TECHNICAL USER'S MANUAL FOR:

MICROSPACE®

EBX-Standard
Pentium Littleboard

MSLB_P5

Low Power Product

#190899-1

DIGITAL-LOGIC®
Nordstrasse 11/F, CH-4542 Luterbach
Tel.: ++41 (0)32 681 53 33 - Fax: ++41 (0)32 681 53 31

REVISION HISTORY:

Prod.-Serialnumber: From: To:	Product Version	Document Version	Date/Vis:	Modification: Remarks, News, Attention:
		V0.1	***10.99 FK***	***Initial version***
		V0.2	11.99 FK	RS422/485, EBX Standard
		V0.3	12.99 FK	Current, VideoInput

Registration Form:

Please register your product under:

http://www.digitallogic.ch -> SUPPORT -> Product Registration

After registration, you will receive driver & software updates, errata information, customer information and news from DIGITAL-LOGIC AG products automatically.

Table of Contents

1 PREFACE 7
1.1 Trademarks 7
1.2 Disclaimer 7
1.3 Who should use this Product 7
1.4 Recycling Information 8
1.5 Technical Support 8
1.6 Limited Warranty 8

2 OVERVIEW 9
2.1 Standard Features 9
2.2 Unique Features 9
Block Diagram 10
2.4 MSLB-P5 Specifications 11
2.5 Thermoscan 15
2.6 Ordering Codes 16
2.7 BIOS History 16
2.8 This product is "YEAR 2000 CAPABLE" 17
2.9 High frequency Radiation (to meet EN55022) 18
2.10 Mechanical Dimensions 19
2.10.1 Mechanical Dimensions Mounting Holes 19
2.10.2 Mechanical Dimensions Connectors 20
2.10.3 Mechanical Dimensions Frontside 21

3 THE PCI, ISA , EPCI AND PC/104 BUS SIGNALS 22
3.1 ISA Signals on the PC/104 and ISA-Bus 22
3.2 Connect PCI peripheral cards to the PC/104plus BUS: 26
3.3 EBX Specification — Ver 1.1 — July 9, 1997 27
3.3.1 EBX DETAILED MECHANICAL DRAWINGS 33

4 DETAILED SYSTEM DESCRIPTION 35
4.1 Power Requirements 35
4.2 CPUs, Boards and RAMs 35
4.2.1 CPUs of this MICROSPACE Product 35
4.3 Interface 36
4.3.1 PS/2-Keyboard 36
4.3.2 PS/2-Mouse Interface 36
4.3.3 Line Printer Port LPT1 36
4.3.4 Serial Ports COM1-COM4 RS232C 37
4.3.5 Serial Ports COM3-COM4 RS422 38
4.3.6 Serial Ports COM3-COM4 RS485 39
4.3.7 Driver for COM3-COM4 RS422/485 40
4.3.8 Floppy disk interface 41
4.3.9 Speaker interface 42
4.4 Controllers 43
4.4.1 Interrupt Controllers 43
4.5 Timers and Counters 43
4.5.1 Programmable Timers 43
4.5.2 Battery backed clock (RTC) 44
4.5.3 Watchdog 44
4.6 BIOS 45
4.6.1 ROM-BIOS Sockets 45
4.6.2 EEPROM Memory for Setup 46
4.6.3 BIOS CMOS Setup 47
4.7 CMOS RAM Map 48
4.7.1 CMOS Setup Harddisk list 54
4.7.2 Harddisk PIO Modes 54

4.7.3 EEPROM saved CMOS setup 55
4.7.4 BIOS Download Function 55
4.7.5 VGA BIOS Download Function 56
4.8 Memory 57
4.8.1 Onboard DRAM Memory 57
4.8.2 System Memory Map 57
4.8.3 System I/O map 58
4.9 BIOS Data Area Definitions 73
4.9.1 Compatibility Service Table 79
4.10 VGA, LCD 80
4.10.1 VGA / LCD Controller C&T69000 (optional C&T69030) 80
4.10.2 VGA / LCD BIOS for 69000 80
4.10.3 Display Modes Supported 81
4.10.4 VGA/LCD BIOS Support 82
4.10.5 Memory 69000 CRT/TFT Panels 83
4.10.6 Memory 69000 Color STN-DD Panels 84
4.10.7 Memory 69000 Mono STN-DD Panels 85
4.11 HiQ Video Multimedia Support 86
4.11.1 HiQVideo Series Programming Examples 87
4.11.2 Introduction 87
4.11.3 Video Playback through PCI/VL Bus 87
4.11.4 Video Capture and Playback Through Video Port 87
4.11.5 ZoomUp 87
4.11.6 Video Capture Using the Video Port 88
4.11.7 How to enable video capture and playback module (Init) 88
4.11.8 How to disable video playback and capture module (Exit) 89
4.11.9 How to start video capture 89
4.11.10 How to stop video capture 90
4.11.11 How to set input video color format 91
4.11.12 How to set interlaced or non-interlaced video input 91
4.11.13 How to enable/disable double buffer 92
4.11.14 How to scale input video (before acquiring into frame buffer) 92
4.11.15 How to crop input video (programming of acquisition rectangle) 93
4.11.16 Video Input with the SAA7111 VIP 98

5 DESCRIPTION OF THE CONNECTORS 100
5.1 Jumpers on this MICROSPACE product 112
5.2 Jumper and Connector Locations 113
5.2.1 Top-view 113
5.2.2 Bottom-view 114

6 CABLE INTERFACE 115
6.1 The Floppy Disk Cable 115
6.2 The Harddisk Cable 40 pins 116
6.3 The Harddisk Cable 44 pins 117
6.4 The COM 1/2 Serial Interface Cable 118

7 SPECIAL PERIPHERALS, OPTIONAL FUNCTIONS 119
7.1 Special Peripherals 119

8 100/10 ETHERNET LAN 120

9 SCSI INTERFACE (OPTION) 122
9.1 SCSI Drivers for operating system support 123

10 SOUNDPORT DRIVER INSTALLATION 124
10.1 ESS1869 124
10.2 Driver for WIN 3.11 124
10.3 Driver for WIN 95 125
10.4 Driver for NT4.0 125
10.5 Bundled Applications, MediaRack 125

11 INSTALLING THE FLASHDISK DOC2000 126
11.1 Enabling and Formatting of the DiskOnChip-Modules 126
12 BUILDING A SYSTEM 127
12.1 Starting up the System 127
12.2 Error on boot time 127
13 DIAGNOSTICS 128
14 BIOS 131
14.1.1 Main Menu Selections 131
14.1.2 Masters and Slaves 131
14.1.3 Memory Cache 132
14.1.4 Memory Shadow 134
14.1.5 Boot Sequence 135
14.1.6 Keyboard Features 136
14.2 Boot Menu 137
14.3 The Advanced Menu 138
14.3.1 Advanced Chipset Control Menu (PCI BIOS) 139
14.3.2 PCI Devices Menu 140
14.3.3 I/O Device Configuration Menu 141
The Security Menu 143
The Power Menu 145
The Exit Menu 147
Saving Values 147
Exit Discarding Changes 147
Load Setup Defaults 148
Discard Changes 148
Save Changes 148
14.4 PhoenixBIOS Messages 149
14.5 Phoenix Phlash Tool 150
14.6 Phoenix QuietBoot 150
14.6.1 Press <ESC> 151
14.6.2 Press <F2> 151
14.6.3 POST Error 151
14.6.4 Keyboard Input Request 151
14.7 Phoenix MultiBoot 151
14.7.1 The Setup Boot Menu 151
14.7.2 Removable Format Menu 152
14.7.3 Fixed and Removable Media Menus 152
14.7.4 The Boot First Menu 153
15 PHOENIX PHLASH 154
15.1 Installation 154
15.2 Create the Crisis Recovery Diskette 154
15.3 Updating the Crisis Recovery Diskette 155
15.4 Executing Phoenix Phlash 155
15.5 Crisis Recovery Mode 156
16 PROGRAMMER'S GUIDE 157
What is a ROM BIOS? 157
Initialize and Configure the computer 158
16.1.1 BIOS Services 158
16.2 System Hardware Requirements 158
16.3 Fixed Disk Tables 159
16.4 PhoenixBIOS Function Keys 160
16.5 POST Errors and Beep Codes 160
16.5.1 Recoverable POST Errors 160
16.5.2 Terminal POST Errors 160
16.5.3 Test Points and Beep Codes 161
16.6 PhoenixBIOS 4.0 Services 164

16.6.1 BIOS32 Service Directory 164
16.6.2 Interrupt 10h–Video Services 166
16.6.3 Interrupt 11h–Return System Information 168
16.6.4 Interrupt 12h–Return Memory Size 168
16.6.5 Interrupt 13h–Diskette Services 168
16.6.6 Interrupt 13h–Fixed Disk Services 171
16.6.7 Interrupt 13h–Extended Fixed Disk Services 174
16.6.8 Interrupt 13h–Bootable CD-ROM Services 175
16.6.9 Interrupt 14h–Serial Services 177
16.6.10 Interrupt 15h–System Services 179
16.6.11 Interrupt 15h–APM Services 181
16.6.12 Interrupt 15h–Big Memory Services 185
16.6.13 Interrupt 15h–PS/2 Mouse Services 186
16.6.14 Interrupt 15h–EISA Services 187
16.6.15 Interrupt 16h–Keyboard Services 189
16.6.16 Interrupt 17h–Parallel Printer Services 190
16.6.17 Interrupt 17h–EPP Services 191
16.6.18 Interrupt 1Ah–Time of Day Services 194
16.6.19 Interrupt 1Ah–General PCI Services 195
16.6.20 PnP Run-Time Services 197
16.6.21 SMBIOS Services 200
16.6.22 SMBIOS Services 201
16.6.23 MultiBoot II Run-Time Services 202
16.7 BIOS Data Area 204
16.7.1 Extended BIOS Data Area 207
16.8 Interrupt Vectors 207

17 INDEX 208

1 PREFACE

This manual is for integrators and programmers of systems based on the MICROSPACE card family. It contains information on hardware requirements, interconnections, and details of how to program the system. The specifications given in this manual were correct at the time of printing; advances mean that some may have changed in the meantime. If errors are found, please notify DIGITAL-LOGIC AG at the address shown on the title page of this document, and we will correct them as soon as possible.

1.1 Trademarks

MICROSPACE, MicroModule	DIGITAL-LOGIC AG
DOS Vx.y, Windows	Microsoft Inc.
PC-AT, PC-XT	IBM
NetWare	Novell Corporation
Ethernet	Xerox Corporation
DR-DOS, PALMDOS	Digital Research Inc. / Novell Inc.
ROM-DOS	Datalight Inc.

1.2 Disclaimer

DIGITAL-LOGIC AG makes no representations or warranties with respect to the contents of this manual and specifically disclaims any implied warranty of merchantability or fitness for any particular purpose. DIGITAL-LOGIC AG shall under no circumstances be liable for incidental or consequential damages or related expenses resulting from the use of this product, even if it has been notified of the possibility of such damage. DIGITAL-LOGIC AG reserves the right to revise this publication from time to time without obligation to notify any person of such revisions. If errors are found, please contact DIGITAL-LOGIC AG at the address listed on the title page of this document.

1.3 Who should use this Product

- Electronic engineers with know-how in PC-technology.
- Without electronic know-how we expect you to have questions. This manual assumes, that you have a general knowledge of PC-electronics.
- Because of the complexity and the variability of PC-technology, we can't give any warranty that the product will work in any particular situation or combination. Our technical support will help you.
- Pay attention to the electrostatic discharges. Use a CMOS protected workplace.
- Power supply OFF when you are working on the board or connecting any cables or devices.

This is a high-technology product.
You need know-how in electronics and PC-technology to install the system !

1.4 Recycling Information

Hardware:	**- Print:**	epoxy with glass fiber wires are of tin-plated copper
	- Components:	ceramics and alloys of gold, silver check your local electronic recycling
Software:	**- no problems:**	re-use the diskette after formatting

1.5 Technical Support

1. Contact your local Digital-Logic Technical Support in your country.
2. Use Internet Support Request form on http://www.digitallogic.ch -> support
3. Send a FAX or an E-mail to DIGITAL-LOGIC AG with a description of your problem.

DIGITAL-LOGIC AG
Technical Support Dept.
Nordstrasse 11/F
CH-4542 Luterbach (SWITZERLAND)

Fax: ++41-32 681 53 31
E-Mail: support@digitallogic.ch

1.6 Limited Warranty

DIGITAL-LOGIC AG warrants the hardware and software products it manufactures and produces to be free from defects in materials and workmanship for one year following the date of shipment from DIGITAL-LOGIC AG, Switzerland. This warranty is limited to the original purchaser of product and is not transferable.

During the one year warranty period, DIGITAL-LOGIC AG will repair or replace, at its discretion, any defective product or part at no additional charge, provided that the product is returned, shipping prepaid, to DIGITAL-LOGIC AG. All replaced parts and products become property of DIGITAL-LOGIC AG.

Before returning any product for repair, customers are required to contact the company.

This limited warranty does not extend to any product which has been damaged as a result of accident, misuse, abuse (such as use of incorrect input voltages, wrong cabling, wrong polarity, improper or insufficient ventilation, failure to follow the operating instructions that are provided by DIGITAL-LOGIC AG or other contingencies beyond the control of DIGITAL-LOGIC AG), wrong connection, wrong information or as a result of service or modification by anyone other than DIGITAL-LOGIC AG. Neither if the user has not enough knowledge of these technologies or has not consulted the product manual or the technical support of DIGITAL-LOGIC AG and therefore the product has been damaged.

Except, as expressly set forth above, no other warranties are expressed or implied, including, but not limited to, any implied warranty of merchantability and fitness for a particular purpose, and DIGITAL-LOGIC AG expressly disclaims all warranties not stated herein. Under no circumstances will DIGITAL-LOGIC AG be liable to the purchaser or any user for any damage, including any incidental or consequential damage, expenses, lost profits, lost savings, or other damages arising out of the use or inability to use the product.

2 OVERVIEW

2.1 Standard Features

The MICROSPACE PC is a miniaturized modular device incorporating the major elements of a PC/AT compatible computer. It includes standard PC/AT compatible elements, such as:

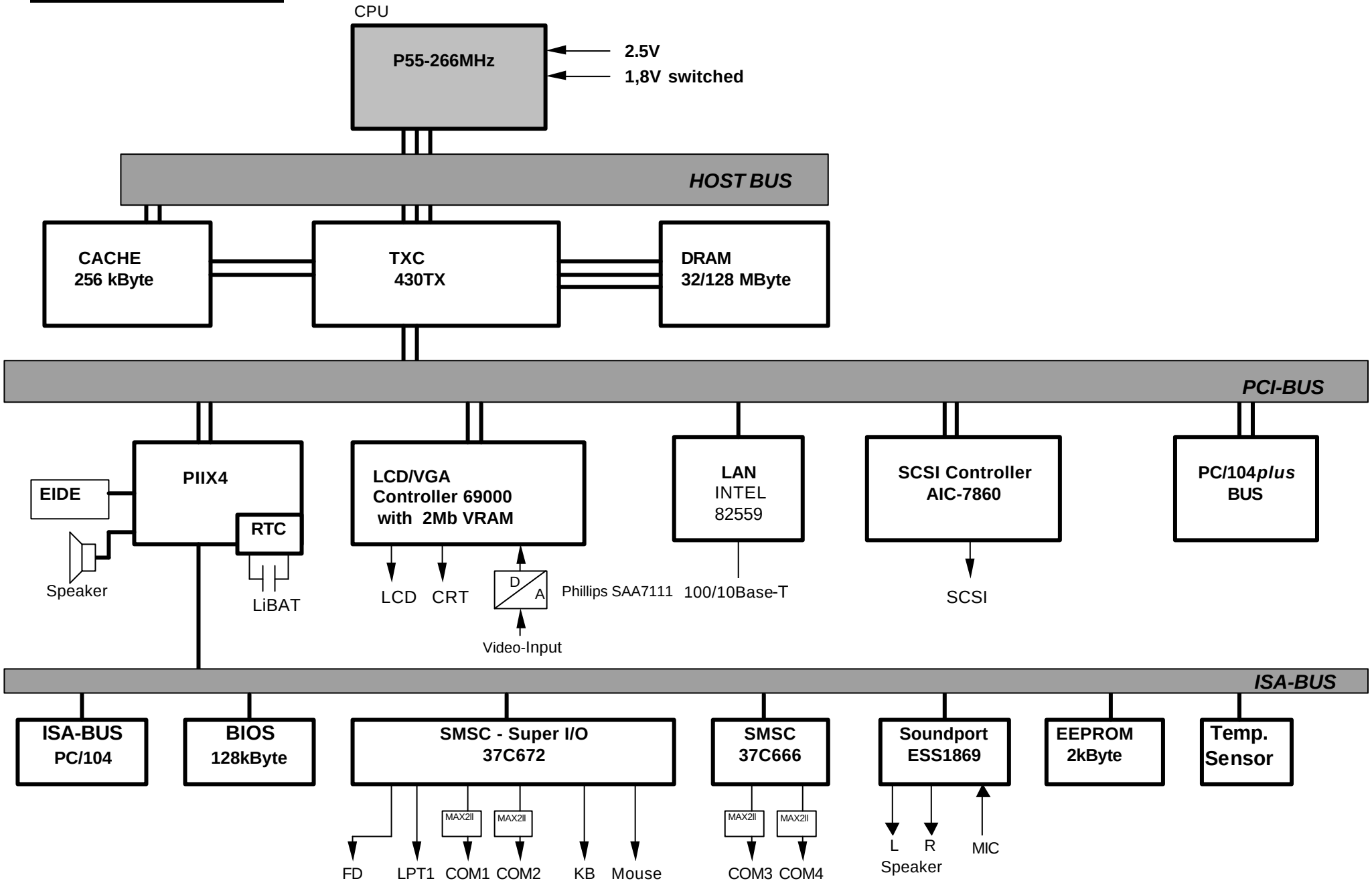
- Powerfull LowVoltage PENTIUM P55 CPU at 166/266MHz
- FLASH BIOS , downloadable
- SDRAM up to 128 MBytes
- 256k burst piplined second level cache
- Timers
- DMA
- Real-time clock with CMOS-RAM and battery buffer
- LPT1 parallel port
- COM1, COM2, COM3, COM4 serial port
- PS/2 keyboard interface
- PS/2 mouse Interface
- SVGA/LC Display interface
- Floppy disk Interface
- E-IDE harddisk interface
- PC/104plus Bus (ISA with PCI signals)
- Ethernet-LAN, 100/10 Base-T
- Video Input for 3 sources
- Soundport
- Optional SCSI-2 with 10MB/s

2.2 Unique Features

The MICROSPACE MSLB_P5 includes all standard PC/AT functions plus unique DIGITAL-LOGIC AG enhancements, such as:

- Low-power consumption, 10 watt
- Single 5 volt supply
- Watchdog
- Power-fail circuit
- EEPROM setup and configuration
- Video Input for 3 video sources PAL or NTSC
- ByteWide socket for DOC2000 (disk on chip) with up to 72MByte
- PC/104+ PCI extension
- UL approved parts

2.3 Block Diagram



2.4 MSLB-P5 Specifications

CPU:	
CPU 64 Bit:	Pentium 166MHz / 266MHz
CPU 16 Bit:	none
Mode:	Real / Protected
Compatibility:	8086 – Pentium
Word Size:	32 Bits
Physical Addressing:	32 lines
Virtual Addressing:	16 Gbytes
Clock Rates:	166 MHz
Socket Standard:	BGA, 3.3V, 1.8V switched, 2.5V linear 3.3V linear)

2nd. Level Cache:	
available	256k onboard, burst pipelined SRAM for max. performance

PC-Chipset:	
Intel	TX430

DMA:	
8237A comp.	4 channels 8 Bits 3 channels 16 Bits

Interrupts:	
8259 comp.	8 + 7 levels, PC compatible

Timers:	
8254 comp.	3 programmable counter/timers

Memory:	
DRAM	168pin SIMM

Video Output:		
Controller:	69000 from C&T	
BUS:	32 Bit highspeed 33 MHz PCI bus	
Enhanced BIOS:	Multi VGA / LCD BIOS	
Video-Memory:	2 MByte - 32 bit, expandable up to 4MByte	
CRT-Monitor:	up to 1600 x 1280 pixels	
Flatpanel:	TFT:	640 x 480, 800 x 600, 1024 x 768
	STN:	640 x 480 color and monochrome
	Plasma:	up to 1280 x 1024
	EL:	640 x 350, 640 x 480
Flatpanelinterface:	Standard: TTL 5V	

Controller Modes:	CRT only; Flatpanel only or simultaneous CRT and Flatpanel
Drivers:	Windows 3.11, WIN95, NT3.5, NT4.0 and other applications

Video Input: (option)

Controller:	69000 from C&T
BUS:	32 Bit highspeed 33 MHz PCI bus
Videoinput Norm:	3 channels with PAL or NTSC (composite video sources = CVBS) 2 channels YC sources (=SVHS) 2 channels CVBS and 1 channel SVHS this configuration may be programmed into the SAA7111
Driversupport:	WIN95, under development for NT4.0
Resolution:	720 x 512
Capture speed:	up to 30 images/second
PAL/NTSC Decoder:	Phillips SAA7111
Y-C resolution:	4:4:4 or 4:2:2 or 4:1:1
RGB resolution:	4:4:4 (16 Bit), gamma corrected

Mass Storage:

FD:	Floppy disk interface, for max. 2 floppies, 34pin connector
HD:	primary IDE 44pin interface for 2 drives (master & slave) secondary IDE 44pin interface for 2 drives (master & slave)
SCSI Devices:	PCI 10MB/s FAST SCSI-2 up to 7 devices, removable boot media AIC-7860 PCI SCSI controller

Standard AT interfaces:

Serial:

Name	FIFO	IRQs	Addr.	Stan-dard	Option
COM1	yes	IRQ4	3F8	RS232C	
COM2	yes	IRQ3	2F8	RS232C	
COM3	yes	IRQ4/11	3E8	RS232C	RS485/RS422
COM4	yes	IRQ3/10	2E8	RS232C	RS485/RS422

(Baudrates: up to 115kBaud)

Parallel:	LPT1 printer interface, in the EPP Mode bidirectional
Keyboard:	PS/2
Mouse:	PS/2
Speaker:	0.1W output drive
RTC:	146818A compatible RTC with CMOS-RAM 128Byte
Backup current:	0.2µA
Battery:	Lithium 3V, Lifetime up to 10 years at 25°C

LAN - Ethernet: (option)

Type:	IEEE 802.3
Controller:	INTEL 82559
Compatibility:	ODI-Novell
Driver:	ODI, packet-driver IEEE 802.3, NT3.5, OS/2, NDIS, NT4.0
Connector:	RJ-45 for 100/10Base-T
Data Rate:	100 or 10 MB/sec
Data-Bus:	32 Bit PCI
Cable Type:	
Remote Boot Socket:	none
RAM Buffer:	4k
Configuration:	with EEPROM

Sound I/O: (option)

Controller:	ESS1869 8Bit Soundblaster compatible
Driver Support:	WIN 3.11, WIN95, NT4
Output channels:	Stereo Output Line Level
Input:	Microphone, Line
Features:	- Compatible with: SoundBlasterPro 8 Bit, AD-Lib, MicroSoft-Windows Sound System - OPL3 Synthesizer built in - 3D Stereo Enhancement - Digital mixer - Game Port, Midi Interface - Programmable IRQs, DRQs and I/O addresses - Supports 16 Bit type F DMA playback

Supervisory:

Watchdog:	LTC1232 with power-fail detection

USB:

Controller:	PIIX4
Transferrate:	12.5MBps / 1.5 MBps
Channels:	2

BUS internal:

PCI	IEEE-996 standard bus, by Intel
Clock:	33MHz with 32bit data path and 100MB/s transferrate

BUS external:

ISA	IEEE-996 standard bus
Clock:	8 MHz or programmable

Embedded BUS:

PC/104	IEEE-996 standard bus
Clock:	8 MHz or programmable
PC/104+	PCI 32bit, 33Mhz bus , this is an assembly option

Power Supply:

Working:	5 Volts ± 5%
Current:	2.3A nominal, using P55-166 MHz
Suspend:	0.9A
Rise Time:	100µs +/- 10% from 0V to 4.75V

Physical Characteristics:

Dimensions:	Length:	204 mm
	Depth:	140 mm
	Height:	40 mm
Weight:	300gr	
PCB Thickness:	1.6 mm / 0.0625 inches nominal	
PCB Layer:	10 with separate ground and VCC plane for low noise	

Operating Environment:

Relative humidity:	5 - 90% non condensing		
Vibration:	5 to 2000 Hz		
Shock:	10 G		
Temperature:	Operating:	Standard version:	-25°C to +70°C
		Enhanced temp. range:	-40°C to +85°C -E48
	Storage:	-55°C to +85°C	

Cooling:

Standard:	5V Fan, with feedback (frequency sensor)
Temperaturesensor:	onboard, located in the center below the CPU
Option:	passive with air flow > 1000ft/Min.

EMI / EMC (IEC1131-2 refer MIL 461/462):

ESD Electro Static Discharge:	IEC 801-2, EN55101-2, VDE 0843/0847 Part 2 metallic protection needed, separate Ground Layer included,15kV single peak
REF Radiated Electromagnetic Field:	IEC 801-3, VDE 0843 Part 3, IEC770 6.2.9. not tested
EFT Electric Fast Transient (Burst):	IEC 801-4, EN50082-1, VDE 0843 Part 4 250V - 4kV, 50 ohms, Ts=5ns Grade 2: 1KV Supply, 500 I/O, 5Khz
SIR Surge Immunity Requirements:	IEC 801-5, IEEE587, VDE 0843 Part 5 Supply: 2 kV, 6 pulse/minute I/O: 500 V, 2 pulse/minute
High-Frequency Radiation:	EN55022

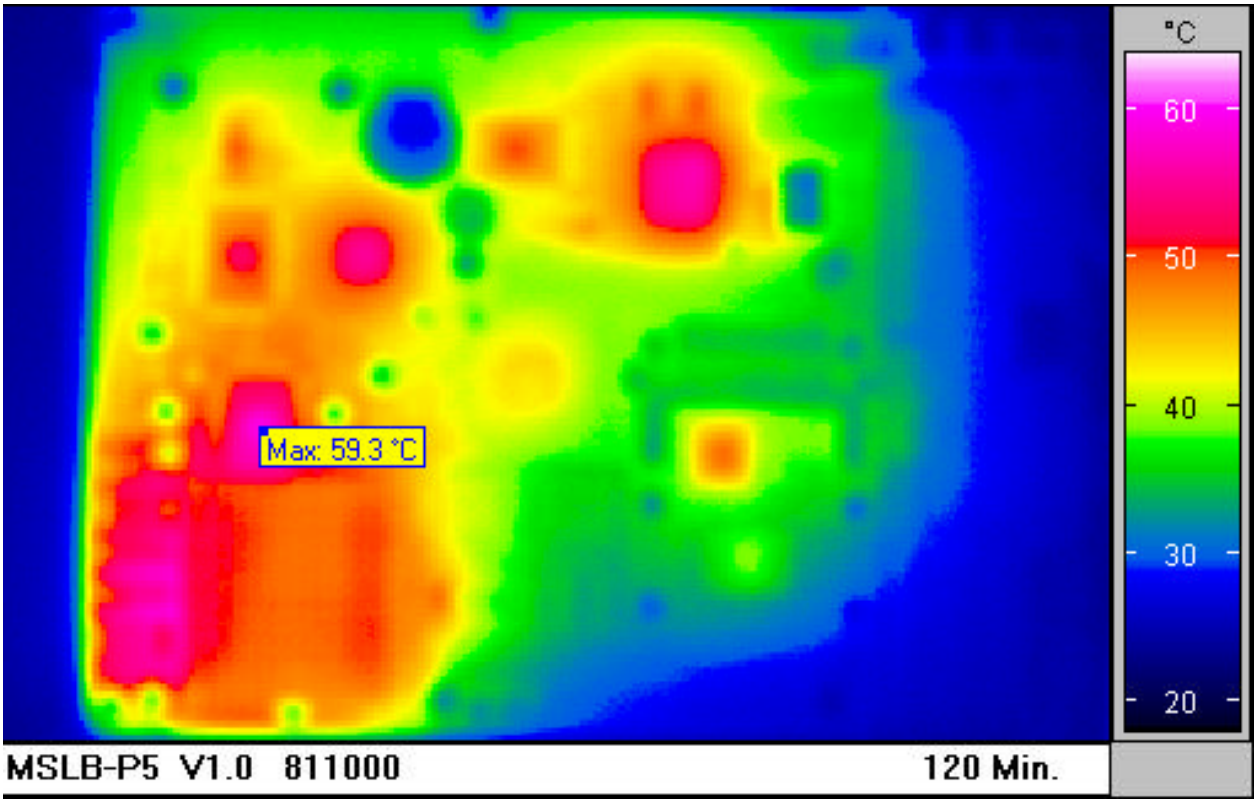
Compatibility:

MSLB:	mechanically compatible to standard EBX single boards

Any information is subject to change without notice.

2.5 Thermoscan

Product: MSLB-P5 Scan time: 60min.



2.6 Ordering Codes

MSLB-P5	MICROSPACE PCCard without CPU, with 256k Cache no Options
Option - E	LAN 100/10BASE-T Option
Option - S	SCSI-2
Option - A	Audio Sound Interface

2.7 BIOS History

Version:	Date:	Status:	Modifications:
BETA			

2.8 This product is "YEAR 2000 CAPABLE"

This DIGITAL-LOGIC product is "YEAR 2000 CAPABLE". This means, that upon installation, it accurately stores, displays, processes, provides and/or receives date data from, into, and between 1999 and 2000, and the 20. and 21. centuries, including leap year calculations, provided that all other technology used in combination with said product properly exchanges date data with it. DIGITAL-LOGIC makes no representation about individual components within the product should be used independently from the product as a whole.
You should understand that DIGITAL-LOGIC's statement that an DIGITAL-LOGIC product is "YEAR 2000 CAPABLE" means only that DIGITAL-LOGIC has verified that the product as a whole meet this definition when tested as a stand-alone product in a test lab, but dies not mean that DIGITAL-LOGIC has verified that the product is "YEAR 2000 CAPABLE" as used in your particular situation or configuration. DIGITAL-LOGIC makes no representation about individual components, including software, within the product should they be used independently from the product as a whole.

DIGITAL-LOGIC customers use DIGITAL-LOGIC products in countless different configurations and in conjunction with many other components ans systems, and DIGITAL-LOGIC has no way to test wheter all those configurations and systems will properly handle the transition to the year 2000. DIGITAL-LOGIC encourages its customers and others to test whether their own computer systems and products will properly handle the transition to the year 2000.

The only proper method of accessing the date in systems is indirectly from the Real-Time-Clock via the BIOS. The BIOS in DIGITAL-LOGIC computerboards contain a century checking and maintenance feature the checks the laest two significant digits of the year stored in the RTC during each BIOS request (INT 1A) to read the date and, if less than '80' (i.e. 1980 is the first year supported by the PC), updates the century byte to '20'. This feature enables operating systems and applications using BIOS date/time services to reliably manipulate the year as a four-digit value.

2.9 High frequency Radiation (to meet EN55022)

Since the boards are very high integrated embedded computers, no peripheral lines are protected against the radiation of high frequency spectrum. To meet a typical EN55022 requirement, all peripherals, they are going outside of the computer case, must be filtered externaly.

Typical signals, they must be filtered:

Interface:	Signals:	onboard filter on the MSLB-P5:
Keyboard:	KBCLK, KBDATA, VCC	onboard filter
Mouse:	MSCLK, MSDATA, VCC	onboard filter
COM1/2/3/4:	All serial signals must be filtered	not filtered
LPT:	All parallel signals must be filtered	not filtered
CRT:	red,blue,green, hsynch, vsynch must be filtered	onboard filter
Powerlines:	+5V, +12V, Ground	not filtered

Typical signals, they must not be filtered, since they are internaly used:

IDE:	connected to the harddisk
Floppy:	connected to the floppy
LCD:	connected to the internal LCD

1. For peripheral cables:

Use for all DSUB connector a filtered version. Select carefully the filter specifications. Place the filtered DSUB connector directly frontside and be shure that the shielding makes a good contact with the case.

9pin DSUB connector from AMPHENOL:	FCC17E09P	820pF
25pin DSUB connector from AMPHENOL:	FCC17B25P	820pF

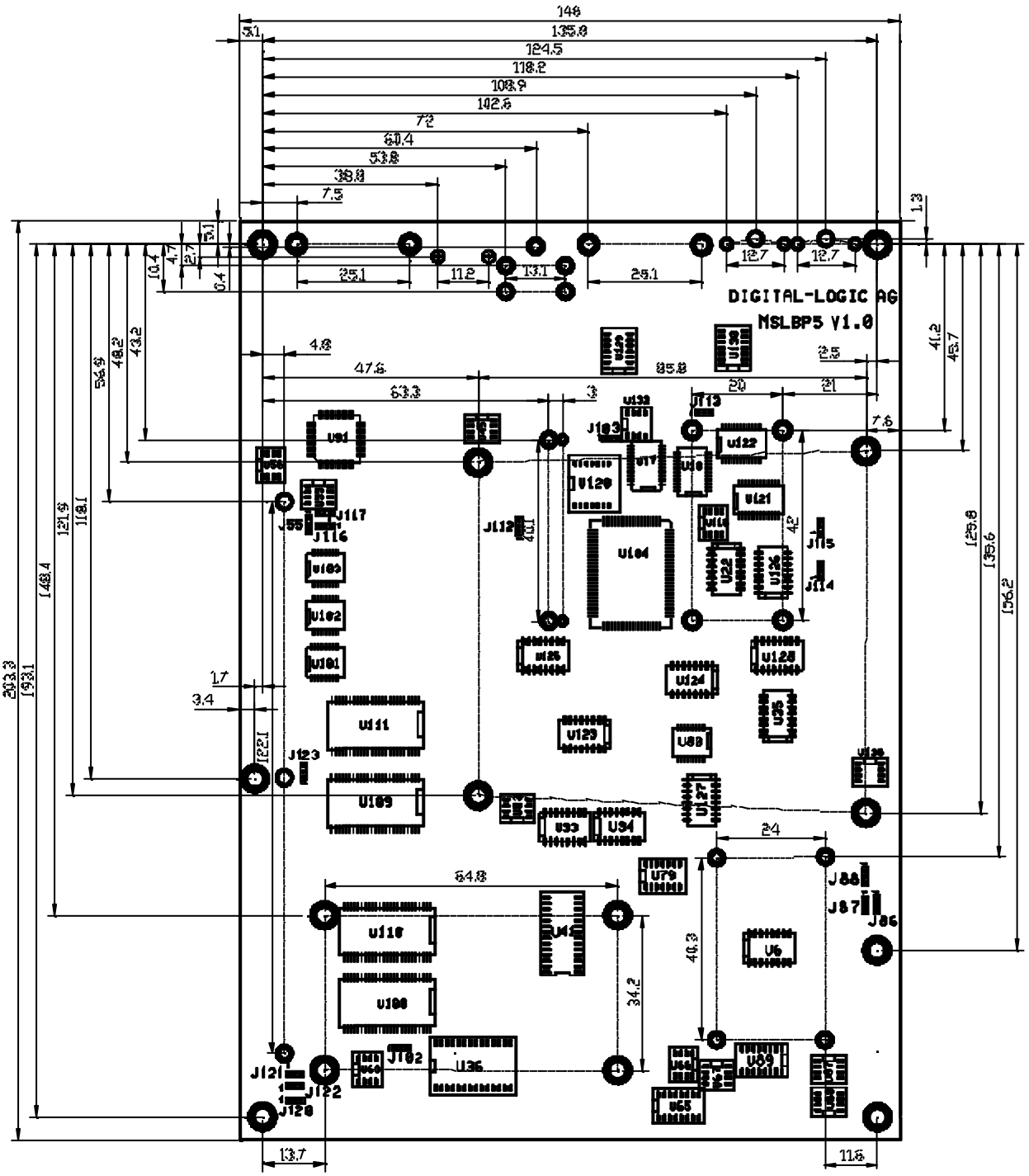
2. Power supply:

Use a currentcompensated dualinductor on the 5V supply

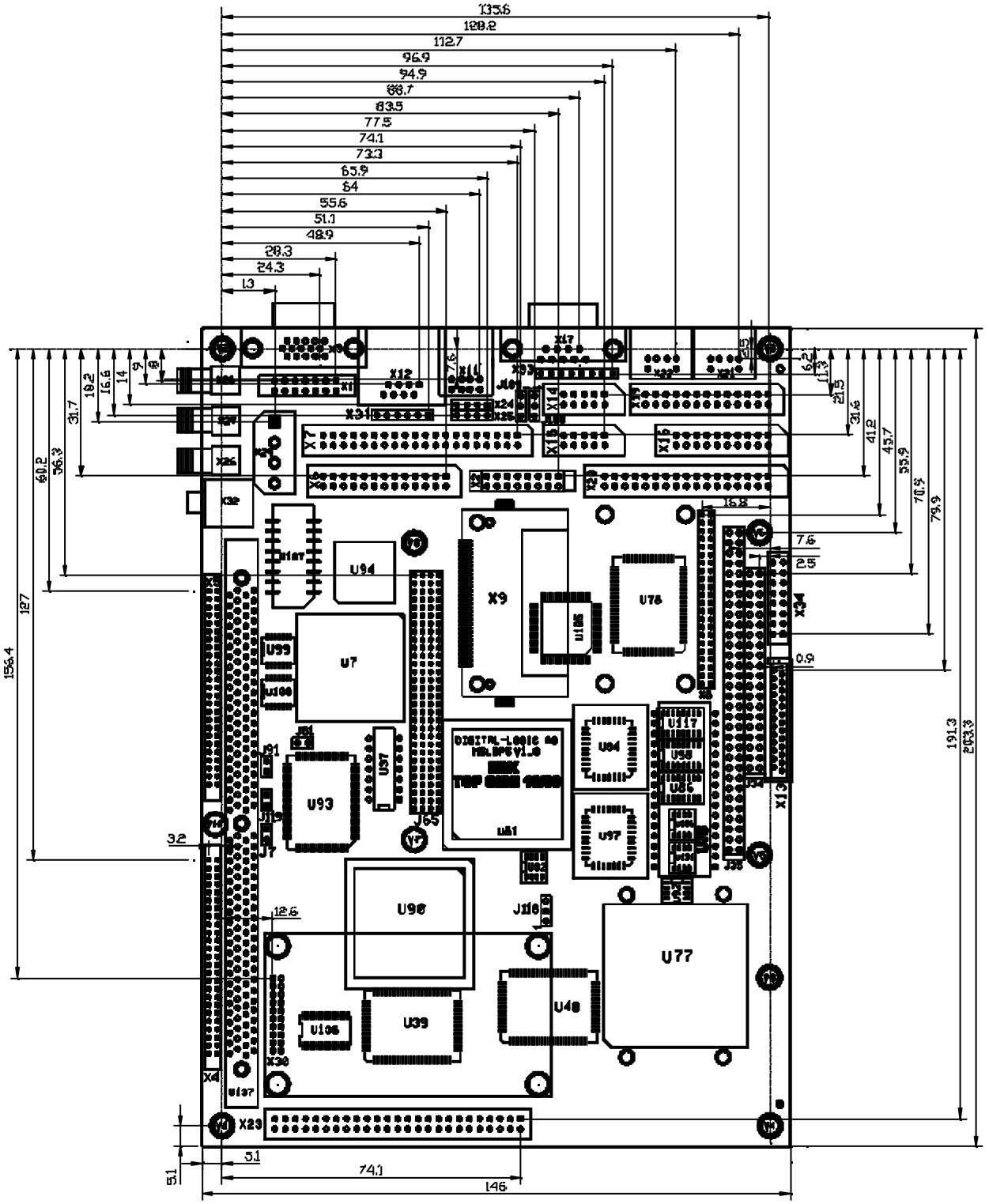
SIEMENS B82721-K2362-N1 with 3.6A , 0.4mH

2.10 Mechanical Dimensions

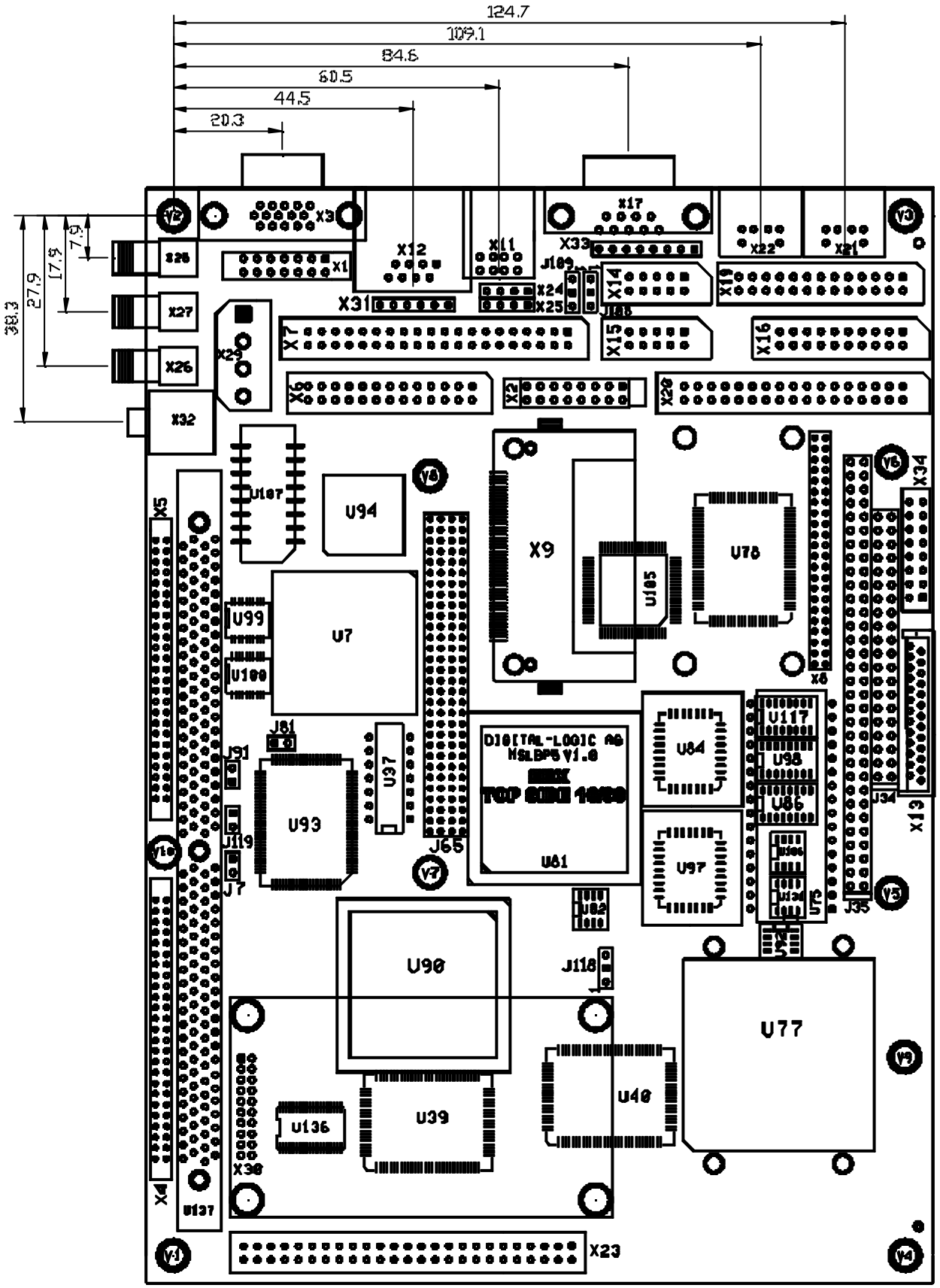
2.10.1 Mechanical Dimensions Mounting Holes



2.10.2 Mechanical Dimensions Connectors



2.10.3 Mechanical Dimensions Frontside



3 THE PCI, ISA , EPCI AND PC/104 BUS SIGNALS

3.1 ISA Signals on the PC/104 and ISA-Bus

AEN, output
Address Enable is used to degate the microprocessor and other devices from the I/O channel to allow DMA transfers to take place. **low = CPU Cycle , high = DMA Cycle**

BALE, output
Address Latch Enable is provided by the bus controller and is used on the system board to latch valid addresses and memory decodes from the microprocessor. This signal is used so that devices on the bus can latch LA17..23. The SA0..19 address lines latched internally according to this signal. BALE is forced high during DMA cycles.

/DACK[0..3, 5..7], output
DMA Acknowledge 0 to 3 and 5 to 7 are used to acknowledge DMA requests (DRQO through DRQ7). They are **active low**. This signal indicates that DMA operation can begin.

DRQ[0..3, 5..7], input
DMA Requests 0 through 3 and 5 through 7 are asynchronous channel requests used by peripheral devices and the I/O channel microprocessors to gain DMA service (or control of the system). A request is generated by bringing a DRQ line to an active level. A DRQ line must be held high until the corresponding DMA Request Acknowledge (DACK/) line goes active. DRQO through DRQ3 will perform 8-bit DMA transfers; DRQ5-7 are used for 16 accesses.

/IOCHCK, input
IOCHCK/ provides the system board with parity (error) information about memory or devices on the I/O channel. **low = parity error , high = normal operation**

IOCHRDY, input
I/O Channel Ready is pulled low (not ready) by a memory or I/O device to lengthen I/O or memory cycles. Any slow device using this line should drive it low immediately upon detecting its valid address and a Read or Write command. Machine cycles are extended by an integral number of one clock cycle (67 nanoseconds). This signal should be held low for no more than 2.5 microseconds. **low = wait, high = normal operation**

/IOCS16, input
I/O 16 bit Chip Select signals the system board that the present data transfer is a 16-bit, 1 wait-state, I/0 cycle. It is derived from an address decode. /IOCS16 is **active low** and should be driven with an open collector (300 ohm pull-up) or tri-state driver capable of sinking 20mA. The signal is driven based only on SA15-SAO (not /IOR or /IOW) when AEN is not asserted. In the 8 bit I/O transfer, the default transfers a 4 wait-state cycle.

/IOR, input/output
I/O Read instructs an I/O device to drive its data onto the data bus. It may be driven by the system microprocessor or DMA controller, or by a microprocessor or DMA controller resident on the I/O channel. This signal is **active low**.

/IOW, input/output

I/O Write instructs an I/O device to read the data on the data bus. It may be driven by any microprocessor or DMA controller in the system. This signal is **active low**.

IRQ[3 - 7, 9 - 12, 14, 15], input
These signals are used to tell the microprocessor that an I/O device needs attention. An interrupt request is generated when an IRQ line is **raised from low to high**. The line must be held high until the microprocessor acknowledges the interrupt request .

/Master, input
This signal is used with a DRQ line to gain control of the system. A processor or DMA controller on the I/0 channel may issue a DRQ to a DMA channel in cascade mode and receive a /DACK.

/MEMCS16, input
MEMCS16 Chip Select signals the system board if the present data transfer is a 1 wait-state, 16-bit, memory cycle. It must be derived from the decode of LA17 through LA23. /MEMCS16 should be driven with an open collector (300 ohm pull-up) or tri-state driver capable of sinking 2OmA.

/MEMR input/output
These signals instruct the memory devices to drive data onto the data bus. /MEMR is active on all memory read cycles. /MEMR may be driven by any microprocessor or DMA controller in the system. When a microprocessor on the I/0 channel wishes to drive /MEMR, it must have the address lines valid on the bus for one system clock period before driving /MEMR active. These signals are **active low**.

/MEMW, input/output
These signals instruct the memory devices to store the data present on the data bus. /MEMW is active in all memory read cycles. /MEMW may be driven by any microprocessor or DMA controller in the system. When a microprocessor on the I/O channel wishes to drive /MEMW, it must have the address lines valid on the bus for one system clock period before driving /MEMW active. Both signals are **active low.**

OSC, output
Oscillator (OSC) is a high-speed clock with a 70 nanosecond period (14.31818 MHz). This signal is not synchronous with the system clock. It has a 50% duty cycle. OSC starts 100us after reset is inactive.

RESETDRV, output
Reset Drive is used to reset or initiate system logic at power-up time or during a low line-voltage outage. This signal is active high. When the signal is active all adapters should turn off or tri-state all drivers connected to the I/O channel. This signal is driven by the permanent Master.

/REFRESH, input/output
These signals are used to indicate a refresh cycle and can be driven by a microprocessor on the I/0 channel. These signals are **active low**.

SAO-SA19, LA17 - LA23 input/output
Address bits 0 through 19 are used to address memory and I/0 devices within the system. These 20 address lines, allow access of up to 1MBytes of memory. SAO through SA19 are gated on the system bus when BALE is high and are latched on the falling edge of BALE. LA17 to LA23 are not latched and addresses the full 16 MBytes range. These signals are generated by the microprocessors or DMA controllers. They may also be driven by other microprocessor or DMA controllers that reside on the I/0 channel. The SA17-SA23 are always LA17-LA23 address timings for use with the MSCS16 signal. This is advanced AT96 design. The timing is selectable with jumpers LAxx or SAxx.

/SBHE, input/output
Bus High Enable (system) indicates a transfer of data on the upper byte of the data bus, SD8 through SD15. Sixteen-bit devices use /SBHE to condition data-bus buffers tied to SD8 through SD15.

SD[O..15], input/output
These signals provide bus bits 0 through 15 for the microprocessor, memory, and I/0 devices. DO is the least-significant bit and D15 is the most significant bit. All 8-bit devices on the I/O channel should use DO through D7 for communications to the microprocessor. The 16-bit devices will use DO through D15. To support 8-bit device, the data on D8 through D15 will be gated to DO through D7 during 8-bit transfers to these devices; 16-bit microprocessor transfers to 8-bit devices will be converted to two 8-bit transfers.

/SMEMR input/output
These signals instruct the memory devices to drive data onto the data bus for the first MByte. /SMEMR is active on all memory read cycles. /SMEMR may be driven by any microprocessor or DMA controller in the system. When a microprocessor on the I/0 channel wishes to drive /SMEMR, it must have the address lines valid on the bus for one system clock period before driving /SMEMR active. The signal is **active low**.

/SMEMW, input/output
These signals instruct the memory devices to store the data present on the data bus for the first MByte. /SMEMW is active in all memory read cycles. /SMEMW may be driven by any microprocessor or DMA controller in the system. When a microprocessor on the I/O channel wishes to drive /SMEMW, it must have the address lines valid on the bus for one system clock period before driving /SMEMW active. Both signals are **active low.**

SYSCLK, output
This is a 8 MHz system clock. It is a synchronous microprocessor cycle clock with a cycle time of 167 nanoseconds. The clock has a 50% duty cycle. This signal should only be used for synchronization. It is not intended for uses requiring a fixed frequency.

TC output
Terminal Count provides a pulse when the terminal count for any DMA channel is reached. The TC completes a DMA-Transfer. This signal is expected by the onboard floppy disk controller.

/OWS, input
The Zero Wait State (/OWS) signal tells the microprocessor that it can complete the present bus cycle without inserting any additional wait cycles. In order to run a memory cycle to a 16-bit device without wait cycles, /OWS is derived from an address decode gated with a Read or Write command. In order to run a memory cycle to an 8-bit device with a minimum of one-wait states, /OWS should be driven active one system clock after the Read or Write command is active, gated with the address decode for the device. Memory Read and Write commands to an 8-bit device are active on the falling edge of the system clock. /OWS is **active low** and should be driven with an open collector or tri-state driver capable of sinking 2OmA.

12V +/- 5%
used only for the flatpanel backligth supply only.

GROUND = 0 Volt
used for the entire system.

VCC, +5V +/- 0.25 Volt
3-4 Amp. nominal, peak current of HD, SCSI-Devices and Cache could go up to 8 Amp..

3.2 Connect PCI peripheral cards to the PC/104plus BUS:

Internal used PCI-Signals: LAN: A_D29, IRQA, REQ1, GNT1
SCSI: A_D30, IRQB, REQ2, GNT2

Signals:	**PC/104plus:**	**1. PCI-Card**	**2. PCI-Card**
	Master	Slave	Slave
PCI_Clock:	PCI_Clock_3 ---------->	PCI_Clock 3	
	PCI_Clock_2 ----------		> PCI_Clock 2
GNT:	GNT0 ---------->	GNT0 ---------->	GNT0
	GNT1 ---------->	GNT1 ---------->	GNT1
	GNT2 ---------->	GNT2 ---------->	GNT2
	GNT3 ---------->	GNT3 ---------->	GNT3
REQ:	REQ0 ---------->	REQ0 ---------->	REQ0
	REQ1 ---------->	REQ1 ---------->	REQ1
	REQ2 ---------->	REQ2 ---------->	REQ2
	REQ3 ---------->	REQ3 ---------->	REQ3
PIRQ	PIRQA ---------->	PIRQA ---------->	PIRQA
	PIRQB ---------->	PIRQB ---------->	PIRQB
	PIRQC ---------->	PIRQC ---------->	PIRQC
	PIRQD ---------->	PIRQD ---------->	PIRQD
IDSEL:	A_D22 ---------->	IDSEL	
	A_D23 ----------		> IDSEL

The number of the PCI cards is limited to totally 4 cards (inclusive PC/104plus).

Following combinations are possible:

onboard SCSI	onboard LAN	PC/104plus Bus	EPCI-Bus	Comments	Total PCI Loads
yes	yes	0	2		8
yes	yes	1	1		8
yes	yes	2	0		8
no *	yes *	0	3		9
no *	yes *	1	2		9
no *	yes *	2	1		9
no *	yes *	3	0		9
no	no	0	4		10
no	no	1	3		10
no	no	2	2		10
no	no	3	1		10
no	no	4	0		10

Attention:
All signals must be routed by observing the definition in the PCI specification !

* The same limitation is valid for SCSI=yes and LAN = no

3.3 EBX Specification — Ver 1.1 — July 9, 1997

1. INTRODUCTION

Until now, embedded system designers had to choose among off-the-shelf backplane solutions, desktop motherboards, and proprietary designs. Size and power consumption constraints hampered finding the right solutions for embedded deployment. Consequently, OEMs wanting to purchase off-the-shelf equipment to shorten time-to-market were often forced to develop proprietary solutions.

Standards are important to the embedded systems market. Popular backplane form-factors — including VME, CompactPCI?, Multibus?, STD32?, and passive backplane ISA — are well documented mechanical and electrical standards. Desktop motherboards, which fit certain high-end embedded applications, also follow standards such as Baby AT, LPX, ATX, and the new NLX standard. All these standards allow vendors and OEMs to create products that are easily packaged in enclosures and readily expanded via open interfaces. However, none of these backplane-based standards satisfy the unique space, power, and reliability constraints of small embedded systems.

The availability of an embedded single-board computer (SBC) standard will ensure that embedded computing solutions can be designed into space constrained environments with off-the-shelf components. The embedded market constantly demands improvements in functionality and performance, while at the same time seeking size and cost reduction. The “Embedded Board, eXpandable” (EBX) standard creates the opportunity for solutions which fit the requirements of embedded system OEMs; takes advantage of trends in the embedded computing market; and offers the convenience, flexibility, risk reduction, and scalability of multi-sourced off-the-shelf products.

The “Embedded Board, eXpandable” (EBX) standard is the result of a collaboration between industry leaders, Motorola and Ampro, to unify the embedded computing industry on a small footprint embedded single-board computer standard. Derived from the Ampro Little Board??form-factor, EBX combines a standard footprint with open interfaces. The EBX form-factor is small enough for deeply embedded applications, yet large enough to contain the functions of a full embedded computer system: CPU, memory, mass storage interfaces, display controller, serial/parallel ports, and other system functions.

This EBX system expansion is based on popular existing industry standards — IEEE P996, PC/104?, PCI, PC/104-*Plus*?, and PCMCIA. IEEE P996 is the governing standard for the PC and PC/AT buses, informally known as the Industry Standard Architecture or “ISA.“ PC/104 places the P996 ISA bus on compact 3.6” x 3.8” modules with self-stacking capability. PC/104-*Plus* adds the power of a PCI bus to PC/104 while retaining the basic form-factor. For further expansion flexibility, PCMCIA offers access to PC Cards from the mobile and handheld computing markets.

The EBX standard integrates all these off-the-shelf standards into a highly embeddable SBC form-factor. EBX supports the legacy of PC/104, hosting the wide variety of embedded system oriented expansion modules from hundreds of companies worldwide. PCMCIA brings the advantages of the latest portable and mobile system expansion technologies to embedded applications. Additionally, the EBX PCI infrastructure and PC/104-*Plus* expansion bus offer true processor independence and high performance standards-based system expansion.

EBX compliant boards have a form-factor large enough to implement a powerful SBC capable of hosting today’s advanced operating systems, yet small enough to fit in the tight spaces of deeply embedded applications. This creates an exciting new opportunity for embedded system OEMs to standardize their designs and take advantage of off-the-shelf modules.

The EBX standard is open to continuing technology advancements, since it is both processor and payload independent. It creates opportunity for economies of scale in chassis, power supply, and peripheral devices. It defines how products interoperate by providing mechanical rules for mandatory features and recommended zones for flexible I/O options. These attributes combine to make EBX the right choice for embedded computing.

The aligning of DIGITAL-LOGIC, Ampro and Motorola, embedded computing industry leaders, brings stability to the embedded board market and offers OEMs assurance that a wide range of products will be available from multiple sources — now and in the future. The EBX specification is freely available to all interested companies, and may be used without licenses or royalties. For further technical information on the EBX standard, please contact:

DIGITAL-LOGIC INC

2. REFERENCE DOCUMENTS

This EBX specification makes reference to, and is based on, the current versions of the following specifications:

IEEE P996 Draft Specification: IEEE Standards Office, Piscataway NJ; phone 908-562-3825, fax 908-562-1571.

PC/104 and PC/104-*Plus* Specification: PC/104 Consortium, Mountain View CA; phone 415-903-8304, fax 415-967-0995.

PCI Local Bus Specification: PCI Special Interest Group, Hillsboro OR; phone 800-443-5177 or 503-693-6232, fax 503-693-8344, email info@pcisig.com.

PC Card Standard: PCMCIA, San Jose CA; phone 408-433-2273, fax 408-433-9558, email office@pcmcia.org.

Technical references about the PCI and ISA buses themselves are available from numerous sources, including Annabooks (toll free 800-462-1042), the Computer Literacy Bookshops (408-435-0744), and others.

3. HORIZONTAL DIMENSIONS AND MOUNTING HOLES

Figure 1 in Appendix A provides the detailed horizontal dimensions and mounting hole locations of the EBX form-factor. With the exception of the four holes labeled “B”, all dimensions indicated in Figure 1 for board size and mounting holes are *mandatory*.

3.1 Horizontal Dimensions

The horizontal dimensions of an EBX board are 5.75 x 8.00 inches (146 by 203 mm).

3.2 Mounting Holes

Eight mounting holes are specified. These are marked “A” in Figure 1. Four of these are located in the corners of the EBX form-factor, and four others correspond to the PC/104-*Plus* module mounting locations defined by the PC/104-*Plus* specification. It is recommended that all eight defined mounting holes be used to provide rugged attachment of the EBX board to its enclosure or parent assembly.

Note that the four holes marked “B” in Figure 1 are optional. These holes are for the screws used to mount a typical PC Card slot connector that meets the mechanical requirements of EBX. The connector that matches these mounting holes is indicated in Section 4.9 of this document.

3.2 PC/104-*Plus* Expansion Stack Location

EBX provides a “PC/104-*Plus* Bus Compatible” module stack location as defined by the PC/104 and PC/104-*Plus* specifications. This location accepts either PC/104-*Plus* (PCI) or PC/104 (ISA) expansion modules, or a combination of both. Figure 1 defines the precise location of the PC/104-*Plus* expansion stack location, based on the location of the PC/104-*Plus* bus connectors and associated mounting holes. Refer to the PC/104 and PC/104-*Plus* specifications for information on the full electrical and mechanical specifications associated with this location.

3.2.1 PC/104-*Plus* Bus Connectors

PC/104-*Plus* defines two buses. One is the 104-pin ISA connector pair (J1/J2) which consists of 64-pin and 40-pin pin-and-socket headers with 0.1 in. pin-to-pin spacing. The second bus is the 120-pin PCI connector (J3), a high density pin-and-socket connector with 2mm pin-to-pin spacing. These bus connectors and their typical vendor part numbers appear in Table 1.

3.2.2 Stackthrough Bus Option

The PC/104-*Plus* specification defines ether stackthrough or non-stackthrough bus connectors. An EBX board may be populated with either of these bus options, as indicated in Table 1. When fitted with stackthrough bus connectors, the EBX board can be plugged onto another circuit board (often called a “baseboard”) and treated like a single-board computer “macrocomponent”.

3.2.3 PC/104-*Plus* Keep Out Area

EBX preserves the mandatory “keep out” areas defined by the PC/104-*Plus* specification.

4. VERTICAL CLEARANCE ZONES

The EBX form-factor is subdivided into zones which are intended for various interfaces and components. Each of these zones, and their associated functions, are defined in Figures 2 and 3 (Appendix A) and are described below. Each zone has a specified vertical dimension within which all components of that zone must fit. Table 2 specifies the maximum component height within each EBX zone.

Figure 2 or 3 will apply, depending on whether the Tall CPU or PC Card option is desired. Many EBX compliant boards have single board computer functions. including memory expansion, PC Card slots, Ethernet ports, mass storage and auxiliary ports, and CRT and LCD interfaces. EBX does not require all these functions, nor does it specify that they *must* appear in a particular location. However, observing these guidelines facilitates interoperability among multiple EBX form-factor products, such as compatibility with multivendor packaging.

Table 2. EBX Vertical Clearance Zones

Zone	Description	Max. Component Height (in.)
A	Memory expansion	1.5
B	Power connector	0.5
C	Video I/O (option)(includes mating connectors)	0.75
D	Misc. primary side components	0.75
E	General purpose I/O, tall region (includes mating connectors)	0.75
F	PC/104- Plus stack location (Primary and secondary side)	See PC/104- Plus spec
G	PC/104- Plus module I/O areas	0.6
H	Tall CPU (option) (includes heat sink)	1.2
I	PC Card slot (option)	0.6
J	General purpose I/O, low profile region (includes mating connectors)	0.5
---	Secondary side components	0.19
---	Board thickness	0.062

4.1 Zone A: Memory Expansion
Most EBX boards will require expansion memory, and this zone is recommended to allow the height profile necessary for industry standard SIMMs or DIMMs.
4.2 Zone B: Power Connector
The 7-pin EBX power connector and external mating connector are located in this zone. Refer to Section 5 of this specification for further information.
4.3 Zone C: Video I/O (Option)
Many EBX boards will provide onboard interface to CRT and/or flat panel displays. It is recommended that the I/O connectors for external display devices be located within this zone. Both the EBX board connectors and the typical mating cable connectors must fit within the defined height profile.
4.4 Zone D: Miscellaneous Primary Side Components
Any primary side components within this zone must fit within the defined height profile.
4.5 Zone E: General Purpose I/O, Tall Region
This zone is defined for I/O expansion interfaces for functions such as IDE, floppy, SCSI, keyboard, mouse, serial ports, parallel ports, etc. Both the EBX board connectors and the typical mating cable connectors must fit within the defined height profile.
4.6 Zone F: PC/104-*Plus* Expansion Stack Location
This zone is for the onboard PC/104-*Plus* expansion stack. For the required height profile within this zone, refer to Figure 4 (Module Dimensions) of the PC/104-*Plus* specification.
4.7 Zone G: PC/104-Plus I/O Areas
The two areas marked “G” correspond to the I/O connector areas of the PC/104 and PC/104-*Plus* module specifications. Components on the EBX board must not be too tall to fit beneath the I/O connectors of the PC/104 (*Plus*) module and must therefore conform to the height profile defined for this zone. Note that the PC/104 (*Plus*) module I/O connectors and mating cable connectors are expected to fit entirely within the two sets of horizontal boundaries indicated by “G” in Figures 2 and 3 of Appendix A.
4.8 Zone H: Tall CPU (Option)
CPUs requiring a tall heatsink or fan attachment are recommended to be located in this zone, as defined in Figure 2. The defined height profile for this zone includes the CPU and its associated heatsink assembly. In this case, use of PC Cards will require a PC/104 or PC/104-*Plus* expansion module or other external adapter.
4.9 Zone I: PC Card Slot (Option)
If an onboard PC Card expansion slot is used, its location should be as defined in Figure 3. When fully inserted, the external edge of the PC Card is flush with the outside edge of the EBX board as indicated in Figure 3; the location of the center of the card is also indicated in Figure 3. Figure 1 indicates four holes marked “B” that correspond to the location of the screws used to mount a specific PC Card connector, Berg part number 95547-XXX (or equivalent).
4.10 Zone J: General Purpose I/O, Low Profile Region
This zone is defined for I/O expansion interfaces for functions such as IDE, floppy, SCSI, keyboard, mouse, serial ports, parallel ports, etc. Both the EBX board connectors and the typical mating cable connectors must fit within the defined height profile.
4.11 Secondary Side Components
All components on the “secondary side” (bottom) of the EBX board, with the exception of the PC/104-*Plus* module area, must fit within this dimension. If the “stackthrough bus” option is employed, secondary side components in the PC/104-*Plus* module area must conform to the secondary side component height requirements specified in the PC/104-*Plus* specification.
4.12 Board Thickness
This dimension specifies the thickness of the EBX PC board material.

5. POWER CONNECTOR AND POWER REQUIREMENTS

The EBX power connector is a 7-pin locking connector. Two options are supported, right-angle and straight, as illustrated in the figure below. The power connector options are Molex part number 26-60-7070 for the right-angle, and 26-60-4070 for the straight (or equivalent). Figures 2 and 3 in Appendix A define the region where the power connector and its mating cable connector must be located. The figure below illustrates the orientation of the two power connector options on the EBX board.

Power Connector Placement

5.1 Mating Connector

The mating connector for either option of power connector consists of a shell and associated pins; these are Molex part numbers 09-50-8073 (shell) with appropriate pins, or equivalent.

5.2 Power Requirements

The EBX specification only defines the available input voltages; it does not specify any electrical requirements for any of the referenced standards such as PC/104, PC/104-Plus, PCMCIA, or the various supported I/O interfaces.

EBX boards are not obligated to use all these voltages. Specified input voltages on the pins of the 7-pin EBX power connector, and the associated maximum currents, are given in Table 3.

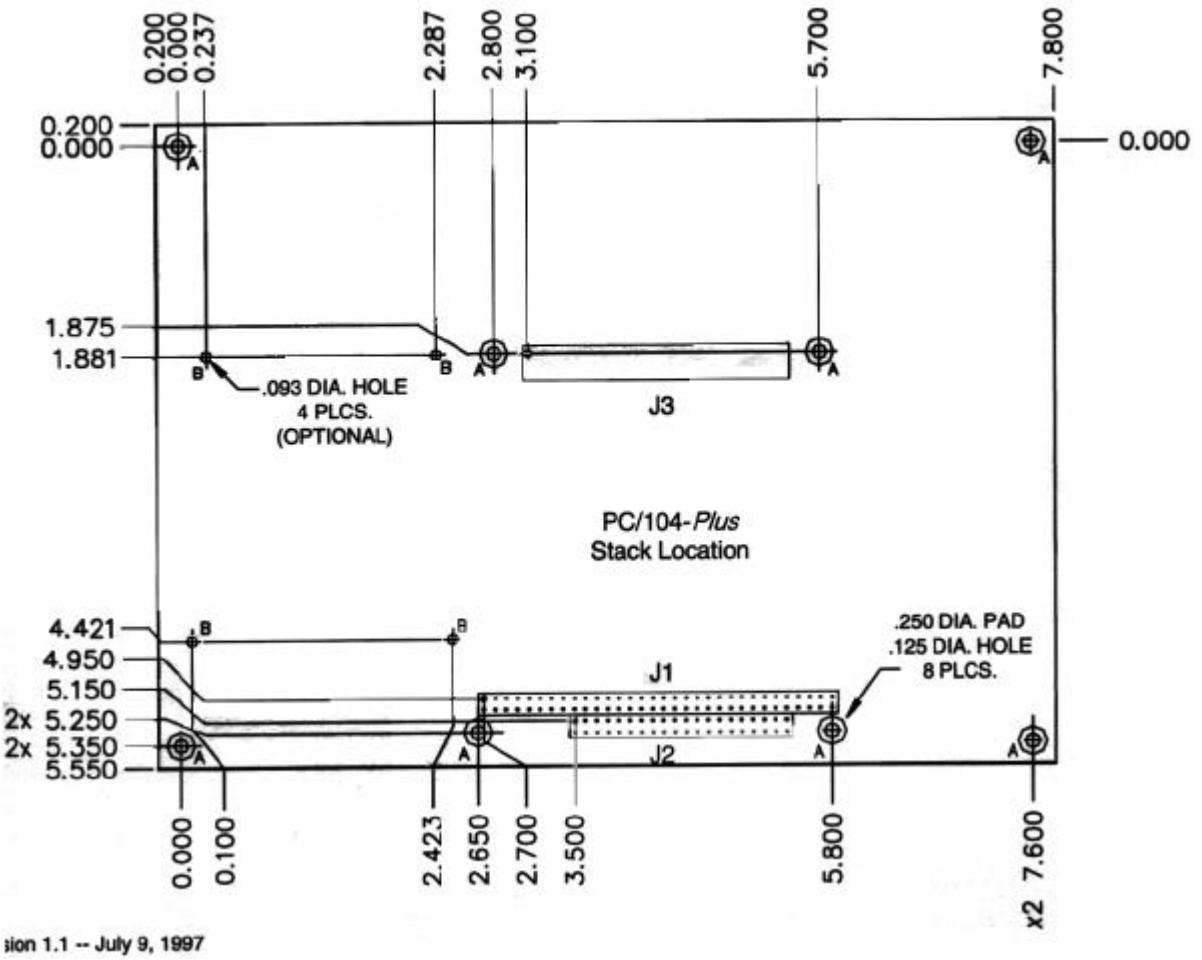
Table 3. Power Connector Pinout and Voltage Requirements

Supply Connector Pin(s)		Maximum-Voltage	Minimum-Voltage
4	+12V	+12.6V	+11.4V
1	+5V	+5.25V	+4.75V
2,3	Ground		

maximum current capacity is 7A per pin

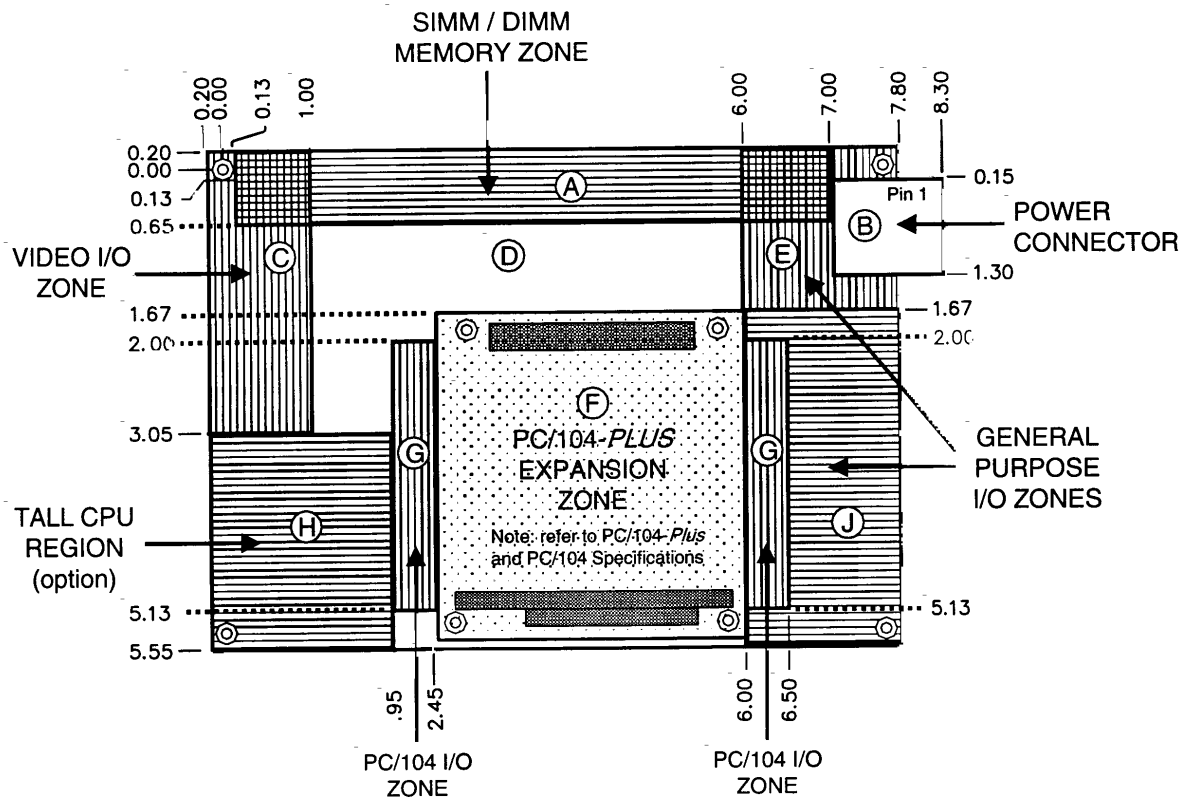
3.3.1 EBX DETAILED MECHANICAL DRAWINGS

Figure 1. EBX Detailed Mechanical Dimensions



sion 1.1 -- July 9, 1997

Figure 2. EBX Component Zones, Tall CPU Option



ease Version 1.1 -- July 9, 1997

4 DETAILED SYSTEM DESCRIPTION

This system has a system configuration based on the ISA architecture. Check the I/O and the Memory map in this chapter.

4.1 Power Requirements

The power is connected through the ISA Bus connector, or the PC/104 Power connector, or the separate power connector on the board. The supply uses only +5V and ground connection. For backplane supply and the Flash BIOS operation, the user has to connect the 12V (only for LCD port).

Warning: Make sure that the power plug is wired correctly before supplying power to the board! A built-in diode protects the board against reverse polarity.

Tolerance of 5 V supply: 5Volt ± 5%; Power-fail signal starts at ± 10% of 5 volt nominal and generates a reset status for the MICROSPACE PC.

ATTENTION: With the harddisk connected to the IDE 44pin interface, the power equirement is high. The peak current must be enough to spin up the HD-motor. The typical spin-up current of the harddisk is 0.8 - 1.5Amp at 5V. Too little current will drop the voltage under 5 volts for a short time. Due to this undervoltage, the system or the harddisk stops or falters. The VGA could also be "snowy".

The precise power requirements of the MICROSPACE PCC-P5 depends on a number of factors, including what functions are present on the board and what peripherals are connected to the board's I/0 ports. For example, AT-keyboards draw their power from the keyboard connector on the MICROSPACE PCC-P5 board, and therefore add keyboard current to the total power drawn by the board from its power supply.

CPU:	Clock:	Memory:	no Harddisk	HD-500MByte:	HD-PowerUp:
P5-166	166MHz	32 MByte	2.3 A	2.8 A	3 A
Suspended		32 Mbyte	1.0 A		
HD start current:		ST-9096A Seagate 2,5" 80 MBytes		ca. 0.8 Amp.	
AT-keyboard:				ca. 10 mA	

4.2 CPUs, Boards and RAMs

4.2.1 CPUs of this MICROSPACE Product

Proposed: Standard: INTEL P55 166 or 266MHz (single 1.8V switched)

The CPU Type must be defined by begin of the order. The CPU may not be changed after the assembly.

4.3 Interface

4.3.1 PS/2-Keyboard

Standard PS2-Keyboard , also available on the utilityconnector

4.3.2 PS/2-Mouse Interface

Standard PS/2 conector , also available on the utilityconnector

4.3.3 Line Printer Port LPT1

A standard bi-directional LPT port is integrated into the MICROSPACE PC, with DMA7 support.

Further information about these signals is available in numerous publications, including the IBM technical reference manuals for the PC and AT computers and from some other reference documents.

The current is: IOH = 12mA IOL = 24mA

The SMC 37C672 may be programmed in the BIOS Setup.

4.3.4 Serial Ports COM1-COM4 RS232C

Select the RS232C Interface with:

J108: 2-3 Select COM3 = RS232C
J109: 2-3 Select COM4 = RS232C

J115: 2-3 Select COM3 = IRQ10 (1-2 = IRQ4 *)
J114: 2-3 Select COM4 = IRQ11 (1-2 = IRQ3 *)

*) if IRQ3/4 used, the driver must handle the shared IRQ3/4 with the COM1/2 !

The serial channels are fully compatible with 16C550 UARTS. COM1 is the primary serial port, and is supported by the board's ROM-BIOS as the PC-DOS 'COM1' device. The secondary serial port is COM2; it is supported as the 'COM2' device.

Standard: COM 1/2: 16C550: 2 x 16C550 with 16 Byte FIFO

Serial Port Connectors - COM1, 2 generally

Pin	Signal Name	Function	in/out	DB25 Pin	DB9 Pin
1	CD	Data Carrier Detect	in	8	1
2	DSR	Data Set Ready	in	6	6
3	RXD	Receive Data	in	3	2
4	RTS	Request To Send	out	4	7
5	TXD	Transmit Data	out	2	3
6	CTS	Clear to Send	in	5	8
7	DTR	Data Terminal Ready	out	20	4
8	RI	Ring Indicator	in	22	9
9	GND	Signal Ground		7	5

The serial port signals are compatible with the RS232C specifications.

4.3.5 Serial Ports COM3-COM4 RS422

Select the RS422 Interface with:

J108: 1-2 Select COM3 = RS422
J109: 1-2 Select COM4 = RS422

J115: 2-3 Select COM3 = IRQ10 (1-2 = IRQ4 *)
J114: 2-3 Select COM4 = IRQ11 (1-2 = IRQ3 *)

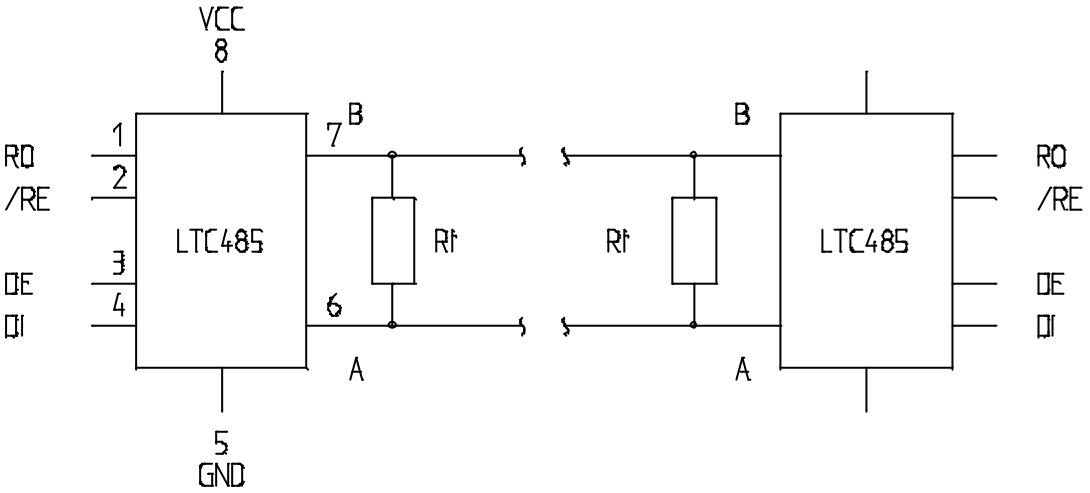
*) if IRQ3/4 used, the driver must handle the shared IRQ3/4 with the COM1/2 !

On the RS422/485-port the signals of UART have the following functions:

Signal of UART	Function of RS485
RxDx	RO (Receive Line)
RTSx	RE/ (Receiver enable) 0 = Enable, 1 = Disable
DTRx	DE (Transceiver enable) 1=Enable, 0= Tristate
TxDx	DI (Transmission line)

x stands for 3 or 4 depending of the COM3 or COM4 interface.

Typical Application



Summary

If the UART should transmit, then set the Bit RTS in the modem control register to '0' and the DTR to '1' befor the databyte is sent to the transmit register.

If the UART should receive, wait on the receive buffer full flag as usual. Nothing special must be done, since the RS422/485 receiver is always enabled.

FUNCTION TABLES
ADM489 Transmitting

INPUTS			Line Condition	OUTPUTS	
/RE = RTSx	TE=DTRx	DI		B	A
0	1	1	no fault	0	1
0	1	0	no fault	1	0
0	0	X	X	Z	Z
0	1	X	Fault	Z	Z

ADM489 Receiving

INPUTS			OUTPUTS
/RE = RTSx	TE=DTRx	A - B	R
0	1	≥ + 0.2V	1
0	1	≤ - 0.2V	0

4.3.6 Serial Ports COM3-COM4 RS485

Select the RS422 Interface with:

J108: 1-2 Select COM3 = RS485
J109: 1-2 Select COM4 = RS485

J115: 2-3 Select COM3 = IRQ10 (1-2 = IRQ4 *)
J114: 2-3 Select COM4 = IRQ11 (1-2 = IRQ3 *)

*) if IRQ3/4 used, the driver must handle the shared IRQ3/4 with the COM1/2 !

On the RS422/485-port the signals of UART have the following functions:

Signal of UART	Function of RS485
RxDx	RO (Receive Line)
RTSx	RE/ (Receiver enable) 0 = Enable, 1 = Disable
DTRx	DE (Transceiver enable) 1=Enable, 0= Tristate
TxDx	DI (Transmission line)

x stands for 3 or 4 depending of the COM3 or COM4 interface.

The RS485 is the same interface as the RS422 with the differenc, that several receiver/transmitter are in a bus line connected. In this case, the driver must have an access protocol, that make it shure, that only one transmitter at one time is enabled.

4.3.7 Driver for COM3-COM4 RS422/485

In the case of RS422 /485 a driver must handle the RTS and DTR control signall, to prevent of buscollisions and to become a proper bus access. This driver s must be programmed operating system dependent and/or application dependent.

The modemcontrol register is defined as follow:

Adress COM3: 3Ech
Adress COM4: 2ECh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserved	Reserved	reserved	reserved	OUT2	OUT1	RTS	DTR

Examples:

To Enable Rx and Tx

```
Mov     dx,3Ech         ; Select register 3Ech for COM3
Mov     al, 02          ; Enable Receiver and Transmitter
Out     dx,al           ; Set the value
```

To Disable Rx and Tx

```
Mov     dx,3Ech         ; Select register 3Ech for COM3
Mov     al, 01          ; Enable Receiver and Transmitter
Out     dx,al           ; Set the value
```

To Enable Rx and Disable Tx (for RS485)

```
Mov     dx,3Ech         ; Select register 3Ech for COM3
Mov     al, 00          ; Enable Receiver and Disable Transmitter
Out     dx,al           ; Set the value
```

For COM use the adress 2Ech.

4.3.8 Floppy disk interface

The onboard floppy disk controller and ROM-BIOS support one or two floppy disk drives in any of the standard PC-DOS and MS-DOS formats shown in the table . 2.88mb floppy are not supported.

4.3.8.1 Supported floppy formats

Capacity	Drive size	Tracks	Data rate	DOS version
1.2 MB	5-1/4"	80	500 KHz	3.0 - 6.22
720 K	3-1/2"	80	250 KHz	3.2 - 6.22
1.44 M	3-1/2"	80	500 KHz	3.3 - 6.22

4.3.8.2 Floppy interface connector

We support only CMOS drives. That means that the termination resistors are 1 kOhm. 5 1/4“-drives are not recommended (TTL interface).

The 34 pin Connector: Ribbon 1,27mm IDT dual row terminal with 2.54mm grid
The 26 pin Connector: Optional: FFC/FPC 0.3mm thick 1.0mm (0.039") pitch (MOLEX 52030 Serie) on the rear side mounted

Floppy Disk Interface Connector

FD34: Pin	FD26: Pin	Signal Name	Function	in/out
2	---	-RPM/-RWC	Speed/Precomp (option)	out
4	---	(Not used)		
6	---	(Not used)		
8	2	-IDX	Index Pulse	in
10	---	-MO1	Motor On 1	out
12	4	-DS2	Drive Select 2	out
14	---	-DS1	Drive Select 1	out
16	10	-M02	Motor On 2	out
18	12	-DIRC	Direction Select	out
20	14	-STEP	Step	out
22	16	-WD	Write Data	out
24	18	-WE	Write Enable	out
26	20	-TRKO	Track 0	in
28	22	-WP	Write Protect	in
30	24	-RDD	Read Data	in
32	26	-HS	Head Select	out
34	6	-DCHG	Disk Change	in
1-33	25,23,21,19,17	GND	Signal grounds	
	1,3,5	VCC	+5 Volt	

4.3.9 Speaker interface

One of the board's CPU device provides the logic for a PC compatible speaker port. The speaker logic signal is buffered by a transistor amplifier, and provides approximately 0.1 watt of audio power to an external 8 ohm speaker. The speaker must be connected to VCC (and not to Ground).

We propose to use a serial capacitor of 1uF with the speaker to eliminate any DC-current to protect the speaker himself from overheating.

4.4 Controllers

4.4.1 Interrupt Controllers

An 8259A compatible interrupt controller, within the chipset device, provides seven prioritized interrupt levels. Of these, several are normally associated with the board's onboard device interfaces and controllers, and several are available on the AT expansion bus.

Interrupt	Sources	onboard used
IRQ0	ROM-BIOS clock tick function, from timer 0	yes
IRQ1	Keyboard controller output buffer full	yes
IRQ2	Used for cascade 2. 8259	yes
IRQ3	COM2	yes
IRQ4	COM1	yes
IRQ5	Free for user	no
IRQ6	Floppy controller	yes
IRQ7	LPT1 parallel printer	yes
IRQ8	Battery backed clock, alarm function of the RTC	yes
IRQ9	Free for user	no
IRQ10	Free for user, COM3/4	yes
IRQ11	Free for user, COM3/4	no
IRQ12	PS/2 mouse	yes
IRQ13	Math coprocessor	yes
IRQ14	Harddisk IDE / SCSI	yes
IRQ15	Free for user (USB)	no

4.5 Timers and Counters

4.5.1 Programmable Timers

An 8253 compatible timer/counter device is also included in the board's ASIC device. This device is utilized in precisely the same manner as in a standard AT implementation. Each channel of the 8253 is driven by a 1.190 MHz clock, derived from a 14.318 MHz oscillator, which can be internally divided down to provide a variety of frequencies.

Timer 2 can also be used as a general purpose timer if the speaker function is not required.

Timer Assignment

Timer	Function
0	ROM-BIOS clock tick (18.2 Hz)
1	DRAM refresh request timing (15 µS)
2	Speaker tone generation time base

4.5.2 Battery backed clock (RTC)

An AT compatible date/time clock is located within the chipset. The device also contains a CMOS static RAM, compatible with that in standard ATs. System configuration data is normally stored in the clock chip's CMOS RAM in a manner consistent with the convention used in other AT compatible computers.

One unique feature of the board's battery-backed clock device is that it contains the backup battery directly on the board. The battery is rated for a minimum of 6 years of clock and internal CMOS RAM backup under conditions of no power to the board. The battery is removable for easy exchange with a new type or replacement when the battery is exhausted.

The battery is DIGITAL-LOGIC replacement part: PCC-P5L 3V-BAT. The battery-backed clock can be set by using the DIGITAL-LOGIC AG SETUP at boot-time.

Addresses:	70h	=	Index register
	71h	=	Data transfer register
RTC-Address MAP:	00 - 0F		RTC (Real time clock)
	10 - 3F		BIOS setup (Standard)
	40 - 7F		Extended BIOS or SuperState BIOS setup

The chipset consumes the following currents:

Typical battery current at 25°C : 0.2 µA Lifetime around 10 years at 25°C

4.5.3 Watchdog

The watchdog timer detects a system crash and performs a hardware reset. After powering-up, the watchdog is always disabled as the BIOS does not send strobes to the watchdog. In case the user wants to take advantage of the watchdog, the application must produce a strobe at least every 800 ms. If no strobe occures within the 800 ms, the watchdog resets the system.

To program the watchdog in user applications DIGITAL-LOGIC AG has implemented a special BIOS extension in Interrupt 60h (function: EBh).

Calling this function by setting a 1 in the AL- Register, turns on the watchdog and performs a strobe. Calling the same function with a 0 in the AL-Register, turns off the watchdog.

The following part has to be implemented in the users application:

Watchdog on: The application has to call interrupt 15h function EBh and set a 1 into the AL-register at least every 800 ms.

Watchdog off: The application has to call interrupt 15h function EBh and set a 0 into the AL-register within 800 ms after the last strobe has been sent while the watchdog was still in function (if the watchdog is not turned off in time, it will reset the system again!).

4.6 BIOS

4.6.1 ROM-BIOS Sockets

An EPROM socket with 8 Bit wide data access normally contains the board's AT compatible ROM-BIOS. The socket takes any of a 27C010 to 29F010 EPROM (or equivalent) device. The board's wait-state control logic automatically inserts four memory wait states in all CPU accesses to this socket. The ROM-BIOS sockets occupies the memory area from C0000H through FFFFFh; however, the board's ASIC logic reserves the entire area from C0000h through FFFFFh for onboard devices, so this area is already usable for ROM-DOS and BIOS expansion modules. Consult the appropriate address map for the MICROSPACE PCC-P5 ROM-BIOS sockets.

4.6.1.1 Standard BIOS FLASH 29F030

DEVICE:	29C020 PLCC32	with 90ns access time
MAP:	E0000 - FFFFFh	Chipset BIOS from AMI including the SCSI BIOS

4.6.1.2 VGA BIOS FLASH 29F010

DEVICE:	29F010 PLCC32	with 90ns access time (with 29F020 are 4 BIOS Segments with 64k jumper selectable)
Segment-MAP:	C0000 - CBFFF	VGA BIOS 48k

4.6.2 EEPROM Memory for Setup

The EEPROM is used for setup and configuration data, stored as an alternative to the CMOS-RTC. Optionally, the EEPROM setup driver may update the CMOS RTC, if the battery is running down, the checksum error would appear and stop the system. The capacity of the EEPROM is 2048 Bytes.

Organisation of the 2048Byte EEPROMs:

Address MAP:	Function:
0000h	**CMOS-Setup valid (01=valid)**
0001h	**Keymatrix-Setup valid (01=valid)**
0003h	**Flag for DLAG-Message (FF=no message)**
0010h-007Fh	**Copy of CMOS-Setup data**
0080h-00FFh	**reserved for AUX-CMOS-Setup**
0100h-010Fh	**Serial-Number**
0110h-0113h	Production date (year/day/month)
0114h-0117h	1. Service date (year/day/month)
0118h-011Bh	2. Service date (year/day/month)
011Ch-011Fh	3. Service date (year/day/month)
0120h-0122h	Booterrors (Autoincremented if any booterror occurs)
0123h-0125h	Setup Entries (Autoincremented on every Setup entry)
0126h-0128h	Low Battery (Autoincremented everytime the battery is low, EEPROM -> CMOS)
0129h-012Bh	Startup (Autoincremented on every poweron start)
0130h	Number of 512k SRAM
0131h	Number of 512k Flash
0132h/0133h	BIOS Version (V1.4 => [0132h]:= 4, [0133h]:=1)
0134h/0135h	BOARD Version (V1.5 => [0124h]:=5, [0125h]:=1)
0136h	BOARD TYPE (‘M’=PC/104, ‘E’=Euro, ‘W’=MSWS, ‘S’=Slot, ‘C’=Custom)
0137h	CPU TYPE (01h=ELAN300/310, 02h=ELAN400, 03h=486SLC, 04h=486DX, 05h=P5).
0200h-03FFh	**Keymatrix-Setup data**
0200h-027Fh	Keymatrix Table
0400h-07FFh	**Free for Customer’s use**

4.6.3 BIOS CMOS Setup

If wrong setups are memorized in the CMOS-RAM, the default values will be loaded after resetting the RTC/CMOS-RAM with the CMOS-RESET jumper. If the battery is down, it is always possible to start the system with the default values from the BIOS.

The following entries may be made:

Date:	The current Real Date of the RTC	
Time:	The current Real Time of the RTC	
Drive A or B:	none	= no drive present, FLASHDisk enabled (if device is loaded)
	360k	= 5,25" low density drive or FLASHDISK
	1,2 MB	= 5,25" high density drive or SRAMDISK
	720 K	= 3,5" low density drive
	1,44 MB	= 3,5" high density drive (Default for A:)
	The A: Drive is the bootable drive.	
Display type:	CRT:	for Mono CRT's but no LCD operating possible.
	40 x 25:	for Color CGA or LCD
	80 x 25:	for Color CGA or LCD (Default)
	VGA:	for VGA
Harddisk type:	defines which drive is connected	
	Type = 0 means no drive is present (Default)!	
	Drive Type 48 and 49 let you define a custom harddisk parameter.	
	PRESS the AUTODETECT FUNCTION to identify the HD!	

WARNING:

On the next Setup pages (switched with PgDn and PgUp) the values for special parameters are modifiable. Normally the parameters are set correctly by DIGITAL-LOGIC AG. Be very careful in modifying any parameter since the system could crash. Some parameters are dependent on the CPU type. The cache parameter is always available, for example. So, if you select too few wait states, the **system will not start** until you reset the CMOS-RAM using the **RTC-Reset jumper**, but the default values are reloaded. If you are not familiar with these parameters, do not change anything.

4.7 CMOS RAM Map

Systems based on the industry-standard specification include a battery backed Real Time Clock chip. This clock contains at least 64 bytes of non-volatile RAM. The system BIOS uses this area to store information including system configuration and initialization parameters, system diagnostics, and the time and date. This information remains intact even when the system is powered down.

The BIOS supports 128 bytes of CMOS RAM. This information is accessible through I/O ports 70h and 71h. CMOS RAM can be divided into several segments:

- Locations 00h - 0Fh contain real time clock (RTC) and status information
- Locations 10h - 2Fh contain system configuration data
- Locations 30h - 3Fh contain System BIOS-specific configuration data as well as chipset-specific information
- Locations 40h - 7Fh contain chipset-specific information as well as power management configuration parameters

The following table provides a summary of how these areas may be further divided.

Beginning	Ending	Checksum	Description
00h	0Fh	No	RTC and Checksum
10h	2Dh	Yes	System Configuration
2Eh	2Fh	No	Checksum Value of 10h - 2Dh
30h	33h	No	Standard CMOS
34h	3Fh	No	Standard CMOS - SystemSoft Reserved
40h	5Bh	Yes	Extended CMOS - Chipset Specific
5Ch	5Dh	No	Checksum Value of 40h - 5Bh
5Eh	6Eh	No	Extended CMOS - Chipset Specific
6Fh	7Dh	Yes	Extended CMOS - Power Management
7Eh	7Fh	No	Checksum Value of 6Fh - 7Dh

Location	Description
00h	Time of day (seconds) specified in BCD
01h	Alarm (seconds) specified in BCD
02h	Time of Day (minutes) specified in BCD
03h	Alarm (minutes) specified in BCD
04h	Time of Day (hours) specified in BCD
05h	Alarm (hours) specified in BCD
06h	Day of week specified in BCD
07h	Day of month specified in BCD
08h	Month specified in BCD
09h	Year specified in BCD
0Ah	Status Register A Bit 7 = Update in progress Bits 6-4 = Time based frequency divider Bits 3-0 = Rate selection bits that define the periodic interrupt rate and output frequency.
0Bh	Status Register B Bit 7 = Run/Halt 0 Run 1 Halt Bit 6 = Periodic Timer 0 Disable 1 Enable Bit 5 = Alarm Interrupt 0 Disable 1 Enable Bit 4 = Update Ended Interrupt 0 Disable 1 Enable Bit 3 = Square Wave Interrupt 0 Disable 1 Enable Bit 2 = Calendar Format 0 BCD 1 Binary Bit 1 = Time Format 0 12-Hour 1 24-Hour Bit 0 = Daylight Savings Time 0 Disable 1 Enable
0Ch	Status Register C Bit 7 = Interrupt Flag Bit 6 = Periodic Interrupt Flag Bit 5 = Alarm Interrupt Flag Bit 4 = Update Interrupt Flag Bits 3-0 = Reserved
0Dh	Status Register D Bit 7 = Real Time Clock 0 Lost Power 1 Power

Continued...

CMOS Map Continued...

Location	Description
0Eh	CMOS Location for Bad CMOS and Checksum Flags bit 7 = Flag for CMOS Lost Power 0 = Power OK 1 = Lost Power bit 6 = Flag for CMOS checksum bad 0 = Checksum is valid 1 = Checksum is bad
0Fh	Shutdown Code
10h	Diskette Drives bits 7-4 = Diskette Drive A 0000 = Not installed 0001 = Drive A = 360 K 0010 = Drive A = 1.2 MB 0011 = Drive A = 720 K 0100 = Drive A = 1.44 MB 0101 = Drive A = 2.88 MB bits 3-0 = Diskette Drive B 0000 = Not installed 0001 = Drive B = 360 K 0010 = Drive B = 1.2 MB 0011 = Drive B = 720 K 0100 = Drive B = 1.44 MB 0101 = Drive B = 2.88 MB
11h	Reserved
12h	Fixed (Hard) Drives bits 7-4 = Hard Drive 0, AT Type 0000 = Not installed 0001-1110 Types 1 - 14 1111 = Extended drive types 16-44. See location 19h. bits 3-0 = Hard Drive 1, AT Type 0000 = Not installed 0001-1110 Types 1 - 14 1111 = Extended drive types 16-44. See location 2Ah. See the *Fixed Drive Type Parameters Table* in Chapter 2 for information on drive types 16-44.
13h	Reserved

Continued...

CMOS Map Continued...

Location	Description
14h	Equipment bits 7-6 = Number of Diskette Drives 00 = One diskette drive 01 = Two diskette drives 10, 11 = Reserved bits 5-4 = Primary Display Type 00 = Adapter with option ROM 01 = CGA in 40 column mode 10 = CGA in 80 column mode 11 = Monochrome bits 3-2 = Reserved bit 1 = Math Coprocessor Presence 0 = Not installed 1 = Installed bit 0 = Bootable Diskette Drive 0 = Not installed 1 = Installed
15h	Base Memory Size (in KB) - Low Byte
16h	Base Memory Size (in KB) - High Byte
17h	Extended Memory Size in (KB) - Low Byte
18h	Extended Memory Size (in KB) - High Byte
19h	Extended Drive Type - Hard Drive 0 See the *Fixed Drive Type Parameters Table* in Chapter 2 for information on drive types 16-44.
1Ah	Extended Drive Type - Hard Drive 1 See the *Fixed Drive Type Parameters Table* in Chapter 2 for information on drive types 16-44.
1Bh	Custom and Fixed (Hard) Drive Flags bits 7-6 = Reserved bit 5 = Internal Floppy Diskette Controller 0 = Disabled 1 = Enabled bit 4 = Internal IDE Controller 0 = Disabled 1 = Enabled bit 3 = Hard Drive 0 Custom Flag 0 = Disable 1 = Enabled bit 2 = Hard Drive 0 IDE Flag 0 = Disable 1 = Enabled bit 1 = Hard Drive 1 Custom Flag 0 = Disable 1 = Enabled bit 0 = Hard Drive 1 IDE Flag 0 = Disable 1 = Enabled

Continued...

CMOS Map Continued...

Location	Description
1Ch	Reserved
1Dh	EMS Memory Size Low Byte
1Eh	EMS Memory Size High Byte
1Fh - 24h	Custom Drive Table 0 These 6 bytes (48 bits) contain the following data: Cylinders 10 bits range 0-1023 Landing Zone 10 bits range 0-1023 Write Precomp 10 bits range 0-1023 Heads 8 bits range 0-15 Sectors/Track 8 bits range 0-254
1Fh	Byte 0 bits 7-0 = Lower 8 Bits of Cylinders
20h	Byte 1 bits 7-2 = Lower 6 Bits of Landing Zone bits 1-0 = Upper 2 Bits of Cylinders
21h	Byte 2 bits 7-4 = Lower 4 Bits of Write Precompensation bits 3-0 = Upper 4 Bits of Landing Zone
22h	Byte 3 bits 7-6 = Reserved bits 5-0 = Upper 6 Bits of Write Precompensation
23h	Byte 4 bits 7-0 = Number of Heads
24h	Byte 5 bits 7-0 = Sectors Per Track
25h - 2Ah	Custom Drive Table 1 These 6 bytes (48 bits) contain the following data: Cylinders 10 bits range 0-1023 Landing Zone 10 bits range 0-1023 Write Precomp 10 bits range 0-1023 Heads 8 bits range 0-15 Sectors/Track 8 bits range 0-254
25h	Byte 0 bits 7-0 = Lower 8 Bits of Cylinders
26h	Byte 1 bits 7-2 = Lower 6 Bits of Landing Zone bits 1-0 = Upper 2 Bits of Cylinders
27h	Byte 2 bits 7-4 = Lower 4 Bits of Write Precompensation bits 3-0 = Upper 4 Bits of Landing Zone

Continued...

CMOS Map Continued...

Location	Description
28h	Byte 3 bits 7-6 = Reserved bits 5-0 = Upper 6 Bits of Write Precompensation
29h	Byte 4 bits 7-0 = Number of Heads
2Ah	Byte 5 bits 7-0 = Sectors Per Track
2Bh	Boot Password bit 7 = Enable/Disable Password 0 = Disable Password 1 = Enable Password bits 6-0 = Calculated Password
2Ch	SCU Password bit 7 = Enable/Disable Password 0 = Disable Password 1 = Enable Password bits 6-0 = Calculated Password
2Dh	Reserved
2Eh	High Byte of Checksum - Locations 10h to 2Dh
2Fh	Low Byte of Checksum - Locations 10h to 2Dh
30h	Extended RAM (KB) detected by POST - Low Byte
31h	Extended RAM (KB) detected by POST - High Byte
32h	BCD Value for Century
33h	Base Memory Installed bit 7 = Flag for Memory Size 0 = 640KB 1 = 512KB bits 6-0 = Reserved
34h	Minor CPU Revision Differentiates CPUs within a CPU type (i.e., 486SX vs 486 DX, vs 486 DX/2). This is crucial for correctly determining CPU input clock frequency. During a power on reset, Reg DL holds minor CPU revision.
35h	Major CPU Revision Differentiates between different CPUs (i.e., 386, 486, Pentium). This is crucial for correctly determining CPU input clock frequency. During a power on reset, Reg DH holds major CPU revision.
36h	Hotkey Usage bits 7-6 = Reserved bit 5 = Semaphore for Completed POST bit 4 = Semaphore for 0 Volt POST (not currently used) bit 3 = Semaphore for already in SCU menu bit 2 = Semaphore for already in PM menu bit 1 = Semaphore for SCU menu call pending bit 0 = Semaphore for PM menu call pending
40h-7Fh	Definitions for these locations vary depending on the chipset.

4.7.1 CMOS Setup Harddisk list

Harddisk parameter selection

Since the BIOS autodetects the harddisk type, no HD-drive parameter table is used.
Go into the BIOS-HD-Setup and press autodetect. The parameters are read out of the IDE harddisk and stored in the CMOS memory.

4.7.2 Harddisk PIO Modes

Block Mode (Multi-Sector) Transfer: Block mode boots IDE drive performance by increasing the amount of data transferred.

No Block Mode: 512 Byte per interrupt
Block Mode: up to 64 kByte per interrupt

LBA Mode: LBA (logical block addressing) is a new method of addressing data on a disk drive. In the standard ST506 (MFM) ISA hard disk, data is accessed via a cylinder - head - sector format.

LBA Mode disabled: max. 528 Mbyte per Disk

LBA Mode enabled: max. 8 Gbyte per Disk

Attention:
The BIOS enables the LBA Mode only, if the harddisk was formatted on a system with enabled LBA. If the drive (capacity > 528MB) is formated on a system with disabled LBA, the AMI BIOS will never enable the LBA mode !

The maximum parameters are:
1024 Cyl. , 16 heads, 63 Sec/Track

32 Bit Transfer: Some operating system can handle two 16bit word as one 32bit access. This accelerates the IDE transfer.

Advanced PIO Modes:

PIO-Mode:		Timing:	Transferspeed:	Remarks:	
IDE	0		600ns	2 Mbyte/sec	Slowest I/O
IDE	1		383ns	5.5Mbyte/sec	Standard I/O
EIDE	2		240ns	8.3Mbyte/sec	Fast I/O, Mem.
EIDE	3		180ns	11,3Mbyte/sec	IORDY Protocol
EIDE	4		120ns	16,6Mbyte/sec	IORDY Protocol
EIDE	DMA 1		160ns	13,3Mbyte/sec	DRQ, ATA-2

Begin always with the PIO-Mode 0
in the manual mode (not autodetect) to test a new drive or if you becomes trouble in the automatic mode. The autodetect mode selects with some drives wrong PIO modes.

4.7.3 EEPROM saved CMOS setup

The EEPROM has different functions, as listed below:

- Backup of the CMOS-Setup values.
- Storing system informations like: version, production date, customisation of the board, CPU type.
- Storing user/application values.

The EEPROM will be updated automatically after exiting the BIOS setup menu. The system will operate also without any CMOS battery. While booting up, the CMOS is automatically updated with the EEPROM values.

Press the Esc-key while powering on the system before the video shows the BIOS message and the CMOS will **not** be updated.

This would be helpful, if wrong parameters are stored in the EEPROM and the setup of the BIOS does not start.

If the system hangs or a problem appears, the following steps must be performed:

1. Reset the CMOS-Setup (use the jumper to reset or disconnect the battery for at least 10 minutes).
2. Press Esc until the system starts up.
3. Enter the BIOS Setup:
   a) load DEFAULT values
   b) enter the settings for the environment
   c) exit the setup
4. Restart the system.

- The user may access the EEPROM through the INT15 special functions. Refer to the chapter SFI functions.
- The system information are read onyl information. To read, use the SFI functions.

4.7.4 BIOS Download Function

The BIOS is stored into a flash device and may be updated with a new version onbaord. To do this , you need an BIOS Updatedisk including the newest BIOS version.

1. Insert the BIOS Update disk into floppy A:, the BIOS-image is the AMIBOOT.ROM file.
2. Restart the system.
3. Press immediatly CONTROL + HOME before the BIOS message appears on the screen
4. The BIOS will be read from the disk and stored into the flashdevice.
5. Restart the system again and check the BIOS version.

4.7.5 VGA BIOS Download Function

The BIOS for the VGA must be downloaded, before an LCD is connected. This could be also a new LCD-Display, which needs a corresponding VGA BIOS.

The following points must be checked before downloading a BIOS:

- Select the Shadow option in the BIOS for BIOS and VGA (if this option is available).
- Disable the EMM386 or other memory managers in the CONFIG.SYS of your bootdisk.
- Make sure, that the DOWN_000.EXE programm and the BIOS to download are on the same path and directory!

How to download a VGABIOS:

1. Restart the system with SHADOW enabled and no EMM386 loaded.
2. Check if you find the DOWN_000.EXE and also the *.000 files on your disk to download.
3. Refer to the VGABIOS.DOC for more information about the VGABIOS files.
4. Insert the floppydisk with the program DOWN.EXE and all VGA-Drivers.
5. Start DOWN_000.EXE.
6. Check if the DOWN_000 program has identified the product and the shadow correctly.
7. Select the function PROGRAMM VGA-BIOS.
8. Select the VGA BIOS out of the proposed file list (UP/DOWN arrows) and press ENTER.
9. Check if the new VGA-Header is displayed on the VGA-INFO screen.

If the download does not work: - Check if no EMM386 is loaded.
- Check if no peripheral card is in the system, which occupies the same memory range. Disconnect this card.

If the screen flickers or is misaligned after reboot: - The previously loaded VGA BIOS is not corresponding 100% or works only on the LCD properly.

If the screen is dark after the reboot of the system:
- A new system BIOS must be programmed. Ask DIGITAL-LOGIC AG for the binary file.

4.8 Memory

4.8.1 Onboard DRAM Memory

Speed:	60ns
Size:	soldered 64bit chips
Bits:	64 Bit
Capacity:	32 MBytes, 64 MBytes
Bank:	allways two banks must be equipped

4.8.2 System Memory Map

The CPU used as a central processing unit on the MICROSPACE PC has a memory address space which is defined by 32 address bits. Therefore, it can address 1GByte of memory. The memory address MAP is as follows:

CPU P5

Address	Size	Function / Comments
000000 - 09FFFFh	640 KBytes	Onboard DRAM for DOS applications
0A0000 - 0BFFFFh	128 KBytes	CGA, EGA, LCD Video RAM 128kB
0C0000 - 0CFFFFh	64 KBytes	Reserved for expansion bus ROMs: C000 - CC00 for VGA BIOS
0D0000 - 0DFFFFh	64 KBytes	Free or DiskOnChip DOC2000 Module from MSystems
0E0000 - 0EFFFFh	64 KBytes	PCI-System BIOS incl. SCSI-BIOS
0F0000 - 0FFFFFh	64 KBytes	PCI-System BIOS incl. SCSI-BIOS
100000 - 1FFFFFh	1 MByte	DRAM for extended onboard memory
200000 - FFFFFFh	14 MBytes	DRAM for extended onboard memory

Attention: Only the D-Segment is free for the user, to use EMM386 or HIMEM programs
If the DOC2000 module is used, the D-Segment is also used, and no EMM386 may be used. For the HIMEM use the EMM386 with the option NOEMS !

Urgent: The AMI POWER BIOS for the P5 has a size of 128k and this value is definitiv, that means may be not decreased !

4.8.3 System I/O map

The following table shows the detailed listing of the I/O port assignments used in the MICROSPACE board:

I/O Address	Read/Write Status	Description
0000h	R / W	DMA channel 0 address byte 0 (low), then byte 1
0001h	R / W	DMA channel 0 word count byte 0 (low), then byte 1
0002h	R / W	DMA channel 1 address byte 0 (low), then byte 1
0003h	R / W	DMA channel 1 word count byte 0 (low), then byte 1
0004h	R / W	DMA channel 2 address byte 0 (low), then byte 1
0005h	R / W	DMA channel 2 word count byte 0 (low), then byte 1
0006h	R / W	DMA channel 3 address byte 0 (low), then byte 1
0007h	R / W	DMA channel 3 word count byte 0 (low), then byte 1
0008h	R	DMA channel 0-3 status register bit 7 = 1 Channel 3 request bit 6 = 1 Channel 2 request bit 5 = 1 Channel 1 request bit 4 = 1 Channel 0 request bit 3 = 1 Terminal count on channel 3 bit 2 = 1 Terminal count on channel 2 bit 1 = 1 Terminal count on channel 1 bit 0 = 1 Terminal count on channel 0

Continued...

I/O Address	Read/Write Status	Description
0008h	W	DMA channel 0-3 command register bit 7 = DACK sense active high/low 0 low 1 high bit 6 = DREQ sense active high/low 0 low 1 high bit 5 = Write selection 0 Late write selection 1 Extended write selection bit 4 = Priority 0 Fixed 1 Rotating bit 3 = Timing 0 Normal 1 Rotating bit 2 = Controller enable/disable 0 Enable 1 Disable bit 1 = Memory-to-memory enable/disable 0 Disable 1 Enable bit 0 = Reserved
0009h	W	DMA write request register
000Ah	R / W	DMA channel 0-3 mask register bits 7-3 = Reserved bit 2 = 0 Clear bit 1 Set bit bits 1-0 = Channel Select 00 Channel 0 01 Channel 1 10 Channel 2 11 Channel 3
00Bh	W	DMA channel 0-3 mode register bits 7-6 = 00 Demand mode 01 Single mode 10 Block mode 11 Cascade mode bit 5 = 0 Address increment select 1 Address decrement select bit 4 = 0 Disable auto initialization 1 Enable auto initialization bits 3-2 = Operation type 00 Verify operation 01 Write to memory 10 Read from memory 11 Reserved bits 1-0 = Channel select 00 Channel 0 01 Channel 1 10 Channel 2 11 Channel 3

Continued...

I/O Address	Read/Write Status	Description
000Ch	W	DMA clear byte pointer flip/flop
000Dh	R	DMA read temporary register
000Dh	W	DMA master clear
000Eh	W	DMA clear mask register
000Fh	W	DMA write mask register
0020h	W	Programmable Interrupt Controller - Initialization Command Word 1 (ICW1) provided bit 4 = 1 bits 7-5 = 000 Used only in 8080 or 8085 mode bit 4 = 1 ICW1 is used bit 3 = 0 Edge triggered mode 1 Level triggered mode bit 2 = 0 Successive interrupt vectors separated by 8 bytes 1 Successive interrupt vectors separated by 4 bytes bit 1 = 0 Cascade mode 1 Single mode bit 0 = 0 ICW4 not needed 1 ICW4 needed
0021h	W	Used for ICW2, ICW3, or ICW4 in sequential order afterICW1 is written to port 0020h **ICW2** bits 7-3 = Address A0-A3 of base vector address for interrupt controller bits 2-0 = Reserved (should be 000) **ICW3** (for slave controller 00A1h) bits 7-3 = Reserved (should be 0000) bits 2-0 = 1 Slave ID **ICW4** bits 7-5 = Reserved (should be 000) bit 4 = 0 No special fully nested mode 1 Special fully nested mode bits 3-2 = Mode 00 Non buffered mode 01 Non buffered mode 10 Buffered mode/slave 11 Buffered mode/master bit 1 = 0 Normal EOI 1 Auto EOI bit 0 = 0 8085 mode 1 8080 / 8088 mode

Continued...

I/O Address	Read/Write Status	Description
0021h	R / W	PIC master interrupt mask register (OCW1) bit 7 = 0 Enable parallel printer interrupt bit 6 = 0 Enable diskette interrupt bit 5 = 0 Enable hard disk interrupt bit 4 = 0 Enable serial port 1 interrupt bit 3 = 0 Enable serial port 2 interrupt bit 2 = 0 Enable video interrupt bit 1 = 0 Enable kybd/pointing device/RTC interrupt bit 0 = 0 Enable interrupt timer
0021h	W	PIC OWC2 (if bits 4-3 = 0) bit 7 = Reserved bits 6-5 = 000 Rotate in automatic EOI mode (clear) 001 Nonspecific EOI 010 No operation 011 Specific EOI 100 Rotate in automatic EOI mode (set) 101 Rotate on nonspecific EOI command 110 Set priority command 111 Rotate on specific EOI command bits 4-3 = Reserved (should be 00) bits 2-0 = Interrupt request to which the command applies
0020h	R	PIC interrupt request and in-service registers programmed by OCW3 **Interrupt request register** bits 7-0 = 0 No active request for the corresponding interrupt line 1 Active request for the corresponding interrupt line **Interrupt in-service register** bits 7-0 = 0 Corresponding interrupt line not currently being serviced 1 Corresponding interrupt line is currently being serviced
0021h	W	PIC OCW3 (if bit 4 = 0, bit 3 = 1) bit 7 = Reserved (should 0) bits 6-5 = 00 No operation 01 No operation 10 Reset special mask 11 Set special mask bit 4 = Reserved (should be 0) bit = Reserved (should be 1) bit 2 = 0 No poll command 1 Poll command bits 1-0 = 00 No operation 01 Operation 10 Read interrupt request register on next read at port 0020 h 11 Read interrupt in-service register on next read at port 0020h

Continued...

I/O Address	Read/Write Status	Description
0022h	R / W	Chipsset Register Adress
0023h	R / W	Chipsset Register Data
0040h	R / W	Programmable Interrupt Time read/write counter 0, keyboard controller channel 0
0041h	R / W	Programmer Interrupt Timer channel 1
0042h	R / W	Programmable Interrupt Timer miscellaneous register channel 2
0043h	W	Programmable Interrupt Timer mode port - control word register for counters 0 and 2 bits 7-0 = Counter select 00 Counter 0 select 01 Counter 1 select 10 Counter 2 select bits 5-4 = 00 Counter latch command 01 R / W counter, bits 0-7 only 10 R / W counter, bits 8-15 only 11 R / W counter, bits 0-7 first, then bits 8-15 bits 3-1 = Select mode 000 Mode 0 001 Mode 1 programmable one shot x10 Mode 2 rate generator x11 Mode 3 square wave generator 100 Mode 4 software-triggered strobe 101 Mode 5 hardware-triggered strobe bit 0 = 0 Binary counter is 16 bits 1 Binary counter decimal (BCD) counter
0048h	R / W	Programmable interrupt timer
0060h	R	Keyboard controller data port or keyboard input buffer
0060h	W	Keyboard or keyboard controller data output buffer

Continued...

I/O Address	Read/Write Status	Description
0064h	R	Keyboard controller read status bit 7 = 0 No parity error 1 Parity error on keyboard transmission bit 6 = 0 No timeout 1 Received timeout bit 5 = 0 No timeout 1 Keyboard transmission timeout bit 4 = 0 Keyboard inhibited 1 Keyboard not inhibited bit 3 = 0 Data 1 Command bit 2 = System flag status bit 1 = 0 Input buffer empty 1 Input buffer full bit 0 = 0 Output buffer empty 1 Output buffer full
0064h	W	Keyboard controller input buffer
0070h	R	CMOS RAM index register port and NMI mask bit 7 = 1 NMI disabled bits 6-0 = 0 CMOS RAM index
0071h	R / W	CMOS RAM data register port
0080h	R / W	Temporary storage for additional page register
0080h	R	Manufacturing diagnostic port (this port can access POST checkpoints)
0081h	R / W	DMA channel 2 address byte 2
0082h	R / W	DMA channel 2 address byte 2
0083h	R / W	DMA channel 1 address byte 2
0084h	R / W	Extra DMA page register
0085h	R / W	Extra DMA page register
0086h	R / W	Extra DMA page register
0087h	R / W	DMA channel 0 address byte 2
0088h	R / W	Extra DMA page register
0089h	R / W	DMA channel 6 address byte 2
008Ah	R / W	DMA channel 7 address byte 2
008Bh	R / W	DMA channel 5 address byte 2
008Ch	R / W	Extra DMA page register
008Dh	R / W	Extra DMA page register
008Eh	R / W	Extra DMA page register
008Fh	R / W	DMA refresh page register

Continued...

I/O Address	Read/Write Status	Description
00A0h - 00A1h are reserved for the slave programmable interrupt controller. The bit definitions are identical to those of addresses 0020h - 0021h except where indicated.		
00A0h	R / W	Programmable interrupt controller 2
00A1h	R / W	Programmable interrupt controller 2 mask bit 7 = 0 Reserved bit 6 = 0 Enable hard disk interrupt bit 5 = 0 Enable coprocessor execution interrupt bit 4 = 0 Enable mouse interrupt bits 3-2 = 0 Reserved bit 1 = 0 Enable redirect cascade bit 0 = 0 Enable real time clock interrupt
00C0h	R / W	DMA channel 4 memory address bytes 1 and 0 (low)
00C2h	R / W	DMA channel 4 transfer count bytes 1 and 0 (low)
00C4h	R / W	DMA channel 5 memory address bytes 1 and 0 (low)
00C6h	R / W	DMA channel 5 transfer count bytes 1 and 0 (low)
00C8h	R / W	DMA channel 6 memory address bytes 1 and 0 (low)
00CAh	R / W	DMA channel 6 transfer count bytes 1 and 0 (low)
00CCh	R / W	DMA channel 7 memory address bytes 1 and 0 (low)
00CEh	R / W	DMA channel 7 transfer count bytes 1 and 0 (low)
00D0h	R	Status register for DMA channels 4-7 bit 7 = 1 Channel 7 request bit 6 = 1 Channel 6 request bit 5 = 1 Channel 5 request bit 4 = 1 Channel 4 request bit 3 = 1 Terminal count on channel 7 bit 2 = 1 Terminal count on channel 6 bit 1 = 1 Terminal count on channel 5 bit 0 = 1 Terminal count on channel 4
00D0h	W	Command register for DMA channels 4-7 bit 7 = 0 DACK sense active low 1 DACK sense active high bit 6 = 0 DREQ sense active low 1 DREQ sense active high bit 5 = 0 Late write selection 1 Extended write selection bit 4 = 0 Fixed Priority 1 Rotating Priority bit 3 = 0 Normal Timing 1 Rotating Timing bit 2 = 0 Enable controller 1 Disable controller bit 1 = 0 Disable memory-to-memory transfer 1 Enable memory-to-memory transfer bit 0 = Reserved

Continued...

I/O Address	Read/Write Status	Description
00D2h	W	Write request register for DMA channels 4-7
00D4h	W	Write single mask register bit for DMA channels 4-7 bits 7-3 = 0 Reserved bit 2 = 0 Clear mask bit, 1 Set mask bit bits 1-0 = Channel select 00 Channel 4 01 Channel 5 10 Channel 6 11 Channel 7
00D6h	W	Mode register for DMA channels 4-7 bits 7-6 = 00 Demand mode 01 Single mode 10 Block mode 11 Cascade mode bit 5 = 0 Address increment select 1 Address decrement select bit 4 = 0 Disable auto initialization 1 Enable auto initialization bits 3-2 = Operation type 00 Verify operation 01 Write to memory 10 Read from memory 11 Reserved bits 1-0 = Channel select 00 Channel 4 01 Channel 5 10 Channel 6 11 Channel 7
00D8h	W	Clear byte pointer flip/flop for DMA channels 4-7
00DAh	R	Read Temporary Register for DMA channels 4-7
00DAh	W	Master Clear for DMA channels 4-7
00DCh	W	Clear mask register for DMA channels 4-7
00DEh	W	Write mask register for DMA channels 4-7
00F0h	W	Math coprocessor clear busy latch
00F1h	W	Math coprocessor reset
00F2h - 00FFh	R / W	Math coprocessor
0140h – 014Fh	R / W	SCSI Controller if installed
I/O addresses 0170h - 0177h are reserved for use with a secondary hard drive. See addresses 01F0h - 01F7h for bit definitions.		
0170h	R / W	Data register for hard drive 1
0171h	R	Error register for hard drive 1
0171h	W	Precomposition register for hard drive 1
0172h	R / W	Sector count - hard drive 1

Continued...

I/O Address	Read/Write Status	Description
0173h	R / W	Sector number for hard disk 1
0174h	R / W	Number of cylinders (low byte) for hard drive 1
0175h	R / W	Number of cylinders (high byte) for hard drive 1
0716h	R / W	Drive/head register for hard drive 1
0177h	R	Status register for hard drive 1
0177h	W	Command register for hard drive 1
01F0h	R / W	Data register base port for hard drive 0
01F1h	R	Error register for hard drive 0 **Diagnostic mode** bits 7-3 = Reserved bits 2-0 = Errors 0001 No errors 0010 Controller error 0011 Sector buffer error 0100 ECC device error 0101 Control processor error **Operation mode** bit 7 = Block 0 Bad block 1 Block not bad bit 6 = Error 0 No error 1 Uncorrectable ECC error bit 5 = Reserved bit 4 = ID 0 ID located 1 ID not located bit 3 = Reserved bit 2 = Command 0 Completed 1 Not completed bit 1 = Track 000 0 Not found 1 Found bit 0 = DRAM 0 Not found 1 Found (CP-3022 always 0)
01F1h	W	Write precomposition register for hard drive 0
01F2h	R / W	Sector count for hard disk 0
01F3h	R / W	Sector number for hard drive 0
01F4h	R / W	Number of cylinders (low byte) for hard drive 0
01F5h	R / W	Number of cylinders (high byte) for hard drive 0

Continued...

I/O Address	Read/Write Status	Description
01F6h	R / W	Drive/Head register for hard drive 0 bit 7 = 1 bit 6 = 0 bit 5 = 1 bit 4 = Drive select 0 First hard drive 1 Second hard drive bits 3-0 = Head select bits
01F7h	R	Status register for hard drive 0 bit 7 = 1 Controller is executing a command bit 6 = 1 Drive is ready bit 5 = 1 Write fault bit 4 = 1 Seek operation complete bit 3 = 1 Sector buffer requires servicing bit 2 = 1 Disk data read completed successfully bit 1 = Index (is set to 1 at each disk revolution) bit 0 = 1 Previous command ended with error
01F7h	W	Command register for hard drive 0
0200h - 020Fh	R / W	Game controller ports
0201h	R / W	I/O data - game port
0220h – 022Fh	R / W	Soundport AD1816 reserved
I/O addresses 0278h - 027Ah are reserved for use with parallel port 2. See the bit definitions for addresses 0378h - 037Ah.		
0278h	R / W	Data port for parallel port 2
0279h	R	Status port for parallel port 2
0279h	W	PnP Adress register (only for PnP devices)
027Ah	R / W	Control port for parallel port 2
02B0h – 02BFh	R / W	Digital I/O for Latch, WDOG, Control
I/O addresses 02E8h - 02EFh are reserved for use with serial port 4. See the bit definitions for I/O addresses 03F8h - 03FFh.		
02E8h	W	Transmitter holding register for serial port 4
02E8h	R	Receive buffer register for serial port 4
02E8h	R / W	Baud rate divisor (low byte) when DLAB = 1
02E9h	R / W	Baud rate divisor (high byte) when DLAB = 1
02E9h	R / W	Interrupt enable register when DLAB = 0
02EAh	R	Interrupt identification register for serial port 4
02EBh	R / W	Line control register for serial port 4
02ECh	R / W	Modem control register for serial port 4
02EDh	R	Line status register for serial port 4
02EEh	R	Modem status register for serial port 4
02EFh	R / W	Scratch register for serial port 4 (used for diagnostics)

Continued...

I/O Address	Read/Write Status	Description
I/O addresses 02F8h - 02FFh are reserved for use with serial port 2. See the bit definitions for I/O addresses 03F8h - 03FFh.		
02F8h	W	Transmitter holding register for serial port 2
02F8h	R	Receive buffer register for serial port 2
02F8h	R / W	Baud rate divisor (low byte) when DLAB = 1
02F9h	R / W	Baud rate divisor (high byte) when DLAB = 1
02F9h	R / W	Interrupt enable register when DLAB = 0
02FAh	R	Interrupt identification register for serial port 2
02FBh	R / W	Line control register for serial port 2
02FCh	R / W	Modem control register for serial port 2
02FDh	R	Line status register for serial port 2
02FEh	R	Modem status register for serial port 2
02FFh	R / W	Scratch register for serial port 2 (used for diagnostics)
0300h – 031Fh	R / W	LAN controller if installed
I/O addresses 0372h - 0377h are reserved for use with a secondary diskette controller. See the bit definitions for 03F2h - 03F7h.		
0372h	W	Digital output register for secondary diskette drive controller
0374h	R	Status register for secondary diskette drive controller
0375h	R / W	Data register for secondary diskette drive controller
0376h	R / W	Control register for secondary diskette drive controller
0377h	R	Digital input register for secondary diskette drive controller
0377h	W	Select register for secondary diskette data transfer rate
0378h	R / W	Data port for parallel port 1 bit 7 = 0 Busy bit 6 = 0 Acknowledge bit 5 = 1 Out of paper bit 4 = 1 Printer is selected bit 3 = 0 Error bit 2 = 0 IRQ has occurred bit 1-0 = Reserved
0379h	R / W	Status port for parallel port 1 bit 7 = 0 Busy bit 6 = 0 Acknowledge bit 5 = 1 Out of paper bit 4 = 1 Printer is selected bit 3 = 0 Error bit 2 = 0 IRQ has occurred bit 1-0 = Reserved

Continued...

I/O Address	Read/Write Status	Description
037Ah	R / W	Control port for parallel port 1 bits 7-5 = Reserved bit 4 = 1 Enable IRQ bit 3 = 1 Select printer bit 2 = 0 Initialize printer bit 1 = 1 Automatic line feed bit 0 = 1 Strobe
03B0h - 03B8h	R / W	Various video registers
I/O addresses 03BCh - 03BEh are reserved for use with parallel port 3. See the bit definitions for addresses 0378h - 037Ah.		
03BCh	R / W	Data port - parallel port 3
03BDh	R / W	Status port - parallel port 3
03BEh	R / W	Control port - parallel port 3
03C0h - 03CFh	R / W	Video subsystem (EGA/VGA)
03C2h - 03D9h	R / W	Various CGA and CRTC registers
03E0h	R / W	PCCARD Adress select
03E1h	R / W	PCCARD Data transfer with 365SL controller
I/O addresses 03E8h - 03EFh are reserved for use with serial port 3. See the bit definitions for I/O addresses 03F8h - 03FFh.		
03E8h	W	Transmitter holding register for serial port 3
03E8h	R	Receive buffer register for serial port 3
03E8h	R / W	Baud rate divisor (low byte) when DLAB = 1
03E9h	R / W	Baud rate divisor (high byte) when DLAB = 1
03E9h	R / W	Interrupt enable register when DLAB = 0
03EAh	R	Interrupt identification register for serial port 3
03EBh	R / W	Line control register for serial port 3
03ECh	R / W	Modem control register for serial port 3
03EDh	R	Line status register for serial port 3
03EEh	R	Modem status register for serial port 3
03EFh	R / W	Scratch register for serial port 3 (used for diagnostics)
03F2h	W	Digital output register for primary diskette drive controller bits 7-6 = 0 Reserved bit 5 = 1 Enable drive 1 motor bit 4 = 1 Enable drive 0 motor bit 3 = 1 Enable diskette DMA bit 2 = 0 Reset controller bit 1 = 0 Reserved bit 0 = 0 Select drive 0 1 Select drive 1

Continued...

I/O Address	Read/Write Status	Description
03F4h	R	Status register for primary diskette drive controller bit 7 = 1 Data register is ready bit 6 = 0 Transfer from system to controller 1 Transfer from controller to system bit 5 = 1 Non-DMA mode bit 4 = 1 Diskette drive controller is busy bits 3-2 = Reserved bit 1 = 1 Drive 1 is busy bit 0 = 1 Drive 0 is busy
03F5h	R / W	Data register for primary diskette drive controller
03F6h	R	Control port for primary diskette drive controller bits 7-4 = Reserved bit 3 = 0 Reduce write current 1 Head select enable bit 2 = 0 Disable diskette drive reset 1 Enable diskette drive reset bit 1 = 0 Disable diskette drive initialization 1 Enable diskette drive initialization bit 0 = Reserved
03F7h	R	Digital input register for primary diskette drive controller bit 7 = 1 Diskette drive line change bit 6 = 1 Write gate bit 5 = Head select 3 / reduced write current bit 4 = Head select 2 bit 3 = Head select 1 bit 2 = Head select 0 bit 1 = Drive 1 select bit 0 = Drive 0 select
03F7h	W	Select register for primary diskette data transfer rate bits 7-2 = Reserved bits 1-0 = 00 500 Kbs mode 01 300 Kbs mode 10 250 Kbs mode 11 Reserved
I/O addresses 03F8h - 03FFh are reserved for use with serial port 1. The bit definitions for these addresses also apply to serial ports 2, 3, and 4.		
03F8h	W	Transmitter holding register for serial port 1 - Contains the character to be sent. Bit 0, the least significant bit, is the first bit sent. bits 7-0 = Data bits 0-7 when the Divisor Latch Access Bit (DLAB) is 0
03F8h	R	Receive buffer register for serial port 1 - Contains the character to be received. Bit 0, the least significant bit, is the first bit received. bits 7-0 = Data bits 0-7 when the Divisor Latch Access Bit (DLAB) is 0

Continued...

I/O Address	Read/Write Status	Description
03F8h	R / W	Baud rate divisor (low byte) - This byte along with the high byte (03F9h) store the data transmission rate divisor. bits 7-0 = Data bits 0-7 when the Divisor Latch Access Bit (DLAB) is 1
03F9h	R / W	Baud rate divisor (high byte) - This byte along with the low byte (03F8h) store the data transmission rate divisor. bits 7-0 = Bits 8-15 when DLAB = 1
03F9h	R / W	Interrupt enable register bits 7-4 = Reserved bit 3 = 1 Modem status interrupt enable bit 2 = 1 Receiver line status interrupt enable bit 1 = 1 Transmitter holding register empty interrupt enable bit 0 = 1 Received data available interrupt enable when DLAB = 0
03FAh	R	Interrupt identification register - serial port 1 bits 7-3 = Reserved bits 2-1 = Identify interrupt with highest priority 00 Modem status interrupt (4th priority) 01 Transmitter holding register empty (3rd priority) 10 Received data available (2nd priority) 11 Receiver line status interrupt (1st priority) bit 0 = 0 Interrupt pending (register contents can be used as a pointer to interrupt service routine) 1 No interrupt pending
03FBh	R / W	Line control register - serial port 1 bit 7 = Divisor Latch Access (DLAB) 0 Access receiver buffer, transmitter holding register, and interrupt enable register 1 Access divisor latch bit 6 = 1 Set break enable. Forces serial output to spacing state and remains there bit 5 = Stick parity bit 4 = Even parity select bit 3 = Parity enable bit 2 = Number of stop bits bit 1 = Word length 00 5-bit word length 01 6-bit word length 10 7-bit word length 11 8-bit word length
03FCh	R / W	Modem control register - serial port 1 bits 7-5 = Reserved bit 4 = 1 Loopback mode for diagnostic testing of serial port. bit 3 = 1 User-defined output 2 bit 2 = 1 User-defined output 1 bit 1 = Force Request To Send active bit 0 = Force Data Terminal Ready active

Continued...

I/O Address	Read/Write Status	Description
03FDh	R	Line status register - serial port 1 bit 7 = Reserved bit 6 = 1 Transmitting shift and holding registers empty bit 5 = 1 Transmitter shift register empty bit 4 = 1 Break interrupt bit 3 = 1 Framing error bit 2 = 1 Overrun error bit 0 = 1 Data ready
03FEh	R	Modem status register - serial port 1 bit 7 = 1 Data Carrier Detect bit 6 = 1 Ring Indicator bit 5 = 1 Data Set Ready bit 4 = 1 Clear To Send bit 3 = 1 Delta Data Carrier bit 2 = 1 Trailing Edge Ring Indicator bit 1 = 1 Delta Data Set Ready bit 0 = 1 Delta Clear To Send
03FFh	R / W	Scratch register - serial port 1 (used for diagnostics)
0A79h	W	PnP Data write register (only for PnP devices)

4.9 BIOS Data Area Definitions

The BIOS Data Area is an area within system RAM that contains information about the system environment. System environment information includes definitions associated with hard disks, diskette drives, keyboard, video, as well as other BIOS functions. This area is created when the system is first powered on. It occupies a 256-byte area from 0400h - 04FFh. The following table lists the contents of the BIOS data area locations in offset order starting from segment address 40:00h.

Location	Description
00h - 07h	I/O addresses for up to 4 serial ports
08h - 0Dh	I/O addresses for up to 3 parallel ports
0Eh - 0Fh	Segment address of extended data address
10h - 11h	Equipment list bits 15-14 = Number of parallel printer adapters 00 = Not installed 01 = One 10 = Two 11 = Three bits 13-12 = Reserved bits 11-9= Number of serial adapters 00 = Not installed 001 = One 010 = Two 011 = Three 100 = Four bit 8 = Reserved bits 7-6 = Number of diskette drives 00 = One drive 01 = Two drives bits 5-4 = Initial video mode 00 = EGA or PGA 01 = 40 x 25 color 10 = 80 x 25 color 11 = 80 x 25 monochrome bit 3 = Reserved bit 2 = (1) Pointing device present bit 1 = (1) Math coprocessor present bit 0 = (1) Diskette drive present
12h	Reserved for port testing by manufacturer bits 7-1 = Reserved bit 0 = (0) Non-test mode (1) Test mode
13h	Memory size in kilobytes - low byte
14h	Memory size in kilobytes - high byte

Continued...

BIOS Data Area Definitions Continued...

Location	Description
15h - 16h	Reserved
17h	Keyboard Shift Qualifier States bit 7 = Insert mode bit 6 = CAPS lock bit 5 = Numlock bit 4 = Scroll Lock bit 3 = Either Alt key bit 2 = Either control key bit 1 = Left Shift key bit 0 = Right shift key 0 = not set / 1 = set
18h	Keyboard Toggle Key States bit 7 = (1) Insert held down bit 6 = (1) CAPS lock held down bit 5 = (1) Num Lock held down bit 4 = (1) Scroll Lock held down bit 3 = (1) Control+Num Lock held down bit 2 = (1) Sys Re held down bit 1 = (1) Left Alt held down bit 0 = (1) Left Control held down
19h	Scratch area for input from Alt key and numeric keypad
1Ah - 1Bh	Pointer to next character in keyboard buffer
1Ch - 1Dh	Pointer to last character in keyboard buffer
1Eh - 3Dh	Keyboard Buffer. Consists of 16 word entries.
3Eh	Diskette Drive Recalibration Flag bit 7 = (1) Diskette hardware interrupt occurred bits 6-4 = Not used bits 3-2 = Reserved bit 1 = (0) Recalibrate drive B bit 0 = (0) Recalibrate drive A

Continued...

BIOS Data Area Definitions Continued...

Location	Description
3Fh	Diskette Drive Motor Status bit 7 = Current operation 0 = Write or Format 1 = Read or Verify bit 6 = Reserved bits 5-4 = Drive Select 00 = Drive A 01 = Drive B bits 3-2 = Reserved 0 = Disable 1 = Enabled bit 1 = Drive B Motor Status 0 = Off 1 = On bit 1 = Drive A Motor Status 0 = Off 1 = On
40h	Diskette Drive Motor Timeout Disk drive motor is powered off when the value via the INT 08h timer interrupt reaches 0.
41h	Diskette Drive Status bit 7 = Drive Ready 0 = Ready 1 = Not ready bit 6 = Seek Error 0 = No error 1= Error occurred bit 5 = Controller operation 0 = Working 1 = Failed bits 4-0 = Error Codes 00h = No error 01h = Invalid function requested 02h = Address mark not located 03h = Write protect error 04h = Sector not found 06h = Diskette change line active (door opened) 08h = DMA overrun error 09h = Data boundary error 0Ch = Unknown media type 10h = ECC or CRC error 20h = Controller failure 40h = Seek operation failure 80h = Timeout
42h - 48h	Diskette Controller Status Bytes
49h	Video Mode Setting
4Ah - 4Bh	Number of Columns on screen
4Ch - 4Dh	Size of Current Page, in bytes
4Eh - 4Fh	Address of Current Page

Continued...

BIOS Data Area Definitions Continued...

Location	Description
50h - 5Fh	Position of cursor for each video page. Current cursor position is stored two bytes per page. First byte specifies the column, the second byte specifies the row.
60h - 61h	Start and end lines for 6845-compatible cursor type. 60h = starting scan line, 61h = ending scan line.
62h	Current Video Display Page
63h - 64h	6845-compatible I/O port address for current mode 3B4h = Monochrome 3D4h = Color
65h	Register for current mode select
66h	Current palette setting
67 - 6Ah	Address of adapter ROM
6Bh	Last interrupt the occurred
6Ch - 6Dh	Low word of timer count
6Eh - 6Fh	High word of timer count
70h	Timer count for 24-hour rollover flag
71h	Break key flag
72h - 73h	Reset flag 1243h = Soft reset. Memory test is bypassed.
74h	Status of last hard disk operation 00h = No error 01h = Invalid function requested 02h = Address mark not located 03h = Write protect error 04h = Sector not found 05h = Reset failed 08h = DMA overrun error 09h = Data boundary error 0Ah = Bad sector flag selected 0Bh = Bad track detected 0Dh = Invalid number of sectors on format 0Eh = Control data address mark detected 0Fh = DMA arbitration level out of range 10h = ECC or CRC error 11h = Data error corrected by ECC 20h = Controller failure 40h = Seek operation failure 80h = Timeout AAh = Drive not ready BBh = Undefined error occurred CCh = Write fault on selected drive E0h = Status error or error register = 0 FFh = Sense operation failed
75h	Number of hard drives
76h - 77h	Work area for hard disk

Continued...

BIOS Data Area Definitions Continued...

Location	Description
78h - 7Bh	Default parallel port timeout values
7Dh - 7Fh	Default serial port timeout values
80h - 81h	Pointer to start of keyboard buffer
82h - 83h	Pointer to end of keyboard buffer
84h - 88h	Reserved for EGA/VGA BIOS
8Ah	Reserved
8Bh	Diskette drive data transfer rate information bits 7-5 = Data rate on last operation 00 = 500 KBS 01 = 300 KBS 10 = 250 KBS bits 5-4 = Last drive step rate selected bits 3-2 = Data transfer rate at start of operation 00 = 500 KBS 01 = 300 KBS 10 = 250 KBS bits 1-0 = Reserved
8Ch	Copy of hard status register
8Dh	Copy of hard drive error register
8Eh	Hard drive interrupt flag
8Fh	Diskette controller information bit 7 = Reserved bit 6 = (1) Drive confirmed for drive B bit 5 = (1) Drive B is multi-rate bit 4 = (1) Drive B supports line change bit 3 = Reserved bit 2 = (1) Drive determined for drive A bit 1 = (1) Drive B is multi-rate bit 0 = (1) Drive B supports line change
90h - 91h	Media type for drives bits 7-6 = Data transfer rate 00 = 500 KBS 01 = 300 KBS 10 = 250 KBS bit 5 = (1) Double stepping required when 360K diskette inserted into 1.2MB drive bit 4 = (1) Known media is in drive bit 3 = Reserved bits 2-0 = Definitions upon return to user applications 000 = Testing 360K in 360K drive 001 = Testing 360K in 1.2 MB drive 010 = Testing 1.2 MB in 1.2 MB drive 011 = Confirmed 360K in 360K drive 100 = Confirmed 360K in 1.2 MB 101 = Confirmed 1.2 MB in 1.2 MB drive 111 = 720K in 720K drive or 1.44 MB in 1.44 MB drive

Continued...

BIOS Data Area Definitions Continued...

Location	Description
92h - 93h	Scratch area for diskette media. Low byte for drive A, high byte for drive B.
94h - 95h	Current track number for both drives. Low byte for drive A, high byte for drive B.
96h	Keyboard Status bit 7 = (1) Read ID bit 6 = (1) Last code was first ID bit 5 = (1) Force to Num Lock after read ID bit 4 = (1) Enhanced keyboard installed bit 3 = (1) Right ALT key active bit 2 = (1) Right Control key active bit 1 = (1) Last code was E0h bit 0 = (1) Last code was E1h
97h	Keyboard Status bit 7 = (1) Keyboard error bit 6 = (1) Updating LEDs bit 5 = (1) Resend code received bit 4 = (1) Acknowledge received bit 3 = Reserved bit 2 = (1) Caps lock LED state bit 1 = (1) Num lock LED state bit 0 = (1) Scroll lock LED state
98h - 99h	Offset address of user wait flag
9Ah - 9Bh	Segment address of user wait flag
9Ch - 9Dh	Wait count, in microseconds (low word)
9Eh - 9Fh	Wait count, in microseconds (high word)
A0h	Wait active flag bit 7 = (1) Time has elapsed bits 6-1 = Reserved bit 0 = (1) INT 15h, AH = 86h occurred
A1h - A7h	Reserved
A8h - ABh	Pointer to video parameters and overrides
ACh - FFh	Reserved
100h	Print screen status byte

4.9.1 Compatibility Service Table

In order to ensure compatibility with industry-standard memory locations for interrupt service routines and miscellaneous tabular data, the BIOS maintains tables and jump vectors.

Location	Description
FE05Bh	Entry Point for POST
FE2C3h	Entry point for INT 02h (NMI service routine)
FE3FEh	Entry point for INT 13h (Diskette Drive Services)
FE401h	Hard Drive Parameters Table
FE6F1h	Entry point for INT 19h (Bootstrap Loader routine)
FE6F5h	System Configuration Table
FE739h	Entry point for INT 14h (Serial Communications)
FE82Eh	Entry point for INT 16h (Keyboard Services)
FE897h	Entry point for INT 09h (Keyboard Services)
FEC59h	Entry point for INT 13h (Diskette Drive Services)
FEF57h	Entry point for INT OEh (Diskette Hardware Interrupt)
FEFC7h	Diskette Drive Parameters Table
FEFD2h	Entry point for INT 17h (Parallel Printer Services)
FF065h	Entry point for INT 10h (CGA Video Services)
FF0A4h	Video Parameter Table (6845 Data Table - CGA)
FF841h	Entry point for INT 12h (Memory Size Service)
FF84Dh	Entry point for INT 11h (Equipment List Service)
FF859h	Entry point for INT 15h (System Services)
Location	**Description**
FFA6Eh	Video graphics and text mode tables
FFE6Eh	Entry point for INT 1Ah (Time-of-Day Service)
FFEA5h	Entry Point for INT 08h (System Timer Service)
FFEF3h	Vector offset table loaded by POST
FFF53h	Dummy Interrupt routine IRET Instruction
FFF54h	Entry point for INT 05h (Print Screen Service)
FFFF0h	Entry point for Power-on
FFFF5h	BIOS Build Date (in ASCII)
FFFFEh	BIOS ID

4.10 VGA, LCD

4.10.1 VGA / LCD Controller C&T69000 (optional C&T69030)

69000 High Performance Flat Panel / CRT HiQVideo™ Accelerator with Integrated Memory

- Highly integrated Flat Panel and CRT GUI Accelerator & Multimedia Engine, Palette/DAC, Clock Synthesizer, and integrated frame buffer
- Integrated High performance SDRAM memory. 2MB integrated memory, 83 MHz SDRAM operation
- HiQColor™ Technology implemented with TMED (Temporal Modulated Energy Distribution)
- Hardware Windows Acceleration
- Integrated composite NTSC / PAL Support
- Hardware Multimedia Support
- High-Performance Flat Panel Display resolution and color depth at 3.3V
- 36-bit direct interface to color and monochrom, single drive (SS), and dual drive (DD), STN & TFT panels
- Advanced Power Management features minimize power usage in:
  - Normal operation
  - Standby (Sleep) modes
  - Panel-Off Power-Saving Mode
- VESA Standards supported
- Fully Compatible with IBM® VGA
- Driver Support for Windows 3.1, Windows 95/98, Windows NT3.1/NT4.0

4.10.2 VGA / LCD BIOS for 69000

VGA BIOS

The 69000 VGA BIOS (hereafter referred to as 69000 BIOS) is an enhanced, high performance BIOS that is used with the 69000 VGA Flat Panel/CRT Controller to provide an integrated Flat panel VGA solution. The BIOS is optimized for 69000 VGA Flat Panel/CRT Controller and provides:

Full compatibility with the IBM VGA BIOS
Support for monochrome LCD, 640x480, 800x600, 1024x768 and 1280x1024 TFT or STN displays.
Optional support for other displays.
Supports VESA BIOS Extensions, including VBE 2.0, VBE/DDC 1.0, and VBE/PM 1.0.
Supports either VESA local bus or PCI bus
Extended BIOS functions which offer easy access to 69000 control ler features and capabilities
Support for simultaneous display
44K BIOS supports 8 panels
48K BIOS supports 16 panels

High Performance Integrated Memory

The integrated SDRAM memory can support up to 83MHz operation, thus increasing the available memory bandwidth for the graphics subsystem. The result is support for additional high color / high resolution graphics modes combined with real-time video acceleration. This additional bandwidth also allows more flexibility in the other graphics functions intensely used in Graphics User Interface (GUIs) such as Microsoft™ Windows™.

Versatile Panel Support

The 69000 support a wide varety of monochrome and color Single-Panel, Single-Drive (SS) and Dual-Panel, Dual-Drive (DD), standard and high-resolution, passive STN and active matrix TFT/MIM LCD, and EL panels. With HiQColor™ technology, up to 256 gray scales are supported on passive STN LCDs. Up to 16.7M different colors can be displayed on passive STN LCDs and up to 16.7M colors on 24bit active matrix LCDs.

The 69000 offers a varety of programmable features to optimize display quality. Vertical centering and streching are provided for handling modes with less than 480 lines on 480-line panels. Horizontal and vertical streching capabilities are also available for both text and graphics modes for optimal display of VGA text and graphics modes on 800x600, 1024x768 and 1280x1024 panels.

Low Power Consumption

The 69000 uses a variety of advanced power management features to reduce power consumption of the display sub-system and to extend battery life. optimized for 3.3V operation, the 69000 internal logic, bus and panel interfaces operate at 3.3V but can tolerate 5V operation.

Software Compatibility / Flexibility

The 69000 is fully compatible with the VGA standard at both the register and BIOS levels. DIGITAL-LOGIC supply a fully VGA compatible BIOS, end-user utilities and drivers for common application programs.

Acceleration for All Panels and All Mode

The 69000 graphics engine is designed to support high performance graphics and video acceleration for all supported display resolutions, display types, and color modes. There is no compromise in performance operating in 8, 16, or 24 bpp color modes allowing true acceleration while displaying up to 16.7M colors.

4.10.3 Display Modes Supported

The 69000 supports the modes which appear in the table below.

Resolution	Color (bpp)	Refresh Rates (Hz)
640x480	8	60, 75, 85
640x480	16	60, 75, 85
640x480	24	60, 75, 85
800x600	8	60, 75, 85
800x600	16	60, 75, 85
800x600	24	60, 75, 85
1024x768	8	60, 75, 85
1024x768	16	60, 75, 85
1280x1024	8	60

4.10.4 VGA/LCD BIOS Support

Each LCD display needs a specific adapted VGA-BIOS.
Standard this product is equiped with the CRT standard VGABIOS.

To connect a LCD Display to this product, you need to perform the following:

1. Check the FP_LIST.PDF if the LCD BIOS is available.
   Ask DIGITAL-LOGIC to get the latest VGA-BIOS file !

IF THE LCD BIOS IS AVAILABLE:

2. In the FLATPANEL-SUPPORT documentation the connection between the LCD and this product will be described.
3. DOWNLOAD the corresponfing LCD-BIOS with the utility DOWN.EXE
   Go the the section 4.6.8 DOWNLOAD THE VGABIOS in this manual and follow those steps.
4. Restart the system and check the VGA-BIOS header message. The LCD name must be visible for only a short time. The VGABIOS message appears as first info page on the screen.
5. Stop the system, connect the LCD to the system and restart again
6. If on the LCD no image appears, as soon as the monitor begins to show the first text, stop the system immediately, otherways the LCD will become damaged.
7. Check the LCD connection again.

FOR A NEW LCD TYPE, NOT AVAILABLE NOW:

If the LCD BIOS for your LCD is not available, DIGITAL-LOGIC will adapt the LCD and provide you with one working cable. To initialise this, we need the following points from you:

1. An order to adapt the LCD (for the costs ask your sales contact)
2. Send the LCD panle, a datasheet, a connector to the LCD and the inverter for the backligth

ATTENTION:

DIGITAL-LOGIC AG is never responsible for a damaged LCD display. Even not if in the BIOS or in any documentation for the LCD is a mistake.

4.10.5 Memory 69000 CRT/TFT Panels

Hor. Resol.	Vert. Resol.	Color bpp	Refr. Hz	DCLK Mhz	MEM kByte	Cursor kByte	FB/C kByte	FB/M kByte	Video Input kByte	Total with Video	Total w/o Video
640	480	8	60	25.175	300	4.2	0	0	300	604	304
640	480	8	72	31.500	300	4.2	0	0	300	604	304
640	480	8	75	31.500	300	4.2	0	0	300	604	304
640	480	8	85	36.000	300	4.2	0	0	300	604	304
640	480	16	60	25.175	600	4.2	0	0	300	904	604
640	480	16	72	31.500	600	4.2	0	0	300	904	604
640	480	16	75	31.500	600	4.2	0	0	300	904	604
640	480	16	85	36.000	600	4.2	0	0	300	904	604
640	480	24	60	25.175	900	4.2	0	0	300	1204	904
640	480	24	72	31.500	900	4.2	0	0	300	1204	904
640	480	24	75	31.500	900	4.2	0	0	300	1204	904
640	480	24	85	36.000	900	4.2	0	0	300	1204	904
800	600	8	60	40.000	469	4.2	0	0	300	773	473
800	600	8	72	50.000	469	4.2	0	0	300	773	473
800	600	8	75	49.500	469	4.2	0	0	300	773	473
800	600	8	85	56.250	469	4.2	0	0	300	773	473
800	600	16	60	40.000	938	4.2	0	0	300	1242	942
800	600	16	72	50.000	938	4.2	0	0	300	1242	942
800	600	16	75	49.500	938	4.2	0	0	300	1242	942
800	600	16	85	56.250	938	4.2	0	0	300	1242	942
800	600	24	60	40.000	1406	4.2	0	0	300	1710	1410
800	600	24	72	50.000	1406	4.2	0	0	300	1710	1410
800	600	24	75	49.500	1406	4.2	0	0	300	1710	1410
800	600	24	85	56.250	1406	4.2	0	0	300	1710	1410
1024	768	16	60	65.000	1536	4.2	0	0	300	1840	1540
1024	768	16	70	75.000	1536	4.2	0	0	300	1840	1540
1024	768	16	75	78.750	1536	4.2	0	0	300	1840	1540
1024	768	16	85	94.500	1536	4.2	0	0	300	1840	1540
1024	768	24	60	65.000	2304	4.2	0	0	300	2608	2308
1024	768	24	72	75.000	2304	4.2	0	0	300	2608	2308
1024	768	24	75	78.750	2304	4.2	0	0	300	2608	2308
1024	768	24	85	94.500	2304	4.2	0	0	300	2608	2308
1280	1024	16	60	108.0	2560	4.2	0	0	300	2864	2564
1280	1024	16	70	128.0	2560	4.2	0	0	300	2864	2564
1280	1024	16	75	135.0	2560	4.2	0	0	300	2864	2564
1280	1024	16	85	157.5	2560	4.2	0	0	300	2864	2564
1280	1024	24	60	108.0	3840	4.2	0	0	300	4144!	3844
1280	1024	24	72	128.0	3840	4.2	0	0	300	4144!	3844
1280	1024	24	75	135.0	3840	4.2	0	0	300	4144!	3844
1280	1024	24	85	157.5	3840	4.2	0	0	300	4144!	3844

! means not possible resolution with the 4Mb Video RAM

4.10.6 Memory 69000 Color STN-DD Panels

Hor. Resol.	Vert. Resol.	Color bpp	Refr. Hz	DCLK Mhz	MEM kByte	Cursor kByte	FB/C kByte	FB/M kByte	Video Input kByte	Total with Video	Total w/o Video
640	480	8	60	25.175	300	4.2	120	0	300	724	424
640	480	8	72	31.500	300	4.2	120	0	300	724	424
640	480	8	75	31.500	300	4.2	120	0	300	724	424
640	480	8	85	36.000	300	4.2	120	0	300	724	424
640	480	16	60	25.175	600	4.2	120	0	300	1024	724
640	480	16	72	31.500	600	4.2	120	0	300	1024	724
640	480	16	75	31.500	600	4.2	120	0	300	1024	724
640	480	16	85	36.000	600	4.2	120	0	300	1024	724
640	480	24	60	25.175	900	4.2	120	0	300	1324	1024
640	480	24	72	31.500	900	4.2	120	0	300	1324	1024
640	480	24	75	31.500	900	4.2	120	0	300	1324	1024
640	480	24	85	36.000	900	4.2	120	0	300	1324	1024
800	600	8	60	40.000	469	4.2	188	0	300	960	660
800	600	8	72	50.000	469	4.2	188	0	300	960	660
800	600	8	75	49.500	469	4.2	188	0	300	960	660
800	600	8	85	56.250	469	4.2	188	0	300	960	660
800	600	16	60	40.000	938	4.2	188	0	300	1429	1129
800	600	16	72	50.000	938	4.2	188	0	300	1429	1129
800	600	16	75	49.500	938	4.2	188	0	300	1429	1129
800	600	16	85	56.250	938	4.2	188	0	300	1429	1129
800	600	24	60	40.000	1406	4.2	188	0	300	1898	1598
800	600	24	72	50.000	1406	4.2	188	0	300	1898	1598
800	600	24	75	49.500	1406	4.2	188	0	300	1898	1598
800	600	24	85	56.250	1406	4.2	188	0	300	1898	1598
1024	768	16	60	65.000	1536	4.2	307	0	300	2147	1847
1024	768	16	70	75.000	1536	4.2	307	0	300	2147	1847
1024	768	16	75	78.750	1536	4.2	307	0	300	2147	1847
1024	768	16	85	94.500	1536	4.2	307	0	300	2147	1847
1024	768	24	60	65.000	2304	4.2	307	0	300	2915	2615
1024	768	24	72	75.000	2304	4.2	307	0	300	2915	2615
1024	768	24	75	78.750	2304	4.2	307	0	300	2915	2615
1024	768	24	85	94.500	2304	4.2	307	0	300	2915	2615
1280	1024	16	60	108.0	2560	4.2	512	0	300	3376	3676
1280	1024	16	70	128.0	2560	4.2	512	0	300	3376	3676
1280	1024	16	75	135.0	2560	4.2	512	0	300	3376	3676
1280	1024	16	85	157.5	2560	4.2	512	0	300	3376	3676
1280	1024	24	60	108.0	3840	4.2	512	0	300	4656!	4356!
1280	1024	24	72	128.0	3840	4.2	512	0	300	4656!	4356!
1280	1024	24	75	135.0	3840	4.2	512	0	300	4656!	4356!
1280	1024	24	85	157.5	3840	4.2	512	0	300	4656!	4356!

! means not possible resolution with the 4Mb Video RAM

4.10.7 Memory 69000 Mono STN-DD Panels

Hor. Resol.	Vert. Resol.	Color bpp	Refr. Hz	DCLK Mhz	MEM kByte	Cursor kByte	FB/C kByte	FB/M kByte	Video Input kByte	Total with Video	Total w/o Video
640	480	8	60	25.175	300	4.2	0	38	300	642	342
640	480	8	72	31.500	300	4.2	0	38	300	642	342
640	480	8	75	31.500	300	4.2	0	38	300	642	342
640	480	8	85	36.000	300	4.2	0	38	300	642	342
640	480	16	60	25.175	600	4.2	0	38	300	942	642
640	480	16	72	31.500	600	4.2	0	38	300	942	642
640	480	16	75	31.500	600	4.2	0	38	300	942	642
640	480	16	85	36.000	600	4.2	0	38	300	942	642
640	480	24	60	25.175	900	4.2	0	38	300	1242	942
640	480	24	72	31.500	900	4.2	0	38	300	1242	942
640	480	24	75	31.500	900	4.2	0	38	300	1242	942
640	480	24	85	36.000	900	4.2	0	38	300	1242	942
800	600	8	60	40.000	469	4.2	0	59	300	832	532
800	600	8	72	50.000	469	4.2	0	59	300	832	532
800	600	8	75	49.500	469	4.2	0	59	300	832	532
800	600	8	85	56.250	469	4.2	0	59	300	832	532
800	600	16	60	40.000	938	4.2	0	59	300	1300	1000
800	600	16	72	50.000	938	4.2	0	59	300	1300	1000
800	600	16	75	49.500	938	4.2	0	59	300	1300	1000
800	600	16	85	56.250	938	4.2	0	59	300	1300	1000
800	600	24	60	40.000	1406	4.2	0	59	300	1769	1469
800	600	24	72	50.000	1406	4.2	0	59	300	1769	1469
800	600	24	75	49.500	1406	4.2	0	59	300	1769	1469
800	600	24	85	56.250	1406	4.2	0	59	300	1769	1469

! means not possible resolution with the 4Mb Video RAM

4.11 HiQ Video Multimedia Support

The 69000 uses independent multimedia capture and display systems on chip. The capture system places data in display memory (usually off screen) and the display system places the data in a window on the screen.
The capture system can receive data from the video portin the 422 YUV format. The YUV data are served from the VideoInputProcessor (VIP) type SAA7111A. The VIP converts the analog CVBS information, coming from a videocamera, into the YUV digital information.
The YUV input data can also be scaled down in the 69000 bevor storage in the display memory. Capture of input data may also be double buffered for smoothing and to prevent image tearing. To better support MPEG2 (DVD) video decompression, the 69000 includes a line buffer to directly support the native format of MPEG2 data of 720 pixels wide.

The capture engine also supports image mirroring and rotation for camera support. This feature is important for applications such as video teleconferencing because it allows the image movements to appear on the display as it actually occurs.

The display system can independently place YUV data from anywhere in the display memory into an on-screen window which can be any size and located at any pixel boundary (YUV data is converted to RGB „on-the-fly“). This is important for the 69000 since the video must be stored in the integrated 2MB frame buffer and thus optimized to require very little space. Interlanced and non-interlanced data are both supported in the capture and display system.

Display Modes Supported

The 69000 supports the modes which appear in the table below.

Resolution:	Color (bpp)	Refresh Rates (Hz)
640 x 480	8	60, 75, 85
640 x 480	16	60, 75, 85
640 x 480	24	60, 75, 85
800 x 600	8	60, 75, 85
800 x 600	16	60, 75, 85
800 x 600	24	60, 75, 85
1024 x 768	8	60, 75, 85
1024 x 768	16	60, 75, 85
1280 x 1024	8	60

4.11.1 HiQVideo Series Programming Examples

4.11.2 Introduction

This application note shows how the CHIPS HiQVideo??Series controllers can be used for video capture and playback. This document includes a description of the hardware configuration, a discussion of the functions, and actual programming examples.

4.11.3 Video Playback through PCI/VL Bus

The new generation of Chips and Technologies, Inc. Multimedia Accelerators (6555x) supports Color Space Conversion and Stretching (Zooming) in the back end with the chroma color key. The color space conversion functionality of the 6555x can be made available to video codecs by implementing the off-screen surface support in the DCI Provider (Windows 3.1 drivers). Only the playback feature of 6555x multimedia module is used to implement extended DCI functionality. This means the video input to the 6555x is kept in the frozen state (not grabbing) when DCI is running. For video playback, the CPU can write YUV, RGB15, and RGB16 data into the off-screen memory and fill the destination rectangle (where video need to be displayed on the visible screen) with the color key. Video can be zoomed up if the destination rectangle is bigger than the source rectangle in the off-screen buffer.

4.11.4 Video Capture and Playback Through Video Port

The new generation of CHIPS Multimedia Accelerators (69000) can also capture live video from the video port into the off-screen memory and play it back with color space conversion onto a color keyed destination rectangle on the visible screen. Playback video can be zoomed up to fill the bigger destination rectangle while incoming video can be scaled down to fit into a smaller off-screen memory buffer or smaller destination rectangle. Zoomed video can be smoothed out with horizontal and vertical interpolation. Scaled down video can also be filtered out at input before capturing into the frame buffer. Input video can be cropped for the extra data which is usually associated with the NTSC or PAL video. The 69000 hardware can accommodate fast or slow capture applications through the CPU Bus by capturing the video frames in one of three methods: continuously, one frame at a time, or one every nth (n = 1-15) frame. Following diagram demonstrates the video capture.

VideoRect comes from the input video stream fed through the Video Port (VAFC / ZV Port) and comprises of one of the following sizes based on the input source.

NTSC:	640x480 60 fields / second (interlaced), Square Aspect Ratio (4:3).
	720x486 60 fields / second (interlaced), Non-square Aspect Ratio.
PAL:	768x576 50 fields / second (interlaced), Square Aspect Ratio (4:3).
	720x576 50 fields / second (interlaced), Non-square Aspect Ratio.
MPEG1:	320x240 30 frames / sec (non-interlaced), Square Aspect Ratio (4:3).
	352x288 30 frames / sec (non-interlaced), Non-square Aspect Ratio.
	Top-left of VideoRect is always at (0,0).

CropRect is defined relative to the **VideoRect**. **CropRect** is used to crop off some pixels from the top, left, right, or bottom to fit the image into a square pixel ratio or to drop some unwanted pixels. **CropRec**t is programmed using the acquisition window registers After cropping, the video is scaled down to fit into a smaller memory buffer or in a smaller display window. The scaled video is captured into off-screen video memory buffer or buffers (as in double buffer mode). There is a horizontal filter to reduce the sampling artifacts caused by input video scaling. Video in the capture buffer is displayed on top of the pixels which matches the color key and/or with a specified rectangular window.

4.11.5 ZoomUp

If client area of a window (**DispRect**) is larger than the capture buffer rectangle (**CaptRect**), the video can be zoomed up to fit into the DispRect.

4.11.6 Video Capture Using the Video Port

We need some additional functions to manage video capture through video port. Some of the initialization and exit code can be merged together with the playback code. Capture code should also include the previously described playback code.

4.11.7 How to enable video capture and playback module (Init)

This code should be executed before video starts flowing into the port.
1. Save and Set XRD0[4] = 1
2. Save and Set SAR04 = 0x2A; // To get wider (> 352) playback buffer width
Static USHORT XR60,XRD0,SAR04;

```
CaptureInit()
{
UCHAR XR_Index;
bVideoFlowingIn = 0; // assume video is not flowing into the port
XR_Index = ReadPortUshort(ulXrAddr); // Save XR Index
WritePortUchar(ulXrAddr,0xd0); // Read XRD0
xrD0 = ReadPortUshort(ulXrAddr); // Save XRD0
WritePortUshort(ulXrAddr,xrD0 | 0x7000);
// Enable video playback/Capture module
//
// Enable video port in 55x for ZV Port Style Video
//
WritePortUchar(ulXrAddr,0x60); // Read XR60
xr60 = ReadPortUshort(ulXrAddr); // Save XRD0
WritePortUshort(ulXrAddr,xr60 | 0x0300);
//
// Program 55x for playback of wider (> 352) video buffer.
//
WritePortUshort(ulXrAddr,0x044e); // Read SAR04
WritePortUchar(ulXrAddr,0x4f);
SAR04 = ReadPortUshort(ulXrAddr); // Save SAR04
WritePortUshort(ulXrAddr,(SAR04 & 0x00ff) | 0x2a00); // SAR04=2a
WritePortUchar(ulXrAddr,XR_Index); // Restore XR_Index
//
// Set video capture buffer address for both buffers.
//
u.d = osbMemAddress; // assign to a DWORD uinon to access bytes
WritePortUshort(ulMrAddr,((UINT)u.b[0] << 8) | MR_VIN_ADDR_1_L); //mr06
WritePortUshort(ulMrAddr,((UINT)u.b[1] << 8) | MR_VIN_ADDR_1_M); //mr07
WritePortUshort(ulMrAddr,((UINT)u.b[2] << 8) | MR_VIN_ADDR_1_H); //mr08
WritePortUshort(ulMrAddr,((UINT)u.b[0] << 8) | MR_VIN_ADDR_2_L); //mr09
WritePortUshort(ulMrAddr,((UINT)u.b[1] << 8) | MR_VIN_ADDR_2_M); //mr0a
WritePortUshort(ulMrAddr,((UINT)u.b[2] << 8) | MR_VIN_ADDR_2_H); //mr0b
WritePortUshort(ulMrAddr,((UINT)u.b[0] << 8) | MR_VDP_ADDR_1_L); //mr22
WritePortUshort(ulMrAddr,((UINT)u.b[1] << 8) | MR_VDP_ADDR_1_M); //mr23
WritePortUshort(ulMrAddr,((UINT)u.b[2] << 8) | MR_VDP_ADDR_1_H); //mr24
WritePortUshort(ulMrAddr,((UINT)u.b[0] << 8) | MR_VDP_ADDR_2_L); //mr25
WritePortUshort(ulMrAddr,((UINT)u.b[1] << 8) | MR_VDP_ADDR_2_M); //mr26
WritePortUshort(ulMrAddr,((UINT)u.b[2] << 8) | MR_VDP_ADDR_2_H); //mr27
//
// Set Aquisition rectangle to NULL (Left=-1, right=0) to avoid capturing of first
// frame. This is needed to latch the capture counter with the new address.
//
WritePortUshort(ulMrAddr,0xff0e); // program Left=-1
WritePortUshort(ulMrAddr,0xff0f);
WritePortUshort(ulMrAddr,0x0010); // program right=0
WritePortUshort(ulMrAddr,0x0011);
WritePortUshort(ulMrAddr,0x0012); // program top=0
WritePortUshort(ulMrAddr,0x0013);
WritePortUshort(ulMrAddr,0x0014); // program bottom=0
WritePortUshort(ulMrAddr,0x0015);
```

4.11.8 How to disable video playback and capture module (Exit)

1. Restore XRD0.
2. Restore SAR04.

```
UCHAR XR_Index;
XR_Index = ReadPortUshort(ulXrAddr); // Save XR Index
WritePortUshort(ulXrAddr,XRD0); // Restore XRD0
WritePortUshort(ulXrAddr,0x044e); // Read SAR04
WritePortUshortAddr,SAR04); // Restore SAR04
WritePortUchar(ulXrAddr,XR_Index); // Restore XR_Index
```

4.11.9 How to start video capture

```
// In 55x VGAs, capture counters are not updated with the new off-screen
// address until the next Input Video VSync. So, the first frame of the input
// video is captured at the old address left in the counters when we froze
// the video. This may cause the memory corruption. To avoid this problem we
// need to ignore the data of the first input video frame. We already set the
// acquisition window to NULL during initialization. Now all we have to do is to
// wait for the first couple of input Vsyncs then set the acquisition rectangle
// to proper values. Acquisition rectangle must not be set till video started
// flowing in (bVideoFlowingIn = 1). bVideoFlowingIn flag is set to 0 at
// initialization time.
// So let us perform the first frame ritual.
//
if(!bVideoFlowingIn)
{ // This is first time to start 550 video, see if video is flowing?
start_time = timeGetTime(); // 1 millisec precision
while(((timeGetTime() - start_time) < 200))
{ // wait for 200 milisec (33.3 ms for 30Hz video) and VSyncActivity
WritePortUchar(ulMrAddr, MR_VIN_CTRL_4);
if(ReadPortUshort(ulMrAddr) & (VIC4_VSYNC << 8))
{
bVideoFlowingIn = 1; // video start flowing
//
// Wait for Input VSync is over.
//
start_time = timeGetTime(); // 1 millisec precision
WritePortUchar(ulMrAddr, MR_VIN_CTRL_4);
while((ReadPortUshort(ulMrAddr) & (VIC4_VSYNC << 8)) &&
((timeGetTime() - start_time) < 200));
//
// Now wait for next VSync.
//
start_time = timeGetTime(); // 1 millisec precision
WritePortUchar(ulMrAddr, MR_VIN_CTRL_4);
while(!(ReadPortUshort(ulMrAddr) & (VIC4_VSYNC << 8)) &&
((timeGetTime() - start_time) < 200));
//
// Restore crop.right
//
SetCropRect((LPRECTL)&rCrop);
break;
}
}
} // FirstTime VideoFlowingIn
mr03 |= (VIC2_START_GRAB << 8); // unfreeze the video (start capturing)
WritePortUshort(ulMrAddr,mr03); // write new value
```

4.11.10 How to stop video capture

```
//---------------------------------------------------------------------------
// FreezeVideo() : Stops capturing the incoming video; whatever is in the
// frame buffer is being displayed.
//
// Enter:
// none
// Exit :
// Nothing
//---------------------------------------------------------------------------
void FreezeVideo()
{
int mr03;
DWORD start_time;
WritePortUchar(ulMrAddr, MR_VIN_CTRL_2);
mr03 = ReadPortUshort(ulMrAddr); // Read current value
if((mr03 & (VIC2_START_GRAB << 8)))
{ // video is running, h/w is grabbing video, wait for input VSync
mr03 &= ~(VIC2_START_GRAB << 8); // turn off the bit to freeze the video
//
// Sometimes if Video Input is not comming thru video port (ZV Port
// Disabled) then we will never get Video VSync (hanging problem). We must
// time out our wait for VSync. If we do not see VSync within 2 frames
// of input VSync (80 Miliseconds for 25Hz Video, worst case) we must get
// out of the waiting loop.
//
start_time = timeGetTime(); // 1 millisec precision
WritePortUchar(ulMrAddr, MR_VIN_CTRL_4);
while( !(ReadPortUshort(ulMrAddr) & (VIC4_VSYNC << 8)) &&
((timeGetTime() - start_time) < 200));
WritePortUshort(ulMrAddr, mr03); // freeze the video
start_time = timeGetTime(); // 1 millisec precision
WritePortUchar(ulMrAddr, MR_VIN_CTRL_4);
while((ReadPortUshort(ulMrAddr) & (VIC4_FRM_READY << 8)) &&
((timeGetTime() - start_time) < 200));
```

4.11.11 How to set input video color format

CHIPS 69000 supports three basic color formats which are YUV4:2:2, RG555 and RGB565. Each format is 16 bit per pixel. The 69000 also allows swapping of the UV positions within a 32 bit dword. The default sequence for YUV4:2:2 is Byte0=Y0, Byte1=U, Byte2=Y1, Byte3=V. The following code shows the input video format selection.

```
//-----------------------------------------------------------------------------
// SetVideoInputFormat() - sets Video Transfer Format bits
//-----------------------------------------------------------------------------
int ChipsVideoOverlay::SetVideoInputFormat(int iVideoFormat)
{
UINT i;
WritePortUchar(ulMrAddr,MR_VIN_CTRL_1); // read video display control reg1
i = ReadPortUshort(ulMrAddr);
i &= ~(VIC1_FORMAT << 8); // clear format bits (0 is YUV4:2:2)
switch(iVideoFormat)
{
case CMM_FMT_YUV_422:
// i |= (VIC1_YUV422 << 8); // 0 is YUV 4:2:2
break;
case CMM_FMT_RGB_555:
i |= (VIC1_RGB555 << 8);
break;
case CMM_FMT_RGB_565:
i |= (VIC1_RGB565 << 8);
break;
default:
return;
}
WritePortUshort(ulMrAddr,i); // write new format
}
```

4.11.12 How to set interlaced or non-interlaced video input

CHIPS 69000 supports interlaced or non-interlaced video sources. Usually, the NTSC/PAL video sources are interlaced and the hardware MPEG decoder generates non-interlaced video source. Following code selects video input type.

```
void SetVideoInputBits(BOOL interlaced)
{ // interlaced = 1 for interlaced video source
int mr02;
WritePortUchar(ulMrAddr, MR_VIN_CTRL_1);
mr02 = ReadPortUshort(ulMrAddr); // Read current value
mr02 &= ~(VIC1_NONINTERLACE << 8); // assume interlaced video
if(interlaced) mr02 |= (VIC1_NONINTERLACE << 8);
WritePortUshort(ulMrAddr, mr02); // write new value
```

4.11.13 How to enable/disable double buffer

CHIPS 69000 supports double buffering for the video capture and playback. Double buffering needs more memory but it minimizes the tearing effect generated by fast changing pictures. We assume that there is enough memory to accommodate both buffers and that the buffer address is programmed in (MR06, MR07, MR08, MR09, MR0A, MR0B). The following code sets/resets double buffering.

```
void SetDoubleBuffer(BOOL double_buffer)
{ // double_buffer = 1 to enable double buffer
int mr04,mr20;
WritePortUchar(ulMrAddr, MR_VIN_CTRL_3);
mr04 = ReadPortUshort(ulMrAddr); // Read current value
mr04 &= ~((VIC3_DB_VLOCK+VIC3_ENABLE_DB) << 8); // assume no double buffer
if(interlaced) mr04 |= ((VIC3_DB_VLOCK+VIC3_ENABLE_DB) << 8);
WritePortUshort(ulMrAddr, mr04); // write new value
//
// Enable double buffer for video playback which locked with the input VSync.
//
WritePortUchar(ulMrAddr, MR_VDP_CTRL_3);
mr20 = ReadPortUshort(ulMrAddr); // Read current value
mr20 &= ~((VDC3_DB_VLOVK+VDC3_DB_TRIGGER) << 8); // assume no double buffer
if(interlaced) mr20 |= ((VDC3_DB_VLOVK+VDC3_DB_TRIGGER) << 8);
WritePortUshort(ulMrAddr, mr20); // write new value
```

4.11.14 How to scale input video (before acquiring into frame buffer)

CHIPS 69000 can scale down the video before capturing into the off-screen buffer.

```
//---------------------------------------------------------------------------
// SetVideoInputScale() : Sets video input scaling factors. Video input scaling
// factor depends on aquisition rectangle and frame buffer rectangle (source
// rectangle).
//
// Enter:
// wCrop = crop rectangle width
// hCrp = crop rectangle height
// wCap = Capture buffer width
// hCap = Capture buffer height
// Exit :
// Nothing
//---------------------------------------------------------------------------
void SetVideoInputScale(int wCrop, int hCrop, int wCap, int hCap)
{
UINT mr03,scale_x,scale_y;
WritePortUchar(ulMrAddr, MR_VIN_CTRL_2);
mr03 = ReadPortUshort(ulMrAddr);
mr03 &= ~((VIC2_SCALE_X | VIC2_SCALE_Y) << 8); // assume no scaling
if(wCrop > wCap)
{ // horizontal input scaling needed
scale_x = (int)(((DWORD)wCap*VIN_SCALE_X_MAX) / (DWORD)wCrop);
WritePortUshort(ulMrAddr, (scale_x << 8) | MR_VIN_SCALE_X);
mr03 |= (VIC2_SCALE_X << 8); // enable input scaling
}
if(hCrop > hCap)
{ // vertical input scaling needed
scale_y = (int)(((DWORD)hCap*VIN_SCALE_Y_MAX) / (DWORD)hCrop);
WritePortUshort(ulMrAddr, (scale_y << 8) | MR_VIN_SCALE_Y);
mr03 |= (VIC2_SCALE_Y << 8); // enable input scaling
}
WritePortUshort(ulMrAddr, mr03); // set scale factors
```

4.11.15 How to crop input video (programming of acquisition rectangle)

Video acquisition rectangle is used to crop the unwanted video input data before the 69000 hardware scales it and grabs it into off-screen buffer. This is also used to crop vertical blank interval data (Closed Caption or Tele Text) from the NTSC video.

```
//-----------------------------------------------------------------------------
// SetCropRect() : Sets cropping rectrangle on input video recctangle.
//
// +----------------+
// | +----------+   |
// | |          |   |
// | | CropRect |   |
// | |          |   |
// | +----------+   |
// | Input Video    |
// +----------------+
//
// Enter:
// lpRect Crop Rectangle withing Input Video rectangle (NTSC/PAL dependent)
// rCrop - local copy of Cropping Rectangle
// bVideoFlowingIn = 0 if video is not flowing into the port, 1 normally.
// Exit :
// Nothing
//-----------------------------------------------------------------------------
void SetCropRect(LPRECTL lpRect)
{
UINT mr0e,mr0f,mr10,mr11,mr12,mr13,mr14,mr15;
union WORD16 u;
if(&rCrop != lpRect) rCrop = *lpRect; // copy crop rectangle into our area
if(!bVideoFlowingIn) return;
// Use NULL Rectangle done by static initialization
u.w = (USHORT)((int)rCrop.left);
mr0e = ((UINT)u.b[0] << 8) + MR_VIN_AQW_XL_L;
mr0f = ((UINT)u.b[1] << 8) + MR_VIN_AQW_XL_H;
u.w = (USHORT)((int)rCrop.right -1);
mr10 = ((UINT)u.b[0] << 8) + MR_VIN_AQW_XR_L;
mr11 = ((UINT)u.b[1] << 8) + MR_VIN_AQW_XR_H;
u.w = (USHORT)((int)rCrop.top);
mr12 = ((UINT)u.b[0] << 8) + MR_VIN_AQW_YT_L;
mr13 = ((UINT)u.b[1] << 8) + MR_VIN_AQW_YT_H;
u.w = (USHORT)((int)rCrop.bottom -1);
mr14 = ((UINT)u.b[0] << 8) + MR_VIN_AQW_YB_L;
mr15 = ((UINT)u.b[1] << 8) + MR_VIN_AQW_YB_H;
WritePortUshort(ulMrAddr,mr0e); // program Left
WritePortUshort(ulMrAddr,mr0f);
WritePortUshort(ulMrAddr,mr10); // program right
WritePortUshort(ulMrAddr,mr11);
WritePortUshort(ulMrAddr,mr12); // program top
WritePortUshort(ulMrAddr,mr13);
WritePortUshort(ulMrAddr,mr14); // program bottom
WritePortUshort(ulMrAddr,mr15);
```

Definition of CHIPSMM.H

```
/*************************************************************************
* Description: Hardware register definitiona file for 6555x *
* Copyright (C) Chips and Technologies, Inc. 1995 *
*************************************************************************/
#define WritePortUchar(p,v) outp((USHORT)p,v)
#define WritePortUshort(p,v) outpw((USHORT)p,v)
#define WritePortUlong(p,v) outpd((USHORT)p,v)
#define ReadPortUchar(p) inp((USHORT)p)
#define ReadPortUshort(p) inpw((USHORT)p)
#define ReadPortUlong(p) inpd((USHORT)p)
//-----------------------------------------------------------------------
// Chips multimedia register description for 6555x registers.
// Any chages here must also be made in CHIPSMMH.INC
//-----------------------------------------------------------------------
#define ADDR_FR 0x03d0 // C&T Flat Panel Register address port
#define DATA_FR 0x03d1 // C&T Flat Panel Register data port
#define ADDR_MR 0x03d2 // C&T Multimedia Register address port
#define DATA_MR 0x03d3 // C&T Multimedia Register data port
#define ADDR_EXTR 0x03d6 // C&T XR Address
#define DATA_EXTR 0x03d7 // C&T XR Data
#define MR_CAPS_REG_1 0x00 // Multimedia capabilities reg 1
#define MR_CAPS_REG_2 0x01 // Multimedia capabilities reg 2
#define MR_VIN_CTRL_1 0x02 // Video Input Control Reg 1
#define MR_VIN_CTRL_2 0x03 // Video Input Control Reg 2
#define MR_VIN_CTRL_3 0x04 // Video Input Control Reg 3
#define MR_VIN_CTRL_4 0x05 // Video Input Control Reg 4(stat reg)
#define MR_VIN_ADDR_1_L 0x06 // Video Input Address Pointer 1 (low)
#define MR_VIN_ADDR_1_M 0x07 // Video Input Address Pointer 1 (mid)
#define MR_VIN_ADDR_1_H 0x08 // Video Input Address Pointer 1 (high)
#define MR_VIN_ADDR_2_L 0x09 // Video Input Address Pointer 2 (low)
#define MR_VIN_ADDR_2_M 0x0A // Video Input Address Pointer 2 (mid)
#define MR_VIN_ADDR_2_H 0x0B // Video Input Address Pointer 2 (high)
#define MR_VIN_PITCH_QD 0x0C // Pitch of Video Input buff in quad
// words (8 bytes = 1QD)
#define MR_VIN_AQW_XL_L 0x0E // Aquisition Window X-Left low
#define MR_VIN_AQW_XL_H 0x0F // Aquisition Window X-Left high
#define MR_VIN_AQW_XR_L 0x10 // Aquisition Window X-Right low
#define MR_VIN_AQW_XR_H 0x11 // Aquisition Window X-Right high
#define MR_VIN_AQW_YT_L 0x12 // Aquisition Window Y-Top low
#define MR_VIN_AQW_YT_H 0x13 // Aquisition Window Y-Top high
#define MR_VIN_AQW_YB_L 0x14 // Aquisition Window Y-Bottom low
#define MR_VIN_AQW_YB_H 0x15 // Aquisition Window Y-Bottom high
#define MR_VIN_SCALE_X 0x16 // Video Input Horizontal scale factor
#define MR_VIN_SCALE_Y 0x17 // Video Input Vertical scale factor
#define MR_VIN_FRMCOUNT 0x18 // Frame Count for Nth Frame capturing
//-----------------------------------------------------------------------
// Video Display Registers:
//-----------------------------------------------------------------------
#define MR_VDP_CTRL_1 0x1E // Video Display Control Reg 1
#define MR_VDP_CTRL_2 0x1F // Video Display Control Reg 2
#define MR_VDP_CTRL_3 0x20 // Video Display Control Reg 3
#define MR_VDP_CTRL_4 0x21 // Video Display Control Reg 4 (status)
#define MR_VDP_ADDR_1_L 0x22 // Video Display Addr Pointer 1 (low)
#define MR_VDP_ADDR_1_M 0x23 // Video Display Addr Pointer 1 (mid)
#define MR_VDP_ADDR_1_H 0x24 // Video Display Addr Pointer 1 (high)
#define MR_VDP_ADDR_2_L 0x25 // Video Display Addr Pointer 2 (low)
#define MR_VDP_ADDR_2_M 0x26 // Video Display Addr Pointer 2 (mid)
#define MR_VDP_ADDR_2_H 0x27 // Video Display Addr Pointer 2 (high)
```

```
#define MR_VDP_PITCH_QD 0x28 // Pitch of Video Display Window in
// quad words (8 bytes = 1QD)
#define MR_VDP_WIN_XL_L 0x2A // Display Window X-Left low
#define MR_VDP_WIN_XL_H 0x2B // Display Window X-Left high
#define MR_VDP_WIN_XR_L 0x2C // Display Window X-Right low
#define MR_VDP_WIN_XR_H 0x2D // Display Window X-Right high
#define MR_VDP_WIN_YT_L 0x2E // Display Window Y-Top low
#define MR_VDP_WIN_YT_H 0x2F // Display Window Y-Top high
#define MR_VDP_WIN_YB_L 0x30 // Display Window Y-Bottom low
#define MR_VDP_WIN_YB_H 0x31 // Display Window Y-Bottom high
#define MR_VDP_ZOOM_X 0x32 // Video Display Horizontal Zoom factor
#define MR_VDP_ZOOM_Y 0x33 // Video Display Vertical Zoom factor
//-----------------------------------------------------------------------
// Color key registers
//-----------------------------------------------------------------------
#define MR_VDP_CKEY_CTRL 0x3C // Video Color Key Control
#define MR_VDP_CKEY_0 0x3F //sw Graphics Color Key Reg 0 (blue)
#define MR_VDP_CKEY_1 0x3E // Graphics Color Key Reg 1 (green)
#define MR_VDP_CKEY_2 0x3D //sw Graphics Color Key Reg 2 (red)
#define MR_VDP_CKEY_M0 0x42 //sw Graphics Color Key Mask Reg0(blue)
#define MR_VDP_CKEY_M1 0x41 // Graphics Color Key Mask Reg1 (green)
#define MR_VDP_CKEY_M2 0x40 //sw Graphics Color Key Mask Reg2 (red)
#define MR_CRT_SCAN_LO 0x43 // Current CRTC Refresh Scanline Line
// Read Counter lo 8 bits
#define MR_CRT_SCAN_HI 0x44 // Current CRTC Refresh Scanline Line
// Read Counter hi 4 bits
//-----------------------------------------------------------------------
// Multimedia capabilities register_1 definitions (MR00):
//-----------------------------------------------------------------------
#define MCAPS_PLAYBACK 0x01 // Play back available
#define MCAPS_CAPTURE 0x02 // Capture available
//-----------------------------------------------------------------------
// Bit definition of Video Input Control Register1 (MR_VIN_CTRL_1)
//-----------------------------------------------------------------------
#define VIC1_NONINTERLACE 0x01 // Interlaced video input
#define VIC1_GAMEFORMAT 0x02 // Game format (duplicate field) video
#define VIC1_YUV422 0x00 // Video Input is YUV
#define VIC1_RGB565 0x04 // RGB16 video input (0 is YUV)
#define VIC1_RGB555 0x0C // RGB15 video input
#define VIC1_FORMAT 0x0E // all format bits
#define VIC1_HSYNC_HI 0x10 // H-Sync Polarity : Hi asserted
#define VIC1_VSYNC_HI 0x20 // V-Sync Polarity : Hi asserted
#define VIC1_FLD_DT_INV 0x40 // Field detect polarity inverted
#define VIC1_FLD_DT_LDE 0x80 // Field detect method leading edge
//-----------------------------------------------------------------------
// Bit definition of Video Input Control Register2 (MR_VIN_CTRL_2)
//-----------------------------------------------------------------------
#define VIC2_START_GRAB 0x01 // 1:start grab, 0:stop grab
#define VIC2_SINGLE 0x02 // 1:single frame, 0:continuous
#define VIC2_FIELD_GRAB 0x04 // 1:field grab, 0:frame grab
#define VIC2_ODD_FIELD 0x08 // 1:odd field grab, 0:even filed grab
#define VIC2_SCALE_X 0x10 // 1:enable x_scaling, 0:full screen
#define VIC2_SCALE_Y 0x20 // 1:enable y_scaling, 0:full screen
#define VIC2_YSCALE_ES 0x40 // 1:y-scale even spaced, 0:normal
#define VIC2_YSCALE_OW 0x80 // y-scale overwrite, 0:as per prev bit
//-----------------------------------------------------------------------
// Bit definition of Video Input Control Register3 (MR_VIN_CTRL_3)
//-----------------------------------------------------------------------
#define VIC3_X_MIRRORED 0x01 // capture direction, 1:right to left,
// 0: left to right
```

```
#define VIC3_Y_FLIPPED 0x02 // capture direction, 1:bottom to top,
// 0: top to bottom
#define VIC3_HFILTER 0x04 // 1:enable horizontal filter at input,
// 0:no h filter
#define VIC3_DB_VLOCK 0x08 // 1:DoubleBuffer Vsync locked,
// 0:DoubleBuffer CPU forced
#define VIC3_ENABLE_DB 0x10 // 1:Enable DoubleBuffer,
// 0:No DoubleBuffer
#define VIC3_PTR1_INUSE 0x20 // 1:PTR 1 in use for DoubleBuffer,
// 0:PTR 0 in use for DoubleBuffer
#define VIC3_CAPTURE_NF 0x80 // 1:Capture Nth Frame/Field,
// 0:Capture single frame
//-------------------------------------------------------------------------
// Bit definition of Video Input Status Register (MR_VIN_CTRL_4)
//-------------------------------------------------------------------------
#define VIC4_FRM_READY 0x01 // 1:Frame is ready for grab by CPU
// (synced with VSync)
#define VIC4_VSYNC 0x08 // VSync after polarity correction
// (read only)
#define VIC4_PQE_PIXEL 0x10 // 1:Pixel Qualifier as valid pixel
// 0:Pixel Qualifier as Blank signal
#define VIC4_PQP_INV 0x20 // 1:Pixel Qualifier polarity inverted,
// 0:Pixel Qualifier normal
#define VIC4_SWAP_UV 0x40 // 1:Swap U & V,
// 0:UV Normal sequence
#define VIC4_HY_LUV 0x80 // 1:Y on high and UV on low pins(VESA)
// 0:UV on high and Y on low pins
//-------------------------------------------------------------------------
// Bit definition of Video Display Control Register1 (MR_VDP_CTRL_1)
//-------------------------------------------------------------------------
#define VDC1_X_MIRRORED 0x01 // 1:mirrored (right to left),
// 0:normal (left to right)
#define VDC1_Y_FLIPPED 0x02 // 1:Flipped (bottom to top),
// 0:normal (top to bottom)
#define VDC1_ZOOM_X 0x04 // 1:enable x_zoom (zoom based on reg),
// 0:normal
#define VDC1_ZOOM_Y 0x08 // 1:enable y_zoom (zoom based on reg),
// 0:normal
#define VDC1_INTERLACE 0x10 // 1:VGA Mode is interlaced,
// 0:non-interlaced mode
//-------------------------------------------------------------------------
// Bit definition of Video Display Control Register2 (MR_VDP_CTRL_2)
//-------------------------------------------------------------------------
#define VDC2_YUV422 0x00 // Video Buf is YUV4:2:2
#define VDC2_UV_SWAP 0x01 // Video Buf is YUV4:2:2 with UV Swap
#define VDC2_SIGNED_UV 0x02 // Video Buf is YUV4:2:2 with Signed UV
#define VDC2_YUV422_UVS 0x01 // Video Buf is YUV4:2:2 with UV Swap
#define VDC2_YUV422_SUV 0x02 // Video Buf is YUV4:2:2 with Signed UV
#define VDC2_YUV422_UV 0x03 // Video Buf is YUV4:2:2 Signed UV&Sawp
#define VDC2_RGB555 0x09 // Video Buffer is RGB15 (5-5-5)
#define VDC2_RGB565 0x08 // Video Buffer is RGB16 (5-6-5)
#define VDC2_FORMAT 0x1F // All format bits
#define VDC2_H_INTERPOL 0x20 // Enable Horizontal Interpolation
#define VDC2_VI_RUNAVRG 0x40 // Vertical Interpolation is done as
// running average method
#define VDC2_V_INTERPOL 0x80 // Enable Vertical Interpolation
//-------------------------------------------------------------------------
// Bit definition of Video Display Control Register3 (MR_VDP_CTRL_3)
//-------------------------------------------------------------------------
#define VDC3_DB_TRIGGER 0x08 // Display new pointer on next VSync
```

```
#define VDC3_DB_CPU_PTR 0x04 // 1:Dbl buf src is buf ptr set by CPU
// 0:Dbl buf src is Input Aqisition's
// last frame
#define VDC3_DB_PTR2 0x10 // 1:CPU buffer is pointer 2
// 0:CPU buffer is pointer 1
#define VDC3_DB_VLOVK 0x20 // 1:Double buffer is VSync locked
// 0:Double buffer is unlocked
//-------------------------------------------------------------------------
// Bit definition of Video Display Status Register4 (MR_VDP_CTRL_4)
//-------------------------------------------------------------------------
#define VDC4_DB_PENDING 0x01 // 1:hasn't displayed CPU set buffer
// 0:CPU Set buffer is displayed or in
// process of being displayed
#define VDC4_DB_USEPTR2 0x02 // 1:PTR2 is being displayed
// 0:PTR1 is being displayed
//-------------------------------------------------------------------------
// Bit definition of Video Color Key Control Register (MR_VDP_CKEY_CTRL)
//-------------------------------------------------------------------------
#define VDC_EV_OVERLAY 0x01 // 1:Enable video overlay
// 0:Display graphics only
#define VDC_EV_COLOR_KEY 0x02 // 1:Enable video display using clr key
// 0:Color Key Disabled
#define VDC_EV_XY_RECT 0x04 // 1:Enable video display in Rect Rgn
// 0:Video Display in Rect Rgn disabled
#define VDC_ENABLE_VAFC 0x08 // 1:Enable external VAFC (like 545)
// for color key only
// 0:Our own video play back
#define VDC_VAFC_18 0x10 // 1:18 bit external VAFC
// 0:16 bit external VAFC
#define VDC_BIT_15_KEY 0x40 // 1:in 16BPP modes MSB is routed thru
// Blue0 for color key
// 0:normal color key
#define VDC_BIT_0_KEY 0x80 // 1:enable blue0 clr key for 16/24BP
// 0:normal color key for 16/24BPP mode
#define VIN_SCALE_X_MAX 0x100 // max value of x_scale reg (8 bit reg)
#define VIN_SCALE_Y_MAX 0x100 // max value of y_scale reg (8 bit reg)
//#define VDP_ZOOM_X_MAX 0x100 // max value of x_zoom reg (ES1 100h)
//#define VDP_ZOOM_Y_MAX 0x100 // max value of y_zoom reg (ES1 100h)
#define VDP_ZOOM_X_MAX 0x40 // max value of x_zoom reg (ES0 40h)
#define VDP_ZOOM_Y_MAX 0x40 // max value of y_zoom reg (ES0 40h)
/*
```

4.11.16 Video Input with the SAA7111 VIP

The 69000 Series integrated configuration reguires only a NTSC/PAL decoder SAA7111 and standard DRAMs, without additional memory. The ITT decoder implements a 3 channel video multiplexer, which may be software controlled over the I2C bus.

- Four analog inputs, internal analog source selectors, e.g. 4 CVBS or 2 Y/C or (1 Y/C and 2 CVBS)
- Two analog preprocessing channels
- Fully programmable static gain for the main channels or automatic gain control for the selected CVBS or Y/C channel
- Switchable white peak control
- Two built-in analog anti-aliasing filters
- Two 8-bit video CMOS analog-to-digital converters
- On-chip clock generator
- Line-locked system clock frequencies
- Digital PLL for horizontal-sync processing and clock generation
- Requires only one crystal (24.576 MHz) for all standards
- Horizontal and vertical sync detection
- Automatic detection of 50 and 60 Hz field frequency, and automatic switching between PAL and NTSC standards
- Luminance and chrominance signal processing for PAL BGHI, PAL N, PAL M, NTSC M, NTSC N, NTSC 4.43, SECAM
- User programmable luminance peaking or aperture correction
- Cross-colour reduction for NTSC by chrominance comb filtering
- PAL delay line for correcting PAL phase errors
- Real time status information output (RTCO)
- Brightness Contrast Saturation (BCS) control on-chip
- The YUV (CCIR-601) bus supports a data rate of:
    - 864 ??fH = 13.5 MHz for 625 line sources
    - 858 ??fH = 13.5 MHz for 525 line sources.
- Data output streams for 16, 12 or 8-bit width with the following formats:
    – YUV 4 :1 :1 (12-bit)
    – YUV 4 :2 :2 (16-bit) = used on the product
    – YUV 4 :2 :2 (CCIR-656) (8-bit)
    – RGB (5, 6, and 5) (16-bit) with dither
    – RGB (8, 8, and 8) (24-bit) with special application.
- Odd/even field identification by a non interlace CVBS input signal
- Fix level for RGB output format during horizontal blanking
- 720 active samples per line on the YUV bus
- One user programmable general purpose switch on an output pin
- Built-in line-21 text slicer
- A 27 MHz Vertical Blanking Interval (VBI) data bypass programmable by I 2 C-bus for INTERCAST applications
- Power-on control
- Two via I 2 C-bus switchable outputs for the digitized CVBS or Y/C input signals AD1 (7 to 0) and AD2 (7 to 0)
- Chip enable function (reset for the clock generator and power save mode up from chip version 3)
- Compatible with memory-based features (line-locked clock)
- Boundary scan test circuit complies with the ‘IEEE Std. 1149.1??1990’ (ID-Code = 0 F111 02 B)
- I2C-bus controlled (full read-back ability by an external controller)
- Low power (0.5 W), low voltage (3.3 V), small package (LQFP64)
- 5 V tolerant digital I/O ports.

The SECAM-processing contains the following blocks:
Baseband ‘bell’ filters to reconstruct the amplitude and phase equalized 0 and 90??FM-signals
Phase demodulator and differentiator (FM-demodulation)

Pe-emphasis filter to compensate the pre-emphasised input signal, including frequency offset compensation (DB or DR white carrier values are subtracted from the signal, controlled by the SECAM-switch signal).
The burst processing block provides the feedback loop of the chroma PLL and contains;

Burst gate accumulator
Colour identification and killer

Comparison nominal/actual burst amplitude (PAL/NTSC standards only)

Loop filter chrominance gain control (PAL/NTSC standards only)

Loop filter chrominance PLL (only active for PAL/NTSC standards)

PAL/SECAM sequence detection, H/2-switch generation

Increment generation for DTO1 with divider to generate stable subcarrier for non-standard signals.

The chrominance comb filter block eliminates crosstalk between the chrominance channels in accordance with the

PAL standard requirements. For NTSC colour standards the chrominance comb filter can be used to eliminate crosstalk from luminance to chrominance (cross-colour) for vertical structures. The comb filter can be switched off if desired. The embedded line delay is also used for SECAM recombination (cross-over switches).
The resulting signals are fed to the variable Y-delay compensation, RGB matrix, dithering circuit and output interface, which contains the VPO output formatter and the output control logic.

Luminance processing
The 8-bit luminance signal, a digital CVBS format or a luminance format (S-VHS, HI8), is fed through a switchable prefilter. High frequency components are emphasized to compensate for loss. The following chrominance trap filter (f0 = 4.43 or 3.58 MHz centre frequency selectable) eliminates most of the colour carrier signal, therefore, it must be bypassed for S-video (S-VHS and HI8) signals. The high frequency components of the luminance signal can be peaked (control for sharpness improvement via I2C-bus) in two band-pass filters with selectable transfer characteristic. This signal is then added to the original (unpeaked) signal. A switchable amplifier achieves
common DC amplification, because the DC gains are different in both chrominance trap modes. The improved luminance signal is fed to the BCS control located in the chrominance processing block.

RGB matrix
Y, Cr and Cb data are converted after interpolation into RGB data in accordance with CCIR-601 recommendations. The realized matrix equations consider the digital quantization:
R = Y + 1.371 Cr
G = Y 0.336 Cb 0.698 Cr
B = Y + 1.732 Cb.
After dithering (noise shaping) the RGB data is fed to the output interface within the VPO-bus output formatter.

VBI-data bypass
For a 27 MHz VBI-data bypass the offset binary CVBS signal is upsampled behind the ADCs. Upsampling of the CVBS signal from 13.5 to 27 MHz is possible, because the ADCs deliver high performance at 13.5 MHz sample clock.
Suppressing of the back folded CVBS frequency components after upsampling is achieved by an interpolation filter. The TUF block on the digital top level performs the upsampling and interpolation for the bypassed CVBS signal.

VPO-bus (digital outputs)
The 16-bit VPO-bus transfers digital data from the output interfaces to a feature box or a field memory, a digital colour space converter (SAA7192 DCSC), a video enhancement and digital-to-analog processor (SAA7165 VEDA2) or a colour graphics board (Targa-format) as a graphical user interface.

5 DESCRIPTION OF THE CONNECTORS

Flat cable

44pin IDE is: IDT Terminal for Dual Row (2.00mm grid) and 1.00mm flat cable
All others are: IDT Terminal for Dual Row 0.1" (2.54mm grid) and 1.27mm flat cable

X29 Power Supply connector

Pin	Signal
Pin 1	VCC Logic, CPU
Pin 2	Ground
Pin 3	Ground
Pin 4	+12V Input for LCD's

X14/X17 = COM1 serial port connector RS-232C
X15 = COM2 serial port connector RS-232C

Channel:		Header onboard	D-SUB connector	Signal
COM	1:	Pin 1	Pin 1	= DCD
X14/ X17		Pin 2	Pin 6	= DSR
		Pin 3	Pin 2	= RxD
		Pin 4	Pin 7	= RTS
		Pin 5	Pin 3	= TxD
		Pin 6	Pin 8	= CTS
		Pin 7	Pin 4	= DTR
		Pin 8	Pin 9	= RI
		Pin 9	Pin 5	= GND
		Pin 10	-	= +5V
COM	2:	Pin 1	Pin 1	= DCD
X15		Pin 2	Pin 6	= DSR
		Pin 3	Pin 2	= RxD
		Pin 4	Pin 7	= RTS
		Pin 5	Pin 3	= TxD
		Pin 6	Pin 8	= CTS
		Pin 7	Pin 4	= DTR
		Pin 8	Pin 9	= RI
		Pin 9	Pin 5	= GND
		Pin 10	-	= +5V

X16 = COM3/4 serial port connector RS-232C

Channel:	Header onboard	9pin D-SUB	Signal
COM 3:	Pin 1	Pin 1	= DCD
X16	Pin 2	Pin 6	= DSR
	Pin 3	Pin 2	= RxD
	Pin 4	Pin 7	= RTS
	Pin 5	Pin 3	= TxD
	Pin 6	Pin 8	= CTS
	Pin 7	Pin 4	= DTR
	Pin 8	Pin 9	= RI
	Pin 9	Pin 5	= GND
	Pin 10	-	= +5V
COM 4:	Pin 11	Pin 1	= DCD
	Pin 12	Pin 6	= DSR
	Pin 13	Pin 2	= RxD
	Pin 14	Pin 7	= RTS
	Pin 15	Pin 3	= TxD
	Pin 16	Pin 8	= CTS
	Pin 17	Pin 4	= DTR
	Pin 18	Pin 9	= RI
	Pin 19	Pin 5	= GND
	Pin 20	-	= +5V

X25 COM3 serial port connector RS-422/485

Pin	Signal
Pin 1	TxD +
Pin 2	TxD -
Pin 3	RxD +
Pin 4	RxD -

X24 COM4 serial port connector RS-422/485

Pin	Signal
Pin 1	TxD +
Pin 2	TxD -
Pin 3	RxD +
Pin 4	RxD -

X12 Ethernet Twisted Pair Interface 10/100Mhz (only if this Option is assembled)

Pin	Signal
Pin 1	TxD +
Pin 2	TxD -
Pin 3	RxD +
Pin 6	RxD -
Pin 4,5,7,8	pull down with 75ohms

X7 (primary) / X8 (secondary) IDE interface connector 44 pins RM2.00mm on rear side

Pin	Signal	Pin	Signal
Pin 1	= Reset (active low)	Pin 2	= GND
Pin 3	= D7	Pin 4	= D8
Pin 5	= D6	Pin 6	= D9
Pin 7	= D5	Pin 8	= D10
Pin 9	= D4	Pin 10	= D11
Pin 11	= D3	Pin 12	= D12
Pin 13	= D2	Pin 14	= D13
Pin 15	= D1	Pin 16	= D14
Pin 17	= D0	Pin 18	= D15
Pin 19	= GND	Pin 20	= NC (keypin)
Pin 21	= DRQ0	Pin 22	= GND
Pin 23	= IOW (active low)	Pin 24	= GND
Pin 25	= IOR (active low)	Pin 26	= GND
Pin 27	= IORD4	Pin 28	= VCC pullup
Pin 29	= DACK0	Pin 30	= GND
Pin 31	= IRQ14	Pin 32	= IOCS16 (active low)
Pin 33	= ADR1	Pin 34	= NC
Pin 35	= ADR0	Pin 36	= ADR2
Pin 37	= CS0 (active low)	Pin 38	= CS1 (active low)
Pin 39	= LED (active low)	Pin 40	= GND
Pin 41	= VCC Logic	Pin 42	= VCC Motor
Pin 43	= GND	Pin 44	= NC

X20 Floppy disk interface connector for 5,25" and 3,5" FD (34 pins header)

Pin	Signal	Pin	Signal
Pin 1,	= GND	Pin 2	= RPM
Pin 3	= GND	Pin 4	= NC
Pin 5	= GND	Pin 6	= NC
Pin 7	= GND	Pin 8	= INDEX
Pin 9	= GND	Pin 10	= Motor on device 0
Pin 11	= GND	Pin 12	= Drive select 1
Pin 13	= GND	Pin 14	= Drive select 0
Pin 15	= GND	Pin 16	= Motor on device 1
Pin 17	= GND	Pin 18	= Head direction
Pin 19	= GND	Pin 20	= Step
Pin 21	= GND	Pin 22	= Write data
Pin 23	= GND	Pin 24	= Write gate
Pin 25	= GND	Pin 26	= Track 00
Pin 27	= GND	Pin 28	= Write protection
Pin 29	= GND	Pin 30	= Read data
Pin 31	= GND	Pin 32	= Head selection 0/1
Pin 33	= GND	Pin 34	= Disk change signal

X13 Optionally assembled on the rear side for micro floppy 3,5" (26 pins FCC-header)

Pin	Signal
Pin 1	= VCC
Pin 2	= Index
Pin 3	= VCC
Pin 4	= Drive select
Pin 5	= VCC
Pin 6	= Disk change signal
Pin 7	= nc
Pin 8	= nc
Pin 9	= nc
Pin 10	= Motor on
Pin 11	= nc
Pin 12	= Dir
Pin 13	= nc
Pin 14	= Step
Pin 15	= GND
Pin 16	= Write data
Pin 17	= GND
Pin 18	= Write gate
Pin 19	= GND
Pin 20	= Track 00
Pin 21	= GND
Pin 22	= Write protect
Pin 23	= GND
Pin 24	= Read data
Pin 25	= GND
Pin 26	= Head select

X19 Printerport connector (LPT1)

The printer connector provides an interface for 8 bit centronics printers.

Header onboard	D-SUB connector on the cable	Signal
Pin 1	Pin 1	= Strobe
Pin 3	Pin 2	= Data 0
Pin 5	Pin 3	= Data 1
Pin 7	Pin 4	= Data 2
Pin 9	Pin 5	= Data 3
Pin 11	Pin 6	= Data 4
Pin 13	Pin 7	= Data 5
Pin 15	Pin 8	= Data 6
Pin 17	Pin 9	= Data 7
Pin 19	Pin 10	= Acknowledge
Pin 21	Pin 11	= Busy
Pin 23	Pin 12	= Paper end
Pin 25	Pin 13	= Select
Pin 2	Pin 14	= Autofeed
Pin 4	Pin 15	= Error
Pin 6	Pin 16	= Init printer
Pin 8	Pin 17	= Shift in (SI)
Pin 10,12,14,16,18	Pin 18 - 22	= Left open
Pin 20,22,24	Pin 23 - 25	= Ground

X22 PS/2 - keyboard connector

The keyboard connector, is a 6pin PS/2 connector that provides an interface for PS/2-keyboards.

J32 Pin	Keyboard Signal
Pin 1	= Keyboard Data
Pin 2	= GND
Pin 3	= GND
Pin 4	= VCC , + 5 Volt
Pin 5	= Keyboard Clock
Pin 6	= GND

X21 PS/2 - mouse connector

The mouse connector, is a 6pin PS/2 connector that provides an interface for PS/2-mouse.

J32 Pin	Keyboard Signal
Pin 1	= Mouse Data
Pin 2	= GND
Pin 3	= GND
Pin 4	= VCC , + 5 Volt
Pin 5	= Mouse Clock
Pin 6	= GND

X3 VGA-CRT connector (HiDens DSUB 15pin)

* DDDA and DDCK are the DIGITAL-DISPLAY interface for power control functions of the monitor.

Pin	Signal
Pin 1	Red
Pin 2	Green
Pin 3	Blue
Pin 5	GND
Pin 6	GND
Pin 7	GND
Pin 8	GND
Pin 10	GND
Pin 12	DDDA *
Pin 13	H-Synch
Pin 14	V-Synch
Pin 15	DDCK *

X4 5V - VGA-LCD (buffered signals) 50pin RM2.00mm connector

Pin	Signal	Function
Pin 1	M/FPM	M-Clock
Pin 2	FLM	Frame
Pin 3	VBACKLSAFE	12V/1A for Backlight
Pin 4	LP	Line pulse
Pin 5	VCC for LCD	5V
Pin 6	GND	0V
Pin 7	VEESAFE	5V / 1A for VEE Generator
Pin 8	SHFCLK	Shift clock
Pin 9	VDDSAFE	5V or +12V/1A for VDD-LCD (selected by J55)
Pin 10	P0	Data 0
Pin 11	P1	Data 1
Pin 12	P2	Data 2
Pin 13	P3	Data 3
Pin 14	P4	Data 4
Pin 15	P5	Data 5
Pin 16	P6	Data 6
Pin 17	P7	Data 7
Pin 18	P8	Data 8
Pin 19	P9	Data 9
Pin 20	P10	Data 10
Pin 21	P11	Data 11
Pin 22	P12	Data 12
Pin 23	P13	Data 13
Pin 24	P14	Data 14
Pin 25	P15	Data 15
Pin 26	GND	GND
Pin 27	P16	Data 16
Pin 28	P17	Data 17
Pin 29	P18	Data 18
Pin 30	P19	Data 19
Pin 31	P20	Data 20
Pin 32	ACT	Activity
Pin 33	P21	Data 21
Pin 34	P22	Data 22
Pin 35	P23	Data 23
Pin 36	P24	Data 24
Pin 37	P25	Data 25
Pin 38	P26	Data 26
Pin 39	P27	Data 27
Pin 40	P28	Data 28
Pin 41	P29	Data 29
Pin 42	P30	Data 30
Pin 43	P31	Data 31
Pin 44	GND	GND
Pin 45	P32	Data 32
Pin 46	P33	Data 33
Pin 47	P34	Data 34
Pin 48	P35	Data 35
Pin 49	VCC	+5V/1A
Pin 50	+12V	+12V/1A

X5 3.3V - VGA-LCD (unbuffered signals) 50pin RM2.00mm connector

Pin	Signal	Function
Pin 1	M/FPM	M-Clock
Pin 2	FLM	Frame
Pin 3	VBACKLSAFE	12V/1A for Backlight
Pin 4	LP	Line pulse
Pin 5	VCC for LCD	3.3V
Pin 6	GND	0V
Pin 7	VEESAFE	5V / 1A for VEE Generator
Pin 8	SHFCLK	Shift clock
Pin 9	VDDSAFE	5V or +12V/1A for VDD-LCD (selected by J55)
Pin 10	P0	Data 0
Pin 11	P1	Data 1
Pin 12	P2	Data 2
Pin 13	P3	Data 3
Pin 14	P4	Data 4
Pin 15	P5	Data 5
Pin 16	P6	Data 6
Pin 17	P7	Data 7
Pin 18	P8	Data 8
Pin 19	P9	Data 9
Pin 20	P10	Data 10
Pin 21	P11	Data 11
Pin 22	P12	Data 12
Pin 23	P13	Data 13
Pin 24	P14	Data 14
Pin 25	P15	Data 15
Pin 26	GND	GND
Pin 27	P16	Data 16
Pin 28	P17	Data 17
Pin 29	P18	Data 18
Pin 30	P19	Data 19
Pin 31	P20	Data 20
Pin 32	ACT	Activity
Pin 33	P21	Data 21
Pin 34	P22	Data 22
Pin 35	P23	Data 23
Pin 36	P24	Data 24
Pin 37	P25	Data 25
Pin 38	P26	Data 26
Pin 39	P27	Data 27
Pin 40	P28	Data 28
Pin 41	P29	Data 29
Pin 42	P30	Data 30
Pin 43	P31	Data 31
Pin 44	GND	GND
Pin 45	P32	Data 32
Pin 46	P33	Data 33
Pin 47	P34	Data 34
Pin 48	P35	Data 35
Pin 49	VCC	+5V/1A
Pin 50	+12V	+12V/1A

X6 ZV-Port connector (26pin RM2.54) (if this option is assembled)

Pin	Signal	Pin	Signal
Pin 2	Data1	Pin 1	Data 0
Pin 4	Data 3	Pin 3	Data2
Pin 6	Data 5	Pin 5	Data 4
Pin 8	Data 7	Pin 7	Data 6
Pin 10	Data 9	Pin 9	Data 8
Pin 12	Data 11	Pin 11	Data 10
Pin 14	Data 13	Pin 13	Data 12
Pin 16	Data 15	Pin 15	Data 14
Pin 18	Vert.Ref	Pin 17	Horiz.Ref
Pin 20	VRDY	Pin 19	P-Clock
Pin 22	SCL	Pin 21	SDA
Pin 24	3.3V	Pin 23	5.0V
Pin 26	GND	Pin 25	GND

X23 SCSI connector (50pin RM2.54) (if this option is assembled)

Pin	Signal	Pin	Signal
Pin 2	SCSI data 0	Pin 1	GND
Pin 4	SCSI data 1	Pin 3	GND
Pin 6	SCSI data 2	Pin 5	GND
Pin 8	SCSI data 3	Pin 7	GND
Pin 10	SCSI data 4	Pin 9	GND
Pin 12	SCSI data 5	Pin 11	GND
Pin 14	SCSI data 6	Pin 13	GND
Pin 16	SCSI data 7	Pin 15	GND
Pin 18	SCSI parity	Pin 17	GND
Pin 20	GND	Pin 19	GND
Pin 22	Terminator	Pin 21	GND
Pin 24	GND	Pin 23	GND
Pin 26	Terminator power	Pin 25	GND
Pin 28	GND	Pin 27	GND
Pin 30	GND	Pin 29	GND
Pin 32	SCSI ATN	Pin 31	GND
Pin 34	GND	Pin 33	GND
Pin 36	SCSI BUSY	Pin 35	GND
Pin 38	SCSI ACK	Pin 37	GND
Pin 40	SCSI Reset	Pin 39	GND
Pin 42	SCSI MSG	Pin 41	GND
Pin 44	SCSI Select	Pin 43	GND
Pin 46	SCSI CD	Pin 45	GND
Pin 48	SCSI REQ	Pin 47	GND
Pin 50	SCSI IO	Pin 49	GND

X2 Utility connector

Pin	Function
Pin 1	VCC
Pin 2	PS/2 Mouse Data
Pin 3	PS/2 Mouse Clock
Pin 4	PS/2 Keyboard Data
Pin 5	PS/2 Keyboard Clock
Pin 6	Ground
Pin 7	Resetinput
Pin 8	Battery Input 3.0 to 3.6V (optional if no onboard battery assembled)
Pin 9	RI Input (activ low)
Pin 10	PowerOn Input (aktiv low)
Pin 11	LID Input (aktiv low)
Pin 12	LAN LED
Pin 13	IrDA RX
Pin 14	IrDA TX
Pin 15	HD LED primary
Pin 16	Speaker Output

X32 Audio Connector

Pin	Function
Pin 1	Speaker Left
Pin 2	GND
Pin 3	Speaker Rigth
Pin 4-8	NC

X26, X27, X28 Video Input connector

Pin	Function
Pin 1	Video
Pin 2	GND

X1 VGA / ZV-Port connector (14pin RM2.54) (if this option is assembled)

Pin	Signal	Pin	Signal
Pin 2	VGA RED	Pin 1	Video IN3
Pin 4	VGA GREEN	Pin 3	Video IN2
Pin 6	VGA BLUE	Pin 5	GND
Pin 8	VGA HSYNCH	Pin 7	Video IN1
Pin 10	VGA VSYNCH	Pin 9	GND
Pin 12	GND	Pin 11	Video Composite In
Pin 14	GND	Pin 13	Nc

X31 Not used connector (6pin RM2.54) (if this option is assembled)

Pin	Signal	Pin	Signal
Pin 2	SMB DATA	Pin 1	SMB CLOCK
Pin 4	GPI 14	Pin 3	GPI 13
Pin 6	GND	Pin 5	+5V

X34 MIDI / GAME Port

Pin	Function
Pin 1	VCC
Pin 2	Game Port A1
Pin 3	Game Port AX
Pin 4	GND
Pin 5	GND
Pin 6	Game Port AY
Pin 7	Game Port A2
Pin 8	VCC
Pin 9	VCC
Pin 10	Game Port B1
Pin 11	Game Port BX
Pin 12	MIDI_OUTPUT
Pin 13	Game Port BY
Pin 14	Game Port B2
Pin 15	MIDI_INPUT
Pin 16	not connected

X33 Audio Interface

Pin	Function
Pin 1	Speaker Stereo Output 1W Left
Pin 2	Speaker Stereo Output 1W Rigth
Pin 3	AUX_Left Input
Pin 4	AUX_Rigth Input
Pin 5	Microphone Input Mono
Pin 6	Line_Left Output
Pin 7	Line_Rigth Output
Pin 8	Analog Ground

X11 Dual USB Connector

Pin	Function
Pin 1	VCC
Pin 2	USB Port 0-
Pin 3	USB Port 0+
Pin 4	Ground
Pin 5	VCC
Pin 6	USB Port 1-
Pin 7	USB Port 1+
Pin 8	Ground

X128 CPU FAN Connector

Pin	Function
Pin 1	+5V
Pin 2	Speed sens TTL input
Pin 3	GND

5.1 Jumpers on this MICROSPACE product

Jumper Locations on the Board

The figure shows the location of all the jumper blocks on the PCC-P5 board. The numbers shown in this figure are silk screened on the board so that the pins can easily be located. This chapter refers to the individual pin for these jumpers. The default jumper settings are written in bold. Be careful when you change some jumpers. Some jumpers are soldering jumpers, you need a miniature soldering station.

The Jumpers of the MSLB-P5 V1.0 (RM2.54mm jumpers)

	CPU Speed select	J86	J87	J88
166Mhz	**factory settings**			
266Mhz	**factory settings**	**1-2**	**1-2**	**2-3**

J7	Clock mode	1-2 =	2-3 =
J91	RTC reset	**close = run**	open = reset
J102	SCSI-2 termination	1-2 = enable	**2-3 = auto**
J108	COM3 interface	1-2 = RS485	**2-3 = RS232**
J109	COM4 interface	1-2 = RS485	**2-3 = RS232**
J114	COM3 interface	1-2 = IRQ3	**2-3 = IRQ10**
J115	COM4 interface	1-2 = IRQ4	**2-3 = IRQ11**
J133	Comp.Flash Select	Open = master	Close = Slave
J55	VDD Save voltage	1-2 = 12V	**2-3 = 5V**
J81	VGA controller configuration	**open**	close = GND

J91	Source the RTC from 3.3V for batteryless systems only	close = low batteryless	**open = battery assembled**

J113	Audio configuration EEPROM		
J112	Audio configuration Mode	1-2 = ext.mode	**2-3 = std.mode**
J103	Audio configuration GPO		

J124	COM3 Receiver RS422	1-2 = RTS3 control	2-3 = hard enable
J125	COM3 Transmitter RS422	1-2 = DTR3 control	2-3 = hard enable
J126	COM4 Receiver RS422	1-2 = RTS3 control	2-3 = hard enable
J127	COM4 Transmitter RS422	1-2 = DTR3 control	2-3 = hard enable

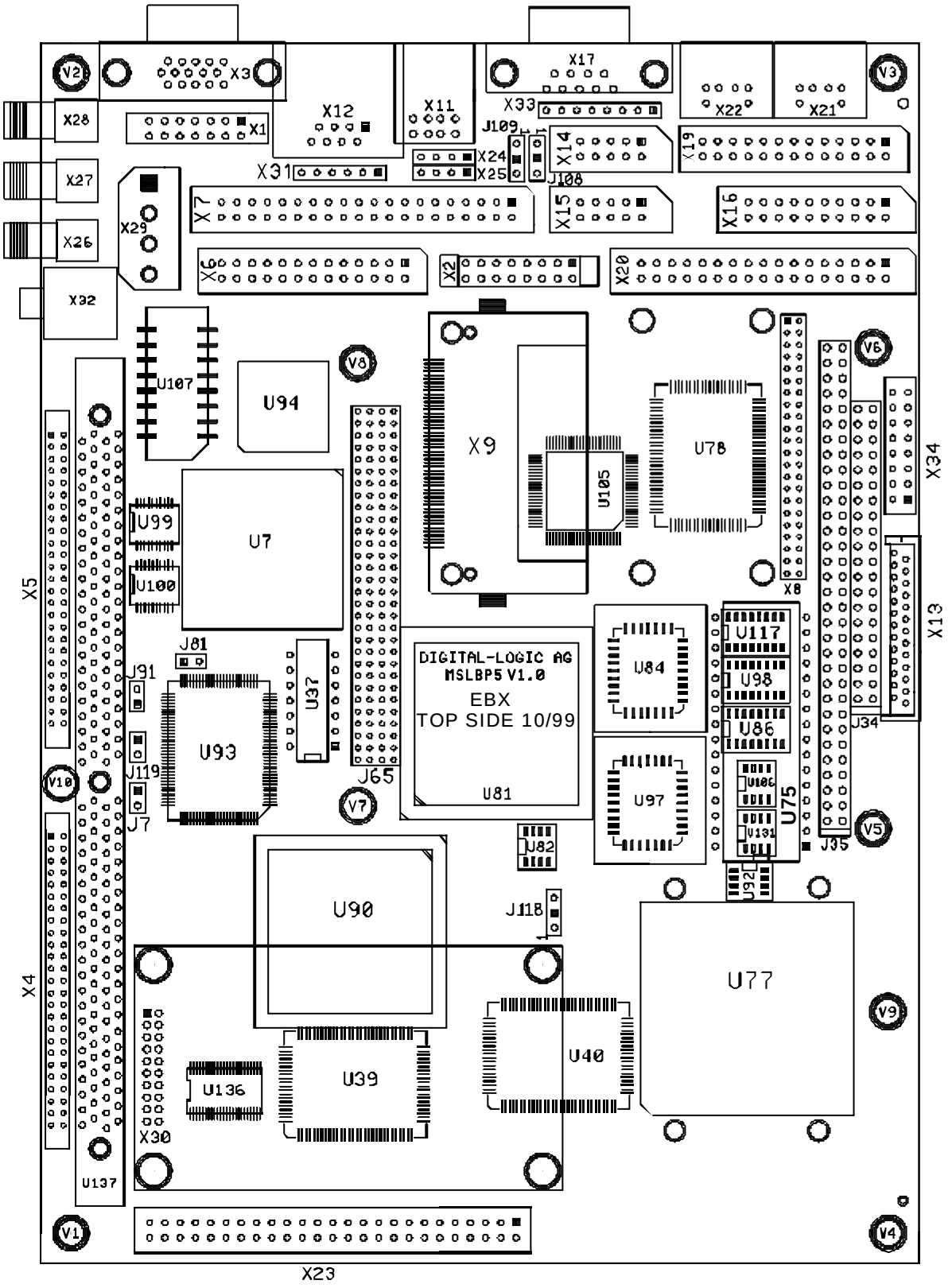
J118	COM2 Select	1-2 = GPD Modul	**2-3=RS232**
J119	GPS Battery Backup	**open=passiv**	close=activ
J120	GPS Preamp	**open=passiv**	close=activ
J121	GPS GPIO2	**open=passiv**	close=activ
J122	GPS GPIO3	**Open = passiv**	Close = activ

J116	Video SDA	1-2 = GP0	2-3 = GP2
J117	Video SCL	1-2 = GP1	2-3 = GP3
J129	Video BIOS Select 0	1-2 = high	2-3 = low
J130	Video BIOS Select 1	1-2 = high	2-3 = low
J131	Video BIOS Select 2	1-2 = high	2-3 = low
J132	Flash select	Open = disable	Close = Enable

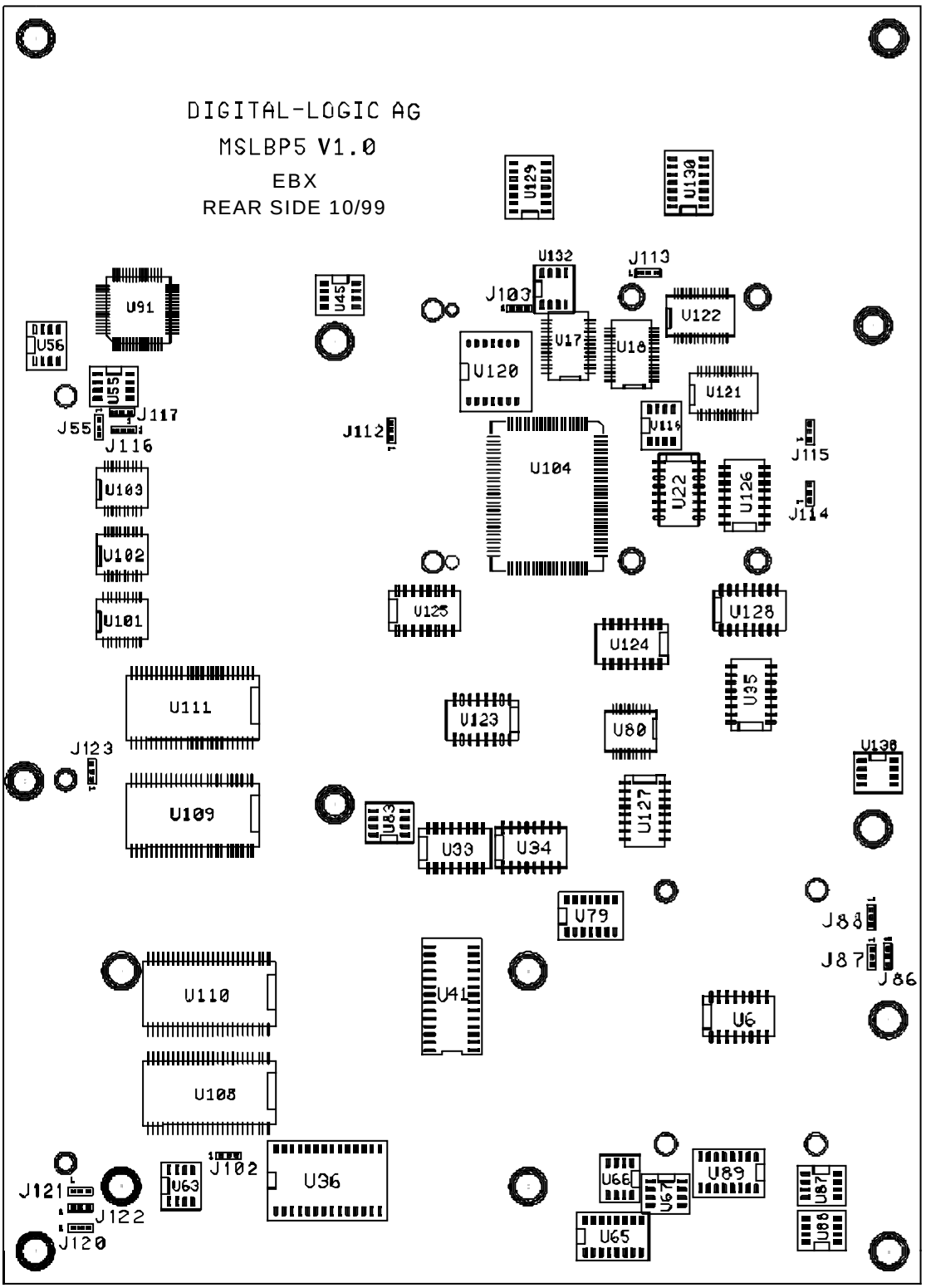
Settings written in bold are defaults!

5.2 Jumper and Connector Locations

5.2.1 Top-view



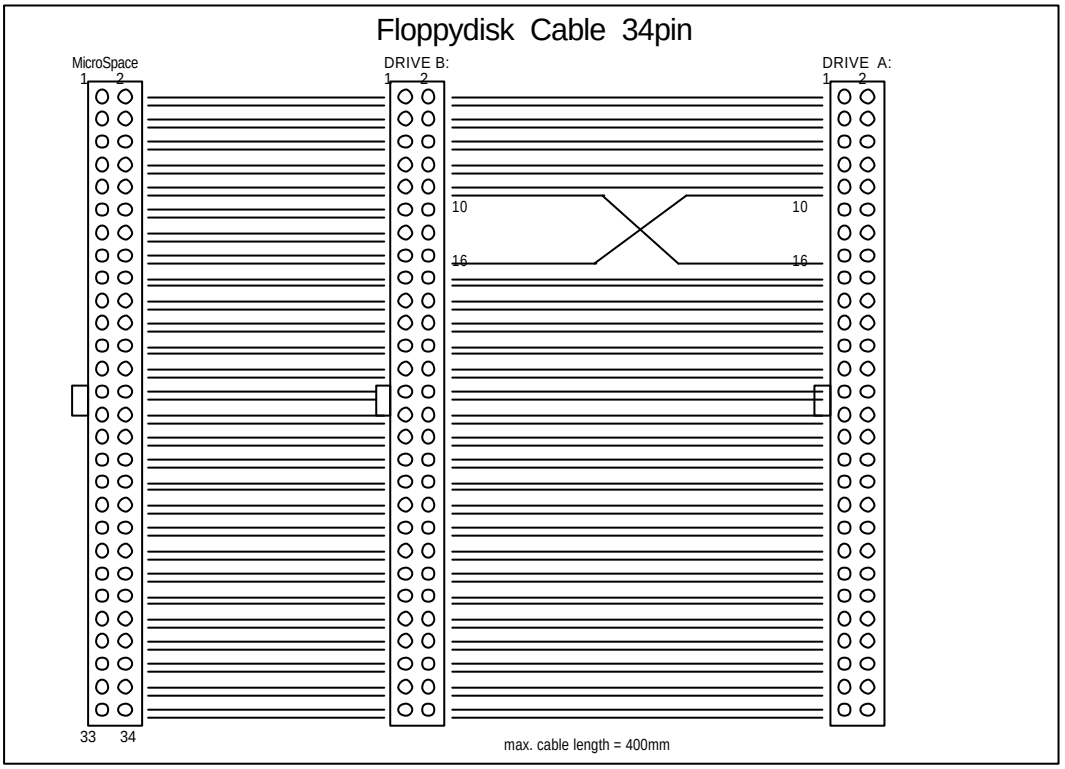
5.2.2 Bottom-view



6 CABLE INTERFACE

6.1 The Floppy Disk Cable

IDT Terminal for Dual Row 0.1" (2.54 mm grid) and 1.27 mm flat cable



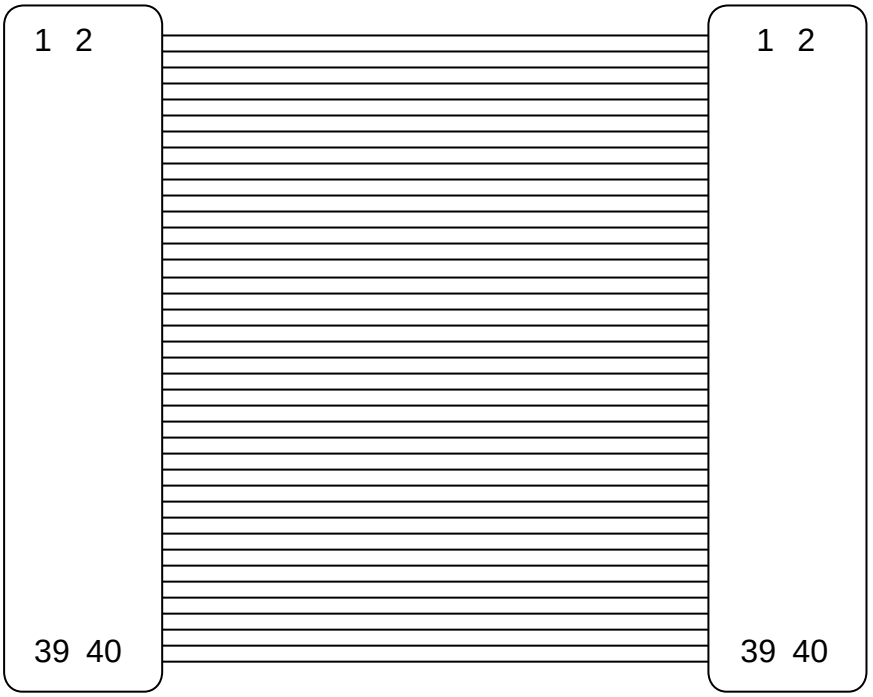
For drive A: the lines 10 to 16 are crossed (180 degrees).

All floppy drives must be selected as drive number 2, because the cable assigns the drive letter A: or B: to the drives. The power must be connected separately. Refer to the technical manual of the floppy drives used.

The last drive must be terminated with 1 Kohms. Do not use 150 ohms terminated floppy drives!

6.2 The Harddisk Cable 40 pins

IDT terminal for dual row 0.1" (2.54 mm grid) and 1.27 mm flat cable. 40 pins signal, power is separately wired. Refer to the technical manual of the harddisk used.



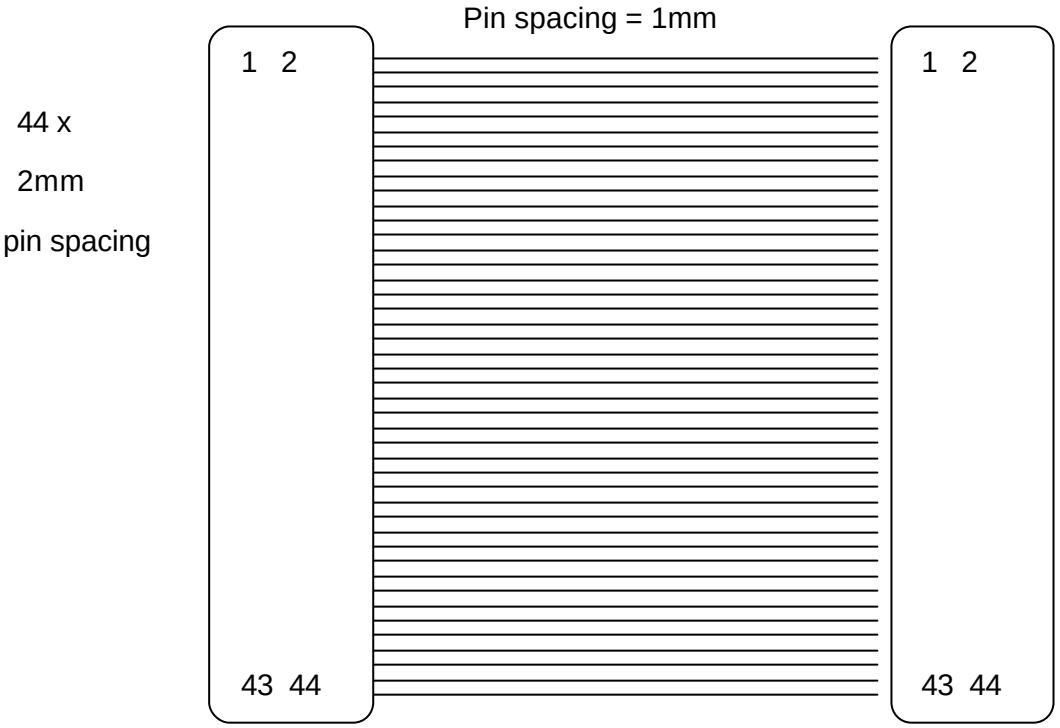
Max. length for IDE cable is 30 cm.

ATTENTION:

A maximum of two IDE drives can be connected to the HD-Interfaces. The first drive must always be the MASTER drive (= C:) and the second is the SLAVE drive. Check the selection of the drive in the technical manual. An inverse connection could destroy the drive or the MICROSPACE PCC-P5. Be very careful. There is no warranty in this case.

6.3 The Harddisk Cable 44 pins

IDT Terminal for Dual Row (2.00 mm grid) and 1.00 mm flat cable. 44 pins = 40 pins signal and 4 pins power.

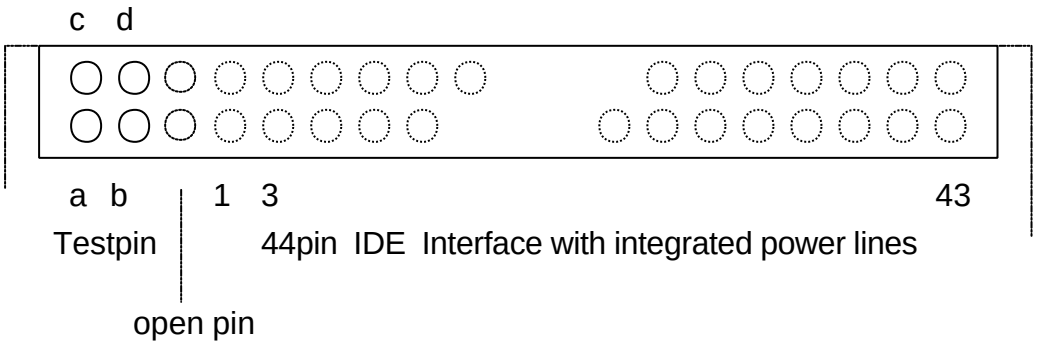


Max. length for the IDE cable is 30 cm.

ATTENTION:

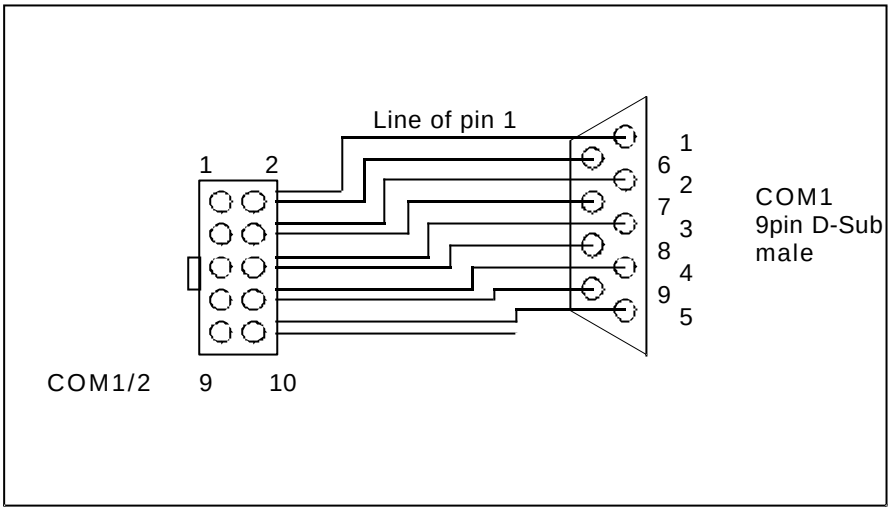
Check the pin 1 marker of the cable and the connector before you power-on. See the technical manual of the drives used, because a wrong cable will immediately destroy the drive and/or the MICROSPACE PCC-P5L board. There is no warranty in this case. Without the technical manual you cannot connect this type of drive.

The 44 pins IDE connector on the drives are normally composed of the 44 pins and 2 open pins and 4 test pins, total: 50 pins. Leave the 4 test pins unconnected .



6.4 The COM 1/2 Serial Interface Cable

DT terminal for dual row 0.1" (2.54 mm grid) and 1.27 mm flat cable



ATTENTION:

- Do not short-circuit these signal lines.
- Never connect any pins either to the same plug or to any other plug on the MICROSPACE PCC-P5L. The +/-10Volt will destroy the MICROSPACE core logic immediately. No warranty in this case!
- Do not overload the output: max. output current of Maxim converters: 10 mA
- The maximum supply current for the mouse is ~ 5mA!

7 SPECIAL PERIPHERALS, OPTIONAL FUNCTIONS

7.1 Special Peripherals

8 100/10 ETHERNET LAN

Required programs and drivers are located in the directory \DRIVERS\NETWORK\82559\

Create a directory C:\LAN100 on your harddisk. Copy the programs and drivers of \DRIVERS\NETWORK\82559\ onto your HD.

1. Load datas into EEPROM. Load with the command „EEUPDATE –ALL LAN100.EEP LAN100.DAT“ the datas of both files LAN100.EEP and LAN100.DAT into the EEPROM (in directory C:\LAN100\UTILITY\E2PROM)
2. Run SETUP.EXE in C:\LAN100.
   Choose Install Network Drivers
   Novell
   DOS ODI Client
   Prefered Server (optional)
   Frame Type 802.2
   Press F10 then select the Name of the Directory for example C:\NETWORK
   Choose Modify AUTOEXEC.BAT to run automatically.
   Exit Setup YES

Change STARTPRO.BAT of C:\NETWORK\ in:
C:
CD \NETWORK
LSL
E100BODI
IPXODI
VLM
CD \

Table LAN100.EEP:

A000 01C9 2345 0000 0000 0301 0701 0000
6494 3903 40C0 0003 8086 0002 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 CHKS

Table LAN100.DAT:

00AA00000000
00AA00000001
00AA00000002
00AA00000003
00AA00000004

Test the transfer with HDTEST with a disk drive on your server!

Driver Installation Windows 95

Copy Intel Pro100+ driver on HD. De-install all networkdrivers under Windows95 in system and software folders and restart Windows. Networkcard Pro10 PCI will be auto-detected and all drivers can be installed (of the copied directory).
Restart Windows and ... failure messages.
Call up the networkcard-preferences in the ... Systemsteuerung and update the drivers.
Choose the Intel 8255x-based PCI Ethernet Adapter (10/100) out of the list.
Shut down Windows and switch of the power supply unit. Boot up again and the net is installed.

9 SCSI INTERFACE (OPTION)

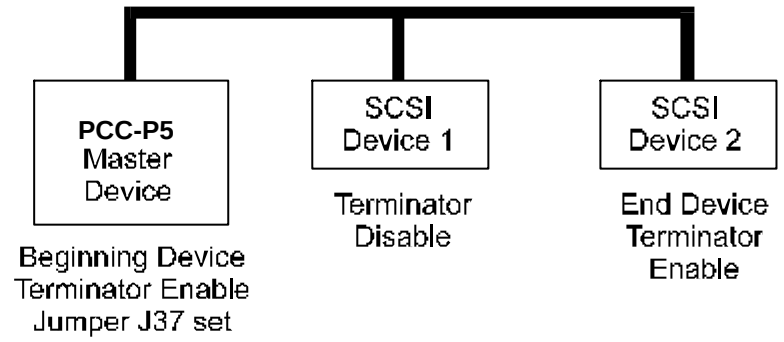
The SCSI-2 interface is realized with the AIC-7880 controller. This chip is fully supported by Adaptec Device Management System (SDMS) software, that supports the Advanced SCSI Protocol Interface (ASPI) and the ANSI Common Access Method (CAM). The AIC-7880 operates the SCSI bus at 5 MB/s asynchronously or 10 MB/s synchronously, and bursts data to the host at full PCI speed up to 110 MB/s (at 33 MHz).

The SCSI-2 controller has full PnP recognition, a 64-Byte DMA FIFO and all SCSI signals are ESD protected up to 2 kV.

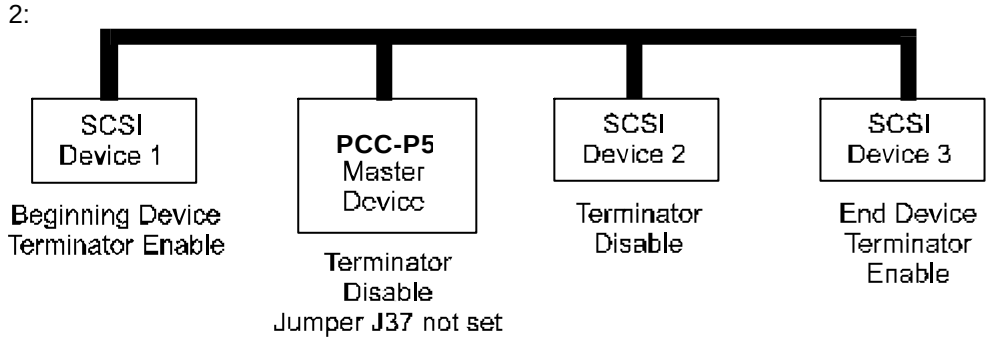
The user can connect up to 7 SCSI devices to the AIC-7880, all the SCSI devices are daisy chained trough one SCSI-Bus cable. This cable must have terminators at both ends; i.e. the beginning device and end device. Without proper installation of the terminators it will cause a SCSI devices malfunction. On the PCC-P5L board the terminator is active, when jumper J102 is set to 1-2. On position 2-3 the termination will be auto dedected.

The following figures illustrate where the terminators can be placed:

Example 1:



Example 2:



The boot devices on the SCSI-2 interface:

- any SCSI-2 harddisk
- CD drives
- Syquest removable media
- Tapes, MO- and ZIP-Drives

9.1 SCSI Drivers for operating system support

10 SOUNDPORT DRIVER INSTALLATION

10.1 ESS1869

To download the latest drivers at http://www.digitallogic.ch

10.2 Driver for WIN 3.11

Insert the DRIVER DISK FOR WIN3.1, after starting windows V3.1. Run „SETUP.EXE“ on the driver floppydisk A:. Choose „INSTALL“ for all the files to be loaded and system files to be modified. The setup utility allows a different directory to be choosen other than default „ C:\ADISOUND“ for copying. Select „CONTINUE“ to begin copying the files from the disk. After the setup has completed type „EXIT“, restart Windows. You may configure the device before restarting Windows by choosing „CONFIG“. This „ESS1869 I/O Configuration“ allows the user to configure the device to something other than default.

After exiting setup, other configuration changes must be made using the program item drivers under ESS1869 Control Panel !

In the Control Panel for the ESS1869 the following setting may be made:

Windows Sound System:

Base Port:	Address 800 - 807h	
IRQ	IRQ5	
DMA Play	DMA 01	
DMA Rec.	DMA 00	

MPU401:

Base Port:	Address 330-331h	
IRQ:	IRQ5	

Sound Blaster System:

Base Port:	220-22fh	

Others:

OPL3 Port:	388h-38Bh	
Game Port:	201h-201h	(not free on MSM-P5!)

After exiting setup and restarting Windows V3.1 the drivers will be loaded. If there is an I/O, DMA or IRQ conflict between the audio drivers and other devices in the system, use the Willow Pond Universal SoundComm Driver setup to change any setting, it is located in „drivers“ under Control Panel. The MPU-401 I/O address and IRQ settings are located in the Roland MPU-401 driver.

10.3 Driver for WIN 95

Win95 will recognize the new hardware by displaying a dialog box „New Hardware Found ESS1869“. Insert DRIVER DISK WIN95 into the floppydisk A:. Select the option „Driver from disk provided by hardware manufacturer“, and hit ENTER. In the next windows select the default driver and hit OK. Select the default driver also for the GAMEPORT. The DRIVER DISK WIN95 must be in the floppy drive and the correct drivers are loaded. Remove the disk and restart the system when prompted. Your soundcard is now active.

10.4 Driver for NT4.0

NT will recognize the new hardware by displaying a dialog box saying „New Hardware Found ESS1869“. Insert DRIVER DISK NT4.0 into the floppydisk A:. Select the option „Driver from disk provided by hardware manufacturer“, and hit ENTER. On the next windows select the default driver and hit OK. Select the default driver also for the GAMEPORT. The DRIVER DISK NT must be in the floppy drive and the correct drivers are loaded. Remove the disk and restart the system when prompted. Your soundcard is now active.

10.5 Bundled Applications, MediaRack

Insert the Disk witch contains all the bundled and optional applications.
Run a:\setup.exe (for Win 3.11, Win95, NT4.0).
The installer will create a program group called „bundled applications“ and „optional applications“.

Included applications:

- MEDIA RACK
- MEDIA LAUNCHER
- WAVE SHAPER
- DOC TALKER TTS and NOTE TALKER TTS
- KARAOKE PRODUCER
- PRESTO ARRANGER

11 INSTALLING THE FLASHDISK DOC2000

On the SSD 36pin socket a DiskOnChip DOC2000 module from M-Systems may be installed with a capacity of 512k to 12MByte. This device is available from DIGITAL-LOGIC AG.

Operating Systems:

DOS, DL-DOS, RTX-DOS, WIN 3.11, ROM-WIN are working with these drives.
All other non DOS compatible systems need a driver.

Give attention to the pin 1 orientation in the 32pin SSD socket.

11.1 Enabling and Formatting of the DiskOnChip-Modules

Enabling:

No handlings need.

Format:

1. Boot up from the standard floppydisk A: or from a harddisk.

2. Enter the tooldisk from M-Systems containing the formattool DFORMAT.EXE

3. Start format utility
   The screen should inform about the status of the flashdisk.

4. Enter the DOS-Bootdisk and transfer the bootfiles with SYS A: C:
   From this moment, the flashdisk is now the bootable drive C: and if any harddisk is conencted it changes to letter D: and E:

12 BUILDING A SYSTEM

To build a system based on the PCC-P5L board you must obtain the following equipment:

- A chassis to hold all the system components. The board size is designed for EURO-RACKS 19" with 6HE.
- A power supply of 5V and 5- 10 Amps depending on the mass storage systems.
- 8 ohm speaker.
- A floppy disk drive (3,5" or 5,25") with a PC floppy cable (34 pin). You need at least one floppy to boot the first time.
- A harddisk IDE 3,5" or 2,5" or 1,8" with the appropriate cable (44 pin and 2mm grid or 40pin 2.54mm)
- Connect an LCD or a monitor to the VGA connector.
- An PS2-compatible keyboard.

12.1 Starting up the System

Power-up the system and wait for the BIOS to show the BIOS activity on the screen. The BIOS diagnoses the system and displays the size of the memory being tested.

CMOS-SETUP

If the CMOS configuration is incorrect, the BIOS tells you to enter the setup screen with <DEL>.
Select the correct options with the arrow keys and save.

12.2 Error on boot time

A. If the display works:

1. Check if you have a bootable floppy or harddisk.
2. Check the CMOS parameter with the setup tools.
3. Reset the CMOS RAM with the reset jumper J91 on the board. Close Jumper J91 for 5 seconds until the power-on procedure. Remove the jumper and the system will start with BIOS-defaults.
4. Re-enter the correct values with Setup.

B. If no display on the screen is available:

1. Check the power circuitry.
2. Check the polarities of the cables.
3. Measure the voltage of the power supply under load and offload.
4. Measure the current between the supply and the MICROSPACE PC.
5. Connect a floppy: does the bezel led light blink?
6. Does the harddisk spindle motor start?
7. Reset the CMOS-RAM: see A.3.

C. If the error appears again

1. Contact your nearest DIGITAL-LOGIC dealer for Technical Support.
2. Or fill out the support request form (SRF) on the Internet: http://www.digitallogic.ch

13 DIAGNOSTICS

Check point Description

Uncompressed INIT code check-points

D0	NMI is Disabled. CPU ID saved. Init code Checksum verification starting.
D1	To do DMA init, Keyboard controller BAT test, start memory refresh and going to 4GB flat mode.
D3	To start Memory sizing.
D4	To come back to real mode. Execute OEM patch. Set stack.
D5	E000 ROM enabled. Init code is copied to segment 0 and control to be transfered to segment 0.
D6	Control is in segment 0. To check <CTRL><HOME> key and verify main BIOS checksum..
D7	Main BIOS runtime code is to be decompressed and control to be passed to main BIOS in shadow RAM.

Boot Block Recovery Code check-points

E0	Onboard Floppy Controller (if any) is initialized. To start base 512K memory test.
E1	To initialize interrupt vector table.
E2	To initialize DMA and interrupt controllers.
E6	To enable floppy and timer IRQ, enable internal cache.
ED	Initialize floppy drive.
EE	Start looking for a diskette in drive A: and read 1st sector of the diskette.
EF	Floppy read error.
F0	Start searching 'AMIBOOT.ROM' file in root directory.
F1	'AMIBOOT.ROM' file not present in root directory.
F2	Start reading FAT table and analyse FAT to find the clusters occupied by 'AMIBOOT.ROM' file.
F3	Start reading 'AMIBOOT.ROM' file cluster by cluster.
F4	'AMIBOOT.ROM' file not of proper size.
F5	Disable internal cache.
FB	Detect Flash type present.
FC	Erase Flash.
FD	Program Flash.
FF	Flash program successful. BIOS is going to restart.

Runtime code is uncompressed in F000 shadow ram

03	NMI is Disabled. To check soft reset/power-on.
05	BIOS stack set. Going to disable Cache if any.
06	POST code to be uncompressed.
07	CPU init and CPU data area init to be done.
08	CMOS checksum calculation to be done next.
0B	Any initialization before keyboard BAT to be done next.
0C	KB controller I/B free. To issue the BAT command to keyboard controller.
0E	Any initialization after KB controller BAT to be done next.
0F	Keyboard command byte to be written.
10	Going to issue Pin-23,24 blocking/unblocking command.
11	Going to check pressing of <INS> , <END> key during power-on.
12	To init CMOS if "Init CMOS in every boot" is set or <END> key is pressed. Going to disable DMA and IRQC
13	Video display is disabled and port-B is initialized. Chipset init about to begin.
14	8254 timer test about to start.
19	About to start memory refresh test.
1A	Memory Refresh line is toggling. Going to check 15us ON/OFF time.
23	To read 8042 input port and disable Megakey GreenPC feature. Make BIOS code segment writeable.
24	To do any setup before Int vector init.
25	Interrupt vector initialization about to begin. To clear password if necessary.
27	Any initialization before setting video mode to be done.
28	Going for monochrome mode and color mode setting.
2A	Different BUSes init (system, static, output devices) to start if present.
2B	To give control for any setup required before optional video ROM check.
2C	To look for optional video ROM and give control.
2D	To give control to do any processing after video ROM returns control.
2E	If EGA/VGA not found then do display memory R/W test.
2F	EGA/VGA not found. Display memory R/W test about to begin.
30	Display memory R/W test passed. About to look for the retrace checking.
31	Display memory R/W test or retrace checking failed. To do alternate Display memory R/W test.
32	Alternate Display memory R/W test passed. To look for the alternate Display retrace checking.
34	Video display checking over. Display mode to be set next.
37	Display mode set. Going to display the power on message.
38	Different BUSes init (input, IPL, general devices) to start if present.
39	Display different BUSes initialization error messages. *(Please see Appendix for details of different BUSes.)*
3A	New cursor position read and saved. To display the Hit <DEL> message.
40	To prepare the descriptor tables.

42	To enter in virtual mode for memory test.
43	To enable interrupts for diagnostics mode.
44	To initialize data to check memory wrap around at 0:0.
45	Data initialized. Going to check for memory wrap around at 0:0 and finding the total system memory size.
46	Memory wrap around test done. Memory size calculation over. About to go for writing patterns to test memory.
47	Pattern to be tested written in extended memory. Going to write patterns in base 640k memory.
48	Patterns written in base memory. Going to find out amount of memory below 1M memory.
49	Amount of memory below 1M found and verified. Going to find out amount of memory above 1M memory.
4B	Amount of memory above 1M found and verified. Check for soft reset and going to clear memory below 1M for soft reset. (If power on, go to check point No. 4Eh.)
4C	Memory below 1M cleared. (SOFT RESET) Going to clear memory above 1M.
4D	Memory above 1M cleared. (SOFT RESET) Going to save the memory size. (Goto check point No. 52h).
4E	Memory test started. (NOT SOFT RESET) About to display the first 64k memory size.
4F	Memory size display started. This will be updated during memory test. Going for seq. and random memory test.
50	Memory testing/initialization below 1M complete. Going to adjust displayed memory size for relocation/shadow.
51	Memory size display adjusted due to relocation/ shadow. Memory test above 1M to follow.
52	Memory testing/initialization above 1M complete. Going to save memory size information.
53	Memory size information is saved. CPU registers are saved. Going to enter in real mode.
54	Shutdown successful, CPU in real mode. Going to disable gate A20 line and disable parity/NMI.
57	A20 address line, parity/NMI disable successful. Going to adjust memory size depending on relocation/shadow.
58	Memory size adjusted for relocation/shadow. Going to clear Hit <DEL> message.
59	Hit <DEL> message cleared. <WAIT...> message displayed. About to start DMA and interrupt controller test.
60	DMA page register test passed. To do DMA#1 base register test.
62	DMA#1 base register test passed. To do DMA#2 base register test.
65	DMA#2 base register test passed. To program DMA unit 1 and 2.
66	DMA unit 1 and 2 programming over. To initialize 8259 interrupt controller.
7F	Extended NMI sources enabling is in progress.
80	Keyboard test started. Clearing output buffer, checking for stuck key, to issue keyboard reset command.
81	Keyboard reset error/stuck key found. To issue keyboard controller interface test command.
82	Keyboard controller interface test over. To write command byte and init circular buffer.
83	Command byte written. Global data init done. To check for lock-key.
84	Lock-key checking over. To check for memory size mismatch with CMOS.
85	Memory size check done. To display soft error and check for password or bypass setup.
86	Password checked. About to do programming before setup.
87	Programming before setup complete. To uncompress SETUP code and execute CMOS setup.
88	Returned from CMOS setup program and screen is cleared. About to do programming after setup.
89	Programming after setup complete. Going to display power on screen message.
8B	First screen message displayed. <WAIT...> message displayed. PS/2 Mouse check and extended BIOS data area allocation to be done
8C	Setup options programming after CMOS setup about to start.
8D	Going for hard disk controller reset.
8F	Hard disk controller reset done. Floppy setup to be done next.
91	Floppy setup complete. Hard disk setup to be done next.
95	Init of different BUSes optional ROMs from C800 to start. *(Please see Appendix-I for details of different BUSes.)*
96	Going to do any init before C800 optional ROM control.
97	Any init before C800 optional ROM control is over. Optional ROM check and control will be done next.
98	Optional ROM control is done. About to give control to do any required processing after optional ROM returns control and enable external cache.
99	Any initialization required after optional ROM test over. Going to setup timer data area and printer base address.
9A	Return after setting timer and printer base address. Going to set the RS-232 base address.
9B	Returned after RS-232 base address. Going to do any initialization before Coprocessor test.
9C	Required initialization before Coprocessor is over. Going to initialize the Coprocessor next.
9D	Coprocessor initialized. Going to do any initialization after Coprocessor test.
9E	Initialization after Coprocessor test is completed. Going to check extd keyboard, keyboard ID and num-lock.
A2	Going to display any soft errors.
A3	Soft error display complete. Going to set keyboard typematic rate.
A4	Keyboard typematic rate set. To program memory wait states.
A5	Going to enable parity/NMI.
A7	NMI and parity enabled. Going to do any initialization required before giving control to optional ROM at E000.
A8	Initialization before E000 ROM control over. E000 ROM to get control next.
A9	Returned from E000 ROM control. Going to do any initialization required after E000 optional ROM control.
AA	Initialization after E000 optional ROM control is over. Going to display the system configuration.
AB	To uncompress DMI data and execute DMI POST init.
B0	System configuration is displayed.
B1	Going to copy any code to specific area.
00	Copying of code to specific area done. Going to give control to INT-19 boot loader.

APPENDIX

The system BIOS gives control to the different BUSes at following check-points to performe various tasks on the different BUSes.

CHECK-POINT	DESCRIPTION OF CHECK-POINT
2A	Different BUSes init (system, static, output devices) to start if present.
38	Different BUSes init (input, IPL, general devices) to start if present.
39	Display different BUSes initialization error messages.
95	Init of different BUSes optional ROMs from C800 to start.

While control is inside the different BUS routines, additional check-points are output to port 80h as WORD to identify the routines under execution. These are WORD check-points, the LOW BYTE of check-point is the system BIOS check-point from where the control is passed to the different BUS routines and the HIGH BYTE of check-point is the indication of which routine is being executed in different BUSes. The details of HIGH BYTE of these check-points are as follows:

HIGH BYTE XY

the upper nibble 'X' indicates the function# is being executed. 'X' can be from 0 to 7.

0 = func#0, disable all devices on the BUS concerned.
1 = func#1, static devices init on the BUS concerned.
2 = func#2, output device init on the BUS concerned.
3 = func#3, input device init on the BUS concerned.
4 = func#4, IPL device init on the BUS concerned.
5 = func#5, general device init on the BUS concerned.
6 = func#6, error reporting for the BUS concerned.
7 = func#7, add-on ROM init for all BUSes.

the lower nibble 'Y' indicates the BUS on which the different routines are being executed. 'Y' can be from 0 to 5.

0 = Generic DIM (Device Initialization Manager).
1 = Onboard System devices.
2 = ISA devices.
3 = EISA devices.
4 = ISA PnP devices.
5 = PCI devices.

14 BIOS

14.1.1 Main Menu Selections

You can make the following selections on the Main Menu itself. Use the sub menus for other selections.

Feature	Options	Description
System Time	HH:MM:SS	Set the system time.
System Date	MM/DD/YYYY	Set the system date.
Diskette 1 Diskette 2	360 kB, 5 ¼" 1.2 MB, 5 ¼" 720 kB, 3 ½" 1.44/1.25 MB, 3 ½" 2.88 MB, 3 ½" Not installed Disabled	Select the type of floppy-disk drive installed in your system. 1.25 MB is a Japanese media format that requires a 3½" 3-Mode Diskette drive.
System Memory	N/A	Displays amount of conventional memory detected during bootup.
Extended Memory	N/A	Displays the amount of extended memory detected during bootup.

You can set the boot sequence of the bootable drives by selecting Boot Sequence on the Main Menu or opening the Boot Menu..

14.1.2 Masters and Slaves

The **Master** and **Slave** settings on the Main Menu control these types of devices:

Hard-disk drives
Removable-disk drives
CD-ROM drives

*Phoenix*BIOS 4.04 supports up to two **IDE disk adapters**, called **primary** and **secondary** adapters. Each adapter supports one **master drive** and one optional **slave drive** in these possible combinations:

1 Master
1 Master, 1 Slave
2 Masters
2 Masters, 1 Slave
2 Masters, 2 Slaves

There is one IDE connector for each adapter on your machine, usually labelled "Primary IDE" and "Secondary IDE." There are usually two connectors on each ribbon cable attached to each IDE connector. When you have connected two drives to these connectors, the one on the end of the cable is the Master.
When you enter Setup, the Main Menu displays the results of **Autotyping–** information each drive provides about its own size and other characteristics–and how they are arranged as Masters or Slaves on your machine.

Note: Do not attempt to change these settings unless you have an installed drive that does not autotype properly (such as an older hard-disk drive that does not support autotyping).
If you need to change your drive settings, use one of the Master or Slave sub-menu as explained in the following.

Use the legend keys listed on the bottom to make your selections and exit to the Main Menu. Use the following chart to configure the hard disk.

Feature	Options	Description
Type	None 1 to 39 User Auto	1 to 39 fills in all remaining fields with values for predefined disk type. See p. **Fehler! Textmarke nicht definiert.** "Fixed Disk Tables." User prompts user to fill in remaining fields. Auto attempts to fill in the fields automatically.
Cylinders	1 to 2048	Number of cylinders.
Heads	1 to 16	Number of read/write heads.
Sectors/Track	1 to 64	Number of sectors per track.
Landing Zone*	1 to 2048	Number of the cylinder specified as the landing zone for the read/write heads.
Write Precomp*	1 to 2048 None	Number of the cylinder at which to change the write timing.

* IDE drives do not require setting Landing Zone and Write Precomp.
WARNING: Incorrect settings can cause your system to malfunction.

14.1.3 Memory Cache

Enabling **cache** saves time for the CPU by holding data most recently accessed in regular memory (dynamic RAM or DRAM) in a special storage area of static RAM (SRAM), which is faster. Before accessing regular memory, the CPU first accesses the cache. If it does not find the data it is looking for there, it accesses regular memory.
Selecting "Memory Cache" from the Main menu displays a menu like the one shown here. The actual features displayed depend on your system's hardware.

```
        PhoenixBIOS Setup - Copyright 1992-1998 Phoenix Technologies Ltd.
    Main
ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
³                      Memory Cache                        ³   Item Specific Help    ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³                                                          ^ ³                       ³
³ External cache:          [Disabled]                      Û ³ Sets the state of the ³
³                                                          Û ³ external system memory³
³ Cache Interleave:        [Disabled]                      Û ³ cache.                ³
³ Cache Write Back:        [Disabled]                      Û ³                       ³
³ Cache Read Cycles:       [2T]                            Û ³                       ³
³ Cache Write Cycles:      [3T]                            Û ³                       ³
³                                                          Û ³                       ³
³ Cache System BIOS:       [Disabled]                      Û ³                       ³
³ Cache Video BIOS:        [Disabled]                      Û ³                       ³
³ Cache E800 - EFFF:       [Disabled]                      Û ³                       ³
³ Cache E000 - E7FF:       [Disabled]                      Û ³                       ³
³ Cache D800 - DFFF:       [Disabled]                      Û ³                       ³
³ Cache D000 - D7FF:       [Disabled]                      ± ³                       ³
³ Cache C800 - CFFF:       [Disabled]                      ± ³                       ³
³                                                          ± ³                       ³
³                                                          Û ³                       ³
³ Non-cacheable Regions                                    Û ³                       ³
³      Region 0, start:    [    0 kB]                      Û ³                       ³
³      Region 0, size:     [Disabled]                      Û ³                       ³
³      Region 1, start:    [    0 kB]                      Û ³                       ³
³      Region 1, size:     [Disabled]                      _ ³                       ³
³                                                            ³                       ³
ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
  F1  Help    XY Select Item    -/+   Change Values        F9  Setup Defaults
  ESC Exit    [Z Select Menu    Enter Select P Sub-Menu    F10 Previous Values
```

(The current BIOS doesn`t support this function.)

Use the legend keys listed on the bottom to make your selections and exit to the Main Menu.

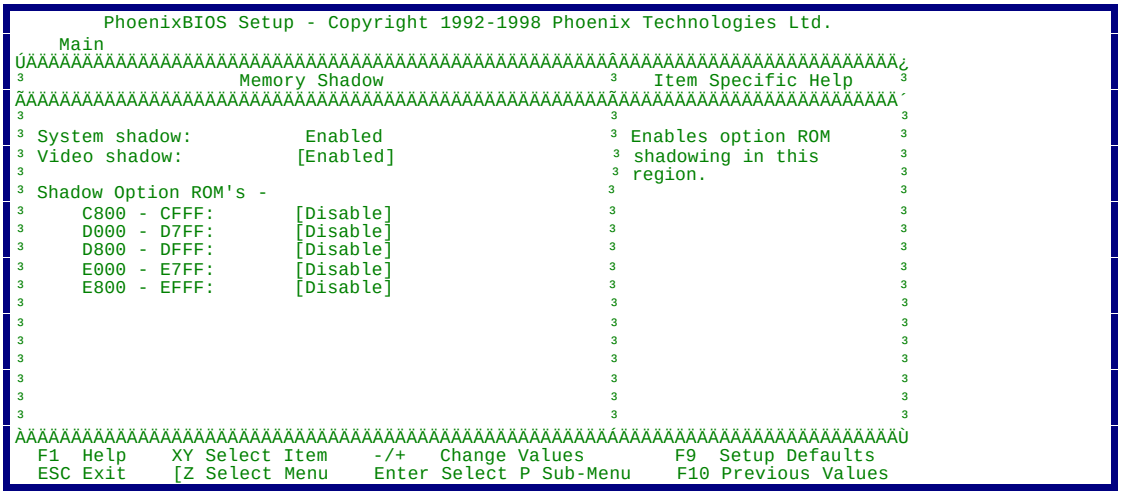
Use the chart on the following page to configure the memory cache.

Feature	Options	Description
External Cache	Enabled Disabled.	Generally enables or disables all memory caching.
Cache Interleave	Enabled Disabled	Interleaving multiple banks of static RAM improves CPU access.
Cache Write Back	Enabled Disabled	Enabled caches both reads and writes to memory. Disabled caches reads only.
Cache Read Cycles	Chipset Dependent	Sets the number of clock pulses for reading from the cache. Shorter number of pulses improves performance.
Cache Write Cycles	Chipset Dependent	Sets the number of clock pulses for writing to the cache. Shorter number of pulses improves performance.
Cache System BIOS	Enabled Disabled	Caches the system BIOS and improves performance.
Cache Video BIOS	Enabled Disabled	Caches the video BIOS and improves performance.
Cache segments, e.g., E800-EFFF	Enabled Disabled	Controls caching of individual segments of memory usually reserved for shadowing system or option ROMs
Non-cacheable regions:		Specifies areas of regular and extended memory as non-cacheable regions.
Region 0, start	0 Multiples of 64	Multiples of 64 define start of non-cacheable region 0 in kilobytes.
Region 0, size	Disabled Multiples of 64	Disabling makes this region available for cache. Multiples of 64 define size of non-cacheable region 0 in kilobytes.
Region 1, start	0 Multiples of 64	Multiples of 64 define start of non-cacheable region 1 in kilobytes.
Region 1, size	Disabled Multiples of 64	Disabling makes this region available for cache. Multiples of 64 define size of non-cacheable region 1 in kilobytes.

WARNING: Incorrect settings can cause your system to malfunction.

14.1.4 Memory Shadow

Selecting "System Shadow" or "Video Shadow" from the Main Menu displays a menu like the one shown here. The actual features displayed depend on the capabilities of your system's hardware.



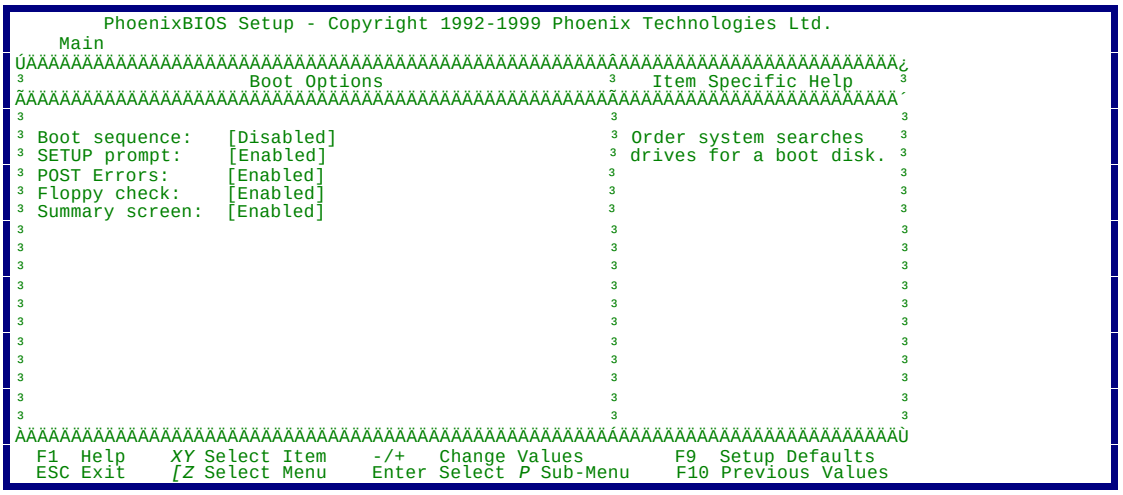
(The current BIOS doesn`t support this function.)

Use the legend keys to make your selections and exit to the Main Menu. Use the following chart to configure memory shadowing.
WARNING: Incorrect settings can cause your system to malfunction.

Feature	Options	Description
System shadow	N/A	Usually permanently enabled.
Video shadow	Enabled Disabled	Shadows video BIOS and improves performance.
Shadow Option ROM	Enabled Disabled	Shadows option ROM located in the specified segments of memory and can improve performance. WARNING: Some option ROMs do not work properly when shadowed.

14.1.5 Boot Sequence

Selecting "Boot Sequence" on the Main Menu displays the Boot Options menu.



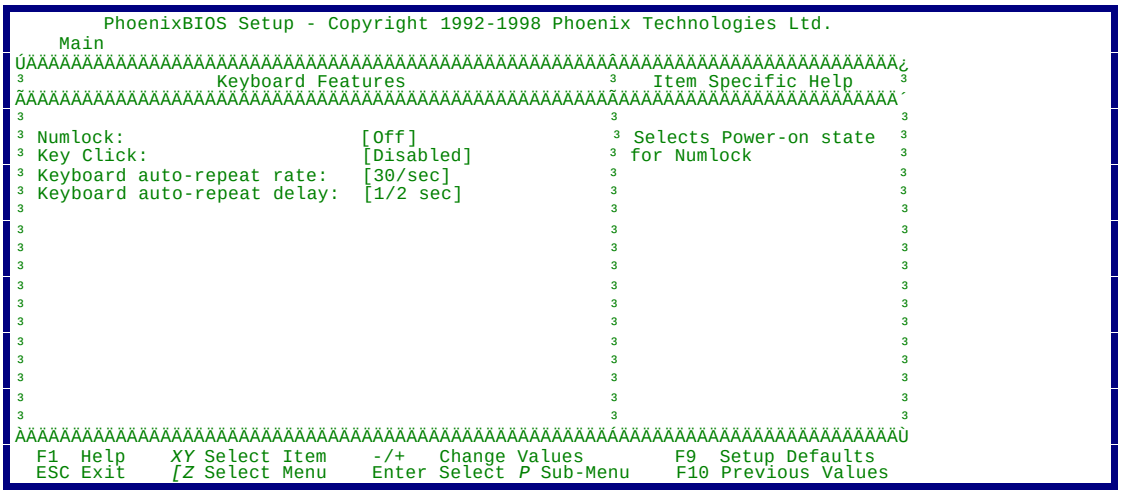
(The current BIOS doesn`t support this function.)

Use the legend keys to make your selections and exit to the Main Menu.
Use the following chart to select your boot options.

Feature	Options	Description
Boot sequence	A: then C; C: then A: C: only	The BIOS attempts to load the operating system from the disk drives in the sequence selected here.
Setup prompt	Enabled Disabled	Displays "Press <F2> for Setup" during bootup.
POST errors	Enabled Disabled	At boot error, pauses and displays "Press <F1> to resume, <F2> to Setup".
Floppy seek	Enabled Disabled	Seeks diskette drives during boo-tup. Disabling speeds boot time.
Summary screen	Enabled Disabled	Displays system summary screen during bootup.

14.1.6 Keyboard Features

Selecting "Numlock" on the Main Menu displays the Keyboard Features menu:



(The current BIOS doesn`t support this function.)

Use the legend keys to make your selections and exit to the Main Menu.
Use the following chart to configure the keyboard features:

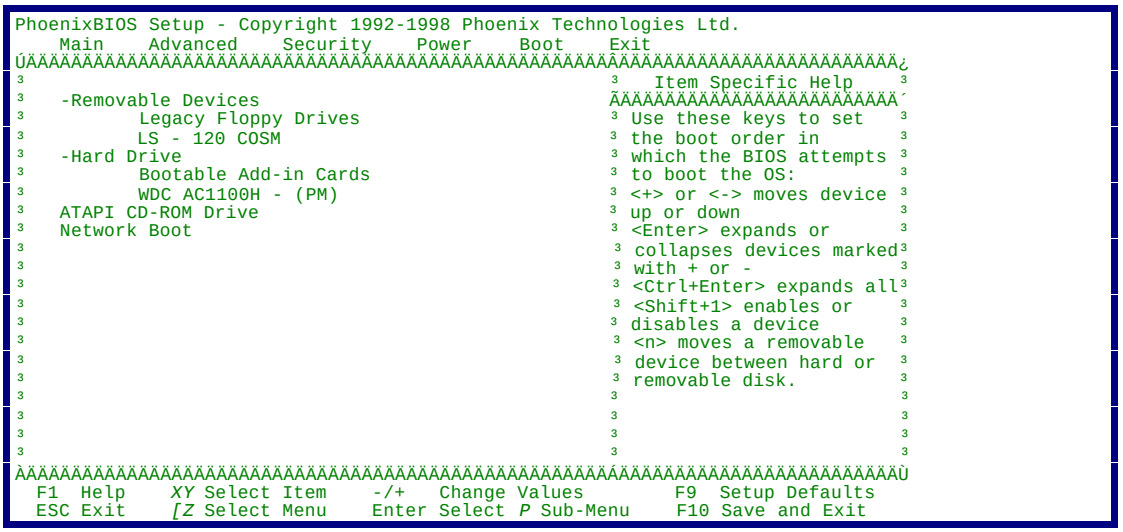
Feature	Options	Description
Numlock	Auto On Off	On or Off turns NumLock on or off at bootup. Auto turns Num-Lock on if it finds a numeric key pad.
Key Click	Enabled Disabled	Turns audible key click on.
Keyboard auto-repeat rate	2/sec 6/sec 10/sec 13.3/sec 21.8/sec 26.7/sec 30/sec	Sets the number of times a sec-ond to repeat a keystroke when you hold the key down.
Keyboard auto-lag delay	¼ sec ½ sec ¾ sec 1 sec	Sets the delay time after the key is held down before it begins to repeat the keystroke.

14.2 Boot Menu

After you turn on your computer, it will attempt to load the operating system (such as Windows 98) from the device of your choice. If it cannot find the operating system on that device, it will attempt to load it from one or more other devices in the order specified in the Boot Menu. Boot devices (i.e., with access to an operating system) can include: hard drives, floppy drives, CD ROMs, removable devices (e.g., Iomega Zip drives), and network cards.

Note: Specifying any device as a boot device on the Boot Menu requires the availability of an operating system on that device. Most PCs come with an operating system already installed on hard-drive C:.

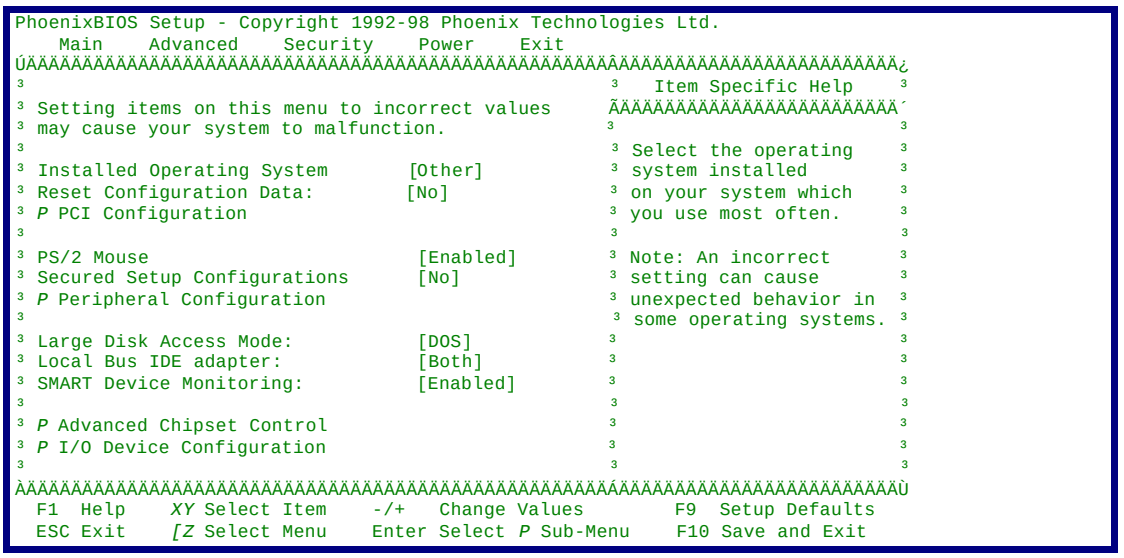
Selecting "Boot" from the Menu Bar displays the Boot menu, which looks like this:



Use this menu to arrange to specify the order of the devices from which the BIOS will attempt to boot the Operating System. Use the <Enter> key to expand or collapse the devices marked with <+> or <->. Press <Ctrl+Enter> to expand all such devices.
To move a device, first select it with the up-or-down arrows, and move it up or down using the <+> and <-> keys. Pressing <n> moves a device between the Removable Devices and Hard Drive. Pressing <Shift+1> enables or disables a device.

14.3 The Advanced Menu

Selecting "Advanced" from menu bar on the Main Menu displays a menu like this:



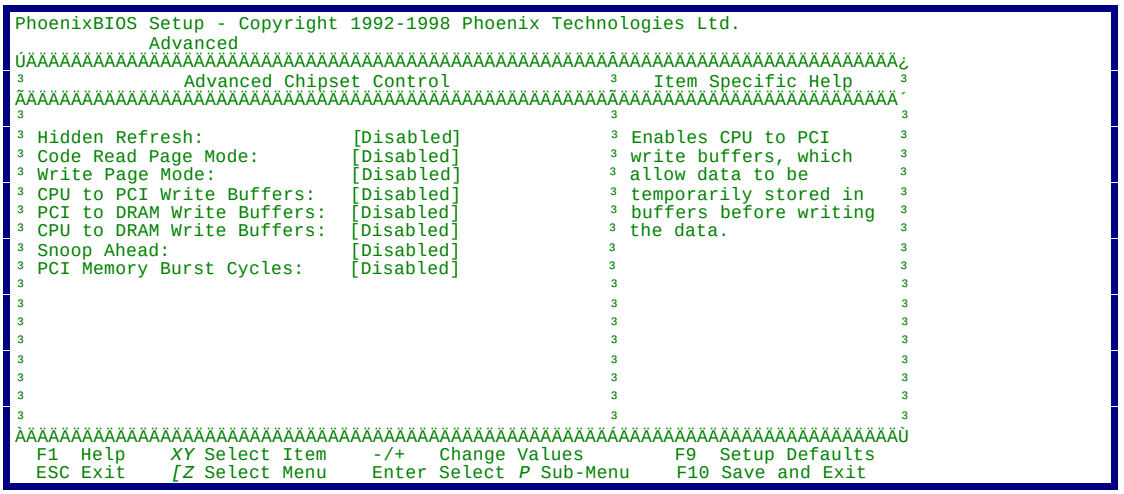
Use the legend keys to make your selections and exit to the Main Menu.

Feature	Options	Description
Installed Operating System	Other Win95 Win98/NT	Select the operating system you use most often.
Reset Configuration Data	Yes No	Yes erases all configuration data in ESCD, which stores the configuration settings for non-PnP plug-in devices. Select Yes when required to restore the manufacturer's defaults.
PS/2 Mouse	Enabled Disabled Auto OS Controlled	Disabled disables any installed PS/2 mouse, but frees up IRQ 12. Auto lets the BIOS control the mouse. OS Controlled lets the operating system control the mouse.
Secured Setup Configurations	Yes No	Yes prevents the Operating System from overriding selections you have made in Setup.
Large Disk Access Mode	DOS Other	Select DOS if you have DOS. Select Other if you have another operating system such as UNIX. A large disk is one that has more than 1024 cylinders, more than 16 heads, or more than 63 tracks per sector.
SMART	Enabled Disabled	Enabled installs Self-Monitoring Analysis-Reporting Technology, which issues a warning if an IDE failure is imminent.

Warning: Incorrect settings can cause your system to malfunction.

14.3.1 Advanced Chipset Control Menu (PCI BIOS)

If the system has a PCI chipset, selecting "Advanced Chipset Control" from the Advanced menu displays a menu like this:



(The current BIOS doesn`t support this function.)

The chipset is a computer chip that acts as an interface between the CPU and the system's hardware. You can use this menu to optimize the performance of your computer.
Use the legend keys to make your selections and exit to the Main Menu.

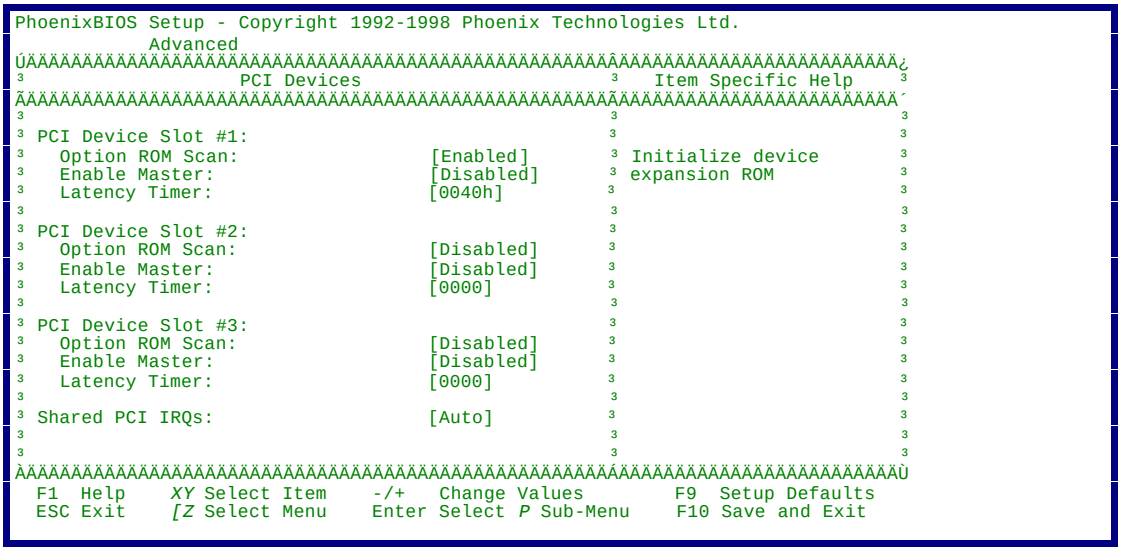
Use the following chart in configuring the chipset:

Feature	Options	Description
Hidden Refresh	Disabled Enabled	Refreshes regular memory without holding up the CPU
Code Read Page Mode	Disabled Enabled	Improves performance when code contains mainly sequential instructions.
Write Page Mode	Disabled Enabled	Improves performance when data is written sequentially.
CPU to PCI Write Buffers	Disabled Enabled	Stores CPU data in buffers before writing to PCI.
PCI to DRAM Write Buffers	Disabled Enabled	Stores PCI data in buffers before writing to DRAM.
CPU to DRAM Write Buffers	Disabled Enabled	Stores CPU data in buffers before writing to DRAM.
Snoop Ahead	Disabled Enabled	Improves PCI bus master access to DRAM.
PCI Memory Burst Cycles	Disabled Enabled	Enables PCI memory burst write cycles.

NOTE: The contents of this menu depend on the chipset installed on your motherboard, and chipsets vary widely. Consult your dealer or the computer manual before changing the items on this menu. **Incorrect settings can cause your system to malfunction.**

14.3.2 PCI Devices Menu

If the system has a PCI bus, selecting "PCI Devices" from menu bar on the Advanced menu displays a menu like this:



(The current BIOS doesn`t support this function.)

PCI Devices are devices equipped for operation with a **PCI** (Peripheral Component Interconnect) **bus**, a standardized hardware system that connects the CPU with other devices. Use this menu to configure the PCI devices installed on your system.
Use the legend keys to make your selections and exit to the Advanced menu.
Use the following chart in configuring the PCI devices:

Feature	Options	Description
PCI Device Slots 1-n:		
Option ROM Scan	Disabled Enabled	Initialize device expansion ROM.
Enable Master	Disabled Enabled	Enables selected device as a PCI bus master. Not every device can function as a master. Check your device documentation.
Latency Timer	0000h to 0280h	Bus master clock rate. A high-priority, high-throughput device may benefit from a greater value.
Shared PCI IRQs	Share One IRQ Share Two IRQs Share Three IRQs Auto	Share ***n*** IRQs: Forces PCI devices to use at most ***n*** IRQs. Auto: Minimizes PCI IRQ Sharing.

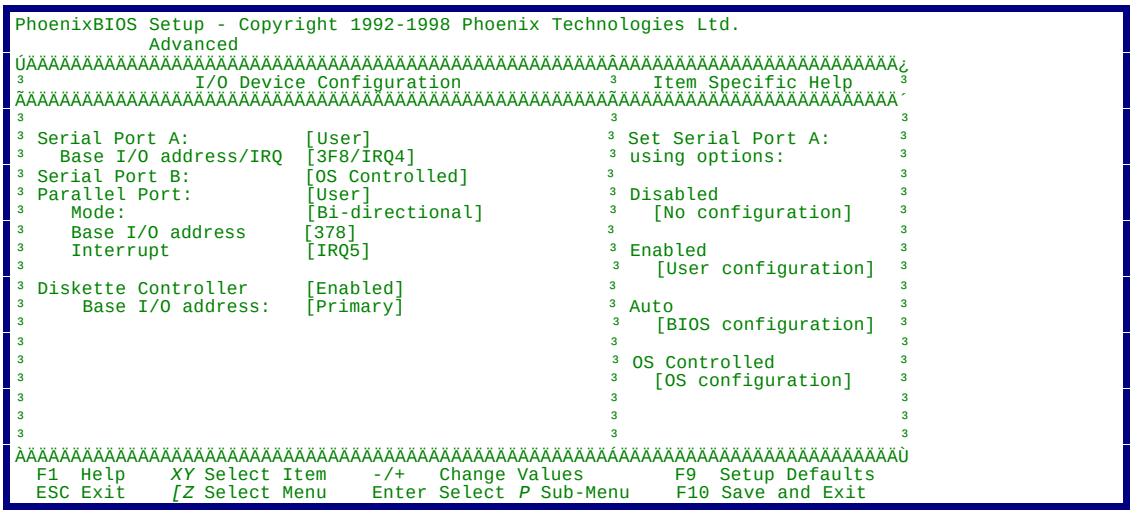
NOTE: The contents of this menu depend on the devices installed on your system. **Incorrect settings can cause your system to malfunction.**

14.3.3 I/O Device Configuration Menu

Most devices on the computer require the exclusive use of **system resources** for operation. These system resources can include Input and Output (I/O) port addresses and Interrupt lines for getting the attention of the CPU. Allocating these resources to various devices is called **device configuration**.

Some systems have devices called **chipsets** that manage a number of things, including the configuration of the serial and parallel ports and the diskette controller. Other systems have, instead, a special I/O chip on the motherboard for configuring these devices.

If your system has a separate on-board I/O chip, select "I/O Device Configuration" on the Advanced Menu to display this menu and specify how you want to configure these I/O Devices:



Use the legend keys to make your selections and exit to the Main Menu.

Use the following chart to configure the Input/Output settings:

Feature	Options	Description
Serial port A: Serial port B:	Disabled Enabled Auto OS Controlled	Disabled turns off the port. Enabled requires you to enter the base Input/Output address and the Interrupt number on the next line. Auto makes the BIOS configure the port automatically during POST. OS Controlled lets the PnP Operating System (such as Windows 95) configure the port after POST.
Base I/O Address/IRQ	3F8, IRQ 4 2F8, IRQ 3	If you select Enabled, choose one of these combinations.
Parallel Port:	Disabled Enabled Auto OS Controlled	Disabled turns off the port. Enabled requires you to enter the base Input/Output address and the Interrupt number below. Auto makes the BIOS autoconfigure the port during POST. OS Controlled lets the PnP Operating System (such as Windows 95) configure the port after POST.
Mode	Output only Bi-directional	Output only is standard one-way protocol for a parallel device. Bi-directional uses two-way protocol of an Extended Capabilities Port (ECP).
Base I/O Address	378 278 3BC	If you select Enabled for the Parallel Port, choose one of these I/O addresses.
Interrupts	IRQ5 IRQ7	If you select Enabled for the Parallel Port, choose one of these interrupt options.
Diskette Controller	Disabled Enabled	Enables the on-board legacy diskette controller. Disabled turns off all legacy diskette drives.
Base I/O Address	Primary Secondary	If you select Enabled for the Diskette Controller, choose Primary for one diskette drive installed or Secondary for two diskette drives installed.

Use this menu to specify how the I/O (Input and Output) ports are configured:

- Manually by you.
- Automatically by the BIOS during POST (See "ROM BIOS Functions" on page **Fehler! Textmarke nicht definiert.**)
- Automatically by a PnP Operating System such as Windows 95 after the Operating System boots.

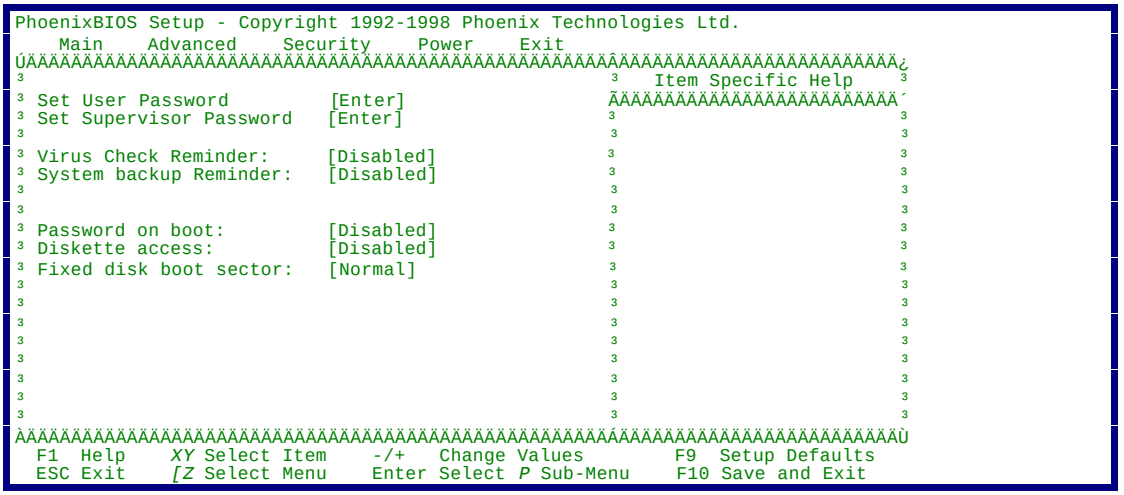
Warning: If you choose the same I/O address or Interrupt for more than one port, the menu displays an asterisk (*) at the conflicting settings. It also displays this message at the bottom of the menu:

```
* Indicates a DMA, Interrupt, I/O, or memory resource conflict with another device.
```

Resolve the conflict by selecting another settings for the devices.

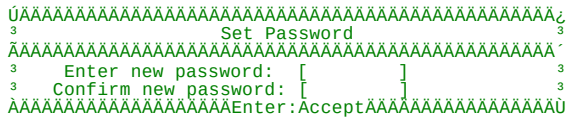
The Security Menu

Selecting "Security" from the Main Menu displays a menu like this:



(The current BIOS doesn`t support this function.)

Use the legend keys to make your selections and exit to the Main Menu.
Enabling "Supervisor Password" requires a password for entering Setup. The passwords are not case sensitive.
Pressing <Enter> at either Set Supervisor Password or Set User Password displays a dialog box like this:



Type the password and press <Enter>. Repeat.

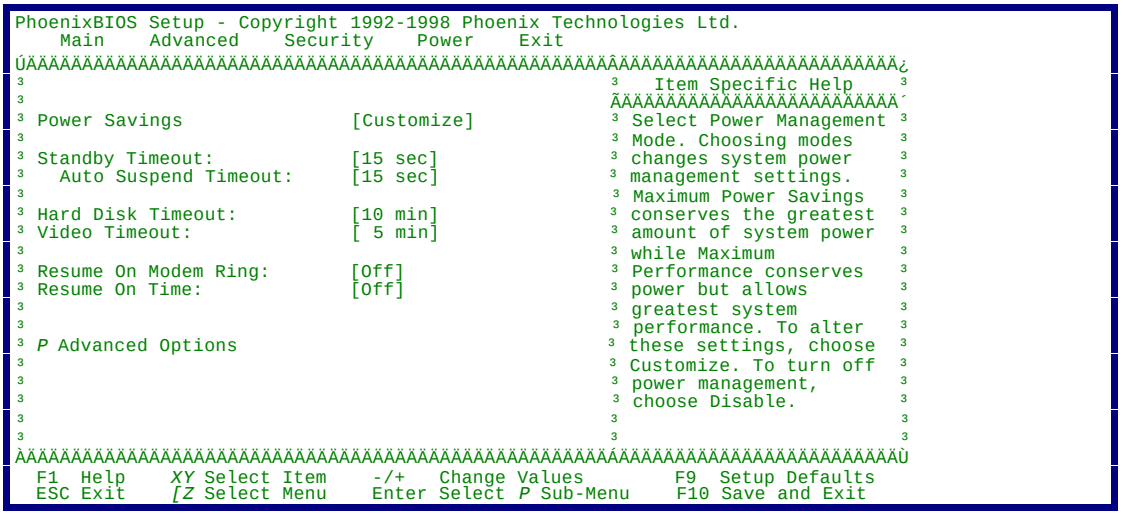
Note: In some systems, the User and Supervisor passwords are related; you cannot have a User password without first creating a Supervisor password. In other systems, you can create and use them independently.

Use the following chart to configure the system-security and anti-virus options.

Feature	Options	Description
Set User Password	Up to seven alphanumeric characters	Pressing <Enter> displays the dialog box for entering the user password. In related systems, this password gives restricted access to SETUP menus.
Set Supervisor Password	Up to seven alphanumeric characters	Pressing <Enter> displays dialog box for entering the supervisor password. In related systems, this password gives full access to Setup menus.
Password on boot	Enabled Disabled	Enabled requires a password on boot. Requires prior setting of the Supervisor password. If supervisor password is set and this option disabled, BIOS assumes user is booting.
Diskette access	Enabled Disabled	Enabled requires a password to boot from or access the floppy disk.
Fixed disk boot sector	Normal Write Protect	Write protects the boot sector on the hard disk for virus protection. Requires a password to format or Fdisk the hard disk.
System backup reminder Virus check reminder	Disabled Daily Weekly Monthly	Displays a message during bootup asking (Y/N) if you have backed up the system or scanned it for viruses. Message returns on each boot until you respond with "Y". Daily displays the message on the first boot of the day, Weekly on the first boot after Sunday, and Monthly on the first boot of the month.

The Power Menu

Selecting "Power" from the menu bar displays a menu like this:



Use this menu to specify your settings for Power Management. Remember that the options available depend upon the hardware installed in your system. Those shown here are from a typical system.

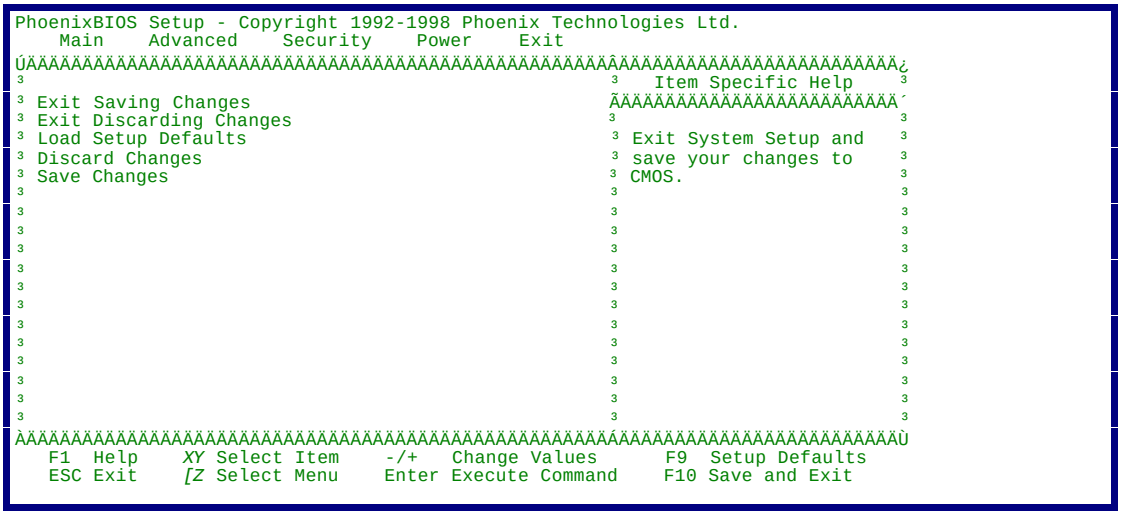
A power-management system reduces the amount of energy used after specified periods of inactivity. The Setup menu pictured here supports a **Full On** state, a **Standby** state with partial power reduction, and a **Suspend** state with full power reduction.

Use the Advanced Options on this menu to specify whether or not the activity of interrupts can terminate a Standby or Suspend state and restore Full On. Do not change these settings without knowing which devices use the interrupts.
Use the legend keys to make your selections and exit to the Main Menu. Use the following chart in making your selections:

Feature	Options	Description
Power Management Mode	Disabled Customize Maximum Power Savings Maximum Performance	Maximum options: pre-defined values. Select Customize to make your own selections from the following fields. Disabled turns off all power management.
Standby Timeout	Off 1 min 2 min 4 min 6 min 8 min 12 min 16 min	Inactivity period required to put system in Standby (partial power shutdown).
Auto Suspend Timeout	Disabled 5 min 10 min 15 min 20 min 30 min 40 min 60 min	Inactivity period required after Standby to Suspend (maximum power shutdown).
Hard Disk Timeout	Disabled 1 min 2 min 4 min 8 min 12 min 16 min	Inactivity period of hard disk required before standby (motor off).
Video Timeout	Disabled 10 sec 15 sec 20 sec 30 sec 45 sec 1 min to 15 min	Set inactivity period required before independently turning off monitor. Disabled turns CRT off in Standby.
Resume On Modem Ring	Off On	Wakes up system when an incoming call is detected on the modem.
Resume On Time	Off On	Wakes up system at predetermined time.
IRQ0...IRQ15 SMI NMI	Disabled Enabled	Enabling interrupt causes it to restore Full On during Standby or Suspend. SMI = System Management Interrupt. NMI = Non-Maskable Interrupt.

The Exit Menu

Selecting "Exit" from the menu bar displays this menu:



The following sections describe each of the options on this menu. Note that <Esc> does not exit this menu. You must select one of the items from the menu or menu bar to exit.

Saving Values

After making your selections on the Setup menus, always select either "Saving Values" or "Save Changes." Both procedures store the selections displayed in the menus in **CMOS** (short for "battery-backed CMOS RAM") a special section of memory that stays on after you turn your system off. The next time you boot your computer, the BIOS configures your system according to the Setup selections stored in CMOS. After you save your selections, the program displays this message:

```
Values have been saved to CMOS!
     Press <space> to continue
```

If you attempt to exit without saving, the program asks if you want to save before exiting.

During bootup, *Phoenix*BIOS attempts to load the values saved in CMOS. If those values cause the system boot to fail, reboot and press **<F2>** to enter Setup. In Setup, you can get the Default Values (as described below) or try to change the selections that caused the boot to fail.

Exit Discarding Changes

Use this option to exit Setup without storing in CMOS any new selections you may have made. The selections previously in effect remain in effect.

Load Setup Defaults

To display the default values for all the Setup menus, select "Load Setup Defaults" from the Main Menu. The program displays this message:

```
ROM Default values have been loaded!
    Press <space> to continue
```

If, during bootup, the BIOS program detects a problem in the integrity of values stored in CMOS, it displays these messages:

```
System CMOS checksum bad - run SETUP
Press <F1> to resume,  <F2> to Setup
```

The CMOS values have been corrupted or modified incorrectly, perhaps by an application program that changes data stored in CMOS.
Press **<F1>** to resume the boot or **<F2>** to run Setup with the ROM default values already loaded into the menus. You can make other changes before saving the values to CMOS.

Discard Changes

If, during a Setup Session, you change your mind about changes you have made and have not yet saved the values to CMOS, you can restore the values you previously saved to CMOS.
Selecting “Discard Changes” on the Exit menu updates all the selections and displays this message:

```
CMOS values have been loaded!
    Press <space> to continue
```

Save Changes

Selecting “Save Changes” saves all the selections without exiting Setup. You can return to the other menus if you want to review and change your selections.

14.4 PhoenixBIOS Messages

The following is a list of the messages that the BIOS can display. Most of them occur during POST. Some of them display information about a hardware device, e.g., the amount of memory installed. Others may indicate a problem with a device, such as the way it has been configured. Following the list are explanations of the messages and remedies for reported problems.
*If your system displays one of the messages marked below with an asterisk (*), write down the message and contact your dealer. If your system fails after you make changes in the Setup menus, reset the computer, enter Setup and install Setup defaults or correct the error.

0200 Failure Fixed Disk
Fixed disk is not working or not configured properly. Check to see if fixed disk is attached properly. Run Setup. Find out if the fixed-disk type is correctly identified.

0210 Stuck key
Stuck key on keyboard.

0211 Keyboard error
Keyboard not working.

***0212 Keyboard Controller Failed**
Keyboard controller failed test. May require replacing keyboard controller.

0213 Keyboard locked - Unlock key switch
Unlock the system to proceed.

0220 Monitor type does not match CMOS - Run SETUP
Monitor type not correctly identified in Setup

0230 Shadow Ram Failed at offset: *nnnn
Shadow RAM failed at offset ***nnnn*** of the 64k block at which the error was detected.

0231 System RAM Failed at offset: *nnnn
System RAM failed at offset ***nnnn*** of in the 64k block at which the error was detected.

0232 Extended RAM Failed at offset: *nnnn Extended memory not working or not configured properly at offset ***nnnn***.

0250 System battery is dead - Replace and run SETUP
The CMOS clock battery indicator shows the battery is dead. Replace the battery and run Setup to reconfigure the system.

0251 System CMOS checksum bad - Default configuration used
System CMOS has been corrupted or modified incorrectly, perhaps by an application program that changes data stored in CMOS. The BIOS installed Default Setup Values. If you do not want these values, enter Setup and enter your own values. If the error persists, check the system battery or contact your dealer.

***0260 System timer error**
The timer test failed. Requires repair of system board.

***0270 Real time clock error** Real-Time Clock fails BIOS hardware test. May require board repair.

0271 Check date and time settings BIOS found date or time out of range and reset the Real-Time Clock. May require setting legal date (1991-2099).

0280 Previous boot incomplete - Default configuration used
Previous POST did not complete successfully. POST loads default values and offers to run Setup. If the failure was caused by incorrect values and they are not corrected, the next boot will likely fail. On systems with control of **wait states**, improper Setup settings can also terminate POST and cause this error on the next boot. Run Setup and verify that the wait-state configuration is correct. This error is cleared the next time the system is booted.

0281 Memory Size found by POST differed from CMOS
Memory size found by POST differed from CMOS.

02B0 Diskette drive A error
02B1 Diskette drive B error
Drive A: or B: is present but fails the BIOS POST diskette tests. Check to see that the drive is defined with the proper diskette type in Setup and that the diskette drive is attached correctly.

02B2 Incorrect Drive A type - run SETUP
Type of floppy drive A: not correctly identified in Setup.

02B3 Incorrect Drive B type - run SETUP
Type of floppy drive B: not correctly identified in Setup.

02D0 System cache error - Cache disabled
RAM cache failed and BIOS disabled the cache. On older boards, check the cache jumpers. You may have to replace the cache. See your dealer. A disabled cache slows system performance considerably.

02F0: CPU ID:
CPU socket number for Multi-Processor error.

***02F4: EISA CMOS not writeable**
ServerBIOS2 test error: Cannot write to EISA CMOS.

***02F5: DMA Test Failed**
ServerBIOS2 test error: Cannot write to extended **DMA** (Direct Memory Access) registers.

***02F6: Software NMI Failed**
ServerBIOS2 test error: Cannot generate software NMI (Non-Maskable Interrupt).

***02F7: Fail-Safe Timer NMI Failed**
ServerBIOS2 test error: Fail-Safe Timer takes too long.

***device* Address Conflict**
Address conflict for specified ***device*.**

Allocation Error for: *device*
Run ISA or EISA Configuration Utility to resolve resource conflict for the specified ***device***.

CD ROM Drive
CD ROM Drive identified.

Entering SETUP ...
Starting Setup program

Failing Bits: *nnnn
The hex number ***nnnn*** is a map of the bits at the RAM address which failed the memory test. Each 1 (one) in the map indicates a failed bit. See errors 230, 231, or 232 above for offset address of the failure in System, Extended, or Shadow memory.
Fixed Disk *n*
Fixed disk ***n*** (0-3) identified.
Invalid System Configuration Data
Problem with NVRAM (CMOS) data.
I/O device IRQ conflict
I/O device IRQ conflict error.
PS/2 Mouse Boot Summary Screen:
PS/2 Mouse installed.
***nnnn* kB Extended RAM Passed**
Where ***nnnn*** is the amount of RAM in kilobytes successfully tested.
***nnnn* Cache SRAM Passed**
Where ***nnnn*** is the amount of system cache in kilobytes successfully tested.
***nnnn* kB Shadow RAM Passed**
Where ***nnnn*** is the amount of shadow RAM in kilobytes successfully tested.
***nnnn* kB System RAM Passed**
Where ***nnnn*** is the amount of system RAM in kilobytes successfully tested.
One or more I2O Block Storage Devices were excluded from the Setup Boot Menu
There was not enough room in the IPL table to display all installed I_2O block-storage devices.
Operating system not found
Operating system cannot be located on either drive A: or drive C:. Enter Setup and see if fixed disk and drive A: are properly identified.
Parity Check 1 *nnnn
Parity error found in the system bus. BIOS attempts to locate the address and display it on the screen. If it cannot locate the address, it displays **????.** Parity is a method for checking errors in binary data. A parity error indicates that some data has been corrupted.
Parity Check 2 *nnnn
Parity error found in the I/O bus. BIOS attempts to locate the address and display it on the screen. If it cannot locate the address, it displays **????.**
Press <F1> to resume, <F2> to Setup,
<F3> for previous
Displayed after any recoverable error message. Press <F1> to start the boot process or <F2>**Fehler! Verweisquelle konnte nicht gefunden werden.** to enter Setup and

14.5 Phoenix Phlash Tool

Phoenix Boot Utilities are:

- Phoenix QuietBoot™
- Phoenix MultiBoot™

Phoenix QuietBoot displays a graphic illustration rather than the traditional POST messages while keeping you informed of diagnostic problems.
Phoenix MultiBoot is a boot screen that displays a selection of boot devices from which you can boot your operating system.

14.6 Phoenix QuietBoot

Right after you turn on or reset the computer, **Phoenix QuietBoot** displays the QuietBoot Screen, a graphic illustration created by the computer manufacturer instead of the text-based POST screen, which displays a number of PC diagnostic messages.
To exit the QuietBoot screen and run Setup, display the MultiBoot menu, or simply display the PC diagnostic messages, you can simply press one of the hot keys described below.
The QuietBoot Screen stays up until just before the operating system loads unless:

You press <Esc> to display the POST screen.
You press <F2> to enter Setup.
POST issues an error message.
The BIOS or an option ROM requests keyboard input.

The following explains each of these situations.

14.6.1 Press <ESC>

Pressing <Esc>**Fehler! Verweisquelle konnte nicht gefunden werden.** switches to the POST screen and takes one of two actions:

1. If MultiBoot is installed, the boot process continues with the text-based POST screen until the end of POST, and then displays the **Boot First Menu**, with these options:
   - ? Load the operating system from a boot device of your choice.
   - ? Enter Setup.
   - ? Exit the Boot First Menu (with <Esc>) and load the operating system from the boot devices in the order specified in Setup.
2. If MultiBoot is not installed, the boot process continues as usual.

14.6.2 Press <F2>

Pressing <F2> at any time during POST switches to the POST screen (if not already displayed) and enters Setup.

14.6.3 POST Error

Whenever POST detects a non-fatal error, QuietBoot switches to the POST screen and displays the errors. It then displays this message:

```
Press <F1> to resume, <F2> to Setup
```

Press <F1> to continue with the boot. Press <F2> if you want to correct the error in Setup.

14.6.4 Keyboard Input Request

If the BIOS or an **Option ROM** (add-on card) requests keyboard input, QuietBoot switches over to the POST screen and the Option ROM displays prompts for entering the information. POST continues from there with the regular POST screen.

14.7 Phoenix MultiBoot

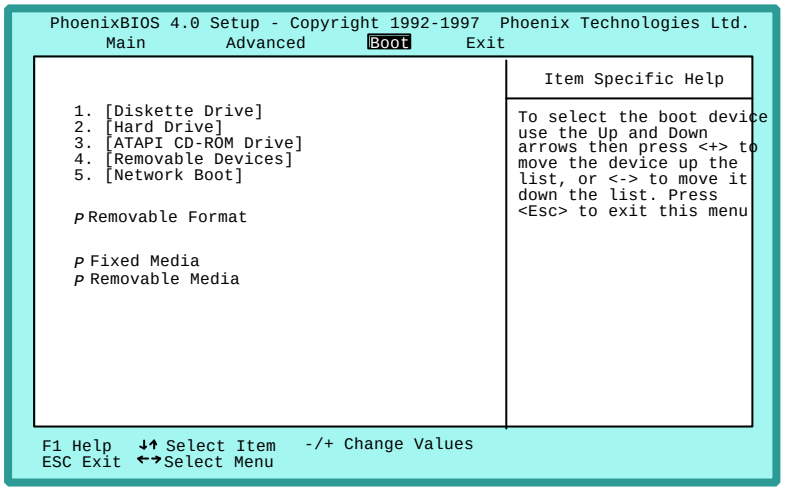
Phoenix MultiBoot expands your boot options by letting you choose your boot device, which could be a hard disk, floppy disk, or CD ROM. You can select your boot device in Setup, or you can choose a different device each time you boot by selecting your boot device in **The Boot First Menu**.
MultiBoot consists of:

- The Setup Boot Menu
- The Removable Format Menu
- The Fixed Disk and Removable Disk Menus
- The Boot First Menu

The following describes each one of these menus.

14.7.1 The Setup Boot Menu

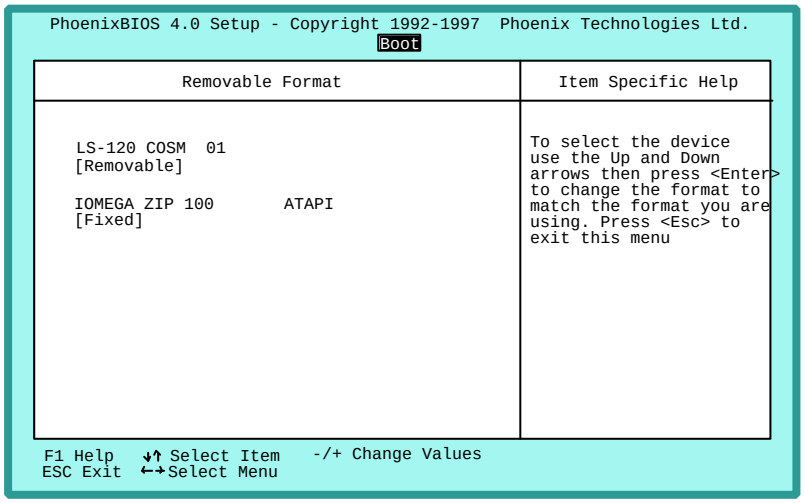
In the Setup **Boot Menu**, shown here, you can select the order of the devices from which the BIOS attempts to boot the operating system. During POST, if the BIOS is unsuccessful at booting from one device, it will try the next one on the list.
The items on this menu each may represent the first of a **class** of items—if you have more than one device of this class installed on your system. For example, if you have more than one hard-disk drive, [Hard Drive] represents the first of such drives as specified in the Fixed Media menu described below.



If you have more than one bootable hard drive, select **Fixed Media** and press <Enter> to display the Fixed Disk menu and choose which drive is represented in the boot-order list.
If you have more than one Removable Media drive, select **Removable Media** and press <Enter> to display the Removable Media menu and choose which drive is represented in the boot-order list.
Select **Removable Format** to display the Removable Format menu for determining whether the removable media is formatted as removable or fixed disk.

14.7.2 Removable Format Menu

Selecting Removable Format brings up a menu like this:



Use this menu to specify whether your removable media is formatted like a hard disk or floppy.

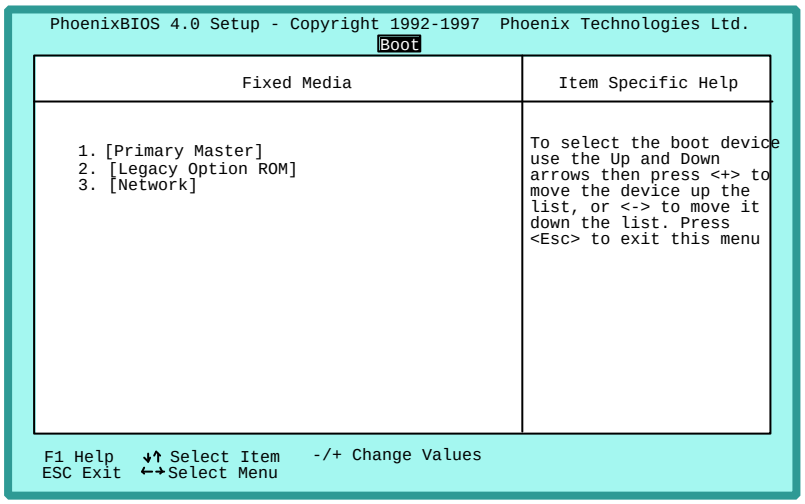
14.7.3 Fixed and Removable Media Menus

Selecting **Fixed Media** or **Removable Media** brings up a menu like the one shown below for determining which item is represented on the order list specified in the Setup Menu described above and displayed in Boot First Menu shown below. “Bootable ISA Cards” refers to devices, such as Legacy (non-PnP) SCSI or network cards, from which you can boot the operating system.
Use these menu to select the order in which POST installs the devices and the operating system assigns device letters:

A:, B:, etc. to floppy drives

C:, D: and E:, etc. to the hard-disk drives

NOTE: There is not always an exact correspondence between the order specified in these menus and the letters assigned by the operating system. Many devices such as Legacy Option ROMs support more than one device, which can be assigned more than one letter.
PhoenixBIOS 4.0 supports up to six floppy devices, to which the operating system may assign, for example, drive letters A;, B:, E:, F:, G:, and H:.
If you want the CD ROM Drive to have a letter coming before the Hard Drive, move it in front of the Hard Drive. MultiBoot attempts to boot only from the first item in this list and ignores the rest.

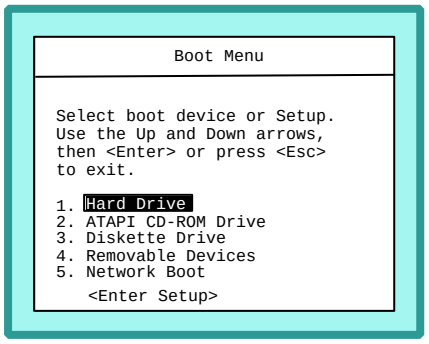


Selecting **Removable Media** brings up a similar menu for selecting the boot order of removable media devices if you have more than one.

14.7.4 The Boot First Menu

Display the Boot First Menu by pressing <Esc> during POST. In response, the BIOS first displays the message, "Entering Boot Menu ..." and then displays the Boot Menu at the end of POST. Use the menu to select any of these options:

1. Override the existing boot sequence (for this boot only) by selecting another boot device. If the specified device does not load the operating system, the BIOS reverts to the previous boot sequence.
2. Enter Setup.
3. Press <Esc> to continue with the existing boot sequence.



If there is more than one bootable hard drive, the first one in the Boot Connection Device Menu is the one represented here.

15 PHOENIX PHLASH

(The current BIOS doesn`t support this function.)

Phoenix Phlash gives you the ability to update your BIOS from a floppy disk without having to install a new ROM BIOS chip.
Phoenix Phlash is a utility for "flashing" (copying) a BIOS to the Flash ROM installed on your computer from a floppy disk. A Flash ROM is a Read-Only Memory chip that you can write to using a special method called "flashing." Use Phoenix Phlash for the following tasks:

Update the current BIOS with a new version.

Restore a BIOS when it has become corrupted.

15.1 Installation

Phoenix Phlash is shipped on a floppy disk with your computer as a compressed file called CRISDISK.ZIP that contains the following files:

CRISDISK.BAT	Executable file for creating the Crisis Recovery Diskette.
PHLASH.EXE	Programs the flash ROM.
PLATFORM.BIN	Performs platform-dependent functions.
BIOS.ROM	Actual BIOS image to be programmed into flash ROM.
MINIDOS.SYS	Allows the system to boot in Crisis Recovery Mode.
MAKEBOOT.EXE	Creates the custom boot sector on the Crisis Recovery Diskette.

To install Phoenix Phlash on your hard disk, follow this simple procedure:

1. Insert the distribution diskette into drive A:
2. Unzip the contents of CRISDISK.ZIP into a local directory, presumably `C:\PHLASH`.
3. Store the distribution diskette in a safe place.

15.2 Create the Crisis Recovery Diskette

If the OEM or dealer from whom you purchased your system has not provided you with one, then you should create a **Crisis Recovery Diskette** before you use the Phlash utility. If you are unable to boot your system and successfully load the Operating System, the BIOS may have been corrupted, in which case you will have to use the Crisis Recovery Diskette to reboot your system. There are several methods that you can use to create the Crisis Recovery Diskette. Below is one recommended procedure.

1. Be sure you have successfully installed the Phlash Utility onto your hard disk.
2. Insert a clean diskette into drive A: or B:
3. From the local directory, enter the following:

   CRISDISK [drive]:

where [*drive*] is the letter of the drive into which you inserted the diskette.
For help, type **/?** or **/h**.

CRISDISK.BAT formats the diskette, then copies MINIDOS.SYS, VGABIOS.EXE (if available), PHLASH.EXE, PLATFORM.BIN and BIOS.ROM to the diskette, and creates the required custom boot sector.

4. Write protect and label the Crisis Recovery Diskette.

NOTE: You can only supply a volume label after the Crisis Recovery Diskette has been formatted and the necessary files copied because MINIDOS.SYS must occupy the first directory entry for the diskette to boot properly.

15.3 Updating the Crisis Recovery Diskette

If the BIOS image (BIOS.ROM) changes due to an update or bug fix, you can easily update the Crisis Recovery Diskette. Simply copy the new BIOS.ROM image onto the Crisis Recovery Diskette. No further action is necessary.

15.4 Executing Phoenix Phlash

You can run Phoenix Phlash in one of two modes:

Command Line Mode

Crisis Recovery Mode

WARNING! For your own protection, be sure you have a Crisis Recovery Diskette ready to use before executing Phlash.

15.4.1.1 Command Line Mode

Use this mode to update or replace your current BIOS. To execute Phlash in this mode, move to the directory into which you have installed Phoenix Phlash and type the following:

```
phlash
```

Phoenix Phlash will automatically update or replace the current BIOS with the one which your OEM or dealer supplies you.
Phlash may fail if your system is using memory managers, in which case the utility will display the following message:

```
Cannot flash when memory managers are present.
```

If you see this message after you execute Phlash, you must disable the memory manager on your system. To do so, follow the instructions in the following sections.

15.4.1.2 Disabling Memory Managers

To avoid failure when flashing, you must disable the memory managers that load from CONFIG.SYS and AUTOEXEC.BAT. There are two recommended procedures for disabling the memory managers. One consists of pressing the <F5> key (only if you are using DOS 5.0 or above), and the other requires the creation of a boot diskette.

15.4.1.3 DOS 5.0 (or later version)

For DOS 5.0 and later, follow the two steps below to disable any memory managers on your system. If you are not using at least DOS 5.0, then you must create a boot diskette to bypass any memory managers (See Create a Boot Diskette, below).

1. Boot DOS 5.0 or later version. (In Windows 95, at the boot option screen, choose Option 8, "Boot to a previous version of DOS.")
3. When DOS displays the “Starting MS-DOS” message, press <F5>.

After you press <F5>, DOS bypasses the CONFIG.SYS and AUTOEXEC.BAT files, and therefore does not load any memory managers.
You can now execute Phlash.

15.4.1.4 Create a Boot Diskette

To bypass memory managers in DOS versions previous to 5.0, follow this recommended procedure:

1. Insert a diskette into your A: drive.
2. Enter the following from the command line:

   ```
   Format A: /S
   ```
4. Reboot your system from the A: drive.

Your system will now boot without loading the memory managers, and you can then execute Phlash.
NOTE: The boot diskette you create here is distinct from a *Crisis Recovery Diskette*. See page 409 for details about creating the Crisis Recovery Diskette.

15.5 Crisis Recovery Mode

You should only have to operate Phoenix Phlash in this mode only if your system does not boot the operating system when you turn on or reset your computer. In these cases, the BIOS on the Flash ROM has probably been corrupted. Boot your system with the Crisis Recovery Diskette taking these steps:

1. Insert the Crisis Recovery diskette (which your dealer supplied or one that you should have created from the instructions above) into drive A:.
2. Reset your computer, power on-off, or press <Ctrl> <Alt> <Del> to reboot the system.

When your system reboots, Phoenix Phlash will restore the BIOS from the diskette and successfully boot the operating system.

16 PROGRAMMER'S GUIDE

This chapter of the User's Manual gives application developers, programmers, and expert computer users a detailed description of the BIOS.
This chapter describes the following subjects:

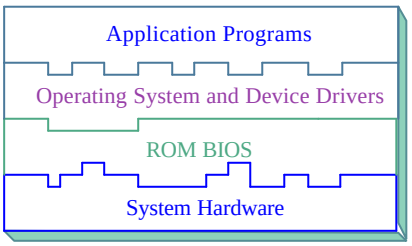
+ What is a ROM BIOS?
+ System Hardware Requirements
+ Fixed-Disk Tables
+ PhoenixBIOS Function Keys
+ POST Errors
+ Run-Time Services

What is a ROM BIOS?

This section briefly explains the function of a BIOS in managing the special features of your system.
A **ROM BIOS** (**Basic Input/Output System**) is a set of programs permanently stored in a **ROM** (Read-Only Memory) chip located on the computer motherboard. These programs micro-manage the hardware devices installed on your computer. When you turn on your computer, the ROM BIOS initializes and tests these devices. During run-time, the ROM BIOS provides the Operating System and application programs with access to these devices. You can also use the BIOS **Setup** program to change your computer's hardware or behavior.

Software works best when it operates in layers, and the ROM BIOS is the bottom-most software layer in the computer. It functions as the interface between the hard-ware and the other layers of software, isolating them from the details of how the hardware works. This arrangement enables you to change hardware devices without having to install a new operating system.

The following diagram shows the function of the ROM BIOS as the interface between the hardware and other layers of software:



ROM BIOS Functions
The *Phoenix*BIOS software performs these functions:

The Setup Program	Using the Setup program, you can install, configure, and optimize the hardware devices on your system (clock, memory, disk drives, etc.).
Initialize Hardware at Boot	At power-on or reset, perform Power-On Self Test (POST) routines to test system resources and run the operating system.
Run-Time Routines	Basic hardware routines that can be called from DOS and Windows applications.

Initialize and Configure the computer

The first job of a ROM BIOS is to initialize and configure the computer hardware when you turn on your computer (system boot). The BIOS runs a series of complex programs called the **Power On Self Test** (POST), which performs a number of tasks, including:

- Test Random Access Memory (RAM)
- Conduct an inventory of the hardware devices installed in the computer
- Configure hard and floppy disks, keyboard, monitor, and serial and parallel ports
- Configure other devices installed in the computer such as CD-ROM drives and sound cards
- Initialize computer hardware required for computer features such as Plug and Play and Power Management
- Run Setup if requested
- Load and run the Operating System such as DOS, OS/2, UNIX, or Windows 95 or NT.

16.1.1 BIOS Services

The second task of the ROM BIOS is to provide the Operating System, device drivers, and application programs with access to the system hardware. It performs this task with a set of program routines called **BIOS Services**, which are loaded into high memory at boot time.
The number of BIOS Services is always changing. The BIOS Services of PhoenixBIOS 4.05 provide precise control of hardware devices such as disk drives, which require careful management and exhaustive checking for errors. They also help manage new computer features such as Power Management, Plug and Play, and MultiBoot.

16.2 System Hardware Requirements

PhoenixBIOS 4.0 requires the following hardware components on the motherboard:

System Board Requirements	
1.	CPU (486 or later)
2.	AT-compatible and MC146818 RTC-compatible chipset.
3.	AT or PS/2-compatible Keyboard controller
4.	At least 1 MB of system RAM

The power on self test (POST) of the BIOS initializes additional ROM BIOS extensions (Option ROMs) if they are accessible in the proper format. The requirements are:

Adapter ROM Requirements
1. The code must reside in the address space between C0000H and F0000H.
2. The code must reside on a 2K boundary.
3. The first two bytes of the code must be 55H and AAH.
4. The third byte must contain the number of 512-byte blocks.
5. The fourth byte must contain a jump to the start of the initialization code.
6. The code must checksum to zero (byte sum).

NOTE: The address space from C0000H to C8000H is reserved for external video adapters (e.g. EGA, VGA). Part of the address space from D0000H to E0000H is typically used by expanded memory (EMS).

16.3 Fixed Disk Tables

PhoenixBIOS 4.0 supports up to four fixed-disk drives. For each drive, it supports 39 pre-defined drive types and four user-defined types (40-43). Below is a table of the pre-defined drive types and their default values.
End users can modify the user-defined drive type for each fixed disk listed in Setup by using the menus of the Setup program. This feature avoids the need for customized software for non-standard drives.

Fixed Disk Tables					
Type	Cylinders	Heads	Sectors	Wrt Pre	Landing
1	306	4	17	128	305
2	615	4	17	300	615
3	615	6	17	300	615
4	940	4	17	512	940
5	940	6	17	512	940
6	615	4	17	-1	615
7	462	8	17	256	511
8	733	5	17	-1	733
9	900	15	17	-1	901
10	820	3	17	-1	820
11	855	5	17	-1	855
12	855	7	17	-1	855
13	306	8	17	128	319
14	733	7	17	-1	733
15	Reserved				
16	612	4	17	0	633
17	977	5	17	300	977
18	977	7	17	-1	977
19	1024	7	17	512	1023
20	733	5	17	300	732
21	733	7	17	300	732
22	733	5	17	300	733
23	306	4	17	0	336
24	612	4	17	305	663
25	612	2	17	300	612
26	614	4	17	-1	614
27	820	6	17	-1	820
28	977	5	17	-1	977

Type	Cylinders	Heads	Sectors	Wrt Pre	Landing
29	1218	15	36	-1	1218
30	1224	15	17	-1	1224
31	823	10	17	512	823
32	809	6	17	128	809
33	830	7	17	-1	830
34	830	10	17	-1	830
35	1024	5	17	-1	1024
36	1024	8	17	-1	1024
37	615	8	17	128	615
38	1024	8	26	-1	1024
39	925	9	17	-1	925
40	User def.				
41	User def.				
42	User def.				
43	User def.				

16.4 PhoenixBIOS Function Keys

The following are the special PhoenixBIOS function keys:

<F2>	Enter SETUP program during POST
Ctrl-Alt-<->	Switch to slow CPU speed
Ctrl-Alt-<+>	Switch to fast CPU speed

The speed switching keys are only operational when speed switching is available. **Fehler! Verweisquelle konnte nicht gefunden werden.**

16.5 POST Errors and Beep Codes

16.5.1 Recoverable POST Errors

Whenever a recoverable error occurs during POST, *Phoenix*BIOS displays an error message describing the problem.
*Phoenix*BIOS also issues a beep code (one long tone followed by two short tones) during POST if the video configuration fails (no card installed or faulty) or if an external ROM module does not properly checksum to zero.
An external ROM module (e.g. VGA) can also issue audible errors, usually consisting of one long tone followed by a series of short tones.

16.5.2 Terminal POST Errors

There are several POST routines that issue a **POST Terminal Error** and shut down the system if they fail. Before shutting down the system, the terminal-error handler issues a beep code signifying the test point error, writes the error to port 80h, attempts to initialize the video, and writes the error in the upper left corner of the screen (using both mono and color adapters).
The routine derives the beep code from the test point error as follows:

1. The 8-bit error code is broken down to four 2-bit groups (Discard the most significant group if it is 00).
2. Each group is made one-based (1 through 4) by adding 1.
3. Short beeps are generated for the numb er in each group.

Example:

Testpoint 01Ah = 00 01 10 10 = 1-2-3-3 beeps

16.5.3 Test Points and Beep Codes

At the beginning of each POST routine, the BIOS outputs the test point error code to I/O address 80h. Use this code during trouble shooting to establish at what point the system failed and what routine was being performed.
Some motherboards are equipped with a seven-segment LED display that displays the current value of port 80h. For production boards which do not contain the LED display, you can purchase a card that performs the same function.
If the BIOS detects a terminal error condition, it halts POST after issuing a terminal error beep code (See above) and attempting to display the error code on upper left corner of the screen and on the port 80h LED display. It attempts repeatedly to write the error to the screen. This may cause "hash" on some CGA displays.
If the system hangs before the BIOS can process the error, the value displayed at the port 80h is the last test performed. In this case, the screen does not display the error code.

The following is a list of the checkpoint codes written at the start of each test and the beep codes issued for terminal errors. Unless otherwise noted, these codes are valid for Phoenix-BIOS 4.0 Release 6.x.

Code	Beeps	POST Routine Description
02h		Verify Real Mode
03h		Disable Non-Maskable Interrupt (NMI)
04h		Get CPU type
06h		Initialize system hardware
07h		Disable shadow and execute code from the ROM.
08h		Initialize chipset with initial POST values
09h		Set IN POST flag
0Ah		Initialize CPU registers
0Bh		Enable CPU cache
0Ch		Initialize caches to initial POST values
0Eh		Initialize I/O component
0Fh		Initialize the local bus IDE
10h		Initialize Power Management
11h		Load alternate registers with initial POST values
12h		Restore CPU control word during warm boot
13h		Initialize PCI Bus Mastering devices
14h		Initialize keyboard controller
16h	1-2-2-3	BIOS ROM checksum
17h		Initialize cache before memory Autosize
18h		8254 timer initialization
1Ah		8237 DMA controller initialization
1Ch		Reset Programmable Interrupt Controller
20h	1-3-1-1	Test DRAM refresh
22h	1-3-1-3	Test 8742 Keyboard Controller
24h		Set ES segment register to 4 GB
28h		Autosize DRAM
29h		Initialize POST Memory Manager
2Ah		Clear 512 kB base RAM
2Ch	1-3-4-1	RAM failure on address line ***xxxx****
2Eh	1-3-4-3	RAM failure on data bits ***xxxx**** of low byte of memory bus
2Fh		Enable cache before system BIOS shadow
32h		Test CPU bus-clock frequency
33h		Initialize Phoenix Dispatch Manager
36h		Warm start shut down
38h		Shadow system BIOS ROM
3Ah		Autosize cache

Code	Beeps	POST Routine Description
3Ch		Advanced configuration of chipset registers
3Dh		Load alternate registers with CMOS values
41h		Initialize extended memory for RomPilot
42h		Initialize interrupt vectors
45h		POST device initialization
46h	2-1-2-3	Check ROM copyright notice
47h		Initialize I20 support
48h		Check video configuration against CMOS
49h		Initialize PCI bus and devices
4Ah		Initialize all video adapters in system
4Bh		QuietBoot start (optional)
4Ch		Shadow video BIOS ROM
4Eh		Display BIOS copyright notice
4Fh		Initialize MultiBoot
50h		Display CPU type and speed
51h		Initialize EISA board
52h		Test keyboard
54h		Set key click if enabled
55h		Enable USB devices
58h	2-2-3-1	Test for unexpected interrupts
59h		Initialize POST display service
5Ah		Display prompt "Press F2 to enter SETUP"
5Bh		Disable CPU cache
5Ch		Test RAM between 512 and 640 kB
60h		Test extended memory
62h		Test extended memory address lines
64h		Jump to UserPatch1
66h		Configure advanced cache registers
67h		Initialize Multi Processor APIC
68h		Enable external and CPU caches
69h		Setup System Management Mode (SMM) area
6Ah		Display external L2 cache size
6Bh		Load custom defaults (optional)
6Ch		Display shadow-area message
6Eh		Display possible high address for UMB recovery
70h		Display error messages
72h		Check for configuration errors
76h		Check for keyboard errors
7Ch		Set up hardware interrupt vectors
7Dh		Initialilze Intelligent System Monitoring
7Eh		Initialize coprocessor if present
80h		Disable onboard Super I/O ports and IRQs
81h		Late POST device initialization
82h		Detect and install external RS232 ports
83h		Configure non-MCD IDE controllers
84h		Detect and install external parallel ports
85h		Initialize PC-compatible PnP ISA devices
86h		Re-initialize onboard I/O ports.
87h		Configure Motheboard Configurable Devices (optional)
88h		Initialize BIOS Data Area
89h		Enable Non-Maskable Interrupts (NMIs)
8Ah		Initialize Extended BIOS Data Area
8Bh		Test and initialize PS/2 mouse
8Ch		Initialize floppy controller
8Fh		Determine number of ATA drives (optional)
90h		Initialize hard-disk controllers
91h		Initialize local-bus hard-disk controllers

Code	Beeps	POST Routine Description
92h		Jump to UserPatch2
93h		Build MPTABLE for multi-processor boards
95h		Install CD ROM for boot
96h		Clear huge ES segment register
97h		Fixup Multi Processor table
98h	1-2	Search for option ROMs. One long, two short beeps on checksum failure
99h		Check for SMART Drive (optional)
9Ah		Shadow option ROMs
9Ch		Set up Power Management
9Dh		Initialize security engine (optional)
9Eh		Enable hardware interrupts
9Fh		Determine number of ATA and SCSI drives
A0h		Set time of day
A2h		Check key lock
A4h		Initialize typematic rate
A8h		Erase F2 prompt
AAh		Scan for F2 key stroke
ACh		Enter SETUP
AEh		Clear Boot flag
B0h		Check for errors
B1h		Inform RomPilot about the end of POST.
B2h		POST done - prepare to boot operating system
B4h	1	One short beep before boot
B5h		Terminate QuietBoot (optional)
B6h		Check password (optional)
B7h		Initialize ACPI BIOS
B9h		Prepare Boot
BAh		Initialize SMBIOS
BBh		Initialize PnP Option ROMs
BCh		Clear parity checkers
BDh		Display MultiBoot menu
BEh		Clear screen (optional)
BFh		Check virus and backup reminders
C0h		Try to boot with INT 19
C1h		Initialize POST Error Manager (PEM)
C2h		Initialize error logging
C3h		Initialize error display function
C4h		Initialize system error handler
C5h		PnPnd dual CMOS (optional)
C6h		Initialize note dock (optional)
C7h		Initialize note dock late
C8h		Force check (optional)
C9h		Extended checksum (optional)
CAh		Redirect Int 15h to enable remote keyboard
CBh		Redirect Int 13h to Memory Technologies Devices such as ROM, RAM, PCMCIA, and serial disk
CCh		Redirect Int 10h to enable remote serial video
CDh		Remap I/O and memory for PCMCIA
CEh		Initialize digitizer and display message
D2h		Unknown interrupt
		The following are for boot block in Flash ROM
E0h		Initialize the chipset
E1h		Initialize the bridge
E2h		Initialize the CPU
E3h		Initialize system timer
E4h		Initialize system I/O

Code	Beeps	POST Routine Description
E5h		Check force recovery boot
E6h		Checksum BIOS ROM
E7h		Go to BIOS
E8h		Set Huge Segment
E9h		Initialize Multi Processor
EAh		Initialilze OEM special code
EBh		Initialize PIC and DMA
ECh		Initialize Memory type
EDh		Initialize Memory size
EEh		Shadow Boot Block
EFh		System memory test
F0h		Initialize interrupt vectors
F1h		Initialize Run Time Clock
F2h		Initialize video
F3h		Initialize System Management Manager
F4h		Output one beep
F5h		Boot to Mini DOS
F6h		Clear Huge Segment
F7h		Boot to Full DOS

* If the BIOS detects error 2C, 2E, or 30 (base 512K RAM error), it displays an additional word-bitmap (***xxxx***) indicating the address line or bits that failed. For example, "2C 0002" means address line 1 (bit one set) has failed. "2E 1020" means data bits 12 and 5 (bits 12 and 5 set) have failed in the lower 16 bits. Note that error 30 cannot occur on 386SX systems because they have a 16 rather than 32-bit bus. The BIOS also sends the bitmap to the port-80 LED display. It first displays the check point code, followed by a delay, the high-order byte, another delay, and then the low-order byte of the error. It repeats this sequence continuously.

16.6 PhoenixBIOS 4.0 Services

The ROM BIOS contains a number of useful run-time **BIOS Services** that are easily called by an outside program. As a programmer, you can execute these services, which are nothing more than subroutines, by invoking one of the BIOS interrupt routines (or, when specified, calling a protected-mode entry point and offset). Invoking a software interrupt causes the CPU to fetch an address from the **interrupt table** in low memory and execute the service routine. Some services return exit values in certain registers. All registers are preserved unless they return data or status.
Generally, a Carry flag set on exit indicates a failed service. A zero on exit in the AH register usually indicates no error; any other value is the service's **exit status code**.

16.6.1 BIOS32 Service Directory

While the standard BIOS services are accessed through the interrupt table, newer services are accessed by a FAR CALL to a service entry point. Programmers can determine the entry point by searching for a particular signature (such as "$PnP") in the BIOS range and finding the entry point in the header.
The **BIOS32 Service Directory** (standard in PhoenixBIOS 4.0) provides a single entry point for all those services in the BIOS that are designed for BIOS clients running in a 32-bit code segment, such as 32-bit operating systems and 32-bit device drivers. The BIOS32 Service Directory itself is a 32-bit BIOS service that provides a single entry point for the other 32-bit services. For a full description of this service, see the **Standard BIOS 32-Bit Service Directory Proposal, Rev 0.4** published by Phoenix and available on the Phoenix Web site at:

http://www.phoenix.com/products/specs.html

Programs calling the 32-bit BIOS services should scan 0E0000h to 0FFFF0h on the 16-byte boundaries for the contiguous 16-byte data structure beginning with the ASCII signature "_32_".
If they do not find this data structure, then the platform does not support the BIOS32 Service Directory. The following chart describes the data structure.

Offset	Size	Description
0h	4 bytes	ASCII signature "_32_" Offset 0 = underscore Offset 1 = "3" Offset 2 = "2" Offset 3 = underscore
4h	4 bytes	Entry point for the BIOS32 Service Directory, a 32-bit physical address
8h	1 byte	Revision level. Currently 00h.
9h	1 byte	Length of this structure in 16-byte units. This structure is 16 bytes long, so the field = 01h.
0Ah	1 byte	Checksum of whole data structure. Result must be 0.
0Bh	5 bytes	Reserved. Must be zero.

Once the data structure is found and verified, the program can do a FAR CALL to the entry point specified in the above structure. The calling environment requires:

1. The CS code segment selector and the DS data segment selector must encompass the physical page of the entry point as well as the following page.
2. The SS stack segment selector must have available 1 kB of stack space.
3. Access to I/O space.

The BIOS32 Service Directory provides a single call that:

1. Determines if the called 32-bit service is available, and, if it is available,
2. Returns three values:
   a) Physical address of the base of the BIOS service.
   b) Length of the BIOS service.
   c) Entry point into the BIOS service (offset of the base).

BIOS32 Service Directory

Entry:
EAX Service Identifier. Four-character string identifying the 32-bit service requested (e.g., "$PCI").
EBX Low-order byte [BL] is the BIOS32 Service Directory Function Selector. Currently, zero supplies the values described below. Upper three bytes are reserved and must be zero on entry.
Exit:
AL Return code:
00h = Service corresponding to the Service Identifier is present.
80h = Service corresponding to the Service Identifier is not present.
81h = Function Selector specified not supported.
EBX Physical address of base of 32-bit service.
ECX Length of BIOS service.
EDX Entry point of BIOS service (offset to base in EBX).

16.6.2 Interrupt 10h–Video Services

The INT 10h software interrupt handles all video services. The results of some of these functions may depend on the active video mode and the particular video controller installed.

Interrupt 10 Video Services	
AH = 00h	**Set video mode**
Entry:	
AL	Mode value (0-7): 0 = 40x25 Black & White 1 = 40x25 Color 2 = 80x25 Black & White 3 = 80x25 Color 4 = 320x200 Color 5 = 320x200 Black & White 6 = 640x200 Black & White 7 = Monochrome only
AH = 01h	**Set cursor size**
Entry:	
CH	Bits 4-0 = Cursor top scan line
CL	Bits 4-0 = Cursor bottom scan line
AH = 02h	**Set cursor position**
Entry:	
BH	Page to set cursor
DL	Character column position
DH	Character row position
AH = 03h	**Get cursor position of page**
Entry:	
BH	Page to return cursor
Exit:	
DL	Character column position
DH	Character row position
CL	Cursor top scan line
CH	Cursor bottom scan line
AH = 05h	**Change displayed (active) page**
Entry:	
AL	Page number to display
AH = 06h	**Scroll active page up**
Entry:	
CL	Upper left column to scroll up
CH	Upper left row to scroll up
DL	Lower right column to scroll up
DH	Lower right row to scroll up
BH	Attribute for blanked space
AL	Number of lines to scroll up 0 = Blank screen
AH = 07h	**Scroll active page down**
Entry:	
CL	Upper left column to scroll down
CH	Upper left row to scroll down
DL	Lower right column to scroll down
DH	Lower right row to scroll down
BH	Attribute for blanked space
AL	Number of lines to scroll down 0 = Blank screen
AH = 08h	**Read character and attribute**
Entry:	
BH	Video page to read character
Exit:	
AL	Character
AH	Character attribute
AH = 09h	**Write character and attribute**
Entry:	
AL	Character to write
BL	Character attribute (alpha) Character color (graphics)
BH	Page to write character
CX	Count of characters to write

continued

AH = 0Ah Write character at cursor
Entry:
BH Page to write character
AL Character to write
CX Count of characters to write

AH = 0Bh Set color palette
Entry:

BH = 00 Set colors:
If mode = 4 or 5, BL = background color
If mode = 0-3, BL = border color
If mode = 6 or 11, BL = foreground color
BL 0-31 = Intense versions of colors 0-15

BH = 01 Set palette for mode 4 or 5
BL 00 Palette = Green (1), Red (2), Yellow (3)
01 Palette = Cyan (1), Magenta (2), White (3)

AH = 0Ch Write graphics pixel
Entry:
AL Color value for pixel
(XORed if bit7=1)
CX Column to write pixel
DX Row to write pixel

AH = 0D Read graphics pixel
Entry:
CX Column to read pixel
DX Row to read pixel
Exit:
AL Value of pixel read

AH = 0E Teletype write character
Entry:
AL Character to write
BL Foreground color (graphics only)

AH = 0F Return Current Video Parameters
Exit:
AL Current video mode
AH Number of character columns
BH Active page

AH = 13h Write string
Entry:
ES:BP Pointer to string
CX Length of string to display
DH Character row for display
DL Character column for display
BL Display attribute
AL Write string mode
0 = Chars only, no cursor update
1 = Chars only, update cursor
2 = Char, Attrib, no cursor update
3 = Char, Attrib, update cursor

16.6.3 Interrupt 11h–Return System Information

This service returns the equipment installed as determined by the BIOS on power-up diagnostics and stored in the BIOS Data Area.

Interrupt 11 Return System Information

Exit:

AX Equipment information:

Bit	Definition
0	Not used
1	Math coprocessor installed
2	PS/2 mouse installed
3	Not used
4,5	Initial video mode: 00 = EGA/VGA 01 = 40x25 CGA 10 = 80x25 CGA 11 = Monochrome
6,7	Diskette drives: 00 = 1 drive 01 = 2 drives 10 = 3 drives 11 = 4 drives
8	Not used
9-11	Number of serial adapters
12	Game Adapter installed
13	Not used
14,15	Number of parallel adapters

16.6.4 Interrupt 12h–Return Memory Size

Returns the amount of system memory determined during the power on diagnostics.

Interrupt 12 Return System Memory Size

Exit:

AX Number of 1-kilobyte memory blocks

16.6.5 Interrupt 13h–Diskette Services

Interrupt 13 is the BIOS software interface for access to the 5-¼" and 3-½" inch diskette drives. When there is a fixed disk in the system, the BIOS assigns Interrupt 13h to the fixed disk and routes diskette calls to Interrupt 40h.
The following table lists the AH error codes.

Int 13 Diskette Exit Status Codes

AH 00h = No error

If Carry = 1:

AH 01h = Illegal BIOS command
02h = Bad address mark
03h = Write-protect occurred
04h = Sector not found
06h = Media changed
09h = DMA crossed 64K boundary
08h = DMA failed
0Ch = Media not found
10h = CRC failed
20h = NEC failed
30h = Drive does not support media sense
31h = No media in drive
32h = Drive does not support media type
40h = Seek failed
80h = Time out occurred

The following table contains the combinations of drive types and media types supported by the INT 13 services 02h to 05h.

Diskette Types

Media	Drive	Sec/Trk	Tracks
360 kB	360 kB	8-9	40
360 kB	1.2 MB	8-9	40
1.2 MB	1.2 MB	15	80
720 kB	720 kB	9	80
720 kB	1.44 MB	9	80
1.44 MB	1.44 MB	18	80
720 kB	2.88 MB	9	80
1.44 MB	2.88 MB	18	90
2.88 MB	2.88 MB	36	80

The following describes the diskette services with their entry and exit values.

Interrupt 13h Diskette Services

AH = 00h **Reset diskette system**

AH = 01h **Return diskette status**

Exit:

AH
00h = No error
01h = Illegal BIOS command
02h = Address mark not found
03h = Write-protect error
04h = Sector not found
06h = Media has been changed
08h = DMA overrun
09h = DMA boundary error
0Ch = Media not found
10h = CRC error
20h = NEC error
40h = Seek error
80h = Time out occurred

AH = 02h **Read diskette sectors**

Entry:

ES:BX Buffer address
DL Drive number (0-1)
DH Head number (0-1)
CH Track number (0-79)
CL Sector number (8-36)
AL Number of sectors (1-15)

Exit:

AL Number of sectors transferred

AH = 03h **Write diskette sectors**

Entry:

ES:BX Buffer address
DL Drive number (0-1)
DH Head number (0-1)
CH Track number (0-79)
CL Sector number (8-36)
AL Number of sectors (1-15)

Exit:

AL Number of sectors transferred

AH = 04h **Verify diskette sectors**

Entry:

DL Drive number (0-1)
DH Head number (0-1)
CH Track number (0-79)
CL Sector number (8-36)
AL Number of sectors (1-15)

Exit:

AL Number of sectors verified

Continued

Continued

AH = 05h Format diskette track
Entry:
ES:BX Buffer address
DL Drive number (0-1)
DH Head number (0-1)
CH Track number (0-79)
CL Sector number (8-36)
AL Number of sectors (1-15)
Exit:
AL Number of sectors formatted

AH = 08h Read drive parameters
Entry:
DL Drive number
Exit:
ES:DI Pointer to parameter table
DH Maximum head number
DL Number of diskette drives present
CH Maximum track number
CL Drive capacity:
Bits 0-5 Maximum sector number
Bits 6-7 Maximum track number
BL Diskette drive type from CMOS:
Bits 0-3:
00 = CMOS not present or invalid
01 = 360 kB
02 = 1.2 MB
03 = 720 kB
04 = 1.44 MB
06 = 2.88 MB
Bits 4-7: 0

AH = 15h Read drive type
Entry:
DL Drive number
Exit:
AH 00 = Drive not present
01 = Drive cannot detect media change
02 = Drive can detect media change
03 = Fixed disk

AH = 16h Detect media change
Entry:
DL Drive Number (0-1)
Exit:
If Carry = 0:
AH 00 = Disk change not active
01 = Invalid drive number
06 = Either disk change line active or
change line not supported
80h = Drive not ready or no drive present:
(timeout)

AH = 17h Set diskette type
Entry:
AL Format:
00 = Invalid Request
01 = 360kB floppy in 360kB drive
02 = 360kB floppy in 1.2MB drive
03 = 1.2MB floppy in 1.2MB drive
04 = 720kB floppy in 720kB (1.44MB not supported)
DL Drive Number (0-1)

AH = 18h Set media type for format
Entry:
CH Maximum track number
CL Diskette parameters:
Bits 0-5: Maximum sector number
Bits 6-7: Maximum track number
DL Drive Number (0-1)
Exit:
ES:DI Pointer to parameter table

AH = 20h Get media type
Entry:
DL Drive number (0-1)
Exit:

Continued

AL	Type of media installed: 00h = 720 kB diskette 01h = 1.44 MB diskette 02h = 2.88 MB diskette 03h = 1 MB diskette 04h = 2 MB diskette 06h = 4 MB diskette

16.6.6 Interrupt 13h–Fixed Disk Services

Interrupt 13h acceses these Services:

- Standard Fixed-Disk Services, 00h-15h
- Enhanced Disk Drive Services, 41h -48h
- Bootable CD-ROM Services, 4Ah-4Dh

The following box lists the error codes:

Int 13h Fixed-Disk Exit Codes

AH	00h = No error If Carry = 1:
AH	01 = Bad command or parameter 02h = Address mark not found 04h = Sector not found 05h = Reset failed 07h = Drive parameter activity failed 0Ah = Bad sector flag detected 10h = ECC data error 11h = ECC data corrected 20h = Controller failure 40h = Seek failed 80h = Time out occurred AAh = Drive not ready BBh = Undocumented controller error CCh = Controller write fault E0h = Unrecognized controller error

The following describes the Standard Fixed-Disk services of PhoenixBIOS 4.0:

Interrupt 13 Standard Fixed Disk Services

AH = 00	**Reset diskette and fixed-disk systems**
AH = 01h	**Read disk status**
Entry:	
DL	Drive number (80h-81h)
Exit:	
AH	001h = Bad command 002h = Bad address mark 004h = Record not found 005h = Controller reset error 007h = Drive initialization error 00Ah = Bad sector 010h = ECC data error 020h = Controller failed 040h = Seek error 0AAh = Drive not ready 0BBh = Invalid controller error 0CCh = Controller write fault 0E0h = Unrecognized controller error

Continued

AH = 02h Read disk sectors

Entry:
ES:BX Buffer address
DL Drive number (80h-81h)
DH Head number (0-15)
CH Track number (0-1023)
Put the two high-order bits (8 and 9)
in the high-order bits of CL
CL Sector number (1-17)
AL Number of sectors (1-80h for read)
(1-79h for long read, includes ECC)
Exit:
AL Number of sectors transferred

AH = 03h Write disk sectors

Entry:
ES:BX Buffer address
DL Drive number (80H-81H)
DH Head number (0-15)
CH Track number (0-1023)
Put the two high-order bits (8 and 9)
in the high-order bits of CL
CL Sector number (1-17)
AL Number of sectors (1-80h for write)
(1-79h for long write, includes ECC)
Exit:
AL Number of sectors transferred

AH = 04h Verify disk sectors

Entry:
ES:BX Buffer address
DL Drive number (80h-81h)
DH Head number (0-15)
CH Track number (0-1023)
Put the two high-order bits (8 and 9)
in the high-order bits of CL
CL Sector number (1-17)
AL Number of sectors (1-80h for write)
(1-79h for long write, includes ECC)
Exit:
AL Number of sectors verified

AH = 05h Format disk cylinder

Entry:
ES:BX Pointer to table containing the
following byte pair for each sector
in the track:
Byte 0: 00h if sector is good
80h if sector is bad
Byte 1: Sector Number (0-255)
DL Drive number (80H-81H)
DH Head number (0-15)
CH Track number (0-1023)
Put the two high-order bits (8 and 9)
in the high-order bits of CL
CL Sector number (1-17)
AL Number of sectors (1-80h for write)
(1-79h for long write, includes ECC)
Exit:
AL Number of sectors formatted

AH = 08h Read drive parameters

Entry:
DL Drive number (80H-81H)
Exit:
CL Maximum sector number
CH Maximum cylinder number
(High bits in CL)
DH Maximum head number
DL Number of responding drives (0-2)
If Carry - 1:
AH 07h = Invalid drive number
AL 0 = Error
CX 0 = Error
DX 0 = Error

Continued

AH = 09h **Initialize drive parameters**
Entry:
DL Drive number (80H-81H)

AH = 0Ah **Read long sector**
Entry:
ES:BX Buffer address
DL Drive number (80H-81H)
DH Head number
CH Cylinder number
CL Sector number/Cyl high
AL Number of sectors
Exit:
AL Number of sectors transferred

AH = 0Bh **Write long sector**
Entry:
ES:BX Buffer address
DL Drive number (80H-81H)
DH Head number
CH Cylinder number
CL Sector number/Cyl high
AL Number of sectors
Exit:
AL Number of sectors transferred

AH = 0Ch **Seek drive**
Entry:
ES:BX Buffer address
DL Drive number (80H-81H)
DH Head number
CH Cylinder number
CL Cylinder high

AH = 0Dh **Alternate disk reset**
Entry:
DL Drive number (80H-81H)

AH = 10h **Test drive ready**
Entry:
DL Drive number (80H-81H)

AH = 11h **Recalibrate drive**
Entry:
DL Drive number (80H-81H)

AH = 14h **Controller diagnostic**
Entry:
DL Drive number (80H-81H)

AH = 15h **Read drive type**
Entry:
DL Drive number (80H-81H)
Exit:
AH 00 = Drive not present
01 = Drive cannot detect media change
02 = Drive can detect media change
03 = Fixed disk
CX High word of number of 512-byte blocks
DX Low word of number of 512-byte blocks

16.6.7 Interrupt 13h–Extended Fixed Disk Services

The following describes the Interrupt 13h Extended Fixed Disk Services, including the *Phoenix*BIOS **Enhanced Disk Drive** (**EDD**) services:

Int 13h Extended Fixed Disk Services

AH = 41h **Check Extensions Present**

Entry:
BX 55AAh
DL Drive Number
Exit:
AH Major version number (20h)
AL Internal use only
BX 55AAh = Extensions present
CX Feature support map:
Bit 0: 1 = Extended disk access
Bit 1: 1 = Removable drive control
Bit 2: 1 = Enhanced Disk Drive Extensions
Bits 3-7, Reserved, must be 0

AH = 42h **Extended Read**

Entry:
DL Drive Number
DS:SI Disk address packet

AH = 43h **Extended Write**

Entry:
AL Verify Bits:
Bit 0: 0 = Write with verify off
1 = Write with verify on
Bits 1-7 Reserved, set to 0
DL Drive number
DS:SI Disk address packet

AH = 44h **Verify Sectors**

Entry:
DL Drive number
DS:SI Disk address packet

AH = 47h **Extended Seek**

Entry:
DL Drive number
DS:SI Disk address packet

AH = 48h **Get Drive Parameters**

Entry:
DL Drive Number
DS:SI Address of Result Buffer
Exit:
DS:SI Pointer to Result Buffer:

```
info_size  dw 30        ;size of this buffer
flags  dw ?             ;info flags (See below)
cylinders  dd ?         ;cylinders on disk
heads  dd ?             ;heads on disk
sec_per_track  dd ?     ;sectors per track
sectors  dq ?           ;sectors on disk
sector_size  dw ?       ;bytes per sector
extended_table dd?      ;extended table ptr
                        ; (See below)
```

info flags:
Bit 00 = DMA boundary errors possible
1 = DMA errors handled
Bit 10 = CHS info not supplied
1 = CHS info valid
Bit 20 = Drive not removable
1 = Drive removable
Bit 30 = No write with verify
1 = Write with verify
Bit 40 = No change-line support
1 = Change-line support
Bit 50 = Drive not lockable
1 = Drive lockable
Bit 60 = CHS values for installed media
1 = Maximum CHS values for drive

Continued

(media absent)

Extended Fixed Disk Parameter Table

Byte	Type	Description
0-1	Word	I/O port address
2-3	Word	Control port address
4	Bit 0-3	Reserved, must be 0
	Bit 4	0 = Master, 1 = Slave
	Bit 5	Reserved, must be 0
	Bit 6	1 = LBA enabled
	Bit 7	Reserved, must be 1
5	Bits0-3	Phoenix Proprietrary
	Bits 4-7	Reserved, must be 0
6	Bits 0-3	IRQ for this drive
	Bits 4-7	Reserved, must be 0
7	Byte	Sector count for multi-sectored transfers
8	Bits 0-3	DMA channel
	Bits 4-7	DMA type
9	Bits 0-3	PIO type
	Bits 1-7	Reserved, must be 0

Byte	Type	Description
10-11	Bit 01 = Fast PIO access enabled	
	Bit 11 = DMA access enabled	
	Bit 21 = Block PIO access enabled	
	Bit 31 = CHS translation enabled	
	Bit 41 = LBA translation enabled	
	Bit 51 = Removable media	
	Bit 6	1 = CD ROM
	Bit 7	1 = 32-bit transfer mode
	Bit 81 = ATAPI Device uses Interrupt DRQ	
	Bits 9-10	CHS Translation Type
	Bits 11-15	Reserved, must be 0
12-13	Byte	Reserved, must be 0
14	Byte	Extension Revision number
15	Byte	Checksum, 2s complement of the sum of bytes 0-14

16.6.8 Interrupt 13h–Bootable CD-ROM Services

Bootable CD-ROM Services 4Ah-4Ch use a pointer to the **Specification Packet**, described here:

Bootable CD-ROM Specification Packet

Offset	Type	Description
0h	Byte	Packet size, currently 13h
1h	Byte	Boot media type: Bits 0-3: 00h = No emulation 01h = 1.2 MB diskette 02h = 1.44 MB diskette 03h = 2.88 MB diskette 04h = Hard disk (drive C:) Bits 05h-07h: Reserved Bit 6: 01h = Sysem has ATAPI driver with 8 & 9 below describing IDE interface. Bit 7: 01h = System has SCSI drivers with 8 & 9 below describing SCSI interface
2h	Byte	Drive number: 00h = Floppy image 80 = Bootable hard disk 81h -FFh = "Non-bootable" or "No emulation"

Continued

Continued

Offset	Type	Description
3h	Byte	Controller index of CD drive
4h-7h	Dword	Logical Block Address
8h-9h	Word	Device specification: For SCSI: Byte 8: LUN and PUN of CD drive Byte 9: Bus number For IDE: Byte 8 LSB: 0 = Master, 1 = Slave
Ah-Bh	Word	User buffer segment
Ch-Dh	Word	Load segment (only for Int 13h 4Ch): 00h = 7C0h
Eh-Fh	Word	Virtual sector count (only for Int 13h 4Ch)
10h	Byte	Low-order bits (0-7) of the cylinder count (Matches returned CH of Int 13h 08h)
11h	Byte	Bits 0-5: Sector count Bits 6-7: High order 2 bits of cylinder count (Matches returned CL of Int 13h 08h)
12h	Byte	Head count (Matches returned DH of Int 13h 0h)

Bootable CD-ROM Service 4Dh uses a pointer to the **Command Packet**, described here:

Bootable CD-ROM Command Packet

Offset	Type	Description
0h	Byte	Packet size in bytes, currently 08h
1h	Byte	Count of sectors in boot catalog to transfer
2-h	Dword	Pointer to destination buffer for boot catalog
6-7h	Word	Beginning sector to transfer, relative to start of the boot catalog. Int 14 4Dh should set this value to 00h.

The following describes the Interrupt 13 Bootable CD-ROM Services of PhoenixBIOS 4.0:

Int 13 Bootable CD-ROM Services

AH = 4Ah	**Initiate disk emulation**
Entry:	
AL	00
DS:SI	Pointer to Specification Packet (See above)
CF	0 = Specified drive emulating 1 = System not in emulation mode
AH = 4Bh	**Terminate disk emulation**
Entry:	
AL	00h = Return status and terminate emulation 01h = Return status only, do not terminate
DL	Drive number to terminate 7Fh = Terminate all
DS:SI	Empty Specification Packet
Exit:	
DS:SI	Completed Specification Packet (See above)
AX	Exit status codes
CF	0 = System released 1 = System not in emulation mode
AH = 4Ch	**Initiate disk emulation and boot**
Entry:	
AL	00h
DS:SI	Specification Packet (See above)
AH = 4Dh	**Return boot catalog**
Entry :	
AL	00h
DS:SI	Point to Command Packet (See above)

16.6.9 Interrupt 14h–Serial Services

The INT 14 software interrupt handles serial I/O service requests. Use the AH register to specify the service to invoke. This describes the UART Modem and Line Status returned by these services. It also includes two services, 04h and 05h, that support the extended communication capabilities of PS/2. The following describes the modem status returned by serial services.

Modem Status

AL	Description
Bit 0	1 = Delta clear to send
Bit 1	1 = Delta data set ready
Bit 2	1 = Trailing edge ring indicator
Bit 3	1 = Delta data carrier detect
Bit 4	1 = Clear to send
Bit 5	1 = Data set ready
Bit 6	1 = Ring indicator
Bit 7	1 = Received line signal detect

The following describes the line status returned by Int 14h Serial Services.

Line Status

AH	Description
Bit 0	1 = Data ready
Bit 1	1 = Overrun error
Bit 2	1 = Parity error
Bit 3	1 = Framing error
Bit 4	1 = Break detect
Bit 5	1 = Trans holding register empty
Bit 6	1 = Trans shift register empty
Bit 7	1 = Time out error

The following describes the serial communication services of *Phoenix*BIOS 4.0:

Interrupt 14h Serial Services

AH = 00 Initialize Serial Adapter

Entry:

AL Init parameters:

Bit 1,0 10 = 7 data bits
11 = 8 data bits

Bit 20 = 1 stop bit
1 = 2 stop bits

Bit 4,3 00 = No parity
10 = No parity
01 = Odd parity
11 = Even parity

Bit 7-5 000 = 110 Baud- 417 divisor
001 = 150 Baud-300 divisor
010 = 300 Baud-180 divisor
011 = 600 Baud-0C0 divisor
100 = 1200 Baud-060 divisor
101 = 2400 Baud-030 divisor
110 = 4800 Baud-018 divisor
111 = 9600 Baud-00C divisor

DX Serial port (0-3)

Exit:

AL Modem status

AH Line status

Continued

AH = 01h Send character

Entry:
AL Character to transmit
DX Serial port (0-3)
Exit:
AH Line status

AH = 02h Receive character

Entry:
DX Serial port (0-3)
Exit:
AL Character received
AH Line Status

AH = 03h Return serial port status

Entry:
DX Serial port (0-3)
Exit:
AH Line status
AL Modem status

AH = 04h Extended Initialize (PS/2)

Entry:
DX 0-3 = Communications adapter
AL 00 = Break
01 = No break
BH Parity:
00 = None
01 = Odd
02 = Even
03 = Stick parity odd
04 = Stick parity even
BL Stop bits:
00 = One
01 = Two if 6,7, or 8-bit word length
One and one-half if 5-bit word length
CH Word length:
00 = 5 bits
01 = 6 bits
02 = 7 bits
03 = 8 bits
CL Baud rate:
00 = 110 baud
01 = 150 baud
02 = 300 baud
03 = 600 baud
04 = 1200 baud
05 = 2400 baud
06 = 6000 baud
07 = 9600 baud
08 = 19200 baud
Exit:
AL Modem status
AH Line status

AH = 05h Extended Communications Port Control (PS/2)

AL = 00 Read modem control register

Entry:
DX Serial port (0-3)
Exit:
BL Modem control register

AL = 01 Write modem control register

Entry:
DX Serial port (0-3)
BL Modem control register
Exit:
AL Modem status
AH Line status

16.6.10 Interrupt 15h–System Services

The INT 15 software interrupt handles a variety of system services:

- Multi-tasking–80h, 81h, 82h, 85h, 90h, and 91h
- Joystick support–84h
- Wait routines–83h and 86h
- Protected-mode support–87h and 89h
- Report extended memory to 64 kB–88h
- System information–C0h
- Advanced Power Management (optional)–53h
- Report extended memory above 64 kB (optional)–8Ah and E8h
- PS/2 Mouse support (optional)–C2h
- EISA Support (optional)–D8h

The first section describes the standard Interrupt 15 services, followed by separate sections describing each of the optional services.

Interrupt 15h System Services

AH = 00-03h Cassette services
Entry:
No longer supported
Exit:
Carry 1 = Not supported

AH = 80h Device open
Entry:
BX Device identifier
CX Process identifier

AH = 81h Device close
Entry:
BX Device identifier
CX Process identifier

AH = 82h Program termination
Entry:
BX Device identifier

AH = 83h Event wait

AL 00 = Set interval
Entry:
ES:BX Pointer to byte in caller's memory that will have bit 7 set when interval expires.
CX Microseconds before post (high byte)
DX Microseconds before post (low byte)
Exit:
AH 83h
AL A value written to CMOS register B
00h = Function busy

AL 01 = Cancel set interval
Exit:
AH 83
AL 00

AH = 84h Joystick support
Entry:
DL 00 = Read switch settings
Exit:
AL Switch settings
DL 01 Return resistive inputs
Exit:
AX Input bit 0 (Joystick A, x coordinate)
BX Input bit 1 (Joystick A, y coordinate)
CX Input bit 2 (Joystick B, x coordinate)
DX Input bit 3 (Joystick B, y coordinate)

Continued

AH = 85h **System request key pressed**
Entry:
AL 00 System request key pressed
AL 01 System request key released

AH = 86h **Wait**
Entry:
CX Number of microseconds to wait (high byte)
DX Number of microseconds to wait (low byte)

AH = 87h **Extended memory move block**
Entry:
CX Number of words to move
ES:SI Pointer to Global Descriptor
Byte 0-1 Bits 0-15 of Segment Limit
Byte 2-3 Bits 0-15 of Base Address
Byte 4 Bits 16-23 of Base Address
Byte 5 Access Rights
Byte 6 Bits 7-4 more Access Rights
Bits 3-0 upper 4 bits of Segment Limit
Byte 7 Bits 24-31 of Base Address
(See Intel programmer's reference)

AH = 88h **Extended memory size**
Exit:
For DOS and Windows 3.x (AT Compatible):
AX Amount of extended memory to 64 MB, in 1 kB blocks
For Windows NT 3.1 and OS/2 2.11 and 2.20:
AX Amount of extended memory to 64 MB in 1 kB blocks
3C00 = 15 MB or > 64 MB (Test further with INT 15 E8)

AH = 89h **Enter protected mode**
Entry:
ES:SI Pointer to Global Descriptor (See service 87)
BH Offset in IDT for IRQ 00-07
BL Offset in IDT for IRQ 08-0F

AH = 90h **Device busy**
Entry:
AL Type code:
00h = Fixed disk (May time out)
01h = Diskette (May time out)
02h = Keyboard (No time out)
03h = Pointing device (May time out)
80h = Network (No time out)
FCh = Fixed disk reset (May time out)
FDh = Diskette drive motor start (May time out)
FEh = Printer (May time out)
ES:BX Points to request block if AL = 80h-FFh
Exit:
Carry 0 = No wait performed
(Driver must perform own wait)
1 = Wait performed (I/O complete or time out)

AH = 91h **Interrupt complete**
Entry:
AL Type code: See service 90h

Continued

Continued

AH = C0h Return system parameters

Exit:

ES:BX Pointer to System Configuration Bytes 1-2 Length of table in bytes (8)

Byte 3 Model (FCh = AT Byte 4 Sub model (01h = AT)

Byte 5 BIOS revision level (0)

Byte 6 Feature information:

Bit 0 0 = Reserved

Bit 1 0 = ISA-type I/O channel

Bit 2 0 = EDBA not allocated

Bit 3 0 = Wait for external event supported

Bit 4 1 = Keyboard intercept (INT 154F) called by INT 09h

Bit 5 1 = Real time clock present

Bit 6 1 = Second PIC present

Bit 7 0 = Fixed disk BIOS does not use DMA channel 3

Byte 7 Reserved

Byte 8 Reserved

AH = C1h Return Extended BIOS Data Area Address

Exit:

ES Extended BIOS Data Area Segment Address

If Carry = 1

AH 86 = Invalid BIOS routine call (No EBDA)

16.6.11 Interrupt 15h–APM Services

The INT 15 software interrupt optionally handles the calls supporting APM (Advanced Power Management).
The following are the APM exit status codes:

APM Service Exit Status Codes

AH 00h = No error

If Carry = 1:

AH 01h = Power Management disabled

02h = Real Mode interface already connected

03h = Interface not connected

05h = 16-bit protected mode interface already connected

06h = 16-bit protected mode interface not supported

07h = 32-bit protected mode interface already connected

08h = 32-bit protected mode interface not supported

09h = Unrecognized Device ID

0Ah = Parameter value out of range

0Bh = Interface not engaged

60h = Unable to enter requested state

80h = No PM events pending

86h = No APM present

following are the Interrupt 15 APM Services of *Phoenix*BIOS 4.0:

Interrupt 15h APM Services

AH = 53h APM 1.0 and APM 1.1 BIOS Services

AL = 00h Installation Check

Entry:

BX 0000h = Power Device ID (APM BIOS)
All other values reserved

Exit:

AH APM major revision in BCD
AL APM minor revision in BCD
BH ASCII "P"
BL ASCII "M"
CX APM information:
Bit 01 = 16 bit Prot Mode supported
Bit 11 = 32 Bit Prot Mode supported
Bit 21 = CPU IDLE slows down CPU speed.
Requires APM CPU Busy service
Bit 31 = BIOS Power Management is disabled
Bit 41 = APM disengaged

AL = 01h Interface Connect

Entry:

BX 0000h = Power Device ID (APM BIOS)
All other values reserved

AL = 02h Protected-mode 16-bit interface connect

Entry:

BX 0000h = Power Device ID (APM BIOS)
All other values reserved

Exit:

AX APM 16-bit code segment (real mode segment base address)
BX Offset of entry point into the BIOS
CX APM 16-bit data segment (real mode segment address)
SI BIOS code segment length
DI BIOS data segment length

AL = 03h Protected-mode 32-bit interface connect

Entry:

BX Power Device ID, 0000h
All other values reserved

Exit:

AX APM 32-bit code segment (real mode segment base address)
EBX Offset of entry point into the BIOS
CX APM 16-bit data segment (real mode segment address)
DX APM data segment (real mode segment address)
SI BIOS code segment length
DI BIOS data segment length

AL = 04h Protected-mode 32-bit interface connect Fehler! Verweisquelle konnte nicht gefunden werden.

Entry:

BX 0000h = Power Device ID (APM BIOS)
All other values reserved

AL = 05h CPU Idle

AL = 06h CPU busy

Continued

AL = 07h Set Power State

Entry:

BX Power Device ID:
0001h = All PM devices managed by the BIOS
01XXh = Display
02XXh = Secondary Storage
03XXh = Parallel Ports
04XXh = Serial Ports
05XXh = Network Adapters
06XXh = PCMCIA Sockets
E000h-EFFFh = OEM-defined power-device IDs
where:
XXh = Unit Number (0 based)
Unit Number FFh = all units in this class

CX Power State:
*0000h = APM enabled
0001h = Standby
0002h = Suspend
0003h = Off
**0004h = Last Request Processing Notification
**0005h = Last Request Rejected
0006h-001Fh = Reserved system states
0020h-003Fh = OEM-defined system states
0040h-007Fh = OEM-defined device states
0080-FFFFh = Reserved device states
* Not supported for Power Device ID 0001h
**Only supported for Power Device ID 0001h

AL = 08h Enable/disable power management

Entry:

BX Power Device ID:
0001h = All PM devices controlled by the BIOS
FFFFh = All PM devices controlled by the BIOS (For compatibility with APM 1.0)
All other values reserved

CX Function code:
0000h = Disable power management
0001h = Enable power management

AL = 09h Restore Power-On Defaults

Entry:

BX Power Device ID:
0001h = All PM devices managed by the BIOS
FFFFh = All PM devices managed by the BIOS (For compatibility with APM 1.0)
All other values reserved

AL = 0Ah Get Power Status

Entry:

BX Power Device ID, 0000h = APM BIOS
All other values reserved

Exit:

BH AC line status:
00h = Off line
01h = On line
02h = On backup power
FFh = Unknown
All other values reserved

BL Battery status:
00h = High
01h = Low
02h = Critical
03h = Charging
FFh = Unknown

CL Percentage of charge remaining:
0-100 = Percentage of full charge
FFh = Unknown
All other values reserved

Continued

AL = 0Bh Get PM Event

Exit:
BX PM event code

AL = 0Ch Get Power State

Entry:
BX Power Device ID:
0001h = All PM devices managed by the BIOS
01XXh = Display
02XXh = Secondary Storage
03XXh = Parallel Ports
04XXh = Serial Ports
05XXh = Network Adapters
06XXh = PCMCIA Sockets
E000h-EFFFh = OEM-defined power-device IDs
All other values reserved
where:
XXh = Unit Number (0 based)

AH = 53h APM 1.1 BIOS Services

AL = 0Dh Enable/Disable power management (APM 1.1 only)

Entry:
BX Power Device ID:
0001h = All PM devices managed by the BIOS
01XXh = Display
02XXh = Secondary Storage
03XXh = Parallel Ports
04XXh = Serial Ports
05XXh = Network Adapters
06XXh = PCMCIA Sockets
E000h-EFFFh = OEM-defined power-device IDs
All other values reserved
where:
XXh = Unit Number (0 based)

AL = 0Eh APM Driver Version (APM 1.1 only)

Entry:
BX 0000h = BIOS device
CH APM Driver major version number (BCD)
CL APM Driver minor version number (BCD)
Exit:
AH APM Connection major version number (BCD)
AL APM Connection minor version number (BCD)

AL = 0Fh Engage/disengage power management (APM 1.1 only)

Entry:
BX Power Device ID:
0001h = All PM devices managed by the BIOS
01XXh = Display
02XXh = Secondary Storage
03XXh = Parallel Ports
04XXh = Serial Ports
05XXh = Network Adapters
06XXh = PCMCIA Sockets
E000h-EFFFh = OEM-defined power-device IDs
All other values reserved
where:
XXh = Unit Number (0 based)
Unit Number FFh = all devices in this class
CX Function code:
0000h = Disengage power management
0100h = Engage power managment

16.6.12 Interrupt 15h–Big Memory Services

The INT 15 software interrupt optionally handles the calls reporting extended memory over 64 MB.
The first function, 8Ah, only supports certain versions of UNIX.
The second function, E8h, incorporates these sub functions:

- Big memory for Windows NT 3.01 and OS/2 2.11 and 2.20–E801h (16 bit) and E881h (32 bit).
- System Memory Map–E820h

Interrupt 15h Big Memory Services

AH = 8Ah Big Memory size, Phoenix definition

Entry:
(For certain versions of UNIX)
AX Low 16-bit value
DX High 16-bit value
= memory above 1024 kB in 1 kB blocks

AH = E8h Big Memory size (over 64 kB)

AL = 01h Big Memory Size, 16 Bit
(Windows NT 3.1 and OS/2 2.11 and 2.20)

Exit:
Carry 0 = E801 Supported
AX Memory 1 MB to 16 MB, in 1 kB blocks
BX Memory above 16 MB, in 64 kB blocks
CX Configured memory 1 MB to 16 MB, in 1 kB blocks
DX Configured memory above 16 MB, in 64 kB blocks

AL = 20h System Memory Map

Entry:
EBX Continuation value
ES:DI Address of Address Range Descriptor
ECX Length of Address Range Descriptor (=> 20 bytes)
EDX "SMAP" signature
Exit:
Carry 0 = E820 Supported
EAX "SMAP" signature
ES:DI Same value as entry
ECX Length of actual reported information in bytes
EBX Continuation value

Structure of Address Range Descriptor:
Bytes 0-3 Low 32 bits of Base Address
Bytes 4-7 High 32 bits of Base Address
Bytes 8-11 Low 32 bits of Length in bytes
Bytes 12-15 High 32 bits of Length in bytes
Bytes 16-20 Type of Address Range:
1 = AddressRangeMemory, available to OS
2 = AddressRangeReserved, not available
3 = AddressRangeACPI, available to OS
4 = AddressRangeNVS, not available to OS
Other = Not defined, not available

NOTE: Each call of this service defines a descriptor buffer and requests the memory status of the address range specified by the continuation value, where zero = first address range. The function fills the buffer and returns the continuation value for the next address range range, where zero = last address range.

AL = 81h Big Memory Size, 32-Bit Protected Mode
(Windows NT 3.1 and OS/2 2.11 and 2.20)

Exit:
Carry 0 = E881 supported
EAX Memory 1 MB to 16 MB, 1 kB blocks
EBX Memory above 16 MB, 64 kB blocks
ECX Configured memory 1 MB to 16 MB, 1 kB blocks
EDX Configured memory above 16 MB, 64 kB blocks

16.6.13 Interrupt 15h–PS/2 Mouse Services

The INT 15 software interrupt optionally supports systems with the PS/2 mouse or similar devices installed on the motherboard. The following table describes the exit status codes:

PS/2 Mouse Exit Status Codes	
AH	00h = No error 01h = Invalid function call 02h = Invalid input value 03h = Interface error 04h = Request for resend received from 8042 05h = No driver installed (i.e., Function C207 has not been called)

The following table describes the Interrupt 15h PS/2 mouse services of *Phoenix*BIOS 4.0:

Interrupt 15h PS/2 Mouse Services	
AH = C2h	**PS/2 Mouse Support**
AL Entry: BH	**00 = Enable/Disable PS/2 Mouse** 00h = Disable 01h = Enable
AL Exit: BH	**01 = Reset PS/2 Mouse** Device ID
AL	**02 = Set Sample Rate**
Entry: BH	 Sample rate: 00h = 10 reports per second 01h = 20 reports per second 02h = 30 reports per second 03h = 40 reports per second 04h = 60 reports per second 04h = 80 reports per second 05h = 100 reports per second 06h = 200 reports per second
AL Entry: BH	**03h = Set resolution** Resolution value: 00h = 1 count per millimeter 01h = 2 counts per millimeter 02h = 4 counts per millimeter 03h = 8 counts per millimeter
AL Exit: BH	**04h = Read Device Type** Device ID
AL Entry: BH	**05h = Initialize PS/2 mouse** Data package size (01-08h, in bytes)
Interrupt 15h-PS/2 Mouse Services,	

continued

AL 06h = Set Scaling or Get Status

Entry:
BH 00 = Return status (See Exit Status below)
01 = Set Scaling Factor to 1:1
02 = Set Scaling Factor to 2:1

Exit:
If Entry BH = 00:
BL Status byte 1:
Bit 01 = Right button pressed
Bit 10 = Reserved
Bit 21 = Left button pressed
Bit 30 = Reserved
Bit 40 = 1:1 Scaling
1 = 2:1 Scaling
Bit 50 = Disable
1 = Enable
Bit 60 = Stream mode
1 = Remote mode
Bit 70 = Reserved

CL Status byte 2:
00h = 1 count per millimeter
01h = 2 counts per millimeter
02h = 4 counts per millimeter
03h = 8 counts per millimeter

DL Status byte 3:
0Ah = 10 reports per second
14h = 20 reports per second
28h = 40 reports per second
3Ch = 60 reports per second
50h = 80 reports per second
64h = 100 reports per second
C8h = 200 reports per second

AL 07 = Set PS/2 mouse driver address

Entry:
ES:BX Pointer to mouse driver

16.6.14 Interrupt 15h–EISA Services

The INT 15 software interrupt optionally supports systems with EISA (Extended Industry Standard Architecture) with these services:

Read slot configuration information–D800h, D880h
Read function configuration information–D801h, D881h
Clear EISA CMOS–D802h , D882h
Write slot configuration information to EISA CMOS–D803h, D883h
Read physical slot information–D804, D884h

The EISA BIOS services accommodate real and protected mode and 16 and 32-bit addressing. See the EISA specifications for descriptions of these services.
The following are the exit status codes for the Int 15 EISA services:

Int 15 EISA Exit Status Codes

AH 00h = No error
If Carry = 1
AH 80h = Invalid slot number
81h = Invalid function number
82h = Extended CMOS corrupted
83h = Empty slot specified
84h = Error writing to CMOS
85h = CMOS is full
86h = Invalid BIOS routine call
87h = Invalid system configuration
88h = Configuration utility not supported

The following are the Interrupt 15 EISA services of *Phoenix*BIOS 4.0:

Interrupt 15h EISA Services

AH = D8h **Access EISA System Information**

AL **00h = Read slot config information**
80h = Read slot config information, 32 bit

Entry:
CL Slot number (0-63)
Exit:
AL Vendor information byte:
Bits 3-0 Duplicate ID number:
0000 = No duplicate ID
0001 = First duplicate ID
Bits 5-4 Slot type:
00 = Expansion slot
01 = Embedded device
10 = Virtual device
11 = Reserved
Bit 6Product ID:
00 = Readable
01 = Not readable
Bit 7Duplicate ID:
00 = No duplicate ID
01 = Duplicate IDs
BH Major revision level of config utility
BL Minor revision level of config utility
CH MSbyte of checksum of config file
LSbyte of checksum of config file
DH Number of device functions
DL Combined function information byte:
Bit 7Reserved
Bit 6Slot has free-form data entries
Bit 5Slot has port initialization entries
Bit 4 Slot has port range entries
Bit 3Slot has DMA entries
Bit 2Slot has IRQ entries
Bit 1Slot has memory entries
Bit 0Slot has function type entries
DI First word of compressed device ID
SI Second word of compressed device ID
(See "Read physical slot information" below)

AL **01h = Read function config information**
81h = Read function config information, 32 bit

Entry:
CH Function number (0 to n-1)
CL Slot number (0-63)
DS:SI Pointer to output data buffer
Exit:
DS Segment for return data buffer
SI Offset to return data buffer (16 bit)
ESI Offset to return data buffer (32 bit)

AL **02h = Clear EISA CMOS configuration**
82h = Clear EISA CMOS configuration 32 bit

Entry:
BH Configuration utility major revision level
BL Configuration utility minor revision level

AL **03h = Write slot config information**
83h = Write slot config information, 32 bit

Entry:
CX Length of data structure in bytes
DS Segment of data table
SI Offset of data table (16-bit call)
ESI Offset of data table (32-bit call)

Continued	
AL	**04h = Read board ID registers** **84h = Read board ID registers, 32 bit**
Entry:	
CL	Slot number (0-63)
Exit:	
DI	First word of compressed ID: Byte 0: Bits 1-0 2nd character of manufacturer code Bits 6-2 1st character of manufacturer code Bit 7 Reserved Byte 1: Bits 4-0 3rd character of manufacturer code Bits 5-7 2nd character of manufacture code, cont.
SI	Second word of compressed ID: Byte 0: Bits 3-0 2nd hex digit of product number Bits 7-4 1st hex digit of product number Byte 1: Bits 3-0 Hex digit of revision number Bits 7-4 3rd hex digit of product number If Carry = 1:

16.6.15 Interrupt 16h–Keyboard Services

The INT 16 software interrupt handles keyboard I/O services. The following describes the keyboard services of *Phoenix*BIOS 4.0:

Interrupt 16h Keyboard Services	
AH = 00h	**Read keyboard input**
Exit:	
AL	ASCII keystroke pressed
AH	Scan code of key
AH = 01h	**Return keyboard status**
Exit:	
AL	ASCII keystroke pressed
AH	Scan code of key
ZF	No keystroke available
NZ	Keystroke in buffer
AH = 02h	**Return shift-flag status**
Exit:	
AL	Current shift status
AH = 03h	**Set typematic rate and delay.**
Entry:	
AL	05 (subfunction number)
BL	00H through 1FH, typematic rate (30 chars/sec to 2 char/sec)
BH	Delay rate: 00h = 250 ms 01h = 500 ms 02h = 750 ms 03h = 1000 ms 04h to 07h = Reserved
AH = 05h	**Add key to Keyboard buffer.**
Entry:	
CL	ASCII code
CH	Scan code
Exit:	
	If Carry = 1:
AL	Keyboard buffer full
AH = 10h	**Read extended character from buffer.**
Exit:	
AL	ASCII keystroke pressed
AH	Scan code of key

Continued

AH = 11h **Return extended buffer status.**

Exit:

AL	ASCII keystroke pressed
AH	Scan code of key
ZF	No keystroke available
NZ	Keystroke in buffer

AH = 12h **Return extended shift status.**

Exit:

AL	Shift status: Bit 71 = Sys Req pressed Bit 61 = Caps Lock active Bit 5 1 = Num Lock active Bit 4 1 = Scroll Lock active Bit 31 = Right Alt active Bit 21 = Right Ctrl active Bit 11 = Left Alt active Bit 01 = Left Ctrl active
AH	Extended shift status: Bit 71 = Insert active Bit 61 = Caps Lock active Bit 51 = Num Lock active Bit 41 = Scroll Lock active Bit 31 = Alt pressed Bit 21 = Ctrl pressed Bit 11 = Left Shift pressed Bit 01 = Right Shift pressed

16.6.16 Interrupt 17h–Parallel Printer Services

The INT 17 software interrupt supports up to 4 parallel adapters. The BIOS stores the standard base addresses for three parallel adapters in the BIOS Data Area at 3FCh, 378h, and 278h. These services use the I/O ports 0278h-027Ah, 0378h-037Ah, and 03BCh-03BEh.

Interrupt 17h Parallel Printer Services

AH = 00h **Print character**

Entry:

AL	Character to print
DX	Printer port (0-3)

Exit:

AH	Printer Status (see below)

AH = 01h **Initialize printer port**

Entry:

DX	Printer port (0-3)

Exit:

AH	Printer Status (see below)

AH = 02h **Return printer status**

Entry:

DX	Printer port (0-3)

Exit:

AH	Printer Status: Bit 01 = Time-out error Bit 1Reserved Bit 2Reserved Bit 31 = I/O error Bit 41 = Printer selected Bit 51 = Out of paper Bit 61 = Acknowledgment from printer Bit 71 = Printer not busy

16.6.17 Interrupt 17h–EPP Services

Use Interrupt 17h 02h to obtain the BIOS entry point (also called the EPP Vector) to Enhanced Parallel Printer (EPP) Services. To use the other EPP services, load AH with an appropriate function value and Far call the EPP Vector.
The following are the EPP exit status codes:

EPP Services Exit Status Codes

AH	00h = No error
	01h = Failed I/O function
	02h = Invalid function
	03h = EPP not supported
	04h = Not an EPP port
	20h = Multiplexor not present
	40h = Multiplexor already locked

The following are the Int 17 EPP services of *Phoenix*BIOS 4.0:

Interrupt 17h EPP Service

AH = 02h EPP Installation check

Entry:

DX	EPP printer port (0-2)
AL	0
CH	45h = "E"
BL	50h = "P"
BH	50h = "P"

Exit:

AL	45h
CX	5050h
DX:BX	EPP BIOS entry point

Vectored EPP Services
(Call entry point)

AH = 00h Query EPP port configuration

Entry:

DL	EPP printer port (0-2)

Exit:

AL	Interrupt level of EPP port (00-15h) FFh = Interrupts not supported
BH	EPP BIOS revision (MMMMnnnn or M.n)
BL	I/O capabilities: Bit 0Multiplexor present Bit 1PS/2 bi-directional capable Bit 2Daisy chain present Bit 3ECP capable
CX	SPP I/O base address
ES:DI	FAR pointer to EPP BIOS manufacturer's info/version text string, zero terminated

AH = 01h Set mode

Entry:

DL	EPP printer port (0-2)
AL	Modes: Bit 0Set compatibility mode Bit 1Set Bi-directional mode Bit 2Set EPP mode Bit 3Set ECP mode Bit 4Set EPP software emulation (via standard parallel port)

AH = 02h Get mode

Entry:

DL	EPP printer port (0-2)

Exit:

AL	Modes: Bit 0In compatibility mode Bit 1In Bi-directional mode Bit 2In EPP mode Bit 3In ECP mode Bit 4In EPP software-emulation mode Bit 7EPP port interrupts enabled

Continued

AH = 03h Interrupt control
Entry:
DL EPP printer port (0-2)
AL 0 = Disable EPP port interrupts
1 = Enable EPP port interrupts

AH = 04h Reset EPP port
Entry:
DL EPP printer port (0-2)

AH = 05h Write address/select device
Entry:
DL EPP printer port (0-2)
AL Device address to write

AH = 06h Read address
Entry:
DL EPP printer port (0-2)
AL Device address to write
Exit:
AL Address/device data returned

AH = 07 Write byte
Entry:
DL EPP printer port (0-2)
AL Data byte

AH = 08 Write block
Entry:
DL EPP printer port (0-2)
CX Number of bytes to write (0 = 64k)
ES:SI Client buffer w/data
Exit:
CX Bytes not transferred (0 = no error)

AH = 09h Read byte
Entry:
DL EPP printer port (0-2)
Exit:
AL Data byte returned

AH = 0Ah Read block
Entry:
DL EPP printer port (0-2)
CX Number of bytes to read (0 = 64k)
ES:DI Client buffer for returned data
Exit:
CX Bytes not transferred (0 = no error)

AH = 0Bh Write address, read byte
Entry:
DL EPP printer port (0-2)
AL Device address
Exit:
AL Data byte returned

AH = 0Ch Write address, write byte
Entry:
DL EPP printer port (0-2)
AL Device address
DH Data byte to write

AH = 0Dh Write address, read block
Entry:
DL EPP printer port (0-2)
AL Device address
CX Number of bytes to read (0 = 64k)
ES:DI Client buffer for data
Exit:
AL Returned byte data
CX Bytes not transferred (0 = no error)

AH = 0Eh Write address, write block
Entry:
DL EPP printer port (0-2)
AL Device address
CX Number of bytes to write
ES:SI Client buffer w/data
Exit:
CX Bytes not transferred (0 = no error)

Continued

AH = 0Fh Lock port

Entry:
DL EPP printer port (0-2)
BL Port address:
Bits 7-4 Daisy chain port number (1-8)
Bits 3-0 Mux device port number (1-8)
0 = No multiplexor

AH = 10h Unlock port

Entry:
DL EPP printer port (0-2)

AH = 11h Device interrupt

Entry:
DL EPP printer port (0-2)
BL The multiplexor device port (1-8)
0 = No multiplexor
AL 0 = Disable device interrupts
1 = Enable device interrupts
ES:DI Far pointer to interrupt-event handler

AH = 12h Real time mode

Entry:
AL 0 = Query if any real-time device present
1 = Add (advertise) real-time device
2 = Remove real-time device
Exit:
AL 0 = No real-time devices present
1 = One or more real-time devices present

AH = 40h Query multiplexor

Entry:
DL EPP printer port (0-2)
Exit:
AL Bit 01 = Channel locked
Bit 11 = Interrupt pending
BL Currently selected port

AH = 41h Query multiplexor device port

Entry:
DL EPP printer port (0-2)
BL The multiplexor device port (1-8)
0 = No multiplexor
Exit:
AL Status flags:
Bit 01 = Port selected
Bit 11 = Port locked
Bit 21 = Interrupts enabled
Bit 31 = Interrupt pending
CX EPP product/Device ID
0 = Undefined

AH = 42h Set product ID

Entry:
DL EPP printer port (0-2)
AL Mapped EPP Mux device port (1-8)
CX EPP Product ID

AH = 50h Rescan daisy chain

Entry:
DL EPP printer port (0-2)
BL The multiplexor device port (1-8)
0 = No multiplexor

AH = 51h Query daisy chain

Entry:
DL EPP printer port (0-2)
BL The multiplexor device port (1-8)
0 = No multiplexor
Exit:
AL Status flags:
Bit 01 = Channel locked
Bit 11 = Interrupt pending
BL Currently selected device
CL Depth of daisy chain on this port
0 = No daisy chain on this port
ES:DI Pointer to ASCII string, driver vendor ID

16.6.18 Interrupt 1Ah–Time of Day Services

The INT 1Ah software interrupt handles the time of day I/O services. A Carry flag set on exit may indicate the clock is not operating.

Interrupt 1Ah Time-of-Day Services	
AH = 00h	**Read current time**
Exit:	
CX	High word of tick count
DX	Low word of tick count
AL	00h = Day rollover has not occurred (Timer count is less than 24 hours since last power on or reset)
AH = 01h	**Set current time (Clear rollover bit)**
Entry:	
CX	High word of tick count
DX	Low word of tick count
AH = 02h	**Read real time clock**
Exit:	
CH	BCD hours
CL	BCD minutes
DH	BCD seconds
DL	00 = Standard Time 01h = Daylight Savings
AH = 03h	**Set the real time clock**
Entry:	
CH	BCD hours
CL	BCD minutes
DH	BCD seconds
DL	01h = Daylight saving 00h = Otherwise
AH = 04h	**Read date from real time clock**
Exit:	
CH	BCD century
CL	BCD year
DH	BCD month
DL	BCD date
AH = 05h	**Set date in real time clock**
Entry:	
CH	BCD century
CL	BCD year
DH	BCD month
DL	BCD date
AH = 06h	**Set real-time alarm**
Entry:	
CH	BCD hours to alarm
CL	BCD minutes to alarm
DH	BCD seconds to alarm
Exit:	
C	1 = Alarm already set
AH = 07h	**Reset real-time alarm**
Exit:	
AL	Value written to CMOS RAM register 0Bh

16.6.19 Interrupt 1Ah–General PCI Services

PhoenixBIOS 4.0 optionally supports General PCI Interrupt 1Ah Services. The following are the exit status codes:

PCI Services Exit Status Codes	
AH	00h = Successful
	If Carry = 1:
AH	81h = Function not supported
	83h = Bad vendor ID
	86h = Device not found
	87h = Bad register number
	88h = Set failed
	89h = Buffer too small

The following are the PCI Services:

Interrupt 1Ah General PCI Services	
AH = B1h	**PCI Services**
AL	**01h = PCI BIOS present**
Exit:	
EDX	"PCI", "P" in [DL], "C" in [DH], etc.
AL	Hardware mechanism:
	Bit Description
	5 Spec. Cycle–Config Mechanism #2 support
	4 Spec. Cycle–Config Mechanism #1 support
	1 Config Mechanism #2 support
	0 Config Mechanism #1 support
BH	Interface level major version
BL	Interface level minor version
CL	Number of last PCI bus
AL	**02h = Find PCI Device**
Entry:	
CX	Device ID (0-65535)
DX	Vendor ID (0-65534)
SI	Index (0-n)
Exit:	
BH	Bus number (0-255)
BL	Bits 7-3 Device number
	Bits 2-0 Function number
AL	**03h = Find PCI class code**
Entry:	
ECX	Class code in lower three bytes
SI	Index (0-n)
Exit:	
BH	Bus number (0-255)
BL	Bits 7-3 Device number
	Bits 2-0 Function number
AL	**06h = Generate special cycle**
Entry:	
BH	Bus number (0-255)
EDX	Special cycle data
AL	**08h = Read configuration byte**
Entry:	
BH	Bus number (0-255)
BL	Bits 7-3 Device number
	Bits 2-0 Function number
DI	Register number (0-255)
Exit:	
CL	Byte read

Continued

AL 09h = Read configuration word
Entry:
BH Bus number (0-255)
BL Bits 7-3 Device number
Bits 2-0 Function number
DI Register number (0, 2, 4,...254)
Exit:
CX Word read

AL 0Ah = Read configuration dword
Entry:
BH Bus number (0-255)
BL Bits 7-3 Device number
Bits 2-0 Function number
DI Register number (0, 4, 8,...252)
Exit:
ECX Dword read

AL 0Bh = Write configuration byte
Entry:
BH Bus number (0-255)
BL Bits 7-3 Device number
Bits 2-0 Function number
DI Register number (0-255)
CL Byte value to write

AL 0Ch = Write configuration word
Entry:
BH Bus number (0-255)
BL Bits 7-3 Device number
Bits 2-0 Function number
DI Register number (0, 2, 4,...254))
CX Word value to write

AL 0Dh = Write configuration dword
Entry:
BH Bus number (0-255)
BL Bits 7-3 Device number
Bits 2-0 Function number
DI Register number (0, 4, 8,...252)
ECX Dword value to write

AL 0Eh = Get PCI IRQ routing options
Entry:
DS Segment or Selector for BIOS data
ES Segment or Selector for Route Buffer parameter
DI 16-bit offset for Route Buffer parameter
EDI 32-bit offset for Route Buffer parameter
Exit:
BX Exclusive-PCI IRQ data map:
Bit 01 = IRQ0 PCI only
Bit 11 = IRQ1 PCI only
...
Bit 15 1 = IRQ15 PCI only

AL 0Fh = Set PCI hardware interrupt
Entry:
BH Bus number (0-255)
BL Bits 7-3 Device number
Bits 2-0 Function number
CL PCI interrupt pin (0Ah...0Dh)
CH IRQ number (0-15)
DS Segment or Selector for BIOS data

16.6.20 PnP Run-Time Services

Plug and Play automatically configures PC hardware and attached devices without requiring you to manually configure the device with jumpers or in Setup. You can install a new device such as sound or fax card ("plug it in") and start working ("begin playing").

To work properly, however, Plug-and-Play must be supported in the hardware and software, including the BIOS, the operating system (such as Microsoft Windows 95), and the hardware drivers.

Each Plug and Play device must have all of the following capabilities:

1. It must be uniquely identified
2. It must state the services it provides and the resources it requires
3. It must allow software to configure it.

Note: To register a new unique vendor ID or manufacturer ID for Plug and Play hardware, please send e-mail to pnpid@microsoft.com.

NOTE: There are a variety of Plug and Play technologies, including BIOS, ISA, SCSI, IDE, CD-ROM, LPT, COM, PCMCIA, and drivers. For complete instructions on using the PnP BIOS Services, consult the *Plug and Play BIOS Specification V. 1.0a.* You can download this specification and other PnP specifications from this Microsoft Web site:
http://www.microsoft.com/hwdev/specs/pnpspecs.htm

PhoenixBIOS 4.0 optionally supports PnP (Plug and Play) Runtime Services in Real and Protected Mode in with the following routines:

PnP Run-Time Services	
00h	Get Number of Device Nodes
01h	Get Device Node
02h	Set Device Node
03h	Get Event
04h	Send Message
05h	Get Docking Station Information
09h	Set Statically Allocated Resources
0Ah	Get Statically Allocated Resources
0Bh	Get APM 1.1 ID Table
40h	Get ISA Configuration Structure
41h	Get ESCD Information
42h	Read ESCD Data Image
43h	Write ESCD Data Image

The following are the exit status codes for the PnP Runtime Services

PnP Runtime Service Exit Status Codes	
AH	00h = No error
	If Carry = 1:
AH	7Fh = Device not set statically
	81h = Unknown or invalid function
	82h = Function not supported
	83h = Handle for Device Node invalid or out of range
	84h = Bad resource descriptors
	85h = Set Device Node function failed
	86h = No events pending
	87h = System currently not docked
	88h = No ISA PnP cards installed
	89h = Cannot determine docking station capabilities
	8Ah = Undocking failed: no battery
	8Bh = Docking failed: conflict with primary boot device
	8Ch = Caller's memory buffer too small
	8Dh = Use ESCD support function instead
	8Eh = Send Message 04h function not supported
	8Fh = Hardware error

To find the PnP entry points, search for the **PnP BIOS Support Installation Check** structure by searching for the "$PnP" signature in system memory staring from F0000h to FFFFFh at every 16-byte boundary. Check the validity of the structure by adding the values of *Length* bytes, including the *Checksum* field, into a 8-bit value. Zero indicates a valid checksum.
The following describes the support structure:

PnP Support Installation Check

Offset	Size	Description
00h	4	ASCII "$PnP" signature
04h	1	Version (10h)
05h	1	Length (21h)
06h	2	Control field
08h	1	Checksum
09h	4	Event-notification flag address
0Dh	2	Real Mode 16-bit offset to entry point
0Fh	2	Real Mode 16-bit code segment address
11h	2	16-bit Protected Mode offset to entry point
13h	4	16-bit Protected Mode code segment base address
17h	4	OEM Device Identifier
1Bh	2	Real Mode 16-bit data segment address
1Dh	4	16-bit Protected Mode data segment base address

Call each service by loading the function parameters on the stack and FAR calling the appropriate entry point. The following are the Runtime Services of *Phoenix*BIOS 4.0, in 'C' syntax.

PnP Runtime-Service Function Parameters

00h Get Number of Device Nodes

```
Entry:
int FAR (*entryPoint)(Function, NumNodes, NodeSize,
BiosSelector);
int  Function;
unsigned char  FAR  *NumNodes;
unsigned int  FAR  *NodeSize;
unsigned int    BiosSelector;
```

01h Get System Device Node

```
Entry:
int FAR (*entryPoint)(Function, Node, devNodeBuffer,
Control, BiosSelector);
int  Function;
unsigned char        FAR  *Node;
struc DEV_NODE   FAR                    *devNodeBuffer;
unsigned int  Control;
unsigned int      BiosSelector;
```

02h Set System Device Node

```
Entry:
int FAR (*entryPoint)(Function, Node, devNodeBuffer,
Control, BiosSelector);
int  Function;
unsigned char  Node;
struc DEV_NODE FAR                      *devNodeBuffer;
unsigned int      Control;
unsigned int      BiosSelector;
```

03h Get Event

```
Entry:
int FAR (*entryPoint)(Function, Message, BiosSelector);
int  Function;
unsigned int  FAR  *Message;
unsigned int      BiosSelector;
```

04h Send Message

```
Entry:
int FAR (*entryPoint)(Function, Message, BiosSelector);
int  Function;
unsigned int  Message:
unsigned int  BiosSelector;
```

Continued

Continued

05h Get Docking Station Information

Entry:
int FAR (*entryPoint)(Function, DockingStationInfo,
BiosSelector);
int Function;
unsigned char FAR *DockingStationInfo;
unsigned int BiosSelector;
Exit:
Docking station info buffer:

Offset	Description
00h	Docking station location identifier
04h	Serial Number
08h	Docking Capabilities: Bits 2-1: 00 = Cold Docking 01 = Warm Docking 10 = Hot Docking Bit 0: 0 = Surprise-style docking 1 = VCR-style docking

09h Set Statically Allocated Resources

Entry:
int FAR (*entryPoint)(Function, Resource Block,
BiosSelector);
int Function;
unsigned char FAR *ResourceBlock;
unsigned int BiosSelector;

0Ah Get Statically Allocated Resources

Entry:
int FAR (*entryPoint)(Function, Resource Block,
BiosSelector);
int Function;
unsigned char FAR *ResourceBlock;
unsigned int BiosSelector;

0Bh Get APM ID Table (For APM 1.1 only)

Entry:
int FAR (*entryPoint)(Function, BufSize, APMIdTAble
BiosSelector);
int Function;
unsigned int FAR *BufSize;
unsigned char FAR *APMIdTable;
unsigned int BiosSelector;
Exit:
APM ID table:

Length	Description
Dword	Device identifier
Word	APM 1.1 identifier

40h Get PnP ISA Configuration Structure

Entry:
int FAR (*entryPoint)(Function, Configuration, BiosSelector);
int Function;
unsigned char FAR *Configuration;
unsigned int BIOS Selector;
Exit:
PnP ISA Configuration structure:

Offset	Description
00h	Structure revision
01h	Number of Card Select Numbers assigned
02h	ISA Read Data port
04h	Reserved

41h Get Extended System Configuration Data (ESCD)

Entry:
int FAR (*entryPoint)(Function, MinESCDWriteSize,
ESCDSize, NVStorageBase, BiosSelector);
int Function;
unsigned int FAR *MinESCDWriteSize;
unsigned int FAR *ESCDSize;
unsigned long FAR *NVStorageBase;
unsigned int BiosSelector;

Continued

42h Read Extended System Configuration Data

```
Entry:
int FAR (*entryPoint)(Function, ESCDBuffer, ESCDSelector,
BiosSelector);
int  Function;
char  FAR     *ESCDBuffer;
unsigned int  ESCDSelector;
unsigned int  BiosSelector;
```

43h Write Extended System Configuration Data (ESCD)

```
Entry:
int FAR (*entryPoint)(Function, ESCDBuffer, ESCDSelector,
BiosSelector);
int  Function;
char  FAR     *ESCDBuffer;
unsigned int  ESCDSelector;
unsigned int  BiosSelector;
```

16.6.21 SMBIOS Services

The **System Management BIOS (SMBIOS),** one of the components of the Desktop Management Interface (DMI), is a method for managing PCs in an enterprise. Using SMBIOS, a Manager of Information Systems can access up-to-date information about the hardware and software installed on every computer on a network.

NOTE: For complete instructions on using these services, see the **System Management BIOS Reference Specification** available at the Phoenix Web site: http://www.phoenix.com/products/specs-smbios.pdf

For descriptions of the DMI architecture, see the Web site of the **Desktop Management Task Force** at: http://www.dmtf.org

The SMBIOS Services are functions 50h through 5Fh of the PnP Run Time Services. See "PnP Run-Time Services" above for a description of how to find the PnP entry points to these SMBIOS Services. The following are the SMBIOS services supported in PhoenixBIOS 4.0:

SMBIOS Services

50h	Get SMBIOS Information
51h	Get SMBIOS Structure
52h	Set SMBIOS Structure
55h	Get GPNV Information
56h	Read GPNV Information
57h	Write GPNV Data

The following are the exit status codes for the SMBIOS Services:

SMBIOS Services Exit Status Codes

```
AX      00h = Function Completed Successfully
AX      81h = Unknown, or invalid, function number passed
        82h = The function is not supported on this system
        83h = SMBIOS Structure number/handle passed is
        invalid or out of range.
        84h = The function detected invalid parameter or, in he case of a “Set SMBIOS Structure”
              request, detected an invalid value for a to-be-changed structure field
        85h = The SubFunction parameter supplied on a SMBIOS control function is not sup-
ported
              by the system BIOS.
        86h = There are no changed SMBIOS structures pending notification.
```

Continued

Continued

40h Get PnP ISA Configuration Structure

Entry:
int FAR (*entryPoint)(Function, Configuration, BiosSelector);
int Function;
unsigned char FAR *Configuration;
unsigned int BIOS Selector;
Exit:
PnP ISA Configuration structure:

Offset	Description
00h	Structure revision
01h	Number of Card Select Numbers assigned
02h	ISA Read Data port
04h	Reserved

41h Get Extended System Configuration Data (ESCD)

Entry:
int FAR (*entryPoint)(Function, MinESCDWriteSize, ESCDSize, NVStorageBase, BiosSelector);
int Function;
unsigned int FAR *MinESCDWriteSize;
unsigned int FAR *ESCDSize;
unsigned long FAR *NVStorageBase;
unsigned int BiosSelector;

42h Read Extended System Configuration Data

Entry:
int FAR (*entryPoint)(Function, ESCDBuffer, ESCDSelector, BiosSelector);
int Function;
char FAR *ESCDBuffer;
unsigned int ESCDSelector;
unsigned int BiosSelector;

43h Write Extended System Configuration Data (ESCD)

Entry:
int FAR (*entryPoint)(Function, ESCDBuffer, ESCDSelector, BiosSelector);
int Function;
char FAR *ESCDBuffer;
unsigned int ESCDSelector;
unsigned int BiosSelector;

16.6.22 SMBIOS Services

The **System Management BIOS (SMBIOS),** one of the components of the Desktop Management Interface (DMI), is a method for managing PCs in an enterprise. Using SMBIOS, a Manager of Information Systems can access up-to-date information about the hardware and software installed on every computer on a network.

NOTE: For complete instructions on using these services, see the **System Management BIOS Reference Specification** available at the Phoenix Web site:
http://www.phoenix.com/products/specs-smbios.pdf

For descriptions of the DMI architecture, see the Web site of the **Desktop Management Task Force** at: http://www.dmtf.org

The SMBIOS Services are functions 50h through 5Fh of the PnP Run Time Services. See "PnP Run-Time Services" above for a description of how to find the PnP entry points to these SMBIOS Services. The following are the SMBIOS services supported in PhoenixBIOS 4.0:

SMBIOS Services	
50h	Get SMBIOS Information
51h	Get SMBIOS Structure
52h	Set SMBIOS Structure
55h	Get GPNV Information
56h	Read GPNV Information
57h	Write GPNV Data

The following are the exit status codes for the SMBIOS Services:

SMBIOS Services Exit Status Codes

AX 00h = Function Completed Successfully
AX 81h = Unknown, or invalid, function number passed
82h = The function is not supported on this system
83h = SMBIOS Structure number/handle passed is invalid or out of range.
84h = The function detected invalid parameter or, in the case of a "Set SMBIOS Structure"
request, detected an invalid value for a to-be-changed structure field
85h = The SubFunction parameter supplied on a SMBIOS control function is not supported
by the system BIOS.
86h = There are no changed SMBIOS structures pending notification.

55h Get General-Purpose NonVolatile Information

Entry:

```
short FAR (*entryPoint) (
        short Function;
        unsigned short FAR *Handle,
        unsigned short FAR *MinGPNVRWSize,
        unsigned short FAR *GPNVSize,
        unsigned long FAR *NVStorageBase,
        unsigned short BiosSelector);
```

56h Read General-Purpose NonVolatile Data

Entry:

```
short FAR (*entryPoint) (
        short Function;
        unsigned short Handle,
        unsigned char FAR *GPNVBuffer,
        short FAR *GPNVLock,
        unsigned short GPNVSelector,
        unsigned short BiosSelector);
```

57h Write General-Purpose NonVolatile Data

Entry:

```
short FAR (*entryPoint)(
        short Function,
        unsigned short Handle,
        unsigned char FAR *GPNVBuffer,
        short GPNVLock,
        unsigned short GPNVSelector,
        unsigned short BiosSelector );
```

16.6.23 MultiBoot II Run-Time Services

An OS or application program can access the features of PhoenixBIOS MultiBoot II during run-time by using the following MultiBoot II Run-Time Services. You can use these services to query the number and type of Initial Program Load (IPL) devices in the system or display an IPL device menu for specifying the boot priority on the next system restart.
MultiBoot II Run-Time Services are extensions to the Plug and Play run-time functions that implement the *BIOS Boot Specification Ver. 1.01.* You can access this specification in Acrobat format from the Phoenix Web site at:
http://www.phoenix.com/desktop/bbs101.pdf

PnP functions 60h through 6Fh are reserved for the BIOS Boot Specification. See Appendix C of the *Plug and Play BIOS Specification* mentioned above for the details of the calling conventions. These functions are available in Real Mode and 16-bit Protected Mode.

MultiBoot II Run-Time Services

60h Get Version and Installation Check

```
Entry:
short FAR (* entryPoint) (Function, Version, BiosSelector);
short Function;
unsigned short FAR *Version;
unsigned short BiosSelector;
```

Continued

61h Get Device Count

```
Entry:
short FAR (* entryPoint) (Function, Switch, Count,
MaxCount, StructSize, BiosSelector);
short Function;
short Switch;
unsigned short FAR *Count;
unsigned short FAR *MaxCount;
unsigned short FAR *StructSize;
unsigned short BiosSelector;
```

62h Get Priority and Table

```
Entry:
short FAR (* entryPoint) (Function, Switch, Priority, Table,
 BiosSelector);
short Function;
short Switch;
unsigned char FAR *Priority;
unsigned char FAR *Table;
unsigned short BiosSelector;
```

63h Set Priority

```
Entry:
short FAR (* entryPoint) (Function, Switch, Priority,
BiosSelector);
short Function;
short Switch;
unsigned byte FAR *Priority;
unsigned short BiosSelector;
```

64h Get IPL Device from Last Boot

```
Entry:
short FAR (* entryPoint) (Function, IPLEntry, BiosSelector);
short Function;
unsigned short FAR *IPLEntry;
unsigned short BiosSelector;
```

16.7 BIOS Data Area

The BIOS keeps information about the current operating environment of the AT system in the BIOS Data Area. The normal way to access this information is by means of the BIOS Services, described above. The BIOS Data Area is located from physical address 400h to 501h.

BIOS Data Area Description

Offset	Size	Description
00	2	Com1 address
02	2	Com2 address
04	2	Com3 address
06	2	Com4 address
08	2	Lpt1 address
0A	2	Lpt2 address
0C	2	Lpt3 address
0E	2	LPT4/EBDA address*
10	2	Equipment installed:
	Bit	**Definition**
	0	Not used
	1	Math coprocessor installed
	2	PS/2 mouse installed
	3	Not used
	4,5	Initial video mode: 00 = EGA/VGA 01 = 40x25 CGA 10 = 80x25 CGA 11 = Monochrome
	6,7	Diskette drives: 00 = 1 drive 01 = 2 drives 10 = 3 drives 11 = 4 drives

Continued

BIOS Data Area, Continued

8	Not used
9-11	Number of serial adapters
12	Game Adapter installed
13	Not used
14,15	Number of parallel adapters

Offset	Size	Description
12	1	Interrupt flag (POST)
13	2	Memory size (K bytes)
15	1	Reserved
16	1	Control flag

Keyboard Data Area

Offset	Size	Description
17	1	Keyboard flag 0:

Bit	Definition
0	Right shift key pressed
1	Left shift key pressed
2	Control key pressed
3	Alt key pressed
4	Scroll lock on
5	Num lock on
6	Caps lock on
7	Insert mode on

18	1	Keyboard flag 1:

Bit	Definition
3	Freeze state
4	Scroll lock pressed
5	Num lock pressed
6	Caps lock pressed
7	Insert mode pressed

19	1	Keypad input byte
1A	2	Key buffer head
1C	2	Key buffer tail
1E	20	Key buffer

Diskette Data Area

3E	1	Seek/recalibrate status
3F	1	Drive motor status
40	1	Motor on time
41	1	Diskette status:

Bit	Definition
7	1 = Drive not ready
6	1 = Seek error occurred
5	1 = Diskette controller failed
4-0	Error codes:
...	01h = Illegal function request
...	02h = Address mark not found
...	03h = Write protected error
...	04h = Sector not found
...	06h = Diskette change line active
...	08h = DMA overrun on operation
...	09h = Data-boundary error (64k)
...	0Ch = Media type not found
...	10h = Uncorrectable ECC or CRC error
...	20h = General controller failure
...	40h = Seek operation failed
...	80h = Device did not respond

42	7	Controller status

Video Data Area

Offset	Size	Description
49	1	Video mode
4A	2	Video columns
4C	2	Video length
4E	2	Video start
50	10	Cursor locations
60	2	Cursor size
62	1	Active page
63	2	6845 address
65	1	Mode register value
66	1	Video palette

Continued

Extended Work Area

67	4	ROM check address
6B	1	CPU rate control

Timer Data Area

6C	2	Timer count low word
6E	2	Timer count high word
70	1	Timer overflow byte

System Data Area

71	1	Break pressed flag
72	2	Soft reset flag

Fixed Disk Data Area

74	1	Fdisk status
75	1	Number of fixed disks
76	1	Fixed disk control
77	1	Reserved

Serial and Parallel Timeout Counters

78	4	Lpt1-4 time-out values
7C	4	Com1-4 time-out values

Extended Keyboard Data Area

80	2	Key buffer start
82	2	Key buffer end

EGA/VGA Data Area

84	1	Number of video rows
85	2	Bytes per character
87	1	EGA Status A
88	1	EGA Status B
89	1	VGA Status A
8A	1	Display Combination Code index

Extended Diskette Area

8B	1	Last diskette data rate

Extended Fixed Disk Area

8C	1	FDisk status
8D	1	FDisk error value
8E	1	FDisk interrupt flag

Additional Extended Diskette Area

Offset	Size	Description
8F	1	Floppy info nibbles
90	4	Floppy state information
94	2	Floppy cylinder number

Additional Extended Keyboard Data Area

96	1	Keyboard control
97	1	Keyboard flag 2:

Bit .... Definition
0..... Scroll LED on
1..... Num lock LED on
2..... Caps lock LED on
4..... Ack code received
5..... Resend received
6..... LED being updated
7..... Keyboard error

Real Time Clock Area

Offset	Size	Description
98	4	RTC user flag
9C	2	RTC time low word
9E	2	RTC time high word
A0	1	RTC wait flag

Network Data Area

A1	7	Network work area

Extended EGA/VGA Data Area

A8	4	EGA/VGA environment pointer

Miscellaneous

AC-FF		Reserved
100	1	Print screen flag

* If the BIOS supports the Extended BIOS Data Area, it uses the LPT4 address in the BIOS data area (Offset 0E) for the Extended BIOS Data Area segment.

16.7.1 Extended BIOS Data Area

The Extended BIOS Data Area (EBDA), located in the top 1k of system RAM, contains information about the pointing device (PS/2 mouse).
INT 15h AH = C1h returns the segment starting address of this table.

Extended BIOS Data Area

Offset	Size	Description
00h	1	Size of EBDA in kbytes
01h	33	Reserved
21h	4	Pointer to device routine
25h	1	First byte of pointer information: Bit .... Definition 4..... Pointer error 5..... Pointer acknowledge 6..... Resend request 7..... Command in progress
26h	1	Second byte of pointer information Bit .... Definition 6..... Enable pointer device 7..... Pointer external device
27h	2	Pointer data package

16.8 Interrupt Vectors

The following table describes the AT system interrupt vectors. Status indicates whether the BIOS supports the interrupt.

INT	Description	Status
00	Divide by zero	Not Supported
01	Single step	Not Supported
02	Non-Maskable interrupt	Supported
03	Breakpoint	Not Supported
04	Overflow	Not Supported
05	Print Screen Interrupt	Supported
06	286 LoadAll Handler	Supported
07	Reserved	Not Supported
08	IRQ0 - System Timer Interrupt	Supported
09	IRQ1 - Keyboard Interrupt	Supported
0A	IRQ2 - Reserved	Not Supported
0B	IRQ3 - COM2: Interrupt	Supported
0C	IRQ4 - COM1: Interrupt	Supported
0D	IRQ5 - LPT2: Interrupt	Supported
0E	IRQ6 - Floppy Disk Interrupt	Supported
0F	IRQ7 - LPT1: Interrupt	Supported
10	BIOS Video Interface	Supported
11	BIOS Equipment Check	Supported
12	BIOS Memory Request	Supported
13	BIOS Fixed Disk/Diskette Interface	Supported
14	BIOS Serial Interface	Supported
15	BIOS System Functions Interface	Supported
16	BIOS Keyboard Interface	Supported
17	BIOS Parallel Printer Interface	Supported
18	BIOS Secondary Boot Request	Supported
19	BIOS Primary Boot Request	Supported
1A	BIOS System Timer Interface	Supported
1B	BIOS Control Break Interrupt	Supported
1C	BIOS User System Timer Interrupt	Supported
1D	BIOS Video Init Parameters	Supported
1E	BIOS Diskette Parameters	Supported
1F	BIOS Video Graphic Characters	Supported
40	BIOS Diskette (when fixed disk present)	Supported
41	BIOS Fixed disk 0 parameters	Supported
46	BIOS Fixed disk 1 parameters	Supported
70	IRQ8 - Real time clock interrupt	Supported
71	IRQ9 - IRQ2 redirection	Supported
72	IRQ10 - Reserved	Not Supported
73	IRQ11 - Reserved	Not Supported
74	IRQ12 - Available/PS/2 Mouse	Supported
75	IRQ13 - Math coprocessor	Supported
76	IRQ14 - Primary IDE HDD	Supported
77	IRQ15 - Available/Secondary IDE HDD	Supported

17 INDEX

1

16C550 UARTS....37

2

27C010 PLCC32....45
27C040 PLCC32....45

8

8253....43
8259A....43

A

active page....166
adapter
  disk....131
adapter ROM....158, 159
Advanced Power Management....See APM
alarm....194
APM....179
  BIOS services....181, 182, 184
  CPU busy....182
  CPU Idle....182
  driver version....184
  enable/disable power management....184
  Enable/disable power management....183
  engage/disengage power management....184
  Get PM Event....184
  Get Power State....184
  Get Power Status....183
  Installation Check....182
  Interface Connect....182
  Protected-mode 16-bit interface connect....182
  Protected-mode 32-bit interface connect....182
  Restore Power_On Defaults....183
  Set Power State....183
Autotype....131

B

Basic Input and Output System....157
Battery backed clock (RTC)....44
battery current....44
BDA....204
beep code....160
Big Memory....179, 185
BIOS....157
  data area....204
  service....158
  services....164
  test points....161
BIOS ROM....45
BIOS.ROM....154
BIOS32 Service Directory....164
Block Diagram....10
Boot First Menu....151, 153
boot options....135
bootable CD ROM....175
build a system....127
Burst....14
BUS....13
Bus currents....18

C

cache....132, 133
Cache....150
cassette....179
CD ROM....175
  Command Packet....176
  Specification Packet....175
centronics....104
check points, POST....161
chipset....141
clock....194
CMOS....147
  error....150
  save Setup values....147
Code Read Page Mode....139
color....167
  palette....167
COM 1/2 cable....118
COM port....142
communications services....177
Compatibility....14
connectors....100
CPU speed keys....160
CRISDISK....154
CRISDISK.BAT....154, 155
CRISDISK.ZIP....154
Crisis disk....154
Crisis Recovery Diskette....154
CrisisRecovery disk....156
CRT Displays....81
current HD....35
cursor
  position....166
cylinder, fixed disk....159

D

date....131
Desktop Management Interface....200, 201
device busy....180
device configuration....141
Device Node....198
Direct Memory Access....149
disk
  adapter....131
  cylinder....172
  sector....172
  status....171
diskette....131
  controller....142

sectors 169
services 168
status 169
type 169, 170
diskette and fixed-disk systems 171
DMA 149
DMI 200, 201
Docking Station 199
DRAM Memory 57
drive
parameter 170, 172
type 170

E

EBDA 207
EDD services 174
EEPROM Memory for Setup 46
EEPROM saved CMOS setup 55
EFT Electric Fast Transient 14
EISA
services 187
EMI 14
EN55022 14
Enhanced Disk Drive services 174
Enhanced Parallel Printer 191
EPP 191
equipment information 168
error 160
address conflict 149
diskette 168
fixed disk 171
keyboard 189
port 80h codes 161
serial service 177
Error on boot time 127
ESCD 199, 201
ESD Electro Static Discharge 14
exit menu 147
exit status codes 164
Int 13 Diskette 168
Int 14h general PCI 195
Int 15 EISA 187
Int17h EPP 191
PnP Runtime Services 197
SMBIOS 2.2 Services 200, 202
Extended BIOS Data Area 181, 207
extended memory 131, 185
move block 180
size 180
Extended System Configuration Data 199, 201

F

fixed disk
diagnostic 173
drive type 159
error codes 171
extended services 174
recalibrate 173
services 171, 173
tables 159
Flash ROM 154
floppy disk cable 115
Floppy disk interface 41
Floppy disk interface connector for 5,25" and 3,5" FD (34 pins header) 103
Floppy disk interface connector for micro floppy 3,5" from TEAC (FD05) (26 pins FCC-header) 103
floppy drive ..... See diskette
floppy seek 135
format diskette track 170
Full On 145
function keys 160

G

Get Drive Parameters 174
graphics 167

H

hard disk ..... See fixed disk
Harddisk cable 40 pins 116
Harddisk cable 44 pins 117
harddisk connected 35
Harddisk list 54
hardware
requirements 158
head, fixed disk 159
High-frequency radiation 14

I

I/O
device error 150
I/O chip 141
I/O map 58
IDE disk adapters 131
IDE interface connector - AT- IDE 2,5" 44 pins RM2.00mm 102
IEC 14
IEEE587 14
initialize
PS/2 mouse 186
Initialize Serial Adapter 177
Int 10h video services 166
Int 11h return system info 168
Int 12h return memory size 168
Int 13h
bootable CD ROM 175
diskette services 168
Extended Fixed Disk Services 174
fixed disk services 171
Int 14h serial services 177
Int 14h services 177
Int 15h
APM services 181
Big Memory services 185
EISA services 187
PS/2 mouse services 186
system services 179
Int 15h services 207
Int 16h keyboard services 189
Int 17h
EPP services 191
Int 17h parallel printer services 190
Int 1Ah
PCI services 195
time of day services 194

interfaces 12
interrupt
non-maskable 207
table 164
vector 207
Interrupt Controllers 43
IRQ 43

J

joystick 179
support 179
Jumper Locations 112

K

key click 136
key repeat 136
keyboard
error 189
servicesInt16H 189
keyboard current 35

L

landing zone 132, 159
Large Disk Mode 138
LCD Controller 80
LPT port 142

M

MAKEBOOT.EXE 154
master drive 131
math coprocessor 204
media
change 170
type for format 170
memory 131
extended 179
refresh 139
system 168
memory address MAP 57
MINIDOS.SYS 154, 155
MultiBoot 150, 151
MultiBoot II
Run-Time Services 202
MultiBoot II Run-Time Services 203
multi-tasking services 179

N

NMI 149
Non-Maskable Interrupt 149
numlock 136
NVRAM
error 150

O

OEM
screen 150
Operating System 158
option ROM 158
QuietBoot 151
Ordering Codes 16
OS/2 180

P

palette 167
parallel printer services 190
Parity Check 150
password 143
PCI 139, 140
devices menu 140
PCI services 195
Peripheral Component Interconnect 140
Phlash 154
PHLASH.EXE 155
PHLASH.EXE 154
PLATFORM.BIN 154
PnP
BIOS support installation check 198
Runtime Services 197, 198, 200, 201
pointer device services 186
port 80h codes 161
POST 158
<ESC> 151
<F2> 151
error 160
option ROM 151
terminal error 160
test points 160
Power Management 145
Power On Self Test 158
Power requirement 35
Power Supply connector 100, 101
Printer 36, 190
Printerport connector (LPT1) 104
Program termination 179
protected mode 180, 182
PS/2 - keyboard connector 104
PS/2 Mouse 150
PS/2 mouse support 186

Q

QuietBoot 150
QuietBoot 150

R

RAM 158
extended 150
read
character 166
device type 186
drive parameters 170
drive type 173
ESCD 200, 201
graphics pixel 167
modem control register 178
real time clock 194
Receive character 178
REF Radiated Electromagnetic Field 14
requirements

option ROM 158
system board 158
Reset diskette system 169
ROM
BIOS 157
default values 147
ROM-BIOS 45
RTC 206
RTC (Real time clock) 44
RTC-Reset 47
Runtime Services 197, 198

S

scroll page 166
SCSI connector 108, 109, 110
SCSI Interface (Option) 122
sector 172
sectors, fixed disk 159
security 143
Send character 178
serial interface cable 118
Serial Port connector 100, 101
Serial Port Signal 37
serial port status 178
Serial Ports 37, 38, 39, 40
serial services 177
service entry point 164
set
cursor 166
video mode 166
Setup 47
get CMOS values 148
get ROM defaults 148
MultiBoot 151
QuietBoot 151
save values to CMOS 147
Setup Boot Menu 151
shadow 134
Shadow 150
signals 22
SIR Surge Immunity Requirements 14
slave drive 131
Snoop Ahead 139
software interrupts 164
Speaker interface 42
Specifications 11
Standby 145
Starting up the System 127
summary screen 135
Support 8
Suspend 145
system
information 168
memory map 185
memory size 168
parameters 181
services 179
System memory map 57
system resources 141

T

Teletype 167
terminal error 160
test points 161
time-of-day 131
services 194
Timer Assignment 43
troubleshooting 161
typematic rate 189

U

Utility connector 109, 110, 111

V

VDE 14
verify
diskette sectors 169
VGA 80
VGA BIOS ROM 45
VGA error 160
VGABIOS.EXE 155
VGA-CRT connector (HiDens DSUB 15pin) 105
VGA-LCD connector 106, 107
video
parameter 167
services 166

W

wait 180
wait states 149
warranty 8
Watchdog 44
Windows NT 180
write 167
buffer 139
character 166
cycle 139
diskette sectors 169
ESCD 200, 201
graphics pixel 167
modem control register 178
page mode 139
pixel 167
precomp 132, 159
string 167
teletype 167