National Semiconductor

ADVANCE INFORMATION

November 1990

PC87120 SuperAT

PC87120 SuperAT™

1.0 General Description

The PC87120, SuperAT, implements most of the medium scale/small scale integration circuits used in building an 80286 processor kernel in one chip. With the PC87120, a complete solution for 10–16 MHz 80286-based motherboard can be designed together with a CPU, memory devices and 16 ICs.

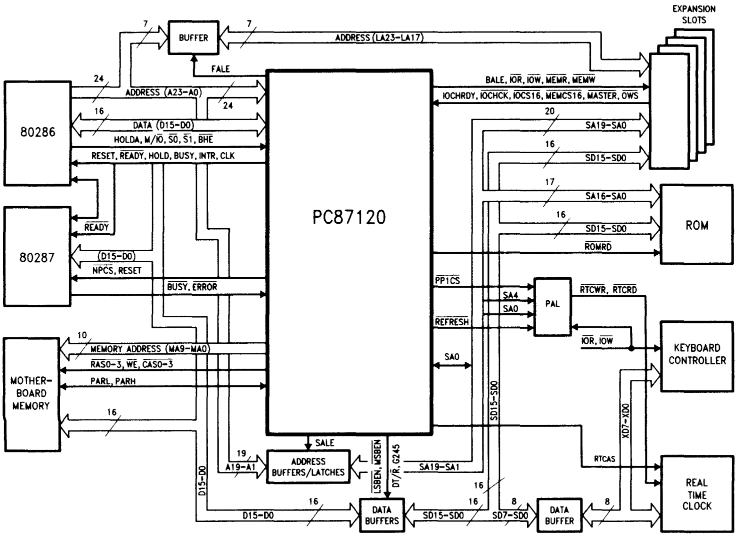
The PC87120 integrates all of the peripheral support required for compatibility in a PC/AT system, along with a programmable clock generator, a programmable bus controller and a programmable memory controller. In addition, the PC87120 incorporates a processor-coprocessor interface, a peripheral chip select, speaker circuitry, ROM BIOS access logic, and reset logic for the processor, coprocessor and the entire system.

The clock generator and bus controller can be programmed to run the processor and the expansion bus synchronously or asynchronously. This allows the designer to increase the system's performance while remaining compatible with the standard 8 MHz expansion bus.

The memory controller can be configured to support 1, 2 or 4 memory banks with zero or one programmed wait state. Conventional mode, two-way page interleaving mode or enhanced two-way page interleaving mode can be selected, allowing the designer to maximize system performance while using low-cost DRAMs.

Features

- SuperAT chip for 10–16 MHz 80286-based PC/AT systems
- Intel® compatible peripherals:
  - Two 8237 DMA controllers
  - Two 8259 Interrupt controllers
  - One 8254 Timer/Counter
- Provides 74LS612 memory mapper
- Software configurable memory organization allows the user to select conventional mode, page mode or two-way page interleaving mode
- Supports 512k to 8 MB motherboard DRAM
- Software configurable wait states
- Shadow RAM support for BIOS, video ROM and video RAM
- Hardware support for LIM EMS 3.2 and 4.0
- Programmable synchronous/asynchronous expansion bus operation
- 8-bit/16-bit BIOS ROM selectable switch
- Supports sleep, suspend, resume modes and slow refresh for DRAM for power savings
- Staggered refresh reduces power supply noise
- Fast GATEA20 and RC optimized for OS/2 operation
- Hardware and software selectable CPU high/low speed
- Low power M2CMOS™ technology
- 160-pin quad flat package



TL/F/11050-1

FIGURE 1. Block Diagram of a Typical PC/AT using the PC87120

RRD-B20M110/Printed in U. S. A.

Table of Contents

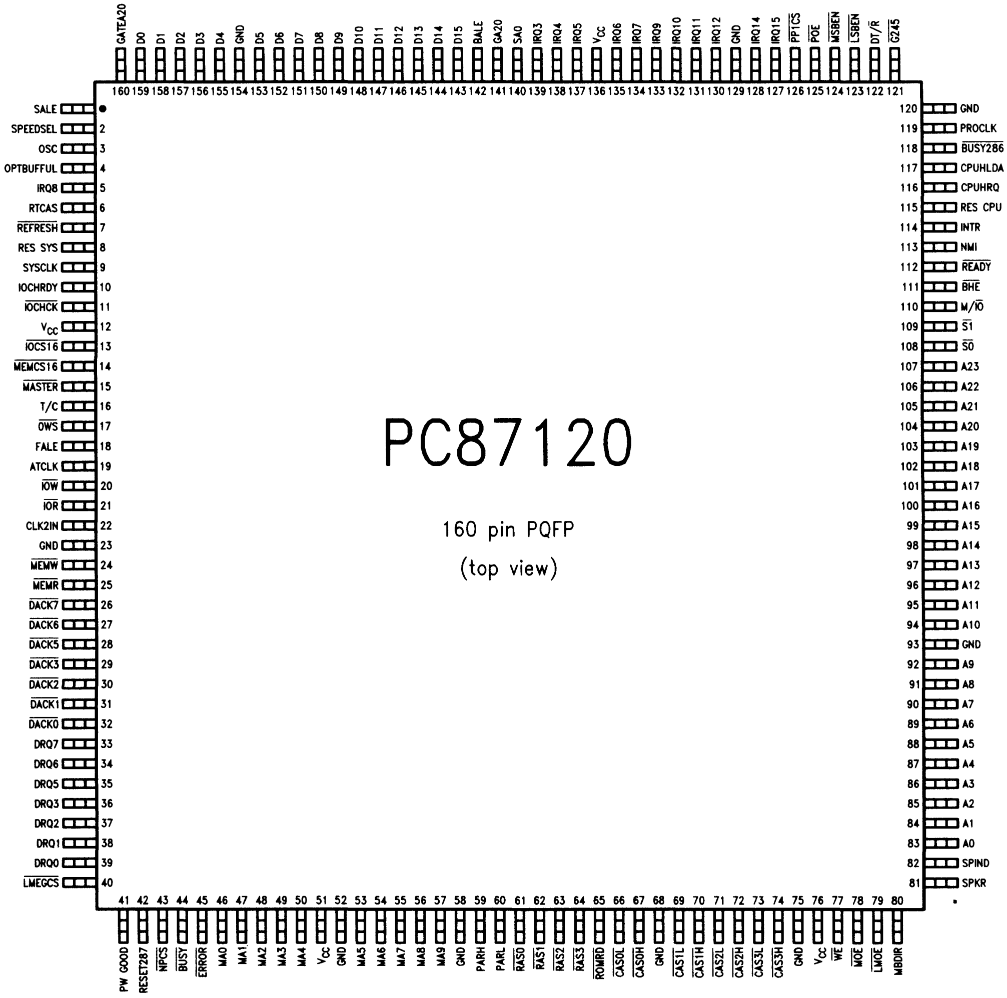
Table of Contents (Continued)

List of Illustrations

List of Tables

2.0 Description

2.1 PIN OUT DIAGRAM OF SuperAT



TL/F/11050-2

FIGURE 2. PC87120 Top View

2.0 Description

PIN DESCRIPTION

Pin #	Description
1	SALE
2	SPEEDSEL
3	OSC
4	OPTBUFFUL
5	IRQ8
6	RTCAS
7	$\overline{\text{REFRESH}}$
8	RES SYS
9	SYSCLK
10	IOCHRDY
11	$\overline{\text{IOCHCK}}$
12	V_{CC}
13	$\overline{\text{IOCS16}}$
14	$\overline{\text{MEMCS16}}$
15	$\overline{\text{MASTER}}$
16	T/C
17	$\overline{\text{OWS}}$
18	FALE
19	ATCLK
20	$\overline{\text{IOW}}$
21	$\overline{\text{IOR}}$
22	CLK2IN
23	GND
24	$\overline{\text{MEMW}}$
25	$\overline{\text{MEMR}}$
26	$\overline{\text{DACK7}}$
27	$\overline{\text{DACK6}}$
28	$\overline{\text{DACK5}}$
29	$\overline{\text{DACK3}}$
30	$\overline{\text{DACK2}}$
31	$\overline{\text{DACK1}}$
32	$\overline{\text{DACK0}}$
33	DRQ7
34	DRQ6
35	DRQ5
36	DRQ3
37	DRQ2
38	DRQ1
39	DRQ0
40	$\overline{\text{LMEGCS}}$
41	PW GOOD
42	RESET287
43	$\overline{\text{NPCS}}$
44	$\overline{\text{BUSY}}$
45	$\overline{\text{ERROR}}$
46	MA0
47	MA1
48	MA2
49	MA3
50	MA4
51	V_{CC}
52	GND
53	MA5
54	MA6
55	MA7
56	MA8
57	MA9
58	GND
59	PARH
60	PARL
61	$\overline{\text{RAS0}}$
62	$\overline{\text{RAS1}}$
63	$\overline{\text{RAS2}}$
64	$\overline{\text{RAS3}}$
65	$\overline{\text{ROMRD}}$
66	$\overline{\text{CAS0L}}$
67	$\overline{\text{CAS0H}}$
68	GND
69	$\overline{\text{CAS1L}}$
70	$\overline{\text{CAS1H}}$
71	$\overline{\text{CAS2L}}$
72	$\overline{\text{CAS2H}}$
73	$\overline{\text{CAS3L}}$
74	$\overline{\text{CAS3H}}$
75	GND
76	V_{CC}
77	$\overline{\text{WE}}$
78	$\overline{\text{MOE}}$
79	$\overline{\text{LMOE}}$
80	MBDIR
81	SPKR
82	SPIND
83	A0
84	A1
85	A2
86	A3
87	A4
88	A5
89	A6
90	A7
91	A8
92	A9
93	GND
94	A10
95	A11
96	A12
97	A13
98	A14
99	A15
100	A16
101	A17
102	A18
103	A19
104	A20
105	A21
106	A22
107	A23
108	$\overline{\text{S0}}$
109	$\overline{\text{S1}}$
110	M/$\overline{\text{IO}}$
111	$\overline{\text{BHE}}$
112	$\overline{\text{READY}}$
113	NMI
114	INTR
115	RES CPU
116	CPUHRQ
117	CPUHLDA
118	$\overline{\text{BUSY286}}$
119	PROCLK
120	GND
121	$\overline{\text{G245}}$
122	DT/$\overline{\text{R}}$
123	$\overline{\text{LSBEN}}$
124	$\overline{\text{MSBEN}}$
125	$\overline{\text{POE}}$
126	$\overline{\text{PP1CS}}$
127	IRQ15
128	IRQ14
129	GND
130	IRQ12
131	IRQ11
132	IRQ10
133	IRQ9
134	IRQ7
135	IRQ16
136	V_{CC}
137	IRQ5
138	IRQ4
139	IRQ3
140	SA0
141	GA20
142	BALE
143	D15
144	D14
145	D13
146	D12
147	D11
148	D10
149	D9
150	D8
151	D7
152	D6
153	D5
154	GND
155	D4
156	D3
157	D2
158	D1
159	D0
160	GATEA20

2.0 Description (Continued)

2.2 ALPHABETICAL PIN DESCRIPTION

TABLE 2-1. Pin Description

Symbol	Pin Number	Type	Description
A0–A9 A10–A23	83–92 94–107	I/O	Address Bus—These bidirectional address signals are inputs during CPU and BUS MASTER cycles; outputs during REFRESH and DMA cycles.
ATCLK	19	I	**AT** Bus **CLocK**—This oscillator input is used for AT bus operation. It provides an independent clock from CPU clock and is only required when the AT bus state machine is not derived from CLK2IN.
BALE	142	O	**B**uffered **A**ddress **L**atch **E**nable—This active High signal latches valid addresses and memory decodes. It is forced High during DMA cycles.
$\overline{\text{BHE}}$	111	I/O	**B**us **H**igh **E**nable—This active Low signal indicates transfer of data on the upper byte of the local data bus (D8–D15).
$\overline{\text{BUSY}}$	44	I	Processor Extension **BUSY**—This active Low signal asserted by the 80287 indicates that it is currently executing a command.
$\overline{\text{BUSY286}}$	118	O	**BUSY 286**—This active Low signal for the CPU indicates that the 80287 numeric processor is in operating mode.
$\overline{\text{CAS0H}}$ $\overline{\text{CAS3H}}$	67, 70 72, 74	O	**C**olumn **A**ddress **S**trobe (**H**igh Byte)—These active Low signals instruct the high byte of DRAM to latch the column address present on the MA0–MA9 pins.
$\overline{\text{CAS0L}}$ $\overline{\text{CAS3L}}$	66, 69 71, 73	O	**C**olumn **A**ddress **S**trobe (**L**ow Byte)—These active Low signals instruct the low byte of DRAM to latch the column address present on the MA0–MA9 pins.
CLK2IN	22	I	**CL**oc**K2 IN**put—This oscillator input provides the fundamental timing for the system. The processor clock is always generated from this input signal and system clock may be derived from this signal.
CPUHLDA	117	I	**CPU H**o**LD A**cknowledge—This active High signal from the CPU indicates that the CPU is relinquishing control of the system.
CPUHRQ	116	O	**CPU H**old **R**e**Q**uest—Active High to CPU to gain control for DMA and Refresh operation.
D15–0	143–153 155–159	I/O	**D**ata Bus—This bidirectional data bus provides a data path to and from the CPU.
$\overline{\text{DACK 7-5}}$ $\overline{\text{DACK 3-0}}$	26–28 29–32	O	**D**MA **ACK**nowledge—These active Low signals acknowledge DMA requests (DRQ7–5, DRQ3–0).
DRQ7–5 DRQ3–0	33–35 36–39	I	**D**MA **R**e**Q**uest—These active High asynchronous signals request DMA services or control of the system. DRQ3–0 are for 8-bit DMA transfers, DRQ7–5 for 16-bit DMA transfers.
DT/$\overline{\text{R}}$	122	O	**D**ata **T**ransmit/**R**eceive—When this signal is High, it indicates data flow from CPU to SD bus. When it is Low, it indicates data flow in the opposite direction. It also controls the direction of data swapping between high byte (SD8–15) and low byte (SD0–7) on the system data bus.
$\overline{\text{ERROR}}$	45	I	Processor Extension **ERROR**—This active Low signal asserted by the 80287, indicates that an unmasked error condition exists.
FALE	18	O	**F**ast **A**ddress **L**atch **E**nable—This signal is an output to latch address A17–A23 to LA17–LA23.
$\overline{\text{G245}}$	121	O	**G**ate **245**—This active Low signal enables data transfer between high byte (SD8–15) and low byte (SD0–7) on the system data bus.
GA20	141	I/O	**G**ated **A**ddress **20**—This signal is an output to generate LA20 as the address for the AT bus during CPU or DMA cycles. It is an input from LA20 during Master cycles.

2.0 Description (Continued)

TABLE 2-1. Pin Description (Continued)

Symbol	Pin Number	Type	Description
GATEA20	160	I	**GATE A20**—This signal is an active Low input from the keyboard controller output port forcing A20 to be low during a CPU shutdown from protected mode.
GND	23, 52, 58 68, 75, 93 120, 129, 154	I	Ground.
INTR	114	O	**INT**errupt **R**equest—This active High signal requests the CPU to suspend its current program execution and service a pending external request. Interrupt requests are maskable.
$\overline{\text{IOCHCK}}$	11	I	**I/O CH**annel Che**CK**—This active Low signal indicates an error condition from an I/O device. The error condition will interrupt the CPU when enabled through the NMI output.
IOCHRDY	10	I	**I/O CH**annel **R**ea**DY**—A Low on this input from an I/O or memory device lengthens the I/O or memory cycle by an integer number of clock cycles.
$\overline{\text{IOCS16}}$	13	I/O	**I/O 16**-bit **C**hip **S**elect—This active Low signal indicates that the current data transfer is a 16-bit IO cycle. This signal should be driven with an open collector or TRI-STATE driver.
$\overline{\text{IOR}}$	21	I/O	**I/O R**ead Command—This active Low signal instructs an I/O device to place the data onto the data bus.
$\overline{\text{IOW}}$	20	I/O	**I/O W**rite Command—This active Low signal instructs an I/O device to read the data on the data bus.
IRQ5-3 IRQ7-6 IRQ12-9 IRQ15-14	137-139 134-135 130-133 127-128	I	**I**nterrupt **R**e**Q**uests—Edge triggered or level triggered input signals which request interrupt services. It should be held high until it's acknowledged.
IRQ8	5	I	**I**nterrupt **R**e**Q**uest 8—Interrupt request from real time clock.
$\overline{\text{LMEGCS}}$	40	O	**L**ow **MEG C**hip **S**elect—Active Low indicates Low Meg memory address space is accessed. This signal is forced Low during refresh cycles. Also, it is used to disable $\overline{\text{SMEMR}}$ and $\overline{\text{SMEMW}}$ signals on the AT bus if accesses are made beyond the 1 Meg Byte address space.
$\overline{\text{LMOE}}$	79	O	**L**ocal **M**emory **O**utput **E**nable—Active Low signal to enable data transfer between memory bus and local bus.
$\overline{\text{LSBEN}}$	123	O	**L**east **S**ignificant **B**yte **EN**able—This active Low signal enables low byte data transfer between local and system buses.
MA0-MA9	46-50 53-57	O	**M**ultiplexed **A**ddress—These are the multiplexed addresses for motherboard DRAM accesses. MA9 is for 1 MB DRAMs only.
$\overline{\text{MASTER}}$	15	I	**MASTER**—This active Low signal is used with DRQ and DACK signals to gain control of the system. Upon receiving DACK, an I/O processor can pull this signal low to gain control of the system buses.
MBDIR	80	O	**M**D/D **B**us **DIR**ection—When High, data flows from D to MD buses; When Low, data flows from MD to D buses.
$\overline{\text{MEMCS16}}$	14	I	**MEM**ory **16**-bit **C**hip **S**elect—This active Low signal indicates that the current data transfer is a 16-bit memory cycle. This signal should be driven with open collector or tri-state driver.
$\overline{\text{MEMR}}$	25	I/O	**MEM**ory **R**ead Command—This active Low signal instructs a memory device to place the data onto the data bus.
$\overline{\text{MEMW}}$	24	I/O	**MEM**ory **W**rite Command—This active Low signal instructs a memory device to read the data on the data bus.

TABLE 2-1. Pin Description (Continued)

Symbol	Pin Number	Type	Description
M/$\overline{\text{IO}}$	110	I	**M**emory/IO—This signal from the CPU indicates a memory cycle when High and an IO cycle when low.
$\overline{\text{MOE}}$	78	O	**M**emory **O**utput **E**nable—This active Low signal should be connected to the DRAM output enable pin to enable data output.
$\overline{\text{MSBEN}}$	124	O	**M**ost **S**ignificant **B**yte **EN**able—This active Low signal enables high byte data transfer between local and system data buses.
NMI	113	O	**N**on-**M**askable **I**nterrupt Request—This active High signal forces the CPU to execute the non-maskable interrupt routine under any condition.
$\overline{\text{NPCS}}$	43	O	**N**umeric **P**rocessor **C**hip **S**elect—This active Low signal drives the $\overline{\text{NPCS}}$ Pin of the 80287 and indicates that the CPU is performing an ESCAPE instruction.
OUTBUFFUL	4	I	**OUT**put **BUF**fer **FULL**—This signal from the keyboard controller generates an interrupt to the CPU when the keyboard buffer is full.
OSC	3	I	**OSC**illator Input—14.318 MHz clock input to generate 1.19 MHz clock for Timer 8254.
PARH	59	I/O	**PAR**ity **H**igh Byte—For the high byte of memory (D8–D15), this signal is used to generate the parity check during read cycles (input) and generates the parity bit for write cycles (output).
PARL	60	I/O	**PAR**ity **L**ow Byte—For the lower byte of memory (D0–D7), this signal is used to generate the parity check during read cycles (input) and generates the parity bit for write cycles (output).
$\overline{\text{POE}}$	125	O	**P**rint **PO**rt **E**nable—This active Low signal enables printer output port.
$\overline{\text{PPICS}}$	126	O	**P**eri**P**heral **C**hip **S**elect—This active Low signal selects I/O devices address from 60H to 7FH.
PROCLK	119	O	**PRO**cessor **CL**oc**K**—This signal is an output to the clock input of the CPU.
PW GOOD	41	I	**P**o**W**er **GOOD**—A Low input to this pin will reset the system.
$\overline{\text{RAS0-3}}$	61–64	O	**R**ow **A**ddress **S**trobe—These active Low signals instruct the DRAM to latch the row addresses present on the MA0–MA9 pins. Four RAS lines can support up to 4-bank memory configuration. A staggered refresh cycle will be performed to reduce power supply noise during RAS switching.
$\overline{\text{READY}}$	112	O	**READY**—This active Low signal to the CPU indicates that the current bus cycle has completed.
$\overline{\text{REFRESH}}$	7	I/O	**REFRESH**—This active Low signal indicates a refresh cycle and can be driven by a microprocessor on the I/O channel during master cycles.
RES CPU	115	O	**RES**et **CPU**—This active High signal resets the CPU during power up, protected mode to real mode switch, and a shutdown cycle.
RES SYS	8	O	**RES**et **SYS**tem Driver—This active High signal resets or initializes system logic during power-up or during a low line voltage outage.
RESET287	42	O	**RESET** 80**287**—This active High signal resets the 80287 coprocessor.
$\overline{\text{ROMRD}}$	65	O	**ROM** Chip **R**ea**D**—This signal accesses data in the ROM and also supports shadow RAM. Once the ROM contents are copied into DRAM space after system initialization, ROMRD output is disabled and ROM addresses are mapped into DRAM physical address space.
RTCAS	6	O	**R**eal **T**ime **C**lock **A**ddress **S**trobe—Active Low signal to latch RAM address into RTC for Read/Write operation.
$\overline{\text{S0}}$ $\overline{\text{S1}}$	108–109	I	Bus Cycle **S**tatus—These input signals from the CPU define the state and type of the CPU cycle.

2.0 Description (Continued)

TABLE 2-1. Pin Description (Continued)

Symbol	Pin Number	Type	Description
SA0	140	I/O	**S**ystem **A**ddress—I/O slot address bit 0. Normally an output, this signal becomes an input during Master cycles.
SALE	1	O	**S**ystem **A**ddress **L**atch **E**nable—Output to latch addresses from local address bus to system address bus to support system addresses.
SPEEDSEL	2	I	**SPEED SEL**ect—Hardware input switch, toggles to change CPU speed.
SPIND	82	O	**SP**eed **IND**icator—When High, it indicates that the machine is running at high speed. When Low, it indicates that the machine is running at low speed.
SPKR	81	O	**SP**ea**K**e**R** Output—Output drives the speaker input.
SYSCLK	9	O	**SYS**tem **CL**oc**K**—This signal is buffered to drive the system clock on the AT bus I/O channel. It may be programmed to be synchronous or asynchronous with the CPU clock (PROCLK).
T/C	16	O	**T**erminal **C**ount—A pulse is active High when the terminal count for any DMA channel is reached.
V_{CC}	12, 51, 76, 136	I	Power: +5V Supply.
$\overline{\text{WE}}$	77	O	**W**rite **E**nable—This output drives the DRAM to enable a write operation. It will be tied to all motherboard DRAM.
$\overline{\text{0WS}}$	17	I	Zero **W**ait **S**tates—This active Low signal from AT bus causes the CPU to complete the current bus cycle without inserting additional wait states. This signal should be driven by open collector or tri-state driver.

3.0 Register Description

The PC87120 registers, each one byte wide, are divided into six groups as shown below. Several of the registers are divided into fields and individual bits which control entirely different functions.

Group Name	Number of Registers
Configuration Registers	10
DMA Registers	
Timer Registers	
Interrupt Controller Registers	9
EMS Registers	
Miscellaneous Registers	4

All of the registers are accessed through the IO port address space.

Following a system reset every register will be set to a default value. Some of the register bits are reserved and have a set value, which is hardwired. These hardwired reserved bits are tied low and will always be read as a 0, even if one attempts to write a 1 to these reserved bits. (These hardwired bits will be shown as **RESERVED:0**, indicating that it will always be read as a 0.) On the other hand, there are some reserved bits which do not control any functions, but are not hardwired. These reserved bits may be read as 0 or 1 and will be shown as **RESERVED**, without any value following.

3.1 CONFIGURATION REGISTERS

The registers in this section are used to configure the PC87120's bus controller, clock generator and memory controller. These configuration registers are mapped into the IO address range FC80h–FC89h.

3.1.1 Configuration Register Access Enable

(IO address FC87h)

(Default = Access Enable Flag is cleared)

In order to write or read from any of the configuration registers (located at IO addresses FC80h–FC86h and FC88h–FC89h), an access to this register (located at address FC87h) must immediately proceed it. When this register is read or written a configuration access enable flag is set in the PC87120. This flag will be cleared on the very next access to any location in the system including any access to the IO address range FC80h–FC89h. If the configuration access enable flag is not set and an access tries to read or write any of the PC87120's configuration registers, the access will be denied. In this way, accidental writes to any of the configuration registers will be ignored, preventing the configuration registers from accidental corruption.

3.1.2 CPU/AT Bus Control Register

(8-Bit, Read/Write—FC80)

(Default = 00110010)

Bit 7—Reserved:0

Bits 6–5—8-Bit AT Cycle Wait States

These two bits determine the number of programmed wait states which will be inserted into each 8-bit access to the AT expansion bus or the peripheral bus.

Bit 6	Bit 5	Wait States
0	0	5 Waits
0	1	4 Waits (default)
1	0	3 Waits
1	1	2 Waits

Bits 4–3—16-Bit AT Cycle Wait States

These two bits determine the number of programmed wait states which will be inserted into each 16-bit access to the AT expansion bus or the peripheral bus.

Bit 4	Bit 3	Wait States
0	0	3 Waits
0	1	2 Waits
1	0	1 Wait (default)
1	1	0 Waits

Bits 2–0—Processor/System CLK Source Select

These three bits determine both the clocks' source(s) and their relative Frequencies. Glitch preventing logic is used to guarantee a smooth transition between various clock frequencies when new values are written to these bits.

Bit 2	Bit 1	Bit 0	PROCLK	SYSCLK
0	X	X	CLK2IN/2	CLK2IN/4 (default)
1	0	1	CLK2IN	CLK2IN/2
1	1	0	CLK2IN	CLK2IN/4
1	1	1	CLK2IN	ATCLK/2

3.1.3 RAM/ROM Configuration Register #1

(8-Bit, Read/Write—FC81)

(Default = 00000000)

Bit 7—Page Interleave Mode Enable

This bit controls the access mode for motherboard memory.

0 = Enable conventional mode (Disable page interleaving mode, this is the default)

1 = Enable page interleaving mode

Bit 6—640 kB to 1 MB RAM Relocation

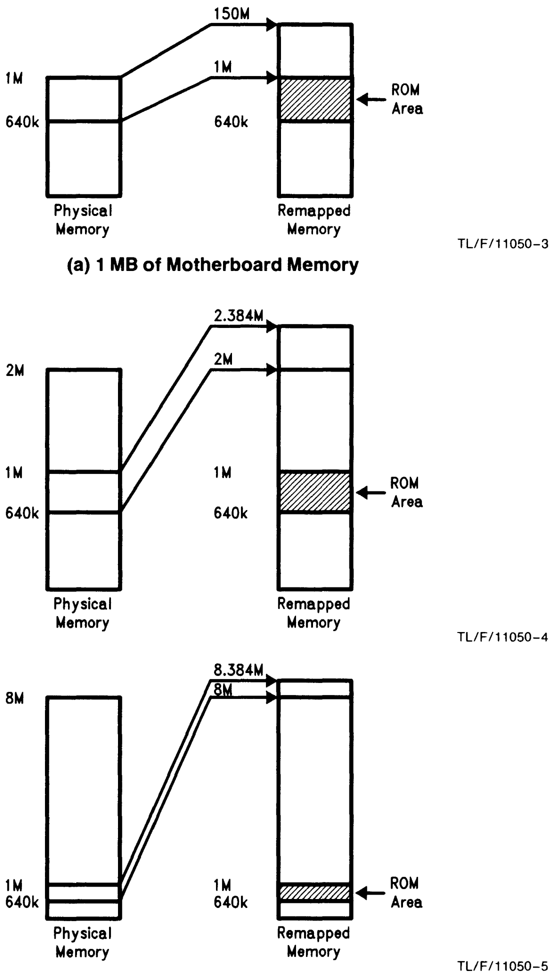
This bit, when set to 1, remaps the memory which coincides with the ROM BIOS area (640 kB to 1 MB), to the top of the supported motherboard memory range. With 1 MB of DRAM, memory will reside from 0 to 640 kB and from 1 MB to 1.384 MB. The ROM BIOS will remain in the 640 kB to 1 MB range.

No remapping takes place when this bit equals 0. This bit must be set equal to 0 if less than 1 MB of DRAM is supported or if BIOS shadow RAM is being used.

0 = No remapping (mandatory for memory configurations 1 & 8)

1 = Relocate RAM area 640 kB – 1 MB

Figures 3-1a, 3-1b and *3-1c* show how the motherboard memory between 640 kB and 1 MB range are remapped when 1, 2, and 8 MB of motherboard memory are configured.



(a) 1 MB of Motherboard Memory

(b) 2 MB of Motherboard Memory

FIGURE 3-1. Remapped Motherboard Memory

3.0 Register Description (Continued)

Bits 5-3—DRAM Configuration

These three bits provide eight common configurations for the DRAM.

Bit 5	Bit 4	Bit 3	Configuration Type	Memory Configuration
0	0	0	1	1-Bank 512 kB
0	0	1	2	1-Bank 2 MB
0	1	0	3	2-Bank 1 MB
0	1	1	4	2-Bank 4 MB
1	0	0	5	4-Bank 2 MB
1	0	1	6	4-Bank 5 MB
1	1	0	7	4-Bank 8 MB
1	1	1	8	2-Bank 640 MB

Configuration Type	DRAM Type			
	Bank 0	Bank 1	Bank 2	Bank 3
1	256 kB	—	—	—
2	1 MB	—	—	—
3	256 kB	256 kB	—	—
4	1 MB	1 MB	—	—
5	256 kB	256 kB	256 kB	256 kB
6	1 MB	1 MB	256 kB	256 kB
7	1 MB	1 MB	1 MB	1 MB
8	256 kB	64 kB	—	—

Bit 2—DRAM Wait States

One programmed wait state is added to every conventional mode motherboard memory access when this bit is set equal to zero. No programmed wait states are added when this bit is set equal to 1.

0 = 1 Programmed Wait State (default)

1 = No Programmed Wait States

Bits 1-0—ROM Wait States

These two bits determine the number of wait states used to access the ROM BIOS. If shadow RAM is used for the BIOS, the number of programmed RAM wait states (determined by bit 2 of this register) will be used for every access to BIOS in the shadow RAM.

Bit 1	Bit 0	Wait States
0	0	3 Waits
0	1	2 Waits
1	0	1 Wait
1	1	0 Waits

3.1.4 RAM/ROM Configuration Register #2

(8-Bit, Read/Write—FC82)

(Default = 00000000)

The values of Bits 7-4 and Bit 0 of this register are determined by system reset strapping options. During a system reset certain outputs of the PC87120 become inputs. If these pins are pulled high by a 10 kΩ resistor during a system reset, then the corresponding bits in this register will be set equal to 1. If these pins are not pulled high during a system reset, then the corresponding bits will be set equal to 0.

The following illustrates how one of the bits in this register is set during a system reset:

If the $\overline{\text{DACK6}}$ output is pulled high with a 10 kΩ resistor during the system reset, Bit 7 of this register will be set equal to 1. On the other hand, if $\overline{\text{DACK6}}$ is not pulled high during a system reset, Bit 7 will be set to 0.

This shows an example of how Bits 7-4 and Bit 0 are configured during a system reset. This register is read only, so once these configuration bits are set they cannot be changed until the next system reset.

Bit	Corresponding Strapping Output
7	$\overline{\text{DACK6}}$
6	$\overline{\text{DACK5}}$
5	$\overline{\text{DACK3}}$
4	$\overline{\text{DACK2}}$
0	$\overline{\text{DACK0}}$

Bit 7—Mode Select for Interleave

Determines if Normal or Advanced two-way page interleaving is to be used for motherboard memory accesses. The state of this bit is ignored if convention mode is selected instead of two-way page interleaving mode (i.e., RAM/ROM Configuration Register #1, Bit 7 = 0).

The default value for this bit is 0, Normal two-way page interleaving mode. If the $\overline{\text{DACK6}}$ output is pulled high by a 10 kΩ resistor during a system reset, this bit will be set equal to 1, advanced two-way page interleaving mode.

0 = Normal Two-Way Page Interleaving Mode (default)

1 = Advanced Two-Way Page Interleaving Mode

Bit 6—8/16-Bit ROM BIOS Select

When set to 0, this bit indicates that 16-bit ROM BIOS will be used. 8-bit ROM BIOS selected when set to 1. The default for this bit is 0, 16-bit ROM BIOS.

If the $\overline{\text{DACK5}}$ output is pulled high by a 10 kΩ resistor during a system reset, this bit will be set to 1, 8-bit ROM BIOS.

0 = 16-bit ROM BIOS (default)

1 = 8-bit ROM BIOS

Bit 5—Page Mode Time Out Enable

When this bit is set to 0, it guarantees that no RAS signal will be held active low for more than 10.0 μs, during page mode accesses. No time out is provided when this bit is set equal to 1.

This bit is ignored when the memory controller is configured to use conventional mode accesses (i.e., RAM/ROM Configuration Register #1, Bit 7 = 0). Conventional memory returns RAS to its inactive high state following every access.

The default for this bit is 0, Enable page mode time out. If the $\overline{\text{DACK3}}$ output is pulled high by a 10 kΩ resistor during a system reset, this bit will be set to 1, disable page mode time out.

0 = Enable page mode time out (default)

1 = Disable page mode time out

Bit 4—Extend RAS During Conventional Memory Access

When this bit is set equal to 1, the RAS and CAS strobes will rise at the same time at the end of every conventional memory access. Some DRAMs require RAS and CAS to rise at the same time.

On the other hand, when this bit is set to 0, RAS will rise a CLK2IN period before CAS. This will reduce the required latency between back to back memory cycles by a CLK2IN period.

This bit is ignored when the memory controller is configured to use two-way page interleaving mode (i.e., RAM/ROM Configuration Register #1, Bit 7 = 1). The default for this bit is 0, RAS rises a CLK2IN period before CAS.

If the $\overline{\text{DACK2}}$ output is pulled high by a 10 kΩ resistor during a system reset, this bit will be set to 1, RAS and CAS rise at the same time.

0 = Off, RAS rises half a CLK2IN period before CAS

1 = On, RAS and CAS rise at the same time

Bit 3—Reserved:0

Bits 2-1—Bank Parity Check Error

These two bits are set when a parity error occurs and indicate which bank of memory the parity error occurred in.

If no parity error has occurred these two bits have no meaning.

Bit 2	Bit 1	Bank No.
0	0	Bank0
0	1	Bank1
1	0	Bank2
1	1	Bank3

Bit 0—ROM BIOS Location

When this bit equals 0, it indicates that the ROM BIOS's data pins are connected to the memory data bus. A 1 in this bit, means that the ROM BIOS's data pins are connected to the peripheral data bus (XD BUS).

The default for this bit is 0, ROM BIOS on the memory data bus.

If the $\overline{\text{DACK0}}$ output is pulled high by a 10 kΩ resistor during a system reset, this bit will be set to 1, ROM BIOS on the peripheral data bus.

Bit 0	Location
0	Memory Data Bus (MD Bus) (default)
1	Peripheral Data Bus (XD Bus)

3.1.5 Shadow RAM Enable Register

(8-Bit, Read/Write—FC83)

(Default = 00000000)

Bits 7-6—ROM Type

These two bits determine the type and size of the ROM or EPROM chips supported. This information along with the 8/16-bit ROM BIOS Select bit determine the amount of ROM BIOS supported.

Bit 7	Bit 6	Type
0	0	16k (27128)
0	1	32k (27256)
1	1	64k (27256)

Bits 5-0—Shadow RAM Enable/Disable Bits

When using shadow RAM for BIOS, only the address range occupied by the BIOS should be enabled. If system BIOS uses shadow RAM, but video BIOS uses ROM, the shadow RAM address range for the system BIOS must be enabled, while the shadow RAM address range that coincides with the video BIOS must be disabled to prevent conflicts.

These six bits (Bits 5-0) are used to enable or disable 64 kB blocks of shadow RAM in the 640 kB to 1 MB BIOS address area. When a bit equals 0, it disables its 64 kB range of shadow RAM. When a bit equals 1, its 64 kB range of shadow RAM is enabled and the corresponding ROM should be disabled.

Bit	Bit Value	Function
5	0	Disables Shadow RAM 0F0000 ~ 0FFFFF default
	1	Enables Shadow RAM 0F0000 ~ 0FFFFF
4	0	Disables Shadow RAM 0E0000 ~ 0EFFFF default
	1	Enables Shadow RAM 0E0000 ~ 0EFFFF
3	0	Disables Shadow RAM 0D0000 ~ 0DFFFF default
	1	Enables Shadow RAM 0D0000 ~ 0DFFFF
2	0	Disables Shadow RAM 0C0000 ~ 0CFFFF default
	1	Enables Shadow RAM 0C0000 ~ 0CFFFF
1	0	Disables Shadow RAM 0B0000 ~ 0BFFFF default
	1	Enables Shadow RAM 0B0000 ~ 0BFFFF
0	0	Disables Shadow RAM 0A0000 ~ 0AFFFF default
	1	Enables Shadow RAM 0A0000 ~ 0AFFFF

3.1.6 ROM Area Enable/Shadow RAM Mode Register

(8-Bit, Read/Write—FC84)

(Default = 00000010)

Bits 7-6—Shadow RAM for Video ROM BIOS Area

These two bits determine access privileges to the shadow RAM for the video ROM BIOS area (located from 0C0000h to 0DFFFFh). When the video ROM BIOS is copied to shadow RAM, the shadow RAM must be configured in either the Write only option or READ/Write option. Otherwise, the copying is impossible.

After the BIOS has been copied to shadow RAM, the shadow RAM must be configured to the Read only option or Read/Write option. This will allow the system to read the video BIOS in shadow RAM. If the Read only option is selected the shadow RAM will act exactly like ROM and cannot be corrupted by accidental writes to this address range.

If the Read/Write option is selected, the video BIOS is susceptible to corruption by misdirected memory writes to this address range.

The states of these two bits are ignored if the shadow RAM for this address range is not enabled in the Shadow RAM Enable Register (i.e., Bits 3–2 set equal to 1 will enable the shadow RAM address range 0C0000h - 0DFFFFh).

Bit 7	Bit 6	Function of Shadow RAM for Video BIOS (0C0000h - 0DFFFFh)
0	0	Disable (default)
0	1	Read Only,
1	0	Write Only
1	1	Read/Write

Bits 5–4—Shadow RAM for Video RAM Area

These two bits determine access privileges to the shadow RAM for the video RAM area (located from 0A0000h to 0BFFFFh).

Bit 5	Bit 4	Function of Shadow RAM for Video RAM (0A0000h - 0BFFFFh)
0	0	Disable (default)
0	1	Read Only,
1	0	Write Only
1	1	Read/Write

Bits 3–2—Shadow RAM for System ROM BIOS Area

These two bits determine access privileges to the shadow RAM for the system ROM BIOS area (located from 0E0000h to 0FFFFFh). When the system ROM BIOS is copied to shadow RAM, the shadow RAM must be configured in either the Write only option or READ/Write option. Otherwise, the copying is impossible.

After the system BIOS has been copied to shadow RAM, the shadow RAM must be configured to the Read only option or Read/Write option. This will allow the system to read the system BIOS in shadow RAM. If the Read only option is selected the shadow RAM will act exactly like ROM and cannot be corrupted by accidental writes to this address range. If the Read/Write option is selected, the video BIOS is susceptible to corruption by misdirected memory writes to this address range.

The states of these two bits are ignored if the shadow RAM for this address range is not enabled in the Shadow RAM Enable Regiser (i.e., Bits 3–2 set equal to 1 will enable the shadow RAM address range 0E0000h - 0FFFFFh).

Bit 3	Bit 2	Function of Shadow RAM for System BIOS (0E0000h - 0FFFFFh)
0	0	Disable (default)
0	1	Read Only,
1	0	Write Only
1	1	Read/Write

Bit 1—ROM BIOS Enable (0E0000h - 0EFFFFh)

This bit enables system ROM BIOS from 0E0000h to 0EFFFFh, when this bit is set to 0. When this bit equals 1, the system ROM BIOS for this memory range is disabled. This BIOS range is only used if more than 64 kB of system BIOS is provided. If less than 64 kB of system BIOS is used only the 0F0000h to 0FFFFFh BIOS range is used and this bit is set equal to zero, disabling this 64 kB block of ROM BIOS. This bit should only be set to zero if more than 64 kB of system BIOS is provided and shadow RAM is not used.

0 = Enable ROM BIOS 0E0000h - 0EFFFFh

1 = Disable ROM BIOS 0E0000h - 0EFFFFh (default)

Bit 0—ROM BIOS Enable (0F0000 - 0FFFFF)

This bit enables system ROM BIOS from 0F0000h to 0FFFFFh, when this bit equals 0. When this bit equals 1, the system ROM BIOS for this memory range is disabled. This system ROM BIOS should always be enabled unless the BIOS has been copied to shadow RAM and the shadow RAM copy is being used to improve system performance.

To prevent any conflicts, the system ROM BIOS and the corresponding shadow RAM system BIOS should never be enabled at the same time.

0 = Enable ROM BIOS 0F0000h - 0EFFFFh (default)

1 = Disable ROM BIOS 0F0000h - 0EFFFFh

3.1.7 DRAM Enable Register

(8-Bit Read/Write—FC86)

(Default = 11110000)

Bits 7–0—DRAM Enable Bits

All eight of these bits enable or disable 64 kB blocks of motherboard DRAM memory between 040000h and 0BFFFFh. These 64 kB blocks must be disabled when EMS expanded memory is mapped into these memory locations. This will prevent conflicts between the expanded memory and the coinciding DRAM memory normally at those physical addresses.

Bit	Bit Value	Function
7	0	Enable RAM B0000h - BFFFFh
	1	Disable RAM B0000h - BFFFFh (default)
6	0	Enable RAM A0000h - AFFFFh
	1	Disable RAM A0000h - AFFFFh (default)
5	0	Enable RAM 90000h - 9FFFFh
	1	Disable RAM 90000h - 9FFFFh (default)
4	0	Enable RAM 80000h - 8FFFFh
	1	Disable RAM 80000h - 8FFFFh (default)
3	0	Enable RAM 70000h - 7FFFFh (default)
	1	Disable RAM 70000h - 7FFFFh
2	0	Enable RAM 60000h - 6FFFFh (default)
	1	Disable RAM 60000h - 6FFFFh
1	0	Enable RAM 50000h - 5FFFFh (default)
	1	Disable RAM 50000h - 5FFFFh
0	0	Enable RAM 40000h - 4FFFFh (default)
	1	Disable RAM 40000h - 4FFFFh

3.1.8 Refresh Control Register

(8-Bit, Read/Write—FC89)

(Default = XXXX0000)

Bits 7-4—Reserved

Bits 3-2—Burst Refresh

These two bits determine the number of refreshes performed for each refresh request. Multiple refreshes reduces latency of the average refresh cycle.

Bit 3	Bit 2	Number of Refreshes per Cycle
0	0	1 (default)
0	1	2
1	0	3
1	1	4

Bits 1-0—Period of Refresh Cycle

These two bits determine the refresh rate for the motherboard DRAM.

Bit 1	Bit 0	Mode (Refresh Rate)
0	0	0 (4 ms) (default)
0	1	1 (8 ms)
1	0	2 (16 ms)
1	1	3 (32 ms)

3.1.9 Sleep Mode/DMA Control Register

(8-Bit Read/Write—FC85)

(Default = 00000000)

Bit 7—DMA Clock Frequency

When this bit is set to 0, the DMA clock (DMACLK) runs at half the frequency of the system clock (SYSCLK). When set to 1, DMACLK runs at the same frequency as SYSCLK.

0 DMACLK = SYSCLK/2 (default)

1 DMACLK = SYSCLK

Bit 6—DMA Command Delays

Determines if a command delay is inserted into every DMA transfer. A command delay forces the command strobe(s) to stay inactive high for half a DMA clock (DMACLK) period at the beginning of each transfer. This will give the address an extra half SYSCLK period to setup before the command strobe(s) become active.

0 = One DMA clock cycle delay (default)

1 = No DMA clock cycle delay

Bits 5-4—DMA Wait State

These two bits determine the number of wait states added to every DMA transfer.

Bit 5	Bit 4	Wait State
0	0	1 wait state (default)
0	1	2 wait states
1	0	3 wait states
1	1	4 wait states

Bit 3—Gate A20 (GA20) Bit

When this bit is set to 0, GA20 will be the same value as the A20 input from the CPU. When this bit is set equal to 1, GA20 will be forced to 0. During real mode operation, GA20 must be forced to 0.

0 = Propagate CPU A20 to A20 line (default)

1 = Set A20 to 0

Bit 2—Suspend/Resume Mode Enable

0 = Enable

1 = Disable

Bit 1—Clocks in Sleep Mode

When set to 1, this bit disables the clocks generated by the PC87120 during sleep mode. The processor clock (PROCLK) and the system clock (SYSCLK) will turn off, reducing power consumption. When set to 0, the clocks will continue to run even in sleep mode.

0 = Clocks are not shut off in sleep mode

1 = Clocks shut off when sleep mode is enabled

Bit 0—Sleep Mode Enable/Disable

When set to 1, this bit enables the PC87120 to enter sleep mode when it decodes a 80286 halt cycle. The PC87120 exits sleep mode whenever an interrupt occurs.

When this bit is set to 0, the PC87120 will lock the system and force a system reset when it decodes a 80286 halt cycle.

0 = Disable

1 = Enable

3.1.10 EMS Control Registers

(8-Bit Read/Write—FC88h)

(Default = XX000000)

Bits 7-6—Reserved

Bit 5—EMS Set Select

This bit determines which EMS register set is in use. Toggling this bit allows one to switch from one set of EMS memory to the other set. This provides efficient support for task switching which requires different EMS memory ranges. Each set of EMS registers may be loaded at the beginning of each task. Then when tasks are switched, the appropriate EMS memory can be enabled by toggling this bit, instead of being forced to reload all of the EMS registers every time tasks are switched.

0 Set 0 in Use (default)

1 Set 1 in Use

Bits 4-3—EMS I/O Port Address Select

These two bits provide a way to remap the IO ports through which the EMS registers may be programmed. The X in the following table corresponds to bits 7-4 of the IO address where the EMS registers are mapped.

For more information about the function of these bits refer to Section 3.5, EMS Registers.

Bit 4	Bit 3	X
0	0	0 (default)
0	1	1
1	0	2
1	1	3

Bits 2-0—On-Board EMS Starting Address

These three bits determine the starting address for all motherboard EMS memory. All motherboard memory below 1 MB is considered real mode memory. All DRAM between 1 MB and the memory selected by these bits will be treated as extended memory. All memory above the indicated EMS starting address will be used as expanded memory.

Bit 2	Bit 1	Bit 0	Starting Address
0	0	0	1 MB (default)
0	0	1	2 MB
0	1	0	4 MB
0	1	1	6 MB
1	0	0	8 MB

3.2 DMA REGISTERS

The registers in this section are used to select the DMA transfer mode as well as the type of transfer for each DMA channel. Additionally, the DMA system registers contain the memory address corresponding to the DMA transfer. Since there are two DMA controllers (DMAC1 and DMAC2) in the PC87120's DMA system, there are two of every register listed in this section, except for the Page Register. The DMA system registers are mapped into the following IO address ranges: 0000-000Fh (DMAC1); even addresses 00C0-00CEh and 00D0-00DEh (DMAC2); and 0080-008Fh (Page Register). Note that all of the DMA system registers must be programmed; there are no default register values.

3.2.1 Command Register

(8-Bit, Write: 0008-DMAC1, 00D0-DMAC2)

Bit 7—DMA Acknowledge Bit (DACK)

0 = DACK signals active low

1 = DACK signals active high

To maintain compatibility with the PC/AT this bit should be set to zero for DMAC2, making its DACK signals active low.

Bit 6—DMA Request Bit (DRQ)

0 = DREQ signals active low

1 = DREQ signals active high

Bit 5—Extended Write Bit (EW)

0 = Extended Write disabled

1 = Extended Write enabled

X if compressed timing is enabled, bit 3 = 1

Bit 4—Rotating Priority Bit (RP)

0 = Fixed Priority

1 = Rotating Priority

Bit 3—Compressed Timing Bit (CT)

0 = Normal Timing

1 = Compressed Timing

X if Memory-to-Memory Transfers are enabled, bit 0 = 1

Bit 2—DMA Controller Disable Bit (CD)

0 = DMA Controller Enabled

1 = DMA Controller Disabled

Bit 1—Address Hold Bit (AH)

This bit determines whether or not the memory address in the DMA access for channel 0 is held constant. It is valid only during a memory-to-memory transfer.

0 = Address Hold Disabled on DMA Channel 0, memory address will increment or decrement as programmed

1 = Address Hold Enabled on DMA Channel 0, memory address is held constant.

X if Memory-to-Memory Transfers are disabled, bit 0 = 0

Bit 0—Memory-to-Memory Transfer Bit (M-M)

0 = Memory-to-Memory Transfers Disabled

1 = Memory-to-Memory Transfers Enabled

3.2.2 Current Address Register

(16-Bit Read/Write)

IO Address for Current Address Register	DMA Channel Assignments	
00h	Channel 0	DMAC1
02h	Channel 1	
04h	Channel 2	
06h	Channel 3	
C0h	Channel 4	DMAC2
C4h	Channel 5	
C8h	Channel 6	
CCh	Channel 7	

Bits 15–0—Current DMA Memory Address Bits

These bits contain the 16 bits of the system memory address produced by the DMA controller and its corresponding 8-bit latch. Note that these bits are read or written in two 8-bit accesses where an internal flip-flop automatically switches between the high and low address bytes.

When programming the Current Address register for 8-bit data transfers (channels 0–3), the bits 15–0 correspond to the system address bits A15–A0. When programming the Current Address register for 16-bit data transfers (channels 5–7), the bits 15–0 correspond to the system address bits A16–A1. Therefore, the address programmed is the system memory address divided by two for 16-bit data transfers

3.2.3 Current Word Count Register

(16-Bit Read/Write)

IO Address for Current Word Count Register	DMA Channel Assignments	
01h	Channel 0	DMAC1
03h	Channel 1	
05h	Channel 2	
07h	Channel 3	
C2h	Channel 4	DMAC2
C6h	Channel 5	
CAh	Channel 6	
CEh	Channel 7	

Bits 15–0—Current Word Count Bits

These bits determine the number of DMA transfers to be made for that channel. The number of DMA transfers will be one more than the contents of this register. Note that these bits are read or written in two 8-bit accesses where an internal flip-flop automatically switches between the high and low byte of the count value.

3.0 Register Description (Continued)

3.2.4 Base Address Register

(16-Bit Write)

IO Address for Current Address Register	DMA Channel Assignments	
00h	Channel 0	DMAC1
02h	Channel 1	
04h	Channel 2	
06h	Channel 3	
C0h	Channel 4	DMAC2
C4h	Channel 5	
C8h	Channel 6	
CCh	Channel 7	

Bits 15-0—Base DMA Memory Bits

These bits contain the address to be loaded into the Current Address Register automatically upon an $\overline{EOP}$ if Autoinitialization has been selected.

3.2.5 Base Word Count Register

(16-Bit Write)

IO Address for Current Word Count Register	DMA Channel Assignments	
01h	Channel 0	DMAC1
03h	Channel 1	
05h	Channel 2	
07h	Channel 3	
C2h	Channel 4	DMAC2
C6h	Channel 5	
CAh	Channel 6	
CEh	Channel 7	

Bits 15-0—Base Word Count Bits

These bits contain the value for the number of DMA transfers which is loaded into the Current Word Count Register automatically upon an $\overline{EOP}$ if Autoinitialize has been selected

3.2.6 Mode Register

(6-Bit, Write: 000B-DMAC1, 00D6-DMAC2)

Bit 7—Mode Select Bit 1 (M1)

Bit 6—Mode Select Bit 0 (M0)

These two bits select the transfer mode for the DMA channel as shown:

Bit 7	Bit 6	Transfer Mode
0	0	Demand Mode
0	1	Single Mode
1	0	Block Mode
1	1	Cascade Mode

Since the PC87120 system uses cascaded DMA Controllers through channel 4 (channel 0 of DMAC2), DMA channel 4 must always be programmed for cascade mode.

Bit 5—Current Address Decrement/Increment (DEC)

0 = Increment Current Address

1 = Decrement Current Address

Bit 4—Autoinitialize Bit (A1)

0 = Disable Autoinitialization

1 = Enable Autoinitialization

Bit 3—Transfer Type Bit 1 (TT1)

Bit 2—Transfer Type Bit 0 (TT0)

These two bits are used together to select the DMA tranfer type for the selected channel in the following manner:

Bit 3	Bit 2	DMA Transfer Type
0	0	Verify Transfer
0	1	Write Transfer
1	0	Read Transfer
1	1	Not Allowed
X	X	If Cascade Mode has been Selected

Bit 1—Channel Select Bit 1 (CS1)

Bit 0—Channel Select Bit 0 (CS0)

These two bits select the DMA channel for which the mode is being programmed as shown:

Bit 1	Bit 0	DMAC1	DMAC2
0	0	Channel 0	Channel 4
0	1	Channel 1	Channel 5
1	0	Channel 2	Channel 6
1	1	Channel 3	Channel 7

3.2.7 Request Register

(8-Bit, Write: 0009-DMAC1, 00D2-DMAC2)

Bits 7-3—Don't Care

Bit 2—Request Bit (RB)

0 = Reset Software DMA Request for the channel

1 = Set Software DMA Request for the channel

Bit 1—Software DMA Request Channel Select Bit 1 (RS1)

Bit 0—Software DMA Request Channel Select Bit 0 (RS0)

These two bits determine the channel for which the software DMA request will be generated or cleared in the following manner:

Bit 1	Bit 0	DMAC1	DMAC2
0	0	Channel 0	Channel 4
0	1	Channel 1	Channel 5
1	0	Channel 2	Channel 6
1	1	Channel 3	Channel 7

3.2.8 Mask Register

(8-Bit, Write: 000A or 000F - DMAC1, 004 or 00DE DMAC2)

The mask register for each DMA Controller can be written in two different methods, each method entered through different IO addresses.

The first method available to mask a DMA request is to mask or unmask the DMA requests one channel at a time. This is done through IO location 0Ah for channels 0-3 in DMAC1 or through IO location D4h for channels 4-7 in DMAC2. The bits are defined in the following manner when using this method to write to the Mask Register.

3.0 Register Description (Continued)

Bits 7-3—Don't Care

Bit 2—Mask Bit (MB)

0 = Clear DMA request Mask for the channel

1 = Set DMA request Mask for the channel

Bit 1—Mask DMA Request for Channel Select Bit 1 (MS1)

Bit 0—Mask DMA Request for Channel Select Bit 0 (MS0)

These two bits determine the channel for which the DMA request will be masked or cleared as follows:

Bit 1	Bit 0	DMAC1	DMAC2
0	0	Channel 0	Channel 4
0	1	Channel 1	Channel 5
1	0	Channel 2	Channel 6
1	1	Channel 3	Channel 7

The second method available to mask a DMA request is to mask or unmask the DMA requests for all channels at the same time. This is done through IO location 0Fh for channels 0-3 in DMAC1 or through IO location DEh for channels 4-7 in DMAC2. The bits are defined in the following manner when using this method to write to the Mask Register.

Bits 7-4—Don't Care

Bit 3—Mask DMAC1 Channel 3 or DMAC2 Channel 7 Bit (MB3(MB7))

0 = Clear DMA request Mask for Channel 3 (7)

1 = Set DMA request Mask for Channel 3 (7)

Bit 2—Mask DMAC1 Channel 2 or DMAC2 Channel 6 Bit (MB2(MB6))

0 = Clear DMA request Mask for Channel 2 (6)

1 = Set DMA request Mask for Channel 2 (6)

Bit 1—Mask DMAC1 Channel 1 or DMAC2 Channel 5 Bit (MB1(MB2))

0 = Clear DMA request Mask for Channel 1 (5)

1 = Set DMA request Mask for Channel 1 (5)

Bit 0—Mask DMAC1 Channel 0 or DMAC2 Channel 4 Bit (MB0(MB4))

0 = Clear DMA request Mask for Channel 0 (4)

1 = Set DMA request Mask for Channel 0 (4)

3.2.9 Status Register

(8-Bit, Read: 0008-DMAC1, 00D0-DMAC2)

Bit 7—DMA Request on Channel 3 (7) (DRQ3(DRQ7))

0 = No pending DMA request on Channel 3 (7)

1 = A DMA request is pending on Channel 3 (7)

Bit 6—DMA Request on Channel 2 (6) (DRQ2(DRQ6))

0 = No pending DMA request on Channel 2 (6)

1 = A DMA request is pending on Channel 2 (6)

Bit 5—DMA Request on Channel 1 (5) (DRQ1(DRQ5))

0 = No pending DMA request on Channel 1 (5)

1 = A DMA request is pending on Channel 1 (5)

Bit 4—DMA Request on Channel 1 (4) (DRQ0(DRQ4))

0 = No pending DMA request on Channel 0 (4)

1 = A DMA request is pending on Channel 0 (4)

Bit 3—$\overline{\text{EOP}}$ (Terminal Count) on Channel 3 (7) (TC3(TC7))

0 = Channel 3 (7) has not encountered $\overline{\text{EOP}}$

1 = Channel 3 (7) has encountered $\overline{\text{EOP}}$

Bit 2—$\overline{\text{EOP}}$ (Terminal Count) on Channel 2 (6) (TC2(TC6))

0 = Channel 2 (6) has not encountered $\overline{\text{EOP}}$

1 = Channel 2 (6) has encountered $\overline{\text{EOP}}$

Bit 1—$\overline{\text{EOP}}$ (Terminal Count) on Channel 1 (5) (TC1(TC5))

0 = Channel 1 (5) has not encountered $\overline{\text{EOP}}$

1 = Channel 1 (5) has encountered $\overline{\text{EOP}}$

Bit 0—$\overline{\text{EOP}}$ (Terminal Count) on Channel 0 (4) (TC0(TC4))

0 = Channel 0 (4) has not encountered $\overline{\text{EOP}}$

1 = Channel 0 (4) has encountered $\overline{\text{EOP}}$

Reading the Status Register will clear the Terminal Count Bits 0 through 3.

3.2.10 Temporary Register

(8-Bit, Read: 0008-DMAC1, 00DA-DMAC2)

Bits 7-0—Memory to Memory Data bits

These bits contain the last byte of data transferred in a Memory-to-Memory transfer.

3.2.11 Page Register

(8-Bit, Read/Write)

IO Hex Address	PC87120 8-Bit Page Register Assignments
087	DMA Channel 0
083	DMA Channel 1
081	DMA Channel 2
082	DMA Channel 3
08B	DMA Channel 5
089	DMA Channel 6
08A	DMA Channel 7
08F	Refresh Cycle

Bits 7-0—Page Boundary Address

These bits make up the upper bits in the address generation for DMA transfers. For 8-bit transfers (channels 0 through 3) these bits will correspond to A16-A23. For 16-bit DMA transfers (channels 5 through 7), these bits will correspond to A17-A23.

3.3 TIMER REGISTERS

The operation of the Programmable Interval Timer (PIT) Registers are explained in greater detail in Section 10.2.2.

The PITs registers and counters power up with random contents. Therefore, each counter must be programmed before it can be used. Counters are programmed by first writing to the Control Word Register followed by writing an initial count to the appropriate counter.

3.3.1 Control Word Register

(8-Bit, Write—0043)

The Control Word Register determines the counter to be programmed, the counter's mode of operation, the method programmed, the counter's mode of operation, the method that the counter is read and written, as well as whether the counter is binary or BCD. Note that the Control Word Register will need to be written once for each counter. When a *Control Word is written for* a counter, all of that counter's control logic is immediately reset and its OUT pin goes to a known initial state.

Bit 7—Select Counter Bit 1 (SC1)
Bit 7—Select Counter Bit 0 (SC0)

These two bits select the counter to be programmed or the Read Back Command as follows:

Bit 7	Bit 6	Counter/Command Selected
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Read Back Command

Bit 5—Read/Write Bit 1 (RW1)
Bit 4—Read/Write Bit 0 (RW0)

These two bits select the method that the counter selected with SC1-SC0 will be read or written, or the Counter Latch Command for the selected counter.

Bit 5	Bit 4	Read/Write Method or Command
0	0	Counter Latch Command
0	1	R/W LSB of Counter Only
1	0	R/W MSB of Counter Only
1	1	R/W LSB first followed by MSB of Counter

The following table summarizes the commands and counter accesses as programmed with the Control Word Register Bits 7-4:

Bit 7	Bit 6	Bit 5	Bit 4	Command/Format
0	0	0	0	Counter Latch Command for Counter 0
0	0	0	1	Counter 0 LSB R/W Only
0	0	1	0	Counter 0 MSB R/W Only
0	0	1	1	Counter 0 LSB and MSB R/W
0	1	0	0	Counter Latch Command for Counter 1
0	1	0	1	Counter 1 LSB R/W Only
0	1	1	0	Counter 1 MSB R/W Only
0	1	1	1	Counter 1 LSB and MSB R/W
1	0	0	0	Counter Latch Command for Counter 2
1	0	0	1	Counter 2 LSB R/W Only
1	0	1	0	Counter 2 MSB R/W Only
1	0	1	1	Counter 2 LSB and MSB R/W
1	1	X	X	Read Back Command

Bit 3—Mode Bit 2 (M2)
Bit 2—Mode Bit 1 (M1)
Bit 1—Mode Bit 0 (M0)

These three bits determine the mode of operation for the selected counter as follows:

Bit 3	Bit 2	Bit 1	Counting Mode	Mode Description
0	0	0	Mode 0	Interrupt on Terminal Count
0	0	0	Mode 1	Hardware Retriggerable One Shot
X	1	0	Mode 2	Rate Generator
X	1	1	Mode 3	Square Wave Mode
1	1	0	Mode 4	Software Triggered Strobe
1	1	1	Mode 5	Hardware Triggered Strobe

Bit 0—Binary Coded Decimal Bit (BCD)

0 = Selected Counter is a 16-bit Binary Counter
1 = Selected Counter is a 4 Decade BCD Counter

3.3.2 Counter Registers

(16-bit, Read/Write—Counter 0: 0040, Counter 1: 0041, Counter 2: 0042)

The process of reading and writing each of the counters is identical, except for the IO address accessed. The counter bits are defined differently when the counter is read and written. Therefore, both reading and writing the three counters will be described here.

For each counter, the Control Word Register must be written before the initial count. The initial count must follow the counter read/write format as programmed in the Control Word Register.

New initial counts may be written into a counter any time without re-issuing a Control Word, as long as the existing format is observed. This will not affect the counters' programmed mode. The counters are 16-bit and are accessed through an 8-bit port. This means that writes to a counter can be performed in one of three ways: 1) LSB only, 2) MSB only, 3) LSB followed by MSB. The method of access is defined by the Control Word format. When a Control Word is written, the Input Holding Registers are automatically set to zeroes.

The Counter's bits are defined as:

Bits 15-0—Count Bits

These bits define the count value loaded in the counter.

Counter Read Operations

It is usually desirable to read the value of a counter without disturbing the present count in process. The PC87210 PIT is able to provide this function in two ways; the Counter Latch and Read-Back Command. A count value may also be directly read from a counter. When this method is used, the CLK input for the selected counter must not be enabled to ensure that a stable count value is read. If the count value is changing at the same instant that the count is read, an inval-

3.0 Register Description (Continued)

id count value will be read. When a counter is read (Counter 0—0040h, Counter 1—0041h and Counter 2—0042h), the 16-bit count value must be read in according to the read/write sequence programmed through the Control Word Register.

The Counter's bits are defined as:

Bits 15-0—Count Bits

These bits define the count value currently in the counter.

Counter Latch Command

This command is written to the Control Word Register (address 0043H).

Bit 7—Select Counter Bit 1 (SC1)
Bit 6—Select Counter Bit 0 (SC0)

These two bits select the counter to be latched as follows:

Bit 7	Bit 6	Counter Selected
0	0	Counter 0 (IO Port 0040h)
0	1	Counter 1 (IO Port 0041h)
1	0	Counter 2 (IO Port 0042h)
1	1	Undefined

Bit 5—Read/Write Bit 1 (RW1)
Bit 4—Read/Write Bit 0 (RW0)

These two bits must equal 0 to issue the Counter Latch Command.

Bits 3-0—Don't Care for the Counter Latch Command

Counter Bits (Counter 0—0040h, Counter 1—0041h, Counter 2—0041h)

The Counter's bits are defined as:

Bits 15-0—Count Bits

These bits define the count value latched with the Counter Latch Command.

Read-Back Command

This command is written to the Control Word Register (address 0043H).

Bit 7—Select Counter Bit 1 (SC1)
Bit 6—Select Counter Bit 0 (SC0)

These two bits must be equal to 1 to issue the Read Back Command.

Bit 5—Count Bit (COUNT\)

0 = Latch Count of the Selected Counters
1 = Do not latch Count of the Selected Counters

Bit 4—Status Bit (STATUS\)

0 = Latch Status of the Selected Counters
1 = Do not latch Status of the Selected Counters

Bit 3—Count 2 Bit (CNT2)

0 = Do not select Counter 2
1 = Select Counter 2

Bit 2—Count 1 Bit (CNT1)

0 = Do not select Counter 1
1 = Select Counter 1

Bit 1—Count 0 Bit (CNT0)

0 = Do not select Counter 0
1 = Select Counter 0

Bit 0—Reserved: 0

The Read-Back Command allows the user to check the value of a selected counter, determine its programmed mode and monitor the status of the OUT pin and Null Count Flag.

This command may be used to latch multiple Output Holding Registers for the counters in the PIT. By setting Bit 5 to zero and selecting the desired counters to be read, all three counter's Output Holding Registers can be latched at the same instant. Thus, enabling all counters to be read at the same instant. When reading the counters, the pre-programmed format should be observed. The specific counter is automatically unlatched when read or when the counter is reprogrammed.

The Counter's bits are defined as:

Bits 15-0—Count Bits

These bits define the count value latched with Read Back Command.

The Read-Back command can also latch status information of selected counters by setting bit 4 to zero. The Status Register must be latched to be read. The status of a counter is obtained by a read from the selected counter when the Status Registers latched.

This is the Status Byte:

Bit 7—Output Bit

0 = OUT is low (0)
1 = OUT is high (1)

Bit 6—Null Count Bit

This bit indicates if the count has been loaded into the counter. The instant when this happens is Mode dependent.

0 = Null Count (Count has not been loaded)
1 = Count can be read (Count has been loaded)

Bit 5—Read/Write Bit 1 (RW1)
Bit 4—Read/Write Bit 0 (RW0)

These bits reflect the counter's programmed read/write format as follows:

Bit 5	Bit 4	Programmed Read/Write Method
0	0	Counter Latch Command
0	1	R/W LSB of Counter Only
1	0	R/W MSB of Counter Only
1	1	R/W LSB First Followed by MSB of Counter

Bit 3—Mode Bit 2 (M2)
Bit 2—Mode Bit 1 (M1)
Bit 1—Mode Bit 0 (M0)

These three bits reflect the counter's programmed mode as follows:

Bit 3	Bit 2	Bit 1	Counting Mode	Mode Description
0	0	0	Mode 0	Interrupt on Terminal Count
0	0	0	Mode 1	Hardware Retriggerable One Shot
X	1	0	Mode 2	Rate Generator
X	1	1	Mode 3	Square Wave Mode
1	1	0	Mode 4	Software Triggered Strobe
1	1	1	Mode 5	Hardware Triggered Strobe

Bit 0—Binary Coded Decimal Bit (BCD)

This bit reflects the programmed counter type.

0 = Counter is programmed to be a 16-bit Binary Counter
1 = Counter is programmed to be a 4 Decade BCD Counter

3.4 INTERRUPT CONTROLLER REGISTERS

The operation and functions of the Programmable Interrupt Controller's (PIC) registers are explained in greater detail in section 10.3.1 Programming the PIC. Refer to section 10.3.1 for more information.

3.4.1 Interrupt Command Word 1 (ICW1)

Master PIC (Write only, even address 0020h–003Eh, Bit 4 = 1)

Slave PIC (Write only, even address 00A0h–00BEh, Bit 4 = 1)

Bits 7–5—Reserved

Bit 4—Must be 1, otherwise the PIC will not recognize the write as a ICW1.

Bit 3—Level Trigger Interrupt Mode (LTIM)

When this bit is set to 1, the PIC will operate in level trigger mode. When set to 0, the PIC operates in edge sensitive trigger mode.

The standard compatible PC/AT sets this bit to 0, edge sensitive mode.

0 = the interrupt request inputs (IRQs) are edge-sensitive; i.e., an interrupt request is generated only by a low-to-high transition of IRQ. The IRQ must return low, then high to generate another request.

1 = the interrupt request inputs (IRQs) are level-sensitive; i.e., requests are generated as long as IRQ remains high.

Bit 2—Address Interval

0 = Interval of 4

1 = Interval of 8

Bit 1—Single PIC (SNGL)

This bit determines if the PIC operates alone or is cascaded with another PIC. When Set to 1, this bit indicates that the PIC is operating alone.

The standard PC/AT sets this bit to 0, indicating that two or more PICs are cascaded (Cascade Mode).

0 = Cascade Mode

1 = Single Mode

Bit 0—Initialization Command 4 (IC4)

When set to 1, ICW4 must be written. When set to 0, ICW4 is not written and all ICW4 functions are set to zero.

The standard compatible PC/AT sets this bit to 1, ICW4 must be written. If this bit is not set to 1, ICW4 will not be written and the PICs internal to the PC87120 will not operate correctly.

0 = ICW4 not written, all ICW4 functions are set to zero

1 = ICW4 must be written

3.4.2 Interrupt Command Word 2 (ICW1)

Master PIC (Write only, odd address 0020h–003Eh, following ICW1 write cycle to Master PIC)

Slave PIC (Write only, odd address 00A0h–00BEh, following ICW1 write cycle to Slave PIC)

Bits 7–3—Interrupt Vector Address (T7–T3)

These five bits may be programmed with the vector address bits 7–3 which the PIC provides the processor to read during the interrupt acknowledge sequence.

Bits 2–0—Not Programmed

These three bits are the decoded address of the active interrupt level during an interrupt acknowledge sequence.

3.4.3 Initialization Command Word 3 (ICW3)

Master PIC (Write only, odd address 0020h–003Eh, following ICW2 write cycle to Master (PIC)

Bits 7–0—Slave Indicator

When a 1 is placed into any of these bits, the Master PIC assumes a Slave PIC is connected to the corresponding interrupt request input (IRQ).

In the PC87120 and all compatible PC/ATs, the slave PIC is fed back into IRQ2 of the Master PIC. So for compatible operations a value of 04h should be written into the Master PIC.

Slave PIC (Write only, odd address 00A0h–00BEh, following ICW2 write cycle to Slave PIC)

Bits 7–3—Reserved.

Bits 2-0—Slave ID

These three bits must equal the interrupt request on the Master PIC which this Slave PIC is cascaded into. In the PC87120 and all compatible PC/ATs, the slave PIC is fed back into IRQ2 of the Master PIC. So for compatible operations a value of 02h should be written into the Slave PIC.

3.4.4 Initialization Command Word 4 (ICW4)

Master PIC (Write only, odd address 0020h–003Eh, following ICW3 write cycle to Master PIC)

Slave PIC (Write only, odd address 00A0h–00BEh, following ICW3 write cycle to Slave PIC)

Bits 7–5—Reserved

Bit 4—Special Fully Nested Mode

(Refer to section 10.3.5.2 for more information about Special Fully Nested Mode)

0 = Enable Normal Nested Mode

1 = Special Fully Nested Mode

Bits 3–2—Buffer/Master-Slave Mode

When bit 3 is set to 0, Non-Buffered Mode will be used and the Master-Slave bit (Bit 2) is a don't care. When bit 3 is set to 1, buffered mode will be used and bit 2 determines whether the PIC is operating as a master or a slave.

Bit 3	Bit 2	Function
0	X	Non-Buffered Mode
1	0	Buffered Mode (Slave)
1	1	Buffered Mode (Master)

Bit 1—Automatic End of Interrupt (AEOI)

0 = Normal EOI

1 = Automatic EOI

Bit 0—Must be set to 1. If this bit is set to 0, the PIC will operate in an incompatible 8085 mode.

3.4.5 Operation Command Word 1 (OCW1)

Master PIC (Read/Write, 8-bit, odd address 0020h–003Fh)

Slave PIC (Read/Write, 8-bit, odd address 00A0h–00BFh)

Bit 7—Interrupt 7 Mask

0 = Interrupt 7 is unmasked

1 = Interrupt 7 is masked

Bit 6—Interrupt 6 Mask

0 = Interrupt 6 is unmasked

1 = Interrupt 6 is masked

Bit 5—Interrupt 5 Mask

0 = Interrupt 5 is unmasked

1 = Interrupt 5 is masked

Bit 4—Interrupt 4 Mask

0 = Interrupt 4 is unmasked

1 = Interrupt 4 is masked

Bit 3—Interrupt 3 Mask

0 = Interrupt 3 is unmasked

1 = Interrupt 3 is masked

Bit 2—Interrupt 2 Mask

0 = Interrupt 2 is unmasked

1 = Interrupt 2 is masked

Bit 1—Interrupt 1 Mask

0 = Interrupt 1 is unmasked

1 = Interrupt 1 is masked

Bit 0—Interrupt 0 Mask

0 = Interrupt 0 is unmasked

1 = Interrupt 0 is masked

3.4.6 Operation Command Word 2 (OCW1)

Master PIC (Read/Write, 8-bit, even address 0020h–003Fh, Bits 4–3 = 00)

Slave PIC (Read/Write, 8-bit, even address 00A0h–00BFh, Bits 4–3 = 00)

Bit 7—Rotate Bit (R)

Bit 6—Select Bit (SL)

Bit 5—End of Interrupt Bit (EOI)

These three bits control the priority Rotation and the EOI modes (and combinations of the two) as shown in the following table:

Bit 7 R	Bit 6 SL	Bit 5 EOI	Function
0	0	1	Non-Specific EOI
0	1	1	Specific EOI
1	0	1	Rotate on Non-Specific EOI
1	0	0	Rotate in Automatic EOI Mode (Set)
0	0	0	Rotate in Automatic EOI Mode (Clear)
1	1	1	Rotate on Specific EOI Command (per L0, L1, L2)
1	1	0	Set Priority Command (per L0, L1, L2)
0	1	0	No Operation

Bits 4–3—Must both be 0, otherwise the PIC will not recognize this access as an OCW2.

Bits 2–0—Level 2–0 (L2, L1, L0)

These bits determine the interrupt level which the Rotate on Specific Command or Set Priority Command is addressed to. If any of the other options are selected by Bits 7–5, these three bits are don't cares.

3.4.7 Operation Command Word 3 (OCW3)

Master PIC (Read/Write, 8-Bit, even address 0020h–003Fh, Bits 4–3 = 01)

Slave PIC (Read/Write, 8-Bit, even address 00A0h–00BFh, Bits 4–3 = 01)

Bit 7—Reserved:0

Bit 6—Enable Special Mask Mode (ESMM)

0 = Prevents the changing of Bit 5, SMM

1 = Always the writing of Bit 5, SMM

Bit 5—Special Mask Mode (SMM)

(Refer to section 10.3.5 for more information about Normal and Special Mask Modes)

0 = Enable Normal Mask Mode

1 = Enable Special Mask Mode

Bit 4—Must be set to 0, otherwise the PIC will not recognize this access as a OCW3.

Bit 3—Must be set to 1, otherwise the PIC will not recognize this access as a OCW3.

Bit 2—Poll Command (P)

The POLL command is used in cases where the INTA sequence is not useable or is not practical. The POLL command, followed by a Read Poll command, is similar to the INTA sequence. The POLL command is effected by issuing an OCW3 with this bit set to 1. The following Read Command (i.e., a read access from IO addresses 0020h–003Fh and 00A0h–00BFh for the Master and Slave PICs respectively) returns a data byte with the following definition:

Bit 7:1 = active Interrupt; 0 = no Interrupt pending

Bits 6–3: zero

Bits 2–0: decode of active (highest priority) requesting interrupt ID (0–7)

0 = No Poll Command Requested

1 = POLL Command

The read cycle following a POLL command is a "Read Poll" regardless of the setting of the Bits 1–0 (RR, RS).

Bit 1—Register Read (RR)

When this bit is set to 0, Bit 0 (R/S) is a "don't care" and no ISR or IRR read command will be generated. When this bit is set to 1, the R/S bit will determine if the ISR or IRR is read by the next read cycle to the PIC.

0 = Disable ISR/IRR read commands

1 = Enable ISR/IRR read commands

Bit 0—Request/Service (R/S)

When this bit is set to 0 and bit 1 (RR) is set to 1, the next read cycle to the PIC will read the Interrupt Request Register (IRR). When this bit is set to 1 and bit 1 (RR) is set to 1, the next read cycle to the PIC (i.e., a read access from IO addresses 0020h–003Fh and 00A0h–00BFh for the Master and Slave PICs respectively) will read the In Service Register (ISR).

When RR = 0, this bit is a "don't care" and ISR and IRR will not be read.

0 = The next read cycle to the PIC will access IRR, if RR = 1.

1 = The next read cycle to the PIC will access ISR, if RR = 1.

If the POLL bit is set, it will take precedence over the RR bit and this bit. The next read cycle will be a Read POLL cycle, not a read ISR or IRR cycle.

3.4.8 Interrupt Request Register (IRR)

This register will be read by the read cycle (i.e., a read access from IO addresses 0020h-003Fh and 00A0h-00BFh for the Master and Slave PICs respectively) following an OCW3 with bits 2-0 set to 010.

This register will indicate which interrupts have been requested, but have not yet received service.

Bit 7—Interrupt Request 7

0 = No pending IRQ7

1 = Pending IRQ7

Bit 6—Interrupt Request 6

0 = No pending IRQ6

1 = Pending IRQ6

Bit 5—Interrupt Request 5

0 = No pending IRQ5

1 = Pending IRQ5

Bit 4—Interrupt Request 4

0 = No pending IRQ4

1 = Pending IRQ4

Bit 3—Interrupt Request 3

0 = No pending IRQ3

1 = Pending IRQ3

Bit 2—Interrupt Request 2

0 = No pending IRQ2

1 = Pending IRQ2

Bit 1—Interrupt Request 1

0 = No pending IRQ1

1 = Pending IRQ1

Bit 0—Interrupt Request 0

0 = No pending IRQ0

1 = Pending IRQ0

3.4.9 In Service Register (ISR)

This register will be read by the read cycle (i.e., a read access from IO addresses 0020h-003Fh and 00A0h-00BFh for the Master and Slave PICs respectively) following an OCW3 with bits 2-0 set to 011.

This register will indicate which interrupt, if any, is presently being serviced.

Bit 7—Interrupt 7 In Service

0 = Interrupt 7 not in service

1 = Interrupt 7 in service

Bit 6—Interrupt 6 In Service

0 = Interrupt 6 not in service

1 = Interrupt 6 in service

Bit 5—Interrupt 5 In Service

0 = Interrupt 5 not in service

1 = Interrupt 5 in service

Bit 4—Interrupt 4 In Service

0 = Interrupt 4 not in service

1 = Interrupt 4 in service

Bit 3—Interrupt 3 In Service

0 = Interrupt 3 not in service

1 = Interrupt 3 in service

Bit 2—Interrupt 2 In Service

0 = Interrupt 2 not in service

1 = Interrupt 2 in service

Bit 1—Interrupt 1 In Service

0 = Interrupt 1 not in service

1 = Interrupt 1 in service

Bit 0—Interrupt 0 In Service

0 = Interrupt 0 not in service

1 = Interrupt 0 in service

3.5 EMS REGISTERS

3.5.1 Page Frame Base Address Register

(8-Bit, Write, IO Port 02XA)

Bit 7—EMS IO Port Enable

This bit will enable or disable all EMS IO ports.

0 = Disable all of the EMS IO ports

1 = Enable all of the EMS IO ports

Bit 6—EMS Wait State Bit

This bit allows the addition of 1 extra wait state when accessing EMS memory in the addressing range C0000h through EFFFFh.

0 = 0 EMS memory wait states

1 = 1 EMS memory wait states

Bits 5, 4—Don't Care

Bits 3-0—Page Bits

These bits determine the starting page frame address for EMS memory in the address range C0000h through EFFFFh. These bits select four 16 KB pages that the EMS memory will be mapped into as shown.

				Page Select			
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**"0"**	**"1"**	**"2"**	**"3"**
0	0	0	0	C0000	C4000	C8000	CC000
0	0	0	1	C4000	C8000	CC000	D0000
0	0	1	0	C8000	CC000	D0000	D4000
0	0	1	1	CC000	D0000	D4000	D8000
0	1	0	0	D0000	D4000	D8000	DC000
0	0	0	1	D4000	D8000	DC000	E0000
0	0	1	0	D8000	DC000	E0000	E4000
0	0	1	1	DC000	E0000	E4000	E8000
0	1	0	0	E0000	E4000	E8000	EC000

3.5.1.2 Page Index Register

(8-Bit, Write, 12xA)

This register is a pointer to the 64 Page Registers for EMS mapping in the address range 40000h-BFFFFh. When an address that is less than the 1 MB limit and is tagged as an EMS page frame (through the DRAM Enable Register) is accessed, a "page index" will be calculated as shown in Table 3-1. This index points to the correct Page Register mapped to the page frame accessed.

For each page index (at 12xA), there is corresponding page register (at 12xB) which holds the address of the page to be mapped into the page frame.

Bit 7—Don't Care

Bit 6—EMS Set Select Bit

This bit determines which EMS register set is in use. Toggling this bit allows one to switch from one set of EMS memory to the other set. This provides efficient support for task switching which requires different EMS memory ranges. Each set of EMS registers may be loaded at the beginning of each task. Then when tasks are switched, the appropriate EMS memory can be enabled by toggling this bit, instead of being forced to reload all of the EMS registers every time tasks are switched. Therefore, this bit selects the EMS Register set (0 or 1) which will be programmed on the next write to 12xB.

TABLE 3-1. EMS Page Index Assignments

Page Frame	Page Index	
	SET 0	SET 1
40000h	00/01	40/41
44000	02/03	42/43
48000	04/05	44/45
4C000	06/07	46/47
50000h	08/09	48/49
54000	0A/0B	4A/4B
58000	0C/0D	4C/4D
5C000	0E/0F	4E/4F
60000h	10/11	50/51
64000	12/13	52/53
68000	14/15	54/55
6C000	16/17	56/57
70000h	18/19	58/59
74000	1A/1B	5A/5B
78000	1C/1D	5C/5D
7C000	1E/1F	5E/5F
80000h	20/21	60/61
84000	22/23	62/63
88000	24/25	64/65
8C000	26/27	66/67
90000h	28/29	68/69
94000	2A/2B	6A/6B
98000	2C/2D	6C/6D
9C000	2E/2F	6E/6F
A0000h	30/31	70/71
A4000	32/33	72/73
A8000	34/35	74/75
AC000	36/37	76/77
B0000h	38/39	78/79
B4000	3A/3B	7A/7B
B8000	3C/3D	7C/7D
BC000	3E/3F	7E/7F

0 = EMS Register Set 0 in use (default)
0 = EMS Register Set 1 in Use

Bits 5-1—Page Index Bits

These bits determine the page index for the page register during the configuration of EMS.

Bit 0—Selects which data byte (high or low) to latch on the next access to the Page Register. This bit will toggle between 0 and 1, selecting pairs of indices.

0 = Low Byte
1 = High Byte

3.5.1.3 Page Registers

The Page Register contains a coded version of the address that will be input to the PC87120's memory controller to access the EMS memory.

Page Register Data—Low Byte

Bit 7—Enable Mapping Bit

0 = Disable EMS Mapping for the page frame
1 = Enable EMS Mapping for the page frame

Bits 6-0—Page Number Bits (PN)

These bits map into the lower six bits of the Page Number (PN) represented by PN [6-0].

Page Register Data—High Byte

Bits 7-4—Don't Care

Bits 3-0—Page Number Bits

These bits map into the bits 10-8 of Page Numbers (PN) represented by PN[10], PN[9], PN[8].

16-Bit, 12xB)

The register 12xB acts as a "window" into the 64 Page Registers for EMS mapping in the address range 40000h-BFFFFh. Through this window, the Page Register information can be seen. Which page register can be seen is controlled by the page index selected through the Page Index Register. The bit definitions in the Page Register are different for high and low bytes.

(8-Bit)

Low Byte	High Byte	Register
02x8	02x9	Page Register 0 for SET 0
42x8	42x9	Page Register 1 for SET 0
82x8	82x9	Page Register 2 for SET 0
C2x8	C2x9	Page Register 3 for SET 0
12x8	12x9	Page Register 0 for SET 1
52x8	52x9	Page Register 1 for SET 1
92x8	92x9	Page Register 2 for SET 1
D2x8	D2x9	Page Register 3 for SET 1

These registers hold part of the memory address that will be mapped into the page frame for EMS mapping in the address range C0000h to EFFFFh. Each page register corresponds to one of the 16 KB page frames programmed in Bits 3-0 of the Page Frame Base Address Register. Page Register 0 corresponds to Page Frame Select "0", and Page Registers 1, 2, and 3 correspond to Page Frame Select "1", "2", and "3" respectively. The registers come in pairs to allow the 16 bits of data to be entered in.

3.6 MISCELLANEOUS REGISTERS

3.6.1 Port B

(0061h)

(Default = 00000000)

Bit 7—Parity Check Flag (Read only bit)

(Default = 0)

This bit will be set high when the PC87120's parity checking circuitry detects a parity error in the motherboard memory. To clear this bit, bit 2 of this register (Enable RAM Parity Check) must be set low. If no motherboard parity errors are detected, this bit will remain low.

Bit 6—IO Channel Check Flag (Read only bit)

(Default = 0)

This bit is set high when an expansion board asserts the active low expansion bus signal IO channel check ($\overline{\text{IOCHCK}}$). To clear this bit, bit 3 of this register (Enable IO Channel Check) must be set low. If no expansion board asserts $\overline{\text{IOCHCK}}$, this bit will remain low.

Bit 5—Timer 2 Output (Read only bit)

(Default = 0)

This bit follows the value of the output of the programmable Timer 2. Timer 2 is used to generate the frequency for the speaker output (SPKR).

Bit 4—Refresh Reference (Read only bit)

(Default = 0)

This bit toggles every time a refresh cycle occurs, providing a way to detect refreshes being performed.

Bit 3—Enable IO Channel Check (Read/Write bit)

(Default = 0)

When set to 0, this bit allows an expansion board to generate a NMI by driving $\overline{\text{IOCHCK}}$ active low. Setting this bit to 1 clears any NMI generated by $\overline{\text{IOCHCK}}$. Also, no $\overline{\text{IOCHCK}}$ NMI can be generated while this bit is set to 1.

Bit 2—Enable Parity Checking (Read/Write bit)

(Default = 0)

When set to 0, this bit allows the PC87120's parity checking circuitry to generate a NMI when a motherboard DRAM parity error occurs. Setting this bit to 1 clears any NMI generated by the parity checker. Also, no parity error NMI can be generated while this bit is set to 1.

Bit 1—Speaker Data (Read/Write bit)

(Default = 0)

When this bit and bit 0 are set to 1, the speaker output (SPKR) will oscillate at the frequency programmed in Timer 2. When set to 0, the SPKR output will stay low and will not oscillate.

Bit 0—Timer 2 Gate Speaker (Read/Write bit)

(Default = 0)

When this bit is set to 1, it enables Timer 2 to oscillate at the timer's programmed frequency. When set low, Timer 2 is disabled. While Timer 2 is disabled, the SPKR output will stay low and will not oscillate.

3.6.2 NMI Enable Flag

(Write—0070h)

Bit 7—NMI Enable

(Default = 1)

When set equal to 0, this bit enables the generation of non-maskable interrupts when a motherboard DRAM parity error occurs or $\overline{\text{IOCHCK}}$ is asserted low. When set equal to 1, NMIs are disabled. Disabling NMIs via this bit does not clear or disable the setting of the Parity Check Flag (Bit 7, Port B) or IO Channel Check Flag (Bit 6, Port B).

The default reset state of this bit is 1, disabling NMIs.

3.6.3 Programmable Option Select Register #94

(8-Bit Read/Write—0094h)

(Default = 10000000)

Bit 7—$\overline{\text{POE}}$ Access Enable

When set to 0, this bit allows one to write a new value into the $\overline{\text{POE}}$ State bit (Bit 7 of the Programmable Option Select Register #102). When set to 1, writing is denied to the $\overline{\text{POE}}$ State bit.

Bits 6-0—Reserved:0

3.6.4 Programmable Option Select Register #102

(8-bit Read/Write—0102h)

(Default = 10011110)

Bit 7—$\overline{\text{POE}}$ State

This bit determines the state of the Printer Port Output Enable pin ($\overline{\text{POE}}$). When set to 0, $\overline{\text{POE}}$ will be driven inactive high (1). When set to 1 $\overline{\text{POE}}$ will be driven active low (0).

Bits 6-0—Reserved

4.0 Reset and Shutdown Logic

The PC87120 resets the CPU, coprocessor and the rest of the system with three separate reset outputs. The three reset outputs are system reset (RESSYS), CPU reset (RESCPU), and coprocessor reset (RES287). The use of three separate resets allows the PC87120 to reset the CPU and/or coprocessor without resetting the entire system.

The PC87120's reset outputs guarantee the following minimum requirements:

Reset Output	Minimum Requirement
RESCPU	active high for 16 processor clock (PROCLK) periods
RES287	active high for 16 processor clock (PROCLK) periods
RESSYS	active high for 16 processor clock (PROCLK) periods

4.1 POWER-ON RESET

When the system's power supply is turned on, it will guarantee that the motherboard has sufficient power before it asserts the active high power good (PWGOOD) signal. While the PWGOOD input to the PC87120 is inactive low, all three of the PC87120's reset outputs will be asserted high. This will reset the entire system including the CPU and coprocessor during power-up. When the PWGOOD signal is asserted high, all three of the reset outputs are deasserted sixteen processor clock (PROCLK) periods later.

4.0 Reset and Shutdown Logic (Continued)

4.2 CPU PROTECTED MODE TO REAL MODE SWITCH

The CPU (80286) powers-up in real mode and may be switched to protected mode via a register bit internal to the 80286. The 80286 does not provide a software means to return to real mode operation. Therefore, the 80286 must be reset in order to return to real mode.

The PC87120 provides a software means to generate a hardware 80286 reset. When the PC87120 decodes a data write of FEh to IO address 0064h, it will assert RESCPU for a minimum of sixteen processor clock (PROCLK) periods. During this "warm" reset only RESCPU is asserted high; RESSYS and RES287 remain deasserted low. Therefore only the 80286 is reset, not the whole system.

In most systems this "warm" reset function is controlled by the keyboard controller, but the PC87120 provides a faster decode and thus enhances system performance.

4.3 COPROCESSOR PROTECTED MODE TO REAL MODE SWITCH

The coprocessor (80287) powers-up in real mode and may be switched to protected mode via a register bit internal to the 80287. The 80287 does not provide a software means to return to real mode operation. So, to switch back to real mode, the 80287 must be reset.

The PC87120 provides a software means to generate a hardware 80287 reset. When the PC87120 decodes any write to IO address 00F1h, it will assert RES287 for a minimum of sixteen processor clock (PROCLK) periods. During this "warm" reset only RES287 is asserted high; RESSYS and RESCPU remain deasserted low. Therefore only the 80286 is reset, not the whole system.

4.4 CPU SHUTDOWN LOGIC

When the CPU (80286) incurs an exception condition in protected mode, it will perform a shutdown cycle. When the PC87120 decodes this shutdown cycle, it will reset the 80286 by asserting RESCPU for sixteen processor clock (PROCLK) cycles. This reset will return the 80286 to real mode and clear the 80286's exception condition.

5.0 CPU/AT Bus Operations

The PC87120 controls the communication between the 80286 and the system expansion and peripheral buses (S-bus and X-bus). The PC87120 accomplishes this function with two separate internal state machines, the CPU state machine and the AT-bus state machines provide this interfacing support. The CPU state machine runs off of the processor clock (PROCLK), and runs synchronously with the 80286 in every mode. This state machine traces every T-state of the 80286 and assures that all CPU interface signals are properly generated. The CPU state machine monitors the 80286's A23–0, M/$\overline{\text{IO}}$, $\overline{\text{S0}}$, $\overline{\text{S1}}$ and $\overline{\text{BHE}}$ outputs to assure proper synchronization. It also controls the latching of the system address for use during expansion bus accesses and produces the active low $\overline{\text{Ready}}$ signal to terminate the present 80286 bus cycle.

The AT-bus state machine runs off of the system clock (SYSCLK) and may run synchronously or asynchronously to the 80286 and the CPU state machine depending on the clock mode selected. The CPU state machine informs the AT-bus state machine of the beginning and type of every access made to the system expansion bus or peripheral bus. The AT-bus state machine monitors $\overline{\text{MEMCS16}}$ and $\overline{\text{IOCS16}}$ at the beginning of every 16-bit expansion bus access to determine if a normal 16-bit access is performed or if a 16-bit to 8-bit transfer is required. After the AT-bus state machine begins its access cycle, it generates a buffered address latch enable (BALE) for the expansion slots, produces the proper command strobe, controls the direction and size of the data buses, and inserts the proper number of programmed wait states. In addition it monitors the $\overline{\text{IOCHRDY}}$ input to determine if extra wait states are to be added, and it monitors the $\overline{\text{0WS}}$ input to determine if the access is to be terminated without inserting any more wait states. When $\overline{\text{0WS}}$ is active low, it takes precedence over programmed wait states as well as $\overline{\text{IOCHRDY}}$ and the access will terminate without adding any extra wait states.

5.1 CLOCK GENERATION

The PC87120 has two TTL oscillator inputs, CLK2IN and ATCLK, which provide the sources to generate the processor clock output (PROCLK) and the system clock output (SYSCLK). PROCLK drives the 80286's clock input with a maximum clock frequency of twice the rated operating frequency of the 80286. The 80286 requires two input clock cycles for each of its T-states, so the 80286 operates at half the frequency of PROCLK.

SYSCLK determines the operating frequency of the system expansion bus. When SYSCLK oscillates at half the frequency of PROCLK, the system expansion bus and the 80286 will operate at the same frequency.

Table 5-1 shows the four common clocking schemes supported by the PC87120.

TABLE 5-1. Bus Mode Clocks

Mode	PROCLK	SYSCLK	Mode Name
0	CLK2IN/2	CLK2IN/4	Non-Turbo Mode
1	CLK2IN	CLK2IN/2	Turbo Bus Mode
2	CLK2IN	CLK2IN/4	Normal Mode
3	CLK2IN	ATCLK/2	Asynchronous Mode

In the original 286-AT, the 80286 and system expansion bus operated at the same frequency. However, as CPU speeds increased, older expansion boards could not maintain the timing requirements. As a result the system expansion bus (AT-bus) usually operates at 8 MHz, while the 80286 runs at or below its rated frequency.

Both the Non-Turbo mode and the Turbo Bus mode run the system expansion bus and the 80286 at the same frequency. Using the same CLK2IN source, the Non-Turbo mode operates the expansion bus and the processor at half the frequency of the Turbo Bus mode.

The Normal mode drives the system expansion bus at half the frequency of the processor. This provides support for a 16 MHz 80286, while maintaining the system expansion bus operation at a compatible 8 MHz.

The Asynchronous mode is the only mode which allows the 80286 to run at an asynchronous frequency from the system expansion bus. This provides support for running the 80286 at 10 or 12.5 MHz, but still allows the system expansion bus to run at 8 MHz. With this mode, a designer does not have to choose between running the bus faster than 8 MHz or running the system expansion bus at half the frequency of the 80286. Thus allowing the designer to maintaining compatibility without reduced system performance.

5.0 CPU/AT Bus Operations (Continued)

The PC87120 defaults to Non-Turbo mode during power-up or following a system reset. Bits 2–0 of the CPU/AT Bus Control Register, located at IO address FC80h, control the bus mode in which the PC87120 will operate. Table 5-2 shows the relationship between bits 2–0 and the mode selected.

TABLE 5-2. Bus Mode Configuration Bits

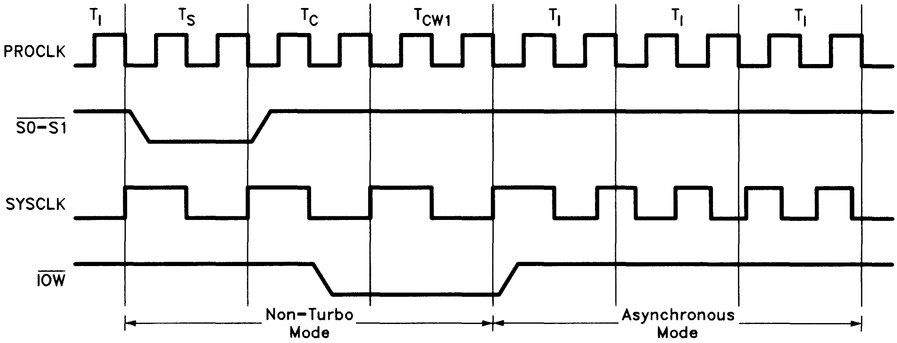
Mode	Bit 2	Bit 1	Bit 0	Mode Name
0	0	X	X	Non-Turbo Mode
1	1	0	1	Turbo Bus Mode
2	1	1	0	Normal Mode
3	1	1	1	Asynchronous Mode

The PC87120 contains logic that guarantees a smooth transition between various clock frequencies when new values are written to the bus mode control bits.

Figure 5-1 shows the timing of a transition from the Non-Turbo mode to the Asynchronous mode.

The PC87120 also supplies an input, called speed select (SPEEDSEL), for a hardware turbo switch. When this input receives a positive transition, via a hardware switch, bit 2 of the CPU/AT Bus Control Register will be toggled. If the PC87120 was operating in the Turbo Bus, Normal or Asynchronous mode before the transition on SPEEDSEL, it will change to the Non Turbo mode after the transition. A second positive transition on the SPEEDSEL input will return the PC87120 to its previous operating mode. Bits 1–0 of the CPU/AT Bus Control Register are not affected.

Again, the PC87120 contains logic used to guarantee a smooth transition between various clock frequencies when SPEEDSEL is toggled.



TL/F/11050-6

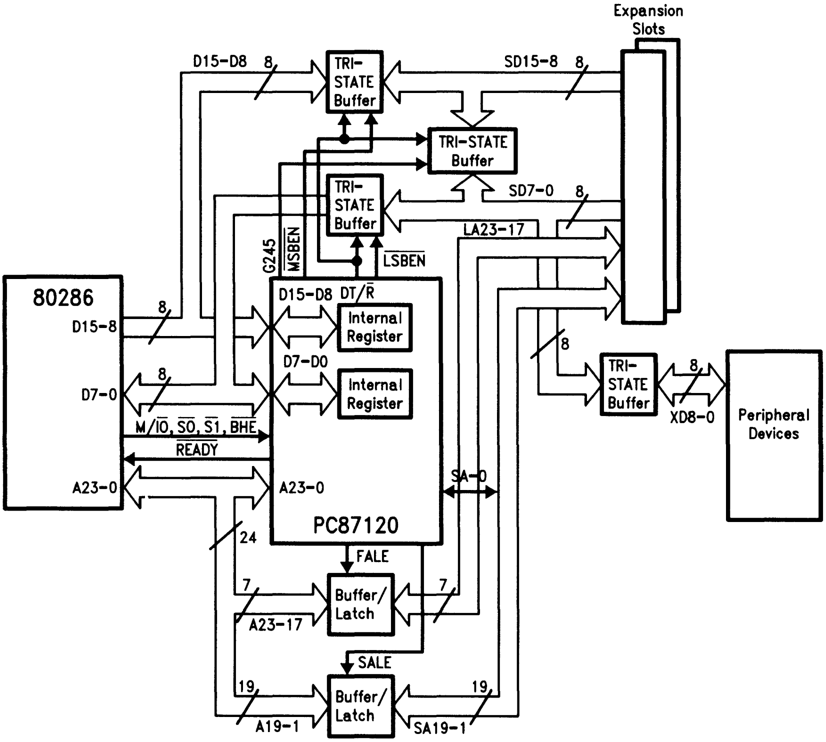
FIGURE 5-1. Non-Turbo Mode to Asynchronous Software Switch

5.2 CPU/AT BUS INTERFACE

The interface between the 80286's local buses, the system expansion buses and the peripheral data bus is shown in *Figure 5-2*. The system address (SA19–0) is latched by SALE which the PC87120's CPU state machine drives active high during phase 2 of the 80286's status T-state (Ts).

Meanwhile, the expansion bus's local address (LA23–17) is latched by the PC87120's fast address latch enable strobe (FALE). The LA23–17 lines are decoded by 16-bit expansion boards for the expressed purpose of driving $\overline{\text{MEMCS16}}$ at the proper time to indicate a 16-bit expansion bus access. To guarantee compatibility, FALE latches LA23–17 from the middle of T_S until the middle of the first expansion bus command T-state (AT-T_C). When FALE is inactive low the local bus's A23–17 values are buffered onto the LA23–17 lines.

The low and high bytes of the local data bus are buffered from the system expansion bus (SD15–0) via two bidirectional buffers enabled by the PC87120's $\overline{\text{LSBEN}}$ and $\overline{\text{MSBEN}}$ respectively. The direction of these buffers is determined by the data transmit/receive signal. DT/$\overline{\text{R}}$.



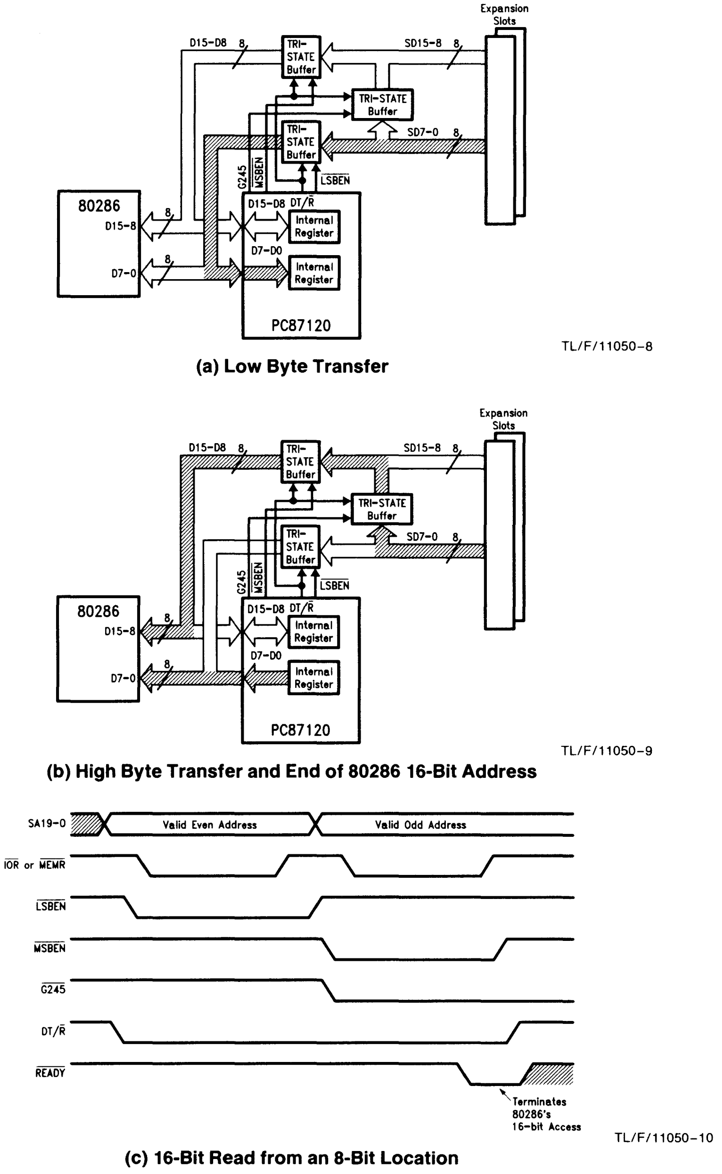
TL/F/11050-7

FIGURE 5-2. CPU/AT Bus Interface

5.0 CPU/AT Bus Operations (Continued)

To allow for 16-bit transfers to or from an 8-bit peripheral, the PC87120 uses its G245 signal to enable an external octal bidirectional buffer to allow high-to-low or low-to-high byte transfers on the expansion data bus. *Figure 5-3* shows the operation of a 16-bit read from an 8-bit peripheral. *Figure 5-3a* shows that during the first 8-bit read (reading from the lower byte), the data byte is latched into a register inside the PC87120. This internal register allows the designer to use an inexpensive bidirectional TRI-STATE buffer instead of a bidirectional TRI-STATE octal register for the systems least significant data byte.

The low byte read is followed immediately by the 8-bit read of the high byte as shown in *Figure 5-3b*. $\overline{\text{G245}}$ enables SD7–0 to drive SD15–8, which in turn drives D15–8 (enabled by $\overline{\text{MSBEN}}$). At the same time the latched lower byte is driven out of the PC87120 and onto the local data bus D7–0 ($\overline{\text{LSBEN}}$ is inactive high to prevent bus contention). *Figure 5-3c* gives the timing of this transfer and indicates that the 80286 reads its data pins D15–0 at the end of the second 8-bit access.



TL/F/11050–8

(a) Low Byte Transfer

TL/F/11050–9

(b) High Byte Transfer and End of 80286 16-Bit Address

TL/F/11050–10

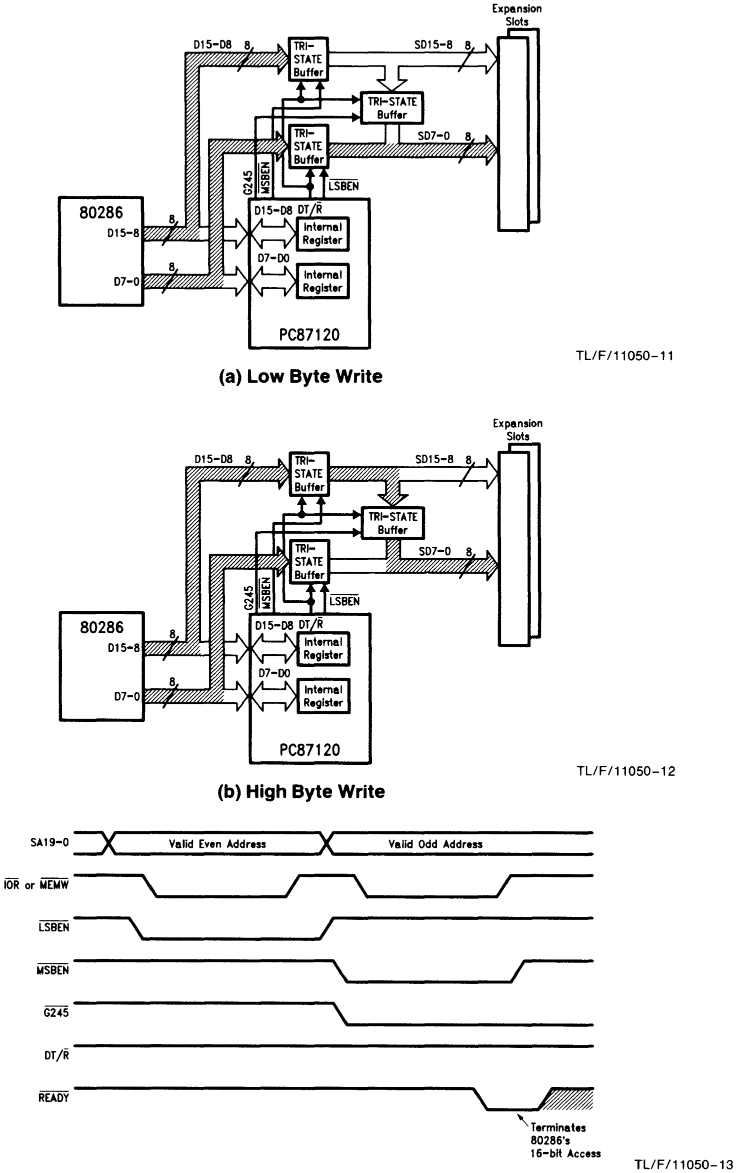
(c) 16-Bit Read from an 8-Bit Location

FIGURE 5-3. 16-Bit Read Operation from an 8-Bit Peripheral

5.0 CPU/AT Bus Operations (Continued)

Figure 5-4 shows the operation of a 16-bit write to an 8-bit peripheral. *Figure 5-4a* shows the low byte data write ($\overline{\text{LSBEN}}$ active low, $\overline{\text{MSBEN}}$ and $\overline{\text{G245}}$ inactive high). The high byte data write, shown in *Figure 5-4b*, immediately follows the low byte write. During this write both $\overline{\text{MSBEN}}$ and $\overline{\text{G245}}$ will be active low, driving D15–0 to SD15–8 and then to SD7–0 ($\overline{\text{LSBEN}}$ will be inactive high to prevent bus contention). The timing diagram for a 16-bit write to an 8-bit peripheral is given in *Figure 5-4c*.

This method of transferring high-to-low byte (i.e., writes) or low-to-high byte (i.e., reads) is also used for all 80286 8-bit data accesses on the high byte data pins (indicated by A0 = 1 and $\overline{\text{BHE}}$ active low). *Figure 5-5a* shows a high byte read from the expansion bus, and *Figure 5-5b* shows a high byte write to the expansion bus.



TL/F/11050–11

(a) Low Byte Write

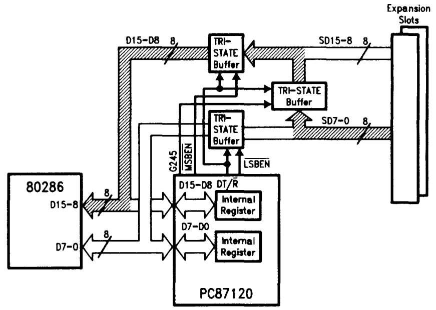
TL/F/11050–12

(b) High Byte Write

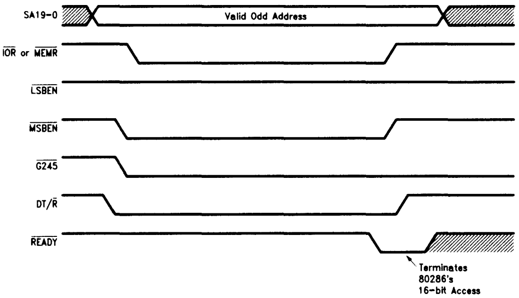
TL/F/11050–13

(c) 16-Bit Write from an 8-Bit Location

FIGURE 5-4. 16-Bit Write Operation to an 8-Bit Peripheral

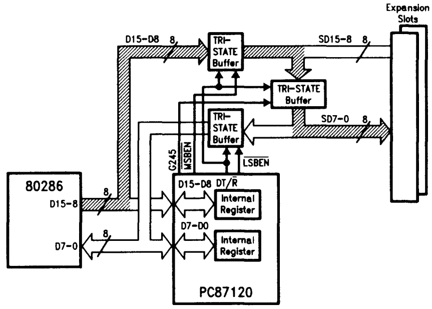


TL/F/11050-14

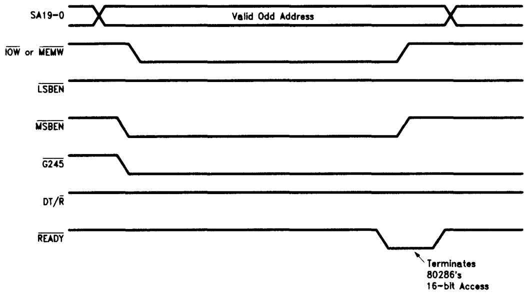


TL/F/11050-15

(a) High Byte Read



TL/F/11050-16



TL/F/11050-17

(b) High Byte Write

FIGURE 5-5. High Byte Read/Write to and from Expansion Bus

5.0 CPU/AT Bus Operations (Continued)

5.3 CPU/AT BUS MODES

This section describes the four modes in which the PC87120 interfaces the CPU and AT-bus.

5.3.1 Non-Turbo Mode

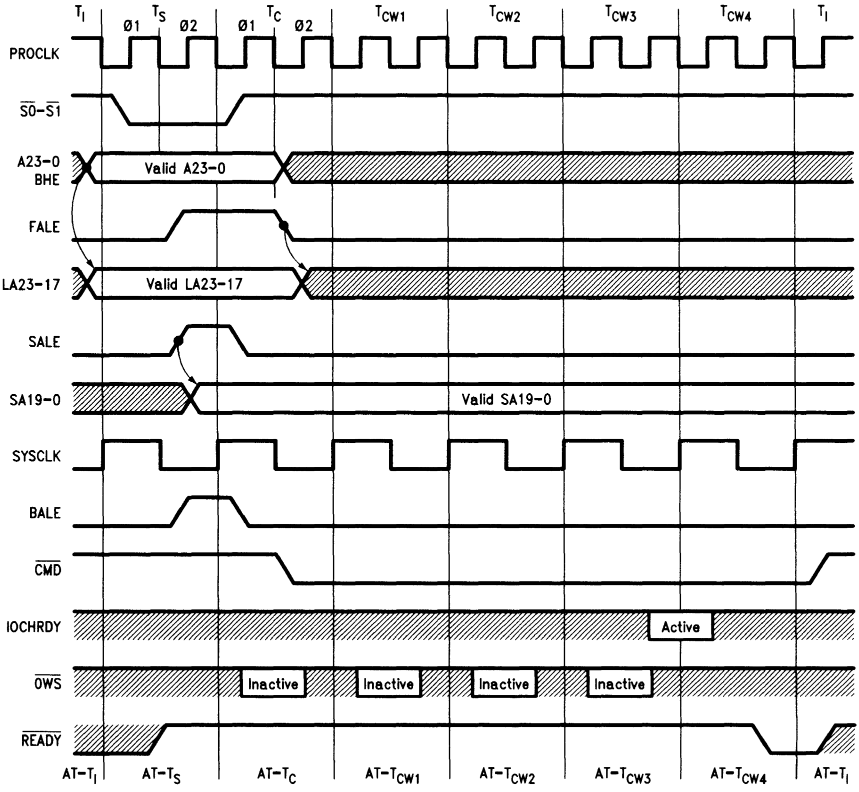
In this mode PROCLK = CLK2IN/2 and SYSCLK = CLK2IN/4. The PC87120's CPU and AT-bus state machines run at the same frequency and are totally synchronous. This means that every 80286 T-state directly corresponds to an expansion bus T-state (AT-T).

Figures 5-6 and *5-7* are the timing diagrams for an 8-bit and a 16-bit access to the expansion bus in Non-Turbo mode. At the top of the figures, the 80286's T-states are shown, while at the bottom the corresponding expansion bus T-states are shown.

As shown, the PC87120 drives BALE, SALE and FALE active in $T_{S\phi2}$. Both BALE and SALE return inactive at the end of T_S, while FALE remains active until the middle of AT-T_C (the same T-state as T_C in this mode). Meanwhile the PC87120 also decodes A23-0, M/$\overline{IO}$, $\overline{S0}$, $\overline{S1}$ and $\overline{BHE}$ during $T_{S\phi2}$ to determine the type and size of the expansion bus access. If the 80286 is making a 16-bit access, the PC87120 monitors the $\overline{MEMCS16}$ and $\overline{IOCS16}$ inputs at the beginning and middle of AT-T_C respectively. If the appropriate 16-bit select signal is driven active low, a normal 16-bit access will be performed, otherwise a 16-bit to 8-bit transfer takes place.

In AT-T_C the PC87120 drives the appropriate command strobe ($\overline{IOR}$, $\overline{IOW}$, $\overline{MEMR}$, or $\overline{MEMW}$) active low for every expansion or peripheral bus access. For all IO accesses as well as 8-bit memory accesses to 8-bit expansion boards, a command delay equal to half an AT-bus T-state is inserted before the command strobe becomes active in the middle of AT-T_C. This command delay supplies a longer system address (SA19–0) set-up time to the command strobe. This extra set-up time is necessary for compatibility wth slower IO or 8-bit boards. In the case of a 16-bit memory access, the command strobe is driven active low at the beginning of AT-T_C, as indicated by the dashed line in *Figure 5-7*.

In the middle of AT-T_C and every AT-bus wait state (AT-T_{CW}) that follows, the zero Wait State signal ($\overline{0WS}$) is checked. If $\overline{0WS}$ is active low, the PC87120 terminates the access cycle at the end of the AT-bus T-state by driving the command strobe inactive high and asserting $\overline{READY}$ to the 80286. An active low $\overline{0WS}$ signal will terminate any expansion bus access, overriding programmed wait states and IOCHRDY. On the other hand, the access is unaffected if $\overline{0WS}$ remains inactive high throughout the access.

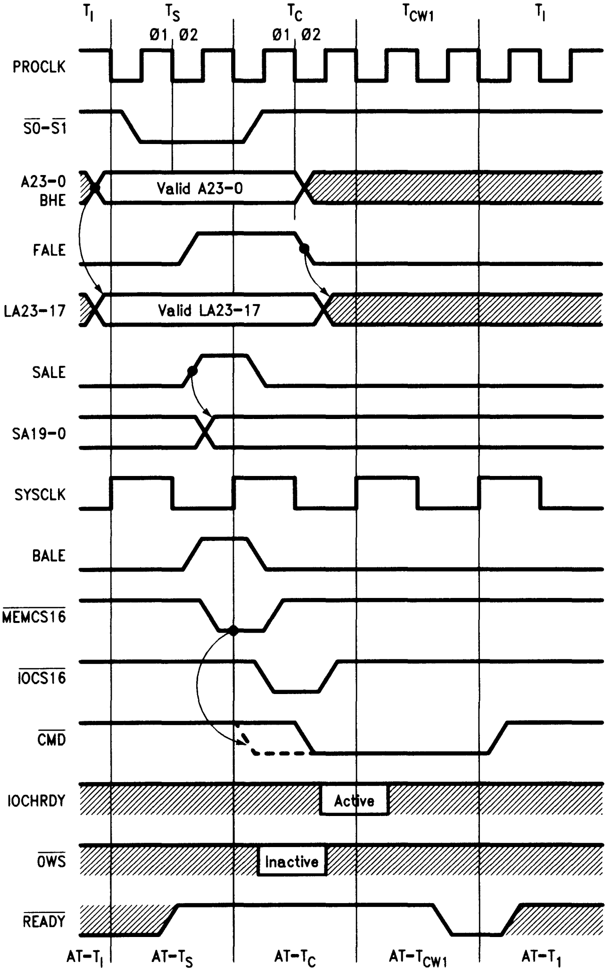


TL/F/11050-18

FIGURE 5-6. 8-Bit Access Cycle in Non-Turbo Mode
(1 Command Delay, 4 Programmed Wait States)

At the end of AT-T_C the appropriate number of programmed wait states will be added. Bits 5 and 6 of the CPU/AT Bus Control Register (IO address FC80h) allow 2 to 5 programmed wait states for the 8-bit AT-bus accesses. Bits 3 and 4 of the same register can be set to provide 0 to 3 programmed wait states for 16-bit AT-bus accesses. For more information refer to the Register Description section 3.1.2.

After all of the programmed wait states are inserted, an expansion board may add more wait states by deasserting IOCHRDY low a full AT-bus T-state before the end of the last programmed wait state. If there are no programmed wait states, IOCHRDY must be deasserted low at the end of AT-T_S. If IOCHRDY remains low, the access is extended indefinitely. When IOCHRDY returns active high, the bus cycle will end within two AT-bus T-states. If both IOCHRDY and $\overline{0WS}$ are low, the $\overline{0WS}$ signal takes precedents and terminates the access.



TL/F/11050–19

FIGURE 5.7 Access to a 16-Bit Board in Non-Turbo Mode
(1 Programmed Wait State)

5.0 CPU/AT Bus Operations (Continued)

5.3.2 Turbo Bus Mode

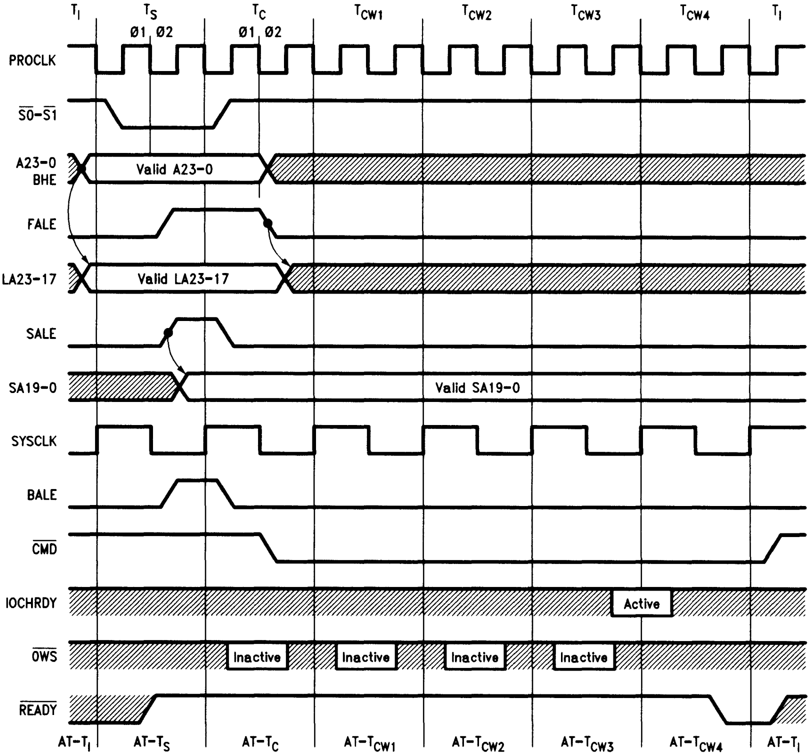
In this mode PROCLK = CLK2IN and SYSCLK = CLK2IN/2. The PC87120's CPU and AT-bus state machines run at the same frequency and are totally synchronous. This means that every 80286 T-state directly corresponds to an expansion bus T-state (AT-T). The relationships between signals in the Turbo Bus mode and the Non-Turbo mode are exactly the same. When switching from Non-Turbo mode to Turbo Bus mode, the frequency of the 80286 and AT-bus doubles. If the final frequency is greater than 8 MHz, the expansion bus is running faster than compatible speed. In some applications, the designer may not be concerned with expansion bus compatibility and in those cases Turbo Bus mode provides a means to run the 80286 and expansion bus synchronously at speeds greater than 8 MHz.

Figures 5-8 and *5-9* are the timing diagrams for an 8-bit and a 16-bit access to the expansion bus in Turbo Bus mode. At the top of the figure the 80286's T-states are shown, while at the bottom, the corresponding expansion bus T-states are shown.

As shown, the PC87120 drives BALE, SALE and FALE active in $T_{S\phi2}$. Both BALE and SALE return inactive at the end of T_S, while FALE remains active until the middle of AT-T_C (the same T-state as T_C in this mode). Meanwhile the PC87120 also decodes A23-0, M/$\overline{\text{IO}}$, $\overline{\text{S0}}$, $\overline{\text{S1}}$ and $\overline{\text{BHE}}$ during $T_{S\phi2}$ to determine the type and size of the expansion bus access. If the 80286 is making a 16-bit access, the PC87120 monitors the $\overline{\text{MEMCS16}}$ and $\overline{\text{IOCS16}}$ inputs at the beginning and middle of AT-T_C respectively. If the appropriate 16-bit select signal is driven active low, a normal 16-bit access will be performed, otherwise a 16-bit to 8-bit transfer takes place.

In AT-T_C, the PC87120 drives the appropriate command strobe ($\overline{\text{IOR}}$, $\overline{\text{IOW}}$, $\overline{\text{MEMR}}$, or $\overline{\text{MEMW}}$) active low for every expansion or peripheral bus access. For all IO accesses as well as 8-bit accesses to 8-bit expansion boards, a command delay equal to half an AT-bus T-state is inserted before the command strobe becomes active in the middle of AT-T_C. This command delay supplies a longer system address (SA19-0) set-up time to the command strobe. This extra set-up time is necessary for compatibility with slower IO or 8-bit boards. In the case of a 16-bit memory access, the command strobe is driven active low at the beginning of AT-T_C, as indicated by the dashed line in *Figure 5-9*.

In the middle of AT-T_C and every AT-bus wait state (AT-T_{CW}) that follows, the zero Wait State signal ($\overline{\text{0WS}}$) is checked. If $\overline{\text{0WS}}$ is active low, the PC87120 terminates the access cycle at the end of the AT-bus T-state by driving the command strobe inactive high and asserting $\overline{\text{READY}}$ to the 80286. An active low $\overline{\text{0WS}}$ signal will terminate any expansion bus access, overriding programmed wait states and IOCHRDY. On the other hand, the access is unaffected if $\overline{\text{0WS}}$ remains inactive high throughout the access.

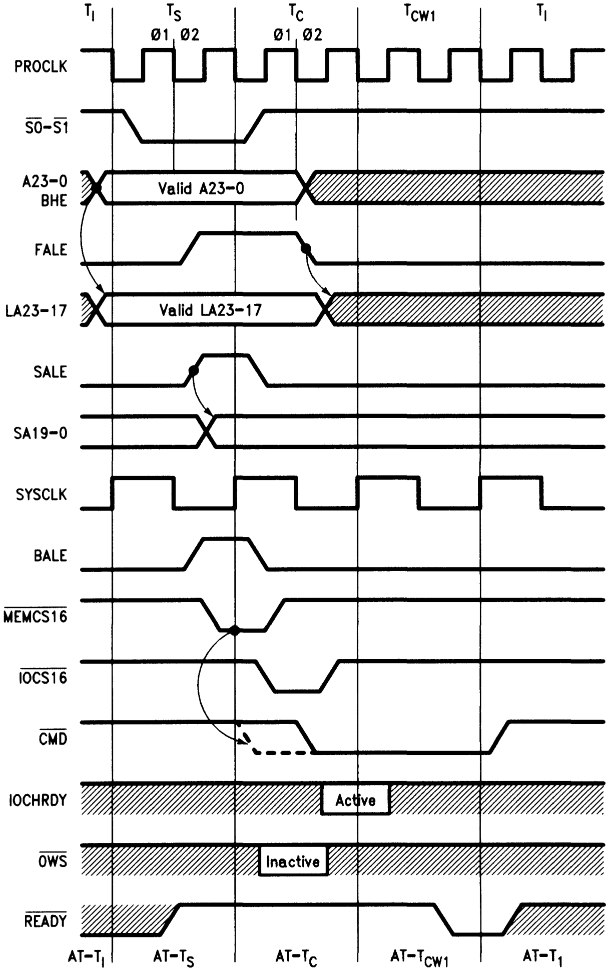


TL/F/11050-20

FIGURE 5-8. 8-Bit Access Cycle In Turbo Bus Mode
(1 Command Delay, 4 Programmed Wait States)

At the end of AT-T_C the appropriate number of programmed wait states will be added. Bits 5 and 6 of the CPU/AT Bus Control Register (IO address FC80h) allow 2 to 5 programmed wait states for the 8-bit AT-bus accesses. Bits 3 and 4 of the same register can be set to provide 0 to 3 programmed wait states for 16-bit AT-bus accesses. For more information refer to the Register Description section 3.1.2.

After all of the programmed wait states are inserted, an expansion board may add more wait states by deasserting IOCHRDY low a full AT-bus T-state before the end of the last programmed wait state. If there are no programmed wait states, IOCHRDY must be deasserted low at the end of AT-T_S. If IOCHRDY remains low, the access is extended indefinitely. When IOCHRDY returns active high, the bus cycle will end within two AT-bus T-states. If both IOCHRDY and $\overline{0WS}$ are low, the $\overline{0WS}$ signal takes precedence and terminates the access.



TL/F/11050-21

FIGURE 5-9. Accesses to a 16-Bit Board in Turbo Bus Mode
(1 Programmed Wait State)

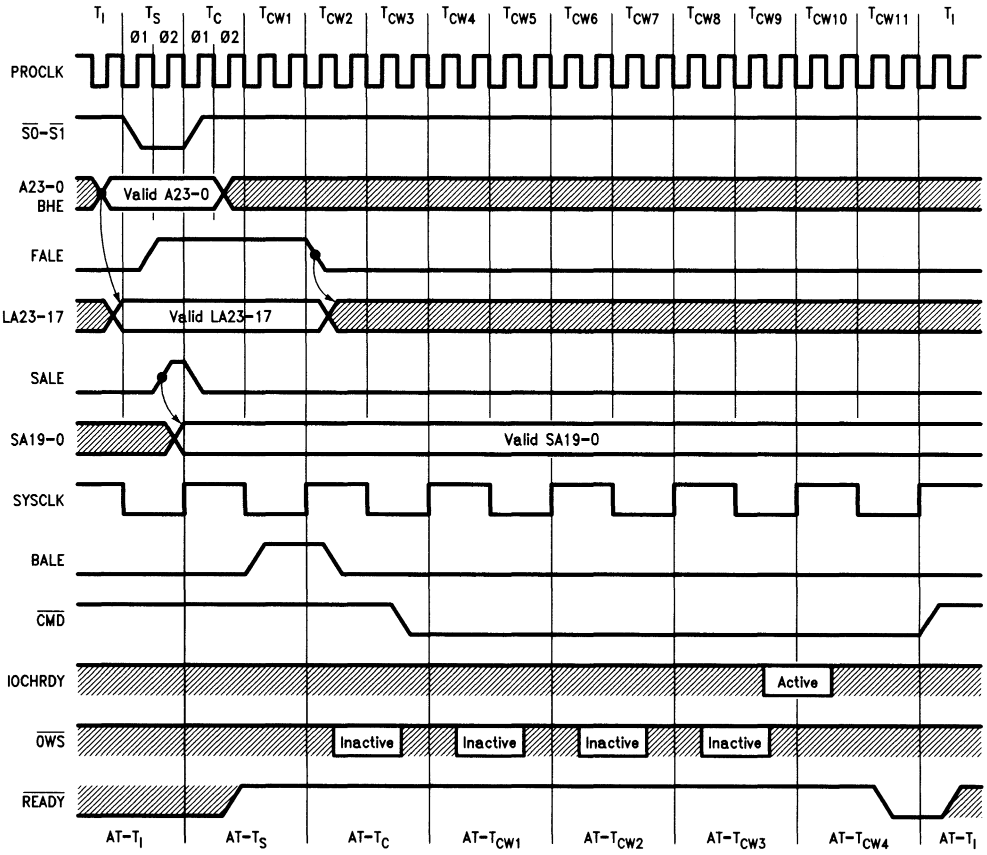
5.3.3 Normal Mode

In this mode PROCLK = CLK2IN and SYSCLK = CLK2IN/4. The PC87120's CPU state machines run at twice the frequency of the AT-bus state machine, indicating that the 80286 runs twice as fast as the expansion bus. This mode provides support for a 16 MHz 80286, while still running the system expansion bus at a compatible 8 MHz. When going from Non-Turbo mode to Normal mode via a software or hardware switch, the speed of the expansion bus remains a constant while the 80286's frequency doubles.

Figures 5-10 and *5-11* are the timing diagrams for typical expansion bus accesses in Normal mode. At the top of the figures the 80286's T-states are shown, while at the bottom are the expansion bus T-states. During every expansion bus access the PC87120 inserts CPU wait states to slow the 80286 down to the expansion bus's speed.

As shown, the PC87120 drives SALE and FALE active in T_{SP2}. Only SALE goes inactive at the end of T_S, while FALE remains active until the middle of AT-T_C. In this mode the expansion bus's buffered address latch enable signal, BALE, is driven active during the second half of AT-T_S. Meanwhile the PC87120's CPU state machine decodes A23–0, $M/\overline{IO}$, $\overline{S0}$, $\overline{S1}$ and $\overline{BHE}$ during $T_{S\phi2}$ to determine the type and size of the expansion bus access. If the 80286 is making a 16-bit access, the PC87120 monitors the $\overline{MEMCS16}$ and $\overline{IOCS16}$ inputs at the beginning and middle of AT-T_C respectively. If the appropriate 16-bit select signal is driven active low, a normal 16-bit access will be performed, otherwise a 16-bit to 8-bit transfer takes place.

In AT-T_C, the PC87120 drives the appropriate command strobe ($\overline{IOR}$, $\overline{IOW}$, $\overline{MEMR}$, or $\overline{MEMW}$) active low for every expansion or peripheral bus access. For all IO accesses as well as 8-bit memory accesses to 8-bit expansion boards, a command delay equal to half an AT-bus T-state is inserted before the command strobe becomes active in the middle of AT-T_C. This command delay supplies a longer system address (SA19-0) set-up time to the command strobe. This extra set-up time is necessary for compatibility with slower IO or 8-bit boards. In the case of a 16-bit memory access, the command strobe is driven active low at the beginning of AT-T_C, as indicated by the dashed line in *Figure 5-11*.



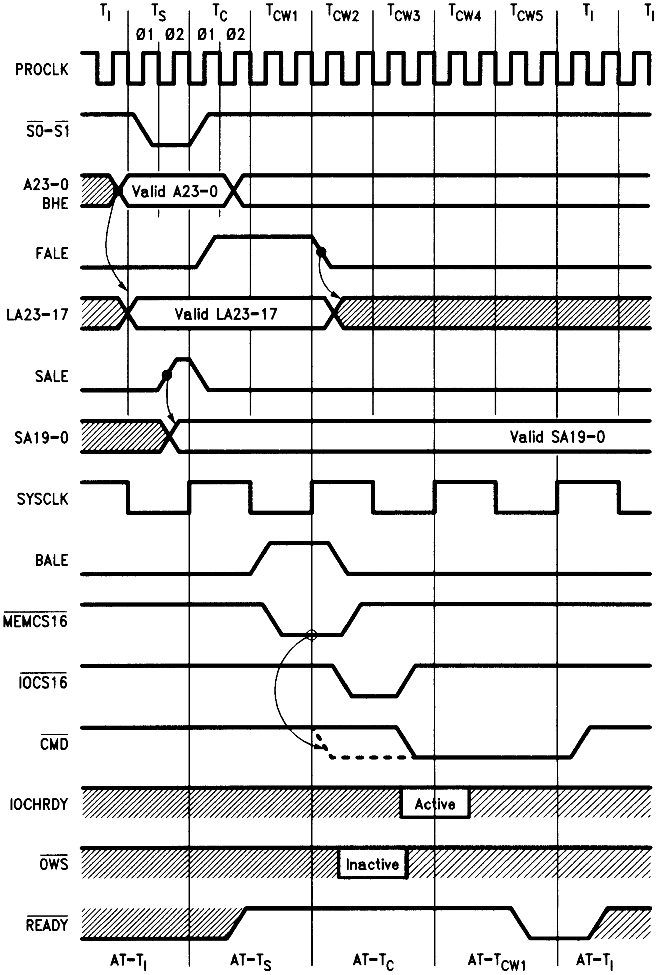
TL/F/11050-22

FIGURE 5-10. 8-Bit Access Cycle In Normal Mode
(1 Command Delay, 4 Programmed Wait States)

In the middle of AT-T_C and every AT-bus wait state (AT-T_{CW}) that follows, the zero Wait State signal ($\overline{0WS}$) is checked. If $\overline{0WS}$ is active low, the PC87120 terminates the access cycle at the end of the AT-bus T-state by driving the command strobe inactive high and asserting $\overline{READY}$ to the 80286. An active low $\overline{0WS}$ signal will terminate any expansion bus access, overriding programmed wait states and IOCHRDY. On the other hand, the access is unaffected if $\overline{0WS}$ remains inactive high throughout the access.

At the end of AT-T_C the appropriate number of programmed wait states will be added. Each wait state is equal to one expansion bus T-state. In this mode, that is equivalent to two CPU wait states. Bits 5 and 6 of the CPU/AT Bus Control Register (IO address FC80h) allow 2 to 5 programmed wait states for the 8-bit AT-bus accesses. Bits 3 and 4 of the same register can be set to provide 0 to 3 programmed wait states for 16-bit AT-bus accesses. For more information refer to the Register Description section 3.1.2.

After all of the programmed wait states are inserted, an expansion board may add more wait states by deasserting IOCHRDY low a full AT-bus T-state before the end of the last programmed wait state. If there are no programmed wait states, IOCHRDY must be deasserted low at the end of AT-Ts. If IOCHRDY remains low, the access is extended indefinitely. When IOCHRDY returns active high, the bus cycle will end within two AT-bus T-states. If both IOCHRDY and $\overline{0WS}$ are low, the $\overline{0WS}$ signal take precedence and terminates the access.



TL/F/11050-23

FIGURE 5-11. Access to a 16-Bit Board In Normal Mode
(1 Programmed Wait State)

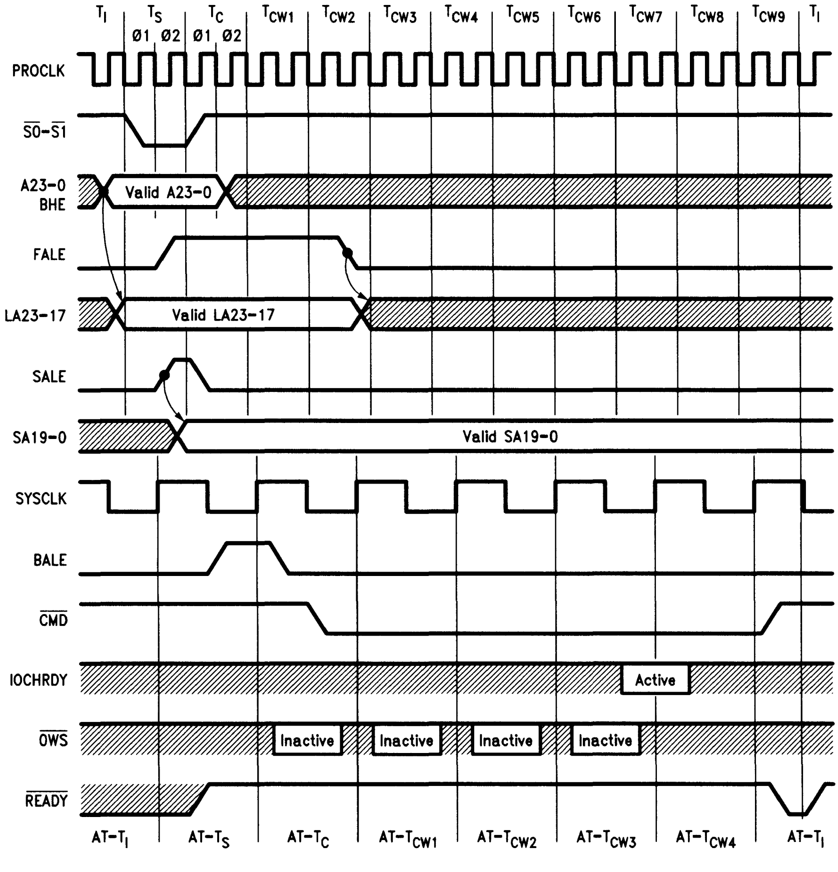
5.3.4 Asynchronous Mode

In this mode PROCLK = CLK2IN and SYSCLK = ATCLK/2. The PC87120's CPU and AT-bus state machines run totally asynchronously in this mode, indicating that the 80286 will run asynchronously to the expansion bus. This provides support for running the 80286 at any frequency between 8 MHz and 16 MHz (typically 10 MHz or 12.5 MHz), but still allows the system expansion bus to run at 8 MHz (i.e., ATCLK driven with a 16 MHz TTL oscillator).

Figures 5-12 and *5-13* are the timing diagrams for typical expansion bus accesses in Non-Turbo mode. At the top of the figures the 80286's T-states are shown, while at the bottom are the expansion bus T-states. During every expansion bus access the PC87120 inserts CPU wait states to slow the 80286 down to the speed of the expansion bus. One should also note that the expansion bus' cycle always begins after and finishes before the corresponding 80286 cycle. This guarantees the proper transfer of data.

As shown, the PC87120 drives SALE and FALE active in $T_{S\phi2}$. Only SALE returns inactive at the end of T_S, while FALE remains active until the middle of AT-T_C. In this mode the expansion bus's buffered address latch enable signal, BALE, is driven active during the second half of AT-T_S. Meanwhile the PC87120's CPU state machine decodes A23-0, M/$\overline{\text{IO}}$, $\overline{\text{S0}}$, $\overline{\text{S1}}$ and $\overline{\text{BHE}}$ during $T_{S\phi2}$ to determine the type and size of the expansion bus access. If the 80286 is making a 16-bit access, the PC87120 monitors the $\overline{\text{MEMCS16}}$ and $\overline{\text{IOCS16}}$ inputs at the beginning and middle of AT-T_C respectively. If the appropriate 16-bit select signal is driven active low, a normal 16-bit access will be performed, otherwise a 16-bit to 8-bit transfer takes place.

In AT-T_C, the PC87120 drives the appropriate command strobe ($\overline{\text{IOR}}$, $\overline{\text{IOW}}$, $\overline{\text{MEMR}}$, or $\overline{\text{MEMW}}$) active low for every expansion or peripheral bus access. For all IO accesses as well as 8-bit memory accesses to 8-bit expansion boards, a command delay equal to half a AT-bus T-state is inserted



TL/F/11050-24

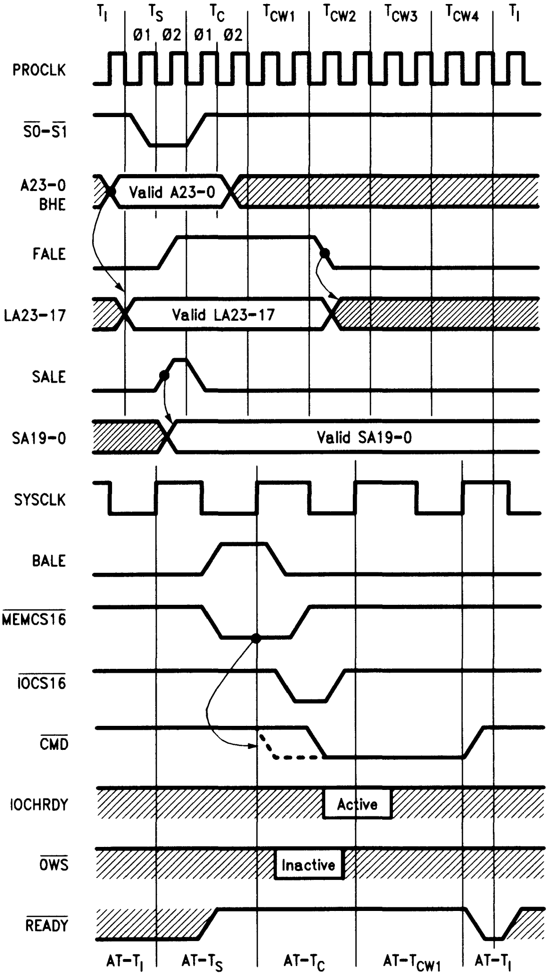
FIGURE 5-12. 8-Bit Access Cycle in Asynchronous Mode
(1 Command Delay, 4 Programmed Wait States)

before the command strobe becomes active in the middle of AT-T_C. This command delay supplies a longer system address (SA19–0) set-up time to the command strobe. This extra set-up time is necessary for compatibility with slower IO or 8-bit boards. In the case of a 16-bit memory access, the command strobe is driven active low at the beginning of AT-T_C, as indicated by the dashed line in *Figure 5-13.*

In the middle of AT-T_C and every AT-bus wait state (AT-T_{CW}) that follows, the zero Wait State signal ($\overline{0WS}$) is checked. If $\overline{0WS}$ is active low, the PC87120 terminates the access cycle at the end of the AT-bus T-state by driving the command strobe inactive high and asserting $\overline{READY}$ to the 80286. An active low $\overline{0WS}$ signal will terminate any expansion bus access, overriding programmed wait states and IOCHRDY. On the other hand, the access is unaffected if $\overline{0WS}$ remains inactive high throughout the access.

At the end of AT-T_C the appropriate number of programmed wait states will be added. Each wait state is equal to one expansion bus T-state. Bits 5 and 6 of the CPU/AT Bus Control Register (IO address FC80h) allow 2 to 5 programmed wait states for the 8-bit AT-bus accesses. Bits 3 and 4 of the same register can be set to provide 0 to 3 programmed wait states for 16-bit AT-bus accesses. For more information refer to the Register Description section 3.1.2.

After all of the programmed wait states are inserted, an expansion board may add more wait states by deasserting IOCHRDY low a full AT-bus T-state before the end of the last programmed wait state. If there are no programmed wait states, IOCHRDY must be deasserted low at the end of AT-Ts. If IOCHRDY remains low, the access is extended indefinitely. When IOCHRDY returns active high, the bus cy-

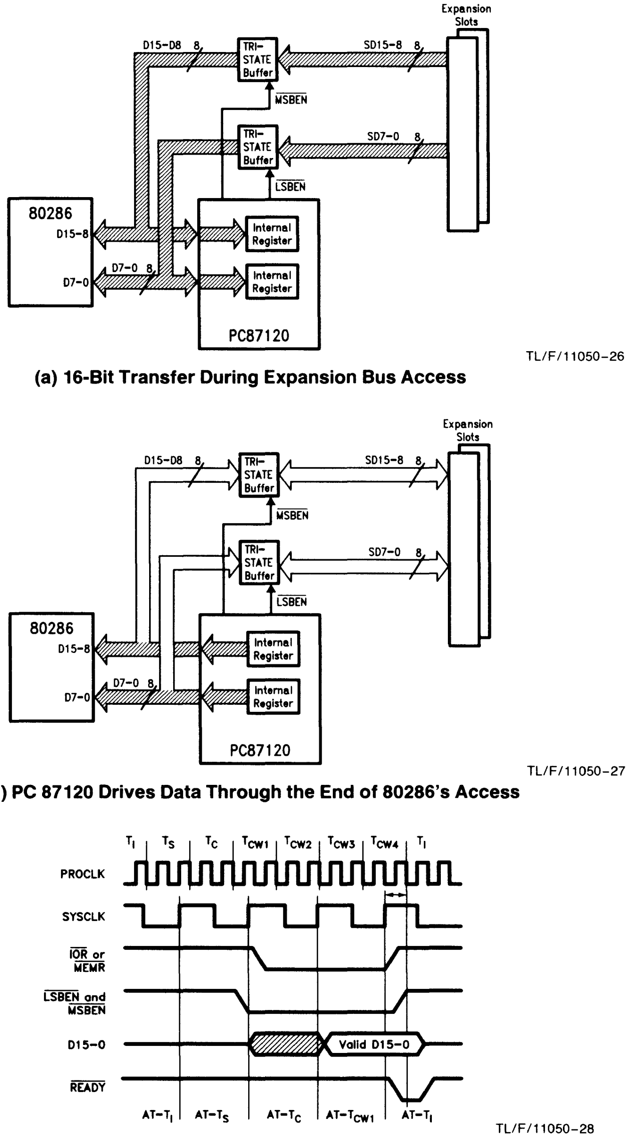


TL/F/11050–25

FIGURE 5-13. Access to a 16-Bit Memory Board in Asynchronous Mode
(1 Programmed Wait State)

cle will end within two AT-bus T-states. If both IOCHRDY and $\overline{\text{0WS}}$ are low, the $\overline{\text{0WS}}$ signal takes precedence and terminates the access.

Figure 5-14 shows the operation of a 16-bit read in asynchronous mode. At the end of a read cycle, the expansion bus cycle may end a maximum of a full CPU T-state before the 80286 finishes its read. To guarantee that the correct data is read by the 80286, the data is latched into two internal PC87120 registers at the end of the expansion bus cycle as shown in *Figure 5-14a*. Then this data is immediately driven back out onto the local data bus. Meanwhile $\overline{\text{MSBEN}}$ and $\overline{\text{LSBEN}}$ return inactive high to prevent bus contention. The PC87120 drives the latched data onto the local data bus until the 80286's read cycle is terminated as shown in *Figure 5-14b*. The timing for this access is shown in *Figure 5-14c*. By providing these two internal registers, the PC87120 allows the designer to use two inexpensive bidirectional TRI-STATE buffers instead of two bidirectional TRI-STATE octal registers for the system data bus buffers.



(a) 16-Bit Transfer During Expansion Bus Access

(b) PC 87120 Drives Data Through the End of 80286's Access

(c) 16-Bit Read Cycle in Asynchronous Mode

FIGURE 5-14. 16-Bit Read Operation in Asynchronous Mode

6.0 Memory Controller

The PC87120's programmable memory controller may be configured to support 512 kB to 8 MB of parity checked motherboard memory distributed in up to four 16-bit wide banks. The memory controller is a state machine which uses PROCLK as its fundamental timing source. PROCLK also determines the operation frequency of the 80286 and the PC87120's CPU state machine. As a result the 80286 and the memory controller run synchronously, providing maximum performance at every operating frequency.

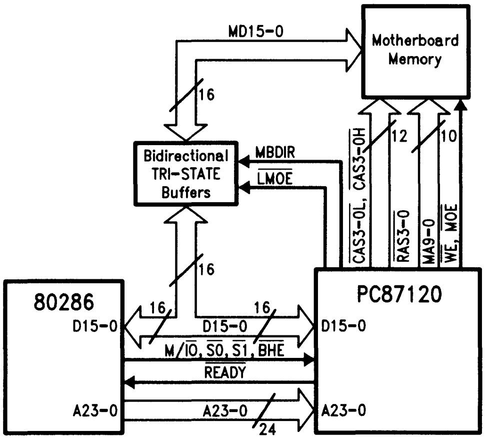
Like the CPU state machine, the memory controller traces every T-state of the 80286. During the status T-state (T_S) of a memory access, the memory controller decodes the local address A23-0 and determines if the access is to the configured motherboard memory. The memory controller ignores all non-motherboard memory accesses and continues to drive the memory control signals at their present states. On the other hand, for all motherboard memory accesses the memory controller provides the correct multiplexed address and generates the appropriate row and column address strobes. In addition the proper write strobe, memory output enable signal and bus control signals are produced. The timing of these access cycles depend on the mode of operation and is described in detail in section 6.4.

6.1 MEMORY DATA BUS INTERFACE

The PC87120's memory controller supports motherboard memory buffered onto the 80286's data bus (D15-0), via two bidirectional buffers controlled by MBDIR and $\overline{\text{LMOE}}$. *Figure 6-1* shows the typical data bus interface.

MBDIR equals 1 during a CPU memory write, allowing data to flow from the local data bus (D15-0) to the memory data bus (MD15-0). During a 80286 memory read cycle MBDIR equals 0 and data flows in the opposite direction, MD15-0 to D15-0.

The local memory output enable signal ($\overline{\text{LMOE}}$) enables the bidirectional buffers whenever the 80286 performs a motherboard memory access. Table 6-1 shows how the DRAM configuration bits (Bits 5-3 of the RAM/ROM Configuration Register #1, located at IO address FC81h) determine the range of addressable motherboard memory. Any memory access outside of the configured motherboard memory range will be addressed to the expansion bus, resulting in an expansion bus memory access.


MD15-0
Motherboard Memory
16
Bidirectional TRI-STATE Buffers
MBDIR
$\overline{\text{LMOE}}$
CAS3-0L, CAS3-0H
12
10
RAS3-0
MA9-0
$\overline{\text{WE}}$, $\overline{\text{MOE}}$
16
80286
D15-0
16
16
D15-0
M/$\overline{\text{IO}}$, $\overline{\text{S0}}$, $\overline{\text{S1}}$, $\overline{\text{BHE}}$
$\overline{\text{READY}}$
A23-0
A23-0
24
PC87120
D15-0
A23-0

TL/F/11050-29

FIGURE 6-1. Typical Memory Data Bus Interface

TABLE 6-1. Motherboard Memory Ranges

RAM/ROM Configuration Register #1			Configuration Type	Motherboard Memory Range
Bit 5	Bit 4	Bit 3		
0	0	0	1	0-512 kB
0	0	1	2	0-2 MB
0	1	0	3	0-1 MB
0	1	1	4	0-4 MB
1	0	0	5	0-2 MB
1	0	1	6	0-5 MB
1	1	0	7	0-8 MB
1	1	1	8	0-640 kB

6.2 SHADOW RAM SUPPORT

The memory address range from 640 kB to 1 MB is reserved for system ROM BIOS, optional system ROMs, video ROM BIOS and video RAM. After power-up the DRAM located in the 640 kB to 1 MB memory address range will be unused.

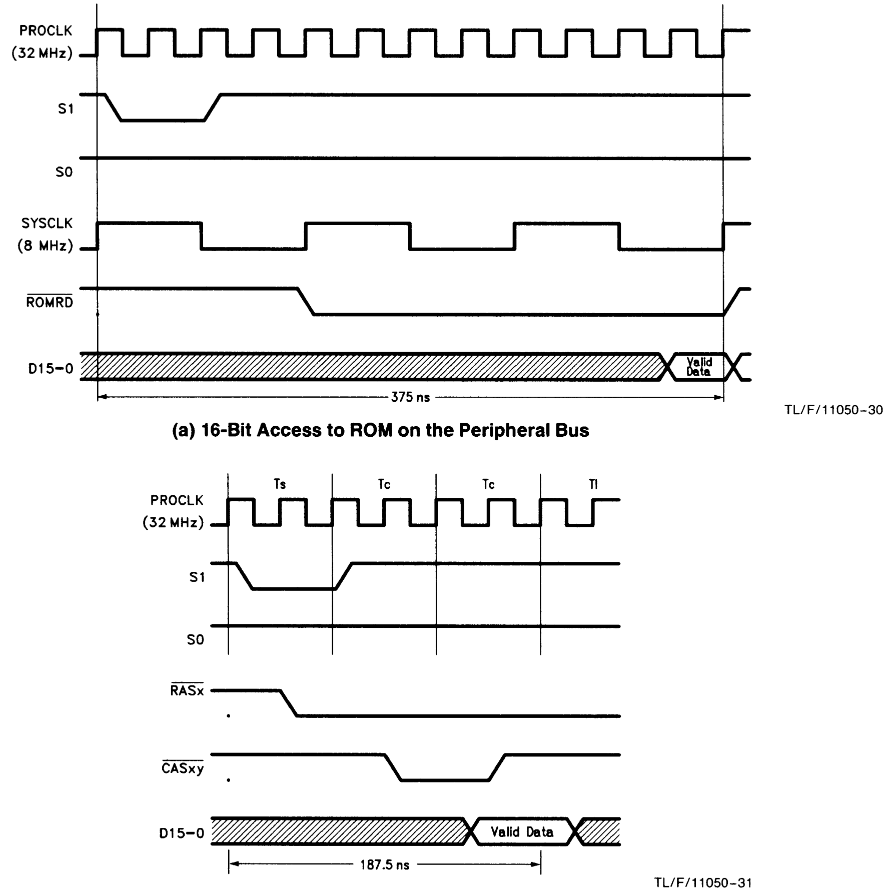
When a minimum of 1 MB of motherboard memory is provided, the PC87120 provides support for copying the BIOS out of slow, cost effective EPROMs and into a duplicate address range in motherboard DRAM. Bits 7-6 and 3-2 of the ROM Area Enable/Shadow RAM Mode Register (located at IO address FC84h) control the access privileges to the Video BIOS and system BIOS shadow RAM, respectively. These bit pairs determine if accesses to the shadow RAM are disabled, read only, write only or both read/write. During the copying process, the shadow RAM area being written to, must have its access privilege set to write only or read/write. The read only or read/write option must be used if the shadow RAM is to be read during BIOS read cycles.

After the BIOS is copied into this shadow RAM the ROM BIOS may be disabled by writing ones to the ROM Enable bits (Bits 1-0 of the ROM Area Enable/Shadow RAM Mode Register, located at IO address FC84h). Then the corresponding shadow RAM may be enabled by writing ones to the appropriate Shadow RAM Enable bits (bits 5-0 of the Shadow RAM Enable Register, located a IO address FC83h). Once the shadow RAM is enabled all accesses to the appropriate memory range will take place at the programmed speed of the DRAM.

The use of shadow RAM can significantly improve performance if slow EPROMs are used. *Figure 6-2a* shows the timing for a 16-bit access to 250 ns EPROM on the peripheral bus. Meanwhile *Figure 6-2b* shows a one wait state Shadow RAM access at 16 MHz.

The Video RAM shadowing operates much like cache memory; the writes are performed on the shadow DRAM and passed to the video adapter's dual port RAM via an expansion bus write cycle. Reads will only access the local shadow RAM and do not require slower expansion bus read cycles.

Refer to sections 3.1.5 and 3.1.6 for more information on the configuration bits that control shadow RAM.



(a) 16-Bit Access to ROM on the Peripheral Bus

(b) 16-Bit Access to Motherboard Memory Using Page Interleaving Mode

FIGURE 6-2. Access Timing for BIOS

6.3 MEMORY REMAPPING

As stated in section 6.2, the memory address range 640 kB to 1 MB is reserved for system ROM BIOS, optional system ROMs, video ROM BIOS and video RAM. The system designer may not wish to use the DRAM located in this address range as shadow RAM. If so, the PC87120 provides a second option: the memory in the 640 kB to 1 MB range may be remapped to the top of the motherboard memory range and used as expanded or extended memory.

When the DRAM Relocation bit (Bit 6 of the RAM/ROM Configuration Register #1, located at IO address FC81h) is set to 1, the 640 kB to 1 MB range is remapped to the top of the motherboard memory. When this bit equals 0, no remapping will take place. Remapping must be disabled if less than 1 MB of DRAM is installed or if shadow RAM is being used.

Table 6-2 shows both the remapped memory range and the addressable motherboard memory when the remapping option is selected.

TABLE 6-2. Remapped Memory Ranges

Configuration Type	Remapped Memory Range	Motherboard Memory Range
1	none	0–512 kB
2	2–2.384 MB	0–640 kB, 1–2.384 MB
3	1–1.382 MB	0–640 kB, 1–1.384 MB
4	4–4.384 MB	0–640 kB, 1–4.384 MB
5	2–2.384 MB	0–640 kB, 1–2.384 MB
6	5–5.384 MB	0–640 kB, 1–5.384 MB
7	8–8.384 MB	0–640 kB, 1–8.384 MB
8	none	0–640 kB

Figures 6-3a, 6-3b and *6-3c* show how the motherboard memory between 640 kB and 1 MB range are remapped when 1, 2, and 8 MB of motherboard memory are configured. Refer to section 3.1.3 for more information about memory configurations.

6.4 MEMORY ACCESS MODES

The PC87120 can be configured to operate in one of three possible memory access modes:

1) Conventional Mode

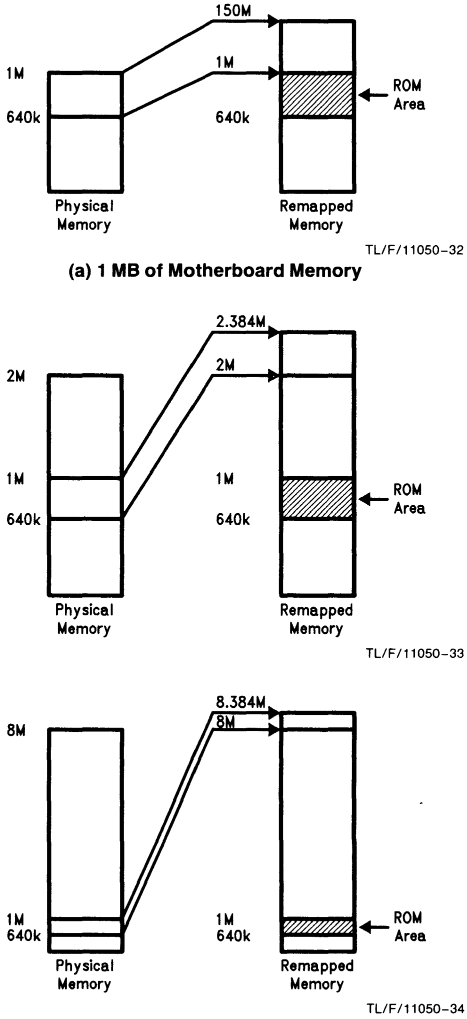
2) Page Interleaving Mode

3) Enhanced Page Interleaving Mode

Each mode provides unique benefits. This allows the designer to select the mode which best suits the system's cost/performance requirements.

Following a system reset, conventional mode is the default mode for memory accesses. To enable one of the page interleaving modes a one (1) must be written to the Page Interleaving Enable bit (bit 7 of the RAM/ROM Configuration Register #1, located at IO address FC81h).

The choice of normal page interleaving or enhanced page interleaving mode is determined by a power-on strapping option. The state of the Interleave Mode Select bit (Bit 7 of the RAM/ROM Configuration Register #2, located at IO address FC82h) is set during a system reset. Once this bit is set, it cannot be changed until the next system reset. (Refer to section 3.3.2 for more information regarding power-on strapping options and the Interleaving Mode Select bit.)



(a) 1 MB of Motherboard Memory

(b) 2 MB of Motherboard Memory

FIGURE 6-3. Remapped Motherboard Memory

The following sections describe the operation of each memory access mode in detail.

6.4.1 Conventional Mode

In conventional mode all of the motherboard memory is treated as one large memory bank and every access to the memory takes the same amount of time. To insert a single wait state into every motherboard memory cycle, the RAM Wait State bit (Bit 2 of the RAM/ROM Configuration Register #1, located at IO address FC81h) must be set to zero (0). Setting this bit to 1 will result in zero wait state accesses to the DRAM in conventional mode.

Figure 6-4 shows a memory read cycle followed by a memory write cycle with one programmed wait state (the default). Every access cycle in this mode begins with all of the memory control signals inactive. MBDIR equal to 1 (i.e., the memory data buffers pointing D15–0 to MD15–0), and the row address multiplexed out onto the memory address lines MA9–0. In both accesses $\overline{\text{RASx}}$ is driven active low in the middle of T_C, latching the row address into the DRAMs.

During a read cycle, MBDIR goes low and $\overline{\text{LMOE}}$ is asserted low during T_C and T_{CW1}, driving MD15–0 onto D15–0. Write enable ($\overline{\text{WE}}$) stays inactive high through the read cycle, and the DRAMs drive their data onto the memory data bus. The low and/or high bit(s) of parity, PARL and PARH respectively, are driven into the PC87120 and along with the appropriate data byte(s) is used to check parity. If parity is correct no action is taken, but if a parity error is detected a non-maskable interrupt (NMI) is generated, if the NMI circuitry is enabled.

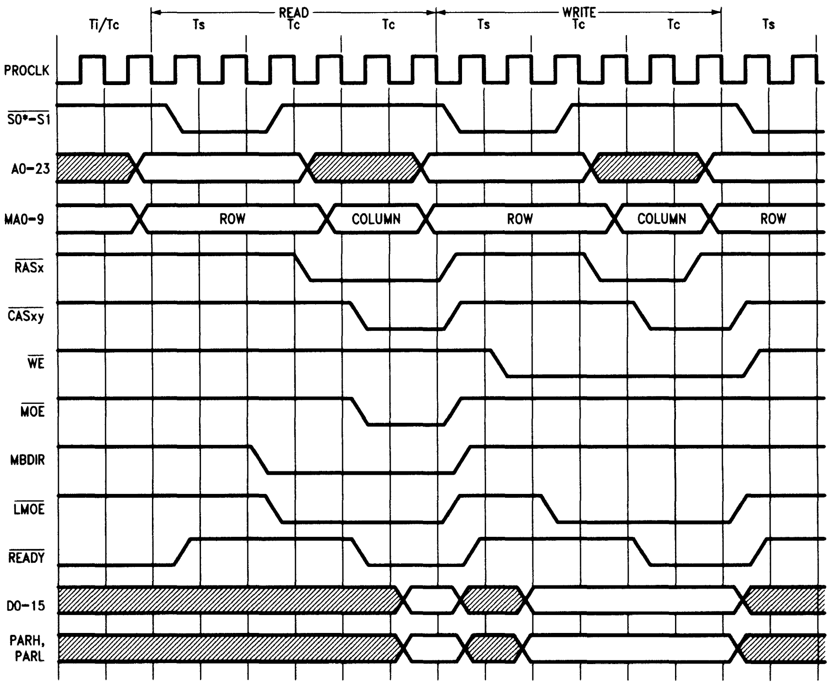
For the write cycle MBDIR stays high and $\overline{\text{LMOE}}$ is again asserted low during T_C and T_{CW1}, driving D15–0 and MD15–0. $\overline{\text{WE}}$ is driven low in the middle of T_S, indicating an early write cycle. The PC87120 determines the appropriate parity for the low and high bytes and drives PARL and PARH out to the DRAMs. When $\overline{\text{CASxy}}$ rises at the end of T_{CW1}, both the data and the parity are latched into the DRAMs.

As the operating frequency of the 80286 increases, the durations of the DRAM access signals shrink proportionally, requiring faster and faster DRAMs. Table 6-3 gives the minimum DRAM access times supported for the common 80286 operating frequencies.

TABLE 6-3. DRAM Access Requirements for Conventional Mode

Operating Frequency of the 80286	Minimum DRAM Access Time Supported (ns)
10 MHz	100 ns
12.5 MHz	80 ns
16 MHz	60 ns

Normally $\overline{\text{RASx}}$ rises a full PROCLK period before $\overline{\text{CASxy}}$. As shown in *Figure 6-4*, $\overline{\text{RASx}}$ rises in the middle of T_{CW1} and stays inactive for a minimum of four (4) PROCLK periods. This amount of time is more than sufficient to guarantee the RAS precharge timing requirements for the DRAMs supported. At the end of the access cycle $\overline{\text{CASxy}}$ rises one PROCLK period after $\overline{\text{RASx}}$ (i.e., at the end of T_{CW1}).



TL/F/11050-35

FIGURE 6-4. Conventional Mode 1 Wait State

6.0 Memory Controller (Continued)

Some DRAMs require $\overline{\text{RASx}}$ and $\overline{\text{CASxy}}$ to rise at the same time. The Extend RAE Access bit, (bit 4 of the RAM/ROM Configuration Register #2, located at IO address FC82h), provides support for DRAMs with this requirement. The state of the Extend RAS Access bit is determined by a power on strapping option during a system reset. Once this bit is set, it cannot be changed until the next system reset. (Refer to Section 3.3.2 for more information about power-on strapping options and the Extend RAS Access bit.) When the Extend RAS Access bit is set to 0, $\overline{\text{RASx}}$ rises one PROCLK period before $\overline{\text{CASxy}}$ (the default), but when it is set to 1 $\overline{\text{RASx}}$ will rise at the same time as $\overline{\text{CASxy}}$.

Figure 6-5 shows a memory read cycle followed by a memory write cycle with one programmed wait state and the Extend RAS Access bit set to 1. $\overline{\text{RASx}}$ and $\overline{\text{CASxy}}$ both rise at the end of T_{CW1}. $\overline{\text{RASx}}$ returns high for a minimum of three PROCLK periods, which is sufficient to meet the RAS precharge timing requirements of the DRAM in Table 6.3.

Figure 6-6 shows a memory read cycle followed by a memory write cycle with no programmed wait states and the Extend RAS Access bit equal to 0. The access timings for a zero wait state access are exactly the same as the access timings for a one wait state access. The only difference is that all of the access signals occur two PROCLK periods earlier. As a result the RAS precharge time shrinks to two PROCLK periods, which remains sufficient to meet the timing requirements of the DRAMs in Table 6-3.

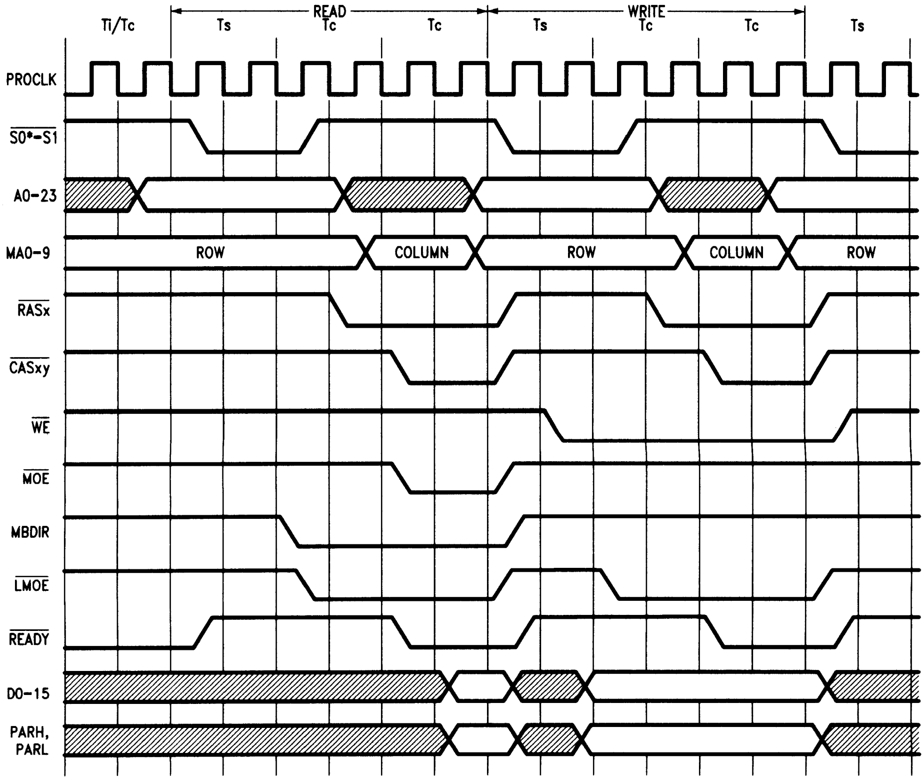
6.4.2 Page Interleaving Mode

Page interleaving is a combination of two methods used to reduce the performance penalty incurred by waiting for the RAS precharge time DRAMs require. It is based on the assumption that most accesses to DRAMs in a 80286 system will be to successive words.

Page mode DRAMs can latch a row address and use it for successive operations with different column addresses. While accesses are on the same "page" and within a maximum RAS low time, the RAS precharge penalty is avoided.

The interleaving feature places successive pages in alternate banks. This reduces the length of sequential accesses to each bank and eliminates the RAS precharge penalty when a new page is addressed in an alternate bank.

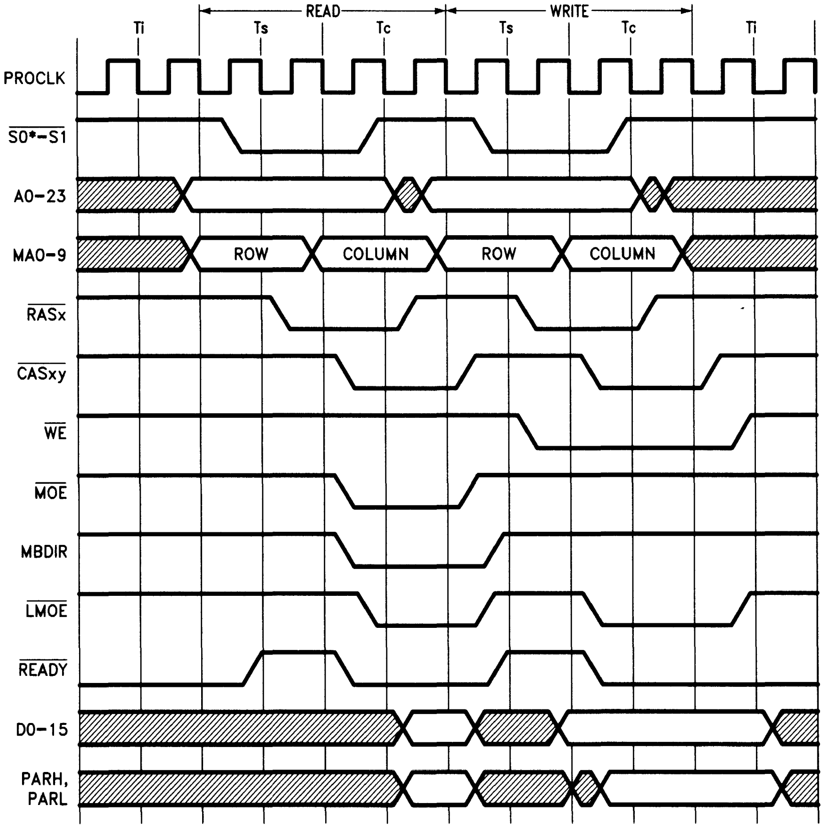
Figure 6-7 shows how pages map into 1, 2 and 4 banks of memory, when page interleaving mode is used with 1 Mb DRAMs. When using 1 M DRAMs each page will be 1 kiloword long (i.e., 2 kB in length). With only one bank of memory as shown in *Figure 6-7a*, interleaving is impossible and by default only page mode accesses are allowed. When two banks are configured as shown in *Figure 6–7b*, successive pages are placed alternately in banks 0 and 1. *Figure 6-7c* shows page interleaving support for four memory banks. In this configuration successive pages are placed alternately in banks 0 and 1 until the memory range of these two banks is exhausted. Then successive pages are placed alternately in banks 2 and 3.



TL/F/11050–36

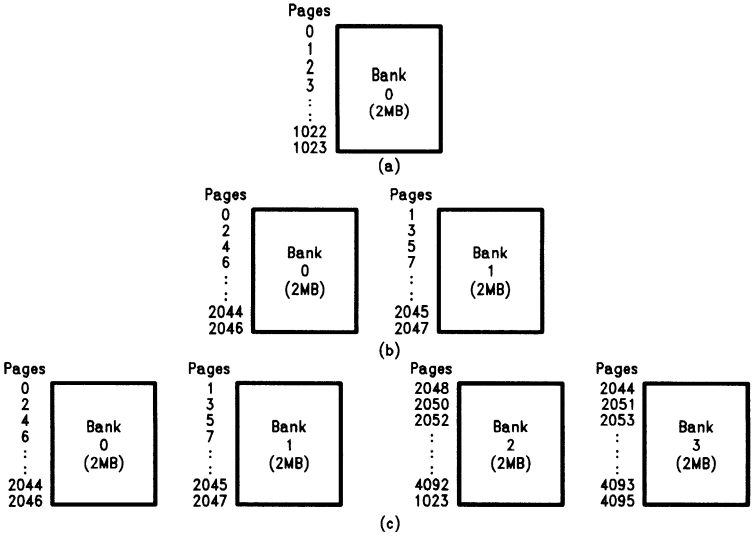
FIGURE 6-5. Conventional Mode 1 Wait State, Extended RAS

6.0 Memory Controller (Continued)



TL/F/11050–37

FIGURE 6-6. Conventional Mode 0 Wait States



TL/F/11050–38

FIGURE 6-7. Page Interleaving Mode Memory Using 1 MB DRAMs

Figures 6-8a, 6-8b and *6-8c* show how pages map into 1, 2 and 4 banks of memory, when page interleaving mode is used with 256 kB DRAMs. In this configuration each page will be 256 words long (i.e., 512 bytes in length). Combinations of 1 MB and 256 kB DRAMs may be used, but all of the DRAMs in each bank must be the same size. For more information regarding what DRAM combinations are supported refer to section 3.1.3.

Page interleaving mode has no programmed wait states, only penalty wait states. A penalty wait state is a CPU wait state which the memory controller must insert to guarantee proper DRAM access times. The memory controller keeps track of the current pages and banks, inserts the appropriate number of wait states for bank and page misses, new RAS generation, and read after write penalties if required.

Three (3) main criteria determine the number of penalty wait states inserted into every memory access in this mode. The three criteria can be classified as follows:

1) Accesses to an inactive page and bank

2) Accesses to an active page (often called a "page-hit")

3) Accesses to an inactive page in an active bank (often called a "page-miss")

6.4.2.1 Accesses to an Inactive Page and Bank

Accesses to an inactive page in an inactive bank begin with all of the memory control signals inactive, indicating that the RAS precharge time is met and no RAS precharge penalty will be paid.

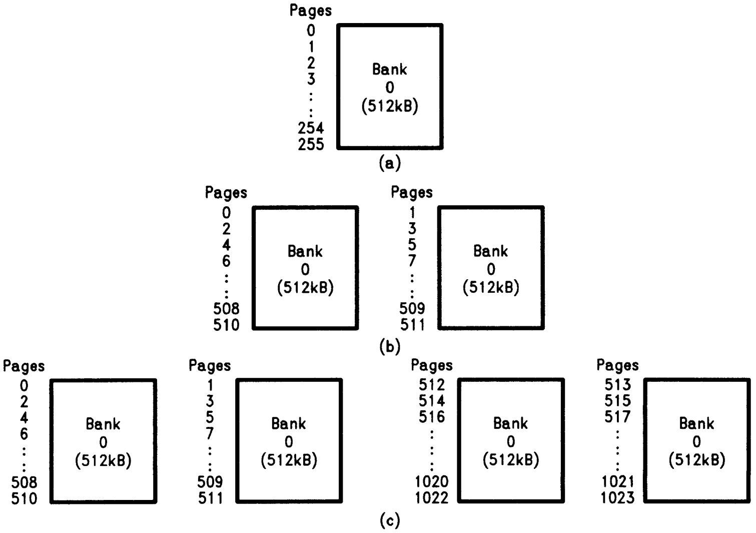
Figure 6-9 shows a write to an inactive page and bank. The row address is multiplexed out onto the memory address lines MA9–0. $\overline{\text{RASx}}$ is driven active low in the middle of T_S, latching the row address into this bank of DRAMs. Then the column address is multiplexed onto MA9–0 and $\overline{\text{CASxy}}$ is asserted low in the middle of T_C, latching the column address into the DRAMs.

During the write cycle MBDIR stays high and $\overline{\text{LMOE}}$ is asserted low in T_C, driving local data onto the memory data bus (i.e., D15–0 to MD15–0). The write enable signal ($\overline{\text{WE}}$) is driven active low in the middle of T_S, indicating an early write cycle. The PC87120 determines the appropriate parity for the low and high bytes and drives PARL and PARH out to the DRAMs. When $\overline{\text{WE}}$ rises half a PROCLK period after the end of T_C, the write cycle ends and both the data and parity are latched into the DRAMs.

$\overline{\text{CASxy}}$ stays active low for half a CPU T-state after the end of T_C to meet the required active low pulse width for the CAS signal. After the write cycle, $\overline{\text{RASx}}$ stays active low to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred. No penalty wait states are added to a write cycle to an inactive page and bank.

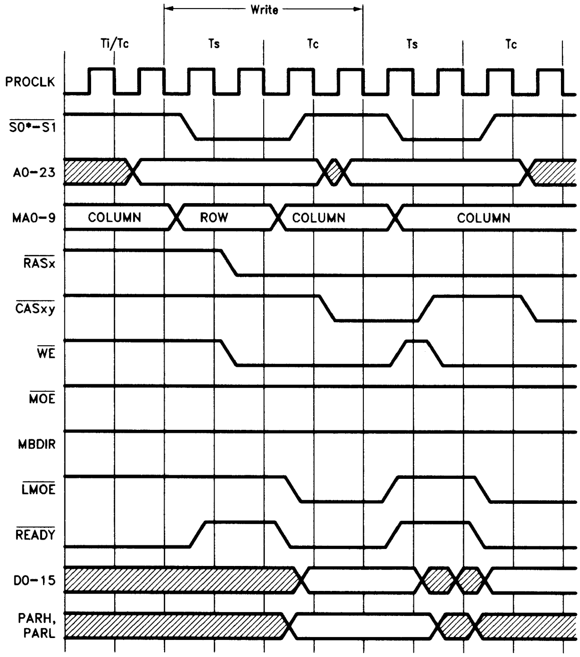
Figure 6-10 shows a read from an inactive page and bank. Like the write cycle, the row address is multiplexed out onto the memory address lines MA9–0. $\overline{\text{RASx}}$ is driven active low in the middle of T_S, latching the row address into this bank of DRAMs. Then the column address is multiplexed onto MA9–0 and $\overline{\text{CASxy}}$ is asserted low in the middle of T_C, latching the column address into the DRAMs.

For the read cycle MBDIR goes low and $\overline{\text{LMOE}}$ is asserted low during the second half of T_C and the first half of T_{CW1}, driving MD15–0 onto D15–0. Write enable ($\overline{\text{WE}}$) stays inactive high through the read cycle, and the DRAMs drive their data onto the memory data bus. The low and/or high bit(s) of parity, PARL and PARH respectively, are driven into the PC87120 and along with the appropriate data byte(s) is used to check parity. If parity is correct no action is taken, but if a parity error is detected a non-maskable interrupt (NMI) is generated, if the NMI circuitry is enabled.



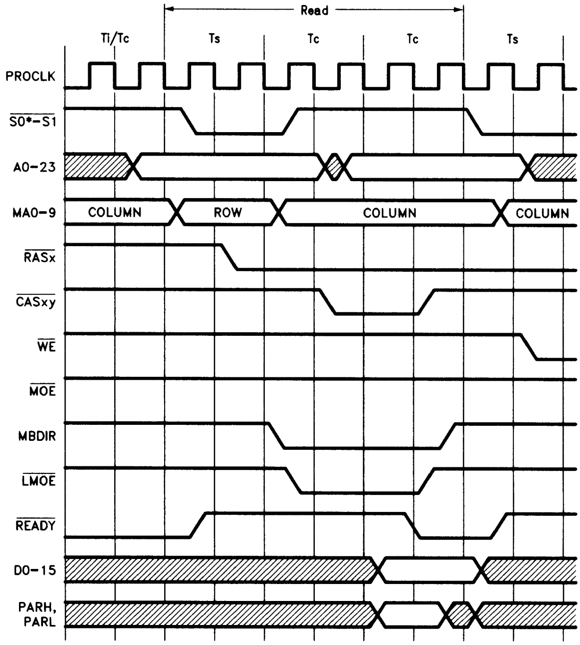
TL/F/11050-39

FIGURE 6-8. Page Interleaving Mode Memory Using 256 kB DRAMs



TL/F/11050–40

FIGURE 6-9. Write to an Inactive Page and Bank in Page Interleaving Mode

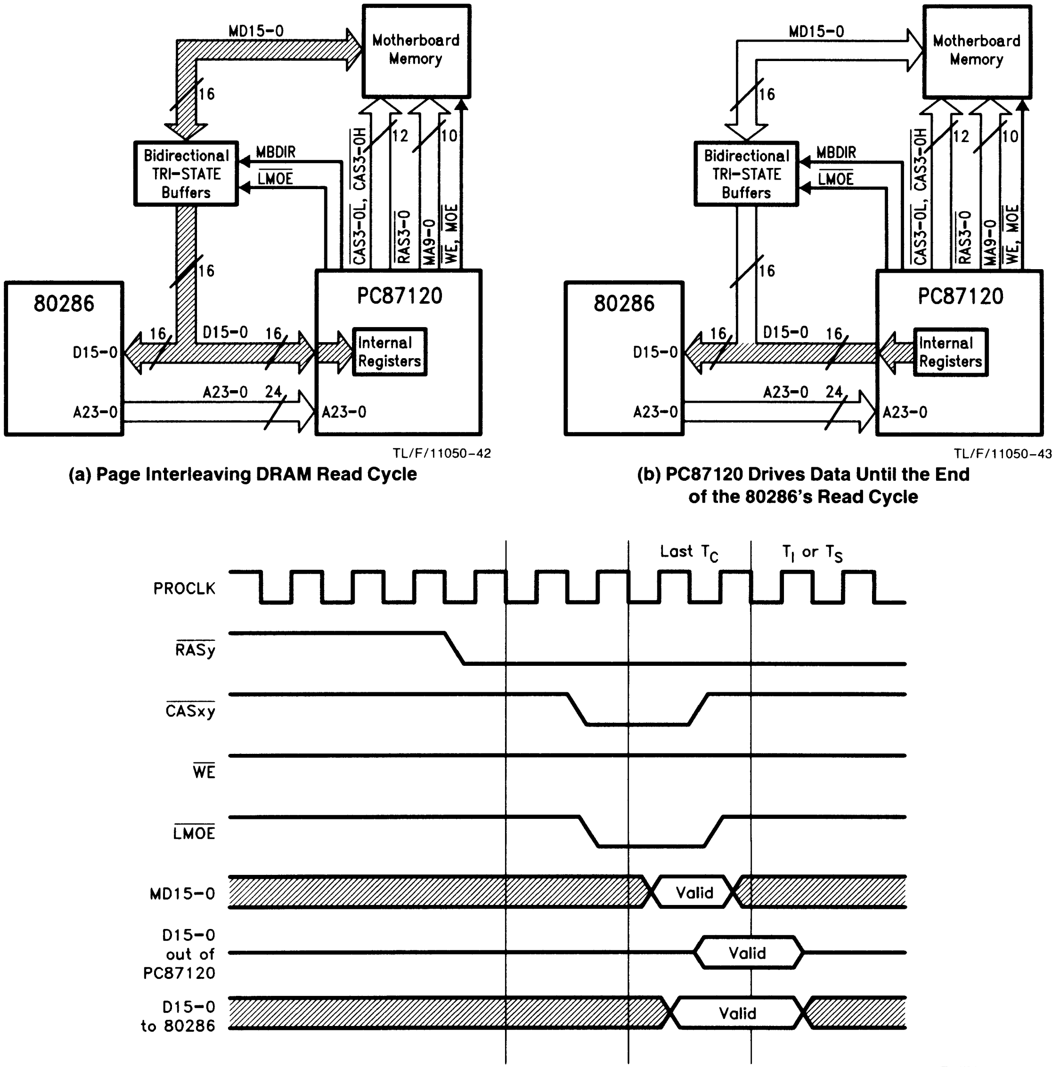


TL/F/11050–41

FIGURE 6-10. Read Cycle to an Inactive Page and Bank in Page Interleaving Mode

As indicated in *Figure 6-10*, one penalty wait state must be added to this read cycle to guarantee sufficient CAS access time. In the middle of the penalty wait state (i.e., T_{CW1}) $\overline{\text{CASxy}}$ rises ending the DRAM access cycle. $\overline{\text{RASx}}$ stays active low after the write cycle to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred.

Figure 6-11 illustrates the operation of a 16-bit memory read in page interleaving mode. At the end of the read cycle, $\overline{\text{CASxy}}$ rises half a CPU T-state before the 80286 finishes its read. To guarantee that the correct data is read by the 80286, the data is latched into two internal PC87120 registers at the end of the memory access cycle (i.e., when $\overline{\text{CASxy}}$ rises), as shown in *Figure 6-11a*. Then this data is immediately driven back out onto the local data bus. Meanwhile $\overline{\text{LMOE}}$ returns inactive high to prevent bus contention. The PC87120 drives the latched data onto the local data bus until the 80286's read cycle is terminated, shown in *Figure 6-11b*. *Figure 6-11c* is the timing diagram for this access. The use of these two internal PC87120 registers, allows the memory controller to reduce the CAS cycle time while still meeting all DRAM access timing requirements.



TL/F/11050-42

(a) Page Interleaving DRAM Read Cycle

TL/F/11050-43

(b) PC87120 Drives Data Until the End of the 80286's Read Cycle

TL/F/11050-44

(c) Page Interleaving Read Cycle

FIGURE 6-11. 16-Bit Page Interleave Read

6.4.2.2 Accesses to an Active Page ("Page-Hit")

When a read or write to a given page is followed by a memory access to the same page, the DRAMs have already latched the row address and only need the new column address to proceed with the access. Since the RAS signal is low from the previous access and does not need to be taken high again, no RAS precharge penalty will be incurred.

Figure 6-12 shows a write to an active page. The row address is latched inside the DRAMs and remains active while RAS stays active low through the entire access. The column address is multiplexed onto MA9–0 and CASxy is asserted low in the middle of T_S, latching the column address into the DRAMs.

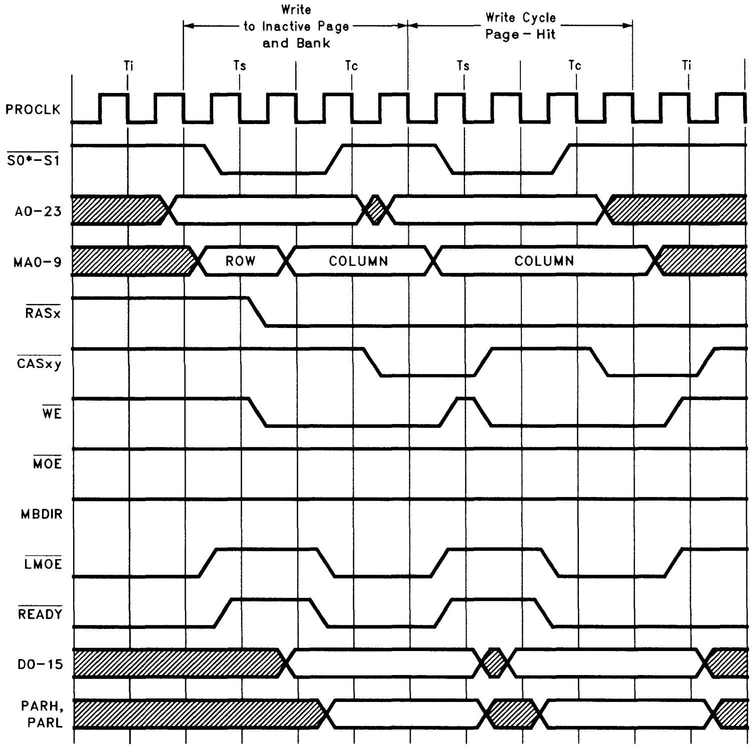
During the write cycle MBDIR stays high and LMOE is asserted low in T_C, driving local data onto the memory data bus (i.e., D15–0 to MD15–0). The write enable signal (WE) is driven active low in the middle of T_S, indicating an early write cycle. The PC87120 determines the appropriate parity for the low and high bytes and drives PARL and PARH out to the DRAMs. When WE rises half a PROCLK period after the end of T_C, the write cycle ends and both the data and parity are latched into the DRAMs.

CASxy stays active low for half a CPU T-state after the end of T_C to meet the required active low pulse width for the CAS signal. After the write cycle, RASx stays active low to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred. No penalty wait states are added to a write cycle to an active page.

Figure 6-13 shows the end of a read cycle followed by a read from the same active page. Like the write cycle, the row address for the second read is latched inside the DRAMs and remains active while RAS stays active low through the entire access. The column address is multiplexed onto MA9–0 and CASxy is asserted low in the middle of T_S, latching the column address into the DRAMs.

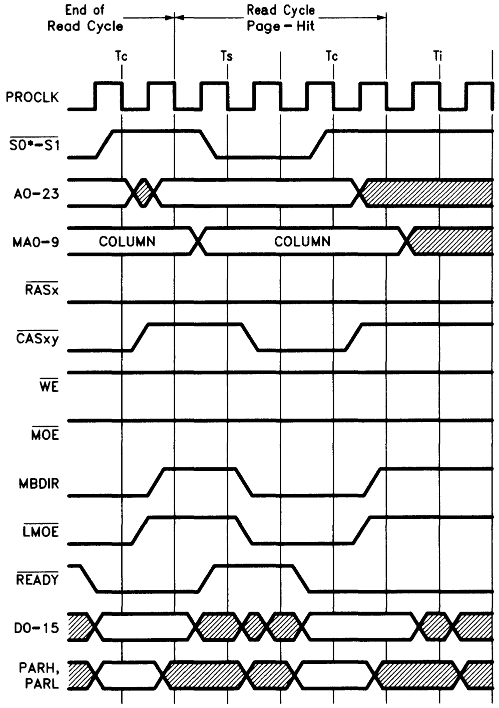
For the read cycle MBDIR goes low and LMOE is asserted low during the second half of T_S and the first half of T_C, driving MD15–0 onto D15–0. Write enable (WE) stays inactive high through the read cycle, resulting in the DRAMs driving their data onto the memory data bus. The low and/or high bit(s) of parity, PARL and PARH respectively, are driven into the PC87120 and along with the appropriate data byte(s) is used to check parity. If parity is correct no action is taken, but if a parity error is detected a non-maskable interrupt (NMI) is generated, if the NMI circuitry is enabled.

As indicated in *Figure 6-13*, no penalty wait state are needed or the second read cycle. In the middle of T_C CASxy rises ending the DRAM access cycle. RASx stays active low after the write cycle to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred.



TL/F/11050–45

FIGURE 6-12. Write Cycle Page-Hit in Page Interleaving Mode



TL/F/11050–46

FIGURE 6-13. Read Cycle Page-Hit in Page Interleaving Mode

Figure 6-11 illustrates the operation for a 16-bit memory read in page interleaving mode. Refer to the last paragraph in section 6.4.2.1 for more information regarding *Figure 6-11.*

Figure 6-14 shows the end of a write cycle followed by a read from the same active page. The row address for the read is latched inside the DRAMs and remains active while RAS stays active low through the entire access. The $\overline{\text{CASxy}}$ signal rises in the middle of T_S completing the write cycle requirements. Then $\overline{\text{CASxy}}$ remains high for two PROCLK periods to satisfy the CAS precharge requirements. Meanwhile, the column address is multiplexed onto MA9–0. In the middle of T_C $\overline{\text{CASxy}}$ is asserted low, latching the column address into the DRAMs.

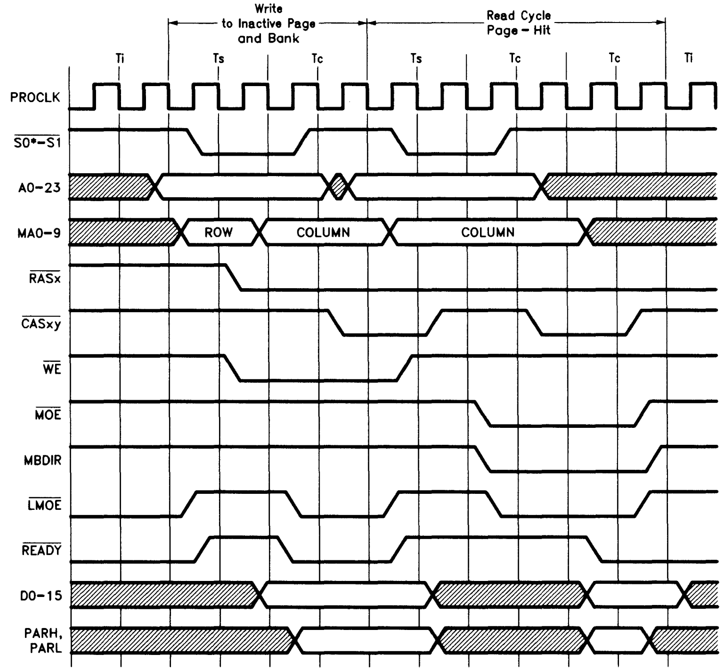
For the read cycle MBDIR goes low and $\overline{\text{LMOE}}$ is asserted low during second half of T_C and the first half of T_{CW1}, driving MD15–0 onto D15–0. Write enable ($\overline{\text{WE}}$) stays inactive high through the read cycle, resulting in the DRAMs driving their data onto the memory data bus. The low and/or high bit(s) of parity, PARL and PARH respectively, are driven into the PC87120 and along with the appropriate data byte(s) is used to check parity. If parity is correct no action is taken, but if a parity error is detected a non-maskable interrupt (NMI) is generated, if the NMI circuitry is enabled.

As indicated in *Figure 6-14*, one penalty wait state is inserted into the read cycle following a write cycle. This penalty wait state provides the time needed to meet the CAS precharge time following a write cycle. In the middle of T_{CW1} $\overline{\text{CASxy}}$ rises ending the DRAM access cycle. $\overline{\text{RASx}}$ stays active low after the write cycle to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred.

6.4.2.3 Accesses to an Inactive Page in an Active Bank ("Page-Miss")

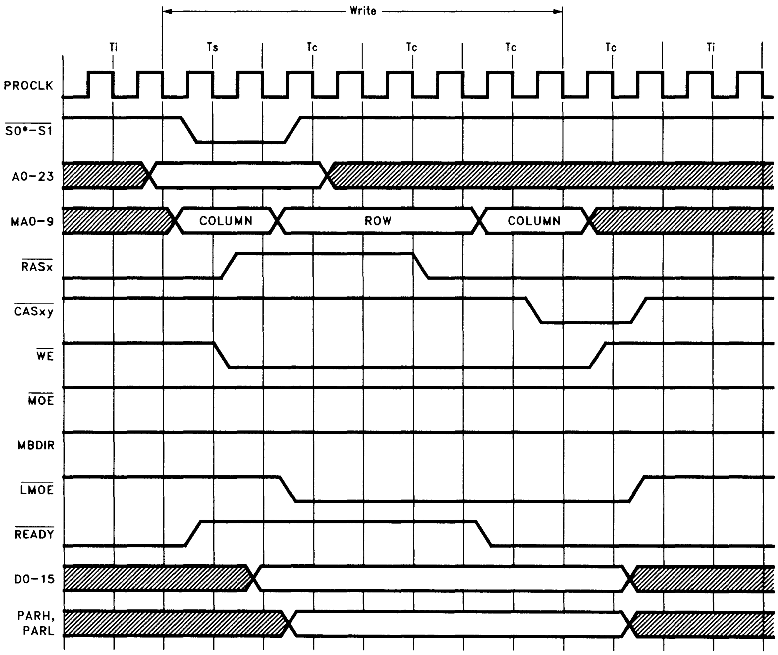
When a read or write to a given page is followed by a memory access to a new page in the same bank, both a new row and column address must be latched into the DRAMs. First, the RAS signal must be returned inactive for the required RAS precharge time. Then RAS may be reasserted low to latch a new row address. This will be followed by a CAS strobe to latch the column address. Meanwhile the memory controller inserts penalty wait states to delay the 80286's access until the DRAM is ready.

Figure 6-15 shows a page-miss write cycle. In the middle of T_S, the memory controller determines that this access is a page-miss and deasserts $\overline{\text{RASx}}$ high. $\overline{\text{RASx}}$ remains high for four (4) PROCLK periods, insuring the proper RAS precharge time.



TL/F/11050-47

FIGURE 6-14. Page Interleaving Read Cycle Page-Hit After Write Cycle



TL/F/11050-48

FIGURE 6-15. Page Interleaving Mode Write Cycle Page-Miss

At the same time, the row address is multiplexed out onto the memory address lines MA9–0. Followed by $\overline{\text{RASx}}$ going active low in the middle of T_{CW1}, latching the row address into this bank of DRAMs. Then the column address is multiplexed onto MA9–0 and $\overline{\text{CASxy}}$ is asserted low in the middle of T_{CW2}, latching the column address into the DRAMs.

During the write cycle MBDIR stays high and $\overline{\text{LMOE}}$ is asserted low in T_C, driving local data onto the memory data bus (i.e., D15–0 to MD15–0). The write enable signal ($\overline{\text{WE}}$) is driven active low in the middle of T_S, indicating an early write cycle. The PC87120 determines the appropriate parity for the low and high bytes and drives PARL and PARH out to the DRAMs. When $\overline{\text{WE}}$ rises half a PROCLK period after the end of T_{CW2}, the write cycle ends and both the data and parity are latched into the DRAMs.

$\overline{\text{CASxy}}$ stays active low for half a CPU T-state after the end of T_C to meet the required active low pulse width for the CAS signal. After the write cycle, $\overline{\text{RASx}}$ stays active low to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred. As shown in *Figure 6-15*, two penalty wait states are added to the page-miss write cycle. The wait states insure the proper RAS precharge and access times.

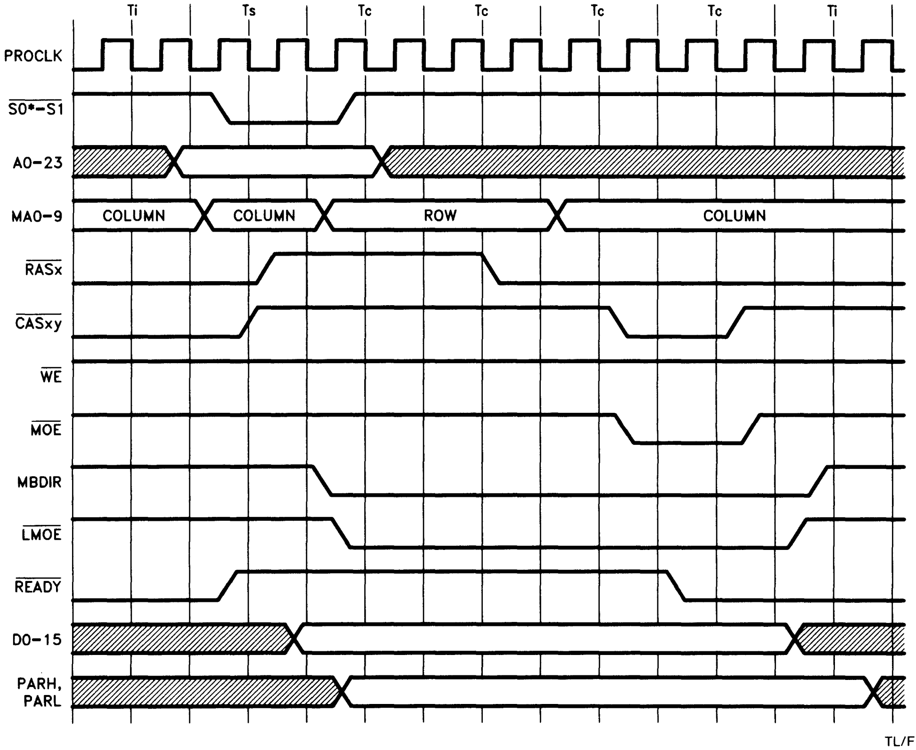
Figure 6-16 shows a page-miss read cycle. In the middle of T_S, the memory controller determines that this access is a page-miss and deasserts $\overline{\text{RASx}}$ high. $\overline{\text{RASx}}$ remains high for four (4) PROCLK periods, insuring the proper RAS precharge time.

At the same time, the row address is multiplexed out onto the memory address lines MA9–0. Followed by $\overline{\text{RASx}}$ going active low in the middle of T_{CW1}, latching the row address into this bank of DRAMs. Then the column address is multiplexed onto MA9–0 and $\overline{\text{CASxy}}$ is asserted low in the middle of T_{CW2}, latching the column address into the DRAMs.

For the read cycle MBDIR goes low and $\overline{\text{LMOE}}$ is asserted low during the second half of T_C through the first half of T_{CW3}, driving MD15–0 onto D15–0. Write enable ($\overline{\text{WE}}$) stays inactive high through the read cycle, resulting in the DRAMs driving their data onto the memory data bus. The low and/or high bit(s) of parity, PARL and PARH respectively, are driven into the PC87120 and along with the appropriate data byte(s) is used to check parity. If parity is correct no action is taken, but if a parity error is detected a non-maskable interrupt (NMI) is generated, if the NMI circuitry is enabled.

In the middle of T_{CW3} $\overline{\text{CASxy}}$ rises ending the DRAM access cycle. $\overline{\text{RASx}}$ stays active low after the write cycle to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred. As indicated in *Figure 6–16*, three penalty wait state are inserted into the page-miss read cycle. These penalty wait states provide the time needed to meet the RAS precharge time and access time for a page-miss read cycle.

Figure 6-11 illustrates the operation of a 16-bit memory read in page interleaving mode. Refer to the last paragraph in section 6.4.2.1 for more information regarding *Figure 6-11*.



TL/F/11050-49

FIGURE 6-16. Page Interleaving Mode Read Cycle Page-Miss

6.4.3 Enhanced Page Interleaving Mode

This mode requires the use of enhanced (also called "fast") page mode DRAM's, which have special latches inside them that allow the column address to flow through as soon as the CAS signal goes active low. This reduces the CAS access time and allows the memory controller to reduce the number of penalty wait states added to read cycles.

The read and write cycles to an inactive page and bank are exactly the same as those in normal page interleaving mode. Refer to the Page Interleaving Mode section for information about these accesses. Also refer to the Page Interleaving Mode section for general operating information pertaining to page interleaving.

The following sections describe the operation of the memory controller during page-hit accesses and page-miss accesses.

6.4.3.1 Accesses to an Active Page ("Page-Hit")

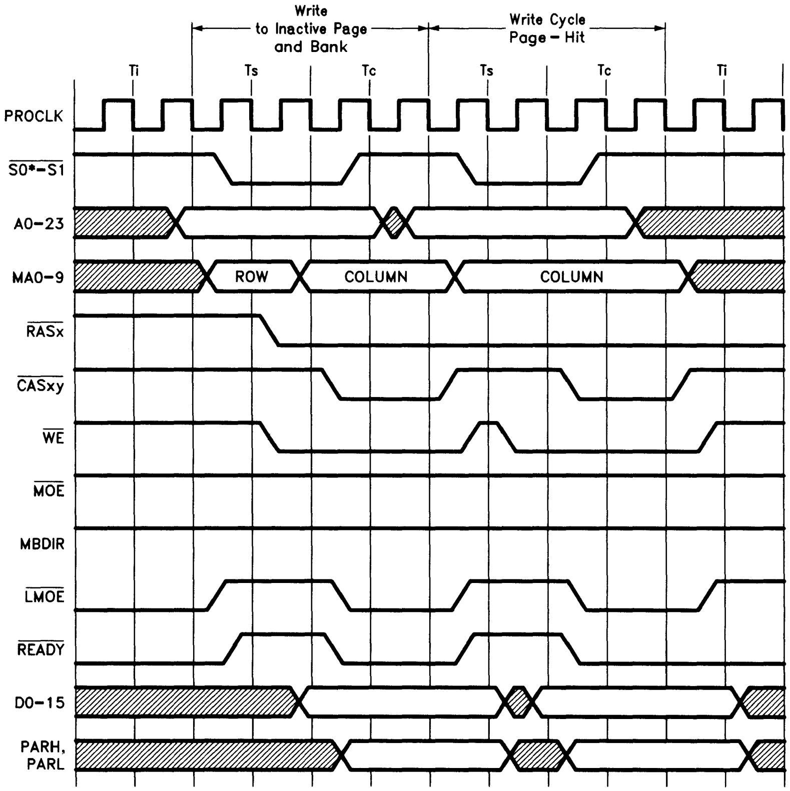
When a read or write to a given page is followed by a memory access to the same page, the DRAMs have already latched the row address and only need the new column address to proceed with the access. Since the RAS signal is low from the previous access and does not need to be taken high again, no RAS precharge penalty will be incurred.

Figure 6-17 shows a write to an active page. The row address is latched inside the DRAMs and remains active while RAS stays active low through the entire access. The column address is multiplexed onto MA9–0 and $\overline{\text{CASxy}}$ is asserted low at the beginning of T_C, latching the column address into the DRAMs.

During the write cycle MBDIR stays high and $\overline{\text{LMOE}}$ is asserted low in T_C driving local data onto the memory data bus (i.e., D15–0 to MD15–0). The write enable signal ($\overline{\text{WE}}$) is driven active low in the middle of T_S, indicating an early write cycle. The PC87120 determines the appropriate parity for the low and high bytes and drives PARL and PARH out to the DRAMs. When CAS rises at the end of T_C, the write cycle ends and both the data and parity are latched into the DRAMs.

After the write cycle, $\overline{\text{RASx}}$ stays active low to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred. No penalty wait states are added to a write cycle to an active page.

Figure 6-18 shows a page-hit read cycle. Like the write cycle, the row address for the read is latched inside the DRAMs and remains active while RAS stays active low through the entire access. The column address is multiplexed onto MA9–0 and $\overline{\text{CASxy}}$ is asserted low in the middle of T_S, latching the column address into the DRAMs.



TL/F/11050–50

FIGURE 6-17. Enhanced Page Interleave Mode Page-Hit Write Cycle

For the read cycle MBDIR goes low and $\overline{\text{LMOE}}$ is asserted low during the second half of T_S and the first half of T_C, driving MD15–0 onto D15–0. Write enable ($\overline{\text{WE}}$) stays inactive high through the read cycle, resulting in the DRAMs driving their data onto the memory data bus. The low and/or high bit(s) of parity, PARL and PARH respectively, are driven into the PC87120 and along with the appropriate data byte(s) is used to check parity. If parity is correct no action is taken, but if a parity error is detected a non-maskable interrupt (NMI) is generated, if the NMI circuitry is enabled.

As indicated in *Figure 6-18*, no penalty wait states are needed for the read cycle. In the middle of T_C $\overline{\text{CASxy}}$ rises ending the DRAM access cycle. $\overline{\text{RASx}}$ stays active low after the write cycle to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred.

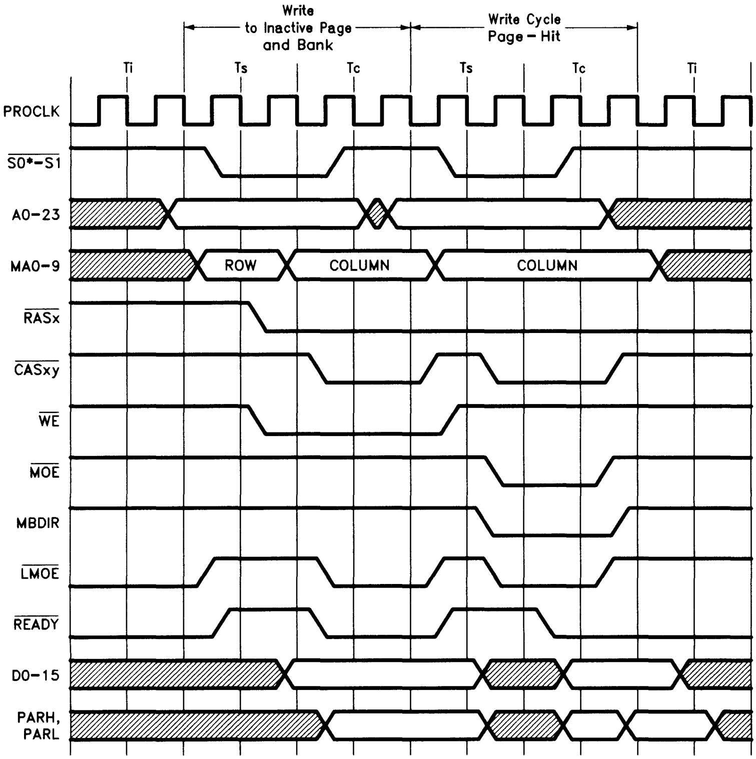
Figure 6-19 illustrates the operation of a 16-bit memory read in enhanced page interleaving mode for a page-hit and an access to an inactive bank and page. At the end of the read cycle, $\overline{\text{CASxy}}$ rises half a CPU T-state before the 80286 finishes its read. To guarantee that the correct data is read by the 80286, the data is latched into two internal PC87120 registers at the end of the memory access cycle (i.e., when $\overline{\text{CASxy}}$ rises), as shown in *Figure 6-19a*. Then this data is immediately driven back out onto the local data bus. Meanwhile $\overline{\text{LMOE}}$ returns inactive high to prevent bus contention. The PC87120 drives the latched data onto the local data bus until the 80286's read cycle is terminated, shown in *Figure 6-19b*. *Figure 6-19c* is the timing diagram for this access. The use of these two internal PC87120 registers allows the memory controller to reduce the CAS cycle time while still meeting all DRAM access timing requirements.

Note that the operation illustrated by *Figures 6-11* and *6-19* are exactly the same.

6.4.3.2 Accesses to an Inactive Page In an Active Bank ("Page-Miss")

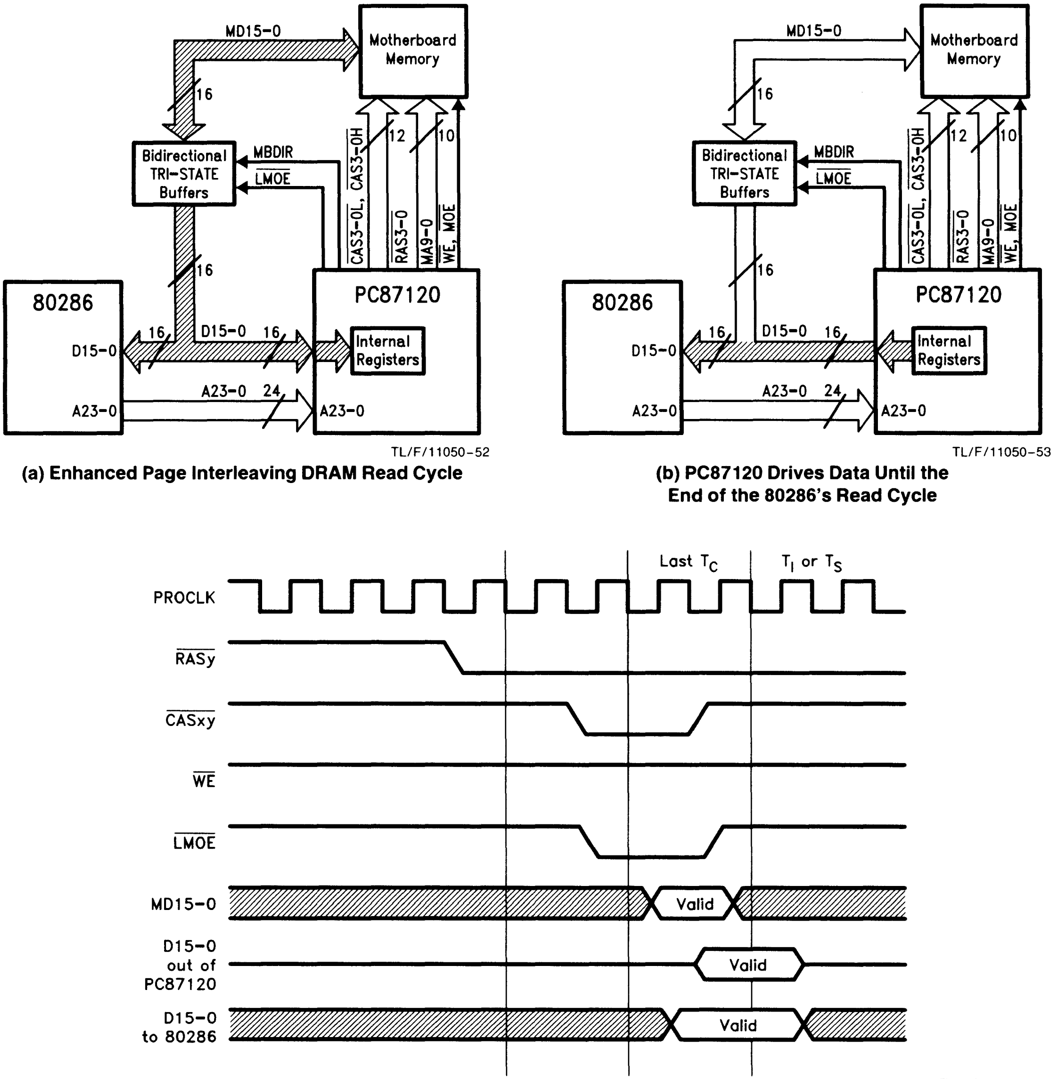
When a read or write to a given page is followed by a memory access to a new page in the same bank, both a new row and column address must be latched into the DRAMs. First, the RAS signal must be returned inactive for the required RAS precharge time. Then RAS may be reasserted low to latch a new row address. This will be followed by a CAS strobe to latch the column address. Meanwhile the memory controller inserts penalty wait states to delay the 80286's access until the DRAM is ready.

Figure 6-20 shows a page-miss write cycle. In the middle of T_S, the memory controller determines that this access is a page-miss and deasserts $\overline{\text{RASx}}$ high. $\overline{\text{RASx}}$ remains high for three (3) PROCLK periods, insuring the proper RAS precharge time.



TL/F/11050-51

FIGURE 6-18. Enhanced Page Interleaving Mode Page-Hit Read Cycle

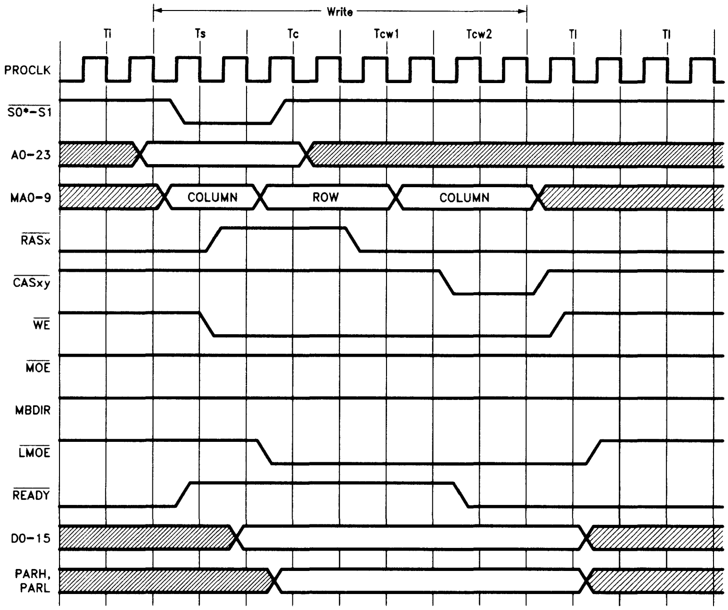


(a) Enhanced Page Interleaving DRAM Read Cycle

(b) PC87120 Drives Data Until the End of the 80286's Read Cycle

(c) Enhanced Page Interleaving Read Cycle

FIGURE 6-19. 16-Bit Enhanced Page Interleave Read



TL/F/11050-55

FIGURE 6-20. Enhanced Page-Interleaving Mode Page-Miss Write Cycle

At the same time, the row address is multiplexed out onto the memory address lines MA9-0. Followed by $\overline{\text{RASx}}$ going active low at the beginning of T_{CW1}, latching the row address into this bank of DRAMs. Then the column address is multiplexed onto MA9-0 and $\overline{\text{CASxy}}$ is asserted low at the beginning of T_{CW2}, latching the column address into the DRAMs.

During the write cycle MBDIR stays high and $\overline{\text{LMOE}}$ is asserted low in T_C, driving local data onto the memory data bus (i.e., D15-0 to MD15-0). The write enable signal ($\overline{\text{WE}}$) is driven active low in the middle of T_S, indicating an early write cycle. The PC87120 determines the appropriate parity for the low and high bytes and drives PARL and PARH out to the DRAMs. When $\overline{\text{CASxy}}$ rises at the end of T_{CW2}, the write cycle ends and both the data and parity are latched into the DRAMs.

After the write cycle, $\overline{\text{RASx}}$ stays active low to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred. As shown in *Figure 6-20*, two penalty wait states are added to this page-miss write cycle. The wait states insure the proper RAS precharge and access times.

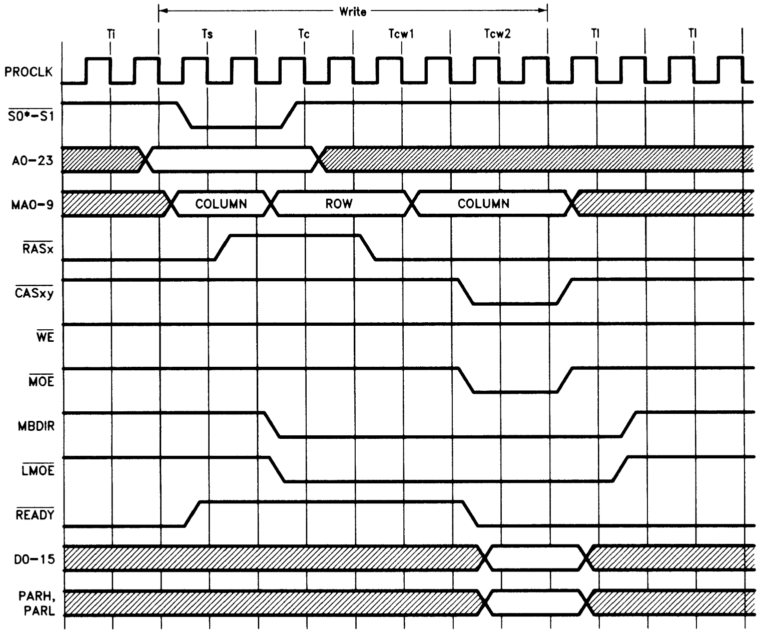
Figure 6-21 shows a page-miss read cycle. In the middle of T_S, the memory controller determines that this access is a page-miss and deasserts $\overline{\text{RASx}}$ high. $\overline{\text{RASx}}$ remains high for three (3) PROCLK periods, insuring the proper RAS precharge time. At the same time, the row address is multiplexed out onto the memory address lines MA9-0. Followed by $\overline{\text{RASx}}$ going active low at the beginning of T_{CW1}, latching the row address into this bank of DRAMs. Then the column address is multiplexed onto MA9-0 and $\overline{\text{CASxy}}$ is asserted low at the beginning of T_{CW2}, latching the column address into the DRAMs.

For the read cycle MBDIR goes low and $\overline{\text{LMOE}}$ is asserted low during T_{CW1} and T_{CW2}, driving MD15-0 onto D15-0. Write enable ($\overline{\text{WE}}$) stays inactive high through the read cycle, resulting in the DRAMs driving their data onto the memory data bus. The low and/or high bit(s) of parity, PARL and PARH respectively, are driven into the PC87120 and along with the appropriate data byte(s) is used to check parity. If parity is correct no action is taken, but if a parity error is detected a non-maskable interrupt (NMI) is generated, if the NMI circuitry is enabled.

At the end of T_{CW2} $\overline{\text{CASxy}}$ rises ending the DRAM read cycle. $\overline{\text{RASx}}$ stays active low after the read cycle to preserve the row address latched inside this bank of DRAMs. If the next access is to the same page a new RAS is not needed and no RAS precharge penalty will be incurred. As indicated in *Figure 6-21*, two penalty wait states are inserted into the page-miss read cycle. These penalty wait states provide the time needed to meet the RAS precharge time and access time for a page-miss read cycle.

As the last two sections have demonstrated that the timing specifications for write cycles in page interleaving mode are different from those in enhanced page interleaving mode, but the number of penalty wait states are the same. On the other hand, enhanced page interleaving mode inserts one less penalty wait state into every page-hit read cycle immediately following a write cycle and one less penalty wait state into every page-miss read cycle. This provides a performance advantage over normal page interleaving mode.

Tables 6-4 and 6-5 list the number of wait states added to memory accesses in the three access modes.



TL/F/11050-56

FIGURE 6-21. Enhanced Page-Interleaving Mode Page-Miss Read Cycle

TABLE 6-4. Programmed Wait States for Conventional Mode

	RAM Wait State bit	
	= 0	= 1
Read Cycle Conventional Mode	1	0
Write Cycle Conventional Mode	1	0

TABLE 6-5. Penalty Wait States for Page Interleaving Modes

	Inactive Page and Bank	Page-Hit After Read or Inactive T-state	Page-Hit Immediately After Write Cycle	Page-Miss
Read Cycle Page Interleaving Mode	1	0	1	3
Write Cycle Page Interleaving Mode	0	0	0	2
Read Cycle Enhanced Page Interleaving Mode	1	0	0	2
Write Cycle Enhanced Page Interleaving Mode	0	0	0	2

6.5 EMS MEMORY SUPPORT

All EMS memory accesses to the address range C0000h to EFFFFh will incur an additional wait state. *Figure 6-22* shows an EMS Read Cycle followed by a non-EMS Write cycle.

6.6 MEMORY REFRESH

The PC87120's memory controller contains a programmable RAS only refresh state machine. This refresh state machine guarantees that all of the DRAM is serviced within a programmed amount of time. The refresh state machine also provides a burst mode, which refreshes multiple rows ("pages") in each bank during every refresh cycle.

To begin a refresh cycle, the PC87120 must perform a Hold ReQuest (HRQ)/HoLD Acknowledge (HLDA) sequence with the 80286, as shown in *Figure 6-22.* All RAS signals are driven high when the hold refresh HLDA is received. After that, the PC87120 drives the refresh signal, $\overline{\text{REF}}$, active low and drives the appropriate address onto the memory address, MA9–0. Then the PC87120 staggers its RAS strobes to reduce the power supply noise during refresh cycles: $\overline{\text{RAS0}}$ and $\overline{\text{RAS2}}$ go active low, followed by $\overline{\text{RAS1}}$ and $\overline{\text{RAS3}}$ going active low a half T-state later. If only one memory bank is supported, only $\overline{\text{RAS0}}$ will go active. Likewise, only $\overline{\text{RAS0}}$ and $\overline{\text{RAS1}}$ will go active if two memory banks are configured.

The RAS signals stay low for 3.5 T-states, then return high ending the row refresh. The refresh state machine increments the refresh row address when $\overline{\text{RAS0}}$ and $\overline{\text{RAS2}}$ rise. The PC87120 saves this new refresh row address and uses it for the next refresh cycle. A CPU T-state after $\overline{\text{RAS1}}$ and $\overline{\text{RAS3}}$ rise, the PC87120 deasserts HRQ low and $\overline{\text{REF}}$ high, ending the refresh cycle.

The refresh state machine also provides a programmable burst refresh mode. Table 6-6 shows how the Burst Refresh bits (Bits 3-2 of the Refresh Control Register, located at IO address FC89h) determine the number of DRAM rows refreshed during each cycle.

TABLE 6-6. Number of Rows Refreshed during Each Refresh Cycle

Bit 1	Bit 0	Number of Refreshes per Cycle
0	0	1 (default)
0	1	2
1	0	2
1	1	4

Figure 6-23 shows a burst refresh cycle with two row refreshes per cycle. The first refresh cycle operates as described above. At the completion of the first row refresh, MA9–0 is incremented to the next row address. All RAS signals return high for two (2) CPU T-states providing the RAS precharge time. Then $\overline{\text{RAS0}}$ and $\overline{\text{RAS2}}$ go active low, followed by $\overline{\text{RAS1}}$ and $\overline{\text{RAS3}}$ going active low a half T-state later.

The RAS signals stay low for 3.5 T-states, then return high ending the second row refresh. The refresh state machine increments the refresh row address when $\overline{\text{RAS0}}$ and $\overline{\text{RAS2}}$ rise. The PC87120 saves this new refresh row address and uses it for the next refresh cycle. A CPU T-state after $\overline{\text{RAS1}}$ and $\overline{\text{RAS3}}$ rise, the PC87120 deasserts HRQ low, ending the refresh cycle.

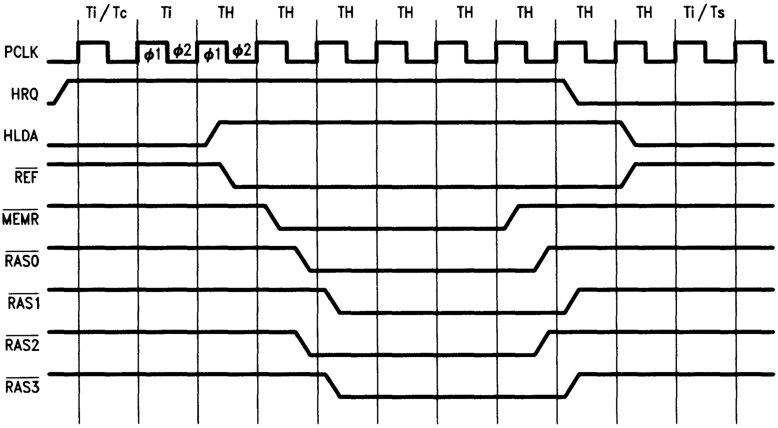
Burst refresh mode reduces the average CPU hold latency at the beginning and end of the refresh cycle. This may increase system performance.

The Refresh Cycle Time bits (Bits 1–0 of the Refresh Control Register, located at IO address FC89h) determine the refresh rate for the motherboard DRAM. Table 6-7 shows the relationship between these bits and the refresh rate.

If slow refresh rate DRAMs are used, fewer refresh cycles are needed and system performance will improve.

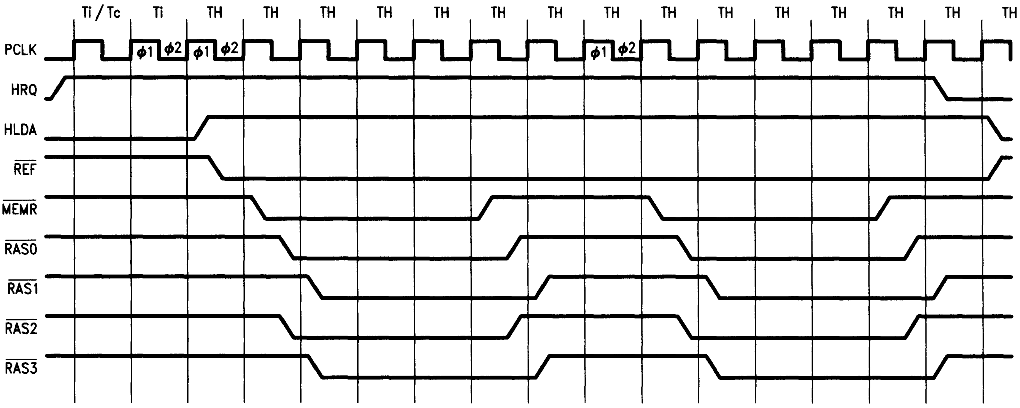
TABLE 6-7. Refresh Rate

Refresh Control Register		Mode (Refresh Rate)
Bit 1	Bit 0	
0	0	0 (4 ms) (default)
0	1	1 (8 ms)
1	0	2 (16 ms)
1	1	3 (32 ms)



TL/F/11050-57

FIGURE 6-22. Staggered Refresh



TL/F/11050-58

FIGURE 6-23. Burst Mode Refresh

7.0 Port B and NMI Enable Logic

Port B is a hardware register located within the PC87120 at IO address 0061h. Port B contains two control bits for non-maskable interrupt (NMI) generation, two control bits for the speaker operation and four read only system status bits. The following describes the function of each bit in Port B:

Port B
(0061h)

(Default = 00000000)

Bit 7—Parity Check Flag (Read only bit)
(Default = 0)

This bit will be set high when the PC87120's parity checking circuitry detects a parity error in the motherboard memory. To clear this bit, bit 2 of this register (Enable RAM Parity Check) must be set low. If no motherboard parity errors are detected, this bit will remain low.

Bit 6—IO Channel Check Flag (Read only bit)
(Default = 0)

This bit is set high when an expansion board asserts the active low expansion bus signal IO channel check (IOCHCK). To clear this bit, bit 3 of this register (Enable IO Channel Check) must be set low. If no expansion board asserts IOCHCK, this bit will remain low.

Bit 5—Timer 2 Output (Read only bit)
(Default = 0)

This bit follows the value of the output of the programmable Timer 2. Timer 2 is used to generate the frequency for the speaker output (SPKR).

Bit 4—Refresh Reference (Read only bit)
(Default = 0)

This bit toggles every time a refresh cycle occurs, providing a way to detect refreshes being performed.

Bit 3—Enable IO Channel Check (Read/Write bit)
(Default = 0)

When set to 0, this bit allows an expansion board to generate a NMI by driving IOCHCK active low. Setting this bit to 1 clears any NMI generated by IOCHCK. Also, no IOCHCK NMI can be generated while this bit is set to 1.

Bit 2—Enable Parity Checking (Read/Write bit)
(Default = 0)

When set to 0, this bit allows the PC87120's parity checking circuitry to generate a NMI when a motherboard DRAM parity error occurs. Setting this bit to 1 clears any NMI generated by the parity checker. Also, no parity error NMI can be generated while this bit is set to 1.

Bit 1—Speaker Data (Read/Write bit)
(Default = 0)

When this bit and bit 0 are set to 1, the speaker output (SPKR) will oscillate at the frequency programmed in Timer 2. When set to 0, the SPKR output will stay low and will not oscillate.

Bit 0—Timer 2 Gate Speaker (Read/Write bit)
(Default = 0)

When this bit is set to 1, it enables Timer 2 to oscillate at the timer's programmed frequency. When set low, Timer 2 is disabled. While Timer 2 is disabled, the SPKR output will stay low and will not oscillate.

Non-Maskable Interrupt (NMI) Enable Logic

An IO write to location 0070h will result in data bit 7 being written into a register which enables or disables all NMIs. When this register is set to 0 NMIs are enabled. Otherwise, all NMIs are disabled when this register is set high. Disabling NMIs via this bit does not clear or disable the setting of the Parity Check Flag (Bit 7, Port B) or IO Channel Check Flag (Bit 6, Port B). The default reset state of this bit is 1, disabling NMIs.

8.0 ROM BIOS Control Circuitry

The PC87120 supports several ROM BIOS options. These options include the choice of ROM size, the choice of 8-bit or 16-bit data bus widths of ROM, the choice of locating the ROM on the peripheral bus or the memory bus, and the choice of RAM shadowing. In addition, the number of wait states inserted during accesses may be selected if the ROM is located on the memory bus. The PC87120 also provides a fully decoded ROM read (ROMRD) strobe, which may drive the ROM's output enable pin without further decoding. Bits 7–6 of the Shadow RAM Enable Register determine the size of EPROM supported. These two bits along with the 8/16-bit ROM BIOS Select bit determine the memory address range where the ROM BIOS resides:

ROM Type		8/16-Bit ROM BIOS Select Bit	Address Range where the ROM BIOS Resides
Bit 7	Bit 6		
0	0	0 (16-bit)	F8000h-FFFFFh (32 kB)
0	1	0 (16-bit)	F0000h-FFFFFh (64 kB)
1	1	0 (16-bit)	E0000h-FFFFFh (128k)
0	0	1 (8-bit)	FC000h-FFFFFh (16 kB)
0	1	1 (8-bit)	F8000h-F7FFFh (32 kB)
1	1	1 (8-bit)	F0000h-FFFFFh (64 kB)

To enable the ROM BIOS mapped into the memory address range from 0F0000h to 0FFFFFh, bit 1 of the ROM Area Enable/Shadow RAM Mode Register must be set to 1. Bit 0 of the same register enables the ROM BIOS mapped into the memory address range 0E0000h to 0EFFFFh. When 64 kB or less of ROM BIOS is supported, bit 1 must be active high, but bit 0 should be set to zero (inactive). Both bits 1 and 0 must be set active high if 64 kB to 128 kB of ROM BIOS is supported. Bits 1 and 0 should both be low if the shadow RAM BIOS is in use.

The 16 bit ROM BIOS option (RAM/ROM Configuration Register, Bit 6 = 0) enables the transfer of 16 bits of information during each BIOS access. The use of 8-bit ROM (RAM/ROM Configuration Register, Bit 6 = 1) can reduce the number of ROMs needed, but slows the system down by requiring twice as many accesses for the same amount of information.

8.1 ROM ON THE MEMORY BUS

The 16-bit ROM option must be selected if the ROM BIOS is placed on the memory bus because there is no way to transfer the ROM BIOS's odd bytes from MD7–0 to the local bus D15–8 lines. With the ROM BIOS on the memory bus, ROM data outputs drive the MD15–0 bus and are buffered onto the local bus (D15–0) via two bidirectional buffers controlled by LMOE and MBDIR.

8.0 ROM BIOS Control Circuitry

(Continued)

To meet the systems cost/performance requirements, the designer may select 0, 1, 2 or 3 wait states for the ROM BIOS. The designer should also note that EPROMs generally have long TRI-STATE disable times, which may result in bus contention on the MD15–0 lines if a ROM read is followed by a zero wait state DRAM read cycle.

8.2 ROM ON THE PERIPHERAL BUS

When the ROM BIOS is located on the peripheral bus (X bus), the ROM's low byte outputs should drive the XD7–0 bus. For the 16-bit ROM option, the ROM's high byte outputs drive the SD15–8 lines. Designers usually choose to place the ROM's low byte on the XD7–0 bus instead of the SD7–0 lines because of load concerns. Typically the loading on SD15–8 is less than SD7–0, so the ROM's high byte outputs can drive SD15–8.

If the 16-bit ROM option is selected the number of 16-bit AT programmed wait states are inserted. For 8-bit ROM the number of 8-bit AT programmed wait states are applied. The PC87120 also controls all 16-bit reads from the 8-bit ROM and guarantees proper operation.

The slower accesses on the peripheral bus allow the use of slow, inexpensive EPROM, but they also reduce the system performance. To use the least expensive components while obtaining the maximum system performance, the ROM BIOS may be placed on the peripheral bus and loaded into RAM after power-up. Once in RAM, the ROM's can be disabled and all BIOS accesses may take place at the maximum speed of the DRAM. The RAM used for this purpose is commonly referred to as shadow RAM.

Refer to section 6.2 for more information regarding the operation and configuration or shadow RAM.

9.0 Math Coprocessor Interface

The PC87120 supports a 286-AT compatible CPU/coprocessor interface. The PC87120 provides a coprocessor (80287) reset (RES287), a numeric processor chip select ($\overline{\text{NPCS}}$), and the logic to wait a CPU (80286) access to the 80287 while the coprocessor is busy. In addition, the PC87120 provides the circuitry to generate an interrupt to the 80286 when the 80287 enters error status.

The 80287's reset strobe, RES287, will be active during power-on reset. RES287 will also become active for sixteen processor clock (PROCLK) periods following any write to IO address 00F1h. For more details on the operation of RES287 refer to section 4.0 RESET and SHUTDOWN LOGIC.

In a 286-AT compatible computer, the 80287 is mapped into the IO address range 00F8h to 00FFh. Whenever an access is made to this address range the active low numeric processor chip select output ($\overline{\text{NPCS}}$) of the PC87120 is asserted. To guarantee proper access timing requirements at the faster clock speeds, a CPU wait state is inserted into every access to the 80287.

When the coprocessor is busy (i.e., $\overline{\text{BUSY}}$ = 0), the PC87120 drives the CPU's busy input active low with the $\overline{\text{BUSY286}}$ output. This will wait any 80286 accesses to the 80287 until the coprocessor is no longer busy (i.e., both $\overline{\text{BUSY}}$ and $\overline{\text{BUSY286}}$ returns inactive high).

If the coprocessor incurs an error condition (i.e., the 80287's active low $\overline{\text{ERROR}}$ output is asserted), the PC87120 will generate an interrupt to the CPU. In addition, the PC87120 will drive $\overline{\text{BUSY286}}$ active low to prevent all accesses to the 80287 while the coprocessor has an error status. $\overline{\text{BUSY286}}$ will stay active low until the interrupt service routine performs a write to IO location 00F0h. Then $\overline{\text{BUSY286}}$ will return inactive high. This hand shaking technique guarantees that the 80286 will be unable to access the 80287 before the $\overline{\text{ERROR}}$ interrupt is serviced. Once $\overline{\text{BUSY287}}$ deasserted high, the interrupt service routine must clear the 80287's error condition.

10.0 Peripheral Functions

10.1 DIRECT MEMORY ACCESS (DMA) DESCRIPTION

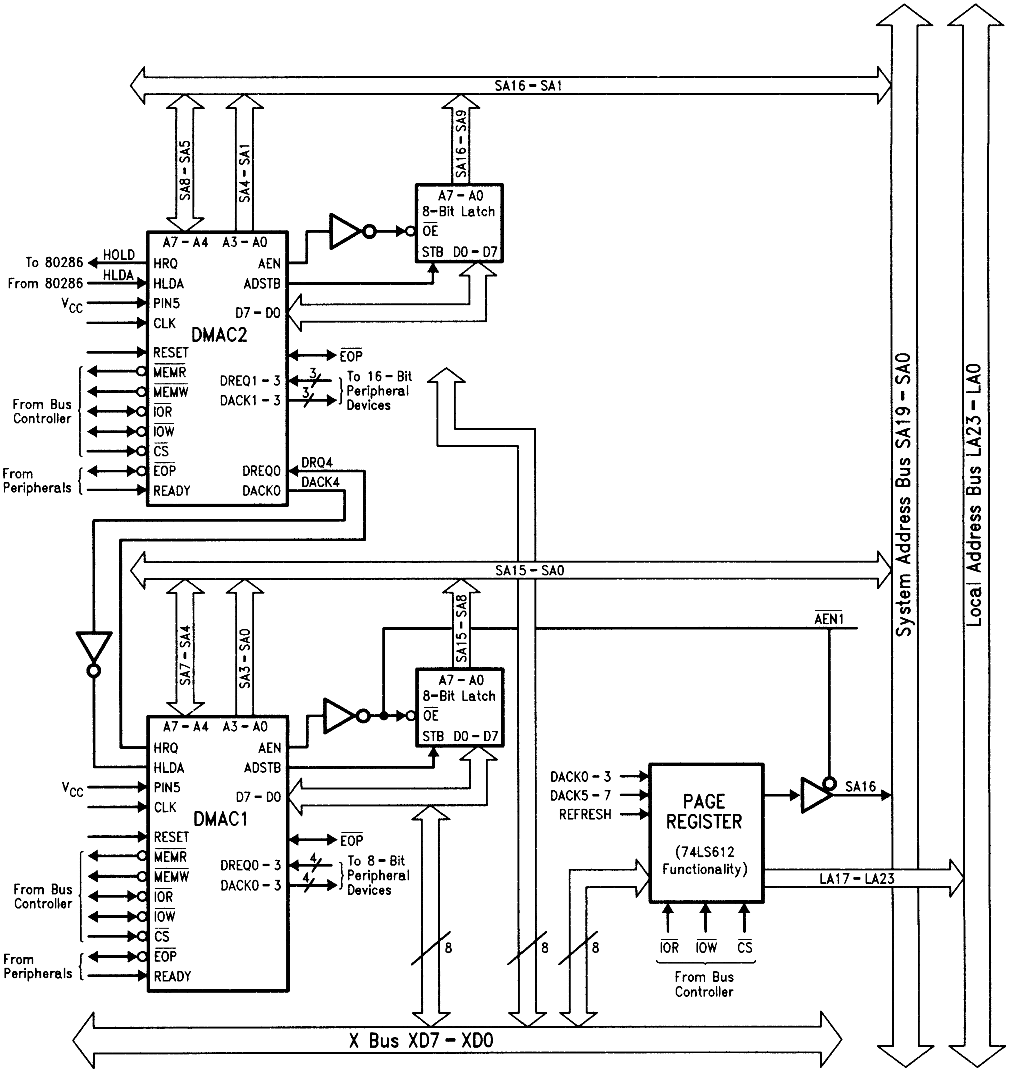
The PC87120 supports direct exchanges of information between peripherals and memory through an internal Direct Memory Access (DMA) system.

10.1.1 DMA Architectural Description

The PC87120 provides a PC/AT compatible DMA system. Therefore, in the PC87120, seven DMA channels are provided through two cascaded DMA controllers. Each DMA controller in the PC87120 is compatible with the Intel 8237A-5 Programmable DMA Controller. For clarity, the DMA channels in the PC87120 will be identified as Channels 0 through 7, denoted as DREQ0–7 and DACK0–7 in *Figure 10-1*. The four DMA channels on the slave DMA controller, or DMA Controller 1 (DMAC1) correspond to channels 0 through 3. These channels are used for 8-bit data transfers between peripheral devices and memory. The four DMA channels on the master DMA controller, or DMA Controller 2 (DMAC2) correspond to channels 4 through 7 (DREQ4–7 and DACK4–7). Three of these channels, channel 5–7, are available for 16-bit data transfers between peripheral devices and memory. Channel 4 is used for cascading the slave DMA controller making it unavailable for external use. *Figure 10-1* illustrates the DMA system implemented in the PC87120.

An 80286 PC/AT system has a maximum of 16 megabyte of system memory. The DMA system in the PC87120 can address the entire range of system memory. The PC87120 DMA addressing circuitry is internally implemented with two 8-bit latches and a 74LS612 Memory Mapper (referred to as the Page Register) as shown in *Figure 10-1*. See Section 10.1.2.3, DMA Address Generation, for a complete description of how the DMA system addresses memory.

Many of the signals in the DMA system are internal to the PC87120. Table 10-1 provides the system designer with a list of the PC87120 pins which are associated with the DMA system.



TL/F/11050-59

FIGURE 10-1. PC87120 DMA System

TABLE 10-1. PC87120 DMA System Pins

DMA System Signal	Corresponding Pin Name	Pin Number	Purpose
DREQ0	DRQ0	39	DMA Request Channel 0
DREQ1	DRQ1	38	DMA Request Channel 1
DREQ2	DRQ2	37	DMA Request Channel 2
DREQ3	DRQ3	36	DMA Request Channel 3
DREQ5	DRQ5	35	DMA Request Channel 5
DREQ6	DRQ6	34	DMA Request Channel 6
DREQ7	DRQ7	33	DMA Request Channel 7
DACK0	$\overline{\text{DACK0}}$	32	DMA Acknowledge Channel 0
DACK1	$\overline{\text{DACK1}}$	31	DMA Acknowledge Channel 1
DACK2	$\overline{\text{DACK2}}$	30	DMA Acknowledge Channel 2
DACK3	$\overline{\text{DACK3}}$	29	DMA Acknowledge Channel 3
DACK5	$\overline{\text{DACK5}}$	28	DMA Acknowledge Channel 5
DACK6	$\overline{\text{DACK6}}$	27	DMA Acknowledge Channel 6
DACK7	$\overline{\text{DACK7}}$	26	DMA Acknowledge Channel 7
HRQ	CPUHRQ	116	CPU Hold Request
HLDA	CPUHLDA	117	CPU Hold Acknowledge

10.1.2 DMA Functional Description

A peripheral device requests DMA service by asserting its preassigned DMA Request (DREQ) channel. When the associated DMA Controller senses that DREQ has become active, it generates a Hold Request (HRQ). In the PC87120, DMAC2 controls the HRQ signal to the 80286. Therefore, asserting DREQ4 through DREQ7 directly generates a HRQ to the 80286. On the other hand, the HRQ from DMAC1 does not connect to the 80286's Hold Request, but to DREQ4 of DMAC2. Therefore, a DMA Request on channels 0 through 3 will be serviced through channel 4 on DMAC2. A HRQ generated by DMAC1 asserts DREQ4 on DMAC2, which then generates the HRQ to the 80286. After the 80286 receives a HRQ, it will relinquish control of the system buses and respond with a Hold Acknowledge (HLDA) signal to the DMAC2. The DMAC2 then signals the peripheral on one of its channels that a DMA transfer is about to begin by asserting the appropriate DMA Acknowledge (DACK4 through DACK7) signal. The HLDA of DMAC1 is controlled by DACK4. Therefore, when DACK4 becomes active, HLDA of DMAC1 is forced active. DMAC1 now signals the appropriate peripheral on one of its channels that the DMA transfer is about to begin by asserting the appropriate DMA Acknowledge (DACK0 through DACK3) signal.

10.1.2.1 Clock

In every CPU/AT Bus mode except Turbo Bus mode, the DMA Controllers in the PC87120 will operate at half the frequency of the system clock (SYSCLK). In Turbo Bus mode, the DMA Controllers will run at one fourth the frequency of SYSCLK. Table 10-2 shows the relationship between CPU/AT Bus, SYSCLK, and the operating frequency of the DMA Controllers.

TABLE 10-2. DMA Clock and CPU Bus Mode Relationship

CPU/AT Bus Mode	SYSCLK	DMA CLK
Non-Turbo	8 MHz	4 MHz
Turbo	16 MHz	4 MHz
Normal	8 MHz	4 MHz
Asynchronous	8 MHz	4 MHz

10.1.2.2 DMA Channels

In a PC/AT compatible system, seven of the eight DMA channels are available to peripheral devices. Channel two is typically assigned to the diskette drive. All other channels are typically available for other peripherals to use.

The priority in which the channels are serviced is programmable to be either fixed or rotating priority. When a DMA Controller is programmed to service the DMA Requests using a fixed priority, channel 0 in that DMA Controller has the highest priority followed by channels 1, 2, and 3 respectively. Therefore, in the PC87120, if DMAC2 is programmed for fixed priority, channels 0 through 3 will have higher priority than channels 5 through 7 since they are serviced through channel 4. Table 10-3 shows the PC87120 channels and their corresponding priority when both DMAC1 and DMAC2 are programmed to service DMA Requests in a fixed priority.

When a DMA Controller is programmed to service the DMA Requests using a rotating priority, the DMA Channel which most recently received a DMA service is given the lowest priority. This allows equal access to DMA service for all channels. Therefore, in the PC87120, if DMAC2 is programmed for rotating priority, channels 0 through 3 will be allowed one DMA service when channel 4 has rotated to the highest priority. After one DMA service, channel 4 (meaning channels 0 through 3) rotate to the lowest priority. The further breakdown in channels 0 through 3 priority is determined according to how DMAC1 is programmed. Thus, if DMAC2 is programmed for rotating priority and DMAC1 is programmed for fixed priority, channel 0 will have the highest priority followed by channels 1 through 3 ONLY when channel 4 has rotated through the priority chain. As an example, refer to Table 10-4 which shows how the channels priority is determined when DMAC2 is programmed for rotating priority and DMAC1 is programmed for fixed priority.

TABLE 10-3. PC87120 Channel Priorities when Both DMAC1 and DMAC2 are Programmed for Fixed Priority

Priority (Highest = 1, Lowest = 7)	PC87120 DMA Channel
1	Channel 0
2	Channel 1
3	Channel 2
4	Channel 3
5	Channel 5
6	Channel 6
7	Channel 7

10.1.2.3 DMA Address Generation

The PC87120 supports 8-bit data and 16-bit data DMA transfers. Both types of DMA transfers can transfer data throughout the entire range of system memory (16 megabyte maximum). However, each type of DMA transfer generates the memory address for the transfer differently resulting in different block sizes for which the transfer can occur. Specifically, 8-bit data transfers can occur in 64 kB blocks on both even and odd byte boundaries whereas 16-bit data transfers can occur in 128 kB blocks on even bytes only.

Figure 10-1 illustrates the DMA memory address generation circuitry implemented in the PC87120. This circuitry includes two 8-bit latches as well as an 74LS612 Memory Mapper, which is referred to as the Page Register. Each DMA controller in the PC87120 has its own 8-bit latch providing for 16 bits of memory address generation. The Page Register extends the DMA memory addressing capability by 8 bits to a total of 24 bits. The Page Register contains 16 internal registers of which eight are used. Each of the seven available DMA channels and Refresh have their own register in the Page Register.

Prior to a DMA transfer the entire memory address must be programmed. This is done in two steps. First, the 16-bit address to be generated by the DMA controller and its associated 8-bit latch is programmed into the Base Address Register of the DMA Controller. This 16-bit memory address must be programmed in the Mode register to increment or decrement following each DMA transfer. Second, the 8-bit address to be supplied by the Page Register is programmed into the Page Register. The address supplied by the Page Register cannot be programmed to automatically increment or decrement. It can only be changed by reprogramming it. This is why the DMA transfers have a limited block size (64 kB blocks for 8-bit data transfers and 128 kB blocks for 16-bit data transfers). See section 10.1.3 Programming the DMA System, for a complete description of how to program the Base Word Register, Mode Register and the Page Register.

The manner in which a DMA controller and 8-bit latch generate a 16-bit memory address will be discussed first. This address generation scheme will then be extended to the PC87120 DMA implementation including the Page Register.

A DMA controller requires an 8-bit latch to create a 16-bit memory address. When the first DMA transfer for a channel occurs, the DMA controller outputs the low address byte of the programmed 16-bit address directly onto the system memory address bus through its own eight address outputs. At the same time, it outputs the high address byte of the programmed 16-bit address onto the data bus through its own eight data outputs. The high byte of the 16-bit programmed memory address is then latched onto the system memory address bus from the data bus through the 8-bit latch. After the DMA transfer, the DMA controller automatically increments or decrements the memory address according to how that channel has been programmed. On subsequent DMA transfers for that channel, the 8-bit latch will only be updated when the incrementing or decrementing of the address causes the low byte of the address to generate a carry or a borrow.

TABLE 10-4. Example of Channel Priorities as Various Channels are Serviced when DMAC2 = Rotating, DMAC1 = Fixed

Priority (Highest = 1, Lowest = 7)	Initial State	State after Channel 2 is Serviced	State after Channel 6 is Serviced	State after Channel 3 is Serviced
1	Channel 0	Channel 5	Channel 5	Channel 5
2	Channel 1	Channel 6	Channel 7	Channel 7
3	Channel 2	Channel 7	Channel 0	Channel 6
4	Channel 3	Channel 0	Channel 1	Channel 0
5	Channel 5	Channel 1	Channel 2	Channel 1
6	Channel 6	Channel 2	Channel 3	Channel 2
7	Channel 7	Channel 3	Channel 6	Channel 3

In the PC87120 DMA memory address generation, each DMA controller and its associated 8-bit latch generate 16 bits of the system memory in the manner just described. DMA transfers made through DMAC1 will be 8-bit data transfers. DMAC1 and its 8-bit latch generate the sixteen address bits A0–A15. The Page Register supplies address bits A16–A23. Since the Page Register must be reprogrammed to define a block boundary, 8-bit data transfers are limited to 64 kB blocks. DMA transfers made through DMAC2 will be 16-bit data transfers. The DMA system in the PC87120 will force address bit A0 low when a 16-bit DMA transfer is made. DMAC2 and its 8-bit latch generate the sixteen address bits A1–A16 and the Page Register supplies address bits A17–A23. Once again, since the Page Register must be reprogrammed to define a block boundary, 16-bit data transfers are limited to 128 kB blocks.

Note that during the DMA accesses, the PC87120 sets the Byte High Enable ($\overline{\text{BHE}}$) signal as follows: 8-bit DMA accesses, $\overline{\text{BHE}} = \overline{\text{A0}}$; 16-bit DMA accesses, $\overline{\text{BHE}} = 0$. Additionally, it forces Buffered Address Latch Enable (BALE) high.

10.1.3 Programming the DMA System

The PC87120 DMA system is programmable. Both DMA controllers and the Page Register must be configured prior to performing a DMA service. The DMA controllers and Page Register may be reconfigured as required. Programming the DMA Controllers is done in the Program Condition.

Upon power up all of the internal DMA controller registers must be initialized. Since DMAC1 is cascaded through channel 4 of DMAC2, DMAC1 should be programmed first to ensure that this channel is correctly initialized.

10.1.3.1 Program Condition

The internal registers of either of the DMA Controllers can only be programmed when that DMA Controller is in the Program Condition. The Program Condition can be entered when no DMA channel is being serviced, thus HLDA is inactive. During these "idle periods", the DMA controller will sample the PC87120 internally produced DMA Controller Chip Select signal to determine if the CPU is attempting to program the internal DMA Registers. It is the responsibility of the system designer to assure that a DMA controller is not programmed while its HLDA signal is active high. The PC87120 internal DMA controller chip select will be set when an IO access is made to a DMA Controller register.

In the program condition, the four lowest address bits become inputs to the DMA Controller. Thus, for DMAC1, address bits A0–A3 become inputs. For DMAC2, address bits A1–A4 are inputs. These address bits are used in conjunction with $\overline{\text{IOW}}$, and $\overline{\text{IOR}}$ to select the appropriate internal DMA Controller Register or Command.

10.1.3.2 Programming the DMA Controller Registers

Once in the Program condition, the internal DMA Controller registers can be programmed to select the DMA transfer mode as well as the type of transfer for each channel. The memory address corresponding to the DMA transfer can also be programmed. Further, many other options associated with the transfer mechanism can be programmed. Following is a complete description of the registers than can be programmed and what they control. (IMPORTANT: When programming a channel, one should first mask DMA requests to that channel to prevent the DMA Controller from attempting to service a request on a channel that has not been completely programmed. If this occurs, the DMA Controller could enter an unknown state.) When programming all the channels on a DMA controller, the DMA controller can first be disabled by setting bit 2 in the command register.

Command Register

This register is used to control the operation of the DMA controller. In the PC87120, each DMA controller has its own command register. The command register is written through IO location 08h for DMA Controller 1 (Channels 0–3) and IO location D0h for DMA Controller 2 (Channels 4–7).

This register can be cleared with a RESET or Master Clear Command (described in section 10.1.3.3).

Bit 7—DMA Acknowledge Bit (DACK)

0 = DACK signals active low
1 = DACK signals active high

To maintain compatibility with the PC/AT this bit should be set to zero for DMAC2, making its DACK signals active low.

Bit 6—DMA Request Bit (DRQ)

0 = DREQ signals active high
1 = DREQ signals active low

Bit 5—Extended Write Bit (EW)

0 = Extended Write disabled
0 = Extended Write enabled
X if compressed timing is enabled, bit 3 = 1

Bit 4—Rotating Priority Bit (RP)

0 = Fixed Priority
1 = Rotating Priority

Bit 3—Compressed Timing Bit (CT)

0 = Normal Timing
1 = Compressed Timing
X if Memory-to-Memory Transfers are enabled, bit 0 = 1

Bit 2—DMA Controller Disable Bit (CD)

0 = DMA Controller Enabled
1 = DMA Controller Disabled

Bit 1—Address Hold Bit (AH)

This bit determines whether or not the memory address in the DMA access for channel 0 is held constant. It is valid only during a memory-to-memory transfer.

0 = Address Hold Disabled on DMA Channel 0, memory address will increment or decrement as programmed

1 = Address Hold Enabled on DMA Channel 0, memory address is held constant.

X if Memory-to-Memory Transfers are disabled, bit 0 = 0

Bit 0—Memory-to-Memory Transfer Bit (M-M)

0 = Memory-to-Memory Transfers Disabled

1 = Memory-to-Memory Transfers Enabled

Current Address Register

Each DMA channel has a corresponding 16-bit current address register which contains 16 bits of the memory address used during a DMA transfer. This address is automatically incremented or decremented following a transfer according to what has been programmed. This register is written or read in 8-bit accesses. The current address is loaded with the base address stored in the Base Address Register when an End of Process ($\overline{EOP}$) is received and if Autoinitialize is selected. Note that the address in channel 0 of the DMA controller can be prevented from incrementing or decrementing if the command register sets the Address Hold (AH) bit high.

IO Address for Current Address Register	DMA Channel Assignments
00h	Channel 0
02h	Channel 1
04h	Channel 2
06h	Channel 3
C0h	Channel 4
C4h	Channel 5
C8h	Channel 6
CCh	Channel 7

Writing to these IO addresses actually writes to both the Base Address and Current Address Registers. Reading from these IO addresses reads only the Current Address Register.

Bits 15–0—Current DMA Memory Address Bits

These bits determine the 16 bits of the system memory address produced by the DMA controller and its corresponding 8-bit latch. Note that these bits are read or written in two 8-bit accesses where an internal flip-flop automatically switches between the high and low address bytes.

When programming the Current Address register for 8-bit data transfers (channels 0–3), the bits 15–0 correspond to the system address bits A15–A0. When programming the Current Address register for 16-bit data transfers (channels 5–7), the bits 15–0 correspond to the system address bits A16–A1. Therefore, the address programmed is the system memory address divided by two for 16-bit data transfers.

Current Word Count Register

The value stored in this 16-bit register is used to determine the number of DMA transfers for a channel. The number of actual transfers made is one greater than the value stored in this register because the value in the register decrements after each transfer until FFFFh is reached. This condition is referred to as Terminal Count (TC). This register is written or read in 8-bit accesses. The Current Word Count is loaded with the base word count stored in the Base Count Register when an End of Process ($\overline{EOP}$) is received if Autoinitialize is selected.

IO Address for Current Word Count Register	DMA Channel Assignments
01h	Channel 0
03h	Channel 1
05h	Channel 2
07h	Channel 3
C2h	Channel 4
C6h	Channel 5
CAh	Channel 6
CEh	Channel 7

Writing to these IO addresses write to both the Base Word Count and Current Word Count Registers. Reading from these IO addresses reads only the Current Word Count Register.

Bits 15–0—Current Word Count Bits

These bits determine the number of DMA transfers to be made for that channel. The number of DMA transfers will be one more than the contents of this register. Note that these bits are read or written in two 8-bit accesses where an internal flip-flop automatically switches between the high and low byte of the count value.

Base Address Register

The 16-bit Base Address Register is automatically programmed when the Current Address Register is programmed. This register cannot be read. The contents of this register will be loaded into the Current Address Register automatically upon a $\overline{EOP}$ if Autoinitialize has been selected.

Base Word Count Register

The 16-bit Base Word Count Register is automatically programmed when the Current Word Count Register is programmed. This register cannot be read. The contents of this register will be reloaded into the Current Word Count Register automatically upon an $\overline{EOP}$ if Autoinitialize has been selected.

Mode Register

Each DMA channel has a 6-bit DMA transfer mode register. The mode register for all channels on a DMA controller are programmed through the same IO location. The channel being programmed is set with 2 bits in the 8-bit register. The mode register for DMAC1 is written through IO location 0Bh. The mode register for DMAC2 is written through IO location D6h.

Bit 7—Mode Select Bit 1 (M1)

Bit 6—Mode Select Bit 0 (M0)

These two bits select the transfer mode for the DMA channel as shown:

Bit 7	Bit 6	Transfer Mode
0	0	Demand Mode
0	1	Single Mode
1	0	Block Mode
1	1	Cascade Mode

Since the PC87120 system uses cascaded DMA Controllers through channel 4 (channel 0 of DMAC2), DMA channel 4 must always be programmed for cascade mode.

Bit 5—Current Address Decrement/Increment (DEC)

0 = Increment Current Address

1 = Decrement Current Address

Bit 4—Autoinitialize Bit (A1)

0 = Disable Autoinitialization

1 = Enable Autoinitialization

Bit 3—Transfer Type Bit 1 (TT1)

Bit 2—Transfer Type Bit 0 (TT0)

These two bits are used together to select the DMA transfer type for the selected channel in the following manner:

Bit 3	Bit 2	DMA Transfer Type
0	0	Verify Transfer
0	1	Write Transfer
1	0	Read Transfer
1	1	Not Allowed
X	X	If Cascade Mode has been selected

Bit 1—Channel Select Bit 1 (CS1)

Bit 0—Channel Select Bit 0 (CS0)

These two bits select the DMA channel for which the mode is being programmed as shown:

Bit 1	Bit 0	DMAC1	DMAC2
0	0	Channel 0	Channel 4
0	1	Channel 1	Channel 5
1	0	Channel 2	Channel 6
1	1	Channel 3	Channel 7

Request Register

This register is used to issue a DMA service request for a specific DMA channel through software. The channel must be programmed to operate in block mode if a software request is made. Note that DMA requests issued through software are not maskable. However, DMA requests issued through software will be serviced according to the priority programmed.

The Request Register is written through IO location 09h for DMAC1, channels 0-3 and through IO location D2h for DMAC2, channels 4-7. The Request Register corresponding to the active DMA channel will be cleared when an external $\overline{\text{EOP}}$ or a terminal count is encountered. The entire request register will be cleared upon a RESET.

Bits 7-3—Don't Care

Bit 2—Request Bit (RB)

0 = Reset Software DMA Request for the channel

1 = Set Software DMA Request for the channel

Bit 1—Software DMA Request Channel Select Bit 1 (RS1)

Bit 0—Software DMA Request Channel Select Bit 0 (RS0)

These two bits determine the channel for which the software DMA request will be generated or cleared in the following manner:

Bit 1	Bit 0	DMAC1	DMAC2
0	0	Channel 0	Channel 4
0	1	Channel 1	Channel 5
1	0	Channel 2	Channel 6
1	1	Channel 3	Channel 7

Mask Register

The bits in this register are used to mask a DMA request generated by a DREQ signal (not software). Each channel has a corresponding mask bit. If a channel is not programmed to Autoinitialize, the channels mask bit in this register will automatically be set when an $\overline{\text{EOP}}$ is received. The PC87120 internal mask register for each DMA controller can be written in two different methods, each method entered through different IO addresses.

The first method available to mask a DMA request is to mask or unmask the DMA requests one channel at a time. This is done through IO location 0Ah for channels 0-3 in DMAC1 or through IO location D4h for channels 4-7 in DMAC2. The bits are defined in the following manner when using this method to write to the Mask Register:

Bits 7-3—Don't Care

Bit 2—Mask Bit (MB)

0 = Clear DMA request Mask for the channel

1 = Set DMA request Mask for the channel

Bit 1—Mask DMA Request for Channel Select Bit 1 (MS1)

Bit 0—Mask DMA Request for Channel Select Bit 0 (MS0)

These two bits determine the channel for which the DMA request will be masked or cleared as follows:

Bit 1	Bit 0	DMAC1	DMAC2
0	0	Channel 0	Channel 4
0	1	Channel 1	Channel 5
1	0	Channel 2	Channel 6
1	1	Channel 3	Channel 7

The second method available to mask a DMA request is to mask or unmask the DMA requests for all channels at the same time. This is done through IO location 0Fh for channels 0-3 in DMAC1 or through IO location DEh for channels 4-7 in DMAC2. The bits are defined in the following manner when using this method to write to the Mask Register:

Bits 7-4—Don't Care

Bit 3—Mask DMAC1 Channel 3 or DMAC2 Channel 7 Bit (MB3(MB7))

0 = Clear DMA request Mask for Channel 3(7)

1 = Set DMA request Mask for Channel 3(7)

Bit 2—Mask DMAC1 Channel 2 or DMAC2 Channel 6 Bit (MB2(MB6))

0 = Clear DMA request Mask for Channel 2(6)

1 = Set DMA request Mask for Channel 2(6)

Bit 1—Mask DMAC1 Channel 1 or DMAC2 Channel 5 Bit (MB1(MB2))

0 = Clear DMA request Mask for Channel 1(5)

1 = Set DMA request Mask for Channel 1(5)

Bit 0—Mask DMAC1 Channel 0 or DMAC2 Channel 4 Bit (MB0(MB4))

0 = Clear DMA request Mask for Channel 0(4)

1 = Set DMA request Mask for Channel 0(4)

Status Register

The status register can be read to determine which channels are requesting a DMA service and which channels have encountered either an internal $\overline{\text{EOP}}$ (a TC) or an external $\overline{\text{EOP}}$. The status for channels 0 through 3 can be read through IO location 08h. The status for channels 4 through 7 are read through IO address D0h.

The status bits are defined as follows:

Bit 7—DMA Request on Channel 3 (7) (DRQ3(DRQ7))

0 = No pending DMA request on Channel 3 (7)

1 = A DMA request is pending on Channel 3 (7)

Bit 6—DMA Request on Channel 2 (6) (DRQ2(DRQ6))

0 = No pending DMA request on Channel 2 (6)

1 = A DMA request is pending on Channel 2 (6)

Bit 5—DMA Request on Channel 1 (5) (DRQ1(DRQ5))

0 = No pending DMA request on Channel 1 (5)

1 = A DMA request is pending on Channel 1 (5)

Bit 4—DMA Request on Channel 1 (4) (DRQ0(DRQ4))

0 = No pending DMA request on Channel 0 (4)

1 = A DMA request is pending on Channel 0 (4)

Bit 3—$\overline{\text{EOP}}$ (Terminal Count) on Channel 3 (7) (TC3(TC7))

0 = Channel 3 (7) has not encountered $\overline{\text{EOP}}$

1 = Channel 3 (7) has encountered $\overline{\text{EOP}}$

Bit 2—$\overline{\text{EOP}}$ (Terminal Count) on Channel 2 (6) (TC2(TC6))

0 = Channel 2 (6) has not encountered $\overline{\text{EOP}}$

1 = Channel 2 (6) has encountered $\overline{\text{EOP}}$

Bit 1—$\overline{\text{EOP}}$ (Terminal Count) on Channel 1 (5) (TC1(TC5))

0 = Channel 1 (5) has not encountered $\overline{\text{EOP}}$

1 = Channel 1 (5) has encountered $\overline{\text{EOP}}$

Bit 0—$\overline{\text{EOP}}$ (Terminal Count) on Channel 0 (4) (TC0(TC4))

0 = Channel 0 (4) has not encountered $\overline{\text{EOP}}$

1 = Channel 0 (4) has encountered $\overline{\text{EOP}}$

Reading the Status Register will clear the Terminal Count Bits 0 through 3.

Temporary Register

This register will contain the last byte of data transferred in a Memory-to-Memory transfer. The register is used as a holding register to store the data between the memory accesses. This register is cleared by a RESET.

Since the PC87120 system does not support Memory-to-Memory Transfers using the DMAC2, its Temporary Register will never be written to. However, it still can be read. To read the Temporary Register for DMAC1, read IO location 0Dh. To read the Temporary Register for DMAC2, read IO location DAh.

10.1.3.3 Programming DMA Commands

There are three software commands that each DMA Controller decodes in its Program Condition. All of these commands are executed upon any write to its IO location. None of the commands decode the specific data written to the IO location. The commands are:

Clear First/Last Flip-Flop

This command is executed in DMAC1 when a write is made to IO location 0Ch and in DMAC2 when a write is made to IO location D8h. This command clears the internal flip-flop which toggles between the lower and upper bytes of the 16-bit registers. This command must be executed prior to writing a current base address or current base word count.

Master Clear

This command is executed in DMAC1 when a write is made to IO location 0Dh and in DMAC2 when a write is made to IO location DAh. This command is equivalent to a hardware RESET. The following registers are all cleared: Command, Request, and Temporary Registers. Further, the internal First/Last Flip Flop is cleared and the Mask Register is set. The DMA Controller will enter the Idle Cycle.

Clear Mask Register

This command is executed in DMAC1 when a write is made to IO location 0Eh and in DMAC2 when a write is made to IO location DCh. This command clears the mask on all of the DMA channels associated with the DMA Controller allowing DMA requests to be made.

10.1.3.4 Programming the DMA Page Boundary Addresses

Page Register

The Page Register consists of 16 8-bit registers accessed through IO addresses 80–9Fh. Seven of these 8-bit registers are assigned to the seven available DMA channels and one of the 8-bit registers is used for Refresh. The contents of the Page Register is used to generate the high address bits of the 24-bit address in the PC/AT. For DMA channels 0 through 3, the Page Register generates address bits 16–23. For DMA channels 5 through 7, the Page Register value is used to generate address bits 17–23. The Page Register Address Map is shown in Table 10-5.

TABLE 10-5. Page Register

IO Hex Address	PC87120 8-Bit Page Register Assignments
087	DMA Channel 0
083	DMA Channel 1
081	DMA Channel 2
082	DMA Channel 3
08B	DMA Channel 5
089	DMA Channel 6
08A	DMA Channel 7
08F	Refresh Cycle

Bits 0–7 Page Boundary Address

These bits make up the upper bits in the address generation for DMA transfers. For 8-bit transfers (channels 0 through 3) these bits will correspond to A16–A23. For 16-bit DMA transfers (channels 5 through 7), these bits will correspond to A17–A23.

10.1.4 Idle Cycle

Upon RESET, or when no DMA requests are being made by a channel, the DMA Controller will be in the Idle Cycle. In the Idle Cycle, the DMA Controller executes State Inactive, SI. While in this cycle, the DMA Controller samples the DMA Request Lines and the internally generated $\overline{\text{CS}}$ to determine if a channel is requesting a DMA service or if the DMA Controller should enter the Program Condition. These samples are made every DMA clock cycle.

When the DMA Controller receives a valid DMA Request from a channel, the DMA Controller will issue a Hold Request and wait for a Hold Acknowledge. Upon issuing the Hold Request, the DMA Controller leaves the Idle Cycle and enters the Active Cycle.

State I: Inactive State

This state occurs in the Idle Cycle. While in the Inactive State, the DMA Controller is inactive, i.e., no DMA requests are being serviced. During this state, the CPU can program the DMA Controller.

10.1.4.1 Program Condition

The DMA Controller can be programmed during the Idle Cycle or state S0 of the Active Cycle. The program condition will occur when the internal DMA Chip select is active low and the HLDA is inactive low. It is the responsibility of the system designer to ensure that the DMA controller will not be programmed when HLDA is active high. In the Program Condition, the CPU can write or read the DMA internal registers or execute one of the three special software commands. See the section 10.1.3 Programming the DMA System for details on programming the DMAC during the program condition.

10.1.5 Active Cycle

The Active Cycle is the cycle in which the actual DMA transfer occurs. This cycle is entered once the DMA Controller has received a valid DMA request and issued a Hold Request to the 80286. Upon entering the Active Cycle, the DMA Controller leaves the state S1 and enters state S0. State S0 remains valid until the CPU returns a Hold Acknowledge. This indicates that the CPU has given control of the busses to the DMA Controller. The DMA Controller can be programmed while in State 0.

After receiving the Hold Acknowledge from the CPU, the DMA Controller leaves State 0 and enters State 1, the first state of the DMA transfer. During State 1, the DMA controller issues the Address Enable for the 8-bit address latch and enters State 2. In State 2, the valid address through the 8-bit latch is generated, and the DMA Controller issues the DMA Acknowledge signal, DACK, to the channel. The DMA channel must hold the DMA Request line DREQ until the corresponding DACK is received. The DMA Controller now enters State 3 in which the appropriate IO and Memory Read and Write signals are generated. The DMA Controller will remain in State 3 until all wait states have been asserted and READY is true. Upon receiving the READY, the DMA Controller enters State 4 where the Memory Read and Write Signals are deasserted.

The DMA Controller can perform the DMA transfer as just described in either the Single Transfer Mode, the Block Transfer Mode, the Demand Transfer Mode, or the Cascade Mode. These transfer modes will be described next.

10.1.5.1 Single Transfer Mode

In this mode, the channel makes one DMA transfer at a time. The DREQ must be held active until DACK becomes active for the DMA transfer to take place. If the DREQ is held active for the entire transfer (through State 4), the DMAC will deassert the Hold Request to the CPU and immediately reassert it. This allows the CPU access to the bus between each transfer.

After each transfer the memory address is incremented or decremented as programmed for the channel. The word count for the channel is decremented for each DMA transfer until it reaches FFFFh which indicates that a terminal count has occurred. The $\overline{\text{EOP}}$ will then be asserted and the DMA transfer is completely terminated for that channel. Once a Terminal Count is reached the channel will Autoinitialize if programmed to do so.

10.1.5.2 Block Transfer Mode

In the Block Transfer Mode, DMA transfers continue until the programmed number of transfers have completed, i.e., a Terminal Count is received. An external $\overline{\text{EOP}}$ will also stop the transfers. The DMA channel need only hold DREQ active until DACK becomes active. This mode is dangerous to use in the AT environment because it locks out DRAM refresh cycles.

10.1.5.3 Demand Transfer Mode

In the Demand Transfer Mode, the DMA transfers continue until one of three conditions is met: a terminal count is reached, an external $\overline{\text{EOP}}$ is received, or DREQ goes inactive before State 4. The third condition allows the peripheral with buffer limitations to stop transfers and catch up. Only the first two conditions will result in an autoinitialize if the channel has been programmed to do so. Once again, this mode can result in DMA refreshes being locked out and therefore is dangerous to use in the AT environment.

10.1.5.4 Cascade Mode

The cascade mode allows additional DMA channels to be added in a fashion which retains the programmed DMA priority system. The additional DMAC devices are daisy chained to an original DMA device by connecting Hold Request and Hold Acknowledge signals of the DMA device to the DREQ and DACK channels of the original device. Note that the channel used to cascade the additional device does not produce any control or addressing signals, it is only used in prioritizing the accesses.

Channel 4 of DMAC2 in the PC87120 is used for cascading DMAC1. Therefore, channel 4 must be programmed for cascade mode.

10.1.6 Types of Transfer

There are four different transfer types which can be performed when transferring in the following three transfer modes: single transfer mode, block transfer mode, or demand transfer mode. The first transfer type transfers data from IO to memory. This transfer type is referred to as a Write Transfer or IO to Memory. The second transfer type, a Read or Memory to IO transfer, moves data from memory to IO. The third transfer type moves data from memory to memory and is referred to as Memory-to-Memory. The final transfer type is actually a pseudo transfer and is referred to as Verify. These transfer types will be further described.

In each of these transfer types, the address signal generated is the memory address. The IO device is determined by the channel priority decoding performed by the DMA Controller.

10.1.6.1 IO to Memory

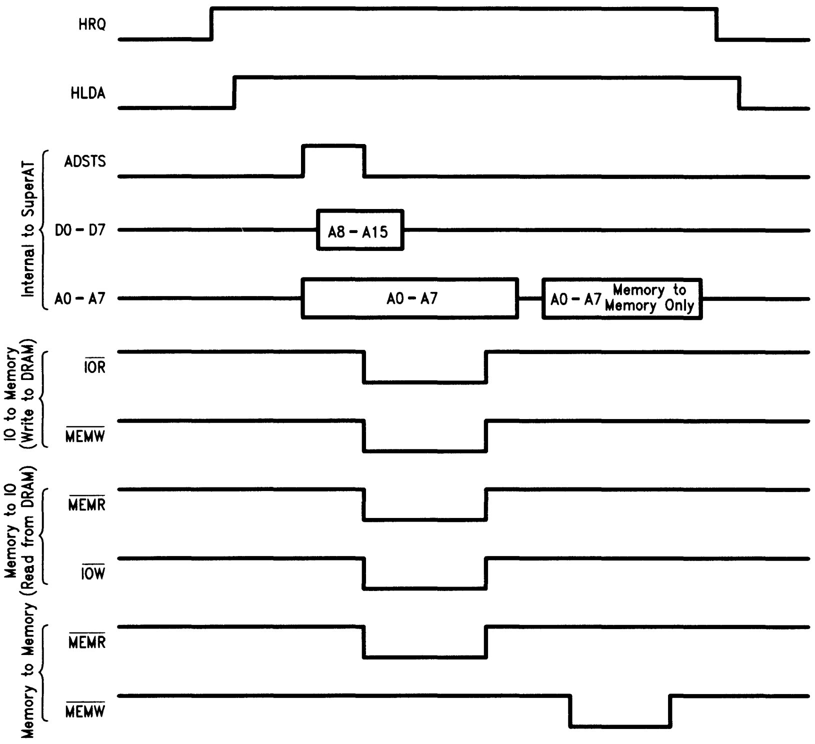
The IO to Memory DMA transfer, or Write Transfer, is activated with the control signals $\overline{\text{MEMW}}$ and $\overline{\text{IOR}}$ are asserted active low. See *Figure 10-2* for the timing of an IO to Memory transfer.

10.1.6.2 Memory to IO

The Memory to IO transfer, or Read Transfer, is activated with the control signals $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$ asserted active low. See *Figure 10-2* for the timing of a Memory to IO transfer.

10.1.6.3 Memory to Memory

This type of transfer requires two DMA cycles. The first moves data from memory to an internal temporary register in the DMAC. The second moves the data from the internal temporary register to memory. This mode allows blocks of data to be transferred from one memory location to another quickly.



TL/F/11050-60

FIGURE 10-2. DMA Transfer Types

The memory to memory transfer requires the use of two channels. This is accomplished by programming in the command register. When memory-to-memory transfer is programmed, channels 0 and 1 operate as the memory-to-memory channels. An implication of this is that the AT does not support 16-bit memory-to-memory transfers since channel 4 (channel 0 of DMAC2) is already physically used for cascading DMAC1. However, 8-bit memory-to-memory transfers are possible.

The memory-to-memory transfer starts by sending a software or hardware DMA Request to channel 0. This channel generates the source memory address. The byte read from the memory location is stored in the DMAC's internal Temporary register. A second DMA transfer is made by channel 1. Channel 1 reads the Temporary register and generates the memory address to transfer it to. The address in channel 0 can be programmed to stay the same. This is useful for initializing large blocks of memory to the same value. The memory-to-memory transfer process is ended when the count of Channel 1 reaches a terminal count or when the external $\overline{EOP}$ is received. See *Figure 10-2* for the timing of a Memory-to-Memory transfer.

10.1.6.4 Verify

The verify transfer is a pseudo read or write transfer. All of the addressing signals and $\overline{EOP}$ responses are generated. However, the IO and memory control signals remain inactive.

10.1.7 Autoinitialization

A channel can be set up to autoinitialize by programming the autoinitialize bit for the channel in the Mode Register. When a channel autoinitializes, the values stored in the Base Address and Base Word Count Register are loaded into the Current Address and Current Word Count Registers respectively. The autoinitialization is initiated by an $\overline{EOP}$ or a Terminal Count.

To autoinitialize channels configured for memory-to-memory transfers, both channel 0 and channel 1 must load the same value into the corresponding Base Word Count registers. Additionally, if an external $\overline{EOP}$ is used to generate the autoinitialize, then the $\overline{EOP}$ should be applied during both channel 0's and channel 1's portion of the Memory-to-Memory transfer cycle.

Autoinitialization should be used when the data to be transferred covers more than one page block. After a terminal count and hence autoinitialization, one would reprogram the page register to a new page boundary. This prevents transferring the same data, or overwriting the previously written data.

10.1.8 Compressed Timing

Compressed timing can be used to reduce the DMA transfer time. In compressed timing, the state S3 is removed.

10.2 TIMER OPERATION

10.2.1 Programming the Timer

The PC87120 contains a Programmable Interval Timer (PIT) that is compatible with the Intel 8254 Programmable Interval Timer.

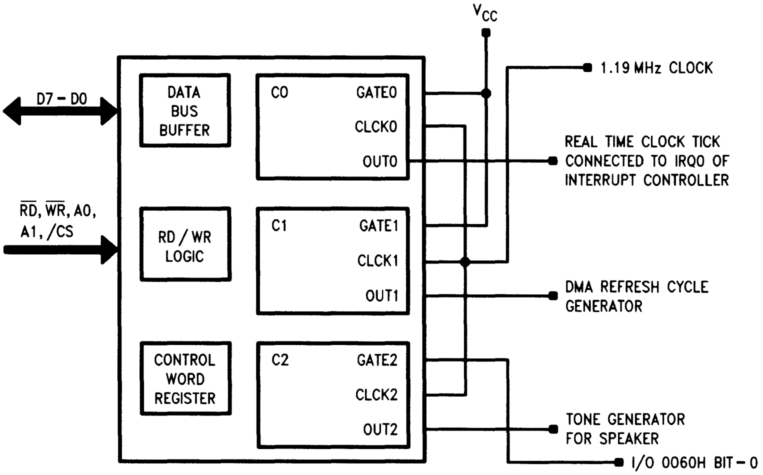
This PC/AT compatible programmable interval timer system has three identical timer channels (CH0–CH2) that can be used to generate accurate time delays under software control. In a PC/AT environment the channels are assigned as follows: (See *Figure 10-3*).

CW0 is a general purpose software interrupt timer and operates in conjunction with the IRQ0 input of the internal Programmable Interrupt Controller to provide the real time clock tick.

CW1 is used internally by the PC87120 to provide the DRAM refresh cycle.

CW2 provides tone generation for the speaker.

The PIT is programmed through I/O accesses to 0040H–0043H for CH0 and CH1 operation and 0040H–0043H, 0061H Bit 0 for CH2 tone generation.



TL/F/11050-61

FIGURE 10-3. PC87120 PIT Block Diagram

Counter Description

The three timer channels, CH0, CH1 and CH2, are outputs of three 16-bit presettable down counters that can be programmed to count in binary or binary coded decimal (BCD). The counters are completely independent. Each can operate as a timer or counter and can be programmed to operate in various modes. The counters are driven from a common 1.19 MHz clock signal generated from dividing the 14.31818 MHz SYSCLK by 12. Logic common to all counters directs read and write data from the 8-bit data port to the appropriate counter.

Each counter is identical and contains a Control Word Register, a Status Register, a 16-bit down counting element, two input holding registers, two output holding registers, a clock input, a gate input and an out signal.

The contents of the Control Word Register determines how the counter operates. When latched, the Status Register contains the current contents of the Control Word Register and the status of the output and null count flag. (See section 10.2.3.2 describing the Read-Back Command.) The 16-bit down counting element is presettable and synchronous.

The Input Holding Registers are two 8-bit latches used to store the LSB and MSB of a count written to the counter until they are transferred to the counting element on the next falling edge of the clock input. The Output Holding Registers are two 8-bit latches used to latch the LSB and MSB of the present count thus enabling the count to be read.

The Clock input provides the clock signal for loading and decrementing the counting element. The Gate input is defined by the mode and controls the counter. The Out signal is determined by the counter mode and status of the counting element.

The counters can operate in six different modes. (See section 10.2.4 describing counter modes.) However in a PC/AT system the Gate inputs for counter 0 and counter 1 are tied to V_{CC}, hence, modes 1 and 5 are not usable as they require a rising edge on the gate to achieve a one shot sequence. Counter 2 is capable of operating in all six modes and is used in mode three as a gated pulse generator for the audio speaker. In counter modes 0, 1, 4 and 5 the counter wraps around to the highest count and continues counting. FFFFH for binary or 9999 for BCD. Modes 2 and 3 are periodic and the counter reloads itself with the initial count and continues from there. In a PC/AT environment the three channels operate in the following modes:

—CH0 operates in Mode 2.

—CH1 operates in Mode 2.

—CH2 operates in Mode 3.

10.2.1 Programming the Timer

The PIT registers and counters power up with random contents. Therefore each counter must be programmed before it can be used. Counters are programmed by first writing to the Control Word Register followed by an initial count for the appropriate counter. The Control Word Register is at address location 0043H and is write only. (See *Figure 10-4.*) The Control Word determines the mode/command and format to be issued to the appropriate counter. (See *Figure 10-5.*) When a Control Word is written to a counter, all control logic is immediately reset and the Out pin goes to a known initial state.

Address	Register	Rights
0040H	Counter 0	RD/WR
0041H	Counter 1	RD/WR
0042H	Counter 2	RD/WR
0043H	Control Register	WR

FIGURE 10-4. Address Locations

Control Word Register (0043H) Write only.

D7	D6	D5	D4	D3	D2	D1	D0
C4	C3	C2	C1	M2	M1	M0	BCD

C4, C3, C2, C1: determine command and format (See Table 10.6).

M2, M1, M0: determine counter mode (See Table 10.7).

BCD: selects binary count (0) or binary coded decimal (1).

FIGURE 10-5. Control Word (00434)

10.2.2 Write Operations

For each counter, the Control Word must be written before the initial count. The initial count must follow the count format specified in the control word. (See *Figure 10-5.*) Because the Control Word specifies to which counter it applies, no special instruction sequence is required as long as the Control Word is written at some time before the initial count.

New initial counts may be written into a counter any time after programming without re-issuing a Control Word, as long as the existing format is observed. This will not affect the counter's programmed mode. The counters are 16-bit and are accessed through an 8-bit port. This means that count writes can only be performed in three ways: 1) LSB only, 2) MSB only, 3) LSB followed by MSB. The method of access is defined by the Control Word format. (See *Figure 3.*) When a Control Word is written, the Input Holding Registers are automatically set to zeroes.

TABLE 10-6. Control Word Command Table

C4	C3	C2	C1	Command/Format
0	0	0	0	Counter Latch Command for Counter 0
0	0	0	1	Counter 0 LSB RD/WR
0	0	1	0	Counter 0 MSB RD/WR
0	0	1	1	Counter 0 LSB and MSB RD/WR
0	1	0	0	Counter Latch Command for Counter 1
0	1	0	1	Counter 1 LSB RD/WR
0	1	1	0	Counter 1 MSB RD/WR
0	1	1	1	Counter 1 LSB and MSB RD/WR
1	0	0	0	Counter Latch Command for Counter 2
1	0	0	1	Counter 2 LSB RD/WR
1	0	1	0	Counter 2 MSB RD/WR
1	0	1	1	Counter 2 LSB and MSB RD/WR
1	1	X	X	Read-Back Comand

TABLE 10-7. Mode Decode

M2	M1	M0	Counting Modes
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	1	0	Mode 4
1	1	1	Mode 5

10.0 Peripheral Functions (Continued)

10.2.3 Read Operations

It is usually desirable to read the value of a counter without disturbing the present count updates. The PC87210 PIT is able to provide this function in two ways; the Counter Latch and Read-back Commands.

10.2.3.1 Counter Latch Command

This command is written to the Control Word Register (address 0043H). The bits C3 and C4 select one of the three counters. Bits C2 and C1 distinguish this command from a Control Word. (See *Figure 10-6.*) When the Counter Latch Command is received by the Control Word Register, the Output Holding Registers latch the count of the selected counter. This count is held until read by the CPU or counter is reprogrammed. If a counter is latched, and then latched again before the CPU has read it, the second count latched will be ignored and the value in the Output Holding Registers will be that of the first count latched. In all cases, the count must be read according to the programmed format.

Counter Latch Command Byte.

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC2	0	0	X	X	X	X

SC1/SC2: Select counter(s) and read-back command.

0	0	counter 0
0	1	counter 1
1	0	counter 2
1	1	read-back command.

D5/D4: Are always 0,0 to designate counter latch command.

FIGURE 10-6. Counter Latch Command

10.2.3.2 Read-Back Command

In the same respect as the Counter Latch Command, the bits C3 and C2 distinguish this command from a Control Word when writing to the Control Word Register. When these bits are 1,1 the Control Word becomes the Read-Back command and provides alternative bit definitions. (See *Figure 10-7.*)

The Read-Back Command allows the user to check the value of a selected counter, determine the programmed mode and monitor the status of the Out pin and Null Count Flag. The counter/counters is/are selected by setting the bits D3,D2,D1 and D0 of the Read-Back command. (See *Figure 10-7.*)

This command may be used to latch multiple Output Holding Registers for the counters in the PIT. By setting D5 to zero and selecting the desired counters to be read, all three counter's Output Holding Registers can be latched at the same instant. Thus, enabling all counters to be read at the same instant. When reading the counters, the programmed format should be observed. The specific counter is automatically unlatched when read or when re-programmed.

The Read-Back command can also latch status information of selected counters by setting bit D4. (See *Figure 10-7.*) The Status Register must be latched to be read. The status of a counter is obtained by a read fom the selected counter when the Status Register is latched. This is the Status Byte. (See *Figure 10-8.*)

Read-Back Command Byte

D7	D6	D5	D4	D3	D2	D1	D0
1	1	$\overline{\text{COU}}$	$\overline{\text{STA}}$	C2	C1	C0	0

D5: 0 = Latch selected counter(s) count

D4: 0 = Latch selected counter(s) status

D3: 1 = Counter 2

D2: 1 = Counter 1

D1: 1 = Counter 0

FIGURE 10-7. Read Back Command

The Status Byte

D7	D6	D5	D4	D3	D2	D1	D0
O/P	NC	C1	C0	M2	M1	M0	BCD

D7: 1 = Out is high

0 = Out is low

D6: 1 = Null Count

0 = Count can be read

D5–D0: Gives counter format and mode. (See *Figure 10-5.*)

FIGURE 10-8. Status Byte

The bits D5 through D0 contain the counter's programmed mode as it was written in the last Control Word. Bit D7 contains the state of the Out pin. Bit D6 indicates when the last count written to the Input Holding Register has been transferred into the Counting Element. Only the counter specified by the Control Word will have its Null Count set to one. If the counter is programmed for two-byte counts the Null Count bit is set to one after the second byte.

Both the Status and the Count of the selected counters can be latched simultaneously by setting bits D4 and D5. The first read after this will always return the status byte. The next reads return latched counts depending on the number of counters selected. Subsequent reads return unlatched counts.

10.2.4 Mode Descriptions

10.2.4.1 Mode 0: Interrupt on Terminal Count

After the Control Word is written, the Out pin is low and remains low until the count reaches zero. Out remains high until a new count or a new mode 0 Control Word is written. Gate = 1 enables counting, Gate = 0 disables counting.

The next clock pulse loads initial count after a Control Word and initial count has been written. This clock pulse does not decrement the count. If a two byte count is written, the first byte disables counting and Out is set low, the second byte allows the new count to be loaded on the next clock pulse. An initial count can be written when Gate = 0 and will still be loaded on the next clock pulse. No clock pulse is needed when Gate goes to a 1 to load the initial count.

10.2.4.2 Mode 1: Hardware Re-triggerable One-Shot

From initially high, Out will go low on the clock pulse after a trigger. This begins the one-shot pulse. Out remains low until the count reaches zero. Out then goes high and remains high until the next trigger then goes low on the next clock pulse. After writing a Control Word and initial count the counter is armed. A trigger loads the initial count and sets Out low on the next clock pulse. The one-shot is Retriggerable and can be repeated without writing the same count or Control Word into a counter. Gate has no effect on Out.

10.2.4.3 Mode 2: Rate Generator

Typically used to generate a Real Time clock tick, Out is initially high. When the initial count is decremented to 1, Out goes low for one clock pulse then goes high again. The counter reloads the initial count and the process is repeated. This same sequence is repeated indefinitely. Gate = 1 enables counting, Gate = 0 disables counting. If Gate goes low during an output pulse, Out goes high immediately. A trigger reloads the counter with the initial count on the next clock pulse. After the trigger, Out goes low when the count has expired. After writing a Control Word and initial count, the counter will be loaded on the next clock pulse. After the initial count is written, Out goes low when the count has expired. Thus, the synchronizing of the counter by software can be achieved. Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the counter will be loaded with the new count on the next clock pulse. Counting will continue from the new count. A count of 1 is illegal in mode 2.

10.2.4.4 Mode 3: Square Wave

Typically used for Baudrate generation, Out will be initially high and goes low when half of the initial count has expired for the remainder of the count. This is periodic and the sequence is continued indefinitely. This is similar to mode 2. Gate = 1 enables counting, Gate = 0 disables counting. If Gate goes low during an output pulse, Out goes high immediately. A trigger reloads the counter with the initial count on the next clock pulse. After the trigger, Out goes low when the count has expired. After writing a Control Word and initial count, the counter will be loaded on the next clock pulse. After the initial count is written, Out goes low when the count has expired. Thus, the synchronizing of the counter by software can be achieved. Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the counter will be loaded with the new count on the next clock pulse. Counting will continue from the new count. Otherwise, the new count will be loaded at the end of the half-cycle.

For even counts: Out is initially high. The initial count is loaded on the next clock pulse and is decremented by two on succeeding clock pulses. When the count expires, Out goes low. The count is reloaded and the process repeats indefinitely.

For odd counts: Out is initially high. The initial count minus one is loaded on the next clock pulse and is decremented by two on succeeding clock pulses. One pulse after the count expires, Out goes low. The count is reloaded with the initial count minus one and the process repeats indefinitely.

10.2.4.5 Mode 4: Software Triggered Strobe

With Out initially high, Out goes low for one clock pulse and then goes high when the initial count expires. Writing the initial count triggers the counter. Gate = 1 enables counting, Gate = 0 disables counting. Gate has no effect on Out. The counter will be loaded on the next clock pulse after writing the Control Word and initial count. This clock pulse does not decrement the count. A new count, written during counting, will be loaded on the next clock pulse and counting will continue from the new count. If a two byte count is written, the first byte will have no effect on counting, the second byte will allow the new count to be loaded on the next clock pulse.

10.2.4.6 Mode 5: Hardware Triggered Strobe

Out is initially high. Counting is triggered by a rising edge on the Gate input. Out will go low for one clock pulse when the count has expired. After writing the initial count and Control Word, the counter will not be loaded until the clock pulse after the trigger. This pulse does not decrement the count. The counter is loaded on the next clock pulse after a trigger. Gate has no effect on Out. If a new count is written during counting, the current count will not be affected. If a trigger is issued before a count expires but after a new count is written, the counter will be loaded with the new count on the next clock pulse and counting will continue from there.

10.2.5 Gate Input

Sampling of the Gate input occurs on the rising edge of the clock input to the counter. The Gate input is level sensitive and the logic level is sampled on the rising edge of the clock in modes 0, 2, 3 and 4. In modes 1, 2, 3 and 5 the Gate input is rising edge sensitive. In these latter modes, a rising edge of the Gate sets an edge sensitive flip-flop in the counter. This flip-flop is sampled on the next rising edge of the clock. Immediately after the flip-flop is sampled, it is reset. This ensures a trigger (rising edge of the Gate input) will always be detected.

10.3 PROGRAMMABLE INTERRUPT CONTROLLER OPERATION

The PC87120 provides two cascaded interrupt controllers (PICs), which are compatible with the Intel 8259-2 Programmable Interrupt Controller. Like the original PC/AT the slave PIC provides support for interrupts 15–8 and is cascaded into the master PIC's interrupt request 2 (IRQ2) input. Besides the cascade signal, the master PIC also supports interrupts 7–3 and 1–0. The interface signals between the two PICs are internal to the PC87120 and may not be accessed from the exterior.

Table 10-8 shows the list of interrupts supported, their functions and their corresponding request input pin (if any). Note that interrupts 13 and 0 are reserved internally for the Coprocessor ERROR interrupt and the Real Time Clock Tick interrupt, respectively.

IRQ1 corresponds to pin 4 of the PC87120 (OPTBUFFUL). In a PC/AT compatible system IRQ1 is used to indicate that the output buffer of the keyboard controller is full.

In a PC/AT compatible system IRQ8 (pin 5 of the PC87120) is used for the Real Time Clock Interrupt. This interrupt request input and IRQ1's input may be used as general purpose interrupt requests if the PC87120 is used in a nonstandard 80286 system.

10.3.1 Programming the PIC

Before operation, the PIC must be initialized by a sequence of four Initialization Command Words (ICW1–ICW4) that establish the PIC's basic mode of operation. After initialization, the PIC operation may be modified by the Operation Command Words (OCW1–OCW3). The OCWs may be written at any time during normal operation.

TABLE 10-8. Interrupt Functions

Interrupt Request	Corresponding Pin Name	Pin Number	Purpose
IRQ0	—	—	Real Time Clock Tick Interrupt
IRQ1	OPTBUFFUL	4	Keyboard Controller Output Buffer Full Interrupt
IRQ2	—	—	Cascade IRQ15–8
IRQ3	IRQ3	139	General Purpose
IRQ4	IRQ4	138	General Purpose
IRQ5	IRQ5	137	General Purpose
IRQ6	IRQ6	135	General Purpose
IRQ7	IRQ7	134	General Purpose
IRQ8	IRQ8	5	Real Time Clock Interrupt
IRQ9	IRQ3	133	General Purpose
IRQ10	IRQ10	132	General Purpose
IRQ11	IRQ11	131	General Purpose
IRQ12	IRQ12	130	General Purpose
IRQ13	—	—	80287 $\overline{\text{ERROR}}$ Interrupt
IRQ14	IRQ14	128	General Purpose
IRQ15	IRQ15	127	General Purpose

10.3.1.1 Initialization Programming

The initialization sequence begins by writing ICW1 to the PIC. For the Master PIC, this is accomplished by writing to an even IO address between 0020h and 003Fh (i.e., 0020h, 0022h, 0024h ... 003Eh) with data bit 4 set to 1. For the Slave PIC, a write to any even IO address between 00A0h and 00BFh (i.e., 00A0h, 00A2h, 00A4h ... 00BEh) with data bit 4 set to 1, will be seen as a ICW1 write cycle.

The writing of ICW1 starts the PIC initialization sequence and automatically establishes the following conditions:

1) The edge-sense logic is reset for all levels. This means that all pending interrupts will be cleared and the PIC must see a low to high transition on an interrupt request input to recognize a new interrupt.
2) The Interrupt Mask is reset for all levels, this will enable all interrupts.
3) Interrupt level 7 is assigned the lowest priority (7).
4) The slave mode addressed is set to 7.
5) Special Mask mode is reset.
6) Status Read pointer is set to IRR.
7) If IC4 = 0, all ICW4 functions are set to zero.

Interrupt Command Word 1 (ICW1)

Master PIC (Write only, even address 0020h-003Eh, Bit 4 = 1)

Slave PIC (Write only, even address 00A0h-00BEh, Bit 4 = 1)

Bits 7-5—Reserved

Bit 4—Must be 1, otherwise the PIC will not recognize the write as a ICW1.

Bit 3—Level Trigger Interrupt Mode (LTIM)

When this bit is set to 1, the PIC will operate in level trigger mode. When set to 0, the PIC operates in edge sensitive trigger mode.

The standard compatible PC/AT sets this bit to 0, edge sensitive mode.

0 = the interrupt request inputs (IRQs) are edge-sensitive; i.e., an interrupt request is generated only by a low-to-high transition of IRQ. The IRQ must return low, then high to generate another request.

1 = the interrupt request inputs (IRQs) are level-sensitive; i.e., requests are generated as long as IRQ remains high.

Bit 2—Address Interval

0 = Interval of 4

1 = Interval of 8

Bit 1—Single PIC (SNGL)

This bit determines if the PIC operates alone or is cascaded with another PIC. When Set to 1, this bit indicates that the PIC is operating alone.

The standard PC/AT sets this bit to 0, indicating that two or more PICs are cascaded (Cascade Mode).

0 = Cascade Mode

1 = Single Mode

Bit 0—Initialization Command 4 (IC4)

When set to 1, ICW4 must be written. When set to 0, ICW4 is not written and all ICW4 functions are set to zero.

The standard compatible PC/AT sets this bit to 1, ICW4 must be written. If this bit is not set to 1, ICW4 will not be written and the PICs internal to the PC87120 will not operate correctly.

0 = ICW4 not written, all ICW1 functions are set to zero

1 = ICW4 must be written

ICW2 must be written during the write cycle that follows the ICW1 write cycle. For the Master PIC ICW2 will be written by a write command to any odd IO address between 0020h to 003Fh (i.e., 0021h, 0023h, 0025 ... 003Fh) immediately following the ICW1 write cycle for the Master PIC. For the Slave PIC ICW2 will be written by a write command to any odd IO address between 00A0h to 00BFh (i.e., 00A1h, 00A3h, 00A5 ... 00BFh) immediately following the ICW1 write cycle for the Slave PIC.

Initialization Command Word 2 (ICW1)

Master PIC (Write only, odd address 0020h-003Eh, following ICW1 write cycle to Master PIC)

Slave PIC (Write only, odd address 00Ah-00BEh, following ICW1 write cycle to Slave PIC)

Bits 7-3—Interrupt Vector Address (T7-T3)

These five bits may be programmed with the vector address bits 7-3 which the PIC provides the processor to read during the interrupt acknowledge sequence.

Bits 2-0—Not Programmed

These three bits are the decoded address of the active interrupt level during an interrupt acknowledge sequence.

ICW3 is written only when there is more than one PIC and they are cascaded; as is the case inside the PC87120 and all compatible PC/ATs. ICW3 is addressed by the write command following ICW2 when ICW1 Bit 1 (SNGL) = 0. For the Master PIC ICW3 will be written by a write command to any odd IO address between 0020h to 003Fh (i.e., 0021h, 0023h, 0025 ... 003Fh) immediately following the ICW2 write cycle for the Master PIC. For the Slave PIC ICW3 will be written by a write command to any odd IO address between 00A0h to 00BFh (i.e., 00A1h, 00A3h, 00A5 ... 00BFh) immediately following the ICW3 write cycle for the Slave PIC.

If SNGL = 1, then ICW3 is not required and the initialization sequence is complete after ICW2 (unless IC4 = 1 = ICW4 must yet be written).

Initialization Command Word 3 (ICW3)

Master PIC (Write only, odd address 0020h-003Eh, following ICW2 write cycle to Master PIC)

Bits 7-0—Slave Indicator

When a 1 is placed into any of these bits, the Master PIC assumes a Slave PIC is connected to the corresponding interrupt request input (IRQ).

In the PC87120 and all compatible PC/ATs, the only slave PIC is fed back into IRQ2 of the Master PIC. So for compatible operations a value of 04h should be written into the Master PIC.

Initialization Command Word 3 (ICW3)

Slave PIC (Write only, odd address 00A0h-00BEh, following ICW2 write cycle to Slave PIC)

Bits 7-3—Reserved

Bits 2-0—Slave ID

These three bits must equal the interrupt request on the Master PIC which this Slave PIC is cascaded into.

In the PC87120 and all compatible PC/ATs, the only slave PIC is fed back into IRQ2 of the Master PIC. So for compatible operations a value of 02h should be written into the Slave PIC.

ICW4 is written only when ICW1 bit 0 (IC4) = 1; otherwise, all of the ICW4 functions are set to 0. ICW4 is addressed by the write cycle following ICW2 if SNGL = 1 and IC4 = 1; or by the write cycle following ICW3 if SNGL = 0 and IC4 = 1. Note, SNGL = 0 and IC4 = 1 is required for PC/AT compatibility.

Writing ICW4 always completes the initialization sequence.

Initialization Command Word 4 (ICW4)

Master PIC (Write only, odd address 0020h-003Eh, following ICW3 write cycle to Master PIC)

Slave PIC (Write only, odd address 00A0h-00BEh, following ICW3 write cycle to Slave PIC)

Bits 7-5—Reserved

Bit 4—Special Fully Nested Mode

(Refer to section 10.3.5.2 for more information about Special Fully Nested Mode)\

0 = Enable Normal Nested Mode

1 = Special Fully Nested Mode

Bits 3-2—Buffer/Master-Slave Mode

When bit 3 is set to 0, Non-Buffered Mode will be used and the Master-Slave bit (Bit 2) is a don't care. When bit 3 is set to 1, buffered mode will be used and bit 2 determines whether the PIC is operating as a master or a slave.

Bit 3	Bit 2	Function
0	X	Non-Buffered Mode
1	0	Buffered Mode (Slave)
1	1	Buffered Mode (Master)

Bit 1—Automatic End of Interrupt (AEOI)

0 = Normal EOI

1 = Automatic EOI

Bit 0—Must be set to 1. If this bit is set to 0, the PIC will operate in an incompatible 8085 mode.

10.3.1.2 Normal Operation Programming

The operation of the PICs may be modified during normal operation by the Operation Command Words (OCW1-OCW3).

Every access of OCW1 is to the Interrupt Mask Register (IMR). Following initialization, all of the bits in the IMR are set to 0, indicating that all interrupts are enabled. To mask off any or all of the interrupts a 1 may be written to the appropriate bit(s) of the IMR using OCW1.

For the Master PIC, OCW1 is addressed by any read or write cycle to an odd address between 0020h-003Fh (i.e., 0021h, 0023h, 0025h ... 003Fh); these accesses exclude any write cycles which are ICW2-4 cycles. For the Slave PIC, OCW1 is addressed by any read or write cycle to an odd address between 00A0h-00BFh (i.e., 00A1h, 00A3h, 00A5h ... 00BFh); these accesses exclude any write cycles which are ICW2-4 cycles.

Operation Command Word 1 (OCW1)

Master PIC (Read/Write, 8-bit, odd address 0020h-003Fh)

Slave PIC (Read/Write, 8-bit, odd address 00A0h-00BFh)

Bit 7—Interrupt 7 Mask

0 = Interrupt 7 is unmasked

1 = Interrupt 7 is masked

Bit 6—Interrupt 6 Mask

0 = Interrupt 6 is unmasked

1 = Interrupt 6 is masked

Bit 5—Interrupt 5 Mask

0 = Interrupt 5 is unmasked

1 = Interrupt 5 is masked

Bit 4—Interrupt 4 Mask

0 = Interrupt 4 is unmasked

1 = Interrupt 4 is masked

Bit 3—Interrupt 3 Mask

0 = Interrupt 3 is unmasked

1 = Interrupt 3 is masked

Bit 2—Interrupt 2 Mask

0 = Interrupt 2 is unmasked

1 = Interrupt 2 is masked

Bit 1—Interrupt 1 Mask

0 = Interrupt 1 is unmasked

1 = Interrupt 1 is masked

Bit 0—Interrupt 0 Mask

0 = Interrupt 0 is unmasked

1 = Interrupt 0 is masked

OCW2 sets/controls the priority Rotation and the EOI modes (and combinations of the two).

For the Master PIC, OCW2 is addressed by any write cycle to an even address between 0020h-003Fh (i.e., 0020h, 0022h, 0024h ... 003Eh) when data bits 3 and 4 are set to 0. For the Slave PIC, OCW1 is addressed by any write cycle to an even address between 00A0h-00BFh (i.e., 00A0h, 00A2h, 00A4h ... 00AEh) when data bits 3 and 4 are set to 0.

Operation Command Word 2 (OCW2)

Master PIC (Read/Write, 8-bit, even address 0020h-003Fh, Bits 4-3 = 00)

Slave PIC (Read/Write, 8-bit even address 00A0h-00BFh, Bits 4-3 = 00)

Bit 7—Rotate Bit (R)

Bit 6—Select Bit (SL)

Bit 5—End of Interrupt Bit (EOI)

These three bits control the priority Rotation and the EOI modes (and combinations of the two) as shown in the following table:

Bit 7 R	Bit 6 SL	Bit 5 EOI	Function
0	0	1	Non-Specific EOI
0	1	1	Specific EOI
1	0	1	Rotate on Non-Specific EOI
1	0	0	Rotate in Automatic EOI mode (Set)
0	0	0	Rotate in Automatic EOI mode (Clear)
1	1	1	Rotate on Specific EOI Command (per L0,L1,L2)
1	1	0	Set Priority Command (per L0,L1,L2)
0	1	0	No Operation

Bits 4-3—Must both be 0, otherwise the PIC will not recognize this access as a OCW2.

Bits 2-0—Level 2-0 (L2, L1, L0)

These bits determine the interrupt level which the Rotate on Specific Command or Set Priority Command is addressed to. If any of the other options are selected by Bits 7-5, these three bits are don't cares.

OCW3 controls the special mask mode and status read pointer. It also effects the POLL command and provides a mechanism to read the In Service Register (ISR) or the Interrupt Request Register (IRR).

For the Master PIC, OCW3 is addressed by any write cycle to an even address between 0020h-003Fh (i.e., 0020h, 0022h, 0024h ... 003Eh) when data bit 3 = 1 and bit 4 = 0. For the Slave PIC, OCW1 is addressed by any write cycle to an even address between 00A0h-00BFh (i.e., 00A0h, 00A2h, 00A4h ... 00AEh) when data bit 3 = 1 and bit 4 = 0.

Operation Command Word 3 (OCW3)

Master PIC (Read/Write, 8-bit, even address 0020h-003Fh, Bits 4-3 = 01)

Slave PIC (Read/Write, 8-bit, even address 00A0h-00BFh, Bits 4-3 = 01)

Bit 7—Reserved:0

Bit 6—Enable Special Mask Mode (ESMM)

0 = Prevents the changing of Bit 5, SMM

1 = Always the writing of Bit 5, SMM

Bit 5—Special Mask Mode (SMM)

(Refer to section 10.3.5.3 for more information about Normal and Special Mask Modes)

0 = Enable Normal Mask Mode

1 = Enable Special Mask Mode

Bit 4—Must be set to 0, otherwise the PIC will not recognize this access as a OCW3.

Bit 3—Must be set to 1, otherwise the PIC will not recognize this access as a OCW3.

Bit 2—Poll Command (P)

The POLL command is used in cases where the INTA sequence is not useable or is not practical. The POLL command, followed by a Read Poll command, is similar to the INTA sequence. The POLL command is effected by issuing an OCW3 with this bit set to 1. The following Read Command returns a data byte with the following definition:

Bit 7: 1 = active Interrupt; 0 = no Interrupt pending

Bits 6-3: zero

Bits 2-0: decode of active (highest priority) requesting interrupt ID (0-7)

0 = No Poll Command Requested

1 = POLL Command

The read cycle following a POLL command is a "Read Poll" regardless of the setting of the Bits 1-0 (RR, R/S).

Bit 1—Register Read (RR)

When this bit is set to 0, Bit 0 (R/S) is a "don't care" and no ISR or IRR read command will be generated. When this bit is set to 1, the R/S bit will determine if the ISR or IRR is read by the next read cycle to the PIC.

0 = Disable ISR/IRR read commands

1 = Enable ISR/IRR read commands

Bit 0—Request/Service (R/S)

When this bit is set to 0 and bit 1 (RR) is set to 1, the next read cycle to the PIC will read the Interrupt Request Register (IRR). When this bit is set to 1 and bit 1 (RR) is set to 1, the next read cycle to the PIC will read the In Service Register (ISR).

When RR = 0, this bit is a "don't care" and ISR and IRR will not be read.

0 = The next read cycle to the PIC will access IRR, if RR = 1.

1 = The next read cycle to the PIC will access ISR, if RR = 1.

If the POLL bit is set, it will take precedence over the RR bit and this bit. The next read cycle will be a Read POLL cycle, not a read ISR or IRR cycle.

10.3.2 General Operation

In the most basic mode of operation, the PIC monitors the multiple interrupt requests coming from other system components (generally, I/O elements) and generates the single Interrupt Request (INTR) signal to the CPU if any of the incoming requests have a higher priority than the level (if any) currently being serviced.

The CPU responds to INT with an Interrupt Acknowledge cycle or POLL sequence; the PIC responds with the vector address of the requesting level with the highest priority.

10.3.3 Interrupt Sequence

The most powerful features of the PIC is its programmability. The interrupt addressing capability enables direct vectoring of software to a specific routine for each interrupt level.

The normal sequence of an interrupt is as follows:

1) One or more Interrupt Requests (IR) inputs (IR0-IR15) become active (high) to set the corresponding bit in the Interrupt Request Register (IRR).

2) The PIC evaluates the request(s) and generates the INTR signal to the 80286 if any of them are of higher priority than the level (if any) currently being serviced.
3) The CPU responds to the INTR with an Interrupt Acknowledge cycle.
4) The PIC responds to the first Interrupt Acknowledge cycle by: first, freezing its priority resolving logic, then setting the appropriate bit in the ISR (In-Service Register) and resetting the corresponding bit in the IRR (Interrupt Request Register) for the requesting level of highest priority.
5) During the first Interrupt Acknowledge cycle, the PIC does not drive any data onto the Data Bus. Then the 80286 will initiate a second Interrupt Acknowledge cycle.
6) During this second and final Interrupt Acknowledge cycle, the PIC releases an 8-bit address pointer onto the Data Bus; the 8-bit value is the vector address of the interrupt service routine and is read by the 80286.
7) The active interrupt level remains "In-Service" until its respective ISR bit is cleared by an EOI (End of Interrupt). If the PIC is programmed for AEOI (Automatic EOI), then ISR is cleared on the trailing-edge of the final Interrupt Acknowledge cycle; otherwise, the ISR remains set until an EOI instruction is issued by the software to end the interrupt sequence.

If no interrupt request (IRQ) is still active during the first command T-state of the first Interrupt Acknowledge cycle (i.e., the request was too short), then the PIC responds as if interrupt level 7 was active (i.e., interrupt 7's vector address will be driven onto the low byte of the local data bus during the second Interrupt Acknowledge cycle), however level 7 does not become "In-Service" (i.e., ISR7 is not set).

10.3.4 End-of-Interrupt (EOI) Modes

An "In-Service" interrupt level remains so until its respective ISR bit is reset. Typically, this occurs at the exit of the respective interrupt handler software routine. The active ISR is cleared by some form of EOI command, depending on the programmed EOI mode.

10.3.4.1 Normal EOI Mode

There are two basic forms of EOI; the Specific EOI and the Non-Specific EOI.

The Specific EOI command is a form of OCW2 (Operation Command Word 2), it clears the ISR of the specific level addressed by the three lower bits of the OCW2. The Specific EOI must be used in cases where the PIC has been programmed for a mode of operation where a fully nested priority structure does not exist (see Priority Nesting section 10.3.5).

The Non-Specific EOI command is another form of OCW2; it clears the ISR of the level with the highest priority that is "In-Service". The Non-Specific EOI may be used in cases where the PIC has been programmed for a mode of operation where a fully nested priority structure exists (see Priority Nesting section 10.3.5).

10.3.4.2 Automatic EOI Mode

If the PIC is programmed for Automatic EOI operation (Bit 1 (AEOI) of ICW4 = 1), a Non-Specific EOI is automatically performed at the trailing-edge of the final INTA of the INTA sequence.

The AEOI mode may be used by the PIC programmed as either Master or Slave.

10.3.5 Priority Nesting

10.3.5.1 Normal Fully Nested Mode

The PIC is in normal fully nested mode following initialization (unless otherwise programmed by SFNM bit of ICW4).

When an interrupt is acknowledged, the vector address of the highest priority request is driven onto the data bus and its corresponding ISR bit is set to 1. The ISR bit remains set until cleared by an EOI (normally immediately before the software exits the interrupt service subroutine). While the level is "In-Service " (ISR bit set to 1), all unmasked requests to this level or levels of lower-priority are inhibited; unmasked requests to levels of higher-priority are acknowledged.

10.3.5.2 Special Fully Nested Mode

This mode is used to enable a master PIC to extend its fully nested priority structure to include that of its slaves. This mode is similar to the normal nested mode; the following are exception:

1) Master PIC interrupt request levels designated as "slave" levels (by corresponding bits of ICW3) are allowed to be "re-entered"; i.e., when such a level is "In-Service", a subsequent IRQ (generated by a higher-priority interrupt request active at the slave) is not "locked-out" by the master (as is the case for normal nested mode).
2) When exiting the interrupt service subroutine, the software must check to see if the request being serviced is the only one from the slave. This is done by issuing a non-specific EOI to the slave, then reading its ISR; if zero, a non-specific EOI may also be sent to the master.

10.3.5.3 Special Mask Mode

This mode is useful in the case where an application may require a certain interrupt routine to dynamically alter the priority structure during its execution.

In the Special Mask Mode, when a mask bit is set, further requests to the corresponding level are inhibited, however, requests to all other unmasked levels (higher and lower) are enabled. Such a disabled level is returned to normal operation by clearing its mask bit.

Note: In this mode, a masked ISR bit is not cleared by a non-specific EOI.

10.3.6 Priority

10.3.6.1 Fixed Priority

By default the interrupt levels (0–7) are assigned an ordered priority of 0–7 (0 = highest, 7 = lowest). The priority is ordered by setting a specific level as the lowest level; the other seven levels assume the next seven higher priorities in ascending order (level 7 wraps to level 0). Following initialization, by default, the lowest priority (7) is assigned to interrupt level 7. The priority assignment may be changed at any time during operation via OCW2.

10.3.6.2 Rotating Priority

A specific rotation may be effected by changing the priority assignment during normal operation via OCW2 (Set Priority Command or Rotate on specific EOI).

Automatic Rotation may be effected via OCW2 (Rotate on non-specific EOI or setting the "Rotate in Automatic EOI" bit). During automatic rotation, the level being serviced (the highest priority by definition) is re-assigned the lowest priority at EOI. In this way no single interrupt level could dominate the interrupt requests and all interrupt levels have equal access to service.

10.3.7 POLL Command

The POLL command is used in cases where the INTA sequence is not useable or is not practical. The POLL command, followed by a Read Poll command, is similar to the INTA sequence. The POLL command is effected by issuing an OCW3 with this bit set to 1. The following Read Command returns a data byte with the following definition:

Bit 7: 1 = active Interrupt; 0 = no Interrupt pending

Bits 6–3: zero

Bits 2–0: decode of active (highest priority) requesting interrupt ID (0–7)

The read cycle following a POLL command is a "Read Poll" regardless of the setting of the Bits 1–0 (RR, R/S).

10.3.8 Reading Status

The Mask Register may be read via OCW1 (A0 = 1).

Either IRR or ISR may be read via OCW3 (A0 = 0). If R/S = 1, ISR is read. If R/S = 0, IRR is read.

If the POLL command has been issued, the next Read is a "Read Poll", regardless of A0 or the current setting of R/S.

10.3.9 Edge and Level Triggered Modes

Programmed by ICW1 bit 3 (LTIM).

If LTIM = 0, the IRQ inputs are edge-sensitive; i.e., an interrupt request is generated only by a low-to-high transition of IRQ. The IRQ must return low, then high to generate another request.

If LTIM = 1, the IRQ inputs are level-sensitive; i.e., requests are generated as long as IRQ remains high.

In either mode, the IRQ must remain high until after the leading-edge of the first INTA or POLL command. Otherwise, IRQ7 will be assumed by the PIC logic.

11.0 EMS

11.1 EXPANDED MEMORY SYSTEM (EMS)

The PC87120 SuperAT supports Lotus, Intel, Microsoft (LIM) EMS 3.2, EEMS 3.2, and EMS 4.0. It supports a maximum memory size of 8 MB and has a 16 KB page memory size.

The Expanded Memory System (EMS) provides a means for the operating system to access memory above 1 MB (often referred to as the "1 MB DOS limit"). This is done by mapping areas in memory that the operating system cannot access (above 1 MB) to addresses which can be accessed (below 1 MB) as shown in *Figure 11-1*.

The memory addresses to be remapped by EMS are called page frames. The two memory address ranges which contain valid EMS page frames are 40000h to BFFFFh and C0000h to EFFFFh. When EMS is enabled, the PC87120 will generate the mapped memory address for the expanded memory. The memory address generated by the PC87120 is programmable. An EMS driver is used to program and enable the PC87120's EMS system.

There are a total of 72 Page Registers used in the PC87120 for mapping the EMS memory. The 72 Page Registers are divided into sets (SET 0 and SET 1), each set with 36 Page registers. The Page Registers for the page frames within memory addresses C0000h to EFFFFh are accessed differently than the Page Registers for the page frames within memory addresses 40000h to BFFFFh.

The Page Registers corresponding to the memory addresses C0000h to EFFFFh have two IO ports that are used to read and write data to each page register. Since there are only 8 of these page registers, only 16 ports are needed to access all the page registers. On the other hand, there are 64 Page Registers for the memory addresses 40000h to BFFFFh. It would be cumbersome to use 128 separate IO ports. Therefore, one IO port is used as an index to the 64 page registers (page indices), and a second IO port is used to transfer the data to or from the Page Register.

11.2 PROGRAMMING THE EMS

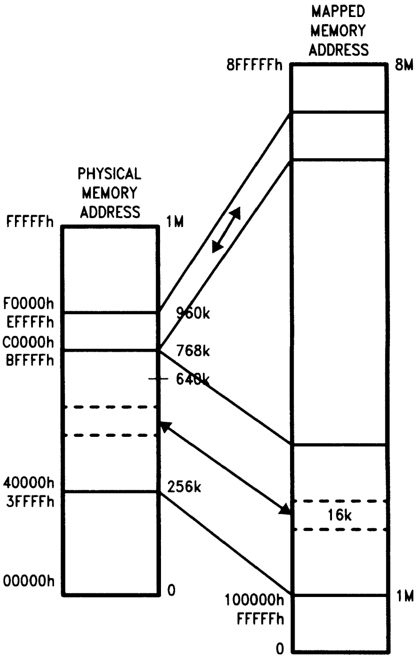
Many of the IO port addresses used for programming the EMS registers have a notation similar to 02xA. The "x" in the IO port address is determined by programming.

11.2.1 Programming EMS for Addresses 40000h to BFFFFh

The registers used in programming EMS to map memory in the address range 40000h to BFFFFh are shown in Table 11-1.

TABLE 11-1. Registers Used to Program EMS Addresses 40000h to BFFFFh

IO Port	Register
FC86	DRAM Enable Register
FC88	EMS Control Register
02xA	Page Frame Base Address Register
12xA	Page Index for Page Registers
12xB	Page Register Data



TL/F/11050-63

FIGURE 11-1. EMS Memory Mapping

DRAM Enable Register

(8-Bit Read/Write —FC86)

Default = 11110000

Bits 7-0—DRAM Enable Bits

All eight of these bits enable or disable 64 KB blocks of motherboard DRAM memory between 040000h and 0BFFFFh. These 64 KB blocks must be disabled when EMS expanded memory is mapped into these memory locations. This will prevent conflicts between the expanded memory and the coinciding DRAM memory normally at those physical addresses.

Bit	Bit Value	Function
7	0	Enable RAM B0000h—BFFFFh
	1	Disable RAM B0000h—BFFFFh (default)
6	0	Enable RAM A0000h—AFFFFh
	1	Disable RAM A0000h—AFFFFh (default)
5	0	Enable RAM 90000h—9FFFFh
	1	Disable RAM 90000h—9FFFFh (default)
4	0	Enable RAM 80000h—8FFFFh
	1	Disable RAM 80000h—8FFFFh (default)
3	0	Enable RAM 70000h—7FFFFh (default)
	1	Disable RAM 70000h—7FFFFh
2	0	Enable RAM 60000h—6FFFFh (default)
	1	Disable RAM 60000h—6FFFFh
1	0	Enable RAM 50000h—5FFFFh (default)
	1	Disable RAM 50000h—5FFFFh
0	0	Enable RAM 40000h—4FFFFh (default)
	1	Disable RAM 40000h—4FFFFh

EMS Control Register

(8-Bit Read/Write—FC88h)

Default = X0000000

Bits 7—Reserved

Bit 6—SPEEDSEL Level/Edge

This bit determines if the SPEEDSEL input is level or edge sensitive. When this bit is set to zero, The SPEEDSEL input will be level sensitive. When this bit is set to one the SPEEDSEL input will be edge sensitive.

0 = SPEEDSEL input is Level sensitive
1 = SPEEDSEL input is Edge sensitive

Refer to section 5.1.1 for more information regarding this bit.

Bit 5—EMS Set Select

This bit determines which EMS register set is in use. Toggling this bit allows one to switch from one set of EMS memory to the other set. This provides efficient support for task switching which requires different EMS memory ranges. Each set of EMS registers may be loaded at the beginning of each task. Then when tasks are switched, the appropriate EMS memory can be enabled by toggling this bit, instead of being forced to reload all of the EMS registers every time tasks are switched.

0 = Set 0 in Use (default)
1 = Set 1 in Use

Bits 4-3—EMS IO Port Address Select

These two bits provide a way to remap the IO ports through which the EMS registers may be programmed. The X in the following table corresponds to bits 7-4 of the IO address where the EMS registers are mapped.

Bit 4	Bit 3	X
0	0	0 (default)
0	1	1
1	1	2
1	1	3

Bits 2-0—On-Board EMS Starting Address

These three bits determine the starting address for all motherboard EMS memory. All motherboard memory below 1 MB is considered real mode memory. All DRAM between 1 MB and the memory selected by these bits will be treated as extended memory. All memory above the indicated EMS starting address will be used as expanded memory.

Bit 2	Bit 1	Bit 0	Starting Address
0	0	0	1 MB (default
0	0	1	2 MB
0	1	0	4 MB
0	1	1	6 MB
1	0	0	8 MB

Page Frame Base Address Register

(8-Bit, IO Port 02xA)

The only bit of significance in this register when programming expanded memory in addresses 40000h through BFFFFh, is bit 7. This bit will enable or disable all EMS IO ports. The remaining bits in this register apply to programming expanded memory for addresses C0000h through EFFFFh. These bits are covered in section 11.2.2.

Bit 7—EMS IO Port Enable

0 = Disable all of the EMS IO ports
1 = Enable all of the EMS IO ports

Bits 6-0—SRAM Bits

These bits apply only to the Registers used to program expanded memory in the ranges C0000h-EFFFFh (see section 11.2.2)

Page Index Register

(8-Bit, IO Port 12xA)

This register port is a pointer to the 64 Page Registers for memory addresses 40000h to BFFFFh. When an address that is less than the 1 MB limit and is tagged as an EMS page frame (through the DRAM Enable Register) is accessed, a "page index" will be calculated as shown in Table 11-2. This index points to the correct Page Register mapped to the page frame accessed.

In programming, each of these page frames are known and therefore, the corresponding page indices are known. Each page frame has a unique page index associated with it.

For each page index (at 12xA), there is corresponding page register (at 12xB) which holds the address of the page to be mapped into the page frame.

The Page Register is a 16-bit register, but the IO port (12xB) is only an 8-bit port. Therefore, the Page Register must be written using two writes. Bit 0 of the Page Index Register selects if the high byte or low byte of the page register will be read or written to. This bit will toggle between 0 and 1, selecting pairs of indices.

Bit 7—Don't Care

Bit 6—EMS Set Select Bit

This bit determines which EMS register set is in use. Toggling this bit allows one to switch from one set of EMS memory to the other set. This provides efficient support for task switching which requires different EMS memory ranges. Each set of EMS registers may be loaded at the beginning of each task. Then when tasks are switched, the appropriate EMS memory can be enabled by toggling this bit, instead of being forced to reload all of the EMS registers every time tasks are switched.

Selects the set (0 or 1) which will be programmed on the next write to 12xB.

0 = Set 0 in Use (default)
1 = Set 1 in Use

Bits 5-1—Page Index Bits

Shown in Table 11-1. These bits determine the page index for the page register during the configuration of EMS.

Bit 0—Selects which data byte (high or low) to latch on the next access to the Page Register.

0 = Low Byte
1 = High Byte

The remaining bits in the page register, bits 12xA[D6,D7] are unused and will always be 0.

When all the Page Register Index Bits are put together, the page index results. The following table shows the indices for each of the 16 KB page frames in the address range 40000h to BFFFFh.

TABLE 11-2. Page Frames and Page Indices

Page Frame	Page Index	
	SET 0	SET 1
40000h	00/01	40/41
44000	02/03	42/43
48000	04/05	44/45
4C000	06/07	46/47
50000h	08/09	48/49
54000	0A/0B	4A/4B
58000	0C/0D	4C/4D
5C000	0E/0F	4E/4F
60000h	10/11	50/51
64000	12/13	52/53
68000	14/15	54/55
6C000	16/17	56/57
70000h	18/19	58/59
74000	1A/1B	5A/5B
78000	1C/1D	5C/5D
7C000	1E/1F	5E/5F
80000h	20/21	60/61
84000	22/23	62/63
88000	24/25	64/65
8C000	26/27	66/67
90000h	28/29	68/69
94000	2A/2B	6A/6B
98000	2C/2D	6C/6D
9C000	2E/2F	6E/6F
A0000h	30/31	70/71
A4000	32/33	72/73
A8000	34/35	74/75
AC000	36/37	76/77
B0000h	38/39	78/79
B4000	3A/3B	7A/7B
B8000	3C/3D	7C/7D
BC000	3E/3F	7E/7F

Page Registers
(8-Bit, IO Port 12xB)

This register acts as a "window" into the 64 Page Registers for memory addresses 40000h to BFFFFh. Through this window, the Page Register information can be seen. Which page register can be seen is controlled by the page index selected through the Page Index Register. The bit definitions in the Page Register data are different for high and low bytes. *Figure 11-2* illustrates this.

The Page Register contains a coded version of the address that will be input to the PC87120's memory controller to access the EMS memory.

Page Register Data—Low Byte

Bit 7—Enable Mapping Bit

0 = Disable EMS Mapping for the page frame
1 = Enable EMS Mapping for the page frame

Bits 6-0—Page Number Bits (PN)

These bits map into the lower six bits of the Page Number (PN) represented by PN [6-0].

Page Register Data—High Byte

Bits 7-4—Don't Care
Bits 3-0—Page Number Bits

These bits map into the bits 10-8 of Page Numbers (PN) represented by PN[10], PN[9], PN[8].

11.2.2 Programming EMS for Addresses C0000h to EFFFFh

The registers used in programming EMS to memory in the address range C0000h to EFFFFh are shown in Table 11-3.

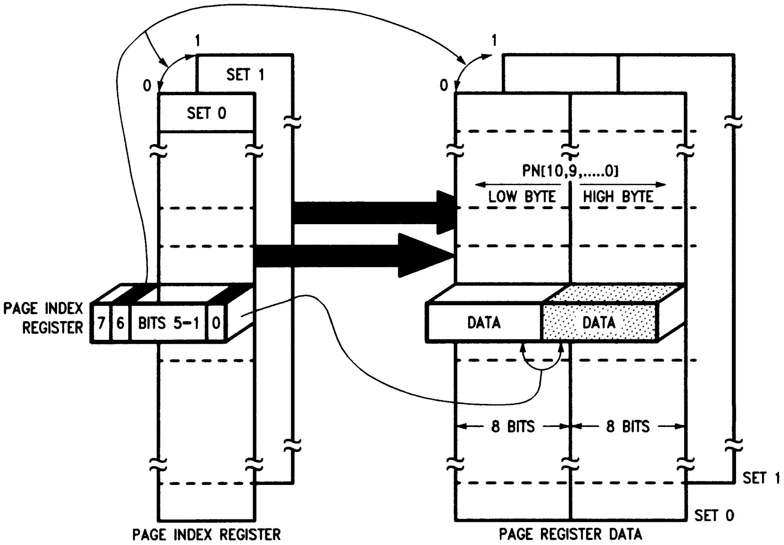
TABLE 11-3. Registers Used to Program EMS Addresses C0000h-EFFFFh

IO Address	Register
FC88	EMS Control Register
02xA	Page Frame Base Address
02x8-02x9	Page Register 0 for SET 0
42x8-42x9	Page Register 1 for SET 0
82x8-82x9	Page Register 2 for SET 0
C2x8-C2x9	Page Register 3 for SET 0
12x8-12x9	Page Register 0 for SET 1
52x8-52x9	Page Register 1 for SET 1
92x8-92x9	Page Register 2 for SET 1
D2x8-D2x9	Page Register 3 for SET 1

EMS Control Register
(8-Bit Read/Write—FC88h)
Default = X0000000

Bit 7—Reserved
Bit 6—SPEEDSEL Level/Edge

This bit determines if the SPEEDSEL input is level or edge sensitive. When this bit is set to zero, the SPEEDSEL input will be level sensitive. When this bit is set to one the SPEEDSEL input will be edge sensitive.



TL/F/11050-64

FIGURE 11-2. Page Indice/Page Registers

11.0 EMS (Continued)

0 = SPEEDSEL input is Level sensitive
1 = SPEEDSEL input is Edge sensitive

Refer to section 5.1.1 for more information regarding this bit.

Bit 5—EMS Set Select

This bit determines which EMS register set is in use. Toggling this bit allows one to switch from one set of EMS memory to the other set. This provides efficient support for task switching which requires different EMS memory ranges. Each set of EMS registers may be loaded at the beginning of each task. Then when tasks are switched, the appropriate EMS memory can be enabled by toggling this bit, instead of being forced to reload all of the EMS registers every time tasks are switched.

0 = Set 0 in Use (default)
1 = Set 1 in Use

Bits 4, 3—EMS IO Port Address Select

These two bits provide a way to remap the IO ports through which the EMS registers may be programmed. The X in the following table corresponds to bits 7-4 of the IO address where the EMS registers are mapped.

Bit 4	Bit 3	X
0	0	0 (default)
0	1	1
1	1	2
1	1	3

Bits 2-0—On-Board EMS Starting Address

These three bits determine the starting address for all motherboard EMS memory. All motherboard memory below 1 MB is considered real mode memory. All DRAM between 1 MB and the memory selected by these bits will be treated as extended memory. All memory above the indicated EMS starting address will be used as expanded memory.

Bit 2	Bit 1	Bit 0	Starting Address
0	0	0	1 MB (default)
0	0	1	2 MB
0	1	0	4 MB
0	1	1	6 MB
1	0	0	8 MB

Page Frame Base Address Register
(8-Bit, IO Port 02xA)

Bit 7—EMS IO Port Enable

0 = Disable all of the EMS IO ports
0 = Enable all of the EMS IO ports

Bit 6—EMS Wait State Bit

This bit allows the addition of 1 extra wait state when accessing EMS memory in the address range C0000h through EFFFFh.

0 = 0 EMS memory wait states
0 = 1 EMS memory wait state

Bits 5, 4—Don't Care

Bits 3-0—Page Bits

These bits determine the starting page frame address for EMS memory in the address range C0000h through EFFFFh. These bits select four 16 KB pages that the EMS memory will be mapped into as shown. Using this method, there is no need for page indices.

Bit 3	Bit 2	Bit 1	Bit 0	Page Select			
				"0"	"1"	"2"	"3"
0	0	0	0	C0000	C4000	C8000	CC000
0	0	0	1	C4000	C8000	CC000	D0000
0	0	1	0	C8000	CC000	D0000	D4000
0	0	1	1	CC000	D0000	D4000	D8000
0	1	0	0	D0000	D4000	D8000	DC000
0	0	0	1	D4000	D8000	DC000	E0000
0	0	1	0	D8000	DC000	E0000	E4000
0	0	1	1	DC000	E0000	E4000	E8000
0	1	0	0	E0000	E4000	E8000	EC000

Page Registers
(8-Bit)

Low Byte	High Byte	Register
02x8	02x9	Page Register 0 for SET 0
42x8	42x9	Page Register 1 for SET 0
82x8	82x9	Page Register 2 for SET 0
C2x8	C2x9	Page Register 3 for SET 0
12x8	12x9	Page Register 0 for SET 1
52x8	52x9	Page Register 1 for SET 1
92x8	92x9	Page Register 2 for SET 1
D2x8	D2x9	Page Register 3 for SET 1

These registers hold part of the memory address that will be mapped into the page frame. Each page register corresponds to one of the 16 KB page frames programmed in Bits 3–0 of the Page Frame Base Address Register. Page Register 0 corresponds to Page Frame Select "0" (see *Figure 11-3*), and Page Registers 1, 2, and 3 correspond to Page Frame Select "1", "2", and "3" respectively.

The registers come in pairs to allow the 16 bits of data to be entered in.

Page Register Data—Low Byte

Bit 7—Enable Mapping Bit

0 = Disable EMS Mapping for the page
1 = Enable EMS Mapping for the page

Bits 6–0—Page Number Bits

These bits map into the lower six bits of the Page Number (PN) represented by PN [6–0]

Page Register Data—High Byte

Bits 7-4—Don't Care

Bits 3-0—Page Number Bits

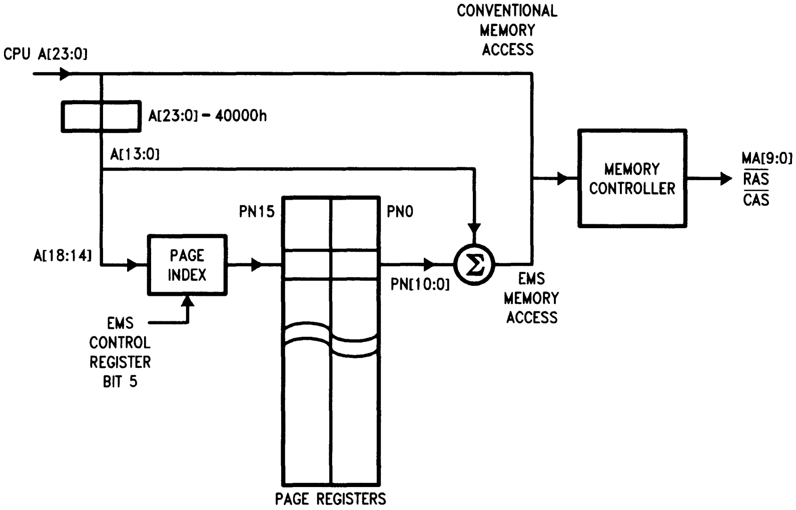
These bits map into the bits 10–8 of Page Number (PN) represented by PN[10], PN[9], PN[8].

11.3 EMS ADDRESS FORMATION

The programmed Page Register Data bits 10–8 and 6–0 are output as PN[10:8] and PN[6:0] and merged together with CPU address bits A[13:0] and sent to the memory controller. Just what position these bits will be in will depend on the type of DRAMs being used. Table 11-4 shows the possible positions.

11.4 EMS MEMORY SUPPORT

All EMS memory accesses to the address range C0000h to EFFFFh will incur an additional wait state. *Figure 11-4* shows an EMS Read cycle followed by a non-EMS Write cycle.

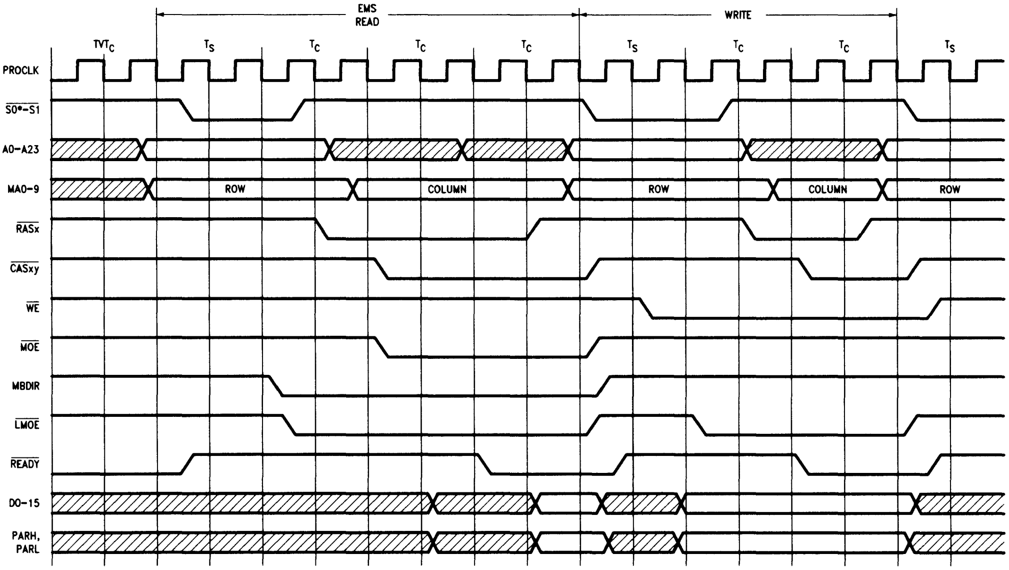


TL/F/11050-65

FIGURE 11-3. EMS Memory Address Generation

TABLE 11-4. EMS Multiplexed Address Generation

Memory	Type	Multiplexed Address Bits									
		MA0	MA1	MA2	MA3	MA4	MA5	MA6	MA7	MA8	MA9
1	$\overline{RAS}$	A10	A11	A12	A13	PN0	PN1	PN2	PN3	PN4	X
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	A9	X
2	$\overline{RAS}$	PN5	A11	A12	A13	PN0	PN1	PN2	PN3	PN4	PN6
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
3	$\overline{RAS}$	PN5	A11	A12	A13	PN0	PN1	PN2	PN3	PN4	X
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	A9	X
4	$\overline{RAS}$	PN5	PN8	A12	A13	PN0	PN1	PN2	PN3	PN4	PN6
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
5	$\overline{RAS}$	PN5	A11	A12	A13	PN0	PN1	PN2	PN3	PN4	X
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	A9	X
6	$\overline{RAS}$	PN5	A11	A12	A13	PN0	PN1	PN2	PN3	PN4	X
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	A9	X
	$\overline{RAS}$	PN5	PN8	A12	A13	PN0	PN1	PN2	PN3	PN4	PN6
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
7	$\overline{RAS}$	PN5	PN8	A12	A13	PN0	PN1	PN2	PN3	PN4	PN6
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
8	$\overline{RAS}$	A10	A11	A12	A13	PN0	PN1	PN2	PN3	PN4	X
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	A9	X
	$\overline{RAS}$	A10	A11	A12	A13	PN0	PN1	PN2	PN3	X	X
	$\overline{CAS}$	A1	A2	A3	A4	A5	A6	A7	A8	X	X



TL/F/11050-66

FIGURE 11-4. EMS Read Followed by Non-EMS Write

12.0 Sleep, Suspend/Resume Mode

The PC87120 supports a Sleep Mode operation to the power consumption of the system. During Sleep Mode the PC87120 holds all output signals in their inactive states. This minimizes the system's current requirements.

A 80286 HALT command will place the PC87120 into sleep mode when the Sleep Mode Enable bit (Bit 0 of the Sleep Mode/DMA Control Register, located at IO address FC85h) is set to 1. If the Sleep Mode Enable bit is set to 0 and the 80286 issues a HALT command, the PC87120 locks the system up and forces a power-on reset.

In Sleep Mode, the clock outputs of the PC87120 will continue to oscillate when the Clocks in Sleep Mode bit (Bit 1 of the Sleep Mode/DMA Control Register) is set to 0 (the default). While the 80286 remains in the HALTed state, the only system activity is when the PC87120 refreshes the motherboard memory. The system remains in sleep mode during refresh cycles.

If the Clocks in Sleep Mode bit is set to 1, the PC87120 will turn off all of its clock outputs, driving them high. When the PC87120 exits Sleep Mode, it turns the clock outputs back on, allowing them to oscillate again. The PC87120 will continue to refresh the memory even with the clocks turned off.

The PC87120 exits Sleep Mode any time it decodes an interrupt. After the system clocks are turned back on (if they were off), the interrupt request is sent to the 80286. Once the 80286 receives this request, it will terminate its HALT state and return to its normal operating mode. To return to Sleep Mode the 80286 must issue another HALT cycle.

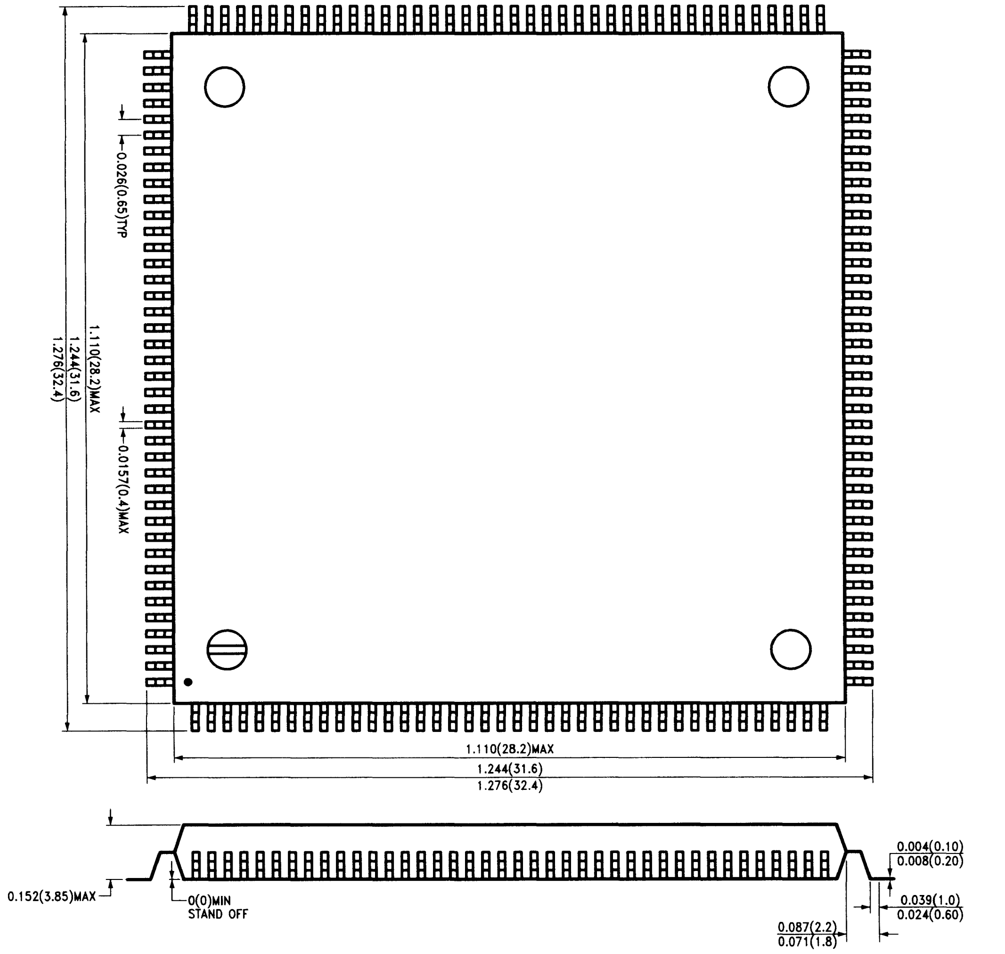
To use Sleep Mode, the software running in the system must be specifically written to take advantage of this feature. Otherwise, the 80286 will never issue the HALT command to put the system into Sleep Mode and there will be no reduction in power consumption.

The Suspend/Resume Flag (Bit 2 of the Sleep Mode/DMA Control Register) is a single latch which may be used as a system flag. This flag does NOT control any function. It may be set to 0 or 1 by software and used by the system's BIOS to indicate any desired condition. This flag is reset to 0 during every system reset.

13.0 Device Specifications

Lit. # 112909

14.0 Package Dimensions



TL/F/11050-62

FIGURE 14-1. Package Dimensions

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: (408) 721-5000
TWX: (910) 339-9240

National Semiconductor GmbH
Industriestrasse 10
D-8080 Furstenfeldbruck
West Germany
Tel: (0-81-41) 103-0
Telex: 527-649
Fax: (08141) 103554

National Semiconductor Japan Ltd.
Sanseido Bldg. 5F
4-15 Nishi Shinjuku
Shinjuku-Ku,
Tokyo 160, Japan
Tel: 3-299-7001
FAX: 3-299-7000

National Semiconductor Hong Kong Ltd.
Suite 513, 5th Floor
Chinachem Golden Plaza,
77 Mody Road, Tsimshatsui East,
Kowloon, Hong Kong
Tel: 3-7231290
Telex: 52996 NSSEA HX
Fax: 3-3112536

National Semicondutores Do Brasil Ltda.
Av. Brig. Faria Lima, 1383
6.0 Andor-Conj. 62
01451 Sao Paulo, SP, Brasil
Tel: (55/11) 212-5066
Fax: (55/11) 211-1181 NSBR BR

National Semiconductor (Australia) PTY, Ltd.
1st Floor, 441 St. Kilda Rd.
Melbourne, 3004
Victory, Australia
Tel: (03) 267-5000
Fax: 61-3-2677458

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.