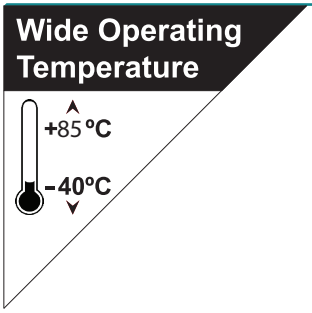


COM-842E

COM Express CPU Module

User's Manual

Version 1.1

2011.06

This page is intentionally left blank.

Table of Contents

Chapter 1

Introduction

1.1 Copyright Notice

The information in this document is subject to change without prior notice in order to improve the reliability, design and function. It does not represent a commitment on the part of the manufacturer.

Under no circumstances will the manufacturer be liable for any direct, indirect, special, incidental, or consequential damages arising from the use or inability to use the product or documentation, even if advised of the possibility of such damages.

1.2 Declaration of Conformity

CE

This product has passed the CE test for environmental specifications when shielded cables are used for external wiring. This kind of cable is available from ARBOR. Please contact your local supplier for ordering information. Test conditions for passing included the equipment being operated within an industrial enclosure. In order to protect the product from being damaged by ESD (Electrostatic Discharge) and EMI leakage, we strongly recommend the use of CE-compliant industrial enclosure products.

FCC Class A

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions : (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

RoHS

ARBOR Technology Corp. certifies that all components in its products are in compliance and conform to the European Union's Restriction of Use of Hazardous Substances in Electrical and Electronic Equipment (RoHS) Directive 2002/95/EC.

The above mentioned directive was published on 2/13/2003. The main purpose of the directive is to prohibit the use of lead, mercury, cadmium, hexavalent chromium, polybrominated biphenyls (PBB), and polybrominated diphenyl ethers (PBDE) in electrical and electronic products. Member states of the EU

are to enforce by 7/1/2006.

ARBOR Technology Corp. hereby states that the listed products do not contain unintentional additions of lead, mercury, hex chrome, PBB or PBDB that exceed a maximum concentration value of 0.1% by weight or for cadmium exceed 0.01% by weight, per homogenous material. Homogenous material is defined as a substance or mixture of substances with uniform composition (such as solders, resins, plating, etc.). Lead-free solder is used for all terminations (Sn(96-96.5%), Ag(3.0-3.5%) and Cu(0.5%)).

1.3 About This User's Manual

This user's manual provides general information and installation instructions about the product. This User's Manual is intended for experienced users and integrators with hardware knowledge of personal computers. If you are not sure about any description in this booklet. please consult your vendor before further handling.

1.4 Warning

Single Board Computers and their components contain very delicate Integrated Circuits (IC). To protect the Single Board Computer and its components against damage from static electricity, you should always follow the following precautions when handling it :

1. Disconnect your Single Board Computer from the power source when you want to work on the inside.

2. Hold the board by the edges and try not to touch the IC chips, leads or circuitry.

3. Use a grounded wrist strap when handling computer components.

4. Place components on a grounded antistatic pad or on the bag that comes with the Single Board Computer, whenever components are separated from the system.

1.5 Replacing the Lithium Battery

Incorrect replacement of the lithium battery may lead to a risk of explosion.

The lithium battery must be replaced with an identical battery or a battery type recommended by the manufacturer.

Do not throw lithium batteries into the trash-can. It must be disposed of in accordance with local regulations concerning special waste.

1.6 Technical Support

If you have any technical difficulties, please do not hesitate to call or e-mail our customer service.

http://www.arbor.com.tw

E-mail:info@arbor.com.tw

1.7 Warranty

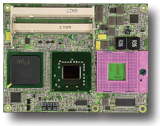
This product is warranted to be in good working order for a period of two years from the date of purchase. Should this product fail to be in good working order at any time during this period, we will, at our option, replace or repair it at no additional charge except as set forth in the following terms. This warranty does not apply to products damaged by misuse, modifications, accident or disaster.

Vendor assumes no liability for any damages, lost profits, lost savings or any other incidental or consequential damage resulting from the use, misuse of, or inability to use this product. Vendor will not be liable for any claim made by any other related party.

Vendors disclaim all other warranties, either expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the hardware, the accompanying product's manual(s) and written materials, and any accompanying hardware. This limited warranty gives you specific legal rights.

Return authorization must be obtained from the vendor before returned merchandise will be accepted. Authorization can be obtained by calling or faxing the vendor and requesting a Return Merchandise Authorization (RMA) number. Returned goods should always be accompanied by a clear problem description.

1.8 Packing List



1 x COM-842E COM Express CPU Module

1 x Driver CD

1 x Quick Installation Guide

If any of the above items is damaged or missing, contact your vendor immediately.

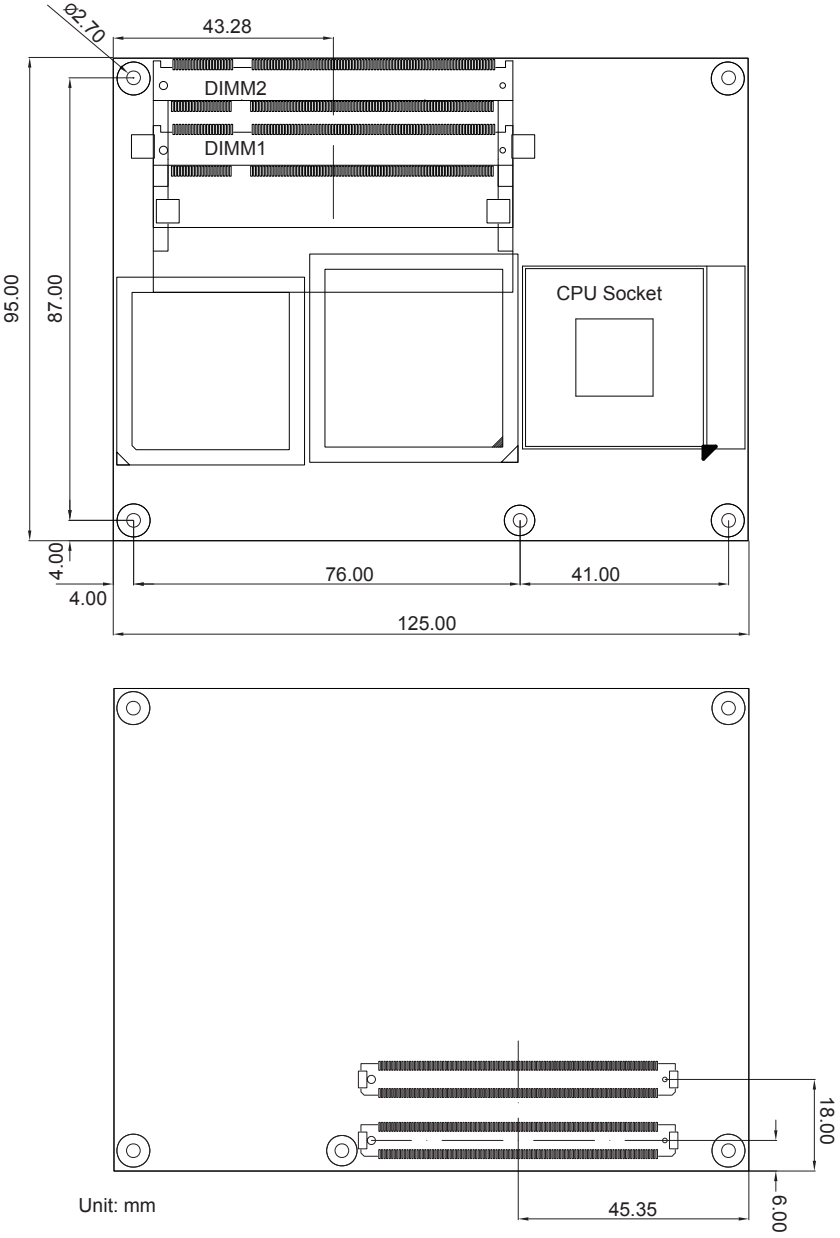
1.9 Ordering Information

COM-842E/L7500	Intel® Core™ 2 Duo L7500 1.6GHz COM Express CPU Module
PBE-1700	COM Express evaluation board in ATX form factor
HS-0842-F1	Heat spreader (114 x 95 x 20.8mm)
CBK-04-1700-00	Cable kit

1.10 Specifications

Form Factor	COM Express Type 2 CPU Module
CPU	Intel® Core™ 2 Duo processor, up to 800MHz FSB Intel® Celeron® M processor with 533/667MHz FSB
Chipset	Intel® GME965 + Intel® ICH8M
System Memory	2 x 200-pin DDR2 SO-DIMM sockets supporting 533/667MHz SDRAM up to 4GB
VGA/LCD Controller	Integrated Intel® Graphics Media Accelerator X3100
Ethernet	1 x RTL8111 PCIe Gigabit Ethernet Controller
BIOS	AMI PnP Flash BIOS
Serial ATA	3 x Serial ATA with 300MB/s HDD transfer rate
IDE Interface	1 x Ultra ATA, supports 2 IDE devices
Universal Serial Bus	8 x USB 2.0
Digital Input/Output	8-bit programmable Digital Input/Output
Expansion Interface	1 x PCI Express x16 5 x PCI Express x1 4 x PCI
Operation Temp.	-40ºC ~ 85ºC (-40ºF ~ 185ºF)
Watchdog Timer	1 ~ 255 levels Reset
Dimension (L x W)	125 x 95 mm (4.9” x 3.7”)

1.11 Board Dimensions



This page is intentionally left blank.

Chapter 2

Installation

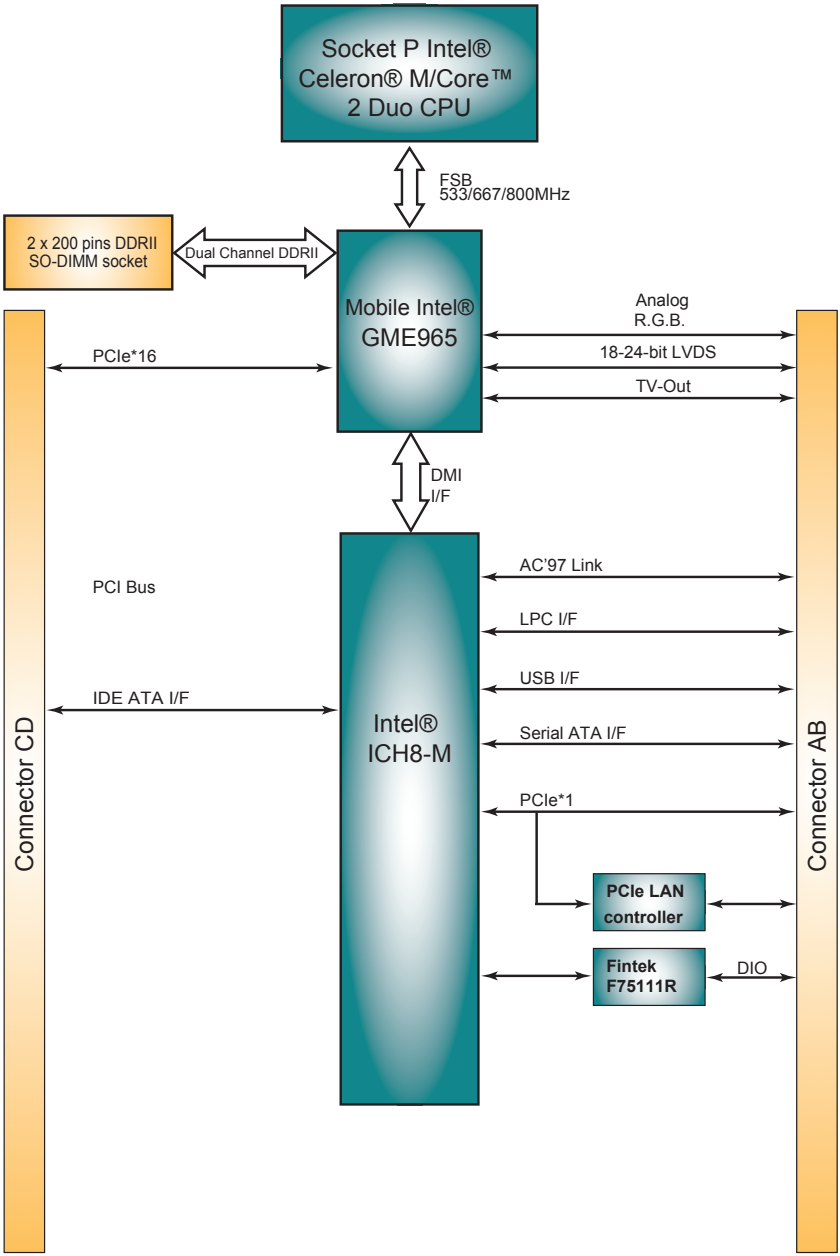
2.1 What is “ COM Express ”?

With more and more demands on small and embedded industrial boards, a multi-functioned COM (Computer-on-Module) is the great one of the solutions.
COM Express, board-to-board connectors consist of two rows of 220 pins each.
Row AB, which is required, provides pins for PCI Express, SATA, LVDS, LCD channel, LPC bus, system and power management, VGA, LAN, and power and ground interfaces.
Row CD, which is optional, provides SDVO and legacy PCI and IDE signals next to additional PCI Express, LAN and power and ground signals.

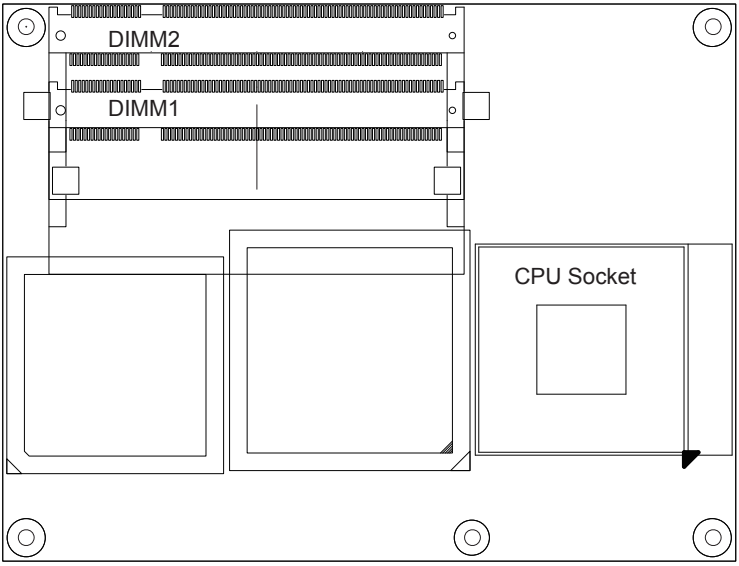
By the way, the target markets of COM will be focused on:

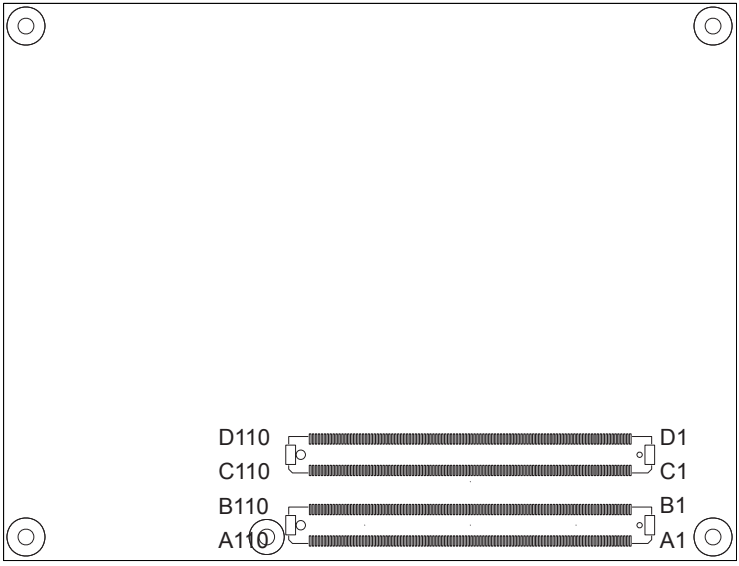
- Retail & Advertising
- Medical
- Test & Measurement
- Gaming & Entertainment
- Industrial & Automation
- Military & Government
- Security

2.2 Block Diagram



2.3 Jumpers and Connectors





2.4 COM Express AB Connector

B1	GND	GND	A1
B2	GBE0_ACT#	GBE0_MDI3-	A2
B3	LPC_FRAME#	GBE0_MDI3+	A3
B4	LPC_AD0	GBE0_LINK100#	A4
B5	LPC_AD1	GBE0_LINK1000#	A5
B6	LPC_AD2	GBE0_MDI2-	A6
B7	LPC_AD3	GBE0_MDI2+	A7
B8	LPC_DRQ0#	GBE0_LINK#	A8
B9	LPC_DRQ1#	GBE0_MDI1-	A9
B10	LPC_CLK	GBE0_MDI1+	A10
B11	GND	GND	A11
B12	PWRBTN#	GBE0_MDI0-	A12
B13	SMB_CK	GBE0_MDI0+	A13
B14	SMB_DAT	GBE0_CTREF	A14
B15	SMB_ALERT#	SUS_S3#	A15
B16	N/C	SATA0_TX+	A16
B17	N/C	SATA0_TX-	A17
B18	SUS_STAT#	SUS_S4#	A18
B19	N/C	SATA0_RX+	A19
B20	N/C	SATA0_RX-	A20
B21	GND	GND	A21
B22	N/C	SATA2_TX+	A22
B23	N/C	SATA2_TX-	A23
B24	PWR_OK	SUS_S5#	A24
B25	N/C	SATA2_RX+	A25
B26	N/C	SATA2_RX-	A26
B27	WDT	BATLOW#	A27
B28	AC_SDIN2	ATA_ACT#	A28
B29	AC_SDIN1	AC_SYNC	A29
B30	AC_SDIN0	AC_RST#	A30
B31	GND	GND	A31
B32	SPKR	AC_BITCLK	A32
B33	I2C_CK	AC_SDOUT	A33
B34	I2C_DAT	BIOS_DISABLE#	A34
B35	THRM#	THRMTRIP#	A35
B36	USB7-	USB6-	A36
B37	USB7+	USB6+	A37
B38	USB_4_5_OC#	USB_6_7_OC#	A38
B39	USB5-	USB4-	A39
B40	USB5+	USB4+	A40
B41	GND	GND	A41
B42	USB3-	USB2-	A42
B43	USB3+	USB2+	A43
B44	USB_0_1_OC#	USB_2_3_OC#	A44
B45	USB1-	USB0-	A45
B46	USB1+	USB0+	A46
B47	EXCD1_PERST#	VCC_RTC	A47
B48	EXCD1_CPPE#	EXCD0_PERST#	A48
B49	SYS_RESET#	EXCD0_CPPE#	A49
B50	CB_RESET#	LPC_SERIRQ	A50
B51	GND	GND	A51
B52	N/C	N/C	A52
B53	N/C	N/C	A53
B54	GPO1	GPI0 D54	A54
B55	N/C	N/C	A55

B56	N/C	N/C	A56
B57	GPO2	GND	A57
B58	N/C	N/C	A58
B59	N/C	N/C	A59
B60	GND	GND	A60
B61	PCIE_RX2+	PCIE_TX2+	A61
B62	PCIE_RX2-	PCIE_TX2-	A62
B63	GPO3	GPI1	A63
B64	PCIE_RX1+	PCIE_TX1+	A64
B65	PCIE_RX1-	PCIE_TX1-	A65
B66	WAKE0#	GND	A66
B67	WAKE1#	GPI2	A67
B68	PCIE_RX0+	PCIE_TX0+	A68
B69	PCIE_RX0-	PCIE_TX0-	A69
B70	GND	GND	A70
B71	LVDS_B0+	LVDS_A0+	A71
B72	LVDS_B0-	LVDS_A0-	A72
B73	LVDS_B1+	LVDS_A1+	A73
B74	LVDS_B1-	LVDS_A1-	A74
B75	LVDS_B2+	LVDS_A2+	A75
B76	LVDS_B2-	LVDS_A2-	A76
B77	N/C	LVDS_VDD_EN	A77
B78	N/C	N/C	A78
B79	LVDS_BKLT_EN	N/C	A79
B80	GND	GND	A80
B81	LVDS_B_CK+	LVDS_A_CK+	A81
B82	LVDS_B_CK-	LVDS_A_CK-	A82
B83	CKLVDS_BKLT_CTRL	LVDS_I2C_CK	A83
B84	VCC_5V_SBY	LVDS_I2C_DAT	A84
B85	VCC_5V_SBY	GPI3	A85
B86	VCC_5V_SBY	KBD_RST#	A86
B87	VCC_5V_SBY	KBD_A20GATE	A87
B88	RSVD	PCIE0_CK_REF+	A88
B89	VGA_RED	PCIE0_CK_REF-	A89
B90	GND	GND	A90
B91	VGA_GRN	RSVD B91	A91
B92	VGA_BLU	RSVD	A92
B93	VGA_HSYNC	GPO0	A93
B94	VGA_VSYNC	RSVD	A94
B95	VGA_I2C_CK	RSVD	A95
B96	VGA_I2C_DAT	GND	A96
B97	TV_DAC_A	VCC_12V	A97
B98	TV_DAC_B	VCC_12V	A98
B99	TV_DAC_C	VCC_12V	A99
B100	GND	GND	A100
B101	VCC_12V	VCC_12V	A101
B102	VCC_12V	VCC_12V	A102
B103	VCC_12V	VCC_12V	A103
B104	VCC_12V	VCC_12V	A104
B105	VCC_12V	VCC_12V	A105
B106	VCC_12V	VCC_12V	A106
B107	VCC_12V	VCC_12V	A107
B108	VCC_12V	VCC_12V	A108
B109	VCC_12V	VCC_12V	A109
B110	GND	GND	A110

2.5 COM Express CD Connector

D1	GND	GND	C1
D2	IDE_D5	IDE_D7	C2
D3	IDE_D10	IDE_D6	C3
D4	IDE_D11	IDE_D3	C4
D5	IDE_D12	IDE_D15	C5
D6	IDE_D4	IDE_D8	C6
D7	IDE_D0	IDE_D9	C7
D8	IDE_REQ	IDE_D2	C8
D9	IDE_IOW#	IDE_D13	C9
D10	IDE_ACK#	IDE_D1	C10
D11	GND	GND	C11
D12	IDE_IRQ	IDE_D14	C12
D13	IDE_A0	IDE_IORDY	C13
D14	IDE_A1	IDE_IOR#	C14
D15	IDE_A2	PCI_PME#	C15
D16	IDE_CS1#	PCI_GNT2#	C16
D17	IDE_CS3#	PCI_REQ2#	C17
D18	IDE_RESET#	PCI_GNT1#	C18
D19	PCI_GNT3#	PCI_REQ1#	C19
D20	PCI_REQ3#	PCI_GNT0#	C20
D21	GND	GND	C21
D22	PCI_AD1	PCI_REQ0#	C22
D23	PCI_AD3	PCI_RESET#	C23
D24	PCI_AD5	PCI_AD0	C24
D25	PCI_AD7	PCI_AD2	C25
D26	PCI_C/BE0#	PCI_AD4	C26
D27	PCI_AD9	PCI_AD6	C27
D28	PCI_AD11	PCI_AD8	C28
D29	PCI_AD13	PCI_AD10	C29
D30	PCI_AD15	PCI_AD12	C30
D31	GND	GND	C31
D32	PCI_PAR	PCI_AD14	C32
D33	PCI_SERR#	PCI_C/BE1#	C33
D34	PCI_STOP#	PCI_PERR#	C34
D35	PCI_TRDY#	PCI_LOCK#	C35
D36	PCI_FRAME#	PCI_DEVSEL#	C36
D37	PCI_AD16	PCI_IRDY#	C37
D38	PCI_AD18	PCI_C/BE2#	C38
D39	PCI_AD20	PCI_AD17	C39
D40	PCI_AD22	PCI_AD19	C40
D41	GND	GND	C41
D42	PCI_AD24	PCI_AD21	C42
D43	PCI_AD26	PCI_AD23	C43
D44	PCI_AD28	PCI_C/BE3#	C44
D45	PCI_AD30	PCI_AD25	C45
D46	PCI_IRQC#	PCI_AD27	C46
D47	PCI_IRQD#	PCI_AD29	C47
D48	PCI_CLKRUN#	PCI_AD31	C48
D49	PCI_M66EN	PCI_IRQA#	C49
D50	PCI_CLK	PCI_IRQB#	C50
D51	GND	GND (FIXED)	C51
D52	PEG_TX0+	PEG_RX0+	C52
D53	PEG_TX0-	PEG_RX0-	C53
D54	PEG_LANE_RV#	TYPE0#	C54
D55	PEG_TX1+	PEG_RX1+	C55

D56	PEG_TX1-	PEG_RX1-	C56
D57	TYPE2#	TYPE1#	C57
D58	PEG_TX2+	PEG_RX2+	C58
D59	PEG_TX2-	PEG_RX2-	C59
D60	GND	GND	C60
D61	PEG_TX3+	PEG_RX3+	C61
D62	PEG_TX3-	PEG_RX3-	C62
D63	RSVD	RSVD	C63
D64	RSVD	RSVD	C64
D65	PEG_TX4+	PEG_RX4+	C65
D66	PEG_TX4-	PEG_RX4-	C66
D67	GND	RSVD	C67
D68	PEG_TX5+	PEG_RX5+	C68
D69	PEG_TX5-	PEG_RX5-	C69
D70	GND	GND	C70
D71	PEG_TX6+	PEG_RX6+	C71
D72	PEG_TX6-	PEG_RX6-	C72
D73	SDVO_CLK	SDVO_DATA	C73
D74	PEG_TX7+	PEG_RX7+	C74
D75	PEG_TX7-	PEG_RX7-	C75
D76	GND	GND	C76
D77	IDE_CBLID#	RSVD	C77
D78	PEG_TX8+	PEG_RX8+	C78
D79	PEG_TX8-	PEG_RX8-	C79
D80	GND	GND	C80
D81	PEG_TX9+	PEG_RX9+	C81
D82	PEG_TX9-	PEG_RX9-	C82
D83	RSVD	RSVD	C83
D84	GND	GND	C84
D85	PEG_TX10+	PEG_RX10+	C85
D86	PEG_TX10-	PEG_RX10-	C86
D87	GND	GND	C87
D88	PEG_TX11+	PEG_RX11+	C88
D89	PEG_TX11-	PEG_RX11-	C89
D90	GND	GND	C90
D91	PEG_TX12+	PEG_RX12+	C91
D92	PEG_TX12-	PEG_RX12-	C92
D93	GND	GND	C93
D94	PEG_TX13+	PEG_RX13+	C94
D95	PEG_TX13-	PEG_RX13-	C95
D96	GND	GND	C96
D97	PEG_ENABLE#	RSVD	C97
D98	PEG_TX14+	PEG_RX14+	C98
D99	PEG_TX14-	PEG_RX14-	C99
D100	GND	GND	C100
D101	PEG_TX15+	PEG_RX15+	C101
D102	PEG_TX15-	PEG_RX15-	C102
D103	GND	GND	C103
D104	VCC_12V	VCC_12V	C104
D105	VCC_12V	VCC_12V	C105
D106	VCC_12V	VCC_12V	C106
D107	VCC_12V	VCC_12V	C107
D108	VCC_12V	VCC_12V	C108
D109	VCC_12V	VCC_12V	C109
D110	GND	GND	C110

2.6 The Installation Paths of CD Driver

Windows 2000 & XP

Driver	Path
CHIPSET	\CHIPSET\INF 9.11
LAN	\ETHERNET\REALTEK\8111_WINXP_5764
VGA	\GRAPHICS\INTEL_2K_XP_32\1432

Windows 7

Driver	Path
CHIPSET	\CHIPSET\INF 9.11
LAN	Windows 7 built-in LAN driver
VGA	\GRAPHICS\INTEL_WIN7_32\1930 \GRAPHICS\INTEL_WIN7_64\1930

This page is intentionally left blank.

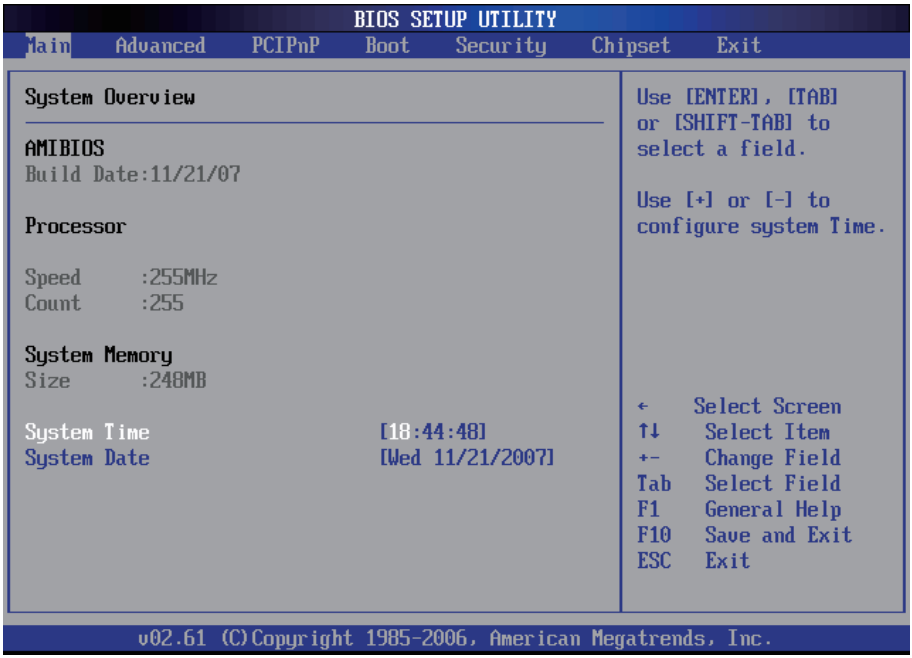
Chapter 3

BIOS

3.1 BIOS Main Setup

The AMI BIOS provides a setup utility program for specifying the system configurations and settings. The BIOS ROM of the system stores the setup utility.

When you turn on the computer, the AMI BIOS is immediately activated. The Main displays system overview status. Use the left/right arrow keys to highlight a particular configuration screen from the top menu bar or use the down arrow key to access and configure the information below.



System Time

Set the system time.
The time format is:

Hour : 00 to 23
Minute : 00 to 59
Second : 00 to 59

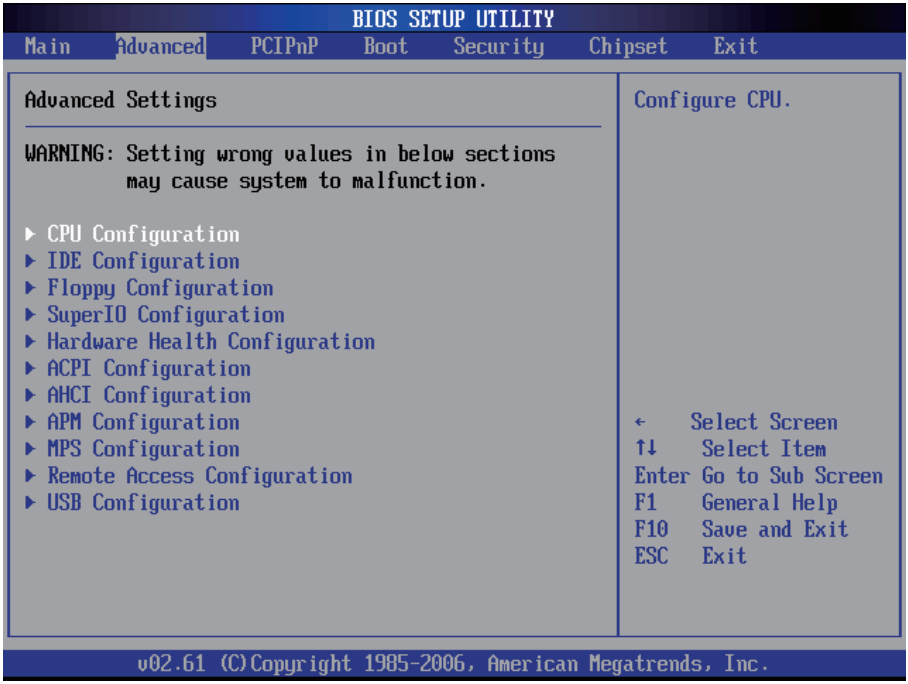
System Date

Set the system date. Note that the 'Day' automatically changes when you set the date.

The date format is:

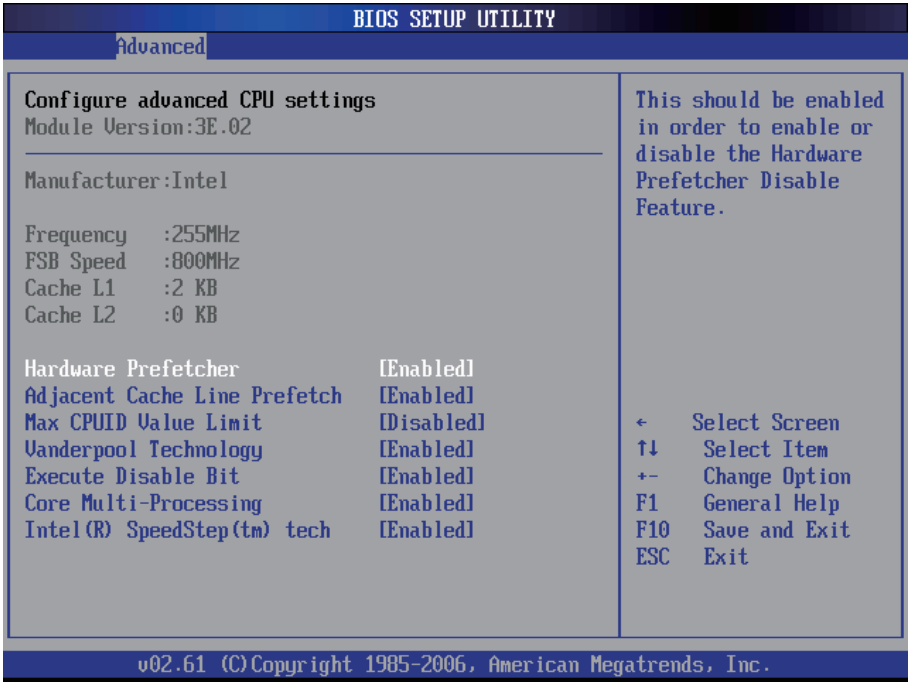
Day : Sun to Sat
Month : 1 to 12
Date : 1 to 31
Year : 1999 to 2099

3.2 Advanced Settings



3.2.1 CPU Configuration

The CPU Configuration setup screen varies depending on the installed processor.



Hardware Prefetcher

This should be enabled in order to enable or disable the Hardware Prefetcher Disable Feature.

Enable - Enable Hardware Prefetcher.
Disabled - Disable Hardware Prefetcher.

Adjacent Cache Line Prefetch

This should be enabled in order to enable or disable the cache Prefetcher Disable Feature.

The choice: Enabled, Disabled.

Cache L1 & L2

CPU Internal Cache & External Cache:
These two categories speed up memory access. However, it depends on CPU/chipset design.

Enable - Enable cache.
Disabled - Disable cache

Max CPUID Value Limit

Disabled for Windows XP.

Vanderpool Technology

Enable this item will allow a platform to run multiple virtual operating systems and applications in independent partitions.

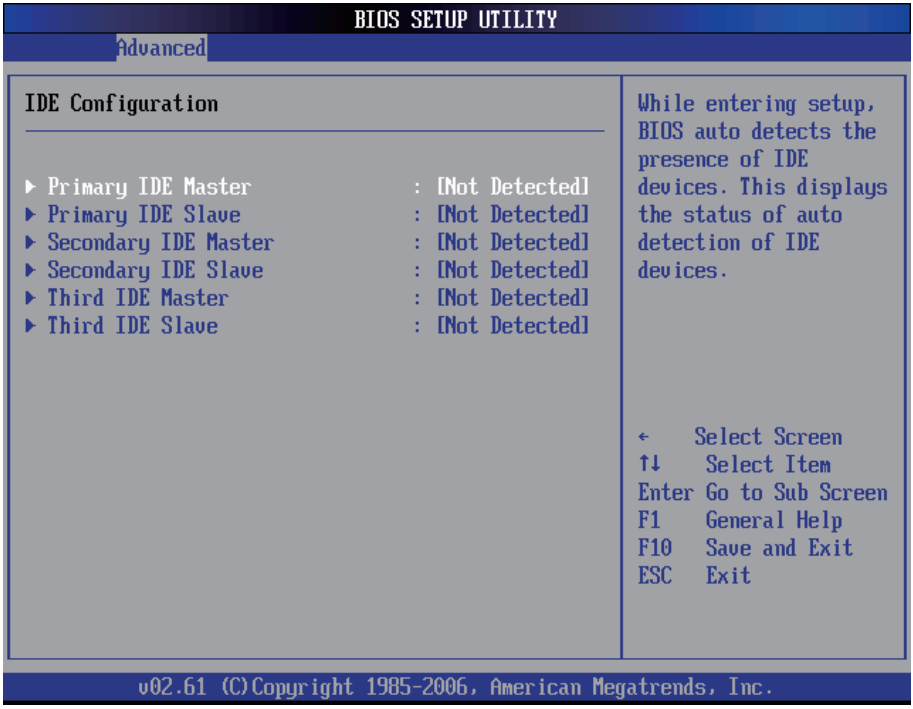
Core Multi-Processing

Enabled or disabled the multi-processing functionality of the Core processor.

Intel® SpeedStep™ Tech

Maximum: CPU speed is set to maximum.
Minimum: CPU speed is set to minimum.
Automatic: CPU speed controlled by Operating system.
Disabled: Default CPU speed.

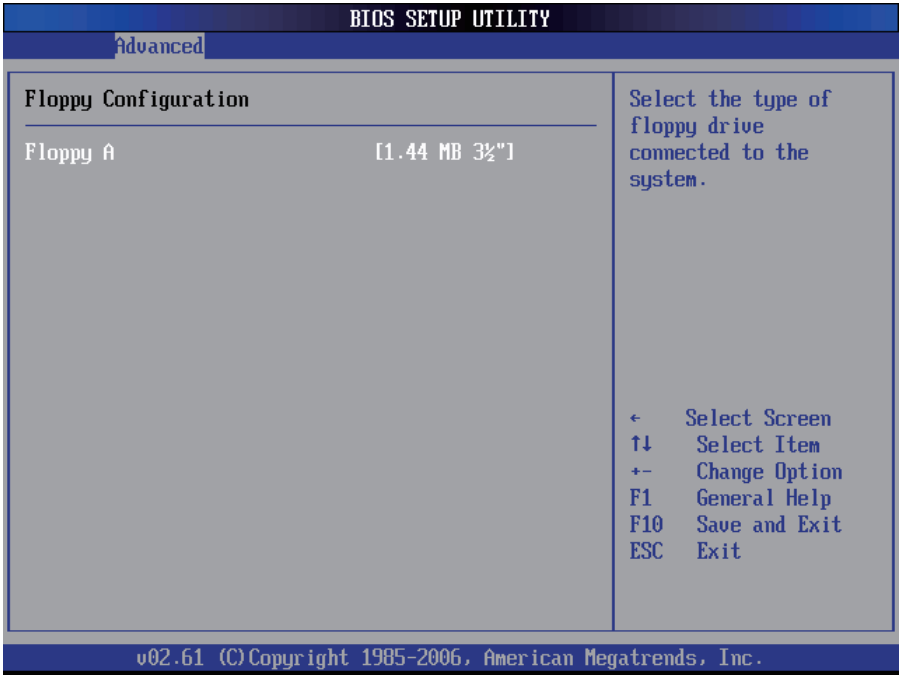
3.2.2 IDE Configuration



Primary/Secondary/Third IDE Master/Slave

Select one of the hard disk drives to configure. Press <Enter> to access its sub menu.

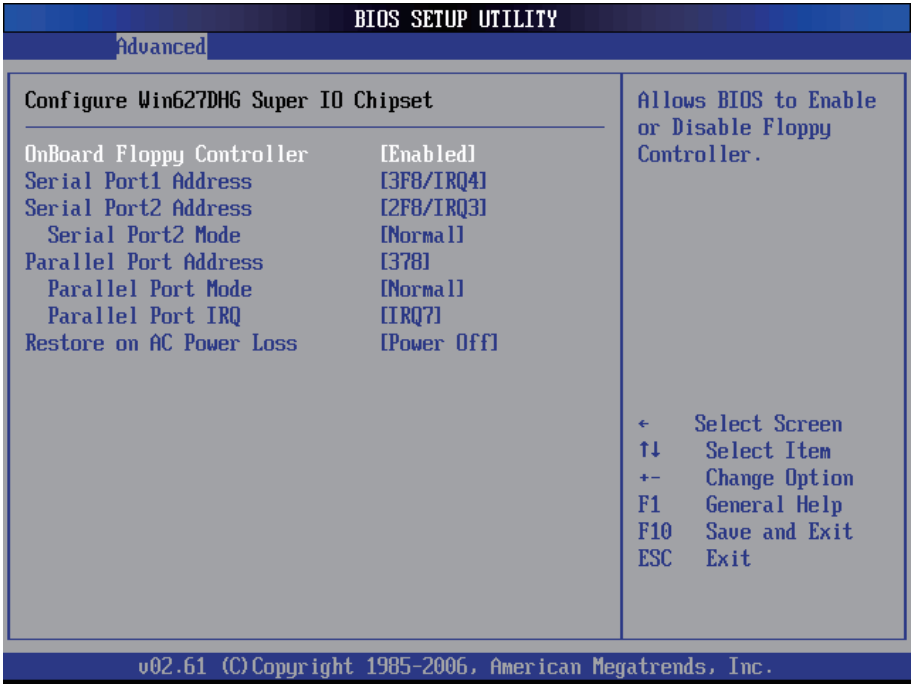
3.2.3 Floppy Configuration



Select the type of floppy disk drive installed in your system.
The choice:

None
360K 5.25”
1.2M 5.25”
720K 3.5”
1.44M 3.5”
2.88M 3.5”

3.2.4 Super IO Configuration



Onboard Floppy Controller

Select “Enabled” if your system has a floppy disk controller (FDC) installed on the system board and you wish to use it. If you didn’t install an FDC or the system has no floppy drive, select Disabled in this field.

The Choice: Enabled, Disabled

Serial Port1 / Port2 Address

Select an address and corresponding interrupt for the first and second serial ports.

The choice:

- 3F8/IRQ4
- 2E8/IRQ3
- 3E8/IRQ4
- 2F8/IRQ3
- Disabled
- Auto

Serial Port2 Mode

Allows BIOS to select mode for serial Port2.

Parallel Port Address

Select an address for the parallel port.
The choice:

- 3BC
- 378
- 278
- Disabled

Parallel Port Mode

Select an operating mode for the onboard parallel port. Select Normal, Compatible or SPP unless you are certain both of your hardware and software support one of the other available modes.
The choice:

- SPP
- EPP
- ECP
- ECP + EPP
- Normal

Parallel Port IRQ

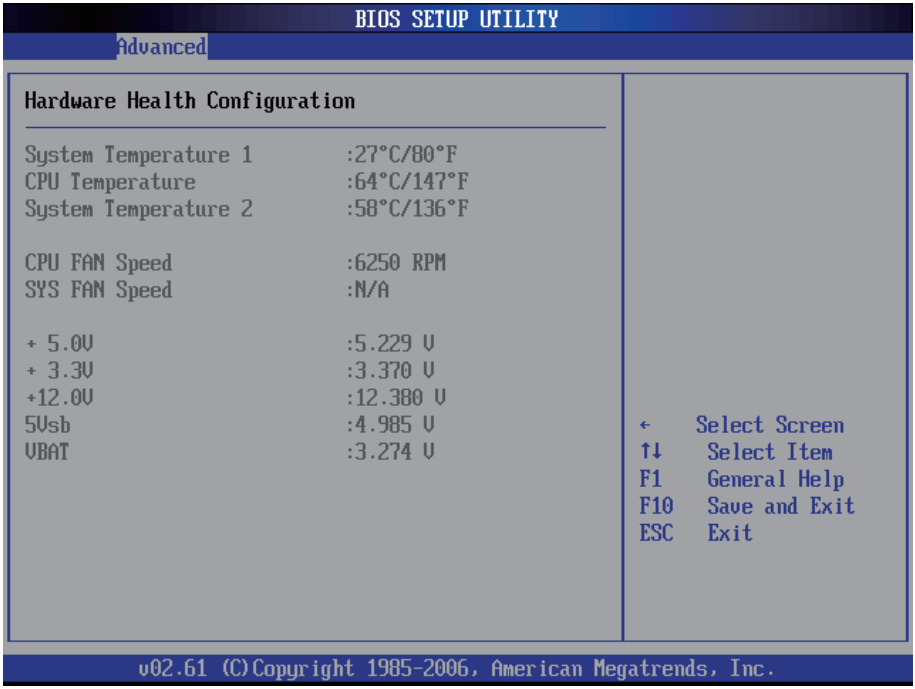
Select an interrupt for the parallel port.
The choice:

- IRQ5
- IRQ7

Restore on AC Power Loss by IO

This item allows you to select if you want to power on the system after power failure.

3.2.5 Hardware Health Configuration



System/ CPU Temperature 1

Displays the current System / CPU fan temperature.

CPU / System Fan Speed

Shows the current CPU / System Fan operating speed.

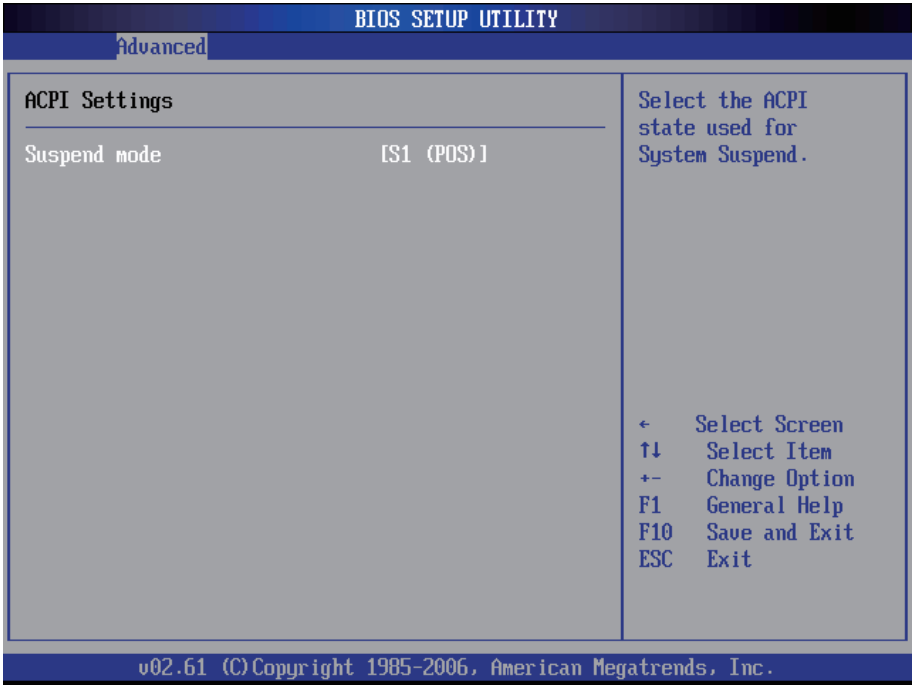
Vcore

Displays the voltage level of CPU (Vcore).

+5.0V / +3.3V / +12.0V / 5Vsb / VBAT

Shows the voltage level of the +3.3V, +5.0V, +12.0V, +5V standby and battery.

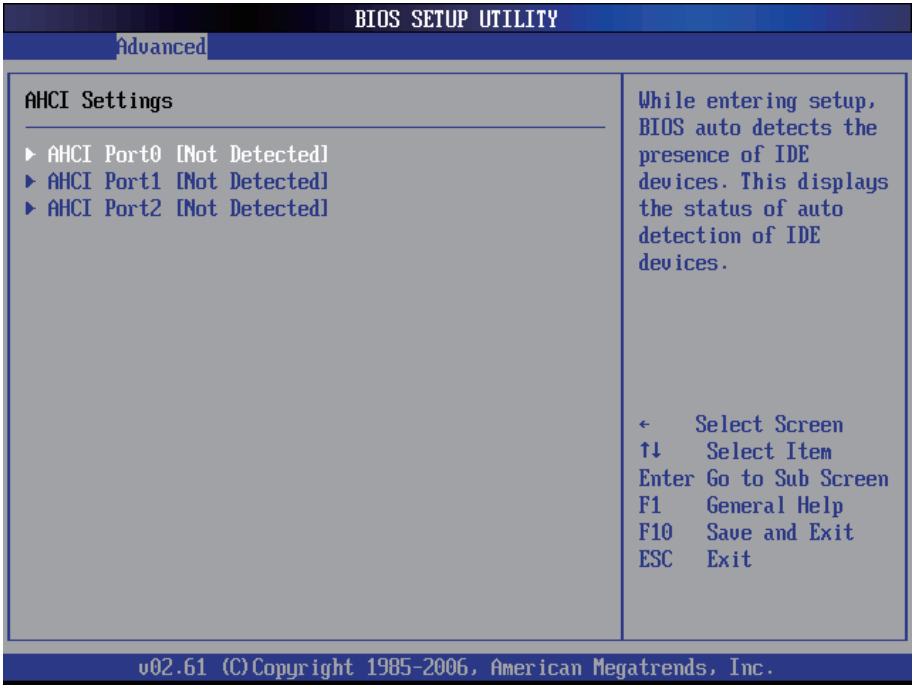
3.2.6 ACPI Configuration



Suspend mode

Select the ACPI state used for System Suspend.
The Choice: S1 (POS)

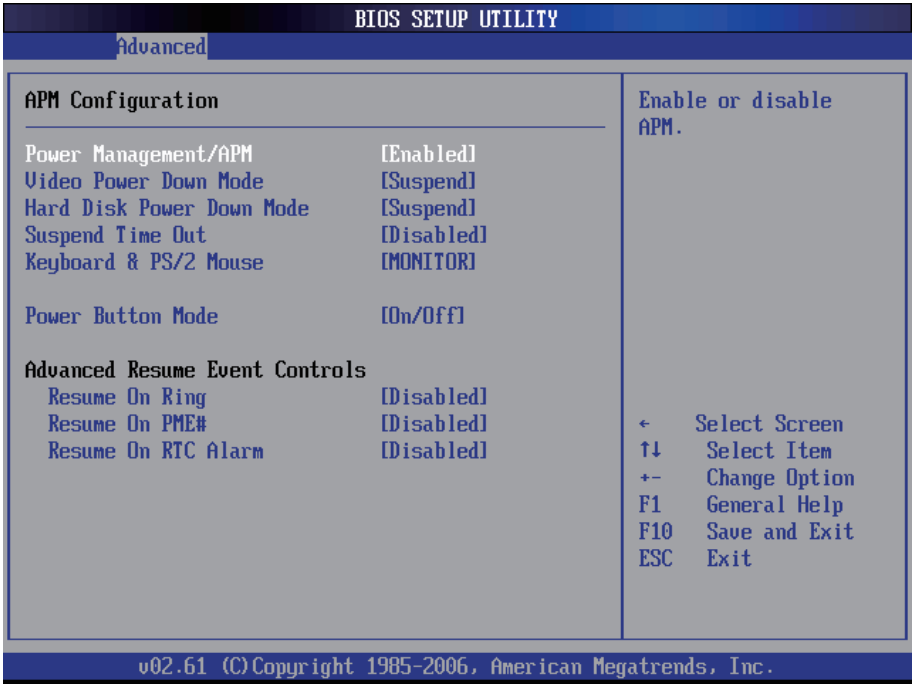
3.2.7 AHCI Configuration



AHCI Port 0 / Port 1 / Port 2

While entering setup, BIOS auto detects the presence of IDE devices. This displays the status of auto detection of IDE devices.

3.2.8 APM Configuration



Power Management/APM

This category allows you to select the type (or degree) of power saving and is directly related to the following modes:
1. HDD Power Down
2. Doze Mode
3. Suspend Mode

Video Power Down Mode

This option defines the level of power-saving mode requires in to power down the video display. As a default, the video powers down both in suspend mode and standby mode.

The Choice: Enabled, Disabled

Hard Disk Power Down Mode

Power Down Hard Disk in Suspend or Standby Mode.

The Choice: Enabled, Disabled

Suspend Time Out
Go into Suspend in the specified time.
The Choice: Enabled, Disabled

Keyboard & PS/2 Mouse
Monitor KBC ports 60/64.

Power Button Mode
Pressing the power button for more than 4 seconds forces the system to enter the Soft-Off state when the system has “hang”.

The Choice: Delay 4 Sec, On/Off

Advanced Resume Event Controls
Resume On Ring
An input signal on the serial Ring Indicator (RI) line (in other words an incoming call on the modem) awakens the system from a soft off state.

The Choice: Enabled, Disabled

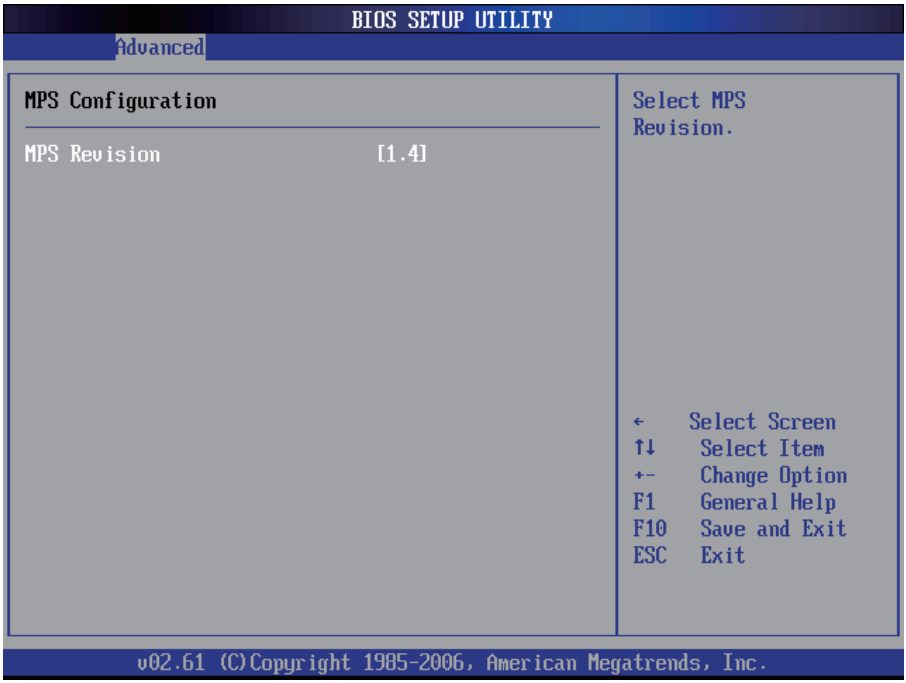
Resume On PME#
An input signal from a PME on the PCI card awakens the system from a soft off state.

The Choice: Enabled, Disabled

Resume On RTC Alarm
When “Enabled”, you can set the date and time at which the RTC (real-time clock) alarm awakens the system from Suspend mode.

The Choice: Enabled, Disabled

3.2.9 MPS Configuration

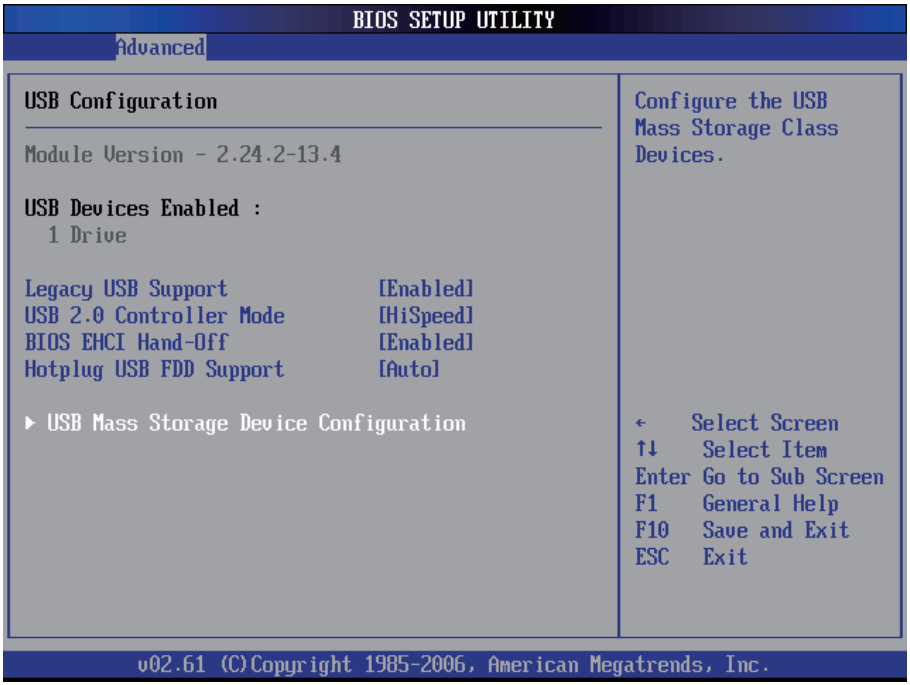


MPS Revision

Select the operating system that is Multi-Processors Version Control for OS.

The Choice: 1.4, 1.1.

3.2.10 USB Configuration



Legacy USB Support

Enables support for legacy USB. AUTO option disables legacy support if no USB devices are connected.

USB 2.0 Controller Mode

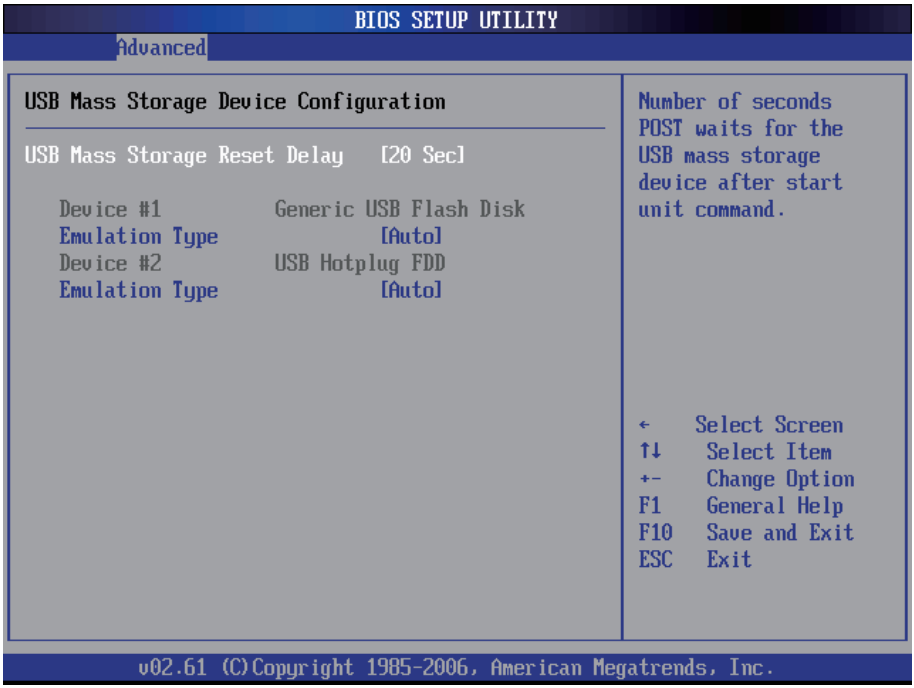
Configures the USB 2.0 controller in High Speed (480Mbps) or Full Speed (12MBPS).

BIOS EHCI Hand-Off

This is a work around for OSs without EHCI hand-Off support. The EHCI ownership change should be claimed by EHCI driver.

USB Mass Storage Reset Delay

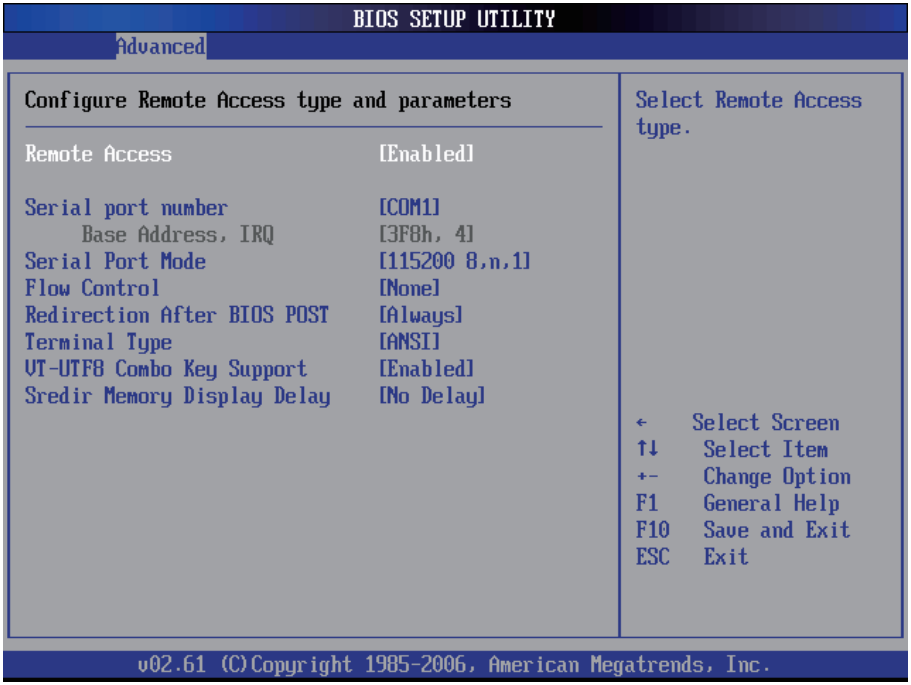
Number of seconds POST waits for the USB mass storage device after start unit command.



Emulation Type

If Auto, USB devices less than 530MB will be emulated as Floppy and remaining as hard drive. Forced FDD option can be used to force a HDD formatted drive to BOOT as FDD. (Ex. ZIP drive).

3.2.11 Remote Access Configuration



Remote Access

Configure Remote Access type and parameters.

The Choice: Enabled, Disabled.

Serial port number

Select Serial Port for console redirection. Make sure the selected port is enabled.

The Choice: COM1, COM2.

Base Address, IRQ

Select Serial Port for console redirection. Make sure the selected port is enabled. The Choice:

3F8, IRQ4
2E8, IRQ3
3E8, IRQ4
2F8, IRQ3

3.3 Advanced PCI/PnP Settings

Clear NVRAM

Clear NVRAM during System BOOT.

The Choice: Yes, No.

Plug & Play O/S

No: Lets the BIOS configure all the devices in the system.
Yes: lets the operating system configure Plug and Play (PnP) devices not required for BOOT if your system has a Plug and Play operating system.

PCI Latency Timer

Value in units of PCI clocks for PCI device latency timer register.

Allocate IRQ to PCI VGA

Yes: Assigns IRQ to PCI VGA card if card requests IRQ.
No: Does not assign IRQ to PCI VGA card even if card requests an IRQ.

IRQ3 - IRQ15

Available: Specified IRQ is available to be used by PCI/PnP devices.
Reserved: Specified IRQ is reserved for use by Legacy ISA devices.

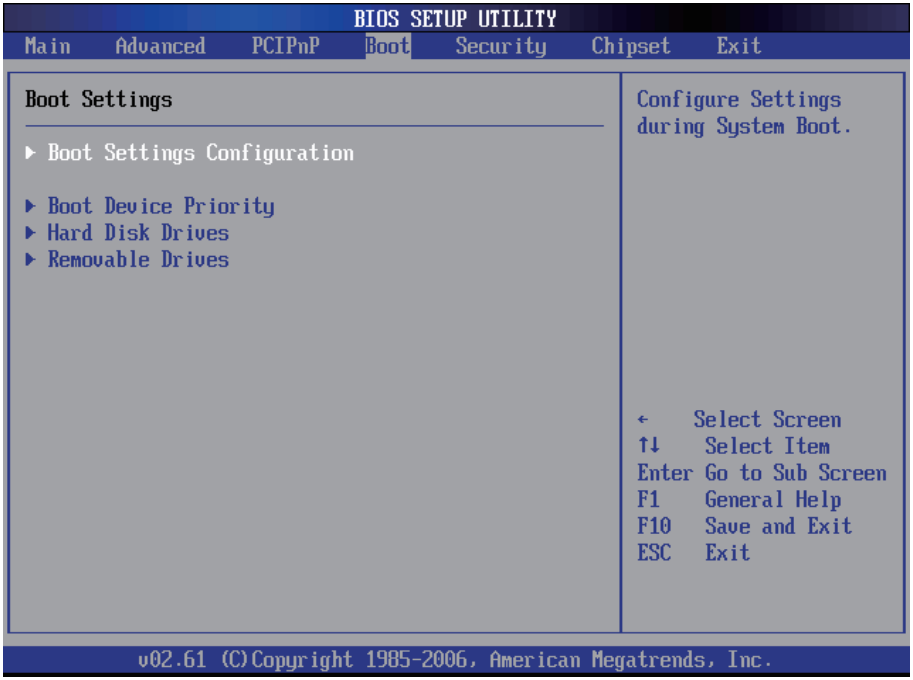
DMA Channel 0 - DMA Channel 7

Available: Specified DMA is available to be used by PCI/PnP devices.
Reserved: Specified DMA is reserved for use by Legacy ISA devices.

Reserved Memory Size

Size of memory block to reserve for legacy ISA devices.

3.4 Boot Settings



Boot Device Priority

Press Enter and it shows Bootable add-in devices.

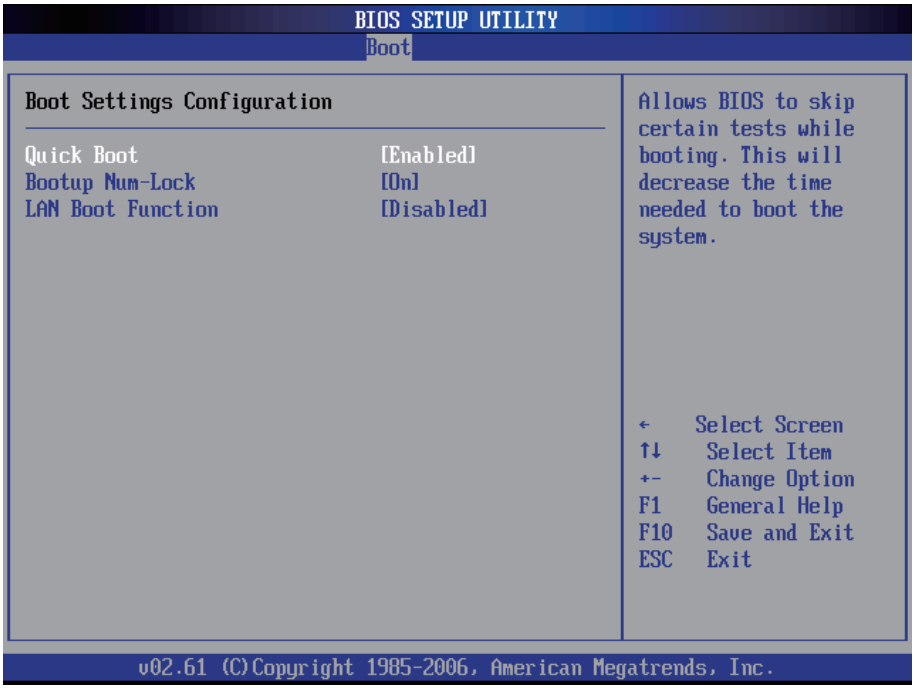
Hard Disk Drives

Press Enter and it shows Bootable and Hard Disk drives.

Removable Drives

Press Enter and it shows Bootable and Removable drives.

3.4.1 Boot Settings Configuration



Quick Boot

Allows BIOS to skip certain tests while booting. This will decrease the time needed to boot the system.

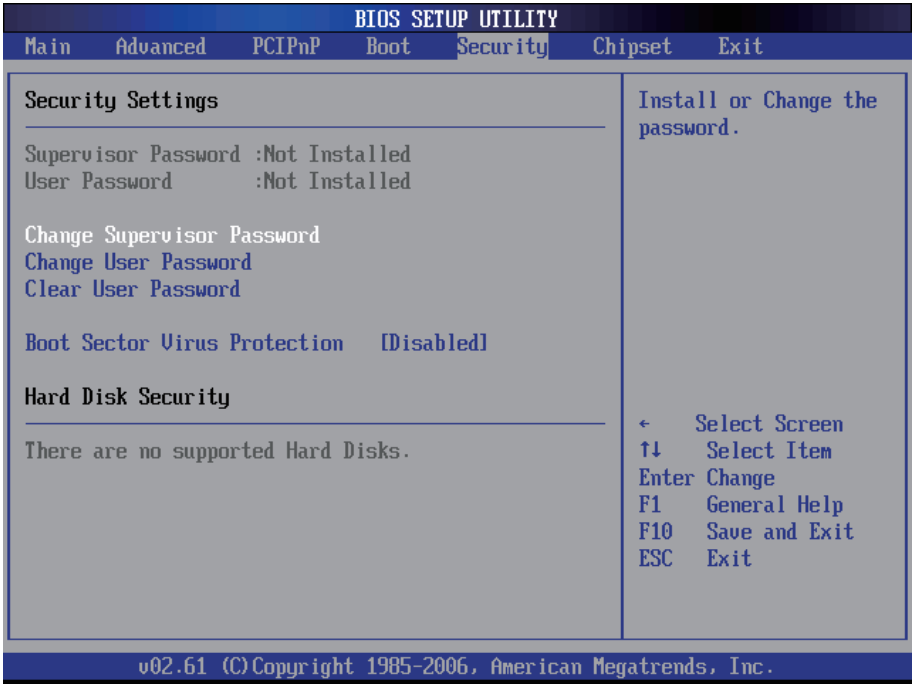
Bootup Num-Lock

Set this value to allow the Number Lock setting to be modified during boot up.

LAN Boot Function

Set this option to LAN add-on Boot ROM function.

3.5 Security



Supervisor Password & User Password

You can set either supervisor or user password, or both of then. The differences between are:

Set ***Supervisor Password***: Can enter and change the options of the setup menus.

Set ***User Password***: Just can only enter but do not have the right to change the options of the setup menus. When you select this function, the following message will appear at the center of the screen to assist you in creating a password.

ENTER PASSWORD:

Type the password, up to eight characters in length, and press <Enter>. The password typed now will clear any previously entered password from CMOS memory. You will be asked to confirm the password. Type the password again and press <Enter>. You may also press <ESC> to abort the selection and not enter a password.

To disable a password, just press <Enter> when you are prompted to enter the password. A message will confirm the password will be disabled. Once the password is disabled, the system will boot and you can enter Setup freely.

PASSWORD DISABLED.

When a password has been enabled, you will be prompted to enter it every time you try to enter Setup. This prevents an unauthorized person from changing any part of your system configuration.

Additionally, when a password is enabled, you can also require the BIOS to request a password every time your system is rebooted. This would prevent unauthorized use of your computer.

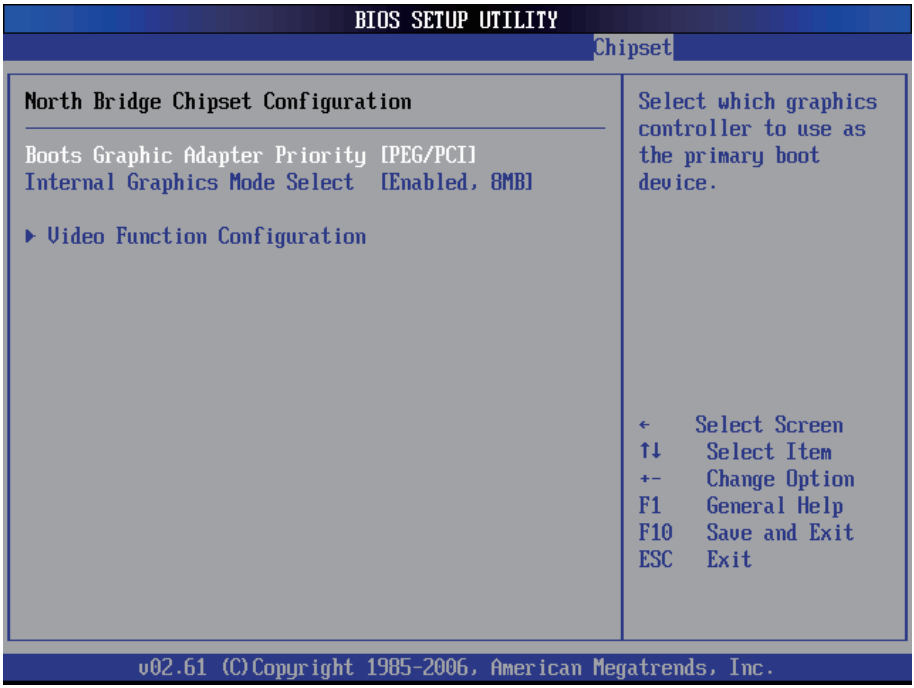
You determine when the password is required within the BIOS Features Setup Menu and its Security option. If the Security option is set to “System”, the password will be required both at boot and at entry to Setup. If set to “Setup”, prompting only occurs when trying to enter Setup.

Boot Sector Virus Protection

Enable/Disable Boot Sector Virus Protection.

3.6 Advanced Chipset Settings

3.6.1 North Bridge Chipset Configuration



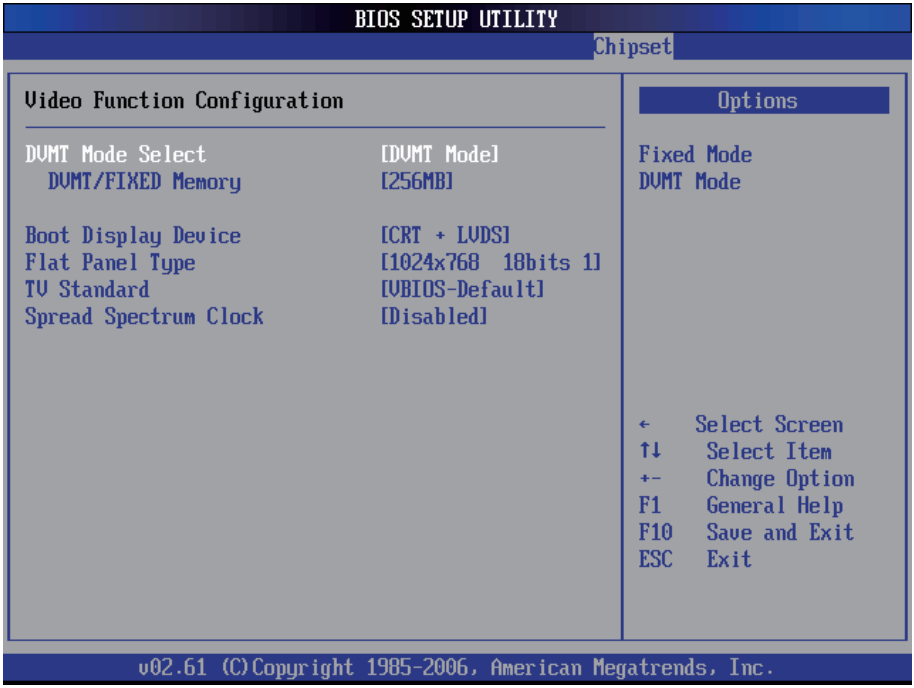
Boots Graphic Adapter Priority

Select which graphics controller to use as the primary boot device.

Internal Graphic Mode Select

Select the amount of system memory used by the Internal graphics device.

Video Function Configuration



DVMT Mode Select

The Choice: FIXED, DVMT (Default), Both.

DVMT/FIXED Memory

The Choice: 64MB, 128MB (Default), 224MB.

Boot Display

The Choice: CRT + LVDS (Default).

Flat Panel Type

It allows you to select the LCD Panel type as below ---
The Choice: 640x480
800x600
1024x768 18bits 1 (Default)
1280x1024
1400x1050
1400x1050
1600x1200
1280x768
1680x1050
1920x1200

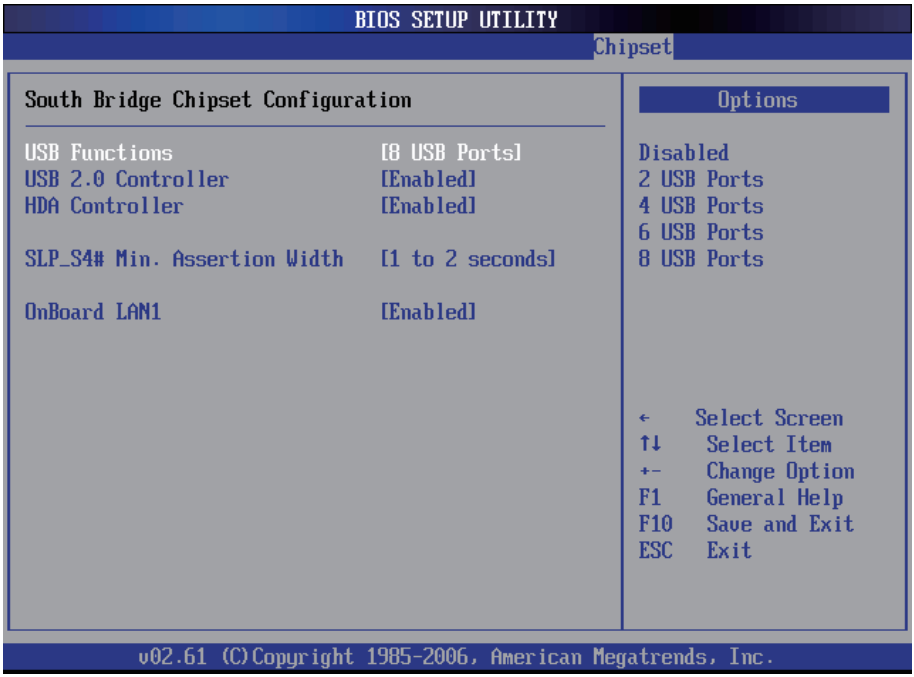
TV Standard

The Choice: VBIOS-Default

Spread Spectrum Clock

It sets the value of the spread spectrum. It is for CE testing use only.
The Choice: Disabled (Default), Enabled.

3.6.2 South Bridge Chipset Configuration



USB Funtion

This item allows you to active USB ports.

The Choice:

- Disabled
- 2 USB Ports
- 4 USB Ports
- 6 USB Ports
- 8 USB Ports
- 10 USB Ports

USB 2.0 Controller

Select “Enabled” if your system contains a Universal Serial Bus 2.0 (USB 2.0) controller and you have USB peripherals.

The Choice: Enabled, Disabled.

HDA Controller

This item allows you to select the chipset family to support High Definition Audio Controller.

The Choice: Enabled, Disabled.

SLP_S4# Min. Assertion Width

The item allows you to select the assertion width of SLP_S4#.
The Choice:

4 to 5 Seconds.
3 to 4 Seconds.
2 to 3 Seconds.
1 to 2 Seconds.

Onboard LAN1

Select “Enabled” if your system has a LAN device installed on the system board and you wish to use it.

The Choice: Enabled, Disabled.

3.7 Exit Options

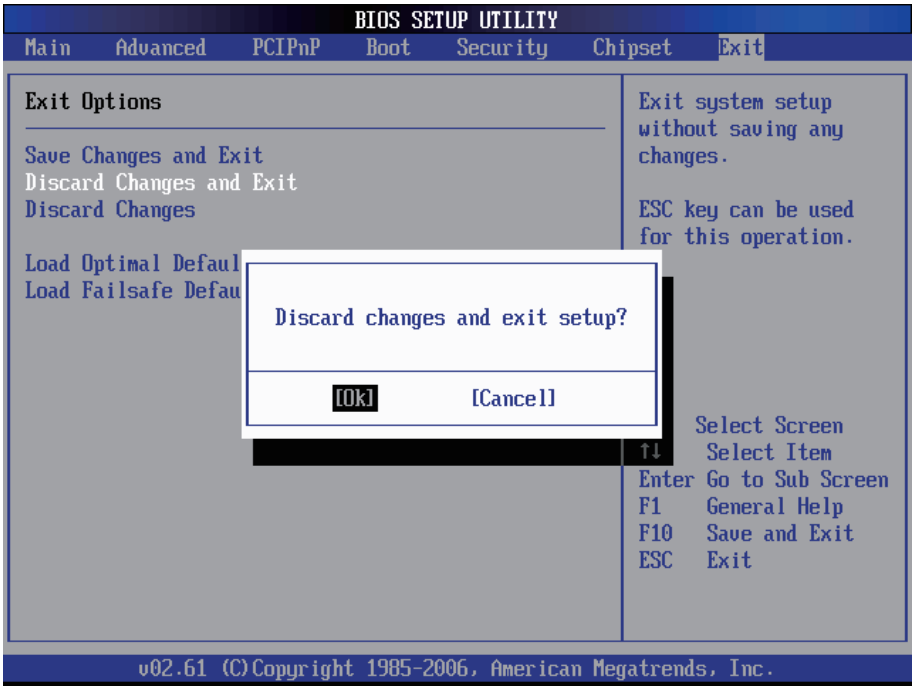
Save Changes and Exit

Pressing <Enter> on this item asks for confirmation:

Save configuration changes and exit setup?

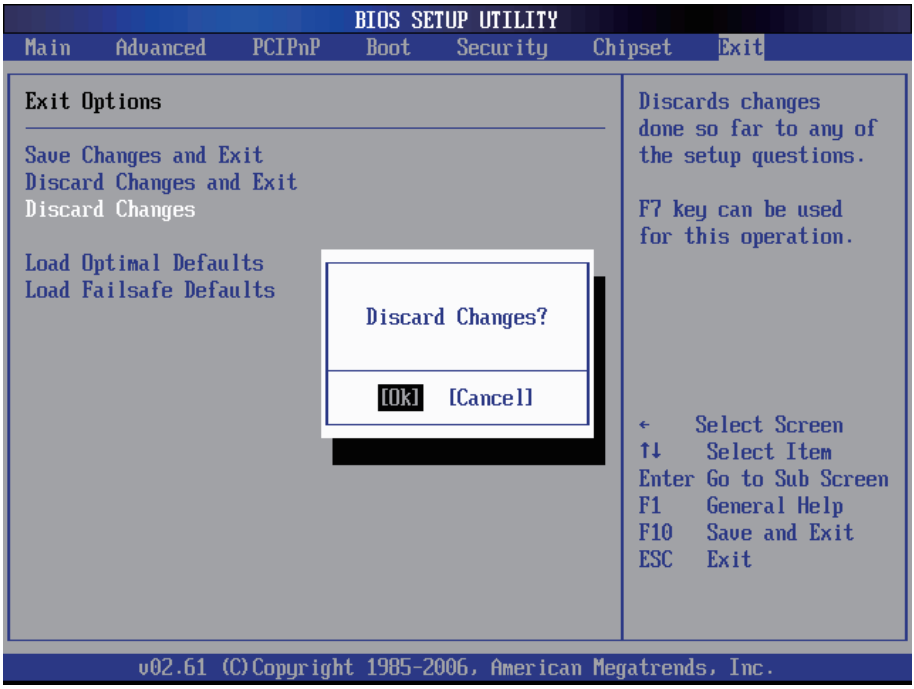
Pressing <OK> stores the selection made in the menus in CMOS - a special section of memory that stays on after you turn your system off. The next time you boot your computer, the BIOS configures your system according to the Setup selections stored in CMOS. After saving the values the system is restarted again.

Discard Changes and Exit



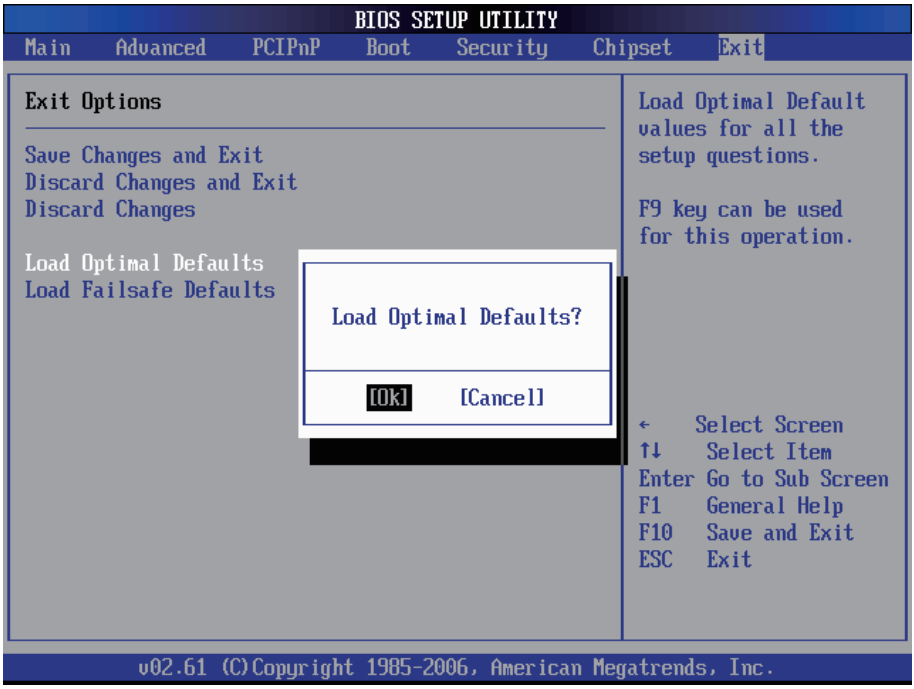
Exit system setup without saving any changes.
<ESC> key can be used for this operation.

Discard Changes



Discards changes done so far to any of the setup questions.
<F7> can be used for this operation.

Load Optimal Defaults



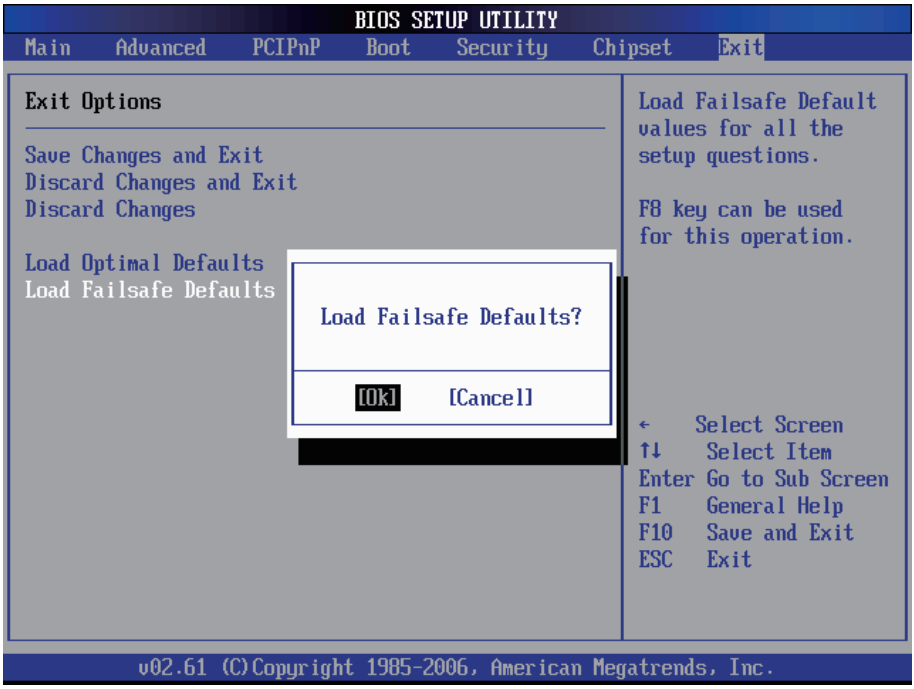
When you press <Enter> on this item you get a confirmation dialog box with a message:

Load Optimal Defaults?
[OK] [Cancel]

Pressing [OK] loads the BIOS Optimal Default values for all the setup questions.

<F9> key can be used for this operation.

Load Failsafe Defaults



When you press <Enter> on this item you get a confirmation dialog box with a message:

Load Failsafe Defaults?
[OK] [Cancel]

Pressing [OK] loads the BIOS Failsafe Default values for all the setup questions.

<F8> key can be used for this operation.

3.8 Beep Sound codes list

3.8.1 Boot Block Beep Codes

Number of Beeps	Description
1	Insert diskette in floppy drive A:
2	'AMIBOOT.ROM' file not found in root directory of diskette in A:
4	Flash Programming successful
5	Floppy read error
6	Keyboard controller BAT command failed
7	No Flash EPROM detected
8	Floppy controller failure
9	Boot Block BIOS checksum error
10	Flash Erase error
11	Flash Program error
12	'AMIBOOT.ROM' file size error
13	BIOS ROM image mismatch (file layout does not match image present in flash device)

3.8.2 POST BIOS Beep Codes

Number of Beeps	Description
1	Memory refresh timer error.
2	Parity error in base memory (first 64KB block)
4	Motherboard timer not operational
5	Processor error
6	8042 Gate A20 test error (cannot switch to protected mode)
7	General exception error (processor exception interrupt error)
8	Display memory error (system video adapter)
9	AMIBIOS ROM checksum error
10	CMOS shutdown register read/write error
11	Cache memory test failed

3.8.3 Troubleshooting POST BIOS Beep Codes

Number of Beeps	Description
1, 2 or 3	Reseat the memory, or replace with known good modules.
4-7, 9-11	Fatal error indicating a serious problem with the system. Consult your system manufacturer. Before declaring the motherboard beyond all hope, eliminate the possibility of interference by a malfunctioning add-in card. Remove all expansion cards except the video adapter. • If beep codes are generated when all other expansion cards are absent, consult your system manufacturer's technical support. • If beep codes are not generated when all other expansion cards are absent, one of the add-in cards is causing the malfunction. Insert the cards back into the system one at a time until the problem
8	If the system video adapter is an add-in card, replace or reset the video adapter. If the video adapter is an integrated part of the system board, the board may be faulty.

3.9 AMI BIOS Checkpoints

3.9.1 Bootblock Initialization Code Checkpoints

The Bootblock initialization code sets up the chipset, memory and other components before system memory is available. The following table describes the type of checkpoints that may occur during the bootblock initialization portion of the BIOS *(Note)*:

Checkpoint	Description
Before D0	If boot block debugger is enabled, CPU cache-as-RAM functionality is enabled at this point. Stack will be enabled from this point.
D0	Early Boot Strap Processo (BSP) initialization like microcode update, frequency and other CPU critical initialization. Early chipset initialization is done.
D1	Early super I/O initialization is done including RTC and keyboard controller. Serial port is enabled at this point if needed for debugging. NMI is disabled. Perform keyboard controller BAT test. Save power-on CPUID value in scratch CMOS. Go to flat mode with 4GB limit and GA20 enabled.
D2	Verify the boot block checksum. System will hang here if checksum is bad.
D3	Disable CACHE before memory detection. Execute full memory sizing module. If memory sizing module is not executed, start memory refresh and do memory sizing in Boot block code. Do additional chipset initialization. Re-enable CACHE. Verify that flat mode is enabled.
D4	Test base 512KB memory. Adjust policies and cache first 8MB. Set stack.
D5	Bootblock code is copied from ROM to lower system memory and control is given to it. BIOS now executes out of RAM. Copy compressed boot block code to memory in right segments. Copy BIOS from ROM to RAM for faster access. Perform main BIOS checksum and update recovery status accordingly.

D6	Both key sequence and OEM specific method are checked to determine if BIOS recovery is forced. If BIOS recovery is necessary, control flows tocheckpoint E0. See *Bootblock Recovery Code Checkpoints* section of document for more information.
D7	Restore CPUID value back into register. The Bootblock-Runtime interface module is moved to system memory and control is given to it. Determine whether to execute serial flash.
D8	The Runtime module is uncompressed into memory. CPUID information is stored in memory.
D9	Store the Uncompressed pointer for future use in PMM. Copying Main BIOS into memory. Leaves all RAM below 1MB Read-Write including E000 and F000 shadow areas but closing SMRAM.
DA	Restore CPUID value back into register. Give control to BIOS POST (ExecutePOSTKernel). See POST Code Checkpoints section of document for more information.
DC	System is waking from ACPI S3 state
E1 - E8 EC - EE	OEM memory detection/configuration error. This range is reserved for chipset vendors & system manufacturers. The error associated with this value may be different from one platform to the next.

3.9.2 Bootblock Recovery Code Checkpoints

The Bootblock recovery code gets control when the BIOS determines that a BIOS recovery needs to occur because the user has forced the update or the BIOS checksum is corrupt. The following table describes the type of checkpoints that may occur during the Bootblock recovery portion of the BIOS *(Note)*:

Checkpoint	Description
E0	Initialize the floppy controller in the super I/O. Some interrupt vectors are initialized. DMA controller is initialized. 8259 interrupt controller is initialized. L1 cache is enabled.
E9	Set up floppy controller and data. Attempt to read from floppy.
EA	Enable ATAPI hardware. Attempt to read from ARMD and ATAPI CDROM.
EB	Disable ATAPI hardware. Jump back to checkpoint E9.
EF	Read error occurred on media. Jump back to checkpoint EB.
F0	Search for pre-defined recovery file name in root directory.
F1	Recovery file not found.
F2	Start reading FAT table and analyze FAT to find the clusters occupied by the recovery file.
F3	Start reading the recovery file cluster by cluster.
F5	Disable L1 cache.
FA	Check the validity of the recovery file configuration to the current configuration of the flash part.
FB	Make flash write enabled through chipset and OEM specific method. Detect proper flash part. Verify that the found flash part size equals the recovery file size.
F4	The recovery file size does not equal the found flash part size.

FC	Erase the flash part.
FD	Program the flash part.
FF	The flash has been updated successfully. Make flash write disabled. Disable ATAPI hardware. Restore CPUID value back into register. Give control to F000 ROM at F000:FFF0h.

3.9.3 POST Code Checkpoints

The POST code checkpoints are the largest set of checkpoints during the BIOS pre-boot process. The following table describes the type of checkpoints that may occur during the POST portion of the BIOS *(Note)*:

Checkpoint	Description
03	Disable NMI, Parity, video for EGA, and DMA controllers. Initialize BIOS, POST, Runtime data area. Also initialize BIOS modules on POST entry and GPNV area. Initialized CMOS as mentioned in the Kernel Variable "wCMOSFlags."
04	Check CMOS diagnostic byte to determine if battery power is OK and CMOS checksum is OK. Verify CMOS checksum manually by reading storage area. If the CMOS checksum is bad, update CMOS with power-on default values and clear passwords. Initialize status register A. Initializes data variables that are based on CMOS setup questions. Initializes both the 8259 compatible PICs in the system
05	Initializes the interrupt controlling hardware (generally PIC) and interrupt vector table.
06	Do R/W test to CH-2 count reg. Initialize CH-0 as system timer.Install the POSTINT1Ch handler. Enable IRQ-0 in PIC for system timer interrupt. Traps INT1Ch vector to "POSTINT1ChHandlerBlock."
07	Fixes CPU POST interface calling pointer.
08	Initializes the CPU. The BAT test is being done on KBC. Program the keyboard controller command byte is being done after Auto detection of KB/MS using AMI KB-5.
C0	Early CPU Init Start -- Disable Cache – Init Local APIC
C1	Set up boot strap processor Information
C2	Set up boot strap processor for POST
C5	Enumerate and set up application processors
C6	Re-enable cache for boot strap processor

C7	Early CPU Init Exit
0A	Initializes the 8042 compatible Key Board Controller.
0B	Detects the presence of PS/2 mouse.
0C	Detects the presence of Keyboard in KBC port.
0E	Testing and initialization of different Input Devices. Also, update the Kernel Variables. Traps the INT09h vector, so that the POST INT09h handler gets control for IRQ1. Uncompress all available language, BIOS logo, and Silent logo modules.
13	Early POST initialization of chipset registers.
20	Relocate System Management Interrupt vector for all CPU in the system.
24	Uncompress and initialize any platform specific BIOS modules. GPNV is initialized at this checkpoint.
2A	Initializes different devices through DIM. See DIM Code Checkpoints section of document for more information.
2C	Initializes different devices. Detects and initializes the video adapter installed in the system that have optional ROMs.
2E	Initializes all the output devices.
31	Allocate memory for ADM module and uncompress it. Give control to ADM module for initialization. Initialize language and font modules for ADM. Activate ADM module.
33	Initializes the silent boot module. Set the window for displaying text information.
37	Displaying sign-on message, CPU information, setup key message, and any OEM specific information.

38	Initializes different devices through DIM. See DIM Code Checkpoints section of document for more information. USB controllers are initialized at this point.
39	Initializes DMAC-1 & DMAC-2.
3A	Initialize RTC date/time.
3B	Test for total memory installed in the system. Also, Check for DEL or ESC keys to limit memory test. Display total memory in the system.
3C	Mid POST initialization of chipset registers.
40	Detect different devices (Parallel ports, serial ports, and coprocessor in CPU, … etc.) successfully installed in the system and update the BDA, EBDA…etc.
52	Updates CMOS memory size from memory found in memory test. Allocates memory for Extended BIOS Data Area from base memory. Programming the memory hole or any kind of implementation that needs an adjustment in system RAM size if needed.
60	Initializes NUM-LOCK status and programs the KBD typematic rate.
75	Initialize Int-13 and prepare for IPL detection.
78	Initializes IPL devices controlled by BIOS and option ROMs.
7C	Generate and write contents of ESCD in NVRam.
84	Log errors encountered during POST.
85	Display errors to the user and gets the user response for error.
87	Execute BIOS setup if needed / requested. Check boot password if installed.
8C	Late POST initialization of chipset registers.
8D	Build ACPI tables (if ACPI is supported)
8E	Program the peripheral parameters. Enable/Disable NMI as selected
90	Initialization of system management interrupt by invoking all handlers. Please note this checkpoint comes right after checkpoint 20h
A1	Clean-up work needed before booting to OS.

A2	Takes care of runtime image preparation for different BIOS modules. Fill the free area in F000h segment with 0FFh. Initializes the Microsoft IRQ Routing Table. Prepares the runtime language module. Disables the system configuration display if needed.
A4	Initialize runtime language module. Display boot option popup menu.
A7	Displays the system configuration screen if enabled. Initialize the CPU's before boot, which includes the programming of the MTRR's.
A9	Wait for user input at config display if needed.
AA	Uninstall POST INT1Ch vector and INT09h vector.
AB	Prepare BBS for Int 19 boot. Init MP tables.
AC	End of POST initialization of chipset registers. De-initializes the ADM module.
B1	Save system context for ACPI. Prepare CPU for OS boot including final MTRR values.
00	Passes control to OS Loader (typically INT19h).

3.9.4 DIM Code Checkpoints

The Device Initialization Manager (DIM) gets control at various times during BIOS POST to initialize different system busses. The following table describes the main checkpoints where the DIM module is accessed *(Note)*:

Checkpoint	Description
2A	Initialize different buses and perform the following functions: Reset, Detect, and Disable (function 0); Static Device Initialization (function 1); Boot Output Device Initialization (function 2). Function 0 disables all device nodes, PCI devices, and PnP ISA cards. It also assigns PCI bus numbers. Function 1 initializes all static devices that include manual configured onboard peripherals, memory and I/O decode windows in PCI-PCI bridges, and noncompliant PCI devices. Static resources are also reserved. Function 2 searches for and initializes any PnP, PCI, or AGP video devices.
38	Initialize different buses and perform the following functions: Boot Input Device Initialization (function 3); IPL Device Initialization (function 4); General Device Initialization (function 5). Function 3 searches for and configures PCI input devices and detects if system has standard keyboard controller. Function 4 searches for and configures all PnP and PCI boot devices. Function 5 configures all onboard peripherals that are set to an automatic configuration and configures all remaining PnP and PCI devices.

While control is in the different functions, additional checkpoints are output to port 80h as a word value to identify the routines under execution. The low byte value indicates the main POST Code Checkpoint. The high byte is divided into two nibbles and contains two fields. The details of the high byte of these checkpoints are as follows:

HIGH BYTE XY

The upper nibble “X” indicates the function number that is being executed. “X” can be from 0 to 7.

0 = func#0, disable all devices on the BUS concerned.
2 = func#2, output device initialization on the BUS concerned.
3 = func#3, input device initialization on the BUS concerned.
4 = func#4, IPL device initialization on the BUS concerned.
5 = func#5, general device initialization on the BUS concerned.
6 = func#6, error reporting for the BUS concerned.
7 = func#7, add-on ROM initialization for all BUSes.
8 = func#8, BBS ROM initialization 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 = On-board System devices.
2 = ISA devices.
3 = EISA devices.
4 = ISA PnP devices.
5 = PCI devices.

3.9.5 ACPI Runtime Checkpoints

ACPI checkpoints are displayed when an ACPI capable operating system either enters or leaves a sleep state. The following table describes the type of checkpoints that may occur during ACPI sleep or wake events *(Note)*:

Checkpoint	Description
AC	First ASL check point. Indicates the system is running in ACPI mode.
AA	System is running in APIC mode.
01, 02, 03, 04, 05	Entering sleep state S1, S2, S3, S4, or S5.
10, 20, 30, 40, 50	Waking from sleep state S1, S2, S3, S4, or S5.

Note:
Please note that checkpoints may differ between different platforms based on system configuration. Checkpoints may change due to vendor requirements, system chipset or option ROMs from add-in PCI devices.

This page is intentionally left blank.

Appendix

Appendix A: I/O Port Address Map

Each peripheral device in the system is assigned a set of I/O port addresses which also becomes the identity of the device.

The following table lists the I/O port addresses used.

0000h - 0000Fh	DMA Controller
0080h - 009Fh	DMA Controller
00C0h - 00DFh	DMA Controller
0020h, 0021h	Programmable Interrupt Controller
00A0h, 00A1h	Programmable Interrupt Controller
0040h - 0043h	System Timer
0044h - 0047h	System Timer
0060h - 0064h	Keyboard Controller
0070h - 0073h	System CMOS/Real Time Clock
00F0h - 00FFh	Math Co-Processor
01F0h-01F7h	Primary IDE
0274h-0277h	ISAPNP Read Data Port
0279h , 0A79h	ISAPnP Configuration
02F8h-02FFh	COM_2 (If use)
0378h-037Ah	Parallel Port (If use)
03B0h-03BFh	MDA/MGA
03C0h-03CFh	EGA/VGA
03D4h-03D9h	CGA CRT register
03F0h-03F7h	Floppy Diskette
03F6h-03F6h	Primary IDE
03F8h-03FFh	COM_1 (If use)
0400h-041F	South Bridge SMB
04D0h-04D1h	IRQ Edge/level control ports
0500h-053Fh	South Bridge GPIO
0800h-087Fh	ACPI
0A00h-0A07h	PME

0A10h-0A17h	Hardware Monitor
0CF8h	PCI Configuration address
0CFCh	PCI Configuration Data

Appendix B: BIOS Memory Map

Item	Address	Description
1	00000h-9FFFFh	DOS Kernel Area
2	A0000h,BFFFFh	EGA and VGA Video Buffer (128KB)
3	C0000h-CFFFFh	EGA/VGA ROM
4	D0000h-DFFFFh	Adaptor ROM
5	E0000h-FFFFFh	System BIOS

Appendix C: Interrupt Request Lines (IRQ)

Peripheral devices use interrupt request lines to notify CPU for the service required. The following table shows the IRQ used by the devices on board.

Level	Function
IRQ 00	System Timer
IRQ 01	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
IRQ 02	VGA and Link to Secondary PIC
IRQ 03	Communications Port (COM2)
IRQ 04	Communications Port (COM1)
IRQ 05	PCI Device
IRQ 06	Standard floppy disk controller
IRQ 07	Parallel Port
IRQ 08	System CMOS/real time clock
IRQ 09	Microsoft ACPI-Compliant System
IRQ 10	PCI Device
IRQ 11	PCI Device
IRQ 12	PS/2 Compatible Mouse
IRQ 13	PFY exception
IRQ 14	Primary IDE Channel
IRQ 15	PCI Device

Appendix D: Digital I/O Setting

Below are the source codes written in assembly & C, please take them for Digital I/O application examples. The default I/O address is 6Eh.

Assembly Code

```
        mov     ax,402h
        mov     dx,ax
        mov     al,00h
        out     dx,al           ; clear i2c bus

        mov     ax,400h
        mov     dx,ax
        mov     al,0ffh
        out     dx,ax           ; clear i2c bus status

        mov     ax,404h
        mov     dx,ax
        mov     al,06eh
        out     dx,ax           ; Set I2C Device Address=6eh

        mov     ax,403h
        mov     dx,ax
        mov     al,010h
        out     dx,ax           ;select GPIO 1 (index=10h)

        mov     ax,405h
        mov     dx,ax
        mov     al,0ffh
        out     dx,ax           ;Set all GPIO 1 pin as output

        mov     ax,402h
        mov     dx,ax
        mov     al,048h
        out     dx,ax           ;start write, active

;---------------------------------------------------------
        mov     ax,402h
        mov     dx,ax
        mov     al,00h
        out     dx,al           ; clear i2c bus
```

```
        mov     ax,400h
        mov     dx,ax
        mov     al,0ffh
        out     dx,ax           ; clear i2c bus status

        mov     ax,404h
        mov     dx,ax
        mov     al,06eh
        out     dx,ax           ; Set I2C Device Address=6eh

        mov     ax,403h
        mov     dx,ax
        mov     al,020h
        out     dx,ax           ;select GPIO 2 (index=20h)

        mov     ax,405h
        mov     dx,ax
        mov     al,0ffh
        out     dx,ax           ;Set all GPIO 2 pin as output

        mov     ax,402h
        mov     dx,ax
        mov     al,048h
        out     dx,ax           ;start write, active

;-------------------------------------------------------------
        mov     ax,402h
        mov     dx,ax
        mov     al,00h
        out     dx,al           ; clear i2c bus

        mov     ax,400h
        mov     dx,ax
        mov     al,0ffh
        out     dx,ax           ; clear i2c bus status

        mov     ax,404h
        mov     dx,ax
        mov     al,06eh
        out     dx,ax           ; Set I2C Device Address=6eh

        mov     ax,403h
        mov     dx,ax
        mov     al,011h
        out     dx,ax           ;select GPIO 1 data register (index=11h)
```

```
        mov     ax,405h
        mov     dx,ax
        mov     al,0ffh
        out     dx,ax           ;Set all GPIO 1 data = high

        mov     ax,402h
        mov     dx,ax
        mov     al,048h
        out     dx,ax           ;start write, active

;-------------------------------------------------------------
        mov     ax,402h
        mov     dx,ax
        mov     al,00h
        out     dx,al           ; clear i2c bus

        mov     ax,400h
        mov     dx,ax
        mov     al,0ffh
        out     dx,ax           ; clear i2c bus status

        mov     ax,404h
        mov     dx,ax
        mov     al,06eh
        out     dx,ax           ; Set I2C Device Address=6eh

        mov     ax,403h
        mov     dx,ax
        mov     al,021h
        out     dx,ax           ;select GPIO 2 Data register (index=21h)

        mov     ax,405h
        mov     dx,ax
        mov     al,0ffh
        out     dx,ax           ;Set all GPIO 2 data = High

        mov     ax,402h
        mov     dx,ax
        mov     al,048h
        out     dx,ax           ;start write, active
```

C Language Code

```
/*-----      Include Header Area -----*/
#include "math.h"
#include "stdio.h"
#include "dos.h"

/*-----      routing, sub-routing -----*/
void main(int argc, char *argv[])
{
        int SMB_PORT_AD  = 0x400;
        int SMB_DEVICE_ADD = 0x6e;      /*75111R's Add=6eh */
        int i,j;

/*      Index x0, GPIO1x Output pin control, Set all pin as output      */
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x10,0xff);
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x20,0xff);
        delay(10);

/*      Index x1, GPIO1x Output Data value, all low  */
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x11,0x00);
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x21,0x00);

        delay(3000);

/*      Index x1, GPIO1x Output Data value, all high*/
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x11,0xff);
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x21,0xff);

        delay(3000);

/*      printf("Digital I/O pin 7,5,3,1 ouput high ...\n"); */
/*      Index x1, GPIO1x Output Data value          */
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x11,0xAA);
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x21,0xAA);

        delay(3000);

/*      printf("Digital I/O pin 6,4,2,0 ouput high ...\n"); */
/*      Index 11, GPIO1x Output Data value          */
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x11,0x55);
        SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x21,0x55);
        delay(1500);
```

```
}


SMB_Byte_WRITE(int SMPORT, int DeviceID, int REG_INDEX, int REG_DATA)
{
        outportb(SMPORT+02, 0x00);              /* clear */
        outportb(SMPORT+00, 0xff);              /* clear */
        delay(10);
        outportb(SMPORT+04, DeviceID);          /* I2C Device Address */
        outportb(SMPORT+03, REG_INDEX);         /* Register Address in device */
        outportb(SMPORT+05, REG_DATA);          /* Data Value */
        outportb(SMPORT+02, 0x48);              /* write, active*/
}
```

This page is intentionally left blank.