4-Level Deep Ping-Pong Buffer During PCI Master Read Cycles



4.5.1.3 X-1-1-1 Support on PCI Master Cycles

During PCI master read and write burst cycles into DRAM, the Viper-MAX Chipset has the capability to do X-1-1-1 cycles on the PCI bus. This increases the PCI bandwidth over 100MB/sec. With the pre-snoop feature of the Viper-MAX, a PCI master can sustain bursting to DRAM till a 4K page boundary is reached.

PCI-to-DRAM

PCI data from the AD bus is driven to the MMD bus via the 82C568 during data transfer phases. 32-bit MMD data from the PCI bus is posted into one of the 4-level deep ping-pong PCI-to-DRAM buffer. Using DLE1#, one set of 64-bit data will be output onto the MD bus and will be written into DRAM. On every PCI clock edge, PCI data will travel through the pipeline into the DRAM, hence achieving a X-1-1-1 throughput on the PCI bus.

DRAM-to-PCI

During PCI master read cycles, 64-bit data from DRAM is latched by one of the two 64-bit DRAM latch sets and 32-bit PCI data is output to the MMD bus on every PCI cycle. While this transfer is proceeding, the next 64-bit data will be read and latched into the second 64-bit DRAM latch and the cycle repeats. On every PCI clock edge, the data will travel through this pipeline into the PCI bus from DRAM, hence achieving a X-1-1-1 throughput on the PCI bus.

Figure 4-28 X-1-1-1 PCI Master Read Cycle

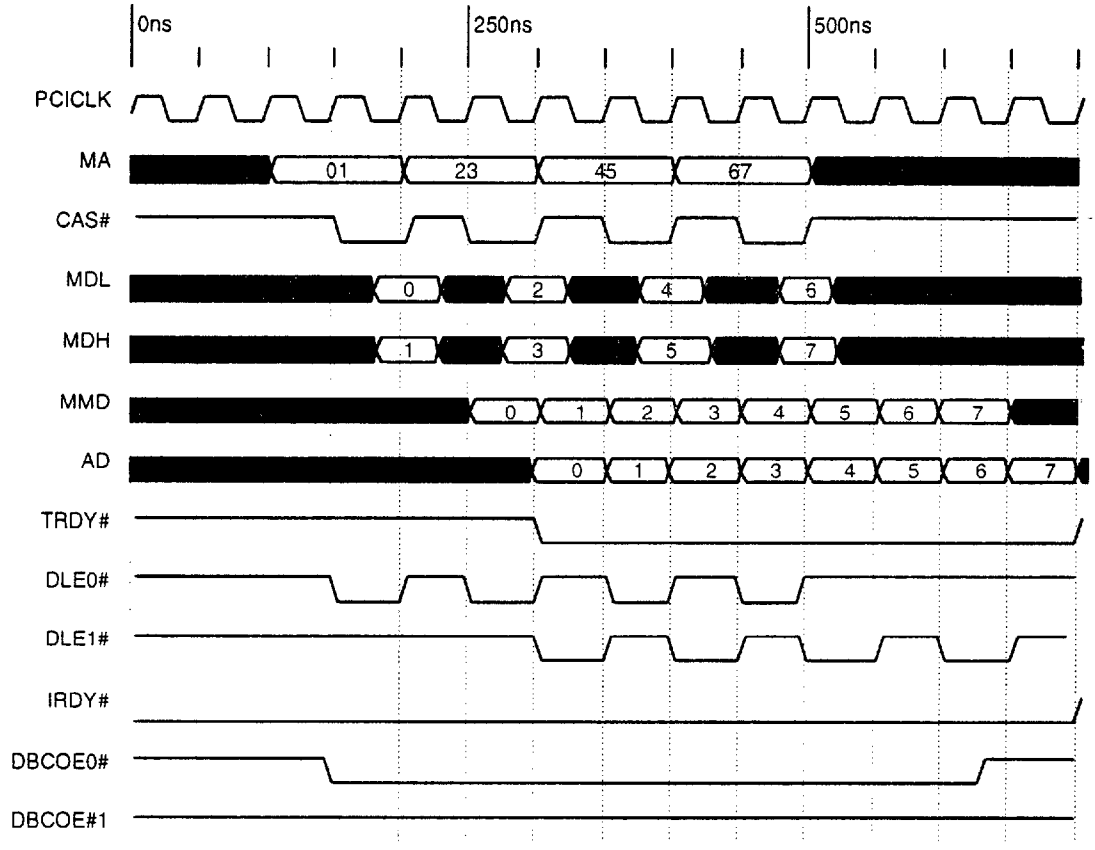
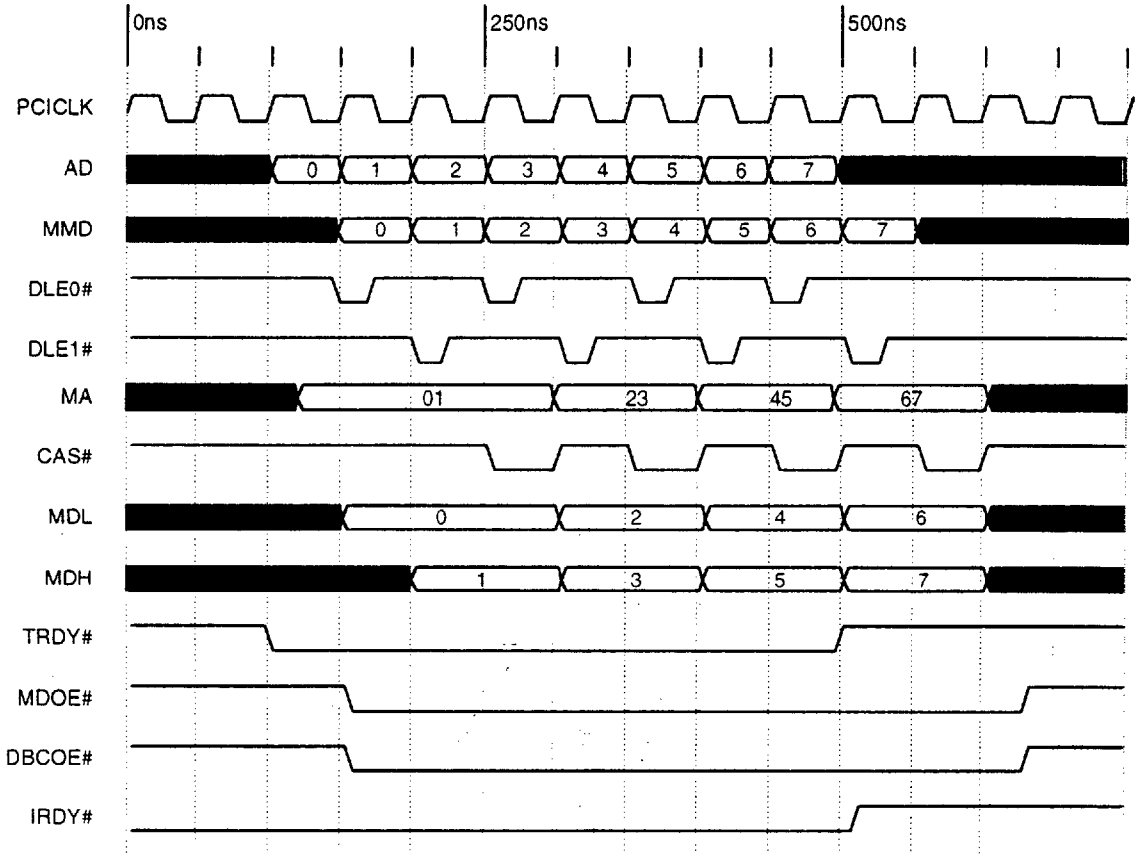


Figure 4-29 X-1-1-1 PCI Master Write Cycle



4.5.1.4 Non-Local Memory Access

The PCI master asserts FRAME# and outputs the address on AD[31:0]. If the access is not to the system memory area, the 82C567 does not assert LMEM# to the 82C568.

All other PCI slaves have up to three PCI CLKs after the start of the PCI cycle to assert DEVSEL#. All read/write access from/to PCI slaves is done directly over AD[31:0].

If no PCI slave responds within three PCI CLKs after the start of the cycle, then the 82C568 starts an ISA cycle. For a read access from the ISA bus, the ISA device outputs the data on SD[15:0] or SD[7:0], depending on whether it is a 16- or 8-bit slave. The 82C568 latches this data and then performs the appropriate data bus conversions and steering (based on the IOCS16#, MEMCS16#, SBHE# signals) and puts the data out on AD[31:0]. For a write access to the ISA bus, the PCI master puts out the data on AD[31:0]. The 82C568 latches this data and then performs the appropriate data bus conversions and steering (based on the IOCS16#, MEMCS16#, SBHE# signals) and outputs the data on SD[15:0] or SD[7:0], depending on whether it is a 16- or 8-bit slave.

4.5.1.5 PCI Master Pre-Snoop

Pre-snooping is a technique with the aid of which a PCI master can sustain bursting to the local memory till a 4K page boundary is reached. If pre-snooping is enabled, then on the first TRDY# of the PCI master cycle, the state machine within the 82C567 increments the HA[12:5] address lines by one and asserts EADS# to the CPU after that. By this time, the earlier cache address would have been latched by HACALE. If the CPU responds with a HITM#, then the current PCI master cycle will be terminated at the cache line boundary to allow the write-back cycle to occur. Enabling pre-snooping allows the Viper-MAX Chipset to continue bursting past a cache line boundary. Table 4-38 shows the register bits associated with the pre-snoop feature.

Table 4-38 Pre-Snoop Control Register Bits

7	6	5	4	3	2	1	0
SYSCFG 0Dh			**Clock Control Register**				**Default = 00h**
Reserved: Must be written to 0.				Enable A0000h-BFFFFh as system memory: 0 = No 1 = Yes	Add one more wait state during PCI master cycle with Intel-type address toggling(1): 0 = No 1 = Yes	Give the 82C567 control of the PCI bus on STOP# generation after HITM# is active: 0 = No 1 = Yes(2)	CPU clock is slowed down to below 33MHz: 0 = No 1 = Yes

(1) If the PCI master does its address toggling in the style of the Intel 486 burst, rather than a linear burst mode style, then one wait state needs to be added.

(2) The 82C567 has control over the PCI bus until the write-back is completed. If PCI master pre-snoop has been enabled (SYSCFG 0Fh[7] = 1), 0Dh[1] should be set to 1.

7	6	5	4	3	2	1	0
SYSCFG 0Fh			**PCI Master Burst Control Register 2**				**Default = 00h**
PCI pre-snoop: 0 = Disable 1 = Enable(1)	Insert wait states for ISA master access: 0 = No 1 = Yes	Reserved: Must be written to 0.	Resynchronize PCI master accesses to system DRAM: 0 = No 1 = Yes(2)	New mode of single cycle NA#: 0 = Disable 1 = Enable	CPU to L2 cache hit cycles, ASDC# generation from chipset: 0 = Enable 1 = Disable(3)	Write pulse duration control for operation with async SRAM: This bit is used when the write cycle takes the form of 3-X-X-X. 0 = 1 CPUCLK 1 = CPUCLK/2 plus the delay of an internal delay line	Cache size selection: This bit along with SYSCFG 02h[1:0] defines the L2 cache size. 0 = < 1MB 1 = ≥1MB

(1) The 82C567 generates a pre-snoop cycle to the CPU assuming that the PCI master will do a burst.

(2) If bit 4 = 1 in sync SRAM mode, PCI master access to system memory will force the master to wait for the current cycle to finish and the CPU-PCI clock to become sync. This is a conservative mode.

(3) SYSCFG 0Fh[2] needs to be set if pipelined sync SRAMs are being used.

Table 4-38 Pre-Snoop Control Register Bits (cont.)

7	6	5	4	3	2	1	0
SYSCFG 16h			**Dirty/Tag RAM Control Register**				**Default = 00h**
DIRTYI pin selection:(1) 0 = Input only 1 = I/O	Reserved: Must be written to 0.	Tag RAM size selection:(2) 0 = 8-bit 1 = 7-bit	Single write hit leadoff cycle in a combined Dirty/Tag implementation:(3) 0 = 5 cycles 1 = 4 cycles	Pre-snoop control: 0 = Pre-snoop for starting address 0 only 1 = Pre-snoop for all addresses except those on the line boundary	Synchronization between the PCI bus clock (LCLK) and the CPU clock (CLK):(4) 0 = LCLK async to CLK 1 = LCLK sync to CLK (skew not to exceed -2ns to 15ns)	Reserved: Must be written to 0.	HDOE# timing control: 0 = Negated normally 1 = Negated one clock before the cycle finishes

(1) If using a x1 SRAM for the Dirty RAM in which there is a separate DirtyIn and a separate DirtyOut bit, then the DIRTYI pin becomes an input only. If using a standard x8 or x9 SRAM, where there is no separate pin for input and output, then the DIRTYI pin becomes an I/O pin.

(2) If a 7-bit Tag is being used and a combined Tag/Dirty RAM is being used, then TAG0 functions as the DIRTYIO signal. In this case, the DIRTYI pin is unused.

(3) If bit 4 is set 1, SYSCFG 22h[0] should be set to 1.

(4) It should be noted that LCLK could be async to CLK also. This bit therefore implies that the PCI clock is either sync to the CPU clock with a skew not to exceed -2ns to 15ns, or that the PCI clock is async to the CPU clock.

7	6	5	4	3	2	1	0
SYSCFG 1Eh			**BOFF# Control Register**				**Default = 00h**
PCI master read cycle: 0 = Wait for IRDY# to be asserted before asserting TRDY# 1 = Generate TRDY# when checking for the status of IRDY#	Reserved: Must be written to 1.	Retry PCI pre-snoop HITM# cycle: 0 = Disable 1 = Enable	BOFF# generation if the PCI retry cycle is in A0000h-BFFFFh range: 0 = Not generated if bit 3 = 1 1 = Generated if bit 3 = 1 If bit 3 is not set to 1, then the setting of this bit has no effect.	Deadlock situation:(1) 0 = No way to avert deadlock situation if the write posting buffer on the PCI-to-PCI bridge has been enabled 1 = BOFF# is asserted to the CPU if deadlock situation occurs	Reserved: Must be written to 1.	When set to 1, PCI bursting will be disabled if BE[7:4]# and/or BE[3:0]# are not all 0.	Reserved: Must be written to 0.

(1) In a situation where there is a PCI-to-PCI bridge in a system and that bridge supports write posting, the following deadlock condition can occur. The bridge posts data from a master on the secondary PCI bus into its FIFO. If at the same time the 82C567 is accessing the bridge as a target, then the bridge will tell the 82C567 to retry its request after it has serviced out its FIFO. This will result in a deadlock situation. Bit 3 needs to be set to 1 if a DEC 21050 PCI-to-PCI bridge (or a similar chip) is being used.

4.5.2 PCI Slave Cycles

4.5.2.1 CPU Master Cycles

Any CPU cycle that is not an access to the system memory area, the 82C567 translates that cycle to a PCI cycle and asserts FRAME# on the PCI bus. All PCI slaves have up to three PCI CLKs after the start of the cycle within which to assert DEVSEL#. The data flow path would be similar to the ones described in the previous section.

4.5.2.2 PCI Byte/Word Merge

This feature, if turned on, allows successive 8-/16-bit writes from the CPU to a PCI slave, to be merged into a 32-bit entity and then sent out to the PCI slave. Byte/word merge is controlled by MDLE# and IRDY# from the 82C567. The number of MDLE# pulses sent out by the 82C567 before it asserts IRDY# determines how much data was sent out with each pulse. There is one additional control provided (in SYSCFG 00h[2:1]) for the byte/word merge implementation. This setting determines the maximum time difference within which consecutive PCI bytes/words could be merged.

To enable the byte/word merge feature, PCIDV1 4Eh[3], PCIDV1 4Eh[1], SYSCFG 17h[2], and SYSCFG 00h[4:3] should be set to 1. Refer to Table 4-39 for information on these register bits.

4.5.2.3 ISA Master Cycles

If the ISA master cycle is not a system memory access, then the 82C568 becomes the initiator and commences a PCI cycle. The data flow path for an ISA master to a PCI slave access is between the SD[15:0]/SD[7:0] lines and the AD[31:0] lines. The 82C568 handles all the data bus conversion and steering logic.

Table 4-39 Byte/Word Merge Feature Register Bits

7	6	5	4	3	2	1	0
PCIDV1 4Eh			**Miscellaneous Control Register - Low Byte**				**Default = 00h**
Reserved: Must be written to 0.				Pipelining with byte merge: 0 = Disable 1 = Enable	EOP configuration: 0 = Output 1 = Input	Byte merging: 0 = Disable 1 = Enable	ISA master data swap: 0 = Enable 1 = Disable
SYSCFG 17h			**PCI Cycle Control Register 2**				**Default = 00h**
Reserved: Must be written to 0.	Generate NA# for PCI slave access in async LCLK mode: 0 = No 1 = Yes This bit will be overridden if bit 7 is set.	Sync two bank select: 0 = Reserved 1 = Set this bit to 1 when two banks of sync SRAM are installed	Reserved: Must be written to 0.	Reserved: Must be written to 0.	Pipelining during byte merge: 0 = Disable 1 = Enable	Sync SRAM type (if SYSCFG 11h[3] = 1): 0 = Standard 1 = Pipelined	Burst type: 0 = Intel burst protocol 1 = Cyrix linear burst protocol
SYSCFG 00h			**Byte Merge/Prefetch & Sony Cache Module Control Register(1)**				**Default = 00h**
Enable pipelining of single CPU cycles to memory: 0 = Disable 1 = Enable	Video memory byte/word read prefetch enable: This setting enables/disables the prefetching of bytes/words/from PCI video memory by the CPU. 0 = Disable 1 = Enable	Sony SONIC-2WP support enable:(2) 0 = No Sony SONIC-2WP installed 1= Sony SONIC-2WP installed	Byte/word merge support: 0 = Disable 1 = Enable	Byte/word merging with CPU pipelining (NA# generation) support: 0 = Disable 1 = Enable	Time-out counter for byte/word merge: This setting determines the maximum time difference between two consecutive PCI byte/word writes to allow merging. 00 = 4 CPU CLKs 01 = 8 CPU CLKs 10 = 12 CPU CLKs 11 = 16 CPU CLKs		Enable internal hold requests to be blocked while performing byte merge: 0 = Disable 1 = Enable

(1) SYSCFG 13h[7] must be set to 1 in order for this register to be mapped correctly (full memory decode).

(2) If bit 5 is set, ensure that the L2 cache has been disabled (i.e., set SYSCFG 02h[3:2] = 00).

Figure 4-30 CPU 32-Bit Read from PCI

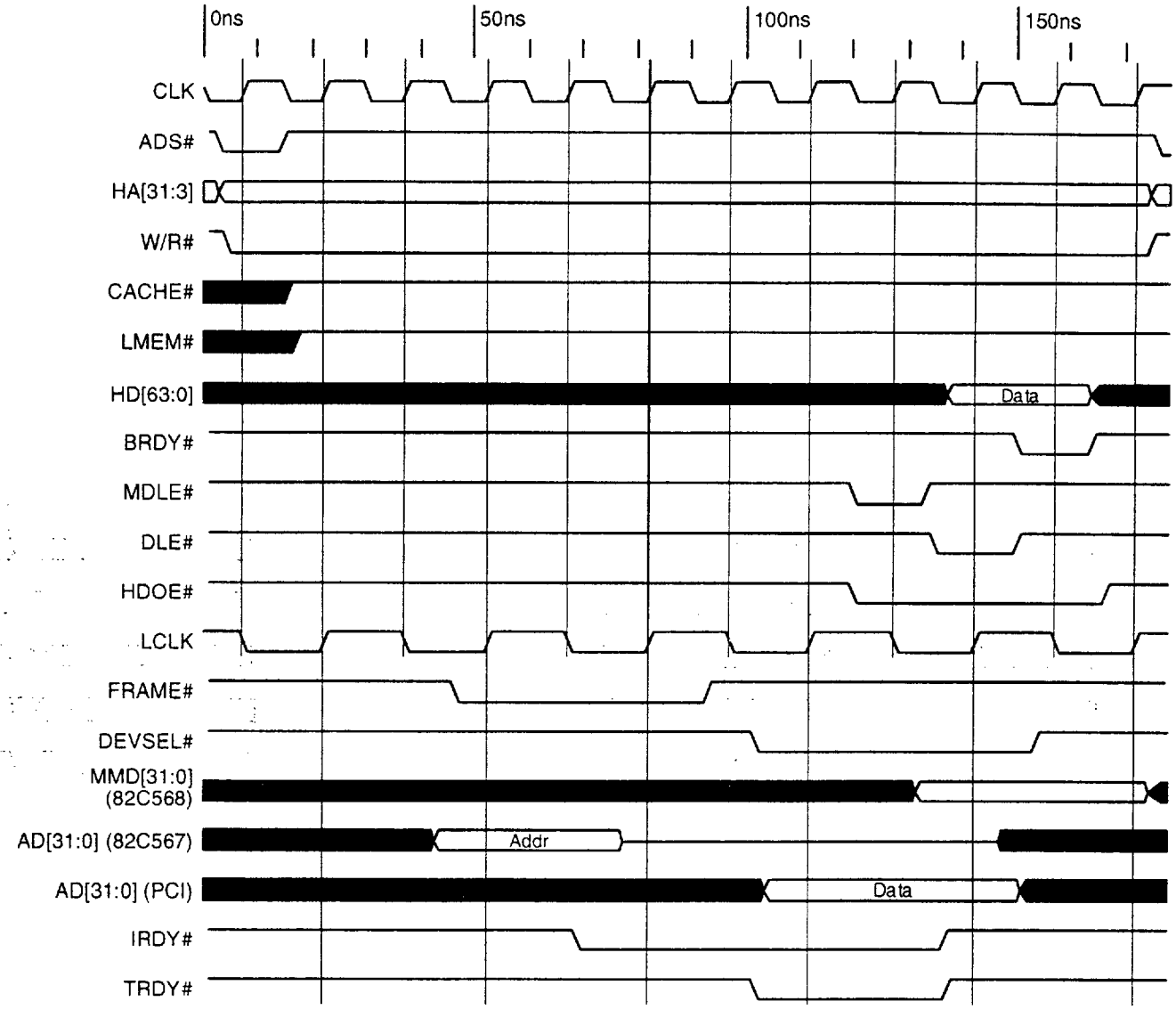


Figure 4-31 CPU 32-Bit Write to PCI

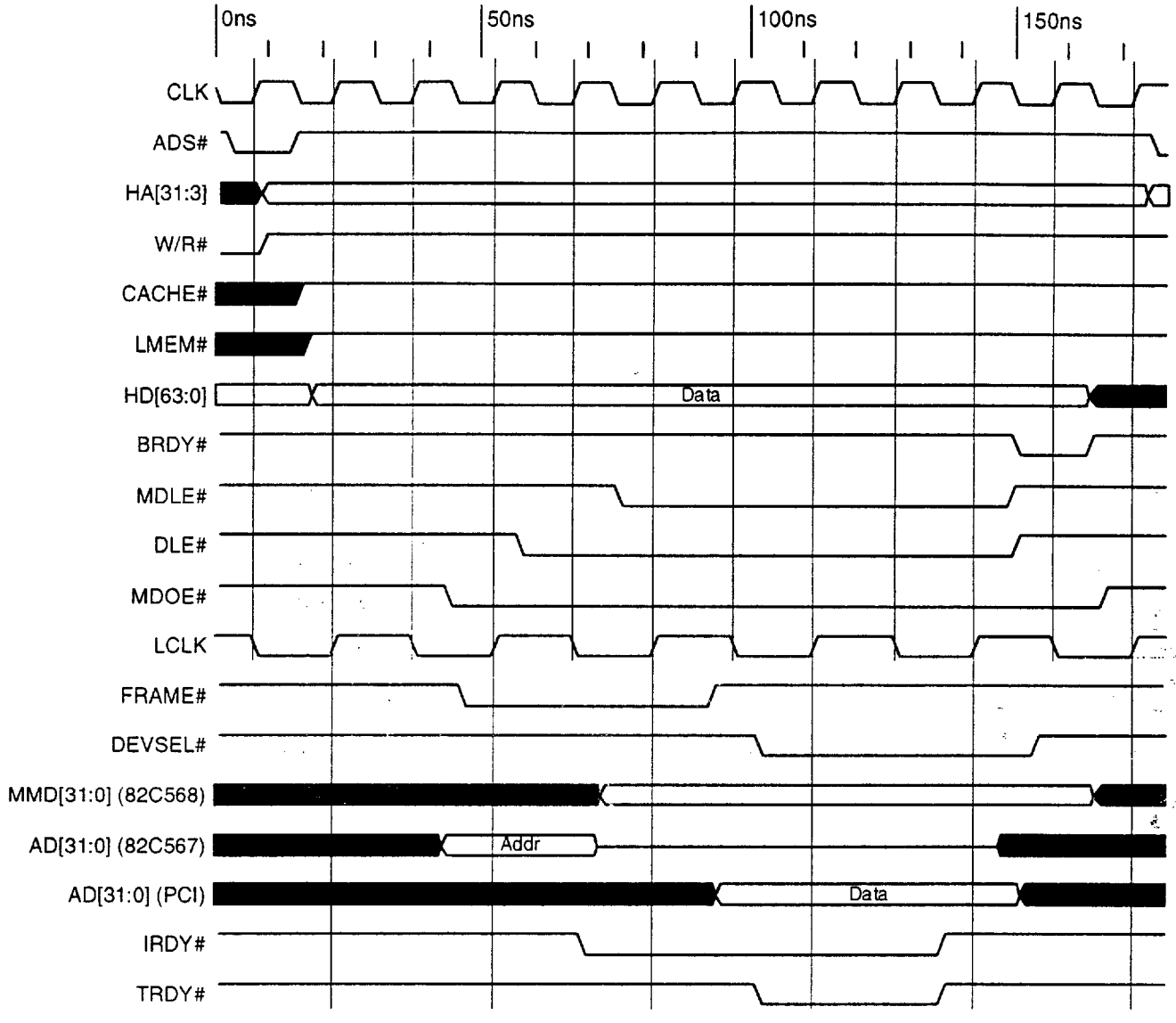


Figure 4-32 CPU 64-Bit Read from PCI

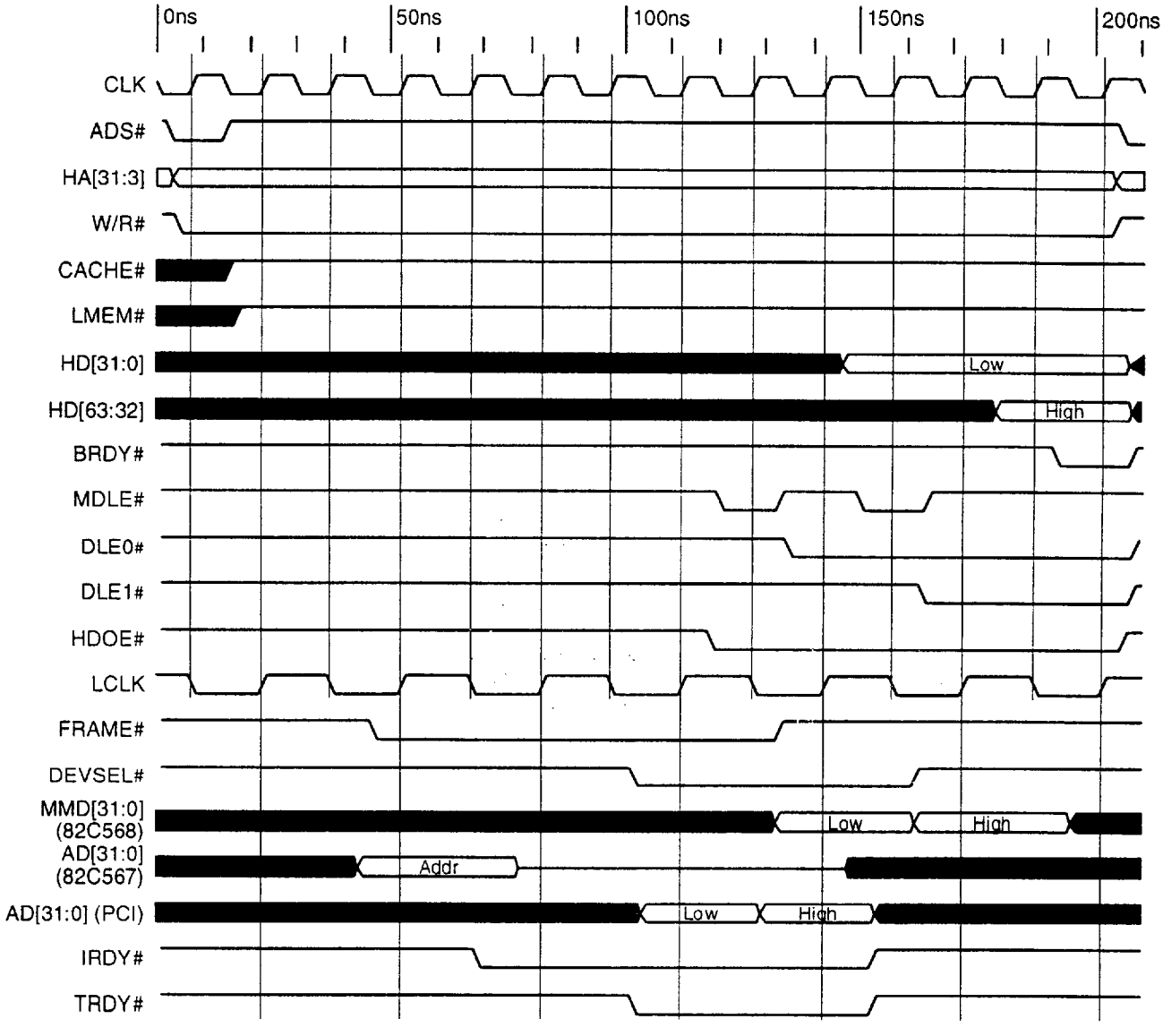


Figure 4-33 ISA Master Read from PCI

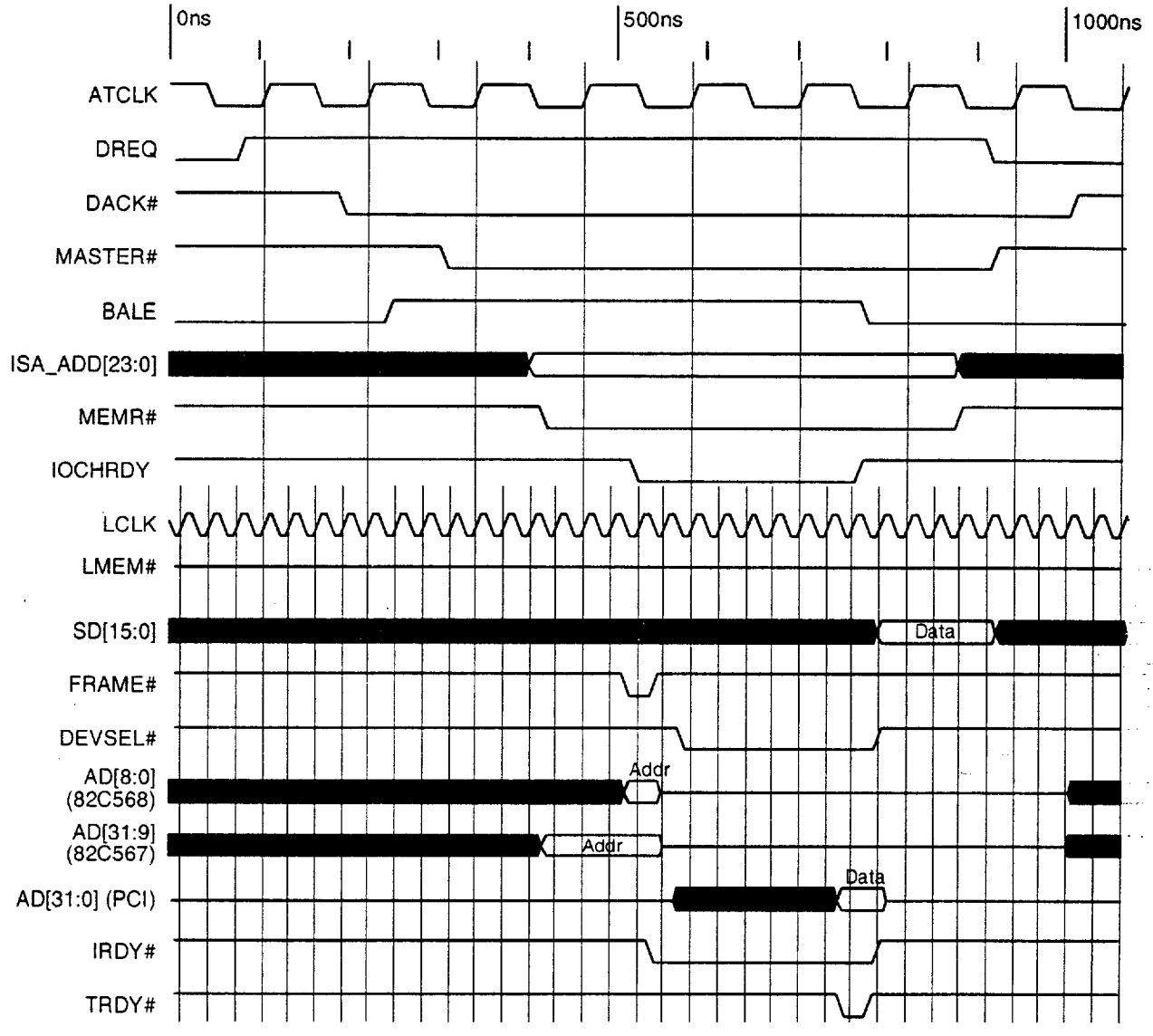


Figure 4-34 ISA Master Write to PCI

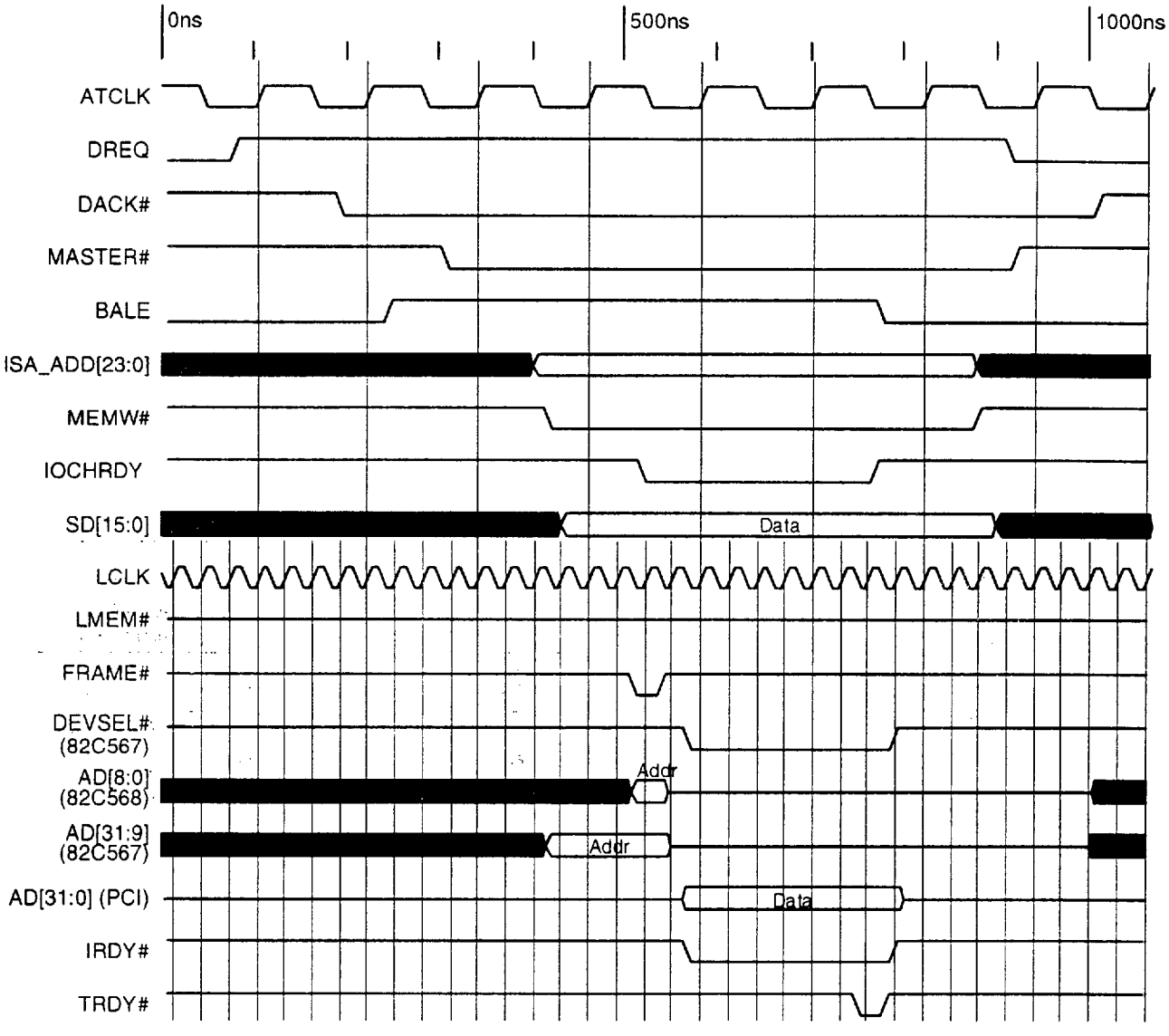


Figure 4-35 ISA Master Read from ISA Slave

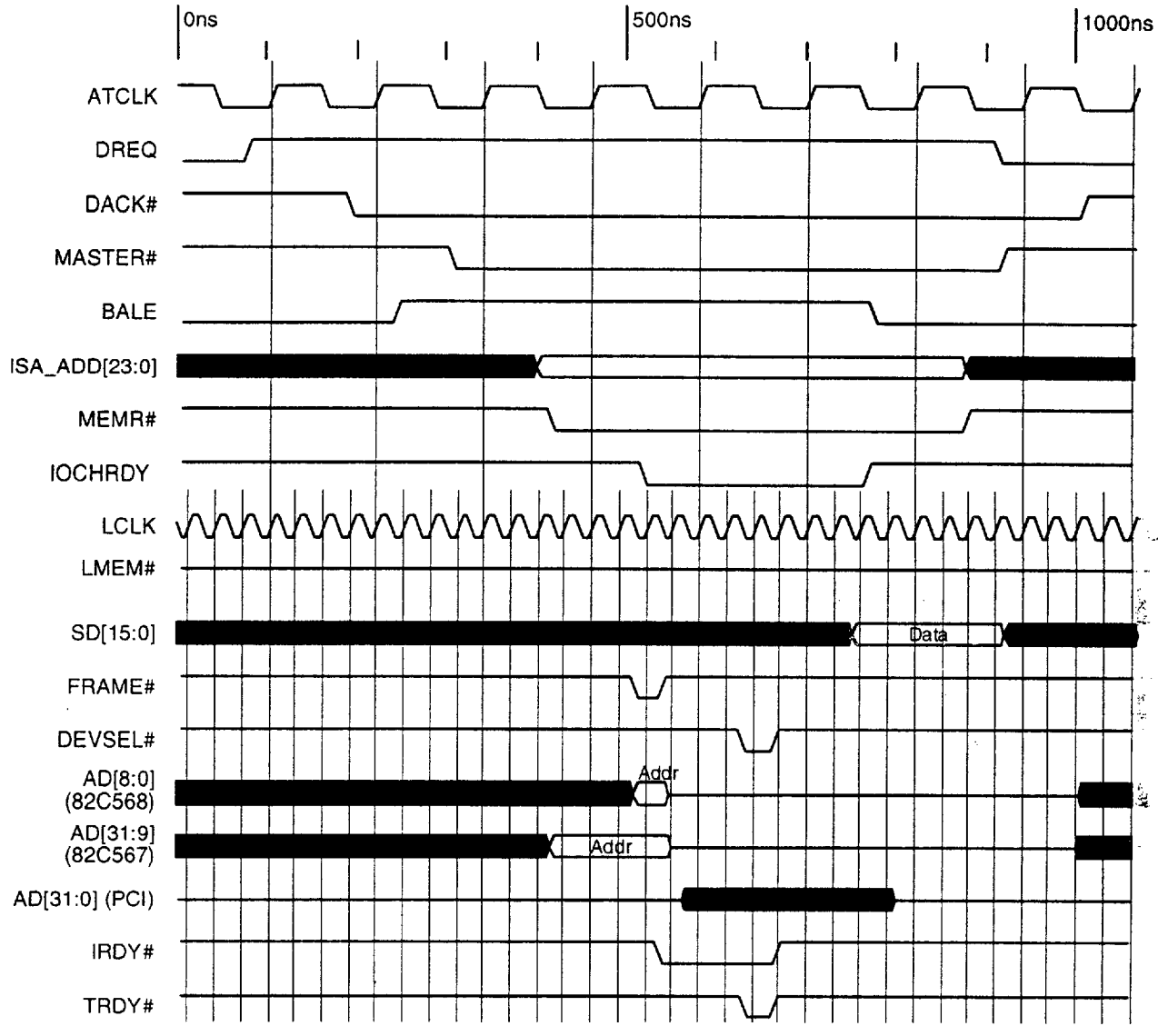
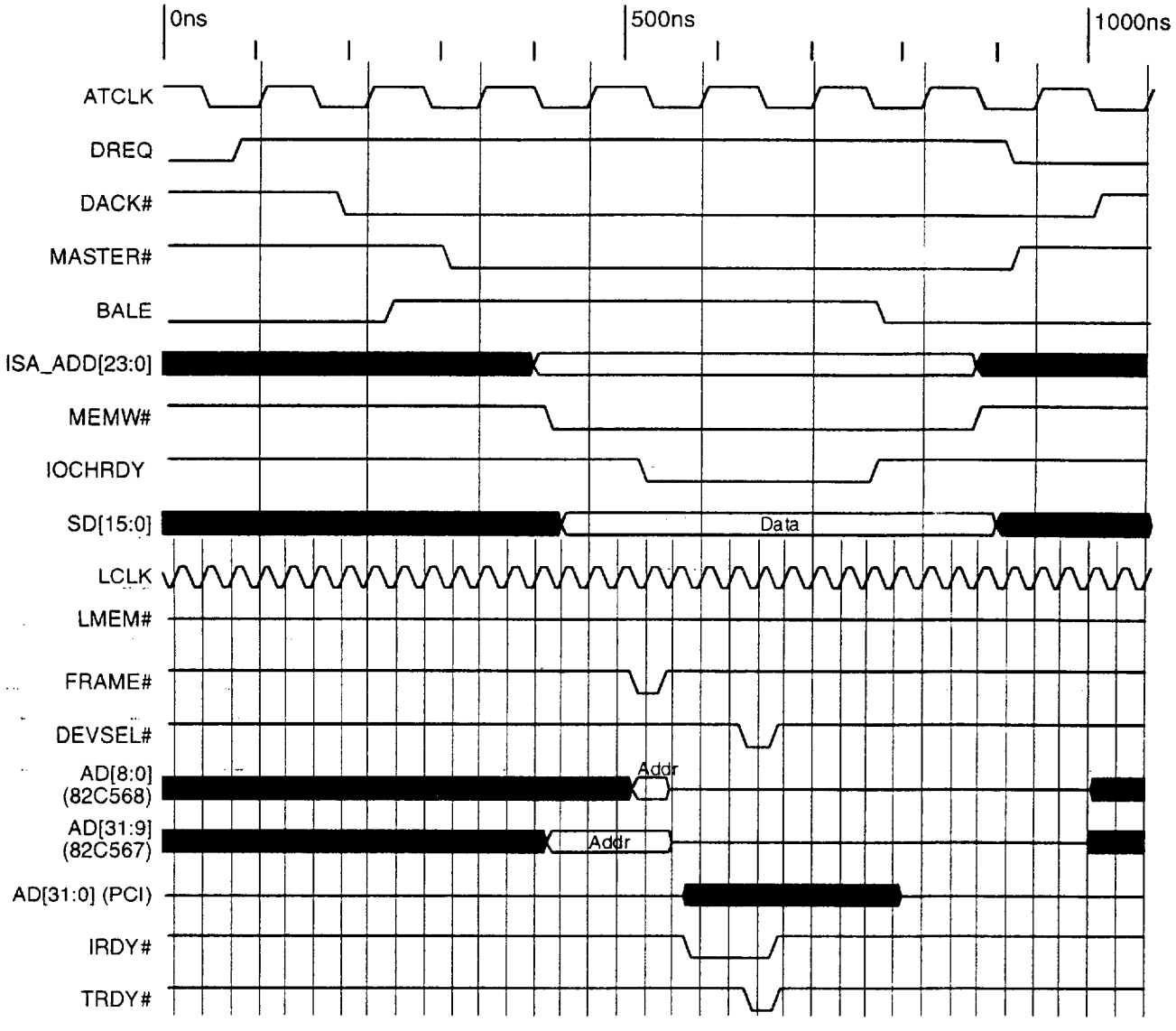


Figure 4-36 ISA Master Write to ISA Slave



4.6 UMA Support

The Viper-MAX Chipset supports a new memory architecture called Unified Memory Architecture (UMA). Under this new architecture, the graphics/video accelerator device will use a part of system memory as its frame buffer. This will eliminate the need for separate graphics DRAM to enable a memory cost savings of 1MB DRAM or more.

Under the UMA specification, the memory bus will be shared between the Host Memory Controller (82C567) and the VGA chip (hereafter referred to as GUI, graphical user interface). OPTi has defined a simple, but efficient, interface between the two. The resulting protocol is described in this section. A typical system block diagram with a UMA interface scheme is shown in Figure 4-37.

4.6.1 Types of Memories Supported

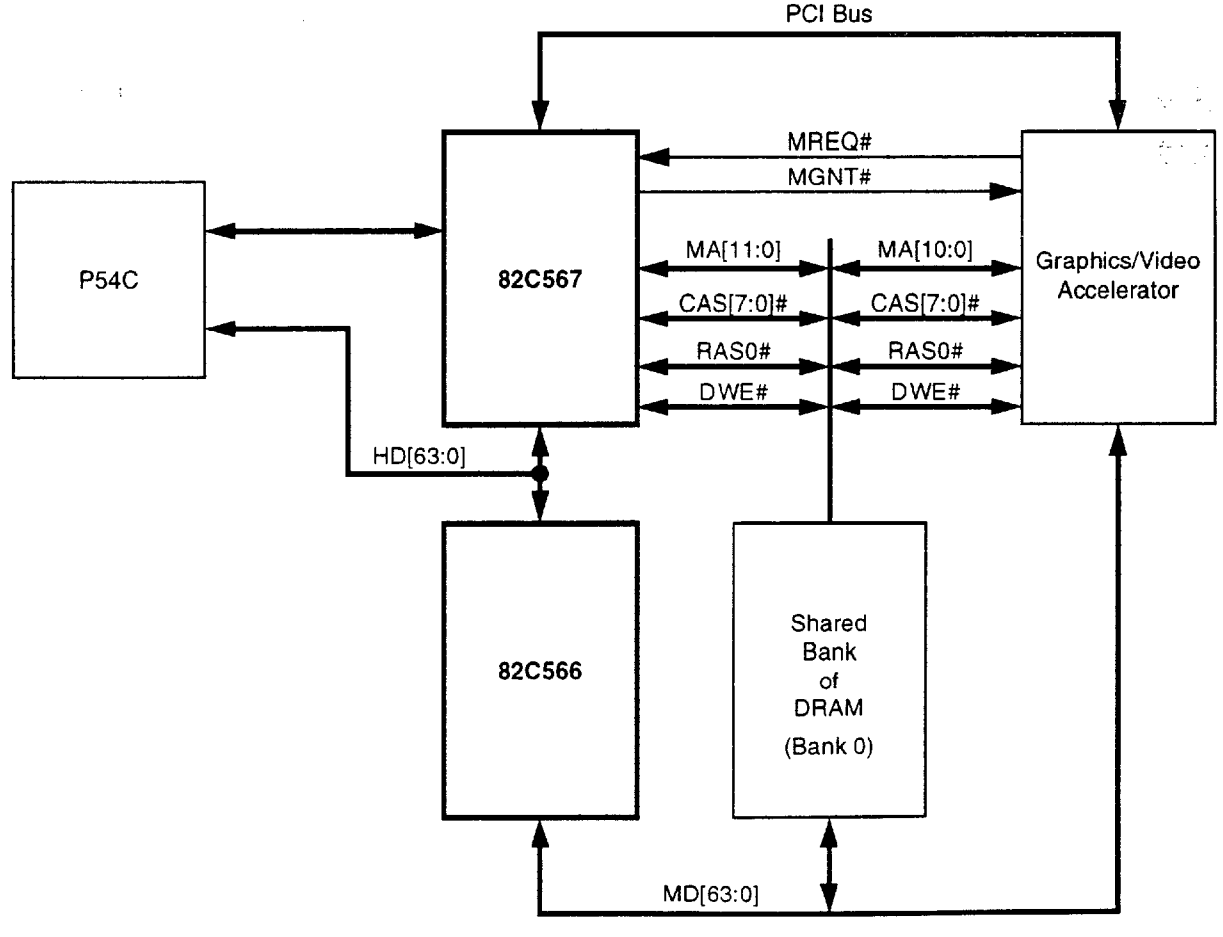
The interface will support fast page mode DRAM and EDO DRAMs at present. To minimize performance degradation to lowest levels, EDO DRAM (60/70ns) running at two clocks CAS cycle time at 66MHz is recommended. The shared memory will be mapped at the top of system memory and sometimes even beyond (making it noncontiguous into the system memory). The 82C567 will be responsible for refresh.

4.6.2 Control Signals

The following signals are shared by the 82C567 and GUI in a system based on UMA.

- RAS0#
  - Active low row address strobe for the shared memory Bank 0.
- CAS[7:0]#
  - Active low column address strobes, one for each byte lane.
- DWE#
  - Active low write enable.

Figure 4-37 UMA Interface System Block Diagram



4.6.3 Information Signals

To support up to 4MB GUI accessible space, the following memory address multiplexing scheme is specified.

- MA[9:0] or MA[10:0] (as per Tables 4-40 and 4-41)
  - Multiplexed memory address
- MD[63:0]
  - 64-bit bidirectional memory data.

4.6.4 Physical Memory Allocation Strategy

Figure 4-38 highlights the sharing of physical memory space between the 82C567 and GUI. The 82C567 can access the entire shared bank. The GUI can only access the top of the memory bank allocated to it.

Table 4-40 Memory Address Multiplexing for Symmetric and Asymmetric DRAMs "x9" and "x10"

Parameter	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
Column	1	A11	A10	A9	A8	A7	A6	A5	A4	A3
Row	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12

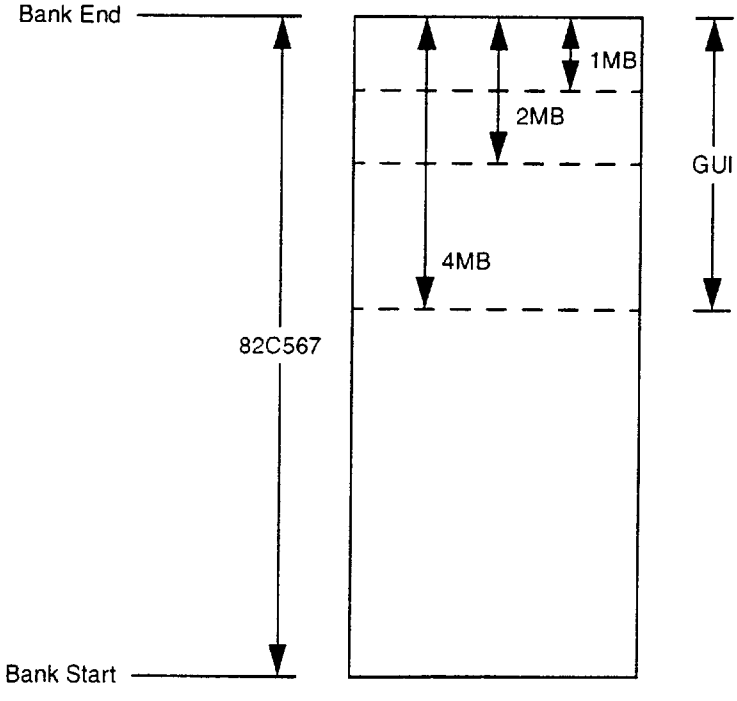
Table 4-41 Memory Address Multiplexing for Asymmetric DRAMs "x8"

Parameter	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
Column	-	-	-	A10	A9	A8	A7	A6	A5	A4	A3
Row	A21	A11	A20	A19	A18	A17	A16	A15	A14	A13	A12

Note: Memory address bit MA11 will not be connected to the GUI. MA10 and MA9 may be connected as per the tables above. When the bus is granted to the GUI, it is the responsibility of the 82C567 to drive MA[11:9] high for one clock and then tristate the output drivers. These address lines will need pull-up resistors to keep them high when they are not being driven.

If less than 4MB frame buffer is used by the GUI, upper MA address lines should be driven high by the GUI during GUI accesses.

Figure 4-38 Shared Memory Bank



4.6.4.1 Memory Mapping in a UMA-based System

In a system with the following properties,

- TD = Total DRAM size
- UME = End of Shared Memory (a portion of DRAM is remapped here)
- UMSZ = Size of Shared memory
- LFB = Linear Frame Buffer

the memory space, as viewed by different devices, is shown in Table 4-42.

The UME is programmable to take any value at integer multiples of 128MB (A[31:27] are programmable). The power-up default is 40000000h. This programmability allows the LFB to be located in any area the video BIOS/driver prefers. It is recommended that the area chosen >= 40000000h (1GB) in the CPU address space because:

- This allows the LFB always to be located at the same address as this is above the maximum host system memory address (768MB).
- This allows system software that does its own memory sizing not to confuse LFB memory with host system memory.

UMSZ is programmable to be 0.5M, 1M, 2M or 4M.

The GUI may optionally map the LFB area dually for CPU access to go through GUI to the same physical memory. Such an area can be chosen from any area that is mapped in the PCI space by the 82C567. Assuming that area is located at NLFB (NLFB >= UME), then the system memory map is as shown in Table 4-43.

Table 4-42 Memory Space

CPU Address	82C567	GUI
00000h to 9FFFFh	DRAM	x
A0000h to BFFFFh	PCI	VGA frame buffer
C0000h to C7FFFh	PCI/DRAM (ROM shadow)	GUI BIOS
C8000h to FFFFFh	PCI/DRAM (ROM shadow)	x
100000h to (TD-UMSZ)	DRAM	x
(TD-UMSZ) to (UME-UMSZ)	PCI	x
(UME-UMSZ) to UME	PCI/DRAM (remapped)	Linear frame buffer
UME to FFFBFFFFh	PCI	x
FFFC0000h to FFFC7FFFh	PCI	
FFFC8000h to FFFFFFFFh	PCI (high ROM)	x

Table 4-43 Memory Map with NLFB>= UME

CPU Address	82C567	GUI
00000h to 9FFFFh	DRAM	x
A0000h to BFFFFh	PCI	VGA frame buffer
C0000h to C7FFFh	PCI/DRAM (ROM shadow)	GUI BIOS
C8000h to FFFFFh	PCI/DRAM (ROM shadow)	x
100000h to (TD-UMSZ)	DRAM	x
(TD-UMSZ) to (UME-UMSZ)	PCI	x
(UME-UMSZ) to UME	PCI/DRAM (remapped)	Linear frame buffer
UME to NLFB	PCI	x
NLFB to (NLFB+UMSZ)	PCI	Alternate frame buffer
(NLFB+UMSZ) to FFFBFFFFh	PCI	x
FFFC0000h to FFFC7FFFh	PCI	
FFFC8000h to FFFFFFFFh	PCI (high ROM)	x

Example

- Total DRAM size = 800000h (TD = 8MB: programmed in 82C567 register)
- End of Shared Memory = 80000000h (UME = 2GB: programmed in 82C567 and GUI)
- Size of Shared memory = 100000h (UMSZ = 1MB: programmed in 82C567 and GUI)
- Alternate Frame Buffer = 80000000h (NLFB = 2GB: programmed in GUI)

Table 4-44 System Memory Map Example

CPU Address	82C567	GUI
00000h to 9FFFFh	DRAM	x
A0000h to BFFFFh	PCI	VGA frame buffer
C0000h to C7FFFh	PCI/DRAM (ROM shadow)	GUI BIOS
C8000h to FFFFFh	PCI/DRAM (ROM shadow)	x
1MB to 7MB	DRAM	x
7MB to 7FEFFFFFh	PCI	x
7FF00000h to 2GB	PCI/DRAM	Linear frame buffer
2GB to 800FFFFFh	PCI	Alternate frame buffer
80100000h to FFFBFFFFh	PCI	x
FFFC0000h to FFFC7FFFh	PCI	
FFFC8000h to FFFFFFFFh	PCI (high-ROM)	x

Note: TD needs to be programmed only in the 82C567. UME and UMSZ needs to be programmed to match values both in the 82C567 and GUI. NLFB needs to be programmed only in the GUI.

Table 4-45 UMA/GUI Associated Register Control Bits

7	6	5	4	3	2	1	0
SYSCFG 23h			**Pre-Snoop Control Register**				**Default = 00h**
Generate internal BREAK signal during master accessing of local memory cycle:(1) 0 = Old Mode 1 = New Mode	Bank 0 is selected as first or last bank: 0 = First bank 1 = Last bank	Pre-snoop for PCI X-1-1-1 write invalidate: 0 = Disable 1 = Enable	Pre-snoop for PCI X-1-1-1 read multiple and read line: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	Reserved: Must be written to 1.	Two clock MREQ# high extention by additional two clocks: 0 = Disable 1 = Enable **Note:** Set this bit to 1 in UMA systems	Reserved: Must be written to 0.
(1) Old Mode conditions: Sync SRAM, starting address AD[4:2] not = 000 or non-linear mode, master L2 cache write-through New Mode conditions: Sync SRAM, starting address AD[4:2] not = 000 or non-linear mode, master L2 cache write-through, L2 cache hit							
SYSCFG 25h			**GUI Memory Location Register**				**Default = 00h**
GUI memory location: A[31:27]					UMA size: 0 = Decided by SYSCFG 26h[5:4] 1 = 0.5MB if SYSCFG 26h[5:4] = 00	Reserved: Must be written to 0.	

Table 4-45 UMA/GUI Associated Register Control Bits (cont.)

7	6	5	4	3	2	1	0
SYSCFG 26h			**UMA Control Register**				**Default = 00h**
ISA master to DRAM cycle CAS width: 0 = Controlled by ISA R/W command pulse width 1 = 2 LCLKs	ISA SA address latch: 0 = SA latch is always transparent (pass-through) 1 = SA latch is on for retry only. (When the first CPU/ISA cycle is retried, the SA address will be latched.)	GUI memory size: 00 = 1MB 01 = 2MB 10 = 3MB 11= 4MB For 0.5MB size, set these bits to 00 and SYSCFG 25h[2] = 1.		5-2-2-2 EDO DRAM read timing at 66MHz: 0 = Disable 1 = Enable	00 = Normal 01 = For low priority GUI request, 82C567 will wait for two more CLKs 11 = GUI is always at high priority		UMA support: 0 = Disable 1 = Enable
SYSCFG 2Ah			**PCI-to-DRAM Deep Buffer Size Register**				**Default = 00h**
Reserved: Must be written to 0.	Time-out selection when there is a GUI request during PCI master read cycles: 00 = Always FP mode, grant DRAM bus to GUI ASAP 01 = Select SDRAM or EDO time-out depending on current bank information 1X = Select FP mode, SDRAM, or EDO depending on current bank information	PCI TRDY# wait state control with PCI-DRAM deep buffer: 0 = Zero wait state (X-1-1-1) 1 = One wait state (X-2-2-2)		Write burst with PCI-DRAM deep buffer: 0 = Disable 1 = Enable	Read burst with PCI-DRAM deep buffer: 0 = Disable 1 = Enable	PCI-to-DRAM deep buffer size: 00 = 16 dword 01 = 24 dword 10 = Reserved 11 = Reserved	
SYSCFG 2Fh			**CAS Address Setup Time Control Register**				**Default = 00h**
Column address to CAS delay for page miss cycles: 0 = Default 1 = 1 CLK	Reserved: Must be written to 0.	Generation of NA# during CPU accesses to non-shared DRAM bank when GUI has the memory bus: 0 = Enable 1 = Disable	Reserved: Must be written to 0.				

4.6.4.2 82C567-GUI DRAM Bus Arbitration Protocol

The GUI will arbitrate with the 82C567 for access to the shared memory through a two signal arbitration scheme. MREQ# is a signal driven by the GUI to the 82C567 and MGNT# is a signal driven by the 82C567 to the GUI. MREQ# and MGNT# are both active low signals which are driven and sampled synchronous to the Host-CPU clock (50, 60, or 66MHz) common to both the 82C567 and GUI. It is recommended that this interface be used with a bus frequency of 60MHz or higher.

The default owner will be the 82C567 and ownership will be transferred to the GUI upon demand. The GUI will return ownership to the 82C567 upon completion of its activities.

There are two signals establishing the communication protocol between the 82C567 and GUI. The MREQ# signal is driven by the GUI to the 82C567 and MGNT# is driven by the 82C567 to the GUI. The interface bus state machine will have five states:

- HOST State
  - The bus is with the 82C567 and no bus request from the GUI is pending.
- LPR State
  - The bus is with the 82C567 and a low priority bus request from the GUI is pending.
- HPR State
  - The bus is with the 82C567 and a bus request that was pending has become a high priority bus request pending.
- GNTD State
  - The bus is with the GUI.
- PRMT State
  - The bus is with the GUI, however, the 82C567 has a preemption request pending.

DRAM Bus Arbitration Rules

1. The GUI asserts MREQ# to generate a low priority request and keeps it asserted until the GUI obtains ownership of the bus through the assertion of MGNT#, unless the GUI wants to either raise a high priority request or raise the priority of an already pending low priority request. In the later case,

   a. if MGNT# is sampled asserted the GUI will not deassert MREQ#. Instead, the GUI will gain bus ownership and maintain MREQ# asserted until it wants to relinquish the bus.

   b. if MGNT# is sampled deasserted, the GUI will deassert MREQ# for one clock and assert it again irrespective of status of MGNT#. After reassertion, the GUI will keep MREQ# asserted until bus ownership is transferred to the GUI through assertion of MGNT# signal.

   The GUI may assert MREQ# only for the purpose of accessing the unified memory area. Once asserted, MREQ# should not be deasserted before MGNT# assertion for any reason other than raising the priority of the request (i.e., low to high). No speculative request and request abortion is permitted. If MREQ# is deasserted to raise the priority, it should be reasserted in the next clock and kept asserted until MGNT# is sampled asserted. If unconditional MREQ# deassertion is permitted, the 82C567 will have severe performance impact because of frequent page miss penalties.

2. Once MGNT# is sampled asserted by the GUI, it gains and retains bus ownership until MREQ# is deasserted.

3. If the GUI is over with its required transaction(s) before the 82C567 needs the bus again it will deassert MREQ#. In response to this, MGNT# will be deasserted in the next clock edge to change bus ownership back to the 82C567.

4. In case the 82C567 needs the bus before the GUI releases the bus on its own, it will deassert MGNT# to signal a preemption request to the GUI. But the GUI still retains ownership of the bus until it deasserts MREQ#.

5. When the GUI deasserts MREQ# to transfer bus ownership back to 82C567, either on its own or because of a preemption request, it should keep MREQ# deasserted for at least two clocks of recovery time before asserting it again to raise a request.

6. The shared signals are all driven by the 82C567 when it is the owner of the bus.

   If GUI requests the bus, the 82C567 asserts MGNT# to relinquish bus ownership and at the same clock edge tristates all the shared signals. The 82C567 will ensure that MA[11:9], CAS[7:0]#,and DWE# are driven high for a minimum of one clock and RAS# is driven high for a minimum of two clocks before tristating them.

   The GUI starts driving all the shared signals at the clock edge where MGNT# is first sampled asserted in response to a bus request. The 82C567 guarantees one clock of switch-over time from the 82C567 to GUI. Hereafter, the GUI drives the shared signals as long as MREQ# is asserted. At the end of its activities, the GUI deasserts MREQ# to relinquish bus ownership and tristates the shared signals at the same clock edge.

   The GUI will ensure that RAS#, CAS[7:0]#, and DWE# are driven high for a minimum of one clock before tristating them. The 82C567 starts driving the shared signals at the clock edge where MREQ# is sampled deasserted to indicate a bus release. Thereby, the GUI guarantees one clock of switch over time from the GUI to 82C567.

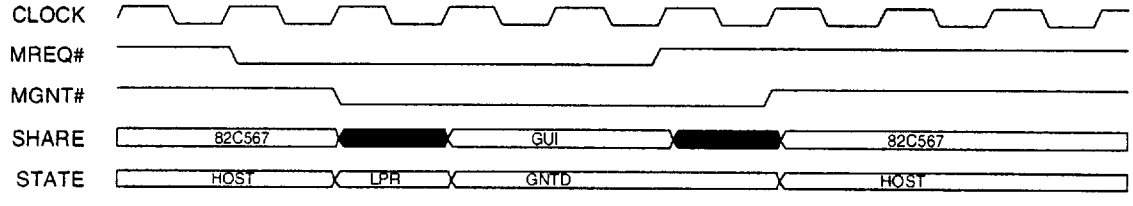
4.6.4.3 Advantage of Driving RAS# High for Two Clocks

The earliest the GUI can drive RAS# low to start a cycle is two clocks after MGNT# is asserted to release the bus to the GUI. So if the 82C567 provides for a minimum of two clock high time on RAS# prior to tristating the outputs and releasing the bus to the GUI, the 82C567 is able to ensure a four clock RAS-precharge time without the GUI bothering about RAS-precharge at the time of bus release to the GUI. Moreover if the Host had not been accessing the shared bank immediately before release of the bus, the 82C567 does not need to wait any extra clocks before releasing the bus to the GUI. Hence saving some clock latency in the arbitration.

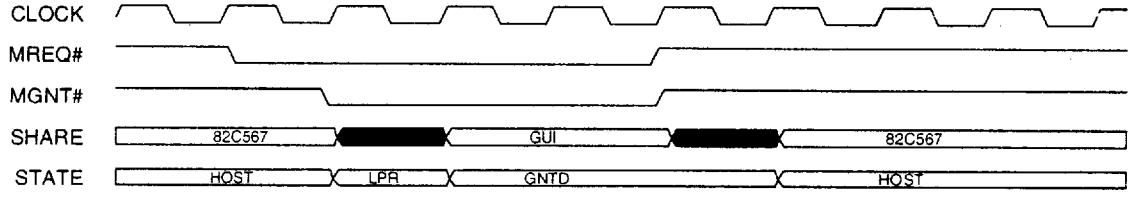
4.6.4.4 Bus Waveform Sample Diagrams

The following diagrams are samples of bus waveforms in different scenarios.

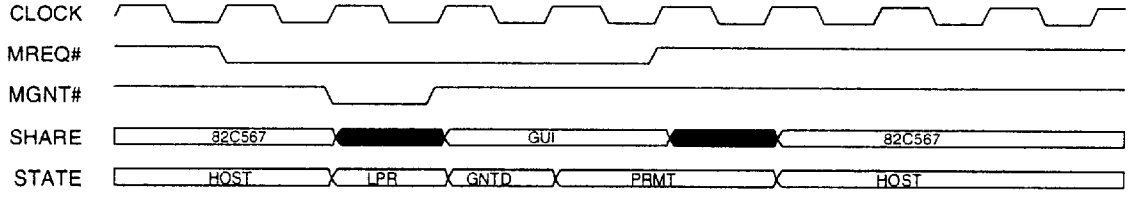
Figure 4-39 Case 1



A. Low priority request and immediate bus release to the GUI.

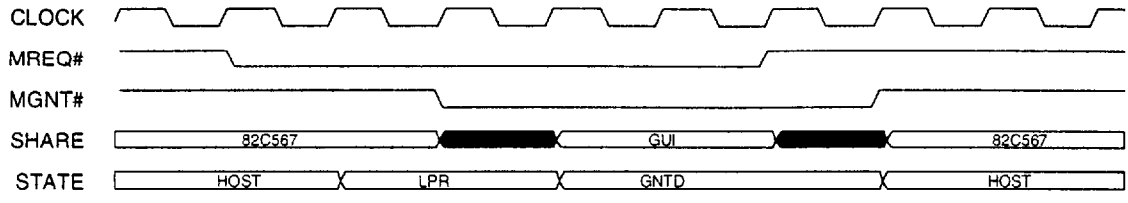


B. Low priority request and immediate bus release to the GUI with preemption where removal of MGNT# and removal of MREQ# coincides.



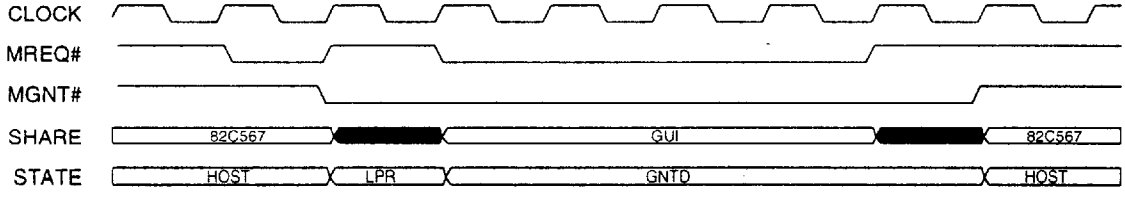
C. Low priority request and immediate bus release to the GUI with preemption where MREQ# is removed after the current transaction because of preemption.

Figure 4-40 Case 2

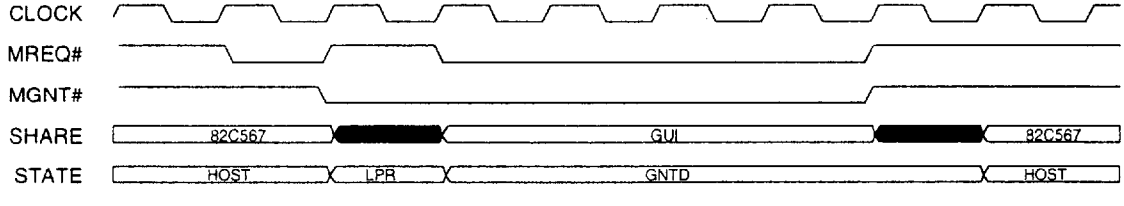


A. Low priority request and delayed bus release to the GUI.

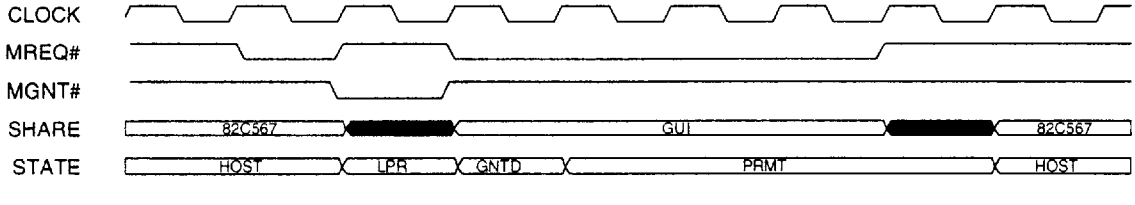
Figure 4-41 Case 3



A. High priority request and immediate bus release to the GUI.

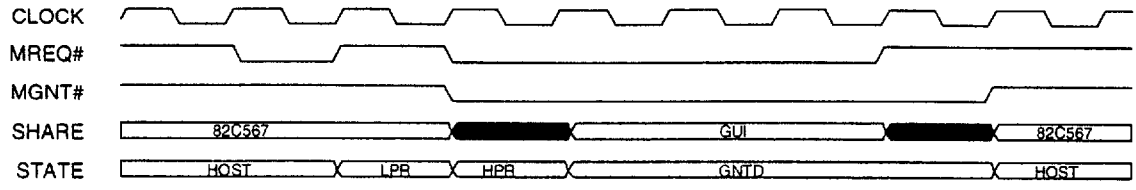


B. High priority request and immediate bus release to the GUI with preemption where MGNT# and MREQ# removal coincides.

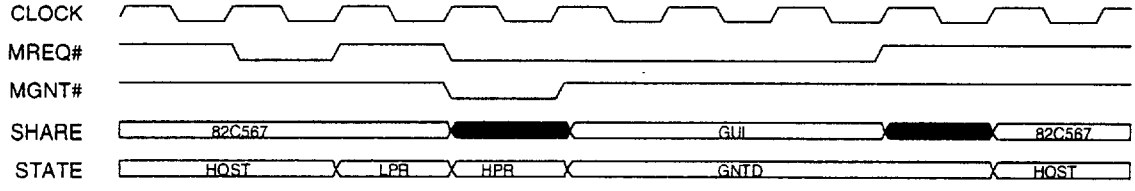


C. High priority request and immediate bus release to the GUI with preemption where MREQ# is removed after the current transaction because of preemption.

Figure 4-42 Case 4

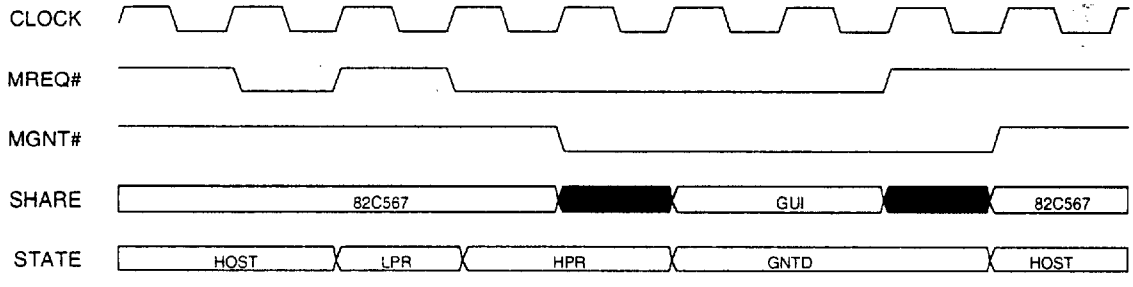


A. High priority request and one clock delayed bus release to the GUI.



B. High priority request and one clock delayed bus release to the GUI with preemption where MREQ# and MGNT# removal does not coincides.

Figure 4-43 Case 5



A. High priority request and delayed bus release to the GUI.

4.7 ISA Bus Interface

The ISA bus state machine gains control when the decoding logic of the 82C568 detects that no PCI device has claimed the cycle. It monitors status signals MEMCS16#, IOCS16#, and IOCHRDY, and performs the necessary synchronization of control and status signals between the ISA bus and the microprocessor. The Viper-MAX Chipset supports 8- and 16-bit memory and I/O devices located on the ISA bus.

An ISA bus cycle is initiated by asserting BALE in ISA-TS1 state. On the trailing edge of BALE, M16# is sampled for a memory cycle to determine the bus size. It then enters ISA-TC state and provides the command signal. For an I/O cycle, IO16# is sampled after the trailing edge of ALE until the end of the command. The command cycle is extended when IOCHRDY is detected inactive. Upon expiration of the wait states, the ISA state machine terminates itself and passes an internal READY to the CPU state machine to output a synchronous BRDY# to the CPU. The ISA bus state machine also routes data and address when an ISA bus master or DMA controller accesses system memory.

The delay between back-to-back ISA cycles is programmable and can be configured by programming PCIDV1 43h[3:2]. See Table 4-46.

4.8 XD Bus Interface

The XD bus is an 8-bit utility that is used to access the 8-bit keyboard controller, BIOS ROM, real-time clock, and non-volatile RAM (NVRAM). The XDIR output signal from the 82C568 is used for the XD bus data buffer direction control. A 1 indicates data transfer from the SD bus to the XD bus. Normally high, XDIR is low for the following conditions:

1) during BIOS ROM accesses, when ROMCS# and MEMR# are both active
2) during reads from I/O Ports 060h, 064h, 070h, and 071h
3) during read accesses from NVRAM

4.9 Bus Arbitration Logic

The 82C567 provides arbitration between the CPU, DMA controller, ISA bus masters, PCI bus masters, and the refresh logic. During DMA, ISA bus master cycles, PCI bus master cycles, and conventional refresh cycles, the 82C567 asserts HOLD to the CPU. The CPU will respond to an active HOLD signal by generating HLDA (after completing its current bus cycle) and placing most of its output and I/O pins in a high impedance state. After the CPU relinquishes the bus, the 82C567 responds by issuing REFRESH# (refresh cycle) or AHOLD (PCI master, ISA bus master, or DMA cycle), depending on the requesting device. During hidden refresh, HOLD remains negated and the CPU continues its current program execution as long as it services internal requests or achieves cache hits.

Table 4-46 Delay Back-to-Back ISA Cycle Register Bit

7	6	5	4	3	2	1	0
PCIDV1 43h			**Interrupt Edge/Level Control Register - High Byte**				**Default = 00h**
ISA IRQ14 recognition control: To use IDE on the PCI bus, the ISA IRQ14 signal needs to be blocked. 0 = ISA IRQ14 is honored 1 = ISA IRQ14 is ignored	ISA IRQ15 recognition control: To use secondary IDE on the PCI bus, the ISA IRQ15 signal needs to be blocked. 0 = ISA IRQ15 is honored 1 = ISA IRQ15 is ignored	DMA/ISA master to preempt PCI master: 0 = Disable 1 = Enable	Fixed/rotating priority between PCI masters: 0 = Rotating 1 = Fixed, priority sequence is PREQ0#, PREQ1#, PREQ2#	Back-to-back ISA I/O cycle delay: 00 = Delay by 3 ATCLKs 01 = Delay by 12 ATCLKs 10 = No delay 11 = Delay by 12 ATCLKs(1)		PCI master access to ISA devices: 0 = Enable 1 = Disable	ISA bus control signals for memory access greater than 16M and for I/O accesses greater than 64K: 0 = Enable 1 = Disable

(1) When bits [3:2] take on the combination of 11, all back-to-back cycles are delayed by 12 AT clocks. This is different from the combinations of 00 and 01 because in the latter case, the delay will be inserted only when an I/O access is followed by a second I/O access with no other type of access occurring in between (e.g., a memory access).

Refresh cycles, DMA cycles, and master cycles are serviced on a first in-first out (FIFO) priority, but DRAM refresh requests (REFRESH#) are internally latched and serviced immediately after the current DMA or master finishes its request, if the refresh request was queued behind an ISA DMA or master (HREQ) request. The 82C567 now requests the CPU bus by asserting HOLD to the CPU. The CPU will complete the ongoing cycle and when it gives up the CPU bus, it will assert HLDA to the 82C567. The 82C567 will grant the CPU bus to the PCI master, ISA DMA or master and assert AHOLD. The HREQ signal must remain active to be serviced if a refresh request comes first. DMA and bus masters share the same request pin; HREQ. To distinguish between DMA and bus master requests during an active AHOLD period, the AEN signal can be used to distinguish between DMA and master cycles. If AEN is active, then it is a DMA cycle. When these signals are inactive, an external bus master controls the system bus.

4.10 Data Bus Conversion/Data Path Control Logic

Data bus conversion from the 64-bit CPU bus to the memory bus is done by the 82C566 (based on control signals from the 82C567). The data bus conversion from the higher order MD bus to the AD bus is done by the 82C568, and the conversion to a 8/16-bit ISA bus is also done by the 82C568. The 82C567 converts the CPU byte enables BE[7:0]# to address A2 and four byte enable signals C/BE[3:0]#, for the PCI bus and the 82C568. The 82C568 uses the C/BE[3:0]#, A2 and the other ISA address (A[1:0], SBHE# and IOCS16#+ MEMCS16#) information to complete the 64-bit to 8/16-bit data conversion for the ISA bus. The 82C568 performs data bus conversion when the CPU accesses 16- or 8-bit devices through 16- or 32-bit instructions. It also handles DMA and ISA master cycles that transfer data between local DRAM or cache memory and locations on the ISA bus. The 82C567 provides all of the signals to control external bidirectional data buffers.

4.11 Special Cycles

4.11.1 System ROM BIOS Cycles

The 82C567 supports both 8- and 16-bit EPROM cycles. If the system BIOS is 16-bits wide, ROMCS# would be connected to MEMCS16# through an open collector gate indicating to the 82C568 that a 16-bit EPROM is responding. The system BIOS resides on the XD bus.

ROMCS# can be generated for both the E0000h-EFFFFh and F0000h-FFFFFFh segments through PCIDV1 4Ah and 4Bh. (Refer to Table 4-47.) If a combined video/system ROM BIOS is desired, these two segments should be used.

4.11.2 System Shutdown/Halt Cycles

The CPU provides special bus cycles to indicate that certain instructions have been executed or certain conditions have occurred internally. These special cycles, such as shutdown and halt, are covered by dedicated handling logic inside the 82C567. The Viper-MAX Chipset will generate INIT for a CPU shutdown cycle.

Table 4-47 Register Bits Associated with ROMCS#

7	6	5	4	3	2	1	0
PCIDV1 4Ah			**ROMCS# Range Control Register - Low Byte**				**Default = 00h**
ROMCS# for F8000h-FFFFFh: 0 = Enable 1 = Disable	ROMCS# for F0000h-F7FFFh: 0 = Enable 1 = Disable	ROMCS# for E8000h-EFFFFh: 0 = Disable 1 = Enable	ROMCS# for E0000h-E7FFFh: 0 = Disable 1 = Enable	ROMCS# for D8000h-DFFFFh: 0 = Disable 1 = Enable	ROMCS# for D0000h-D7FFFh: 0 = Disable 1 = Enable	ROMCS# for C8000h-CFFFFh: 0 = Disable 1 = Enable	ROMCS# for C0000h-C7FFFh: 0 = Disable 1 = Enable
PCIDV1 4Bh			**ROMCS# Range Control Register - High Byte**				**Default = 00h**
ROMCS# for FFFF8000h-FFFFFFFFh segment: 0 = Enable 1 = Disable	ROMCS# for FFFF0000h-FFFF7FFFh segment: 0 = Enable 1 = Disable	ROMCS# for FFFE8000h-FFFEFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFE0000h-FFFE7FFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFD8000h-FFFDFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFD0000h-FFFD7FFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFC8000h-FFFCFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFC0000h-FFFC7FFFh segment: 0 = Disable 1 = Enable

4.12 Internal Integrated 82C206

The following subsections give detailed operational information about the internal integrated 82C206 in the 82C568.

4.12.1 Top Level Decoder and Configuration Register

The top level decoder of the 82C206 provides eight separate enables to various internal subsystems. The following is a truth table for the top level decoder.

Address Range	Selected Device
000h-00Fh	DMA8 - 8-bit DMA Controller
020h-021h	INTC1 - Interrupt Controller 1
022h-023h	CONFIG - Configuration Register
040h-043h	CTC - Counter/Timer
080h-08Fh	DMAPAGE - DMA Page Register
0A0h-0A1h	INTC2 - Interrupt Controller 2
0C0h-0DFh	DMA16 - 16-Bit DMA Controller

Refer to Section 5.0, Register Descriptions, to program the various 82C206 registers.

4.12.2 DMA Subsystem

The 82C206 contains two 8237 DMA controllers. Each controller is a four channel DMA device which will generate the memory address and control signals necessary to transfer data between a peripheral device and memory directly. The two DMA controllers are internally cascaded to provide four DMA channels for transfers to 8-bit peripherals (DMA8) and three channels for transfers to 16-bit peripherals (DMA16). Channel 0 of DMA16 provides the cascade interconnection of the two DMA controllers, hence maintaining PC/AT compatibility. Hereafter, the description of the DMA subsystem pertains to both DMA8 and DMA16 unless otherwise noted.

Table 4-48 gives the I/O address map of the 82C206's DMA subsystem. The mapping is fully PC/AT compatible.

4.12.2.1 DMA Operation

During normal operation, the DMA subsystem of the 82C206 will be in one of three modes: the Idle mode, Program mode, or the Active mode. When the DMA controller is in the Idle mode, it only executes idle state cycles. The DMA controller will remain in the Idle mode unless it has been initialized to work and one of the DMA request pins has been asserted. In this case, the DMA controller will exit the Idle mode and enter the Active mode. The DMA controller will also exit the Idle mode and enter the Program mode when the CPU attempts to access its internal registers.

Idle Mode

If no peripheral requests service, the DMA subsystem will enter the Idle mode and perform only idle states. During this time, the 82C206 will sample the DREQ input pins every clock cycle to determine if any peripheral is requesting a DMA service. The internal select from the top level decoder and HLDA input pin will also sample at the same time to determine if the CPU is attempting to access the internal registers. With either of the above conditions, the DMA subsystem will exit the Idle mode and enter either the Program or Active mode. Note that the Program mode has priority over the Active mode since a CPU cycle has already started before the DMA was granted use of the bus.

Program Mode

The DMA subsystem will enter the Program mode whenever HLDA is inactive and an internal select from the top level decoder is active. During this time, the address lines A[3:0] become inputs if DMA8 is selected or A[4:1] become inputs if DMA16 is selected. These address inputs are used to decode which registers in the DMA controller are to be accessed. The IOR# and IOW# signals are used to select and time the CPU reads or writes. When DMA16 is selected, A0 is not used to decode and is ignored. Due to the large number and size of the internal registers of the DMA controller, an internal byte pointer flip-flop is used to supplement the addressing of the 16-bit word and count address registers. This byte pointer is used to determine the upper or lower byte of word count and address registers and is cleared by a hardware reset or a master clear command. It may also be set or cleared by the CPU's set byte pointer flip-flop or clear byte pointer flip-flop commands.

The DMA subsystem supports some special commands when in the Program mode. These commands do not use the data bus, but are derived from a set of address, the internal select, and IOR# or IOW#. These commands are listed at the end of Table 4-48. Erratic operation of the 82C206 can occur if a request for service occurs on an unmasked DMA channel which is being programmed. The channel should be masked or the DMA should be disabled to prevent the 82C206 from attempting to service a peripheral with a channel which is only partially programmed.

Table 4-48 DMA I/O Address Map

Address		Operation			
DMA8	DMA16	XIOR#	XIOW#	Byte Pointer	Register Function
000h	0C0h	0	1	0	Read Channel 0's current address low byte
		0	1	1	Read Channel 0's current address high byte
		1	0	0	Write Channel 0's base and current address low byte
		1	0	1	Write Channel 0's base and current address high byte
001h	0C2h	0	1	0	Read Channel 0's current word count low byte
		0	1	1	Read Channel 0's current word count high byte
		1	0	0	Write Channel 0's base and current word count low byte
		1	0	1	Write Channel 0's base and current word count high byte
002h	0C4h	0	1	0	Read Channel 1's current address low byte
		0	1	1	Read Channel 1's current word count high byte
		1	0	0	Write Channel 1's base and current address low byte
		1	0	1	Write Channel 1's base and current address high byte
003h	0C6h	0	1	0	Read Channel 1's current word count low byte
		0	1	1	Read Channel 1's current word count high byte
		1	0	0	Write Channel 1's base and current word count low byte
		1	0	1	Write Channel 1's base and current word count high byte
004h	0C3h	0	1	0	Read Channel 2's current address low byte
		0	1	1	Read Channel 2's current address high byte
		1	0	0	Write Channel 2's base and current address low byte
		1	0	1	Write Channel 2's base and current address high byte
005h	0CAh	0	1	0	Read Channel 2's current word count low byte
		0	1	1	Read Channel 2's current word count high byte
		1	0	0	Write Channel 2's base and current word count low byte
		1	0	1	Write Channel 2's base and current word count high byte
006h	0CCh	0	1	0	Read Channel 3's current address low byte
		0	1	1	Read Channel 3's current address high byte
		1	0	0	Write Channel 3's base and current address low byte
		1	0	1	Write Channel 3's base and current address high byte
007h	0CEh	0	1	0	Read Channel 3's current word count low byte
		0	1	1	Read Channel 3's current word count high byte
		1	0	0	Write Channel 3's base and current word count low byte
		1	0	1	Write Channel 3's base and current word count high byte
008h	0D0h	0	1	X	Read Status Register
		1	0	X	Write Command Register
009h	0D2h	0	1	X	Read DMA Request Register
		1	0	x	Write DMA Request Register
00Ah	0D4h	0	1	X	Read Command Register
		1	0	X	Write single bit DMA Request Mask Register
00Bh	0D6h	0	1	X	Read Mode Register
		1	0	X	Write Mode Register
00Ch	0D8h	0	1	X	Set byte pointer flip-flop
		1	0	X	Clear byte pointer flip-flop
00Dh	0DAh	0	1	X	Read Temporary Register
		1	0	X	Master clear
00Eh	0DCh	0	1	X	Clear Mode Register counter
		1	0	X	Clear all DMA Request Mask Register bits
00Fh	0DEh	0	1	X	Read all DMA Request Mask Register bits
		1	0	X	Write all DMA Request Mask Register bits

Active Mode
The DMA subsystem will enter the Active mode whenever a software request occurs or a DMA request occurs on an unmasked channel which has already been programmed. An example of this would be a DMA read cycle. After receiving a DREQ, the 82C206 will issue HOLD to the CPU. Until an HLDA is returned from the CPU, the DMA subsystem will remain in an idle state. On the next clock cycle, the DMA will exit the idle state and enter an S0 state. During S0, the DMA will resolve priority and issue DACK on the highest priority channel which is requesting service. The DMA then enters the S1 state where the multiplexed addresses are output and latched. Next, the DMA enters the S2 state where the 82C206 asserts the MEMR# command. Then the DMA will enter the S3 state where the 82C206 asserts the IOW# command. The DMA will then remain in the S3 state until the wait state counter has expired and IOCHRDY is high. Note that at least one additional S3 will occur unless compressed timing is programmed. Once a ready condition is detected, the DMA will enter S4 where MEMR# and IOW# are negated.

In the Compressed and Demand modes, subsequent transfers will begin in S2 unless the intermediate addresses require updating. In these subsequent transfers, the lower addresses are changed in S2.

4.12.2.2 DMA Transfer Modes

There are four transfer modes supported by the DMA subsystem: Single, Block, Demand, and Cascade. The DMA subsystem can be programmed on a channel-by-channel basis to operate in one of these four modes.

Single Transfer Mode
In the Single Transfer mode, the DMA will execute only one cycle at a time. DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the Single Transfer, the 82C206 will negate HOLD and release the bus to the system once the transfer is complete. After HLDA has gone inactive, the 82C206 will again assert HOLD and execute another transfer on the same channel unless a request from a higher priority channel has been received.

During the Single Transfer mode, the CPU is ensured of at least one full machine cycle execution between DMA transfers. Following each transfer, the Word Count Register is decreased and the Address Register is increased or decreased (depending on the DEC bit of the Mode Register). When the word count decrements from 0000h to FFFFh, the terminal count bit in the Status Register is set and a pulse is output to the TC pin. If auto-initialization is selected, the channel will reinitialize itself for the next service - otherwise, the DMA will set the corresponding DMA request bit mask and suspend transferring on that channel.

Block Transfer Mode
In the Block Transfer mode, the DMA will begin transfers in response to either a DREQ or a software reset. If DREQ starts the transfer, it needs to be held active until DACK becomes active. The transfers will continue until the word count decrements from 0000h to FFFFh, at which time the TC pin is pulsed and the terminal count bit in the Status Register is set. Once more, an auto-initialization will occur at the end of the last service if the channel has been programmed to do so.

Demand Transfer Mode
In the Demand Transfer mode, the DMA will begin transfers in response to the assertion of DREQ and will continue until either the terminal count is reached or DREQ becomes active. The Demand Transfer mode is normally used for peripherals which have limited buffering capacity. The peripheral can initiate a transfer and continue until its buffer capacity is exhausted. The peripheral may then re-establish service by again asserting DREQ. During idle states between transfers, the CPU is released to operate and can monitor the operation by reading intermediate values from the Address and Word Count Registers. Once DREQ is negated, higher priority channels are allowed to intervene. Reaching the terminal count will result in the generation of a pulse on the TC pin, the setting of the terminal count bit in the Status Register, and auto-initialization if programmed to do so.

Cascade Mode
The Cascade mode is used to interconnect more than one DMA controller to extend the number of DMA channels while preserving the priority chain. While in this mode, the master DMA controller does not generate address or control signals. The DREQ and DACK signals of the master are used to interface the HOLD and HLDA signals of the slave DMA devices. Once the master has received an HLDA from the CPU in response to a DREQ caused by the HOLD from a slave DMA controller, the master DMA controller will ignore all inputs except HLDA from the CPU and DREQ on the active channel. This prevents conflicts between the DMA devices.

Figure 4-44 shows the cascade interconnection for two levels of DMA devices. Note that Channel 0 of DMA16 is internally connected for the Cascade mode to DMA8. Additional devices can be cascaded to the available channels in either DMA8 or DMA16 since the Cascade mode is not limited to two levels of DMA controllers.

When programming cascaded controllers, begin with the device which is actually generating HRQ to the system (first level device) and then proceed to the second level devices. RESET causes the DACK outputs to become active low and are placed in the inactive state. To allow the internal cascade between DMA8 and DMA16 to operate correctly, the active low state of DACK should not be modified. The first level device's DMA request mask bits will prevent the second level cascaded devices from generating unwanted hold requests during the initialization process.

4.12.2.3 Transfer Types

There are three types of transfers:

- Read Transfers
- Write Transfers
- Verify Transfers

The Single, Block, and Demand Transfer modes can perform any of the three transfer types.

Read Transfers move data from memory to an I/O peripheral by generating the memory address and asserting MEMR# and IOW# during the same transfer cycle.

Write Transfers move data from an I/O peripheral to memory by generating the memory address and asserting MEMW# and IOR# during the same transfer cycle.

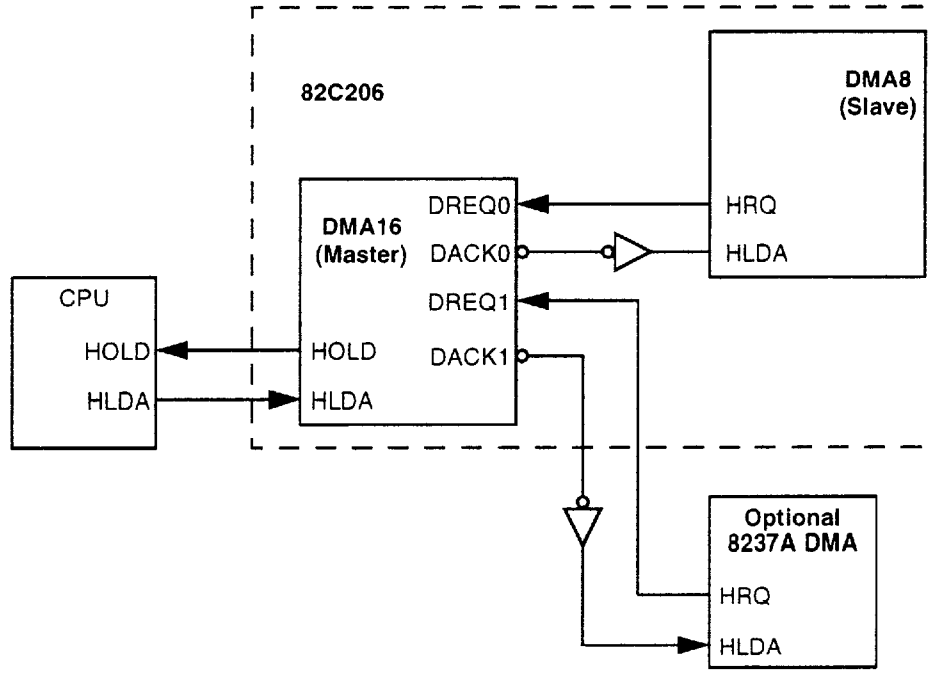
Verify Transfers are pseudo transfers. In this type of transfer, the DMA will operate as in Read or Write Transfers by generating HOLD, DACK, memory addresses and respond to the terminal count, but it does not activate the memory or I/O command signals. Since no transfer actually takes place, IOCHRDY is also ignored during Verify Transfers.

In addition to the three transfer types mentioned above, there is also a memory-to-memory transfer which can only be used on DMA Channels 0 and 1. The memory-to-memory transfer is used to move a block of memory from one location in memory to another. DMA Channels 0 and 1 may be programmed to operate as memory-to-memory channels by setting a bit in the DMA Command Register. Once programmed, the transfer can be started by generating either a software or an external request to Channel 0. During the transfer, Channel 0 provides the address for the source block during the memory write portion of the same transfer. During the read portion of the transfer, a byte of data is latched in the internal Temporary Register of the DMA. The contents of this register are then output on the SD[7:0] output pins during the write portion of the transfer and subsequently written to the memory location. Channel 0 may be programmed to maintain the same source address on every transfer. This allows the CPU to initialize large blocks of memory with the same value. The DMA subsystem will continue performing transfers until Channel 1 reaches the terminal count.

Auto-initialization

The Mode Register of each DMA channel contains a bit which will cause the channel to reinitialize after reaching the terminal count. During auto-initialization, the Base Address and Base Word Count Registers (which were originally programmed by the CPU) are reloaded into the Current Address and Current Word Count Registers. The Base Registers remain unchanged during DMA active cycles and can only be changed by the CPU. If the channel has been programmed to auto-initialize, the request mask bit will remain cleared upon reaching the terminal count. This allows the DMA to continue operation without CPU intervention. In memory-to-memory transfers, the Word Count Registers of Channels 0 and 1 must be programmed with the same starting value for full auto-initialization.

Figure 4-44 Cascade Mode Interconnect



DREQ Priority

The 82C206 supports two types of software programmable priority schemes: fixed and rotating. Fixed priority assigns priority based on channel position. With this method, Channel 0 is assigned the highest priority and Channel 3 is the lowest. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is completed.

In the rotating priority scheme, the ordering of priority from Channel 0 to Channel 3 is maintained, but the actual assignment of priority changes. The channel most recently serviced will be assigned the lowest priority and since the order of priority assignment remains fixed, the remaining three channels rotate accordingly. Table 4-49 shows the rotating priority scheme. In cases where multiple requests occur at the same time, the 82C206 will issue HOLD but will not freeze the priority logic until HLDA is returned. After HLDA becomes active, the priority logic is frozen and DACK is asserted on the highest requesting channel. Priority will not be reevaluated until HLDA has been deactivated.

Address Generation

During active cycles of the DMA, eight intermediate bits of the address are multiplexed onto the data lines. This reduces the number of pins required by the DMA subsystem. During an S1 state, the intermediate addresses are output on data lines SD[7:0]. These addresses should be externally latched and used to drive the system address bus. Since DMA8 is used for 8-bit transfers and DMA16 is used for 16-bit transfers, a one bit skew occurs in the intermediate address fields. DMA8 will therefore output address on LA[15:8] on the data bus at this time whereas DMA16 will output LA[16:9]. A separate set of latch and enable signals are provided for both DMA8 and DMA16 to accommodate the address skew.

During 8-bit DMA transfers in which DMA8 is active, the 82C206 will output the lower eight bits of address on SA[7:0]. LA[23:16] are also generated at this time from a DMA page register in the 82C206. Note that A16 is output on the A16 pin of the device.

During 16-bit DMA transfers in which DMA16 is active, the 82C206 will output the lower eight bits of address on SA[8:1]. LA[23:17] are also generated at this time from a DMA page register in the 82C206. Note that SA0 and LA16 remain tristated during 16-bit DMA transfers

The DMA page registers are a set of 16 8-bit registers in the 82C206 which are used to generate the high order addresses during DMA cycles. Only eight of the registers are actually used, but all 16 were included to maintain PC/AT compatibility. Each DMA channel has a page register associated with it except Channel 0 of DMA16 which is used for cascading to DMA8. Assignment of each of these registers is shown in Table 4-50 along with its CPU I/O read/write address.

During Demand and Block Transfers, the 82C206 generates multiple sequential transfers. For most of these transfers, the information in the external address latches will remain the same, thus eliminating the need to be relatched. Since the need to update the latches occurs only when a carry or borrow from the lower eight bits of the address counter exists, the 82C206 will only update the latch contents when necessary. The 82C206 execute an S1 state only when necessary and improve the overall system throughput.

Table 4-49 Rotating Priority Scheme

Priority	First Arbitration	Second Arbitration	Third Arbitration
Highest	Channel 0	Channel 2 - Cycle Grant	Channel 3 - Cycle Grant
	Channel 1 - Cycle Grant	Channel 3	Channel 0
	Channel 2	Channel 0	Channel 1
Lowest	Channel 3	Channel 1	Channel 2

Channel X = Requested Channel

Table 4-50 DMA Page Register I/O Address Map

I/O Addr	Type	Register Function
080h	R/W	Unused
081h	R/W	DMA8 Channel 2 (DACK2)
082h	R/W	DMA8 Channel 3 (DACK3)
083h	R/W	DMA8 Channel 1 (DACK1)
084h	R/W	Unused
085h	R/W	Unused
086h	R/W	Unused
087h	R/W	DMA8 Channel 0 (DACK0)
088h	R/W	Unused
089h	R/W	DMA16 Channel 2 (DACK6)
08Ah	R/W	DMA16 Channel 3 (DACK7)
08Bh	R/W	DMA16 Channel 1 (DACK5)
08Ch	R/W	Unused
08Dh	R/W	Unused
08Eh	R/W	Unused
08Fh	R/W	DRAM Refresh Cycle

Compressed Timing

The DMA subsystem in the 82C206 can be programmed to transfer a word in as few as two DMA clock cycles. Normal transfers require four DMA clock cycles since S3 is executed twice (due to the one wait state insertion). In systems capable of supporting higher throughput, the 82C206 can be programmed to omit one S3 and assert both commands in S2. S2 begins the cycle by generating the address and asserting both commands. One S3 cycle is executed and the cycle terminates in S4. If compressed timing is selected, TC will be output in S2 and S1 cycles which will be executed as necessary to update the address latch. Note that compressed timing is not allowed for memory-to-memory transfers.

4.12.3 DMA Register Descriptions

The following subsections are descriptions of the 82C206's internal peripherals controller DMA registers. The complete bit descriptions to these registers can be found in Section 5.0, Register Descriptions.

4.12.3.1 Current Address Register

Each DMA channel has a 16-bit Current Address Register which holds the address used during transfers. Each channel can be programmed to increment or decrement this register whenever a transfer is completed. This register can be read or written by the CPU in consecutive 8-bit bytes. If auto-initialization is selected, this register will be reloaded from the Base Address Register upon reaching the terminal count in the Current Word Count Register. Channel 0 can be prevented from incrementing or decrementing by setting the address hold bit in the Command Register.

4.12.3.2 Current Word Count Register

Each channel has a Current Word Count Register which determines the number of transfers. The actual number of transfers performed will be one greater than the value programmed into the register. The register is decremented after each transfer until it goes from 0 to FFFFh. When this rollover occurs, the 82C206 will generate TC and either suspend the operation on that channel and set the appropriate request mask bit, or auto-initialize and continue.

4.12.3.3 Base Address Register

Associated with each Current Address Register is a Base Address Register. This is a write-only register which is loaded by the CPU when writing to the Current Address Register. The purpose of this register is to store the initial value of the Current Address Register for auto-initialization. The contents of this register are loaded into the Current Address Register whenever the terminal count is reached and the auto-initialize bit is set.

4.12.3.4 Base Word Count Register

This register preserves the initial value of the Current Word Count Register. It too is a write-only register which is loaded by writing to the Current Word Count Register. The Base Word Count Register is loaded into the Current Word Count Register during auto-initialization.

4.12.3.5 Command Register

The Command Register controls the overall operation of the DMA subsystem. This register can be read or written by the CPU and is cleared by either a reset or master clear command.

4.12.3.6 Mode Register

Each DMA channel has a Mode Register associated with it. All four Mode Registers reside at the same I/O address. Bits 0 and 1 of the Write Mode Register command determine which channel Mode Register gets written. The remaining six bits control the mode of the selected channel. Each channel Mode Register can be read by sequentially reading the Mode Register location. A Clear Mode Register Counter command is provided to allow the CPU to restart the mode read process at a known point. During mode read operations, bit 0 and 1 will both equal 1.

4.12.3.7 Request Register

This 4-bit register is used to generate software requests (DMA service can be requested either externally or under software control). Request Register bits can be set or reset independently by the CPU. The register mask has no effect on software generated requests. All four bits are read in one operation and appear in the lower four bits of the byte. Bits 7 through 4 are read as 1s. All four request bits are cleared to 0 by a reset.

4.12.3.8 Request Mask Register

The Request Mask Register is a set of four bits which are used to inhibit external DMA requests from generating transfer cycles. This register can be programmed in two ways. Each channel can be independently masked by writing to the Write Single Mask bit location.

Alternatively, all four mask bits can be programmed in one operation by writing to the write all mask bits address.

All four mask bits are set following a reset or a Master Clear command. Individual channel mask bits will be set as a result of the terminal count being reached, if auto-initialize is disabled. The entire register can be cleared, enabling all four channels by performing a Clear Mask Register operation.

4.12.3.9 Status Register

The status of all four channels can be determined by reading the Status Register. Information is available to determine if a channel has reached the terminal count and whether an external service request is pending.

4.12.3.10 Temporary Register

The Temporary Register is used as a temporary holding register for data during memory-to-memory transfers. The register is loaded during the first cycle of a memory-to-memory transfer from SD[7:0]. During the second cycle of the transfer, the data in the Temporary Register is output on the SD[7:0] pins. Data from the last memory-to-memory; transfer will remain in the register.

4.12.4 Special Commands

Five special commands are provided to make the task of programming the 82C206 easier. These commands are activated as a result of a specific address and assertion of either IOR# or IOW#. For these special commands, the data bus is ignored by the 82C206 whenever an IOW# activated command is issued. Data returned on IOR# activated commands is undefined.

- Clear Byte Pointer Flip-Flop: This command is normally executed prior to reading or writing to the Address or Word Count Registers. This initializes the flip-flop to point to the low byte of the register and allows the CPU to read or write the register bytes in correct sequence.
- Set Byte Pointer Flip-Flop: Setting the byte pointer flip-flop allows the CPU to adjust the pointer to the high byte of an Address or Word Count Register.
- Master Clear: This command has the same effect as a hardware reset. The Command Register, Status Register, Request Register, Temporary Register, Mode Register counter, and byte pointer flip-flop are cleared and the Request Mask Register is set. Immediately following a Master Clear or reset, the DMA will be in the Idle mode.
- Clear Request Mask Register: This command enables all four DMA channels to accept requests by clearing the mask bits in the register.
- Clear Mode Register Counter: In order to allow access to the four Mode Registers while only using one address, an internal counter is used. After clearing the counter, all four Mode Registers may be read by successive reads to the Mode Register. The order in which the registers are read is Channel 0 first and Channel 3 last.

4.12.5 Interrupt Controller Subsystem

The programmable interrupt controllers in the 82C206 serve as a system wide interrupt manager. They accept requests from peripherals, resolve priority on pending interrupts and interrupts in service, issue an interrupt request to the CPU, and provide a vector which is used as an index by the CPU to determine which interrupt service routine to execute.

A variety of priority assignment modes are provided which can be reconfigured at any time during system operation. This allows the complete subsystem to be restructured based on the system environment.

4.12.5.1 Interrupt Controller Subsystem Overview

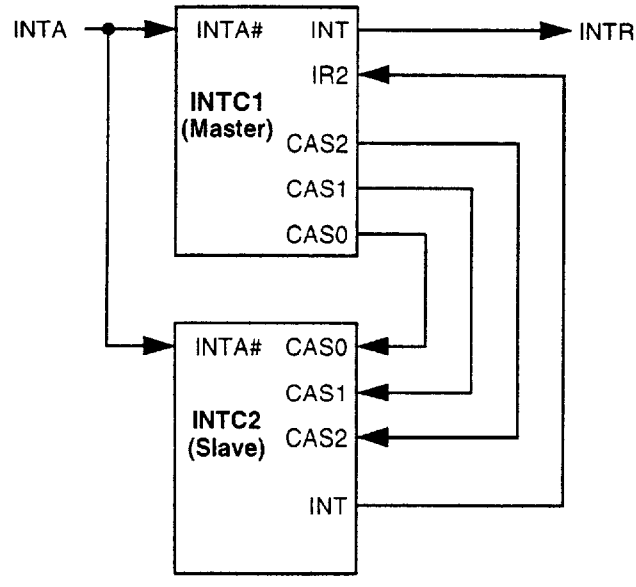
There are two interrupt controllers, INTC1 and INTC2, included in the 82C206. Each of the interrupt controllers is equivalent to an 8259A device operating in X86 mode. The two devices are interconnected and must be programmed to operate in the Cascade mode for all 16 interrupt channels to operate properly. Figure 4-45 shows the internal Cascade interconnection.

INTC1 is located at addresses 020h-021h and is configured for master operation in the Cascade mode. INTC2 is a slave device and is located at 0A0h-0A1h. The interrupt request output signal (INT) from INTC2 is internally connected to the interrupt request input Channel 2 (IR2) of INTC1. The address decoding and cascade interconnection matches that of the PC/AT.

Two additional interconnections are made to the interrupt request inputs of the interrupt controllers. The output of Timer 0 in the counter/timer subsystem is connected to Channel 0 (IR0) of INTC1. Interrupt request from the real-time clock is connected to Channel 0 (IR0) of INTC2. Table 4-51 lists the 16 interrupt channels and their interrupt request sources.

Description of the interrupt subsystem will pertain to both INTC1 and INTC2 unless otherwise noted. Wherever register addresses are used, the address for the INTC1 Register will be listed first and the address for the INTC2 Register will follow in parenthesis. Example: 02h (0A0h).

Figure 4-45 Internal Cascade Interconnect



Note: INTA will be active when the CPU initiates an interrupt acknowledge cycle.

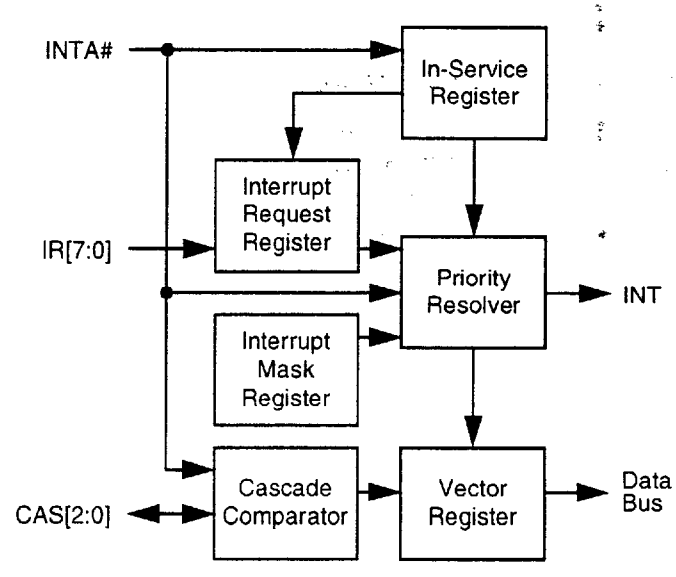
Table 4-51 Interrupt Request Source

Interrupt Controller	Channel Name	Interrupt Request Source
INTC1	IR0	Counter/Timer OUT0
INTC1	IR1	IRQ1 input pin
INTC1	IR2	INTC2 cascade interrupt
INTC1	IR3	IRQ3 input pin
INTC1	IR4	IRQ4 input pin
INTC1	IR5	IRQ5 input pin
INTC1	IR6	IRQ6 input pin
INTC1	IR7	IRQ7 input pin
INTC2	IR0	Real-time clock IRQ
INTC2	IR1	IRQ9 input pin
INTC2	IR2	IRQ10 input pin
INTC2	IR3	IRQ11 input pin
INTC2	IR4	IRQ12 input pin
INTC2	IR5	IRQ13 input pin
INTC2	IR6	IRQ14 input pin
INTC2	IR7	IRQ15 input pin

4.12.5.2 Interrupt Controller Operation

Figure 4-46 is a block diagram of the major components in the interrupt controller subsystem. The Interrupt Request Register (IRR) is used to store requests from all of the channels which are requesting service. The IRR's bits are labeled using the channel name IR[7:0]. The In-Service Register (ISR) contains all the channels which are currently being serviced (more than one channel can be in service at a time). The ISR's bits are labeled IS[7:0] and correspond to IR[7:0]. The Interrupt Mask Register (IMR) allows the CPU to disable any or all of the interrupt channels. The Priority Resolver evaluates inputs from the IRR, ISR, and IMR, issues an interrupt request, and latches the corresponding bit into the ISR. During interrupt acknowledge cycles, a master controller outputs a code to the slave device which is compared in the Cascade Buffer/Comparator with a 3-bit ID code previously written. If a match occurs in the slave controller, it will generate an interrupt vector. The contents of the Vector Register are used to provide the CPU with an interrupt vector during interrupt acknowledge (INTA) cycles.

Figure 4-46 Interrupt Controller Block Diagram

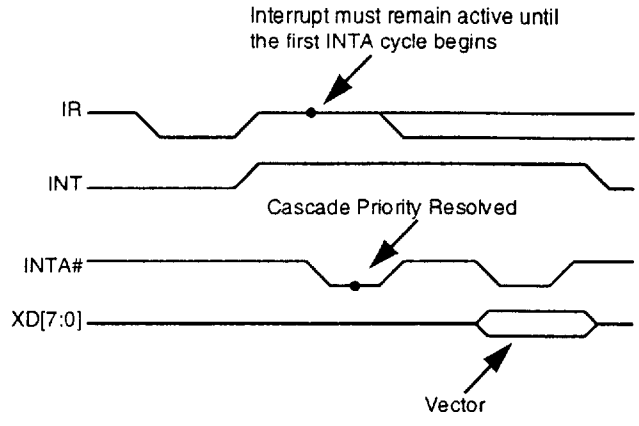


4.12.5.3 Interrupt Sequence

The 82C206 allows the CPU to perform an indirect jump to a service routine in response to a request for service in response to a request for service from as peripheral device. The indirect jump is based on a vector which is provided by the 82C206 on the second of two CPU generated INTA cycles (the first INTA cycle is used for resolving priority and the second is for transferring the vector to the CPU (see Figure 4-47). The events which occur during an interrupt sequence are as follows:

1. One or more of the interrupt requests (IR[7:0]) becomes active, setting the corresponding IRR bit(s).
2. The interrupt controller resolves priority based on the state of the IRR, IMR, and ISR and asserts the INTR output if needed.
3. The CPU accepts the interrupt and responds with an INTA cycle.
4. During the first INTA cycle, the highest priority ISR bit is set and the corresponding IRR bit is reset. The internal cascade address is generated.
5. The CPU will execute a second INTA cycle, during which the 82C206 will drive an 8-bit vector onto the data pins XD[7:0], which is read by the CPU. The format of this vector is shown in Table 4-52. Note that V[7:3] in Table 4-52 are programmable by writing to ICW2 (Initialization Command Word 2).
6. At the end of the second INTA cycle, the ISR bit will be cleared if the Automatic End of Interrupt Mode is selected (see below). Otherwise, the ISR bit must be cleared by an End of Interrupt (EOI) command from the CPU at the end of the interrupt service routine to allow further interrupts. If no interrupt request is present at the beginning of the first INTA cycle (i.e., a spurious interrupt), INTC1 will issue an interrupt level 7 vector during the second INTA cycle.

Figure 4-47 Interrupt Sequence



4.12.5.4 End of Interrupt (EOI)

EOI is defined as the condition which causes an ISR bit to be reset. Determination of which ISR bit is to be reset can be done by a CPU command (specific EOI) or the Priority Resolver can be instructed to clear the highest priority ISR bit (nonspecific EOI). The 82C206 can determine the correct ISR bit to reset when operated in modes which do not alter the fully nested structure since the current highest priority ISR bit is the last level acknowledged and serviced. In conditions where the fully nested structure is not preserved, a specific EOI must be generated at the end of the interrupt service routine. An ISR bit that is masked, in the Special Mask Mode by an IMR bit, will not be cleared by a nonspecific EIO command. The interrupt controller can optionally generate an Automatic End of Interrupt (AEOI) on the trailing edge of the second INTA cycle.

Table 4-52 Interrupt Vector Byte

Interrupt	D7	D6	D5	D4	D3	D2	D1	D0
IR7	V7	V6	V5	V4	V3	1	1	1
IR6	V7	V6	V5	V4	V3	1	1	0
IR5	V7	V6	V5	V4	V3	1	0	1
IR4	V7	V6	V5	V4	V3	1	0	0
IR3	V7	V6	V5	V4	V3	0	1	1
IR2	V7	V6	V5	V4	V3	0	1	0
IR1	V7	V6	V5	V4	V3	0	0	1
IR0	V7	V6	V5	V4	V3	0	0	0

4.12.5.5 Priority Assignment

Assignment of priority is based on an interrupt channel's position relative to the other channels in the interrupt controller. After the initialization sequence, IR0 has the highest priority, IR7 the lowest, and priority assignment is Fixed. Priority assignment can be rotated either manually (Specific Rotation Mode) or automatically (Automatic Rotation Mode) by programming Operational Command Word 2 (OCW2).

Fixed Priority Mode

This is the default condition which exists unless rotation (either manual or automatic) is enabled, or the controller is programmed for Polled Mode. In the Fixed Priority Mode, interrupts are fully nested with priority assigned as shown:

	Lowest							Highest
Priority Status	7	6	5	4	3	2	1	0

Nesting allows interrupts of a higher priority to generate interrupt requests prior to the completion of the interrupt service. When an interrupt is acknowledged, priority is resolved, the highest priority request's vector is placed on the bus, and the ISR bit for that channel is set. This bit remains set until an EIO (automatic or CPU generated) is issued to that channel. While the ISWR bit is set, all interrupts of equal or lower priority are inhibited. Note that a higher priority service routine will only be acknowledged if the CPU has internally re-enabled interrupts.

Specific Rotation Mode

Specific Rotation allows the system software to reassign priority levels by issuing a command which redefines the highest priority channel. Before rotation:

	Lowest							Highest
Priority Status	7	6	5	4	3	2	1	0

(Specific Rotation command issued with Channel 5 specified.) After rotation:

	Lowest							Highest
Priority Status	5	4	3	2	1	0	7	6

Automatic Rotation Mode

In applications where a number of equal priority peripherals are requesting interrupts, Automatic Rotation may be used to equalize the priority assignment. In this mode, after a peripheral is serviced it is assigned the lowest priority. All peripherals connected to the controller will be serviced at least once in eight interrupt requests to the CPU from the controller. Automatic Rotation will occur, if enabled, due to the occurrence of an EOI (automatic or CPU generated).

Before rotation (IR3 is the highest priority request being serviced):

ISR Status Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
	1	1	0	0	1	0	0	0
	Lowest							Highest
Priority Status	7	6	5	4	3	2	1	0

(Specific Rotation command issued with Channel 4 specified.) After rotation:

ISR Status Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
	1	1	0	0	0	0	0	0
	Lowest							Highest
Priority Status	3	2	1	0	7	6	5	4

4.12.5.6 Programming the Interrupt Controller

Two types of commands are used to control the 82C206's interrupt controllers: Initialization Command Words (ICWs) and Operational Command Words (OCWs).

Initialization Command Words (ICWs)

The initialization process consists of writing a sequence of four bytes to each interrupt controller. The initialization sequence is started by writing the first Initialization Command Word (ICW1) to address 020h (0A0h) with a 1 on bit 4 of the data byte. The interrupt controller interprets this as the start of an initialization sequence and does the following:

1) The Initialization Command Word Counter is reset to 0.
2) ICW1 is latched into the device.
3) Fixed Priority Mode is selected.
4) IR0 is assigned the highest priority.
5) The Interrupt Mask Register is cleared.
6) The Slave Mode Address is set to 7.
7) Special Mask Mode is disabled.
8) IRR is selected for status read operations.

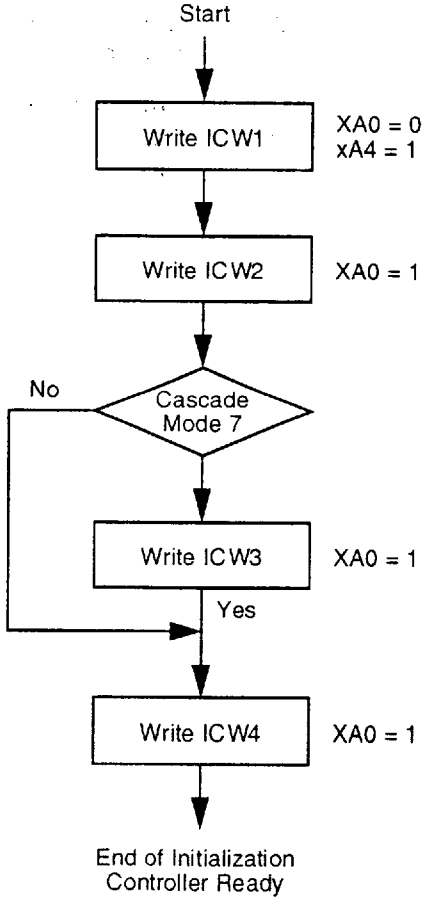
The next three I/O writes to address 021h (0A1h) will load ICW2 through ICW4. See Figure 4-48 for a flow chart of the initialization sequence. The initialization sequence can be terminated at any point (all four bytes must be written for the controller to be properly initialized) by writing to address 020h (0A0h) with a 0 in data bit 4. Note this will cause OCW2 or OCW3 to be written.

Operational Command Words (OCWs)
Operational Command Words (OCWs) allow the 82C206's interrupt controllers to be controlled or reconfigured at any time while operating. Each interrupt has three OCWs which can be programmed to affect the proper operating configuration and a status register to monitor controller operation.

OCW1 is located at address 021h (0A1h) and may be written any time the controller is not in the Initialization Mode. OCW2 and OCW3 are located at address 020h (0A0h). Writing to address 020h (0A0h) with a 0 in bit 4 will place the controller in the operating mode and load OCW2 (if data bit 3 = 0) or OCW3 (if data bit 3 = 1).

IRR, ISR, & Poll Vector
IRR, ISR, and Poll Vector are the same address, 020h (0A0h). The selection of the registers depends on the programming of ITC. If the latest OCW3 issued the poll command (PM = 1), the poll vector is selected for the next read. Before another poll command is issued, subsequent reads to the address will select IRR or ISR depending on the latest OCW3, if RR = 1 and RIS = 0, ISR is selected. Note that the poll command is cleared after the first read to the ITC. After initialization (ICW1 or reset), IRR is selected.

Figure 4-48 Initialization Sequence



4.12.6 Counter/Timer Subsystem

The 82C206 contains an 8254 compatible counter/timer. The counter/timer can be used to generate accurate time delays under software control. It contains three 16-bit counters (Counters 2 through 0) which can be programmed to count in binary or binary-coded decimal (BCD). Each counter operates independently of the other and can be programmed for operation as a timer or a counter.

All counters in this subsystem are controlled by a common control logic as shown in Figure 4-49. The control logic decodes and generates the necessary commands to load, read, configure, and control each counter. Counter 0 and Counter 1 can be programmed for all six modes, but Mode 1 and Mode 5 have limited usefulness because their gate is hard-wired to GND internally. Counter 2 can be programmed to operate in any of the six modes:

- Mode 0 - Interrupt on terminal count
- Mode 1 - Hardware retriggerable one-shot
- Mode 2 - Rate generator
- Mode 3 - Square wave generator
- Mode 4 - Software triggered strobe
- Mode 5 - Hardware retriggerable strobe

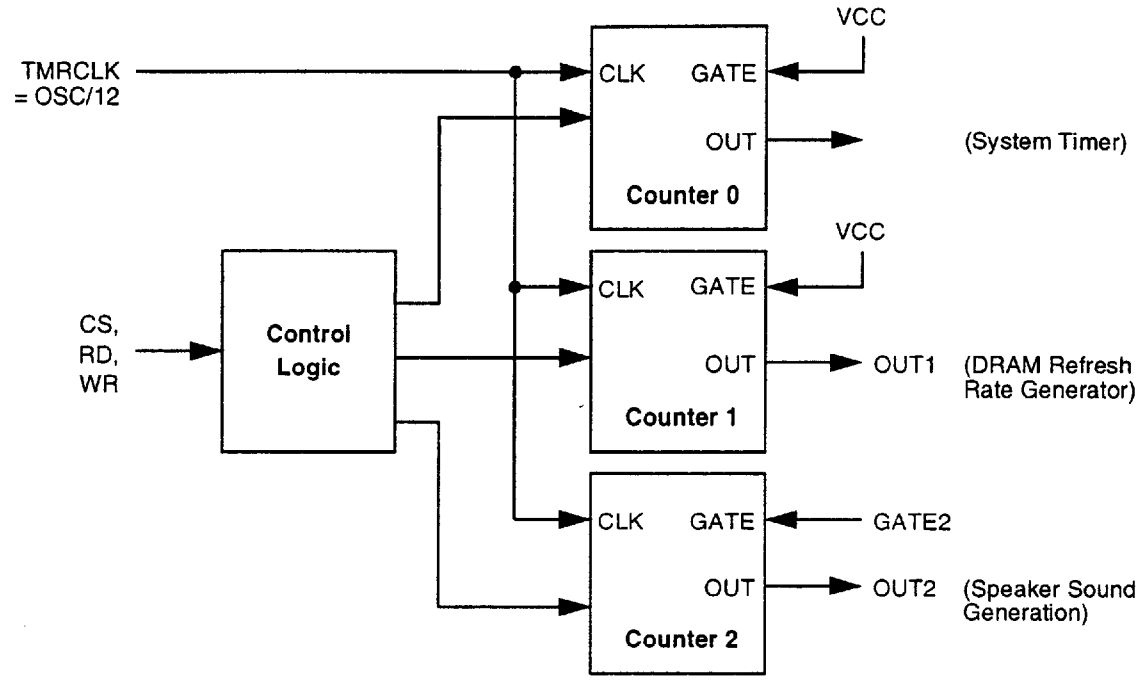
The internal timer counter use an internal signal TMRCLK which is derived from the OSC input of the 82C206. For the sake of simplicity, all references to the timer counter clock will be TMRCLK in the following description. All three counters are driven from a common clock input, TMRCLK (TMRCLK = OSC/12) Counter 0's output (OUT0) is internally connected to IRQ of INTC1 and is used as an interrupt to the system for time keeping and task switching. Counter 1 may be programmed to generate pulses or square waves for external devices. Counter 2 is a full function counter/timer. It can be used as an interval timer, a counter, or as a gated rate/pulse generator. In a PC/AT compatible design, Counter 0 is used as a system timer, Counter 1 is used as a DRAM refresh rate generator, and Counter 2 is used for speaker sound generation.

4.12.6.1 Counter Description

Each counter in this subsystem contains a control register, a status register, a 16-bit counting component, a pair of 8-bit counter input latches, and a pair of 8-bit counter output latches. Each counter shares the same clock input (TMRCLK). GATE0, GATE1, and OUT0 are not externally accessible. This is fully compatible with a PC/AT-based design. Output of OUT0 is dependent on the counter mode.

The control register stores the mode and command information used to control the counter. It may be loaded by writing a byte to the write control word at Port 043h. The status register allows the software to monitor counter conditions and read back the contents of the control register.

Figure 4-49 Counter/Timer Block Diagram



The 16-bit counting component is a loadable synchronous down counter. It is loaded or decremented on the falling edge of TMRCLK. The counting component contains a maximum count when a 0 is loaded, which is equivalent to 65536 in binary operation or 1000 in BCD. The counting component does not stop when it reaches 0. In Modes 2 and 3, the counting component will be reloaded and in all other modes it will wrap around to 0FFFFh in binary operation or 9999 in BCD.

The counting component is indirectly loaded by writing one or two bytes (optional) to the counter input latches, which are in turn loaded into the counting component. Thus, the counting component can be loaded or reloaded in one TMRCLK cycle. The counting component is also read indirectly by reading the contents of the counter output latches. The counter output latches are transparent latches which can be read while transparent or latched (see Latch Counter Command).

Programming the Counter/Timer

After a system reset, the contents of the control registers, counter registers, counting components, and the output of all counters are undefined. Each counter must be programmed before it can be used. Each counter is programmed by writing its control register with a control word and then giving an initial count to its counting component. Table 4-53 lists the I/O address map used by the counter/timer subsystem.

Table 4-53 Counter/Timer I/O Address Map

Address	Function
040h	Counter 0 read/write
041h	Counter 1 read/write
042h	Counter 2 read/write
043h	Control register write only

Read/Write Counter Command

Each counter has a write only control register. This control register is written with a control word to the I/O address 043h.

When programming to a counter, the following steps must sequentially occur:

1) Each counter's control register must be written with a control word before the initial count is written.

2) Writing the initial count must follow the format specified in the control word (least significant bit only, most significant bit only, or least significant bit and then most significant bit.

A new initial count can be written into the counter at any time after programming without rewriting the control word.

Counter Latch Command
When a counter latch command is issued, the counter's output latches latch the current state of the counting component. The counter's output latches remain latched until read by the CPU or the counter is reprogrammed. After that, the output latches then returns to a "transparent" condition. Counter latch commands may be issued to more than one counter before reading the first counter to which this command was issued. Also, multiple counter latch commands issued to the same counter without reading the counter will cause all but the first command to be ignored.

Read-Back Command
The read-back command allows the user to check the count value, mode, and state of the OUT signal and null count flag of the selected counter(s).

Each counter's latches remain latched until either the latch is read or the counter is reprogrammed. If both LSTATUS and LCOUNT are 0, the status will be returned on the next read from the counter. The next one or two reads (depending on whether the counter is programmed to transfer one or two bytes) from the counter result in the count being returned. Multiple read-back commands issued to the same counter without reading the counter will cause all but the first command to be ignored.

4.12.6.2 Counter Operation

Since Counter 1 and 0 have limitations in some of their operation modes, Counter 2 will be used to describe the various counter operating modes. However, the description of Modes 0, 2, 3, and 4 are suitable for all counters. The following terms are defined for describing the counter/timer operation.

- TMRCLK pulse - A rising edge followed by a falling edge of the 82C206's TMRCLK (0SC/12).
- Trigger - The rising edge of the GATE2 input.
- Counter Load - the transfer of the 16-bit value in counter input latches to the counting element.
- Initialized - A control word written and the counter input latches loaded.
- Counter 2 can operate in one of the following modes:
  - Mode 0 - Interrupt on terminal count
  - Mode 1 - Hardware retriggerable one-shot
  - Mode 2 - Rate generator
  - Mode 3 - Square wave generator
  - Mode 4 - Software triggered strobe
  - Mode 5 - Hardware triggered strobe

Mode 0 - Interrupt on Terminal Count
Mode 0 is usually used for event counting. After a counter is written with the control word, OUT2 of that counter goes low and remains low until the counting element reaches 0, at which time it goes back high and remains high until a new count or control word is written. Counting is enabled when GATE2 = 1 and disabled when GATE2 = 0. GATE2 has no effect on OUT2.

The counting component is loaded at the first TMRCLK pulse after the control word and initial count are loaded. When both initial count bytes are required, the counting component is loaded after the high byte is written. This TMRCLK pulse does not decrement the count, so for an initial count of N, OUT2 does not go high until (N + 1) TMRCLK pulses after initialization. Writing a new initial count to the counter reloads the counting element on the next TMRCLK pulse and counting continues from the new count. If an initial count is written with GATE2 = 0, it will still be loaded on the next TMRCLK pulse. But counting does not progress until GATE2 = 1. When GATE2 goes high, OUT2 will go high after N TMRCLK pulses later.

Mode 1 - Hardware Retriggerable One-Shot
Writing the control word causes OUT2 to go high initially. Once initialized, the counter is armed and a trigger causes OUT2 to go low on the next TMRCLK pulse. OUT2 then remains low until the counter reaches 0. An initial count of N results in a one-shot pulse N TMRCLK cycles long. Any subsequent triggers while OUT2 is low cause the counting component to be reloaded, extending the length of the pulse. Writing a new count to the counter input latches will not affect the current one-shot pulse unless the counter is retriggered. In the latter case, the counting component is loaded with the new count and the one-shot pulse continues until the new count expires.

Mode 2 - Rate Generator
This mode functions as a divide-by-N counter. After writing the control word during initialization, the counter's OUT2 is set to high. When the initial count is decremented to 1, OUT2 goes low on the next TMRCLK pulse. The following TMRCLK pulse returns OUT2 high, reloads the CE, and the process is repeated. In Mode 2, the counter continues counting (if GATE2 = 1) and will generate an OUT2 pulse every N TMRCLK cycles. Note that a count of 1 is illegal in Mode 2.

GATE2 = 0 disables counting and forces OUT2 high immediately. A trigger reloads the CE on the next TMRCLK pulse. Thus, GATE 2 can be used to synchronize the counter to external events.

Writing a new count while counting does not affect current operation unless a trigger is received. Otherwise, the new count will be loaded at the end of the current counting cycle.

Mode 3 - Square Wave Generator
Mode 3 is similar to Mode 2 in every respect except for the duty cycle of OUT2. OUT2 is set high initially and remains high for the first half of the count. When the first half of the initial count expires, OUT2 goes low for the remainder of the count.

If the counter is loaded with an even count, the duty cycle of OUT2 will be 50% (high = low = N/2). For odd count values,

OUT2 is high one TMRCLK cycle longer than it is low. Therefore, high = (N + 1)/2 and low = (N - 1)/2.

Mode 4 - Software Triggered Strobe
Writing the Control Word causes OUT2 To go high initially. Expiration of the initial count causes OUT2 to go low for one TMRCLK cycle. GATE2 = 0 disables counting but has no effect on OUT2. Also, a trigger will not reload CE.

The counting sequence is started by writing the initial count. The CE is loaded on the TMRCLK pulse after initialization. The CE begins decrementing one TMRCLK pulse later, OUT2 will go low for one TMRCLK cycle, (N + 1) cycles after the initial count is written.

If a new initial count is written during a counting sequence, it is loaded into the CE on the next TMRCLK pulse and the sequence continues from the new count. This allows the sequence to be "retriggerable" by software.

Mode 5 - Hardware Triggered Strobe
Writing the Control Word causes OUT2 to go high initially. Counting is started by a trigger. The expiration of the initial count causes OUT2 to go low for one TMRCLK cycle. GATE2 = 0 disables counting.

The CE is loaded on the TMRCLK pulse after a trigger. Since loading the CE inhibits decrementing, OUT2 will go low for one TMRCLK cycle, (N + 1) TMRCLK cycles after the trigger.

If a new count is loaded during counting, the current counting sequence will not be affected unless a trigger occurs. A trigger causes the counter to be reloaded from CIL and CIH making the counter "retriggerable".

GATE2
In Modes 0, 2, 3, and 4 GATE2 is level-edge sensitive and is sampled on the rising edge of TMRCLK. In Modes 1, 2, 3, and 5 the GATE2 input is rising-edge sensitive. This rising edge sets an internal flip-flop whose output is sampled on the rising edge of TMRCLK. The flip-flop resets immediately after being sampled. Note that in Modes 2 and 3, the GATE2 input is both edge and level sensitive. Table 4-54 details this operation.

Table 4-54 GATE2 Pin Function

Mode	GATE2		
	Low	Rising	High
0	Disables counting		Enables counting
		A) Initiates counting B) Reset OUT2 pin	
2	A) Disables counting B) Forces OUT2 pin high	Initiates counting	Enables counting
3	A) Disables counting B) Forces OUT2 pin high	Initiates counting	Enables counting
4	Disables counting		Enables counting
5		Initiates counting	

4.13 Serial IRQ Interface

The serial IRQ is a mechanism for communicating IRQ status between, ISA legacy components, PCI components, and PCI system controllers. The IRQ/data serializer is a wired OR structure that passes the state of one or more device's IRQ(s) and/or data to the host controller (82C568). A transfer, called a serial IRQ cycle, consists of three frame types: one start frame, several IRQ/data frames, and one stop frame. This protocol uses the PCI clock as its clock source and conforms to the PCI bus electrical specification. Pin 1 (SERIRQ#) is used to implement this interface.

4.13.1 Timing Diagrams For Serial IRQ Cycle

Figure 4-50 through Figure 4-53 are timing diagrams for various serial interrupt cycles.

Figure 4-50 Serial Interrupt - Time Slots

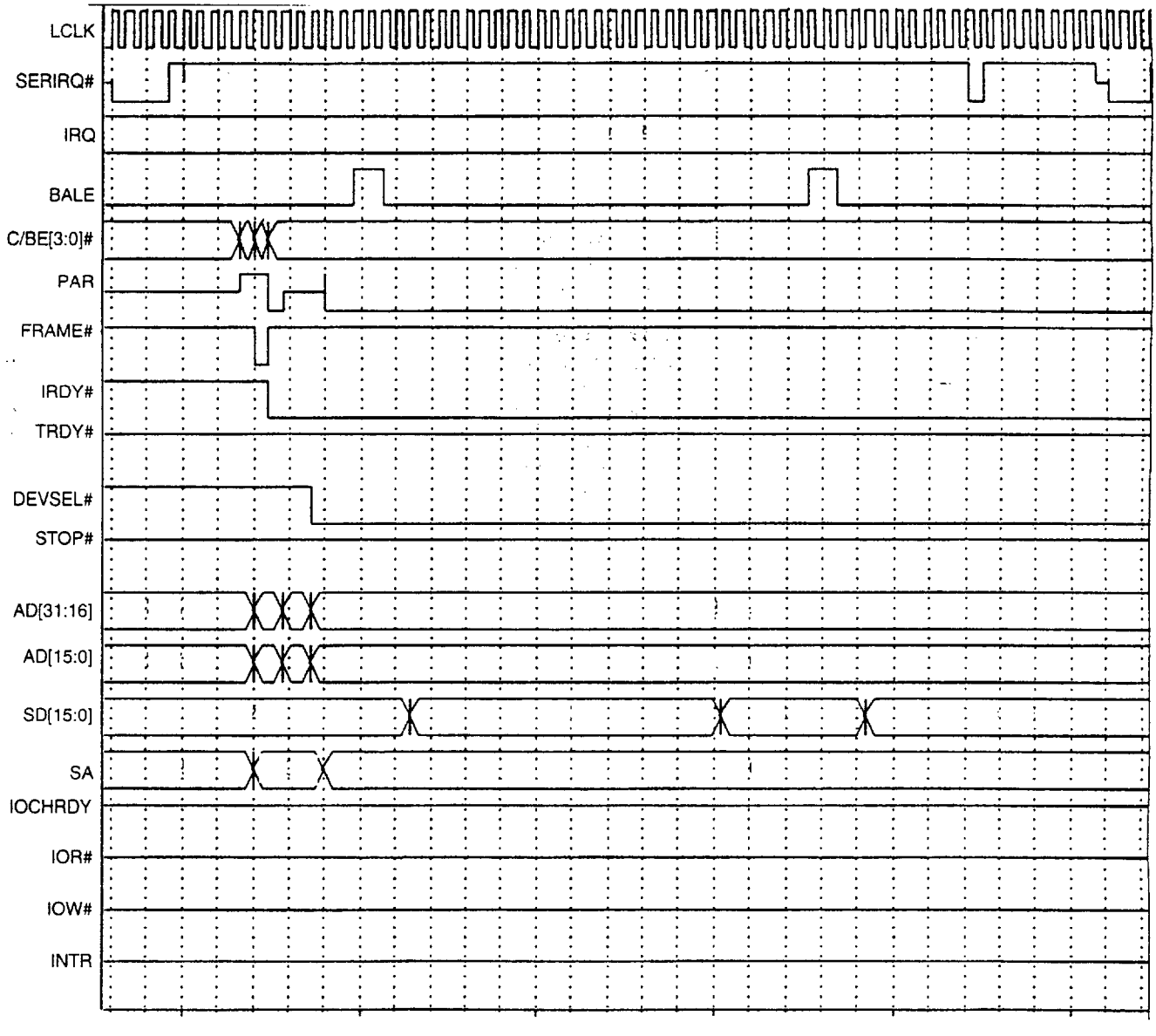


Figure 4-51 Serial Interrupt - Acknowledge Cycle

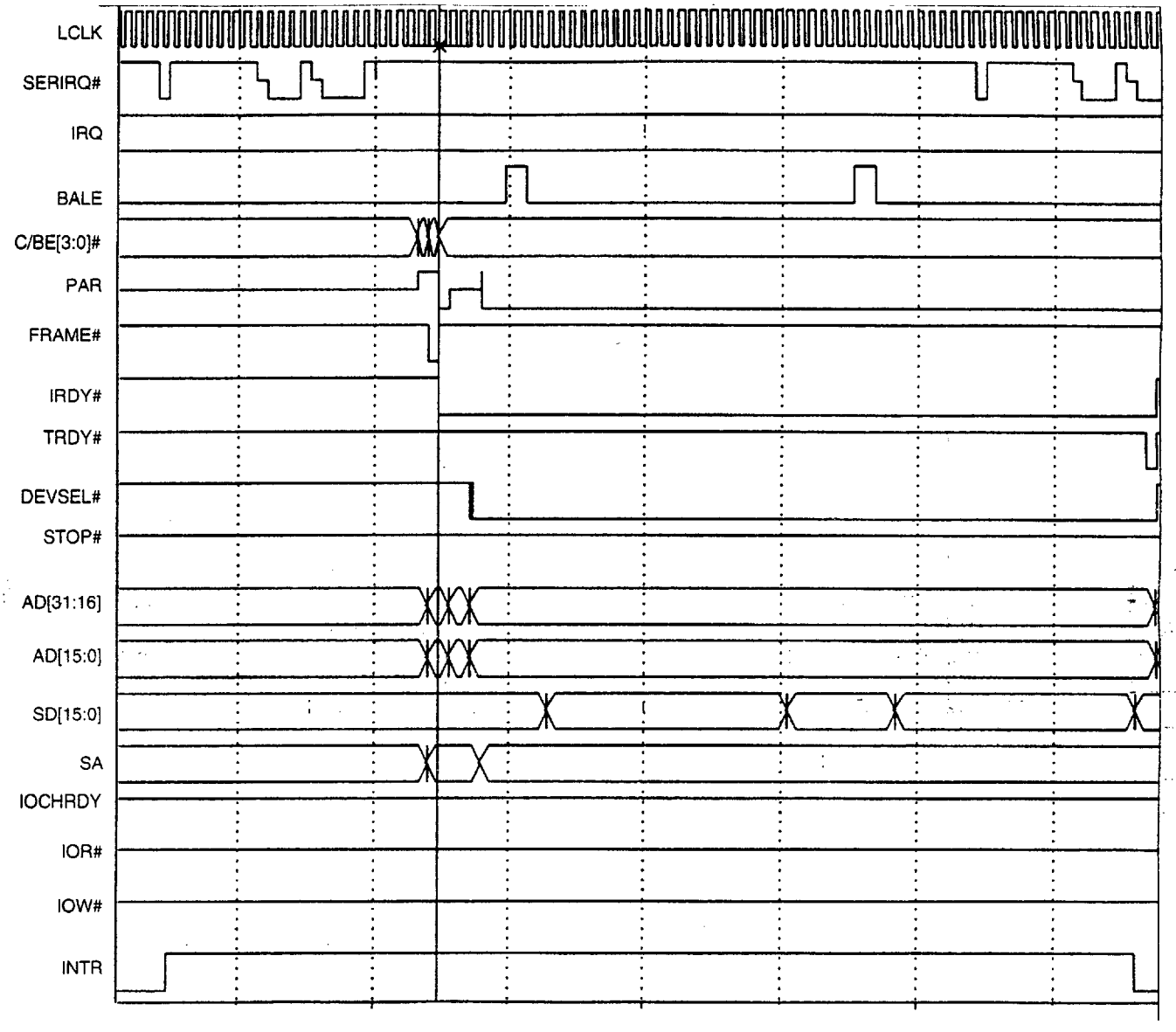


Figure 4-52 Serial Interrupt - EOI Latency Handling

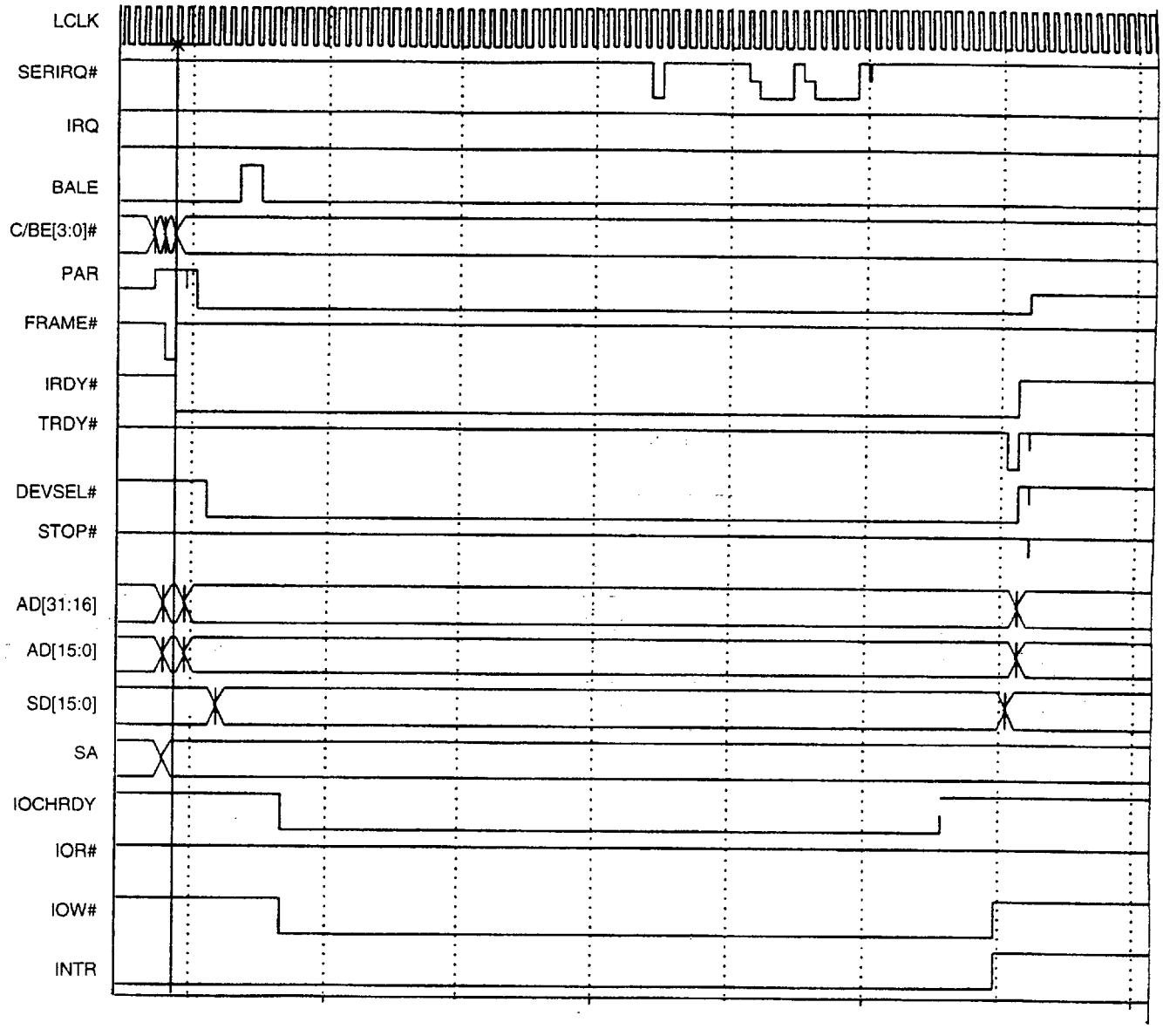
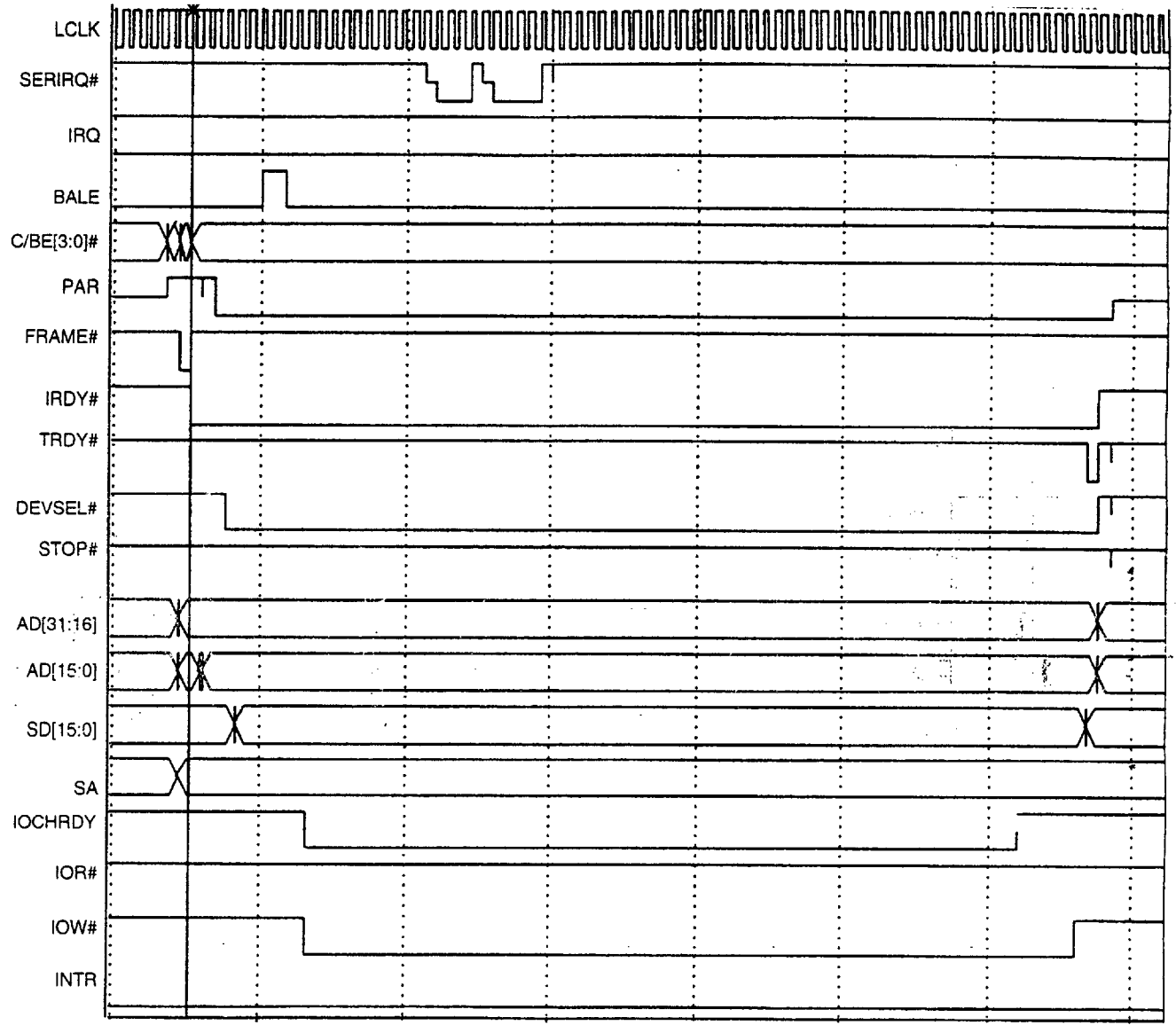


Figure 4-53 Serial Interrupt - ISR Latency Handling



4.13.1.1 Start and Stop Frame Timing

- Start frame pulses can be four, six, or eight clocks wide.
- Stop pulses are two clocks wide for Quiet mode, three clocks wide for Continuous mode.
- There may be none, one or more Idle states during the stop frames.
- The next SERIRQ# cycle's start frame pulse may not start immediately after the turnaround clock of the stop frame.

Refer to Figure 4-54.

4.13.2 Serial IRQ Cycle Control

There are two modes of operation for the serial IRQ start frame - the Quiet (Active) mode and the Continuous mode.

4.13.2.1 Quiet (Active) Mode

Any device may initiate a start frame by driving SERIRQ# low for one clock, while SERIRQ# is Idle. After driving low for one clock, SERIRQ# must immediately be tristated without driving high at any time. A start frame may not be initiated while SERIRQ# is Active. SERIRQ# is Idle between stop and start frames. SERIRQ# is Active between start and stop frames. This mode of operation allows SERIRQ# to be Idle when there are no IRQ/data transitions (which should be most of the time).

Once a start frame has been initiated, the 82C568 will take over driving SERIRQ# low in the next clock and will continue driving it low for a programmable period of three to seven clocks more. This makes a total low pulse width of four to eight clocks. Finally, the 82C568 will drive SERIRQ# back high for one clock, then tristate.

Any serial IRQ device which detects any transition on an SERIRQ# line for which it is responsible, must initiate a start frame in order to update the 82C568 unless SERIRQ# is already in an serial IRQ cycle and the IRQ/data transition can be delivered in that serial IRQ cycle.

4.13.2.2 Continuous (Idle) Mode

Only the 82C568 can initiate a start frame to update IRQ/data line information. All other SERIRQ# agents become passive and may not initiate a start frame. SERIRQ# will be driven low for four, six, or eight clocks by the 82C568. This mode has two functions:

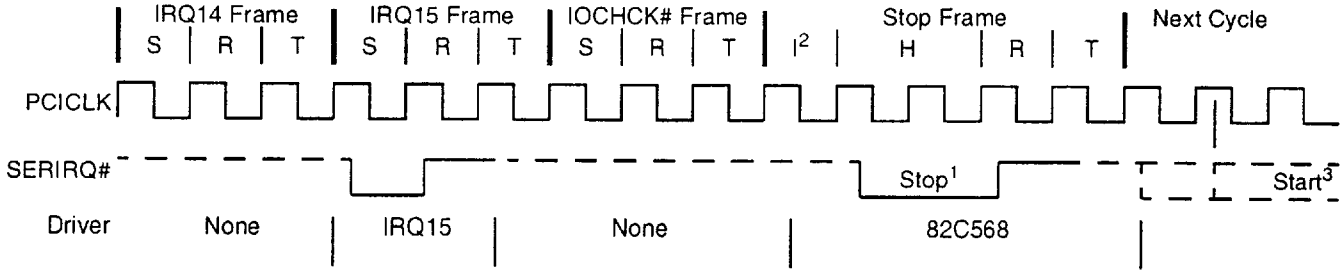
1) It can be used to stop or idle SERIRQ# or,
2 the 82C568 can operate SERIRQ# in a Continuous mode by initiating a start frame at the end of every stop frame.

An serial IRQ mode transition can only occur during the stop frame. Upon reset, the SERIRQ# bus is defaulted to Continuous mode, therefore only the 82C568 can initiate the first start frame. Slaves must continuously sample the stop frame's pulse width to determine the next serial IRQ cycle's mode.

4.13.3 IRQ/Data Input Detection

To assure that SERIRQ# does not miss narrow IRQ/data input pulses, it is recommended that serial IRQ devices detect IRQ/data input through a low level extender. Serial IRQ devices should detect any high-to-low transition on an IRQ/data input and keep it latched until this transition is successfully transmitted to the 82C568 through the SERIRQ# bus.

Figure 4-54 Start/Stop Frame Timing



Note: H = Host Control
I = Idle
R = Recovery
S = Sample
T = Turnaround

4.13.4 IRQ/Data Frame

Once a start frame has been initiated, all serial IRQ devices must watch for the rising edge of the start pulse and start counting IRQ/data frames from there. Each IRQ/data frame is three clocks:

- Sample phase
- Recovery phase
- Turnaround phase

During the Sample phase, the serial IRQ devices must drive the SERIRQ# low, if and only if, its last detected IRQ/data was low. If its detected IRQ/data value is high, SERIRQ# must be left tristated.

During the Recovery phase, the serial IRQ devices must drive SERIRQ# high, if and only if, it had driven SERIRQ# low during the previous Sample phase.

During the Turnaround phase, all serial IRQ devices must be tristated. All serial IRQ devices must drive SERIRQ# low at the appropriate sample point if its associated IRQ/data line is low, regardless of which device initiated the start frame.

The Sample phase for each IRQ/data follows the low-to-high transition of the start frame pulse by a number of clocks equal to the IRQ/data frame times three, minus one, (e.g., the IRQ5 Sample clock is the sixth IRQ/data frame, (6x3)-1=17th clock after rising edge of the start pulse.)

At the end of each Sample phase, the 82C568 will sample the state of the SERIRQ# line and replicate the status of the original IRQ/data line at the input to its interrupt controller. Slot #21-18 (INTB#/INTC#/INTD#) is optional (selected through register programming, refer to Table 4-56). Table 4-55 highlights the sampling period for each IRQ line.

4.13.5 Stop Cycle Control

Once all IRQ/data frames have completed, the 82C568 will terminate SERIRQ# activity by initiating a stop frame. Only the 82C568 can initiate the stop frame. A stop frame is indicated when SERIRQ# is low for two or three clocks.

If the stop frame's low time is two clocks, then the next SERIRQ# cycle's sampled mode is the Quiet mode; and any serial IRQ device may initiate a start frame in the third clock or more after the rising edge of the stop frame's pulse.

If the stop frame's low time is three clocks, then the next serial IRQ cycle's sampled mode is the Continuous mode; and only the 82C568 may initiate a start frame in the third clock or more after the rising edge of the stop frame's pulse.

Table 4-55 SERIRQ# Sampling Periods

IRQ/Data Frame	Signal Sampled	# of Clocks Past Start
1	IRQ0	2
2	IRQ1	5
3	SMI#	8
4	IRQ3	11
5	IRQ4	14
6	IRQ5	17
7	IRQ6	20
8	IRQ7	23
9	IRQ8	26
10	IRQ9	29
11	IRQ10	32
12	IRQ11	35
13	IRQ12	38
14	IRQ13	41
15	IRQ14	44
16	IRQ15	47
17	IOCHCK#	50
18	INTA#	53
19	INTB#	56
20	INTC#	59
21	INTD#	62

4.13.5.1 Serial IRQ Cycle Maximum Sampling Period

Based on the maximum of 21 sampling slot assignments, plus start/stop pulse width variations, the longest SERIRQ# cycle is approximately 70 PCI clocks. There are 17 IRQ/data frames defined for IRQs, SMI#, and IOCHCK#, and there are four optional IRQ/data frames, 18 through 21.

4.13.6 Latency

Latency for IRQ/data updates over the SERIRQ# signal in bridge-less systems with the minimum for IRQ/data frames of 17, will range up to 96 clocks.

4.13.6.1 EOI/ISR Read Latency

Any serialized IRQ scheme has a potential implementation issue related to IRQ latency. IRQ latency could cause an EOI or ISR read to precede an IRQ transition that it should have followed. This could cause a system fault. The 82C568 will ensure that these latency issues are well covered. By delaying EOIs and ISR reads to the interrupt controller by the same amount as the serial IRQ cycle latency in order to ensure that these events do not occur out of order.

Table 4-56 Serial Interrupt Control Related Registers

7	6	5	4	3	2	1	0
PCIDV1 55h			**PCI Master Control Register - High Byte**				**Default = 00h**
Reserved: Must be written to 0.			SERIRQ# muxing on Pin 1: 0 =Disable 1 =Enable(1)	Interrupt request register recover: 0 = Disable 1 = Enable	Select DMA current or base address and counter to be read: 0 = Current 1 = Base	ISA retry for CPU/PCI master access ISA cycle: 0 = Disable 1 = Enable	Use of AHOLD signal during CPU-to-PCI cycles:(2) 0 = Disable 1 = Enable
(1) Also these PCIDV1 register bits must be set: 54h[4] = 1, 59h[3] = 0, and 5Fh[4] = 0. (2) Bit 0 is used only if PCIDV1 54h[4] = 1.							
PCIDV1 56h			**Serial Interrupt Source Register - Low Byte**				**Default = 00h**
Interrupt resource for IRQ Ch. 7: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 6: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 5: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 4: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 3: 0 = ISA 1 = Serial interrupt	Interrupt resources for SMI#, IOCHK#, PCIRQ[3:0]#: 0 = Original 1 = Serial interrupt	Interrupt resource for IRQ Ch. 1: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 0: 0 = ISA 1 = Serial interrupt
PCIDV1 57h			**Serial Interrupt Source Register - High Byte**				**Default = 00h**
Interrupt resource for IRQ Ch. 15: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 14: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 13: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 12: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 11: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 10: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 9: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 8: 0 = ISA 1 = Serial interrupt
PCIDV1 58h			**Serial Interrupt Mode Control Register**				**Default = 00h**
Serial interrupt control mode: 00 = Continuous mode 01 = Idle mode 1x = Active mode		Reserved: Must be written to 0.	Data frame slot support:(1) 0 = 17 slots 1 = 21 slots	Reserved: Must be written to 0.		Serial interrupt start frame pulse width control: 00 = 4 CLK in Continuous mode or 3 CLK in Active mode 01 = 6 CLK in Continuous mode or 5 CLK in Active mode 10 = 8 CLK in Continuous mode or 7 CLK in Active mode 11 = Reserved	
(1) 17 slot support: IRQ[1:0], IRQ[15:3], SMI#, and IOCHK# 21 slot support: All of the above plus PCIRQ[3:0]#							

4.14 Distributed DMA

Distributed DMA provides a facility to support legacy DMA by two or more devices on the PCI bus. The distributed DMA (DDMA) master interface in the 82C568 snoops on any PCI master I/O access to the legacy DMA I/O space and claims and internally executes any such cycle. The DDMA module will gain control of the PCI bus and execute the required remapped cycles. Once a cycle is posted inside the 82C568, the cycle is posted with a retry response on the PCI bus till such time DDMA operation is complete. All remapped PCI cycles are executed as single byte cycles by taking the multi-byte PCI master I/O access internally and breaking it into proper single byte cycles. DDMA will generate the remapped cycles only for those channels which are enabled in the slave channel configuration registers.

Table 4-57 Register Bits Associated with DDMA

7	6	5	4	3	2	1	0
SYSCFG 22h			**Inquire Cycle Control Register**				**Default = 00h**
0 = AHOLD+ BOFF# 1 = HOLD+ HLDA If SYSCFG 21h[0] = 0, this bit must be set to 1. If 21h[0] = 1, this bit must be set to 0.	OCAWE# and OCDOE# pin functionality in single bank cache: 0 = No change 1 = OCAWE# becomes ECA4 and OCDOE# becomes ECA3(1)	Reserved: Must be written to 0.	HRQ is sync to LCLK: 0 = No 1 = Yes (Must = 1 for DDMA operation)	Reserved: Must be written to 0.	EADS# generation: 00 = Normal for inquire cycle 01 = 1 CPU CLK earlier 10 = 1 CPU CLK earlier with async clock 2 CPU CLK earlier with sync clock 11 = Reserved		Single write hit leadoff cycle in a combined Dirty/Tag implementation:(2) 0 = 5 cycles 1 = 4 cycles
(1) This is to reduce cache address 0/1 (CA4/3) loading to improve timing if necessary. Do not set bit 6 if two banks of cache are in the system. (2) If bit 1 is set 1, SYSCFG 16h[4] should be set to 1.							
PCIDV1 5Ah-5Bh			**Distributed DMA Master Base Address Register**				**Default = 00h**
PCIDV1 5Ch			**Distributed DMA Control Register**				**Default = 00h**
Channel 7: 0 = Disable 1 = Enable	Channel 6: 0 = Disable 1 = Enable	Channel 5: 0 = Disable 1 = Enable	Channel 3: 0 = Disable 1 = Enable	Channel 2: 0 = Disable 1 = Enable	Channel 1: 0 = Disable 1 = Enable	Channel 0: 0 = Disable 1 = Enable	Master Distributed DMA (DDMA): 0 = Disable 1 = Enable

4.15 Fast GATEA20 and Reset Emulation

The 82C567 will intercept commands to Ports 060h and 064h so that it can emulate the keyboard controller, allowing the generation of the fast GATEA20 and fast INIT signals. The decode sequence is software transparent and requires no BIOS modifications to function. The fast GATEA20 generation sequence involves writing "D1h" to Port 064h, then writing data "02h" to Port 060h. The fast CPU "warm reset" function is generated when a Port 064h write cycle with data "FEh" is decoded. A write to Port 064h with data "D0h" will enable the status of GATEA20 (bit 1 of Port 060h) and the warm reset (bit 0 of Port 060h) to be readable.

If keyboard emulation is disabled, (i.e., PCIDV1 41h[4] = 1, in the system control register space of the 82C568) then the keyboard must generate the GATEA20 and KBRST signals. In this case, the A20M#+KBRST signal on the 82C568 functions as a KBRST input signal. It samples the KBRST output from the keyboard and generates an INIT to the CPU. The keyboard GATEA20 signal should be connected to the CPU A20M# input through a voltage translator. The keyboard GATEA20# signal is a 5.0V output signal and the CPU A20M# signal is a 3.3V input signal. Figure 4-55 shows the connectivity when keyboard emulation has been disabled.

Figure 4-55 Connections with Keyboard Emulation disabled

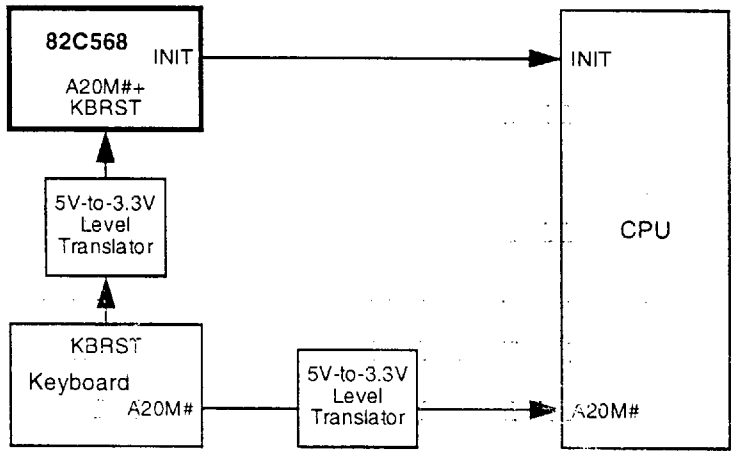


Table 4-58 Keyboard Emulation Associated Register Bits

7	6	5	4	3	2	1	0
PCIDV1 41h			**Keyboard Control Register - High Byte**				**Default = 00h**
Keyboard port read (RO): 0 = Does not say anything 1 = Keyboard controller has received command D0h and has not received the following 60h read	Keyboard port write (RO): 0 = Does not say anything 1 = Keyboard controller has received command D1h and has not received the following 60h write	Immediate INIT generation: 0 = Generate INIT immediately on FEh command 1 = Wait for halt before generating INIT on receiving the keyboard RESET	Keyboard emulation: 0 = Enable - Pin 12 functions as A20M# output 1 = Disable - Pin 12 functions as KBRST input	Selects which IRQ signal is to be generated when PIRQ3# has been triggered: 000 = IRQx* 001 = IRQ5 010 = IRQ9 011 = IRQ10 100 = IRQ11 101 = IRQ12 110 = IRQ14 111 = IRQ15 *Selection controlled by PCIDV1 50h[7:0].			Works with PCIDV1 40h[7:6]

4.16 Type F DMA

The Viper-MAX Chipset provides a form of compressed timing on the DMA called "Type F DMA". This mode provides ISA compatible timing for fast DMA slave devices. Type F timing basically runs at 360ns/cycle or three ISA clock cycles during the repeated portion of a Block or Demand mode transfer. Timing on DMA cycles is controlled by programming specific bits of registers in the I/O Register Space of the 82C568.

Programming I/O Address 40Bh[5:4] = 00 will yield compatible timings on DMA Channels 0 through 3. Programming 40Bh[5:4] = 11 will yield compressed timing or Type F DMA timing on Channels 0 through 3.

Likewise, programming I/O Address 4D6h[5:4] = 00 will yield compatible timings on DMA Channels 5 through 7. Programming 4D6h[5:4] = 11 will yield compressed timing or Type F DMA timing on Channels 5 through 7.

Table 4-59 Type F DMA Register Programming

7	6	5	4	3	2	1	0
I/O Address 40Bh			**DMA Channel Select Register 1**				**Default = 00h**
Reserved: Must be written to 0.		A setting of 00 on these bits gives compatible timing on DMA transfers. When set to 11, Type F DMA timing can be obtained.		Reserved: Must be written to 0.		DMA channel select: 00 = Ch. 0 10 = Ch. 2 01 = Ch. 1 11 = Ch. 3	
I/O Address 4D6h			**DMA Channel Select Register 2**				**Default = 00h**
Reserved: Must = 0		A setting of 00 on these bits gives compatible timing on DMA transfers. When set to 11, Type F DMA timing can be obtained.		Reserved: Must = 0		DMA channel select: 00 = Ch. 4 10 = Ch. 6 01 = Ch. 5 11 = Ch. 7	

4.17 Power Management

The Viper-MAX Chipset supplies an optimum GREEN solution by providing a green power management port for controlling desktop subsystems which includes clock control to the CPU's clock, STPCLK# signal to the CPU, and monitoring shutdown.

The Viper-MAX Chipset has a Green Event Timer (GET) used to activate the AUTO_GREEN or SMI_GREEN modes. The GET can be reloaded by any IRQ, PCI bus requests, DMA requests, keyboard, video, hard disk and floppy accesses, programmable I/O subsystem activity and optional external sources.

The AUTO_GREEN mode is available for dynamic CPUs which do not support the SMI protocol. The SMI_GREEN mode enables a much higher degree of software control for GREEN capabilities. The SMI_GREEN mode can only be utilized in systems that support the SMI# protocol.

4.17.1 Definition of Power Management Modes

The following subsections will define the various power management modes used when configuring systems with the Viper-MAX Chipset to run in the AUTO_GREEN and SMI_GREEN modes.

4.17.1.1 Normal Mode

In this mode, the system is running at full speed. No power management features have been activated.

4.17.1.2 AUTO_GREEN Mode

This mode is used to accommodate non-SL Enhanced CPUs. It allows for power management through hardware control. The AUTO_GREEN mode is entered when any enabled GREEN event occurs. When any enabled GREEN event occurs, the power control information in SYSCFG EAh is output to the SD[7:0] bus and the external power control latch is pulsed to match this value. The system can resume out of the AUTO_GREEN mode by any wake-up event that has been enabled in the power management registers. When a wake-up event takes place, the power control information in SYSCFG EBh is put out onto the SD[7:0] bus and the external power control latch is pulsed to latch this value. While returning to the NORMAL mode, the CPU clock first runs at full speed for 20ms before the ISA bus clock is switched back to the synchronous mode.

4.17.1.3 SMI_GREEN Mode

The SMI_GREEN mode is used to accommodate SMI supported CPUs. It allows power management through the SMI# protocol. When any event, that has been configured to generate a SMI#, occurs then a SMI# is generated from the Viper-MAX Chipset to the CPU. In response, the CPU saves the state of all its internal registers, asserts SMIACT# to the Viper-MAX Chipset and then begins executing the SMI code. In the SMI code, SYSCFG ECh can be directly written to the external power control latch, thus allowing power management through hardware too.

The system can resume out of the SMI_GREEN mode by any enabled wake-up event programmed in the power management register. During this Resume state, the system can be allowed to return to the NORMAL mode. The CPU clock first runs at full speed for 20ms before the ISA bus clock is switched back to the synchronous mode.

4.17.2 System Activity Detection

4.17.2.1 GREEN Events

The following is a list of events that can be programmed to be GREEN.

- GREEN Event Timer (GET) time-out
- EPMI# trigger
- Software bit trigger - This is a bit in the system power management registers which if set, causes a GREEN event
- Device Timer time-out - There are two general purpose timers that can be programmed (in SYSCFG F0h-F7h) to monitor user specified address locations. When either of them times out, a GREEN event is generated.

4.17.2.2 Reload GET/Wake-up Events

Any of the following events, if enabled, will cause the GREEN Event Timer to reload its initial count from SYSCFG E2h. These events, if enabled, will cause the system to generate a hardware PPWRL# (AUTO_GREEN mode) or SMI# (SMI_GREEN Mode) or both.

- All IRQs (except IRQ2)
- One programmable IO/MEM range access
- PREQ# signals from the PCI bus
- All DREQs (except DREQ4)
- Keyboard access:
  - I/O Ports 060h and 064h
- Video access:
  - 0A0000-0BFFFF address trap (graphics buffer)
  - I/O Port 3B0h-3DFh (VGA command registers)
- Hard/floppy disk access:
  - I/O Port 1F0h-1F7h and/or 3F6h, 170h-177h (hard disk)
  - I/O Port 3F5h (floppy)
- COM Ports:
  - COM1/3: COM1 (3F8-3FF) and COM3 (3E8-3EF)
  - COM2/4: COM2 (2F8-2FF) and COM4 (2E8-2EF)
- LPT Ports:
  - LPT1 (3B0-3BF), LPT2 (378-37F), and LPT3 (278-27F)
- External EPMI source

4.17.3 System Management Interrupt (SMI)

Most modern processors offer a System Management Interrupt (SMI) that allows external logic to signal to the CPU that a high priority event has occurred and must be serviced but should not in any way interfere with the application currently being processed. When the CPU senses its SMI input active, it saves the context of its current application and loads the context of its System Management Mode (SMM) handler routine from a protected part of RAM. SMM code can then proceed to determine the reason for the interrupt, service it appropriately, and return to application processing through a special RESUME instruction that restores the context as it originally was before the SMI. Entry to and exit from SMM is completely hardware-controlled.

4.17.3.1 SMI Implementation

During the NORMAL mode of operation, CPU accesses in the A0000h-BFFFFh are diverted to the ISA bus. During SMM, the SMIACT# signal is used for recognizing SMM addresses and these addresses are always mapped to the A0000h-BFFFFh range in DRAM (which is initialized with SMM code during boot-up). It is not required to flush the cache before executing SMM code due to the following reasons:

- The A0000-BFFFFh range is always made non-cacheable whether the CPU is in SMM mode or not.
- The SMBASE Register in the CPU is always initialized to A0000h.

If the SMBASE Register in the CPU were programmed with an address other than an address in the A0000h-BFFFFh range, it would be necessary to flush the cache.

4.17.4 Hardware Power Management Support without an External Latch

The Viper-MAX Chipset can support hardware power management through an external latch or internally through programmable pins on the 82C568. At power-on reset, if XDIR is sampled high, then hardware power management is done through an external latch. If XDIR is sampled low, then the PPWRL# and GPCS0# pins on the 82C568 change functionality to support hardware power management without a latch.

The PPWRL# pin becomes PPWRL2 and the GPCS0# pin becomes PPWRL1. With this implementation, the 82C568 will not be able to generate memory parity for PCI master writes. Hence, maximum performance for PCI master writes will be sacrificed.

4.18 IDE Controller

The IDE controller in the Viper-MAX Chipset is based on OPTi's Bus Master PCI IDE Module (MIDE) which is designed as a fast and flexible interface between the PCI bus and two channels of IDE devices (up to four devices). An integrated 12-level (48-byte) read prefetch FIFO and 12-level (48-byte) posted write FIFO supports bus mastering burst read and write operations on the PCI bus. This substantially improves the performance over the typical slave IDE implementations. The Enhanced ATA Specification can be supported by programming the internal registers up to IDE PIO Mode 4 and Single- and/or Multi-Word DMA Mode 2 timing.

4.18.1 Overview

The Viper-MAX Chipset provides a full function PCI local bus IDE controller capable of master mode operation. The chipset is capable of arbitrating for the ownership of the PCI local bus and transfer data between the IDE device and local memory. The IDE controller in the Viper-MAX Chipset conforms to the ATA Standard for IDE disk controllers.

By performing the IDE data transfers as a bus master instead of a slave, the chipset off-loads the CPU from having to perform the transfers. This benefit is realized in the form of the CPU not having to perform programmed I/O transfers to effect the data transfer between the disk and the memory. This feature improves system performance dramatically, especially in systems that operate in multitudinous environments.

Since the Viper-MAX Chipset implements this feature, the user's need to develop indigenous software to provide a complete product is greatly reduced.

The master mode of operation is an extension to the standard IDE controller model. Thus, systems can still revert back to slave mode IDE if they so desire. The master mode of operation is designed to work with any IDE device that supports DMA transfers on the IDE bus. Devices that do not support DMA on the IDE bus can revert to transfers across the IDE bus using programmed I/O.

The master mode of operation simply defines a simple scatter/gather mechanism allowing large blocks of data to be scattered to or gathered from memory. Such a mechanism improves system performance by reducing the number of interrupts to the CPU and the number of interventions by the CPU.

4.18.1.1 Scatter/Gather

The Viper-MAX Chipset provides another useful feature in the form of the scatter/gather functionality. Primarily scatter/gather functionality provides the user with the ability to transfer multiple buffers of data between memory and I/O without any intervention by the CPU. This facilitates the CPU to continue its processes without any interruption, thereby increasing the overall performance of the system.

In scatter/gather, the DMA reads memory addresses and word counts from multiple buffer descriptors located in the system memory (PCI or ISA). These buffer descriptors are called the scatter/gather descriptor (SGD) table. With the aid of these descriptors, the DMA controller can sustain DMA transfers until all the buffers in the descriptor table are transferred. The SGD table pointer address holds the address of the next buffer descriptor in the SGD table.

In order to perform a scatter/gather transfer the following needs to be done:

1) Software prepares the SGD table in system memory. A typical SGD will consist of an address pointer to the starting address and the transfer count of the memory buffer to be transferred.
2) Initialize the DMA channel mode registers with transfer specific information like 8/16-bit I/O device, transfer mode, etc.
3) Software provides the starting address of the SGD table by loading the SGD table pointer register.
4) Initiate the scatter/gather function by writing a start command to the Scatter/Gather Command Register.
5) The Mask Register should be the last register to be cleared to prevent the DMA from starting the DMA cycle with a partially loaded command description.
6) Once the register set is loaded and the channel is unmasked, DMA can begin.

4.18.2 Physical Region Descriptor Table

Before the IDE controller starts a master transfer it is given a pointer to a Physical Region Descriptor Table. This table contains some number of Physical Region Descriptors (PRDs) which describe areas of memory that are involved in the data transfer. The descriptor table must be aligned on a 4-byte boundary and the table cannot cross a 64K boundary in memory.

4.18.2.1 Physical Region Descriptor

The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The data transfer will proceed until all the regions described by the PRDs in the table have been transferred.

Each Physical Region Descriptor entry is eight bytes in length. The first four bytes specify the start address of a physical memory region. The next four bytes specify the size of the region in bytes (64K byte limit per region). A value of zero in these two bytes indicates 64K. Bit 7 of the last byte indicates the end of the table; bus master operation terminates when the last descriptor has been retired.

4.18.2.2 Bus Master IDE Registers

The bus master IDE function uses 16 bytes of I/O space. The base address of this block of I/O space is pointed to by the Bus Master IDE Base Address Register (IO5) in the PCI Configuration space. All bus master IDE I/O space registers can be accessed as byte, word, or dword quantities. The description of the 16 bytes of I/O registers is shown in Table 4-61 (refer to Section 5.0 for individual bit formats in each register).

Table 4-60 Physical Region Descriptor Table Entry

Bit(s)	Name	Default	Function
Byte-0, bit 0	---	0	0 (RO)
Byte-[3:1] Byte-0, bits [7:1]	BASE	xxxx xxxx	Memory Region Physical Base Address [31:1]
Byte-4, bit 0	---	0	0 (RO)
Byte-5 Byte-4, bits [7:1]	COUNT	xxxx	Byte Count [15:1]
Byte-6	---	xx	Reserved
Byte-7, bits [6:0]	---	xx	Reserved
Byte-7, bit 7	EOT	x	End of Table

Note: The memory region specified by the descriptor is further restricted such that the region cannot straddle a 64K boundary. This means the byte count is limited to 64K and the incrementer for the current address register only extends from bit 1 to bit 15.

Table 4-61 Bus Master IDE Registers

Offset from Base Address	Register Access	Register Name/Function
00h	R/W	Bus Master IDE Command Register for Primary IDE
01h		Device-specific
02h	RWC	Bus Master IDE Status Register for Primary IDE
03h		Device-specific
04h-07h	R/W	Bus Master IDE PRD Table Address for Primary IDE
08h	R/W	Bus Master IDE Command Register for Secondary IDE
09h		Device-specific
0Ah	RWC	Bus Master IDE status Register for Secondary IDE
0Bh		Device-specific
0Ch-0Fh	R/W	Bus Master IDE PRD Table Address for Secondary IDE

4.18.3 Programming the IDE Controller Registers

4.18.3.1 Standard Programming Sequence for Bus Mastering Operations

To initiate a bus master transfer between memory and an IDE DMA slave device, the following steps are required:

1) Software prepares a PRD Table in system memory. Each PRD is eight bytes long and consists of an address pointer to the starting address and the transfer count of the memory buffer to be transferred. In any given PRD Table, two consecutive PRDs are offset by eight bytes and are aligned on a 4-byte boundary. Bit 7 of the eighth byte in the PRD Table will be set to a 1 if that is the last entry in the table.
2) Software provides the starting address of the PRD Table by loading the PRD Table Pointer Register. The direction of the data transfer is specified by setting the Read/Write Control bit. Clear the Interrupt bit and Error bit in the Status Register.
3) Software issues the appropriate DMA transfer command to the disk device.
4) Engage the bus master function by writing 1 to the Start bit in the Bus Master IDE Command Register for the appropriate channel.
5) The controller transfers data to/from memory responding to DMA requests from the IDE device.
6) At the end of the transfer, the IDE device signals an interrupt.
7) In response to the interrupt, software resets the Start/Stop bit in the Command Register. Next it reads the controller's status and then the drive's status to determine if the transfer completed successfully.

4.18.3.2 Programming the IDE Mode Timing

Table 4-63 and Table 4-64 show the timing and recommended register settings for various IDE modes defined in the Enhanced IDE Specifications. They include PIO transfer, Single-Word DMA transfer, and Multi-Word DMA transfer modes. The actual cycle time equals the sum of actual command active time and actual command inactive (command recovery and address setup) time. These three timing requirements shall be met. In some cases, the minimum cycle time requirement is greater than the sum of the command pulse and command recovery time. This means either the command active (command pulse) or command inactive time (command recovery and address setup) can be lengthened to ensure that the minimum cycle times are met.

Figure 4-56 is a flow chart that describes how to program the primary channel of the MIDE interface. For the secondary channel, a similar procedure can be done by changing all the indexes from 1Fxh to 17xh.

4.18.3.3 Programming the IDE Interrupt Routing

Table 4-65 details the interrupt routing mechanism for the MIDE Module while in the Legacy and Native Modes. The system BIOS needs to program them accordingly.

Table 4-62 REGTIMx Programming Options

REGTIM0 1F3h/173h[2]	REGTIM1 1F3h/173h[3]	REGTIM2 1F3h/173h[7]	Drive 0 Control	Drive 1 Control
1(1)	0	1	Index-0	Index-1
0	1	1	Index-1	Index-0
0	0	1	Index-1	Index-1
1	0	0	Index-0	Default(2)
0	2	0	Default(2)	Index-0
0	0	0	Default(2)	Default(2)
1	1	x	Index-0	Index-0

(1) Recommended configuration.
(2) Refer to PCI Configuration Address Offset 40h[1:0] for default values.

Table 4-63 16-Bit Timing Parameters with 33MHz PCI Bus

Parameter: Register Bits	Dimension	IDE Transfer Modes											
		PIO Modes						Multi-Word DMA Modes			Single-Word DMA Modes		
		0	1	2	3	4	5	0	1	2	0	1	2
Address Setup: 1F6h/176h[5:4]	Bit values in hex	2	1	1	1	0	0	0	0	0	0	0	0
	Timing in LCLKs(1)	3	2	2	2	1	1	1	1	1	1	1	1
	Enhanced IDE Spec in ns(2)	70	50	30	30	25	N/S	N/A	N/A	N/A	N/A	N/A	N/A
R/W Command Pulse: 1F0h/170h/1F1h/171h[7:4], Index-0/1	Bit values in hex	5	4	3	2	2	2	7	2	2	F	8	4
	Timing in LCLKs(1)	6	5	4	3	3	3	8	3	3	16	9	5
	Enhanced IDE Spec in ns(2)	165	125	100	80	70	N/S	215	80	70	480	240	120
R/W Recovery Time: 1F0h/170h/1F1h/171h[3:0], Index-0/1	Bit values in hex	9	4	0	0	0	0	6	0	0	D	4	0
	Timing in LCLKs(1)	11	6	2	1	0	0	8	1	0	15	6	2
	Enhanced IDE Spec in ns(2)	N/S	N/S	N/S	70	25	N/S	215	50	25	N/S	N/S	N/S
Enhanced Mode: 43h bits [7:6], [5:4], [3:2], or [1:0]	Bit values in hex	0	0	0	1	2	2	0	1	2	0	0	0
DRDY: 1F6h/176h[3:1]	Bit values in hex	0	0	0	0	0	0	0	0	0	0	0	0
	Timing in LCLKs(1)	2	2	2	2	2	2	2	2	2	2	2	2
Cycle Time	Timing in LCLKs	20	13	8	6	5	4	17	5	4	32	16	8
	Enhanced IDE Spec in ns(2)	600	383	240	180	120	N/S	480	150	120	960	480	240

N/S = Not Specified, N/A = Not Applicable

(1) The actual timing (in LCLKs) that will be generated by the IDE controller if the recommended bit values in hex are programmed.

(2) The timing (in ns) as specified in the Enhanced IDE Specification.

Table 4-64 16-Bit Timing Parameters with 25MHz PCI Bus

Parameter: Register Bits	Dimension	IDE Transfer Modes											
		PIO Modes						Multi-Word DMA Modes			Single-Word DMA Modes		
		0	1	2	3	4	5	0	1	2	0	1	2
Address Setup: 1F6h/176h[5:4]	Bit values in hex	1	1	0	0	0	0	0	0	0	0	0	0
	Timing in LCLKs(1)	2	2	1	1	1	1	1	1	1	1	1	1
	Enhanced IDE Spec in ns(2)	70	50	30	30	25	N/S	N/A	N/A	N/A	N/A	N/A	N/A
R/W Command Pulse: 1F0h/170h/1F1h/171h[7:4], Index-0/1	Bit values in hex	4	3	2	2	1	1	5	2	1	D	6	3
	Timing in LCLKs(1)	5	4	3	3	2	2	6	3	2	13	7	4
	Enhanced IDE Spec in ns(2)	165	125	100	80	70	N/S	215	80	70	480	240	120
R/W Recovery Time: 1F0h/170h/1F1h/171h[3:0], Index-0/1	Bit values in hex	6	2	0	0	0	0	4	0	0	8	2	0
	Timing in LCLKs(1)	8	4	2	1	0	0	6	1	0	10	4	1
	Enhanced IDE Spec in ns(2)	N/S	N/S	N/S	70	25	N/S	215	50	25	N/S	N/S	N/S
Enhanced Mode: 43h bits [7:6], [5:4], [3:2], or [1:0]	Bit values in hex	0	0	0	1	2	2	0	1	2	0	0	1
DRDY: 1F6h/176h[3:1]	Bit values in hex	0	0	0	0	0	0	0	0	0	0	0	0
	Timing in LCLKs(1)	2	2	2	2	2	2	2	2	2	2	2	2
Cycle Time	Timing in LCLKs	15	10	6	5	4	3	13	4	3	24	12	6
	Enhanced IDE Spec in ns(2)	600	383	240	180	120	N/S	480	150	120	960	480	240

N/S = Not Specified, N/A = Not Applicable

(1) The actual timing (in LCLKs) that will be generated by the MIDE Module if the recommended bit values in hex are programmed.

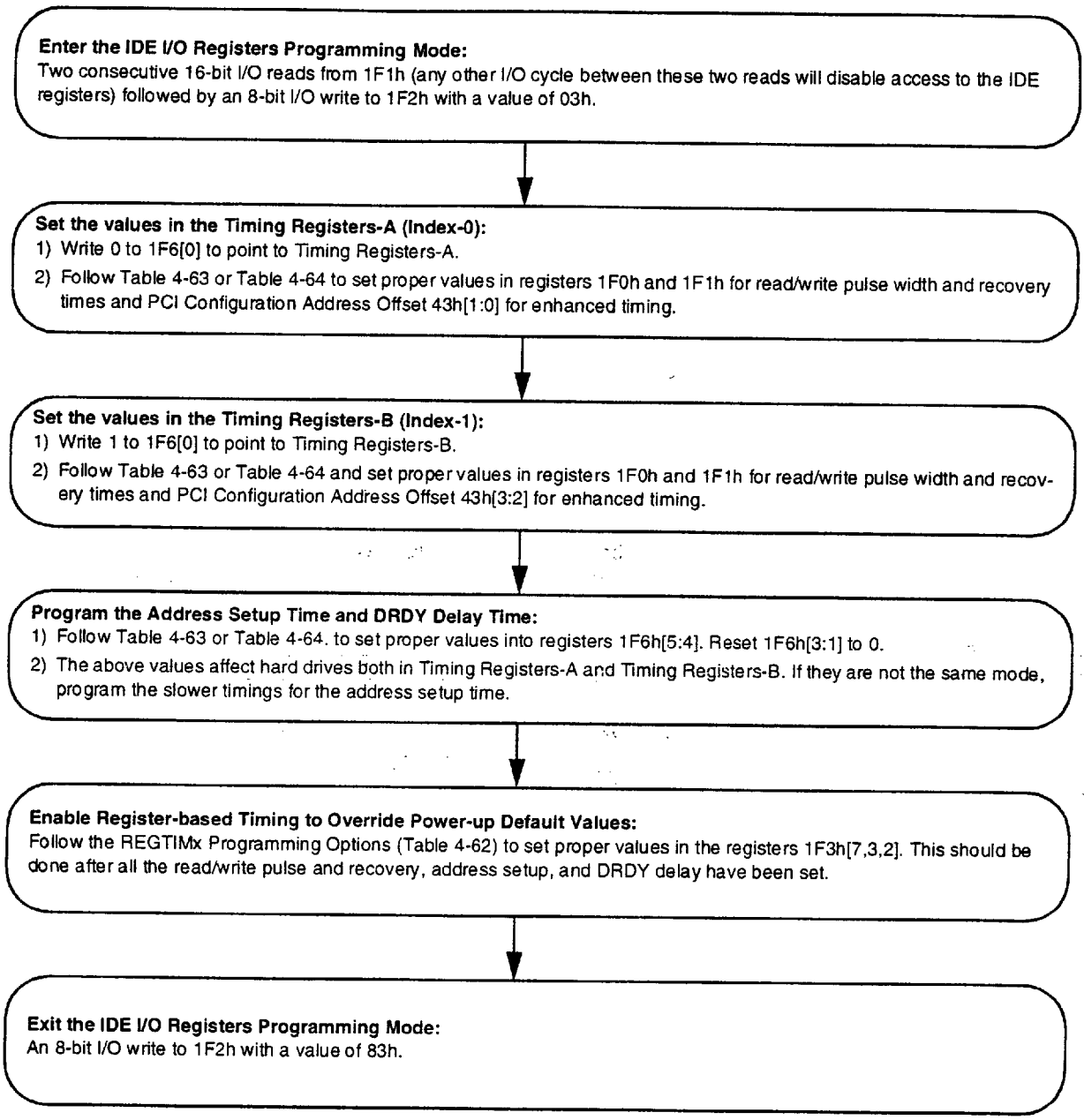
(2) The timing (in ns) as specified in the Enhanced IDE Specification.

Table 4-65 IDE Interrupt Routing Chart

Functions		82C568 PCI Configuration Register Setting					Interrupt Controller			
		PCI Bus Intrf, Dev = 01h Func = 0	IDE Module, Dev = 01h Func = 1(1)				82C568 Interrupt Input		IDE Interrupts Output	
IDE Modes		4Fh[6]	04h[0]	40h[3]	40h[2]	09h[3:0]	Pin 135(2)	Pin 136(2)	Primary	Secondary
Primary	Secondary	IDE Module Enable	IDE I/O Enable	2nd IDE Disable	Native Mode Enable	Native/ Legacy Mode	IRQ14 or DINT0	IRQ15 or DINT1	8259 or PCI INTC	8259 or PCI INTC
Disable		0	PCI Config. Register Space cannot be accessed				ISA IRQ14	ISA IRQ15	N/A	N/A
		1	0	x	x	xxxx				
Legacy(3)	Disable	1	1	1	0	xxxx	DINT0	ISA IRQ15	8259 IRQ14	N/A
		1	1	1	1	xx10				
Native	Disable	1	1	1	1	xx11	DINT0	ISA IRQ15	PIRQ3#(4)	N/A
Legacy(3)	Native	1	1	0	1	1110	DINT0	DINT1	8259 IRQ14	PIRQ3#(4)
Native	Legacy(3)	1	1	0	1	1011	DINT0	DINT1	PIRQ3#(4)	8259 IRQ15
Legacy(3)	Legacy(3)	1	1	0	0	xxxx	DINT0	DINT1	8259 IRQ14	8259 IRQ15
		1	1	0	1	1010				
Native	Native	1	1	0	1	1111	DINT0	DINT1	PIRQ3#(4)	PIRQ3#(4)

(1) It assumes that SYSCFG FFh[4] of the 82C568 is set to 1, the IDE module is mapped as Device #01h, Function #1.
(2) The ISA IRQ14 (ISA IRQ15) will not be available to the ISA bus if the on-board primary (secondary) IDE is enabled.
(3) The 8259 IRQ14 (8259 IRQ15) will not be available for PIRQ[3:0]# if the on-board primary (secondary) IDE is enabled.
(4) In Native mode, IDE interrupts are shared with PIRQ3# from the PCI bus. It is routed in the same way as PIRQ3# to the interrupt controller and is controlled by PCIDV1 40h[11:9], 42h[7:1], and 50h[7:6] of the 82C568 (Device #01h, Function #0). Using this mode requires that the IDE device's Interrupt Service Routine support interrupt sharing.

Figure 4-56 IDE Interface Primary Channel Programming Flow Chart



5.0 Register Descriptions

There are three broad classes of configuration registers spaces in the Viper-MAX Chipset:

1) PCI Configuration Register Space
2) System Control Register Space
3) I/O Register Space

All three of these register spaces can be found in both the 82C567 and 82C568. Table 5-1 and 5-2 details the locations and access mechanisms for registers located within these register spaces.

Table 5-1 82C567 Register Locations and Access Mechanisms

PCI Config. Register Space		System Control Register Space		I/O Register Space	
Location	**Access Mechanism**	**Location**	**Access Mechanism**	**Location**	**Access Mechanism**
PCIDV0 00h-47h	Through Mechanism #1 as: Bus #0, Device #0, Function #0	SYSCFG 00h-2Fh	Index loaded in 022h, Data to/from index through 024h	I/O Address CF8h, CFCh, I/O Address 8Fh	CPU Direct I/O R/W

Table 5-2 82C568 Register Locations and Access Mechanisms

PCI Config. Register Space		System Control Register Space		I/O Register Space	
Location	**Access Mechanism**	**Location**	**Access Mechanism**	**Location**	**Access Mechanism**
PCIDV1 00h-FFh	Through Mechanism #1 as: Bus #0, Device #1, Function #0	SYSCFG E0h-FFh	Index loaded in 022h, Data to/from index through 024h	I/O Address 060h, 061h, 064h, 092h I/O Address 8Fh	CPU Direct I/O R/W
Integrated Internal 82C206					
		SYSCFG2 01h	Index loaded in 022h, Data to/from index through 023h	I/O Address 000h-00Fh, 020h-021h, 040h-043h, 0A0h-0A1h, 0C0h-0DEh, 40Bh, 481h-4D6h	CPU Direct I/O R/W
		SYSCFG 80h-8Eh, 90h-96h	Index loaded in 022h, Data to/from index through 024h		
Integrated Internal IDE					
PCIIDE 00h-43h	Through Mechanism #1 as: Bus #0, Device #1, Function #1			I/O Address 1F0h-1F6h, 170h-176h, 370h-376h, 3F0h-3F6h I/O Address 8Fh	CPU Direct I/O R/W

PCI Configuration Register Access Method

The following briefly describes how to access the Viper-MAX Chipset and PCI devices on the slots. The Viper-MAX Chipset uses PCI Configuration Mechanism #1 to access the configuration spaces. Two I/O locations are used in this mechanism. The first I/O location, CF8h (which must be a double-word), references a read/write register that is called CONFIG_ADDRESS. The second I/O address, CFCh (which can be byte, word, or double-word), references a register called CONFIG_DATA. The general mechanism for accessing the configuration space is to write a value into CONFIG_ADDRESS that specifies the PCI bus, the device on that bus, and the configuration register in that device being accessed. A read or write to CONFIG_DATA will then cause the Viper-MAX Chipset to translate that CONFIG_ADDRESS value to the requested configuration cycle on the PCI bus. Below is an example to read PCIDV0 00h (the register located at 00h in the PCI Configuration Space of the 82C567):

```
MOV   EAX,80000800h  ;specifies the Device, Function,
                      and Register number
MOV   DX,0CF8h       ;CONFIG_ADDRESS
OUT   DX,EAX
MOV   DX,0CFCh       ;CONFIG_DATA
IN    EAX,DX
```

The content of the CONFIG_ADDRESS shown above possesses the following meanings (device number 10000b means the 82C567 is designed to use AD11 as the IDSEL) as shown in Table 5-3.

Table 5-3 CONFIG_ADDRESS Example

31	30 24	23 16	15 11	10 8	7 2	1	0
1	Reserved	Bus Number	Device Number	Function Number	Register Number	0	0
1	000 0000	0000 0000	0000 1	000	0000 00	0	0
80h		00h	08h		00h		

5.1 82C566 Register Space

The 82C566 does not have any registers that can be accessed directly. However, in the PCI Configuration Space of the 82C567 some register bits have been provided for configuring the 82C566.

When the CPU outputs a configuration write cycle to the 82C567, the 82C567 will initiate a special 82C566 configuration write to let the 82C566 snoop the HD bus and latch data into its internal 32-bit registers. Refer to PCIDV0 44h[2:0] for programming information.

5.2 82C567 Register Space

This section details the locations, access mechanisms, and bit formats for registers located within the 82C567.

5.2.1 82C567 PCI Configuration Register Space (PCIDV0)

The PCI Configuration Register Space of the 82C567 (PCIDV0) is accessed through Configuration Mechanism #1 as Bus #0, Device #0, and Function #0. Table 5-4 gives the bit formats for the registers accessed in PCIDV0. All bits are read/write and their default value is 0 unless otherwise specified.

Table 5-4 82C567 PCI Configuration Registers: PCIDV0 00h-47h

7	6	5	4	3	2	1	0
PCIDV0 00h **PCIDV0 01h**			**Vendor Identification Register (RO)**				**Default = 45h** **Default = 10h**
PCIDV0 02h **PCIDV0 03h**			**Device Identification Register (RO)**				**Default = 67h** **Default = C5h**
PCIDV0 04h			**Command Register - Byte 0**				**Default = 07h**
Address/data stepping (RO): 0 = Disable (always)	PERR# output pin: 0 = Disable (always)	Reserved: Must be written to 0.	Memory write and invalidate cycle generation (RO): Must = 0 (always) No memory write and invalidate cycles will be generated by the 82C567.	Special cycles (RO): Must = 0 (always) The 82C567 does not respond to the PCI special cycle.	Bus master operations (RO): Must = 1 (always) This allows the 82C567 to perform bus master operations at any time. (Default = 1)	Memory access (RO): Must = 1 (always) The 82C567 allows a PCI bus master access to memory at anytime. (Default = 1)	I/O access (RO): Must = 1 (always) The 82C567 allows a PCI bus master I/O access at any time. (Default = 1)
PCIDV0 05h			**Command Register - Byte 1**				**Default = 00h**
Reserved: Must be written to 0.						Fast back-to-back to different slaves: 0 = Disable 1 = Enable	SERR# output pin (RO): 0 = Disable (always)
PCIDV0 06h			**Status Register - Byte 0**				**Default = 80h**
Fast back-to-back capability (RO): 0 = Not Capable 1 = Capable (Default = 1)	Reserved: Must be written to 0.						

Table 5-4 82C567 PCI Configuration Registers: PCIDV0 00h-47h (cont.)

7	6	5	4	3	2	1	0
PCIDV0 07h			**Status Register - Byte 1**				**Default = 02h**
Detected parity error (RO): Must = 0 (always)	SERR# status (RO): Must = 0 (always)	Master abort status (RO): Must = 0 (always)	Received target abort status (RO): 0 = No target abort 1 = Target abort occurred	Signaled target abort status (RO): Must = 0 (always)	DEVSEL# timing status (RO): Must = 01 (always) Indicates medium timing selection; the 82C567 asserts the DEVSEL# based on medium timing. (Default = 01)		Data parity detected (RO): Must = 0 (always)
PCIDV0 08h			**Revision Identification Register (RO)**				**Default = 00h**
PCIDV0 09h			**Class Code Register (RO)**				**Default = 00h**
PCIDV0 0Ah							**Default = 00h**
PCIDV0 0Bh							**Default = 06h**
PCIDV0 0Ch			**Reserved Register** Reserved: Must be written to 0.				**Default = 00h**
PCIDV0 0Dh			**Master Latency Timer Register (RO)**				**Default = 00h**
PCIDV0 0Eh			**Header Type Register (RO)**				**Default = 00h**
PCIDV0 0Fh			**Built-In Self-Test (BIST) Register (RO)**				**Default = 00h**
PCIDV0 10h-3Fh			**Reserved Register (RO)** Reserved: Must be written to 0.				**Default = 00h**
PCIDV0 40h			**Memory Control Register - Byte 0**				**Default = 00h**
PCI video frame buffer write posting hole: These bits map onto address bits A[31:30]. Together with PCIDV0 41h[7:0] they define the 4MB window where write posting can be masked.	Reserved: Must be written to 0.	PCI to ISA decoding mode: 0 = Subtractive 1 = Slow	If this bit is set to 0, then control of writes being posted on the PCI bus is determined by the settings of SYSCFG 15h[5:4]. If this bit is set to 1, then no writes will be posted on the PCI bus except writes to the video memory and frame buffer areas.	Write posting to the video frame buffer control: If PCIDV0 40h[3] = 0: 0 = Enable 1 = Disable If PCIDV0 40h[3] = 1: 0 = Disable 1 = Enable	Write posting to the video memory (A0000h-BFFFFh) control: If PCIDV0 40h[3] = 0: 0 = Enable 1 = Disable If PCIDV0 40h[3] = 1: 0 = Disable 1 = Enable		I/O cycle write post control: 0 = Disable 1 = Enable
PCIDV0 41h			**Memory Control Register - Byte 1** PCI video frame buffer write posting hole: These bits map onto address bits A[29:22]. Together with PCIDV0 40h[7:6] they define the 4MB window where write posting can be masked.				**Default = 00h**

Table 5-4 82C567 PCI Configuration Registers: PCIDV0 00h-47h (cont.)

7	6	5	4	3	2	1	0
PCIDV0 42h			**Memory Control Register - Byte 2**				**Default = 00h**
Reserved: Must be written to 0.						Reserved: Must be written to 1.	HA drive-back during CPU accesses to memory: 0 = Disable 1 = Enable
PCIDV0 43h			**Memory Control Register - Byte 3** Reserved: Must be written to 0.				**Default = 00h**
PCIDV0 44h			**82C566 Control Register 1**				**Default = 00h**
6DW FIFO for CPU write to PCI: 0 = Disable 1 = Enable	24DW FIFO for PCI read from DRAM: 0 = Disable 1 = Enable	24DW FIFO for PCI write to DRAM: 0 = Disable 1 = Enable	6QW FIFO for CPU write to DRAM: 0 = Disable 1 = Enable	Memory read accesses in the 82C566 if PCIDV0 44h[0] = 1 and 47h[7] = 1: 0 = SDRAM 1 = BEDO	82C566 ping-pong buffer used for PCI master write X-1-1-1: 0 = Disable 1 = Enable	82C566 ping-pong buffer used for PCI master read X-1-1-1: 0 = Disable 1 = Enable	Memory read accesses in the 82C566: 0 = FP Mode 1 = EDO/ SDRAM/ BEDO
PCIDV0 45h			**82C566 Control Register 2**				**Default = 00h**
Reserved: Must be written to 0.				Memory parity generation and checking if PCIDV0 45h[1] = 0: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	Byte merge for CPU write to DRAM: 0 = Disable 1 = Enable	MD bus internal pull-up resistors: 0 = Enable 1 = Disable
PCIDV0 46h			**82C566 Control Register 3** Reserved: Must be written to 0.				**Default = 00h**
PCIDV0 47h			**82C566 Control Register 4**				**Default = 00h**
SDRAM/BEDO memory read accesses in 82C566: 0 = Disable 1 = Enable	CPU-to-PCI FIFO clearing when combination changed: 0 = Do not clear 1 = Clear	PCI-to-DRAM FIFO clearing when combination changed:(1) 0 = Do not clear 1 = Clear	CPU-to-DRAM FIFO clearing when combination changed: 0 = Do not clear 1 = Clear	82C566 register is writable: 0 = Enable 1 = Disable (cnfg-writes blocked within 82C566)	Reserved: Must be written to 0.		

(1) Bit 5 must be set to 1 whenever PCI to DRAM FIFO is turned on in the system.

5.2.2 82C567 System Control Register Space (SYSCFG)

An indexing scheme is used to access the System Control Register Space (SYSCFG) of the 82C567. Port 022h is used as the Index Register and Port 024h as the Data Register. Each access to a register within this space consists of:

1) a write to Port 022h, specifying the desired register in the data byte,
2. followed by a read or write to Port 024h with the actual register data.

The index resets after every access; so every data access (via Port 024h) must be preceded by a write to Port 022h even if the same register is being accessed consecutively.

Index 023h is the Data Register for DMA clock select.

Table 5-5 gives the bit formats for the System Control Registers accessed in SYSCFG. All bits are read/write and their default value is 0 unless otherwise specified.

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh

7	6	5	4	3	2	1	0
SYSCFG 00h			**Byte Merge/Prefetch & Sony Cache Module Control Register(1)**				**Default = 00h**
Enable pipelining of single CPU cycles to memory: 0 = Disable 1 = Enable	Video memory byte/word read prefetch enable: This setting enables/disables the prefetching of bytes/words/ from PCI video memory by the CPU. 0 = Disable 1 = Enable	Sony SONIC-2WP support enable:(2) 0 = No Sony SONIC-2WP installed 1= Sony SONIC-2WP installed	Byte/word merge support: 0 = Disable 1 = Enable	Byte/word merging with CPU pipelining (NA# generation) support: 0 = Disable 1 = Enable	Time-out counter for byte/word merge: This setting determines the maximum time difference between two consecutive PCI byte/word writes to allow merging. 00 = 4 CPU CLKs 01 = 8 CPU CLKs 10 = 12 CPU CLKs 11 = 16 CPU CLKs		Enable internal hold requests to be blocked while performing byte merge: 0 = Disable 1 = Enable

(1) SYSCFG 13h[7] must be set to 1 in order for this register to be mapped correctly (full memory decode).
(2) If bit 5 is set, ensure that the L2 cache has been disabled (i.e., set SYSCFG 02h[3:2] = 00).

SYSCFG 01h		**DRAM Control Register 1**			**Default = 00h**
Row address hold after RAS# active: 0 = 2 CLKs 1 = 1 CLK	RAS# active/ inactive on entering master mode: 0 = Normal page mode, RAS# active when starting master cycle 1 = RAS# inactive when starting a master cycle	RAS pulse width used during refresh: 00 = 7 CLKs 01 = 6 CLKs 10 = 5 CLKs 11 = 4 CLKs	CAS pulse width during reads: 0 = 3 CLKs 1 = 2 CLKs	CAS pulse width during writes: 0 = 3 CLKs 1 = 2 CLKs	RAS precharge time: 00 = 6 CLKs 01 = 5 CLKs 10 = 4 CLKs 11 = 3 CLKs

SYSCFG 02h		**Cache Control Register 1**			**Default = 00h**
L2 cache size selection: **If SYSCFG 0Fh[0] = 0** 00 = Reserved 01 = Reserved 10 = 256K 11 = 512K	**If SYSCFG 0Fh[0] = 1** 00 = 1MB 01 = 2MB 10 = Reserved 11 = Reserved	L2 cache write policy: 00 = L2 cache write-through 01 = Adaptive Write-back Mode 1 10 = Adaptive Write-back Mode 2 11 = L2 cache write-back	L2 cache operating mode select: 00 = Disable 01 = Test Mode 1; External Tag Write (Tag data write-through SYSCFG 07h) 10 = Test Mode 2; External Tag Read (Tag data read from SYSCFG 07h) 11 = Enable L2 cache	DRAM posted write: 0 = Disable 1 = Enable	CAS precharge time: 0 = 2 CLKs 1 = 1 CLK

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0

SYSCFG 03h — Cache Control Register 2 — Default = 00h

Timing for burst writes to L2 cache:	Leadoff cycle time for writes to L2 cache:	Timing for burst reads to L2 cache:	Leadoff cycle time for reads to L2 cache:
00 = X-4-4-4 10 = X-2-2-2 01 = X-3-3-3 11 = X-1-1-1	00 = 5-X-X-X 10 = 3-X-X-X 01 = 4-X-X-X 11 = 2-X-X-X(1)	00 = X-4-4-4 10 = X-2-2-2 01 = X-3-3-3 11 = X-1-1-1	00 = 5-X-X-X 10 = 3-X-X-X 01 = 4-X-X-X 11 = 2-X-X-X(1)

(1) Sync SRAM double bank implementation does not support this timing.

SYSCFG 04h — Shadow RAM Control Register 1 — Default = 00h

CC000h-CFFFFh read/write control:	C8000h-CBFFFh read/write control:	Sync SRAM pipelined read cycle 1-1-1-1 enable:(1)	E0000h-EFFFFh range selection:	C0000h-C7FFFh read/write control:
00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	0 = Implies leadoff T-state for read pipe-lined cycle = 2(2) 1 = Enables leadoff T-state for read pipe-lined cycle = 1(3)	Determines whether this region will be treated like the F0000 BIOS area or whether it will always be non-cacheable. 0 = E0000h-EFFFFh area will always be non-cacheable 1 = E0000h-EFFFFh area will be treated like the F0000h BIOS area. If this bit is set, then SYSCFG 06h[3:2] and [1:0] Should be set identically.	00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM

(1) If SYSCFG 11h[3] = 1 (i.e., sync SRAM chosen) and if SYSCFG 03h[3:2] = 11, then this register setting comes into play.
(2) It will be a 3-1-1-1 cycle followed by a 2-1-1-1 cycle, or a 3-1-1-1 cycle for successive piped cycles.
(3) It will be a 3-1-1-1 cycle followed by a 1-1-1-1 cycle for successive piped cycles. This is valid only for a single bank case.

SYSCFG 05h — Shadow RAM Control Register 2 — Default = 00h

DC000h-DFFFFh read/write control:	D8000h-DBFFFh read/write control:	D4000h-D7FFFh read/write control:	D0000h-D3FFFh read/write control:
00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PC I/ write to DRAM 11 = Read/write DRAM	00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 06h			**Shadow RAM Control Register 3**				**Default = 00h**
DRAM hole in system memory from 80000h-9FFFFh:(1) 0 = No hole in memory 1 = Enable hole in memory	Wait state addition for PCI master snooping: 0 = Do not add a wait state for the cycle access finish to do the snooping 1 = Add a wait state for the cycle access to finish and then do the snooping	C0000h-C7FFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh cacheability: 0 = Not Cacheable 1 = Cacheable in L1 and L2 (L1 disabled by SYSCFG 08h[0])	F0000h-FFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM If SYSCFG 04h[2] = 1, then the E0000h-EFFFFh read/write control should have the same setting as this.		E0000h-EFFFFh read/write control: 00 = Read/write PCI bus 01 = Read from DRAM / write to PCI 10 = Read from PCI / write to DRAM 11 = Read/write DRAM	
(1) This setting gives the user the option to have some other device in the address range 80000h-9FFFFh instead of system memory. When bit 7 is set, the 82C567 will not start the system DRAM controller for accesses to this particular address range.							
SYSCFG 07h			**Tag Test Register** Data from this register is written to the tag, if in Test Mode 1 (refer to SYSCFG 02h). Data from the tag is read into this register, if in Test Mode 2 (refer to SYSCFG 02h).				**Default = 00h**
SYSCFG 08h			**CPU Cache Control Register**				**Default = 00h**
L2 cache single/double bank select: 0 = Double bank (If async SRAM, then the banks are interleaved. If sync SRAM, they are not interleaved.) 1 = Single bank (non-interleaved)	Snoop filtering for bus masters:(1) 0 = Disable 1 = Enable	CPU HITM# pin sample timing: 0 = Delay 1 CLK (HITM# sampled on 3rd rising edge of LCLK after EADS# assertion) 1 = No delay (HITM# sampled on 2nd rising edge of LCLK after EADS# assertion)	Parity checking: 0 = Disable 1 = Enable	Tag/Dirty RAM implementation: 0 = Tag and Dirty are on separate chip (i.e., a separate x1 or x8 SRAM for the Dirty RAM) 1 = Tag and Dirty are on the same chip (i.e., could be either a x9 or x8 Tag/Dirty RAM)	CPU address pipelining: 0 = Disable 1 = Enable	L1 cache write-back and write-through control: 0 = Write-through only 1 = Write-back enabled	BIOS area cacheability in L1 cache: Determines if system BIOS area E0000h-FFFFFh (if SYSCFG 04h[2] = 1) or F0000h-FFFFFh (if SYSCFG 04h[2] = 0), and video BIOS area C0000h-C7FFFh is cacheable in L1 or not. 0 = Cacheable 1 = Non-Cacheable
(1) For a master request if the subsequent read/write is within the same cache line, CPU 'Inquire' cycles are not done until there is a cache line miss (i.e., line comparator not activated for accesses within the same cache line).							
SYSCFG 09h			**System Memory Function Register**				**Default = 00h**
DRAM Hole B size: 00 = 512KB 10 = 2MB 01 = 1MB 11 = 4MB Address for this hole is specified in SYSCFG 0Bh[7:0] and 0Ch[3:2]		DRAM Hole B control mode: 00 = Disable 01 = WT for L1 and L2 10 = Non-cacheable for L1 and L2 11 = Enable hole in DRAM		DRAM Hole A size: 00 = 512KB 10 = 2MB 01 = 1MB 11 = 4MB Address for this hole is specified in SYSCFG 0Ah[7:0] and 0Ch[1:0]		DRAM Hole A control mode: 00 = Disable 01 = WT for L1 and L2 10 = Non-cacheable for L1 and L2 11 = Enable hole in DRAM	

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 0Ah	**DRAM Hole A Address Decode Register 1**						**Default = 00h**
DRAM Hole A starting address: These bits along with SYSCFG 0Ch[1:0] are used to specify the starting address of DRAM Hole A. These bits, AST[7:0], map onto HA[26:19] lines.							
SYSCFG 0Bh	**DRAM Hole B Address Decode Register 2**						**Default = 00h**
DRAM Hole B starting address: These bits along with SYSCFG 0Ch[3:2] are used to specify the starting address of DRAM Hole A. These bits, BST[7:0], map onto HA[26:19] lines.							
SYSCFG 0Ch	**DRAM Hole Higher Address**						**Default = 00h**
Reserved: Must be written to 0.	Fast BRDY# generation for DRAM write page hits. BRDY# for DRAM writes generated on: 0 = 4th CLK 1 = 3rd CLK	HACALE cycle: 0 = Normal timing 1 = HACALE one-half a clock cycle early	Cache WE# pulse width: 0 = Normal (i.e., ~15ns) 1 = Wider (i.e., ~17.5ns)	DRAM Hole B starting address: These bits are used in conjunction with the bits in SYSCFG 0Bh to specify the starting address of DRAM Hole B. These bits, BST[9:8], map onto HA[28:27].		DRAM Hole A starting address: These bits are used in conjunction with the bits in SYSCFG 0Ah to specify the starting address of DRAM Hole A. These bits, AST[9:8], map onto HA[28:27].	
SYSCFG 0Dh	**Clock Control Register**						**Default = 00h**
Reserved: Must be written to 0.				Enable A0000h-BFFFFh as system memory: 0 = No 1 = Yes	Add one more wait state during PCI master cycle with Intel-type address toggling(1): 0 = No 1 = Yes	Give the 82C567 control of the PCI bus on STOP# generation after HITM# is active: 0 = No 1 = Yes(2)	CPU clock is slowed down to below 33MHz: 0 = No 1 = Yes
SYSCFG 0Eh	**PCI Master Burst Control Register 1**						**Default = 00h**
Reserved: Must be written to 0.	Pin 55 and 57 functionality: 0 = Pin 55 is OCDOE# or HLDA or SDRAS# and pin 57 is OCAWE#, CAS1O#, HOLD, or SDCAS# 1 = Pin 55 is MEMR# and Pin 57 is MEMW#	Reserved: Must be written to 0.	Reserved: Must be written to 0.	Parity check during master cycles (if SYSCFG 08h[4] = 1): 0 = Enable 1 = Disable	Fast NA# generation: 0 = Disable 1 = Enable for every single transfer cycle	Write protection for L1 BIOS: 0 = No 1 = Yes	PCI line comparator (if SYSCFG 08h[6] = 1): 0 = Use line comparator in PCI master 1 = Generate inquire cycle for every new FRAME#

(1) If the PCI master does its address toggling in the style of the Intel 486 burst, rather than a linear burst mode style, then one wait state needs to be added.

(2) The 82C567 has control over the PCI bus until the write-back is completed. If PCI master pre-snoop has been enabled (SYSCFG 0Fh[7] = 1), 0Dh[1] should be set to 1.

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 0Fh			**PCI Master Burst Control Register 2**				**Default = 00h**
PCI pre-snoop: 0 = Disable 1 = Enable(1)	Insert wait states for ISA master access: 0 = No 1 = Yes	Reserved: Must be written to 0.	Resynchronize PCI master accesses to system DRAM: 0 = No 1 = Yes(2)	New mode of single cycle NA#: 0 = Disable 1 = Enable	CPU to L2 cache hit cycles, ASDC# generation from chipset: 0 = Enable 1 = Disable(3)	Write pulse duration control for operation with async SRAM: This bit is used when the write cycle takes the form of 3-X-X-X. 0 = 1 CPUCLK 1 = CPUCLK/2 plus the delay of an internal delay line	Cache size selection: This bit along with SYSCFG 02h[1:0] defines the L2 cache size. 0 = < 1MB 1 = ≥1MB

(1) The 82C567 generates a pre-snoop cycle to the CPU assuming that the PCI master will do a burst.
(2) If bit 4 = 1 in sync SRAM mode, PCI master access to system memory will force the master to wait for the current cycle to finish and the CPU-PCI clock to become sync. This is a conservative mode.
(3) SYSCFG 0Fh[2] needs to be set if pipelined sync SRAMs are being used.

7	6	5	4	3	2	1	0
SYSCFG 10h			**Miscellaneous Control Register 1**				**Default = 00h**
CPU to PCI/ ISA slave cycle triggered: 0 = After 2nd T2 1 = After 1st T2	Cache modified write cycle timing: 0 = No delay on CA4 1 = In a two bank cache, CA4 is delayed one-half clock	Leadoff cycle for a pipelined read: 0 = 3-X-X-X read followed by a 3-X-X-X pipelined read cycle 1 = 3-X-X-X read followed by a 2-X-X-X pipelined read cycle	2-X-X-X pipelined write hit cycles: 0 = Disable 1 = Enable	Move the write pulse one-half a clock later in X-2-2-2 write hit cycles: 0 = No 1 = Yes	Move the write pulse one-half a clock earlier in 3-X-X-X write hit cycles: 0 = No 1 = Yes	Reserved: Must be written to 1.	LCLK select control:(1): LCLK <= 1/2 CPUCLK period before CPUCLK LCLK <= 0.5ns after CPUCLK 0 = LCLK is async to the CPUCLK 1 = LCLK is sync to the CPUCLK **Note:** In the synch LCLK option, LCLK = CPUCLK/2.

(1) If bit 0 is set, (i.e., a sync PCI implementation is being used) then the timing constraints between the LCLK and CPUCLK inputs to the 82C567 must be met.

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 11h			**Miscellaneous Control Register 2**				**Default = 00h**
Reserved: Must be written to 0.		Cache inactive during Idle state control: This bit controls the chip selects of the SRAMs. 0 = SRAM active always 1 = SRAM inactive during Idle state	Next address (NA#) mode control:(1) 0 = Normal NA# timing used with async SRAMs 1 = New NA# timing for sync SRAMs - used only when CPU operating at 50MHz	SRAM type: 0 = Async SRAM 1 = Sync SRAM	Page miss posted write: 0 = Enable 1 = Disable	ISA/DMA IOCHRDY control: 0 = Old mode, no IOCHRDY during line hit 1 = Drive IOCHRDY low until cycle is finished	Delay start: 0 = Old mode, do not delay internal master cycle cycles after an inquire cycle 1 = Delay internal master cycles by one LCLK after inquire cycle

(1) If the CPU is used at a 50MHz operating frequency, then a 2-1-1-1 cycle on read/write hits to the sync SRAM can be obtained. To obtain this performance, the ADS# output of the CPU needs to be connected to the ADSP# input of the sync SRAM directly and bit 4 needs to be set. By setting bit 4, generation of the NA# signal from the chipset to the CPU is controlled.

SYSCFG 12h		**Refresh Control Register**			**Default = 00h**
REFRESH# pulse source: 0 = From 82C568 or ISA master is source of the REFRESH# input 1 = From 32kHz clock	Reserved: Must be written to 0.	Suspend mode refresh: 00 = From CLK state machine 01 = Self-refresh based on 32KHz only 10 = Normal refresh based on 32KHz only 11 = Undefined	Slow refresh: Refresh on: 00 = Every REFRESH#/32KHz falling edge 01 = Alternate REFRESH#/32KHz falling edge 10 = One in four REFRESH#/ 32KHz falling edge 11 = Every REFRESH#/32KHz toggle	LA[23:17] enable from 8Fh during refresh: 0 = Disable 1 = Enable	Reserved: Must be written to 0.

SYSCFG 13h	**Memory Decode Control Register 1**		**Default = 00h**
Memory decode select: This bit must be set to 1 for full decode (maximum flexibility in choosing different DRAM configurations)	Full decode for logical Bank 1 (RAS1#) if SYSCFG 13h[7] is set: 000 = 0Kx36 100 = 2Mx36 001 = 256Kx36 101 = 4Mx36 010 = 512Kx36 110 = 8Mx36 011 = 1Mx36 111 = 16Mx36	SMRAM: 0 = Disable 1 = Enable	Full decode for logical Bank 0 (RAS0#) if SYSCFG 13h[7] is set: 000 = 0Kx36 100 = 2Mx36 001 = 256Kx36 101 = 4Mx36 010 = 512Kx36 110 = 8Mx36 011 = 1Mx36 111 = 16Mx36

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 14h				**Memory Decode Control Register 2**			**Default = 00h**
82C566 mode: 0 = Normal mode 1 = Clocked mode (Must = 1 for EDO timing)	Full decode for logical Bank 3 (RAS3#) if SYSCFG 13h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36			SMRAM control: Inactive SMIACT#: 0 = Disable SMRAM 1 = Enable SMRAM(1) Active SMIACT#: 0 = Enable SMRAM for both Code and Data(1) 1 = Enable SMRAM for Code only (1)	Full decode for logical Bank 2 (RAS2#) if SYSCFG 13h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36		

(1) If SYSCFG 13h[3] is set.

SYSCFG 15h		**PCI Cycle Control Register 1**		**Default = 00h**
CPU master to PCI memory slave write IRDY# control: 00 = 3 LCLKs after data 01 = 2 LCLKs after data 10 = 1 LCLK after data 11 = 0 LCLK after data	CPU master to PCI slave write posting, bursting control: 00 = No posting, no bursting 01 = Posting only, no bursting 10 = Posting, with conservative bursting 11 = Posting, with aggressive bursting	Master retry timer: Selects the delay before retry is attempted. 00 = 10 PCICLKs 01 = 18 PCICLKs 10 = 34 PCICLKs 11 = 66 PCICLKs	Reserved: Must be written to 0.	PCI FRAME# generation control: 0 = Conservative mode in CPU pipelined cycle 1 = Aggressive mode

SYSCFG 16h			**Dirty/Tag RAM Control Register**				**Default = 00h**
DIRTYI pin selection:(1) 0 = Input only 1 = I/O	Reserved: Must be written to 0.	Tag RAM size selection:(2) 0 = 8-bit 1 = 7-bit	Single write hit leadoff cycle in a combined Dirty/Tag implementation:(3) 0 = 5 cycles 1 = 4 cycles	Pre-snoop control: 0 = Pre-snoop for starting address 0 only 1 = Pre-snoop for all addresses except those on the line boundary	Synchronization between the PCI bus clock (LCLK) and the CPU clock (CLK):(4) 0 = LCLK async to CLK 1 = LCLK sync to CLK (skew not to exceed -2ns to 15ns)	Reserved: Must be written to 0.	HDOE# timing control: 0 = Negated normally 1 = Negated one clock before the cycle finishes

(1) If using a x1 SRAM for the Dirty RAM in which there is a separate DirtyIn and a separate DirtyOut bit, then the DIRTYI pin becomes an input only. If using a standard x8 or x9 SRAM, where there is no separate pin for input and output, then the DIRTYI pin becomes an I/O pin.

(2) If a 7-bit Tag is being used and a combined Tag/Dirty RAM is being used, then TAG0 functions as the DIRTYIO signal. In this case, the DIRTYI pin is unused.

(3) If bit 4 is set 1, SYSCFG 22h[0] should be set to 1.

(4) It should be noted that LCLK could be async to CLK also. This bit therefore implies that the PCI clock is either sync to the CPU clock with a skew not to exceed -2ns to 15ns, or that the PCI clock is async to the CPU clock.

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 17h			**PCI Cycle Control Register 2**				**Default = 00h**
Reserved: Must be written to 0.	Generate NA# for PCI slave access in async LCLK mode: 0 = No 1 = Yes This bit will be overridden if bit 7 is set.	Sync two bank select: 0 = Reserved 1 = Set this bit to 1 when two banks of sync SRAM are installed	Reserved: Must be written to 0.	Reserved: Must be written to 0.	Pipelining during byte merge: 0 = Disable 1 = Enable	Sync SRAM type (if SYSCFG 11h[3] = 1): 0 = Standard 1 = Pipelined	Burst type: 0 = Intel burst protocol 1 = Cyrix linear burst protocol
SYSCFG 18h			**Tristate Control Register**				**Default = 00h**
Reserved: Must be written to 0.	Drive strength on RAS lines: 0 = 4mA 1 = 16mA	CAS lines voltage selection: 0 = 5.0V 1 = 3.3V	Drive strength on memory address lines and write enable line: 0 = 4mA 1 = 16mA	Tristate CPU interface during Suspend and during CPU power-off: 0 = Disable 1 = Enable	Tristate PCI interface during Suspend and during PCI power-off: 0 = Disable 1 = Enable	Tristate cache interface during Suspend and during cache power-off: 0 = Disable 1 = Enable	Pull-up/-down resistors active during Suspend and power-off: 0 = Disable 1 = Enable

SYSCFG 19h	**Memory Decode Control Register 3**				**Default = 00h**
Pin 189 functionality:(1) 0 = DIRTYWE# 1 = RAS5#	Full decode for logical bank 5 (RAS5#) if SYSCFG 13h[7] is set and 19h[7] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = 8Mx36 111 = 16Mx36		Pin 71 functionality:(2) 0 = MA11 1 = RAS4#	Full decode for logical bank 4 (RAS4#) if SYSCFG 13h[7] is set and 19h[3] is set: 000 = 0Kx36 001 = 256Kx36 010 = 512Kx36 011 = 1Mx36 100 = 2Mx36 101 = 4Mx36 110 = Undefined 111 = Undefined	

(1) If six DRAM banks have been chosen, the DIRTYWE# line will become RAS5# if bit 7 = 1. If six banks of DRAM are chosen, then a combined Dirty/Tag SRAM solution must be implemented or else it will not have a Dirty RAM.

(2) If five DRAM banks have been chosen, the MA11 line will become RAS4# if bit 3 = 1. If bit 3 is set to 1, none of the DRAM banks will support the 8Mx36 or 16Mx36 options.

SYSCFG 1Ah	**Memory Shadow Control Register 1**			**Default = 00h**
Reserved: Must be written to 0.	Time that CPU is ensured for bus utilization during every 15µs of system operation:(1) 00 = No bandwidth guarantee 01 = 1µs guarantee 10 = 2µs guarantee 11 = 4µs guarantee	C8000h-DFFFFFh shadowing granularity: 0 = 16KB 1 = 8KB	Read and write control of CE000h-CFFFFh for shadowing if SYSCFG 1Ah[4] = 1: 00 = Read/write PCI 10 = Read from PCI / write to DRAM 11 = Read from DRAM / write to DRAM 01 = Read from DRAM / write to PCI	Read and write control of CA000h-CBFFFh for shadowing if SYSCFG 1Ah[4] = 1: 00 = Read/write PCI 10 = Read from PCI / write to DRAM 11 = Read from DRAM / write to DRAM 01 = Read from DRAM / write to PCI

(1) Bits [6:5] allow the user to guarantee the CPU a percentage of the total available bus bandwidth, if he so desires. When these bits are programmed, the CPU is ensured of utilization of the bus for up to 4µs of every 15µs of operation of the system. This is achieved by not granting bus ownership to other requesting devices.

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 1Bh			**Memory Shadow Control Register 2**				**Default = 00h**
Read and write control of DE000h-DFFFFh for shadowing if SYSCFG 1Ah[4] = 1: 00 = Read/write PCI 10 = Read from PCI / write to DRAM 11 = Read from DRAM / write to DRAM 01 = Read from DRAM / write to PCI		Read and write control of DA000h-DBFFFh for shadowing if SYSCFG 1Ah[4] = 1: 00 = Read/write PCI 10 = Read from PCI / write to DRAM 11 = Read from DRAM / write to DRAM 01 = Read from DRAM / write to PCI		Read and write control of D6000h-D7FFFh for shadowing if SYSCFG 1Ah[4] = 1: 00 = Read/write PCI 10 = Read from PCI / write to DRAM 11 = Read from DRAM / write to DRAM 01 = Read from DRAM / write to PCI		Read and write control of D2000h-D3FFFh for shadowing if SYSCFG 1Ah[4] = 1: 00 = Read/write PCI 10 = Read from PCI / write to DRAM 11 = Read from DRAM / write to DRAM 01 = Read from DRAM / write to PCI	
SYSCFG 1Ch			**EDO DRAM Control Register**				**Default = 00h**
EDO DRAM usage: Each bit is set to a 1 if the user is using EDO DRAMs in each of the available six banks. Bit 2 corresponds to Bank 0 and bit 7 corresponds to Bank 5, yielding a total of six banks that the user can populate. 0 = Standard page mode DRAM 1 = EDO DRAM						Viper-MAX Chipset operating at a frequency of 50MHz:(1) 0 = No 1 = Yes	CAS pulse width during DRAM accesses: 0 = CAS pulse width determined by SYSCFG 01h[3] 1 = CAS pulse width is one CPUCLK(2)
SYSCFG 1Dh			**Miscellaneous Control Register 3**				**Default = 00h**
Reserved: Must be written to 0.	DWE# timing selection:(1) 0 = Normal 1 = Removed one CLK earlier	DRAM read leadoff cycle: 0 = Normal 1 = Reduced by one CLK	DMA accesses from system memory: 0 = Enable 1 = Disable	Reserved: Must be written to 1 if 08h[4] = 1 (if parity is enabled).		Accesses to B0000h-BFFFFh during SMM mode: 0 = Accesses go to main memory 1 = Accesses go to PCI bus	Accesses to A0000h-AFFFFh during SMM mode: 0 = Accesses go to main memory 1 = Accesses go to PCI bus

(1) Bit 1 is set by the user when the chipset is operating at a frequency of 50MHz. The setting of this bit could potentially improve DRAM access times even if the user is not using EDO DRAMs.

(2) The width of the pulse is one CPU clock if the Viper-MAX Chipset is operating at 50MHz (selected by the setting of bit 1) if it is interfaced to EDO DRAMs (selected by bits [7:2]).

(1) When using a buffered DWE# solution and the DRAM load is substantial, bit 5 may have to be set if the system begins to malfunction.

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 1Eh			**BOFF# Control Register**				**Default = 00h**
PCI master read cycle: 0 = Wait for IRDY# to be asserted before asserting TRDY# 1 = Generate TRDY# when checking for the status of IRDY#	Reserved: Must be written to 1.	Retry PCI pre-snoop HITM# cycle: 0 = Disable 1 = Enable	BOFF# generation if the PCI retry cycle is in A0000h-BFFFFh range: 0 = Not generated if bit 3 = 1 1 = Generated if bit 3 = 1 If bit 3 is not set to 1, then the setting of this bit has no effect.	Deadlock situation:(1) 0 = No way to avert deadlock situation if the write posting buffer on the PCI-to-PCI bridge has been enabled 1 = BOFF# is asserted to the CPU if deadlock situation occurs	Reserved: Must be written to 1.	When set to 1, PCI bursting will be disabled if BE[7:4]# and/or BE[3:0]# are not all 0.	Reserved: Must be written to 0.

(1) In a situation where there is a PCI-to-PCI bridge in a system and that bridge supports write posting, the following deadlock condition can occur. The bridge posts data from a master on the secondary PCI bus into its FIFO. If at the same time the 82C567 is accessing the bridge as a target, then the bridge will tell the 82C567 to retry its request after it has serviced out its FIFO. This will result in a deadlock situation. Bit 3 needs to be set to 1 if a DEC 21050 PCI-to-PCI bridge (or a similar chip) is being used.

7	6	5	4	3	2	1	0
SYSCFG 1Fh			**EDO Timing Control Register**				**Default = 00h**
0 = Normal 1 = Generate conflict during EDO detection (bit 6 set) if necessary	0 = Normal (fast page mode) 1 = Detect EDO	NA# generation: 0 = Aggressive (enabled with EDO or 50MHz operation and X-2-2-2 timing selection 1 = Normal	DRAM read cycle leadoff reduced by 1 clock to support 5-2-2-2 at 50MHz: 0 = No (Normal) 1 = Yes	Reserved: Must be written to 0.	Chip selects and write enables for async SRAM: 0 = 8 CS# and 1 WE# 1 = 1 CS# and 8 WE# (CS# is OCAWE# in this mode)(1) (In sync SRAM mode, ADSC# and ADV# are swapped.)	Block AHOLD during Hidden Refresh: 0 = Enable (Normal) 1 = Disable	0D0000-0DFFFFh is cacheable in L1 and L2:(2) 0 = No 1 = Yes

(1) This is only good for single bank cache. ECAWE# pin will become CS#.

(2) Before turning on bit 0, 0D0000-0DFFFFh needs to be readable/writable and shadowed. When cached into L1, it will be in write-back mode if SYSCFG 08h[1] is on. There is no write protection in this region if bit 0 is set.

7	6	5	4	3	2	1	0
SYSCFG 20h			**DRAM Burst Control Register**				**Default = 00h**
Reserved: Must be written to 1.	DRAM post write during HITM# cycle during PCI master access: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	PCI master parity: 0 = Disable 1 = Enable	DRAM write burst cycle control during PCI master cycles: 00 = Reserved 01 = X-3-3-3 10 = X-2-2-2 11 = X-1-1-1		DRAM read burst cycle control during PCI master cycles: 00 = Reserved 01 = X-3-3-3 10 = X-2-2-2 11 = X-1-1-1	

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 21h			**PCI Concurrence Control Register**				**Default = 00h**
Concurrence timer: 0 = Conservative 1 = Aggressive	00 = No concurrence on PCI master and CPU/L2 If bit 1 = 1 then: X1 = PCI master and CPU/L2 concurrence for PCI write invalid cycles 1X = PCI master and CPU/L2 concurrence for PCI read multiple and read line cycles		Reserved: Must be written to 0.	0 = Normal Tag write 1 = If bit 1 is set, always write invalid Tag during line fill.	0 = If Tag = 1101 1111b => invalid combination 1 = If cache size = 256K, Tag = 0000 1100b => invalid combination (C-F0000h). If cache size > 256K, Tag = 1011 1111b => invalid combination Valid only when bit 1 = 1.	L2 cache write mode during master cycle: 0 = Write-through 1 = Write-back	CPU bus arbitration protocol: 0 = HOLD+ HLDA 1 = BOFF#+ AHOLD
SYSCFG 22h			**Inquire Cycle Control Register**				**Default = 00h**
0 = AHOLD+ BOFF# 1 = HOLD+ HLDA If SYSCFG 21h[0] = 0, this bit must be set to 1. If 21h[0] = 1, this bit must be set to 0.	OCAWE# and OCDOE# pin functionality in single bank cache: 0 = No change 1 = OCAWE# becomes ECA4 and OCDOE# becomes ECA3(1)	Reserved: Must be written to 0.	HRQ is sync to LCLK: 0 = No 1 = Yes (Must = 1 for DDMA operation)	Reserved: Must be written to 0.	EADS# generation: 00 = Normal for inquire cycle 01 = 1 CPU CLK earlier 10 = 1 CPU CLK earlier with async clock 2 CPU CLK earlier with sync clock 11 = Reserved		Single write hit leadoff cycle in a combined Dirty/Tag implementation:(2) 0 = 5 cycles 1 = 4 cycles
(1) This is to reduce cache address 0/1 (CA4/3) loading to improve timing if necessary. Do not set bit 6 if two banks of cache are in the system.							
(2) If bit 1 is set 1, SYSCFG 16h[4] should be set to 1.							
SYSCFG 23h			**Pre-Snoop Control Register**				**Default = 00h**
Generate internal BREAK signal during master accessing of local memory cycle:(1) 0 = Old Mode 1 = New Mode	Bank 0 is selected as first or last bank: 0 = First bank 1 = Last bank	Pre-snoop for PCI X-1-1-1 write invalidate: 0 = Disable 1 = Enable	Pre-snoop for PCI X-1-1-1 read multiple and read line: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	Reserved: Must be written to 1.	Two clock MREQ# high extention by additional two clocks: 0 = Disable 1 = Enable **Note:** Set this bit to 1 in UMA systems	Reserved: Must be written to 0.
(1) Old Mode conditions: Sync SRAM, starting address AD[4:2] not = 000 or non-linear mode, master L2 cache write-through New Mode conditions: Sync SRAM, starting address AD[4:2] not = 000 or non-linear mode, master L2 cache write-through, L2 cache hit							

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 24h			**Asymmetric DRAM Configuration Register**				**Default = 00h**
Logical Bank 3 DRAM type: 00 = Sym DRAM 01 = Asym DRAM - x8 type 10 = Asym DRAM - x9 type 11= Asym DRAM - x10 type		Logical Bank 2 DRAM type: 00 = Sym DRAM 01 = Asym DRAM - x8 type 10 = Asym DRAM - x9 type 11= Asym DRAM - x10 type		Logical Bank 1 DRAM type: 00 = Sym DRAM 01 = Asym DRAM - x8 type 10 = Asym DRAM - x9 type 11= Asym DRAM - x10 type		Logical Bank 0 DRAM type: 00 = Sym DRAM 01 = Asym DRAM - x8 type 10 = Asym DRAM - x9 type 11= Asym DRAM - x10 type	
SYSCFG 25h			**GUI Memory Location Register**				**Default = 00h**
GUI memory location: A[31:27]					UMA size: 0 = Decided by SYSCFG 26h[5:4] 1 = 0.5MB if SYSCFG 26h[5:4] = 00	Reserved: Must be written to 0.	
SYSCFG 26h			**UMA Control Register**				**Default = 00h**
ISA master to DRAM cycle CAS width: 0 = Controlled by ISA R/W command pulse width 1 = 2 LCLKs	ISA SA address latch: 0 = SA latch is always transparent (pass-through) 1 = SA latch is on for retry only. (When the first CPU/ISA cycle is retried, the SA address will be latched.)	GUI memory size: 00 = 1MB 01 = 2MB 10 = 3MB 11= 4MB For 0.5MB size, set these bits to 00 and SYSCFG 25h[2] = 1.		5-2-2-2 EDO DRAM read timing at 66MHz: 0 = Disable 1 = Enable	00 = Normal 01 = For low priority GUI request, 82C567 will wait for two more CLKs 11 = GUI is always at high priority		UMA support: 0 = Disable 1 = Enable
SYSCFG 27h			**Self Refresh Timing Register**				**Default = 00h**
Reserved: Must be written to 0.			Generate AHOLD at 2nd T2 on CPU single write hit not dirty cycle: 0 = Disable 1 = Enable	Fast NA# with L2 cache: 0 = Disable 1 = Enable	Self refresh: 000 = Disable, use external refresh pin 001 = Reserved 010 = Reserved 011 = Reserved 100 = 66MHz external CPU clock 101 = 60MHz external CPU clock 110 = 50MHz external CPU clock 111 = 40MHz external CPU clock		
SYSCFG 28h			**SDRAM Burst and Latency Control Register**				**Default = 00h**
Reserved: Must be written to 0.	SDRAM CAS# latency: 000 = Reserved 001 = 1 010 = 2 011 = 3	100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved		Reserved: Must be written to 0.	Burst length control: 000 = 1 001 = 2 010 = 4 011 = 8	100 = Reserved 101 = Reserved 110 = Reserved 111 = Full page	

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 29h			**SDRAM Selection Register**				**Default = 00h**
When set to 1, at least one CLK will be forced between the current command and next cycle.(1)	Controls latency between bank active and command at 50MHz: 0 = 2 CLK latency 1 = 1 CLK latency	SDRAM pre-charge control: 00 = 2 CLKs 10 = 3 CLKs 01 = 4 CLKs 11 = Reserved		SDRAM enable on each bank: 0 = Disable 1 = Enable			
(1) Otherwise, if next cycle is a page-and/or bank-miss, it could start right after the current command has been issued.							
SYSCFG 2Ah			**PCI-to-DRAM Deep Buffer Size Register**				**Default = 00h**
Reserved: Must be written to 0.	Time-out selection when there is a GUI request during PCI master read cycles: 00 = Always FP mode, grant DRAM bus to GUI ASAP 01 = Select SDRAM or EDO time-out depending on current bank information 1X = Select FP mode, SDRAM, or EDO depending on current bank information	PCI TRDY# wait state control with PCI-DRAM deep buffer: 0 = Zero wait state (X-1-1-1) 1 = One wait state (X-2-2-2)	Write burst with PCI-DRAM deep buffer: 0 = Disable 1 = Enable	Read burst with PCI-DRAM deep buffer: 0 = Disable 1 = Enable	PCI-to-DRAM deep buffer size: 00 = 16 dword 01 = 24 dword 10 = Reserved 11 = Reserved		
SYSCFG 2Bh			**EDO/SDRAM Time-Out Register**				**Default = 00h**
SDRAM time-out count when there is a GUI request: The register value plus 9 is the number of CPU clocks delaying the GUI request to stop the DRAM controller.				EDO time-out count when there is a GUI request: The register value plus 6 is the number of CPU clocks delaying the GUI request to stop the DRAM controller.			
SYSCFG 2Ch			**CPU-to-DRAM Buffer Control Register**				**Default = 00h**
Reserved: Must be written to 0.					BOFF# assertion for DRAM read cycles: 0 = Disable 1 = Enable	Data merge when CPU has ownership of DRAM bus: 0 = Disable 1 = Enable	Buffer write data while buffer is flushing: 0 = Disable 1 = Enable
SYSCFG 2Dh			**Bank-wise EDO Timing Selection Register**				**Default = 00h**
Split buffer concurrency: 0 = Disable 1 = Enable	Predictive reading: 0 = Disable (normal) 1 = Enable	Bank-wise selection for 5-X-X-X at 66MHz or 4-X-X-X at 50MHz EDO DRAM read cycle: 0 = Default setting 1 = 5-X-X-X/4-X-X-X enabled					

Table 5-5 82C567 System Configuration Registers: SYSCFG 00h-2Fh (cont.)

7	6	5	4	3	2	1	0
SYSCFG 2Eh			**PCI Master - GUI Retry Control Register**				**Default = 00h**
Pin 189 functionality: 0 = DIRTYWE# or RAS5# 1 = SDCKE	Pin 100 functionality: 0 = USBCLK 1 = REFRESH#	Pin 121 functionality and MSGN2S/ MSGS2N bus enabling: 0 = AEN 1 = MSGN2S	Reserved: Must be written to 0.	CPU-to-PCI FIFO control module: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	PCI Master HITM# cycle, if GUI high priority request jumps in before first BRDY#: 0 = Retry all PCI cycles 1 = Retry only PCI master read	PCI master requests retried during GUI cycles: 0 = All PCI master requests are retried 1 = PCI master reads are retried, writes are accepted
SYSCFG 2Fh			**CAS Address Setup Time Control Register**				**Default = 00h**
Column address to CAS delay for page miss cycles: 0 = Default 1 = 1 CLK	Reserved: Must be written to 0.		Generation of NA# during CPU accesses to non-shared DRAM bank when GUI has the memory bus: 0 = Enable 1 = Disable	Reserved: Must be written to 0.			

5.2.3 82C567 I/O Register Space

The I/O Register Space of the 82C567 is accessed normally (i.e., CPU Direct I/O R/W). There is no indexing scheme. Table 5-6 shows these registers.

Table 5-6 82C567 I/O Register Space

7	6	5	4	3	2	1	0
I/O Address 8Fh			**Refresh Page Register (Write-Only)**				
Values programmed into this register gets driven on LA[23:17] during refresh cycles. It is recommended that the user write 00h into this register.							
I/O Address CF8h			**Configuration/NVM Address Register**				
32-bit writes to this register are captured by the internal configuration address register of the 82C567. When bit 31 of this register is set, access to configuration data register is translated to PCI configuration or special cycle according to Configuration Mechanism #1. When bit 31 of this register is reset, access to configuration data register goes out as I/O access to the PCI bus. When bit 31 of this register is reset and bits [30:14] are set, NVMCS is generated during Configuration Data Register access and the content of Configuration Address Register bits [13:0] is put out as address [15:12, 9:0] and address [11:10] are driven to 0 on PCI and ISA buses. When bit 31 is reset and any of the bits [30:14] are reset, access to the Configuration Data Register proceeds as a normal I/O cycle.							
I/O Address CFCh			**Configuration/NVM Data Register**				

5.3 82C568 Register Space

This section details the locations, access mechanisms, and bit formats for registers located within the 82C568.

5.3.1 82C568 PCI Configuration Register Space (PCIDV1)

The PCI Configuration Register Space of the 82C568 (PCIDV1) is accessed through Configuration Mechanism #1 as Bus #0, Device #1, and Function #0. Table 5-7 gives the bit formats for the registers accessed in PCIDV1. All bits are read/write and their default value is 0 unless otherwise specified.

Table 5-7 82C568 PCI Configuration Registers: PCIDV1 00h-FFh

7	6	5	4	3	2	1	0
PCIDV1 00h **PCIDV1 01h**	**Vendor Identification Register (RO)**						**Default = 45h** **Default = 10h**
PCIDV1 02h **PCIDV1 03h**	**Device Identification Register (RO)**						**Default = 68h** **Default = C5h**
PCIDV1 04h	**Command Register - Byte 0**						**Default = 07h**
Address/data stepping (RO): 0 = Disable (always)	PERR# output pin: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	Memory write and invalidate cycle generation (RO): Must = 0 (always) No memory write and invalidate cycles will be generated by the 82C568.	Special cycles: 0 = Disable 1 = Enable The 82C568 responds to Stop Grant special cycle.	Bus master operations: 0 = Disable 1 = Enable PCI cycle generation during DMA/ISA master may be disabled by this bit. (Default = 1)	Memory access (RO): Must = 1 (always) The 82C568 allows a PCI bus master access to memory at anytime. (Default = 1)	I/O access (RO): Must = 1 (always) The 82C568 allows a PCI bus master I/O access at any time. (Default = 1)
PCIDV1 05h	**Command Register - Byte 1**						**Default = 00h**
Reserved: Must be written to 0.						Fast back-to-back to different slaves (RO): 0 = Disable (always)	SERR# output pin: 0 = Disable 1 = Enable
PCIDV1 06h	**Status Register - Byte 0**						**Default = 80h**
Fast back-to-back capability (RO): 0 = Not Capable 1 = Capable (Default = 1)	Reserved: Must be written to 0.						
PCIDV1 07h	**Status Register - Byte 1 (RO)**						**Default = 02h**
Parity error status (RO): 0 = No parity error 1 = Parity error has occurred	SERR# status (RO): 0 = No system error 1 = System error has occurred	Master abort status (RO): Must = 0 (always)	Received target abort status (RO): 0 = No target abort 1 = Target abort occurred	Signaled target abort status (RO): Must = 0 (always)	DEVSEL# timing status (RO): Must = 01 (always) Indicates medium timing selected; the 82C568 asserts the DEVSEL# based on medium timing. (Default = 01)		Data parity status (RO): 0 = No data parity detected 1 = Data parity detected

Table 5-7 82C568 PCI Configuration Registers: PCIDV1 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIDV1 08h			**Revision Identification Register (RO)**				**Default = 12h**
PCIDV1 09h **PCIDV1 0Ah** **PCIDV1 0Bh**			**Class Code Register (RO)**				**Default = 00h** **Default = 01h** **Default = 06h**
PCIDV1 0Ch			**Reserved Register** Reserved: Must be written to 0.				**Default = 00h**
PCIDV1 0Dh			**Master Latency Timer Register (RO)**				**Default = 00h**
PCIDV1 0Eh			**Header Type Register (RO)**				**Default = 00h**
PCIDV1 0Fh			**Built-In Self-Test (BIST) Register (RO)**				**Default = 00h**
PCIDV1 10h-3Fh			**Reserved Register (RO)** Reserved: Must be written to 0.				**Default = 00h**
PCIDV1 40h			**Keyboard Control Register - Byte 0**				**Default = 00h**
Selects which IRQ signal is to be generated when PIRQ2# has been triggered: See 41h[3:1] for decode.		Selects which IRQ signal is to be generated when PIRQ1# has been triggered: See 41h[3:1] for decode.			Selects which IRQ signal is to be generated when PIRQ0# has been triggered: See 41h[3:1] for decode.		
PCIDV1 41h			**Keyboard Control Register - Byte 1**				**Default = 00h**
Keyboard port read (RO): 0 = Does not say anything 1 = Keyboard controller has received command D0h and has not received the following 60h read	Keyboard port write (RO): 0 = Does not say anything 1 = Keyboard controller has received command D1h and has not received the following 60h write	Immediate INIT generation: 0 = Generate INIT immediately on FEh command 1 = Wait for halt before generating INIT on receiving the keyboard RESET	Keyboard emulation: 0 = Enable - Pin 12 functions as A20M# output 1 = Disable - Pin 12 functions as KBRST input	Selects which IRQ signal is to be generated when PIRQ3# has been triggered: 000 = IRQx* 001 = IRQ5 010 = IRQ9 011 = IRQ10 100 = IRQ11 101 = IRQ12 110 = IRQ14 111 = IRQ15 *Selection controlled by PCIDV1 50h[7:0].			Works with PCIDV1 40h[7:6] –
PCIDV1 42h			**Interrupt Edge/Level Control Register - Byte 0**				**Default = 00h**
Triggering for IRQ15:(1) 0 = Edge 1 = Level	Triggering for IRQ14:(1) 0 = Edge 1 = Level	Triggering for IRQ12:(1) 0 = Edge 1 = Level	Triggering for IRQ11:(1) 0 = Edge 1 = Level	Triggering for IRQ10:(1) 0 = Edge 1 = Level	Triggering for IRQ9:(1) 0 = Edge 1 = Level	Triggering for IRQ5:(1) 0 = Edge 1 = Level	Pin 122 functionality: 0 = DREQ6 1 = EPMI0#
(1) If a PCI interrupt is routed to a particular ISA IRQ, then that ISA IRQ needs to become level triggered (except IRQ14 for IDE).							

Table 5-7 82C568 PCI Configuration Registers: PCIDV1 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIDV1 43h			**Interrupt Edge/Level Control Register - Byte 1**				**Default = 00h**
ISA IRQ14 recognition control: To use IDE on the PCI bus, the ISA IRQ14 signal needs to be blocked. 0 = ISA IRQ14 is honored 1 = ISA IRQ14 is ignored	ISA IRQ15 recognition control: To use secondary IDE on the PCI bus, the ISA IRQ15 signal needs to be blocked. 0 = ISA IRQ15 is honored 1 = ISA IRQ15 is ignored	DMA/ISA master to preempt PCI master: 0 = Disable 1 = Enable	Fixed/rotating priority between PCI masters: 0 = Rotating 1 = Fixed, priority sequence is PREQ0#, PREQ1#, PREQ2#	Back-to-back ISA I/O cycle delay: 00 = Delay by 3 ATCLKs 01 = Delay by 12 ATCLKs 10 = No delay 11 = Delay by 12 ATCLKs(1)		PCI master access to ISA devices: 0 = Enable 1 = Disable	ISA bus control signals for memory access greater than 16M and for I/O accesses greater than 64K: 0 = Enable 1 = Disable

(1) When bits [3:2] take on the combination of 11, all back-to-back cycles are delayed by 12 AT clocks. This is different from the combinations of 00 and 01 because in the latter case, the delay will be inserted only when an I/O access is followed by a second I/O access with no other type of access occurring in between (e.g., a memory access).

PCIDV1 44h		**Pin Functionality Register 1 - Byte 0**	**Default = 00h**
Pin 111 functionality:(1) 0X = Controlled by bits [1:0] 10 = DACK7# 11 = Reserved If set to 10, the setting on bits [1:0] will not affect the functionality that this pin takes on.	Pin 109 functionality:(2) 0X = Controlled by bits [1:0] 10 = DACK6# 11 = Reserved If set to 10, the setting on bits [1:0] will not affect the functionality that this pin takes on.	Pin 108 functionality: 0X = Controlled by bits [1:0] 10 = DACK5# 11 = GPCS0# or PPWRL1 To select the GPCS0# or PPWRL1 function set these bits to 11 and PCIDV1 44h[1:0] to 11. Then set SYSCFG FDh[1] = 1 for GPCS0# or SYSCFG FDh[1] = 0 for PPWRL1.	DACK/PIRQ[3:2]# group-wise programmable pin functionalities: 00 = Explicit DACK[3:0]#, PIRQ[3:2]# 01 = Explicit DACK[7:5,3,1,0]#, GPCS0#, PIRQ[3:2]# 10 = Encoded EDACK[2:0], EDACKEN# 11 = Encoded EDACK[2:0], EDACKEN#, PIRQ[3:2]# Pin-wise, these functions may be overridden by GPCS[x]#, EPMI[x]#, and DACK[7:5]# (for DACK3#, DACK1#, and DACK0#).

(1) Pin 111 can take on the following functionalities - DACK3#, EDACK2, or DACK7#. DACK3# and EDACK2 are group-wise programmable, and both are pin-wise programmable with DACK7#.

(2) Pin 109 can take on the following functionalities - DACK1#, EDACK1, or DACK6#. DACK1# and EDACK1 are group-wise programmable, and both of them are pin-wise programmable with DACK6#.

Table 5-7 82C568 PCI Configuration Registers: PCIDV1 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIDV1 45h			**Pin Functionality Register 1 - Byte 1**				**Default = 00h**
Reserved: Must be written to 0.		Pin 143 functionality:(1) If PCIDV1 51h[4] = 0: 0 = Controlled by PCIDV1 44h[1:0] 1 = EPMI2# (If this bit is set and 51h[4] = 0, pin 143 takes on the EPMI2# functionality regardless of the setting of 44h[1:0])	Pin 141 functionality: This bit determines the group-wise functionality of the PIRQ2#+ GPCS0#. These two functionalities are group-wise programmable. 0 = Controlled by PCIDV1 44h[1:0] and PCIDV1 51h[3] = 0 1 = Reserved	Pin 140 functionality: 00 = PIRQ1# 01 = IRQ0 1X = Reserved		Pin 139 functionality: 00 = PIRQ0# 01 = EPMI1# 1X = Reserved	

(1) This pin can take on any of the these functionalities - PIRQ3#, GPCS1#, or EPMI2#. PIRQ3# and GPCS1# functionalities are group-wise programmable, and those two functionalities are pin-wise programmable with EPMI2#.

PCIDV1 46h			**Cycle Control Register 1 - Byte 0**				**Default = 00h**
DMA/ISA access to PCI slave: 0 = Never 1 = When LMEM# is not asserted Master retry always unmasked after 16 LCLKs.	XDIR control: 0 = XDIR is achieved for accesses to/ from ROM, Kybd controller, RTC, or NVRAM 1 = XDIR achieved only during access to/from ROM or NVRAM	Conversion of PERR# to SERR#: 0 = Disable 1 = Enable	Address parity checking: 0 = Disable 1 = Enable	SERR# generation for target abort: 0 = Disable 1 = Enable	Fast back-to-back capability: 0 = Disable 1 = Enable (Default = 1) A change on this bit will reflect on PCIDV1 06h[7]	Subtractive decoding sample point: 0 = Typical sample point 1 = Slow sample point	Reserved: Must be written to 0.

PCIDV1 47h			**Cycle Control Register 1 - Byte 1**				**Default = 00h**
Write protect ISA bus ROM (ROMCS# for writes): 0 = Enable 1 = Disable	Refresh select: 0 = Normal 1 = Hidden	ATCLK frequency select: 00 = LCLK÷4 10 = LCLK÷2 01 = LCLK÷3 11 = LCLK		CPU master to PCI slave write (turnaround between address and data phases): 0 = 1 LCLK 1 = 0 LCLK	PCI master to PCI master preemption timer (preempt after unserviced request pending for X LCLKs): 000 = No preemption 100 = 36 LCLKs 001 = 260 LCLKs 101 = 20 LCLKs 010 = 132 LCLKs 110 = 12 LCLKs 011 = 68 LCLKs 111 = 5 LCLKs		

PCIDV1 48h		**Pin Functionality Register 2 - Byte 0**		**Default = 00h**
Pin 120 functionality: 00 = DREQ3 01 = DREQ3/7 10 = DREQ7 11 = Reserved	Pin 117 functionality: 00 = DREQ1 01 = DREQ1/6 10 = DREQ6 11 = Reserved	Pin 116 functionality: 00 = DREQ0 01 = DREQ0/5 10 = DREQ5 11 = Reserved	Pin 145 functionality: 0X = PREQ1# 1X = Reserved Bit 1 of this register pair must be 0 for the PREQ1# functionality. All other combinations are reserved.	

Table 5-7 82C568 PCI Configuration Registers: PCIDV1 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIDV1 49h			**Pin Functionality Register 2 - Byte 1**				**Default = 00h**
Pin 136 functionality: 0 = IRQ15 1 = Reserved	Pin 134 functionality: 0X = IRQ12 10 = MPIRQ2#/3# 11 = Reserved		Pin 132 functionality: 0 = IRQ10 1 = MIRQ10/12	Pin 128 functionality: 0 = IRQ6 1 = MPIRQ0#/1#	Pin 126 functionality: 0 = IRQ4 1 = MIRQ4/6	Pin 123 functionality: 00 = DREQ7 01 = EPMI1# 1X = Reserved	
PCIDV1 4Ah			**ROMCS# Range Control Register - Byte 0**				**Default = 00h**
ROMCS# for F8000h-FFFFFh: 0 = Enable 1 = Disable	ROMCS# for F0000h-F7FFFh: 0 = Enable 1 = Disable	ROMCS# for E8000h-EFFFFh: 0 = Disable 1 = Enable	ROMCS# for E0000h-E7FFFh: 0 = Disable 1 = Enable	ROMCS# for D8000h-DFFFFh: 0 = Disable 1 = Enable	ROMCS# for D0000h-D7FFFh: 0 = Disable 1 = Enable	ROMCS# for C8000h-CFFFFh: 0 = Disable 1 = Enable	ROMCS# for C0000h-C7FFFh: 0 = Disable 1 = Enable
PCIDV1 4Bh			**ROMCS# Range Control Register - Byte 1**				**Default = 00h**
ROMCS# for FFFF8000h-FFFFFFFFh segment: 0 = Enable 1 = Disable	ROMCS# for FFFF0000h-FFFF7FFFh segment: 0 = Enable 1 = Disable	ROMCS# for FFFE8000h-FFFEFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFE0000h-FFFE7FFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFD8000h-FFFDFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFD0000h-FFFD7FFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFC8000h-FFFCFFFFh segment: 0 = Disable 1 = Enable	ROMCS# for FFFC0000h-FFFC7FFFh segment: 0 = Disable 1 = Enable
PCIDV1 4Ch-4Dh			**Reserved Register 3** Reserved: Must be written to 0.				**Default = 00h**
PCIDV1 4Eh			**Miscellaneous Control Register - Byte 0**				**Default = 00h**
Reserved: Must be written to 0.				Pipelining with byte merge: 0 = Disable 1 = Enable	EOP configuration: 0 = Output 1 = Input	Byte merging: 0 = Disable 1 = Enable	ISA master data swap: 0 = Enable 1 = Disable
PCIDV1 4Fh			**Miscellaneous Control Register - Byte 1**				**Default = 00h**
Pin 112 functionality: 0 = DACK5# (also see PCIDV1 44h[1:0]) 1 = PPWRL# +PPWRL2	IDE functionality support: 0 = Disable 1 = Enable	82C568 silicon revision usage in system (RO): 0 = 82C568 Rev 0	Extend the cycle on the ISA bus by inserting one wait state in AT commands:(1) 0 = Yes 1 = No	Reserved: Must be written to 0.	Reserved: Must be written to 0.	Pin 113 functionality: 0 = Controlled by PCIDV1 44h[1:0] 1 = GPCS2# If set to 1, the setting on PCIDV1 44h[1:0] will not affect the functionality that this pin takes on.	Clock source for multiplexing/demultiplexing IRQs: 0 = 14.318MHz clock 1 = LCLK

(1) Usually a cycle on the ISA bus, during 8-bit accesses, consists of a command phase that is one clock long, followed by four wait states, and then finally followed by a one clock long data phase. Setting this bit to a 0 will cause the number of wait states inserted in the cycle to increase by one to five clocks, making the duration of the entire cycle to seven clocks instead of six clocks.

Table 5-7 82C568 PCI Configuration Registers: PCIDV1 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIDV1 50h		**Interrupt Trigger Control Register - Byte 0**					**Default = 00h**
Selects which IRQ signal is generated when PIRQ3# has been triggered: 00 = Disable 10 = IRQ4 01 = IRQ3 11 = IRQ7 If PIRQ3# is routed onto any of these ISA IRQs, ensure that PCIDV1 41h[3:1] = 000.		Selects which IRQ signal is generated when PIRQ2# has been triggered: 00 = Disable 10 = IRQ4 01 = IRQ3 11 = IRQ7 If PIRQ2# is routed onto any of these ISA IRQs, ensure that PCIDV1 41h[0] and PCIDV1 40h[7:6] = 000.		Selects which IRQ signal is generated when PIRQ1# has been triggered: 00 = Disable 10 = IRQ4 01 = IRQ3 11 = IRQ7 If PIRQ1# is routed onto any of these ISA IRQs, then make sure that PCIDV1 40h[5:3] = 000.		Selects which IRQ signal is generated when PIRQ0# has been triggered: 00 = Disable 10 = IRQ4 01 = IRQ3 11 = IRQ7 If PIRQ0# is routed onto any of these ISA IRQs, then make sure that PCIDV1 40h[2:0] = 000.	
PCIDV1 51h		**Interrupt Trigger Control Register - Byte 1**					**Default = 00h**
Pin 104 functionality:(1) 0 = 32KHz 1 = PREQ3#	Pin 90 functionality: 0 = ZEROWS# 1 = PGNT3#	Pin 110 functionality: 0 = DACK2# 1 = GPCS2#	Pin 143 functionality: 0 = PIRQ3# 1 = GPCS1#	Pin 141 functionality: 0 = PIRQ2# 1 = GPCS0#	Triggering for IRQ3: 0 = Edge 1 = Level	Triggering for IRQ4: 0 = Edge 1 = Level	Triggering for IRQ7: 0 = Edge 1 = Level
(1) For SDRQ2 function on this pin, see PCIDV1 5Eh[6].							
PCIDV1 52h		**Interrupt Multiplexing Control Register - Byte 0**					**Default = 00h**
Triggering for IRQ6: 0 = Edge 1 = Level	IRQ signal generation when GMIRQ is triggered: 0000 = Disabled 1000 = Reserved 0001 = Reserved 1001 = IRQ9 0010 = Reserved 1010 = IRQ10 0011 = IRQ3 1011 = IRQ11 0100 = IRQ4 1100 = IRQ12 0101 = IRQ5 1101 = Reserved 0110 = IRQ6 1110 = IRQ14 0111 = IRQ7 1111 = IRQ15				Reserved: Must be written to 1.	Priority scheme: 0 = Disable 1 = Enable A setting of 1 will employ a priority scheme that guarantees higher priority for PCI masters during arbitration over DMA and ISA masters for the first 7µs interval after every refresh cycle.	Concurrent refresh and IDE cycle: 0 = Disable 1 = Enable ISA devices that rely on accurate refresh addresses for proper operation should disable this bit.
PCIDV1 53h		**Interrupt Multiplexing Control Register - Byte 1**					**Default = 00h**
Pin functionality: 0 = Pin #: 125 = IRQ3 127 = IRQ5 128 = IRQ6 129 = IRQ7 131 = IRQ9 133 = IRQ11 1 = Pin #: 125 = MIRQ3/5 127 = MIRQ7/9 128 = MIRQ11/15 129 = EPMI1# 131 = EPMI2# 133 = GMIRQ	Pin 146 functionality: 0 = PREQ2# 1 = EPMI0#	IRQ1 latching: 0 = Disable 1 = Enable	IRQ12 latching: 0 = Disable 1 = Enable	DACKEN#: 0 = Active low 1 = Active high	External device owns system bus: 0 = No 1 = Yes If set to 1, preemption of PGNT0# is disabled, EOP is tristated, BALE is driven high, and internal decoding of the DMA subsystem registers is blocked.	Locking of flash ROM: 0 = Disable 1 = Enable Setting this bit to a 1 will block writes to flash ROM until the next RESET pulse	Reserved: Must be written to 0.

Table 5-7 82C568 PCI Configuration Registers: PCIDV1 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIDV1 54h			**PCI Master Control Register - Byte 0**				**Default = 00h**
PCI master write X-1-1-1: 0 = Disable 1 = Enable	PCI master read X-1-1-1: 0 = Disable 1 = Enable	PCI master/IDE concurrence: 0 = Disable 1 = Enable (Also see IDE 42h[3])	New AHOLD protocol: 0 = Disable 1 = Enable (use HREQ to latch AHOLD)	Non-contiguous byte enables for PCI masters: 0 = Disable 1 = Enable	Reserved: Must be written to 0.	Hardware PMU and IDE functions operate simultaneously: 0 = Disable 1 = Enable	ISA refresh: 0 = Enable 1 = Disable, to increase PCI master bandwidth
PCIDV1 55h			**PCI Master Control Register - Byte 1**				**Default = 00h**
Reserved: Must be written to 0.			SERIRQ# muxing on Pin 1: 0 = Disable 1 = Enable(1)	Interrupt request register recover: 0 = Disable 1 = Enable	Select DMA current or base address and counter to be read: 0 = Current 1 = Base	ISA retry for CPU/PCI master access ISA cycle: 0 = Disable 1 = Enable	Use of AHOLD signal during CPU-to-PCI cycles:(2) 0 = Disable 1 = Enable

(1) Also these PCIDV1 register bits must be set: 54h[4] = 1, 59h[3] = 0, and 5Fh[4] = 0.

(2) Bit 0 is used only if PCIDV1 54h[4] = 1.

7	6	5	4	3	2	1	0
PCIDV1 56h			**Serial Interrupt Source Register - Byte 0**				**Default = 00h**
Interrupt resource for IRQ Ch. 7: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 6: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 5: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 4: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 3: 0 = ISA 1 = Serial interrupt	Interrupt resources for SMI#, IOCHK#, PCIRQ[3:0]#: 0 = Original 1 = Serial interrupt	Interrupt resource for IRQ Ch. 1: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 0: 0 = ISA 1 = Serial interrupt
PCIDV1 57h			**Serial Interrupt Source Register - Byte 1**				**Default = 00h**
Interrupt resource for IRQ Ch. 15: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 14: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 13: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 12: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 11: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 10: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 9: 0 = ISA 1 = Serial interrupt	Interrupt resource for IRQ Ch. 8: 0 = ISA 1 = Serial interrupt
PCIDV1 58h			**Serial Interrupt Mode Control Register**				**Default = 00h**
Serial interrupt control mode: 00 = Continuous mode 01 = Idle mode 1x = Active mode	Reserved: Must be written to 0.	Data frame slot support:(1) 0 = 17 slots 1 = 21 slots	Reserved: Must be written to 0.			Serial interrupt start frame pulse width control: 00 = 4 CLK in Continuous mode or 3 CLK in Active mode 01 = 6 CLK in Continuous mode or 5 CLK in Active mode 10 = 8 CLK in Continuous mode or 7 CLK in Active mode 11 = Reserved	

(1) 17 slot support: IRQ[1:0], IRQ[15:3], SMI#, and IOCHK#
21 slot support: All of the above plus PCIRQ[3:0]#

Table 5-7 82C568 PCI Configuration Registers: PCIDV1 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIDV1 59h			**Pin Functionality Register 3**				**Default = 00h**
Reserved: Must be written to 0.	Zero wait state CPU R/W for I/O accesses: 0 = Disable 1 = Enable	SMI output control: 0 = SMI enabled 1 = SMI disabled	Reserved: Must be written to 0.	EPMI0# muxing on Pin 1: 0 = Disable 1 = Enable(1)	Refresh preemption: 0 = Enable 1 = Disable	Reserved: Must be written to 0.	Pin 106 functionality:(2) 0 = RTCRD# 1 = PGNT3#
(1) Also these PCIDV1 register bits must be set: 54h[4] = 1, 55h[4] = 0, and 5Fh[4] = 0.							
(2) For SDACK2# function on this pin, see PCIDV1 5Eh[6].							
PCIDV1 5Ah-5Bh			**Distributed DMA Master Base Address Register**				**Default = 00h**
PCIDV1 5Ch			**Distributed DMA Control Register**				**Default = 00h**
Channel 7: 0 = Disable 1 = Enable	Channel 6: 0 = Disable 1 = Enable	Channel 5: 0 = Disable 1 = Enable	Channel 3: 0 = Disable 1 = Enable	Channel 2: 0 = Disable 1 = Enable	Channel 1: 0 = Disable 1 = Enable	Channel 0: 0 = Disable 1 = Enable	Master Distributed DMA (DDMA): 0 = Disable 1 = Enable
PCIDV1 5Dh			**Reserved Register** Reserved: Must be written to 0.				**Default = 00h**
PCIDV1 5Eh			**Steerable DRQ Control Register**				**Default = 00h**
SDRQ/SDACK# functions on pins 105 and 107: 0 = Disable 1 = Enable(1)	SDRQ/SDACK# functions on pins 104 and 106: 0 = Disable 1 = Enable(2)	DRQ/DACK signal generation when SDRQ1/SDACK1# is triggered: 000 = Disable 001 = DRQ1/DACK1# 010 = DRQ2/DACK2# 011 = DRQ3/DACK3# 100 = DRQ0/DACK0# 101 = DRQ5/DACK5# 110 = DRQ6/DACK6# 111 = DRQ7/DACK7#		DRQ/DACK signal generation when SDRQ2/SDACK2# is triggered: 000 = Disable 001 = DRQ1/DACK1# 010 = DRQ2/DACK2# 011 = DRQ3/DACK3# 100 = DRQ0/DACK0# 101 = DRQ5/DACK5# 110 = DRQ6/DACK6# 111 = DRQ7/DACK7#			
(1) Pin 105 functions as SDRQ1 and pin 107 functions as SDACK1#.							
(2) Pin 104 functions as SDRQ2 and pin 106 functions as SDACK2#. Also refer to PCIDV1 register bits 51h[7] and 59h[0].							
PCIDV1 5Fh			**Steerable IRQ Control Register**				**Default = 00h**
Pin 52 functionality: 0 = Reserved 1 = MSGN2S	Pin 154 functionality: 00 = Reserved 01 = Reserved 10 = MSGS2N 11 = USBGNT#	SIRQ muxing on Pin 1: 0 = Disable 1 = Enable(1)	IRQ signal generation when SIRQ is triggered: 0000 = Disabled 0001 = Reserved 0010 = Reserved 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5 0110 = IRQ6 0111 = IRQ7 1000 = Reserved 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = IRQ12 1101 = Reserved 1110 = IRQ14 1111 = IRQ15				
(1) Also these PCIDV1 register bits must be set: 54h[4] = 1, 55h[4] = 0, and 59h[3] = 0.							

Table 5-7 82C568 PCI Configuration Registers: PCIDV1 00h-FFh (cont.)

7	6	5	4	3	2	1	0
PCIDV1 60h			**USB Interrupt Control Register**				**Default = 00h**
IOCHRDY generation through MDLE#: 0 = Enable 1 = Disable	Buffered DMA (data transfer to/ from DRAM thru PCI master) 00 = Original DMA with old protocol 01 = Reserved 10 = Original DMA with PCI master capability 11 = Buffered DMA enable		Pins 105 and 107 functionality: 0 =RTCAS+ SDRQ1 on pin 105 and RTCWR#+ SDACK1# on pin 107 1 =PREQ4# on pin 105 and PGNT4# on pin 107	IRQ signal generation when USBIRQ is triggered: 0000 = Disabled 0001 = Reserved 0010 = Reserved 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5 0110 = IRQ6 0111 = IRQ7		1000 = Reserved 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = IRQ12 1101 = Reserved 1110 = IRQ14 1111 = IRQ15	
PCIDV1 61h-FCh			**Reserved Register** Reserved: Must be written to 0.				**Default = 00h**
PCIDV1 FDh			**Reserved Register**				**Default = xxh**
PCIDV1 FEh			**Stop Grant Cycle Control Register** A byte write to this register indicates a Stop Grant cycle. The data is "don't care". A read from this register is undefined. The 82C567 propagates the CPU Stop Grant cycle as a configuration write to this register. The BIOS should never write to this configuration register. The 82C568 should ignore any write to this space unless it has driven STPCLK# active.				**Default = xxh**
PCIDV1 FFh			**Host Memory Parity Error Register** A byte write to this register indicates that a host memory parity error has occurred. A read from this register is undefined. The 82C567 propagates any host memory parity error it detects as a configuration write to this register.				**Default = xxh**

Table 5-8 Pin Multiplexing Chart 1

	SYSCFG																		
	42h		44h									48h				4Fh	45h		
Default Pin Names	Bit 0	Bits [7:6]	Bits [5:4]		Bits [3:2]		Bits [1:0][A]			Bits [7:6]		Bits [5:4]		Bits [3:2]		Bit 1	Bit 6	Bit 5	
	1[B]	10[B]	10[B]	11[B]	10[B]	11[B]	11[C]	10[C]	01[C]	10[B]	01[B]	10[B]	01[B]	10[B]	01[B]	1[B]	1[B]	1[B]	
DREQ0														DREQ 5	DREQ 0/5				
DREQ1												DREQ 6	DREQ 1/6						
DREQ2																			
DREQ3										DREQ 7	DREQ 3/7								
DREQ5																			
DREQ6	EPMI0#																		
DREQ7																			
DACK0#					DACK 5#	GPCS 0#	EDACK 0#	EDACK 0	DACK 0#										
DACK1#			DACK 6#	GPCS 2#			EDACK 1	EDACK 1	DACK 1#										
DACK2#							EDACK EN#	EDACK EN#	DACK 2#										
DACK3#		DACK 7#					EDACK 2	EDACK 2	DACK 1#										
									GPCS 0#										
									DACK 5#										
									DACK 6#										
									DACK 7#										
PIRQ2#							PIRQ 2#		PIRQ 2#										
PIRQ3#							PIRQ 3#		PIRQ 3#										EPMI#
PGNT2#																	GPCS 1#		

A. Group-wise programming can be overridden by pin-wise programming.

B. Pin-wise programming

C. Group-wise programming

Table 5-9 Pin Multiplexing Chart 2

Default Pin Names	SYSCFG 42h												SYSCFG 51h								SYSCFG 49h			
	Bit 7		Bit 6		Bit 5		Bit 4		Bit 3		Bit 2		Bit 1		Bit 2		Bit 1		Bit 0		Bits [6:5]	Bit 4	Bit 3	Bit 2
	1	0*	1	0*	1	0*	1	0*	1	0*	1	0*	1	0*	1	0*	1	0*	1	0*	10	1	1	1
IRQ1[A]																								
IRQ3															L	E								
IRQ4																	L	E						IRQ4/6
IRQ5													L	E										
IRQ6[A]																							PCIRQ0/1#	
IRQ7																			L	E				
IRQ8#[A]																								
IRQ9											L	E												
IRQ10									L	E												IRQ10/12		
IRQ11								L	E															
IRQ12							L	E																
IRQ14				L	E																PCIRQ2/3#			
IRQ15	L	E																						

*: Default

A: IRQ1, IRQ6 and IRQ8# are hard-wired as edge trigger (internal signal IRQ0, IRQ13 are also edge trigger).

E: Edge trigger

L: Level trigger

Table 5-10 Pin Multiplexing Chart 3

	SYSCFG 41h-40h								SYSCFG 50h			
	PCI IRQ Internal Routing[F]								PCI IRQ Internal Routing[G]			
Default Pin Names	111	110	101	100	011	010	001	000*	11	10	01	00
PREQ0# (PCI_REQ0#)												
PREQ1# (PCI_REQ1#)												
PREQ2# (PCI_REQ2#)												
PGNT0# (PCI_GRANT0#)												
PGNT1# (PCI_GRANT1#)												
PGNT2# (PCI_GRANT2#)												
PIRQ0# (PCI_IRQ0#)[E]	IRQ15	IRQ14	IRQ12	IRQ11	IRQ10	IRQ9	IRQ5	Disable	IRQ7[A]	IRQ4[A]	IRQ3[A]	Disable
PIRQ1# (PCI_IRQ1#)[E]	IRQ15	IRQ14	IRQ12	IRQ11	IRQ10	IRQ9	IRQ5	Disable	IRQ7[B]	IRQ4[B]	IRQ3[B]	Disable
PIRQ2# (PCI_IRQ2#)[E]	IRQ15	IRQ14	IRQ12	IRQ11	IRQ10	IRQ9	IRQ5	Disable	IRQ7[C]	IRQ4[C]	IRQ3[C]	Disable
PIRQ3# (PCI_IRQ3#)[E]	IRQ15	IRQ14	IRQ12	IRQ11	IRQ10	IRQ9	IRQ5	Disable	IRQ7[D]	IRQ4[D]	IRQ3[D]	Disable

*: Default

A: Only valid with SYSCFG 41h[3:1] - 000.

B: Only valid with SYSCFG 41h[0] and 40h[7:6] - 000.

C: Only valid with SYSCFG 40h[5:3] - 000.

D: Only valid with SYSCFG 40h[2:0] - 000.

E: Refer to Table 5-9 for additional muxing options on IRQ6 and IRQ12.

F: SYSCFG 40h[2:0] for PIRQ0#, 40h[5:3] for PIRQ1#, 40h[7:6]+41h[0] for PIRQ2#, 41h[3:1] for PIRQ3#.

G: SYSCFG 50h[1:0] for PIRQ0#, 50h[3:2] for PIRQ1#, 50h[5:4] for PIRQ2#, 50h[7:6] for PIRQ3#.

5.3.2 82C568 Power Management Registers

The Power Management Registers are located in the 82C568 and are accessed by an indexing scheme.

Port 022h is used as the Index Register and Port 024h is the Data Register. Each access to a control register consists of:

1) a write to Port 022h, specifying the desired register in the data byte,
2) followed by a read or write to Port 024h with the actual register data.

The index resets after every access; so every data access (via Port 024h) must be preceded by a write to Port 022h even if the same register is being accessed consecutively.

Table 5-11 gives the bit formats for the Power Management Registers accessed in SYSCFG. All bits are read/write and their default value is 0 unless otherwise specified.

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG E0h			**GREEN Mode Control/Enable Status**				**Default = 00h**
SMI# generation: Setting this bit allows power management to be run through the SMI# protocol. 0 = Disable (the only way power management functions can now be utilized is by setting bit 3 = 1). 1 = Enable	GREEN event SMI# generation/status:(1)	Wake-up event SMI# generation/status:(2)	GREEN status bit (RO): 0 = NORMAL 1 = GREEN	Hardware PPWRL# generation enable bit: Allows a GREEN/wake-up event to generate a PPWRL# pulse. 0 = Disable 1 = Enable	Hardware PPWRL# generation for GREEN events: Allows generation of a hardware PPWRL# for GREEN events. 0 = Disable, Causes system not to generate a hardware PPWRL# for GREEN events. 1 = Enable, Causes system to generate a hardware PPWRL# for GREEN events (if bit 3 = 1).	Hardware PPWRL# generation for wake-up events: Allows control of hardware PPWRL# generation for reload GET / wake-up events. 0 = Disable, Causes system not to generate a hardware PPWRL# for wake-up events. 1 = Enable, Causes system to generate a hardware PPWRL# for wake-up events (if bit 3 = 1).	Software GREEN event generation (can be used by APM): Setting this bit via software generates a GREEN event (if SYSCFG E1h[0] is enabled). 0 = No action is taken. 1 = Causes the generation of a GREEN event (if SYSCFG E1h[0] = 1).

1) Written to: Allows generation of SMI# on occurrence of any GREEN event (if bit 7 = 1).
0 = Disable, GREEN event occurrence will not cause an SMI# to be generated (this will also cause the status bit to be 0, and a read from the status bit will always yield a 0).
1 = Enable, GREEN event will cause an SMI# to be generated.
Read from:
Reflects if GREEN event caused SMI#.
0 = SMI# is not caused by GREEN event.
1 = GREEN event caused the system to generate a SMI#.
The BIOS should read this bit to identify whether a GREEN event caused the generation of the SMI#.

2) Written to: Allows generation of SMI# on occurrence of any reload GET/wake-up event (if bit 7 = 1).
0 = Disable, Wake-up event occurrence will not cause a SMI# to be generated (this will also cause the status bit to be 0, and a read from the status bit will always yield a 0).
1 = Enable, Wake-up event will cause a SMI# to be generated.
Read from: Reflects if wake-up event caused SMI#.
0 = SMI# not caused by wake-up event.
1 = Wake-up event caused the system to generate a SMI#.
The BIOS should read this bit to identify whether a wake-up event caused the generation of the SMI#

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG E1h			**EPMI Control / GREEN Event Timer**				**Default = 00h**
GREEN Event Timer (GET) CLK selection: This specifies the period of the CLK used by the GET. 00 = 119µs CLK period 01 = 12.25ms CLK period 10 = 1.94s CLK period 11 = 62.5s CLK period		EPMI0# polarity select: 0 = EPMI0# is an active low input 1 = EPMI0# is an active high input	EPMI0# debounce select: 0 = Disable debounce 1 = Enable debounce (EPMI0# width needs to stay stable for >= 5ms to be recognized	This bit is used in conjunction with bit 5 to determine the polarity of the EPMI0# signal. 0 = Polarity of EPMI0# is determined by bit 5. 1 = Regardless of bit 5 setting, EPMI0# will be triggered on the falling edge as well as the rising edge. However, care must be taken to ensure that the level on EPMI0# stays unchanged for a minimum period of 20ms for the ViperMAX Chipset to recognize the level change and act on it. (1)	Timer time-out into GREEN enable/status:(2) Written to: 0 = Disables the GET time-out from causing GREEN event 1 = Enables the GET time-out to cause GREEN event Read from: Reflects whether a GREEN event has been invoked by the timer time-out (if 0 had been written to it then reading this bit will always show a 0). 0 = GREEN event has not been caused by the GET time-out 1 = GREEN was caused due to the GET time-out	EPMI0# trigger into GREEN enable/status:(3) Written to: 0 = Disables an EPM0# trigger from causing GREEN event 1 = Enables an EPMI0# trigger to cause GREEN event Read from: Reflects whether a GREEN event has been invoked by an EPMI0# trigger (if 0 had been written to it then reading this bit will always show a 0). 0 = GREEN event is not invoked due to an EPMI0# trigger 1 = GREEN event is invoked due to an EPMI0# trigger	Sftwr trigger into GREEN enable/status:(4) Written to: 0 = Disables a software trigger from causing a GREEN event 1 = Enables a software trigger to cause a GREEN event Read from: Reflects whether a GREEN event has been invoked by a software trigger. (if 0 had been written to it then reading this bit will always show a 0). 0 = GREEN event is not invoked by a software trigger 1 = GREEN event is invoked by a software trigger

1) In addition, since this pin can also take on the functionality of DREQ6 (pin number 122 on the 82C568), the polarity on DREQ6, when this pin is used for the DREQ6 functionality, will be active low when this bit is a 0 and will be active high when this bit is a 1.
When the combination of bits 3 and 5 is 00, then a falling edge on EPMI0# will be recognized as an interrupt. When the combination on the aforementioned bits is 01, then a rising edge on the EPMI0# signal will be recognized as an interrupt. When the combination of the bits is 1X, then EPMI0# will be recognized on a transition to the opposite level. However, as noted earlier, the new level to which EPMI0# switches to will have to be maintained for a minimum period of 20ms for the internal circuit to recognize the transition and act on it.
2) The BIOS should read this bit to identify the cause of the GREEN event. If it is caused by the GET time-out, then the BIOS should clear the bit by writing a 0.
3) The BIOS should read this bit to identify the cause of the GREEN event. If it is caused by an EPMI trigger, then the BIOS should clear the bit by writing a 0.
4) The BIOS should read this bit to identify the cause of the GREEN event. If it is caused by a software trigger, then the BIOS should clear the bit by writing a 0. After that, to enable the functionality the BIOS should write a 1.

SYSCFG E2h	**GREEN Event Timer Initial Count Register**	**Default = 00h**
GREEN Event Timer count: Specifies the initial count (01h-FFh) for the GET. This count, along with SYSCFG E1h[7:6], specify the time-out period for the GET. **Note:** There is a latency of two clocks, so the actual time-out period will be: (the count selected +2) * (clock period defined in SYSCFG E1h[7:6])		

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG E3h			**IRQ Event Enable Register 1**				**Default = 00h**
IRQ7 monitor: Disable/enable IRQ7 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ6 monitor: Disable/enable IRQ6 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ5 monitor: Disable/enable IRQ5 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ4 monitor: Disable/enable IRQ4 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ3 monitor: Disable/enable IRQ3 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ[15:0] deglitch select: 0 = No deglitch 1 = Sample the IRQ[15:0] lines using an internal clock and reduce glitching	IRQ1 monitor: Disable/enable IRQ1 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ0 monitor: Disable/enable IRQ0 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable
SYSCFG E4h			**IRQ Event Enable Register 2**				**Default = 00h**
IRQ15 monitor: Disable/enable IRQ15 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ14 monitor: Disable/enable IRQ14 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ13 monitor: Disable/enable IRQ13 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ12 monitor: Disable/enable IRQ12 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ11 monitor: Disable/enable IRQ11 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ10 monitor: Disable/enable IRQ10 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ9 monitor: Disable/enable IRQ9 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	IRQ8 monitor: Disable/enable IRQ8 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable
SYSCFG E5h			**DREQ Event Enable Register**				**Default = 00h**
DREQ7 monitor: Disable/enable DREQ7 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable(1)	DREQ6 monitor: Disable/enable DREQ6 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable(1)	DREQ5 monitor: Disable/enable DREQ5 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable(1)	Reserved: Must = 0	DREQ3 monitor: Disable/enable DREQ3 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable(1)	DREQ2 monitor: Disable/enable DREQ2 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable(1)	DREQ1 monitor: Disable/enable DREQ1 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable(1)	DREQ0 monitor: Disable/enable DREQ0 from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable(1)
1) Enable if SYSCFG EFh[6] = 1 (this then becomes a wake-up event).							
SYSCFG E6h			**Device Cycle Monitor Enable Register**				**Default = 00h**
Programmable IO/MEM range:(1) Disable/enable the device cycle to the address range specified from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	Parallel ports access detection: Disable/enable accesses to the parallel ports (3B0h-#BFh, 378h-37Fh, 278h-27Fh) from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	Video (A0000-BFFFFh, 3B0h-3DFh) access detection: Disable/enable video accesses from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	Hard disk access detection: Disable/enable accesses to the hard disk ports (170h-177h, 376h, 1F0h-1F7h, 3F6h) from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	Floppy disk access detection: Disable/enable accesses to the accesses to floppy disk (3F5h) from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	Keyboard access detection: Disable/enable accesses to the keyboard (60h, 64h) from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	COM Ports 1/3 access detection: Disable/enable accesses to the COM Ports 1/3 (3F8h-3FFh, 3E8h-3EFh) from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable	COM Ports 2/4 access detection: Disable/enable accesses to the COM ports 2/4 (2F8h-2FFh, 2E8h-2EFh) from/to becoming a reload GET / wake-up event. 0 = Disable 1 = Enable
1) Setting this bit enables the IO/MEM address ranges specified in SYSCFG E7h through E9h to be monitored.							

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG E7h			**Wake-up Source/Programmable I/O/Memory Address Mask Register**				**Default = 00h**
PREQ# detection: Allows PCI bus requests to be monitored. 0 = Disables a PCI bus master cycle from generating a reload GET / wake-up event. 1 = Enables a PCI bus master cycle to generate a reload GET / wake-up event if SYSCFG EFh[7] is set.	LDEV#/ DEVSEL# detection: Allows all accesses to the local bus be monitored. 0 = Disables generation of a reload GET / wake-up event on any local bus slave access. 1 = Enables generation of a reload GET / wake-up event on any local bus slave access.	Reload GET on an EPMI0# trigger: 0 = Disables generation of a reload GET / wake-up event on an EPMI0# trigger. 1 = Enables generation of a reload GET / wake-up event on an EPMI0# trigger.	Reserved: Must = 0	IO/MEM (non-system memory) selection for the programmable address ranges specified in SYSCFG E8h and E9h: Determines whether the address specified in SYSCFG E8h and E9h is an I/O or non-system memory address. 0 = I/O 1 = Memory	Mask bits for the programmable IO/MEM (non-system memory) address range (SYSCFG E8h): 000 = Mask no bits 001 = Mask lowest bit 010 = Mask lowest 2 bits 011 = Mask lowest 3 bits 100 = Mask lowest 4 bits 101 = Mask lowest 5 bits 110 = Mask lowest 6 bits 111 = Mask lowest 7 bits		
SYSCFG E8h			**Programmable IO/MEM Address Range Register**				**Default = 00h**
Programmable IO/MEM (non-system memory) address range: Depending on the selection in SYSCFG E7h[3], the address range specified in this register and in SYSCFG E9h corresponds to an I/O or a non-system memory address. Bits [7:0] map onto address lines A[7:0] for an I/O address and map onto address lines A[23:16] for a non-system memory address. SYSCFG E7h[2:0] specify the masking range.							
SYSCFG E9h			**Programmable IO/MEM Address Range Register**				**Default = 00h**
Programmable IO/MEM (non-system memory) address range: Bits [7:0] map onto address lines A[15:8] for an I/O address and map onto address lines A[31:24] for a non-system memory address.							
SYSCFG EAh			**Enter GREEN State Port Register**				**Default = 01h**
Hardware power management information: This port provides the GREEN state values for the external power control latch (SYSCFG ECh). When the hardware PPWRL# is strobed to enter the GREEN state, this register will transfer its contents to SYSCFG ECh. GREEN state Power Port (GPP) bits [7:0] contain the information that is transferred to SYSCFG ECh.							
SYSCFG EBh			**Return to NORMAL State Configuration Port Register**				**Default = 01h**
NORMAL state configuration information: This port provides the return to NORMAL state values for the external power control latch (SYSCFG ECh). When the hardware PPWRL# is strobed to return to the NORMAL mode, this register will transfer its contents to SYSCFG ECh. NORMAL state Power Port (NPP) bits [7:0] contain the NORMAL state configuration information that is transferred to SYSCFG ECh.							
SYSCFG ECh			**Shadow Register for External Power Control Latch Register**				**Default = 01h**
This port shadows the value of the external power control latch (SYSCFG ECh). Power Port (PP) bits [7:0] contain the value that the external power control latch has. A write to SYSCFG ECh will generate a PPWRL# pulse.							

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG EDh			**Device Cycle Detection Enable / Status Register**				**Default = 00h**
Programmable IO/MEM range monitor for SMI# generation/status: (1)	LPT access monitor for SMI# generation/status:(2)	Video access monitor for SMI# generation/status:(3)	Hard disk access monitor for SMI# generation/status (4):	Floppy disk access monitor for SMI# generation/status:(5)	Keyboard access monitor for SMI# generation/status:(6)	COM Ports 1/3 access monitor for SMI# generation/status:(7)	COM Ports 2/4 access monitor for SMI# generation:(8)

(1) Written to: Allows an access to the programmed IO/MEM address range (specified by SYSCFG E8h and E9h), to become a wake-up event. If SYSCFG E0h[5] and E0h[7] = 1, an SMI# will be generated.
0 = Disable an access to programmed range from generating a SMI#.
1 = Enable an access to programmed range to generate a SMI#.
Read from: If a 0 has been written into this bit, then a read from it will always show a 0.
0 = SMI# has not been invoked by an access to programmed range.
1 = SMI# has been invoked by an access to programmed range.
The BIOS should read this bit to identify the cause of the SMI#. If it is caused by a device cycle to the above programmed address range, then the BIOS should clear the bit by writing a 0.

(2) Written to: If set, allows an LPT access to become a wake-up event. If SYSCFG E0h[5] and E0h[7] are set to 1, then it will cause a SMI# to be generated.
0 = Disable an LPT access from generating a SMI#.
1 = Enable an LPT access to generate a SMI#.
Read from: If a 0 has been written into this bit, then a read from it will always show a 0.
0 = SMI# has not been invoked by an LPT access cycle.
1 = SMI# has been invoked by an LPT access cycle.
The BIOS should read this bit to identify the cause of the SMI#. If it is caused due to an LPT access, then the BIOS should clear the bit by writing a 0.

(3) Written to: Allows a video access to become a wake-up event. If SYSCFG E0h[5] and E0h[7] are set to 1, an SMI# will be generated.
0 = Disable a video access from generating a SMI#.
1 = Enable a video access to generate a SMI#.
Read from: If a 0 has been written into this bit, then a read from it will always show a 0.
0 = SMI# has not been invoked by a video access cycle.
1 = SMI# has been invoked by a video access
The BIOS should read this bit to identify the cause of the SMI#. If it is caused by a device cycle to the above programmed address range, then the BIOS should clear the bit by writing a 0.

(4) Written to: If set allows a hard disk access to become a wake-up event. If SYSCFG E0h[5] and E0h[7] are set to 1, then it will cause a SMI# to be generated.
0 = Disables a hard disk access from generating a SMI#.
1 = Enables a hard disk access to generate a SMI#.
Read from: If a 0 has been written into this bit, then a read from it will always show a 0.
0 = SMI# has not been invoked by a hard disk access cycle.
1 = SMI# has been invoked by a hard disk access cycle.
The BIOS should read this bit to identify the cause of the SMI#. If it is caused due to a hard disk access, then the BIOS should clear the bit by writing a 0.

(5) Written to: If set, allows a floppy disk access to become a wake-up event. If SYSCFG E0h[5] and E0h[7] are set to 1, then it will cause a SMI# to be generated.
0 = Disables a floppy disk access from generating a SMI#.
1 = Enables a floppy disk access to generate a SMI#.
Read from: If a 0 has been written into this bit, then a read from it will always show a 0.
0 = SMI# has not been invoked by a floppy disk access cycle.
1 = SMI# has been invoked by a floppy disk access cycle.
The BIOS should read this bit to identify the cause of the SMI#. If it is caused due to a floppy disk access, then the BIOS should clear the bit by writing a 0.

(6) Written to: If set, allows a keyboard access to become a wake-up event. If SYSCFG E0h[5] and E0h[7] are set to 1, then it will cause a SMI# to be generated.
0 = Disables a keyboard access from generating a SMI#.
1 = Enables a keyboard access to generate a SMI#.
Read from: If a 0 has been written into this bit, then a read from it will always show a 0.
0 = SMI# has not been invoked by a keyboard access cycle.
1 = SMI# has been invoked by a keyboard access cycle.
The BIOS should read this bit to identify the cause of the SMI#. If it is caused due to a keyboard access, then the BIOS should clear the bit by writing a 0.

(7) Written to: If set, allows an access to COM Ports 1/3 to become a wake-up event. If SYSCFG E0h[5] and E0h[7] are set to 1, then it will cause a SMI# to be generated.
0 = Disables a COM1/COM3 access from generating a SMI#.
1 = Enables a COM1/COM3 access to generate a SMI#.
Read from: If a 0 has been written into this bit, then a read from it will always show a 0.
0 = SMI# has not been invoked by a COM1/COM3 access cycle.
1 = SMI# has been invoked by a COM1/COM3 access cycle.
The BIOS should read this bit to identify the cause of the SMI#. If it is caused due to an access to COM Ports 1/3, then the BIOS should clear the bit by writing a 0.

(8) Written to: If set, will allow an access to COM Ports 2/4 to become a wake-up event. If SYSCFG E0h[5] and E0h[7] are set to 1, then it will cause a SMI# to be generated.
0 = Disables a COM2/COM4 access to generate a SMI#.
1 = Enables a COM2/COM4access to generate a SMI#.
Read from: If a 0 has been written into this bit, then a read from it will always show a 0.
0 = SMI# has not been invoked by a COM2/COM4 access cycle.
1 = SMI# has been invoked by a COM2/COM4 access cycle.
The BIOS should read this bit to identify the cause of the SMI#. If it is caused due to an access to COM Ports 2/4, then the BIOS should clear the bit by writing a 0.

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG EEh			**STPCLK# Modulation Register**				**Default = 00h**
CPU STOP-CLK state support: 0 = Disable 1 = Enable	CPU on hold when in STOP-CLK state: 0 = No 1 = Yes This setting is mainly used for further lowering power consumption.	Reserved: Must = 0		STPCLK# modulation: 0 = Disable 1 = Enable	STPCLK# modulation duty cycle: 000 = STPCLK# = 1 always (i.e., no modulation) 001 = STPCLK# = 1 for 1/2 period 010 = STPCLK# = 1 for 1/4 period 011 = STPCLK# = 1 for 1/8 period 100 = STPCLK# = 1 for 1/16 period 101 = STPCLK# = Reserved 110 = STPCLK# = Reserved 111 = STPCLK# = Reserved Comes into effect only if bit 3 = 1.		
SYSCFG EFh			**Miscellaneous Register**				**Default = 00h**
PREQ# wake-up system: This bit is used in conjunction with SYSCFG E7h[7] to enable a local bus request to wake-up the system. 0 = Disable 1 = Enable	DREQ# wake-up system: This bit is used in conjunction SYSCFG E5h to enable any DREQ# to wake-up the system. 0 = Disable 1 = Enable	Reserved: Must = 0	GPCS1#, GPCS2# generation: Setting this bit enables generation of GPCS1# and GPCS2# for the address ranges specified in SYSCFG F4h through F7h. 0 = Disable 1 = Enable	Reserved: Must = 0	Clock selection bit to initiate the hardware PPWRL# signal: 0 = 14MHz clock 1 = 32kHz clock If CLK SYNTH is in the power-down mode, then a clock that is still running will have to be selected in order for the wake-up event to trigger a hardware PPWRL# signal.	Read current count of timers:(1) 0 = Return the current count value on reads to registers. 1 = Return the originally programmed value into the register on reads.	Reserved: Must = 0

1) This bit must be set to a 0 to read the current count of the various timers used in power management of the Viper-MAX Chipset. The user needs to program various initial counts in the registers at SYSCFG E0h, E1h, E2h, EDh, F0h, F1h, F2h, FCh, FDh, and FEh. The current count value of the timers associated with these registers can be read back from these registers if this bit set to 0. If this bit is set to 1, reads from the aforementioned registers will return the originally programmed initial count of the respective registers.

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG F0h		**Device Timer CLK Select / Enable Status Register**					**Default = 00h**
Clock period for Device Timer 1: 00 = 119µs 10 = 1.94s 01 = 12.25ms 11 = 62.5s		Clock period for Device Timer 0: 00 = 119µs 10 = 1.94s 01 = 12.25ms 11 = 62.5s		Device Timer 1 time-out GREEN event enable/ status:(1)	Device Timer 0 time-out GREEN event enable/ status:(2)	Device 1 cycle generate wake-up event enable/ status:(3)	Device 0 cycle generate wake-up event enable/ status:(4)

(1) Written to: If set, Device Timer 1 time-out becomes a GREEN event.
0 = Disable Device Timer 1 time-out from becoming a GREEN event.
1 = Enable Device Timer 1 time-out to become a GREEN event.
Read from: If this has not been enabled (i.e., not set to 1), then a read from it will always fetch 0.
1 = GREEN event has been invoked by Device Timer 1 time-out.
0 = GREEN event has not been invoked by Device Timer 1 time-out.
The BIOS should read this bit to identify the cause of the GREEN event. If it is caused due to Device Timer 1 time-out, then the BIOS should clear the bit by writing a 0. After that, to enable the functionality again, the BIOS should write a 1.

(2) Written to: If set, Device Timer 0 time-out becomes a GREEN event.
0 = Disable Device Timer 0 time-out from becoming a GREEN event.
1 = Enable Device Timer 0 time-out to become a GREEN event.
Read from: If this has not been enabled (i.e., not set to 1) then a read from it will always fetch 0.
1 = GREEN event has been invoked by Device Timer 0 time-out.
0 = GREEN event has not been invoked by Device Timer 0 time-out.
The BIOS should read this bit to identify the cause of the GREEN event. If it is caused due to Device Timer 0 time-out, then the BIOS should clear the bit by writing a 0. After that, to enable the functionality again, the BIOS should write a 1.

(3) Written to: If enabled, will allow any access to the Device 1 address range (SYSCFG F6h, F7h) to become a wake-up event.
0 = Disables a Device 1 cycle from becoming a wake-up event.
1 = Enables a Device 1 cycle to become a wake-up event.
Read from: If a 0 had been written to it, then a read from it will always fetch 0.
0 = Wake-up event has not been invoked by an access cycle to device 1.
1 = Wake-up event has been invoked by an access cycle to device 1.
The BIOS should read this bit to identify the cause of the wake-up event. If it is caused by an access to the Device 1 address range, then the BIOS should clear the bit by writing a 0. After that, to enable the functionality again, the BIOS should write a 1.
If Device Timers 1 and 2 are not utilized, then the F1h-F7h register space can be used as scratch pad space.

(4) Written to: This bit, if enabled, will allow any access to the Device 0 address range (SYSCFG F5h, F4h) to become a wake-up event.
0 = Disables a Device 0 cycle from becoming a wake-up event.
1 = Enables a Device 0 cycle to become a wake-up event.
Read from: If a 0 had been written to it, then a read from it will always fetch 0.
0 = Wake-up event has not been invoked by an access cycle to device 0.
1 = Wake-up event has been invoked by an access cycle to device 0.
The BIOS should read this bit to identify the cause of the wake-up event. If it is caused by an access to the Device 0 address range, then the BIOS should clear the bit by writing a 0. After that, to enable the functionality again, the BIOS should write a 1.

SYSCFG F1h	**Device Timer 0 Initial Count Register**	**Default = 00h**
	Device Timer 0 initial count: Bits [7:0] specify the initial count (00h-FFh) for Device Timer 0. These bits, in conjunction with SYSCFG F0h[5:4], define the time-out period for Device Timer 0.	

SYSCFG F2h	**Device Timer 1 Initial Count Register**	**Default = 00h**
	Device Timer 1 initial count: Bits [7:0] specify the initial count (00h-FFh) for Device Timer 1. These bits, in conjunction with SYSCFG F0h[7:6], define the time-out period for Device Timer 1.	

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG F3h			**Device Timer IO/MEM Select, Mask Bits Register**				**Default = 00h**
IO/MEM selection for Device 1: This bit determines whether the address range specified for Device 1 in SYSCFG F6h and F7h, is an I/O address or a memory address. 0 = I/O address 1 = Memory address	Mask bits for Device 1 IO/MEM address range: 000 = Mask no bits 001 = Mask the lowest bit 010 = Mask the lowest 2 bits 011 = Mask the lowest 3 bits 101 = Mask the lowest 5 bits 101 = Mask the lowest 5 bits 110 = Mask the lowest 6 bits 111 = Mask the lowest 7 bits			IO/MEM selection for Device 0: This bit determines whether the address range specified for Device 0 in SYSCFG F5h and F4h, is an I/O address or a memory address 0 = I/O address 1 = Memory address	Mask bits for Device 0 IO/MEM address range: 000 = Mask no bits 001 = Mask the lowest bit 010 = Mask the lowest 2 bits 011 = Mask the lowest 3 bits 101 = Mask the lowest 5 bits 101 = Mask the lowest 5 bits 110 = Mask the lowest 6 bits 111 = Mask the lowest 7 bits		
SYSCFG F4h			**Device 0 IO/MEM Address Register**				**Default = 00h**
Device 0 IO/MEM address: This register is used in conjunction with SYSCFG F5h[7:0] to specify the I/O or the memory address of Device 0. If SYSCFG F3h[3] = 0, these bits map onto A[7:0] as an I/O address. If SYSCFG F3h[3] = 1, these bits map onto A[23:16] as a memory address. SYSCFG F3h[2:0] specify which of these bits are to be masked.							
SYSCFG F5h			**Device 0 IO/MEM Address Register**				**Default = 00h**
Device 0 IO/MEM address: This register is used in conjunction with SYSCFG F4h[7:0] to specify the I/O or the memory address of Device 0. If SYSCFG F3h[3] = 0, these bits map onto A[15:8] as an I/O address. If SYSCFG F3h[3] = 1, these bits map onto A[31:24] as a memory address. If SYSCFG EFh[4] = 1, an access to the address range specified here causes GPCS1# to be asserted.							
SYSCFG F6h			**Device 1 IO/MEM Address Register**				**Default = 00h**
Device 1 IO/MEM address: This register is used in conjunction with SYSCFG F7h[7:0] to specify the I/O or the memory address of Device 1. If SYSCFG F3h[7] = 0, these bits map onto A[7:0] as an I/O address. If SYSCFG F3h[7] = 1, these bits map onto A[23:16] as a memory address. SYSCFG F3h[6:4] specify which of these bits are to be masked.							
SYSCFG F7h			**Device 1 IO/MEM Address Register**				**Default = 00h**
Device 1 IO/MEM address: This register is used in conjunction with SYSCFG F6h[7:0] to specify the I/O or the memory address of Device 1. If SYSCFG F3h[7] = 0, these bits map onto A[15:8] as an I/O address. If SYSCFG F3h[7] = 1, these bits map onto A[31:24] as a memory address. If SYSCFG EFh[4] = 1, an access to the address range specified here causes GPCS2# to be asserted.							
SYSCFG FAh-FBh			**Reserved Register 1**				**Default = 00h**
Reserved: Must be written to 0.							

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG FCh			**Power Management Control Register 1**				**Default = 00h**
EPMI1# becomes GREEN event:(1) 0 = Disable EPMI1# from triggering a GREEN event 1 = Enable triggering of a GREEN event when EPMI1# is active	Reload wake-up GREEN state timer whenever EPMI1# is active:(1) During writes: 0 = Disable EPMI1# from loading wake-up GREEN state timer 1 = Enable loading of wake-up GREEN state timer on EPMI1# During reads: 0 = GREEN event not invoked by EPMI1# 1 = GREEN event invoked by EPMI1#	EPMI1# polarity:(2) 0 = EPMI1# polarity determined by bit 4 1 = An event will be triggered dictated by bits 7 and 6 on a transition of EPMI1# (provided timing requirements are met on EPMI1#)	EPMI1# polarity: Determines the polarity of EPMI1# as long as bit 5 is not set. Listed is the combination of bits 5 and 4 and the polarity on EPMI1# that they define: 00 = Falling edge on EPMI1# causes a triggering of an event 01 = Rising edge on EPMI1# causes a trigger of the event 1X = A transition on EPMI1# causes a triggering of an event (as long as timing requirements are met)	EPMI1# debounce effect:(3) 0 = No debounce effect has been enabled on EPMI1# 1 = Debounce effect has been selected on EPMI1# (again, EPMI1# should remain stable for a minimum period of 5ms for proper recognition)	Reserved: Must be written to 0.		Pin functionality: 0 = Pins 131, 139, and 143 all function as EPMI0#. 1 = Pin 131 functions as EPMI0#, pin 139 functions as EPMI1#, and pin 143 functions as EPMI2#.

(1) Bits 7 and 6 cannot be set to a 1 at the same time, as EPMI1# is interpreted differently for each of the settings. During reads of the register (reads of the register is primarily done to determine status) if bit 7 returns a 1, the user should interpret that as a GREEN event is invoked by the EPMI1# signal and if a 0 is returned, the user should interpret that as a GREEN event not being invoked by the EPMI1# signal.

(2) Bit 5 determines the polarity of EPMI1# in conjunction with bit 4. When bit 5 = 0, the polarity of EPMI1# is determined by bit 4. However, bit 5 = 1, then regardless of the setting of bit 4, a transition on EPMI1# will trigger an event based on the settings of bits 7 and 6 (explained above). It should be noted that when bit 5 = 1, it is the user's responsibility to guarantee that the transition on EPMI1# lasts for a minimum period of 20ms before it departs from the current logic level.

(3) Bit 3 is used to select the debounce effect on EPMI1#. Normally, external interrupts to the power management unit can cause glitches in the system. The user can avoid this glitching by setting this bit to a 1. Note that to avoid the glitching, the EPMI1# signal needs to stay stable for a minimum period of 5ms.

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG FDh			**Power Management Control Register 2**				**Default = 00h**
EPMI2# becomes GREEN event:(1) During writes: 0 = Disable EPMI2# from loading wake-up GREEN state timer 1 = Enable loading of wake-up GREEN state timer on EPMI2# During reads: 0 = GREEN event not invoked by EPMI2# 1 = GREEN event invoked by EPMI2#	Reload wake-up GREEN state timer whenever EPMI2# is active: (1) 0 = Disable EPMI2# from loading wake-up GREEN state timer 1 = Enable loading of wake-up GREEN state timer on EPMI2#	EPMI2# polarity:(2) 0 = EPMI2# polarity determined by bit 4 1 = An event will be triggered dictated by bits 7 and 6 on a transition of EPMI2# (provided timing requirements are met on EPMI2#)	EPMI2# polarity: Determines the polarity of EPMI2# as long as bit 5 is not set. The combination of bits 5 and 4 and the polarity on EPMI2# that they define: 00 = Falling edge on EPMI2# causes a triggering of an event 01 = Rising edge on EPMI2# causes a triggering of an event 1X = Transition on EPMI2# causes a trigger of an event (as long as timing requirements are met)	EPMI2# debounce effect:(3) 0 = No debounce effect has been enabled on EPMI2# 1 = Debounce effect has been selected on EPMI2# (EPMI2# should remain stable for a minimum period of 5ms for proper recognition.)	Reserved: Must be written to 0.	Pin 108 functionality: 0 = GPCS0# 1 = PPWRL1 To activate these functions on pin 108, PCIDV1 44h[3:2] must = 11 and 44h[1:0] must = 01.	Reserved: Must be written to 0.

(1) Bits 7 and 6 cannot be set to a 1 at the same time, as EPMI2# is interpreted differently for each of the settings. During reads of the register (done to determine the status of the event) if bit 7 returns a 1, the user should interpret that as a GREEN event is invoked by the EPMI2# signal and if a 0 is returned, the user should interpret that as a GREEN event not being invoked by the EPMI2# signal.

(2) Bit 5 determines the polarity of EPMI2# in conjunction with bit 4. If bit 5 = 0, the polarity of EPMI2# is determined by bit 4. However, if bit 5 = 1, then regardless of the setting of bit 4, a transition on EPMI2# will trigger an event based on the settings of bits 7 and 6 (explained above). It should be noted that when this bit is set to a 1, it is the user's responsibility to guarantee that the transition on EPMI2# signal lasts for a minimum period of 20ms before it departs from the current logic level.

(3) Bit 3 is used to select the debounce effect on EPMI2#. Normally, external interrupts to the power management unit can cause glitches in the system. The user can avoid this glitching by setting this bit to a 1. Note that to avoid glitching, EPMI2# signal needs to stay stable for a minimum period of 5ms.

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG FEh			**Power Management Control Register 3**				**Default = 00h**
EPMI3# becomes GREEN event:(1) During writes: 0 = Disable EPMI3# from loading wake-up GREEN state timer 1 = Enable loading of wake-up GREEN state timer on EPMI3# During reads: 0 = GREEN event not invoked by EPMI2# 1 = GREEN event invoked by EPMI2#	Reload wake-up GREEN state timer whenever EPMI3# is active: (1) 0 = Disable EPMI3# from loading wake-up GREEN state timer 1 = Enable loading of wake-up GREEN state timer on EPMI3#	EPMI3# polarity:(2) 0 = EPMI3# polarity determined by bit 4 1 = An event will be triggered, dictated by bits 7 and 6 on a transition of EPMI3# (provided EPMI3# timing requirements are met)	EPMI3# polarity: Determines the polarity of EPMI3# as long as bit 5 is not set. The combination of bits 5 and 4 and the polarity on EPMI2# that they define: 00 = Falling edge on EPMI3# causes a triggering of an event 01 = Rising edge on EPMI3# causes a triggering of an event 1X = Transition on EPMI3# causes a trigger of an event (as long as timing requirements are met)	EPMI3# debounce effect: (3) 0 = No debounce effect has been enabled EPMI3# 1 = Debounce effect has been selected on EPMI3# (EPMI3# should remain stable for a minimum period of 5ms for proper recognition)	Reserved: Must be written to 0.		

(1) Bits 7 and bit 6 cannot be set to a 1 at the same time, as EPMI3# is interpreted differently for each of the settings. During reads of the register (done to determine the status of the event) if bit 7 returns a 1, the user should interpret that as a GREEN event is invoked by the EPMI3# signal and if a 0 is returned, the user should interpret that as a GREEN event not being invoked by the EPMI3# signal.

(2) Bit 5 determines the polarity of EPMI3# in conjunction with bit 4. When bit 5 = 0, the polarity of EPMI3# is determined by bit 4. However, if bit 5 = 1, then regardless of the setting of bit 4, a transition on EPMI3# will trigger an event based on the settings of bits 7 and 6 (explained above). It should be noted that when this bit is set to a 1, it is the user's responsibility to guarantee that the transition on EPMI3# signal lasts for a minimum period of 20ms before it departs from the current logic level.

(3) Bit 3 is used to select the debounce effect on EPMI3#. Normally, external interrupts to the power management unit can cause glitches in the system. The user can avoid this glitching by setting this bit to a 1. Note that to avoid the glitching, the EPMI3# signal needs to stay stable for a minimum period of 5ms.

Table 5-11 82C568 Power Management Registers: SYSCFG E0h-FFh

7	6	5	4	3	2	1	0
SYSCFG FFh	**General Purpose Chip Select Control Register**						**Default = 00h**
CPU type: Provides CPU vendor information to the chipset so it can provide the stop clock functionality. 0 = CPU vendor is Intel/ AMD. 1 = CPU vendor is Cyrix and the CPU is M1.	Reserved: Must be written to 0.		Reserved: Must be written to 1.	Reserved: Must be written to 0.	Address bit masking capability: Used in conjunction with SYSCFG F3h[6:4] to increase the granularity in the address space of external devices. This table controls GPCS2#.(1)	Address bit masking capability: Used in conjunction with SYSCFG F3h[2:0] to increase the granularity in the address space of external devices This table controls GPCS1#.(1)	Address bit masking capability: Used in conjunction with SYSCFG E7h[2:0] to increase the granularity in the address space of external devices This table controls GPCS0#.(1)

(1)

Bit 2 and SYSCFG F3h[6:4], Bit 1 and SYSCFG F3h[2:0], (or) Bit 0 and SYSCFG E7h[2:0]				Bit 2 and SYSCFG F3h[6:4], Bit 1 and SYSCFG F3h[2:0], (or) Bit 0 and SYSCFG E7h[2:0]			
0	000	=	Mask no bit	1	000	=	Mask lowest 8 bits
0	001	=	Mask first lowest bit	1	001	=	Mask lowest 9 bits
0	010	=	Mask lowest 2 bits	1	010	=	Mask lowest 10 bits
0	011	=	Mask lowest 3 bits	1	011	=	Mask lowest 11 bits
0	100	=	Mask lowest 4 bits	1	100	=	Mask lowest 12 bits
0	101	=	Mask lowest 5 bits	1	110	=	Mask lowest 13 bits
0	110	=	Mask lowest 6 bits	1	110	=	Mask lowest 14 bits
0	111	=	Mask lowest 7 bits	1	111	=	Mask lowest 15 bits

5.3.3 82C568 I/O Register Space

The I/O Register Space of the 82C568 is accessed normally (i.e., CPU Direct I/O R/W). There is no indexing scheme.

Table 5-12 I/O Port Registers

7	6	5	4	3	2	1	0
Port 061h				**Port B Register**			
System parity check (RO)	I/O channel check (RO)	Timer OUT2 detect (RO)	Refresh detect (RO)	I/O channel check: 0 = Enable 1 = Disable	Parity check: 0 = Enable 1 = Disable	Speaker output: 0 = Disable 1 = Enable	Timer 2 Gate: 0 = Disable (from CPU address) 1 = Enable
Port 060h & 064h				**Keyboard I/O Control Registers**			
The 82C568 will intercept commands to Ports 060h and 064h so that it may emulate the keyboard controller, allowing the generation of the fast GATEA20 and fast CPURST signals. The decode sequence is software transparent and requires no BIOS modifications to function. The fast GATEA20 generation sequence involves writing 'D1h' to Port 64h, then writing data '02h' to Port 060h. The fast CPU warm reset function is generated when a Port 064h write cycle with data 'FEh ' is decoded. A write to Port 064h with data D0h will enable the status of GATEA20 (bit 1 of Port 060h) and the system reset (bit 0 of Port 060h) to be readable							
Port 092h				**PS/2 Reset Control Registers**			**Default = 00h**
Reserved: Must be written to 0.						A20M#: 0 = A20M# active 1 = A20M# inactive	Fast Reset (automatically clears back to 0): 1 = INIT sent to the 3.3V CPU

5.3.4 Internal Integrated 82C206 Registers

The internal integrated 82C206 configuration register is accessed by indexing I/O Registers 022h and 023h. Index Register 01h should be set to the default value of C0h. All individual DMA, Interrupt, and Timer subsystem registers are mapped in CPU I/O Address Space.

Following Table 5-13 are sections tables that explain the subsystem registers (DMA, Interrupt Controller and Counter/Timer) of the 82C206.

Table 5-13 Configuration Register (Index Port 022h, Data Port 023h) - SYSCFG2 01h

7	6	5	4	3	2	1	0
These bits control the number of wait states inserted when the CPU accesses the registers of the 82C206. Wait states are counted as SYSCLK cycles and are not affected by the DMA clock selection. 00 = One R/W wait state 01 = Two R/W wait states 10 = Three R/W wait states 11 = Four R/W wait states Default = 11		These bits control the number of wait states inserted in 16-bit DMA cycles. Further control of the DMA cycle length is available through the use of the IPC's IOCHRDY pin. During DMA cycles, this pin is used as an input to the wait state generation logic to extend the cycle if necessary. 00 = One 16-bit DMA wait state 01 = Two 16-bit DMA wait states 10 = Three 16-bit DMA wait states 11 = Four 16-bit DMA wait states Default = 00		These bits control the number of wait states inserted in 8-bit DMA cycles. Further control of the DMA cycle length is available through the use of the IPC's IOCHRDY pin. During DMA cycles, this pin is used as an input to the wait state generation logic to extend the cycle if necessary. 00 = One 8-bit DMA wait state 01 = Two 8-bit DMA wait states 10 = Three 8-bit DMA wait states 11 = Four 8-bit DMA wait states Default = 00		This bit enables the early internal DMAMEMR# function. In a PC/AT-based system, DMAMEMR# is delayed one clock cycle later than SMEMR#. 0 = Start DMAMEMR# 1 = Start DMAMEMR# at the time as SMEMR# Default = 0	0 = SYSCLK input is ÷2 and is used to drive both 8- and 16-bit DMA subsystems. 1 = SYSCLK will directly drive the DMA subsystems. Whenever the state of this bit is changed, an internal synchronizer controls the actual switching of the clock to prevent a short clock pulse from causing a DMA malfunction. Default = 0

5.3.4.1 DMA Subsystem Register Descriptions

Table 5-14 DMA Subsystem Registers

7	6	5	4	3	2	1	0
I/O Address 008h/0D0h			**Command Register (WO)**				
Determines if the DACK output pin will be active low or active high. 0 = Active low 1 = Active high	Determines if the DREQ input pin will be active low or active high. 0 = Active low 1 = Active high	Extended write:(1) 0 = Disabled 1 = Enabled	Rotating priority: 0 = Fixed (Default) 1 = Rotating	Compressed timing: This bit enables the compressed timing feature. 0 = Compressed timing 1 = Normal timing (Default)	Controller disable:(2) 0 = Enable 1 = Disable	Address hold: Setting this bit to 1 enables the address hold feature in Channel 0 when performing memory-to-memory transfers. 0 = Disable 1 = Enable	Memory-to-Memory: This bit enables Channels 0 and 1 to be used for memory-to-memory transfers. 0 = Disable 1 = Enable

(1) When the extended write feature is enabled, it causes the write command to be asserted one DMA cycle earlier during a transfer. Thus, read and write commands both begin in the S2 state.

(2) Setting this bit to 1 disables the DMA subsystem (DMA8 or DMA16). This function is normally used whenever the CPU needs to reprogram one of the channels to prevent DMA cycles from occurring.

I/O Address 00Bh/0D6h	**Mode Register**			
Mode select: 00 = Demand mode 01 = Single cycle mode 10 = Block mode 11 = Cascade mode	Decrement: Writing a 1 to this bit decrements the address after each transfer.	Auto-initialization: Writing a 1 to this bit enables the auto-initialization function.	Transfer type: 00 = Verify 01 = Write transfer 10 = Read transfer 11 = Illegal	Channel selection: These bits determine which channel's Mode Register will be written. A read-back of a Mode Register will cause these bits to both be 1. 00 = Ch. 0 10 = Ch. 2 01 = Ch. 1 11 = Ch. 3

I/O Address 009h/0D2h	**Request Register Write Format**	
Don't care	Request bit: Writing a 1 to this bit sets the request bit.	Request select: These bits determine which channel's request bit will be set. 00 = Ch. 0 10 = Ch. 2 01 = Ch. 1 11 = Ch. 3

I/O Address 009h/0D2h	**Request Register Read Format**
Reserved: Always reads 1.	Request channel: These bits contain the state of the request bit associated with each request channel. The bit position corresponds to the channel number.

Request Mask Register Set/Reset Format		
Don't Care	Mask bit: Writing a 1 to this bit sets the request mask bit and inhibits external requests.	Mask select: These bits determine which channel's request bit will be set. 00 = Ch. 0 10 = Ch. 2 01 = Ch. 1 11 = Ch. 3

Table 5-14 DMA Subsystem Registers (cont.)

7	6	5	4	3	2	1	0

I/O Address 00Fh/0DEh — Request Mask Register Read/Write Format

Reserved:	Mask bits:
Always reads 1.	These bits contain the state of the request mask bit associated with each request channel. The bit position corresponds to the channel number.

I/O Address 008h/0D0h — Status Register (RO)

Data Request:	Terminal Count:
These bits show the status of each channel request and are not affected by the state of the Mask Register bits. Reading a 1 means "request" occurs and bits 7 through 4 represent Channels 3 through 0, respectively. These bits can be cleared by a reset, Master Clear, of the pending request being deasserted.	These bits indicate which channel has reached the terminal count reading 1. These bits can be cleared by a reset, Master Clear, or each time a status read takes place. The channel number corresponds to the bit position.

I/O Address 40Bh — DMA Channel Select Register 1 — Default = 00h

Reserved:		Reserved:	DMA channel select:
Must be written to 0.	A setting of 00 on these bits gives compatible timing on DMA transfers. When set to 11, Type F DMA timing can be obtained.	Must be written to 0.	00 = Ch. 0 10 = Ch. 2 01 = Ch. 1 11 = Ch. 3

I/O Address 481h — DMA Channel 2 High Page Register — Default = 00h

These bits from the eight most significant address bits AD[31:24] when formatting the address for a DMA transfer.

I/O Address 482h — DMA Channel 3 High Page Register — Default = 00h

These bits from the eight most significant address bits AD[31:24] when formatting the address for a DMA transfer.

I/O Address 482h — DMA Channel 1 High Page Register — Default = 00h

These bits from the eight most significant address bits AD[31:24] when formatting the address for a DMA transfer

I/O Address 487h — DMA Channel 0 High Page Register — Default = 00h

These bits from the eight most significant address bits AD[31:24] when formatting the address for a DMA transfer.

I/O Address 488h — Reserved Register — Default = 00h

Reserved: Must be written 0.

I/O Address 489h — DMA Channel 6 High Page Register — Default = 00h

These bits from the eight most significant address bits AD[31:24] when formatting the address for a DMA transfer.

I/O Address 48Ah — DMA Channel 7 High Page Register — Default = 00h

These bits from the eight most significant address bits AD[31:24] when formatting the address for a DMA transfer.

I/O Address 48Bh — DMA Channel 5 High Page Register — Default = 00h

These bits from the eight most significant address bits AD[31:24] when formatting the address for a DMA transfer.

I/O Address 4D6h — DMA Channel Select Register 2 — Default = 00h

Reserved:		Reserved:	DMA channel select:
Must = 0	A setting of 00 on these bits gives compatible timing on DMA transfers. When set to 11, Type F DMA timing can be obtained.	Must = 0	00 = Ch. 4 10 = Ch. 6 01 = Ch. 5 11 = Ch. 7

5.3.4.2 Interrupt Controller Subsystem Register Descriptions

Table 5-15 Interrupt Controller Subsystem

7	6	5	4	3	2	1	0
I/O Address 020h (0A0h)			**ICW1 Register (WO)**				
Don't Care			Must be set to 1 for ICW1 since ICW1, OCW2, and OCW3 share the same address, 020h (0A0h).	Level trigger mode: Selects either the level or edge triggered mode input to the IR. (1)	Don't Care	Single mode: This bit selects between the Single and Cascade modes.(2)	Don't Care
1) If a 1 is written to LTM, a high level on the IR input will generate an interrupt request and the IR must be removed prior to EOI to prevent another interrupt. In the edge triggered mode, a low-to-high will generate an interrupt request. In either mode, IR must be held high until the first INTA cycle is started in order to generate the proper vector. IR7 vector will be generated if the IR input is negated early. 2) The Single mode is used whenever only one interrupt controller (INTC1) is used and is not recommended for this device. The Cascade mode allows the two interrupt controllers to be connected through IR2 of INTC1. INTC1 will allow INTC2 to generate its own interrupt vectors if the Cascade mode is selected and the highest priority IR pending is from an INTC2 input. INTC1 and INTC2 must be programmed for the Cascade mode.							
I/O Address 021h (0A1h)			**ICW2 Register (WO)**				
Vector bits 5 through 0: These bits are the upper five bits of the interrupt vector and are programmable by the CPU. INTC1 and INTC2 need not be programmed with the same value in ICW2. Usually INTC1 is programmed with 08h and INTC2 with 70h.					Vector bits 2 through 0: The lower three bits of the vector are generated by the Priority Resolver during INTA.		
I/O Address 021h			**ICW3 Register - Format for INTC1 (WO)**				
Slave mode: These bits select which IR inputs have Slave mode controller connected. ICW3 in INTC1 must be written with 04h (IRQ2) for INTC2 to function correctly.							
I/O Address 0A1h			**ICW3 Register - Format for INCT1 (WO)**				
Don't Care					Identify bits: Determines the Slave mode address the controller will respond to during the cascade INTA sequence. ICW3 in INTC2 should be written with a 02h (IRQ2 of INTC1) for operation in the Cascade mode.		
I/O Address 021h (0A1h)			**ICW4 Register (WO)**				
Don't Care			Enable multiple interrupts: This bit will enable multiple interrupts from the same channel in the Fixed Priority mode.(1)	Don't Care		Auto end of interrupt: An AEOI is enabled when this bit is 1. The interrupt controller will perform a non-specific EOI on the trailing edge of the second INTA cycle.(2)	Don't Care
1) This allows INTC2 to fully nest interrupts when the Cascade and Fixed Priority mode are both selected, without being blocked by INTC1. Correct handling in this type of mode requires the CPU to issue a non-specific EOI command to zero when exiting an interrupt service routine. If zero, a non-specific EOI command should be sent to INTC1. If non-zero, no command is issued. 2) Note this function should not be used in a device with fully nested interrupts unless the device is a cascade master type.							

Table 5-15 Interrupt Controller Subsystem (cont.)

7	6	5	4	3	2	1	0

I/O Address 021h (0A1h) — **OCW1 Register**

Mask bits:

These bits control the state of the Interrupt Mask Register. Each Interrupt Register can be masked by writing a 1 in the appropriate bit position (M0 controls IR0, etc.). Setting an IMR bit has no affect on lower priority requests. All IMR bits are cleared by writing ICW1.

I/O Address 020h (0A0h) — **OCW2 Register (WO)**

These bits are used to select various operating functions. Writing a 1 in bit 7 causes one of the rotate functions to be selected. Writing a 1 in bit 6 causes a specific or immediate function to occur. All specific commands require L[2:0] to be valid except no operation. Writing a 1 in bit 5 causes a function related to EOI to occur. 000 = Clear Rotate in Auto-EOI mode 001 = Non-specific EOI Command 011 = No Operation 011 = Specific EOI Command* 100 = Set Rotate in Auto-EOI Mode 101 = Rotate on Non-specific EOI Command 110 = Set Priority Command* 111 = Rotate on specific EOI Command *L[2:0] are used by these commands.	These bits must be set to 0 to indicate that OCW2 is selected, because ICW1, OCW2, and OCW3 share the same address. 020h (0A0h).	These three bits are internally decoded to select which interrupt channel is to be affected by the Specific command. L[2:0] must be valid during three of the four specific cycles.

I/O Address 020h (0A0h) — **OCW3 Register (WO)**

Reserved: This bit must be set to 0.	Enable Special Mask mode:(1) 0X = No operation 10 = Reset Special Mask mode to Normal Mask mode 11 = Set Special Mask mode	These bits must be set to 0 to indicate that OCW3 is selected because ICW1, OCW2, and OCW3 share the same address, 020h (0A0h).	Polled mode: Writing a 1 to this bit of OCW3 enables the Polled mode.(2)	Read Register: A 1 to this bit enables the contents of IRR or ISR (determined by RIS) to be placed on XD[7:0] when reading the Status Port at address 020h (0A0h). Asserting PM forces RR to reset. 0X = No Operation 10 = Read IRR on the next read 11 = Read ISR on the next read

(1) Writing a 1 in bit 5 enables the set/reset Special Mask mode function. ESMM allows the other functions in OCW3 to be accessed and manipulated without affecting the Special Mask mode (SMM) state.
During SMM, writing a 1 to any bit position of OCW1 inhibits interrupts and a 0 enables interrupts on the associated channel by causing the Priority Resolver to ignore the condition of the ISR.

(2) Writing OCW3 with the Polled mode acts like the first INTA cycle, freezing all interrupt request lines and resolving priority. The next read operation to the controller acts like a second INTA cycle and polled vector is output to the data bus. The format of polled vector is described later.

I/O Address 020h (0A0h) — **IIR Register**

Interrupt request bits:

These bits correspond to the interrupt request bits of the Interrupt Request Register. A 1 on these bits indicate that an interrupt request is pending on the corresponding line.

I/O Address 020h (0A0h) — **ISR Register**

Interrupt service bits:

These bits correspond to the interrupt service bits of the Interrupt Service Register. A 1 on these bits indicate that an interrupt is being serviced on the corresponding IS bits of the ISR.

Table 5-15 Interrupt Controller Subsystem (cont.)

7	6	5	4	3	2	1	0
I/O Address 20h (0A0h)				**Poll Vector**			
Interrupt: A 1 on this bit indicates that a pending interrupt is polled. If there is no pending interrupt request or the request is removed before the poll command, this bit is 0.	Don't Care				Vector bits: These bits are the binary encoding of the highest priority level pending interrupt request being polled. If no pending interrupt has been polled, all three bits are equal to 1.		

Interrupt Controller Subsystem Shadow Registers

The registers listed in this section (Interrupt Controller Subsystem) are accessed with Port 022h used as the Index Register and Port 024h as the Data Register. Each access to a register within this space consists of:

1) a write to Port 022h, specifying the desired register in the data byte,
2. followed by a read or write to Port 024h with the actual register data.

The index resets after every access; so every data access (via Port 024h) must be preceded by a write to Port 022h even if the same register is being accessed consecutively.

These registers shadow the write-only registers in the Interrupt Controller Subsystem. The values programmed into the "write-only" registers can be read back by accessing these registers.

Table 5-16 Interrupt Controller Subsystem Shadow Register

7	6	5	4	3	2	1	0
SYSCFG 80h	**PIC1ICW1 Register (RO)** Reads back the contents of the ICW1 Register at Address 020h.						
SYSCFG 81h	**PIC1ICW2 Register (RO)** Reads back the contents of the ICW2 Register at Address 021h.						
SYSCFG 82h	**PIC1ICW3 Register (RO)** Reads back the contents of the ICW3 Register at Address 021h.						
SYSCFG 83h	**PIC1ICW4 Register (RO)** Reads back the contents of the ICW4 Register at Address 021h.						
SYSCFG 84h	**Reserved**						
SYSCFG 85h	**PIC1OCW2 Register (RO)** Reads back the contents of the OCW2 Register at Address 020h.						
SYSCGF 86h	**PIC1OCW3 Register (RO)** Reads back the contents of the OCW3 Register at Address 020h.						
SYSCFG 87h	**Reserved**						
SYSCGF 88h	**PIC2ICW1 Register (RO)** Reads back the contents of the ICW1 Register at Address 0A0h.						
SYSCGF 89h	**PIC2ICW2 Register (RO)** Reads back the contents of the ICW2 Register at Address 0A1h.						
SYSCGF 8Ah	**PIC2ICW3 Register (RO)** Reads back the contents of the ICW3 Register at Address 0A1h.						
SYSCGF 8Bh	**PIC2ICW4 Register (RO)** Reads back the contents of the ICW4 Register at Address 0A1h.						
SYSCFG 8Ch	**Reserved**						
SYSCGF 8Dh	**PIC2OCW2 Register (RO)** Reads back the contents of the OCW2 Register at Address 0A0h.						
SYSCGF 8Eh	**PIC2OCW3 Register (RO)** Reads back the contents of the OCW3 Register at Address 0A0h.						

5.3.4.3 Counter/Timer Subsystem Register Descriptions

Table 5-17 Counter/Timer Subsystem Registers

7	6	5	4	3	2	1	0
I/O Address 043h			**Control Word Format (WO)**				
Select counter: These bits select which counter this control word is written to. 00 = Counter 0 01 = Counter 1 10 = Counter 2 11 = Rsrvd for read-back command		Read/write: These bits determine the counter read/write word size. 00 = Rsrvd for counter latch command 01 = R/W LSB only 10 = R/W MSB only 11 = R/W LSB first, then MSB		Mode select: These bits select the counter operating mode. 000 = Mode 0 001 = Mode 1 X10 = Mode 2 X11 = Mode 3 100 = Mode 4 101 = Mode 5			Binary coded decimal:(1) During R/W counter commands control word writing: 1 = binary coded decimal count format 0 = binary counting format
(1) During read-back command word writing, this bit 0 must = 0.							
I/O Address 043h			**Counter Latch Command Format (WO)**				
Select counter: These bits select which counter is being latched. 00 = Counter 0 01 = Counter 1 10 = Counter 2 11 = Rsrvd for read-back command		These bits must be 0 for the counter latch command.		Don't care			
I/O Address 043h			**Read-Back Command Format (WO)**				
These bits must be 1 for the read-back command.		Latch count: 0 = Latches the count of the counting component of the selected counter(s).	Latch status: 0 = Latches the status information of the selected counter(s).	Counter select: These bits select which counter(s) the read-back command is applied to. 0XX = Counter 2 X0X = Counter 1 XX0 = Counter 0			Reserved: Write as 0.
I/O Address 043h			**Status Format (RO)**				
Out: This bit contains the state of the OUT signal of the counter.	Null Count: This bit contains the condition of the null count flag.(1)	Read/Write Word: These bits indicate the counter read/write word size. This information is useful in determining where the high byte, the low byte, or both must be must be transferred during counter read/write operations.		Mode bits: These bits reflect the operating mode of the counter and are interpreted in the same manner as in the write control word format.			Binary Coded Decimal: This bit indicates the counting element is operating in binary format or BCD format.
1) This flag is used to indicate that the contents of the counting element are valid. It will be set to 1 during a write to the control register or the counter. It is cleared to a 0 whenever the counter is loaded from the counter input register.							
I/O Address 040h			**Counter 0 Count Value (WO)** Contains the initial count value of Counter 0.				
I/O Address 41h			**Counter 1 Count Value (WO)** Contains the initial count value of Counter 1.				
I/O Address 42h			**Counter 2 Count Value (WO)** Contains the initial count value of Counter 2.				

Counter/Timer Subsystem Shadow Register Descriptions
The remaining registers listed in this section (Counter/Timer Subsystem) are accessed with Port 022h used as the Index Register and Port 024h as the Data Register. Each access to a register within this space consists of:

1) a write to Port 022h, specifying the desired register in the data byte,
2. followed by a read or write to Port 024h with the actual register data.

The index resets after every access; so every data access (via Port 024h) must be preceded by a write to Port 022h even if the same register is being accessed consecutively.

These registers shadow the write-only registers in the Counter/Timer Subsystem. The values programmed into the "write-only" registers can be read back by accessing these registers.

Table 5-18 Counter/Timer Subsystem Shadow Registers

7	6	5	4	3	2	1	0
SYSCFG 90h	**CTSC0LB Register (RO)** Reads back the lower byte value of Counter 0 at Address 40h.						
SYSCFG 91h	**CTSC0HB Register (RO)** Reads back the higher byte value of Counter 0 at Address 40h.						
SYSCFG 92h	**CTSC1LB Register (RO)** Reads back the lower byte value of Counter 1 at Address 41h.						
SYSCFG 93h	**CTSC1HB Register (RO)** Reads back the higher byte value of Counter 1 at Address 41h.						
SYSCFG 94h	**CTSC2LB Register (RO)** Reads back the lower byte value of Counter 2 at Address 42h.						
SYSCFG 95h	**CTSC2HB Register (RO)** Reads back the higher byte value of Counter 2 at Address 42h.						
SYSCFG 96h	**Byte Pointer Register (RO)** The contents of this register is the Byte 2 pointer value.						

5.4 82C568 IDE Register Space

5.4.1 IDE Configuration Registers

The configuration space for the IDE Controller can be mapped in different locations in different OPTi chipsets. In the case of the 82C568 (Viper-MAX Chipset), it can be mapped into two locations. It is controlled by SYSCFG FFh[4]. If this bit is cleared to 0, the configuration space is mapped as Device 14h (AD31 = 1,) Function 0. If FFh[4] is set to 1, the IDE controller is mapped as Device 01h (AD12 = 1), Function 1. This section describes the registers implemented in the 256-byte configuration space. All registers not implemented always return zero during read cycles.

Table 5-19 IDE Configuration Registers

7	6	5	4	3	2	1	0
PCIIDE 00h **PCIIDE 01h**			**Vendor ID Register (RO)** OPTi ID				**Default = 45h** **Default = 10h**
PCIIDE 02h **PCIIDE 03h**			**Device ID Register (RO)** Identifies the ID of the IDE controller. If SYSCFG FFh[4] = 1, this register returns C568h.				**Default = 21h** **Default = C6h**
PCIIDE 04h			**Command Register - Low Byte**				**Default = 4xh**
Reserved (RO)	Parity checking: 0 = Parity checking ignored 1 = IDE controller generates PERR# if a parity error occurs during I/O write cycles For I/O read cycles, the IDE controller always generates parity bit.	Reserved (RO)	Memory write and invalidate: 0 = Memory write generates command. 1 = IDE controller generates command.	Reserved (RO)	IDE controller becomes a PCI master to generate PCI accesses: 0 = Disable 1 = Enable **Note:** This bit must be explicitly programmed.	Reserved	I/O accesses: IDE controller uses this bit to enable/disable I/O accesses. 0 = Disable 1 = Enable (Default = 1)
PCIIDE 05h			**Command Register - High Byte** Reserved (RO)				**Default = 00h**
PCIIDE 06h			**Status Register - Low Byte**				**Default = 80h**
Fast back-to-back transactions (RO): 0 = Disable 1 = Enable (Default = 1)	Reserved (RO)						
PCIIDE 07h			**Status Register - High Byte**				**Default = 02h**
Parity error: This bit is set whenever the IDE controller detects a parity error. It is cleared by writing 8000h to this register.	Reserved (RO)	Master abort: As a PCI master, the IDE controller sets this bit to 1 when its transaction is terminated with a master abort.	Target abort: As a PCI master, the IDE controller sets this bit to 1 when its transaction is terminated with a target abort.	Reserved (RO)	Select timing (RO): These read-only bits indicate the allowable timing assertion for DEVSEL#. (Default = 01)		Data parity: As a PCI master, the IDE controller sets this bit to 1 when it detected a data parity error.

Table 5-19 IDE Configuration Registers (cont.)

7	6	5	4	3	2	1	0
PCIIDE 08h			**Revision ID Register (RO)**				**Default = 00h**
			Identifies the revision number of the IDE controller				
PCIIDE 09h			**Class Code Register - Low Byte**				**Default = 80h**
Bus mastering IDE signature (RO): This bit is set to 1 to indicate master mode support. (Default = 1)	Reserved (RO)			Writability of the Native/Legacy bit for Secondary IDE (RO): Determines if bit 2 is RO or R/W. 0 = Bit 2 is RO 1 = Bit 2 is R/W This bit is set only when both 40h[3] and 40h[2] = 1.	Native/Legacy Mode for Secondary IDE: 0 = Legacy 1 = Native	Writability of the Native/Legacy bit for Primary IDE (RO): Determines if bit 0 is RO or R/W. 0 = Bit 0 is RO 1 = Bit 0 is R/W If 40h[2] = 0, this bit is 0. When the 40h[2] = 1, this bit is 1.	Native/Legacy Mode for Primary IDE: 0 = Legacy 1 = Native
PCIIDE 0Ah-0Bh			**Class Code Register (RO) - High Byte**				**Default = 0101h**
			The MSB indicates the base class code for the mass storage controller. The LSB byte indicates the sub-class code (IDE controller).				
PCIIDE 0Ch-0Dh			**Reserved Register (RO)**				**Default = 00h**
PCIIDE 0Eh			**Header Type Register (RO)**				**Default = 00h**
			Configuration bit for single (default) or multi-function device. If SYSCFG FFh[4] is set to 1, this register returns 80h denoting a multi-function device.				
PCIIDE 0Fh			**Built-In Self-Test Register (RO)**				**Default = 00h**
PCIIDE 10h-13h			**Primary IDE Command Block Base Address Register**				**Default = 1F1h w/40h[2] = 1**
			This register is the I/O space indicator for the Drive Command Block. The address block has a size of eight bytes. Bits [2:0] are read-only and default to 001. Bits [31:3] are writable if the 40h[2] bit is set to 1. If 40h[2] = 0, bits [31:0] are read-only and return 0.				
PCIIDE 14h-17h			**Primary IDE Control Block Base Address Register**				**Default = 3F5h w/40h[2] = 1**
			This register is the I/O space indicator for the Drive Control Block. The address block has a size of four bytes. Bits [1:0] are read-only and default to 01. Bits [31:3] are writable if the 40h[2] bit is set to 1. If 40h[2] = 0, bits [31:0] are read-only and return 0.				

Table 5-19 IDE Configuration Registers (cont.)

7	6	5	4	3	2	1	0
PCIIDE 18h-1Bh			**Secondary IDE Command Block Base Address Register**				**Default = 171h w/40h[2] = 1 and 40h[3] = 0**
This register is the I/O space indicator for the Drive Command Block. The address block has a size of eight bytes. Bits [2:0] are read-only and default to 001. Bits [31:3] are writable if the 40h[2] bit is set to 1. If 40h[2] = 0, bits [31:0] are read-only and return 0.							
PCIIDE 1Ch-1Fh			**Secondary IDE Control Block Base Address Register**				**Default = 375h w/40h[2] = 1 and 40h[3] = 0**
This register is the I/O space indicator for the Drive Control Block. The address block has a size of four bytes. Bits [1:0] are read-only and default to 01. Bits [31:3] are writable if the 40h[2] bit is set to 1. If 40h[2] = 0, bits [31:0] are read-only and return 0.							
PCIIDE 20h-23h			**Bus Master IDE Base Address Register**				**Default = 01000080h**
This register is the I/O base address indicator for the Bus Master IDE Registers. The address block has a size of 16 bytes. Bits [3:0] are read-only and default to 0001. Bits [31:4] are writable.							
PCIIDE 24h-3Bh			**Reserved Register (RO)**				**Default = 00h**
PCIIDE 3Ch			**Interrupt Line Register**				**Default = 14h**
This register indicates which input of the system interrupt controller the PCIIRQ3# interrupt pin is routed to.							
PCIIDE 3Dh			**Interrupt Pin Register (RO)**				**Default = 1**
The contents of this register is 1.							
PCIIDE 3Eh-3Fh			**Reserved Register (RO)**				**Default = 00h**
PCIIDE 40h			**IDE Initialization Control Register**				**Default = 00h**
Bus master IDE early request for PCI bus. The bus master IDE requests the PCI bus whenever the FIFO is: 00 = Disable, 32 bytes filled (1 cache line) 01 = 30 bytes filled 10 = 28 bytes filled 11 = 26 bytes filled		Enhanced Slave: 0 = 82C621A-compatible mode, uses a 16-byte FIFO in PIO Mode 1 = Enhanced mode, uses a 32-byte FIFO in PIO Mode	Reserved: Must be written to 0.	Secondary IDE: 0 = Enable 1 = Disable	Address relocation: Determines if I/O space addresses are relocatable via programming configuration space registers. 0 = Fixed I/O addresses (1F0h-1F7h, 3F6h for primary; 170h-177h, 376h for secondary) 1 = Relocatable I/O addresses		Mode: These bits control the default 16-bit cycle times for all IDE devices and can be overridden by programming the IDE I/O Registers. 00 = ≥ 600ns cycle time (PIO Mode 0) 01 = ≥ 383ns cycle time (PIO Mode 2) 10 = ≥ 240ns cycle time (PIO Mode 1) 11 = ≥ 180ns cycle time (PIO Mode 3)

Table 5-19 IDE Configuration Registers (cont.)

7	6	5	4	3	2	1	0
PCIIDE 41h			**Reserved Register (RO)**				**Default = 00h**
PCIIDE 42h			**IDE Enhanced Feature Register**				**Default = 00h**
Reserved: Must be written 0.		Slave IDE FIFO to ISA bus preemption: 0 = Enable 1 = Disable, 82C568 waits to generate ISA cycles until all data in the IDE FIFO is flushed out	IDE arbiter support for PCI/IDE concurrency: 0 = Disable 1 = Enable	PCI memory read line/write and invalidate commands: 0 = Disable 1 = Enable	Concurrent PCI master IDE and IDE cycle: 0 = Disable 1 = Enable (a 48-byte FIFO is used, master IDE to local memory and IDE devices will be accessed concurrently)	X-1-1-1 MIDE PCI master read/write transfers: 0 = Disable 1 = Enable	Reserved: Must be written 0.
PCIIDE 43h			**IDE Enhanced Mode Register**				**Default = 00h**
Enhanced Mode for Drive 1 on Secondary Channel: Sets 16-bit cycle times for IDE PIO Modes 3 and 4 or Multi-Word DMA Modes 1 and 2. 00 = Disabled, control by corresponding Timing Registers Set 01 = Command recovery in 1 LCLK 10 = Command recovery in 0 LCLK 11 = Reserved The corresponding 170h/171h[3:0] must be set to 0 before these two bits are set to 01 or 10.		Enhanced Mode for Drive 0 on Secondary Channel: Sets 16-bit cycle times for IDE PIO Modes 3 and 4 or Multi-Word DMA Modes 1 and 2. 00 = Disabled, control by corresponding Timing Registers Set 01 = Command recovery in 1 LCLK 10 = Command recovery in 0 LCLK 11 = Reserved The corresponding 170h/171h[3:0] must be set to 0 before these two bits are set to 01 or 10.		Enhanced Mode for Drive 1 on Primary Channel: Sets 16-bit cycle times for IDE PIO Modes 3 and 4 or Multi-Word DMA Modes 1 and 2. 00 = Disabled, control by corresponding Timing Registers Set 01 = Command recovery in 1 LCLK 10 = Command recovery in 0 LCLK 11 = Reserved The corresponding 1F0h/1F1h[3:0] must be set to 0 before these two bits are set to 01 or 10.		Enhanced Mode for Drive 0 on Primary Channel Sets 16-bit cycle times for IDE PIO Modes 3 and 4 or Multi-Word DMA mode 1 and 2. 00 = Disabled, control by corresponding Timing Registers Set 01 = Command recovery in 1 LCLK 10 = Command recovery in 0 LCLK 11 = Reserved The corresponding 1F0h/1F1h[3:0] must be set to 0 before these two bits are set to 01 or 10.

5.4.2 IDE I/O Registers

5.4.2.1 Primary IDE I/O Registers

The register addresses are referred to in this section by their power-up default addresses If the power-up default is modified by writing to Configuration Register IO1, then these registers will be relocated accordingly.

The IDE controller contains registers at seven I/O ports accessible after two consecutive 16-bit I/O reads from address 1F1h, followed by a byte write 03h to 1F2h. Any other I/O cycle between these two reads will disable access to the IDE controller registers. Refer to Section 4.18.3.2, Programming the IDE Mode Timing, for programming details.

Table 5-20 Primary IDE I/O Registers

7	6	5	4	3	2	1	0
I/O Address 1F2h				**Internal ID Register**			**Default = xxh**
Configuration disable (WO): 0 = Enable accesses to internal IDE controller registers 1 = Disable accesses to internal IDE controller registers until another two consecutive I/O reads from 1F1h. (Default = 1)	Configuration off (WO): 0 = Enable accesses to internal IDE controller registers 1 = Disable all accesses to internal IDE controller registers until power-down or reset.	Reserved (RO): Must be written 0. (Default = xxxx)				Reserved: Must be written 11. If not, all writes to the IDE I/O Registers will be blocked.	
I/O Address 1F0h, Index-0				**Read Cycle Timing Register-A**			**Default = xxh**
Read pulse width: The value programmed in this register plus one determines the DRD# pulse width in LCLKs (for a 16-bit read from the IDE Data Register). See Table 4-63 or Table 4-64. (Default = xxxx)				Read recovery time: The value programmed in this register plus two determines the recovery time between the end of DRD# and the next DA[2:0]/DCSx# being presented (after a 16-bit read from the IDE Data Register), measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xxxx)			
Note: Read Cycle Timing Register-A shares the I/O address with Read Cycle Timing Register-B, indexed by 1F6h[0]. It controls the read cycle timing of the IDE Data Register for the drive selected by 1F3h[3:2].							
I/O Address 1F0h, Index-1				**Read Cycle Timing Register-B**			**Default = xxh**
Read pulse width: The value programmed in this register plus one determines the DRD# pulse width in LCLKs (for a 16-bit read from the IDE Data Register). See Table 4-63 or Table 4-64. (Default = xxxx)				Read recovery time: The value programmed in this register plus two determines the recovery time between the end of DRD# and the next DA[2:0]/DCSx# being presented (after a 16-bit read from the IDE Data Register), measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xxxx)			
Note: Read Cycle Timing Register-B shares the I/O address with Read Cycle Timing Register-A, indexed by 1F6h[0]. It controls the read cycle timing of the IDE Data Register for the drive not selected by 1F3h[3:2] if 1F3h[7] = 1.							

Table 5-20 Primary IDE I/O Registers (cont.)

7	6	5	4	3	2	1	0
I/O Address 1F1h, Index-0			**Write Cycle Timing Register-A**				**Default = xxh**
Write pulse width: The value programmed in this register plus one determines the DWR# pulse width in LCLKs (for a 16-bit write from the IDE Data Register). See Table 4-63 or Table 4-64. (Default = xxxx)				Write recovery time: The value programmed in this register plus two determines the recovery time between the end of DWR# and the next DA[2:0]/DCSx# being presented (after a 16-bit write from the IDE Data Register), measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xxxx)			
Note: Write Cycle Timing Register-A shares the I/O address with Write Cycle Timing Register-B, indexed by 1F6h[0]. It controls the write cycle timing of the IDE Data Register for the drive selected by 1F3h[3:2].							
I/O Address 1F1h, Index-1			**Write Cycle Timing Register-B**				**Default = xxh**
Write pulse width: The value programmed in this register plus one determines the DWR# pulse width in LCLKs (for a 16-bit write from the IDE Data Register). See Table 4-63 or Table 4-64. (Default = xxxx)				Write recovery time: The value programmed in this register plus two determines the recovery time between the end of DWR# and the next DA[2:0]/DCSx# being presented (after a 16-bit write from the IDE Data Register), measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xxxx)			
Note: Write Cycle Timing Register-B shares the I/O address with Write Cycle Timing Register-A, indexed by bit 1F6h[0]. It controls the write cycle timing of the IDE Data Register for the drive not selected by 1F3h[3:2] if 1F3h[7] = 1.							
I/O Address 1F3h			**Control Register**				**Default = xxh**
Enable Timing Registers-B: Cycle Timing Registers-B (1F0h and 1F1h of Index-1) and Miscellaneous Timing Register 1F6h[5:1] to override the IDE timing set by 40h[1:0] for any drive not selected by 1F3h[3:2]. 0 = Disable 1 = Enable	Reserved (RO): Must be written 0.	Enable one wait state read: 0 = 2 WS minimum 1 = 1 WS minimum for data reads	Enable Timing Registers-A, Drive 1: Cycle Timing Registers-A (1F0h and 1F1h of Index-0) to override the IDE timing set 40h[1:0] for Drive 1. 0 = Disable 1 = Enable	Enable Timing Registers-A, Drive 0: Cycle Timing Registers-A (1F0h and 1F1h of Index-0) to override the IDE timing set by 40h[1:0] Drive 0. 0 = Disable 1 = Enable	Reserved: Must be written 0.	Reserved (RO): Must be written 1. (Default = 1)	
Note: For all new software controls the IDE timing through registers programming, bits 2, 3 and 7 of the Control Register should be enabled after the Cycle Timing Registers and Miscellaneous Register are programmed. See Table 4-62 for programming options.							
I/O Address 1F5h			**Strap Register**				**Default = xxh**
Reserved (RO): Must be written 1.	Revision number (RO): When the value of this register is set to 11, the contents of REVID Register (08h) should be used to find the revision level of the chip.	DINTR status (RO): Returns the state of the DINTR input.	Mode (RO): Returns information about drive speed as determined by 40h[1:0].		Reserved (RO): Must be written 1. (Default = 1)	PCI CLK speed: 0 = 33MHz 1 = 25MHz	

Table 5-20 Primary IDE I/O Registers (cont.)

7	6	5	4	3	2	1	0
I/O Address 1F6h			**Miscellaneous Register**				**Default = xxh**
Reserved: Must be written 0.	Read prefetch: 0 = Disable 1 = Enable	Address setup time: The value programmed in this register plus one determines the address setup time between DRD# or DWR# going active and DA[2:0], DCS3#, DCS1# being presented, measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xx)		Delay: The value programmed in this register plus two determines the minimum number of LCLKs between DRDY# going high and DRD# or DWR# going inactive. See Table 4-63 or Table 4-64. (Default = xxx)			Index-0: This bit is used to select between Cycle Timing Registers-A and -B located at 1F0h and 1F1h.

5.4.2.2 Secondary IDE I/O Registers

The register addresses are referred to in this section by their power-up default addresses. If the power-up default is modified by writing to Configuration Register IO3, then these registers will be relocated accordingly.

The IDE controller contains registers at seven I/O ports accessible after two consecutive 16-bit I/O reads from address 171h, followed by a byte write 03h to 1F2h. Any other I/O cycle between these two reads will disable access to the IDE controller registers. Refer to Section 4.18.3.2, Programming the IDE Mode Timing, for more details.

Table 5-21 Secondary IDE I/O Registers

7	6	5	4	3	2	1	0
I/O Address 172h			**Internal ID Register**				**Default = xxh**
Configuration disable: 0 = Enable accesses to internal IDE controller registers 1 = Disable accesses to internal IDE controller registers until another two consecutive I/O reads from 171h. (Default = 1)	Configuration off (WO): 0 = Enable accesses to internal IDE controller registers 1 = Disable all accesses to internal IDE controller registers until power-down or reset.	Reserved (RO): Must be written 0. (Default = xxxx)				Reserved: Must be written 11. If not, all writes to IDE I/O Registers will be blocked.	

I/O Address 170h, Index-0	**Read Cycle Timing Register-A**	**Default = xxh**
Read pulse width: The value programmed in this register plus one determines the DRD# pulse width in LCLKs (for a 16-bit read from the IDE Data Register). See Table 4-63 or Table 4-64. (Default = xxxx)	Read recovery time: The value programmed in this register plus two determines the recovery time between the end of DRD# and the next DA[2:0]/DCSx# being presented (after a 16-bit read from the IDE Data Register), measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xxxx)	

Note: Read Cycle Timing Register-A shares the I/O address with Read Cycle Timing Register-B, indexed by 176h[0]. It controls the read cycle timing of the IDE Data Register for the drive selected by 173h[3:2].

Table 5-21 Secondary IDE I/O Registers (cont.)

7	6	5	4	3	2	1	0
I/O Address 170h, Index-1			**Read Cycle Timing Register-B**				**Default = xxh**
Read pulse width: The value programmed in this register plus one determines the DRD# pulse width in LCLKs (for a 16-bit read from the IDE Data Register). See Table 4-63 or Table 4-64. (Default = xxxx)				Read recovery time: The value programmed in this register plus two determines the recovery time between the end of DRD# and the next DA[2:0]/DCSx# being presented (after a 16-bit read from the IDE Data Register), measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xxxx)			
Note: Read Cycle Timing Register-B shares the I/O address with Read Cycle Timing Register-A, indexed by 176h[0]. It controls the read cycle timing of the IDE Data Register for the drive not selected by 173h[3:2] if 173h[7] = 1.							
I/O Address 171h, Index-0			**Write Cycle Timing Register-A**				**Default = xxh**
Write pulse width: The value programmed in this register plus one determines the DWR# pulse width in LCLKs (for a 16-bit write from the IDE Data Register). See Table 4-63 or Table 4-64. (Default = xxxx)				Write recovery time: The value programmed in this register plus two determines the recovery time between the end of DWR# and the next DA[2:0]/DCSx# being presented (after a 16-bit write from the IDE Data Register), measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xxxx)			
Note: Write Cycle Timing Register-A shares the I/O address with Write Cycle Timing Register-B, indexed by 176h[0]. It controls the write cycle timing of the IDE Data Register for the drive selected by 173h[3:2.							
I/O Address 171h, Index-1			**Write Cycle Timing Register-B**				**Default = xxh**
Write pulse width: The value programmed in this register plus one determines the DWR# pulse width in LCLKs (for a 16-bit write from the IDE Data Register). See Table 4-63 or Table 4-64. (Default = xxxx)				Write recovery time: The value programmed in this register plus two determines the recovery time between the end of DWR# and the next DA[2:0]/DCSx# being presented (after a 16-bit write from the IDE Data Register), measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xxxx)			
Note: Write Cycle Timing Register-B shares the I/O address with Write Cycle Timing Register-A, indexed by 176h[0]. It controls the write cycle timing of the IDE Data Register for the drive not selected by 173h[3:2] if 173h[7] = 1.							
I/O Address 173h			**Control Register**				**Default = xxh**
Enable Timing Registers-B: Cycle Timing Registers-B (170h and 171h of Index-1) and Miscellaneous Timing Register 176h[5:1] to override the IDE timing set by 40h[1:0] for any drive not selected by 173h[3:2]. 0 = Disable 1 = Enable	Reserved: Must be written 0.			Enable Timing Registers-A, Drive 1: Cycle Timing Registers-A (170h and 171h of Index-0) to override the IDE timing set by 40h[1:0] for Drive 1. 0 = Disable 1 = Enable	Enable Timing Registers-A, Drive 0: Cycle Timing Registers-A (170h and 171h of Index-0) to override the IDE timing set by 40h[1:0] for Drive 1. 0 = Disable 1 = Enable	Reserved: Must be written 0.	Reserved (RO): Must be written 1. (Default = 1)
Note: For all new software controls the IDE timing through programming, bits 2, 3, and 7 of the Control Register should be enabled after the Cycle Timing Registers and Miscellaneous Register are programmed. See Table 4-62 for programming options.							

Table 5-21 Secondary IDE I/O Registers (cont.)

7	6	5	4	3	2	1	0
I/O Address 175h			**Strap Register**				**Default = xxh**
Reserved: Must be written 1.	Revision number (RO): When the value of this register is set to 11, the contents of REVID Register (08h) should be used to find the revision level of the chip.		SDINTR status (RO): Returns the state of the SDINTR input. (Default = x)	Reserved: Must be written 0.		Reserved (RO): Must be written 1.	Reserved (RO): Must be written 0.
I/O Address 176h			**Miscellaneous Register**				**Default = xxh**
Reserved: Must be written 0.	Read prefetch: 0 = Disable 1 = Enable	Address setup: The value programmed in this register plus one determines the address setup time between DRD# or DWR# going active and DA[2:0], DCS3#, DCS1# being presented, measured in LCLKs. See Table 4-63 or Table 4-64. (Default = xx)		DRDY delay: The value programmed in this register plus two determines the minimum number of LCLKs between DRDY# going high and DRD# or DWR# going inactive. See Table 4-63 or Table 4-64. (Default = xxx)			Index-0: This bit is used to select between Cycle Timing Registers-A and -B located at 170h and 171h.

5.4.3 Bus Master IDE Registers

The bus master IDE function uses 16 bytes of I/O space. The base address of this block of I/O space is pointed to by the Bus Master IDE Base Address Register (IO5) in the PCI Configuration space. All bus master IDE I/O space registers can be accessed as byte, word, or dword quantities. The description of the 16 bytes of I/O registers is shown in Table 5-22 and the individual bit formats for each register follow in Table 5-23.

Table 5-22 Bus Master IDE Registers

Offset from Base Address	Register Access	Register Name/Function
00h	R/W	Bus Master IDE Command Register for Primary IDE
01h		Device-specific
02h	RWC	Bus Master IDE Status Register for Primary IDE
03h		Device-specific
04h-07h	R/W	Bus Master IDE PRD Table Address for Primary IDE
08h	R/W	Bus Master IDE Command Register for Secondary IDE
09h		Device-specific
0Ah	RWC	Bus Master IDE status Register for Secondary IDE
0Bh		Device-specific
0Ch-0Fh	R/W	Bus Master IDE PRD Table Address for Secondary IDE

Table 5-23 Bus Master IDE Register Formats

7	6	5	4	3	2	1	0
Base Address + 00h			**Bus Master IDE Command Register for Primary IDE**				**Default = 00h**
Reserved: Must be written 0.				Read or write control: Sets the direction of the bus master transfer. 0 = PCI bus master reads 1 = PCI bus master writes This bit must not be changed when the bus master function is active.	Reserved: Must be written 0.		Start/Stop bus master: Writing a 1 to this bit enables bus master operation of the controller. Bus master operation begins when this bit is detected changing from 0 to 1. The controller will transfer data between the IDE device and memory only when this bit is set.(1)

(1) Master operation can be halted by writing 0 to this bit. All state information is lost when a 0 is written; master mode operation cannot be stopped and then resumed. If this bit is reset while bus master operation is still active (i.e., the Bus Master IDE Active bit of the Bus Master IDE Status Register for that IDE channel is set) and the drive has not yet finished its data transfer (the Interrupt bit in the Bus Master IDE Status Register for that IDE channel is not set), the bus master command is said to be aborted and data transferred from the drive may be discarded before being written to memory. This bit is intended to be reset after the data transfer is completed, as indicated by either the Bus Master IDE Active bit or the Interrupt bit of the Bus Master IDE Status Register for that IDE channel.

Table 5-23 Bus Master IDE Register Formats (cont.)

7	6	5	4	3	2	1	0
Base Address + 02h			**Bus Master IDE Status Register for Primary IDE**				**Default = 00h**
Simplex only (RO): This bit indicates that both bus master channels (primary and secondary) can be operated at the same time.	Drive 1 DMA capable: This bit is set by device-dependent code (BIOS or device driver) to indicate that Drive 1 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance.	Drive 0 DMA capable: This bit is set by device-dependent code (BIOS or device driver) to indicate that Drive 0 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance.	Reserved: Must be written 0.		Interrupt: This bit is set by the rising edge of the IDE interrupt line. It is cleared when a 1 is written to it by software. Software can use this bit to determine if an IDE device has asserted its interrupt line. When this bit is read as a 1, all data transferred from the drive is visible in system memory.	Error: This bit is set when the IDE controller encounters an error transferring data to/from memory. The exact error condition is bus-specific and can be determined in a bus-specific manner. This bit is cleared when a 1 is written to it by software.	Bus Master IDE Active: This bit is set when the Start bit is written to the Command Register. It is cleared when the last transfer for a region is performed, where EOT (end of transfer) for that region is set in the region descriptor. It is also cleared when the Start bit is cleared in the Command Register. When this bit is read as 0, all data transferred from the drive during the previous bus master command is visible in system memory, unless the bus master command was aborted.
Base Address + 04h			**Descriptor Table Pointer Register for Primary IDE**				**Default = 00h**
Bits [1:0] - Reserved Bits [31:2] - Base Address of Descriptor Table: Corresponds to A[31:2]. **Note:** The Descriptor Table must be dword aligned and must not cross a 64K boundary in memory.							

Table 5-23 Bus Master IDE Register Formats (cont.)

7	6	5	4	3	2	1	0
Base Address + 08h				**Bus Master IDE Command Register for Secondary IDE**			
Reserved: Must be written 0.				Read or write control: This bit sets the direction of the bus master transfer. 0 = PCI bus master reads 1 = PCI bus master writes This bit must not be changed when the bus master function is active.	Reserved: Must be written 0.		Start/Stop bus master: Writing a 1 to this bit enables bus master operation of the controller. Bus master operation begins when this bit is detected changing from 0 to 1. The controller will transfer data between the IDE device and memory only when this bit is set.(1)

(1) Master operation can be halted by writing 0 to this bit. All state information is lost when a 0 is written; master mode operation cannot be stopped and then resumed. If this bit is reset while bus master operation is still active (i.e., the Bus Master IDE Active bit of the Bus Master IDE Status Register for that IDE channel is set) and the drive has not yet finished its data transfer (the Interrupt bit in the Bus Master IDE Status Register for that IDE channel is not set), the bus master command is said to be aborted and data transferred from the drive may be discarded before being written to memory. This bit is intended to be reset after the data transfer is completed, as indicated by either the Bus Master IDE Active bit or the Interrupt bit of the Bus Master IDE Status Register for that IDE channel.

Table 5-23 Bus Master IDE Register Formats (cont.)

7	6	5	4	3	2	1	0
Base Address + 0Ah			**Bus Master IDE Status Register for Secondary IDE**				**Default = 00h**
Simplex only (RO): This bit indicates that both bus master channels (primary and secondary) can be operated at the same time.	Drive 1 DMA Capable: This bit is set by device dependent code (BIOS or device driver) to indicate that Drive 1 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance.	Drive 0 DMA Capable: This bit is set by device dependent code (BIOS or device driver) to indicate that Drive 0 for this channel is capable of DMA transfers and that the controller has been initialized for optimum performance.	Reserved: Must be written 0.		Interrupt: This bit is set by the rising edge of the IDE interrupt line. It is cleared when a 1 is written to it by software. Software can use this bit to determine if an IDE device has asserted its interrupt line. When this bit is read as a 1, all data transferred from the drive is visible in system memory.	Error: This bit is set when the controller encounters an error transferring data to/from memory. The exact error condition is bus-specific and can be determined in a bus-specific manner. This bit is cleared when a 1 is written to it by software.	Bus Master IDE Active: This bit is set when the Start bit is written to the Command Register. It is cleared when the last transfer for a region is performed, where EOT (end of transfer) for that region is set in the region descriptor. It is also cleared when the Start bit is cleared in the Command Register. When this bit is read as 0, all data transferred from the drive during the previous bus master command is visible in system memory, unless the bus master command was aborted.
Base Address + 0Ch			**Descriptor Table Pointer Register for Secondary IDE**				**Default = 00h**
Bits [1:0] - Reserved Bits [31:2] - Base Address of Descriptor Table: Corresponds to A[31:2].							
Note: The Descriptor Table must be dword aligned and must not cross a 64K boundary in memory.							

5.5 Register Summary

Tables 5-24 and 5-25 are summary tables of registers located within the PCI Configuration, System Control Register and I/O Register Spaces of the 82C567 and 82C568, respectively.

Table 5-24 82C567 Register Summary

Loc.	Register Name	Default Value
PCIDV0 - PCI Specific		
00h-01h	Vendor Identification (RO)	4510h
02h-03h	Device Identification (RO)	67C5h
04h-05h	Command	0700h
06h-07h	Status	8002h
08h	Revision Identification (RO)	00h
09h-0Bh	Class Code (RO)	000006h
0Ch	Reserved	00h
0Dh	Master Latency Timer (RO)	00h
0Eh	Header Type (RO)	00h
0Fh	Built-In Self-Test (BIST) (RO)	00h
10h-3Fh	Reserved (RO)	00h
PCIDV0 - System Control		
40h-41h	Memory Control	00h
42h-43h	Reserved	00h
44h	82C566 Control Register 1	00h
45h	82C566 Control Register 2	00h
46h	82C566 Control Register 3	00h
47h	82C566 Control Register 4	00h
SYSCFG - System Control		
00h	Byte Merge/Prefetch & Sony Cache Module Control	00h
01h	DRAM Control Register 1	00h
02h	Cache Control Register 1	00h
03h	Cache Control Register 2	00h
04h	Shadow RAM Control Register 1	00h
05h	Shadow RAM Control Register 2	00h
06h	Shadow RAM Control Register 3	00h
07h	Tag Test	00h
08h	CPU Cache Control	00h
09h	System Memory Function	00h
0Ah	DRAM Hole A Address Decode Reg. 1	00h
0Bh	DRAM Hole B Address Decode Reg. 2	00h
0Ch	DRAM Hole Higher Address	00h
0Dh	Clock Control	00h

Table 5-24 82C567 Register Summary (cont.)

Loc.	Register Name	Default Value
0Eh	PCI Master Burst Control Register 1	00h
0Fh	PCI Master Burst Control Register 2	00h
10h	Miscellaneous Control Register 1	00h
11h	Miscellaneous Control Register 2	00h
12h	Refresh Control Register	00h
13h	Memory Decode Control Register 1	00h
14h	Memory Decode Control Register 2	00h
15h	PCI Cycle Control Register 1	00h
16h	Dirty/Tag RAM Control Register	00h
17h	PCI Cycle Control Register 2	00h
18h	Tristate Control Register	00h
19h	Memory Decode Control Register 3	00h
1Ah	Memory Shadow Control Register 1	00h
1Bh	Memory Shadow Control Register 2	00h
1Ch	EDO DRAM Control	00h
1Dh	Miscellaneous Control Register 3	00h
1Eh	BOFF# Control	00h
1Fh	EDO Timing Control	00h
20h	DRAM Burst Control	00h
21h	PCI Concurrence Control	00h
22h	Inquire Cycle Control	00h
23h	Pre-Snoop Control	00h
24h	Asymmetric DRAM Configuration	00h
25h	GUI Memory Location	00h
26h	UMA Control	00h
27h	Self Refresh Timing	00h
28h	SDRAM Burst and Latency Control	00h
29h	SDRAM Selection	00h
2Ah	PCI-to-DRAM Deep Buffer Size	00h
2Bh	EDO/SDRAM Time-Out	00h
2Ch	CPU-to-DRAM Buffer Control	00h
2Dh	Bank-wise EDO Timing Selection	00h
2Eh	PCI Master - GUI Retry Control	00h
2Fh	CAS Address Setup Time Control	00h
I/O Address		
8Fh	Refresh Page Register (WO)	
CF8h	Configuration/NVM Address Register	
CFCh	Configuration/NVM Data Register	

Table 5-25 82C568 Register Summary

Loc.	Register Name	Default Value
PCIDV1 - PCI Specific Registers		
00h-01h	Vendor Identification (RO)	4510h
02h-03h	Device Identification (RO)	68C5h
04h-05h	Command	0700h
06h-07h	Status	8002h
08h	Revision Identification (RO)	12h
09h-0Bh	Class Code (RO)	000106h
0Ch	Reserved	00h
0Dh	Master Latency Timer (RO)	00h
0Eh	Header Type (RO)	00h
0Fh	Built-In Self-Test (BIST) (RO)	00h
10h-3Fh	Reserved (RO)	00h
PCIDV1 - System Control Registers		
40h-41h	Keyboard Control	00h
42h-43h	Interrupt Edge/Level Control	00h
44h-45h	Pin Functionality Register 1	00h
46h-47h	Cycle Control Register 1	00h
48h-49h	Pin Functionality Register 2	00h
4Ah-4Bh	ROMCS# Range Control	00h
4Ch-4Dh	Reserved	00h
4Eh-4Fh	Miscellaneous Control	00h
50h-51h	Interrupt Trigger Control	00h
52h-53h	Interrupt Multiplexing Control	00h
54h-55h	PCI Master Control	00h
56h-57h	Serial Interrupt Source	00h
58h	Serial Interrupt Mode Control	00h
59h	Pin Functionality Register 3	00h
5Ah-5Bh	Distributed DMA Master Base Address	00h
5Ch	Distributed DMA Control	00h
5Dh	Reserved	00h
5Eh	Steerable DRQ Control	00h
5Fh	Steerable IRQ Control	00h
60h	USB Interrupt Control	00h
61h-FCh	Reserved	00h
FDh	Reserved	xxh
FEh	Stop Grant Cycle Control	xxh
FFh	Host Memory Parity Error	xxh

Table 5-25 82C568 Register Summary (cont.)

Loc.	Register Name	Default Value
SYSCFG - Power Mgmnt Registers		
E0h	GREEN Mode Control/Enable Status	00h
E1h	EPMI Control / GREEN Event Timer	00h
E2h	GREEN Event Timer Initial Count	00h
E3h	IRQ Event Enable Register 1	00h
E4h	IRQ Event Enable Register 2	00h
E5h	DREQ Event Enable	00h
E6h	Device Cycle Monitor Enable	00h
E7h	Wake-up Source/Programmable IO/ Memory Address Mask	00h
E8h	Programmable IO/MEM Address Range	00h
E9h	Programmable IO/MEM Address Range	00h
EAh	Enter GREEN State Port	01h
EBh	Return to NORMAL State Configuration Port	01h
ECh	Shadow Register for External Power Control Latch	01h
EDh	Device Cycle Detection Enable / Status	00h
EEh	STPCLK# Modulation	00h
EFh	Miscellaneous Register	00h
F0h	Device Timer CLK Select / Enable Status	00h
F1h	Device Timer 0 Initial Count	00h
F2h	Device Timer 1 Initial Count	00h
F3h	Device Timer IO/MEM Select, Mask Bits	00h
F4h-F5h	Device 0 IO/MEM Address	00h
F6h-F7h	Device 1 IO/MEM Address	00h
FAh-FBh	Reserved	00h
FCh	Power Management Control Register 1	00h
FDh	Power Management Control Register 2	00h
FEh	Power Management Control Register 3	00h
FFh	General Purpose Chip Select Control	00h
I/O Address - I/O Port Registers		
061h	Port B Register	
060h, 064	Keyboard I/O Control Registers	
092h	PS/2 Reset Control Registers	00h
Integrated Internal 82C206		
SYSCFG2 - Index Port 022h, Data Port 023h		

Table 5-25 82C568 Register Summary (cont.)

Loc.	Register Name	Default Value
01h	Configuration Register	C0h
I/O Address - DMA Subsystem Registers		
008h/0D0h	Command Register (WO)	
00Bh/0D6h	Mode Register	
009h/0D2h	Request Register Write Format	
009h/0D2h	Request Mask Register Set/Reset Format	
00Fh/0DEh	Request Mask Register Read/Write Format	
008h/0D0h	Status Register (RO)	
40Bh	DMA Channel Select Register 1	00h
481h	DMA Channel 2 High Page Register	00h
482h	DMA Channel 3 High Page Register	00h
482h	DMA Channel 1 High Page Register	00h
487h	DMA Channel 0 High Page Register	00h
488h	Reserved	00h
489h	DMA Channel 6 High Page Register	00h
48Ah	DMA Channel 7 High Page Register	00h
48Bh	DMA Channel 5 High Page Register	00h
4D6h	DMA Channel Select Register 2	00h
I/O Address - Interrupt Controller Subsystem Registers		
020h/0A0h	ICW1 Register (WO)	
021h/0A1h	ICW2 Register (WO)	
021h	ICW3 Register - Format for INTC1 (WO)	
0A1h	ICW3 Register - Format for INCT1 (WO)	
021h/0A1h	ICW4 Register (WO)	
021h/0A1h	OCW1 Register	
020h/0A0h	OCW2 Register (WO)	
020h/0A0h	OCW3 Register (WO)	
020h/0A0h	IIR Register	
020h/0A0h	ISR Register	
020h/0A0h	Poll Vector	
SYSCFG - Interrupt Controller Subsystem Shadow Registers		
80h	PIC1ICW1 Register (RO)	
81h	PIC1ICW2 Register (RO)	
82h	PIC1ICW3 Register (RO)	
83h	PIC1ICW4 Register (RO)	
84h	Reserved	

Table 5-25 82C568 Register Summary (cont.)

Loc.	Register Name	Default Value
85h	PIC1OCW2 Register (RO)	
86h	PIC1OCW3 Register (RO)	
87h	Reserved	
88h	PIC2ICW1 Register (RO)	
89h	PIC2ICW2 Register (RO)	
8Ah	PIC2ICW3 Register (RO)	
8Bh	PIC2ICW4 Register (RO)	
8Ch	Reserved	
8Dh	PIC2OCW2 Register (RO)	
8Eh	PIC2OCW3 Register (RO)	
I/O Address - Counter/Timer Subsystem Registers		
043h	Control Word Format (WO)	
043h	Counter Latch Command Format (WO)	
043h	Read-Back Command Format (WO)	
043h	Status Format (RO)	
040h	Counter 0 Count Value (WO)	
041h	Counter 1 Count Value (WO)	
042h	Counter 2 Count Value (WO)	
SYSCFG - Counter/Timer Subsystem Shadow Registers		
90h	CTSC0LB Register (RO)	
91h	CTSC0HB Register (RO)	
92h	CTSC1LB Register (RO)	
93h	CTSC1HB Register (RO)	
94h	CTSC2LB Register (RO)	
95h	CTSC2HB Register (RO)	
96h	Byte Pointer Register (RO)	
Integrated Internal IDE		
PCIIDE - PCI Specific Registers		
00h-01h	Vendor ID (RO)	4510h
02h-03h	Device ID (RO)	21C6h
04h-05h	Command	4x00h
06h-07h	Status	8002h
08h	Revision ID (RO)	00h
09h-0Bh	Class Code	800101h
0Ch-0Dh	Reserved (RO)	00h
0Eh	Header Type (RO)	00h
0Fh	Built-In Self-Test Register (RO)	00h

Table 5-25 82C568 Register Summary (cont.)

Loc.	Register Name	Default Value
10h-13h	Primary IDE Command Block Base Address Register	1F1h w/ 40h[2]=1
14h-17h	Primary IDE Control Block Base Address Register	3F5h w/ 40h[2]=1
18h-1Bh	Secondary IDE Command Block Base Address Register	171h w/ 40h[2]=1 and 40h[3]=0
1Ch-1Fh	Secondary IDE Control Block Base Address Register	375h w/ 40h[2]=1 and 40h[3]=0
20h-23h	Bus Master IDE Base Address Register	0100 0080h
24h-3Bh	Reserved (RO)	00h
3Ch	Interrupt Line	14h
3Dh	Interrupt Pin (RO)	1
3Eh-3Fh	Reserved (RO)	00h
PCIIDE - System Control Registers		
40h	IDE Initialization Control	00h
41h	Reserved (RO)	00h
42h	IDE Enhanced Feature	00h
43h	IDE Enhanced Mode Register	00h
I/O Address - Primary IDE I/O Registers		
1F2h	Internal ID	

Table 5-25 82C568 Register Summary (cont.)

Loc.	Register Name	Default Value
1F0h, Index-0	Read Cycle Timing Register-A	xxh
1F0h, Index-1	Read Cycle Timing Register-B	xxh
1F1h, Index-0	Write Cycle Timing Register-A	xxh
1F1h, Index-1	Write Cycle Timing Register-B	xxh
1F3h	Control	xxh
1F5h	Strap	xxh
1F6h	Miscellaneous	xxh
I/O Address - Secondary IDE I/O Registers		
172h	Internal ID	xxh
170h, Index-0	Read Cycle Timing Register-A	xxh
170h, Index-1	Read Cycle Timing Register-B	xxh
171h, Index-0	Write Cycle Timing Register-A	xxh
171h, Index-1	Write Cycle Timing Register-B	xxh
173h	Control	xxh
175h	Strap	xxh
176h	Miscellaneous	xxh

6.4 82C566 AC Characteristics (66MHz - Preliminary)

Sym	Parameter	Min	Max	Unit	Figure	Condition
t101	HD[63:0] to MD[63:0] bus valid	2	22	ns		
t102	DLE1# low to HD[63:0] bus valid	2	22	ns		
t103	DLE0# low to HD[63:0] bus valid	2	22	ns		
t104	MD[31:0]# setup to DLE0# low	5		ns		
t105	MD[63:32] setup to DLE1# low	5		ns		
t106	HD[31:0] setup to DLE0# low	5		ns		
t107	HD[63:32] setup to DLE1# low	5		ns		
t108	MD[31:0]# hold from DLE0# high	5		ns		
t109	MD[63:32] hold from DLE1# high	5		ns		
t110	HD[31:0] hold from DLE0# high	8		ns		
t111	HD[63:32] hold from DLE0# high	8		ns		
t112	HDOE# high to HD[63:0] high-Z	2	11	ns		
t113	HDOE# high to HD[31:0] high-Z	2	11	ns		
t114	MDOE# high to MD[63:0] high-Z	2	15	ns		

6.5 82C567 AC Characteristics (66MHz - Preliminary)

Sym	Parameter	Min	Max	Unit	Figure	Condition
t201	CPUCLK/LCLK to BRDY# active delay	5	15	ns	6-3	
t202	CPUCLK/LCLK to BRDY# inactive delay	5	15	ns		
t203	ECA4, OCA4 delay from CPUCLK/LCLK rising	5	15	ns	6-3	
t204	HACALE delay from CPUCLK/LCLK rising	5	15	ns		
t205	ECDOE#, even bank cache, falling edge valid delay from CPUCLK/LCLK rising	5	15	ns	6-3	
t206	OCDOE#, odd bank cache, falling edge valid delay from CPUCLK/LCLK rising	5	15	ns		
t207	ADS# setup to CLK high	2		ns		
t208	ADS# hold time from CLK high	1		ns	6-2	
t209	M/IO#, D/C#, W/R#, CACHE# setup to CLK high	1		ns	6-1	Sampled one CLK after ADS#
t210	CPUCLK/LCLK to DIRTYWE# active delay	5	14	ns	6-3	
t211	CPUCLK/LCLK to DIRTYWE# inactive delay	5	14	ns		
t212	CPUCLK/LCLK to TAGWE# active delay	5	14	ns	6-3	
t213	CPUCLK/LCLK to TAGWE# inactive delay	5	14	ns		
t214	ECAWE#, even bank cache, falling edge valid delay from CPUCLK/LCLK high	5	15	ns	6-3	
t215	OCAWE#, odd bank cache, falling edge valid delay from CPUCLK/LCLK high	5	15	ns	6-3	
t216	CPUCLK/LCLK to NA# active delay	5	15	ns	6-3	

6.0 Maximum 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	5.0V Supply Voltage	4.5	5.5	NA	NA	V
VDD	3.3V Supply Voltage	NA	NA	2.97	3.63	
VI	Input Voltage	–0.5	VCC + 0.5	–0.5	VDD + 0.33	V
VO	Output Voltage	–0.5	VCC + 0.5	–0.5	VDD + 0.33	V
TOP	Operating Temperature	0	70	0	70	°C
TSTG	Storage Temperature	–40	125	–40	125	°C

6.2 DC Characteristics: 5.0 Volt (VCC = 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		240	mA	In a 66/100MHz system

6.3 DC Characteristics: 3.3 Volt (VDD = 3.3V ±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	VDD + 0.33	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 = VDD
IOZ	Tristate Leakage Current		10.0	µA	
CIN	Input Capacitance		10.0	pF	
COUT	Output Capacitance		10.0	pF	
ICC	Power Supply Current		115	mA	In a 66/100MHz system

- The total average Viper-MAX Chipset power dissipation for a system running at 66/100MHz is 1550mW (2 x 175 + 600 + 600 = 1550mW). The average power dissipation for each device within the chipset:
  - 82C566 = 175mW
  - 82C567 = 600mW
  - 82C568 = 600mW

82C567 AC Characteristics (66MHz - Preliminary) (cont.)

Sym	Parameter	Min	Max	Unit	Figure	Condition
t217	CPUCLK/LCLK to NA# inactive delay	5	15	ns		
t218	TAG[7:0] data read to BRDY# low		5	ns	6-1	
t219	CPUCLK/LCLK to ADSC# active delay (for sync SRAM)	5	15	ns	6-3	
t220	CPUCLK/LCLK to ADV# active delay (for sync SRAM)	5	15	ns	6-3	
t221	CPUCLK/LCLK to SYNCS0#, SYNCS1# active delay (for sync SRAM)	5	15	ns	6-3	
t222	CPUCLK/LCLK to CAWE[7:0]# active delay (for sync SRAM)	5	15	ns	6-3	
t223	HA[31:3] valid delay from LCLK high	2	18	ns	6-3	
t224	AHOLD valid delay from CLK high	5	15	ns	6-3	
t225	EADS# valid delay from CLK high	5	15	ns	6-3	
t226	RESET rising edge valid from CLK high	5	15	ns	6-3	
t227	RESET falling edge valid delay from CLK high	5	15	ns	6-3	
t228	KEN#/LMEM# valid delay from CPUCLK/LCLK high	5	15	ns	6-3	
t229	RAS[3:0]# valid delay from CPUCLK high/LCLK high	2	15	ns	6-3	
t230	CAS[7:0]# valid delay from CPUCLK high/LCLK high	2	15	ns	6-3	
t231	MA[11:0] valid delay from CPUCLK high/LCLK high	2	15	ns	6-3	
t232	DWE# valid delay from CPUCLK high/LCLK high	2	15	ns	6-3	
t233	MA[11:0] propagation delay from HA[28:3]	2	22	ns		
t234	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, DEVSEL# valid delay from LCLK rising	2	11	ns	6-3	
t235	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, DEVSEL# setup time to LCLK rising	7		ns	6-1	
t236	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, DEVSEL# hold time from LCLK rising	0		ns		
t237	AD[31:0] valid delay from LCLK high	2	11	ns	6-3	
t238	AD[31:0] setup time to LCLK high	7		ns	6-1	
t239	AD[31:0] hold time from LCLK high	0		ns	6-2	
t240	DBCOE[1:0]#, MDOE#, HDOE# valid delay from CLK/ LCLK high	2	15	ns	6-3	
t241	DLE[1:0]# valid delay from CLK\LCLK high	2	15	ns	6-3	
t242	MDLE# valid delay from CLK\LCLK high		15	ns	6-3	
t243	LA[23:9] valid delay from CLK high (2nd or 3rd T2)	2	25	ns	6-3	
t244	HREQ setup time to CLK high	2		ns	6-1	
t245	HOLD valid delay from CLK high	5	15	ns	6-3	
t246	HLDA setup time to CLK high	2		ns	6-2	
t247	HLDA hold time from CLK high	1		ns	6-2	

6.6 82C568 AC Characteristics (66MHz - Preliminary)

Sym	Parameter	Min	Max	Unit	Figure	Condition
t301	FRAME#, TRDY#, IRDY#, STOP#, DEVSEL#, PAR, SERR#, PERR# valid delay from LCLK rising	2	11	ns	6-3	
t302	PGNT[2:0]# valid delay from LCLK rising	2	12	ns	6-3	
t303	PIRQ[3:0]# valid delay from LCLK rising	2	16	ns	6-3	
t304	MMD[31:0] valid delay from LCLK rising	2	20	ns	6-3	
t305	C/BE[3:0]#, AD[31:0], FRAME#, IRDY#, TRDY#, STOP#, DEVSEL#, PLOCK#, PAR, SERR#, PERR# setup time to LCLK rising	7		ns	6-1	
t306	C/BE[3:0]#, AD[31:0], FRAME#, IRDY#, TRDY#, STOP#, DEVSEL#, PLOCK#, PAR, SERR#, PERR# hold time from LCLK rising	0		ns	6-2	
t307	PREQ[2:0]# setup time to LCLK rising	12		ns		
t308	PREQ[2:0]# hold time from LCLK rising	0		ns	6-2	
t309	PIRQ[3:0]# setup time to LCLK rising	5		ns	6-1	
t310	PIRQ[3:0]# hold time from LCLK rising	3		ns	6-2	
t311	PPWRL# valid delay from ATCLK falling	5	15	ns		
t312	MMD[31:0] setup time to MDLE# low	5		ns		
t313	MMD[31:0] hold time from MDLE# high	5		ns		

6.7 AC Timing Diagrams

Figure 6-1 Setup Timing Waveform

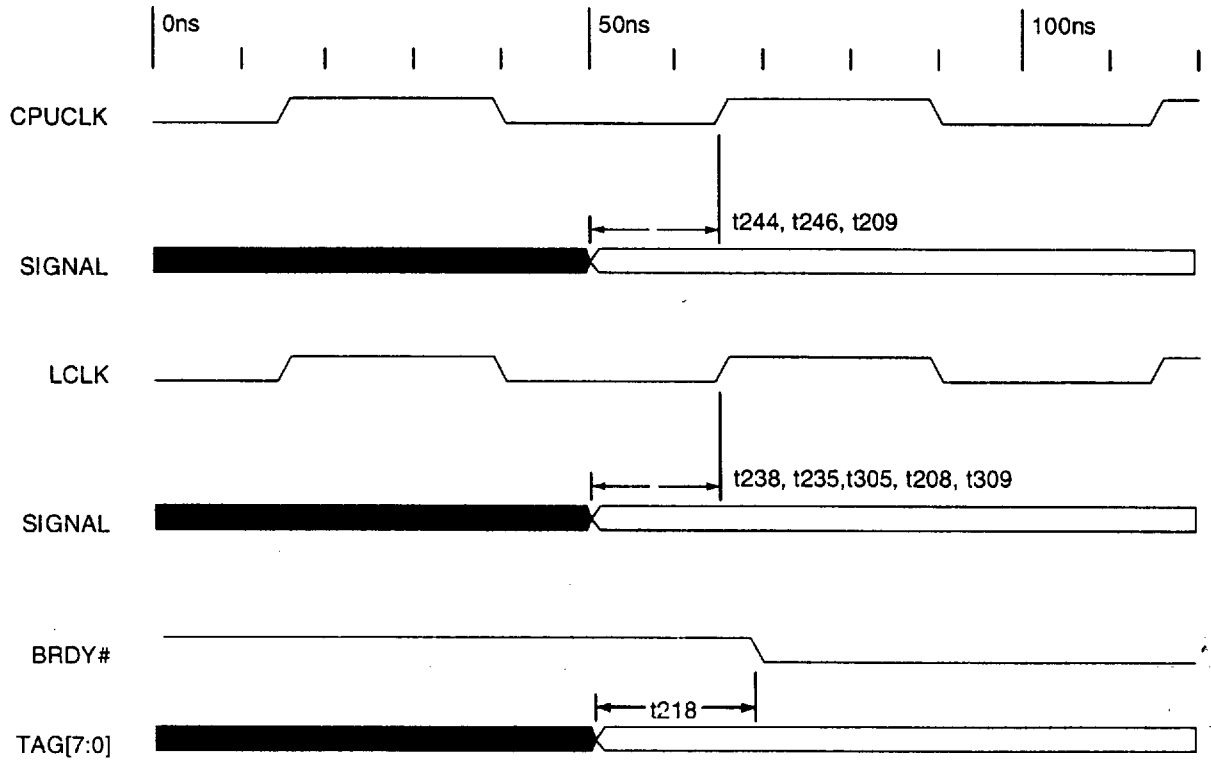


Figure 6-2 Hold Timing Waveform

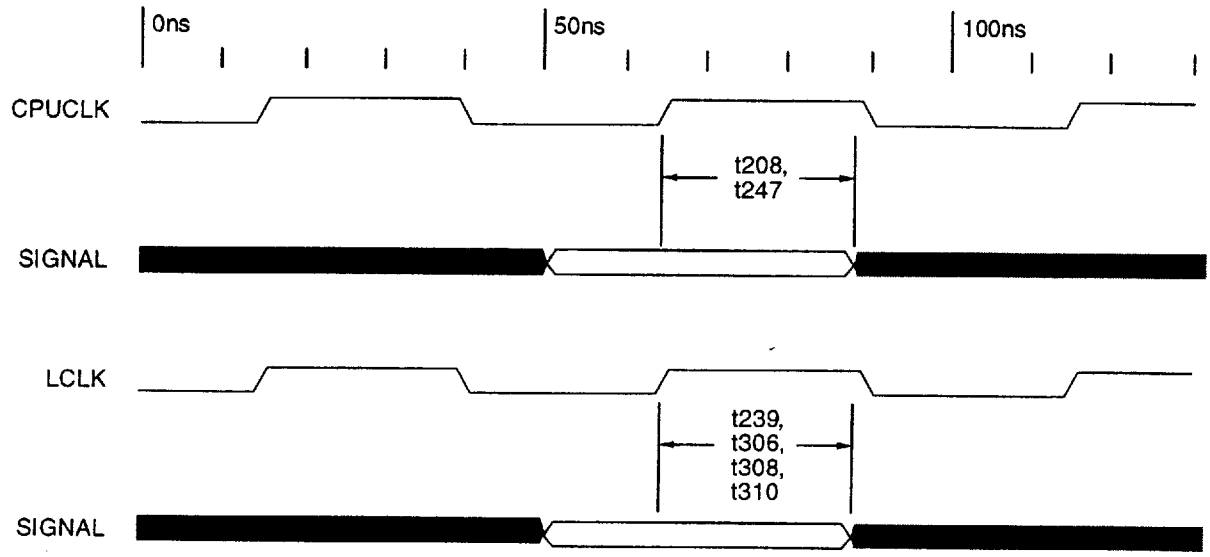
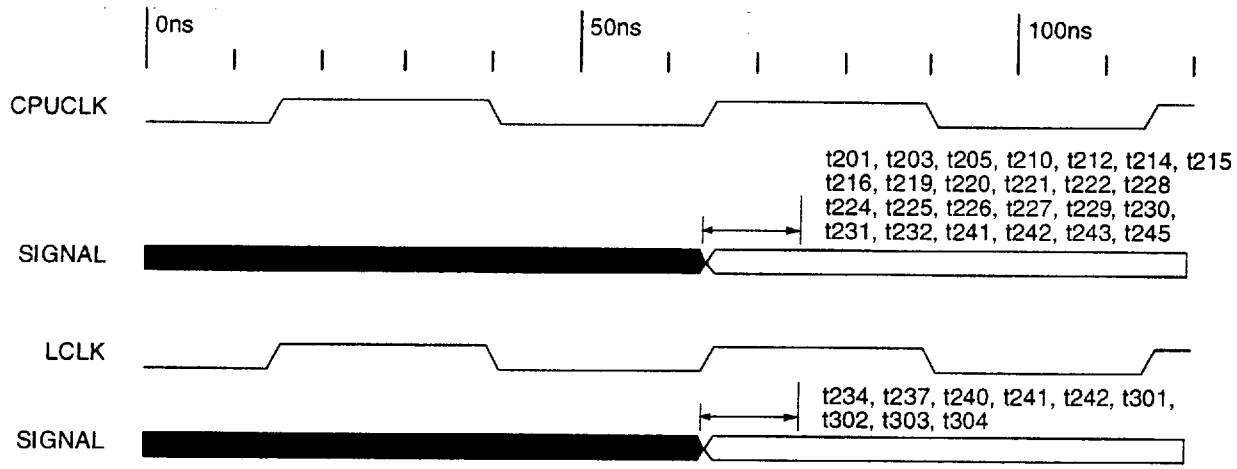


Figure 6-3 Output Delay Timing Waveform



7.0 Test Mode Information

Each device in the Viper-MAX Chipset can be forced into various test modes for board-level testing automatic test equipment (ATE).

- 82C566
  - Test Mode 0: All outputs and bidirectional pins are tristated.
  - Test Mode 1 (NAND tree test): All bidirectionals are tristated and the end of the input and bidirectional NAND chain is present on pin 19.
- 82C567
  - Test Mode 0: All outputs and bidirectional pins are tristated.
  - Test Mode 1 (NAND tree test): All bidirectionals are tristated and the end of the input and bidirectional NAND chain is present on pin 127.
- 82C568
  - Test Mode 0: All outputs and bidirectional pins are tristated.
  - Test Mode 1 (NAND tree test): All bidirectionals are tristated and the end of the input and bidirectional NAND chain is present on pin 92.
  - Test Mode 2: All even numbered output pins are driven high and all odd numbered output pins are driven low.
  - Test Mode 3: All even numbered output pins are driven low and all odd numbered output pins are driven high.

The following subsections explain how to enable the devices in the Viper-MAX Chipset into the above mentioned test modes.

7.1 82C566 Testability

When the 82C566 decodes the RESET combination, it generates an internal reset signal and straps in the value of the MMD[2:1] lines. The RESET combination is defined by the following condition being valid for more than ten clocks:

- MMDOE#, HDOE#, MDOE#, DLE1#, and DLE0# = 11111, DBCTL0# and DBCTL1# = 00, and DBCOE# = 0

If MMD1 is sampled low at this point of the RESET combination, then the 82C566 enters the Testing Mode:

- 0 = Testing Mode enable
- 1 = Testing Mode disable

Once the Testing Mode is enabled, the strap on MMD2 selects which mode is enabled:

- 0 = Test Mode 1 enable (NAND tree test)
- 1 = Test Mode 0 enable (i.e., all outputs and bidirectional pins are tristated)

7.1.1 82C566 NAND Tree Test (Test Mode 1)

The NAND tree mode is used to test input and bidirectional pins which will be part of the NAND tree chain. The NAND tree chain starts at pin 20 and the output of the chain is pin 19. Table 7-1 gives the pins of the NAND tree chain.

7.2 82C567 Testability

The 82C567 samples all its strap information during RESET. If the HREQ pin is sampled low during RESET, the 82C567 enters the test mode. Strap information is sampled 4096 LCLKs after the rising edge of PWRGD.

- HREQ = 0 at the rising edge of RESET: Testing Mode enable
- HREQ = 1 at the rising edge of RESET: Testing Mode disable

If the Testing Mode is enabled, then the strap information on the MASTER#, MSGS2N, and AEN lines selects which test mode is enabled:

- MASTER# = 0, AEN = 0, and MSGS2N = 1: Test Mode 0 is enabled (i.e., all outputs and bidirectional pins are tristated)
- MASTER# = 0, AEN = 1, and MSGS2N = 1: Test Mode 1 (NAND tree test) is enabled (i.e., all bidirectionals are tristated and the end of the input and bidirectional NAND chain is present on pin 123).

7.2.1 82C567 NAND Tree Test (Test Mode 1)

The NAND tree test is used to test input and bidirectional pins which will be part of the NAND tree chain. The NAND tree chain starts at pin 124 and the output of the chain is pin 123. Table 7-2 gives the pins of the NAND tree chain.

During NAND tree testing, the following recommendations regarding pin conditions should be met:

- LCLK and CLK should toggle as clocks to enable the strap information on the MASTER#, MSGS2N, and AEN lines. After the 82C567 is strapped into the NAND tree test mode, LCLK and CLK become regular inputs which should toggle only when they are needed as a part of the NAND-chain testing.
- PWRGD is also a part of the NAND-chain, however, it should not be toggled during NAND-chain testing.

The following is a NAND-chain test example:

	Pin 51 CACHE# = 1;	Pin 55 OCDOE# = 0;
	Pin 55 OCDOE# = 1;	Pin 57 OCAWE# = 0;
*	Pin 57 OCAWE# = 1;	Pin 70 PWRGD = 1;
*	Pin 70 PWRGD = 1;	Pin 100 USBCLK = 0;
	Pin 100 USBCLK = 1;	Pin 101 LA23 = 0;

*PWRGD should not toggle

Table 7-1 82C566 NAND Tree Test Mode Pins

Pin #	Remarks
20	NAND Tree Input Start
22	NAND Tree Input
23	NAND Tree Input
24	NAND Tree Input
25	NAND Tree Input
26	NAND Tree Input
27	NAND Tree Input
28	NAND Tree Input
29	NAND Tree Input
30	NAND Tree Input
31	NAND Tree Input
32	NAND Tree Input
34	NAND Tree Input
36	NAND Tree Input
37	NAND Tree Input
38	NAND Tree Input
39	NAND Tree Input
40	NAND Tree Input
41	NAND Tree Input
42	NAND Tree Input
43	NAND Tree Input
44	NAND Tree Input
45	NAND Tree Input
46	NAND Tree Input
47	NAND Tree Input
48	NAND Tree Input
49	NAND Tree Input
50	NAND Tree Input
51	NAND Tree Input
54	NAND Tree Input
55	NAND Tree Input
56	NAND Tree Input
57	NAND Tree Input
58	NAND Tree Input
59	NAND Tree Input
60	NAND Tree Input
61	NAND Tree Input
62	NAND Tree Input
64	NAND Tree Input
65	NAND Tree Input
67	NAND Tree Input
68	NAND Tree Input
69	NAND Tree Input
70	NAND Tree Input
71	NAND Tree Input

Pin #	Remarks
72	NAND Tree Input
74	NAND Tree Input
75	NAND Tree Input
76	NAND Tree Input
77	NAND Tree Input
78	NAND Tree Input
79	NAND Tree Input
81	NAND Tree Input
82	NAND Tree Input
83	NAND Tree Input
84	NAND Tree Input
85	NAND Tree Input
86	NAND Tree Input
87	NAND Tree Input
88	NAND Tree Input
89	NAND Tree Input
90	NAND Tree Input
93	NAND Tree Input
94	NAND Tree Input
95	NAND Tree Input
96	NAND Tree Input
97	NAND Tree Input
98	NAND Tree Input
99	NAND Tree Input
100	NAND Tree Input
101	NAND Tree Input
102	NAND Tree Input
103	NAND Tree Input
106	NAND Tree Input
107	NAND Tree Input
108	NAND Tree Input
109	NAND Tree Input
110	NAND Tree Input
111	NAND Tree Input
112	NAND Tree Input
113	NAND Tree Input
114	NAND Tree Input
115	NAND Tree Input
117	NAND Tree Input
118	NAND Tree Input
119	NAND Tree Input
121	NAND Tree Input
122	NAND Tree Input
123	NAND Tree Input
124	NAND Tree Input

Pin #	Remarks
125	NAND Tree Input
126	NAND Tree Input
127	NAND Tree Input
128	NAND Tree Input
130	NAND Tree Input
131	NAND Tree Input
132	NAND Tree Input
133	NAND Tree Input
134	NAND Tree Input
135	NAND Tree Input
136	NAND Tree Input
137	NAND Tree Input
138	NAND Tree Input
139	NAND Tree Input
140	NAND Tree Input
141	NAND Tree Input
143	NAND Tree Input
145	NAND Tree Input
146	NAND Tree Input
147	NAND Tree Input
148	NAND Tree Input
149	NAND Tree Input
150	NAND Tree Input
151	NAND Tree Input
152	NAND Tree Input
153	NAND Tree Input
154	NAND Tree Input
155	NAND Tree Input
158	NAND Tree Input
159	NAND Tree Input
160	NAND Tree Input
161	NAND Tree Input
162	NAND Tree Input
163	NAND Tree Input
164	NAND Tree Input
165	NAND Tree Input
166	NAND Tree Input
167	NAND Tree Input
168	NAND Tree Input
170	NAND Tree Input
172	NAND Tree Input
173	NAND Tree Input
174	NAND Tree Input
175	NAND Tree Input
176	NAND Tree Input

Pin #	Remarks
177	NAND Tree Input
178	NAND Tree Input
179	NAND Tree Input
180	NAND Tree Input
181	NAND Tree Input
183	NAND Tree Input
184	NAND Tree Input
185	NAND Tree Input
186	NAND Tree Input
187	NAND Tree Input
189	NAND Tree Input
190	NAND Tree Input
191	NAND Tree Input
192	NAND Tree Input
193	NAND Tree Input
194	NAND Tree Input
196	NAND Tree Input
197	NAND Tree Input
198	NAND Tree Input
199	NAND Tree Input
200	NAND Tree Input
201	NAND Tree Input
203	NAND Tree Input
204	NAND Tree Input
205	NAND Tree Input
206	NAND Tree Input
207	NAND Tree Input
2	NAND Tree Input
3	NAND Tree Input
4	NAND Tree Input
5	NAND Tree Input
6	NAND Tree Input
7	NAND Tree Input
8	NAND Tree Input
9	NAND Tree Input
10	NAND Tree Input
12	NAND Tree Input
13	NAND Tree Input
14	NAND Tree Input
15	NAND Tree Input
16	NAND Tree Input
18	NAND Tree Input
19	NAND Tree Output

Table 7-2 82C567 NAND Tree Test Mode Pins

Pin #	Remarks
124	NAND Tree Input Start
125	NAND Tree Input
126	NAND Tree Input
127	NAND Tree Input
128	NAND Tree Input
131	NAND Tree Input
132	NAND Tree Input
133	NAND Tree Input
134	NAND Tree Input
135	NAND Tree Input
136	NAND Tree Input
137	NAND Tree Input
138	NAND Tree Input
139	NAND Tree Input
140	NAND Tree Input
141	NAND Tree Input
143	NAND Tree Input
144	NAND Tree Input
145	NAND Tree Input
146	NAND Tree Input
147	NAND Tree Input
148	NAND Tree Input
149	NAND Tree Input
150	NAND Tree Input
151	NAND Tree Input
152	NAND Tree Input
153	NAND Tree Input
154	NAND Tree Input
155	NAND Tree Input
158	NAND Tree Input
159	NAND Tree Input
160	NAND Tree Input
161	NAND Tree Input
162	NAND Tree Input
163	NAND Tree Input
164	NAND Tree Input
166	NAND Tree Input

Pin #	Remarks
167	NAND Tree Input
168	NAND Tree Input
169	NAND Tree Input
170	NAND Tree Input
171	NAND Tree Input
172	NAND Tree Input
173	NAND Tree Input
175	NAND Tree Input
178	NAND Tree Input
179	NAND Tree Input
180	NAND Tree Input
181	NAND Tree Input
190	NAND Tree Input
191	NAND Tree Input
192	NAND Tree Input
193	NAND Tree Input
194	NAND Tree Input
195	NAND Tree Input
196	NAND Tree Input
197	NAND Tree Input
198	NAND Tree Input
205	NAND Tree Input
206	NAND Tree Input
207	NAND Tree Input
2	NAND Tree Input
9	NAND Tree Input
10	NAND Tree Input
11	NAND Tree Input
12	NAND Tree Input
13	NAND Tree Input
14	NAND Tree Input
16	NAND Tree Input
17	NAND Tree Input
18	NAND Tree Input
19	NAND Tree Input
20	NAND Tree Input

Pin #	Remarks
21	NAND Tree Input
22	NAND Tree Input
23	NAND Tree Input
24	NAND Tree Input
25	NAND Tree Input
26	NAND Tree Input
27	NAND Tree Input
28	NAND Tree Input
29	NAND Tree Input
30	NAND Tree Input
31	NAND Tree Input
32	NAND Tree Input
33	NAND Tree Input
34	NAND Tree Input
35	NAND Tree Input
36	NAND Tree Input
37	NAND Tree Input
39	NAND Tree Input
40	NAND Tree Input
41	NAND Tree Input
42	NAND Tree Input
43	NAND Tree Input
44	NAND Tree Input
45	NAND Tree Input
46	NAND Tree Input
47	NAND Tree Input
48	NAND Tree Input
49	NAND Tree Input
50	NAND Tree Input
51	NAND Tree Input
55	NAND Tree Input
57	NAND Tree Input
70	NAND Tree Input
100	NAND Tree Input
101	NAND Tree Input
102	NAND Tree Input
103	NAND Tree Input

Pin #	Remarks
106	NAND Tree Input
107	NAND Tree Input
108	NAND Tree Input
109	NAND Tree Input
110	NAND Tree Input
111	NAND Tree Input
112	NAND Tree Input
113	NAND Tree Input
114	NAND Tree Input
115	NAND Tree Input
116	NAND Tree Input
117	NAND Tree Input
121	NAND Tree Input
122	NAND Tree Input
123	NAND Tree Output

7.3 82C568 Testability

The 82C568 samples all its strap information on the falling edge of RESET. If the HREQ pin is sampled low on the falling edge of RESET, the 82C568 enters the Testing Mode.

- HREQ = 0 at the falling edge of RESET:
  Testing Mode enable
- HREQ = 1 at the falling edge of RESET:
  Testing Mode disable

If the Testing Mode is enabled, then the strap information on the IRQ1 and IRQ3 lines selects which test mode is enabled:

- IRQ1 = 0 and IRQ3 = 0:
  Test Mode 0 enable (i.e., all outputs and bidirectional pins are tristated)
- IRQ1 = 1 and IRQ3 = 0:
  Test Mode 1 (NAND tree test) enable (i.e., all bidirectionals are tristated and the end of the input and bidirectional NAND chain is present on pin 92).
- IRQ = 0 and IRQ3 = 1:
  Test Mode 2 enable (i.e., all even numbered output pins are driven high and all odd numbered output pins are driven low).
- IRQ = 1 and IRQ3 = 1:
  Test Mode 3 enable (i.e., all even numbered output pins are driven low and all odd numbered output pins are driven high).

7.3.1 82C568 NAND Tree Test (Test Mode 1)

The NAND tree test is used to test input and bidirectional pins which will be part of the NAND tree chain. The NAND tree chain starts at pin 5 and the output of the chain is pin 92. Table 7-4 gives the pins of the NAND tree chain.

During NAND tree testing, the following recommendations should be met:

- Four clock transitions of LCLK and OSC clock before, during, and after RESET condition must occur before the start of NAND tree test.
- No LCLK or OSC transition must occur during RESET edge transitions.
- Before the start of NAND tree tests, LCLK and OSC clocks must be shut off.
- All inputs and bidirectional pins must be forced to 1.
- Toggle the first input of the NAND tree and work upwards till the end of the tree while monitoring the output of the NAND tree.

7.3.2 Drive High/Drive Low Test (Test Modes 2 and 3)

The 82C568 supports two types of drive high/drive low tests.

If Test Mode 2 is enabled, all even numbered output pins are driven high and all odd numbered output pins are driven low. Table 7-4 shows which pins are driven high/low during this test.

If Test Mode 3 is enabled, all odd numbered output pins are driven high and all even numbered output pins are driven low. Table 7-3 shows which pins are driven high/low during this test.

Table 7-3 Drive High/Drive Low Test Mode Pins

Even Number Pins	Odd Number Pins
4	3
8	9
52	13
74	53
80	81
84	85
92	105
106	107
108	109
110	111
112	113
148	115
150	149
154	

Table 7-4 82C568 NAND Tree Test Mode Pins

Pin#	Remark
5	NAND Tree Input Start
1	NAND Tree Input
6	NAND Tree Input
7	NAND Tree Input
11	NAND Tree Input
12	NAND Tree Input
16	NAND Tree Input
17	NAND Tree Input
18	NAND Tree Input
19	NAND Tree Input
20	NAND Tree Input
21	NAND Tree Input
22	NAND Tree Input
23	NAND Tree Input
24	NAND Tree Input
25	NAND Tree Input
27	NAND Tree Input
28	NAND Tree Input
29	NAND Tree Input
30	NAND Tree Input
31	NAND Tree Input
32	NAND Tree Input
33	NAND Tree Input
34	NAND Tree Input
35	NAND Tree Input
36	NAND Tree Input
39	NAND Tree Input
40	NAND Tree Input
41	NAND Tree Input
42	NAND Tree Input
44	NAND Tree Input
45	NAND Tree Input
46	NAND Tree Input
47	NAND Tree Input
48	NAND Tree Input
49	NAND Tree Input
50	NAND Tree Input
51	NAND Tree Input
52	NAND Tree Input
54	NAND Tree Input

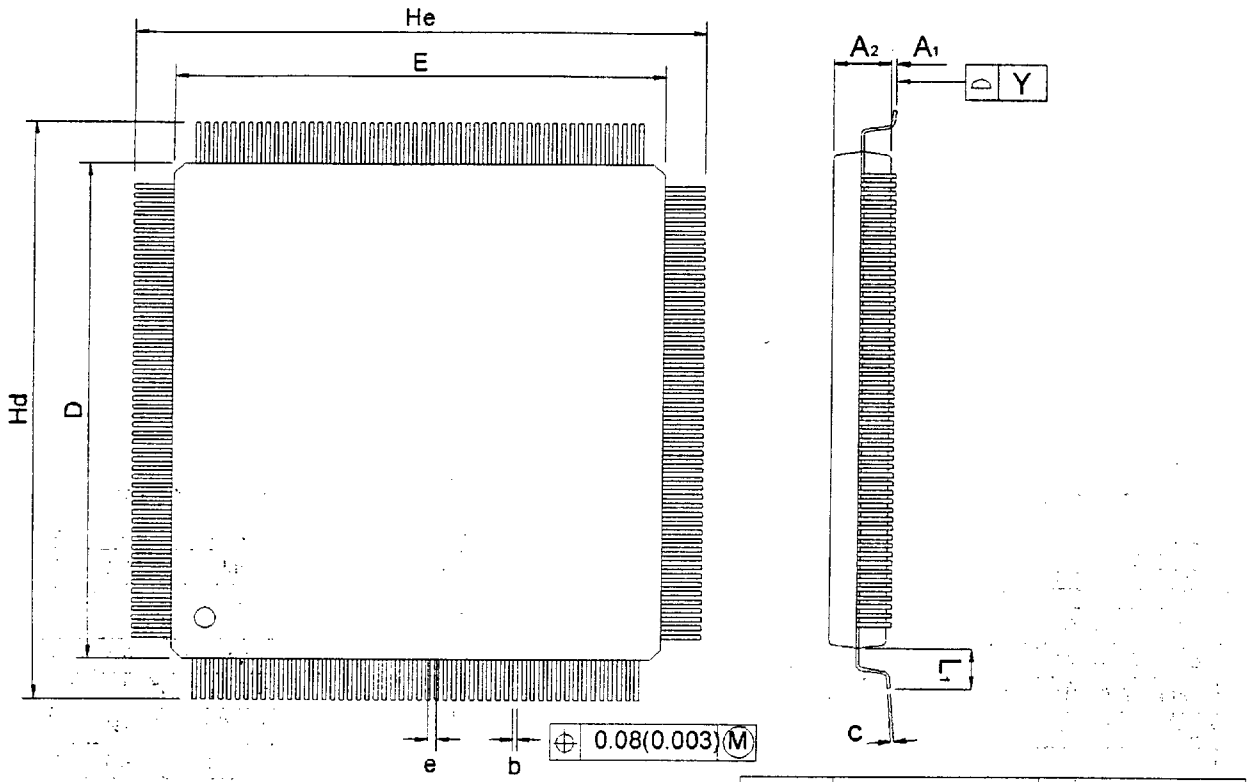
Pin#	Remark
55	NAND Tree Input
56	NAND Tree Input
57	NAND Tree Input
58	NAND Tree Input
59	NAND Tree Input
62	NAND Tree Input
63	NAND Tree Input
64	NAND Tree Input
65	NAND Tree Input
66	NAND Tree Input
67	NAND Tree Input
68	NAND Tree Input
69	NAND Tree Input
70	NAND Tree Input
71	NAND Tree Input
73	NAND Tree Input
75	NAND Tree Input
76	NAND Tree Input
77	NAND Tree Input
78	NAND Tree Input
79	NAND Tree Input
82	NAND Tree Input
83	NAND Tree Input
86	NAND Tree Input
87	NAND Tree Input
88	NAND Tree Input
89	NAND Tree Input
90	NAND Tree Input
91	NAND Tree Input
93	NAND Tree Input
94	NAND Tree Input
95	NAND Tree Input
98	NAND Tree Input
99	NAND Tree Input
100	NAND Tree Input
101	NAND Tree Input
102	NAND Tree Input
103	NAND Tree Input
104	NAND Tree Input
105	NAND Tree Input

Pin#	Remark
116	NAND Tree Input
117	NAND Tree Input
118	NAND Tree Input
120	NAND Tree Input
121	NAND Tree Input
122	NAND Tree Input
123	NAND Tree Input
124	NAND Tree Input
125	NAND Tree Input
126	NAND Tree Input
127	NAND Tree Input
128	NAND Tree Input
129	NAND Tree Input
130	NAND Tree Input
131	NAND Tree Input
132	NAND Tree Input
133	NAND Tree Input
134	NAND Tree Input
135	NAND Tree Input
136	NAND Tree Input
138	NAND Tree Input
139	NAND Tree Input
140	NAND Tree Input
141	NAND Tree Input
143	NAND Tree Input
144	NAND Tree Input
145	NAND Tree Input
146	NAND Tree Input
151	NAND Tree Input
152	NAND Tree Input
153	NAND Tree Input
155	NAND Tree Input
156	NAND Tree Input
157	NAND Tree Input
158	NAND Tree Input
159	NAND Tree Input
160	NAND Tree Input
161	NAND Tree Input
162	NAND Tree Input
163	NAND Tree Input
166	NAND Tree Input

Pin#	Remark
167	NAND Tree Input
168	NAND Tree Input
169	NAND Tree Input
170	NAND Tree Input
171	NAND Tree Input
172	NAND Tree Input
173	NAND Tree Input
174	NAND Tree Input
175	NAND Tree Input
176	NAND Tree Input
177	NAND Tree Input
178	NAND Tree Input
179	NAND Tree Input
181	NAND Tree Input
182	NAND Tree Input
183	NAND Tree Input
184	NAND Tree Input
187	NAND Tree Input
188	NAND Tree Input
189	NAND Tree Input
190	NAND Tree Input
191	NAND Tree Input
192	NAND Tree Input
193	NAND Tree Input
194	NAND Tree Input
195	NAND Tree Input
196	NAND Tree Input
197	NAND Tree Input
198	NAND Tree Input
199	NAND Tree Input
202	NAND Tree Input
203	NAND Tree Input
204	NAND Tree Input
205	NAND Tree Input
206	NAND Tree Input
207	NAND Tree Input
208	NAND Tree Input
2	NAND Tree Input
92	NAND Tree Output

8.0 Mechanical Package Outlines

Figure 8-1 208-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	0.10	0.20	0.30	0.004	0.008	0.012
c	0.10	0.15	0.20	0.004	0.006	0.008
D	27.90	28.00	28.10	1.098	1.102	1.106
E	27.90	28.00	28.10	1.098	1.102	1.106
e		0.50			0.020	
Hd	30.35	30.60	30.85	1.195	1.205	1.215
He	30.35	30.60	30.85	1.195	1.205	1.215
L	0.50	0.60	0.75	0.020	0.024	0.030
L_1		1.30			0.051	
Y			0.08			0.003
θ	0		7	0		7

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