Intel386™ MicroComputer Model 302

Board Technical Reference Manual

intel

302





Intel386[™] MicroComputer Model 302 Board Technical Reference Manual

Order Number: 505067-002

Revision F



WARNING

This equipment has been tested with a Class B computing device and has been found to comply with the limits for a class B computing device, pursuant to subpart J of part 15 FCC rules. Only peripherals (computer input/output devices, terminals, printers, etc.) which comply with the class B limits may be attached to this computer product. Operation with noncompliant peripherals is likely to result in interference to radio and TV reception.

All cables used to connect to peripherals must be shielded and grounded. Operation with cables, connected to peripherals, which are not shielded and grounded may result in interference to radio and TV reception.

This equipment meets or exceeds requirements for safety in the US (UL 478 5th Edition), Canada (CSA C22.2 No. 220), and Europe (IEC 380, IEC 435, IEC 950, and VDE 0806).

This equipment has been tested for radio frequency emissions and has been verified to meet VDE 0871 Class B.

This digital apparatus does not exceed the Class B limits for radio noise emissions set out in the radio interference regulations of the Canadian Department of Communications.

Ce dispositif digital, s'il est utilise suivant les instructions et recommandations du constructeur, ne depasse pas les limites de la Classe B pour le bruit des frequences radio, etablies par les Regles sur l'interference radio du Ministere Canadian des Communications.

RADIO FREQUENCY INTERFERENCE NOTICE

This equipment generates and uses radio frequency energy and if not installed properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type-tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 FCC rules, which are designed to provide reasonable protection against interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If the equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna.
- Relocate the system with respect to the receiver.
- Move the system away from the receiver.
- Plug the system into a different outlet so that the system and receiver are on different branch circuits.
- Move the cables connected to the system to minimize the interference.
- Tighten all screws on cables and the system housing.
- Install blank panels, originally supplied with the system, in all unused card slots.

If necessary the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet, prepared by the Federal Communications Commission, helpful:

"How to Identify and Resolve Radio-TV Interference Problems"

This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402. Stock No. 004-00398-5.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation Literature Sales P.O. Box 58130 Santa Clara, CA 95052-8130

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software and firmware products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication, or disclosure is subject to restrictions stated in Intel's Software License Agreement, or in the case of software delivered to the U.S. government, in accordance with the software license agreement as defined in FAR 52.227-7013.

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Intel Corporation.

Intel Corporation retains the right to make changes to these specifications at any time, without notice. Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and its affiliates, and may be used only to identify Intel products:

Above	iMDDX	iRMX	Promware
BITBUS	iMMX	iSBC	QueX
COMMputer	Inboard	iSBX	QUEST
CREDIT	Insite	iSDM	Quick-Erase
Data Pipeline	Intel®	iSXM	Quick Pulse Programming
ETOX	int _e l®	KEPROM	Ripplemode
Genius	Intel376	Library Manager	RMX/80
i486	Intel386	MAPNET	RUPI
i860	intelBOS	MCS®	Seamless
ф	Intel Certified	Megachassis	SLD
j®	Intelevision	MICROMAINFRAME	SugarCube
I ² CE	inteligent Identifier	MULTIBUS	UPI
ICE	inteligent Programming	MULTICHANNEL	VLSiCEL
iCEL	Intellec®	MULTIMODULE	376
ICEVIEW	Intellink	ONCE	386
iCS	iosp	OpenNET	386SX
iDBP	iPAT	OTP	387
iDIS	iPDS	PC BUBBLE	387SX
iLBX	iPSC®	Plug-A-Bubble	4-SITE
i _m ®	iRMK	PROMPT	486

MULTIBUS is a patented Intel bus.

IBM is a registered trademark of International Business Machines, Inc.

PC, XT, and AT are trademarks of International Business Machines, Inc.

Microsoft, MS-DOS, OS/2, and XENIX are registered trademarks of Microsoft Corporation.

Multiplan is a trademark of Microsoft Corporation.

UNIX is a trademark of AT&T.

Lotus 1-2-3 and Symphony are registered trademarks of Lotus Development Corporation.

Framework II and dBASE III PLUS are registered trademarks of Ashton-Tate.

MultiMate is a trademark of Ashton-Tate.

Centronics is a trademark of Centronics Data Computer Corporation.

Copyright © 1989,1990 Intel Corporation

REV.	REVISION HISTORY	DATE
-001	Preliminary Release	11/89
-002	Original Issue (Release 1.0)	1/90



Contents

About This Manual Purpose xxvii Library Plan..... xxviii Audience..... XXVIII Organization..... xxix How to Obtain More Information xxxi Notational Conventions. xxxi Related Publications xxxiv 1 **Board Overview** 11 Introduction..... 1-1 1.2 Overview 1-1 1.3 Feature Set Description..... 1-6 Central Processing Unit (CPU)..... 1.3.1 1-6 1.3.2 Memory 1-6 AT32 Bus..... 1.3.3 1-6 1.3.4 Intel ISA Bus 1-7 1.3.5 Programmable Interval Timer..... 1-7 1.3.6 Programmable Interrupt Controller 1-7 1.3.7 DMA Controller 1-7 1.3.8 Real-time CMOS Clock 1-7 1.3.9 Input/Output Expansion Slots..... 1-8 1.3.10 Input/Output Ports..... 1-8 1.3.11 Keyboard Interface..... 1-8 Special Board Interfaces..... 1.3.12 1-8 Firmware 1.3.13 1-9

2	Central Processing Core	
2.1	Introduction	2-1
2.2	Overview	2-1
2.3	CPU	2-3
2.3.1	Real Mode Architecture	2-3
2.3.2	Protected Mode Architecture	2-3
2.3.3	Virtual-8086 Mode	2-4
2.3.4	CPU Signals	2-4
2.3.5	Basic CPU Bus Operations	2-7
2.4	Numeric Coprocessor	2-10
2.4.1	387™ Numeric Coprocessor Data Types	2-10
2.4.2	387™ Coprocessor Programming Interface	2-11
2.4.3	Weitek 3167 Interface	2-13
2.5	Cache Memory and Tag	2-14
2.5.1	Cache Memory	2-15
2.5.2	Cache Tag Memory	2-15
2.5.3	Cache Memory Read Requests	2-15
2.5.4	Cache Memory Write Requests	2-16
2.6	Address and Data Buffers	2-16
2.7	CPU/Cache Control	2-16
3	Onboard Memory	
3.1	Introduction	3-1
3.2	Onboard Memory Overview	3-1
3.3	302 Board DRAM	3-3
3.4	Configuring AT32 Memory	3-4
3.5	DRAM Refresh	3-5
4	Intel AT32 Bus Interface	
4.1	Introduction	4-1
4.2	AT32 Bus Overview	4-1

5	Intel ISA Bus	
5.1	Introduction	5-1
5.2	Bus Agents	5-1
5.2.1	Configuring Bus Agents	5-2
5.2.2	Agent Functional Model	5-3
5.2.3	General ISA Bus Attributes	5-4
5.3	Signal Groups	5-5
5.3.1	Address Signal Group	5-6
5.3.2	Data Signal Group	5-8
5.3.3	Cycle Control Signal Group	5-9
5.3.4	Central Control Signal Group	5-11
5.3.5	Interrupt Signal Group	5-13
5.3.6	Direct Memory Access Signal Group	5-13
5.3.7	Power Group	5-14
6	8254 Programmable Interval Timer	
6.1	Introduction	6-1
6.2	Counter Description	6-2
6.2.1	Control and Status Registers	6-4
6.2.2	Counting Element (CE)	6-4
6.3	Programming the PIT	6-5
6.3.1	Control Register (043H)	6-5
	Read/Write Counter Command	6-7
	Latch-Counter Command	6-7
	Read-back Command	6-8
7	8259 Programmable Interrupt Controllers	
7.1	Introduction	7-1
7.2	Interrupt Controller Architecture	7-4
7.2.1	Interrupt Request Register (IRR) and In-service Register (ISR)	7-4
7.2.2	Interrupt Mask Register	7-4
7.2.3	Priority Resolver	7-4
7.3	Interrupts	7-5
7.3.1	Maskable Interrupts	7-5
7.3.2	Non-maskable Interrupts	7-7

7	8259 Programmable Interrupt Controllers (continued)	
7.4	Programming the PIC	7-8
7.4.1	Initialization Command Word	7-9
7.4.2	Operational Command Word	7-15
8	Direct Memory Access (DMA)	
8.1	Introduction	8-1
8.2	DMA Operation	8-4
8.2.1	Idle Cycle	8-4
8.2.2	Active Cycle	8-4
	Single-Transfer Mode	8-5
	Block-Transfer Mode	8-5
	Demand-Transfer Mode	8-5
	Cascade Mode	8-6
8.3	Transfer Types	8-6
8.4	DMA Channels	8-7
8.5	Programming The DMA Controllers	8-8
8.6	DMA Internal Registers	8-9
8.6.1	Address and Count Registers	8-10
	Current Address Register	8-11
	Current Word-Count Register	8-11
	Base Address and Base Word-Count Registers	8-11
8.6.2	Program-control Registers	8-12
8.6.3	Command Register	8-12
8.6.4	Mode Register	8-14
8.6.5	Request Register	8-15
8.6.6	Mask Register	8-15
8.6.7	Status Register	8-17
8.6.8	Temporary Register	8-19
8.6.9	Non-register Program Controls	8-19
8.7	Page Registers	8-20

9	1287 Real-time Clock (RTC)	
9.1	Introduction	9-1
9.2	RTC RAM I/O Operations	9-1
9.3	RTC Addressable Locations	9-3
9.3.1	Time, Calendar, and Alarm Bytes	9-5
9.4	Status Registers	9-7
9.4.1	Status Register A	9-7
9.4.2	Status Register B	9-8
9.4.3	Status Register C	9-10
9.4.4	Status Register D	9-11
9.5	Configuration Bytes	9-11
9.5.1	Diagnostic Status Byte	9-12
9.5.2	Shutdown Status Byte	9-13
9.5.3	Floppy Disk Drive Type Byte	9-14
9.5.4	Fixed Disk Drive Type Byte	9-15
9.5.5	Equipment Byte	9-16
9.5.6	Low and High Base Memory Bytes	9-17
9.5.7	Low and High Memory Expansion Bytes	9-17
9.5.8	Drive C Extended Byte	9-18
9.5.9	Drive D Extended Byte	9-18
9.5.10	Feature Installed Byte	9-18
9.5.11	Fixed Disk Type 48 Parameters	9-19
9.5.12	Shadow and Setup Byte	9-20
9.5.13	CMOS RAM Checksum	9-20
9.5.14	Low and High Extended Memory Byte	9-21
9.5.15	Date Century Byte	9-21
9.5.16	Setup Information Byte	9-22
9.5.17	CPU Speed Byte	9-22
9.5.18	Fixed Disk Type 49 Parameters	9-23
10	Communication Ports	
10.1	Introduction	10-1
10.2	Serial Communication Ports	10-1
10.2.1	CPU Interfacing	10-2
10.2.2	Connectors and Pinouts	10-2
10.3	Parallel Printer Port	10-2
10.3.1	Programming	10-3
10.3.2	Connector and Pinouts	10-5

Keyboard Controller	
Introduction	11-1
Keyboard Controller System Interface	11-1
Status Register	11-3
Output Buffer	11-4
Input Buffer	11-4
Input and Output Ports	11-5
Controller Commands	11-6
Keyboard Interface	11-7
Keyboard Data Stream	11-8
Receiving Data from the Keyboard	11-9
Sending Data to the Keyboard	11-11
Keyclick Volume	11-15
302 Board Special Interfaces	
Introduction	12-1
Keylock Interface	12-1
Speaker Interface	12-2
Reset Interface	12-3
Power-on Self Test and Setup	
Introduction	13-1
Power-on Self Test	13-1
Setup Program Overview	13-2
Running The SETUP Program	13-3
Moving Through Setup Screen 1	13-5
Setting System Time	13-5
Setting System Date	13-5
Selecting Floppy disk Drive Types	13-6
Setting Fixed disk Drive Types	13-6
Setting Base Memory	13-7
Setting Extended Memory	13-7
Setting Video Display Controller Type	13-8
Setting Keyboard Availability	13-9
Setting CPU Speed	13-9
Numeric Coprocessor	13-10
	Introduction Keyboard Controller System Interface Status Register Output Buffer Input Buffer Input and Output Ports Controller Commands Keyboard Interface Keyboard Data Stream Receiving Data from the Keyboard Sending Data to the Keyboard Sending Data to the Keyboard Keyclick Volume 302 Board Special Interfaces Introduction Keylock Interface Speaker Interface Reset Interface Reset Interface Power-on Self Test and Setup Introduction Power-on Self Test Setup Program Overview Running The SETUP Program Moving Through Setup Screen 1 Setting System Time Setting System Date Selecting Floppy disk Drive Types Setting Base Memory Setting Base Memory Setting Video Display Controller Type Setting Keyboard Availability

13	Power-on Self Test and Setup (continued)	
13.4.2	Moving Through Setup Screen 2	13-10
	Shadowing System BIOS and Video BIOS	13-11
	Cache Memory	13-11
	AT32 I/O	13-12
	Above 16M	13-12
	512-640 K	13-12
	Speaker	13-13
	Enter SETUP	
13.5	Exiting SETUP	13-13
13.6	Error Messages	13-14
A	Specifications	
A.1	302 Board Specifications	A-1
A.2	302 Board Physical Characteristics	A-2
A.3	302 Board Power Requirements	A-3
В	System BIOS Specifications	
B.1	Introduction	B-1
B.2	Non-maskable Interrupt (INT 02H)	B-1
B.3	Print Screen (INT 05H)	B-2
B.4	System Timer H/W Interrupt (INT 08H)	B-2
B.5	Keyboard H/W Interrupt (INT 09H)	B-3
B.6	Disk H/W Interrupt (INT 0EH)	B-3
B.7	Video I/O (INT 10H)	B-4
B.7.1	Set Display Mode	B-5
B.7.2	Set Cursor Size	B-6
B.7.3	Set Cursor Position	B-6
B.7.4	Read Cursor Position	B-7
B.7.5	Read Light Pen Position	B-7
B.7.6	Select Active Display Page	B-7
B.7.7	Scroll Window Up	B-8
B.7.8	Scroll Window Down	B-8
B.7.9	Read Attribute/Character at Current Cursor Position	B-9
B.7.10	Write Attribute/Character at Current Cursor Position	B-9
B.7.11	Write Character Only at Current Cursor Position	B-10

B System BIOS Specifications (continued)

B.7.12	Set Color Palette	B-10
B.7.13	Write Dot	B-11
B.7.14	Read Dot	B-11
B.7.15	Write Character as Teletype to Active Page	B-11
B.7.16	Get Current Video Mode	B-12
B.7.17	Write String	B-12
B.8	Equipment Determination (INT 11H)	B-13
B.9	Memory Size Determination (INT 12H)	B-14
B.10	Floppy Disk Drive I/O (INT 13H, Part 1)	B-14
B.10.1	Reset Disk Drive	B-15
B.10.2	Read Status	B-15
B.10.3	Disk Read	B-16
B.10.4	Disk Write	B-16
B.10.5	Disk Verify	B-17
B.10.6	Format Disk Track	B-17
B.10.7	Disk Drive Parameters	B-18
B.10.8	Read DASD Type	B-18
B.10.9	Disk Change Line Status	B-19
B.10.10	Set DASD Type for Format	B-19
B.10.11	Set Media Type for Format	B-20
B.11	Fixed Disk I/O (INT 13H, Part 2)	B-21
B.11.1	Disk Reset	B-22
B.11.2	Read Status	B-23
B.11.3	Read Disk	B-23
B.11.4	Write Disk	B-23
B.11.5	Verify Disk Sectors	B-24
B.11.6	Format Disk Track	B-24
B.11.7	Disk Drive Parameters	B-25
B.11.8	Initialize Disk Parameters	B-25
B.11.9	Disk Read Long	B-25
B.11.10	Disk Write Long	B-26
B.11.11	Disk Seek	B-26
B.11.12	Disk Alternate Reset	B-27
B.11.13	Disk Ready Test	B-27
B.11.14	Disk Recalibrate	B-27
B.11.15	Disk Diagnostics	B-28
B.11.16	Read DASD Type	B-28

B System BIOS Specifications (continued)

B.12	RS-232 I/O (INT 14H)	B-29
B.12.1	Initialize the Communication Port	B-30
B.12.2	Send a Character	B-31
B.12.3	Receive a Character	B-32
B.12.4	Return the Communication Port Status	B-32
B.13	System Service Routines (INT 15H)	B-33
B.13.1	Keyboard Intercept	B-34
B.13.2	Device Open	B-35
B.13.3	Device Close	B-35
B.13.4	Program Termination	B-35
B.13.5	Event Wait	B-36
B.13.6	Joystick Support	B-36
B.13.7	System Request Key Pressed	B-37
B.13.8	Wait	B-37
B.13.9	Move Block	B-38
B.13.10	Extended Memory Size Determination	B-38
B.13.11	Switch CPU to Protected Mode	B-39
B.13.12	Device Busy	B-39
B.13.13	Interrupt Complete	B-40
B.13.14	Return System Configuration Parameters	B-40
B.14	Keyboard I/O (INT 16H)	B-42
B.14.1	Read Next Character	B-43
B.14.2	Read Buffer Status	B-43
B.14.3	Return Shift Status	B-44
B.14.4	Set Typematic Rate and Delay	B-45
B.14.5	Place ASCII Character/Scan Code in Keyboard Buffer	B-46
B.14.6	Extended Read Interface for the Enhanced Keyboard	B-46
B.14.7	Extended Buffer Status for the Enhanced Keyboard	B-46
B.14.8	Return the Extended Shift Status for the Enhanced Keyboard	B-47
B.15	Printer I/O (INT 17H)	B-48
B.15.1	Print Character	B-48
B.15.2	Initialize Printer Port	B-49
B.15.3	Read Printer Status	B-49
B.16	System Boot (INT 19H)	B-50

В	System BIOS Specifications (continued)	
B.17	Clock Services (INT 1AH)	B-50
B.17.1	Read the System Timer Count	B-51
B.17.2	Set the System Timer Count	B-51
B.17.3	Read the Real-time Clock Time	B-51
B.17.4	Set the Real-time Clock Time	B-52
B.17.5	Read the Real-time Clock Date	B-52
B.17.6	Set the Real-time Clock Date	B-53
B.17.7	Set the Real-time Clock Alarm	B-53
B.17.8	Reset the Real-time Clock Alarm	B-53
B.18	Real-Time Clock (INT 70H)	B-54
B.19	Coprocessor Interrupt (INT 75H)	B-54
B.20	Fixed Disk H/W Interrupt (INT 76H)	B-54
C	Jumper Settings	
C.1	Introduction	C-1
C.2	Total Onboard DRAM	C-3
C.3	Onboard ROM Size	C-3
C.4	Onboard RAM	C-4
C.5	Video Display Type	C-5
C.6	Printer Acknowledge	C-5
C.7	Parallel Printer Port Selection	C-6
C.8	Serial Communications Port Selection	C-6
D	Changing CPU Speed	
D.1	Introduction	D-1
D.2	Overview	D-1
D.3	Keyboard Method	D-2
D.4	Setup Method	D-2
D.5	Software Method	D-3

E	Messages	
E.1 E.2 E.2.1	IntroductionPOST and Boot MessagesPOST and Boot Error Messages	E-1 E-1 E-2
E.2.2	POST and Boot Information Messages	E-9
E.3	Run-Time Messages	E-11
E.4	302 Board Errors	E-12
F	Device Mapping	
F.1	Introduction	F-1
F.2	System Memory Map	F-2
F.3	I/O Addresses	F-3
F.4	Interrupt Priority Levels	F-7
F.5	DMA Channel Assignments	F-8
F.6	RtC Mapping	F-8
G	Hot Keys	
Н	Pin Assignments for Major Signals	
H.1	Introduction	H-1
H.2	CPU Pin Assignments	H-1
H.3	AT32 Bus Pin Assignments	H-3
H.4	ISA Bus Pin Assignments	H-4
H.5	Serial Communications Port Pinouts	H-6
H.6	Parallel Printer Port Pinout	H-8
H.7	Keyboard Pinout	H-9

I Installation 1.1 Introduction..... 1-1 Installing SIMMs..... 1.2 1-1 1.3 Removing SIMMS I-3 Installing a Numeric Coprocessor 1.4 1-4 Removing a Numeric Coprocessor..... 1.5 I-6

Glossary

Index



Figures

1	Board Overview	
1-1 1-2 1-2	302 Board Component Layout	1-3 1-4 1-5
2	Central Processing Core	
2-1 2-2 2-2 2-3	Central Processing Core	2-2 2-8 2-9 2-14
6	8254 Programmable Interval Timer	
6-1	Internal Block Diagram of a Counter	6-3
7	8259 Programmable Interrupt Controllers	
7-1 7-2 7-3 7-4 7-5 7-6 7-7	PIC Block Diagram	7-2 7-3 7-9 7-10 7-11 7-13 7-14 7-16
7-9	Command Word Format (OCW3)	7-17

8	Direct Memory Access (DMA)	
8-1 8-2	DMA Controller Block Diagram DMA Logic	8-2 8-3
9	1287 Real-time Clock (RTC)	
9-1	RTC Memory Map	9-2
11	Keyboard Controller	
11-1	Keyboard Controller	11-2
13	Power-on Self Test and Setup	
13-1 13-2	Typical Setup Screen 1Typical Setup Screen 2	13-4 13-10
A	Specifications	
A-1	302 Board Dimensions	A-2
C	Jumper Settings	
C-1	302 Board Jumper Locations	C-2
D	Changing CPU Speed	
D-1 D-2	Setting Deturbo Mode - Software Method	D-4 D-5
I	Installation	
I-1 I-2 I-3	Installing SIMMsRemoving SIMMsInstalling a Numeric Coprocessor	I-2 I-4 I-5



Tables

3	Onboard Memory	
3-1 3-2 3-3 3-4	Memory Address Map Memory Resource Hierarchy Memory Installed Address Range	3-2 3-4 3-5 3-5
5	Intel ISA Bus	
5-1 5-2	Number of Agents SupportedISA Bus Cycles	5-2 5-3
6	8254 Programmable Interval Timer	
6-1 6-2 6-3 6-4 6-5	Counter Functions, Gates, and Signals Counter/Timer Address Map Control Register Bit Definition Read-back Command Format Format of a Latched Status Byte	6-2 6-5 6-6 6-8 6-9
7	8259 Programmable Interrupt Controllers	
7-1 7-2	Interrupt LevelsInterrupt Controller I/O Address and I/O Data	7-6 7-8

8	Direct Memory Access (DMA)	
8-1	DMA Channel Assignment	8-7
8-2	DMA Internal Registers	8-9
8-3	I/O Port Addresses for DMA Address and Count Registers	8-10
8-4	Control Register I/O Port Addresses	8-12
8-5	Command Register Bit Assignments	8-13
8-6	Mode Register Bit Assignments	8-14
8-7	Request Register Update	8-15
8-8	Write Single Mask Bit Assignments	8-16
8-9	Write All Mask Bit Assignments	8-17
8-10	Status Register Bit Assignments	8-18
8-11	Lower-page Register Addresses for Each Channel	8-20
8-12	Upper-page Register Addresses for Each Channel	8-21
9	1287 Real-time Clock (RTC)	
9-1	Real-time Clock Address Map	9-4
9-2	Time, Calendar, and Alarm Data Format	9-6
9-3	Status Register A (0AH)	9-7
9-4	Status Register B (0BH)	9-8
9-5	Status Register C (0CH)	9-10
9-6	Status Register D (0DH)	9-11
9-7	Diagnostic Status Byte (0EH)	9-12
9-8	Shutdown Status Byte (0FH)	9-13
9-9	Floppy Disk Drive Type Byte (10H)	9-14
9-10	Fixed Disk Type Byte (12H)	9-15
9-11	Equipment Byte (14H)	9-16
9-12	Low and High Base Memory Byte (15H and 16H)	9-17
9-13	Low and High Memory Expansion Bytes (17H and 18H)	9-17
9-14	Feature Installed Byte (1FH)	9-18
9-15	Fixed Disk Type 48 Parameters (20H:27H)	9-19
9-16	Shadow and Setup Byte (28H)	9-20
9-17	Low and High Extended Memory Bytes (30H and 31H)	9-21
9-18	Setup Information Byte (33H)	9-22
9-19	CPU Speed Byte (34H)	9-22
9-20	Fixed Disk Type 49 Parameters (35H:3CH)	9-23

10	Communication Ports	
10-1	Selection of Addresses and Interrupt Levels	10-1
10-2	Port Address and Interrupt Levels	10-3
10-3	Input Instructions	10-3
10-4	Output Instructions	10-4
10-5	Parallel Port Registers (Bits 7-4)	10-4
10-6	Parallel Port Registers (Bits 3-0)	10-5
11	Keyboard Controller	
11-1	Status Register Bit Assignments	11-3
11-2	Input Port Bit Assignments	11-5
11-3	Output Port Bit Assignments	11-6
11-4	Controller Commands	11-6
11-5	Data Stream Bits	11-8
12	302 Board Special Interfaces	
12-1	J23 Keylock Pinout	12-1
12-2	J23 Speaker Header Pinout	12-2
12-3	J23 Reset Interface Pinouts	12-3
A	Specifications	
A-1	302 Board Specifications	A-1
A-2	Physical Characteristics	A-3
A-3	Power Consumption	A-3
A-4	Maximum Current Ratings	A-4
A-5	Maximum +5V Current Drain on 302 Board	A-5

System BIOS Specifications	
Video I/O Floppy Disk Drive I/O (part 1) Fixed Disk I/O (part 2) RS-232 I/O System Service Routines System Descriptor Table Gate A20 Control Routine Keyboard I/O Printer I/O Clock Services	B-4 B-14 B-29 B-33 B-41 B-41 B-42 B-48 B-50
Jumper Settings	
Total Onboard RAM Onboard ROM Size Onboard RAM Type Onboard RAM Speed. Video Display Type Printer Acknowledge. Parallel Printer Port Selection. Serial Communications Port Selection.	C-3 C-4 C-4 C-5 C-5 C-6
Messages	
Beep Codes for Fatal Errors Beep Codes for Non-fatal Errors	E-13 E-14
Device Mapping	
Memory Address Map	F-2 F-3 F-4 F-5 F-6 F-7 F-8 F-9
	Video I/O Floppy Disk Drive I/O (part 1) Fixed Disk I/O (part 2) RS-232 I/O System Service Routines System Descriptor Table Gate A20 Control Routine Keyboard I/O Printer I/O Clock Services Jumper Settings Total Onboard RAM Onboard RAM Size Onboard RAM Type Onboard RAM Speed Video Display Type Printer Acknowledge Parallel Printer Port Selection Serial Communications Port Selection Messages Beep Codes for Fatal Errors Beep Codes for Non-fatal Errors Beep Codes for Non-fatal Errors Beep Codes for Hat Assignment I/O Address Map I/O Address Map I/O Address 78H Bit Assignments I/O Address 79H Bit Assignments I/O Address 79H Bit Assignments Interrupt Levels

H Pin Assignments for Major Signals H-1 CPU Pin Assignments.... H-1 H-2 AT32 Bus Connector Pin Assignments..... H-3 H-3 ISA Bus 8-bit Connector Pin Assignments H-4 H-4 ISA Bus 16-bit Connector Pin Assignments H-5 Nine-pin Serial Port (COM1) Connector Pin Assignments H-5 H-6 Nine-pin Serial Port (COM2) Connector Pin Assignments H-6 H-6 H-7 25-pin Serial Port (COM2) Connector Pin Assignment..... H-7 H-8 Parallel Printer Port Connector Pin Assignments..... H-8 H-9 Keyboard Connector Pin Assignment H-9



About This Manual

PURPOSE

This manual provides reference data for the Intel386™ MicroComputer Model 302 Board. This gives system designers sufficient information to allow them to integrate the board into systems, and to test and evaluate that integration. System designers may also use this manual on a continuing basis to support their customers, solve problems, or expand the system. For example, this manual may be used to:

- Design or select system components such as expansion boards, peripheral devices, and replacement parts
- Select or design systems and applications software
- Solve system integration and interfacing problems
- Troubleshoot at an advanced level
- Program complex applications and systems software such as device drivers, interrupt handlers, etc.

LIBRARY PLAN

This manual is part of a set of three manuals written for the Intel386 MicroComputer Model 302 Board. A brief description of these manuals is as follows:

- Intel386™ MicroComputer Model 302 System Technical Reference Manual. This manual and its companion volume, Intel386™ MicroComputer Model 302 Board Technical Reference Manual, contain detailed technical information. The system manual focuses on the major elements contained in the system module chassis. It provides a general description of the system, a brief overview of the 302 board, all internal interfaces to other components, and all external interfaces. Basic installation and removal procedures for system components and peripheral devices are also included. A detailed description of the 302 board is contained in a separate manual (see following).
- Intel386™ MicroComputer Model 302 Board Technical Reference Manual.
 This manual describes the 302 board in detail, and is written for engineers who design system accessories and for programmers who require information on hardware and firmware specifications.
- Intel386™ MicroComputer Model 302 System User's Guide. This manual is written for an end-user. It describes all system features, installation and operation of the system, how to install or remove system components, and basic troubleshooting procedures should problems occur.

AUDIENCE

This manual is written for an Original Equipment Manufacturer (OEM), or a system engineer or hardware or software designer. As such, it assumes you are familiar with the general terminology used in the field of microprocessor and microcomputer design.

ORGANIZATION

This manual is organized as follows:

Chapter 1	Board Overview — provides an overview of the 302 board. Included is a list of features, a block diagram of the board and a description of the feature set.
Chapter 2	Central Processing Core — describes the operation of the CPU, the 387 numeric coprocessor, cache memory, cache tag memory, and data and controller buffers.
Chapter 3	Onboard Memory — describes the onboard dynamic random access memory (DRAM), single in-line memory modules (SIMMs), and 32-bit expansion board DRAM.
Chapter 4	Intel AT32 Bus Interface — provides a brief description of the AT32 Bus Interface.
Chapter 5	Intel ISA Bus — introduces the ISA bus, provides general attributes of the ISA bus, and describes the ISA bus signal groups.
Chapter 6	8254 Programmable Interval Timer (PIT) — describes the operation of the PIT.
Chapter 7	8259 Programmable Interrupt Controllers (PICs) — describes the operation of the PICs.
Chapter 8	Direct Memory Access (DMA) — describes the operation of the dual 8237 DMA controllers.
Chapter 9	1287 Real-time Clock (RTC) — describes the operation of the RTC.
Chapter 10	Communications Ports — describes the system serial and parallel communications ports.
Chapter 11	Keyboard Controller — describes the operation of the keyboard controller.

Chapter 12	302 Board Special Interfaces — describes three special board interfaces: keylock, speaker, and the reset.
Chapter 13	Power-on Self Test and Setup — describes the power-on self test and the setup utility stored in ROM.
Appendix A	Specifications — provides system specifications for the 302 board.
Appendix B	System BIOS Specifications — provides BIOS specifications for the 302 board.
Appendix C	Jumper Settings — provides jumper settings for the 302 board.
Appendix D	Changing CPU Speed — describes how to effectively reduce the processor clock speed to 8 MHz.
Appendix E	Messages — describes the various screen messages and error beep codes. Included are POST messages, run-time messages, board error messages, and beep codes.
Appendix F	Device Mapping — provides tables that list mapping and addressing information related to system memory and onboard devices.
Appendix G	Hot Keys — lists keystroke sequences used to invoke special system functions.
Appendix H	Pin Assignments for Major Signals — provides pin assignments for all major signals present in the 302 system.
Appendix I	Component Installation — describes how to install the Intel 387™ numeric coprocessor and SIMMs.

Glossarv Glossary — defines the standard acronyms and

technical terms used in the field of microcomputer

design.

Index Index — provides important terms arranged in

alphabetical order for quick reference.

HOW TO OBTAIN MORE INFORMATION

You may be interested in obtaining further information about products and services relating to the Intel386 MicroComputer Model 302 Board. Or you may require more detailed information than is provided in this manual.

Please contact your local Intel Sales Office if you desire additional information.

NOTATIONAL CONVENTIONS

Certain notational conventions are used throughout this manual and others in the library. Refer to the glossary for specific definitions. Notational conventions include:

Throughout this manual, the term "system" applies to the system

Intel386™ MicroComputer Model 302 System.

board Throughout this manual, the term "302 board" or "board"

applies to the Intel386™ MicroComputer Model 302 Board.

F1 A letter, number, symbol, or word enclosed in a double

rectangle, and printed in small type represents a a key on your

keyboard. For example, the instruction "press [1]" means

press the key labeled "F1" on your keyboard.

This manual refers to most keys by the symbol, letter, or name Backspace

printed on the key. The exception is the Backspace key. The

Backspace key is called Backspace to distinguish it from the left

arrow kev.

Enter	This manual uses Enter to refer to the two Enter keys. Other manuals refer to the Enter keys as RETURN, CARRIAGE RETURN, or use an arrow. All these items are interchangeable.
x + y	Two or three key names with plus signs between them indicate multiple-key entries. For example, [ctr] + [Att] + [Dtr] means hold down the [ctr] and [Att] keys and press the [Dtr] key.
*	In signal definitions, the asterisk (*) following a signal name indicates an active low signal; for example IOCHCK*.
Н	An H suffix to a numerical value denotes hexadecimal numbers. For example, 0F8H means 0F8 (hexadecimal). See hex.
hex	Denotes hexadecimal numbers. Memory addresses are always listed in hexadecimal notation and are indicated by the term "hex" preceding or following the number; i.e., 0A hex or hex 8A. See H.
К	A K (upper case) suffix to a numerical value is used to indicate size in kilobytes; i.e., 7168K, 640K, etc. Note, that while a kilobyte is defined as 1024 bytes, the lower case k prefix used in other measurements indicates a quantity of 1,000. The K suffix is synonymous with KB or Kbyte. See Glossary.
Kb	A Kb suffix to a numerical value indicates size in kilobits. For example: 512Kb. (One kilobit is defined as 1024 bits.)
М	An M suffix to a numerical value is used to indicate size in megabytes; i.e., 1M, 256M, etc. Note, however, that while a megabyte is defined as 1,048,576 bytes the M prefix used in other measurements indicates a quantity of 1,000,000. The M suffix is synonymous with MB or megabyte. See Glossary.
Mb	An Mb suffix to a numerical value indicates size in megabits. For example: 4Mb. (One megabit is defined as 1,048,576 bits.)

All system messages (screen display) are shown in a non-proportional font to simulate the appearance of a screen display.

An italicized word or phrase is used to represent a variable, a publication title, or occasionally, to lend emphasis in textual descriptions. Where shown, DOS, UNIX or XENIX files, path names and directories are also italicized.

The longer POST and Boot error and informational messages in the index are followed by an ellipsis (three periods). This convention is used to denote that the message in the index is incomplete. The portion appearing in the index is of sufficient length to make a unique identification.

Four kinds of special notices are used throughout the text to emphasize specific information. Examples of each type of notice are as follows:



Notes are used to provide the reader with important or explanatory information that stands out from the rest of the text.

D DANGER

DANGER indicates the presence of a hazard that will cause death or severe personal injury if the hazard is not avoided.



WARNING

WARNING indicates the presence of a hazard that can cause death or severe personal injury if the hazard is not avoided.



CAUTION

CAUTION indicates the presence of a hazard that can or will cause minor personal injury or damage to hardware or software.

3

RELATED PUBLICATIONS

Refer to the following publications for additional information relating to the Intel386 MicroComputer Model 302 Board and its operating environment.

- Intel386™ MicroComputer Model 302 System Technical Reference Manual (Intel order number 505066-001)
- Intel386™ MicroComputer Model 302 User's Guide (Intel order number 505068-001)
- Introduction to the 80386 (Intel order number 231252-001)
- 80386 Programmer's Reference Manual (Intel order number 230985-001)
- 80387 Programmer's Reference Manual (Intel order number 231917-001)
- 386™ Microprocessor Hardware Reference Manual (Intel order number 231732-003)
- 80386 System Software Writer's Guide (Intel order number 231499-001)
- Microprocessor and Peripheral Handbook (Volume I and II) (Intel order number 230843-006)

Board Overview

1.1 INTRODUCTION

This chapter provides an overview of the 302 board. Included in this chapter is a list of features, a block diagram of the board, and a description of the feature set.

1.2 OVERVIEW

The 302 board contains the following components:

- 25 MHz 386 central processing unit (CPU)
- 64K cache memory and cache tag memory
- ASIC device for CPU control logic (CAT)
- 1, 2, 4, or 8M of onboard memory
- 64K read-only memory (ROM)
- AT32 bus interface
- ASIC device for bus interface and memory control logic (DBC)
- Two direct memory access (DMA) controllers

- Two DMA page registers for accesses to memory throughout the full AT32 memory range.
- Two programmable interrupt controllers (PICs)
- Programmable interval timer (PIT)
- Real-time CMOS clock/calendar with integral lithium battery
- ASIC device for I/O decode logic (RIO)
- A 121-pin extended numeric coprocessor socket
- Eight I/O expansion slots (two 32-bit, five 16-bit, one 8-bit)
- I/O ports (two serial and one parallel)
- ROM-based setup program, BIOS, and power-on self test
- Keyboard controller and ports
- Reset interface
- Speaker interface
- Keylock interface

Figure 1-1 illustrates the component layout of the 302 board. Figure 1-2 shows a functional block diagram of the 302 board.

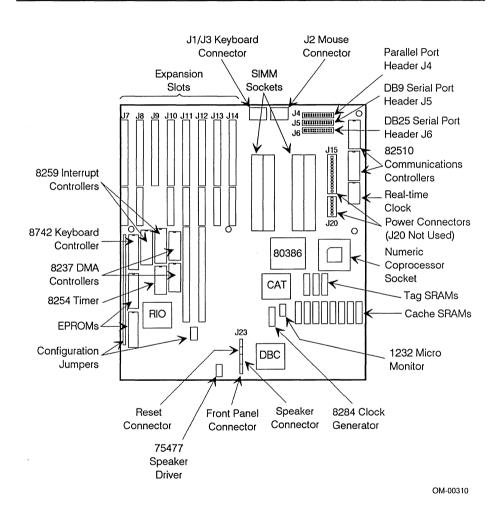


Figure 1-1. 302 Board Component Layout

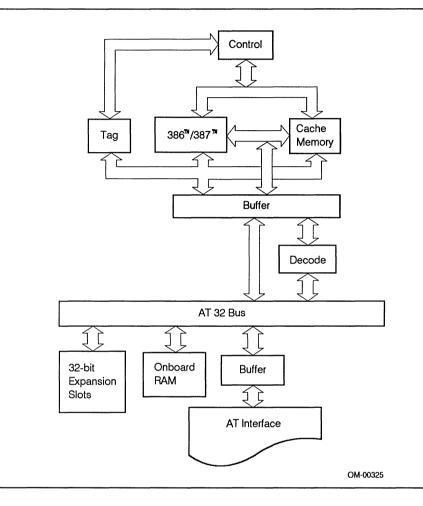


Figure 1-2. 302 Board Functional Block Diagram (sheet 1 of 2)

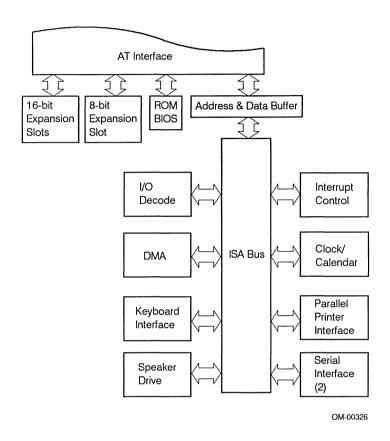


Figure 1-2. 302 Board Functional Block Diagram (sheet 2 of 2)

1.3 FEATURE SET DESCRIPTION

This section gives detailed description of the features listed earlier.

1.3.1 Central Processing Unit (CPU)

The CPU incorporates multitasking support, memory management, address translation caches, and a high-speed 32-bit bus interface. The CPU runs at a clock speed of 25 MHz resulting in a system speed of 40 ns per cycle. For applications requiring slower operation (such as installing some copy-protected software), a deturbo mode is provided. The deturbo mode reduces the effective system operating speed to 8 MHz by inserting wait states into the CPU cycle.

1.3.2 Memory

The 302 board contains three types of memory: read-only memory, cache and tag memory, and DRAM.

- The 302 board contains 64K of ROM. The ROM contains the BIOS, power-on self test (POST), and setup program.
- The 64K direct-mapped cache memory consists of eight static random access memory (SRAM) chips. The cache provides zero wait-state read performance and one wait-state write performance during CPU accesses. The tag memory consists of three SRAM chips and supplies 12-bit tag data for address comparison.
- The 302 board contains 4M (expandable to 8M) of DRAM.

1.3.3 AT32 Bus

The AT32 bus supports 28-bit physical addresses and has a physical memory range of up to 256M. The AT32 bus is a high performance 32-bit extension to the Intel ISA bus. The AT32 bus provides a standard interface supporting additional memory, and I/O. The 302 board contains two AT32 slots, each capable of accepting 16M of DRAM.

1.3.4 Intel ISA Bus

The system is compatible with the Intel ISA bus. I/O expansion boards communicate with the system via the ISA bus.

1.3.5 Programmable Interval Timer

Three general-purpose programmable interval timers (PITs), in an 82C54 integrated circuit, generate timing for memory refresh, software timing control, and speaker frequency generation.

1.3.6 Programmable Interrupt Controller

Two programmable interrupt controllers (PICs) provide 15 interrupt levels minimizing the software and real-time overhead in handling multilevel interrupts.

1.3.7 DMA Controller

Two 8237 direct memory access (DMA) controllers provide seven DMA channels for data transfer between main memory and I/O devices. Each DMA controller generates the memory addresses and controls signals needed to transfer data.

1.3.8 Real-time CMOS Clock

The Real-time Clock (RTC) provides 50 bytes of CMOS RAM and a battery backup power source for keeping the system clock/calendar and system configuration parameters in nonvolatile memory. This protects the contents of both the RAM and the clock during system power-up and power-down.

1.3.9 Input/Output Expansion Slots

The system has eight I/O expansion slots:

- One slot (J9) accepts 8-bit expansion boards only.
- Five slots (J7, J8, J10, J13, and J14) accept 8-bit or 16-bit expansion boards.
- Two slots (J11 and J12) accept 8-, 16-, or 32-bit expansion boards that are compatible with the AT32 bus.

1.3.10 Input/Output Ports

The 302 board provides a parallel printer port configured as either LPT1 or LPT2, and two serial communications ports configured as COM1 and COM2. Both serial ports can be disabled by changing the jumper settings.

1.3.11 Keyboard Interface

The keyboard connects to the 302 board through a bidirectional synchronous serial port. The bidirectional serial interface converts signals and sends the data to and from a 101- or 102-key keyboard.

1.3.12 Special Board Interfaces

The 302 board incorporates three special interfaces: keylock, speaker, and reset.

1.3.13 Firmware

The 302 board firmware consists of a POST that performs automatic system diagnostics, a setup program for setting system configuration parameters, and Phoenix Technologies' basic input/output system (BIOS).

The POST runs automatically and checks the CPU, keyboard, display, and system memory each time the system is turned on or rebooted.

The setup program is contained in the ROM BIOS on the 302 board and is used to store system configuration information. The information can be changed at any time by rerunning the setup program. The type of configuration information maintained by the setup program is as follows:

- Time and date
- Number and capacity of floppy disk drives
- Number and type of fixed disk drives
- Amount of base and extended memory
- Availability and type of primary display controller
- Keyboard present
- Coprocessor present
- CPU speed
- Shadow or do not shadow system BIOS and video BIOS
- Enable or disable cache memory, AT32 I/O, above 16M memory, speaker, and preboot SETUP

Onboard ROM consists of two 27256 EPROMS containing 64K of memory. The EPROMs contain the ROM code for the BIOS. Functions of the ROM BIOS are as follows:

- System initialization
- Power-on diagnostics
- System configuration
- Disk bootstrap loading
- Character bit patterns of the ASCII character set
- Storage of frequently needed input and output routines

If enabled in the setup program, the BIOS initializes the DRAM-shadowed BIOS option for increased system performance.

Central Processing Core

2.1 INTRODUCTION

This chapter describes the central processing core of the 302 board. The 386 CPU, the 387 numeric coprocessor, cache memory, cache tag memory, and address and data buffers are discussed. For more detailed information on these components, refer to the *Intel Microprocessor and Peripheral Handbook*.

2.2 OVERVIEW

The central processing core contains the following components:

- A 386 32-bit CPU
- A 387 numeric coprocessor
- Cache memory
- Cache tag memory
- Address/data buffers
- An ASIC device for CPU and cache control (CAT)

Figure 2-1 diagrams the central processing core.

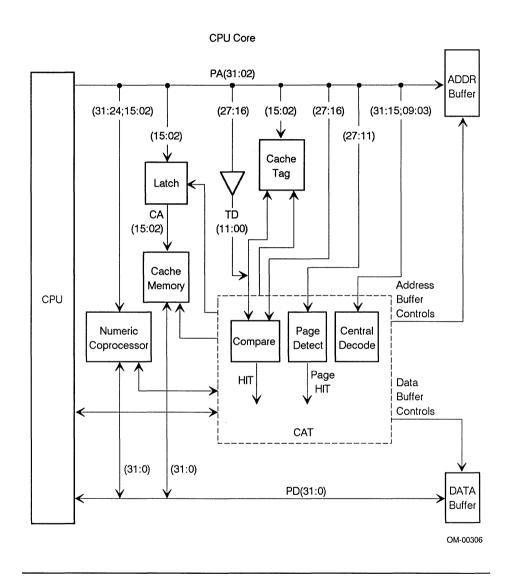


Figure 2-1. Central Processing Core

2.3 CPU

The CPU is a 386 microprocessor operating at 25 MHz. The CPU has separate 32-bit data and address paths, 32-bit registers, and on-chip memory management and protection. The CPU supports multiuser and multitasking systems, memory management, virtual memory, and task or memory isolation. Refer to "Related Publications" in About This Manual for a list of Intel reference manuals that provide detailed information on the 386 microprocessor.

2.3.1 Real Mode Architecture

The CPU defaults to real mode upon reset. Real mode is compatible with 8086/8088 and 80286 CPUs at the object code level and has the same capabilities and limitations. In real mode addressable physical memory is limited to 1M via segment registers, with a 64K limitation on segment size. Real mode does not provide memory protection features.

Real mode addresses are formed, as in the 8086, by combining the base address from a segment register with the offset value provided by the instruction. The CPU shifts the 16-bit base address value in the segment register left four bits, and adds the 16-bit offset value forming the 20-bit real address.

2.3.2 Protected Mode Architecture

In protected mode, the CPU increases the linear address space to 4 gigabytes and lets the user run programs of almost unlimited size (up to 64 terabytes). In this mode, the integrated memory management and protection mechanism translates virtual addresses to physical addresses. Protected mode also isolates the operating system and enforces the protection rules that are necessary for maintaining task integrity in a multitasking environment. This is useful in a multitasking and multiuser environment where resources are shared.

Protected mode provides memory paging, I/O protection, virtual-8086 mode, and a full 32-bit extended instruction set. Protected mode also provides source-code compatibility with the 8086/8088 and 80286 CPUs. This allows the direct execution of 16-bit applications at higher speeds.

2.3.3 Virtual-8086 Mode

The virtual-8086 mode is an extension of the protected mode. In this mode, the CPU provides compatibility with applications developed for the 8086/8088 while simultaneously providing a full 32-bit, large linear address programming environment in its protected mode.

Virtual memory allows programs to overcome the limitations of physical memory. The paging hardware allows the concurrent running of multiple virtual-8086 mode tasks and also provides protection and operating system isolation. The system divides virtual memory into many different segments which are mapped into physical memory during virtual memory execution. The memory management system transfers code and data between physical memory and disk memory.

2.3.4 CPU Signals

The following text defines the signal functions of the CPU. A signal name followed by an "I" in parentheses indicates an input signal, an "O" indicates an output signal and an "I/O" indicates an input/output signal. For more detailed information on the CPU signal functions, refer to the *Intel Microprocessor and Peripheral Handbook*.

CLK2 (I)	The CLK2 signal is a 50-MHz system clock signal. It provides the basic timing for the system. The CPU divides CLK2 by two to achieve a 25-MHz working clock.
D(31:0) (I/O)	The D(31:0) signals are the data bus lines. These tristate bidirectional signals input data during memory and I/O read cycles and output data during memory and I/O write cycles.
A(31:2) (O)	The A(31:2) signals are the address bus lines. These tristate signals output physical memory and I/O

addresses.

BE(3:0)* (O)	The BE(3:0)* signals are the byte enables. These signals indicate which bytes of the data lines will carry data for the current transfer.
ADS* (O)	The ADS* signal is the address status. When ADS* is asserted, it indicates the beginning of a bus cycle. While ADS* is asserted, address lines A(31:2) are valid.
M/IO* (O)	The M/IO (memory or I/O) signal indicates whether the access is to memory or an I/O device. When asserted (high), M/IO indicates that a memory or halt/shutdown cycle is in progress. When deasserted (low), M/IO indicates that a I/O or interrupt-acknowledge cycle is in progress.
D/C* (O)	The D/C (data/control) signal is dual-purpose. When asserted (high), D/C indicates that a data cycle is in progress. When deasserted (low), D/C indicates that a control cycle is in progress.
W/R* (O)	The W/R (write/read) signal is dual-purpose. When asserted (high), W/R indicates that a write cycle is in progress. When deasserted (low), W/R indicates that a read cycle is in progress.
LOCK* (O)	The LOCK* signal, when asserted, indicates a locked bus. It prevents access by an external coprocessor until the CPU completes a read or modifying a byte in memory.
READY* (I)	The READY* signal, when asserted, indicates the termination of a bus cycle and that the bus is available. Bus cycles are extended until terminated.
HOLD (I)	The HOLD signal is the bus hold request. The signal requests ownership of the CPU local bus.

HLDA (O) The CPU asserts HLDA (hold acknowledge) as a response to HOLD. The CPU tri-states its buses. When HOLD is deasserted, the CPU deasserts HLDA. INTR (I) The INTR (interrupt request) signal, when asserted, requests that the CPU suspend program execution and service the interrupt request. The CPU samples the INTR line at the beginning of each processing cycle. The INTR signal must be asserted at least two processing cycles before the current instruction ends. NMI (I) The NMI signal is the non-maskable interrupt. NMI is edge-triggered and cannot be masked by software. NMI has the highest priority of all interrupts. PEREQ (I) The PEREQ signal is the coprocessor extension operand request and acknowledge signal. It coordinates data transfer between the CPU and the numeric coprocessor. When asserted, PEREQ requests the CPU to perform a data operand transfer to the numeric coprocessor. BUSY* (I) The BUSY* signal is the coprocessor extension busy signal. It indicates the operating condition of the numeric coprocessor to the CPU. The CPU program execution stops as long as the signal remains asserted. ERROR* (I) The ERROR* signal, when asserted, indicates the operational status of the numeric coprocessor to the CPU. RESET (I) The RESET signal, when asserted, causes a system reset. It clears the internal logic of the CPU. The RESET signal initializes the CPU with a low-to-high transition.

2.3.5 Basic CPU Bus Operations

The bus control unit manages all bus operations and generates the address, data, and command signals for external memory and I/O operations. The bus control unit also transfers instructions to the instruction pre-fetch unit. Instructions are stored in a 16-byte pre-fetch queue while waiting for decoding and execution. The execution unit does not need to wait for the completion of a bus cycle before accepting a new instruction. This results in faster execution of instructions.

The instruction pre-decode unit receives and decodes the instructions from the pre-fetch queue. It then places the instructions in the decoded instruction queue for use by the execution unit. Instructions can be placed three-deep in the decoded instruction queue.

The execution unit performs the basic processing functions. It accepts the decoded instructions from the instruction pre-decode unit and executes them. The execution unit uses the bus unit to fetch and store operands during the execution of instructions.

The address paging unit and the segmentation unit provide memory management and protection services for the CPU. They also translate logical addresses into physical addresses for use by the bus unit. A register cache in the address unit contains the information used for performing the various memory translation and protection checks for each bus cycle.

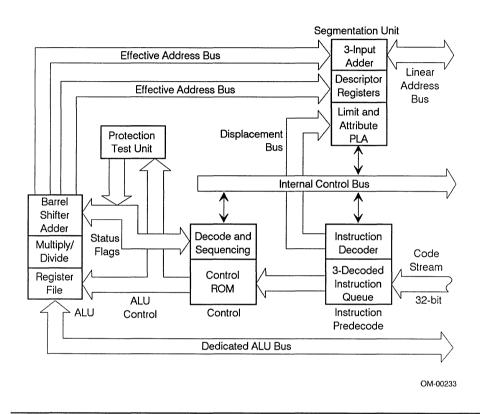


Figure 2-2. CPU Block Diagram (sheet 1 of 2)

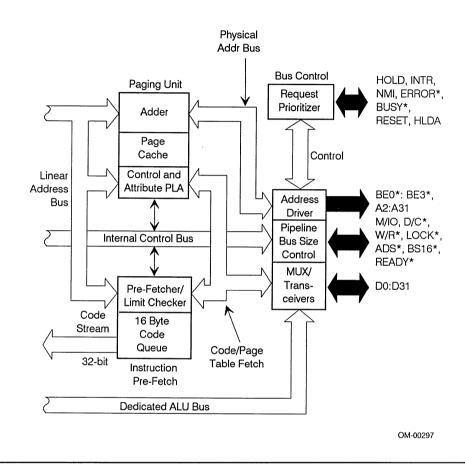


Figure 2-2. CPU Block Diagram (sheet 2 of 2)

The CPU uses a 50 MHz clock to control bus timing. The CPU divides this clock by two, which creates the 25 MHz internal processing clock and determines the bus cycle speed.

2.4 NUMERIC COPROCESSOR

A 121-pin extended numeric coprocessor (ENP) socket installed on the 302 board accommodates either an Intel 387 or Weitek 3167 numeric coprocessor. Both use the same clock generator as the CPU and fully support single-, double-, and extended-precision operations. The 387 numeric coprocessor is mapped on the I/O address space and the Weitek 3167 numeric coprocessor is mapped on memory address space.

A numerical coprocessor allows the CPU to perform high-speed mathematical calculations, logarithmic functions, and trigonometric functions.

Both numeric coprocessors provide the CPU with additional data types, registers, instructions, and interrupts specifically designed for high-speed numeric coprocessing. The registers in the numeric coprocessors hold constants and temporary results generated during calculations. These registers reduce memory-access time and improve bus availability. The numeric coprocessor register space can be used as a stack or fixed register set.

2.4.1 387™ Numeric Coprocessor Data Types

The 387 numeric coprocessor fully implements the ANSI/IEEE 754 standard for binary floating-point arithmetic. The 387 numeric coprocessor works with seven data types: 32-, 64-, 80-bit floating point, 16-, 32-, 64-bit integers, and 18-digit packed BCD.

The 387 directly extends the CPU instruction set. Extending the instruction set includes trigonometric, logarithmic, exponential, and arithmetic instructions for all data types.

2.4.2 387™ Coprocessor Programming Interface

The 387 numeric coprocessor functions as an I/O device through the I/O ports using addresses 800000F8H and 800000FCH for sending opcodes and operands, as well as for receiving and storing results. The CPU outputs address 800000F8H when writing a command or reading status, and outputs address 800000FCH when writing or reading data.

The CPU has three input signals (BUSY*, PEREQ, and ERROR*) that are used for controlling data transfers to and from the 387 numeric coprocessor.

The BUSY* signal informs the CPU that the 387 numeric coprocessor is executing an instruction and cannot accept another. The WAIT instruction informs the CPU to wait until the 387 numeric coprocessor completes execution of the current instruction.

The PEREQ signal indicates the 387 numeric coprocessor needs to transfer data to or from memory. Because the 387 numeric coprocessor is never a bus master, all input and output data transfers are performed by the CPU. The PEREQ signal is deasserted before the BUSY* signal is deasserted.

The 387 numeric coprocessor asserts an ERROR* signal after an instruction results in an error that is not masked by the 387 numeric coprocessor's control register. If an error occurs, the ERROR* signal is asserted before the BUSY* signal is deasserted. As a result, the CPU receives an interrupt. If a higher priority interrupt does not exist, the CPU services the interrupt. Interrupts report exception conditions.

The 387 numeric coprocessor detects six different conditions that may occur during instruction execution. If the proper exception mask is not set in the control register, the 387 numeric coprocessor asserts an ERROR* signal. The ERROR* signal generates a hardware interrupt (BUSY*) holding the 387 numeric coprocessor in a busy state. This state is cleared by writing zeroes to I/O port address F0H. The numeric exception conditions recognized by the 387 are as follows:

- 1. Invalid operations (stack fault or IEEE standard invalid operation)
- 2. Divide-by-zero
- 3. Denormalized operand
- Numeric overflow
- 5. Numeric underflow
- 6. Inexact result (precision)

The POST code in the system ROM enables the hardware interrupt. If the POST detects a 387 numeric coprocessor present, it sets the hardware interrupt vector to point to an interrupt routine in ROM. This routine clears the latch on the BUSY* signal and transfers control to the address pointed to by the nonmaskable interrupt (NMI) vector. The NMI interrupt handler reads the 387 numeric coprocessor's status and determines if the coprocessor caused the NMI. If the 387 coprocessor is not the interrupt source, control passes to the original NMI interrupt handler.

While the CPU executes numeric programs in either real or protected mode, interrupts report exception conditions. Refer to the 80387 Programmer's Reference Manual (Intel order number 231917-001) for detailed descriptions of 387 interrupts and exceptions.

All communication between the CPU and the 387 numeric coprocessor is transparent to applications software. The 387 numeric coprocessor operates whether the CPU executes instructions in real-address mode, protected mode, or virtual-8086 mode. The CPU handles all memory accesses. The 387 numeric coprocessor operates on instructions and values passed to it by the CPU and is not aware of the mode of the CPU.

For complete information on programming the 387, refer to the 80387 *Programmer's Reference Manual*, (Intel order number 231917-001).

2.4.3 Weitek 3167 Interface

The Weitek 3167 memory-mapped numeric coprocessor communicates with the CPU over the same address bus that connects the main memory to the CPU. Instructions are defined by the 14 least-significant address bits, A(15:2), as well as three of the four byte enable bits (BE2:0).

The 3167 numeric coprocessor responds to memory addresses C0000000H through C1FFFFFH. Although addresses C0000000H to C000FFFH are normally used, it is important to be sure that other components in the system do not conflict with the address space decoded by the 3167. Writing to this address space causes the 3167 numeric coprocessor to execute instructions and reading causes the 3167 to drive the data bus.

2.5 CACHE MEMORY AND TAG

The cache memory and tag logic together implement a direct mapped 64K write-through cache for the CPU. Figure 2-3 shows the cache memory and tag logic.

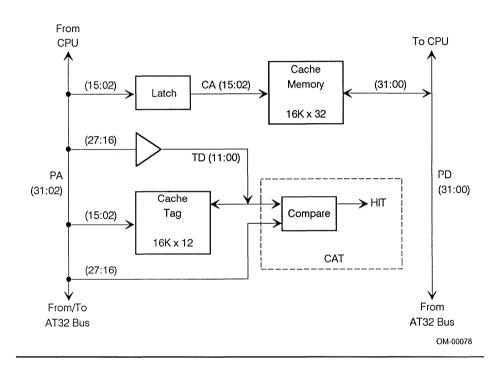


Figure 2-3. Cache and Tag Memory

2.5.1 Cache Memory

The cache memory resides between the CPU and the AT32 bus. The cache memory holds copies of information contained in the system memory for fast CPU access. The cache contains 64K of memory organized into 16K words of 32 bits each. The cache uses a direct-mapped design with write-through to main memory on all write cycles. A CPU access or DMA access to memory locations contained in the cache is called a "hit." For cache hits during read operations, the access occurs at full speed with no wait states. For cache hits during write operations, the access occurs at full speed with one wait state.

Eight 16K \times 4 SRAMs comprise the memory portion of the cache. The SRAMs make up the 32-bit word for the bus and are addressed by the least-significant 14 bits of the address bus.

2.5.2 Cache Tag Memory

The cache tag memory consists of three 16K x 4 SRAMs. They supply the 12-bit tag address to the address comparator. The least-significant 14 bits of the address bus address the tag SRAMs. A write path between the address bus and the cache tag SRAMs provides a means for cache tag updates.

2.5.3 Cache Memory Read Requests

Memory read requests originate from the CPU. Read requests cause the cache data SRAMs to drive data onto the CPU bus. Tag SRAMs are also accessed and the tag data are compared to bits 27 through 16 of the address bus. When tag data matches (a hit), the requested information is present in cache memory and is delivered to the CPU. When tag data does not match (a miss), the requested information is not contained in the cache. In this case, the cache read aborts, and the CPU read request is routed to main memory on the AT32 bus. Data returned from the AT32 bus is updated to the CPU and to the cache memory. Tag SRAMs are updated at the same time as the data from the processor address bus. During an AT32 bus memory read cycle, the CPU remains in a wait state.

2.5.4 Cache Memory Write Requests

Memory write requests result in a cache update if the memory location is present in the cache. Write requests are also posted to the AT32 bus so system memory remains coherent with the cache. Updating the cache is the same as a read cache miss update, but without writing to the tag SRAMs.

Memory writes originating from another bus master use the same tag comparison logic for the addressed memory location. If the addressed location is present in the cache, a cache update takes place. Updating the cache is the same as a processor-initiated write request update. This ensures that changes to the system memory are always reflected in the cache memory regardless of who makes the changes.

2.6 ADDRESS AND DATA BUFFERS

Bidirectional buffers transfer address and data signals between the CPU bus and the AT32 bus. Four buffers transfer CPU data, D(31:0), and four buffers transfer CPU addresses, A(27:02) and byte enable signals, BE(3:0)*.

2.7 CPU/CACHE CONTROL

An ASIC device, CPU/Cache Control AT (CAT), contains the integrated functionality of the CPU core control logic for the system board.

This device provides the following functions:

- Tracking CPU bus cycle initiation and terminating the cycle based on the specific requirements
- Control signals for address and data buffers
- Control signals for cache and tag memory
- Centralized arbitration mechanism on the AT32 bus among the system CPU, DRAM refresh, ISA bus DMA, and two AT32 coprocessors
- Numeric coprocessor interface and control signals.

Onboard Memory

3

3.1 INTRODUCTION

This chapter describes the onboard memory that is available for the 302 board in basic and extended configurations.

3.2 ONBOARD MEMORY OVERVIEW

The 302 board memory map consists of onboard ROM, onboard 32-bit DRAM, expansion board 32-bit RAM, and the remaining memory mapped to the ISA bus. The system supports up to 40M of fast 32-bit wide memory. The 302 board supports 1, 2, 4, or 8M of onboard DRAM with parity. In addition, two expansion slots are provided for two 32-bit expansion boards (ATMEM4, ATMEM8, or ATMEM16).

The system BIOS and video BIOS may be shadowed in 32-bit memory for enhanced performance. All 32-bit memory except shadowed system BIOS is cacheable. The shadowing and caching features are controlled by the system ROM setup program. See Chapter 13 and Appendix B for more information on ROM. Appendix C specifies the 302 board jumper settings for memory configuration. Table 3-1 shows the system memory address map.

Table 3-1. Memory Address Map

Address	Name	Function
000000:09FFFFH	512K system board	302 board memory (0 - 640K)
0A0000:0BFFFFH	128K video RAM	Reserved for video display controller
0C0000:0C7FFFH	32K video ROM	Reserved for video display controller, BIOS ROM, and video BIOS ROM shadow
0C8000:0DFFFFH	96K I/O expansion ROM/RAM	Reserved for ROM and RAM on I/O adapters
0E0000:0FFFFH	128K system ROM BIOS/shadow RAM	Reserved for system ROM BIOS & shadow of system and video ROM BIOS
100000:FDFFFFH	Extended memory	Extended Memory space
FE0000:FFFFFFH	128K reserved on 302 board	Duplicates code assignment at address 0E0000H
1000000:FFFFFFFH	Extended Memory	Extended Memory space

3.3 302 BOARD DRAM

RAM on the 302 board can support configurations of 1, 2, 4, or 8M with parity. This memory is organized as one or two banks of four Single In-line Memory Modules (SIMMs). SIMMs are small boards containing several dynamic random access memory (DRAM) chips. The SIMMs are organized as 256K-bit \times 9 or 1M-bit \times 9. Each bank is 32 data bits wide with 1 parity bit for each byte. Both banks must have the same size SIMMs installed.

The 302 board may be configured to support 100 ns or 85 ns speed SIMMs. The default access speed is 100 ns. All SIMMs must operate at the configured speed or faster.

The 302 board supports three modes of memory operation. The modes are RAS/CAS, fast-paged, and static column. The capability to support the fast-paged and static column modes is selected by jumpers on the 302 board. Although different mode SIMMs can be mixed on the board, the system requires that the jumpers be set for the SIMMs with the lowest performance.

RAS/CAS mode is the standard access mode and has the lowest performance. If the special requirements of fast-paged and static-column mode accesses are not met, then a RAS/CAS mode access will occur even though the board is configured for one of the advanced modes.

Fast-page mode is the default configuration for the 302 board. Fast-page mode DRAM chips are organized into pages. If the current and previous memory cycles are both made to the same DRAM page, then a fast-paged mode access will be executed. Fast-page mode accesses are significantly faster than RAS/CAS mode accesses.

Static-column mode is another advanced mode of operation. It is similar to fast-paged mode in that a static-column access will be executed if the current and previous memory cycles are both made to the same DRAM page. Static-column mode is slightly faster than fast-page mode.

3.4 CONFIGURING AT32 MEMORY

The system hardware automatically configures up to 16M of 32-bit memory into one contiguous address range. The 32-bit memory consists of all onboard memory plus any AT32 memory expansion boards installed in the system. POST tests for and reports the amount of memory available.

At reset time the 302 board hardware monitors signals from the total onboard RAM jumpers and configuration signals from the AT32 expansion slots. From this data, the 302 board configures the way memory addresses are allocated to each memory resource. Table 3-2 defines the hierarchy of memory resource allocation.

Table 3-2. Memory Resource Hierarchy

Memory Resource	Priority Level	Assigned Address Space
Onboard bank 0	1 - highest	Lowest
Onboard bank 1	2 - :	:
Expansion slot 0	3 - :	:
Expansion slot 1	4 - lowest	Highest

The onboard bank 0 is always assigned first and starts at address 00000000H. If present, memory in onboard bank 1 is assigned the next address space, followed by memory in AT32 expansion slot 0. Finally, memory in AT32 expansion slot 1, if any, is assigned. If the total of the onboard and AT32 expansion slot memory is less than 16M, accesses between the top of this 32-bit memory and 16M (address 00FFFFFFH) are made to the ISA bus. Tables 3-3 and 3-4 list examples of some typical memory configurations.

Table 3-3. Memory Installed

302 Boa 0	ard Bank 1	Expan 0	sion Slots 1	Total
4M	•	-	-	4M
1M	1M	8M	-	10M
4M	4M	8M	16M	32M
1M	-	-	8M	9M

Table 3-4. Address Range

Onboard	Slot 0	Slot 1	ISA
4M	-		0400000 - 0FFFFFFH
2M	0200000 - 09FFFFH	-	0A00000 - 0FFFFFFH
8M	0800000 - 0FFFFFH	1000000 - 2FFFFFFH	-
1M	-	0100000 - 08FFFFFH	0900000 - 0FFFFFH

3.5 DRAM REFRESH

All dynamic random access memory requires each memory cell to be refreshed at least once every four ms. To meet this requirement a system wide refresh cycle occurs approximately every 15 μ s. The 302 board performs the refresh cycle to both ISA bus memory and 32-bit memory simultaneously. During the refresh cycle the processor can still access cache.

An ISA bus secondary requesting agent must initiate refresh cycles at appropriate intervals if it has control of the system for greater than 15 μ s. The 302 board will execute the refresh cycle, and then return control to the secondary requesting agent. Refer to Chapter 5 for more information on the ISA bus.



Intel AT32 Bus Interface

4.1 INTRODUCTION

This chapter provides an overview of the AT32 bus. Please contact your local Intel Sales Office if you would like more information on the AT32 bus.

4.2 AT32 BUS OVERVIEW

The AT32 bus is a high-performance 32-bit extension to the ISA bus. The AT32 bus supports the CPU with an efficient, high performance interface to standard DRAM.

The AT32 bus is a physical and electrical extension of the Intel ISA bus. All AT32 bus memory expansion boards use the same physical form factor as an ISA bus expansion board. In addition to the two standard ISA bus connectors the AT32 bus adds a third connector. This connector provides additional signals and power which support a full-32 bit data path and enhanced addressing capability.

The 302 board has two AT32 expansion slots J11 and J12. These slots support 8-bit and 16-bit ISA bus expansion boards. In addition, they can accommodate high performance AT32 memory boards with 4, 8, or 16M of DRAM installed on each board. With a maximum of 8M of memory on the boards this allows up to 40M of memory in the system. See Appendix H for the connector pinouts of J11 and J12.

Configuration of AT32 memory boards into the system is jumperless and automatic. The configuration process is a function of the 302 board and is described in Chapter 3.

Intel ISA Bus

5.1 INTRODUCTION

This chapter provides an overview of the ISA bus interface as implemented on the 302 board. Note that the following description only applies to the Intel implementation of the ISA bus.

5.2 **BUS AGENTS**

The ISA bus allows several different bus agents. A bus agent is a physical unit which has an interface directly to the ISA bus. A memory expansion board, a LAN controller, and a modem are all examples of bus agents. The two basic types of bus agents are requesting agents and replying agents. The applicable bus agent type definitions are listed below.

Requesting agent

(RA)

The requesting agent initiates an ISA bus cycle. A requesting agent can be either primary or secondary, as

explained next.

Primary requesting agent (PRA)

The 302 board is the primary requesting agent. It has immediate access to the ISA bus when control has not been granted to an secondary requesting agent.

Secondary requesting The SRA is an optional requesting agent that normally agent (SRA) does not have immediate control of the ISA bus. The

SRA requests control from the primary requesting agent. Multiple SRAs are allowed. An SRA must have a 16-bit bus interface. Eight-bit SRAs are not allowed on the ISA

bus.

Replying agent Any agent which responds to ISA bus cycles initiated by

a requesting agent is a replying agent. A replying agent

cannot initiate ISA bus cycles.

5.2.1 Configuring Bus Agents

The ISA bus services eight agents via 8- and 16-bit portions of the bus. The number of each agent type that is supported is listed in Table 5-1.

Table 5-1. Number of Agents Supported

Agent Type	No. Supported	Notes
Agent Requesting Primary Requesting Secondary Requesting Replying	1 - 9 1 - 8 1 0 - 7 0 - 8	General agents; PRA + 8 PRA+7 PRA alone 16-bit interface only 8- or 16-bit interface

5.2.2 Agent Functional Model

Table 5-2 describes the types of ISA bus cycles in which ISA bus agents may participate.

Table 5-2. ISA Bus Cycles

Type of Agent	Agent's Action		
Primary Requesting Agent	Initiates:	Memory access I/O access DMA access Global refresh	
	Responds to:	Interrupt request DMA request Bus arbitration request	
Secondary Requesting Agent	Initiates:	Memory access I/O access	
	Responds to:	Global refresh	
	Gains Bus Ownership by:	DMA request Assuming bus ownership on DMA grant Assuming responsibility for refresh initiation	
Replying Agent	Responds to:	Memory access I/O access DMA access Global refresh	
	Seeks PRA service through:	Interrupt request DMA request	

5.2.3 General ISA Bus Attributes

Specific attributes of the ISA bus are:

- The memory address is 24 bits long, and the data path is 16 bits wide.
 This provides a 16M memory address space with 8- and 16-bit data transfers.
- The I/O address is 16 bits long and the data path is 16 bits wide. This
 provides a 64K I/O address space with 8- and 16-bit transfers.
- Interrupt lines support signalling between agents on the bus and the PRA.
- The DMA capability allows 8- or 16-bit data transfers between memory and I/O agents without direct intervention of the CPU on the PRA.
- The PRA performs refresh cycles. The PRA also initiates refresh cycles at the request of an SRA that is in control of the bus, in order to maintain the integrity of the data in system DRAM.
- The ISA bus supports multiple agents. It can support up to nine agents, including one PRA and up to seven SRAs or eight replying agents. A PRA (the 302 board itself) is required in all implementations. SRAs and replying agents are optional.

5.3 SIGNAL GROUPS

The ISA bus contains seven groups of signals. These signal groups are:

- Address
- Data
- Cycle control
- Central control
- Interrupt
- Direct memory access (DMA)
- Power

The signal groups support a PRA, which has an onboard DMA controller, optional SRAs, and replying agents providing DMA or I/O memory expansion. In the following signal descriptions the input and output direction designations for each signal are referenced to the PRA. See Appendix H for information about the signal/ISA bus connector pin correspondence.

5.3.1 Address Signal Group

At the beginning of a bus cycle, the requesting agent drives signals in the address signal group to specify the address and data transfer width.

A(19:00) (I/O)

Address signals, A(19:00), are latched outputs driven by a requesting agent. They are the least-significant 20 bits of the address value. Signals A(19:00) become valid when BUSALE is asserted, and can be latched by responding agents on the falling edge of BUSALE.

Address signals, A(19:16), are driven low during I/O cycles. During refresh cycles, the PRA drives A(07:00) with the DRAM row address to be refreshed.

LA(23:17) (I/O)

The unlatched address, LA(23:17), bus signals are driven by a requesting agent. These signals are not latched by the PRA. However, they are valid when BALE is asserted, and they may be latched on the falling edge of this signal. Signals LA(23:17) represent bits 17 through 23 of the memory address presented on the bus. These signals are used by 16-bit replying agents when generating SRDY*, MCS16*, and IOCS16*.

The requesting agent drives LA(23:17) during any transfer cycle. During I/O cycles and refresh cycles, LA(23:17) are driven low. During SRA cycles, LA(23:17) must be valid throughout the entire transfer cycle, and BUSALE must be asserted by the PRA. The PRA drives LA(23:17) to 0 during refresh cycles.

SBHE* (I/O)

The system bus high enable signal, SBHE*, is asserted by a requesting agent to indicate a data transfer on lines D(15:08) for repliers that support 16-bit transfers. The signals SBHE* and A0 are used in 16-bit transfers to determine which bytes are being transferred over the data bus.

BUSALE (O)

The bus address latch enable signal, BUSALE, is an address strobe driven by the PRA. Signals LA(23:17) are valid when BUSALE is asserted, and they can be latched on the falling edge of BUSALE. Address signals A(19:00) are latched by the PRA on the leading edge of BUSALE during bus cycles initiated by the PRA.

All agents must be level sensitive with respect to BUSALE. This means that although the address signal group signals or decodes generated from them may be latched by agents on the falling edge of BUSALE, all agents must monitor the address signal group signals whenever BUSALE is asserted. This is especially important during DMA cycles, SRA cycles, and refresh cycles.

For all DMA controller cycles (including secondary-requesting-agent cycles), the PRA asserts BUSALE, thereby allowing addresses LA(23:17) to pass through transparent address latches to the bus.

AEN (O)

The DMA address enable signal, AEN, is asserted by the PRA when its CPU is in the hold mode and its DMA controller has control of the bus. The AEN signal is negated by the PRA when its CPU is in control of the bus or when the DMA controller has granted the bus to an SRA. When AEN is asserted, all agents other than the PRA must tri-state their address signal group and cycle control signal group outputs to the ISA bus.

During DMA cycles, the validity of LA(23:17) and A(19:00) is indicated by the assertion of both AEN and BUSALE. SRAs cannot conduct DMA cycles, because only the PRA can drive the DACKn* and AEN signals.

5.3.2 Data Signal Group

The data signal group consists of one set of 16 data bits. Data transfers may occur over either of the two bytes independently of one another.

D(15:0) (I/O)

On the data bus, D(15:00), D15 is the most significant bit and D0 is the least significant bit. All 8-bit replying agents must connect only to the least significant 8 data lines, D(07:00). To support communication of 8-bit replying agents to 16-bit requesting agents, the PRA supports both data swapping and transfer reformatting. During odd-byte transfers between a 16-bit requesting agent and an 8-bit replying agent, the PRA drives the data appearing on D(07:00) onto D(15:08). During 16-bit accesses to 8-bit agents, the PRA performs transfer reformatting of 8-bit data by using two consecutive 8-bit ISA bus cycles. This allows the PRA to reformat the replying agent's 8-bit data into the required 16-bit data. The PRA tri-states D(15:00) during refresh operations.

Cycle Control Signal Group 5.3.3

This group of signals controls the duration and type of cycles. The group consists of six commands signals, two ready signals, and three signals that specify the cycle type. The command signals define the address space (memory or I/O) and the data direction (read or write). The ready signals modify the command pulse widths to lengthen or shorten the default cycle timing.

MEMR*, MRDC* (I/O) The memory read signal, MRDC*, is asserted when the requesting agent is ready for a replying agent to drive the data bus with the contents of the memory location specified by LA(23:17) and A(19:00). Signal MEMR* is identical in function to MRDC*, except that MEMR* is asserted when the memory read access falls below 1M. Eight-bit agents receive only MEMW*.

> The PRA asserts MEMR* and MRDC* during refresh cycles initiated by an SRA in control of the ISA bus.

MEMW*, MWTC* (I/O)

The memory write signal, MWTC*, is asserted during a write cycle when the requesting agent is driving the data bus. Signal MEMW* is identical in function to MWTC*, except that MEMW* is asserted when the memory write access falls below 1M.

IORC* (I/O)

The I/O read signal, IORC*, is asserted when the requesting agent is ready for a replying agent to drive the data bus with the data available from the I/O port specified by A(15:00).

IOWC* (I/O)

The I/O write signal, IOWC*, is asserted during an I/O write cycle when the requesting agent is driving the data bus, and it is deasserted when a replying agent must clock the data to the I/O port specified by A(15:00).

MCS16* (I)

The memory cycle select signal, MCS16*, is asserted by a 16-bit memory agent to indicate to the requesting agent that a 16-bit cycle can be executed. Replying agents generate MCS16* based on a decode of LA(23:17). Timing requirements placed on MCS16* prevent using the memory command signals MEMR*/MRDC* and MEMW*/MWTC* in the generation of MCS16*. The requesting agent ignores MCS16* on I/O cycles.

IOCS16* (I/O)

The I/O cycle select signal, IOCS16*, is asserted by a 16-bit I/O agent to indicate to the requesting agent that a 16-bit cycle can be executed. Replying agents generate IOCS16* based on a decode of A(15:00). Timing requirements placed on IOCS16* prevent the use of IOWC* and IORC* in generating IOCS16*. The requesting agent ignores IOCS16* on memory cycles.

IOCHRDY (I)

The I/O channel ready signal, IOCHRDY, is an asynchronous ready signal from a replying agent. It is deasserted to force the requesting agent to lengthen the bus cycle by inserting an integral number of wait states (one wait state equals 62.5 ns). The signal IOCHRDY must not be deasserted for longer than 15 μ s. The PRA ignores IOCHRDY during zero-wait state cycles.

SRDY*

The synchronous ready signal, SRDY*, is asserted by the replying agent to terminate the current bus cycle without any further wait states. The absolute minimum command-pulse width is nominally one SYSCLK period (125 ns) and is known as a zero-wait-state cycle.

SRAs are not required to support SRDY*. The PRA ignores SRDY* when accessing 8-bit replying agents.

MEMREF* (I/O)

The refresh signal, MEMREF*, is asserted during a DRAM refresh cycle. Only memory-read cycles may occur while MEMREF* is asserted. The address present on A(07:00) is the refresh row address used by the memory bus agent.

AN SRA can, if it is the current bus owner, tri-state its address, command, and data drivers and assert MEMREF*. Doing so forces the PRA to conduct a refresh cycle. SRAs must force a refresh every 15 μ s if they retain ownership of the bus, or the contents of the system DRAM will be lost. When a refresh cycle is initiated in this manner, the PRA asserts A(07:00) and MEMR*/MRDC*. Because refresh cycles occur at a period of 15 μ s, each of the 256 refresh addresses is refreshed at least once every 4 ms.

5.3.4 Central Control Signal Group

The central control group consists of special timing, control, and error signals. This section describes each signal of the group.

SECMAST*

The secondary master signal, SECMAST*, is asserted by an SRA to gain control of the bus after receiving the appropriate DACKn* from the PRA. When SECMAST* is asserted, all other requesting agents must tri-state their address, data, and control signals. After SECMAST* is asserted, the SRA must wait at least one SYSCLK period before driving the address and data group signals, and it must wait at least two SYSCLK periods before driving the cycle control group signals.

If SECMAST* is asserted for longer than 15 μ s, the SRA must initiate refresh cycles to maintain DRAM data integrity. Note that only DMA channels programmed in the cascade mode may be used by SRAs wishing to gain control of the ISA bus.

IOCHCK* (I)

The I/O channel check signal, IOCHCK*, can be asserted by any agent to mark an error condition that cannot be corrected, such as a memory parity error. Signal IOCHCK* must be asserted for at least 15 ns for the PRA to recognize that an error condition has occurred.

RSTDEV (O)

The reset signal, RSTDEV, is asserted by the PRA to initialize all agents on the ISA bus after power-up or during a low-voltage condition.

SYSCLK (O)

The system clock signal, SYSCLK, has a frequency of 8 MHz with a 50% duty cycle, and is driven by the PRA. Bus cycle times are directly proportional to the clock period. All synchronous signals on the ISA bus are synchronous to SYSCLK.

Bus cycles are lengthened by IOCHRDY or shortened by SRDY* in integer multiples of one-half the SYSCLK period. For example, SRDY* could be asserted during a 16-bit cycle to reduce the command pulse width to 1.5 SYSCLK periods. Likewise, IOCHRDY could lengthen a 16-bit cycle to N + 2.5 SYSCLK periods, where N is the number of wait states that the accessed device requests. Since the DMA controller operates off of a 4-MHz clock, DMA cycles are extended in multiples of 2 SYSCLK periods.

84OSC (O)

The oscillator output signal, 84OSC, is a 50% duty cycle clock signal with a frequency of 14.31818 MHz. Signal 84OSC is not synchronous with either SYSCLK or any other signals on the ISA bus, so it must not be used in applications which require synchronization with the bus. It is intended for use in timing or counting operations only.

5.3.5 Interrupt Signal Group

The interrupt signal group consists of a set of signals replying agents use to obtain interrupt service from a requesting agent.

IRQ (I)

An interrupt can be requested by asserting an IRQ line. The line must remain asserted until the interrupt is acknowledged by the appropriate interrupt-service software routine. The interrupt request lines are IRQ(15,14,12:09, 07:03)

5.3.6 Direct Memory Access Signal Group

The DMA signal group controls DMA service and transfer of ISA bus ownership from the PBA to an SBA

DRQ(I)

The DMA request signals, DRQ(7:5, 3:0), are asynchronous channel requests used to gain either DMA service or control of the ISA bus from the PRA. DMA service or bus control can be attained by asserting a DRQ line and keeping it asserted until the corresponding DACK* line is asserted by the PRA. When second requesting agents wish to gain control of the bus, they must only use DMA channels that have been programmed to operate in the cascade mode.

DACK* (O)

The DMA request acknowledge signals, DACK(7:5, 3:0), are driven by the PRA to acknowledge a DMA request DRQ(7:5, 3:0). I/O repliers use DMA acknowledge signals for address selection during DMA cycles when AEN is asserted.

TC (O)

The terminal count signal, TC, is asserted by the PRA when any one of its DMA channels has reached its terminal count, signalling the end of the preprogrammed DMA transfer.

5.3.7 Power Group

The ISA bus provides DC power at \pm 5V, \pm 12V, and 0V (ground).

+ 5 Volts	Pins 3 and 29 on the non-component side of the 8-bit connector, as well as, pin 16 on the non-component side of the 16-bit connector supply current at a +5V level for 16-bit agents. Pins 3 and 29 on the non-component side of the 8-bit connector supply current at a +5V level for 8-bit agents.
- 5 Volts	Pin 5 on the non-component side of the 8-bit connector supplies current at a $-5V$ level.
+ 12 Volts	Pin 9 on the non-component side of the 8-bit connector supplies current at a +12V level.
- 12 Volts	Pin 7 on the non-component side of the 8-bit connector supplies current at a $-12V$ level.
Ground	Pins 1, 10, and 13 on the non-component side of the 8-bit connector, as well as, pin 18 on the non-component side of the 16-bit connector provide a return path for the current supplied by the other power pins for 16-bit agents. Pins 1, 10, and 13 on the non-component side of the 8-bit connector provide this return path for 8-bit agents.

8254 Programmable Interval Timer

6.1 INTRODUCTION

The system timer for the 302 board is an Intel 82C54 programmable interval timer (PIT). The PIT contains three independent 16-bit counters counting down in binary coded decimal (BCD) or binary. The counters are read, written, and configured using common control logic. Each counter operates in one of the following six modes:

Mode 0: interrupt on terminal count

Mode 1: hardware triggered one-shot

Mode 2: rate generator

Mode 3: square wave generator

Mode 4: software triggered strobe

Mode 5: hardware triggered strobe

6.2 COUNTER DESCRIPTION

Common clock input pins (CLKC) drive each of the three PIT counters. The frequency of the common clock is 1.19318 MHz. The output of counter 0 (PITIRQ) is connected to IRQ of the interrupt controller module (INTC1). This output provides the system-timer interrupt for time-of-day, disk timeout, and other system timing functions. System hardware uses counter 1 to generate a DRAM refresh-operation request signal. Counter 2 generates the tone for the system speaker.

Each counter contains the following elements: a control register, a status register, a 16-bit counting element (CE), a pair of 8-bit-counter-input latches (CIL, CIH), and a pair of 8-bit-counter-output latches (COL, COH). Each counter also has a clock input for loading and decrementing the CE. The CE is a mode-defined input for controlling both the counter and an OUT signal. The counter mode and condition of CE controls the OUT signal state and function. Table 6-1 lists the functions of the counters, the respective gates, and the clock in and clock out signals.

Table 6-1. Counter Functions, Gates, and Signals

Counter	Signal	Function
0, System timer	GATE 0 CLK0 OUT0C	Always on 1.193 MHz (CLKC) IRQ0
1, Refresh request	GATE 1 CLK1 O1	Always on 1.193 MHz (CLKC) Request refresh (REFTIME)
2, Speaker frequency	GATE 2 CLK2 O2	Enable speaker (ENBSPK) 1.193 MHz (CLKC) Speaker signal (SPKFRQ)

Figure 6-1 illustrates the internal registers of a counter. The figure shows the status register, the actual counter, count registers (CR), output latches (OL). The subscripts M and L stand for "most significant byte" and "least significant byte" respectively.

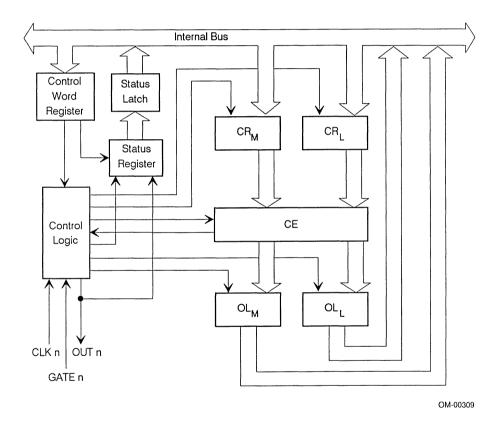


Figure 6-1. Internal Block Diagram of a Counter

6.2.1 Control and Status Registers

The control register stores the mode and command information for each of the counters. Writing a byte to address (043H) loads the control register. The byte contains a pointer to the desired counter, the type of command, and information about the count format.

System software uses a status register to monitor the counters and read back the contents of the control register.

6.2.2 Counting Element (CE)

The counting element is a 16-bit synchronous down counter. Writing one or two bytes in the counter input loads the CE. The CE loads or decrements on the falling edge of the CLKC signal. The CE contains the maximum count when loading 0 and wraps around to FFFFH in binary operation.

6.3 PROGRAMMING THE PIT

Programming the PIT requires writing a control word and initial counts to the three PIT counters at power-up. Register F receives the control word while each counter receives the initial count at the addresses listed in Table 6-2.

Table 6-2. Counter/Timer Address Map

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

6.3.1 Control Register (043H)

Programming a counter requires writing control words to control register 043H. Control words specify the counter, command, mode, and numeric format (BCD or binary). Control register 043H is write-only.

Bits 7 through 4 of this register indicate the counter and the command to be executed. Bits 3 through 1 of this register select the mode of operation as defined in Section 6.1. Bit 0 indicates the countdown format. Table 6-3 lists the contents of the fields and the associated commands.

Table 6-3. Control Register Bit Definition

Bit	Function	
Bits 7:4 0000 0001 0010 0011	Command to be executed Latch counter 0 Read/write counter 0, LSB only Read/write counter 0, MSB only Read/write counter 0, LSB then MSB	
0100 0101 0110 0111	Latch counter 1 Read/write counter 1, LSB only Read/write counter 1, MSB only Read/write counter 1, LSB then MSB	
1000 1001 1010 1011	Latch counter 2 Read/write counter 2, LSB only Read/write counter 2, MSB only Read/write counter 2, LSB then MSB	
11xx	Read-back command	
Bits 3:1 000 001 x10 x11 100 101	Operating mode selection (These bits are "don't care" during the latch counter command.) Mode 0 Mode 1 Mode 2 Mode 3 Mode 4 Mode 5	
Bit 0 0 1	Binary or BCD count down format (This bit must be 0 during read/write command.) Binary (16-bit) count down BCD count down (4 decades)	

READ/WRITE COUNTER COMMAND

Observe the following conventions when loading a counter using the read/write counter command:

- Each counter control word must be written before loading the initial count.
- Writing an initial count must follow the format specified in the control word.
 Load either the least significant byte (LSB) only, most significant byte (MSB) only, or load the LSB then MSB.
- When writing the LSB and MSB, take care to avoid relinquishing control.

A new initial count can be written into the counter at any time after programming. Writing a new initial count does not require rewriting the control word as long as the programmed format is observed.

When reading a counter, the following two conventions must be observed:

- The count must be latched.
- When reading the LSB and MSB, control must not be relinquished between modes.

LATCH-COUNTER COMMAND

When issuing a latch counter command, the counter output latches (COL and COH) latch the current state of the counting element. COL and COH remain latched until read by the CPU or until the counter is reprogrammed. Once read or reprogrammed, COL and COH return to the value currently in the counting element.

Latch-counter commands can be issued to more than one counter before reading the first counter. Multiple latch-counter commands issued to the same counter without reading the counter are ignored, except for the first command.

READ-BACK COMMAND

The read-back command checks the count value, programmed mode, and the current state of the OUT signal and NULL count flag of the selected counters. Latched status bytes (OUT and NULL) remain until the counter is read or reprogrammed. Table 6-4 lists the bit values and function for the read-back command byte.

Table 6-4. Read-back Command Format

Bit/Value	Function
Bits 7:6 Bit 7= 1 Bit 6= 1	Read-back command specified Specifies the read-back command. Bits 7 and 6 are always set to 1.
Bits 5:4 Bit 5 = 0 Bit 4 = 0	Latch counting element/status (LC and LS) If LS and LC = 0, status returns on the first read from the counter. The next one or two reads from the counter results in the count being returned. Latch the state of the counting element in COL and COH. Latch the status of selected counters into the status register.
Bits 3:0 Bit 3 = 1 Bit 2 = 1 Bit 1 = 1 Bit 0 = 0	Counter selection (C2:C0) Select counter 2. Select counter 1. Select counter 0. Must be zero.

Table 6-5 lists the bit values and function for the latched status byte.

Table 6-5. Format of a Latched Status Byte

Bit/Value	Function
Bit 7 0 1	OUT signal state OUT signal 0 (low) OUT signal 1 (high)
Bit 6 0 1	NULL count flag condition Counter loaded from the counter input registers, count can be read. Write to the control register or the counter, but the new value has not been loaded into CE.
Bits 5:4 00 01 10 11	Byte transfer during read/write commands Reserved R/W least-significant byte R/W most-significant byte R/W least-significant byte then most-significant byte
Bits 3:1 000 001 010 011 100	Operating mode selection Mode 0 selected Mode 1 selected Mode 2 selected Mode 3 selected Mode 4 selected Mode 5 selected
Bit 0 0 1	Binary or BCD count down format Binary (16-bit) count down BCD count down (4 decades)

8259 Programmable Interrupt Controllers

7.1 INTRODUCTION

The system interrupt controllers (U32 and U45) are Intel 82C59 programmable interrupt controllers (PICs) (Figure 7-1 diagrams a PIC). U32 is located at addresses 20H and 21H and configured for master operation. U45 is a slave device located at addresses A0H and A1H. The interrupt request output signal from U32 connects to the interrupt request input channel 2 (IR2) of the PIC master (see Figure 7-2).

U45 has seven prioritized interrupt levels and the PIC slave has eight. The PIC minimizes the software and real time overhead in handling multi-level priority interrupts.

The PIC functions as an overall manager in an interrupt-driven system environment. It accepts incoming interrupt requests from the various peripherals attached to the system unit. The PIC checks the interrupt requests for priority, determines whether the incoming request has a higher priority value than the current level, then sends an INTR pulse to the CPU. The CPU acknowledges the INT request with an INTA signal.

Configuring a variety of priority assignment modes any time during system operation structures the system interrupt (based on the system environment).

Each peripheral device has a special program associated with its specific functional or operational requirements called an "Interrupt Service Routine." The PIC, after receiving an interrupt request from the peripheral device, sends the address information to the CPU. This forces the program counter to the starting interrupt vector address. Interrupt vector addresses are stored in a table in low memory (locations 0 to 1024).

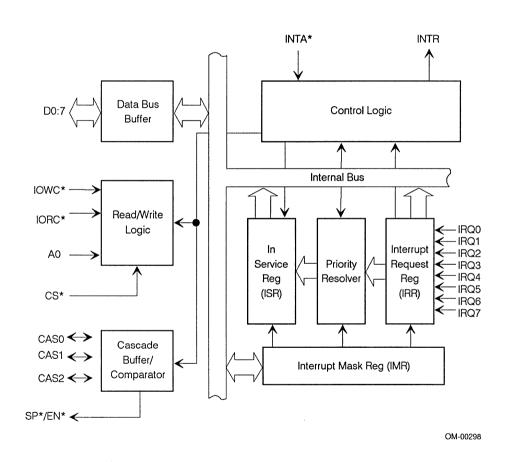


Figure 7-1. PIC Block Diagram

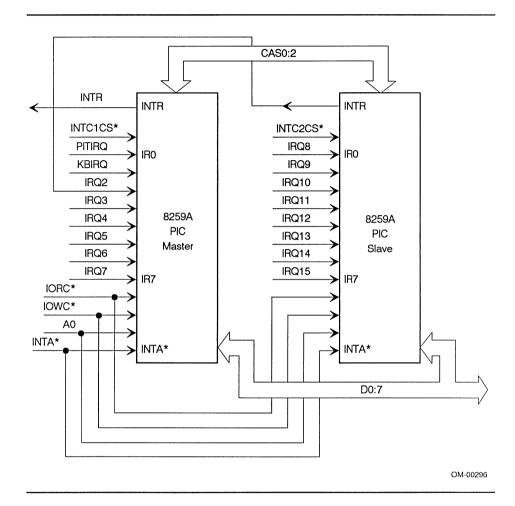


Figure 7-2. Maskable Interrupt Logic

7.2 INTERRUPT CONTROLLER ARCHITECTURE

Each PIC contains an interrupt request register, an in-service register, an interrupt mask register, and a priority resolver.

7.2.1 Interrupt Request Register (IRR) and In-service Register (ISR)

The IRR and ISR handle the interrupts at the input lines. The IRR stores interrupt levels which are requesting service and the ISR stores interrupt levels currently being serviced.

7.2.2 Interrupt Mask Register

Each interrupt line can be individually masked by the interrupt mask register (IMR). Disabling an interrupt line prevents the IMR from generating an interrupt. Masking a higher priority input does not affect the interrupt request lines of lower priority.

7.2.3 Priority Resolver

The priority resolver determines the priorities of the bits set in the IRR. The priority resolver, following the first INTA pulse from the CPU, selects the highest priority and strobes it into the corresponding bit of the ISR.

7.3 INTERRUPTS

The system module utilizes the following four interrupt types:

- Maskable interrupts
- Nonmaskable interrupts
- Hardware interrupts
- Software interrupts

The CPU executes instructions which allow software to enable or disable the maskable interrupt (INTR). External hardware on the 302 board, under software control, masks the nonmaskable interrupt (NMI). The NMI cannot be masked off within the CPU. The NMI mask register appears at I/O address 70H. The NMI mask register shares address 70H with the real-time clock and the configuration memory device (lower six bits).

7.3.1 Maskable Interrupts

All maskable hardware interrupts to the CPU are processed by the PICs. These devices generate interrupts on the CPU interrupt line (INTR). These interrupts can be masked in the CPU by using the clear (CLI) instruction. Any or all interrupts can be masked using the interrupt mask register (IMR). However, the interrupt vector of the PIC must be initialized in advance.

Fifteen interrupt levels are available by cascading the two interrupt controllers. The slave controller signals the master to cause an interrupt. The base I/O addresses are: INTC1 (master) 20H and INTC2 (slave) A0H.

Table 7-1 lists each interrupt priority level, the PIC that monitors that level, the name of the interrupt, and the source of the interrupt.

Table 7-1. Interrupt Levels

Priority	PIC No.	Int No.	Interrupt Source	
1	†	NMI	Parity error detected	
2	1	IRQ0	Interval timer (PIT), counter 0 output	
3	1	IRQ1	Full keyboard output buffer	
	1	IRQ2	Interrupt from controller 2 (cascade)	
4	2	IRQ8	Real-time clock INT	
5	2	IRQ9	Software redirected to INT 0AH (IRQ2)	
6	2	IRQ10	Reserved	
7	2	IRQ11	Reserved	
8	2	IRQ12	Auxiliary device	
9	2	IRQ13	INT from coprocessor	
10	2	IRQ14	Fixed disk controller	
11	2	IRQ15	Reserved	
12	1	IRQ3	COM2	
13	1	IRQ4	COM1 (primary)	
14	1	IRQ5	LPT2	
15	1	IRQ6	Floppy disk controller	
16	1	IRQ7	LPT1 (primary)	

[†] I/O address 70H, bit 7, controls the NMI signal.

There is always the possibility that more than two interrupts will demand servicing at the same time. The PICs determine the priority of each interrupt and process the requests one at a time by transferring the control of the CPU to the higher priority service routine first.

7.3.2 Non-maskable Interrupts

Non-maskable interrupts (NMI) are caused by:

- Detection of parity error during a memory read on either the 302 board or AT32 bus memory boards.
- Detection of parity errors on expansion boards or on any 8- or 16-bit board that pulls the IOCHCK* line low.
- Detection of a software interrupt to the NMI routine (BIOS call).

Enable or disable the NMI interrupt by:

- Clearing mask flip-flop: OUT 70H, 00H. Executing the instruction enables the NMI.
- Setting mask flip-flop: OUT 70H, 80H. Executing the instruction disables the NMI.

At power-on, the NMI is disabled until system software executes the clear mask instruction.

7.4 PROGRAMMING THE PIC

Each PIC is programmed using initialization command words (ICWs) and operation command words (OCWs). The ICWs bring the PIC to a starting point before beginning normal system operation. The OCWs tells the PIC which interrupt mode to operate in. Table 7-2 lists the I/O addresses and data for the two interrupt controllers.

Table 7-2. Interrupt Controller I/O Address and I/O Data

Interrupt Controller	I/O Port	Read Data	Write Data
INTC1 Master (IRQ0-IRQ7)	IRR 0020H	ICW1 ISR	OCW2 OCW3
	0021H	ICW3 IMR OCW1	ICW2
INTC2 Slave	IRR 00A0H	ICW1 ISR	OCW2 OCW3
(IRQ8-IRQ15)	00A1H	ICW3 IMR OCW1	ICW2

7.4.1 Initialization Command Word

The initialization command word (ICW) initializes the system upon power-on. Before normal operation begins, the PICs must be brought to a starting point by writing a sequence of four bytes to each controller. Refer to Figures 7-3 and 7-4 for a description of the formats of ICWs. Figure 7-5 shows the initialization sequence of the ICWs.

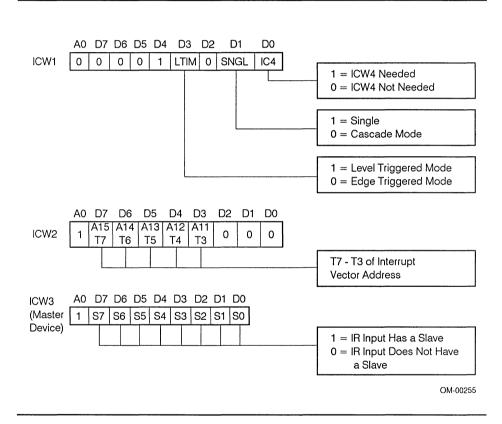


Figure 7-3. Command Word Format (ICW1, ICW2, and ICW3)

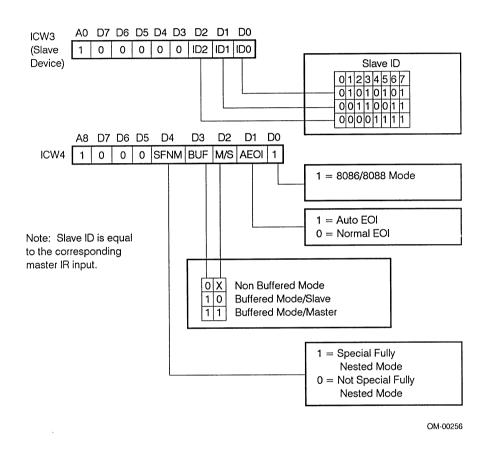


Figure 7-4. Initialization Command Word Format (ICW3 and ICW4)

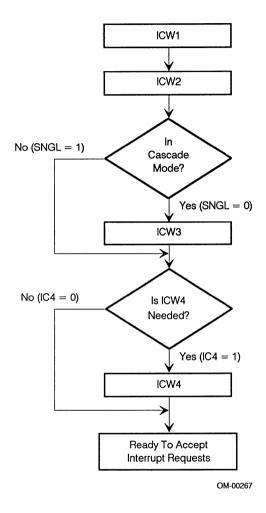


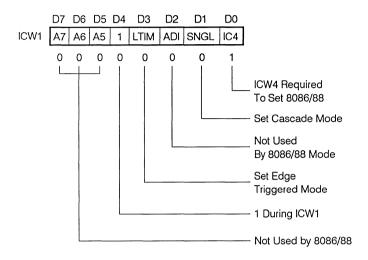
Figure 7-5. Initialization Sequence Diagram

The initialization sequence starts by writing byte ICW1 to address 020H (0A0H) with bit 4 set to 1. The interrupt controller interprets this as the start of an initialization sequence and does the following:

- 1. Resets the ICW counter to zero
- 2. Selects the fixed priority mode
- 3. Assigns the highest priority to IR0
- 4. Clears the interrupt mask register and in-service register
- 5. Sets the slave mode address to 7
- 6. Disables the special mask mode
- 7. Selects the interrupt request register for status read operations.

The next three I/O writes to address 021H (0A1H) load bytes ICW(2:4). Figures 7-6 and 7-7 illustrate the initialization formats from the power-on test routine.

OUT 020H AL AL = 11H (Master 8259A) OUT 0A0H AL AL = 11H (Slave 8259A)



OUT 021H, AL AL = 08H (Master 8259A) OUT 0A1H, AL AL = 70H (Slave 8259A)

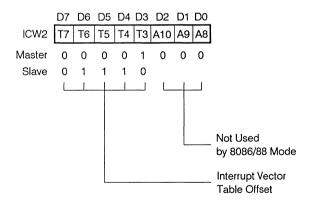


Figure 7-6. Power-on Formats (ICW1 and ICW2)

OM-00257

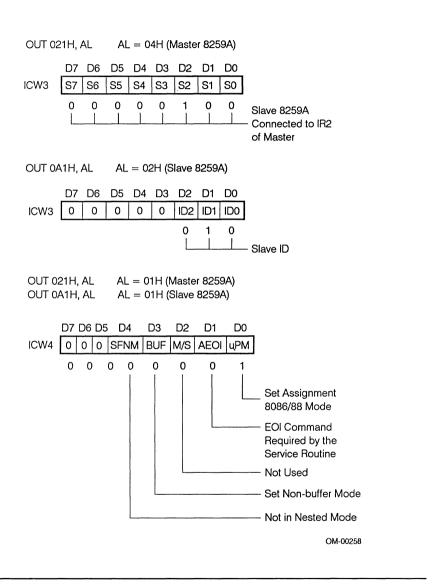


Figure 7-7. Power-on Formats (ICW3 and ICW4)

7.4.2 Operational Command Word

Operation command words (OCWs) command the PICs to operate in various interrupt PIC modes. These modes are: fully nested, rotating priority, special mask, and polled. Each PIC has three OCWs which can be programmed to change the configuration and to monitor controller operation.

OCW1 can be written to address 021H (0A1H) any time the controller is not in initialization mode. OCW2 and OCW3 are written to address 020H (0A0H). Writing to address 020H (0A0H), with bit 4 set to 0, places the controller in operational mode and loads OCW2 (if data bit 3=0) or OCW3 (if data bit 3=1). Figures 7-8 and 7-9 show the OCW instruction sequence.

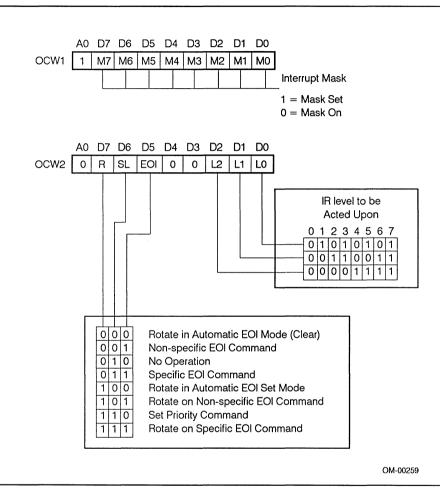


Figure 7-8. Command Word Format (OCW1 and OCW2)

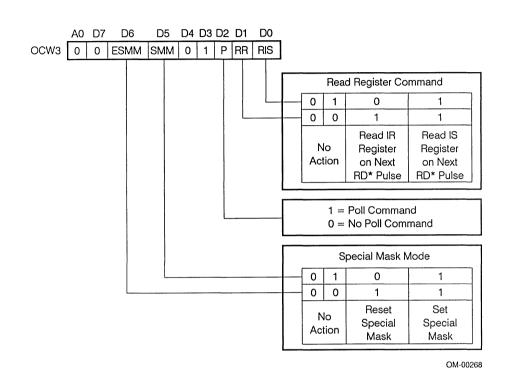


Figure 7-9. Command Word Format (OCW3)

Direct Memory Access (DMA)

8.1 INTRODUCTION

Direct memory access (DMA) is a process where data is transferred between I/O devices (peripherals) and system DRAM without the direct participation of the CPU. The CPU establishes the parameters for the data transfer, but the DMA controllers perform the actual data transfers. This leaves the CPU free for other tasks. Figure 8-1 illustrates the major functional blocks of a DMA controller.

The 302 board uses two DMA controllers (DMA1 and DMA2). Each DMA controller has four DMA channels that are independently programmable, and each controller generates the memory addresses and control signals necessary to transfer information. The controllers are cascaded together to provide seven DMA channels: four channels for transfers to 8-bit peripherals (DMA1) and three channels for transfers to 16-bit peripherals (DMA2). DMA2 port 0 provides the cascade interconnection for the two DMA controllers (see Figure 8-2).

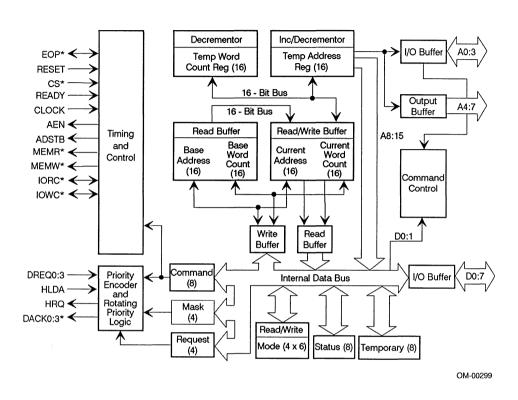


Figure 8-1. DMA Controller Block Diagram

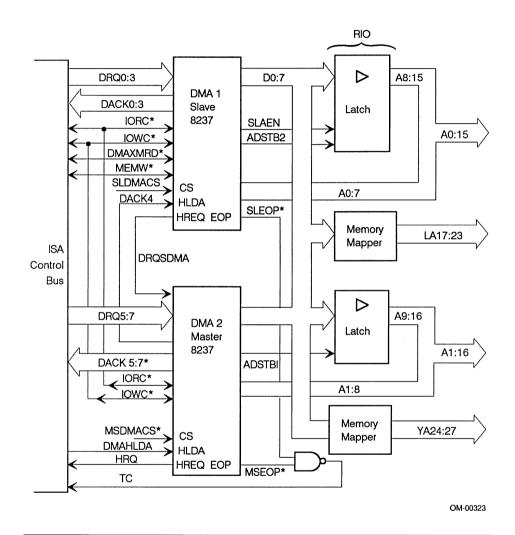


Figure 8-2. DMA Logic

8.2 DMA OPERATION

The DMA controller and I/O devices use the DMA request (DRQ) and DMA acknowledge (DACK*) signals for handshaking. An I/O device activates the DRQ line when requesting an 8- or 16-bit data transfer. The I/O device asserts the byte or data word on the data bus after receiving an active DACK* signal from the DMA controller. During normal operation, the DMA controller can be in an idle cycle, program condition, or active cycle.

8.2.1 Idle Cycle

The DMA controller enters the idle cycle when no I/O device requests service. During the idle cycle, the DMA controller samples the DRQ input pins at a frequency of 4.0 MHz. The DMA controller also samples the chip select (CS*) signal and determines if the CPU is attempting to access the DMA controller registers. If the CPU is attempting an access, the controller remains in the idle cycle. If a DRQ signal occurs on an unmasked channel, the DMA controller exits the idle cycle.

8.2.2 Active Cycle

The DMA controller enters the active cycle when a DRQ signal occurs on an unmasked channel. During an active cycle, the DMA controller operates in one of the following four modes:

- Single-transfer mode
- Block-transfer mode
- Demand-transfer mode
- Cascade mode

SINGLE-TRANSFER MODE

During single-transfer mode, the DMA controller makes one transfer. After each transfer, the current word count decrements, and the current address either decrements or increments. When the current word count changes from 0 to FFFFH, TC is generated and auto-initialization occurs. Auto-initialization, enabled by a bit in the mode register, restores channels to their original condition.

An I/O device must hold the DRQ signal active until a corresponding DACK* signal becomes active. If the I/O device holds the DRQ signal active throughout the single transfer, the hold request (HRQ) signal goes inactive for one full cycle, before it is reasserted. This gives the CPU an opportunity to execute a bus access between successive DMA transfers. Another single transfer begins when the DMA controller receives a new HLDA signal.

BLOCK-TRANSFER MODE

During block-transfer mode, a DRQ signal activates the DMA controller. The DRQ signal remains active during an active DACK* signal. The controller continually makes block transfers during the service until it encounters a TC signal, caused by the word count going to FFFFH, or an external end of process (EOP*) signal. If the DMA channel is programmed for auto-initialization, an auto-initialization occurs at the end of the block-transfer service.

DEMAND-TRANSFER MODE

During demand-transfer mode, the DMA controller continually makes DMA transfers until it encounters a TC signal, an external EOP* signal, or when the current DRQ signal goes inactive. An inactive DRQ signal, before the last working state of the cycle, prevents another transfer. Current address registers store intermediate values of addresses between transfers. Similarly, current word count registers store intermediate values of word counts between transfers. An EOP* signal causes an auto-initialization at the end of the DMA controller service

CASCADE MODE

During cascade mode, both DMA controllers are cascaded together for system expansion. The HRQ and HLDA signals from the slave DMA controller connect to the DRQ and DACK* signals of the master DMA controller. This allows the master controller to request the slave controller to propagate through the priority network circuitry of the master device. The cascade channel of the master controller prioritizes the slave controller and does not output any address or control signals of its own. The DMA controller responds only to signals DRQ and DACK*. All other outputs, except the HRQ signal, are disabled.

8.3 Transfer Types

Each of the four transfer modes perform three types of transfers: read, write, and verify.

Read Transfer A read transfer moves data from memory to an I/O

device by generating the memory address and asserting the MEMR* and IOW* signals during the same cycle.

Write Transfer A write transfer moves data from an I/O device to

memory by generating the memory address and

asserting the MEMW* and IOR* signals during the same

cycle.

Verify Transfer A verify transfer is used for diagnostics. In this type of

transfer, the DMA operates as if it is performing a read or write transfer by generating a memory address, signal HRQ, and signal DACK*. However, memory and I/O

control lines remain inactive.

8.4 DMA CHANNELS

Each DMA controller has four channels. DMA1 supports channels zero through three (for 8-bit transfers), and DMA2 supports channels four through seven (for 16-bit transfers). Channel four is used to cascade from DMA1 to DMA2. All HOLD requests for DMA1 are processed via DMA2, channel four. This forces all channels in DMA1 to operate at a higher priority than those in DMA2. Channel zero has the highest priority and channel seven the lowest. DMA channels 0-3 are 8-bit channels and DMA channels 5-7 are 16-bit channels. Table 8-1 lists each channel's function.

Table 8-1. DMA Channel Assignment

Channel	Controller	Function
0 1 2 3 4 5 6 7	1 1 1 2 2 2 2	Refresh Streaming tape (typical) Disk (floppy) Spare Cascade Spare Spare Spare Spare

The system assigns an upper- and lower-page register to each DMA channel. The page registers supply the top 12 bits of each address for DMA1 (8-bit transfers) and the top 11 bits of each address for DMA2 (16-bit transfers). The DMA controller supplies the bottom 16 bits of each address. The page registers are implemented external to the DMA controllers.

The DMA controllers and page registers provide 28 bits of address for DMA1 and 27 bits of the address for DMA2. Generated addresses do not cross page boundaries (64K for channels 0 to 3, and 128K for channels 5 to 7).

The DMA controllers and page registers must be initialized before beginning a DMA transfer. Initialization consists of loading the starting address, the number of bytes, and the direction of transfer. The CPU reads and writes the DMA controllers' internal registers when the controller is selected while the CPU controls the bus. All registers, including those not used, are written to by the system BIOS at power-up.

8.5 PROGRAMMING THE DMA CONTROLLERS

The controllers can be programmed any time except when the HLDA signal is asserted. The CPU ensures that no DMA activity occurs on the channel being programmed. To prevent a conflict, disable the DMA controller or mask the DMA channel before programming any registers. Disabling external interrupts protects the sections of code that set up the registers. This prevents another routine from changing the state of the circuit.

8.6 DMA INTERNAL REGISTERS

The DMA internal registers control DMA transfers. Table 8-2 lists all DMA internal registers and their sizes.

Table 8-2. DMA Internal Registers

Register Name	Size	Number
Base-address registers	16 bits	4
Base-word-count registers	16 bits	4
Current-address registers	16 bits	4
Current-word-count registers	16 bits	4
Temporary-address register	16 bits	1
Temporary-word-count register	16 bits	1
Status register	8 bits	1
Command register	8 bits	1
Temporary register	8 bits	1
Mode registers	6 bits	4
Mask register	4 bits	1
Request register	4 bits	1

8.6.1 Address and Count Registers

Address registers and count registers specify the address and length of the transfer. These registers are 16 bits, requiring two successive eight-bit read or write operations. The byte pointer flip-flop must be cleared before accessing these registers. This guarantees that the first operation transfers the low-order byte and the second operation transfers the high-order byte. After clearing the byte flip-flop, the program must retain control until both bytes are written or read.

Table 8-3 lists the I/O port address assignments for the DMA address and count registers.

Table 8-3. I/O Port Addresses for DMA Address and Count Registers

Controller	I/O Address	Command Codes
	000H	CH0 base and current address
	001H	CH0 base and current word count
	002H	CH1 base and current address
DMA1	003H	CH1 base and current word count
	004H	CH2 base and current address
	005H	CH2 base and current word count
	006H	CH3 base and current address
	007H	CH3 base and current word count
:	0C0H	CH4 base and current address
	0C2H	CH4 base and current word count
	0C4H	CH5 base and current address
DMA2	0C6H	CH5 base and current word count
	0C8H	CH6 base and current address
	0CAH	CH6 base and current word count
	0CCH	CH7 base and current address
	0CEH	CH7 base and current word count

CURRENT ADDRESS REGISTER

Each DMA channel has a 16-bit current-address register for storing the value of the address used during DMA transfers. The address automatically increments or decrements after each transfer, and the intermediate values of the address are stored in the register during the transfer. The CPU reads or writes the register in successive bytes while in the program state. During auto-initialize, the original values of the current-address register are automatically restored from the base count register after the EOP* signal goes active. An I/O device or TC generates EOP*. Note that EOP* is not available as an input of the 302 board due to external gating and no ISA bus signal dedicated to this function.

CURRENT WORD-COUNT REGISTER

Each DMA channel has a 16-bit current-word-count register for determining the number of transfers performed. The actual number of transfers is one more than the number programmed in the register (programming a count of 25 results in 26 transfers). The register loads or reads in successive bytes by the CPU in the program condition. The word count decrements after each transfer until the value in the register goes from zero to FFFFH. At FFFFH, the TC is generated and auto-initialization occurs. During auto-initialization, the original values of the current-address register are automatically restored from the base-count register after the EOP* signal is asserted. An I/O device or TC generates EOP*. If it is not auto-initialized, the register will have a count of FFFFH after TC.

BASE ADDRESS AND BASE WORD-COUNT REGISTERS

Each DMA channel has a pair of base-address registers and base-word-count registers. These 16-bit registers are programmed with the desired address and word count. During auto-initialization, these values are transferred to the current-address registers and current-word-count registers. The base registers are automatically loaded when corresponding current registers are written. The base registers cannot be read by the CPU.

8.6.2 Program-control Registers

Program-control registers assign DMA transfer operations. Table 8-4 lists the I/O port addresses for the control registers.

Table 8-4. Control Register I/O Port Addresses

Add DMAC1	ress DMAC2	Operation	Command Codes
008H 008H 009H 00AH 00BH 00CH 00DH 00DH 00EH 00FH	ODOH ODOH OD2H OD4H OD6H OD8H ODAH ODAH ODAH ODCH ODEH	Read Write Write Write Write Write Read Write Write Write	Read status register Write command register Write request register Write single-mask register bit Write mode register Clear byte-pointer flip-flop Read temporary register Master-clear register Clear mask register Write all mask register bits

8.6.3 Command Register

The 8-bit command register controls the overall operation of the DMA subsystem. The CPU programs the command register during a program condition. A reset or a master clear instruction clears the command register. Table 8-5 shows the bit assignments for the command register.

Table 8-5. Command Register Bit Assignments

Bit/Value	Function
Bit 7 1 0	DACK* polarity configuration bit DACK* active-high DACK* active low (default)
Bit 6	DRQ polarity configuration bit
1	DRQ active low
0	DRQ active high (default)
Bit 5 1 0	Extended write bit Extended write selected Late write selected (default)
Bit 4	Priority scheme select bit
1	Rotating-priority scheme is used
0	Fixed-priority scheme is used (default)
Bit 3	Compressed timing bit
1	Compressed timing is enabled
0	Normal timing is used (default)
Bit 2	Enable/disable bit
1	Controller is disabled
0	Controller is enabled (default)
Bit 1	Channel address hold-enable bit
1	CH0 address hold is enabled
0	CH0 address hold is disabled (default)
Bit 0	Memory-to-memory enable bit
1	Memory-to-memory is enabled for CH0 and CH1
0	Memory-to-memory is disabled (default)

8.6.4 Mode Register

Each DMA channel has an associated 6-bit mode register. The mode registers specify the channel operating mode. When the CPU writes to a mode register during a program state, bits 0 and 1 specify the channel selected. Table 8-6 lists the bit assignments for the mode register.

Table 8-6. Mode Register Bit Assignments

Bit/Value	Function
Bits 7:6 00 01 10 11	Mode selection bits Demand mode is selected Single mode is selected Block mode is selected Cascade mode is selected
Bit 5 1 0	Address increment/decrement bit Address decrement is selected Address increment is selected
Bit 4 1 0	Auto-initialization enable bit Auto-initialization is enabled Auto-initialization is disabled
Bits 3:2 00 01 10 11	Transfer bits Verify transfer Write transfer Read transfer Illegal
Bits 1:0 00 01 10 11	Channel select bits CH 0 is selected CH 1 is selected CH 2 is selected CH 3 is selected

8.6.5 Request Register

Each DMA controller has an associated 4-bit request register. Each channel has a request bit associated with it in the 4-bit request register. The system software sets or resets each request-register bit separately to request use of the corresponding DMA channel. A TC or EOP* signal clears each request-register bit. A system reset clears the entire request register. The DMA channel must be in the block transfer mode and the appropriate registers must be set before initiating this request. Table 8-7 lists the format for writing to the request register.

Table 8-7. Request Register Update

Bit/Value	Function
Bits 7:3	Not used
Bit 2 1 0	Set/reset request bit Request bit is set Request bit is reset
Bits 1:0 00 01 10 11	Channel selection bits CH 0 is selected CH 1 is selected CH 2 is selected CH 3 is selected

8.6.6 Mask Register

Each DMA channel has an associated mask bit. The mask bit disables incoming DRQ signals. Each mask register bit is set when the associated DMA channel activates the EOP* signal, if auto-initialization is not programmed in the DMA channel. The system software sets or clears each mask-register bit separately. A system reset sets the entire mask register and disables all DMA requests.

Two commands are associated with the mask register: Write single mask bit and Write all mask bits. Tables 8-8 and 8-9 list the bit positions for both commands.

Table 8-8. Write Single Mask Bit Assignments

Bit/Value	Function
Bits 7:3	Not used
Bit 2 1 0	Set/reset mask bit Set mask bit Reset mask bit
Bits 1:0 00 01 10 11	Select channel mask bits CH 0 mask bit selected CH 1 mask bit selected CH 2 mask bit selected CH 3 mask bit selected

Table 8-9. Write All Mask Bit Assignments

Bit/Value	Function
Bits 7:4	Not used
Bit 3	Set/reset channel 3 mask bit
1	Set CH 3 mask bit
0	Reset CH 3 mask bit
Bit 2	Set/reset channel 2 mask bit
1	Set CH 2 mask bit
0	Reset CH 2 mask bit
Bit 1	Set/reset channel 1 mask bit
1	Set CH 1 mask bit
0	Reset CH 1 mask bit
Bit 0	Set/reset channel 0 mask bit
1	Set CH 0 mask bit
0	Reset CH 0 mask bit

8.6.7 Status Register

The status register contains the status of the DMA controller at the time of readout. The status information tells which channel has reached the terminal count and which channel has a DRQ signal pending. Bits 3 through 0 are set every time the corresponding channel reaches the terminal count or when a DMA channel activates the EOP* signal. Reading or resetting the status register clears the status-register bits. Status register bits 7 through 4 are set when the corresponding DMA channel requests service. Table 8-10 lists the status register bit positions.

Table 8-10. Status Register Bit Assignments

Bit/Value	Function
Bit 7	Channel 3 DMA request bit
1	Channel 3 DMA requested
0	No DMA request
Bit 6	Channel 2 DMA request bit
1	Channel 2 DMA requested
0	No DMA request
Bit 5	Channel 1 DMA request bit
1	Channel 1 DMA requested
0	No DMA request
Bit 4	Channel 0 DMA request bit
1	Channel 0 DMA requested
0	No DMA request
Bit 3	Channel 3 terminal-count-reached bit
1	Channel 3 DMA process completed
0	Status register is read or reset
Bit 2	Channel 2 terminal-count-reached bit
1	Channel 2 DMA process completed
0	Status register is read or reset
Bit 1	Channel 1 terminal-count-reached bit
1	Channel 1 DMA process completed
0	Status register is read or reset
Bit 0	Channel 0 terminal-count-reached bit
1	Channel 0 DMA process completed
0	Status register is read or reset

8.6.8 Temporary Register

The temporary register is used to hold data during memory-to-memory transfers. Following completion of the transfers, the last word moved can be read by the CPU in the program condition. The temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a reset.

8.6.9 Non-register Program Controls

Special software commands are executed by reading or writing to the addresses specified in the program-control-register I/O. The software commands do not depend on any specific bit pattern on the data bus. For example, accessing the I/O port with a write of any value executes the function. The three software commands are: clear-byte pointer flip-flop, master clear, and clear-request mask register.

Clear-Byte Pointer

Flip-flop

This command resets (clears) the byte pointer flip-flop so the next CPU access to the register addresses the least significant byte. This command must be executed before reading or writing new address or word count information.

Master Clear

This command resets the DMA controller and has the same effect as a hardware reset. During a master clear, the command, status, request, temporary, and internal first/last flip/flop registers reset (clear) and the mask register sets. The DMA controller enters the idle condition after a master clear.

Clear-Request Mask Register This command resets (clears) the mask bits of all four

DMA channels.

8.7 PAGE REGISTERS

The system assigns each channel a 4-bit upper-page register and an 8-bit lower-page register (8-bit DMA). The upper- and lower-page registers extend the address space for the 16-bit DMA controller to 28 bits.

Writing to the DMA controller's lower-page register or current-address counter sets the upper-page register to zero, hence, the lower-page register must be programmed before the upper-page register is programmed.

When a secondary requesting agent has control of the bus, it drives address lines 23 through 0, the upper-page register drives its addresses on the bus, and the lower-page register remains disabled. During a refresh cycle, both the upper- and lower-page registers are active only while BUSALE is asserted.

Tables 8-11 and 8-12 list the I/O addresses for the lower- and upper-page registers.

Table 8-11. Lower-page Register Addresses for Each Channel

I/O Address	Function (In/Out)
081H	8-bit DMA channel 2 (DACK*2)
082H	8-bit DMA channel 3 (DACK*3)
083H	8-bit DMA channel 1 (DACK*1)
087H	8-bit DMA channel 0 (DACK*0)
089H	16-bit DMA channel 6 (DACK*6)
HA80	16-bit DMA channel 7 (DACK*7)
08BH	16-bit DMA channel 5 (DACK*5)
08FH	Refresh-cycle-page register

Table 8-12. Upper-page Register Addresses for Each Channel

I/O Address	Function (In/Out)
481H	8-bit DMA channel 2 (DACK*2)
482H	8-bit DMA channel 3 (DACK*3)
483H	8-bit DMA channel 1 (DACK*1)
487H	8-bit DMA channel 0 (DACK*0)
489H	16-bit DMA channel 6 (DACK*6)
48AH	16-bit DMA channel 7 (DACK*7)
48BH	16-bit DMA channel 5 (DACK*5)
48FH	Refresh-cycle-page register

	•		

1287 Real-time Clock (RTC)

9.1 INTRODUCTION

The board uses a 1287 real-time clock (RTC) module as its real-time clock and configuration memory. The RTC module combines a complete time-of-day clock with alarm, 100-year calendar, a programmable periodic interrupt, 50 bytes of low-power SRAM to store system configuration information, and a battery. System provisions allow the RTC to operate in a low-power mode and protect the contents of both the RAM and clock during system power-up and power-down. The battery maintains clock and calendar information in the RAM. The system does not charge the battery. If the battery fails, the real-time clock chip must be replaced. Figure 9-1 illustrates the RTC memory map. An additional 14 bytes of CMOS RAM is used for the internal clock circuitry.

9.2 RTC RAM I/O OPERATIONS

The RTC maintains the system time and date. The RTC updates the system time at one-second intervals and automatically adjusts at the end of months and leap years.

Writing the appropriate index address to I/O port 70H allows reading and writing to the 64 locations in the RTC. The RTC address register latches the address and points to the specified byte in the RTC.

Values can be written to or read from all 64 bytes except for the following, which are read only:

- Status registers C and D
- High-order bit (bit 7) of status register A
- High-order bit (bit 7) of the seconds register

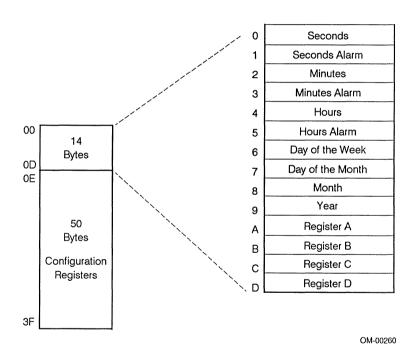


Figure 9-1. RTC Memory Map

Perform the following two steps when writing data into the RTC/RAM:

- Write the RAM address (data = 00H through 3FH) into I/O port address 70H.
- Write the data byte into I/O port 71H.

Perform the following two steps when reading data from the RTC/RAM:

- Write the RAM address (data = 00H through 3FH) into I/O port address 70H.
- 2. Read the data byte from I/O port 71H.



I/O port address 70H is also an output port for the NMI mask. Data bus bit 7 connects to the NMI mask bit and bits 0 through 5 to the RTC/RAM address lines. To protect the RTC from an accidental write to port 71H set 80H to point to status register C or D.

During normal operation, the RTC performs an update cycle every second. Divider bits DV(2:0) and the set bit in register B determine the performance of an update cycle. Divider bits DV(2:0) must be set to a binary value of 010 and the register B set bit must be cleared. During an update, the lower ten registers are not available to the CPU. The update cycle increments the clock/calendar registers and compares them to the alarm registers. An interrupt is issued to the CPU if a match occurs between the two sets of registers (with alarm and interrupt control bits enabled).

9.3 RTC ADDRESSABLE LOCATIONS

The 64 addressable locations in the RTC are divided into ten bytes containing the time, calendar, alarm data, four control and status bytes, and 50 general purpose RAM bytes as listed in Table 9-1. Table 9-1 also details the internal register/RAM organization of the RTC.

Table 9-1. Real-time Clock Address Map

Function	Index
Time, Calendar, and Alarm Bytes Seconds register Seconds alarm register Minutes register Minutes alarm register Hours register Hours alarm register Day of week register Date of month register Month register Year register	00H 01H 02H 03H 04H 05H 06H 07H 08H
Status Registers Status register A Status register B Status register C Status register D	0AH 0BH 0CH 0DH
General Configuration Bytes Diagnostic status byte Shutdown status byte Floppy disk drive type byte Reserved Fixed disk type byte Reserved Equipment byte Low base memory byte High base memory byte Low expansion memory byte High expansion memory byte Drive C extended type byte Drive D extended type byte Reserved Features installed byte Drive type 48 parameters byte Cache, shadow, and setup byte	0EH 0FH 10H 11H 12H 13H 14H 15H 16H 17H 18H 19H 1AH 1BH-1EH 1FH 20H-27H 28H

(continued)

Table 9-1. Real-time Clock Address Map (continued)

Function	Index	
Reserved 2-byte CMOS RAM checksum byte Low extended memory byte High extended memory byte Date century byte Setup information byte System speed byte	29H-2DH 2EH-2FH 30H 31H 32H 33H	
Drive type 49 parameters byte Reserved	35H-3CH 3DH-3FH	

9.3.1 Time, Calendar, and Alarm Bytes

The CPU obtains time and calendar information by reading the appropriate locations in the RTC. Writing to these locations initializes the time, calendar, and alarm information. Information stored in these locations is in BCD format.

Before initializing the internal bytes, the set bit in byte B must be set to 1 to prevent RTC updates. Once set, the CPU initializes the first ten locations in BCD format and the set bit is cleared.

Once initialized and enabled, the RTC performs clock/calendar updates at a 1 Hz rate. During updates, the ten bytes of time, calendar, and alarm information are not available to be read or written by the CPU for 2 ms. The update in progress (UIP) bit, in status byte A, is set while an update is occurring.



System software sets the real-time clock to BCD data mode.

Table 9-2 lists the format for the clock, calendar, and alarm locations.

Table 9-2. Time, Calendar, and Alarm Data Format

Function	Index	BCD Data
Seconds	00	00:59
Seconds alarm	01	00:59
Minutes	02	00:59
Minute alarm	03	00:59
Hours	04	
(12 hour mode)		01:12 (AM)
(12 Hodi Mode)		81:92 (PM)†
	·	01.52 (FW))
(24 hour mode)		00:23
Hours alarm	05	
(12 hour mode)		01:12 (AM)
(12 Hour Houe)		81:92 (PM) [†]
		61.92 (FIVI) ·
(24 hour mode)		00:23
(=		
Day of week	06	01:07
Date of month	07	01:31
Month	80	01:12
Year	09	00:99

[†] In 12 hour mode the most significant bit of each byte indicates pm when set. For example, 81 BCD (10000001B) indicates 1 pm.



The RTC does not affect the 50 bytes of RAM from index address 0EH to 3FH. These bytes are accessible only during the update cycle.

9.4 STATUS REGISTERS

The four control and status bytes (status bytes A through D) control the operation and monitor the status of the RTC. These bytes, located at index addresses 0AH through 0DH, are accessible by the CPU at all times.



A setup program must initialize status registers A through D when setting the time and date.

9.4.1 Status Register A

Status register A (index 0AH) contains information on the divider selection bits, the rate selection bits and the update in progress bit. Bits 6-0 are read/write. Bit 7 is read-only. Table 9-3 defines register A.

Table 9-3. Status Register A (0AH)

Bit/Value	Function
Bit 7 1 0	Update in progress (UIP) bit The time update cycle will start soon. The current date and time can be accessed.
Bits 6:4	Divider selection DV(2:0) bits These bits turn the oscillator on and off and reset the countdown chain. When the bits are set to 010, the oscillator is turned on allowing the RTC to keep time. When they are set to X11, the oscillator is enabled and the countdown chain is held in reset. (X = don't care)
Bits 3:0	Rate selection RS(3:0) bits These bits select the divider output frequency. The system initializes to 0110, which selects a 1024 Hz divider frequency and an interrupt rate of 976.562 μ s.

9.4.2 Status Register B

Status register B (index 0BH) contains the periodic, alarm, and update-ended interrupt enable bits. The register also contains the set update cycle bit and the square wave enable bits. Table 9-4 defines register B.

Table 9-4. Status Register B (0BH)

Bit/Value	Function
Bit 7 1	Set update cycle (SET) bit Aborts the update cycle in progress. Set to 1 for system initialization.
0	Enables normal update cycle of one count per second.
Bit 6	Periodic interrupt enable (PIE) bit This read/write bit selects an interrupt occurring at the rate specified by the rate and divider selection bits in Status Register A Enables the generation of periodic interrupts
Ö	Disables the interrupt (default)
Bit 5 1 0	Alarm interrupt enable (AIE) bit Enables the alarm interrupt Disables the alarm interrupt (default)
Bit 4 1 0	Update-ended interrupt enable (UIE) bit Enables the update-ended interrupt Disables the update-ended interrupt (default)
Bit 3 1	Square wave enabled (SQWE) bit Enables the square-wave frequency set by the rate selection bits in Status Register A Disables square-wave frequency (default)

(continued)

Table 9-4. Status Register B (continued)

Bit/Value	Function
Bit 2 1 0	Date mode (DM) bit Indicates whether the time and date calendar updates are in binary or BCD format. Selects the binary format Selects the BCD format (default)
Bit 1 1 0	24/12-hour (24/12) bit Determines the format of the hour-byte, either 12-hour or 24-hour mode. Indicates the 24-hour mode set (default) Indicates the 12-hour mode set
Bit 0 1 0	Daylight savings enabled (DSE) bit Enables daylight savings time Disables daylight savings time (default)

9.4.3 Status Register C

Status register C (index 0CH) contains the interrupt request, periodic interrupt, alarm interrupt, and update ended interrupt flags. Table 9-5 defines register C.

Table 9-5. Status Register C (0CH)

Bit	Function
Bit 7 1	IRQF (interrupt request flag) The condition causing the interrupt is true and the interrupt enable for that condition is true. The condition causing the interrupt is false or the interrupt enable for that condition is false
Bit 6 1 0	PF (periodic interrupt flag) This bit becomes active, independent of the condition of the PIE control bit. The PF bit generates an interrupt and sets IRQF if PIE = 1. A transition, selected by RS(3:0), occurred in the divider chain. The transition did not occur
Bit 5 1 0	AF (alarm interrupt flag) This flag is independent of the condition of the AIE, and generates an interrupt if AIE is true. A match occurred between the time bytes and alarm bytes during an update cycle. No match occurred.
Bit 4 1 0	UF (update ended interrupt flag) This flag is also independent of the condition of the UIE, and generates an interrupt if UIE is true. End of update Update in progress
Bits 3:0	Reserved

9.4.4 Status Register D

Status register D (index ODH) contains the valid RAM and time bit. Table 9-6 defines register D.

Table 9-6. Status Register D (0DH)

Bit/Value	Function
Bit 7 1 0	Valid RAM and time (VRT) bit This read-only bit determines the condition of the RTC internal battery. Battery operational A low power sense. A dead battery in the RTC.
Bits 6:0	Reserved

9.5 CONFIGURATION BYTES

The remaining bytes in RTC address space are for general configuration and status. The rest of the chapter discusses these general configuration and status bytes.

9.5.1 Diagnostic Status Byte

The diagnostic status byte (index 0EH) contains the RTC battery power status, checksum status indicator, incorrect configuration information, and the memory size comparison. Table 9-7 defines the byte usage.

Table 9-7. Diagnostic Status Byte (0EH)

Bit/Value	Function
Bit 7 1	Real-time clock chip battery power status Battery is dead or was found dead in the power-on self-test (steady bit) Battery is operational
Bit 6 1 0	Configuration record — checksum status indicator Checksum not valid Checksum valid
Bit 5 1 0	Incorrect configuration information Checks the equipment byte of the configuration record when the system powers up Configuration information not valid Configuration information valid
Bit 4 1 0	Memory size comparison Memory size different from configuration record Memory size the same as configuration record
Bit 3 1 0	Initial state of drive C or fixed disk controller Wrong controller or drive C. System cannot boot from drive C. Correct controller and drive. The system can boot from drive C.
Bit 2 1 0	Time status indicator (post checks) Time not valid Time valid
Bits 1:0	Reserved



In order for the configuration information to be valid, power-on check requires at least one floppy disk drive to be installed (bit 0 of the equipment byte set to 1) and the video display jumper to match with the display controller installed.

9.5.2 Shutdown Status Byte

When the CPU resets, the shutdown status byte (index 0FH) is set. The reset code identifies the type of reset and signals the system what to do after the reset. It also provides a method of resetting the system without losing previously stored data or returning the system to the real mode from protected mode. Table 9-8 defines the byte usage.

Table 9-8. Shutdown Status Byte (0FH)

Value	Function
00H 09H 01H:08H 0AH:FFH	System reset or power up User software reset (return from protected mode) Used by hardware self-test Reserved

9.5.3 Floppy Disk Drive Type Byte

The floppy disk drive type byte (index 10H) contains data indicating the floppy disk drive types installed. Table 9-9 defines the byte usage.

Table 9-9. Floppy Disk Drive Type Byte (10H)

Bit/Value	Function
Bits 7:4 0000 0001 0010 0011 0100 0101:1111	First floppy disk drive type No floppy disk drive 360K drive (5.25-inch) 1.2M high-density drive (5.25-inch) 720K (3.5-inch drive) 1.4M (3.5-inch drive) Reserved
Bits 3:0 0000 0001 0010 0011 0100 0101:1111	Second floppy disk drive type No floppy disk drive 360K drive (5.25-inch) 1.2M high-density drive (5.25-inch) 720K (3.5-inch drive) 1.4M (3.5-inch drive) Reserved

9.5.4 Fixed Disk Drive Type Byte

The fixed disk drive type byte (index 12H) contains data indicating the fixed disk drive types installed. Table 9-10 defines the byte usage.

Table 9-10. Fixed Disk Type Byte (12H)

Bit/Value	Function
Bits 7:4	First fixed disk drive type (drive C)
0000	No fixed disk drive installed
0001:1110	Types 1 - 14
1111	Types 16 - 255 (refer to extended byte 19H)
Bits 3:0	Second fixed disk drive type (drive D)
0000	No fixed disk drive installed
0001:1110	Types 1 - 14
1111	Types 16 - 255 (refer to extended byte 1AH)

9.5.5 Equipment Byte

The equipment byte (index 14H) contains equipment information for use by the power-on self test (POST). Table 9-11 shows how the byte is used.

Table 9-11. Equipment Byte (14H)

Bit/Value	Function
Bit 7:6 00 01	Number of floppy disk drives installed 1 drive
10 11	2 drives Reserved Reserved
Bits 5:4 00 01	Type of video display controller used Extended functionality controller (EGA, VGA, PGA). Color graphic video display controller in the 40-column mode.
10 11	Color graphic video display controller in the 80-column mode. Monochrome display controller.
Bits 3:2	Reserved
Bit 1 1 0	Presence of a numeric coprocessor Numeric coprocessor installed No numeric coprocessor
Bit 0 1 0	Presence of floppy disk drive Floppy disk drive installed No floppy disk drive

9.5.6 Low and High Base Memory Bytes

Low and high base memory bytes (index 15H and 16H) contain the low and high bytes of the base memory size, respectively. These two bytes indicate the base memory size. Table 9-12 defines the byte usage.

Table 9-12. Low and High Base Memory Byte (15H and 16H)

Value	Typical Values	
0100H 0200H	256K RAM 512K RAM	
0280H	640K RAM	

9.5.7 Low and High Memory Expansion Bytes

Low and high memory expansion byte (index 17H and 18H) form a 16-bit value that indicates the total amount of expansion memory (above 1M) set by the system configuration program. Byte index 17H contains the low byte and byte index 18H contains the high byte. Table 9-13 defines byte usage.

Table 9-13. Low and High Memory Expansion Bytes (17H and 18H)

Value	Typical Values	
0200H 0400H 0600H 3C00H	512K RAM expansion 1024K RAM expansion 1536K RAM expansion 15360K RAM expansion	

9.5.8 Drive C Extended Byte

Drive C extended byte (index 19H) contains the drive C extended type byte indicating drive type. Values from 10H to FFH indicate drive types 16 to 255, respectively. All other values for this byte are reserved.

9.5.9 Drive D Extended Byte

Drive D extended byte (index 1AH) contains the drive D extended type byte indicating drive type. Values from 10H to FFH indicate drive types 16 to 255, respectively. All other values of byte 1AH are reserved.

9.5.10 Feature Installed Byte

The feature installed byte (index 1FH) contains the bits to indicate what features are installed. Table 9-14 defines byte usage.

Table 9-14. Feature Installed Byte (1FH)

Bit	Function
Bit 7:3	Reserved
Bit 2	Floppy disk drive A installed
Bit 1	Video display installed
Bit 0	Keyboard BIOS installed

9.5.11 Fixed Disk Type 48 Parameters

The fixed disk type 48 parameters (index 20H-27H) are eight bytes that contain information about various disk use and operating parameters. Table 9-15 lists the bytes and their index values.

Table 9-15. Fixed Disk Type 48 Parameters (20H:27H)

Index	Byte
20H 21H 22H 23H 24H 25H	Cylinder low byte Cylinder high byte Number of heads Write pre-compensation start cylinder low byte Write pre-compensation start cylinder high byte Landing zone cylinder low byte
26H 27H	Landing zone cylinder high byte Sectors per track.

9.5.12 Shadow and Setup Byte

Shadow and setup byte (index 28H) contains the enable and disable bits for the options in the setup program. Table 9-16 defines byte usage.

Table 9-16. Shadow and Setup Byte (28H)

Bit	Function
Bit 7	1 = Cache disabled
Bit 6	1 = Above 16M disabled
Bit 5	1 = Speaker off
Bit 4	1 = 640K enabled
Bit 3	1 = Enter setup at pre-boot only
Bit 2	1 = AT32 I/O enabled
Bit 1	1 = Video shadow disabled
Bit 0	1 = BIOS shadow disabled

Note

Video BIOS, alone, cannot be shadowed. It must be shadowed with system BIOS. Video BIOS shadowing can be disabled separately. If the system BIOS shadow is disabled, the video BIOS shadow must also be disabled. Both video BIOS shadow and system BIOS shadow are disabled when the corresponding bits are set.

9.5.13 CMOS RAM Checksum

The CMOS RAM checksum byte (index 2EH-2FH) contains the sum of the values from addresses 10H through 2DH. Byte index 2EH contains the checksum high byte and byte index 2FH contains the checksum low byte.

9.5.14 Low and High Extended Memory Byte

The low and high extended memory bytes (index 30H and 31H) represent the total extended memory (above 1M) determined during system power-up. System interrupt 15H determines extended memory size. Byte index 30H contains the low byte of the extended memory size and byte index 31H contains the high byte. Table 9-17 defines the byte usage.

Table 9-17. Low and High Extended Memory Bytes (30H and 31H)

Value	Typical Values
0200H	512K RAM extended
0400H	1024K RAM extended
0600H	1536K RAM extended
3C00H	15360K RAM extended

9.5.15 Date Century Byte

The date century byte (index 32H) is the century part of the current date encoded in BCD format. The BIOS sets and reads this byte.

9.5.16 Setup Information Byte

The setup information byte (index 33H) contains information on the setup program. Table 9-18 defines the byte usage.

Table 9-18. Setup Information Byte (33H)

Bit	Function
Bit 7 Bit 6 Bit 5 Bit 4	128K ROM expansion Enable user message after initial setup Reserved Copy of 386 CR0 ET bit
Bits 3:0	Reserved

9.5.17 CPU Speed Byte

CPU speed byte (index 34H) contains information on the CPU speed. Table 9-19 defines the byte usage.

Table 9-19. CPU Speed Byte (34H)

Bit	Function
Bit 7:3	Reserved
Bit 2:0	CPU speed

9.5.18 Fixed Disk Type 49 Parameters

The fixed disk type 49 parameters (index 35H-3CH) are eight bytes that contain information about various disk use and operating parameters. Table 9-20 lists the bytes and their index.

Table 9-20. Fixed Disk Type 49 Parameters (35H:3CH)

Index	Byte
35H 36H 37H	Cylinder low byte Cylinder high byte Number of heads
38H 39H 3AH 3BH 3CH	Write pre-compensation start cylinder low byte Write pre-compensation start cylinder high byte Landing zone cylinder low byte Landing zone cylinder high byte Sectors per track

Communication Ports

10.1 INTRODUCTION

The 302 board contains two RS-232C serial communication ports and one parallel printer port. This chapter provides reference data about the ports, including port addresses and interrupt levels.

10.2 SERIAL COMMUNICATION PORTS

The 302 board provides two RS-232-C serial communication ports (COM1 and COM2). Each port is implemented by an 82510 serial communication controller. Two 82510 asynchronous serial controllers provide interfacing between the communication ports and the CPU. Refer to Table 10-1 for selection of port addresses and interrupt levels.

Table 10-1. Selection of Addresses and Interrupt Levels

Port	Address	Interrupt
COM1	3F8H-3FFH	IRQ4
COM2	2F8H-2FFH	IRQ3

10.2.1 CPU Interfacing

Two 82510 controllers provide the interfacing between the communication ports and the CPU. The 82510 is a demultiplexed bus interface using a bidirectional, buffered, 8-bit data bus and a 3-bit address bus. Thirty-five registers, divided into four banks, control and configure the 82510 controllers.

10.2.2 Connectors and Pinouts

Headers J5 and J6 on the 302 board are 2×13 (26-pin) headers supplying the serial communication channels. A ribbon cable maps J5 to two DB9 connectors (serial connector 1 and serial connector 2). J6 is a header for the optional serial connector 1. Another ribbon cable maps J6 to a DB25 connector. Appendix H list the pinouts for J5 and J6.

10.3 PARALLEL PRINTER PORT

The 302 board provides a Centronics-compatible parallel printer port. The port can be designated as either Port 1 or Port 2 (LPT1 or LPT2). Two sets of onboard jumper pins are used to set up this port. The jumpers select the leading or trailing edge of the Printer Acknowledge signal and enable interrupt generation. Discrete logic provides interfacing between the parallel printer port and the printer. Table 10-2 shows the port address and interrupt request levels for the two ports.

Table 10-2. Port Address and Interrupt Levels

Port	Address	Interrupt
LPT1	0378H-037FH	IRQ7
LPT2	0278H-027FH	IRQ5

10.3.1 Programming

The 302 board uses the read, write, status, and control signal registers to transmit data and status to and from the printer. System software performs all printer controls. Tables 10-3 and 10-4 list input and output instruction information.

Table 10-3. Input Instructions

Input Instruction	Port LPT1	Port LPT2
Data Read	0378H	0278H
Status Read	0379H	0279H
Control Signal Read	037AH	027AH

Table 10-4. Output Instructions

Output Instruction	Address Port 1	Address Port 2	Function	Data
Write Data	0378H	0278H	Character to print	As desired
Write Control	037AH	027AH	STROBE	Set bit 0=1 Reset bit 0=0
			AUTO FEED	Set bit 1=1 Reset bit 1=0
			INITIALIZE	Set bit 2=1 Reset bit 2=0
			SLCTIN	Set bit 3=1 Reset bit 3=0
			Enable int	Set bit 4=1 Reset bit 4=0

Tables 10-5 and 10-6 list the parallel port registers.

Table 10-5. Parallel Port Registers (Bits 7-4)

REGISTER	ВІТ7	BIT6	BIT5	BIT4
Read port	PD7	PD6	PD5	PD4
Read status	BUSY	ACK	PE	SELECT
Read control	1	1	1	IRQ ENB
Write port	PD7	PD6	PD5	PD4
Write control	1	1	1	IRQ ENB

Table 10-6. Parallel Port Registers (Bits 3-0)

REGISTER	ВІТ3	BIT2	BIT1	BIT0
Read port	PD3	PD2	PD1	PD0
Read status	ERROR	1	1	1
Read control	SELECT	INT	AUTOFEED	STROBE
Write port	PD3	PD2	PD1	PD0
Write control	SELECT	INIT	AUTOFEED	STROBE

10.3.2 Connector and Pinouts

J4 is a 2×13 (26-pin) header connected to a DB25 connector. Refer to Appendix E for pinout information on the parallel printer connector.



Keyboard Controller

11.1 INTRODUCTION

The board supports a 101- or 102-key enhanced keyboard. The keyboard is controlled by the 8742 single-chip microcomputer. This chapter describes the keyboard interface on the 302 board (see Figure 11-1.)

11.2 KEYBOARD CONTROLLER SYSTEM INTERFACE

The keyboard controller communicates with the system through a status register, an output buffer, and an input buffer. The status register is an 8-bit read-only register at I/O address 64H. The output buffer is read only and at I/O address 60H. The input buffer consists of data write at I/O address 60H and command write at I/O address 64H.

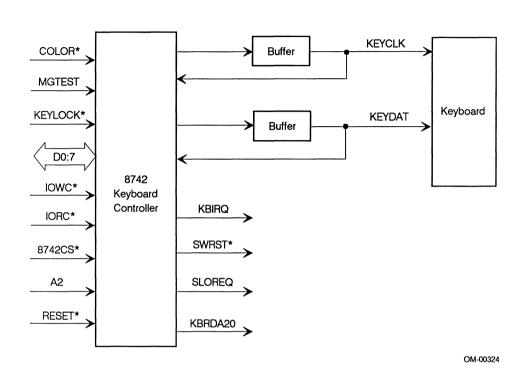


Figure 11-1. Keyboard Controller

11.2.1 Status Register

The status register contains information about the state of the controller and keyboard interface. Table 11-1 lists the status register bit assignments.

 Table 11-1.
 Status Register Bit Assignments

Bit/Value	Function
Bit 7 1 0	Parity error This bit indicates whether the last byte of data received from the keyboard had odd or even parity. In normal operation, the byte sent by the keyboard should have odd parity. Even parity (error) Odd parity (no error)
Bit 6 1 0	Receive time-out This bit indicates whether the data reception from the keyboard terminated normally within the programmed receive time-out delay or not. Did not terminate within time-out delay Terminated within time-out delay
Bit 5 1 0	Transmit time-out This bit indicates whether the keyboard controller transmission terminated normally within the specified time or not. Did not terminate within the specified time. Terminated within the specified time
Bit 4 1 0	Front panel keylock status Keyboard not inhibited Keyboard inhibited

(continued)

Table 11-1. Status Register Bit Assignments (continued)

Bit/Value	Function
Bit 3 1 0	Command/Data Status in Input Buffer A command byte was written (via port 64H) A data byte was written (via port 60H)
Bit 2 1 0	System flag Self-test was successful Power-on reset occurred
Bit 1 1 0	Input Buffer Full A write to either 60H or 64H sets this bit to 1. This bit is set to 0 when the keyboard controller reads the buffer. Data from the CPU in the input buffer No data from the CPU in the input buffer
Bit 0	Output Buffer Full This bit is set to 0 when the CPU reads the buffer (DBBIN). The keyboard controller has loaded the output buffer with data
0	No data from the keyboard in the output buffer (DBBOUT)

11.2.2 Output Buffer

The controller output buffer is an 8-bit read-only register at I/O address 60H. The controller sends keyboard scan codes and command-requested data bytes to the system via the output buffer. The significant data in the output buffer can be read only when bit 0 of the status register is 1.

11.2.3 Input Buffer

The keyboard controller input buffer is an 8-bit write-only register at I/O addresses 60H or 64H. The input buffer is used to receive data from the system. Data can be written to the input buffer only if bit 1 of the status register is 0. Full output buffers should always be serviced first.

Writing to address 60H clears the command/data bit (bit 3) of the status register. Once bit 3 clears, the controller processes the data in the input buffer as a data byte. Data written to address 60H is sent to the keyboard unless a system command instructs the controller to wait for control data.

Writing to address 64H sets the command/data bit (bit 3) of the status register to 1. Once bit 3 is set, the controller processes the data in the input buffer as a command byte.

11.2.4 Input and Output Ports

The input port consists of four signals: two signals driven to the controller by the keyboard, and two signals indicating the keylock state and color/monochrome bit setting. The output port consists of seven signals driven by the controller to the keyboard. These ports are accessed by sending the appropriate read or write command to the controller. Tables 11-2 and 11-3 list the input and output port bit assignments respectively.

Table 11-2. Input Port Bit Assignments

Bit	Function
7	Keylock (lock = 0)
6	Color/mono (color = 0)
5	$MFG\ TEST\ (enabled=0)$
4:1	Reserved
0	Keyboard data in

Table 11-3. Output Port Bit Assignments

Bit	Function		
7	Keyboard data out		
6	Keyboard clock out		
5	Reserved		
4	IRQ01 (keyboard interrupt)		
3	Reserved		
2	Deturbo (slow down)		
1	Gate address line 20		
0	Reset CPU (software reset)		

11.3 CONTROLLER COMMANDS

The CPU uses controller commands to govern the operation of the keyboard controller and sense its status. The CPU writes controller commands into the input buffer through I/O address 64H. Keyboard controller commands are listed in Table 11-4 and described individually on the following pages.

Table 11-4. Controller Commands

Code	Description
20H:3FH 60H:7FH AAH ABH ACH ADH AEH	Read keyboard controller command byte Write keyboard controller command byte Self-test Test interface NO-OP Disable keyboard Enable keyboard
COH DOH D1H E0H E1H:EFH F0H:FFH	Read input port Read output port Write output port Read test input port Reserved Output pulse

11.4 KEYBOARD INTERFACE

The keyboard connects to the controller through a bidirectional-synchronous serial interface cable with a 5-pin DIN connector (J9). The controller supplies the keyboard with DC power of $+5V \pm 10\%$ at a maximum 300 mA of current. Some keyboards require the system to send a RESET signal to the keyboard when a system reset occurs. This ensures the keyboard buffers are cleared. See Appendix H for the keyboard connector pinout.

The controller and the keyboard communicate using data and clock lines for synchronous serial communication. Open-collector drivers, at both ends of the cable, drive the data and clock lines.

At power-up, the keyboard scans the signals on the clock and data lines and establishes a line protocol. A bidirectional serial interface in the keyboard converts the clock and data signals and transfers them to and from the keyboard through the keyboard cable. Signals include keyboard control commands from the controller and keyboard scan and acknowledgment codes transferred to the controller.

The serial data from the keyboard is called a scan code. Each keyboard key has an associated 11-bit scan code. Pressing and releasing a key generates a make or break scan code. The keyboard detects all keys pressed and transfers each scan code in the correct sequence to the controller.

The controller receives serial data from the keyboard, checks the parity of the data, and translates the 8-bit scan code into system codes. It also interrupts the CPU to transfer data to the system. The controller interrupts the system when data is placed in its output buffer, or waits for the system to poll its status register to determine when data is available.

The controller sends various commands to the keyboard at any time. When the controller sends data to the keyboard, it sets the data line to an inactive state and allows the clock line to go to an active state. This action serves as both a request-to-send (RTS) and a start bit. If the clock line is set to an inactive state, keyboard transmission is inhibited.

If the controller has to send data to the keyboard when the keyboard is transmitting to the controller, the controller first clamps the clock signal line to request a keyboard transmission halt. This clock line must remain low for at least 60 ms to ensure that the keyboard recognizes the controller request.

During the basic assurance test (BAT) or when no data transmission is occurring, the clock line is at an active (high) state. The data line is held at the active (high) state by the keyboard.

A signal at an inactive (low) state has a value between 0V and +0.7V (logical 0). A logical 1 is an active signal with a value between +2.4V and +5.5V. These voltages are measured between a signal source and the DC network ground.

11.4.1 Keyboard Data Stream

The 11-bit data stream (scan code), transferred serially over the data line, consists of one start bit, eight data bits, one odd parity bit, and one stop bit. A logical 1 indicates an active level and a logical 0 an inactive level. The 8 data bits plus the parity bit always have an odd number of 1's. Bit 1 of the Data Stream bits is the first bit transmitted. Table 11-5 lists the functions of the Data Stream bits.

Table 11-5. Data Stream Bits

Bit	Function			
1	Start bit (always 0)			
2	Data bit 0 (LSB)			
3	Data bit 1			
4	Data bit 2			
5	Data bit 3			
6	Data bit 4			
7	Data bit 5			
8	Data bit 6			
9	Data bit 7 (MSB)			
10	Parity bit (odd parity)			
11	Stop bit (always 1)			

11.4.2 Receiving Data from the Keyboard

The keyboard transfers data to the controller in a serial format using an 11-bit frame. The first bit is a start bit, followed by 8 data bits, an odd parity bit, and a stop bit. A clock, supplied by the keyboard, synchronizes the data transfer.

Before transferring data, the keyboard checks for a transmission inhibit or controller RTS status on the clock and data lines. If transmission is inhibited (clock line inactive), keycodes are transferred to the keyboard buffer. During controller RTS status, data is also stored in the buffer while the keyboard receives system data.

The keyboard transfers keycodes to the controller only when both the clock and data signals are active. At the end of a transfer, the controller disables the interface until the system accepts the data byte.

If a parity check error occurs, the controller signals the keyboard to transfer the data again. If the controller does not receive the data correctly after a number of retries, an FFH code is sent to the output buffer. The parity bit in the status register is also set to 1 indicating a receive parity error.

The controller times each data byte transfer from the keyboard. If a keyboard transfer does not end within 2 ms, the receive time-out bit in the status register is set. An FFH code is sent to the keyboard controller output buffer. No retries are attempted on a receive time-out error.

The following commands are sent from the keyboard to the controller.

Overrun or Key Detection Error (00 or FFH) If the keyboard is using scan code set 1, the code is FFH. For sets 2 and 3, the code is 00H. The overrun or key detection error condition are:

- The keyboard sends a key detection error character if conditions in the keyboard make it impossible to identify a switch closure.
- When the buffer in the keyboard is full, an overrun character replaces the last transmitted code in the buffer. This code is sent to the controller when it reaches the top of the buffer queue.

Keyboard ID (83ABH) The keyboard ID consists of two bytes, 83ABH. The keyboard controller responds to the read ID command with ACK, discontinues scanning, and sends the two ID bytes. The low byte is sent first followed by the high byte. Following output of the keyboard ID, the keyboard controller resumes scanning.

BAT Completion Code Each time the system is powered on or POST is run, the (AAH)

keyboard performs a self-test operation called the BAT. The BAT consists of a keyboard processor test, a checksum of the ROM, and a RAM test. Activity on the clock and data lines is ignored during a BAT. The AAH command is sent to the controller following satisfactory completion of the BAT. Any other code indicates a keyboard failure.

BAT Failure Code

(FCH)

If a BAT failure occurs, the keyboard sends FCH. discontinues scanning, and waits for a controller

response and reset.

Echo (EEH)

When the controller issues the Echo command to the keyboard, the keyboard sends EEH as a response.

Acknowledge (FAH)

Resend (FEH)

The keyboard issues ACK to any valid input other than an Echo and a Resend command. If the keyboard is interrupted while sending ACK, it discards ACK and responds to the new commands.

The keyboard sends this command when it receives invalid input or any input with incorrect parity. This command signals the controller to send the input again. If the controller sends nothing to the keyboard, no

response is required.

11.4.3 Sending Data to the Keyboard

The controller sends data to the keyboard in the same serial format that it uses to receive data from the keyboard. Before the controller sends data to the keyboard, it checks the keyboard and determines whether the keyboard is transferring data. If the keyboard is transferring data, but has not reached the tenth clock signal, the controller overrides the keyboard output by asserting the keyboard clock signal. If the keyboard transfer is beyond the tenth clock signal, the controller waits until the keyboard completes its transmission before sending data.

If the controller overrides the keyboard output, or if the keyboard is not sending data, the controller sets the clock line inactive for more than 60 μ s while preparing to send data. When the controller sends the start bit, the clock signal is asserted.

Each controller command or data transmission to the keyboard requires a response before the controller sends its next output. After the keyboard receives a controller command, it returns an ACK code to the controller. If the keyboard response is invalid or has a parity error, FEH is placed in the controller output buffer and the transmit time-out or parity error bits are set to 1 in the status register.

The controller sets a programmed time limit (20 to 25 ms) for the keyboard to respond. If the keyboard cannot complete the send-out data process within this time period, the controller places FEH in its output buffer and sets the transmit and receive time-out error bits to 1 in its status register. No retries are attempted by the controller for any transmission error. The following commands are sent from the controller to the keyboard.

Echo (EEH)

This command is used as a diagnostic aid to test the keyboard command process. When the keyboard receives this command, it issues an EEH response and continues scanning.

Invalid Commands (EFH and F1H) EFH and F1H are invalid commands and are not supported. If EFH or F1H is sent, the keyboard does not acknowledge the command. Instead, it returns a Resend command and continues in its prior scanning state. No other activities occur.

Select Alternate Scan Codes (F0H)

This command instructs the keyboard to select one of three sets of scan codes. The keyboard acknowledges receipt of this command with the ACK signal, then clears both the output buffer and the typematic key (if one is active). When the controller sends the option byte, the keyboard responds with another ACK signal. An option byte value of 01H selects scan code set 1, 02H selects set 2, and 03H selects set 3. The byte value 00H causes the keyboard to respond with the ACK signal and send a byte, thereby signaling the controller which scan code set is in use.

Read ID (F2H)

This command requests identification information from the keyboard. The keyboard responds with the ACK signal, discontinues scanning, and sends the two keyboard ID bytes. The second byte must follow completion of the first byte within 500 μ s. After the output of the second ID byte, the keyboard resumes scanning.

Set Typematic Rate/Delay (F3H) This command sets the typematic rate and delay. The keyboard responds with the ACK signal, stops scanning, and waits for the controller to issue the rate/delay value byte. Once issued, the keyboard responds with another ACK signal, sets the rate and delay to the values indicated, and resumes scanning.

The contents of the rate/delay value byte following the command determine the parameters for these two functions. Bits 4:0 set the typematic rate, bits 5 and 6 set the delay parameter, and bit 7 is set to 0. The following equations show the calculation for the delay parameters and the typematic rate:

- Delay = $(1 + C) \times 250 \text{ ms} \pm 20\%$
- Period T = $(8 + A) \times 2^B \times 0.00417$ seconds
- Typematic Rate = 1/T ± 20% (default is 10 characters per second)

Where: C = Binary value of bits 6 and 5 (default is 500 ms)

A = Binary value of bits 2, 1, and 0

B = Binary value of bits 4 and 3

T = Interval from one typematic output to the next

Enable (F4H)

When the keyboard receives this command, it responds with the ACK signal, clears its output buffer, clears the last typematic key, and starts scanning.

Default Disable (F5H)

This command resets all conditions to the power-on default state. The keyboard responds with the ACK signal, clears its output buffer, sets the default key types and typematic rate/delay, and clears the last typematic key. The keyboard stops scanning and waits for further instructions from the controller.

Set Default (F6H)

This command is similar to the default disable command F5H. However, the keyboard continues scanning instead of stopping and waiting for further instructions.

Set All Keys (F7H, F8H, F9H, and FAH) Commands F7H, F8H, F9H, and FAH instruct the keyboard to set all keys to typematic, make/break, make, and typematic/make/break, respectively. The keyboard responds with the ACK signal, clears its output buffer, sets all keys to the type indicated by the command, and continues scanning. Although these commands are sent using any scan-code set, they affect only scan-code set 3.

Set Key Type

Commands FBH, FCH, and FDH instruct the keyboard (FBH, FCH, and FDH) to set individual keys to typematic, make/break, and make respectively. The keyboard responds with the ACK signal, clears its output buffer, and prepares to receive key identification. The controller identifies each key by its scan-code value as defined in scan code set 3. Only scan-code set 3 values are valid for key identification. The type of each identified key is determined by the value indicated by the command.

Resend (FEH)

The controller transmits this command when it detects an error in any transfer from the keyboard. The controller requests the keyboard to retransmit the code that was detected as an error. This command is only transmitted after a keyboard transfer and before the controller allows the next keyboard output. When a resend command is received, the keyboard transfers the previous output again. If the previous output was Resend, the keyboard transfers the last byte before the Resend command.

Reset (FFH)

The controller issues this command to start a program reset and a keyboard internal self-test. The keyboard responds with the ACK signal and ensures the controller accepts the ACK signal before executing the command. The controller signals acceptance of ACK by raising the clock and data lines for at least 500 μ s. The keyboard remains disabled from the time it receives the reset command until ACK is accepted, or until another command is sent that overrides the previous command. Following the acceptance of the ACK signal, the keyboard is reinitialized and performs the BAT. After returning the completion code, the keyboard defaults to scan-code set 2.

11.5 KEYCLICK VOLUME

The keyclick volume feature controls the eight volume levels of the keyboard. When a key is pressed, the 302 board emits a clicking sound, giving audio feedback for controller keyboards with little or no audio or tactile feedback. Holding down the or and the keys then pressing the key, on the numeric keypad, increases the volume to the next highest level. If the volume is at the highest level, pressing the key sequence decreases the volume to the lowest level. At power-up, the volume is set to the second-lowest level.

302 Board Special Interfaces

12.1 INTRODUCTION

This chapter discusses three special interfaces: the keylock interface, the speaker interface, and the reset interface.

12.2 KEYLOCK INTERFACE

The keylock interface can be used with a key-activated switch to electrically lock out the keyboard. The keylock interfaces with part of connector J23 on the 302 board. Table 12-1 lists the keylock interface pinout.

Table 12-1. J23 Keylock Pinout

Signal/Function	Pin
+ 5V	13
Not connected	14
Ground	15
KEYLOCK*	16
Ground	17

12.3 SPEAKER INTERFACE

A 75477 high current analog driver provides audible tone generation for the speaker. The 75477 uses an OSC signal derived from a counter on the 8254 programmable interval timer (PIT) module to drive the SPKOUT signal. The OSC signal is gated on and off by bit 0 (ENABLE SPEAKER) of port 61H. Bit 1 (SPEAKER DATA) supplies the data for sounding the speaker. Port 79H, bit 6 is an additional speaker enable signal to enable the 75477 to drive the speaker. Part of the header (J23) provides the connection between the speaker and the system. Table 12-2 lists the speaker interface pinout.

Table 12-2. J23 Speaker Header Pinout

Signal/Function	Pin
SPKOUT speaker drive out	8
Key (pin missing)	9
Ground	10
+5V	11

12.4 RESET INTERFACE

There are several ways in which to reset the 302 board: hardware, software, and keyboard.

Systems with a front panel reset feature will force a hard reset by activating the reset switch. The interface to this function is through J23 on the 302 board. Table 12-3 lists the pin definitions for J23. This reset will cause the hardware of the entire 302 board to re-initialize.

Table 12-3. J23 Reset Interface Pinouts

Signal/Function	Pin
Ground	5
FPRESET*	6

The function of a software reset is to force a reset of the CPU only. No other hardware including the 387 numeric coprocessor is effected. This is a function of the 8742 microcontroller provided by the keyboard interface. Bit 0 of the keyboard controller output port is toggled via the pulse output port command. See Chapter 11 for details on the keyboard controller.

A keyboard reset is initiated by holding down the [cri], [Art], and [Delete] keys on the keyboard at the same time. The effect of this function is to cause the CPU to go to the address specified by the reset vector. This type of reset does not directly effect any hardware.

Power-on Self Test and Setup

13.1 INTRODUCTION

This chapter describes the power-on self test (POST), which executes automatically each time a 302 system is booted, and the setup program. Both the POST and the setup program are stored in ROM on the 302 board. This chapter assumes that the 302 board has been correctly integrated into a computer system environment with all necessary I/O devices connected, peripheral devices installed, and configuration performed.

13.2 POWER-ON SELF TEST

Each time the 302 system is turned on or reset, the POST runs automatically and checks the CPU, keyboard, video display, memory, and most peripheral devices.

During the POST memory test, the amount of memory being tested is displayed on the screen. Depending on the amount of extended memory installed, the POST memory test takes 3 to 15 seconds to complete.

During a soft boot, the system executes all POST tests except memory.

When POST completes the system beeps once, if no configuration errors are detected, and displays a message similar to the following:

Phoenix 80386 ROM BIOS PLUS Version x.xx yy.yy Copyright (C) 1985-1988 Phoenix Technologies Ltd. All Rights Reserved If configuration errors are detected, the system beeps twice and displays a message similar to the following:

640K Base Memory, 01024K Extended
Invalid configuration information - please run SETUP program
Strike the F1 key to continue, F2 to run the setup utility



It is normal for this message to appear the first time you start the system.

If you press the [1] key to continue, the system will operate, but will not be correctly configured.

13.3 SETUP PROGRAM OVERVIEW

The setup program, located in ROM BIOS on the board, enables you to check or change the system configuration information.

The setup program does not require you to see a particular operating system or special diskette to change system configuration values. Values changed through the setup program are effective when the system is rebooted.

You can check or change the following configuration values using the setup program:

- Date and time
- Number and type of floppy disk drives and fixed disk drives
- Amount of base memory and extended memory
- Availability and type of primary monitor controller
- Availability of keyboard
- Availability of math coprocessor
- Shadow or do not shadow system BIOS and video BIOS

- Enable or disable cache memory
- Enable or disable AT32 I/O
- Enable or disable access to memory above 16M
- Enable or disable the speaker
- Enable or disable post-boot setup
- Enable or disable system memory at 512K-640K

When the system boots, the POST checks the stored setup information against the hardware configuration. If the data does not agree, the POST displays an invalid configuration message. If such a message appears, you can run setup to enter the correct configuration parameters.

- Displays date, time, and current values for system options.
- Allows you to accept current (default) values or enter different values for time, date, and system options.
- Provides instructions on booting the system so new settings take effect.

13.4 RUNNING THE SETUP PROGRAM

Because the setup program is permanently stored in ROM, it can be run at any time.

To run setup, wait for POST to complete (you'll hear a short beep), then immediately press [ctrl] + [Att] + [ms].

A display similar to Figure 13-1 will appear. The actual display you will see depends on the specific configuration of your system.

```
Phoenix Technologies Ltd. Version
System Configuration Setup x.xx yy
```

Time: 08:20:20

Date: Fri Jan 29, 1990

Diskette A:

3.5 Inch, 1.4 MB

Diskette B:

5.25 Inch, 1.2 MB Hd Cyl

Hard Disk C:

Type 44 6 -1 820 17

Pre Lz

Hard Disk D:

Not Installed

Base Memory:

640 KB

Extended Memory:

1024 KB

Display:

VGA/EGA Installed

Keyboard: CPU speed:

Turbo

Coprocessor:

Not Installed

PgUp for advanced options. Up/Down Arrow to select. Left/Right Arrow to change entries.

F1 for help. F10 to exit and save changes.

Esc to reboot for changes to take effect.

Figure 13-1. Typical Setup Screen 1

To change an option, use the 1 or 1 key to move the cursor to the option you want to change. The cursor will move only to options that you can change.

To change the base memory and extended memory, enter the correct numeric values. If you enter an incorrect number, you can correct the entry by using the Backspace key. To change other options, press the 🗐 or 🗐 key to display the possible options. Each time you press a key, a new value is displayed.

When you finish setting options, exit setup by pressing Fig or Esc. You can exit setup from either screen.

Size

Sec

40

Press F10 to exit the setup program without booting the system. Changes are saved, but only the date and time information take effect. The other changes take effect when the system boots. Press Esc to save the setup changes and boot the system.

13.4.1 Moving Through Setup Screen 1

Make sure the first setup screen is displayed (refer to Figure 13-1. Setup Screen 1). If it is not, press Page Up.

SETTING SYSTEM TIME

The time option contains three fields: hours, minutes, and seconds. You can change the hours and minutes fields to specific values. You can reset the seconds field to 00.

To set the correct time, press the \square or \square key. Press the \square key to increment the time; press the \square key to decrement the time. Set the hour first, and then press the \square key to move to the minutes field. Press the \square or \square key to set the minutes, and then press the \square key to move to the seconds field. Press either the \square or \square key to reset the seconds field to 00.

SETTING SYSTEM DATE

The date option contains four fields: day of the week, month, day of the month, and year. You can set any field except the day of the week. When you change any of the other three fields, the setup program resets the day of the week.

Press the fi or key to move to a different date field. Press the or key to toggle among the entries in the date fields.

SELECTING FLOPPY DISK DRIVE TYPES

The setup program maintains information about two floppy disk drives, drive A and drive B. Press the 1 or 1 key to move the cursor to the field you want to change.

To change the floppy disk drive types, press the \blacksquare or \blacksquare key. The following options will be displayed:

```
3.5 Inch, 1.44 MB
3.5 Inch, 720 KB
5.25 Inch, 1.2 MB
5.25 Inch, 360 KB
Not Installed
```

If only one floppy disk drive is installed, it will always be drive A. In this case, set drive B to "Not Installed." You cannot change the individual fields within the options.

SETTING FIXED DISK DRIVE TYPES

The setup program maintains drive type information for two fixed disk drives, drive C and drive D. Press the 🗓 or 🗓 key to move the cursor to the field you want to change.

Press the 🖃 or 🖼 key to display the drive type options. The following options will be displayed:

```
1-49
Not Installed
```

For standard (non-SCSI) fixed disk drives, the drive type number is usually located on a sticker on the top or rear of the drive unit.

If you do not know the drive type, the setup program contains a list of the major specifications for fixed disk drive types 1-47. To access this list from the setup screen, position the cursor in either the fixed disk C or D field and press the relative twice.

If the list does not include the drive type in the system, choose the user-configurable disk type (Type 48 for drive 1 and Type 49 for drive 2). Set the number of cylinders, number of heads, write precompensation, landing zone, and sectors per track to match the specifications of the fixed disk.



CAUTION

Specifying an incorrect drive type may damage data on the fixed disk.

Select "Not Installed" if a SCSI fixed disk drive is installed or if no fixed disk drive is installed. If only one fixed disk drive is installed, it will always be drive C. In this case, select "Not Installed" for drive D.

SETTING BASE MEMORY

If the setup screen indicates an incorrect value for base memory, press the \blacksquare or \blacksquare key to move the cursor to the base memory field.

To change the base memory, type in the base memory size in kilobytes. There are only two valid choices for base memory size: 512K and 640K.

SETTING EXTENDED MEMORY

Extended memory is random access memory (RAM) above 1024K (1M).

The first 1M of memory is used for base memory and dedicated memory. This amount must be subtracted from the total amount of RAM in the system (onboard RAM + expansion board RAM) to compute extended memory. For example, 4M of onboard memory plus 8M of expansion board memory equals 12M of extended memory. Twelve megabytes (12M) minus the first 1M of base memory leaves 11M (11264K) of extended memory.

If the POST indicates "Invalid configuration information," when the setup program is executed, it will attempt to correct the error if it is due to an invalid memory configuration.

Press the 🗈 or 🗓 key to move the cursor to the extended memory field.

To change the extended memory value, type in the total extended memory size in kilobytes, according to the preceding formula.

The setup program will auto-configure the memory size when the system is first powered up.



Some memory expansion boards add memory in varying increments. The setup program is not restricted to set increments and accepts any memory size.

SETTING VIDEO DISPLAY CONTROLLER TYPE

You must specify the type of monitor controller installed in the system for the self-test to function correctly.

Press the figure or leaves to move the cursor to the monitor controller field.

Press the ☐ or ☐ key to display the following options:

vga/ega: Enhanced graphics, video graphics, or any controller

with its own BIOS

cga40: Color/graphics adapter in 40-column mode

cga80: Color/graphics adapter in 80-column mode

Monochrome controller

Not Installed: No controller is installed. This setting allows the self-

test to suppress video display errors, and permits systems such as network servers to operate without

monitors.

Select the VGA/EGA option if an enhanced graphics, video graphics, or video controller with its own BIOS is installed, no matter what mode the controller is set for or if another controller is designated as the primary one. Otherwise, select the type of the primary monitor controller, or "Not Installed" if no monitor controller is installed (or if none will be installed in the final configuration).

SETTING KEYBOARD AVAILABILITY

You must specify the presence or absence of a keyboard for the self-test to function correctly.

Press the 🗓 or 🗓 key to move the cursor to the keyboard field.

Press the ☐ or ☐ key to display the Installed or Not Installed options.

Select "Installed" if a keyboard is attached to the system unit. Select "Not Installed" if no keyboard is attached. This setting allows the self-test to suppress keyboard errors and errors which would require pressing the key, to continue permitting systems such as network servers to operate without keyboards.

SETTING CPU SPEED

The CPU speed setting determines the effective speed used by the system each time you turn on the power.

Press the $\mathbf{1}$ or $\mathbf{1}$ key to move the cursor to the CPU speed field.

Press the \blacksquare or \blacksquare key to display the turbo (25 MHz) or deturbo (8 MHz) options.

The 25 MHz speed is the normal setting. Try 8 MHz only if you experience problems while running programs. This setting can also be changed via the keyboard (see Appendix D).

NUMERIC COPROCESSOR

You cannot change the math coprocessor status entry. Every time it runs, the system self-test checks the math coprocessor status. Setup simply displays the status (80387 or Not Installed).

13.4.2 Moving Through Setup Screen 2

Press the Page Up key. A screen display similar to the one shown in Figure 13-2 appears.

Phoenix Technologies Ltd.
Additional Options

Time: 08:24:30

Date: Fri Jan 29, 1990

Shadow BIOS ROM:

System and Video BIOS

CACHE Memory:

Enabled Disabled

AT32 I/0: Above 16 Meg:

Enabled

Speaker:

Enabled |

512 - 640KB:

Enabled

Enter SETUP:

Always

PgUp for main menu. Up/Down Arrow to select. Left/Right Arrow to change entries.

F1 for help. F10 to exit and save changes. Esc to reboot for changes to take effect.

Figure 13-2. Typical Setup Screen 2

SHADOWING SYSTEM BIOS AND VIDEO BIOS

The 302 board maintains an area of 32-bit RAM into which it can copy the system BIOS and video BIOS. This memory, called shadow memory, is write-protected and has the same addresses as the system BIOS and video BIOS ROM locations. System performance increases significantly because the information is maintained in fast 32-bit RAM instead of in ROM.

Press the 🗓 or 🗓 key to move the cursor to the shadow BIOS ROM field.

Press the ☐ or ☐ key to display the following options:

Shadow BIOS ROM: System and Video BIOS

Shadow BIOS ROM: System BIOS Only

Shadow BIOS ROM: Disabled



The video BIOS cannot be shadowed independently of the system BIOS.

The recommended option is to shadow both the system BIOS and video BIOS to allow programs fast access to the BIOS in RAM. Disable the shadow BIOS only if you are experiencing problems when running certain application programs. Select the System BIOS Only option if you experience problems using monitor controllers.

CACHE MEMORY

This setting determines whether the cache will be enabled or not.

Press the \P or \P key to move the cursor to the cache memory field.

Press the \blacksquare or \blacksquare key to display the enabled or disabled options.

Cache memory enabled is the recommended option. Cache memory disabled is the lower-performance option. Disable the cache memory only if you are experiencing problems when running with the cache memory enabled.

AT32 I/O

This setting determines whether I/O addresses in the range 8000:90FFH are decoded onto the AT32 bus or the I/O bus. Enabling this feature allows special AT32 boards to respond to these addresses.

Press the 🗓 or 🗓 key to move the cursor to the AT32 I/O field.

Press the \blacksquare or \blacksquare key to display the enabled or disabled options.

The disabled option is the recommended setting. Enable the AT32 I/O bus only if you are installing a board that requires a 32-bit I/O slot and responds to I/O addresses in the address range 8000:90FFH.

ABOVE 16M

The POST detects and tests contiguous extended memory. However, the ROM BIOS creates a hole below 16M (0E00000H-0FFFFFH) in the extended memory space in systems with more than 16M extended memory. This option enables users to configure address range 0E00000H-0FFFFFH as either ROM BIOS or extended memory.

Press the \blacksquare or \blacksquare key to move the cursor to the Above 16M field.

Press the \blacksquare or \blacksquare key to display the enabled or disabled options.

The default setting is disabled. If you have more than 16M in the system, it should be enabled otherwise you will be unable to access the extended memory beyond 16M.

512-640 K

If the base memory in the system is 640K, this option must be enabled. If base memory is 512K, this option must be disabled.

Press the 1 or 4 key to move the cursor to the Base Memory field.

Press the \blacksquare or \blacksquare key to display the 512K or 640K options.

SPEAKER

This setting determines whether the speaker will be enabled or not.

Press the 🗓 or 🗓 key to move the cursor to the speaker field.

Press the \blacksquare or \blacksquare key to display the enabled or disabled options.

Speaker enabled is the recommended setting. When the speaker is disabled through the setup program, writing to port 61H will not enable it. It must be re-enabled using the setup program.

ENTER SETUP

The Enter Setup option determines when you can enter the setup program by pressing the [Tit] + [Att] + [Ins] key sequence.

Press the figure or leaves to move the cursor to the Enter Setup field.

Press the
☐ or ☐ key to display the always or pre-boot options.

Pre-boot is the recommended setting. Selecting pre-boot allows you to enter setup only during the brief period between the time of the beep when you turn the power on and before the boot sequence. This setting prevents you from accidentally entering setup from an operating system or an application program.

If you select always, you can enter setup at any time after POST by pressing the following sequence: [CIT] + [AIT] + [ITS].

13.5 EXITING SETUP

To exit setup after changing all desired options, press the setup from either setup screen 1 or 2.

If you press the key to exit, the setup program saves the new information and reboots the system. The new configuration changes are effective immediately.

If you press the Fig. key to exit, the system will not reboot. The new configuration changes are saved, but only the date and time information are effective immediately. You have to reboot the system for the other changes to be effective.

13.6 ERROR MESSAGES

The following basic message can be followed by any combination of the error messages numbered one through six. These errors occur only when you run setup.

Message

Errors have been found during the power-on self-test in your computer. The errors were:

- 1. Clock chip lost power
- 2. CMOS checksum invalid
- 3. Incorrect configuration data in CMOS
- 4. Memory size in CMOS invalid
- 5. Disk C: failed initialization
- 6. Time or Date in CMOS is invalid

Strike any key to continue

Possible Cause

The configuration information stored in the RTC does not

agree with the hardware configuration of the 302

system.

Solution

Make sure the jumper settings on the 302 board agree

with the hardware configuration. Run setup and enter

the correct configuration information.

Specifications



A.1 302 BOARD SPECIFICATIONS

Table A-1 lists general specifications for the 302 board.

Table A-1. 302 Board Specifications

Attributes		Specification
CPU CPU clock rate ISA bus speed Data path		386 32-bit microprocessor 25 MHz 8 MHz 8-, 16-, 32-bits
Addressing	Physical Virtual Supported	256M 64 terabytes 40M
Cache Memory	Size Speed	64K zero wait-state on read hit one wait-state on write cycle
ROM	Size	64K
DRAM	Size	0, 1, 2, 4, 8M
AT32 Expansion DRAM		4, 8, or 16M per slot (2)
Expansion Slots		Two 8-, 16- and 32-bit slots Five 8- and 16-bit slots One 8-bit slot

A.2 302 BOARD PHYSICAL CHARACTERISTICS

Figure A-1 shows the 302 board dimensions.

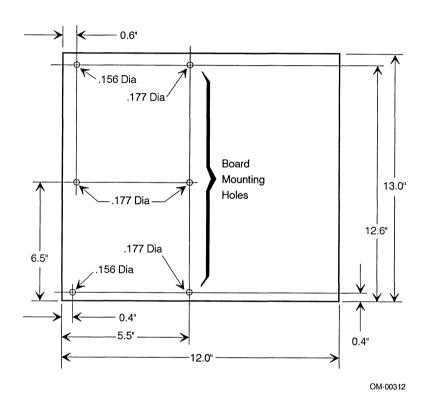


Figure A-1. 302 Board Dimensions

Table A-2 lists the physical characteristics for the 302 board.

Table A-2. Physical Characteristics

Attributes		Specification
Dimensions	Width Length Height Weight	13.0 inches (317.2 mm) 12.6 inches (320.0 mm) 0.75 inch (1.8 cm) 43 oz. (1219 gm)
Environment	Operating temp. Storage temp. Operating humidity Altitude Shock Vibration	13 °C to 35 °C (55 °F to 95 °F) -40 °C to 65 °C (-40 °F to 149 °F) To 80% To 10,000 feet (3048 m) maximum 50 G, 11 ms, 1/2 sine wave Random vibration 10 - 1,000 Hz

A.3 302 BOARD POWER REQUIREMENTS

The 302 board requires the power consumption and current ratings listed in Table A-3 for normal operation.

Table A-3. Power Consumption

Voltage	Nominal Current	Power (Watts)
+5.0V ±5%	6.0A	30.0
+12.0V ±10%	0.06A	0.72
-12.0V ±10%	0.08A	0.96

The 302 board contains eight expansion slots: one 8-bit, five 16-bit, and two 32-bit.

The maximum allowable +5V current available to any expansion slot depends on the number of +5V pins in the slot providing current, the power supply capacity, the power connectors, the demands of both the 302 board and all other slots in use, and the 302 board circuit traces. Each expansion slot pin can carry a maximum of 3A. Table A-4 lists the current ratings for each slot type.

Table A-4. Maximum Current Ratings

Maximum Allowable Current per Slot	
6A (3A per pin with 2 pins)	
9A (3A per pin with 3 pins) 21A (3A per pin with 7 pins)	



CAUTION

The ISA bus connectors are limited to 3A per pin maximum. The power supply connectors are limited to 5A per pin maximum. These limitations plus the maximum capacity of the power supply itself may be more restrictive than the current limitations of the pins/connectors and expansion slots. As an over limitation, do not exceed 20A total when computing total current drain for the system board. This avoids damage to the power supply and system board.

Table A-5 specifies the nominal current drain for various system board configurations.

Table A-5. Maximum +5V Current Drain on 302 Board

Element	Maximum +5V Current Required
302 board - 4M/8M onboard RAM, no numeric coprocessor, no expansion boards	12.00A
387 numeric coprocessor	0.25A
ATMEM16 memory expansion board 4M configuration 8M configuration 16M configuration	1.39A 1.48A 1.70A

Note

The values for the ATMEM16 memory expansion boards are the quiescent currents. Only one 4M bank of total RAM in the system can be active at a given time. Total RAM includes all onboard RAM and ATMEM16 RAM (if present). The current drain for the 4M on active RAM is included in the 302 board current specified in Table A-5.

For example, assume your system has 8M of onboard RAM, a numeric coprocessor, and one ATMEM16 board with 16M of RAM. The total +5V current drain would be 13.95A (12.00A + 0.25A + 1.70A).

System BIOS Specifications



B.1 INTRODUCTION

The basic input output system (BIOS) is a software interface that isolates operating systems and application programs from specific hardware devices. The BIOS routines allow assembly language programmers to perform block and character-level operations without concern for device addresses or hardware operating characteristics. The BIOS also provides system services such as time-of-day and memory-size determination.

Application programs should make functional requests to BIOS rather than directly manipulating I/O port control words. This appendix provides a description and list of the BIOS routines.

B.2 NON-MASKABLE INTERRUPT (INT 02H)

The 302 board and I/O channel parity-error signals connect to the NMI pin of the CPU. When a memory parity error occurs, the hardware invokes this routine to examine a register and ensure the reason for the interrupt is a memory parity error. If no memory parity error occurred, the routine returns to the interrupted operation. The parity checking aspect is never impaired. Note the I/O channel error and parity error sources of an NMI can be disabled.

Input:

None

Output:

Error message; system halts

B.3 PRINT SCREEN (INT 05H)

Invoking this routine prints the screen. The current cursor position when the routine is invoked is saved and restored upon completion. During execution of this routine, the print screen key is ignored.

Input:

None

Output:

Location 50:00H

= 0 Normal termination or print

screen not busy

= 1 Print screen in progress

= FFH Error during print screen

B.4 SYSTEM TIMER H/W INTERRUPT (INT 08H)

This interrupt occurs when channel zero of the system timer counts down to zero. The BIOS handler keeps a count of interrupts starting at power-on for use as a system clock. The BIOS handler also decrements the disk drive motor control counter and shuts off the drive when the counter expires. In addition, the BIOS handler invokes INT 1CH allowing a user routine to execute.

Input:

None

Output:

B.5 KEYBOARD H/W INTERRUPT (INT 09H)

This interrupt occurs each time a key is pressed. The BIOS handler checks to see if certain keys are being pressed or if a combination of keys are being pressed, for example \(\text{Ctrl} + \text{Att} + \text{Delete} \) or \(\text{Pause} \). If either situation occurs, the BIOS handler takes appropriate action. Otherwise, the key scan-code and its ASCII value, if applicable, are moved into the keyboard buffer (provided sufficient space is available).

Input:

None

Output:

None

B.6 DISK H/W INTERRUPT (INT 0EH)

This interrupt occurs when the disk drive requires attention. During multi-sector transfers, the controller interrupts for each transfer. A completed operation also generates an interrupt. The BIOS handler sets bit 7 of the byte at 40:30H indicating an interrupt occurred.

Input:

None

Output:

B.7 VIDEO I/O (INT 10H)

This program interface allows control of the video display. Table B-1 defines the AH values for the video I/O.

Table B-1. Video I/O

AH Value	Function
00H	Set display mode
01H	Set cursor size
02H	Set cursor position
03H	Read cursor position
04H	Read light pen position
05H	Select active display page (valid only for text modes)
06H	Scroll window up
07H	Scroll window down
08H	Read attribute/character at current cursor position
09H	Write attribute/character at current cursor position
OAH	Write character only at current cursor position
0BH	Set color palette
0CH	Write dot
0DH	Read dot
0EH	Write character as teletype to active page
0FH	Get current video mode
13H	Write string

These functions are contained in the system BIOS and can be enhanced or replaced by the functions contained in the BIOS ROM on a plug-in video controller. Each function is described in the following text.

B.7.1 Set Display Mode

This function sets the display mode for the active controller. The monochrome display controller only uses modes 0 and 7.

Input:

AH = 00H

AL Mode value

Text Mode:

AL = 00H 40 x 25 B&W AL = 01H 40 x 25 color AL = 02H 80 x 25 B&W AL = 03H 80 x 25 color AL = 07H 80 x 25 B&W card

Graphics Mode:

AL = 04H 320 x 200 color AL = 05H 320 x 200 B&W AL = 06H 640 x 200 B&W AL = 11H 640 x 480 color

Enhanced Graphics Mode:

AL = 0DH 320 x 200 16 colors AL = 0EH 640 x 200 16 colors AL = 0FH 640 x 350 B&W

 $AL = 10H 640 \times 350 16/64 \text{ colors}$

Output:

B.7.2 Set Cursor Size

This function programs the CRTC cursor start/end register to set the desired cursor size.

Input:

AH = 01H

Set cursor type

CH (bits 4:0)

Start line for cursor

(bit 5)

Control cursor display

0 = Normal 1 = No cursor

٠,

Set to 0

(bit 6)

CL (bits 4:0) End line for cursor

Output:

None

B.7.3 Set Cursor Position

This function programs the CRTC cursor position register to display the cursor at the desired location. When the specified page number differs from the active display page, no visible reaction occurs.

Input:

AH = 02H

DH,DL BH Row, column (0,0 upper left corner)

Page number (must be 0 for graphics

modes)

Output:

B.7.4 Read Cursor Position

This function returns the current cursor position of the specified page.

Input:

AH = 03H

BH

Page number (must be 0 for graphics

modes)

Output:

DH,DL

Row, column of current cursor position

CH,CL

Cursor size parameters

B.7.5 Read Light Pen Position

This function returns the current light pen position.

Input:

AH = 04H

Output:

AH = 00H

Light pen switch not down/not

triggered

AH = 01H

Valid light pen value in registers

DH,DL

Row, column of light pen character position

CH

Raster line (0:199)

BX

Pixel column (0:319, 0:639)

B.7.6 Select Active Display Page

This function selects the active display page and is valid only for text modes.

Input:

AH = 05H

AL

New page value (0:7 for modes 0 & 1,

0:3 for modes 2 & 3)

Output:

B.7.7 Scroll Window Up

This function scrolls up the given area in the active page.

Input:

AH = 06H

AL

Number of lines blanked at

bottom of window

0 = Blank entire window

CH,CL

Row, column of upper left corner of

window

DH,DL

Row, column of lower right corner of

window

BH

Attribute used on blank line

Output:

None

Scroll Window Down B.7.8

This function scrolls down the given area in the active page.

Input:

AH = 07H

ΑL

Number of input lines blanked at top of

window

0 = Blank entire window

CH,CL DH,DL Row, column of upper left corner of window

Row, column of lower right corner of window

BH

Attribute used on blank line

Output:

B.7.9 Read Attribute/Character at Current Cursor Position

This function reads the attribute and character displayed at the cursor position.

Input:

AH = 08H

BH

Display page (used for text modes only)

Output:

AL

Character read

AΗ

Attribute of character read (text modes only)

B.7.10 Write Attribute/Character at Current Cursor Position

This function writes the specified attribute and character to the display at the current cursor position. Cursor position does not change.

Input:

AH = 09H

AL

Character to write

BH

Display page (used for text modes only)

BL

Attribute of character (text) or

CX

color of character (graphics)

Number of times to write the character and

attribute

In graphics mode the write does not

continue to the next line

Output:

B.7.11 Write Character Only at Current Cursor Position

This function writes the character on the display at the current cursor position. Cursor position is unchanged.

Input:

AH = 0AH

AL

Character written

BH CX Display page (used for text modes only)

Number of times to write the character (in

graphics mode the write does not continue

to the next line)

Output:

None

B.7.12 Set Color Palette

This function programs the CRTC border control register to set the desired color.

Input:

AH = OBH

BH

Color ID set (0-1)

BL BH = 00H Color value used with color ID Set background color for 320 x 200

graphics modes

Set border color for alphanumeric modes Set foreground color for 640 x 200 graphics

BL = 0-31

BH = 01H

Select palette for 320 x 200 graphics

BL = 0BI = 1 Green (1)/red (2)/brown (3)

Cyan (1)/magenta (2)/white (3)

Output:

B.7.13 Write Dot

This function writes a dot at the specified location.

Input:

AH = 0CH

Αl

Color value

The color value is exclusive read with the

current color at that location if

bit 7 = 1

CX DX Column number Row number

Output:

None

B.7.14 Read Dot

This function reads the dot at the specified location.

Input:

AH = 0DH

CX

Column number

DX

Row number

Output:

AL

Color data

B.7.15 Write Character as Teletype to Active Page

This function writes a character at the cursor position of the active page and moves the cursor to the next position. The attribute remains the same in text mode. The cursor moves to the next line and scrolls the screen, if applicable.

Input:

AH = 0EH

AL

Character to write

BL

Foreground color in graphics mode

Output:

B.7.16 Get Current Video Mode

This function returns the current video mode.

Input:

AH = OFH

Output:

AΗ

Number of columns on screen

ΑI

Mode currently set

BH

Current active display page

Write String B.7.17

This function displays a string of characters on the display.

Input:

AH = 13H

ES:BP Pointer to character string to write CX Length of character string to write

DH.DL

Row, column string write BH Page number

AL = 00H

BL = Attribute, string {char, ..., char}

Cursor does not move

AL = 01H

BL = Attribute, string {char, ..., char}

Cursor moves

AL = 02H

String {char, attr, ..., char, attr}

Cursor does not move

AL = 03H

String {char, attr, ..., char, attr}

Cursor moves

Output:

B.8 EQUIPMENT DETERMINATION (INT 11H)

This program interface describes the hardware installed in the system.

Input:	None	
Output:	AX Bits 15:14 Bit 13 Bit 12 Bits 11:9 Bit 8 Bits 7:6 Bits 5:4	Number of printers attached Internal modem Not used Number of RS-232 ports attached Not used Number of floppy disk drives (00 = 1; 01 = 2 only if bit 0 = 1) Initial video mode 00 = Not used 01 = 40 x 25 color 10 = 80 x 25 color 11 = 80 x 25 monochrome
	Bit 3 Bit 2	Not used Pointing device 1 = Pointing device installed 2 = Pointing device not installed
	Bit 1	Numeric coprocessor 1 = Numeric coprocessor installed 0 = Numeric coprocessor not installed
	Bit 0	Floppy disk drive 1 = Floppy disk drive installed 0 = Floppy disk drive not installed

B.9 MEMORY SIZE DETERMINATION (INT 12H)

This routine returns the RAM size below address 10000H.

Input:

None

Output:

AX

Number of contiguous 1K blocks of

memory

B.10 FLOPPY DISK DRIVE I/O (INT 13H, PART 1)

This interface provides access to the disk drives supported by the system. Table B-2 defines the AH values for the Floppy Disk Drive I/O (part 1).

Table B-2. Floppy Disk Drive I/O (part 1)

AH Value	Function	
00H	Reset disk drive	
01H	Read status	
02H	Disk read	
03H	Disk write	
04H	Disk verify	
05H	Format disk track	
08H	Disk drive parameters	
15H	Read direct access storage device (DASD) type	
16H	Disk change line status	
17H	Set DASD type for format	
18H	Set media type for format	

Input:	AH	Function number
•	AL	Number of sectors
	CH	Track number
	CL	Sector number
	DH	Head number
	DL	Drive number
	ES:BX	Transfer address

Output:

Carry flag = 1 If there is an operation error

Carry flag = 0 If operation is correct AH Status of operation

AH = 00H

No error

AH = 01HAH = 02H Invalid function request Address mark not found

AH = 03H

Write protect error

AH = 04H

Requested sector not found

AH = 06H

Media changed DMA overrun

AH = 08HAH = 09H

Attempt to DMA across 64K

boundary

AH = 0CHAH = 10H Media type not found CRC error on disk read

AH = 20HAH = 40H General controller failure Seek operation failed

H08 = HA

Timeout

B.10.1 Reset Disk Drive

Input:

AH = 00H

Output:

DL

Drive number (0-based)

Bit 7=0 floppy disk

B.10.2 Read Status

Input:

AH = 01H

Output:

DL

Drive number (0-based)

Bit 7=0 floppy disk (value checked)

B.10.3 Disk Read

Input: AH = 02H

AL Number of sectors
CH Track number
CL Sector number
DH Head number
DL Drive number
ES:BX Address of buffer

Output: AH Status of operation

AL Number of sectors transferred

CY = 1 Error CY = 0 No error

B.10.4 Disk Write

Input: AH = 03H

AL Number of sectors
CH Track number
CL Sector number
DH Head number
DL Drive number
ES:BX Address of buffer

Output: AH Status of operation

AL Number of sectors transferred

CY = 1 Error CY = 0 No error

B.10.5 Disk Verify

Input: AH = 04H

AL Number of sectors
CH Track number
CL Sector number
DH Head number
DL Drive number

Output: AH Status of operation

AL Number of sectors verified

CY = 1 Error CY = 0 No error

B.10.6 Format Disk Track

Input: AH = 05H

AL Number of sectors
CH Track number
DH Head number
DL Drive number

ES:BX Address of buffer containing a series of

4 byte fields for each sector:

Byte 1 = Track
Byte 2 = Head
Byte 3 = Sector
Byte 4 = Bytes/sector
0 = 128 bytes/sector
1 = 256 bytes/sector
2 = 512 bytes/sector
3 = 1024 bytes/sector

Output: AH Status of operation

CY = 1 Error CY = 0 No error

B.10.7 Disk Drive Parameters

Input:

H80 = HA

DL

Drive number

Output:

ES:DI

Pointer to drive parameter table

CH

Maximum number of tracks

(low order 8 bits)

CL (bits 7:6)

Maximum number of tracks

(high order 2 bits)

(bits 5:0)

Maximum sectors per track

DH

Maximum head number

DL

Number of floppy disk drives installed

BH

0

BL (bits 7:4)

(bits 3:0)

3:0) Valid drive type value in CMOS RAM

01 - 360K drive

02 - 1.2M drive 03 - 720K drive

04 - 1.4M drive

AX

0

B.10.8 Read DASD Type

Input:

AH = 15H

DL

Drive number 0-3=floppy

Output:

AH = 00H

Drive not present

AH = 01H Disk, AH = 02H Disk.

Disk, no change line available Disk, change line available

AH = 03H Fixed disk drive

CY = 1 Error CY = 0 No error

B.10.9 Disk Change Line Status

Input:

AH = 16H

DL

Drive number

Output:

AH = 01H Invalid parameter

AH = 80H Drive not ready

CY = 1H

AH=0

AH = 0H Floppy disk change line not active
AH = 01H Invalid floppy disk parameter
AH = 06H Floppy disk change line active

AH = 80H Floppy disk drive not ready

B.10.10 Set DASD Type for Format

Input:

AH = 17H

AL = 00H

Not used

 $\begin{array}{ll} \text{AL} = 01 \text{H} & 360 \text{K floppy disk in } 360 \text{K drive} \\ \text{AL} = 02 \text{H} & 360 \text{K floppy disk in } 1.2 \text{M drive} \\ \text{AL} = 03 \text{H} & 1.2 \text{M floppy disk in } 1.2 \text{M drive} \\ \text{AL} = 04 \text{H} & 720 \text{K floppy disk in } 720 \text{K drive} \\ \end{array}$

All other are invalid

DL

Drive number 0-3

Output:

ΑH

Status of operation

CY = 1

Error No error

CY = 0

B.10.11 Set Media Type for Format

Input:

AH = 18H

CH

Maximum number of tracks

(low order 8 bits)

CL (bits 7:6)

Maximum number of tracks

(high order 2 bits)

(bits 5:0)

Maximum sectors per track

DL

Drive number

Output:

ES:DI

Pointer to drive parameter table for this

media type. Unchanged if AH is non-zero

CY = 0

AH = 00H

Successful. Track and sector

combination supported

CY = 1

AH = 01H Function not available

AH = 0CH Track and sector combination not

supported

B.11 FIXED DISK I/O (INT 13H, PART 2)

This interface provides access to fixed disk drives through the fixed/floppy combo controller. Table B-3 defines the AH values for the Fixed Disk I/O.

Table B-3. Fixed Disk I/O (part 2)

AH Value	Function	
00H	Disk reset	
01H	Read status	
02H	Read disk	
03H	Write disk	
04H	Verify disk sectors	
05H	Format disk track	
H80	Disk drive parameters	
09H	Initialize disk parameters	
0AH	Disk read long	
0BH	Disk write long	
0CH	Disk seek	
0DH	Disk alternate reset	
10H	Disk ready test	
11H	Disk recalibrate	
14H	Disk diagnostics	
15H	Read DASD type	

Input:	AH	Function number
	AL	Number of sectors
	CH	Cylinder number (0:1023)
	CL	Sector number (1:17)
		High 2 bits of cylinder number are
		placed in the high 2 bits of the CL register
	DH	Head number (0:15 are allowed)
	DL	Drive number (80H:81H)
	ES:BX	Transfer address

See individual functions

Output:

Output: Carry flag = 1 If there is an operation error

Carry flag = 0 If operation is correct

AH Status of operation

AH = 00H No error

AH = 01H Invalid function passed or invalid parameter

AH = 02H Address mark not found AH = 04H Requested sector not found

AH = 05H Reset failed

AH = 07H Drive parameter activity failed AH = 08H DMA overrun on operation

AH = 09H Attempt to DMA across 64K boundary

AH = 0AH Bad sector flag detected AH = 0BH Bad cylinder detected

AH = 0DH Invalid number of sectors on format
AH = 0EH Control data address mark detected
AH = 0FH DMA arbitration level out of range
AH = 10H Uncorrectable ECC or CRC error

AH = 11H ECC corrected data error AH = 20H General controller failure AH = 40H Seek operation failed

AH = 80H Timeout

AH = AAH Drive not ready or not selected AH = BBH Undefined error occurred

AH = CCH Write fault error on selected drive AH = E0H Status error/error register = 0

AH = FFH Sense operation failed

B.11.1 Disk Reset

Input: AH = 00H

DL Drive number

Output: AH Status CY = 1 Error

CY = 0 No error

B.11.2 Read Status

Input: AH = 01H

DL Drive number bit 7=1

Output: AH Status of the system

CY = 1 Error CY = 0 No error

B.11.3 Read Disk

Input: AH = 02H

AL Number of sectors

CH Cylinder number (low order 8 bits)
CL (bits 7:6) Cylinder number (high order 2 bits)

(bits 5:0) Sector number

DL Drive number bit 7=1

DH Head number ES:BX Address of buffer

Output: AH Status of operation

CY = 1 Error CY = 0 No error

B.11.4 Write Disk

Input: AH = 03H

AL Number of sectors

CH Cylinder number (low order 8 bits)
CL (bits 7:6) Cylinder number (high order 2 bits)

(bits 5:0) Sector number
DH Head number
DL Drive number
ES:BX Address of buffer

Output:

ΑН

Status of operation

CY = 1CY = 0 Error No error

B.11.5 **Verify Disk Sectors**

Input:

AH = 04H

ΑL

Number of sectors

CH CL (bits 7:6) Cylinder number (low order 8 bits) Cylinder number (high order 2 bits)

(bits 5:0)

Sector number

DL

Drive number

DH

Head number

Output:

CY = 1

Error

CY = 0

No error

Format Disk Track B.11.6

Input:

AH = 05H

AL

Number of sectors

CH

Cylinder number (low order 8 bits) CL (bits 7:6) Cylinder number (high order 2 bits)

(bits 5:0)

Sector number

DL

Drive number

DH

Head number

ES:BX

Address of buffer for reads and writes

This points to a 512 byte buffer

The first 2 x (number of sectors/track) bytes

contain F, N for each sector

where: F = 00H for a good sector,

80H for a bad sector and N = logical sector number

Output:

AΗ

Status of operation

CY = 1

Error

CY = 0

No error

B.11.7 Disk Drive Parameters

Input:

H80 = HA

DL

Drive number

Output:

AX = 0

BH = 0

DL Number of disk drives installed
DH Maximum usable head number
CH Cylinder number (low order 8 bits)
CL (bits 7:6) Cylinder number (high order 2 bits)

(bits 5:0) Sector number

CY = 1 Error CY = 0 No error

B.11.8 Initialize Disk Parameters

Input:

AH = 09H

DL

Drive number

Output:

AH

Status

CY = 1

Error

CY = 0

No error

B.11.9 Disk Read Long

Input:

AH = 0AH

AL

Number of sectors

CH

Cylinder number (low order 8 bits) Cylinder number (high order 2 bits)

CL (bits 7:6) Cy (bits 5:0) Se

Sector number

DH DL Head number Drive number

ES:BX

Buffer address (Must accommodate four

bytes of ECC per sector)

Output:

AΗ

Status of operation

CY = 1CY = 0 Error No error

B.11.10 **Disk Write Long**

Input:

AH = OBH

AL

Number of sectors

CH CL (bits 7:6) Cylinder number (low order 8 bits) Cylinder number (high order 2 bits)

(bits 5:0)

Sector number

DH DL

Head number Drive number

ES:BX

Buffer address

Each sector of data is followed by the 4-byte

ECC

Output:

ΑН

Status of operation

CY = 1

Error

CY = 0

No error

Disk Seek B.11.11

Input:

AH = 0CH

CH

Cylinder number (low order 8 bits)

CL (bits 7:6) (bits 5:0) Cylinder number (high order 2 bits)

DH

Sector number

DL

Head number Drive number

Output:

AΗ

Status of operation

CY = 1

Error

CY = 0

No error

B.11.12 Disk Alternate Reset

This function is the same as function 00H.

Input:

AH = 0DH

DL

Drive number

Output:

ΑН

Status

CY = 1

Error

CY = 0

No error

B.11.13 Disk Ready Test

Input:

AH = 10H

DL

Drive number

Output:

AΗ

Status (if fixed disk drive present)

CY = 1CY = 0 Error No error

B.11.14 Disk Recalibrate

Input:

AH = 11H

DL

Drive number

Output:

AΗ

Status (if fixed disk drive present)

CY = 1

Error

CY = 0

No error

Disk Diagnostics B.11.15

Input:

AH = 14H

DL

Drive number

Output:

AΗ CY = 1 Status Error

CY = 0

No error

Read DASD Type B.11.16

This function is the same as the Floppy Disk function (15H).

Input:

AH = 15H

DL

Drive number

Output:

AH = 00HAH = 01H Drive not present

AH = 02H

Disk, no change line available Disk, change line available

AH = 03HFixed disk drive

CX:DX

Contains total number of sectors

CY = 1

Invalid drive number

CY = 0No error

B.12 RS-232 I/O (INT 14H)

This program interface provides access to the asynchronous communication I/O port. The communication port number is not checked. Table B-4 defines the AH values for the RS-232 I/O.

Table B-4. RS-232 I/O

AH Value	Function	
00H 01H 02H 03H	Initialize the communication port Send a character Receive a character Return the communication port status	

Input: AH Function number

DX Communication port number (0:3)

Output: See individual functions

B.12.1 Initialize the Communication Port

Input: AH = 00H

DX Communication port number (0:3)

AL Parameters for initialization

Bits 7:5 Baud rate

000 = 110 bps 001 = 150 bps 010 = 300 bps 011 = 600 bps 100 = 1200 bps

101 = 2400 bps 110 = 4800 bps111 = 9600 bps

Bits 4:3 Parity

00 = No parity 01 = Odd parity 10 = No parity

11 = Even parity

Bit 2 Stop bit

0 = 1 stop bit 1 = 2 stop bits

Bits 1:0 Word length

10 = 7 bits

11 = 8 bits

Output:	AH Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	Line status: Time out Transmitter shift register empty Transmitter holding register empty Break detected Framing error Parity error Overrun error Data ready
	AL Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	Modem status: Received line signal detect Ring indicator Data set ready Clear to send Delta receive line signal detect Trailing edge ring detector Delta data set ready Delta clear to send

B.12.2 Send a Character

Input:

AH = 01H

AL DX Character

Communication port number (0:3)

Output:

AL AH Character

Line status

(see Section B.12.1 for descriptions)

B.12.3 Receive a Character

Input:

AH = 02H

DX

Communication port number (0:3)

Output:

AH = 00H

No error

AL

Character received

AH

Line status

(see Section B.12.1 for descriptions)

B.12.4 Return the Communication Port Status

Input:

AH = 03H

DX

Communication port number (0:3)

Output:

Refer to Section B.12.1.

B.13 SYSTEM SERVICE ROUTINES (INT 15H)

This program interface allows access to the 386 extended functions. Table B-5 defines the AH values for the System Service Routines.

Table B-5. System Service Routines

AH Value	Function	
4F	Keyboard intercept (null)	
80	Device open (null)	
81	Device close (null)	
82	Program termination (null)	
83	Event wait (null)	
84	Joystick support	
85	System request key pressed (null)	
86	Wait	
87	Move block	
88	Extended memory size determination	
89	Switch coprocessor to virtual mode	
90	Device busy (null)	
91	Interrupt complete (null)	
CO	Return system configuration parameters	

Input: AH Function number

Output: See individual functions

B.13.1 Keyboard Intercept

The hardware keyboard interrupt INT 09H routine calls the keyboard intercept asynchronously. This allows keystroke changes. Normally the system returns with the scan code unchanged, but the operating system can redirect interrupt 15H to its own routine and

- Replace (AL) with a different scan code and return with the carry flag set, effectively changing the keystroke.
- Process the keystroke and return with the carry flag cleared, causing the interrupt 09H routine to ignore the keystroke.

The RET 2 instruction must be used when returning from scan code handling. This clears the stack and prevents the flag just set from being overwritten with its previous state.

Input: AH = 4FH

Output: AL Scan code

CY = 1 Handle code CY = 0 Ignore keystroke

B.13.2 Device Open

This function reserved for the operating system.

H08 = HA

Input:

BX Device ID

CX Process ID

Output: AH = 0

CY = 0

B.13.3 Device Close

This function reserved for the operating system.

Input: AH = 81H

BX Device ID

CX Process ID

Output: AH = 0

CY = 0

B.13.4 Program Termination

This function reserved for the operating system.

Input: AH = 82H

BX Device ID

Output: AH = 0

CY = 0

B.13.5 Event Wait

This function sets a timer which counts off CX:DX microseconds. When the timer expires, the high bit of the byte pointed to by ES:BX is set. Make sure the bit is cleared initially, and monitor the bit after invoking this function. The real-time clock periodic interrupt counts the interval (1/1024th of a second). The event wait handler takes this into account, however, the shortest interval available is 1 ms.

Input: AH = 83H

AL = 0 Set interval AL = 1 Cancel

ES:BX Pointer to a byte

CX:DX Number of microseconds to wait

Output: CY = 0 Not busy (AL not equal to 0)

CY = 1 Busy (AL = 0)

AH = 0

B.13.6 Joystick Support

This routine supports an interface between a joystick interface and the system.

Input: AH = 84H

DX = 0 Read current switch settings

DX = 1 Read resistive inputs

Output: CY = 1 Invalid call

If DX = 0

AL = Switch setting (bits 7:4)

If DX = 1

AX = a(x) value BX = a(y) value CX = b(x) value DX = b(y) value

B.13.7 System Request Key Pressed

Pressing the system request (SysRq) key loads the BIOS keyboard hardware interrupt routine, loads AX with 8500H, and invokes INT 15H. This vector can be revectored. Releasing the key invokes INT 15H, this time with 8501H in AX.

Input:

AH = 85H

AL = 00 key pressed AL = 01 key released

Output:

AH = 0

CY = 0

B.13.8 Wait

This function waits for the specified number of microseconds before returning to caller. The real-time clock periodic interrupt counts the interval (1/1024th of a second). The event wait handler takes this into account, however, the shortest interval available is 1 ms.

Input:

AH = 86H

CX.DX

Number of microseconds to elapse before

return to caller

Output:

CY = 0

Function successful

CY = 1

Wait function already in progress

B.13.9 Move Block

This function transfers up to 32,768 words to or from extended memory.

Input:

AH = 87H

CX

Number of words to move, maximum count

= 8000H (32K) words

ES:SI

Global descriptor table (GDT) pointer. User

must set up the GDT.

Output:

AH = 00H

Successful

AH = 01H

RAM parity (parity error cleared on return)

AH = 02H

Exception interrupt occurred

AH = 03H

Gate address line 20 failed

CY = 0

Successful

CY = 1

Error

B.13.10 Extended Memory Size Determination

This routine returns the number of consecutive 1K blocks above 1M of memory. CMOS RAM locations 30 and 31H, set when the system was started, obtains this value.

Input:

AH = 88H

Output:

ΑX

Number of consecutive 1K blocks starting

at 1M

B.13.11 Switch CPU to Protected Mode

This routine sets the CPU to protected mode.

Input:

AH = 89H

ES:SI BH Pointer to GDT. User must set up the GDT Offset into interrupt descriptor table (IDT)

where first eight 8259 interrupts are to

occur

BL

Offset into IDT where second eight 8259

interrupts occur

Output:

AH = 0

If successful

Segment registers set for protected mode

operation, AX and BP destroyed

B.13.12 Device Busy

This function reserved for the operating system.

Input:

AH = 90H

AL

Type code

Output:

CY = 0CY = 1 No error Timeout

<u>Type</u>	Description	Timeout
00	Fixed disk	yes
01	Disk	yes
02	Keyboard	no
80	Network	no
FC	Fixed disk reset	yes
FD	Drive motor start	yes
FE	Printer	yes

B.13.13 Interrupt Complete

This function reserved for the operating system.

Input:

AH = 91H

ΑL

Type code

Output:

None, AX destroyed

B.13.14 Return System Configuration Parameters

This routine gives information about the model of the board, the BIOS revision level, and hardware features.

Input:

AH = COH

Output:

CY = 0

AH = 0

ES:BX

Pointer to system descriptor table in ROM

Table B-6 provides information on the system configuration.

Table B-6. System Descriptor Table

Size	Description	Value
Word Byte	Descriptor length (bytes) Model	008 FC
Byte	Submodel	01
Byte	BIOS revision level	00
Byte	Feature information	70
Bit 7 = 1	BIOS uses DMA channel 3	
Bit 6 = 1	One interrupt controller	
Bit 5 = 1	Real-time clock present	
Bit 4 = 0	System hook in keyboard interrupt routine	
Bit 3 = 0	Reserved	
Bit 2 = 0	Extended BIOS data area allocated	
Bit 1 = 0	Reserved	
Bit 0 = 0	Reserved	

The A20 Gate control routine will reside at absolute memory location 0F000H:0FF82H. Table B-7 lists information used to determine if services for the A20 Gate are present.

Table B-7. Gate A20 Control Routine

ES:BX + Offset	Table Offset	Description	
0A	0	A word entry indicating the nominal size of the table (0008H).	
0C	2	Byte entry	
0D	3	Byte entry	
0E	4	String: 'PTL'	
11	7	Phoenix Feature Byte 1	
12	8	Phoenix Feature Byte 2 (nominally 0)	
13	9	Phoenix Feature Byte 3 (nominally 0)	

Bit 7 of Phoenix Feature byte 1 is set if the Gate A20 control routine is supported, otherwise it will be cleared.

Input:	AH = 8 AL = 0 AL = 1 AL = 2	Identify the call as an A20 call Attempt to disable A20 Attempt to enable A20 The current state of A20 is returned in AH If enabled, AH = 1 If disabled, AH = 0
Output:	CY = 1	An error has been encountered in processing
	CY = 0	No error

B.14 KEYBOARD I/O (INT 16H)

This program interface provides an interface to access keys placed into the keyboard buffer by the keyboard hardware interrupt. Table B-8 defines the AH values for the Keyboard I/O.

Table B-8. Keyboard I/O

AH Value	Function	
00H 01H 02H 03H 05H 10H 11H	Read next character Read buffer status Return shift status Set typematic rate and delay Place ASCII character/scan code in keyboard buffer Extended read interface for the enhanced keyboard Extended buffer status for the enhanced keyboard Return the extended shift status for the enhanced keyboard	

B.14.1 Read Next Character

This function returns the scan code and ASCII code of the next character in the keyboard buffer, and updates the buffer pointer. If the keyboard buffer is empty, the function waits for a key.

Input:

AH = 00H

Output:

AH AL Scan code

ASCII character (0 for special keys, function

keys, etc.)

B.14.2 Read Buffer Status

This function returns the keyboard buffer status (indicates whether a keystroke is available or not). The buffer pointer is not updated, even though the key returns. Use function 00H to update the buffer pointer.

Input:

AH = 01H

Output:

ZF = 1

No keystroke queued

ZF = 0

Keystroke available in queue, key in AX

B.14.3 Return Shift Status

Pressing a shift key does not transfer the scan code/ASCII code into the keyboard buffer. Instead, a shift status bit is set. This function indicates which key is pressed.

Input:	AH = 02H
ii iput.	/ (I I OEI I

Output:	AL	Shift status
	Bit $0 = 1$	Right-Shift pressed
	Bit 1 = 1	Left-Shift pressed
	Bit $2 = 1$	Ctrl-Shift pressed
	Bit $3 = 1$	Alt-Shift pressed
	Bit $4 = 1$	Scroll Lock state
	Bit $5 = 1$	Num Lock state

Bit 6 = 1

Bit 7 = 1 Insert state
All other registers are restored

Caps Lock state

B.14.4 Set Typematic Rate and Delay

Pressing most keys causes the keyboard to repeat the key until the key is released. This function is used to set the typematic rate and delay time. The typematic rate is the rate at which the keyboard repeats the key. Delay time is the time between the first keystroke and the repeated keystrokes.

Input:	AH = 03H			
	BL	Typema	atic rate (bits 4:0)	
	BL = 00H	30.0	BL = 10H	7.5
	BL = 01H	26.7	BL = 11H	6.7
	BL = 02H	24.0	BL = 12H	6.0
	BL = 03H	21.8	BL = 13H	5.5
	BL = 04H	20.0	BL = 14H	5.0
	BL = 05H	18.5	BL = 15H	4.6
	BL = 06H	17.1	BL = 16H	4.3
	BL = 07H	16.0	BL = 17H	4.0
	BL = 08H	15.0	BL = 18H	3.7
	BL = 09H	13.3	BL = 19H	3.3
	BL = 0AH	12.0	BL = 1AH	3.0
	BL = 0BH	10.9	BL = 1BH	2.7
	BL = 0CH	10.0	BL = 1CH	2.5
	BL = 0DH	9.2	BL = 1DH	2.3
	BL = 0EH	8.5	BL = 1EH	2.1
	BL = 0FH	8.0	BL = 1FH	2.0
	BH	Delay v	alue (bits 1:0)	
	BH = 00H	250 ms		
	BH = 01H	500 ms		
	BH = 02H	750 ms		
	BH = 03H	1000 m	S	

None

Output:

B.14.5 Place ASCII Character/Scan Code in Keyboard Buffer

This function places an ASCII character/scan code combination into the keyboard buffer.

Input:

AH = 05H

CL CH ASCII code Scan code

Output:

AL = 00H

Successful operation

AL = 01H

Keyboard buffer full

B.14.6 Extended Read Interface for the Enhanced Keyboard

This function corresponds to AH = 0, but supports the enhanced keyboard.

Input:

AH = 10H

Output:

AΗ

Scan code

AL

ASCII code

B.14.7 Extended Buffer Status for the Enhanced Keyboard

This function corresponds to AH = 1, but supports the enhanced keyboard.

Input:

AH = 11H

Output:

ZF = 1

No keystroke queued

ZF = 0

Keystroke is available, key in AX

B.14.8 Return the Extended Shift Status for the Enhanced Keyboard

This function corresponds to AH = 2, but supports the enhanced keyboard.

Input:

AH = 12H

Output:

AL

Bit 0 = 1

Bit 1 = 1

Bit 2 = 1

Control key pressed

Bit 3 = 1

Alt key pressed

Control key pressed

Alt key pressed

Control key pressed

Control key pressed

Bit 4 = 1 Scroll Lock state
Bit 5 = 1 Num Lock state
Bit 6 = 1 Caps key state
Bit 7 = 1 Insert state

Output:

ΑН

Bit 0 = 1Left-Ctrl-Shift Bit 1 = 1Left-Alt-Shift Bit 2 = 1Right-Ctrl-Shift Bit 3 = 1Right-Alt-Shift Bit 4 = 1Scroll-Lock-Shift Bit 5 = 1Num-Lock-Shift Bit 6 = 1Caps-Lock-Shift Bit 7 = 1SysRq-Shift

B.15 PRINTER I/O (INT 17H)

This program interface allows access to the printer. Table B-9 defines the AH values for the Printer I/O.

Table B-9. Printer I/O

AH Value	Function
00H	Print character
01H	Initialize printer port
02H	Read printer status

Input: AH Function number DX Device number (0:2)

Output: See individual functions

B.15.1 Print Character

This function writes the character in AL to the specified printer.

Input: AH = 00HAL Character DΧ Device number (0:2) Output: AH = OFHInvalid port AΗ Printer status Bit 0 = 1Timeout Bits 1:2 Reserved Bit 3 = 1I/O error Bit 4 = 1Printer selected Bit 5 = 1Out of paper Bit 6 = 1Acknowledge

Bit 7 = 1

Not busy

B.15.2 Initialize Printer Port

Input:

AH = 01H

DX

Device number (0:2)

Output:

 $\begin{array}{ll} \text{AH} & \text{Printer status} \\ \text{Bit 0} = 1 & \text{Timeout} \\ \text{Bits 1:2} & \text{Reserved} \\ \text{Bit 3} = 1 & \text{I/O error} \end{array}$

Bit 4 = 1 Printer selected
Bit 5 = 1 Out of paper
Bit 6 = 1 Acknowledge
Bit 7 = 1 Not busy

B.15.3 Read Printer Status

Input:

AH = 02H

DX

Device number (0:2)

Output:

AH Printer status
Bit 0 = 1 Timeout
Bits 1:2 Reserved
Bit 3 = 1 I/O error

 $\begin{array}{lll} \text{Bit } 4 = 1 & \text{Printer selected} \\ \text{Bit } 5 = 1 & \text{Out of paper} \\ \text{Bit } 6 = 1 & \text{Acknowledge} \\ \text{Bit } 7 = 1 & \text{Not busy} \end{array}$

B.16 SYSTEM BOOT (INT 19H)

This routine reads the boot-sector from the disk drive into main memory and executes it. Memory does not clear unless invoked by the BIOS self-test and initialization. If a disk time-out is encountered, the routine attempts to boot from the fixed disk drive. If this fails, INT 18H is invoked.

Input:

None

Output:

None

B.17 CLOCK SERVICES (INT 1AH)

This BIOS routine allows the clock to be set or read. Table B-10 defines the AH values for the Clock Services.

Table B-10. Clock Services

AH Value	Function		
00H	Read the system timer count		
01H	Set the system timer count		
02H	Read the real-time clock time		
03H	Set the real-time clock time		
04H	Read the real-time clock date		
05H	Set the real-time clock date		
06H	Set the real-time clock alarm		
07H	Reset the real-time clock alarm		
I			

Input:

Function number

Output:

See individual functions

B.17.1 Read the System Timer Count

This function reads the system timer from the BIOS data area. Each system timer interrupt (INT 08H), approximately 18.2 times per second, increments this value.

Input: AH = 00H

Output:

CX High count word

DX Low count word

AL = 0 Timer has not passed midnight since last

read

AL > 0

Timer has passed midnight since last read

AH = 0

B.17.2 Set the System Timer Count

This function sets the system timer in the BIOS data area.

Input:

AH = 01H

CX

High count word

DΧ

Low count word

Output:

AH = 0

B.17.3 Read the Real-time Clock Time

This function reads hours, minutes, and seconds from the real-time clock.

Input:

AH = 02H

Output:

CH
CL
Minutes in BCD
DH
Seconds in BCD
DL = 1
DL = 0
CY = 0
CY = 1

Hours in BCD
Daylight savings time
Standard time
Successful
CY = 1

Clock is busy

B.17.4 Set the Real-time Clock Time

This function sets hours, minutes, and seconds in the real-time clock.

Input: AH = 03H

CH Hours in BCD
CL Minutes in BCD
DH Seconds in BCD
DL = 1 Daylight savings time

DL = 0 Standard time

Output: AH = 00H

CY = 0 Successful CY = 1 Clock is busy

B.17.5 Read the Real-time Clock Date

This function reads century, year, month, and day from the real-time clock.

Input: AH = 04H

Output: CH Century in BCD (19 or 20)

CL Year in BCD
DH Month in BCD
DL Day in BCD

AH = 00H

CY = 0 Successful CY = 1 Clock busy

B.17.6 Set the Real-time Clock Date

This function sets century, year, month, and day into the real-time clock.

Input:

AH = 05H

CH CL Century in BCD (19 or 20) Year in BCD

DH DL

Month in BCD Day in BCD

Output:

AH = 00H

CY = 0

Successful

CY = 1

Clock busy

B.17.7 Set the Real-time Clock Alarm

This function sets the alarm (hours, minutes, and seconds) to the real-time clock. When the time is up, INT 4AH is invoked.

Input:

AH = 06H

СН

Hours in BCD Minutes in BCD

CL DH

Seconds in BCD

Output:

AH = 00H

CY = 0CY = 1 Successful Clock busy

B.17.8 Reset the Real-time Clock Alarm

This function resets the alarm and turns off the alarm enable.

Input:

AH = 07H

Output:

AH = 00H

CY = 0

Successful

CY = 1

Clock busy

B.18 REAL-TIME CLOCK (INT 70H)

The following three conditions in the real-time clock may cause this interrupt:

- The current time matches the time set in the alarm.
- The periodic interval counter expires.
- The real-time clock completes an internal update cycle.

Each separately enabled condition must be enabled for an interrupt to occur. In a standard system, only the periodic interrupt is enabled. BIOS uses this periodic interrupt to time certain events in the system. If an alarm interrupt occurs, BIOS invokes INT 4AH and gives the program a chance to handle the alarm condition. The update cycle complete interrupt is not used.

B.19 COPROCESSOR INTERRUPT (INT 75H)

This interrupt is caused by the 387 numeric coprocessor error which occurs with interrupts enabled at the coprocessor. To retain compatibility with 8088-based systems, the IRQ13 handler issues a software INT 02H command. The software INT 02H command causes a software generated NMI.

If an application preempts the NMI vector to handle coprocessor errors (as it must to retain compatibility with 8088-based systems), it must be sensitive to NMIs generated by hardware, and pass them on to the system NMI.

B.20 FIXED DISK H/W INTERRUPT (INT 76H)

This interrupt occurs when the fixed disk drive requires attention. During multi-sector transfers, the controller interrupts after each transfer. A completed operation also causes an interrupt. The BIOS handler moves 0FFH to 40:8EH indicating the interrupt occurred.

Jumper Settings



C.1 INTRODUCTION

Jumper pins allow specific system operating parameters to be set in the system. Various options require changing jumper blocks to reflect the correct system configuration.

Figure C-1 diagrams the 302 board jumper pin locations. Jumper pin blocks are arranged into the following functional groups:

- Total onboard DRAM
- Onboard ROM size
- Onboard RAM size
- Onboard RAM type
- Onboard RAM speed
- Video display type
- Printer acknowledge
- Parallel printer port
- Serial communication ports

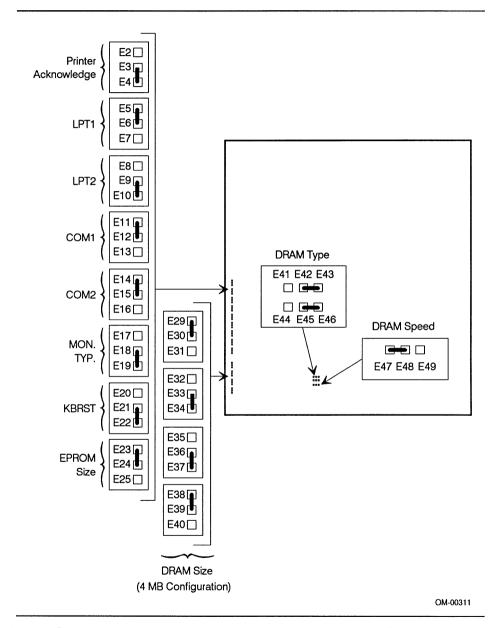


Figure C-1. 302 Board Jumper Locations

C.2 Total Onboard DRAM

The total onboard DRAM jumpers allow you to configure the 302 board for the amount of DRAM installed in the SIMM sockets. Table C-1 lists the four options.

Table C-1. Total Onboard RAM

Amount	Jumper Pin Setting			
1M	E35 to E36, E30 to E31 E33 to E34, E39 to E40			
2M	E32 to E33, E35 to E36 E39 to E40, E30 to E31			
4M	E33 to E34, E36 to E37 E38 to E39, E29 to E30			
8M	E32 to E33, E36 to E37 E38 to E39, E29 to E30			

C.3 ONBOARD ROM SIZE

Total onboard ROM size jumpers allow you to configure the 302 board for the amount of ROM installed. Table C-2 lists the two options.

Table C-2. Onboard ROM Size

Amount	Jumper Pin Setting	
64K (27256)	E23 to E24 [†]	
128K (27512)	E24 to E25	

[†] Factory default

C.4 ONBOARD RAM

Onboard RAM jumpers include type and speed. Tables C-3 and C-4 list the RAM options for configuring the 302 board.

Table C-3. Onboard RAM Type

RAM Type	Jumper Pin Setting
Static-column	E42 to E43 E44 to E45
Fast paged mode	E42 to E43† E45 to E46 [†]
Standard RAS/CAS	E41 to E42 E44 to E45

[†] Factory default

Table C-4. Onboard RAM Speed

Speed (ns)	Jumper Pin Setting	
100	E47 to E48 [†]	
85	E48 to E49	

[†] Factory default

C.5 VIDEO DISPLAY TYPE

The video display type jumper allows you to indicate the type of video display connected to the 302 board. Table C-5 lists the two options.

Table C-5. Video Display Type

Monitor Type	Jumper Pin Setting
Color	E18 to E19 [†]
Monochrome	E18 to E17

Factory default

C.6 PRINTER ACKNOWLEDGE

This jumper allows you to set the printer acknowledge signal. Table C-6 lists the two options.

Table C-6. Printer Acknowledge

Condition	Jumper Pin Setting
Leading edge	E2 to E3
Trailing edge	E3 to E4 [†]

[†] Factory default

C.7 PARALLEL PRINTER PORT SELECTION

This jumper allows you to configure the 302 board for the desired parallel printer port. Table C-7 lists the two options.

Table C-7. Parallel Printer Port Selection

Condition	Jumper Pin Setting
LPT1 enabled	E5 to E6 [†]
LPT1 disabled	E6 to E7
LPT2 enabled	E8 to E9
LPT2 disabled	E9 to E10 [†]

[†] Factory default

C.8 SERIAL COMMUNICATIONS PORT SELECTION

This jumper allows you to configure the 302 board for the desired serial communications port. Table C-8 lists the options.

Table C-8. Serial Communications Port Selection

Condition	Jumper Pin Setting
COM1 enabled	E11 to E12 [†]
COM1 disabled	E12 to E13
COM2 enabled	E14 to E15 [†]
COM2 disabled	E15 to E16

[†] Factory default

Changing CPU Speed



D.1 INTRODUCTION

This section describes changing the system speed using the setup program, the keyboard method, and the software method.

D.2 OVERVIEW

The CPU runs at a clock speed of 25 MHz on the 302 board. This gives the 302 board over three times the performance of an 8 MHz IBM AT. However, there are some applications that require a slower operation speed and cannot operate at a clock speed of 25 MHz. To meet this requirement, a special mode can be initiated which enables the 302 board to effectively operate at a slower speed to emulate the performance of a 8 MHz IBM AT. This special mode of operation is called deturbo mode. The opposite of deturbo mode (full speed system operation) is sometimes referred to as turbo mode. Deturbo mode can be enabled in the setup program, from the keyboard, or with a software program. Conversely, the normal or turbo mode can be enabled in the same manner.

Deturbo mode does not actually affect the clock rate of the CPU or the 387 numeric coprocessor (if one is installed). Deturbo mode disables the cache and inserts CPU wait states between each instruction, resulting in an effective CPU speed of 8 MHz.

If the system is put in deturbo mode, it will remain in deturbo mode even after warm reboots (Ctrl + Att + Delete) or other software resets. However, a power-on reset will put the system back to the mode selected by the Setup program.

D.3 KEYBOARD METHOD

The following keystroke sequences can be used to set the system speed.

Eth + All + 1 Hold down Eth and All and press 1 on the numerical

keypad to place the system in deturbo mode.

Hold down cm and Att and press 2 on the numerical

keypad to place the system in turbo mode.

An audible tone occurs when you change the system speed via the keyboard. A low-pitched tone will be emitted when the system is placed into deturbo mode. A high-pitched tone will be emitted when the system is returned to normal operation.

D.4 SETUP METHOD

The system speed can be changed using the setup program. The system speed chosen will determine the speed used by the system each time you turn on the power.

To invoke the setup program, press $\Box + \Box + \Box + \Box$. Press the \Box or \Box key to move the cursor to the CPU speed field. Press the \Box or \Box key to display the turbo or deturbo options.

Exit the setup program by pressing [55]. Setup will save the new system speed and reboot the system.

D.5 SOFTWARE METHOD

Figure D-1 lists the assembly language code needed to place the system in deturbo mode. Figure D-2 shows the code to return the system to full speed operation. Both of these programs run under DOS and can be assembled using the Microsoft Macro Assembler (MASM). Assuming that the program in Figure D-2 is contained in the file FAST.ASM, the following commands can be used to assemble the program and make it ready to run. The procedure for the program shown in Figure D-1 is analogous.

C:>del fast.obj

C:>del fast.exe

```
name
                          slow
                          'SLOW.COM -- puts board into deturbo mode (emulated 8 MHz)'
               title
cr
              egu
                          0dh
                                            ; ASCII carriage return
1f
              equ
                          0ah
                                            ; ASCII line feed
cmnd
              equ
                          64h
                                            ; command register
               segment public
code
              org
                          100h
                                            ; COM file
              assume
                          cs:code,ds:code
start:
                                            ; set ds equal to cs
                          ax, cs
              mov
                          ds.ax
              mov
                          al,Oeah
                                            ; value for deturbo mode
              mov
                          cmnd,al
                                            ; write to command port
              out
                          dx,offset deturbo; tell user that
              mov
                                            ; switch occurred
                          ah.9
                                            ; use DOS function 9 to
              mov
                          21h
                                            ; print the string
              int
              mov
                          ax,4c00h
                                            ; exit back to DOS with
               int
                          21h
                                            ; a return code of zero
deturbo db
              cr, lf, 'Now running in deturbo mode.', cr, lf, '$'
code
              ends
end
               start
```

Figure D-1. Setting Deturbo Mode - Software Method

	name	fast	
	title	'FAST.COM put	s board into turbo mode'
cr	equ	Odh; ASCII carri	- -
1f	equ	Oah; ASCII line	feed
cmnd	equ	64h; command reg	gister
code	segment p	ublic	
	org	100h	; COM file
	assume	cs:code,ds:code	
start:			
	mov	ax, cs	; set ds equal to cs
	mov	ds.ax	·
		uo,ux	
	mov	al,0e5h	; value for turbo mode
	out	cmnd,al	; write to command port
	mov	dx,offset turbo	; tell user that
	11104	ax, or race carbo	; switch occurred
	mov	ah,9	: use DOS function 9 to
	int	21h	; print the string
	•		, prime and acrimg
	mov	ax,4c00h	; exit back to DOS with
	int	21h	; a return code of zero
turbo	db	cr,lf,'Now runni	ng in turbo mode.',cr,lf,'\$'
code	ends		
end	start		

Figure D-2. Setting Turbo Mode - Software Method

Messages



E.1 INTRODUCTION

This appendix describes the various 302 system screen messages and error beep codes. Information is grouped as follows:

- POST and boot error messages
- POST and boot informational messages
- Run-time messages
- Board errors
- Beep codes for fatal errors
- Beep codes for non-fatal errors

E.2 POST AND BOOT MESSAGES

The POST displays messages to indicate errors in hardware, software, or firmware, or to provide other information.

If the POST can display a message on the video display screen, it will beep the speaker twice as the message appears. However, when an error occurs before the video display is initialized, the POST cannot display messages on the screen. POST sounds a series of beeps instead.

The next two sections provide a general grouping of messages, with each group arranged in alphabetical order. Each message is accompanied by a short paragraph describing the message and a recommended solution to the problem.

Italics indicate variable parts of a message such as memory addresses. These variable parts of the message may differ at each occurrence.

E.2.1 POST and Boot Error Messages

Message CACHE memory failure - Disabling CACHE - Strike

the F1 key to continue, F2 to run the SETUP

utility

Possible Cause

The cache memory is defective.

Solution

Contact your service representative.

Message

Floppy disk drive 0 seek failure

Possible Cause

Drive A has either failed or is missing.

Solution

Check that drive A is present and the floppy disk is inserted properly. If they are, then drive A may have failed. If the problem persists, contact your service

representative.

Message

Floppy disk drive 1 seek failure

Possible Cause

Drive B has either failed or is missing.

Solution

Check that drive B is present and the floppy disk is inserted properly. If they are, then drive B may have failed. If the problem persists, contact your service

representative.

Message Floppy disk read failure - strike F1 to retry

boot, F2 for SETUP utility

Possible Cause A nonbootable or defective floppy disk, or the drive

heads may need cleaning.

Solution Replace the floppy disk with a bootable floppy disk or

another copy and try again. Clean the floppy disk drive

heads, if necessary,

Message Floppy disk subsystem reset failed

Possible Cause The floppy disk drive control cable has failed.

Solution Check the floppy disk drive control cable. If the problem

persists, contact your service representative.

Message Display adapter failed; using alternate

Possible Cause The video display type jumper is set incorrectly or the

primary monitor controller has failed.

Solution Check to ensure that the monitor type jumper is set

correctly. Check the primary monitor controller. If the problem persists, contact your service representative.

Message Gate A20 failure

Possible Cause The computer cannot switch into protected mode.

Solution Contact your service representative.

Message Hard disk configuration error

Possible Cause The specified configuration is incorrect.

Solution Rerun the setup program and enter the correct fixed

disk drive type number. If the problem persists, contact

your service representative.

Message Hard disk controller failure

Possible Cause The drive controller board has failed.

Solution Check the connections at both ends of the fixed disk

control and data cables and reseat the drive controller.

If the problem persists, contact your service

representative.

Message Hard disk failure or Hard disk read failure -

strike F1 to retry boot, F2 for Setup utility

Possible Cause The fixed disk is defective.

Solution Check the system configuration and drive type, and

rerun the setup program. Check the connections at both ends of the fixed disk control and data cables and reseat the drive controller. Check the fixed disk drive jumper and termination resistor. If the problem persists,

contact your service representative.

Message Hex-value optional ROM bad checksum = hex-value

Possible Cause A peripheral board contains a defective ROM or its

address conflicts with another board.

Solution Replace the ROM or the peripheral board, or correct the

address conflict. If the problem persists, contact your

service representative.

Message Invalid configuration information - please run

the Setup program

Possible Cause The memory size, the monitor/controller type, or the

number of floppy disk drives is incorrect.

Solution Check the system configuration and rerun the setup

program.

Message Keyboard clock line failure

Possible Cause Either the keyboard or the keyboard cable connection is

defective.

Solution Check the keyboard connection. If the connection is

good, the keyboard may have failed. Try running the keyboard diagnostic test. If the problem persists,

contact your service representative.

Message Keyboard controller failure

Possible Cause The keyboard controller located on the 302 board has

failed.

Solution Contact your service representative.

Message Keyboard data line failure

Possible Cause Either the keyboard or the keyboard cable connection is

defective.

Solution Check the keyboard connection. If the connection is

good, the keyboard may have failed. If the problem

persists, contact your service representative.

Message Keyboard is locked - please unlock - Strike the

F1 key to continue, F2 to run the Setup utility

Possible Cause The system unit keylock is locked.

Solution Unlock the keyboard and try again. If the problem

persists, contact your service representative.

Message Keyboard stuck key failure

Possible Cause One or more keys are pressed or stuck.

Solution Release the key or keys and try again. If the key is still

stuck, there may be debris in the keyboard. Try to shake it loose. If the problem persists, contact your

service representative.



The following seven messages have the same possible cause and solution.

Message Memory address line failure at hex-value, read

hex-value, expecting hex-value

Message Memory data line failure at hex-value, read

hex-value, expecting hex-value

Message Memory high address line failure at hex-value,

read hex-value, expecting hex-value

Message Memory double word logic failure at hex-value,

read hex-value, expecting hex-value

Message Memory odd/even logic failure at hex-value,

read hex-value, expecting hex-value

Message Memory parity failure at hex-value, read hex-

value expecting, hex-value

Message Memory write/read failure at hex-value, read

hex-value, expecting hex-value

Possible Cause One of the SIMMs or associated circuitry has failed.

Solution Check that all SIMMs are installed correctly.

Message No boot device available - strike F1 to retry

boot, F2 for the Setup utility

Possible Cause If booting from a floppy disk, it is nonbootable or

defective, or the floppy disk drive is defective. If booting from a fixed disk, it may not be formatted as a system disk or is defective. The problem could also be in the

drive controller board.

Solution Make sure that the floppy disk in drive A or the fixed disk

contains an operating system. Check the connections at both ends of the fixed disk control and data cables and reseat the drive controller. If the problem persists,

contact your service representative.

Message No boot sector on hard disk - strike F1 to

retry boot, F2 for the Setup utility

Possible Cause The fixed disk does not contain an operating system.

Solution Format the disk with the /S option



CAUTION

This procedure will destroy data on the disk. Refer to your MS-DOS Operations Reference Manual for instructions.

Message No timer tick interrupt

Possible Cause The timer chip on the 302 board may have failed.

Solution Contact your service representative.

Message Not a boot floppy disk - strike F1 to retry

boot, F2 for the Setup utility

Possible Cause The floppy disk in drive A is not formatted as a system

floppy disk.

Solution Replace the floppy disk with a bootable system floppy

disk and try again.

Message Shadow of System BIOS failed - Executing from

ROM - Strike the F1 key to continue, F2 to run

the Setup utility

Possible Cause The system RAM is defective.

Solution Check the installation of the SIMMs.

Message Shadow of Video BIOS failed - Executing from

ROM - Strike the F1 key to continue, F2 to run

the Setup utility

Possible Cause The system RAM is defective or the video BIOS cannot

be shadowed.

Solution Check the installation of all SIMMs. Rerun the setup

program and turn video BIOS shadow off.

Message Shutdown failure

Possible Cause The keyboard controller or its associated logic has

failed.

Solution Contact your service representative.

Message Time-of-day clock stopped

Possible Cause The integral battery in the RTC is probably dead.

Solution Contact your service representative.

Message Time-of-day not set - Please run the Setup

program

Possible Cause The date and time information is not set in the RTC.

Solution Run the setup program and set the date and time.

Message Timer chip counter 2 failed

Possible Cause The PIT on the 302 board may have failed.

Solution Contact your service representative.

Message Timer or interrupt controller bad

Possible Cause The PIT or the PICs on the 302 board may have failed.

Solution Contact your service representative.

Message Unexpected interrupt in protected mode

Possible Cause The system received an interrupt when in protected

mode, probably while testing memory.

Solution Contact your service representative.

E.2.2 POST and Boot Information Messages

These messages do not indicate error conditions.

Message Hex-value Base Memory

Meaning Indicates the amount of base memory that has been

tested successfully.

Message Hex-value extended

Meaning Indicates the amount of extended memory that has been

tested successfully.

Message Decreasing available memory

Meaning This message immediately follows any memory error

message, informing you that memory modules are failing. Check that all SIMMs are installed correctly. Check expansion board memory (if installed), and check

the SIMM jumpers on the 302 board.

Message Memory test terminated by keystroke

Meaning The spacebar was pressed during the memory test.

Reboot the system if you want to rerun the self-test.

Message Strike the F1 key to continue, F2 to run the

Setup utility

Meaning The self-test detected an error prior to boot. Pressing

Fill lets the computer try to boot. Pressing Fill runs the

setup program.

E.3 RUN-TIME MESSAGES

Message I/O card parity interrupt at address hex-value.

Type (S)hut off NMI, (R)eboot, other keys to

continue

Possible Cause A peripheral board has failed.

Solution Type S to shut off the nonmaskable interrupt (NMI). This

will temporarily allow you to continue. If the problem

persists, contact your service representative.

Message Memory parity interrupt at address hex-value.

Type (S)hut off NMI, (R)eboot, other keys to

continue

Possible Cause One or more memory modules has failed.

Solution Type S to shut off the nonmaskable interrupt (NMI). This

will temporarily allow you to continue. Check the

installation of all SIMMs. If the problem persists, contact

your service representative.

Message Unexpected HW interrupt at address hex-value.

Type (R)eboot, other keys to continue

Possible Cause This could be any hardware-related problem.

Solution Recheck all cables, connections, jumpers, and boards.

If the problem persists, contact your service

representative.

Message Unexpected SW interrupt at address hex-value.

Type (R)eboot, other keys to continue.

Possible Cause There is an error in the software program.

Solution Try turning the computer off and then on again. If the

problem persists, contact your software manufacturer's

representative.

E.4 302 BOARD ERRORS

If the POST finds an error and cannot display a message, the system emits a series of beeps to indicate the error and places a value in I/O port 80H.

For example, a failure of bit 3 in the first 64K of RAM is indicated by a 2-1-4 beep code (a burst of two beeps, a single beep, and a burst of four beeps).

Tables E-1 and E-2 list the beep codes and I/O the values that the POST writes to I/O port 80H when it encounters error conditions. Table E-1 lists fatal errors (errors that halt the system). Table E-2 lists the non-fatal errors (errors that are not serious enough to halt the system). The tables also list other conditions that have no beep codes.

In addition to the codes listed in the tables, a long beep followed by one or more short beeps indicates a monitor controller failure. No beep code is sounded if a test is aborted in progress. Many beep codes indicate problems which can only be solved by replacing or repairing boards or other system components.

Table E-1. Beep Codes for Fatal Errors

Beep Code	Description of Error	Contents of I/O Port 80H
none	386 register test in progress	01H
1-1-3	Real-time clock write/read failure	02H
1-1-4	ROM BIOS checksum failure	03H
1-2-1	Programmable interval timer failure	04H
1-2-2	DMA initialization failure	05H
1-2-3	DMA page register write/read failure	06H
1-3-1	RAM refresh verification failure	08H
none	1st 64K RAM test in progress	09H
1-3-3	1st 64K RAM chip or data line failure multi-bit	0AH
1-3-4	1st 64K RAM odd/even logic failure	0BH
1-4-1	1st 64K RAM address line failure	0CH
1-4-2	1st 64K RAM parity test in progress or failure	0DH
2-1-1	Bit 0 1st 64K RAM failure	10H
2-1-2	Bit 1 1st 64K RAM failure	11H
2-1-3	Bit 2 1st 64K RAM failure	12H
2-1-4	Bit 3 1st 64K RAM failure	13H
2-2-1	Bit 4 1st 64K RAM failure	14H
2-2-2	Bit 5 1st 64K RAM failure	15H
2-2-3	Bit 6 1st 64K RAM failure	16H
2-2-4	Bit 7 1st 64K RAM failure	17H
2-3-1	Bit 8 1st 64K RAM failure	18H
2-3-2	Bit 9 1st 64K RAM failure	19H
2-3-3	Bit A 1st 64K RAM failure	1AH
2-3-4	Bit B 1st 64K RAM failure	1BH
2-4-1	Bit C 1st 64K RAM failure	1CH
2-4-2	Bit D 1st 64K RAM failure	1DH
2-4-3	Bit E 1st 64K RAM failure	1EH
2-4-4	Bit F 1st 64K RAM failure	1FH

(continued)

Table E-1. Beep Codes for Fatal Errors (continued)

Beep Code	Description of Error	Contents of I/O Port 80H
3-1-1 3-1-2 3-1-3 3-1-4 none 3-2-4 none	Slave DMA register failure Master DMA register failure Master interrupt mask register failure Slave interrupt mask register failure Interrupt vector loading in progress Keyboard controller test failure Real-time clock power failure or checksum failure	20H 21H 22H 23H 25H 27H 28H

Table E-2. Beep Codes for Non-fatal Errors

Beep Code	Description of Error	Contents of I/O Port 80H
none	Real-time clock configuration	29H
3-3-4	Screen memory test failure	2BH
3-4-1	Screen initialization failure	2CH
3-4-2	Screen retrace test failure	2DH
none	Search for video ROM in progress	2EH
none	Screen running with video ROM	30H
none	Monochrome display operable	31H
none	Color display (40 column) operable	32H
none	Color display (80 column) operable	33H

Device Mapping



F.1 INTRODUCTION

This appendix provides a series of tables listing mapping and address information related to system memory and onboard devices. Topics presented include:

- System memory map
- I/O addresses
 - Control port bit assignments (61H)
 - Auxiliary control port bit assignments (78H and 79H)
- Interrupt priority levels
- DMA controller channel assignments
- Real-time clock map

You can find detailed information concerning the topics discussed in this appendix in other sections of this manual.

F.2 SYSTEM MEMORY MAP

Table F-1. Memory Address Map

Address	Name	Function
000000:09FFFFH	512K system board	302 board memory (0 - 640K)
0A0000:0BFFFFH	128K video RAM	Reserved for video display controller.
0C0000:0C7FFFH	32K video ROM	Reserved for video display controller, BIOS ROM, and video BIOS ROM shadow.
0C8000:0DFFFFH	96K I/O expansion ROM/RAM	Reserved for ROM and RAM on I/O adapters.
0E0000:0FFFFH	128K system ROM BIOS/shadow RAM	Reserved for system ROM BIOS & shadow of system and video ROM BIOS
100000:FDFFFFH	Extended memory	Extended Memory space
FE0000:FFFFFFH	128K reserved on 302 board	Duplicates code assignment at address 0E0000H.
1000000:FFFFFFFH	Extended Memory	Extended Memory space.

If operating with the 16M option in the setup program enabled, expansion RAM starts at 1000000H and goes to FFFFFFH. There is no duplication at FE0000:FEFFFFH. The setup program provides this memory addressing option.

F.3 I/O ADDRESSES

Table F-2 lists the I/O locations that are addressable from the ISA bus. The 302 board provides three control ports directly on the ISA bus. These ports are described following Table F-2.

Table F-2. I/O Address Map

Range	Device
000H-01FH	DMA controller-1, 8237, for 8-bit device
020H-3FH	Interrupt controller-1, 8259, master PIC
040H-5FH	System timer, PIT 8254
060H, 064H	Keyboard controller, 8742
61H	Control port (bits described in Table F-3)
70H	NMI mask (bit 7)
70H-71H	Real-time clock
78H-79H	Auxiliary control ports (described in
	Tables F-4 and F-5)
080H-08FH,	DMA page registers, 74LS612
480H-48FH	
0A0H-0BFH	Interrupt controller-2, 8259, slave PIC
0C0H-0DFH	DMA controller-2, 8237
0F0H	Clear numeric coprocessor BUSY
0F8H-0FFH	Numeric coprocessor
1F0H-1F8H	Fixed disk controller
200H-207H	Game I/O port
278H-27FH	Parallel printer port 2 (LPT2)
2F8H-2FFH	Serial port 2 (COM2)
378H-37FH	Parallel printer port 1 (LPT1)
3B0H-3BFH	Monochrome display/printer port
3C0H-3CFH	Enhanced graphics controller
3D0H-3DFH	Color/graphics controller
3F0H-3F7H 3F8H-3FFH	Floppy disk drive controller
400H-44FH	Serial port 1 (COM1) Multi-terminal adapter board
400H-44FH 46E8H	Video display controller
40C0H 4BC4H-4BC5H	Video display controller Video display controller
56E8H	Video display controller
66E8H	Video display controller
76E8H	Video display controller Video display controller
702011	Flace display controller

I/O address 61H is an 8-bit control port. Table F-3 lists the 61H bit assignments.

Table F-3. I/O Address 61H Bit Assignment

Bit/Value	Function	Access
Bit 7 1 0	AT32 parity error Parity error No parity error	Read only
Bit 6 1 0	ISA parity error Parity error No parity error	Read only
Bit 5 1 0	Speaker signal Speaker on Speaker off	Read only
Bit 4	Refresh signal The refresh signal toggles with each refresh	Read only
Bit 3 1 0	Enable ISA parity error Parity error disabled Parity error enabled	R/W
Bit 2 1 0	Enable AT32 memory parity error Parity error disabled Parity error enabled	R/W
Bit 1 1 0	Speaker data Speaker data on Speaker data off	R/W
Bit 0 1 0	Enable speaker Speaker enabled Speaker disabled	R/W

I/O address 78H is an 8-bit auxiliary control port. Table F-4 lists the 78H bit assignments.

Table F-4. I/O Address 78H Bit Assignments

Bit/Value	Function	Access
Bit 7 1 0	Force memory parity error Force bad parity Normal operation	R/W
Bit 6 1 0	Cache misses Enable cache Force cache misses	R/W
Bit 5:4 00 10 01 11	Video and system BIOS Normal operation; neither shadowed Downloading video BIOS to RAM Only system BIOS shadowed Both system and video BIOS shadowed	R/W
Bit 3	Reserved	
Bit 2 1 0	A20 enabled Gate A20 enabled Gate A20 disabled	R/W
Bit 1 1 0	AT32 I/O AT32 I/O enabled ISA I/O enabled	R/W
Bit 0	Reserved	

I/O address 79H is an eight bit auxiliary control port. Table F-5 lists the 79H bit assignments.

Table F-5. I/O Address 79H Bit Assignments

Bit/Value	Function	Access
Bit 7 1 0	BIOS ROM at 16M Memory fill BIOS ROM at 16M enabled	R/W
Bit 6 1 0	Secondary enable speaker (In addition to port 61H bit 0) Speaker disabled Speaker enabled	R/W
Bit 4:0	Reserved	
Bit 5 1 0	Deturbo mode Deturbo (8 MHz CPU emulation Normal (turbo)	R/W

F.4 INTERRUPT PRIORITY LEVELS

Table F-6 lists the interrupt priority assignments for the system.

Table F-6. Interrupt Levels

Priority	PIC No.	Int No.	Interrupt Source	
1	†	NMI	Parity error detected	
2	1	IRQ0	Interval timer (PIT), counter 0 output	
3	1	IRQ1	Full keyboard output buffer	
	1	IRQ2	Interrupt from controller 2 (cascade)	
4	2	IRQ8	Real-time clock INT	
5	2	IRQ9	Software redirected to INT 0AH (IRQ2)	
6	2	IRQ10	Reserved	
7	2	IRQ11	Reserved	
8	2	IRQ12	Auxiliary device	
9	2	IRQ13	INT from coprocessor	
10	2	IRQ14	Fixed disk controller	
11	2	IRQ15	Reserved	
12	1	IRQ3	COM2	
13	1	IRQ4	COM1 (primary)	
14	1	IRQ5	LPT2	
15	1	IRQ6	Floppy disk controller	
16	1	IRQ7	LPT1 (primary)	

[†] I/O address 70H, bit 7, controls the NMI signal.

There is always the possibility that more than two interrupts will demand servicing at the same time. The PICs determine the priority of each interrupt and process the requests one at a time by transferring the control of the CPU to the higher priority service routine first.

F.5 DMA CHANNEL ASSIGNMENTS

Each DMA controller has four ports, DMA(4:1). DMA1 supports channels zero through three (for 8-bit transfers) and DMA2 supports channels four through seven (for 16-bit transfers). Channel four is used to cascade from DMA2 to DMA1. All HOLD requests for DMA1 are processed via DMA2, channel four. This forces all channels in DMA1 to operate at a higher priority than those in DMA2. Channel zero has the highest priority and channel seven the lowest. DMA channels 0-3 are 8-bit channels and DMA channels 5-7 are 16-bit channels. Table 8-1 lists each channel's function.

Table F-7.	DMA	Channel	Assignment
------------	-----	---------	------------

Channel	Controller	Function
0 1 2 3 4 5 6 7	1 1 1 2 2 2 2 2	Refresh Streaming tape (typical) Disk (floppy) Spare Cascade Spare Spare Spare Spare

F.6 RTC MAPPING

The 64 addressable locations in the RTC are divided into ten bytes containing the time, calendar, alarm data, four control and status bytes, and 50 general purpose RAM bytes as listed in Table F-8. Table F-8 also details the internal register/RAM organization of the RTC.

Table F-8. RTC Address Map

Function	Index
Time, Calendar, and Alarm Bytes Seconds register Seconds alarm register Minutes register Minutes alarm register Hours register Hours alarm register Day of week register Date of month register Month register Year register	00H 01H 02H 03H 04H 05H 06H 07H 08H 09H
Status Registers Status register A Status register B Status register C Status register D	OAH OBH OCH ODH
General Configuration Bytes Diagnostic status byte Shutdown status byte Floppy disk drive type byte Reserved Fixed disk type byte Reserved Equipment byte Low base memory byte High base memory byte Low expansion memory byte High expansion memory byte Drive C extended type byte Drive D extended type byte Reserved Features installed byte Drive type 48 parameters byte Cache, shadow, and setup byte	0EH 0FH 10H 11H 12H 13H 14H 15H 16H 17H 18H 19H 1BH-1EH 1FH 20H-27H 28H

(continued)

Table F-8. Real-time Clock Address Map (continued)

Function	Index
Reserved 2-byte CMOS RAM checksum byte Low extended memory byte High extended memory byte Date century byte Setup information byte System speed byte Drive type 49 parameters byte	29H-2DH 2EH-2FH 30H 31H 32H 33H 34H 35H-3CH
Reserved	3DH-3FH

Hot Keys



This appendix lists the "hot keys." Hot keys are keystroke sequences used to invoke special 302 system functions. Note that in these descriptions, all numbers refer to numeric pad keys.

Keystroke sequence	<u>Function</u>
Ctrl + Ait + Ins	Enter ROM-based setup program.
Ctrl + Ait + 0	Enter ROM-based setup program.
Ctrl + Ait + 1	Set deturbo mode. CPU emulates 8 MHz CPU.
Ctrl + Alt + 2	Set turbo mode. CPU runs at 25 MHz.

Pin Assignments for Major Signals



H.1 INTRODUCTION

This appendix provides pin assignments for all major signals present in the 302 system.

H.2 CPU PIN ASSIGNMENTS

Table H-1 lists the connector pinouts for the CPU. The pinouts are listed in functional groups.

Table H-1. CPU Pin Assignments

Pin/Signal	Pin/Signal	Pin/Signal	Pin/Signal
N2 A31 P1 A30 M2 A29 L3 A28 N1 A27 M1 A26 K3 A25 L2 A24 L1 A23 K2 A22	M5 D31 P3 D30 P4 D29 M6 D28 N5 D27 P5 D26 N6 D25 P7 D24 N8 D23 P9 D22	A1 V _{CC} A5 V _{CC} A7 V _{CC} A10 V _{CC} A14 V _{CC} C5 V _{CC} C12 VCC D12 V _{CC} G2 V _{CC} G3 V _{CC}	A2 V _{ss} A6 V _{ss} A9 V _{ss} B1 V _{ss} B5 V _{ss} B11 V _{ss} B14 V _{ss} C11 V _{ss} F2 V _{ss} F3 V _{ss}

(continued)

Table H-1. CPU Pin Assignments (continued)

Pin/Signal	Pin/Signal	Pin/Signal	Pin/Signal
K1 A21	N9 D21	G12 V _{cc}	F14 V _{ss}
J1 A20	M9 D20	G14 V _{CC}	J2 V _{ss}
H3 A19	P10 D19	L12 V _{CC}	J3 V _{SS}
H2 A18	P11 D18	M3 V _{CC}	J12 V _{ss}
H1 A17	N10 D17	M7 V _{CC}	J13 V _{ss}
G1 A16	N11 D16	M13 V _{CC}	M4 V _{ss}
F1 A15	M11 D15	N4 V _{cc}	M8 V _{ss}
E1 A14	P12 D14	N7 V _{CC}	M10 V _{SS}
E2 A13	P13 D13	P2 V _{cc}	N3 V _{ss}
E3 A12.	N12 D12	P8 V _{cc}	P6 V _{ss}
D1 A11	N13 D11		P14 V _{SS}
D2 A10	M12 D10		
D3 A9	N14 D9	F12 CLK2	A4 N.C.
C1 A8	L13 D6		B4 N.C.
C2 A7	K12 D7	E14 ADS*	B6 N.C.
C3 A6	L14 D6		B12 N.C.
B2 A5	K13 D5	B10 W/R*	C6 N.C.
B3 A4	K14 D4	A11 D/C*	C7 N.C.
A3 A3	J14 D3	A12 M/IO*	E13 N.C.
C4 A2	H14 D2	C10 LOCK*	F13 N.C.
A13 BE3*	H13 D1		
B13 BE2*	H12 D0	D13 NA*	C8 PEREQ
C13 BE1*		C14 BS16*	B9 BUSY*
E12 BE0*		G13 READY*	A8 ERROR*
	D14 HOLD		
C19 RESET	M14 HLDA	B7 INTR	B8 NMI

H.3 AT32 BUS PIN ASSIGNMENTS

Table H-2 lists the pin assignments for the AT32 bus.

Table H-2. AT32 Bus Connector Pin Assignments

Pin	Component Side	Non- Component Side	Pin	Component Side	Non- Component Side
1 2	GND	5VDC	23	YA3	YA2
	YD0	YD1	24	YA5	YA4
3	YD2	YD3	25	YA7	YA6
4	YD4	YD5	26	YA9	YA8
5	YD6	YD7	27	GND	YA10
6	YD8	YD9	28	YA11	YA12
7	YD10	YD11	29	YA13	YA14
8	YD12	YD13	30	YA15	YA16
9	YD14	YD15	31	YA25	YA24
	GND	5VDC	32	YA27	YA26
11	YD16	YD17	33	YSELn*†	GND
12	YD18	YD19	34	YLOCK*	YASTB*
13 14	YD20 YD22	YD21 YD23 YD25	35 36	YMIO* YRSRQ* YRSP0	YCASSTB* YWR* YRSP1
15 16 17	YD24 YD26	YD27	37 38 39	YCACHEN* YSPD1*	YPSIZE YSPD0*
18	YD28 YD30	YD29 YD31	40	GND	5VDC YREQn*†
19	GND	5VDC	41	YRDY	YREQn*†
20	Reserved	Reserved	42	YPARERR*	YPRYn*†
21	YBE1*	YBE0*	43	YPARFRC	YGNTn*†
22	YBE3*	YBE2*	Ş	II ARITIO	- IGIVIII

[†] The AT32 bus signals with a lower case n following the signal name are connected to only one expansion connector (numbered n). All signals without an n suffix are bussed to all AT32 bus expansion connectors.

H.4 ISA BUS PIN ASSIGNMENTS

These are the pin assignments looking down onto the top (open card-receiving side) of the connectors. Table H-3 lists the pin assignments for the 8-bit slot.

Table H-3. ISA Bus 8-bit Connector Pin Assignments

Pin No.	Signal	Pin No.	Signal
B1	Ground	A1	IOCHCK*
B2	RSTDEV	A2	D07
B3	5V	A3	D06
B4	IRQ09	A4	D05
B5	-5V	A5	D04
B6	DRQ2	A6	D03
B7	-12V	A7	D02
B8	SRDY*	A8	D01
B9	12V	A9	D00
B10	Ground	A10	IOCHRDY
B11	MEMW*	A11	AEN
B12	MEMR*	A12	A19
B13	IOWC*	A13	A18
B14	IORC*	A14	A17
B15	DACK3*	A15	A16
B16	DRQ3	A16	A15
B17	DACK1*	A17	A14
B18	DRQ1	A18	A13
B19	MEMREF*	A19	A12
B20	SYSCLK	A20	A11
B21	IRQ07	A21	A10
B22	IRQ06	A22	A09
B23	IRQ05	A23	80A
B24	IRQ04	A24	A07
B25	IRQ03	A25	A06
B26	DACK2*	A26	A05
B27	TC	A27	A04
B28	BUSALE	A28	A03
B29	5V	A29	A02
B30	84OSC	A30	A01
B31	Ground	A31	A00

Table H-4 lists the pin assignments for the 16-bit slot.

Table H-4. ISA Bus 16-bit Connector Pin Assignments

Pin No.	Signal	Pin No.	Signal
D1	MCS16*	C1	SBHE*
D2	IOCS16*	C2	LA23
D3	IRQ10	C3	LA22
D4	IRQ11	C4	LA21
D5	IRQ12	C5	LA20
D6	IRQ15	C6	LA19
D7	IRQ14	C7	LA18
D8	DACK0*	C8	LA17
D9	DRQ0	C9	MRDC*
D10	DACK5*	C10	MWTC*
D11	DRQ5	C11	D08
D12	DACK6*	C12	D09
D13	DRQ6	C13	D10
D14	DACK7*	C14	D11
D15	DRQ7	C15	D12
D16	5V	C16	D13
D17	SECMAST*	C17	D14
D18	Ground	C18	D15

H.5 SERIAL COMMUNICATIONS PORT PINOUTS

The pin assignments for header J5 (9-pin versions of COM1 and COM2) are listed in Tables H-5 and H-6.

Table H-5. Nine-pin Serial Port (COM1) Connector Pin Assignments

J5 Pin No.	DB9 Pin No.	Signal/Function
1 2 3 4 5 6 7 8 9 23 24	1 6 2 7 3 8 4 9 5	DCD/carrier detect DSR/data set ready RXD/receive data RTS/request to send TXD/transmit data CTS/clear to send DTR/data terminal ready RI/ring indicator Ground Key (pin missing) MGTEST*/mfg. test

Table H-6. Nine-pin Serial Port (COM2) Connector Pin Assignments

J5 Pin No.	DB9 Pin No.	Signal/Function
10 11 12 13 14 15 16 17 18 23 24	1 6 2 7 3 8 4 9 5	DCD/data carrier detect DSR/data set ready RXD/receive data RTS/request to send TXD/transmit data CTS/clear to send DTR/data terminal ready RI/ring indicator Ground Key (pin missing) MGTEST*/mfg. test

The pinout assignments for header J6 (25-pin version of COM2) is listed in Table H-7.

Table H-7. 25-pin Serial Port (COM2) Connector Pin Assignment

J6 Pin No.	DB25 Pin No.	Signal/Function
15 11 5 7 3 9 14	- 6 3 4 2 5 20 22	DCD/data carrier detect DSR/data set ready RXD/receive data RTS/request to send TXD/transmit data CTS/clear to send DTR/data terminal ready RI/ring indicator
13 24 23	25 -	Ground MGTEST*/mfg. test Key (pin missing)

H.6 PARALLEL PRINTER PORT PINOUT

The pin assignment for the parallel printer connector is shown in Table H-8.

Table H-8. Parallel Printer Port Connector Pin Assignments

J4 Pin No.	DB25 Pin No.	36-Pin Ribbon Cable Pin No.	Signal/Function
1	1	1	STROBE*
3	2	2	PRTD0/data bit 0
5	3	3	PRTD1/data bit 1
7	4	4	PRTD2/data bit 2
9	5	5	PRTD3/data bit 3
11	6	6	PRTD4/data bit 4
13	7	7	PRTD5/data bit 5
15	8	8	PRTD6/data bit 6
17	9	9	PRTD7/data bit 7
19	10	10	ACK*/acknowledge
21	11	11	BUSY
23	12	12	PE/paper end
25	13	13	SLCT/select
2	14	14	AUTOFDXT*/auto feed
4	15	32	ERROR*
6	16	31	INIT*/initializing printer
8	17	36	SLCTIN*/select input
10,12,14,	18-25	19-30,33	Ground
16,18,22,24			
-	-	17	Chassis ground
20	-	-	Key (pin missing)
26	-	-	No connection

H.7 KEYBOARD PINOUT

Table H-9 lists the pinout for the keyboard connector.

Table H-9. Keyboard Connector Pin Assignment

Pin #	Signal	Signal Type
1 2 3 4 5 6	KBD CLK KBD DATA Reset Ground 5.0VDC Frame ground	Input/output Input/output Output Signal ground Power

Component Installation



I.1 INTRODUCTION

This appendix describes how to install and remove SIMMs and the 387 numeric coprocessor on the 302 board.

I.2 INSTALLING SIMMS

Installing SIMMs requires inserting two modules each in sockets J16 and J17. For each socket, insert the first SIMM into the left-hand slot and the second SIMM into the right-hand slot. To install SIMMs, perform the following:



CAUTION

Do not touch any electronic component unless you are properly grounded. Proper grounding can be established by wearing a grounded wrist strap or touching an exposed metal part of the system module chassis. A static discharge from your fingers can result in permanent damage to electronic components.



CAUTION

Use extreme care when installing SIMMs. The plastic retaining clips on the sockets are easily broken by using too much force.

- 1. Holding the SIMM by the edges only, remove it from the antistatic package.
- Position the SIMM correctly (see Figure I-1) and insert the bottom edge into the socket slot, beginning with the empty slot farthest to the left.
 Press down firmly while maintaining the angle of insertion.
- 3. Make sure the SIMM seats correctly. If not, gently spread the retaining clips just enough to permit the top edge of the SIMM to be pulled away from the clips. Reseat the SIMM.
- 4. When the SIMM seats correctly, hold it at each end, and gently push the top edge toward the slot retaining clips until it snaps into place.
- 5. Repeat steps one through four and install the remaining SIMMs into the socket slots, working from left to right.

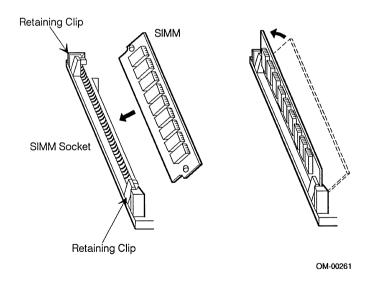


Figure I-1. Installing SIMMs

I.3 REMOVING SIMMS

Use the following procedure to remove SIMMs from the 302 board.

When removing SIMMs, remove them one at a time from right to left. That is, remove the right-hand SIMM first, and the left-hand SIMM last.



CAUTION

Apply only enough pressure on the retaining clips to release the SIMM. Too much pressure can break the retaining clips or damage the socket slot.



CAUTION

Do not touch any electronic component unless you are properly grounded. Proper grounding can be established by wearing a grounded wrist strap or touching an exposed metal part of the system module chassis. A static discharge from your fingers can result in permanent damage to electronic components.

To remove a SIMM:

- Locate the SIMM in the right-hand slot of the right-most socket of the group to be removed (see Figure I-2).
- 2. Gently spread the retaining clips just enough to pull the top edge of the SIMM away from the retaining clips.
- 3. Carefully lift the SIMM away from the socket and store it in a suitable static-free protective wrapper.
- Repeat steps two and three, as necessary, to remove the remaining SIMMs.

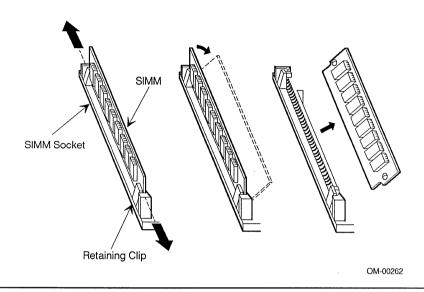


Figure I-2. Removing SIMMs

I.4 INSTALLING A NUMERIC COPROCESSOR



CAUTION

Do not touch any electronic component unless you are properly grounded. Proper grounding can be established by wearing a grounded wrist strap or touching an exposed metal part of the system module chassis. A static discharge from your fingers can result in permanent damage to electronic components.



CAUTION

Be sure to orient the beveled corner of the numeric coprocessor with the beveled cutout in the socket center. Failure to install the numeric coprocessor correctly may destroy the component and could damage the 302 board.

The numeric coprocessor plugs directly into the socket on the board (see Figure I-1). Follow these procedures to install the numeric coprocessor:

- 1. Remove the numeric coprocessor from its antistatic package, being careful not to touch the pins on the chip.
- 2. Align the numeric coprocessor's pins with the socket contacts.
- 3. Position the numeric coprocessor's pins in the socket receptacle. Press the chip down firmly until it seats (see Figure I-3). Be careful not to bend the pins.

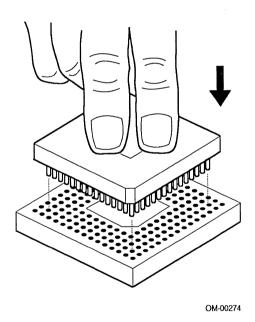


Figure I-3. Installing a Numeric Coprocessor

I.5 REMOVING A NUMERIC COPROCESSOR

Use the following procedures to remove the numeric coprocessor from the 302 board:



CAUTION

Do not touch any electronic component unless you are properly grounded. Proper grounding can be established by wearing a grounded wrist strap or touching an exposed metal part of the system module chassis. A static discharge from your fingers can result in permanent damage to electronic components.

- 1. Remove the numeric coprocessor from its socket on the 302 board (see Figure 1-1) using a grid-array device removal tool. Be careful not to touch the device pins.
- 2. Place the numeric coprocessor in an antistatic container to protect the device from static electricity.



Glossary

Α

A

Ampere.

AC

Alternating current. A current that periodically reverses its direction of flow.

accuracy

In scientific measurements, accuracy is the degree of conformity to an absolute standard. For example, a specification of $\pm 5V \pm 10\%$, signifies how accurate $\pm 5V$ is with respect to the absolute volt as defined by the U.S. National Bureau of Standards or other governing standards organizations. Do not confuse accuracy with precision. Contrast with precision.

active high

Designates a signal that has to go high to produce an effect.

active low

Designates a signal that has to go low to produce an effect.

adapter

- 1. An auxiliary device or unit used to extend the operation of another system.
- 2. An electronic part used to connect two dissimilar parts or machines.

address

- A name, label, or number identifying a location in storage, a device in a network, or any other data source.
- 2. A number that identifies the location of data in memory.

address bus

One or more conductors used to carry the binary-coded address from the processor throughout the rest of the system.

algorithm

A finite set of well-defined rules for the solution of a problem in a finite number of steps.

ampere(A)

The basic unit of electric current.

analog

Pertaining to data in the form of continuously variable physical quantities. Contrast with digital.

application

A program or set of programs used to do work on the computer. Some categories of application programs are word processors, database managers, spreadsheet managers, and project managers. Specific examples of application programs are MultiMate, dBase III PLUS, Lotus 1-2-3, Framework II, and Symphony.

array

An arrangement of elements in one or more dimensions.

ASCII

American Standard Code for Information Interchange. The code developed by ANSI for information interchange among data processing systems, data communications systems, and associated equipment. The ASCII character set consists of 7-bit control characters and symbolic characters.

asynchronous

In data communications, a method of transmission in which the bits included in a character or block of characters occur during a specific time interval. However, the start of each character or block of characters can occur at any time during this interval. Contrast with synchronous.

AUTOEXEC.BAT

A special-purpose batch file. When you turn on your computer, or restart it by pressing the the latter + the latter + the latter + latter

B

base address

The beginning address for resolving symbolic references to locations in storage.

base memory

Up to 640K of memory accessible to DOS. This is also referred to as conventional memory. Contrast with expanded memory and extended memory.

base register

A general purpose register that the programmer chooses to contain a base address.

BASIC

A programming language that uses common English words.

basic input/output system

The feature of a computer that provides a basic level of control of the major I/O devices, and relieves programmers of having to learn about system hardware device characteristics.

See BIOS.

batch file

A file that saves time and effort and which is identified by the .BAT extension following the file name. If you use a sequence of DOS commands frequently, you can create a batch file containing the commands, and then execute the entire sequence by typing the name of the file. This reduces the number of keystrokes needed to execute a sequence of commands.

binary

- 1. Involving a a choice of two conditions, such as on-off or ves-no.
- 2. Pertaining to a fixed radix numeration system having a radix of 2, wherein the binary digits are 0 and 1.

BIOS

The feature of a computer that provides a basic level of control of the major I/O devices, and relieves programmers of having to learn about system hardware device characteristics.

Acronym for basic input/output system.

bit

Synonym for binary digit. Either of the binary digits 0 or 1 used in computers to store information. (see also byte)

bits per second (bps)

A unit of measurement representing the number of discrete binary digits transmitted by a device in one second.

board

A rectangular piece of fiberglass that has pins on one side and electronic parts on the other; also called a card, PC board or PCB (printed circuit board). The system is always supplied with a board. Other boards can include a video adapter board, a disk controller board, a network communication board, memory boards, and multifunction boards.

boot

(see bootstrap)

bootstrap

A technique or device designed to bring itself into a desired state by means of its own action; for example, a machine routine whose first few instructions are sufficient to bring the rest of itself into the computer from an input device. For example, a computer that runs DOS boots itself by causing the computer to read the first few of its instructions from disk. Those instructions are sufficient to read in the rest of DOS from disk.

bps

Bits per second.

buffer

- 1. An area of storage that is temporarily reserved for use in performing an input/output operation into which data is written or from which data is read. Synonymous with I/O area.
- 2. A portion of memory storage for temporarily holding input or output data.

bus

One or more conductors used for transmitting signals or power.

bvte

- 1. A sequence of eight adjacent binary digits that are operated upon as a unit.
- 2. A binary character operated upon as a unit.
- 3. The amount of storage used to represent one character.

\mathbf{C}

C

- Celsius (centigrade).
- 2. A programming language.

cache memory

A small, high-speed memory block that improves CPU performance by minimizing the number of memory accesses on the bus. This action is accomplished by storing the CPU's most recently used data/instructions in the cache memory rather than main memory.

card

(see Board)

CAS

Column address strobe; a signal that latches the column addresses in a memory chip.

Cathode ray tube (CRT)

A vacuum tube in which a stream of electrons is projected onto a fluorescent screen producing a luminous spot. The location of the spot can be controlled. A CRT is the main element in a video display or monitor.

Celsius (C)

A temperature scale: also called Centigrade. Contrast with Fahrenheit.

Central processing unit (CPU)

Term for processing unit; i.e. 80386.

CFR Part 15 Subpart J

Federal Communications Commission Specification for EMI suppression.

channel

A path along which signals can be sent; for example, data channel, output channel.

character

A letter, digit, or other symbol.

character generator

- 1. In computer graphics, a functional unit that converts the coded representation of a graphics character into the shape of the character for display.
- In word processing, the means within equipment for generating visual characters or symbols from coded sets.

character key

A keyboard key that allows the user to enter the character shown on the key. Compare with function key.

character set

A group of characters used for a specific reason; for example, the set of characters a printer can print or a keyboard can support.

Class A device

Broadly defined, a Class A device complies with the various regulatory agencies that certify equipment for operation in a commercial (office or factory) environment.

Class B device

Broadly defined, a Class B device complies with the various regulatory agencies that certify equipment for operation in a home or residential environment.

CMOS

Complementary metal oxide semiconductor. A logic circuit family that uses very little power. It works with a wide range of power supply voltages.

COM1, COM2, COM3, COM4

These are the names DOS assigns to the serial communications port(s). Some systems only provide one serial port; others provide two.

CONFIG.SYS file

A special-purpose file that provides DOS with information about the special kinds of hardware or software used with the computer. Whenever the computer is turned on or rebooted by pressing $\boxed{\text{Ctrl}}$ + $\boxed{\text{Alt}}$ + $\boxed{\text{Del}}$, DOS searches the system disk for the CONFIG.SYS file. If DOS finds one, it reads the commands from the file and uses them to prepare the computer for operation.

configuration

- The arrangement of a computer system or network as defined by the nature, number, and chief
 characteristics of its functional units. More specifically, the term configuration may refer to a
 hardware configuration or a software configuration.
- 2. The devices and programs that make up a system, subsystem or network.

connector

A device allowing the connection of various electrical elements. See edge connector.

conventional memory

(see base memory)

cps

Characters per second.

CPU

Central processing unit.

CRT

Cathode ray tube.

CSA 22.2 # 220

Canadian Standards Association Standard for Safety of Information Processing and Business Equipment.

cursor

- 1. In computer graphics, a movable marker that is used to indicate position on a display.
- A displayed symbol that acts as a marker to help the user locate a point in text, in a system command, or in storage.
- 3. A movable spot of light on the screen of a display device, usually indicating where the next character is to be entered, replaced, or deleted.

cycle time

Defines the minimum amount of time in which subsequent accesses to a DRAM device can occur.

cylinder

All fixed disk or diskette tracks that can be read or written without moving the disk drive or diskette drive read/write mechanism

D

data base

A collection of data that can be immediately accessed and operated upon by a data processing system for a specific purpose.

dB

(see Decibel)

DC

Direct current.

decibel

- 1. A unit that expresses the ratio of two power levels on a logarithmic scale.
- 2. A unit for measuring relative power.

Deutsche Industrie Norm (DIN)

- 1. German Industrial Norm.
- 2. The committee that sets German dimension standards.

diagnostic

Pertaining to the detection and isolation of a malfunction or mistake.

digital

Pertaining to data in the form of digits. Contrast with analog.

DIN

Deutsche Industrie Norm.

DIN connector

One of the connectors specified by the DIN committee.

DIP

Dual in-line package. DIPs have pins in two parallel rows. The pins are spaced 1/10 inch apart. (See also DIP switch.)

DIP switch

One of a set of small switches mounted in a dual in-line package.

direct current (dc)

A current that always flows in one direction.

direct memory access (DMA)

A method of transferring data between main storage and I/O devices that does not require processor intervention.

disable

To stop the operation of a circuit or device; specifically applied to communications ports.

disabled

Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. Synonymous with masked.

disk

Loosely, a magnetic disk.

disk drive

A device for storing data on and retrieving data from a fixed disk or diskette.

diskette

A thin, flexible magnetic disk, permanently sealed in a protective jacket, that is used to store information. Synonymous with floppy and flexible disk. Most commonly available in 5.25-inch and 3.5-inch sizes

display

- 1. A visual presentation of data.
- 2. A device for visual presentation of information on any temporary character imaging device.
- 3. To present data visually.
- 4. See cathode ray tube.

DMA

Direct memory access.

DOS

Disk Operating System. (see operating system)

double precision

Pertaining to the use of two computer words to represent a number in accordance with the required precision. Contrast with single precision.

DRAM

Dynamic RAM. A type of RAM comprised of capacitive cells that require periodic refresh to maintain data. While the DRAM is a slower than the SRAM, its cell is much smaller. This enables the DRAM to be a higher density device and generally less expensive.

drop card

Drop cards are expansion boards that gain extra surface area by dropping down immediately after the 8-bit edge connector. Drop cards are not compatible in 16-bit and 32-bit expansion slots because the dropped portion of the card interferes with the unused connectors in these longer slots.

dual in-line package (DIP)

A widely used container for an integrated circuit.

dynamic memory

Random access memory (RAM). Read/write memory. See DRAM.

E

edge connector

A terminal block with a number of contacts attached to the edge of a printed-circuit board to facilitate plugging into a foundation circuit.

EIA

Electronic Industries Association.

EMI

Electromagnetic Interference.

enable

To initiate the operation of a circuit or device; specifically applied to communications ports.

EPROM

Erasable programmable read-only memory. A PROM that allows the user to change its code.

E2PROM

Electrically erasable programmable read-only memory. An EPROM which allows the user to change its code by means of appropriate electrical signals.

ESDI

Enhanced Small Device Interface, which achieves faster throughput than standard fixed disk controllers by shifting functionality to the fixed disk drive.

expanded memory

Certain expansion boards can provide additional memory to a personal computer. Expanded memory is distinguished from conventional memory in that it cannot be addressed directly by DOS but must be accessed through the expanded memory manager. It is used directly by application programs such as Symphony and Framework. Contrast with extended memory.

expansion slot

A series of connectors mounted on the board into which expansion boards can be inserted. Depending upon the system model, the type and number of expansion slots will vary.

extended memory

Memory whose addresses start at 1M and which can be accessed only when the processor is running in protected mode. Extended memory can be used by operating systems, such as UNIX, that run in protected mode, and by certain DOS programs, such as the RAMDRIVE.SYS virtual disk, that switch in and out of protected mode to perform special operations. Ordinary DOS applications cannot directly access extended memory.

F

Fahrenheit (F)

A temperature scale. Contrast with Celsius (C).

falling edge

Synonym for negative-going edge.

fast paged mode

A dynamic memory mode of operation that allows successive addresses to the same DRAM page (defined by the row addresses strobed (RAS) into the devices by simply changing the column addresses. In this mode, RAS is held active and the various accesses are initiated by strobing the new column addresses with the falling edge of CAS. The mode is available in CMOS parts and allows fewer cycle times.

FCC

Federal Communications Commission.

field

- 1. In a record, a specified area used for a particular category of data.
- 2. In a data base, the smallest unit of data that can be referred to by name.

firmware

- 1. Instructions or programs stored permanently in read-only memory (ROM) and unchangeable.
- 2. Internal connections that permanently determine the function of a device or system.

fixed disk

A nonflexible, flat, circular plate with a magnetizable surface layer on which data can be stored by magnetic recording. Synonymous with hard disk. Fixed disks are permanently mounted within a fixed disk drive.

fixed disk drive

A computer unit that consists of nonremovable magnetic disks, and a device for storing and retrieving data from the disks. Synonymous with hard disk drive.

flexible disk

See diskette.

floppy disk

Synonym for flexible disk. See diskette.

floppy disk drive

A computer unit that can store and retrieve data from floppy disks. See diskette.

function keys

Keys that request actions but do not display or print characters. Included are the keys that normally produce a printed character, but when used with the code key produce a function instead. Compare with character key.

G

G

A symbol used to represent the prefix giga. When describing computer storage capacity, common usage has made G synonymous with GB, G-byte or gigabyte.

GB

Abbreviation for gigabyte.

giga

A prefix normally used to indicate a quantity of 1,000,000,000. However, when referring to computer storage capacity, the prefix giga represents a quantity of 1,073,741,824 or 2 raised to the 30th power.

gigabyte

A term used when referring to computer storage capacity. A gigabyte is defined as 1,073,741,824 bytes.

gram (g)

A unit of weight equivalent to 0.035 ounces.

graphics

type of data created from fundamental drawing units such as lines, splines, curves, polygons, and so forth.

H

hard disk

(See fixed disk.)

hardware

- Physical equipment used in data processing, as opposed to programs, procedures, rules, and associated documentation.
- 2. Contrast with software.

head

A device that reads, writes, or erases data on a storage medium; for example, a small electromagnet used to read, write, or erase data on a magnetic disk.

header

A connector located on the board. Usually consisting of one or more rows of evenly-spaced pins.

Hertz (Hz)

A unit of frequency equal to one cycle per second.

hex

Common abbreviation for hexadecimal.

hexadecimal

- Pertaining to a selection, choice, or condition that has 16 possible different values or states.
 These values or states are usually symbolized by the ten digits 0 through 9, and the six letters A through F.
- 2. Pertaining to a fixed radix numeration system having a radix of 16.

See	Hertz.	
See I	Hertz.	

Ι

icon

Icon is a term used to describe graphic display symbols commonly used on video displays. A small symbol that can be easily identified with a device or function; e.g., a graphics symbol of a printer or keyboard. Selecting the icon will allow you to access the device or function it represents.

IEC 435

TUV certified in accordance with International Electrotechnical Commissions Standard for Safety of Information Technology Equipment.

IEC 950

TUV certified in accordance with International Electrotechnical Commissions Standard for Safety of Information Technology Equipment including Electrical Business Equipment.

input/output (I/O)

- Pertaining to a device or to a channel that may be involved in an input process and, at a different time, in an output process. Input/output may be used in place of input/output data, input/output signal, and input/output terminals, when such usage is clear in a given context.
- 2. Pertaining to a device whose parts can be performing an input process and an output process at the same time.
- Pertaining to either input or output, or both.

instruction

A statement that specifies an operation to be performed by the computer, along with the values or locations of operands, if any exist. This statement represents the programmer's request to the processor to perform a specific operation.

instruction set

The set of instructions of a computer, of a programming language, or of the programming languages in a programming system.

interface

A device that alters or converts actual electrical signals between distinct devices, programs, or systems.

interleave

To arrange parts of one sequence of things or events so that they alternate with parts of one or more other sequences of the same nature and so that each sequence retains its identity.

interrupt

- 1. A suspension of a process, such as the execution of a computer program, caused by an event external to that process and performed in such a way that the process can be resumed.
- 2. In a data transmission, to take an action at a receiving station that causes the transmitting station to terminate a transmission.
- 3. Synonymous with interruption.

I/O

Input/Output.

K

K

A symbol used to represent kilobyte, a computer storage quantity representing 1024 bytes, wherein 1024 is equal to 2 raised to the 10th power. Common usage has made it synonymous with KB, Kbyte or kilobyte. See kilobyte. Contrast with k.

k

A symbol used to represent the prefix kilo; 1,000.

KB

Abbreviation for kilobyte.

Kb

A symbol used to represent kilobit, a computer storage quantity representing 1024 bits, wherein 1024 is equal to 2 raised to the 10th power.

kevlock

A device that can deactivate a keyboard (if implemented) and locks the cover on for security.

kilo

A prefix used to indicate a quantity of 1000. Abbreviation symbol k. Contrast with K.

kilobyte

A term used when referring to computer storage capacity. A kilobyte is defined as 1024 bytes. Note that in all other usages, the prefix kilo (k) indicates a quantity of 1,000.

kilogram (kg)

1000 grams.

kilohertz (kHz)

1000 Hertz.

L

leading edge

The first occurring edge of a pulse.

LIM

Lotus/Intel/Microsoft Expanded Memory Manager specification.

LED

Light-emitting diode.

LPT1, LPT2, LPT3

These are the names DOS assigns to the parallel printer ports in a system. The three names reflect the fact that DOS permits as many as three parallel printer ports in a system.

M

M

A symbol used to represent the prefix mega. When describing computer storage capacity, common usage has made M synonymous with MB, Mbyte or megabyte. See mega.

m

- 1. Prefix milli; 0.001.
- Meter.

mΑ

Milliampere; 0.001 ampere.

machine language

A language that can be used directly by a computer without intermediate processing.

magnetic disk

A flat circular plate with a magnetizable surface layer on which data can be stored by magnetic recording.

main storage

- Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing.
- 2. Contrast with mass storage.

mapping

Pertaining to the geographic location for a resource within the address space.

masked

Synonym for disabled.

mass storage

Auxiliary storage in a computer system as differentiated from RAM. Mass storage most commonly refers to floppy and fixed disks and magnetic tape.

MB

Abbreviation for megabyte.

Mb

A symbol used to represent megabit, a computer storage quantity representing 1,048,576 bits, wherein 1,048,576 is equal to 2 raised to the 20th power.

mega

A prefix normally used to indicate a quantity of 1,000,000. However, when referring to computer storage capacity, the prefix mega represents a quantity of 1,048,576 or 2 raised to the 20th power.

megabyte

A term used when referring to computer storage capacity. A megabyte is defined as 1,048,576 bytes.

memory

Storage on electric memory such as random access memory (RAM), read-only memory (ROM), or CPU registers.

MFM

Modified frequency modulation.

MHz

Megahertz; 1,000,000 Hertz.

micro (µ)

Prefix 0.000 001.

microprocessor

An integrated circuit that accepts coded instructions for execution; the instructions may be entered, integrated, or stored internally.

microsecond (µs)

0.000 001 second.

modified frequency modulation (MFM)

The process of varying the amplitude and frequency of the write signal. MFM pertains to the number of bytes of storage that can be stored on the recording media. The number of bytes is twice the number contained in the same unit area of recording media at single density. MFM recording is commonly used on floppy- and fixed disk drives.

monitor

- 1. A device for visual presentation of information as temporary images. A video display.
- 2. Synonym for cathode ray tube display (CRT display).

N

nanosecond (ns)

0.000 000 001 second.

negative-going edge

The edge of a pulse or signal changing in a negative direction. Synonymous with falling edge.

negative true

Synonym for active low.

network

A group of computers connected and configured such that they can share resources.

nonrecoverable error

An error that makes recovery impossible without the use of recovery techniques external to the computer program run.

ns

nanosecond; 0.000 000 001 second.

O

OEM

Original Equipment Manufacturer.

offline

Pertaining to the operation of a functional unit without the continual control of a computer.

online

Pertaining to the operation of a functional unit under the continual control of a computer.

operating system

Software that controls the execution of programs; an operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

P

paged mode

The same basic functionality as fast paged-mode, except that the access time is the same as a normal RAS/CAS access. This mode is a feature on NMOS-type DRAM parts.

PAL

Program array logic

parallel

- 1. Pertaining to the concurrent or simultaneous operation of two or more devices, or to the concurrent performance of two or more activities.
- 2. Pertaining to the concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels.
- 3. Pertaining to the simultaneity of two or more processes.
- 4. Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts.
- 5. An alternative to serial.

parameter

- A variable that is given a constant value for a specified application and that may denote the application.
- 2. A name in a procedure that is used to refer to an argument passed to that procedure.

pel

Picture element.

picture element (pel)

In computer graphics, the smallest element of a display space that can be independently assigned color and intensity. Synonymous with pixel.

pixel

The smallest displayable unit on a monitor or picture tube element. Synonymous with pel.

platform system

A basic OEM product-line system which combines computers or computer subsystems with special, unique, and proprietary hardware and/or software for added value.

port

An access point for data entry or exit.

positive-going edge

The edge of a pulse or signal changing in a positive direction. Synonymous with rising edge.

positive true

Synonym for active high.

POST

Acronym for power-on self test.

power-on self test

A series of diagnostic tests that are run each time the computer's power is turned on. See POST.

power supply

A device that produces the power needed to operate electronic equipment.

precision

In science, a measure of the ability to differentiate quantities; the degree of agreement of repeated measurements of a quantity. Not to be confused with accuracy.

printed circuit

A pattern of conductors (corresponding to the wiring of an electronic circuit) formed on a board of insulating material.

printed-circuit board

Usually a copper-clad fiberglass board used to make a printed circuit. Also, refers to a board on which a printed circuit has been made.

processing unit

A functional unit that consists of one or more processors and all or part of internal memory.

processor

- 1. In a computer, a functional unit that interprets and executes instructions.
- A functional unit, a part of another unit such as a terminal or processing unit, that interprets and executes instructions. (see microprocessor)

program

A file containing a set of instructions conforming to a particular programming language syntax.

PROM

Programmable read-only memory. A type of ROM that contains a programmed set of code. A PROM code cannot be changed once programmed. See also EPROM, E²PROM and ROM.

protected mode

A mode of the 80386 microprocessor enabling it to provide advanced features, such as accessing large amounts of memory and enforcing hardware protection of memory segments. Current versions of DOS do not support protected mode operation, except for special utilities such as the RAMDRIVE.SYS virtual disk.

R

RAM

Random access memory. Read/write memory.

RAS

Row address strobe; a technique used in dynamic RAM addressing.

RAS/CAS

A mode of DRAM operation where every access is begun by strobing the row addresses with RAS and column addresses with CAS.

rastei

In computer graphics, a predetermined pattern of lines that provides uniform coverage of a display space.

read

To acquire or interpret data from a storage device, from a data medium, or from another source.

read-only memory (ROM)

A storage device whose contents cannot be modified. The memory is retained when power is removed.

recoverable error

An error condition that allows continued execution of a program.

register

- A storage device, having a specified storage capacity such as a bit, a byte, or a computer word, and usually intended for a special purpose.
- 2. A storage device in which specific data is stored.

reverse video

A form of highlighting a character, field, or cursor by reversing the color of the character, field, or cursor with its background; for example, changing a red character on a black background to a black character on a red background.

RFI

Radio frequency interference.

ROM

Read-only memory. See also PROM, EPROM, and E2PROM.

ROM BIOS

The ROM resident basic input/output (BIOS) system which controls the major I/O devices in a computer system.

RS-232C

A standard by the Electronics Industries Association (EIA) for serial communication between computers and external equipment.

S

scratch disk

A scratch disk is usually a formatted floppy disk that can be used for test purposes. A floppy disk that contains no information of value. If data on a scratch disk is lost or destroyed during tests, it is of no consequence.

sector

That part of a track or band on a magnetic drum, a magnetic disk, or a disk pack that can be accessed by the magnetic heads in the course of a predetermined rotational displacement of the particular device.

serial

- Pertaining to the sequential performance of two or more activities in a single device. The
 modifiers serial and parallel usually refer to devices, as opposed to sequential and consecutive,
 which refer to processes.
- Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel.
- 3. Pertaining to the sequential processing of the individual parts of a whole, using the same facilities for successive parts.
- 4. An alternative to parallel.

setup

- In a computer that consists of an assembly of individual computing units, the arrangement of interconnections between the units, and the adjustments needed for the computer to operate.
- 2. The preparation of the system for normal operation.

shadow memory

A portion of RAM to which selected BIOS information is copied from ROM. This ROM-to-RAM copying technique is referred to as shadowing. Shadow memory is write-protected and often has the same addresses as the original ROM locations. Shadowing greatly enhances system performance because ROM information is available from fast 32-bit RAM chips instead of the slower ROM chips.

signal

A variation of physical quantity, used to convey data.

SIMM

Single in-line memory module. A small plug-in board containing nine DRAM chips. A SIMM DRAM chip is organized in a specific configuration; i.e. $256\text{Kb} \times 1$ or $1\text{Mb} \times 1$ organization. For example, eight $256\text{Kb} \times 1$ DRAM devices combine to form 256K of memory. The ninth device provides parity checking.

single precision

Pertaining to the use of one computer word to represent a number in accordance with the required precision. Contrast with double precision and precision.

software

- Computer programs, procedures, and rules concerned with the operation of a data processing system.
- 2. Contrast with hardware.

SRAM

Static RAM. RAM comprised of static RAM chips. Unlike DRAMs, SRAMs require no refresh and are faster devices. The SRAM cell is larger than the DRAM cell and for this reason, SRAMs are lower density devices.

static column

A mode of operation of DRAM operation that allows successive accesses to the same DRAM page (defined by the row addresses initially strobed into the devices) by simply changing the column addresses. This mode differs from fast paged-mode because both RAS and CAS are held active, whereas CAS is strobed in fast paged-mode. The access time is limited to the address access time of the part as new column addresses are presented.

static memory

RAM using flip-flops as the memory elements. Data is retained as long as power is applied to the flip-flops. Contrast with dynamic memory.

storage

- 1. A storage device.
- 2. A device, or part of a device that can retain data.
- 3. The retention of data in a storage device.
- 4. The placement of data into a storage device.

synchronization

The process of adjusting the corresponding significant instants of two signals to obtain the desired phase relationship between these instants.

synchronous

- Data transmission in which the time of transmission occurrence of each signal representing a bit is related to a fixed time frame.
- Data transmission in which the sending and receiving devices are operating continuously at substantially the same frequency and are maintained in a desired phase relationship by means of correction.

Contrast with asynchronous.

Т

Т

A symbol used to represent the prefix tera. When describing computer storage capacity, common usage has made T synonymous with TB, Tbyte or terabyte.

TB

Abbreviation for terabyte.

tera

A prefix normally used to indicate a quantity of 1,000,000,000. However, when referring to computer storage capacity, the prefix tera represents a quantity of 1,099,571,627,300 or 2 raised to the 40th power.

terabyte

A term used when referring to computer storage capacity. A terabyte is defined as 1,099,571,627,300 bytes.

tpi

Tracks per inch. A specification used in formatting floppy disks and fixed disks.

TTL

Transistor-transistor logic. A popular logic circuit family that uses multiple-emitter transistors.

track

- The path or one of the set of paths, parallel to the reference edge on a data medium, associated with a single reading or writing component as the data medium moves past the component.
- The portion of a moving data medium such as a drum, or disk, that is accessible to a given reading head position.

trailing edge

The second edge of a pulse.

TUV

Technischer Ueberwachungs-Verein. TUV is an testing organization that evaluates and certifies electronic data processing equipment to specific International safety standards.

typematic key

A key that repeats its function multiple times when held down.

U

UL 478

Underwriter Laboratories Standard for Safety of Information Processing and Business Equipment.

V

ν

Volt.

VAC

Volts (alternating current).

VDE 0806/IEC 380

TUV certified in accordance with International Electrotechnical Commissions Standard for Safety of Electrical Energized Office Machines.

VDE 0871

Verband Deutscher Electrotechnikes Specification for EMI Suppression.

VDC

Volts (direct current).

video

Computer data or graphics displayed on a CRT, monitor, or display.

video adapter

A special board that provides a suitable interface between a computer and a video display device such as a CRT or monitor. A video controller.

video controller

A special board that provides a suitable interface between a computer and a video display device such as a CRT or monitor. A video adapter.

video display

A device for visual presentation of information as temporary images. A monitor. See also CRT.

virtual address

A 32-bit address on the internal bus intended to be translated by memory management.

volt

The basic unit of electric pressure. The potential that causes electrons to flow through a circuit.

•	•	•
1		
- 1		ì

W

Watt.

Watt (W)

The basic unit of electric power.

word

A character string or bit string considered as an entity in computer architecture.

write

To make a permanent or transient recording of data in a storage device or on data medium.

write precompensation

The varying of the timing of the head current from the outer tracks to the inner tracks of the diskette to keep a constant write signal.

Index

387 coprocessor 2-11, 2-13 binary floating-point arithmetic 2-10 install I-5 interrupts 2-12 1287 real-time clock (RTC) module 9-1 8237A DMA controllers 8-1 8254 programmable interval timer 1-7 8259 programmable interrupt controllers 1-7 82510 serial communication controller 10-1

A

Access ports B-29 Address map RTC 9-3,F-8 AT32 4-1 AT32 bus 2-15, 7-7, 13-12, H-3 buffers 2-16 Auto-initialization 8-5, 8-11, 8-15

В

Binary floating-point arithmetic 2-10 BIOS 1-10, 3-1, 7-7, 8-8, 9-21, 13-2, 13-11, B-1

clock services B-50, B-53 coprocessor interrupt B-54 disk H/W interrupt B-3 equipment determination B-13 fixed disk H/W interrupt B-54 fixed disk I/O B-21, B-28 floppy disk drive I/O B-14, B-20 keyboard H/W interrupt B-3 keyboard I/O B-42, B-47 memory size determination B-14 non-maskable interrupt B-1 print screen B-2 printer I/O B-48, B-49 real-time clock B-54 RS-232 I/O B-29, 32 system boot B-50 system service routines B-33, B-42 system timer H/W interrupt B-2 video I/O B-4, B-12 Bus timing 2-10 control unit 2-7

C

Cache memory 2-14, 2-15, 13-11 read requests 2-15 write requests 2-16 Cache tag 2-15 Central processing core 2-1

Changing the system speed 1-6, D-1	active cycle 8-4	
keyboard method D-2	address registers 8-10, 8-11	
software method D-3	auto-initialization 8-5, 8-11, 8-15	
using Setup D-2	channels 8-7, F-8	
Clock speed	command register 8-12	
changing 13-9	count registers 8-10, 8-11	
Configuration 13-14	idle cycle 8-4	
Connector pinouts	internal registers 8-9	
CPU H-1	mask register 8-15	
serial port H-7	mode register 8-14	
Control port	modes 8-4	
61H F-4	block-transfer 8-5	
78H F-5	cascade 8-6	
79H F-6	demand-transfer 8-5	
Counters	single-transfer 8-5	
PIT 6-2	page registers 8-7, 8-8, 8-20	
Counting element (CE) 6-2, 6-4	program-control registers 8-12	
CPU 1-6, 2-3, 2-10, 2-15, 8-1,	programming 8-8, 8-20	
9-3, 9-5, 9-7, 9-13, D-1, H-1	request register 8-15	
address paging unit 2-7	signals 8-4, 8-5, 8-6, 8-8	
addressable physical memory 2-3	software commands 8-19	
bus control unit 2-7	status register 8-17	
bus cycle initiation 2-16	temporary register 8-19	
bus timing 2-10	transfers 8-8, 8-9, 8-11, 8-12	
core control logic 2-16	DMA cycles 5-7, 5-13	
numeric coprocessors 2-10	DRAM 3-1	
pre-decode unit 2-7		
pre-fetch unit 2-7		
protected mode 2-3		
real mode 2-3		
segmentation unit 2-7	E	
signal definitions 2-6	_	
virtual memory 2-3	Environment A-3	
virtual-8086 mode 2-4	Error beep messages E-12	
CPU/cache control AT (CAT) 2-16	Error messages	
	setup 13-14	
	Execution unit 2-7	
	Expansion memory 3-1, 3-4, 9-17	
D	Expansion slots 1-8, A-4	
D	J1 1-8	
Daturba mada D.1 D.4	J2 1-8	
Deturbo mode D-1, D-4	J3 1-8	
Dimensions A-3	J4 1-8	
Direct Memory Access 1-7	J5 1-8	
DMA 8-1	J6 1-8, 4-2	
DMA controller 8-1	J7 1-8, 4-2	

J8 1-8

Extended memory 9-21, B-38

I K 1/0 Keyboard controller address 5-4 connector 11-7 address map F-3 controller commands 11-6 cycles 5-6, 5-10 data stream 11-8 I/O ports data transfers 11-9, 11-11 COM1 1-8 input buffer 11-4, 11-5 COM2 1-8 input port 11-5 LPT1 1-8 output buffer 11-4, 11-11 LPT2 1-8 output port 11-5 ICW 7-9 scan codes 11-4, 11-8, 11-12 Initialization sequence 7-12 status register 11-3, 11-5, 11-11 Interrupts 9-3 Keylock interface maskable interrupt 7-5 connector J20 12-1 NMI 2-12, 7-5 PIC 7-5, 15 request levels 10-1, 10-2 ISA bus 1-7, 3-1, 3-5, 4-1, H-4 M address signals 5-6, 5-7 bus agents 5-2, 5-3 central control signals 5-11, 5-12 Memory 9-17, 13-7 cycle control signals 5-9, 5-11 address 5-4 data signals 5-8 cache 2-14, 2-15, 13-11 DMA signals 5-13 cache memory 1-6 I/O address 5-4 cache tag 2-15 map F-3 cache tag memory 1-6 interrupt signals 5-13 DMA 8-1 memory address 5-4 **DRAM 3-1** power 5-14 expansion 3-4 extended 9-21, B-38 extended memory configuration 13-7 parity error B-1 random access memory (RAM) 1-6, 3-3, 13-11, B-14 Jumpers C-1 read-only memory monitor type C-5 (ROM) 1-6, 1-10, 13-2, 13-11 parallel port C-6 SIMM 3-3

printer acknowledge C-5

serial port C-6

SRAM 9-1	keyboard H-9
Memory map	parallel port H-8
RTC 9-3, F-8	serial port H-6
1110 0-0,1-0	PIT 1-7, 12-2
	counters 6-2
	control register 6-4
	functions 6-2
N	status register 6-4
	counting element (CE) 6-2, 6-4
NMI 2-12, 7-7, B-1	modes 6-1
	programming 6-5
mask register 7-5	control register 043H 6-5
Numeric coprocessor 13-10, D-1, I-5	latch counter command 6-7
instruction set 2-10	read-back command 6-8
registers 2-10	
removal I-6	read/write counter
	command 6-7
	Ports
	access B-29
	parallel 1-8, 10-2
0	input instructions 10-3
O	output instructions 10-3
	registers 10-4
OCW 7-15	serial 1-8
Onboard ROM 1-6	connector J5 10-2
	connector J6 10-2
	interfacing 10-2
	POST 1-9, 2-12, 13-1
P	configuration errors 13-2
-	memory test 13-1
D	reboot routine 13-1
Parallel port 1-8, 10-2, 10-3, 10-4,	startup message 13-1
C-6, H-8	POST and boot messages E-1
Parity error B-1	POST messages
PIC 1-7	error E-9
in-service register 7-4	error beep E-12, E-14
interrupt mask register 7-4, 5	informational E-10
interrupt request register 7-4	run-time E-11
priority resolver 7-4	Power requirements
programming	302 board A-3
ICW 7-8, 7-9	
initialization sequence 7-12	current drain A-5
	expansion slots A-4
OCW 7-8, 15	Programming
Pin assignments	DMA controller 8-8, 8-20
AT32 bus H-3	PIC 7-8, 12
CPU H-2	PIT 6-5, 6-7, 6-8
ISA bus H-4, 5	

5-11, 8-20 Reaisters R address registers 8-10, 8-11 command register 8-12 RAM 3-3, 13-11, B-14 count registers 8-10, 8-11 Read requests 2-15 DMA internal registers 8-9 Real-time clock 1-7, 9-1 in-service register 7-4 address map 9-4, F-9 interrupt mask register 7-4, 7-5 addressable locations 9-3, F-8 interrupt request 7-4 battery 9-1, 9-12 interrupt request register 7-4 bytes 9-11 mask register 8-15 CMOS RAM checksum mode register 8-14 byte 9-20 numeric coprocessors 2-10 CPU speed byte 9-22 page registers 8-7, 8-8, 8-20 date century byte 9-21 PIT 6-4, 6-5 diagnostic status byte 9-12 program-control registers 8-12 drive C extended byte 9-18 request register 8-15 drive D extended byte 9-18 status register 8-17, 11-3, equipment byte 9-16 11-5, 11-11 feature installed byte 9-18 status register A 9-7 fixed disk drive type byte 9-15 status register B 9-8 fixed disk type 48 status register C 9-10 parameter 9-19 status register D 9-11 fixed disk type 49 status registers 8-19 parameters 9-23 Reset interface 12-3 floppy disk drive type byte 9-14 connector J18 12-3 low and high hardware reset 12-3 base memory bytes 9-17 keyboard reset 12-3 low and high software reset 12-3 extended memory bytes 9-21 ROM 13-2, 13-11 low and high RS-232-C serial communication memory expansion bytes 9-17 ports 10-1 setup information byte 9-22 RTC 1-7 shadow and setup byte 9-20 shutdown status byte 9-13 format 9-6 initialization 9-5 memory map 9-2 operations 9-1 registers 9-7 Scan codes B-3 status register A 9-7 Serial port 1-8, 10-2, C-6, H-6 status register B 9-8 Setup program 1-9, 9-22, 13-2, D-2 status register C 9-10 512-640K 13-12 status register D 9-11

above 16M 13-12

AT32 bus 13-12

Refresh cycles 3-5, 5-4, 5-6, 5-8, 5-9,

update cycle 9-3

base memory 13-7 cache memory 13-11 changing options 13-4 computing extended memory 13-7 CPU speed 13-9 date 13-5 error messages 13-14 exiting 13-4, 13-13 extended memory 13-7 fixed disk drives 13-6 floppy disk drives 13-6 keyboard 13-9 monitor 13-8 numeric coprocessor 13-10 pre-boot setup 13-13 running 13-3 setup screen 1 13-3, 13-5 setup screen 2 13-10 speaker 13-13 system BIOS 13-11 time 13-5 video BIOS 13-11 **SIMM 3-3** installing I-1 Speaker interface 12-2 connector J19 12-2 Specifications A-1 SRAM 9-1 System configuration C-1 System reboot procedure 13-1

W

Weitek 3167 coprocessor 2-13

T

Transfer operations

DMA controller 8-6, 8-8, 8-9,
8-11, 8-12
read 8-6
verify 8-6
write 8-6

Transfer operations
keyboard controller 11-9, 11-11
Turbo mode D-1, D-5



INTEL CORPORATION 3065 Bowers Avenue Santa Clara, CA 95052-8130