OPTi

82C895

System/Power Management Controller

Data Book

Revision 1.0
912-3000-015
September 1994

Copyright

Disclaimer

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

Trademarks

OPTi and OPTi Inc. are registered trademarks of OPTi Inc.

All other trademarks and copyrights are the property of their respective holders.

OPTi Inc.
2525 Walsh Avenue
Santa Clara, CA 95051
Tel: (408) 980-8178
Fax: (408) 980-8860
BBS: (408) 980-9774

Table of Contents

Table of Contents (Cont.)

Table of Contents (Cont.)

Table of Contents (Cont.)

Table of Contents (Cont.)

List of Figures

List of Figures (Cont.)

List of Tables

List of Tables (Cont.)

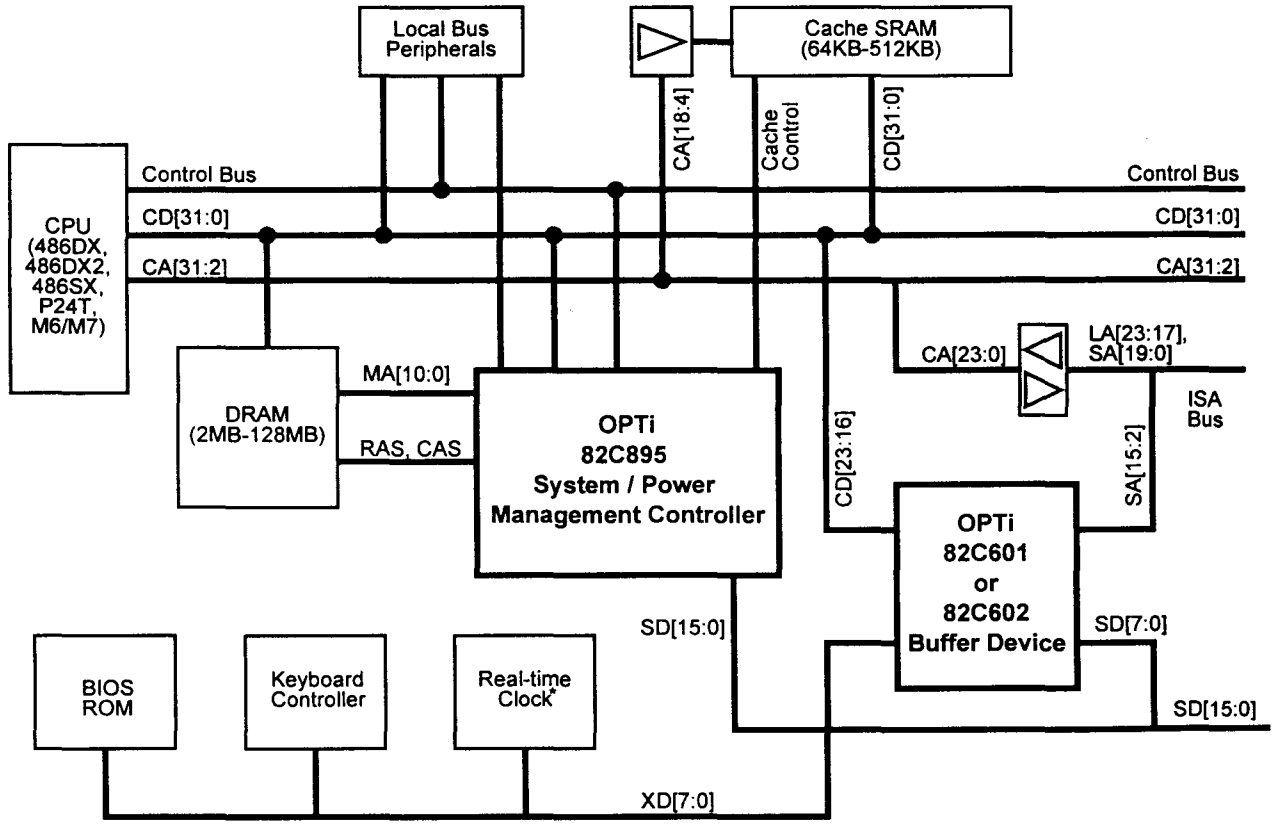
List of Tables (Cont.)

System Power Management Controller

1.0 Features

- Processor interface:
  - Intel® 80486SX, DX, DX2, SLe, DX4, P24T, P24D
  - AMD® 486DX, DX2, DXL, DXL2, Plus
  - Cyrix® DX, DX2, M7
  - CPU frequencies supported 20, 25, 33, 40 and 50MHz
- Cache interface:
  - Direct Mapped Cache
  - Two banks interleaved or single bank non-interleaved
  - 64, 128, 256 and 512K cache sizes
  - Programmable wait states for L2 cache reads and writes
  - 2-1-1-1 read burst and zero wait state write @ 33MHz
  - No Valid bit required
  - Supports CPUs with L1 write-back support
- DRAM interface:
  - Up to 128MB main memory support
  - Supports 256KB, 1MB, 4MB, and 16MB single- and double-sided SIMM modules
  - Read page-hit timing of 3-2-2-2 at 33MHz
  - Supports hidden, slow, and CAS-before-RAS refresh
  - Four RAS lines to support four banks of DRAM
  - Programmable wait states for DRAM reads and writes
  - Enhanced DRAM configuration map
- Power management:
  - Support for SMM (System Management Mode) for system power management implementations
  - Programmable power management
  - Programmable wake-up events through hardware, software and external SMI source
  - Multiple level GREEN support (NESTED_GREEN)
  - STPCLK# protocol support
  - One programmable GREEN event timer
- ISA interface:
  - 100% IBM® PC/AT® ISA compatible
  - Integrates DMA, timer and interrupt controllers
  - Optional PS/2 style IRQ1 and 12 latching

Figure 1-1 82C895 and 82C601/602-Based System Block Diagram



*Included as a part of 82C602.

OPTi Inc. · 2525 Walsh Avenue · Santa Clara, CA 95051 · (408) 980-8178

Features (Cont.)

- VESA VL interface:
  - Conforms to the VESA V2.0 specification
  - Optional support for up to two VL masters
- Miscellaneous features:
  - Full support for shadow RAM, write protection, L1/L2 cacheability for video, adapter and system BIOS
  - Enhanced arbitration scheme
  - Transparent 8042 emulation for fast CPU reset and GATEA20 generation
- Packaging:
  - Higher integration
  - Reduced TTL count
  - Low-power, high-speed 0.8-micron CMOS technology
  - 208-pin PQFP (Plastic Quad Flat Pack)

2.0 Overview

The 82C895 provides a highly integrated solution for fully compatible, high performance PC/AT platforms. This chipset will support 486SX/DX/DX2/DX4 and P24T microprocessors in the most cost effective and power efficient designs available today. For power users, this chipset offers optimum performance for systems running up to 50MHz.

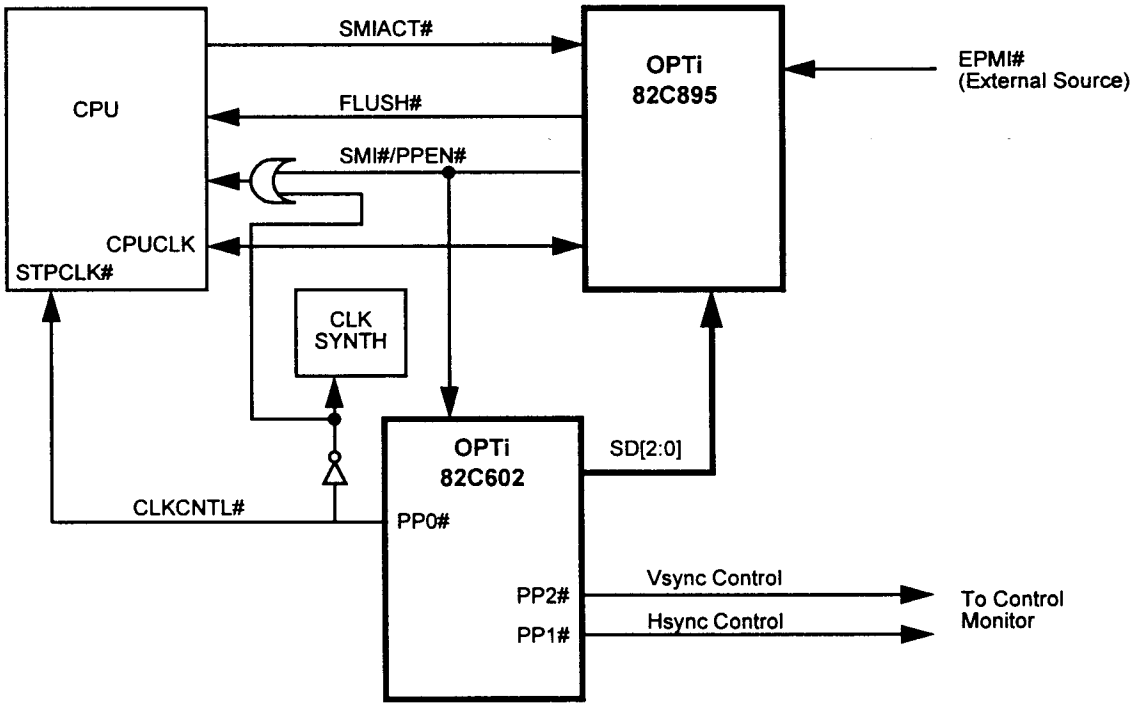
Based fundamentally on OPTi's proven 82C801 and 82C802 design architectures, the 82C895 adds additional memory configurations and extensive power management control for the processor and other motherboard components.

The 82C895 supports the latest write-back processor designs from Intel, AMD, and Cyrix, as well as supporting the AT bus and VESA local bus for compatibility and performance. It also includes an 82C206-compatible Integrated Peripherals Controller (IPC), all in a single 208-pin PQFP (Plastic Quad Flat Pack) package for low cost.

2.1 Power Management

This block diagram exemplifies the flexibility of the 82C895/82C602 GREEN strategy. System designs can easily accommodate both SLe and non-SLe CPUs. If an Intel non-SLe CPU is used, SMI#, SMIACT#, and FLUSH# are no connects. One design can easily accommodate both types of processors with minimal changes for upgrades.

Figure 2-1 Power Management Block Diagram



3.0 Signal Definitions

Figure 3-1 Pin Diagram

Table 3-1 Numerical Pin Cross-Reference List

Pin No.	Pin Name
1	IRQ75
2	IRQ8
3	IRQ9
4	IRQ1110
5	IRQ14
6	IRQ1512
7	RFSH#
8	ALE
9	ATCLK
10	VCC
11	SD0
12	SD1
13	SD2
14	SD3
15	GND
16	DRQ0
17	SD4
18	SD5
19	SD6
20	SD7
21	IORD#
22	IOWR#
23	MEMR#
24	MEMW#
25	MCS16#
26	DRQ1
27	GND
28	CLK1
29	IOCS16#
30	SA0
31	SA1
32	SBHE#
33	SD8
34	SD9
35	SD10
36	SD11
37	GND
38	GND
39	SD12
40	SD13
41	SD14
42	SD15
43	VCC
44	TC
45	LGNT0#
46	AHOLD
47	HITM#/DC#
48	SPKD
49	ROMCS#/KBDCS#
50	0WS#/LRDYI#
51	IOCHRDY
52	A20M#

Pin No.	Pin Name
53	LMGCS#
54	XDIR#
55	HLBLTH#
56	HLBOE1#
57	HLBOE2#
58	CHCK#
59	PWRGD
60	GND
61	VCC
62	EPMI#/MDIR#
63	MP0
64	MP1
65	MP2
66	MP3
67	D0
68	D1
69	D2
70	D3
71	D4
72	D5
73	D6
74	D7
75	D8
76	D9
77	D10
78	GND
79	GND
80	D11
81	D12
82	D13
83	D14
84	D15
85	D24
86	D25
87	D26
88	D27
89	D28
90	D29
91	D30
92	D31
93	DRQ2
94	FERR#
95	IGNNE#
96	GND
97	VCC
98	NMI
99	A26
100	LDEV#
101	DRQ3
102	HOLD
103	HLDA
104	INTR

Pin No.	Pin Name
105	BLAST#
106	EADS#
107	KEN#
108	DC#/TAG7
109	MIO#
110	WR#
111	ADS#
112	BRDY#
113	RDY#
114	VCC
115	CPURST
116	A2
117	A3
118	A4
119	GND
120	GND
121	A5
122	A6
123	A7
124	A8
125	A9
126	A10
127	A11
128	A12
129	A13
130	GND
131	A14
132	A15
133	A16
134	A17
135	A18
136	A19
137	A20
138	A21
139	A22
140	A23
141	GND
142	GND
143	A24
144	A25
145	LREQ0#
146	DRAMS#
147	VCC
148	TAG0
149	TAG1
150	TAG2
151	TAG3
152	TAG4
153	TAG5
154	TAG6
155	DRTY
156	SMIACT#

Pin No.	Pin Name
157	FLUSH#
158	BEA3
159	BEA2OA3
160	SMI#/PPEN#
161	TAGW#
162	BEOE#
163	BOOE#
164	GND
165	VCC
166	ECAWE#
167	OCAWE#
168	BE0#
169	BE1#
170	BE2#
171	BE3#
172	MA0
173	MA1
174	MA2
175	MA3
176	MA4
177	MA5
178	MA6
179	MA7
180	CAS0#
181	CAS1#
182	GND
183	CAS2#
184	CAS3#
185	MA8
186	MA9
187	MA10
188	RAS0#
189	RAS1#
190	RAS2#
191	RAS3#
192	LREQ1#
193	LGNT1#
194	NC
195	MA11
196	DWE#
197	DRQ5
198	DRQ6
199	OSC
200	GND
201	VCC
202	DRQ7
203	DACK0
204	DACK1
205	DACK2
206	IRQ1
207	IRQ43
208	IRQ6

Table 3-2 Alphabetical Pin Cross-Reference List

Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
A2	116	D5	72	GND	182	OCAWE#	167
A3	117	D6	73	GND	200	OSC	199
A4	118	D7	74	HITM#/DC#	47	PWRGD	59
A5	121	D8	75	HLBOE1#	56	RAS0#	188
A6	122	D9	76	HLBOE2#	57	RAS1#	189
A7	123	D10	77	HLBLTH#	55	RAS2#	190
A8	124	D11	80	HLDA	103	RAS3#	191
A9	125	D12	81	HOLD	102	RDY#	113
A10	126	D13	82	IGNNE#	95	RFSH#	7
A11	127	D14	83	INTR	104	ROMCS#/KBDCS#	49
A12	128	D15	84	IOCHRDY	51	SA0	30
A13	129	D24	85	IOCS16#	29	SA1	31
A14	131	D25	86	IORD#	21	SBHE#	32
A15	132	D26	87	IOWR#	22	SD0	11
A16	133	D27	88	IRQ1	206	SD1	12
A17	134	D28	89	IRQ1110	4	SD2	13
A18	135	D29	90	IRQ14	5	SD3	14
A19	136	D30	91	IRQ1512	6	SD4	17
A20	137	D31	92	IRQ43	207	SD5	18
A20M#	52	DACK0	203	IRQ6	208	SD6	19
A21	138	DACK1	204	IRQ75	1	SD7	20
A22	139	DACK2	205	IRQ8	2	SD8	33
A23	140	DC#/TAG7	108	IRQ9	3	SD9	34
A24	143	DRAMS#	146	KEN#	107	SD10	35
A25	144	DRQ0	16	LDEV#	100	SD11	36
A26	99	DRQ1	26	LGNT0#	45	SD12	39
ADS#	111	DRQ2	93	LGNT1#	193	SD13	40
AHOLD	46	DRQ3	101	LMGCS#	53	SD14	41
ALE	8	DRQ5	197	LREQ0#	145	SD15	42
ATCLK	9	DRQ6	198	LREQ1#	192	SMIACT#	156
BEA2OA3	159	DRQ7	202	MA0	172	SMI#/PPEN#	160
BEA3	158	DRTY	155	MA1	173	SPKD	48
BEOE#	162	DWE#	196	MA2	174	TAG0	148
BE0#	168	EADS#	106	MA3	175	TAG1	149
BE1#	169	ECAWE#	166	MA4	176	TAG2	150
BE2#	170	EPMI#/MDIR#	62	MA5	177	TAG3	151
BE3#	171	FERR#	94	MA6	178	TAG4	152
BLAST#	105	FLUSH#	157	MA7	179	TAG5	153
BOOE#	163	GND	15	MA8	185	TAG6	154
BRDY#	112	GND	27	MA9	186	TAGW#	161
CAS0#	180	GND	37	MA10	187	TC	44
CAS1#	181	GND	38	MA11	195	VCC	10
CAS2#	183	GND	60	MCS16#	25	VCC	43
CAS3#	184	GND	78	MEMR#	23	VCC	61
CHCK#	58	GND	79	MEMW#	24	VCC	97
CLK1	28	GND	96	MIO#	109	VCC	114
CPURST	115	GND	119	MP0	63	VCC	147
D0	67	GND	120	MP1	64	VCC	165
D1	68	GND	130	MP2	65	VCC	201
D2	69	GND	141	MP3	66	WR#	110
D3	70	GND	142	NC	194	XDIR#	54
D4	71	GND	164	NMI	98	0WS#/LRDY#	50

3.1 Signal Descriptions

3.1.1 CPU Interface Signals

Signal Name	Pin No.	Signal Type	Signal Description
A[26:24]	99, 144, 143	I	CPU address lines 26, 25, and 24.
A[23:17]	140:134	I/O	CPU address lines 23 through 17: These signals are inputs during CPU, refresh, and master cycles and are outputs during DMA cycles.
A[16:8]	133:131, 129:124	I/O	CPU address lines 16 through 8: These signals are inputs during non-DMA cycles. A[16:9] become outputs which transmit DMA address lines A[16:9] by latching SD[7:0] during 16-bit DMA cycles. A[15:8] transmit DMA address lines A[15:8] by latching SD[7:0] during 8-bit DMA cycles.
A[7:2]	123:121, 118:116	I/O	CPU address lines 7 through 2: These signals are outputs during DMA cycles.
DRAMS#	146	I	DRAM controller upper address decode input. All the CPU address lines are not decoded by the 82C895. An external decoder or an upper address line should be connected to this input.
D[31:24], D[15:0]	92:85, 84:80, 77:67	I/O	CPU data bus bits 31 through 24 and 15 through 0.
BE[3:0]#	171:168	I/O	The byte enable signals indicate active bytes during read and write cycles.
MIO#	109	I/O	Memory or I/O cycle definition pin: When MIO# is high, it indicates a memory cycle and, if low, an I/O cycle. MIO# becomes an output during master and DMA cycles for local device accesses.
DC#/TAG7	108	I/O	CPU data / code cycle status or Tag bit 7: As DC#, this pin is used to indicate data transfer operations when high, or control operations (code fetch, halt, etc.) when low. As TAG7, this pin is used to expand the cacheable address range of the DRAM. When MP2 is sampled high during reset, this pin operates as DC#. When MP2 is sampled low during reset, it operates as TAG7. TAG7's functionality may be ignored by setting Index Register 20h, bit 4. **MP2 Function** 1 D/C# 0 TAG7
HITM#/DC#	47	I	L1 write-back hit or DC#: This pin is an active low input from an L1 write-back capable CPU (such as the P24T) used to indicate that the current cache inquiry address has been found in the internal cache and that dirty data exists in that cache line. This pin is either HITM#, or both HITM# and DC#. This pin can be in either configuration by the sampling of MP2 during reset. When strapped low, it functions as HITM# during inquiry cycles and DC# for all other cycles. In this configuration, the DC# and HITM# signals from the CPU must be ANDed into this pin. The strapping option also affects Pin 108. **MP2 Function** 1 HITM# 0 HITM# and DC#

Signal Descriptions (Cont.)

Signal Name	Pin No.	Signal Type	Signal Description
WR#	110	I/O	Write or read cycle definition pin: If WR# is high, it indicates a write cycle and, if low, a read cycle. WR# becomes an output during master and DMA cycles for local device accesses.
ADS#	111	I/O	Address status input indicates that a valid bus cycle definition and addresses are available on the cycle definition pins and address bus. It becomes an output during master or DMA cycles to the local bus.
RDY#	113	I/O	RDY# indicates that the current non-bus cycle is complete. RDY# becomes an input during local device cycles.
BRDY#	112	I/O	BRDY# indicates the completion of a bus cycle.
BLAST#	105	I	BLAST# indicates the end of a CPU burst cycle.
EADS#	106	O	External address: EADS#, when asserted, indicates that an external address has been driven onto the CPU address lines. This address is used to perform an internal cache snoop cycle.
AHOLD	46	O	Address Hold: This pin is driven to force the CPU to float address lines A[31:2] on the next clock cycle.
A20M#	52	O	Address bit 20 mask: A20M# is asserted to force the CPU for real mode operation. Up reset, this pin is driven high and can be asserted by writing to Port 92h or keyboard registers.

3.1.2 VESA Local Bus Interface Signals

Signal Name	Pin No.	Signal Type	Signal Description
LDEV#	100	I	Local bus device cycle: LDEV# indicates that a local bus device has captured the current cycle. This signal is sampled by the 82C895 at the end of the first T2 or at the end of the second T2 at 50MHz.
LREQ0#	145	I	Local bus request 0 input from the VESA local bus master.
LGNT0#	45	O	Local bus grant 0 output to the VESA local bus master.
LREQ1#	192	I	Local bus request 1 input from the VESA local bus master. MP1 must be pulled down with a 1K resistor (refer to MP[3:0] signal description).
LGNT1#	193	O	Local bus grant 1 output to the VESA local bus master. MP1 must be pulled down with a 1K resistor (refer to MP[3:0] signal description). **MP1 Pin 192 Pin 193** 0 LREQ1# LGNT1# 1 Reserved Reserved Function of pins 192 and 193 are dependent on MP1 sampling during reset.

Signal Descriptions (Cont.)

3.1.3 AT Bus Interface Signals

Signal Name	Pin No.	Signal Type	Signal Description
SA[1:0]	31:30	I/O	ISA bus address lines 1 and 0: These pins are inputs during master cycles and outputs during CPU, DMA, or refresh cycles.
SD[15:0]	42:39, 36:33, 20:17, 14:11	I/O	ISA bus data lines 15 through 0.
ALE	8	O	ISA bus address latch enable: When asserted, ALE indicates that the SBHE#, SA and LA lines are valid on the ISA bus.
SBHE#	32	I/O	ISA bus byte high enable: When asserted, SBHE# indicates that a byte is being transferred on SD[15:8] of the ISA data bus. During master cycles this pin is an input, otherwise it is always an output from the 82C895.
IORD#	21	I/O	ISA I/O read command: This pin is an input during master cycles and an output during CPU and DMA cycles.
IOWR#	22	I/O	ISA I/O write command: This pin is an input during master cycles and an output during CPU and DMA cycles.
MEMR#	23	I/O	ISA memory read command: This pin is an input during master cycles and an output during CPU and DMA cycles.
MEMW#	24	I/O	ISA memory write command: This pin is an input during master cycles and an output during CPU and DMA cycles.
LMGCS#	53	O	Memory space below one megabyte indicator: This signal is asserted during ISA refresh or when the address lines 20 to 26 and DRAMS# are low. This signal should be wired OR'd externally with MEMR# and MEMW# to generate SMEMR# and SMEMW#, respectively.
MEMCS16#	25	I/O	ISA 16-bit memory chip select: This is driven by an ISA slave to indicate that it is a 16-bit memory device. This is driven low during master cycles.
IOCS16#	29	I	ISA 16-bit I/O chip select: This is driven by an ISA slave to indicate that it supports 16-bit I/O bus cycles.
IOCHRDY	51	I/O	I/O channel ready: This input is from the ISA bus indicating that additional time is required to complete the current ISA cycle.
0WS#/LRDYI#	50	I	Zero wait state input from ISA bus or local ready: This pin is either 0WS# or LRDYI# and is determined by a strapping option of MP0. 0WS is asserted by the ISA slave to indicate that the system controller can shorten the current ISA cycle. If configured as LRDYI#, the VESA local by LRDY# signal should be connected to this input. Consequently, the 82C802G will assert CPURDY# in response to the LRDY# from the VL device. MP0 is sampled during reset to determine the function of this pin. **MP0 Pin 50** 0 LRDYI# 1 0WS#

Signal Descriptions (Cont.)

Signal Name	Pin No.	Signal Type	Signal Description
ROMCS#/ KBDCS#	49	O	BIOS ROM output enable: During memory cycles, this signal is used for system BIOS ROM accesses and can be either 8- or 16-bit. It will be asserted from the end of the first T2 to the end of the last T2. Keyboard Controller Chip Select: This is also driven during I/O accesses to port 60h or 64h and can be connected to the keyboard controller chip select.

3.1.4 Bus Arbitration Interface Signals

Signal Name	Pin No.	Signal Type	Signal Description
RFSH#	7	I/O	Refresh: When RFSH# is active, it indicates that a refresh cycle is in progress. As an input, this signal is driven by the ISA bus masters to initiate refresh cycles.
HOLD	102	O	Hold: This signal is driven to the CPU to request the CPU bus.
HLDA	103	I	Hold acknowledge must be driven by the CPU to grant the CPU bus to ISA or VL devices.

3.1.5 Numeric Processor Interface Signals

Signal Name	Pin No.	Signal Type	Signal Description
FERR#	94	I	Numeric coprocessor error: FERR# is driven by the CPU when a floating point error occurs. This active low signal is used to generate IGNNE# for the 486 CPU.
IGNNE#	95	O	Ignore numeric coprocessor error: This signal goes active once FERR# is active. An I/O write to Port F0h or a CPU reset will force this signal inactive.

3.1.6 Cache Interface Signals

Signal Name	Pin No.	Signal Type	Signal Description
KEN#	107	O	Cacheable or non-cacheable status for the 486 CPU's internal cache: KEN# is asserted if the current cycle is cacheable in the CPU internal cache.
FLUSH#	157	O	Flush CPU internal cache: FLUSH# is driven active before an SMM occurs and during a wake up from SMI_GREEN mode to flush the CPU internal cache.
BEA3	158	O	Cache address line A3: In the single bank cache, this pin is connected to the A3 line of the cache RAM. In the double-bank interleaved cache, it is connected to the even bank A3 line of the cache RAM.
BEA2OA3	159	O	Cache address line A2/A3: In the single bank cache, this pin is connected to the A2 line of the cache RAM. In the double-bank interleaved cache, it is connected to the odd bank A3 line of the cache RAM.
BEOE#	162	O	Even bank cache output enable.
BOOE#	163	O	Odd bank cache output enable.
ECAWE#	166	O	Even bank cache write enable.

Signal Descriptions (Cont.)

Signal Name	Pin No.	Signal Type	Signal Description
OCAWE#	167	O	Odd bank cache write enable.
TAG[6:0]	154:148	I/O	Cache tag: Tag 0 through 6 connected to the Tag SRAM data bus.
TAGW#	161	O	Tag write enable: TAGW# is asserted during CPU read miss cycles and during write hit cycles when the dirty bit is updated.
DIRTY	155	I/O	Dirty bit of tag RAM: Indicates that the current line is modified.

3.1.7 DRAM Interface Signals

Signal Name	Pin No.	Signal Type	Signal Description
MA[11:0]	195, 187:185, 179:172	O	DRAM address: MA[10:0] provides row and column addresses to the DRAMs.
RAS[3:0]#	191:188	O	DRAM row address strobe: These signals are used to latch the row addresses on the MA[11:0] bus into the DRAMs.
CAS[3:0]#	184:183, 181:180	O	DRAM column address strobe: These signals are used to latch the column addresses on the MA[10:0] bus into the DRAMs.
DWE#	196	O	DRAM write enable signal.
MP[3:0]	66:63	I/O	DRAM parity Bits 3 through 0. In addition, these MP lines are used for power on strapping options. MP lines are sampled at the rising edge of reset and they must be pulled down with a 1Kohm resistor to detect a "0" during reset. **MP0** **Pin 50** 0 LRDYI# 1 0WS# **MP1** **Pin 192** **Pin 193** 0 LREQ1# LGNT1# 1 Reserved Reserved **MP2** **Pin 108** **Pin 47** 0 TAG7 HITM# and DC# 1 D/C# HITM# **MP3** **Pin 62** 0 MDIR# 1 EPMI#

3.1.8 DMA and Interrupt Controller Interface Signals

Signal Name	Pin No.	Signal Type	Signal Description
DRQ[7:5], DRQ[3:0]	202, 198, 197, 101, 93, 26, 16	I	DMA request lines 7 through 5 and 3 through 0

Signal Descriptions (Cont.)

Signal Name	Pin No.	Signal Type	Signal Description
DACK[2:0]	205:203	O	Encoded DMA acknowledgment lines 2 through 0. These three lines should be decoded externally using a 3 - 8 decoder to drive the ISA DACK lines. DACK 2 1 0 — ISA DACK lines 0 0 0 — DACK0# 0 0 1 — DACK1# 0 1 0 — DACK2# 0 1 1 — DACK3# 1 0 1 — DACK5# 1 1 0 — DACK6# 1 1 1 — DACK7#
TC	44	O	TC is asserted as a terminal count indicator.
IRQ1	206	I	ISA interrupt request 1.
IRQ6	208	I	ISA interrupt request 6.
IRQ8	2	I	ISA interrupt request 8.
IRQ9	3	I	ISA interrupt request 9.
IRQ14	5	I	ISA interrupt request 14.
IRQ43	207	I	ISA interrupt requests 4 and 3: When ATCLK is low, this pin is IRQ4. When ATCLK is high, this pin is IRQ3.
IRQ75	1	I	ISA interrupt requests 7 and 5: When ATCLK is low, this pin is IRQ7. When ATCLK is high, this pin is IRQ5.
IRQ1110	4	I	ISA interrupt requests 11 and 10: When ATCLK is low, this pin is IRQ11. When ATCLK is high, this pin is IRQ10.
IRQ1512	6	I	ISA interrupt requests 15 and 12: When ATCLK is low, this pin is IRQ15. When ATCLK is high, this pin is IRQ12.
INTR	104	O	Interrupt request to the CPU.

3.1.9 Buffer Control Interface Signals

Signal Name	Pin No.	Signal Type	Signal Description
XDIR#	54	O	SD[7:0] to XD[7:0] direction control. This pin is driven active for all ROM cycles and I/O accesses to Ports 60h, 64h, 70h, and 71h.
HLBLTH#	55	O	Byte 2 data latch enable: This signal becomes high during the CPU AT byte 2 read cycle and during DMA or master cycles.
HLBOE1#	56	O	Byte 2 data buffer output enable: This signal becomes active during CPU DRAM cycles for parity checking and generation, during CPU AT byte 2 write cycle in 486 mode and during DMA or master byte 2 read DRAM or local device cycles.
HLBOE2#	57	O	Byte 2 data latch output enable: This signal becomes active during CPU AT byte 2 read cycle and during DMA or master byte 2 write to local DRAM or local device cycles.

Signal Descriptions (Cont.)

3.1.10 Reset Signals

Signal Name	Pin No.	Signal Type	Signal Description
CPURST	115	O	CPU reset: This reset signal can be connected to CPURST, SRESET, or INIT to the CPU depending on the CPU type. This signal is driven high for 64 clocks after a low-to-high transition on the PWRGD input. Additionally, the signal is also driven during Port 92 and keyboard reset.
PWRGD	59	I	Power good status or reset switch on indication: A high on this pin indicates that VCC from the power supply is stable. When sampled low with CLKI present, all the 82C802G's internal state machines will be reset. A low-to-high transition is used to generate CPURST.

3.1.11 Clock Signals

Signal Name	Pin No.	Signal Type	Signal Description
OSC	199	I	14.31818MHz oscillator input: This is a 14.31818MHz clock input used by the internal 8254 timer and power management unit.
ATCLK	9	O	AT clock (to AT bus): This is a free running clock output, programmable to be CLKI/3, CLKI/4, CLKI/5, CLKI/6 or OSC/2. During the GREEN Mode, the ATCLK output will always be equal to OSC/2.
CLK1	28	I	CLK 1x input: The single-phase clock input provides basic timing and operating frequency for the 82C895. The 82C895 supports 25, 33, 40 and 50MHz operation. This clock should be in phase with CPU clock.

3.1.12 Miscellaneous Signals

Signal Name	Pin No.	Signal Type	Signal Description
CHCK#	58	I	I/O channel check: CHCK# is driven by the ISA bus devices when a parity or uncorrectable error occurred on the ISA bus. If NMI is enabled and CHCK# is asserted and NMI will be generated to the CPU.'
NMI	98	O	Non-maskable interrupt (to the CPU): If NMI is enabled, NMI will be asserted to the CPU due to a system parity error or ISA bus channel check.
SPKD	48	O	Speaker data output: This signal is generated by the output of counter 2 and controlled by Port 61h, bit 1.
NC	194		No connection

Signal Descriptions (Cont.)

3.1.13 Power Management Signals

Signal Name	Pin No.	Signal Type	Signal Description
EPMI#/MDIR#	62	I/O	This pin is either MDIR# or EPMI#. When configured as MDIR#, the 82C895 will use this signal to buffer the DRAM data bus. This will be the direction control pin to the 74F245s. When active, the DRAM (MD) data bus will drive the CD bus. Otherwise, the CD bus will always drive the MD bus. When this pin is configured as EPMI#, it is an input which will signal the system to generate an SMI# or PPEN# depending on the configuration. A configuration register is available to control the functionality of this pin. MP3 is sampled during reset to determine the function of this pin. **MP3 Function** 1 EPMI# 0 MDIR#
SMIACT#	156	I	System management interrupt active: This is an input signal from the CPU which indicates SMM (System Management Mode).
SMI#/PPEN#	160	O	System management interrupt or power port enable: When an SLe CPU is used, this pin will be used as both PPEN# and SMI#. It will be used as SMI# to allow the system to go into the SMI_GREEN Mode and PPEN# when the system is to return to the NORMAL Mode. When a non-SLe CPU is used, this pin will be PPEN# and used as a strobe to the 82C602's GPM Port.

3.1.14 Power and Ground Pins

Signal Name	Pin No.	Signal Type	Signal Description
VCC	10, 43, 61, 97, 114, 147, 165, 201	I	Power connection: +5.0V
GND	15, 27, 37, 38, 60, 78, 79, 96, 119, 120, 130, 141, 142, 164, 182, 200	I	Ground connection

4.0 Functional Description

The following sub-sections will explain the various cycle and power management operations of the 82C895.

4.1 Reset Logic

The RST1# input to the 82C895 is used to generate the CPU reset (CPURST) signal. RST1# 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. This reset signal is used to force the system to begin execution at a known state. When PWRGD is sensed inactive, the 82C895 will assert CPURST. CPURST is also generated when a shutdown condition is decoded from the CPU bus definition signals. CPURST is asserted for 128 CLK cycles.

For systems with SLe CPUs, CPURST is the global reset while SRESET is active during a global reset and also during any warm reset. When SRESET is activated, the SMBASE Register does not change and UP# is not sampled. SRESET leaves the status of the on-chip FPU and SMBASE Register intact while resetting other units including the on-chip L1 cache. CPURST for the SLe CPU, is generated through the 82C602. The 82C602 asserts RSTDRV and LRESET# for 128 CLK cycles after PWRGD is active. Figure 4-1 shows the proper way to configure the resets.

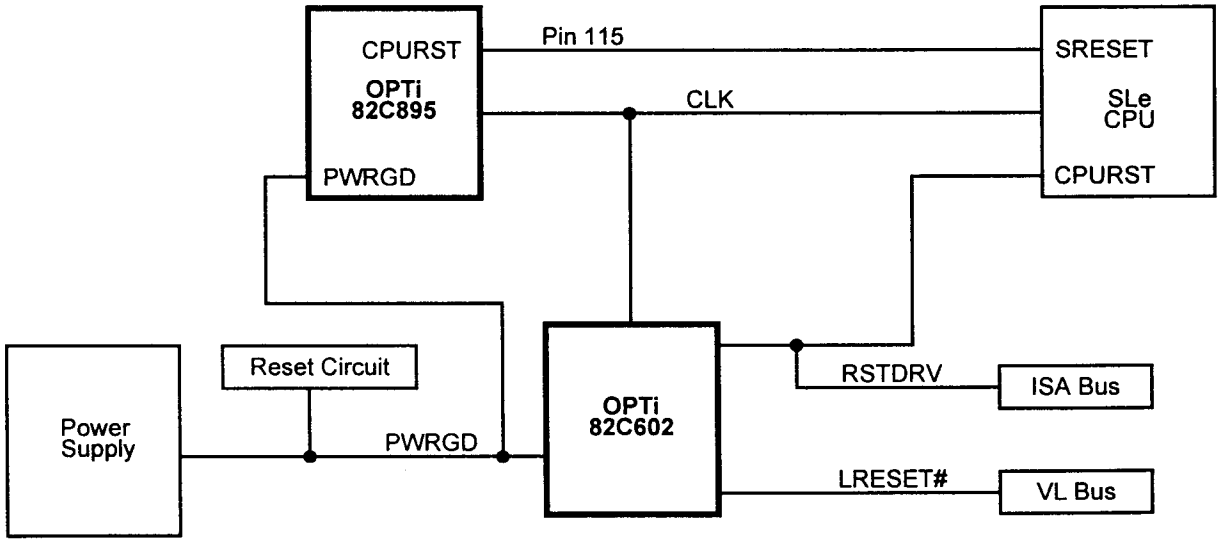
The 82C895 emulates the keyboard reset function. The keyboard reset is intercepted by monitoring the I/O write cycle "FE" command to Port 64h. This fast CPU reset from the chipset will be generated directly after the I/O write is decoded unless bit 1 of Index Register 20h is cleared to 0, in which case the reset will not start until a "HALT" instruction is executed.

4.2 System Clock Generation

The 82C895 has a single high-frequency clock input, CLK. CLK is a master single-phase clock which is used to drive all the host CPU synchronous signals and all of the 82C895's internal state machines. This clocking scheme provides operation to support platforms at system speeds up to 50MHz.

The 82C895 generates the AT bus clock (ATCLK) from an internal division of CLK. The ATCLK frequency is programmable and can be set to any of four synchronous mode clock division options by programming Register 25h, bits 1 and 0. In addition to the CLK source, there is an asynchronous mode available by clearing bit 1 of Register 27h, which generates the ATCLK by dividing the ATBUS OSC oscillator by 2 (OSC/2). This asynchronous mode is important when entering the GREEN Mode, where the CPU clock rate can change and thereby, generate unsuitable ATCLK frequencies if left in the synchronous mode. This allows the system designer to tailor the AT bus clock frequency to support a wide range of system designs and performance platforms, as well as to function reliably during power saving modes.

Figure 4-1 Reset Connection Example



4.3 CPU Burst Mode Control

The 82C895 chipset fully supports 486 burst cycles. The 82C895 cache and DRAM controllers insure that data is burst into the CPU whenever the 486 requests a burst linefill. The secondary cache provides data on read-hits and the DRAM supplies the data during cache read-misses.

For a cache read-hit cycle, BRDY# (Burst Ready) is asserted during the first T2 state when a 2-1-1-1 (zero wait state) cache burst cycle is chosen, otherwise it is asserted during the second T2 state when one wait state is required. If a read-miss occurs, the DRAM controller will burst new data into both the cache memory and CPU simultaneously. BRDY# will be asserted for each double-word during these cache read-miss update cycles. For a zero wait state cache burst read cycle, 2-1-1-1, BRDY# will be asserted during the first T2 and remain active until BLST# (Burst Last) from the CPU is detected. For a 3-2-2-2 cache burst read cycle, BRDY# will be toggled active on an every-other-clock basis to accommodate the more relaxed data access timing required above 33MHz operation. BRDY# is never active during DMA or master cycles.

The 82C895 contains separate burst counters to support DRAM and external cache burst cycles. The read/write DRAM burst counter performs the cache read-miss linefill (DRAM to external cache/CPU) and the cache burst counter supports the 486 burst linefill (external cache to the 486 CPU). The access order of the burst counter exactly matches the double-word address sequencing expected by the 486 CPU. The DRAM burst counter is used for cache read-miss cycles and dirty linefill write operations.

4.4 L1 Write-Back Timing Description

Level 1 Write-Back Support

The L1 cache can contain modified data that is not contained in the L2 cache or DRAM. The CPU will not allow external devices to access its internal cache. The 82C895 will execute an inquire cycle to the L1 cache for all master accesses to the system memory area. Master devices, whether local or on the ISA bus, must snoop the L1 cache during every access to system memory. If valid information is in the L1 cache and this information has been modified without being updated to the system memory, the HITM# signal will be generated. A write-back cycle must be generated whenever a modified line is hit. In this case, the CPU will write the line back to the L2 cache.

VESA Local Master Cycles

The L1 cache inquire cycle begins with the CPU relinquishing the bus with the assertion of HLDA. On sampling HLDA, the local bus card will generate ADS#. EADS# will be generated by the 82C895 for one clock following the ADS# generation. If the CPU does not respond with assertion of HITM#, the 82C895 will complete the cycle from the L2 cache or the system memory. If HITM# is asserted, the 82C895 will expect a castout cycle from the L1 cache. HITM# is connected to the WBACK# signal on the VL bus which will abort the VL cycle and allow the CPU to perform its castout cycle. The 82C895 will release hold to the CPU and generate RDY# to terminate the local bus cycle. Next, the CPU will write-back its L1 contents to cache/system memory.

Master/DMA Write Cycle

HOLD is generated to the CPU in response to an ISA master or DMA cycle. The CPU then relinquishes the bus with the assertion of HLDA. The 82C895 issues AHOLD to the CPU to tristate the CPU's address bus. At this time, the DMA or master device drives the address onto the CPU bus and CHRDY is released. EADS# is generated by the 82C895 and HITM# will be generated if the address is a modified line in the cache. The CPU will then perform its castout cycle always starting at the address 0X0 of the 16-byte line. After the castout cycle, the CPU deasserts HITM# and issues HLDA. The ISA master or DMA device can then finish its cycle.

4.5 Cache Subsystem

The integrated cache controller, which uses a direct-mapped, bank-interleaved scheme 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, and sizes of 64KB, 128KB, 256KB, and 512KB are supported. Provisions for two programmable non-cacheable regions are provided. The cache controller operates in a non-pipeline mode with a fixed 16-byte line size (optimized to match a 486 burst linefill) in order to simplify the motherboard design without increasing cost or degrading system performance. For 486 systems, the secondary cache operates independently and in addition to the CPU's internal cache.

4.5.1 Cache Bank Interleave

In order to support cache burst cycles at elevated frequencies and still utilize conventional speed SRAMs, a bank-interleave cache access method is employed. The addresses are applied to the cache memory one cycle earlier, while cache output enable signals control even/odd bank selection and enable cache RAM data to the CPU data bus. Since the output enable time is about one-half of the address access time, the 82C895 can achieve a high-performance cache burst mode without using more expensive high-speed SRAMs.

The 82C895 supports one or two cache banks. Two cache banks are required to interleave and optimally realize the performance advantages of this cache scheme. Cache sizes of 128KB and 512KB are single-bank caches, while 64KB and 256KB cache sizes are double-bank configurations. When using a double-bank configuration, the even and odd banks

receive the same address lines. Signals BEA2OA3, BEA3, ECAWE#/OCAWE#, and BEOE#/BOOE# are used to dictate the even or odd bank access.

4.5.2 Write-Back Cache

The write-back cache scheme derives its superior performance by optimizing write cycles. There is no performance penalty in the cache write cycle, since the cache controller does not need to wait for the much slower DRAM controller to finish its import before proceeding to the next cycle.

4.5.2.1 Tag RAM

A built-in tag comparator improves system performance while reducing component count on the system board. The comparator internally detects the cache hit/miss status by comparing the high-order address bits (for the memory cycle in progress) with the stored tag bits from previous cache entries (see Table 4-1). When a match is detected, and the location is cacheable, a cache hit cycle takes place. If the comparator does not match, or a non-cacheable location is accessed (based on the internal non-cacheable region registers), the current cycle is a cache miss.

The tag is invalidated automatically during memory reads when the cache is disabled; each memory read will write into the corresponding tag location a non-cacheable address (such as A0000h or B0000h of the video memory area). To flush the cache, simply disable the cache in Configuration Register 21h and read a block of memory equal to the size of the cache. The advantage of this invalidation scheme is that no valid bit is necessary and expensive SRAM can be conserved.

Table 4-1 details which CPU address bits are stored as tags for the various cache sizes supported in the 82C895.

Table 4-1 Address to Tag Bit Mapping

TAG Bit	64KB	128KB	256KB	512KB
7*	A23	A24	A25	A26
6	A22	A22	A22	A22
5	A21	A21	A21	A21
4	A20	A20	A20	A20
3	A19	A19	A19	A19
2	A18	A18	A18	A25
1	A17	A17	A24	A24
0	A16	A23	A23	A23

* TAG7 is invalid when either MP2 is pulled up or Register 20h, bit 4 is = 1. Therefore, cacheable memory size is reduced to one-half its original size, respectively.

4.5.3 Dirty Bit Mechanism

The "dirty bit" is a mechanism for monitoring data coherency between the external cache subsystem and DRAM. 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 82C895 to determine whether the data in memory is "stale" and needs to be updated before a new memory location is allowed to overwrite the currently indexed cache entry. The write-back cycle causes an entire cache line (16 bytes) to be written back to memory, followed by a line burst from the new memory location into the cache and CPU. Normally, the performance advantage of completing fast writes to the cache outweigh the "write-back" read-miss penalties which are incurred while operating the write-back scheme.

Possible cache cycles are detailed next:

Cache Read-Hit
The secondary cache provides the data to the CPU directly. The 82C895 cache controller follows the CPU's burst protocol to fill the processor's internal cache line.

Cache Read-Miss (Dirty Bit Negated): Import Cycle
The cache controller does not need to update system memory with the cache's current data, because that data has not been modified (evidenced by the dirty bit negation). The cache controller asserts TAGW#, causing the tag RAMs to update with the new address, and asserts BEOE#/BOOE#, causing the cache memory to update with data from the new DRAM line. Data is presented to the CPU and the secondary cache concurrently (following the 486 burst protocol).

Cache Read-Miss (Dirty Bit Asserted): Castout Cycle
The cache controller must update the system memory with data from the cache location that is going to be overwritten. The 82C895's cache controller writes the 16-byte line from cache memory into DRAM, then reads the new line from DRAM into the cache memory and deasserts the dirty bit. The cache controller asserts TAGW#, BEOE#/BOOE#, and DRTYW# during this linefill. This new data is presented to the CPU and to the secondary cache concurrently (following the 486 burst protocol).

Cache Write-Hit
In a write-back cache, the memory controller does not need to update the much slower DRAM memory or write-hit. If the cache controller is set for zero wait state operation and the dirty bit is set, the controller will run a zero wait state cycle.

Cache Write-Miss
The cache controller bypasses the cache entirely and writes the data directly into DRAM. The dirty bit is unchanged. No import cycle to the cache takes place.

Table 4-2 shows the various cache cycles at different operational speeds.

Table 4-3 shows the cache sizes supported by the 82C895, with the corresponding tag RAM address bits, tag RAM size, cache RAM address bits, cache RAM size, and cacheable main memory size.

Table 4-4 shows recommended data and tag SRAM speeds for relative CPU clock rates.

Table 4-2 Cache Cycles

Cycle Type	25MHz	33MHz	40MHz	50MHz
Cache hit	2-1-1-1	2-1-1-1	2-2-2-2	3-2-2-2/4-2-2-2*
Cache miss, page hit	3-2-2-2	3-2-2-2	4-3-3-3	5-4-4-4/6-4-4-4*
Cache miss, page miss	8-2-2-2	8-2-2-2	9-3-3-3	12-4-4-4-/13-4-4-4*
No L2 cache, page-hit	3-2-2-2	3-2-2-2	4-3-3-3	5-4-4-4/6-4-4-4*
No L2 cache, page miss	8-2-2-2	8-2-2-2	9-3-3-3	12-4-4-4/13-4-4-4*
RAS inactive	5-2-2-2	5-2-2-2	7-3-3-3	8-4-4-4/9-4-4-4

* If the AT bus clock is equal to the system clock divided by 6, ADS# will be delayed one T-state.

Table 4-3 Cache SRAM Requirements

Cache Size (KB)	Tag Field Address Tag RAM Size	Dirty SRAMs Size	Cache SRAM Address Qty/Cache RAM Size	Cacheable Main Memory (MB)
64	A[23:16] 8Kx8	16Kx1	A[15:2] 8 ea./8Kx8	16 8*
128	A[24:17] 8Kx8	16Kx1	A[16:2] 4 ea./32Kx8	32 16*
256	A[25:18] 32Kx8	16Kx1	A[17:2] 8 ea./32Kx8	64 16*
512	A[26:19] 32Kx8	64Kx1	A[18:2] 4 ea./128Kx8	128 64*

* Cacheable address range when used with 7-bit tag.

Table 4-4 SRAM Speed Requirements

Speed (MHz)	Cache SRAM (ns)	Tag SRAM (ns)	DRAM Speed	Note 1
16	25	25	80	Cache write 0 wait state Cache read burst 2-1-1-1 Single/double bank
20	25	25	80	Cache write 0 wait state Cache read burst 2-1-1-1 Single/double bank
25	20	25	80	Cache write 0 wait state Cache read burst 2-1-1-1 Single bank
25	25	25	80	Cache write 0 wait state Cache read burst 2-1-1-1 Double bank
33	20	15	80	Cache write 0 wait state Cache read burst 3-2-2-2 Single bank
33	20 (Note 2)	15	80	Cache write 0 wait state Cache read burst 2-1-1-1 Double bank cache only
40	20	15	80	Cache write 1 wait state Cache read burst 3-2-2-2 Single/double bank
50	20	15	80	Cache write 1 wait state Cache read burst 3-2-2-2 Single/double bank

Notes: 1. DRAM and cache cycles are at their minimum wait states.
2. 20ns SRAM with Tdoe equal to or less than 10ns

4.6 Local DRAM Control Subsystem

The 82C895 supports up to eight banks of page-mode local DRAM memory for configurations of up to 64MB. 256KB, 1MB, 4MB, or 16MB page-mode DRAM devices may be used. The DRAM configuration is programmable through Configuration Register 24h. DRAM performance features are programmable through Configuration Register 25h.

Table 4-5 gives the possible DRAM configurations. For additional information, please refer to the DRAM Control Register 1 (Index 24h) for programming options.

Table 4-5 DRAM Configurations

Bank 0	Bank 1	Bank 2	Bank3	Total	Reg. 24h [6:4, 2:0]
256Kx36	256Kx36	x	x	2M	000,000
1Mx36	x	x	x	4M	000,010
256Kx36	256Kx36	256Kx36	256Kx36	4M	000,001
256Kx36	1Mx36	x	x	5M	101,011
256Kx36	256Kx36	1Mx36	x	6M	000,011
1Mx36	1Mx36	x	x	8M	000,101
1Mx36	x	1Mx36	x	8M	000,100
256Kx36	256Kx36	1Mx36	1Mx36	10M	000,110
1Mx36	x	1Mx36	1Mx36	12M	000,111
4Mx36	x	x	x	16M	001,001
1Mx36	1Mx36	1Mx36	1Mx36	16M	001,000
256Kx36	4Mx36	x	x	17M	101,100
1Mx36	4Mx36	x	x	20M	101,101
1Mx36	x	4Mx36	x	20M	101,111
4Mx36	4Mx36	x	x	32M	001,011
4Mx36	x	4Mx36	x	32M	001,010
16Mx36	x	x	x	64M	101,110
4Mx36	4Mx36	4Mx36	4Mx36	64M	001,100
16Mx36	16Mx36	x	x	128M	110,001
16Mx36	x	16Mx36	x	128M	110,000

Table 4-6 describes how the DRAM address lines are multiplexed for any memory device types.

Table 4-6 CPU Address to MA Bus Mapping

Memory Address	Column Address	Row Address
MA0	A2	A13
MA1	A3	A14
MA2	A4	A15
MA3	A5	A16
MA4	A6	A17
MA5	A7	A18
MA6	A8	A19
MA7	A9	A11
MA8	A10	A12
MA9	A21	A20
MA10	A23	A22
MA11	A25	A24

4.7 Parity Generation/Detection Logic

During local DRAM write cycles, the 82C895 generates a parity bit for each byte of write data from the processor. Parity bits are stored into 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 82C895 generates a parity error to the CPU. The parity error will invoke the NMI interrupt, providing the parity check is enabled in the Configuration Register 21h, bit 5. Parity check must also be enabled in the Port B (61h) Register, bits 2 and 3.

4.8 Refresh Logic

The 82C895 supports both normal and hidden refresh. Normal refresh refers to the classical refresh implementation which places the CPU on "hold" while a refresh cycle takes place to both the local DRAM and any AT bus memory. This is the default condition at power-up. However, hidden refresh is performed independent of the CPU and does not suffer from the performance restriction of losing processor bandwidth by forcing the CPU into its hold state.

Hidden refresh delivers higher system performance and is recommended over normal refresh. As long as the CPU does not try to access local memory or the AT bus during a hidden refresh cycle, refresh will be transparent to the CPU. The CPU can continue to execute from its internal cache and execute internal instructions during hidden refresh without any loss in performance due to refresh arbitration. If a local memory or AT bus access is required during hidden refresh, wait states will be added to the CPU cycle until the resource becomes available. Hidden refresh also separates refreshing of the AT bus and local DRAM.

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

The 82C895 implements refresh cycles to the local DRAM using CAS-before-RAS timing. CAS-before-RAS refresh has lower power consumption 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 local memory.

The output of internal Counter1/Timer1 (OUT1) inside the 82C895 is programmed as a rate generator to produce the periodic refresh request signal which occurs every 15.9µs. Requests for refresh cycles are generated by two sources: internally by Counter1/Timer1, or alternatively, a 16-bit ISA master may initiate a refresh cycle. These ISA masters 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.

By programming Configuration Register 22h, bit 0, slow refresh may be enabled to further divide the 15.9µs period by four to provide a 63.6µs "slow refresh" interval. (Slow refresh DRAMs must be used with the slow refresh feature.)

4.9 Shadow RAM and BIOS Cacheability

When using the 82C895, the procedures listed below should be followed to properly setup and configure 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# enabled manually in Index Register 2Dh.
2. Enable ROM contents to be copied to into DRAM. Enabling shadow RAM copy enable (read from ROM, write to DRAM) for address range C0000h-EFFFFH is controlled in Index Register 26h[6]. The F0000h-FFFFFh segment copy enable bit is in Index Register 22h[7]. These bits must be set so that reads to these segments will be executed out of the ROM and writes will be translated into DRAM.
3. Enable shadow RAM area to permit DRAM read/write accesses. The next step is to enable the individual shadow RAM areas so that both read and write accesses will be executed in DRAM exclusively. At this point, the ROMCS# generation bits that were previously necessary to access the original ROM code must be disabled (Index Register 2Dh). These ROMCS# generation bits will over-ride shadow RAM settings, which makes the disabling of these bits necessary. The following registers control shadow RAM enabling for the individual segments.

   C0000h-CFFFFh Index 26h[3:0] (with 16KB granularity)
   D0000h-DFFFFh Index 23h[3:0] (with 16KB granularity)
   E0000h-EFFFFh Index 23h[7:4] (with 16KB granularity)
   F0000h-FFFFFh Index 22h[7] (with 64KB granularity)

   (The F0000h-FFFFFh segment is handled uniquely; this segment will be shadowed, write-protected, and cached in the L2 cache with the clearing of a single bit: Index 22h[7].)
4. Write-protect shadow RAM areas. Generally, shadow RAM areas should be write protected to ensure the integrity of the code. This can be accomplished by setting the following Index Registers:

   26h[5] for the C0000h-CFFFFh segment
   22h[4] for the D0000h-DFFFFh segment
   22h[3] for the E000h-EFFFFh segment

5. Cache shadow RAM area in L2/L1 caches (optional). Caching of the individual code segments can be accomplished through Index Registers 2Eh and 2Fh. Although write protection of these areas will still be honored in the L2 (external) cache, 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.

4.10 System ROM BIOS Cycles

The 82C895 supports both 8- and 16-bit EPROM cycles. If the system BIOS is 16 bits wide, ROMCS# should be connected to M16# through an open collector gate indicating to the 82C895 that a 16-bit EPROM is responding. The system BIOS resides on the XD bus.

ROMCS# is generated by default for the system BIOS (F0000-FFFFFh) segment. In addition, Configuration Register 2Dh may be programmed to generate ROMCS# for the C0000h-EFFFFh block with 32KB granularity. This feature is extremely useful when integrating various adapters on the system board.

4.11 AT Bus State Machine

The AT bus state machine gains control when the 82C895's decoding logic detects a non-local memory cycle. It monitors status signals M16#, IOCS16#, CHRDY, and NOWS# and performs the necessary synchronization of control and status signals between the AT bus and the microprocessor. The 82C895 supports 8- and 16-bit memory and I/O devices located on the AT bus.

An AT bus cycle is initiated by asserting ALE in AT-TS1 state. On the trailing edge of ALE, M16# is sampled for a memory cycle to determine the bus size. It then enters AT-TC state and provides the command signal. For an I/O cycle, IOCS16# is sampled after the trailing edge of ALE until the end of the command.

Typically, the wait state for an AT 8/16-bit transaction is 5/1, respectively. The command cycle is extended when CHRDY is detected inactive, or the cycle is terminated when the zero wait state request signal (NOWS#) from the AT bus is active. Upon expiration of the wait states, the AT state machine terminates itself and passes an internal READY to the CPU state machine for outputting a synchronous RDY# to the CPU. Index Register 20h, bit 2, allows for the addition of an AT cycle wait state. Bit 3 of this same register allows for the generation of a single ALE instead of multiple ALEs during bus conversion cycles. The AT bus state machine also routes data and address when an AT bus master or DMA controller accesses memory.

4.12 Bus Arbitration Logic

The 82C895 provides arbitration between the CPU, DMA controller, AT bus masters, and the refresh logic. During DMA, AT bus master, and conventional refresh cycles, the 82C895 asserts HOLD to the CPU. The CPU responds 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 82C895 responds by issuing RFSH# (refresh cycle) or generating the appropriate DRQ (AT 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 (refer to the refresh section for additional information).

The AT bus controller in the 82C895 arbitrates between DMA/master and refresh requests, deciding which will own the bus once the CPU relinquishes control with the HLDA signal. The arbitration between refresh and DMA/master is based on a FIFO (first in-first out) priority. However, a refresh request (RFSH#) is internally latched and serviced immediately after the DMA or master finishes its term if queued after. DRQs must remain active to be serviced if a refresh request comes first. The "MASTER#" signal from the AT bus indicates an active AT bus master cycle.

4.13 Local Bus Interface

The 82C895 allows peripheral devices to share the "local bus" with the CPU. The performance of these devices (which may include the video subsystem, hard disk adapters, LAN, and other PC/AT controllers) will dramatically increase when allowed to operate in this high-speed environment. These devices are responsible for their own address and bus cycle decodes and must be able to operate compatibly at the elevated frequencies required for operation on the local CPU bus.

The LDEV# input signal to the 82C895 indicates that a local device is intercepting the current cycle. If this signal is sampled at the end of the first T2 clock cycle (end of the second T2 at 50MHz), then the 82C895 will allow the responding local device to assume responsibility for the current local cycle. When the device has completed its operation, it must terminate the cycle by asserting RDY# or BRDY# to the CPU. RDY# and BRDY# are bidirectional pins on the 82C895 and may be driven by a local bus peripheral or the chipset to terminate their respective cycles.

4.14 Data Bus Conversion/Data Path Control Logic

The 82C895 performs data bus conversion when the CPU accesses 16- or 8-bit devices through 16- or 32-bit instructions. It also handles DMA and AT master cycles that transfer data between local DRAM or cache memory and locations on the AT bus. The 82C895 provides all of the signals to control external bidirectional data buffers.

4.15 Turbo/Slow Mode Operations

The Turbo Mode is controlled through Index Register 27h, bit 3. The system will run at full speed if this bit is set to 1 and Non-turbo (slow) Mode when this bit is set to 0. The slow mode operation is implemented by applying a periodic clock to the HOLD input of the CPU. OSC12 is the clock source used for this operation. OSC12 is internally derived from the 14.31818MHz OSC clock input to the 82C895. HOLD is maintained for approximately two-thirds of the time, while the CPU is allowed to perform normal external operations during the remaining one-third interval.

4.16 Fast GATEA20 and RESET Emulation

The 82C895 will intercept commands to Ports 60h and 64h so that it can 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 60h. The fast CPU "warm reset" function is generated when a Port 64h write cycle with data "FEh" is decoded. A write to Port 64h with data "D0h" will enable the status of GATEA20 (bit 1 of Port 60h) and the warm reset (bit 0 of Port 60h) to be readable.

4.17 Special Cycles

The 486 microprocessor provides special bus cycles to indicate that certain instructions have been executed, or certain conditions have occurred internally. Special cycles such as shutdown and halt cycles are covered by dedicated handling logic in the 82C895. This logic decodes the CPU bus status signals MIO#, DC#, and WR# and executes the appropriate action.

4.18 Power Management Features

The OPTi 82C895 along with the 82C602 provides for an optimum GREEN solution. The 82C602 provides a GREEN power management port for controlling desktop subsystems which may include clock control to the CPU's clock (STPCLK# signal to the CPU) to monitor shutdown conditions.

The 82C895 provides a Green Event Timer (GET) used to activate the AUTO_GREEN or SMI_GREEN Modes. The GET can be reloaded by any IRQ, local bus, DMA request, keyboard, video, and hard/floppy disk accesses. It can also be reloaded by a programmable I/O subsystem activity and an optional external source.

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. This SMI_GREEN Mode requires SL enhanced (SLe) CPUs.

4.18.1 System Activity Detection

The GET countdown timer will reload under the following events: (Selection of these events are fully programmable in the chipset's indexed registers.)

- All IRQs
- One programmable I/O range
- LDEV# and LREQ# signals from the VESA local bus
- All DREQs
- Keyboard access:
  - I/O Ports 60h and 64h
- 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)
- External EPMI source:
  - Additional input pin to the 82C895 from an external PMI source
- All interrupt vector addresses (00h-0FFh, corresponding to address 00h-3FFh) with two maskable vector addresses

Any of the following conditions will allow the system to return to the NORMAL state if the event was programmed to allow the system to go into the GREEN Mode.

- All IRQs
- External EPMI
- One programmable I/O
- Keyboard access
- Video access
- Hard/floppy disk access
- External EPMI source
- All interrupt vector addresses

4.18.2 Definition of Power Management Modes

The following sub-sections define the various power management modes used when configuring systems with OPTi's 82C895 and 82C602 to run in the AUTO_GREEN and AUTO_SMI Modes.

4.18.2.1NORMAL Mode

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

4.18.2.2AUTO_GREEN Mode

This mode is used to accommodate non-SLe CPUs. It allows for power management through hardware control. The AUTO_GREEN Mode is entered when either the chipset's GET expires or an EPMI# occurs. The 82C895 automatically switches the AT bus clock to the asynchronous mode (which is derived from the 14.318MHz clock). It then sends PPEN# to the 82C602's PPEN# pin. This sends the 82C895's GREEN Latch Register onto the SD[3:0] bus and allows the 82C602's Green Power Management (GPM) Port to latch this on its outputs. These outputs support some power management functions such as sending a SLWCLK# bit to a clock synthesizer to slow the CPU's clock within specification. They may also be used to control shutdown of the monitor and other system peripherals.

The system can resume from the AUTO_GREEN Mode by any event programmed in the System Activity Registers. PPEN# will reload the default values into the 82C602. While returning to the NORMAL Mode, the CPU clock first runs at full speed for 20ms before the AT bus clock is switched back to the synchronous mode. Figure 4-2 gives a flowchart for the AUTO_GREEN Mode.

4.18.2.3SMI_GREEN Mode

The SMI_GREEN Mode is used to accommodate SMI supported CPUs. It allows power management through the SMI# protocol. After either the GET expires, an EPMI# occurs, or a forced SMI (bit 3 of Index Register E1h) happens, an SMI# is generated from the 82C895 to the CPU. The 82C895 flushes the L1 cache and then remaps all 3XXXX memory accesses with the assertion of SMIACT#. The CPU will save all of its internal registers and then begin executing the SMI code. In the SMI code, the 82C602's GPM Port can be written to via Index Register FAh. This register can control the CPUCLK, STPCLK#, and monitor syncs.

The system can resume out of the SMI_GREEN Mode by any event programmed in the System Activity 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 AT bus clock is switched back to the synchronous mode. Figure 4-3 shows a flowchart of the SMI_GREEN Mode.

Figure 4-2 AUTO_GREEN Mode Flowchart

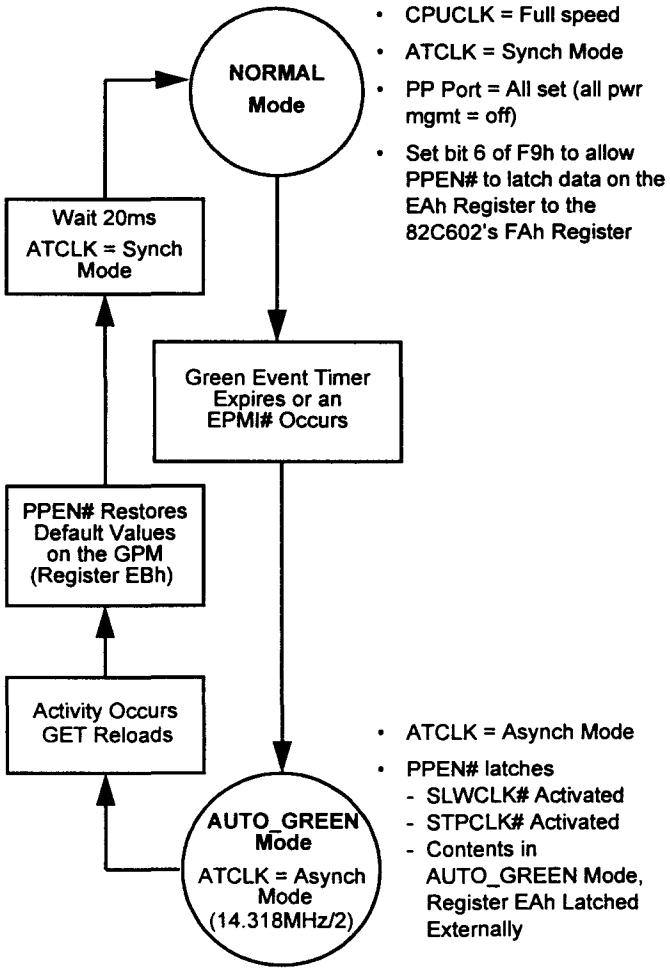
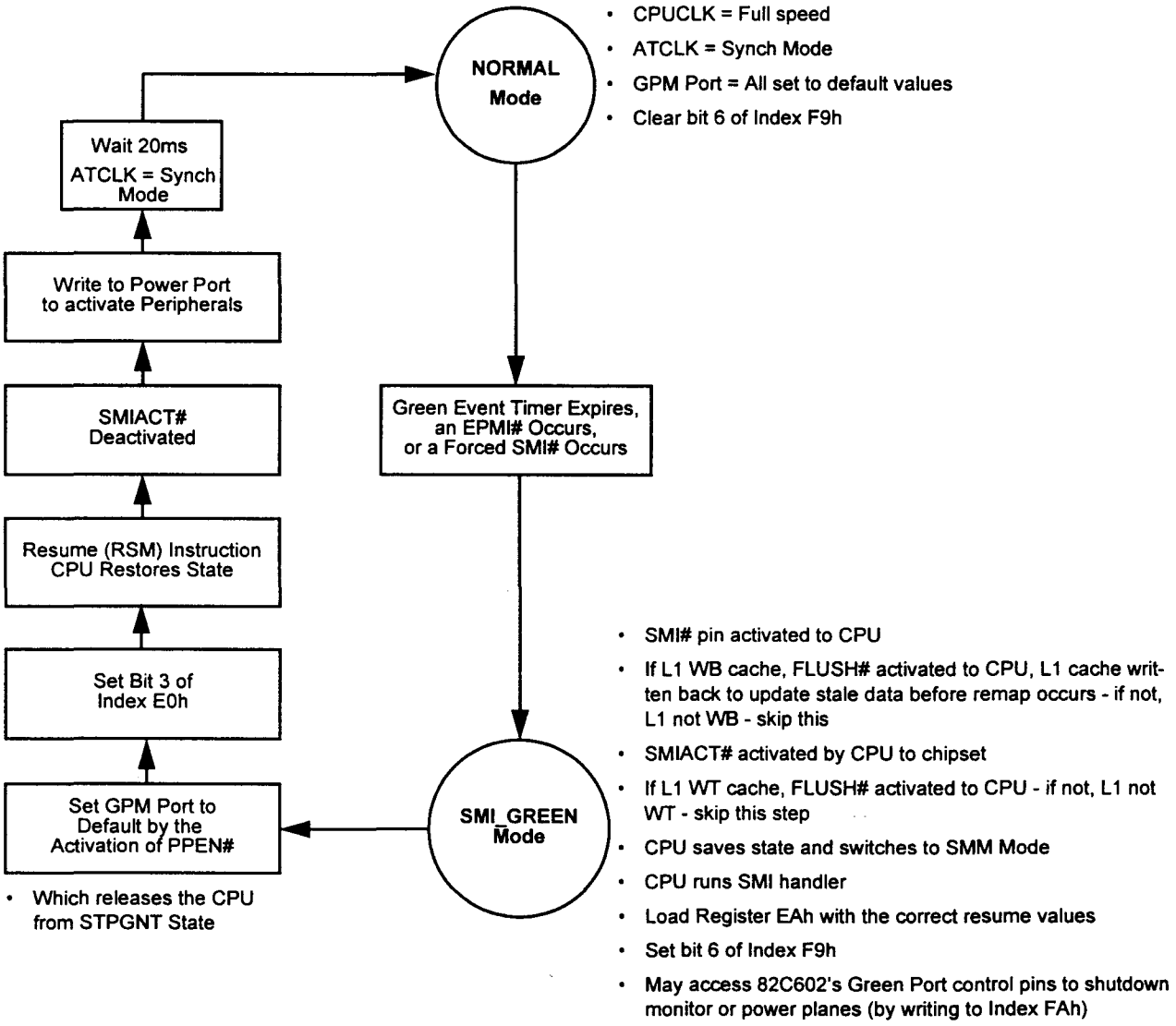


Figure 4-3 SMI_GREEN Mode Flowchart



Notes: 1.For back-to-back SMIs, a 6µs delay will prevent another SMI from triggering.
2.For SRESET occurring during resume, a 14µs (or 64 CPU clock cycles) delay has been added before the SRESET can occur, after SMIACT# goes away.

4.19 Internal Peripherals Controller

The following subsections will give detailed operational information about the 82C895's internal peripherals controller (IPC).

4.19.1 Top Level Decoder & Configuration Register

The IPC's top level decoder 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 5.4 to program the various IPC registers.

4.19.2 DMA Subsystem

The IPC 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 4-7 gives the I/O address map of the IPC's DMA subsystem. The mapping is fully PC/AT compatible.

4.19.2.1 DMA Operation

During normal operation, the DMA subsystem of the IPC 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 the S1 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.

4.19.2.1.1 Idle Mode

If no peripheral requests service, the DMA subsystem will enter the Idle Mode and perform only S1 idle states. During this time, the IPC 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 Mode 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.

4.19.2.1.2 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 4-7. Erratic operation of the IPC 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 IPC from attempting to service a peripheral with a channel which is only partially programmed.

Table 4-7 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

4.19.2.1.3 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 IPC will issue a 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 IPC asserts the MEMR# command. Then the DMA will enter the S3 state where the IPC 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 deasserted.

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.

4.19.2.2 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.

4.19.2.2.1 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 IPC will deassert HOLD and release the bus to the system once the transfer is complete. After HLDA has gone inactive, the IPC 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.

4.19.2.2.2 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.

4.19.2.2.3 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 deasserted, 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.

4.19.2.2.4 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 4-4 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.

4.19.2.3 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.

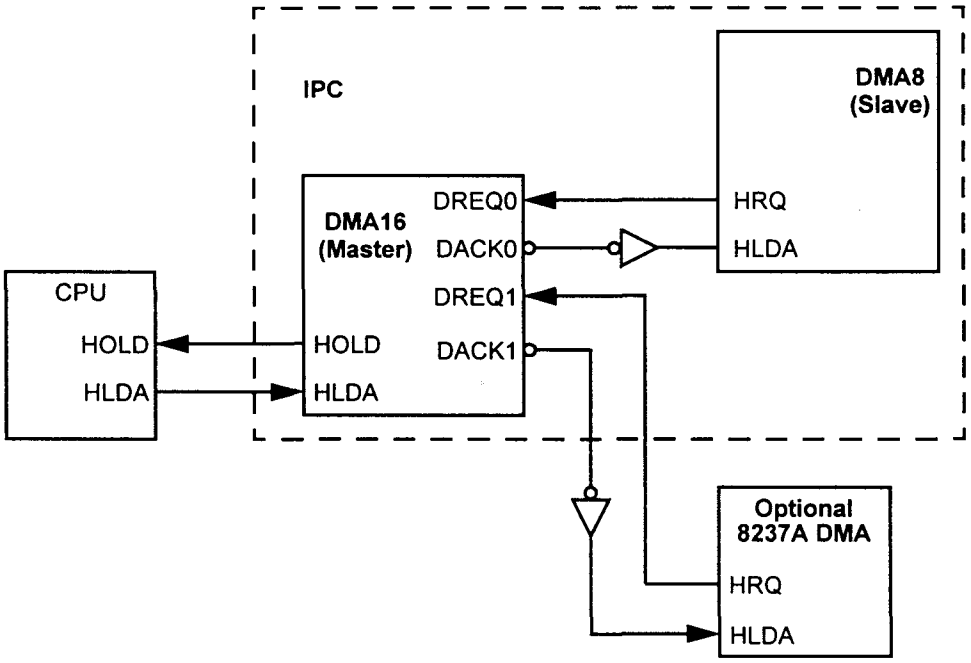
Read Transfers move data from memory to an I/O peripheral by generating the memory address and asserting MEMR# and IOW# during the same transfer cycle.

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.

Figure 4-4 Cascade Mode Interconnect



4.19.2.3.1 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.

4.19.2.3.2 DREQ Priority

The IPC 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 4-8 shows the rotating priority scheme. In cases where multiple requests occur at the same time, the IPC 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.

4.19.2.3.3 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 A[15:8] on the data bus at this time whereas DMA16 will output A[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 IPC will output the lower eight bits of address on A[7:0]. A[23:16] are also generated at this time from a DMA page register in the IPC. Note that A16 is output on the A16 pin of the device.

During 16-bit DMA transfers in which DMA16 is active, the IPC will output the lower eight bits of address on A[8:1]. A[23:17] are also generated at this time from a DMA page register in the IPC. Note that A0 and A16 remain tristated during 16-bit DMA transfers

The DMA page registers are a set of 16 8-bit registers in the IPC 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 4-9 along with its CPU I/O read/write address.

Table 4-8 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

During Demand and Block Transfers, the IPC 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 IPC will only update the latch contents when necessary. The IPC will there only execute an S1 state when necessary and improve the overall system throughput.

Table 4-9 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

4.19.2.3.4 Compressed Timing

The DMA subsystem in the IPC 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 IPC 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.

4.19.3 DMA Register Descriptions

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

4.19.3.1 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.

4.19.3.2 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 IPC will generate TC and either suspend the operation on that channel and set the appropriate request mask bit, or auto-initialize and continue.

4.19.3.3 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.

4.19.3.4 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.

4.19.3.5 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.

4.19.3.6 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.

4.19.3.7 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 request. 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.

4.19.3.8 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.

4.19.3.9 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.

4.19.3.10 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.

4.19.4 Special Commands

Five Special Commands are provided to make the task of programming the IPC 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 IPC 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.

4.19.5 Interrupt Controller Subsystem

The programmable interrupt controllers in the IPC serve as a system wide interrupt manager in an X86 system. 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.

4.19.5.1 Interrupt Controller Subsystem Overview

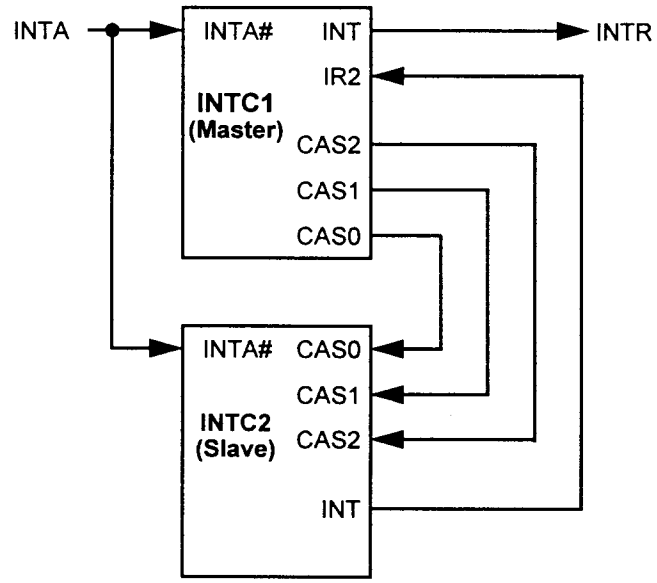
There are two interrupt controllers, INTC1 and INTC2, included in the IPC. Each of the interrupt controllers is equivalent to an 8259A device operating in X86 mode. The two devices are interconnected and must be programmed to operate in the Cascade Mode for all 16 interrupt channels to operate properly. Figure 4-5 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 4-10 lists the 16 interrupt channels and their interrupt request sources.

Description of the interrupt subsystem will pertain to both INCT1 and INCT2 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 4-5 Internal Cascade Interconnect



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

Table 4-10 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

4.19.5.2 Interrupt Controller Operation

Figure 4-6 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.

4.19.5.3 Interrupt Sequence

The IPC 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 IPC 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 4-7). 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 IPC 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 4-11. Note that V[7:3] in Table 4-11 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), INCT1 will issue an interrupt level 7 vector during the second INTA cycle.

Figure 4-6 Interrupt Controller Block Diagram

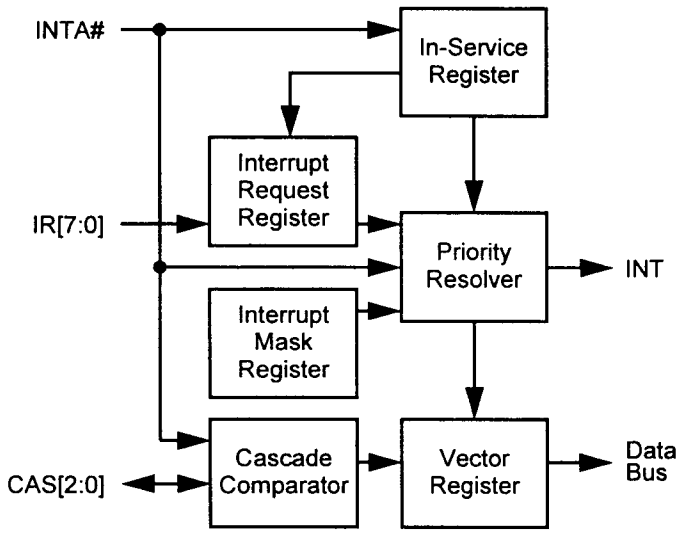


Figure 4-7 Interrupt Sequence

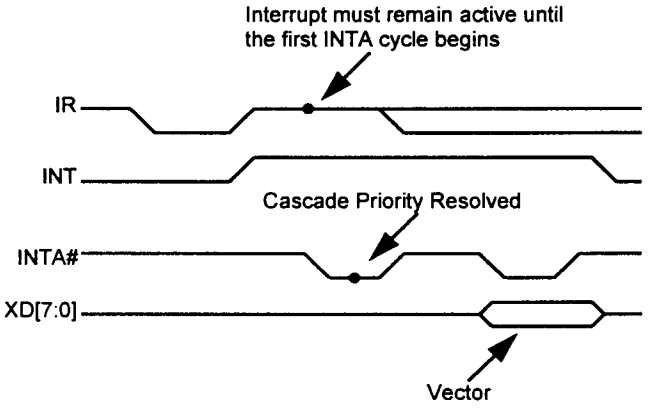


Table 4-11 Interrupt Vector Byte

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

4.19.5.4 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 IPC 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.

4.19.5.5 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).

4.19.5.5.1 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.

4.19.5.5.2 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

4.19.5.5.3 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

4.19.5.6 Programming the Interrupt Controller

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

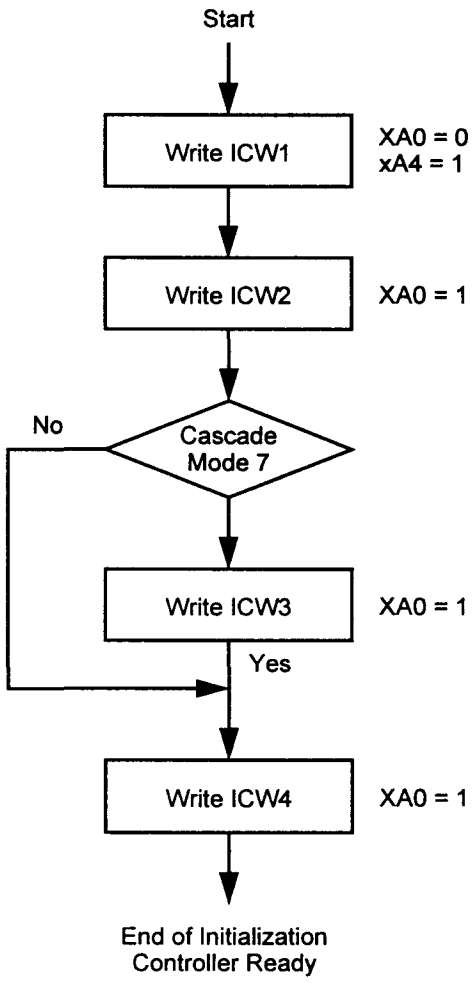
4.19.5.6.1 Initialization Command Words (ICWs)

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 4-8 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.

Figure 4-8 Initialization Sequence



4.19.5.6.2 Operational Command Words (OCWs)

Operational Command Words (OCWs) allow the IPC'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).

4.19.5.6.3 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.

Table 5-50 through Table 5-52 give each of these registers' formats.

4.19.6 Counter/Timer Subsystem

The IPC 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 4-9. 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

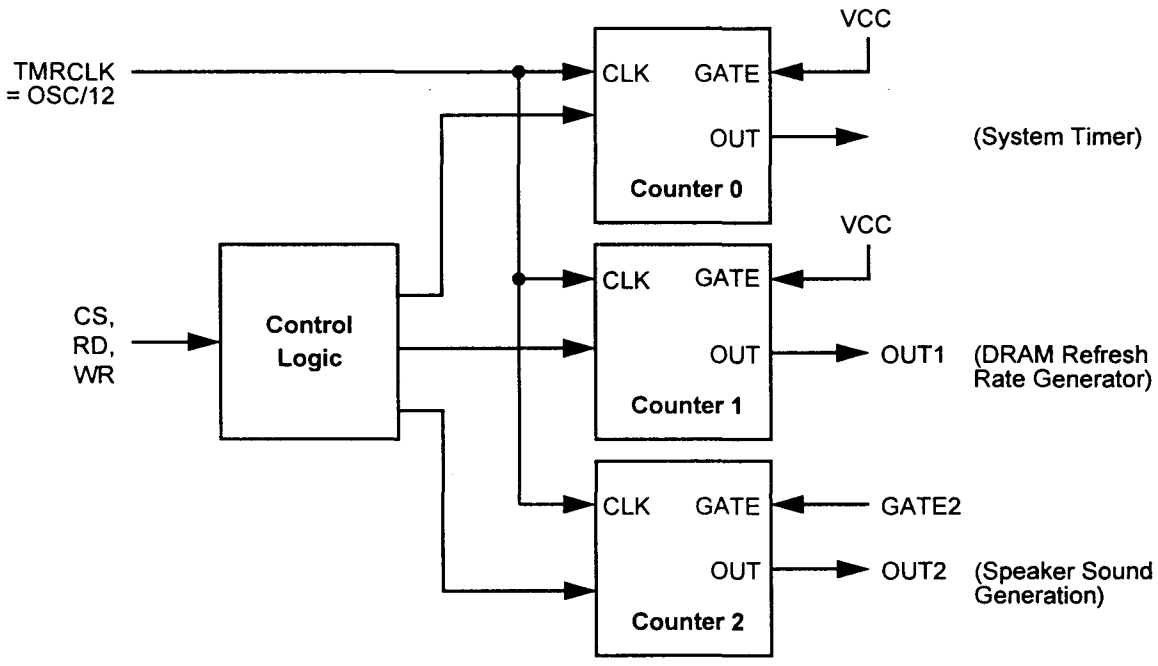
The internal timer counter use an internal signal TMRCLK which is derived from the OSC input of the IPC. For the sake of simplicity, all references to the timer counter clock will be TMRCLK in the following description. All three counters are driven from a common clock input, TMRCLK (TMRCLK = OSC/12) 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.

4.19.6.1 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 4-9 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).

4.19.6.1.1 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 4-12 lists the I/O address map used by the counter/timer subsystem.

Table 4-12 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

4.19.6.1.2 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.

4.19.6.1.3 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.

4.19.6.1.4 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.

4.19.6.2 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 IPC'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

4.19.6.2.1 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.

4.19.6.2.2 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.

4.19.6.2.3 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.

4.19.6.2.4 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.

4.19.6.2.5 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.

4.19.6.2.6 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".

4.19.6.2.7 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 4-13 details this operation.

Table 4-13 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	

5.0 82C895 Registers

There are two sets of Index Registers in the 82C895. The first main set of Index Registers are for the 82C895 and the second set of Index Registers are for the internal IPC (Integrated Peripherals Controller). The following sub-sections will summarize the 82C895's registers and explain how to access them.

5.1 Register Accessing

The 82C895's registers can be accessed by indexing I/O Registers 22h and 24h. I/O Register 22h contains the index of the register to be accessed and I/O Register 24h contains the data to be read from or written to.

The embedded IPC registers are accessed by indexing I/O Registers 22h and 23h. Index Register 01h should be set to the default value of C0h.

5.2 Control Register Descriptions

Table 5-1 is a summary of the 82C895's main set of control registers and the tables that follow explains each register's bits functions in detail.

Table 5-1 Control Registers Summary (Index Port 22h, Data Port 24h)

Index	Name	7	6	5	4	3	2	1	0
20h	Control Reg. 1	Revision of 82C895 (RO)		L2 cache burst wait state control for both reads and writes	TAG7 bit sampling	Single or multiple ALE during bus conversion cycle	Wait state control bit for standard AT bus cycles	Keyboard emulation reset control	Fast reset
21h	Control Reg. 2	Master mode byte swap	Reserved	DRAM parity check	External (L2) cache	Cache size		Cache lead-off wait state control for write and burst write cycles	Cache read leadoff cycle wait state control
22h	Shadow RAM Control Reg. 1	ROM (000F0000-000FFFFF) chip select	Reserved		R/W control for D0000h-DFFFFh shadow RAM area	R/W control for E0000h-EFFFFh shadow RAM area	Refresh type	Fast Gate A20	Slow refresh
23h	Shadow RAM Control Reg. 2	Shadow RAM at EX000h-EXFFFh area				Shadow RAM at DX000h-DXFFFh area			
24h	DRAM Control Reg. 1	SMI handler upload	DRAM configuration			Reserved	DRAM configuration		
25h	DRAM Control Reg. 2	MDIR#/ LMEM# (pin 62 functionality)	DRAM read cycle wait state control		DRAM write cycle wait state control	Reserved		AT clock selection	
26h	Shadow RAM Control Reg. 3	ROMCS# generation for write cycles to F0000h-FFFFFh	Shadow RAM copy enable for C0000h-EFFFFh	Shadow write protect at C0000h-CFFFFh	Reserved	Shadow RAM for C0000h-CFFFFh			
27h	Control Reg. 3	Global cache enable	Fast AT cycle	Back-to-back I/O delay control	L2 cache for F0000h-FFFFFh	Turbo bit	L1 write-back	AT clock change	Reserved

Control Registers Summary (Cont.)

Index	Name	7	6	5	4	3	2	1	0
28h	Non-Cacheable Block 1 Reg.	Size of non-cacheable memory block 1			Unused		Address bits of A[26:24] of non-cacheable memory block 1		
29h	Non-Cacheable Block 1 Reg. 2	Address bit A[23:16] of non-cacheable memory block 1							
2Ah	Non-Cacheable Block 2 Reg. 1	Size of non-cacheable memory block 2			Unused		Address bits of A[26:24] of non-cacheable memory block 2		
2Bh	Non-Cacheable Block 2 Reg. 2	Address bit A[23:16] of non-cacheable memory block 2							
Note: Index 2Ch is not decoded by the 82C895, hence this register should not be accessed.									
2Dh	ROMCS# Control Reg.	IRQ12 latching to support PS2 mouse controller	IRQ1 latching to support PS2 keyboard controller	ROMCS# control for areas C0000h to EFFFFh					
2Eh	Cacheable Addr Range 1	Used to control L1 cacheability for areas C0000h-FFFFFh				EC000h-EFFFFh cacheable in L2	E8000h-EBFFFh cacheable in L2	E4000h-E7FFFh cacheable in L2	E0000h-E3FFFh cacheable in L2
2Fh	Cacheable Addr Range 1	DC000h-DFFFFh cacheable in L2	D8000h-DBFFFh cacheable in L2	D4000h-D7FFFh cacheable in L2	D0000h-D3FFFh cacheable in L2	CC000h-CFFFFh cacheable in L2	C8000h-CBFFFh cacheable in L2	C4000h-C7FFFh cacheable in L2	C0000h-C3FFFh cacheable in L2

Table 5-2 Control Register 1 - Index: 20h

Bit(s)	Type	Default	Function
7:6	RO	00	Revision of 82C895 and is read-only.
5	R/W	0	L2 cache burst wait state control for both reads and writes. 0 = L2 cache cycle is X-1-1-1 1 = L2 cache cycle is X-2-2-2 Index Register 21h, bit 0, determines the leadoff cycle wait state for read bursts. Index Register 21h, bit 1 determines the leadoff cycle wait state for write bursts.
4	R/W	0	TAG7 bit sampling: This bit is used to enable or disable TAG7 sampling when MP2 is sampled low during reset. If MP2 is sampled high, this bit is ignored (takes no action). 0 = TAG7 enabled 1 = TAG7 disabled
3	R/W	0	The 82C895 will activate a single ALE instead of multiple ALEs during a bus conversion cycle if this bit is set. 0 = Multiple ALEs 1 = Single ALEs
2	R/W	0	Wait state control bit for standard AT bus cycles: 0 = No wait state 1 = One wait state
1	R/W	0	Keyboard emulation reset control: 0 = A CPU reset is generated only after executing a "halt" instruction following a write to Port 64h. 1 = A CPU reset is generated immediately after a write to Port 64h.
0	R/W	0	Fast reset: The 82C895 generates a CPU reset whenever a "halt" instruction is executed. 0 = Disable 1 = Enable

Table 5-3 Control Register 2 - Index: 21h

Bit(s)	Type	Default	Function
7	R/W	0	Master mode byte swap: 0 = Disable 1 = Enable
6	R/W	0	Reserved: This bit must be left at the default value = 0 (internally used for debugging purposes).
5	R/W	0	DRAM Parity check: 0 = Enable 1 = Disable
4	R/W	0	External (L2) cache: 0 = Disable 1 = Enable
3:2	R/W	00	Cache size: 00 = 64KB 10 = 256KB 01 = 128KB 11 = 512KB
1	R/W	0	Cache leadoff wait state control for write and burst write cycles. 0 = 1 wait state write (3-X-X-X) 1 = 0 wait state (2-X-X-X)
0	R/W	0	Cache read leadoff cycle wait state control: 0 = 3-X-X-X 1 = 2-X-X-X

Table 5-4 Shadow RAM Control Register 1 - Index: 22h

Bit(s)	Type	Default	Function
7	R/W	1	ROM (000F0000-000FFFFF) chip select: 0 = Read from write-protected DRAM. Writes are ignored. 1 = Read from ROM, write to DRAM. ROMCS# is generated during read access (see Register 26h, bit 7).
6:5	R/W	11	Reserved: These bits must be left at the default value = 1 (internally used for debugging purposes).
4	R/W	0	Read/write control for D0000h-DFFFFh shadow RAM area (write protect): 0 = Read/write 1 = Read only
3	R/W	0	Read/write control for E0000h-EFFFFh shadow RAM area (write protect): 0 = Read/write 1 = Read only
2	R/W	1	Refresh type: 0 = Hidden refresh: HOLD will not be asserted to the CPU during refresh. 1 = Conventional refresh: HOLD will be asserted to the CPU during refresh.
1	R/W	0	Fast Gate A20: This bit controls the A20 signal to the CPU. 0 = A20M# is controlled by keyboard emulation. 1 = A20M# pin is always high. No address wrap-around will exist above the 1MB boundary.
0	R/W	0	Slow refresh (four times slower than the normal refresh): 0 = Disable 1 = Enable (This feature can be enabled only when the DRAMs support the slow refresh feature.)

Table 5-5 Shadow RAM Control Register 2 - Index: 23h

Bit(s)	Type	Default	Function
7:4	R/W	0000	These bits enables shadow RAM for the E0000h-EFFFFh segment with a 16KB granularity. If set to 0, the cycles are forwarded to the ISA bus. If set to 1, the corresponding area in the shadow RAM will be enabled. 0 = Disable shadow 1 = Enable shadow Bit 7 = EC000h-EFFFFh Bit 5 = E4000h-E7FFFh Bit 6 = E8000h-EBFFFh Bit 4 = E0000h-E3FFFh
3:0	R/W	0000	These bits enables shadow RAM for the D0000h-DFFFFh segment with a 16KB granularity. If set to 0, the cycles are forwarded to the ISA bus. If set to 1, the corresponding area in the shadow RAM will be enabled. 0 = Disable shadow 1 = Enable shadow Bit 3 = DC000h-DFFFFh Bit 1 = D4000h-D7FFFh Bit 2 = D8000h-DBFFFh Bit 0 = D0000h-D3FFFh

Table 5-6 DRAM Control Register 1 - Index: 24h

Bit(s)	Type	Default	Function
7	R/W	0	SMI handler upload: This bit, when set, will remap the 3XXXX memory to BXXXX memory range. This bit should always be set to write the handler in SMM memory space. 0 = Normal Mode 1 = Remaps memory 3XXXXh-BXXXXh This bit should be set only when loading the SMI handler.
6:4	R/W	000	DRAM configuration: See Table 4-5 "DRAM Configurations".
3	R/W	0	Reserved: Always set this bit to 0 (internally used for debugging purposes).
2:0	R/W	000	DRAM configuration: See Table 4-5 "DRAM Configurations".

Table 5-7 DRAM Control Register 2 - Index: 25h

Bit(s)	Type	Default	Function
7	R/W	0	MDIR#/LMEM# (pin 62 functionality): If MP3 is sampled low during reset, this bit defines pin 62 as follows: 0 = MDIR# 1 = LMEM#
6:5	R/W	11	DRAM read cycle wait state control: **6 5 DRAM Burst Add'l Write Wait State Default** 0 0 3-2-2-2 33MHz or less 0 1 4-3-3-3 Page-miss = 1 add'l ws 1 0 4-3-3-3 Page-miss = 0 add'l ws 1 1 5-4-4-4 50MHz
4	R/W	1	DRAM write cycle wait state control: 0 = 3-2-2-2 1 = 4-3-3-3
3:2	R/W	11	Reserved: Must be set to 11 (internally used for debugging purposes).
1:0	R/W	00	ATCLK selection: 00 = ATCLK = CLKI/6 01 = ATCLK = CLKI/5 10 = ATCLK = CLKI/4 11 = ATCLK = CLKI/3 **Note:** In ATCLK = CLKI/6 setting, the ADS# signal will be delayed by one CLK inside the 82C895. In this setting there will be a one CLK penalty in all cache/DRAM timings.

Table 5-8 Shadow RAM Control Register 3 - Index: 26h

Bit(s)	Type	Default	Function
7	R/W	0	ROMCS# generation for write cycles: 0 = Disable 1 = Enable (Generates ROMCS# for write cycles to support flash ROMs)
6	R/W	0	Shadow RAM copy enable for address area C0000h-EFFFFh: This bit, when set, will allow reads in this range to be executed out of ROM on the ISA bus and will be written to the DRAM. 0 = Disable - Will not copy contents to shadow RAM. 1 = Enable - Copy to shadow RAM.
5	R/W	0	Read/write control for C0000h-CFFFFh shadow RAM area (write protect): 0 = Read/write 1 = Read only
4	R/W	1	Reserved: This bit should always be set to 1.
3:0	R/W	0000	These bits enables shadow RAM for the C0000h-CFFFFh segment with a 16KB granularity. If set to 0, the cycles are forwarded to the ISA bus. If set to 1, the corresponding area in the shadow RAM will be enabled. (Bit 4 must be set.) 0 = Disable shadow 1 = Enable shadow Bit 3 = CC000h-CFFFFh Bit 1 = C4000h-C7FFFh Bit 2 = C8000h-CBFFFh Bit 0 = C0000h-C3FFFh

Table 5-9 Control Register 3 - Index: 27h

Bit(s)	Type	Default	Function
7	R/W	1	Global cache enable: This bit determines whether all cycles are cacheable in L1 and L2 cache. 0 = Enable L1 and L2 cache 1 = Disable L1 and L2 cache
6	R/W	1	Fast AT cycle: Determines when the AT cycle will begin. ADS# to BALE delay will be shortened by one ATCLK# when enabled. 0 = Disable - Cycle will be normal. 1 = Enable - Cycle will be shortened by one ATCLK.
5	R/W	0	Back-to-back I/O delay control: 0 = Three ATCLKs inserted on back-to-back I/O delay 1 = No back-to-back I/O delay
4	R/W	1	L2 cache for F0000h-FFFFFh area: This bit controls the write protection of L2 cache to F0000h-FFFFFh area. This is effective only when the F0000h-FFFFFh region is shadowed. 0 = Write enable 1 = Write protected
3	R/W	1	Turbo bit: When in Non-turbo Mode, a HOLD will be asserted on every other refresh. 0 = Non-turbo Mode (slow) 1 = Normal Mode (fast)
2	R/W	1	L1 write-back: This bit will sample the HITM# signal from the CPU and allow burst writes. 0 = CPU in write-back mode 1 = CPU in write-through mode
1	R/W	1	AT clock change: The AT bus clock can be set to either the synchronous mode, which is controlled by Index 25h[1:0] or the asynchronous mode which is OSC/2. 0 = Asynchronous mode 1 = Synchronous mode When the 82C895 goes into the power management mode, ATLCK will always be OSC/2.
0	R/W	0	Reserved

Table 5-10 Non-Cacheable Block 1 Register - Index: 28h

Bit(s)	Type	Default	Function
7:5	R/W	111	Size of non-cacheable memory block 1: 000 = 64KB 001 = 128KB 010 = 256KB 011 = 1MB 1xx = Disabled
4:3	RO	11	Unused
2:0	R/W	000	Address bits of A[26:24] of non-cacheable memory block 1

This register is used in conjunction with Index Register 29h to define a non-cacheable block. The starting address for the non-cacheable block must have the same granularity as the block size. For example, if a 512KB non-cacheable block is selected, its starting address is a multiple of 512KB; consequently, only address bits of A[23:19] are significant, A[18:16] are "don't care".

Table 5-11 Non-Cacheable Block 1 Register 2 - Index: 29h

Bit(s)	Type	Default	Function
7:0	R/W	0001 xxxx	Address bits A[23:16] of non-cacheable memory block 1:

	Valid Starting Address Bits							
Block Size	**A23**	**A22**	**A21**	**A20**	**A19**	**A18**	**A17**	**A16**
64KB	V	V	V	V	V	V	V	V
128KB	V	V	V	V	V	V	V	x
256KB	V	V	V	V	V	V	x	x
1MB	V	V	V	V	V	x	x	x

x = Don't care
V = Valid bit

Table 5-12 Non-Cacheable Block 2 Register 1 - Index: 2Ah

Bit(s)	Type	Default	Function
7:5	R/W	111	Size of non-cacheable memory block 2: 000 = 64KB 001 = 128KB 010 = 256KB 011 = 1MB 1xx = Disabled
4:3	R/W	00	Unused
2:0	R/W	000	Address bits of A[26:24] of non-cacheable memory block 2

This register is used in conjunction with Index Register 2Bh to define a non-cacheable block. The starting address for the non-cacheable block must have the same granularity as the block size. For example, if a 512KB non-cacheable block is selected, its starting address is a multiple of 512KB; consequently, only address bits of A[23:19] are significant, A[18:16] are "don't care".

Table 5-13 Non-Cacheable Block 2 Register 2 - Index: 2Bh

Bit(s)	Type	Default	Function
7:0	R/W	0001 xxxx	Address bits A[23:16] of non-cacheable memory block 2:

	Valid Starting Address Bits							
Block Size	**A23**	**A22**	**A21**	**A20**	**A19**	**A18**	**A17**	**A16**
64KB	V	V	V	V	V	V	V	V
128KB	V	V	V	V	V	V	V	x
256KB	V	V	V	V	V	V	x	x
1MB	V	V	V	V	V	x	x	x

x = Don't care
V = Valid bit

Note: Index 2Ch is not decoded by the 82C895, hence this register can not be accessed.

Table 5-14 ROM Chip Select (ROMCS#) Control Register - Index: 2Dh

Bit(s)	Type	Default	Function
7	R/W	1	IRQ12 latching to support PS2 mouse controller: When enabled, the latch will be cleared on I/O reads to Port 60h. 0 = IRQ12 latched by 82C895 1 = IRQ12 not latched by 82C895
6	R/W	1	IRQ1 latching to support PS2 keyboard controller: When enabled, the latch will be cleared on I/O reads to Port 60h. 0 = IRQ1 latched by 82C895 1 = IRQ1 not latched by 82C895
5:0	R/W	000000	ROMCS# control for areas C0000h to EFFFFh with a 32KB granularity. If set to 1, ROMCS# will be asserted for the correspond area when reading this area. For write cycles, if Register 26h, bit 7 is enabled then ROMCS# will be asserted for the corresponding area. If set to 0, ROMCS# will not be asserted. 0 = Disable 1 = Enable Bit 5 = E8000h-EFFFFh Bit 2 = D0000h-D7FFFh Bit 4 = E0000h-E7FFFh Bit 1 = C8000h-CFFFFh Bit 3 = D8000h-DFFFFh Bit 0 = C0000h-C7FFFh

Table 5-15 Cacheable Address Range 1 Register - Index 2Eh

Bit(s)	Type	Default	Function
7:4	R/W	0000	These bits are used to control L1 cacheability for areas C0000h-FFFFFh. If set to 1, the corresponding area will be cached in the L1 cache (CPU internal cache). If set to 0, the corresponding area will not be cacheable in the L1 cache. 0 = Disable 1 = Enable Bit 7 = F0000h-FFFFFh Bit 5 = D0000h-DFFFFh Bit 6 = E0000h-EFFFFh Bit 4 = C0000h-CFFFFh
3:0	R/W	0000	These bits are used to control the L2 cacheability of the E0000h-EFFFFh region. This region will be cacheable in L2, only if shadow is enabled. This region is write protected if write protection is enabled for this region in Index Register 22h, bit 3. 0 = Disable 1 = Enable Bit 3 = EC000h-EFFFFh Bit 1 = E4000h-E7FFFh Bit 2 = E8000h-EBFFFh Bit 0 = E0000h-E3FFFh

Table 5-16 Cacheable Address Range 2 Register - Index 2Fh

Bit(s)	Type	Default	Function
7:4	R/W	0000	These bits are used to control the L2 cacheability of the D0000h-DFFFFh region. This region will be cacheable in L2, only if shadow is enabled. This region is write protected if write protection is enabled for this region in Index Register 22h, bit 4. 0 = Disable 1 = Enable Bit 7 = DC000h-DFFFFh Bit 5 = D4000h-D7FFFh Bit 6 = D8000h-DBFFFh Bit 4 = D0000h-D3FFF
3:0	R/W	0000	These bits are used to control the L2 cacheability of the C0000h-CFFFFh region. This region will be cacheable in L2, only if shadow is enabled. This region is write protected if write protection is enabled for this region in Index Register 26h, bit 5. 0 = Disable 1 = Enable Bit 3 = CC000h-CFFFFh Bit 1 = C4000h-C7FFFh Bit 2 = C8000h-CBFFFh Bit 0 = C0000h-C3FFFh

5.3 Power Management Registers Description

Table 5-17 is a summary of the 82C895's power management registers and the tables that follow explains each register's bits functions in detail.

Table 5-17 Power Management Registers Summary (Index Port 22h, Data Port 24h)

Index	Name	7	6	5	4	3	2	1	0
E0h	PMU Timer & Pwr Mgmt Port Enable Reg.	Power management mode selection	PPEN# generation for GREEN Mode	SMI# generation for GREEN Mode	IRQ15 for CPUs not supporting SMI_GREEN functionality	Power management status	Timer enable/status bit	EPMI# enable/status bit	Forced GREEN enable/status bit
E1h	PMU Mode Event Timer 1	Any accesses to the CPU interrupt vector table will be considered as a wake-up event	Reload GET when EPMI# transitions	EPMI# pulse width	EPMI# pulse polarity	Forced GREEN	GREEN event timer selection		
E2h	GREEN Mode Event Timer 2	IRQ7 monitor	IRQ6 monitor	IRQ5 monitor	IRQ4 monitor	IRQ3 monitor	Reserved	IRQ1 monitor	Reserved
E3h	GREEN Mode Event Timer 2	IRQ15 monitor	IRQ14 monitor	Reserved	IRQ12 monitor	IRQ11 monitor	IRQ10 monitor	IRQ9 monitor	IRQ8 monitor
E4h	DRQ Detection	DRQ7 monitor	DRQ6 monitor	DRQ5 monitor	Reserved	DRQ3 monitor	DRQ2 monitor	DRQ1 monitor	DRQ0 monitor
E5h	Video/ Hard & Floppy Disk Monitor	Programmable I/O range 0	Video access A0000-BFFFF detection	I/O Port 3B0h-3DFh video detection	I/O Port 1F0h-1F7h and 3F6h hard disk detection	I/O Port 3F5h floppy detection	I/O Port 60h and 64h keyboard detection	LDEV# detection	LREQ# detection
E6h	Programmable I/O Addr Detection	Programmable I/O Port Address A[7:0]							
E7h	Programmable I/O Range Detection	Reserved	Mask Bits			Reserved		Programmable I/O port address A[9:8]	
E8h	Interrupt Trap Mask Reg. 1	Programmable interrupt vector mask area							
E9h	Interrupt Trap Mask Reg. 2	Programmable interrupt vector mask area							
EAh	GREEN Mode Config. Port	Reserved					GPP2# (HSYNC control)	GPP1# (VSYNC control)	GPP0# (CLKCNT)

Power Management Registers Summary (Cont.)

Index	Name	7	6	5	4	3	2	1	0
EBh	Return from GREEN Mode Config. Port	Reserved					NPP2# (HSYNC control)	NPP1# (VSYNC control)	NPP0# (CLKCNT)
ECh	Scratch Reg. 1	Scratch Register							
EDh	Scratch Reg. 2	Scratch Register							
EEh	Index Reg.	Local master wake-up	DMA request to generate PPEN#	Reserved	NESTED_ GREEN operation in GREEN Mode	Reserved			SMI# generation for a wake-up event
EFh	Mode Reg.	Reserved	Reserved	Reserved	Reserved	82C601 or 82C602 Mode	Reserved		

Table 5-18 PMU Timer and Power Management Port Enable Register - Index E0h

Bit(s)	Type	Default	Function
7	R/W	0	Power management mode selection: (See Note) 0 = AUTO_GREEN Mode (Non-SLe CPU system) 1 = SMI_GREEN Mode (SLe CPU system)
6	R/W	0	PPEN# generation for GREEN Mode: (See Note) 0 = Disable 1 = Enable
5	R/W	0	SMI# generation for GREEN Mode: (See Note) 0 = Disable 1 = Enable
4	R/W	0	This bit can be enabled to use IRQ15 generation for SMI_GREEN functionality. This feature can be used with CPUs not supporting the power management function. 0 = Disable 1 = Enable
3	RO	0	Power management status: 0 = System in GREEN Mode 1 = System in NORMAL Mode This bit will be changed dynamically by the power management logic.
2	R/W	0	Timer enable/status bit: When written to (timer enable): 0 = Disable - Timer time-out will not trigger system into GREEN Mode 1 = Enable - Timer time-out will trigger system into GREEN Mode When read from (status bit for timer): 0 = Timer did not cause system to go to GREEN Mode 1 = Timer caused system to go to GREEN Mode The BIOS should read the bit to identify if it is a GREEN or wake-up event. If it is a GREEN event, clear the bit by writing 0. If it is a wake-up event, then resume NORMAL operation.
1	R/W	0	EPMI# enable/status bit: When written to (enable EPMI#): 0 = Disable - EPMI# will not trigger system into GREEN Mode 1 = Enable - EPMI# will trigger system into GREEN Mode When read from (status bit for EPMI#): 0 = EPMI# did not cause system to go to GREEN Mode 1 = EPMI# caused system to go to GREEN Mode The BIOS should read the bit to identify if it is a GREEN or wake-up event. If it is a GREEN event, clear the bit by writing 0. If it is a wake-up event, then resume NORMAL operation.
0	R/W	0	Forced GREEN enable/status bit: When written to (enable bit for software GREEN): 0 = Disable - Forced GREEN Mode 1 = Enable - Forced GREEN Mode When read from (status bit for software GREEN): 0 = Forced GREEN did not cause system to go to the GREEN Mode 1 = Forced GREEN caused system to go to the GREEN Mode The BIOS should read the bit to identify if it is a GREEN or wake-up event. If it is a GREEN event, clear the bit by writing 0. If it is a wake-up event, then resume NORMAL operation.

Note: Power Management Mode Selection:

Function	Bit 7	Bit 6	Bit 5
82C601 Auto_GREEN	0	1	0
82C602 Auto_GREEN	0	1	0
82C601 SMI_GREEN	1	1	1
82C602 SMI_GREEN	1	0	1

Table 5-19 PMU Mode Event Timer 1 - Index E1h

Bit(s)	Type	Default	Function
7	R/W	0	Any accesses to the CPU interrupt vector table will be considered as a wake-up event. The vector can be masked using E8h and E9h. 0 = Disable 1 = Enable
6	R/W	0	Reload GET when EPMI# transitions: If set any EPMI# will both reload the GET and be used as an activity to return to the NORMAL Mode. When cleared, any EPMI# will not reload the GET or return the system to NORMAL Mode (while in GREEN Mode). 0 = Disable, an EPMI# will not reload GET or send system into NORMAL Mode. 1 = Enable, an EPMI# reloads the GET (during NORMAL Mode) and returns system to NORMAL Mode (during GREEN Mode)
5	R/W	0	EPMI# pulse width: This feature can be used for EPMI# switch debouncing. 0 = At least 100ns 1 = At least 5ms
4	R/W	0	EPMI# pulse polarity: 0 = Active low 1 = Active high
3	R/W	0	Forced GREEN: If bit 0 of Index Register E0h is enabled, a 1 will trigger the system into the forced GREEN Mode and a 0 takes no action.
2:0	R/W	000	GREEN event timer selection. 000 = 15 seconds 100 = 30 minutes 001 = 2 minutes 101 = 45 minutes 010 = 5 minutes 110 = 60 minutes 011 = 15 minutes 111 = 240 minutes

The power management unit operates based on a Green Event Timer (GET) which is programmed through bits 2:0, Index Register E1h.

Table 5-20 GREEN Mode Event Timer 2 - Index E2h

Bit(s)	Type	Default	Function
7:3, 1	R/W	000000	In the GREEN Mode, if any of these bits are enabled and the corresponding IRQ is detected, a wake-up event will be generated. A wake-up event always reloads the GET and asserts PPEN#. In the NORMAL Mode, if any of these bits are enabled and the corresponding IRQ is detected, the GET will be reloaded. 0 = Disable 1 = Enable Bit 7 = IRQ7 Bit 4 = IRQ4 Bit 6 = IRQ6 Bit 3 = IRQ3 Bit 5 = IRQ5 Bit 1 = IRQ1
2, 0	R/W	00	Reserved: Must always = 00 (internally not used).

Table 5-21 GREEN Mode Event Timer 2 - Index E3h

Bit(s)	Type	Default	Function
7, 6, 4:0	R/W	000000	In the GREEN Mode, if any of these bits are enabled and the corresponding IRQ is detected, a wake-up even will be generated. A wake-up event always reloads the GET and asserts PPEN#. In the NORMAL Mode, if any of these bits are enabled and the corresponding IRQ is detected, the GET will be reloaded. 0 = Disable 1 = Enable Bit 7 = IRQ15 Bit 2 = IRQ10 Bit 6 = IRQ14 Bit 1 = IRQ9 Bit 4 = IRQ12 Bit 0 = IRQ8 Bit 3 = IRQ11
5	R/W	0	Reserved

Table 5-22 DRQ Detection Register - Index E4h

Bit(s)	Type	Default	Function
7:5, 3:0	R/W	000000	In the GREEN Mode, if these bits are enabled, the corresponding DRQ will reload the GET. If EEh[6] is also enabled, PPEN# will be generated. In the NORMAL Mode, if any of these bits are enabled, the corresponding DRQ will reload the GET. 0 = Disable 1 = Enable Bit 7 = DRQ7 Bit 2 = DRQ2 Bit 6 = DRQ6 Bit 1 = DRQ1 Bit 5 = DRQ5 Bit 0 = DRQ0 Bit 3 = DRQ3
4	R/W	0	Reserved

Table 5-23 Video/Hard & Floppy Disk Monitor - Index E5h

Bit(s)	Type	Default	Function
7	R/W	0	Programmable I/O range 0: In the GREEN Mode, if this bit is enabled and the I/O port address range specified by Index E6h and the lowest bits of Index E7h is detected, a wake-up event is generated which reloads the GET and PPEN# is generated. In the NORMAL Mode, the I/O port address range specified by Index E6h and the lowest bits of Index E7h is detected, only the GET is reloaded. 0 = Disable 1 = Enable
6:2	R/W	0	In the GREEN Mode, if any of these bits are enabled and an access is detected in the corresponding area, a wake-up event is generated which reloads the GET and PPEN# is generated. In the NORMAL Mode, if any of these bits are enabled and an access is detected in the corresponding area, only the GET is reloaded. 0 = Disable 1 = Enable Bit 6 = A0000h-BFFFFh Bit 5 = I/O Port 3B0h-3DFh Bit 4 = I/O Port 1F0h-1F7h and 3F6h Bit 3 = I/O Port 3F5h Bit 2 = I/O Port 60h and 64h
1	R/W	0	In the GREEN Mode, if this bit is enabled and LDEV# are detected, a wake-up event is generated which reloads the GET and PPEN# is generated. In the NORMAL Mode, if this bit is enabled and LDEV# is detected, only the GET is reloaded. 0 = Disable 1 = Enable
0	R/W	0	In the GREEN Mode, if this bit and EEh[7] is enabled and LREQ# is detected, a wake-up event is generated which reloads the GET and PPEN# is generated. In the NORMAL Mode, if this bit is enabled and LREQ# is detected, only the GET is reloaded. 0 = Disable 1 = Enable

Table 5-24 Programmable I/O Address Detection - Index E6h

Bit(s)	Type	Default	Function
7:0	R/W	0000 0000	Programmable I/O port address: Bits A[7:0]. The range will be specified on a byte boundary. (See Index E7h for masking range.)

Table 5-25 Programmable I/O Range Detection- Index E7h

Bit(s)	Type	Default	Function
7	R/W	0	Reserved
6:4	R/W	000	Mask Bits: 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
3:2	R/W	00	Reserved
1:0	R/W	00	Programmable I/O port address: Bits A[9:8]. The range will be specified on a byte boundary.

Table 5-26 Interrupt Trap Mask Register 1 - Index E8h

Bit(s)	Type	Default	Function
7:0	R/W	0000 1000 (Corresponds to INT8)	Programmable interrupt vector mask area. Corresponds to bits A[9:2]. Masking may be for 0h to FFh vector areas. When set to 00h, INT0 vector will be masked. When set to 01h, INT0 vector will be masked; etc.

Table 5-27 Interrupt Trap Mask Register 2- Index E9h

Bit(s)	Type	Default	Function
7:0	R/W	0000 1000 (Corresponds to INT8)	Programmable interrupt vector mask area: Corresponds to bits A[9:2]. Masking may be for 0h to FFh vector areas.

Table 5-28 GREEN Mode Configuration Port - Index EAh

Bit(s)	Type	Default	Function
7:3	R/W	00000	Reserved
2	R/W	0	GPP2# (HSYNC control)
1	R/W	0	GPP1# (VSYNC control)
0	R/W	0	GPP0# (CLKCNT)

This port provides the GREEN state values for the 82C895/82C602 GPM (GREEN Power Management) Port. This register will transfer its information to Index FAh when PPEN# is strobed to go to the GREEN Mode. When Index F9h, bit 6, is cleared, the 82C602 will not load the GPM when PPEN# is strobed. When a GREEN event and PPEN# occurs, the contents of this register is placed on the lower SD bus.

Table 5-29 Return from GREEN Mode Configuration Port - Index EBh

Bit(s)	Type	Default	Function
7:3	R/W	11111	Reserved
2	R/W	1	NPP2# (HSYNC control)
1	R/W	1	NPP1# (VSYNC control)
0	R/W	1	NPP0# (CLKCNT)

This port provides the return from GREEN state values for the 82C895/82C602 GPM Port. This register will transfer its information to Index FAh when PPEN# is strobed to return to the NORMAL Mode. When Index F9h, bit 6, is cleared, the 82C602 will not load the GPM when PPEN# is strobed. When a GREEN event and PPEN# occurs, the contents of this register is placed on the lower SD bus.

Table 5-30 Scratch Register 1 - Index ECh

Bit(s)	Type	Default	Function
7:0	R/W	0000 0000	Scratch register: Used to store configuration information.

Table 5-31 Scratch Register 2 - Index EDh

Bit(s)	Type	Default	Function
7:0	R/W	0000 0000	Scratch register: Used to store configuration information.

Table 5-32 Index Register EEh

Bit(s)	Type	Default	Function
7	R/W	0	During a write operation: 0 = No action 1 = Enable local bus master request to generate PPEN# During a read operation: This bit is a flag which, when set, identifies that an event woke the system up. This bit is cleared when it is read.
6	WO	0	This bit allows a DMA request to generate PPEN#. 0 = No action 1 = Enable DMA request to generate PPEN#
5	R/W	0	Reserved: Must always = 0 (internally used for debugging purposes).
4	R/W	0	NESTED_GREEN Mode: The 82C895 allows multiple levels of GREEN events and a wake-up event will always make the system go into the NORMAL mode. 0 = Disable 1 = Enable
3:1	R/W	000	Reserved: Must always = 0 (internally used for debugging purposes).
0	R/W	0	Determines whether an SMI# will be generated for a wake-up event: 0 = Pin 160 is PPEN# 1 = Pin 160 is SMI# and PPEN#

Table 5-33 Mode Register - Index EFh

Bit(s)	Type	Default	Function
7	R/W	0	Reserved: Must always clear to 0.
6	R/W	0	Reserved: Must always be set to 1.
5:4	R/W	00	Reserved: Must always clear to 00.
3:2	R/W	00	82C601 or 82C602 Mode: 11 = 82C601 Mode - The 82C895 will always generate PPEN# and never generate an SMI# strobe directly. An external latch should be used to generate SMI#. 00 = 82C602 Mode - The 82C895 will generate an SMI# when the system goes from the NORMAL to GREEN Mode. 01 = Reserved 10 = Reserved
1:0	R/W	00	Reserved: Must always be set to **01** (internally used for debugging purposes).

5.4 82C895 Internal Peripherals Controller Register Descriptions

The internal IPC (Integrated Peripherals Controller) registers are accessed by indexing I/O Registers 22h and 23h. Index Register 01h should be set to the default value of C0h.

Following Table 5-34 are tables that explain the subsystem registers of the internal IPC of the 82C895.

Table 5-34 Configuration Register (Index Port 22h, Data Port 23h) - Index: 01h

Bit(s)	Type	Default	Function
7:6	R/W	11	These bits control the number of wait states inserted when the CPU accesses the registers of the IPC. 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)
5:4	R/W	00	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 82C895'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 (Default) 01 = Two 16-bit DMA wait states 10 = Three 16-bit DMA wait states 11 = Four 16-bit DMA wait states
3:2	R/W	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 82C895'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 (Default) 01 = Two 8-bit DMA wait states 10 = Three 8-bit DMA wait states 11 = Four 8-bit DMA wait states
1	R/W	0	This bit enables the early internal DMAMEMR# function. In a PC/AT-based system, DMAMEMR# is delayed one clock cycle later than SMEMR#. If set to 1, it will start DMAMEMR# at the time as SMEMR#. If set to 0, it will start DMAMEMR#.
0	R/W	0	If this bit is set to 0, the SYSCLK input is divided by two and is used to drive both 8- and 16-bit DMA subsystems. If this bit is set to 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.

5.4.1 DMA Subsystem Registers

Table 5-35 Command Register

Bit(s)	Type	Function
7	R/W	The setting of this bit determines if the DACK output pin will be active low or active high. 0 = Active low 1 = Active high
6	R/W	The setting of this bit determines if the DREQ input pin will be active low or active high. 0 = Active low 1 = Active high
5	R/W	Extended write: The extended write feature is controlled by this bit. When 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. 0 = Disabled 1 = Enabled
4	R/W	Rotating priority: This bit selects wether the priority scheme is fixed or rotating. 0 = Fixed (Default) 1 = Rotating
3	R/W	Compressed timing: This bit enables the Compressed Timing feature. 0 = Compressed Timing 1 = Normal Timing (Default)
2	R/W	Controller disable: 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. 0 = Enable 1 = Disable
1	R/W	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
0	R/W	Memory-to-Memory: This bit enables Channel 0 and 1 to be used for memory-to-memory transfers. 0 = Disable 1 = Enable

Table 5-36 Mode Register

Bit(s)	Type	Function
7:6	R/W	Mode select bits 1 and 0: These bits are used to select the mode for each channel. 00 = Demand Mode 10 = Block Mode 01 = Single Cycle Mode 11 = Cascade Mode
5	R/W	Decrement: Writing a 1 to this bit decrements the address after each transfer.
4	R/W	Auto-initialization: Writing a 1 to this bit enables the auto-initialization function.
3:2	R/W	Transfer type bits 1 and 0: These bits control the type of transfer to performed. 00 = Verify 10 = Read Transfer 01 = Write Transfer 11 = Illegal
1:0	R/W	Channel selection bits 1 and 0: These bits determine which channel's Mode Register will be written. Read back of a Mode Register will cause these bits to both be 1. 00 = Select Channel 0 10 = Select Channel 2 01 = Select Channel 1 11 = Select Channel 3

Table 5-37 Request Register Write Format

Bit(s)	Type	Function
7:3	W	Don't care
2	W	Request bit: Writing a 1 to this bit sets the request bit.
1:0	W	Request select bits 1 and 0: These bits determine which channel's request bit will be set. 00 = Select Channel 0 10 = Select Channel 2 01 = Select Channel 1 11 = Select Channel 3

Table 5-38 Request Register Read Format

Bit(s)	Type	Function
7:4	R	Reserved: Always reads 1.
3:0	R	Request channel bits 3 through 0: These bits contain the state of the request bit associated with each request channel. The bit position corresponds to the channel number.

Table 5-39 Request Mask Register Set/Reset Format

Bit(s)	Type	Function
7:3		Don't Care
2		Mask bit: Writing a 1 to this bit sets the request mask bit and inhibits external requests.
1:0		Mask select bits 1 and 0: These bits determine which channel's request bit will be set. 00 = Select Channel 0 10 = Select Channel 2 01 = Select Channel 1 11 = Select Channel 3

Table 5-40 Request Mask Register Read/Write Format

Bit(s)	Type	Function
7:4	R/W	Reserved: Always reads 1.
3:0	R/W	Mask Bits 3 through 0: These bits contain the state of the request mask bit associated with each request channel. The bit position corresponds to the channel number.

Table 5-41 Status Register

Bit(s)	Type	Function
7:4	R	Data Request bits 3 through 0: 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.
3:0	R	Terminal Count bits 3 through 0: 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.

5.4.2 Interrupt Controller Subsystem

Table 5-42 ICW1 Register - Address: 020h (0A0h)

Bit(s)	Type	Function
7:5	W	Don't Care
4	W	Must be set to 1 for ICW1 since ICW1, OCW2, and OCW3 share the same address, 020h (0A0h).
3	W	Level Trigger Mode: This bit selects either the Level Triggered Mode or Edge Triggered Mode input to the IR. 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 deasserted early.
2	W	Don't Care
1	W	Single Mode: This bit selects between the Single and Cascade Modes. 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.
0	W	Don't Care

Table 5-43 ICW2 Register - Address: 021h (0A1h)

Bit(s)	Type	Function
7:3	W	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.
2:0	W	Vector bits 2 through 0: The lower three bits of the vector are generated by the Priority Resolver during INTA

Table 5-44 ICW3 Register - Format for INTC1 - Address: 021h

Bit(s)	Type	Function
7:0	W	Slave Mode bits 7 through 0: 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.

Table 5-45 ICW3 Register - Format for INCT1 - Address: 0A1h

Bit(s)	Type	Function
7:3	W	Don't Care
2:0	W	Identify bits 2 through 0: 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.

Table 5-46 ICW4 Register - Address 021h (0A1h)

Bit(s)	Type	Function
7:5	W	Don't Care
4	W	Enable Multiple Interrupts: This bit will enable multiple interrupts from the same channel in the Fixed Priority Mode. 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.
3:2	W	Don't Care
1	W	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. Note this function should not be used in a device with fully nested interrupts unless the device is a cascade master type.
0	W	Don't Care

Table 5-47 OCW1 Register - Address: 021h (0A1h)

Bit(s)	Type	Function
7:0	R/W	Mask bits 7 through 0: 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.

Table 5-48 OCW2 Register - Address: 020h (0A0h)

Bit(s)	Type	Function
7:5	W	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. **7 6 5 Function** 0 0 0 Clear Rotate in Auto-EOI mode 0 0 1 Non-specific EOI Command 0 1 1 No Operation 0 1 1 Specific EOI Command* 1 0 0 Set Rotate in Auto-EOI Mode 1 0 1 Rotate on Non-specific EOI Command 1 1 0 Set Priority Command* 1 1 1 Rotate on specific EOI Command *L[2:0] are used by these commands.
4:3	W	These bits must be set to 0 to indicate that OCW2 is selected, because ICW1, OCW2, and OCW3 share the same address. 020h (0A0h).
2:0	W	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.

Table 5-49 OCW3 Register - Address 020h (0A0h)

Bit(s)	Type	Function
7	W	Reserved: This bits must be set to 0.
6:5	W	Enable Special Mask Mode: 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. **6 5 Function** 0 X No operation 1 0 Reset Special Mask Mode to Normal Mask Mode 1 1 Set Special Mask Mode
4:3	W	These bits must be set to 0 to indicate that OCW3 is selected because ICW1, OCW2, and OCW3 share the same address, 020h (0A0h).
2	W	Polled Mode: Writing a 1 to this bit of OCW3 enables the Polled Mode. 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.
1:0	W	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. **1 0 Function** 0 X No Operation 1 0 Read IRR on the next read 1 1 Read ISR on the next read

Table 5-50 IIR Register - Address: 020h (0A0h)

Bit(s)	Type	Function
7:0		Interrupt Request bits 7 through 0: 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.

Table 5-51 ISR Register - Address 020h (0A0h)

Bit(s)	Type	Function
7:0		Interrupt Service bits 7 through 0: 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 5-52 Poll Vector - Address 200h (0A0h)

Bit(s)	Type	Function
7		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.
6:3		Don't Care
2:0		Vector bits 2 through 0: 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.

5.4.3 Counter/Timer Subsystem

Table 5-53 Control Word Format (Write Only)

Bit(s)	Type	Function
7:6	W	Select counter bits 1 and 0: These bits select which counter this control word is written to. 00 = Select Counter 0 01 = Select Counter 1 10 = Select Counter 2 11 = Reserved for read-back command
5:4	W	Read/write bits 1 and 0: These bits determine the counter read/write word size. 00 = Reserved for counter latch command 01 = Read/write LSB only 10 = Read/write MSB only 11 = Read/write LSB first, then MSB
3:1	W	Mode select bits 2 through 0: These bits select the counter operating mode. 000 = Select Mode 0 001 = Select Mode 1 X10 = Select Mode 2 X11 = Select Mode 3 100 = Select Mode 4 101 = Select Mode 5
0	W	Binary coded decimal: During read/write counter commands control word writing, a 1 selects binary coded decimal count format. A 0 selects binary counting format. During read-back command word writing, this bit must be 0.

Table 5-54 Counter Latch Command Format (Write Only)

Bit(s)	Type	Function
7:6	W	Select counter bits 1 and 0: These bits select which counter is being latched. 00 = Select Counter 0 01 = Select Counter 1 10 = Select Counter 2 11 = Reserved for read-back command
5:4	W	These bits must be 0 for the counter latch command.
3:0	W	Don't care

Table 5-55 Read-Back Command Format (Write Only)

Bit(s)	Type	Function
7:6	W	These bits must be 1 for the read-back command
5	W	Latch count: A 0 in this bit will latch the count of the counting component of the selected counter(s);
4	W	Latch status: A 0 in this bit will latch the status information of the selected counter(s).
3:1	W	Counter select bits 2 through 0: These bits select which counter(s) the read-back command is applied to. 0XX = Select Counter 2 X0X = Select Counter 1 XX0 = Select Counter 0
0	W	Reserved: Write as 0.

Table 5-56 Status Format

Bit(s)	Type	Function
7	R	Out: This bit contains the state of the OUT signal of the counter.
6	R	Null Count: This bit contains the condition of the null count flag. 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.
5:4	R	Read/Write Word bits 1 and 0: 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.
3:1	R	Mode bits 2 through 0: These bits reflect the operating mode of the counter and are interpreted in the same manner as in the write control word format.
0	R	Binary Coded Decimal: This bit indicates the counting element is operating in binary format or BCD format.

5.5 I/O Port 60h

The 82C895 emulates the Port 60h and 64h registers of the keyboard controller, allowing the generation of a faster gate A20 signal. The sequence here is BIOS transparent, and there is no need for the modification of the current BIOS. The sequence involves writing data D1h to Port 64h, then writing data 02h to Port 60h.

Table 5-57 I/O Port 61h (Port B)

Bit(s)	Type	Function
7	R	System Parity Check: This bit indicates that an on-board RAM parity error has occurred. It can only be set if bit 2 (Parity Check Enable) = 0. This bit should be cleared by writing a 1 to bit 2.
6	R	I/O Channel Check: This bit indicates that a peripheral device is reporting an error. It can only be set if bit 3 (I/O Channel Check Enable) = 0. This bit should be cleared by writing a 1 to bit 3.
5	R	Timer OUT2 Detect: This bit indicates the current state of the OUT2 signal from the on-board timer.
4	R	Refresh Detect: This bit is tied to a toggle flip-flop which is clocked by REFRESH. It toggles the opposite state every time a refresh cycle occurs.
3	R/W	I/O Channel Check Enable: When this bit is set low, it allows an NMI to be generated if the IOCHCK# input is pulled low. Otherwise, the IOCHCK# input is ignored and can not generate an NMI.
2	R/W	Parity Check Enable: When this bit is set low, it allows parity errors from on-board RAM memory to cause an NMI. When high, on-board RAM parity errors will not cause an NMI.
1	R/W	Speaker Output Enable: This bit is gated with the output of Counter 2 from the on-board timer. When this bit is high, it allows the OUT2 frequency to be passed out on the SPKR pin. When low, the SPKR output is forced low.
0	R/W	Timer 2 Gate: This bit goes to the GATE2 input of the on-board timer to enable Counter 2 to produce a speaker frequency.

5.6 I/O Port 64h

82C895 I/O Port 64h emulates the register inside the keyboard controller by generating a fast reset pulse. Writing data FEh to Port 64h asserts the reset pulse. The pulse is generated immediately after an I/O write, if bit 1 of Index Register 20h is set. If AAh is written to Port 64h, the 82C895 will set the A20M# pin to 1.

5.7 Port 70h

The NMI generation is controlled through Port 70h. This is a write only port. If set to 1, NMI will be disabled and if set to 0, NMI will be enabled.

5.8 Port 92h

Port 92h is the System Controller Port A, PS/2 compatibility port. If bit 1 is set to 1, Fast GATEA20 will be generated. If bit 0 is set to 1, Fast Reset will be generated.

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	Min	Max	Unit
VCC	Supply Voltage		+6.5	V
VI	Input Voltage	-0.5	VCC +0.5	V
VO	Output Voltage	-0.5	VCC +0.5	V
TOP	Operating Temperature	0	+70	°C
TSTG	Storage Temperature	-40	+125	°C

6.2 DC Characteristics

TA = 0°C to +70°C, VCC = 5.0V ±5%

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	
VOH	Output High Voltage	2.4		V	
IIL	Input Leakage Current		10	µA	
IOZ	Tristate Leakage Current		10	µA	
CIN	Input Capacitance		10	pF	
COUT	Output Capacitance		10	pF	
ICC	Power Supply Current		100	mA	

6.3 AC Characteristics

Up to 50MHz - TA = 0°C to +70°C, VCC = 5V ±5%

6.3.1 Cache Timing

Symbol	Parameter	Min	Max	Unit	Condition
t1	CPU bus definition valid to BEOE#/BOOE# active delay (for 2-X-X-X leadoff cycles only)	10	20	ns	
t2	CLK↑ to BEOE#/BOOE# active delay	5	11	ns	
t3	CLK↑ to BEOE#/BOOE# inactive delay		22	ns	
t4	CLK↑ to BRDY# active delay	5	15	ns	
t5	CLK↑ to BRDY# inactive delay	5	15	ns	
t6	CPU bus definition valid to ECAWE#/OCAWE# active delay (for 2-X-X-X leadoff cycles only)	10	20	ns	
t7	CLK↓ to ECAWE#/OCAWE# active delay		15	ns	
t8	CLK↓ to ECAWE#/OCAWE# active delay (0 wait state write	15	20	ns	
t9	CLK↓ to ECAWE#/OCAWE# inactive delay (0 wait state write)		12	ns	
t10	CLK↓ to TAGW# active delay (for updating DIRTY bit)		15	ns	
t11	CLK↓ to TAGW# inactive delay (for updating DIRTY bit)	5	15	ns	
t12	CLK↑ to BRDY# active delay (for cache write cycles)	5	20	ns	
t13	CLK↑ to BRDY# inactive delay (for cache write cycles)	5	10	ns	
t14	CLK↑ to TAGW# active delay (for updating TAG)	5	15	ns	
t15	CLK↑ to TAGW# inactive delay (for updating TAG)	5	15	ns	
t16	CPU bus definition valid to BEA3/BEA2OA3 active delay		16	ns	
t17	CLK↑ to BEA3/BEA2OA3 inactive delay		10	ns	

6.3.2 DRAM Timing

Symbol	Parameter	Min	Max	Unit	Condition
t18	CLK↑ to CAS# active delay	5	15	ns	
t19	CLK↑ to CAS# inactive delay	5	20	ns	
t20	CLK↓ to CAS# active delay (for 3-2-2-2 DRAM burst cycles only)	5	15	ns	
t22	CLK↑ to RAS# active delay	5	20	ns	
t23	CLK↑ to RAS# inactive delay	5	15	ns	
t24	CLK↑ to column address valid delay	5	30	ns	
t25	CLK↑ to row address hold time	8	30	ns	

AC Characteristics (Cont.)

Symbol	Parameter	Min	Max	Unit	Condition
t26	CLK↑ to DWE# active delay	5	30	ns	
t27	CLK↑ to DWE# inactive delay	5	30	ns	
t28	CLK↑ to new row address delay	20	60	ns	
t29	RAS# precharge time	80		ns	
t30	CAS# precharge time	20		ns	
t31	CAS# active to RAS#[1:0] active delay (refresh)	25		ns	
t32	CAS# inactive to RAS#[1:0] inactive delay (refresh)	25		ns	
t33	RASx# active to RASx# active delay (during refresh)			ns	
t34	RASx# active to RASx# inactive delay (during refresh)			ns	

6.3.3 Cache with DRAM Timing

Symbol	Parameter	Min	Max	Unit	Condition
t35	MRD#/MWE# active to BEOE/BOOE# active delay		280	ns	
t36	MRD#/MWE# inactive to BEOE/BOOE# inactive delay		100	ns	
t37	MRD#/MWE# active to RAS# active delay		20	ns	
t38	MRD#/MWE# inactive to RAS# inactive delay		18	ns	
t39	MRD#/MWE# active to CAS# active delay	70	100	ns	
t40	MRD#/MWE# inactive to CAS# inactive delay		18	ns	
t41	MRD#/MWE# active to column address valid delay			ns	
t42	MRD#/MWE# active to row address valid delay			ns	
t43	MWE# active to ECAWE#/OCAWE# active delay	40	80	ns	
t44	MWE# active to DWE# active delay		20	ns	
t45	MWE# inactive to DWE# inactive delay		18	ns	

6.3.4 AT Bus Timing

Symbol	Parameter	Min	Max	Unit	Condition
t46	ATCLK↓ to ALE active delay	5	30	ns	
t47	ATCLK↑ to ALE inactive delay	5	30	ns	
t48	ATCLK↓ to CMD active delay (1 CMD delay)	5	30	ns	
t49	ATCLK↑ to CMD active delay (0 CMD delay)	5	30	ns	
t50	ATCLK↑ to CMD inactive delay	5	30	ns	
t51	M16# to ATCLK↑ setup time	8		ns	
t52	M16# to ATCLK↑ hold time	8		ns	
t53	IO16# to ATCLK↑ setup time	10		ns	
t54	IO16# to ATCLK↑ hold time	10		ns	
t55	CHRDY to ATCLK↑ setup time	12		ns	

AC Characteristics (Cont.)

Symbol	Parameter	Min	Max	Unit	Condition
t56	CHRDY to ATCLK↑ hold time	12		ns	
t57	ATCLK↓ to HOLD active delay	5	16	ns	
t58	ATCLK↑ to HOLD inactive delay	5	16	ns	
t59	ATCLK↑ to REF# active delay	8	30	ns	
t60	ATCLK↑ to REF# inactive delay	8	30	ns	
t70	ATCLK↑ to MEMRD# active delay	5	25	ns	
t71	ATCLK↑ to MEMRD# inactive delay	5	25	ns	

6.3.5 Reset Timing

Symbol	Parameter	Min	Max	Unit	Condition
t72	RST1# inactive to CPURST active delay			CLKI	
t73	CPURST active delay from CLK↑	4	20	ns	
t74	CPURST inactive delay from CLK↑	4	20	ns	

6.3.6 VL Timing

Symbol	Parameter	Min	Max	Unit	Condition
t75	LDEV# setup time to CLK↑	5		ns	
t76	LDEV# hold time to CLK↑	5		ns	
t77	KEN# active delay from CLK↑		15	ns	
t78	KEN# inactive delay from address		20	ns	
t79	RDYI# setup time to CLK↑	5		ns	
t80	RDYI# hold time to CLK↑	5		ns	

6.3.7 Address and Data Bus Timing

Symbol	Parameter	Min	Max	Unit	Condition
t81	D[31:0] valid to SD[15:0] valid delay		30	ns	
t82	D[31:0] invalid to SD[15:0] invalid delay		25	ns	
t83	D[31:0] valid to MP[3:0] valid delay		20	ns	
t84	D[31:0] invalid to MP[3:0] invalid delay		25	ns	
t85	A[9:0] valid to KBDCS# active delay		30	ns	
t86	A[9:0] invalid to KBDCS# inactive delay		30	ns	

6.3.8 L1 Write-back Timing

Symbol	Parameter	Min	Max	Unit	Condition
t87	CLK↑ to EADS# active delay	8	20	ns	
t88	CLK↑ to EADS# inactive delay	8	20	ns	

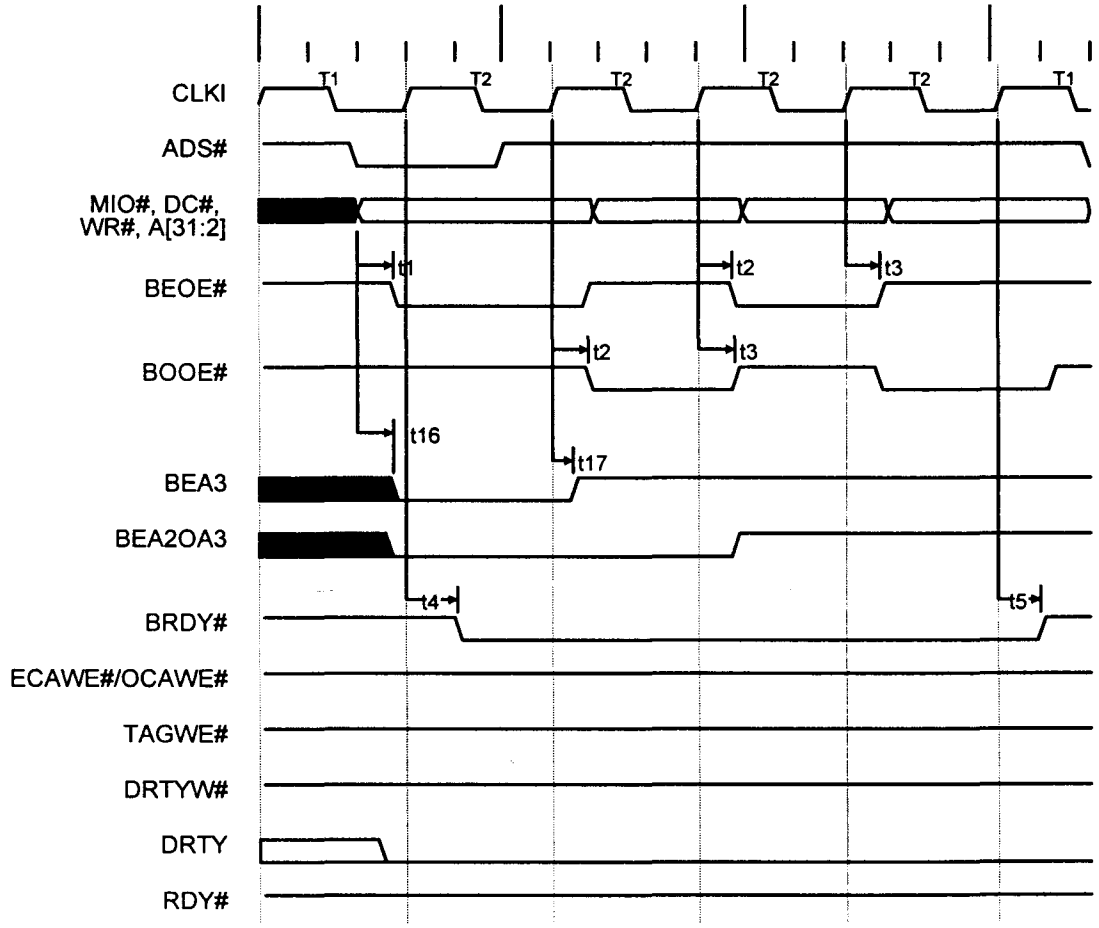
AC Characteristics (Cont.)

Symbol	Parameter	Min	Max	Unit	Condition
t89	HITM# to CLK↑ setup time	8		ns	
t90	CLK↑ to AHOLD active delay	8	20	ns	

Notes: 1. ↑ means rising edge
2. ↓ means falling edge
3. The capacitance loading is 50pf.

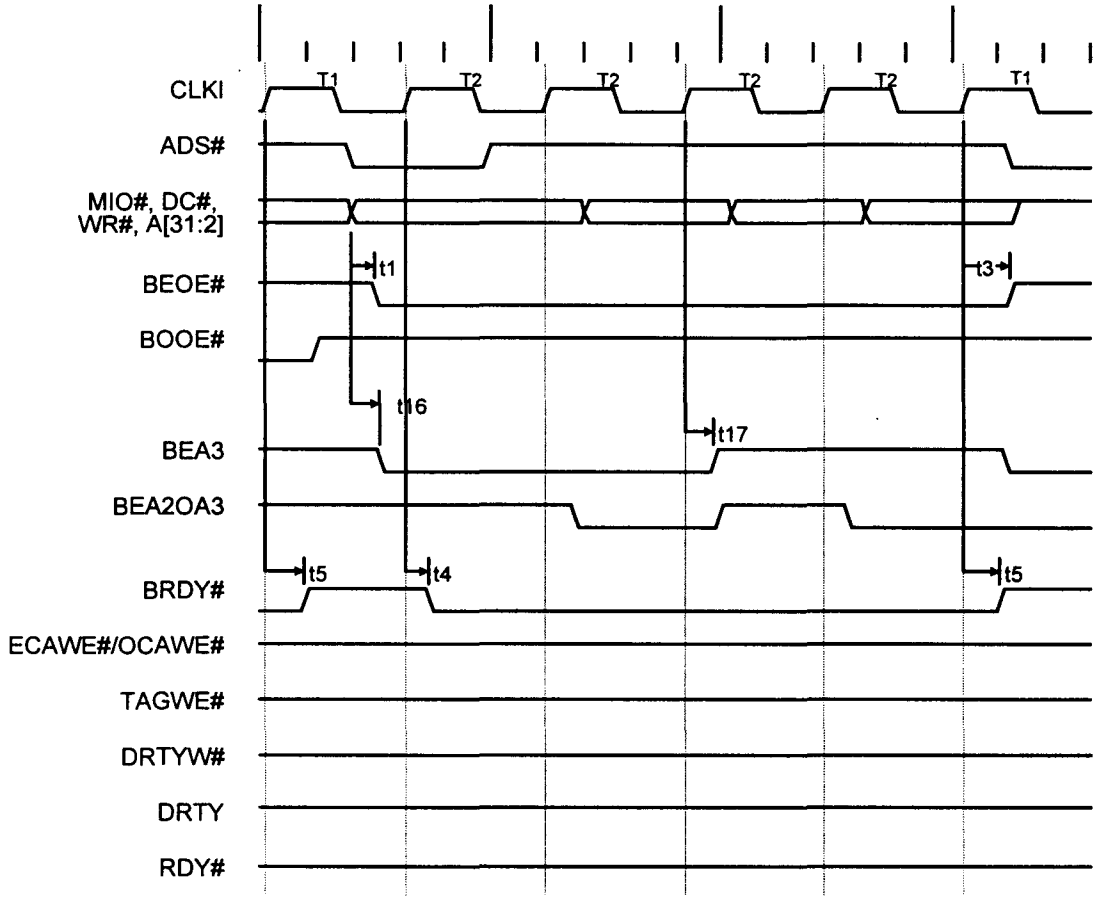
6.3.9 AC Timing Waveforms

Figure 6-1 Cache Read Hit



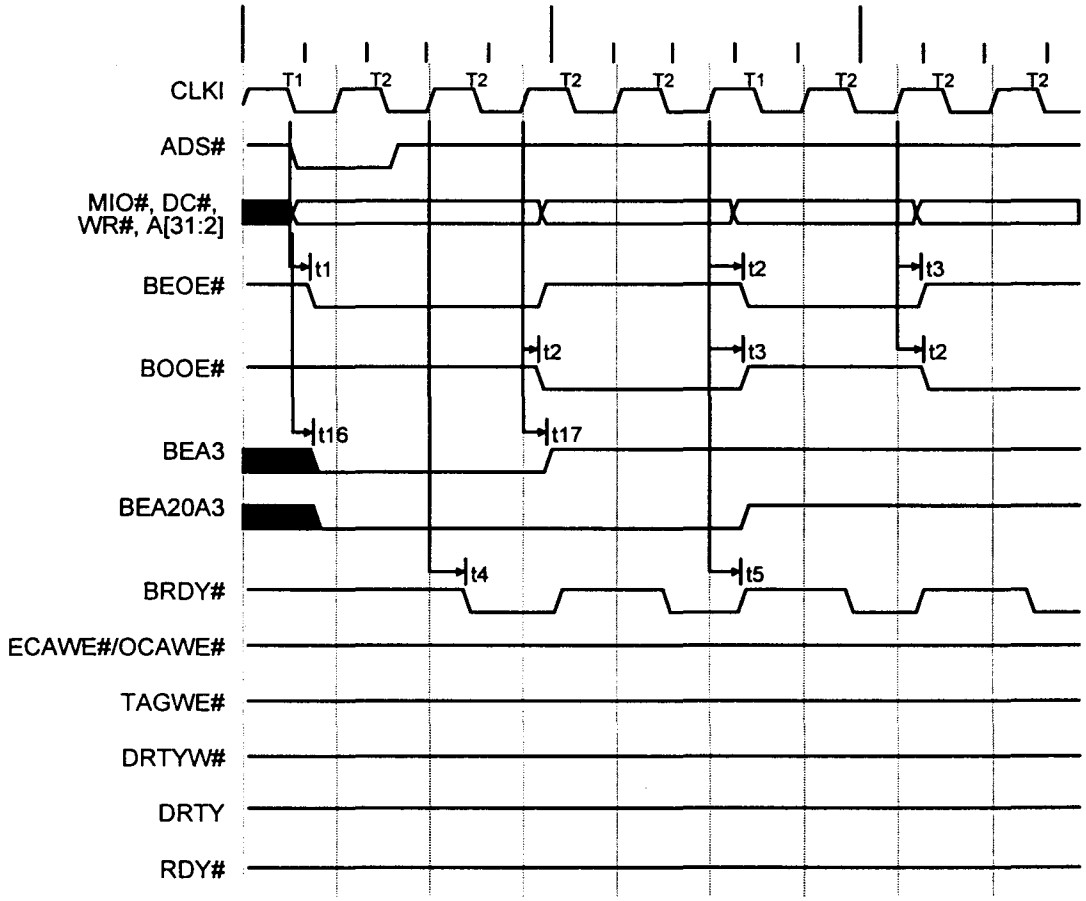
Note: Double Bank, 2-1-1-1 Cache Cycle

Figure 6-2 Cache Read Hit



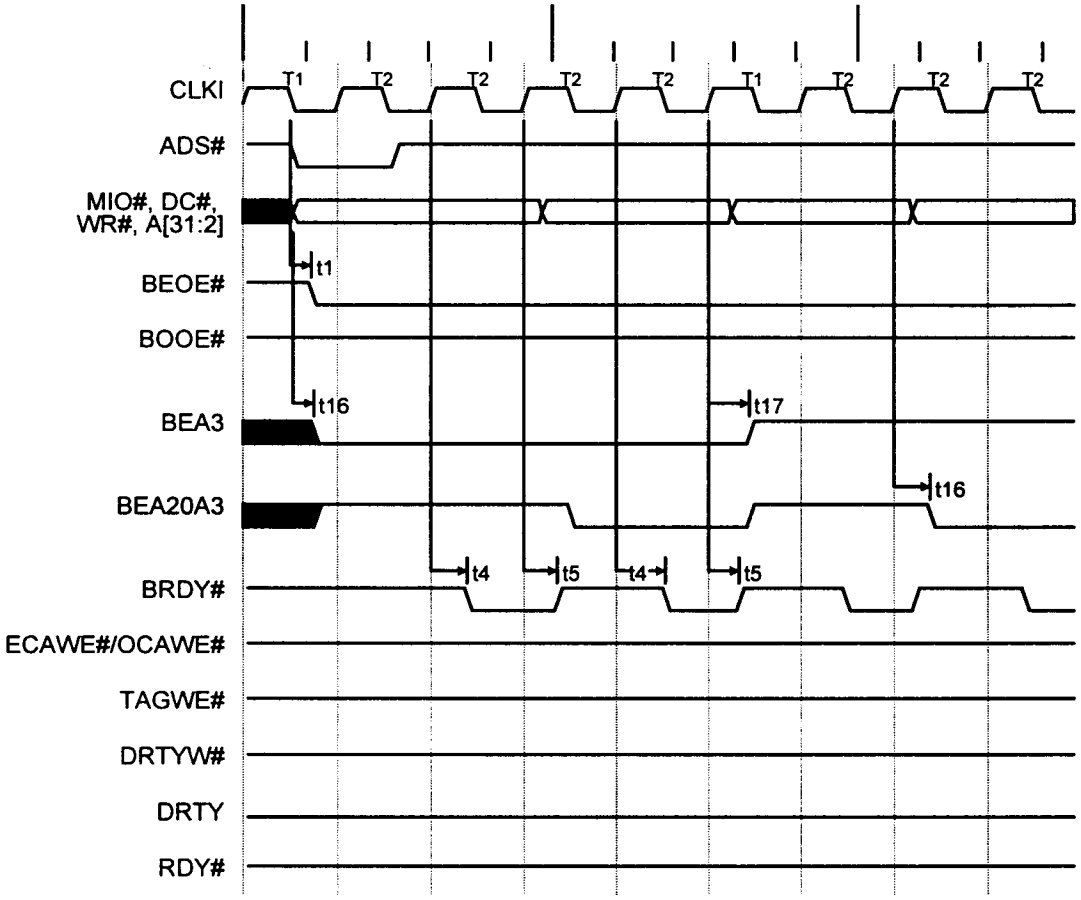
Note: Single Bank, 2-1-1-1 Cache Cycle

Figure 6-3 Cache Read Hit



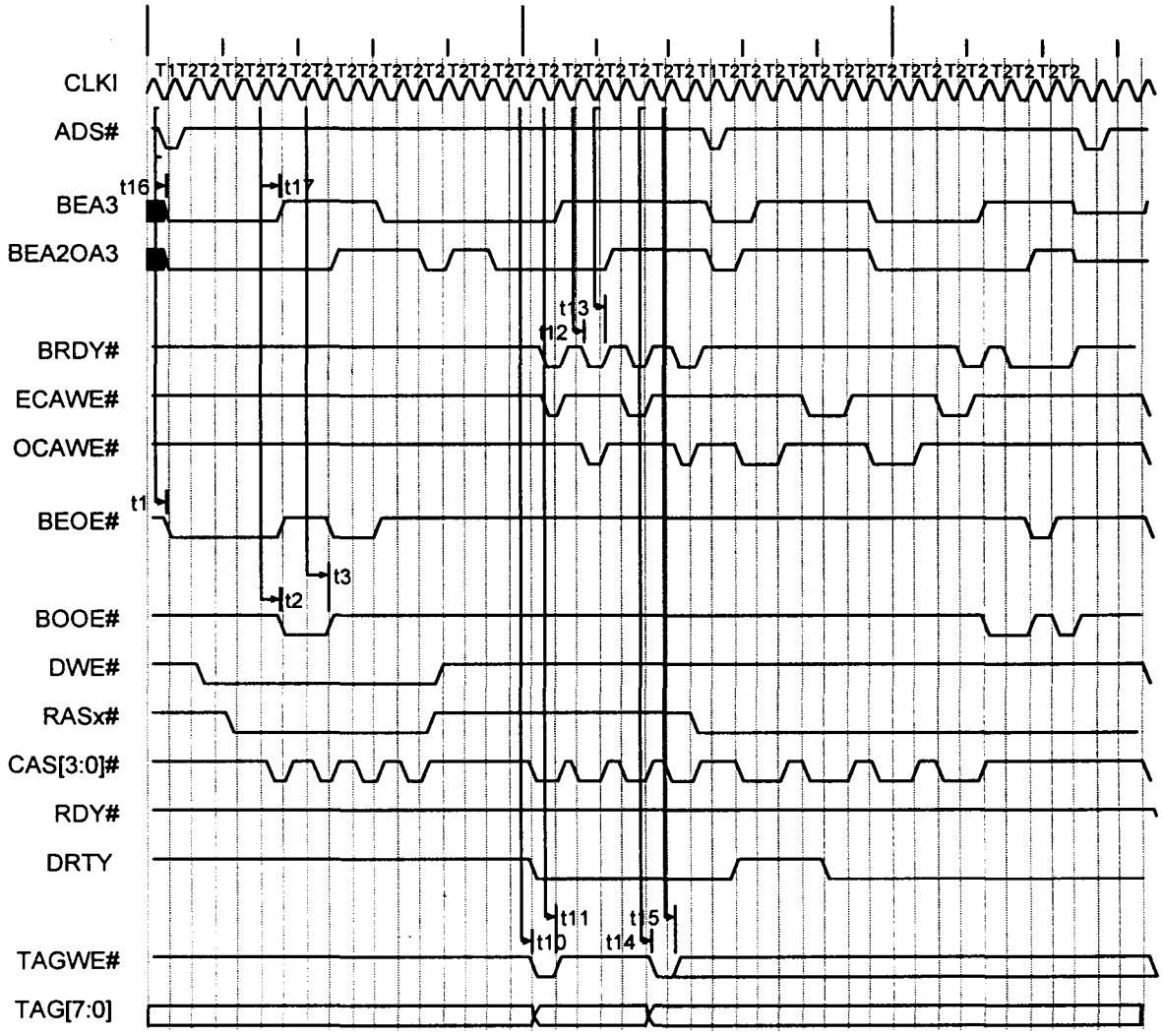
Note: Double Bank, 3-2-2-2 Cache Cycle

Figure 6-4 Cache Read Hit



Note: Single Bank, 3-2-2-2 Cache Cycle

Figure 6-5 Cache Read Miss (Dirty)



Note: Double Bank, 3-2-2-2 DRAM Read, 0 Wait State DRAM Write Cycle

Figure 6-6 Cache Write Hit (0WS/Not Dirty)

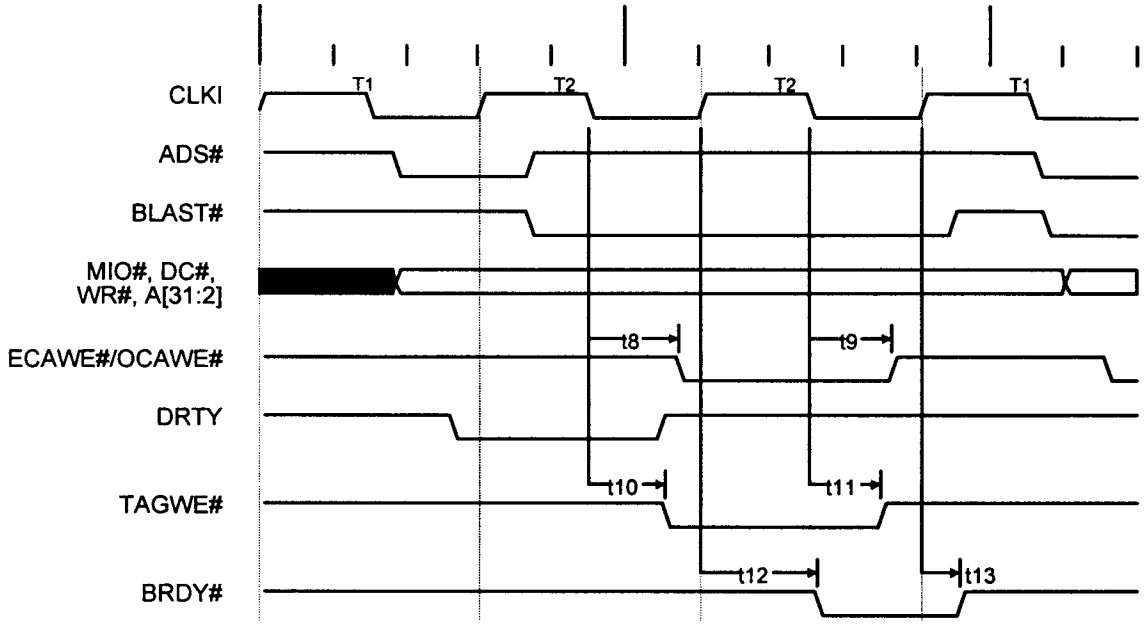


Figure 6-7 Cache Write Hit (0WS/Dirty)

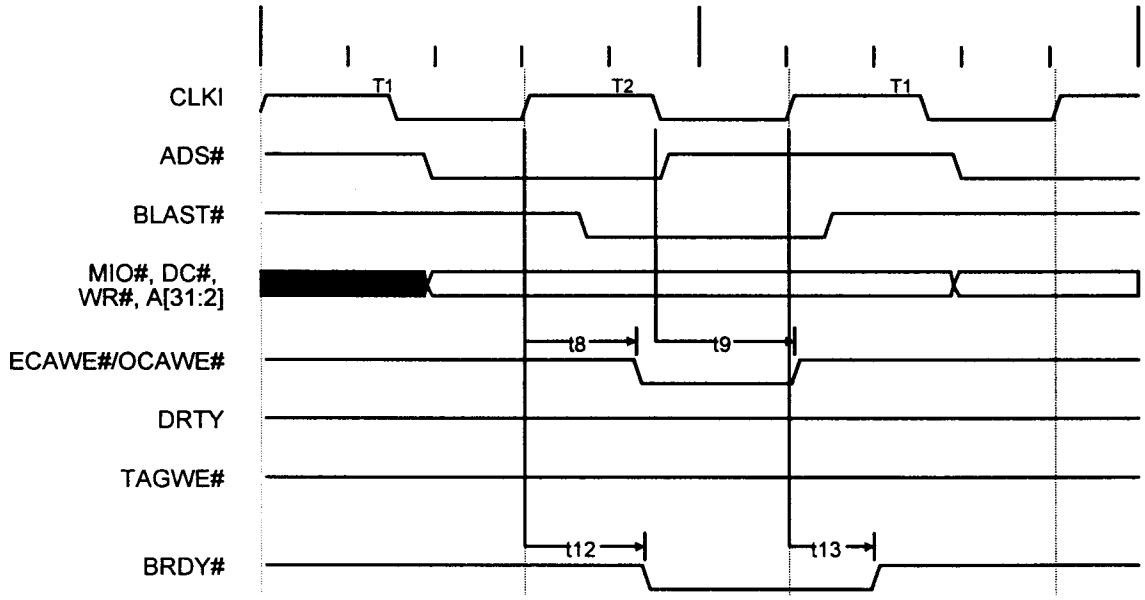


Figure 6-8 Cache Write Hit (1WS/Dirty)

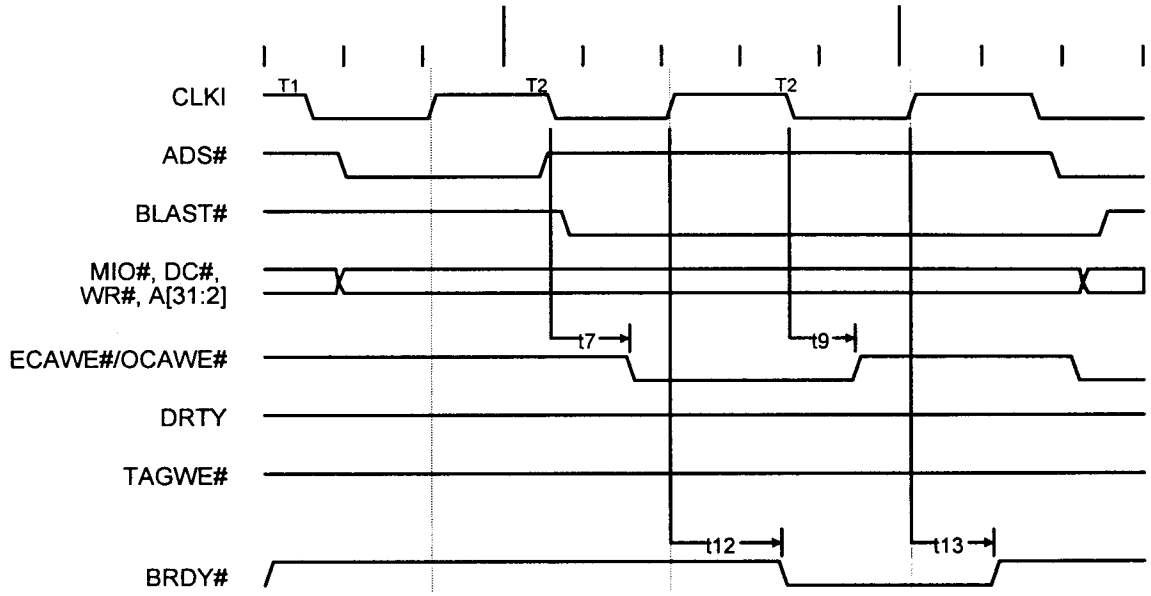
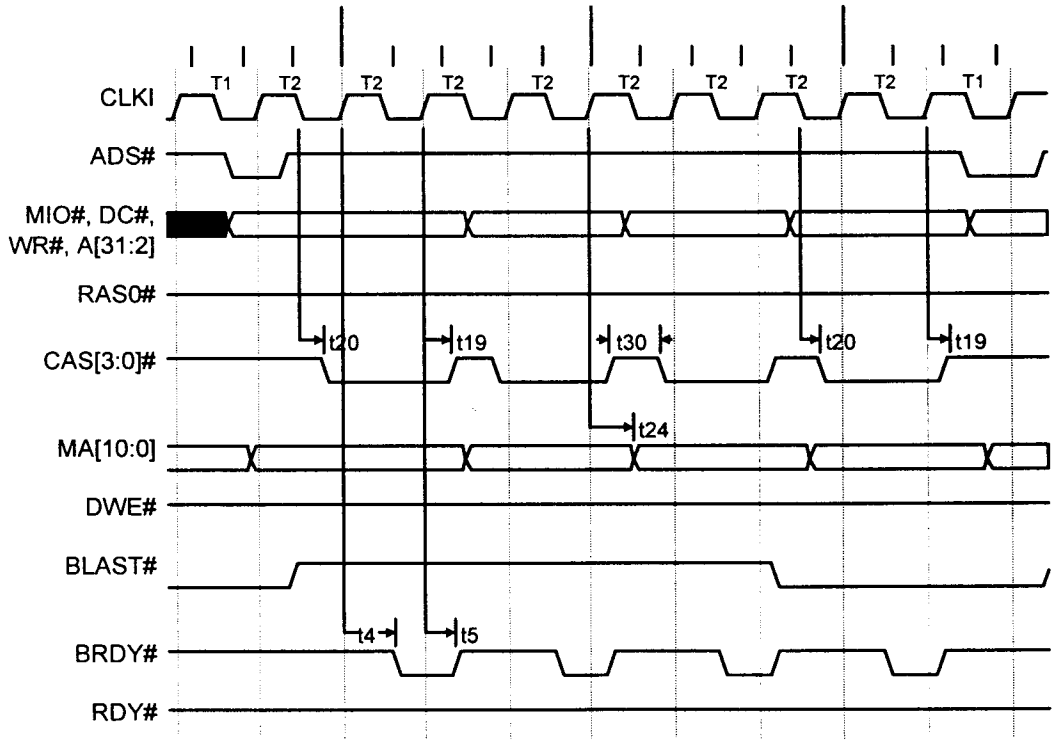
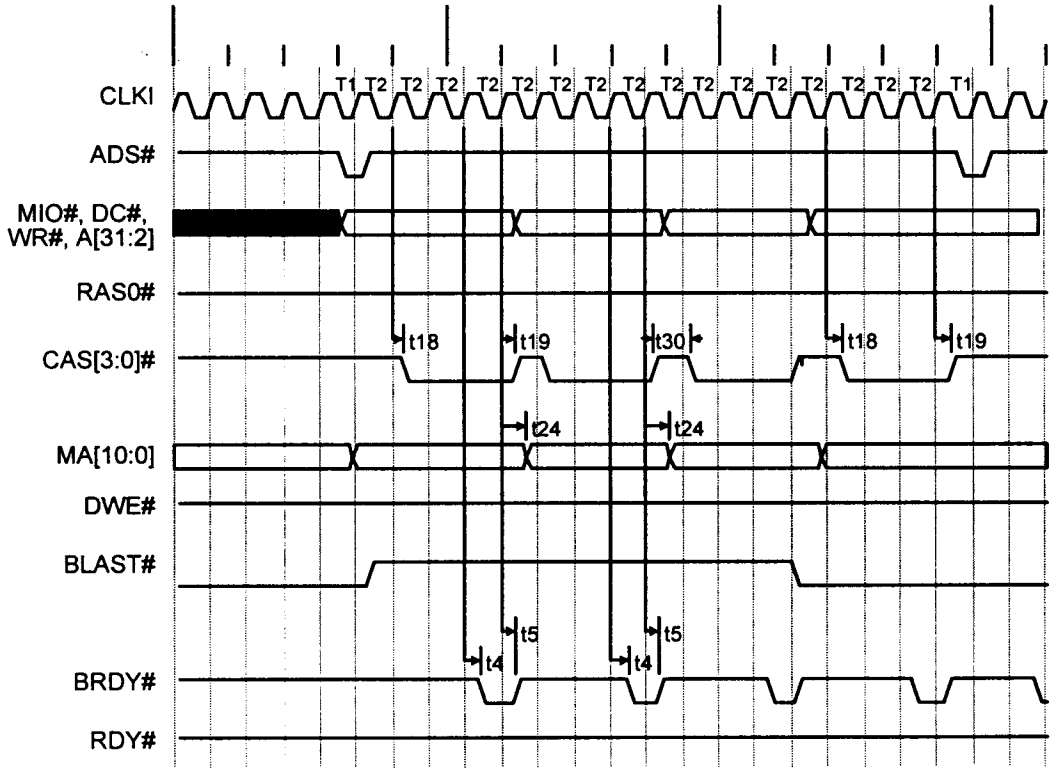


Figure 6-9 DRAM Read (Page Hit)



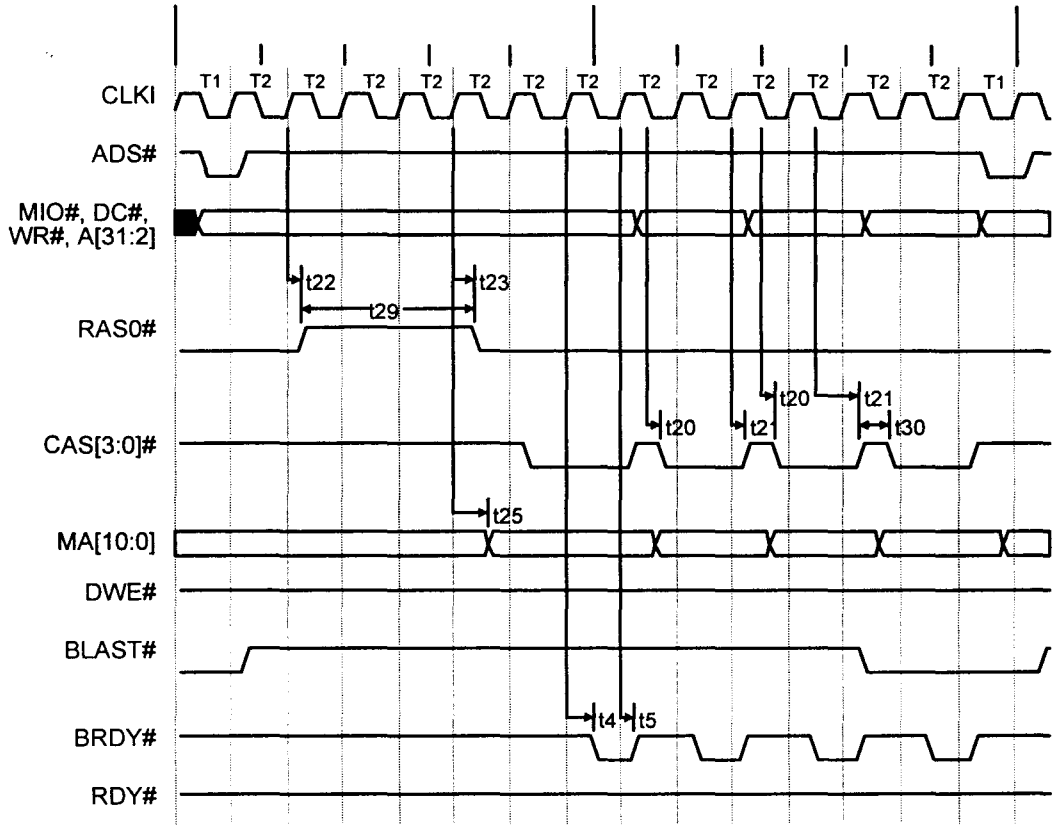
Note: 3-2-2-2 DRAM Read Cycle

Figure 6-10 DRAM Read (Page Hit)



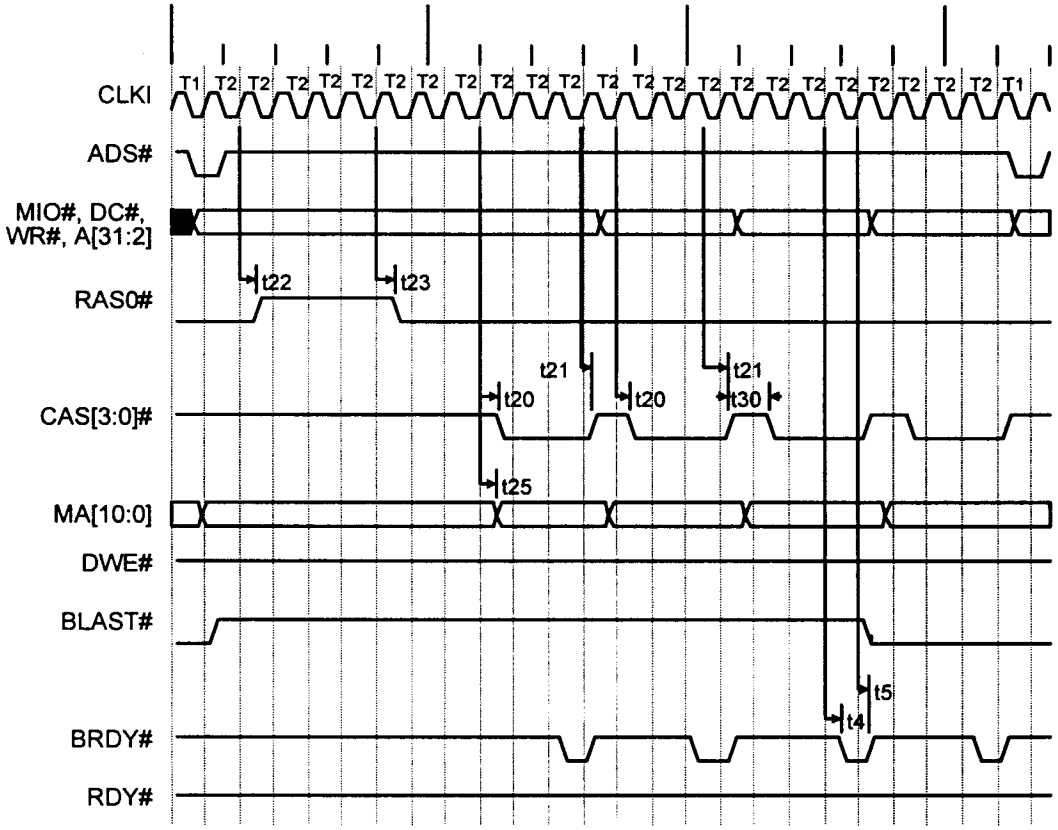
Note: 5-4-4-4 DRAM Read Cycle

Figure 6-11 DRAM Read (Page Miss)



Note: 3-2-2-2 DRAM Read Cycle

Figure 6-12 DRAM Read (Page Miss)



Note: 5-4-4-4 DRAM Read Cycle

Figure 6-13 DRAM Write (0WS/Page Hit)

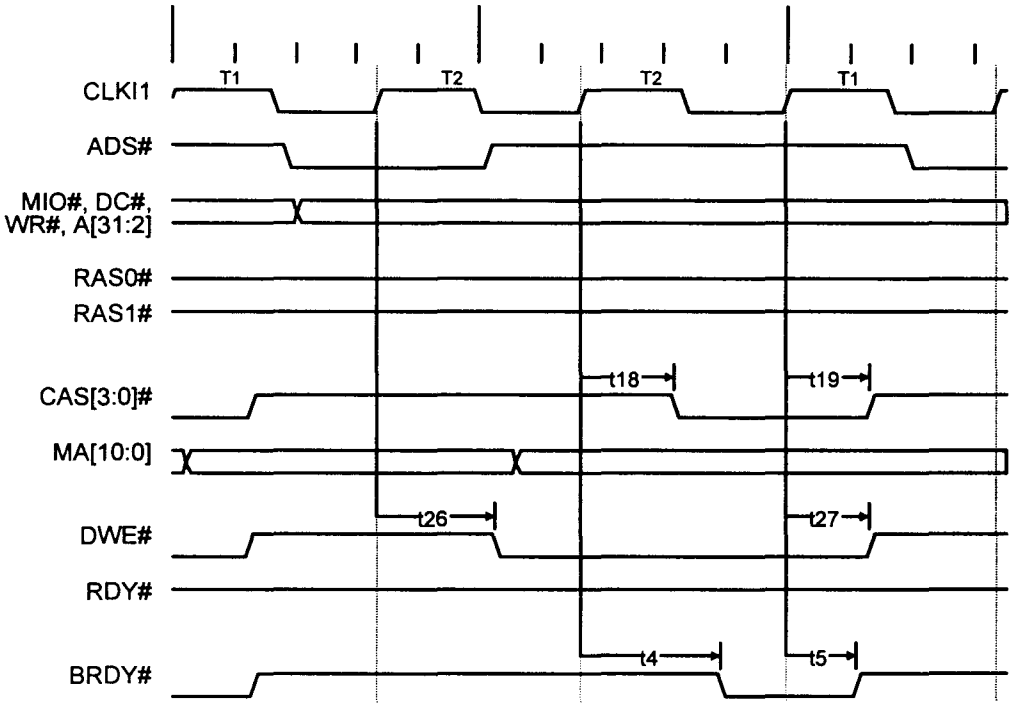


Figure 6-14 DRAM Write (0WS/Page Miss)

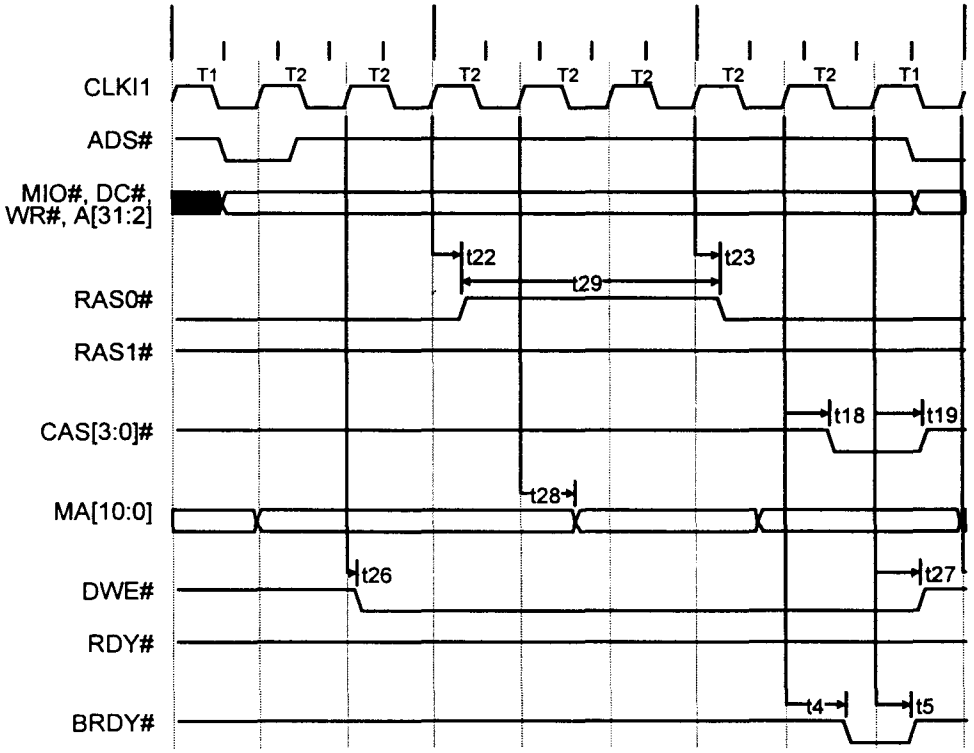


Figure 6-15 DRAM Write (1WS/Page Hit)

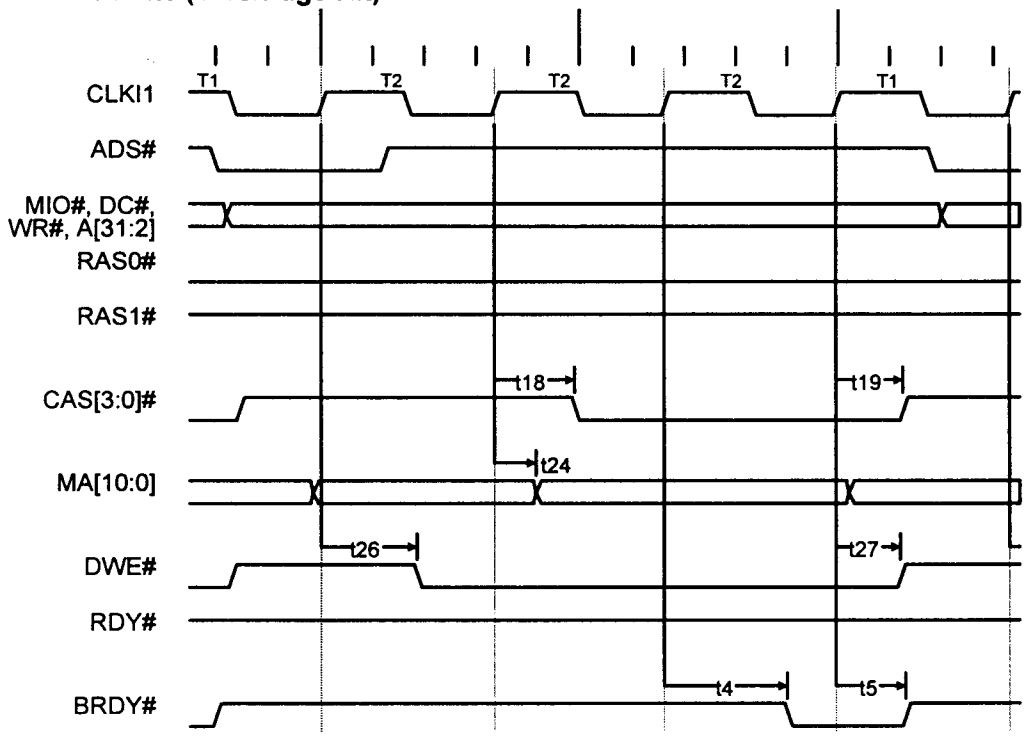


Figure 6-16 Refresh Cycle

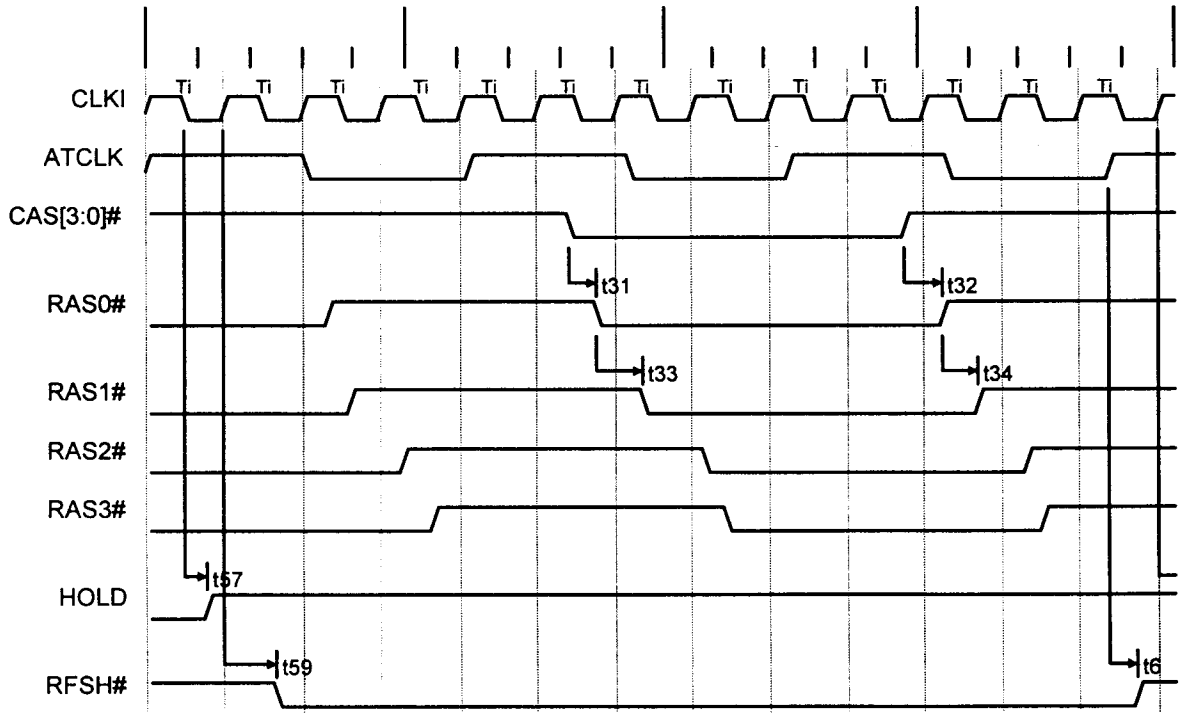


Figure 6-17 ISA Cycle

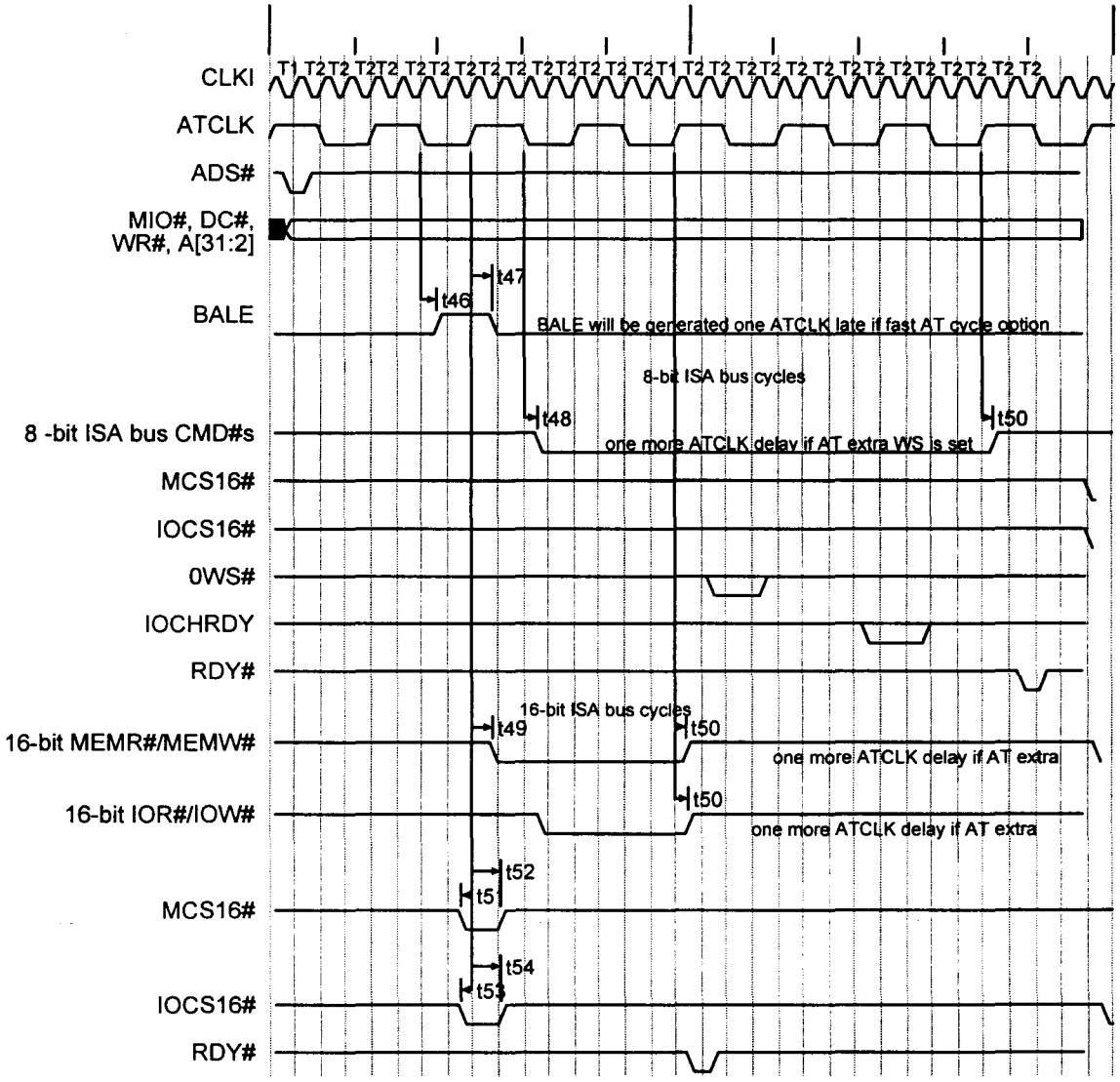


Figure 6-18 DMA Read Cycle

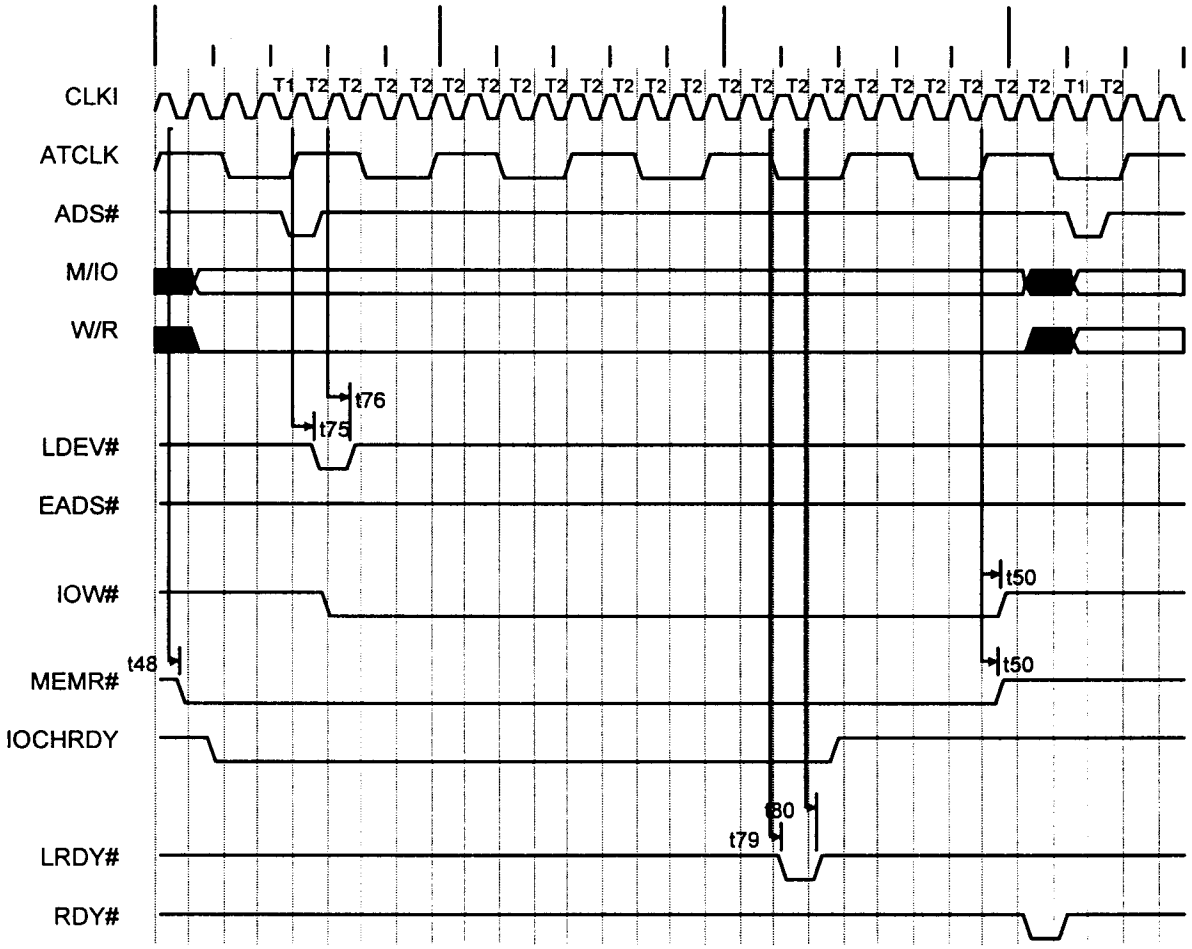
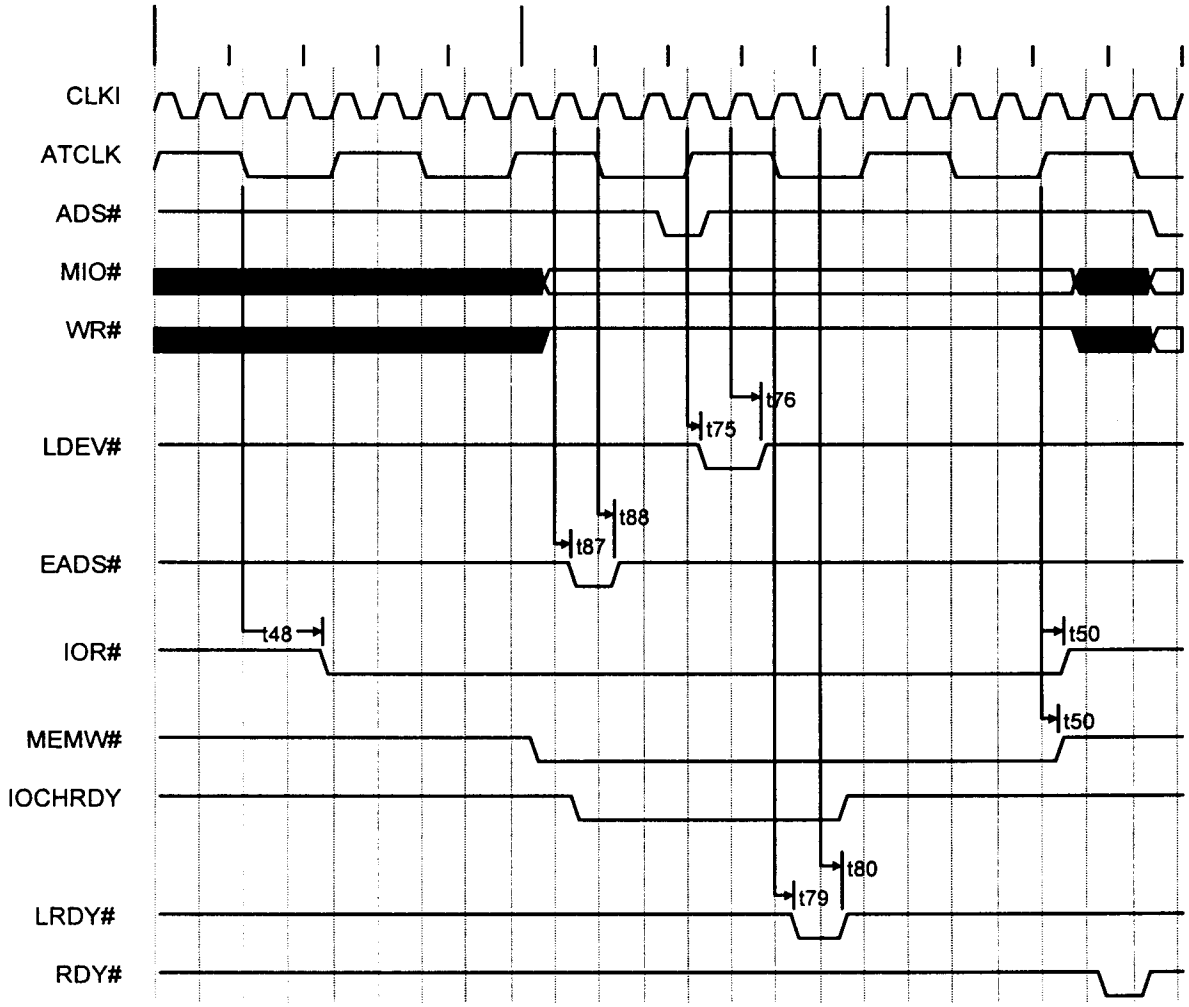


Figure 6-19 DMA Write Cycle



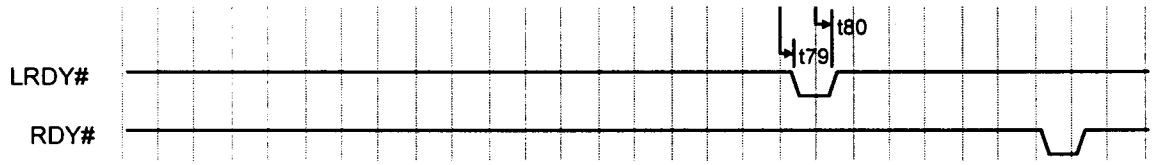

t80
t79
LRDY#
RDY#

Figure 6-21 ISA Master Write Cycle

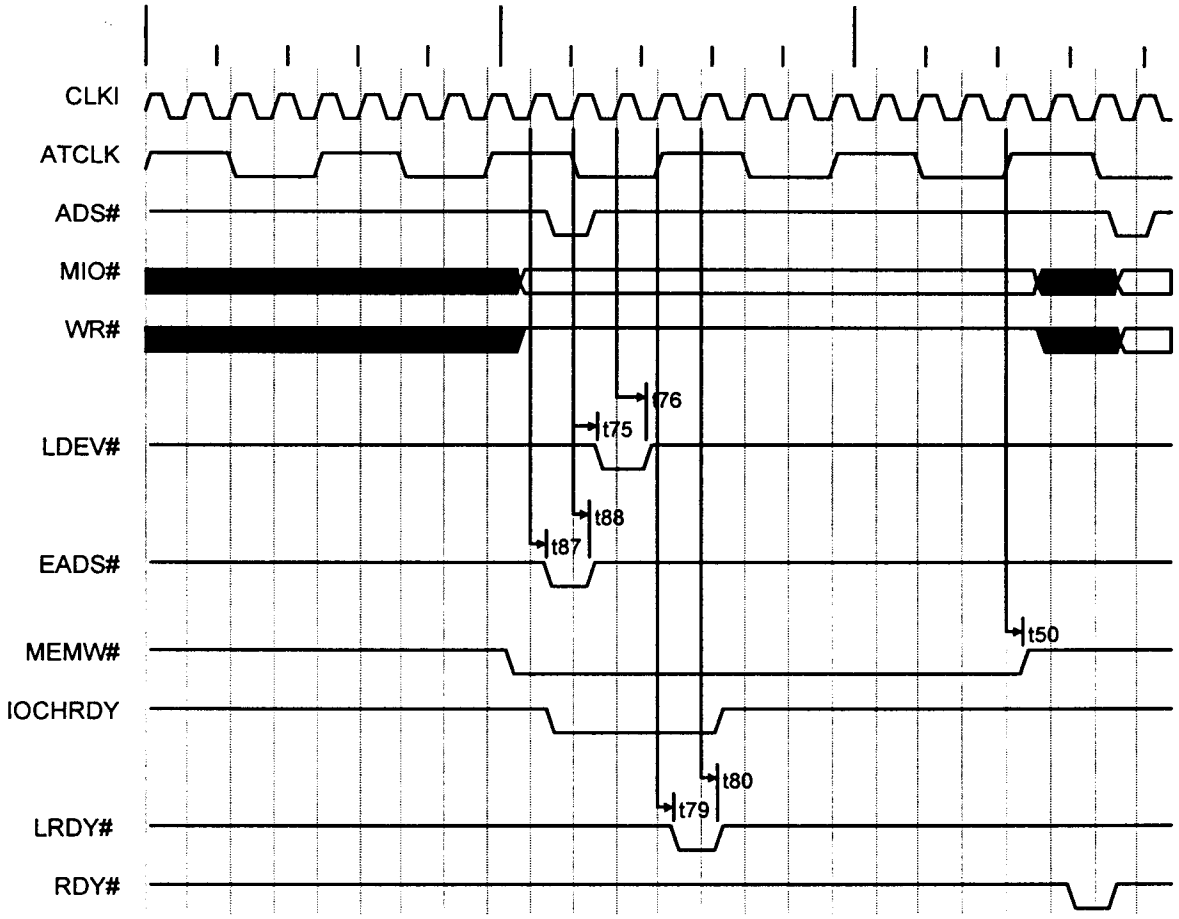


Figure 6-22 ROM Access Cycle

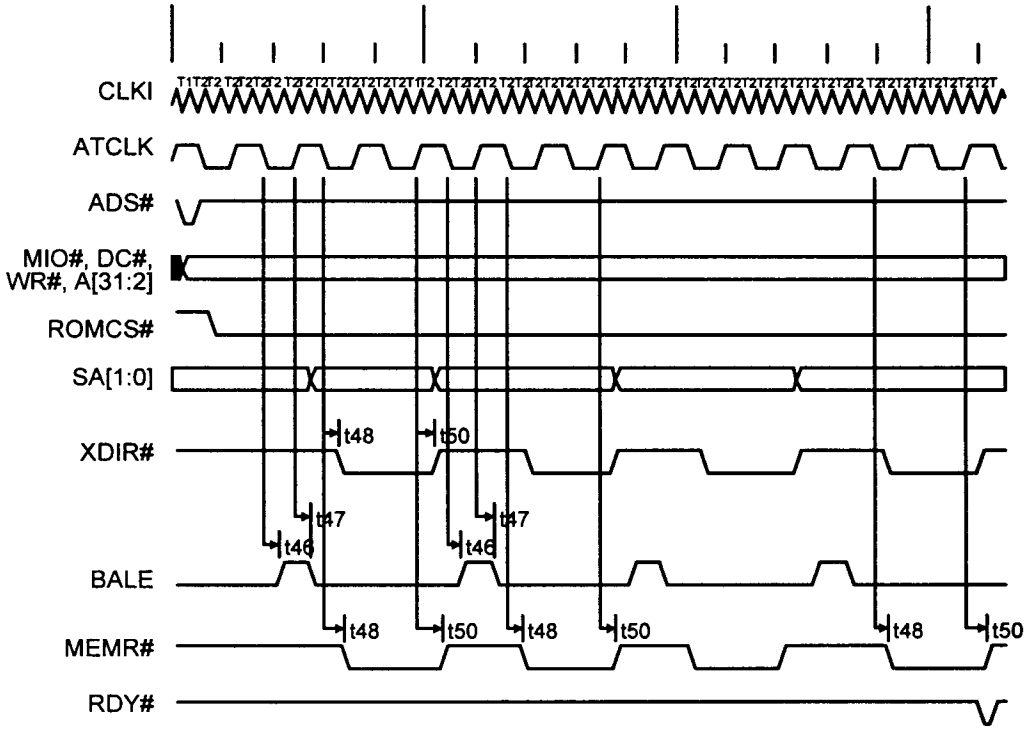
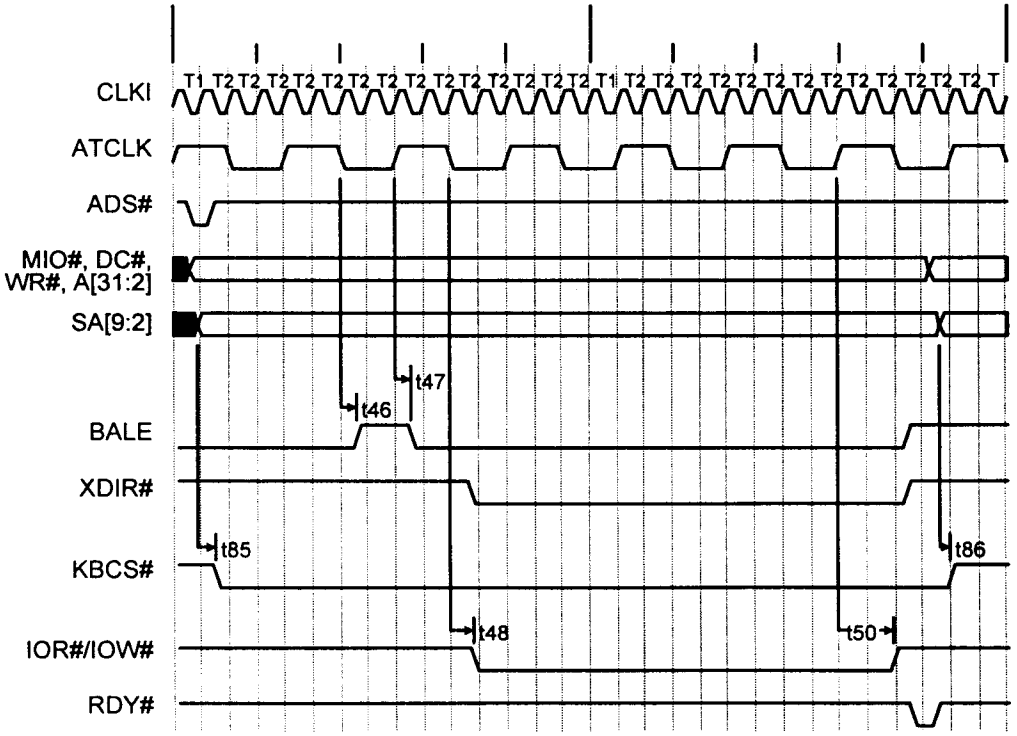


Figure 6-23 Keyboard Controller Access Cycle



7.0 82C895 Testability

The 82C895 can be forced into a test mode for board level testing working automatic test equipment (ATE). There are three kinds of tests available when the 82C895 is in the test mode. They are NAND tree, drive high, and drive low.

The following input combination would enable the 82C895 to go into the test mode:

- Pins 146 and 103 = 1 (high)
- Pins 29, 50, 100, 108, 145, 94 = 0 (low)

With the above input condition and a transition of Pin #99 from 0 to 1 will enable the test mode operation of the 82C895. The 82C895 will latch the test mode input condition whenever Pin #99 makes a transition from 0 to 1. If the input condition is not correct, it will not go into the test mode.

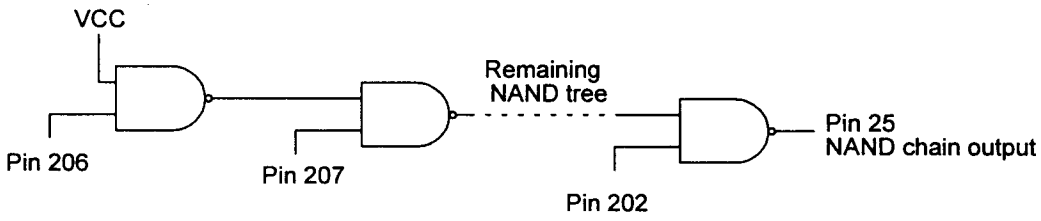
7.1 NAND Tree Test

The NAND tree testing can be done if Pin #99 is held high after entering the test mode. 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 #206 and the output of the chain is Pin #25. The following table gives the pins on the NAND tree chain.

Table 7-1 NAND Tree Test Mode Pins

Pin No.	Remarks	Pin No.	Remarks	Pin No.	Remarks	Pin No.	Remarks
206	NAND tree input start	36	NAND tree input	87	NAND tree input	133	NAND tree input
207	NAND tree input	39	NAND tree input	88	NAND tree input	134	NAND tree input
208	NAND tree input	40	NAND tree input	89	NAND tree input	135	NAND tree input
1	NAND tree input	41	NAND tree input	90	NAND tree input	136	NAND tree input
2	NAND tree input	42	NAND tree input	91	NAND tree input	137	NAND tree input
3	NAND tree input	47	NAND tree input	92	NAND tree input	138	NAND tree input
4	NAND tree input	50	NAND tree input	93	NAND tree input	139	NAND tree input
5	NAND tree input	51	NAND tree input	94	NAND tree input	140	NAND tree input
6	NAND tree input	58	NAND tree input	99	NAND tree input	143	NAND tree input
7	NAND tree input	59	NAND tree input	100	NAND tree input	144	NAND tree input
10	NAND tree input	62	NAND tree input	101	NAND tree input	145	NAND tree input
11	NAND tree input	63	NAND tree input	103	NAND tree input	146	NAND tree input
12	NAND tree input	64	NAND tree input	105	NAND tree input	148	NAND tree input
13	NAND tree input	65	NAND tree input	108	NAND tree input	149	NAND tree input
14	NAND tree input	66	NAND tree input	109	NAND tree input	150	NAND tree input
16	NAND tree input	67	NAND tree input	110	NAND tree input	151	NAND tree input
17	NAND tree input	68	NAND tree input	111	NAND tree input	152	NAND tree input
18	NAND tree input	69	NAND tree input	112	NAND tree input	153	NAND tree input
19	NAND tree input	70	NAND tree input	113	NAND tree input	154	NAND tree input
20	NAND tree input	71	NAND tree input	116	NAND tree input	155	NAND tree input
21	NAND tree input	72	NAND tree input	117	NAND tree input	156	NAND tree input
22	NAND tree input	73	NAND tree input	118	NAND tree input	157	NAND tree input
23	NAND tree input	74	NAND tree input	121	NAND tree input	168	NAND tree input
24	NAND tree input	75	NAND tree input	122	NAND tree input	169	NAND tree input
26	NAND tree input	76	NAND tree input	123	NAND tree input	170	NAND tree input
28	NAND tree input	77	NAND tree input	124	NAND tree input	171	NAND tree input
29	NAND tree input	80	NAND tree input	125	NAND tree input	192	NAND tree input
30	NAND tree input	81	NAND tree input	126	NAND tree input	197	NAND tree input
31	NAND tree input	82	NAND tree input	127	NAND tree input	198	NAND tree input
32	NAND tree input	83	NAND tree input	128	NAND tree input	199	NAND tree input
33	NAND tree input	84	NAND tree input	129	NAND tree input	202	NAND tree input
34	NAND tree input	85	NAND tree input	131	NAND tree input	25	NAND tree output
35	NAND tree input	86	NAND tree input	132	NAND tree input		

Figure 7-1 NAND Tree Block Diagram



7.2 Drive High/Drive Low Test

The drive high/drive low test can be done on the output pins of the 82C895 if Pin #99 is low after entering the test mode. Additionally, Pin #146 will determine the drive high or dive low modes.

Pin #99	Pin #146	Function
0	0	Drive high mode: All odd and even numbered output and bidirectional pins will be driven low and high, respectively.
0	1	Drive low mode: All even and odd numbered output and bidirectional pins will be driven low and high, respectively.

The following output pins will be driven in the drive high/drive low tests.

Table 7-2 Drive High/Drive Low Test Mode Pin

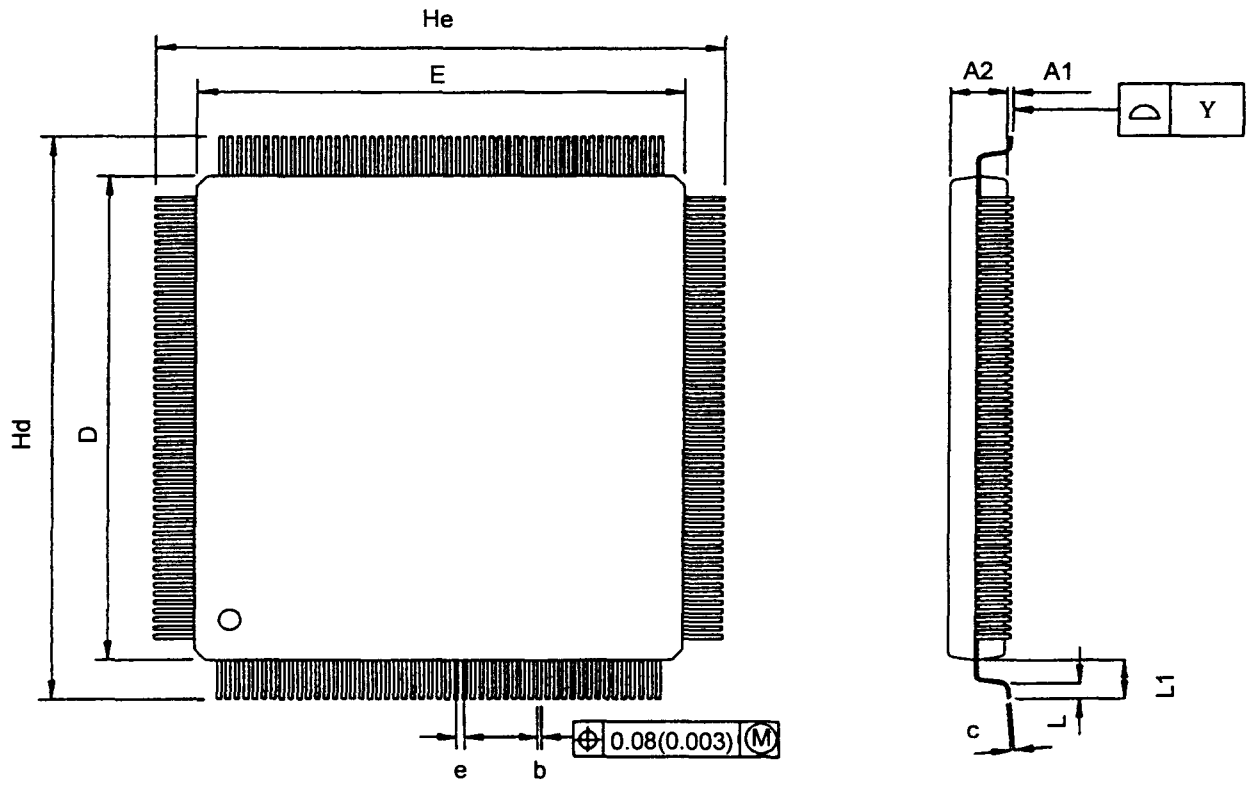
Even Pin No.	Odd Pin No.
8	11
10	13
12	17
14	19
16	21
18	23
20	31
22	33
24	35
30	39
32	41
34	45
36	49
40	53
42	55
44	57
46	63
48	65
52	67
54	69
56	71
62	73
64	75
66	77
68	81

Even Pin No.	Odd Pin No.
70	83
72	85
74	87
76	89
80	91
82	95
84	107
86	109
88	111
90	117
92	121
98	123
102	125
104	127
106	129
108	131
110	133
116	135
118	137
122	139
124	149
126	151
128	153
132	155
134	157

Even Pin No.	Odd Pin No.
136	169
138	171
140	173
148	175
150	177
152	179
154	181
160	183
168	185
170	187
172	189
174	191
176	193
178	195
180	203
184	205
186	
188	
190	
192	
194	
196	
204	

8.0 Mechanical Package Outline

Figure 8-1 208-Pin Plastic Quad Flat Pack



Symbol	Millimeter			Inch		
	Min	Nom	Max	Min	Nom	Max
A1	0.05	0.25	0.50	0.002	0.010	0.020
A2	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	0.35	0.50	0.65	0.014	0.020	0.026
L1		1.30			0.051	
U			0.08			0.003
θ	0		10	0		10