ITX-i45M2

Mini ITX Industrial Motherboard

User's Manual

Version 1.1

2009.11

This page is intentionally left blank.

Table of Contents

This page is intentionally left blank.

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 About this User's Manual

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 User's Manual, please consult your vendor before further handling.

1.3 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 came with the Single Board Computer, whenever components are separated from the system.

1.4 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.

1.5 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

1.6 Warranty

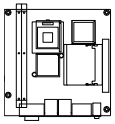
This product is warranted to be in good working order for a period of two years from the date of purchase. Should this product fail to be in good working order at any time during this period, we will, at our option, replace or repair it at no additional charge except as set forth in the following terms. 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 merchantibility 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.

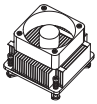
Return authorization must be obtained from the vendor before returned merchandise will be accepted. Authorization can be obtained by calling or faxing the vendor and requesting a Return Merchandise Authorization (RMA) number. Returned goods should always be accompanied by a clear problem description.

1.7 Packing List

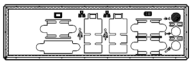


1 x ITX-i45M2 Industrial Motherboard

1 x Driver CD
1 x Quick Installation Guide



1 x CPU Cooler



1 x I/O bracket

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

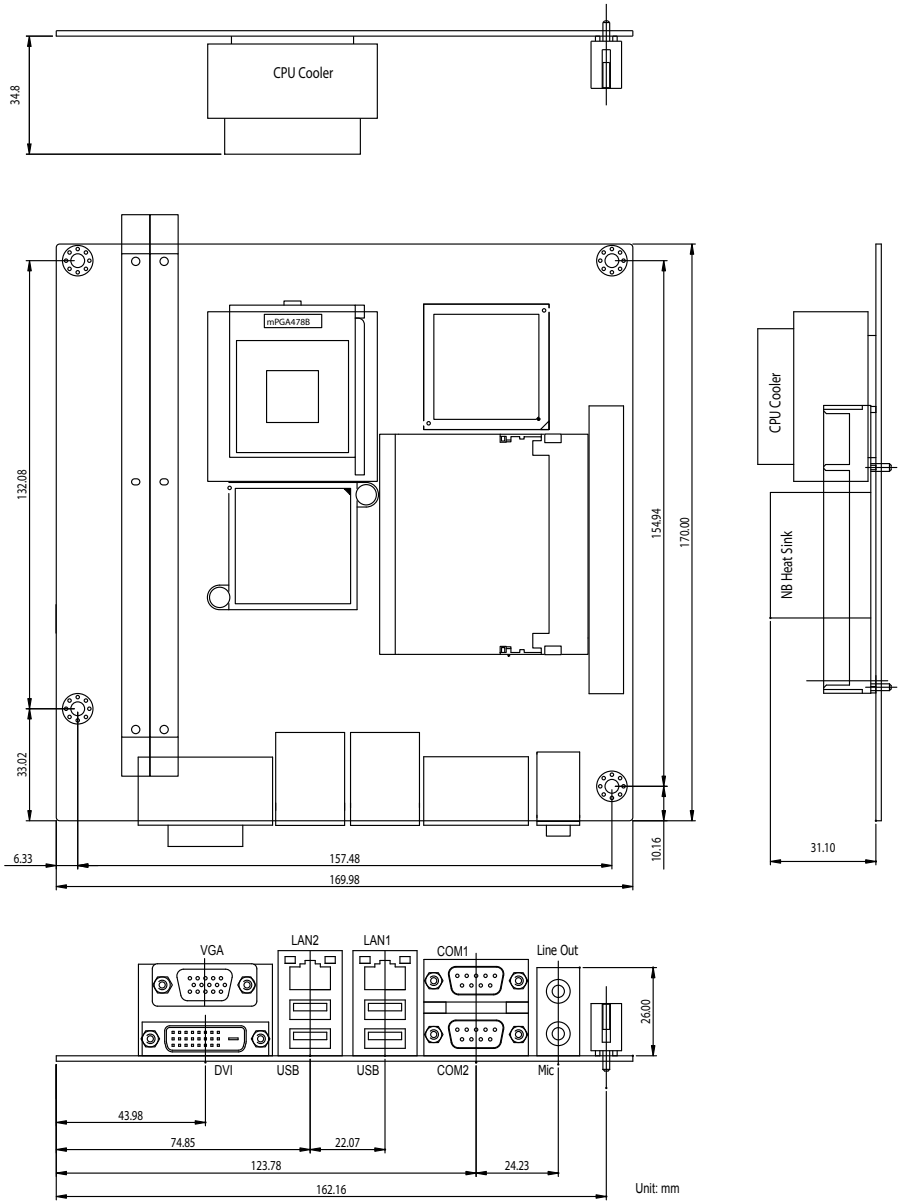
1.8 Ordering Information

ITX-i45M2	Intel® Core™ 2 Duo/ Celeron® M embedded Mini-ITX motherboard
CBK-06-4500-00	Cable kit 1 x USB Cable 1 x LPT to FDD Cable 1 x SATA Cable 1 x LPT Cable 1 x PS/2 Keyboard & Mouse Cable 1 x TV-out Cable

1.9 Specifications

Form Factor	Mini-ITX industrial motherboard
CPU	Socket-P Intel® Penryn Core™ 2 Duo with 667/800/1066MHz FSB, Celeron® M with 800MHz FSB Processor
Chipset	Intel® GM45 + Intel® ICH9M
System Memory	2 x 240-pin DIMM Sockets up to 4GB DDR2 SDRAM with 667/800MHz
VGA/ LCD Controller	Integrated Intel® Graphics Media Accelerator 4500MHD with Analog RGB/ Dual Channels LVDS/ TV-out/ DVI
Ethernet	2 x Realtek 8111B PCIe Gigabit Ethernet Controllers
I/O Chips	Winbond W83627HG
BIOS	AMI PnP Flash BIOS
Audio	Realtek ALC888 HD Audio CODEC, MIC-in/ Line-In/ Line-Out, default type is without Line-In
Serial ATA	4 x Serial ATA ports with 300MB/s HDD transfer rate
Serial Port	2 x COM ports (COM1: RS-232, COM2: RS-232/422/485 selectable)
Parallel Port/ Floppy	1 x SPP/EPP/ECP mode selectable, shared with FDD port
KBMS	1 x 6-pin Mini-DIN for PS/2 interface Keyboard and Mouse via Y-Cable
Universal Serial Bus	8 x USB 2.0 ports
DIO	8-bit programmable Digital Input/Output
Expansion Interface	1 x PCI Slot, 1 x Mini-PCI Socket
Operation Temp.	-20°C ~ 70°C (-4°F ~ 158°F)
Watchdog Timer	1~255 levels Reset
Dimension (L x W)	170 x 170 mm (6.7” x 6.7”)

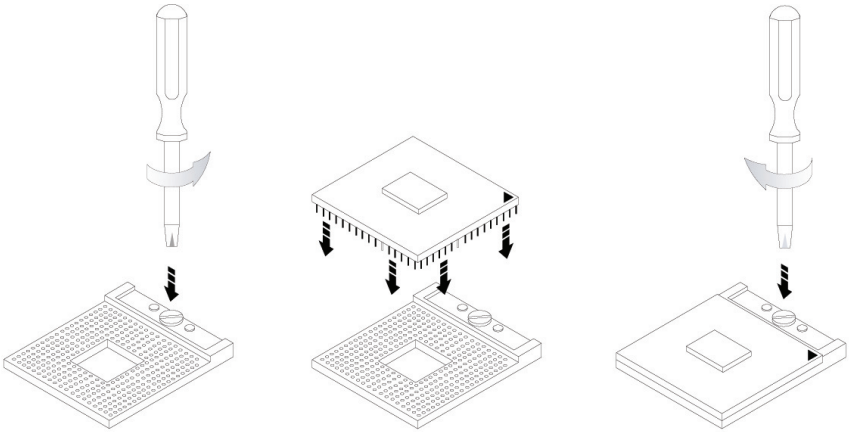
1.10 Board Dimensions



1.11 Installing the CPU

The processor socket comes with a screw to secure the CPU. As showing in the picture as bellow, loose the screw first before inserting the CPU.

Place the CPU into the socket by making sure the notch on the corner of the CPU corresponding with the notch on the inside of the socket. Once the CPU has slide into the socket, lock the screw.



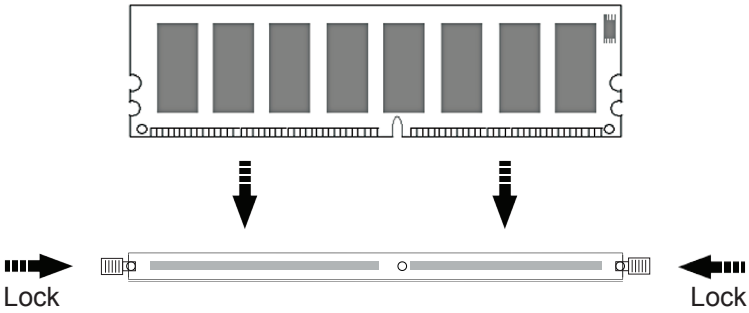
Make sure that heat sink of the CPU top surface is in complete contact to avoid the CPU overheating problem.

If not, it would cause your system or CPU to be hanged, unstable, damaged.

1.12 Installing the Memory

To install the Memory module, locate the Memory DIMM slot on the board and perform as below:

1. Hold the Memory module so that the key of the Memory module align with those on the Memory DIMM slot.
2. Gently push the Memory module in an upright position and a right way until the clips of the DIMM slot close to lock the Memory module in place, when the Memory module touches the bottom of the DIMM slot.
3. To remove the Memory module, just pressing the clips of DIMM slot with both hands.

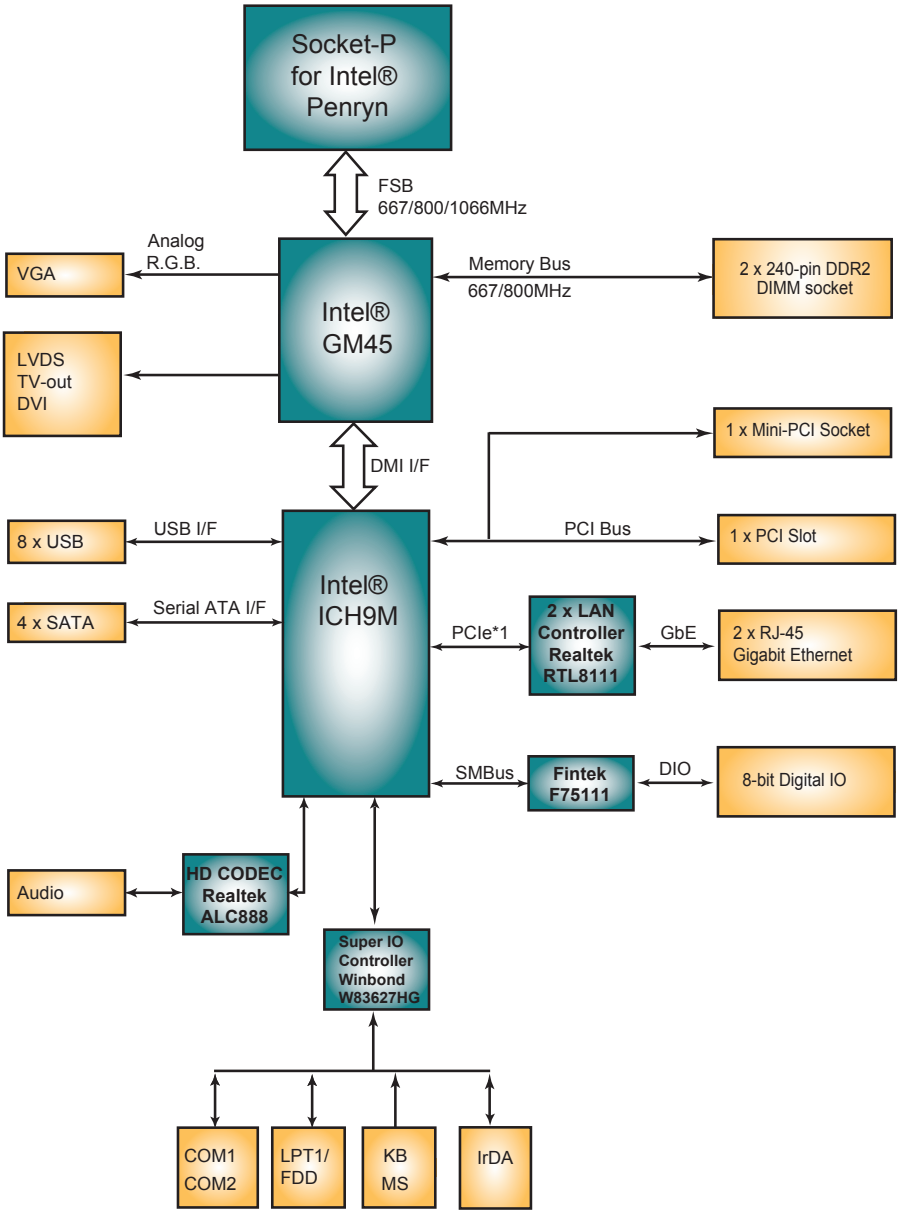


This page is intentionally left blank.

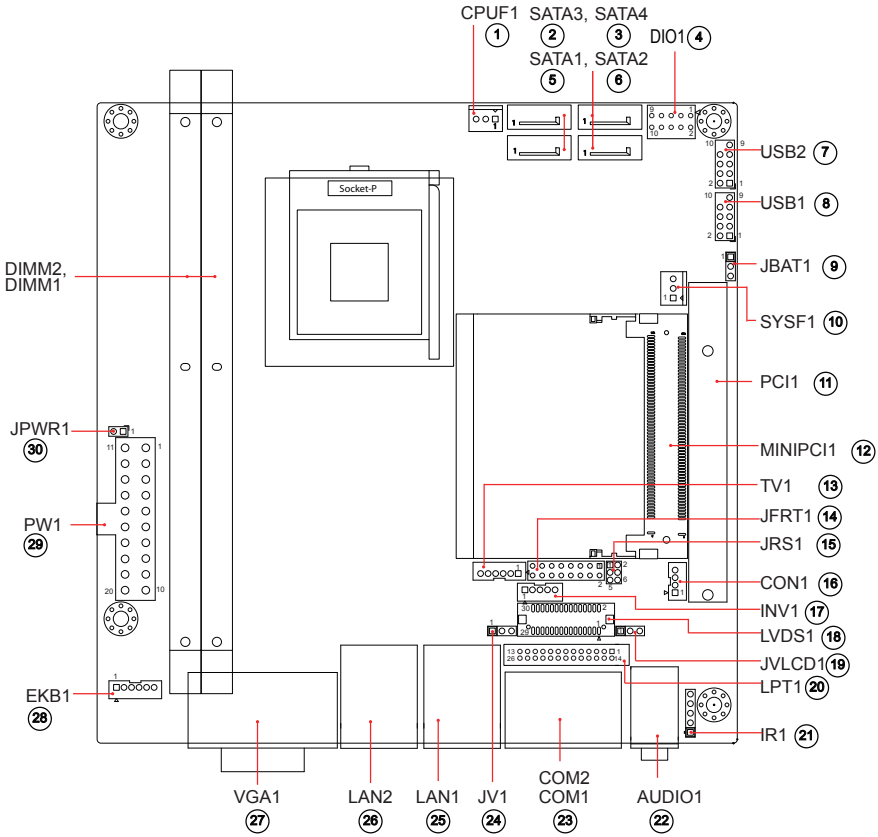
Chapter 2

Installation

2.1 Block Diagram



2.2 Jumpers and Connectors



Jumpers

JBAT1: Clear CMOS Setting (9)

If the board refuses to boot due to inappropriate CMOS settings, here is how to proceed to clear (reset) the CMOS to its default values.
Connector type: 2.54 mm pitch 1x3-pin headers

Pin	Mode	
1-2	Keep CMOS (Default)	
2-3	Clear CMOS	

You may need to clear the CMOS if your system cannot boot up because you forgot your password, the CPU clock setup is incorrect, or the CMOS settings need to be reset to default values after the system BIOS has been updated. Refer to the following solutions to reset your CMOS setting:

Solution A:

1. Power off the system and disconnect the power cable.
2. Place a shunt to short pin 1 and pin 2 of JBAT1 for five seconds.
3. Place the shunt back to pin 2 and pin 3 of JBAT1.
4. Power on the system.

Solution B:

If the CPU Clock setup is incorrect, you may not be able to boot up. In this case, follow these instructions:

1. Turn the system off, then on again. The CPU will automatically boot up using standard parameters.
2. As the system boots, enter BIOS and set up the CPU clock.

Note:

If you are unable to enter BIOS setup, turn the system on and off a few times.

JRS1: COM2 RS-232/422/485 Selection (15)

The onboard COM2 port can be configured to operate in RS-422 or RS-485 modes. RS-422 modes differ in the way RX/TX is being handled. Jumper JRS1 switches between RS-232 or RS-422/485 mode. When JRS1 is set to RS-422 or RS-485 mode, there will be only +12V output let while JRS1 is set. All RS-232/422/482 modes are available on COM2.
It can be configured COM2 to operate in RS-232, RS-422 or RS-485 mode.
Connector type: 2.00mm pitch 2x3-pin headers.

Mode	RS-232 (Default)	RS-422	RS-485
1-2	ON	OFF	OFF
3-4	OFF	ON	OFF
5-6	OFF	OFF	ON
			

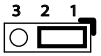
JVLCD1: LCD Panel Voltage Selection (19)

The voltage of LCD panel could be selected by JVLCD1 in +5V or +3.3V.
Connector type: 2.54 mm pitch 1x3-pin headers

Pin	Voltage	
1-2	+5V	
2-3	+3.3V (Default)	

JV1: COM port Power Special Support (24)

The COM1 port's voltage could be selected by LV1 respectively to +5V.
Connector type: 2.54mm pitch 1x3-pin headers.

Pin	Setup	
1-2	Standard signal for Pin-9. (Default)	
2-3	+5V	

JPWR1: AT/ATX Power Mode (30)

The power mode jumper selects the power mode for the system.
Connector type: 2.54mm pitch 1x2-pin headers.

Pin 1-2	Mode	
Short	AT Mode	
Open	ATX Mode (Default)	

Connectors

CPUF1: CPU Fan Connector (1)

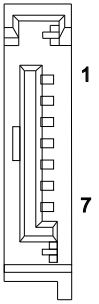
CPUF1 is 3-pin headers for the system fan. The fan must be a +12V fan.

Pin	Description
1	GND
2	+12V
3	FAN_Detect

SATA1~ 4: Serial ATA Connectors (2, 3, 5, 6)

The ITX-i45M2 on board supports four SATA connectors, second generation SATA drives transfer data at speeds as high as 300MB/s, twice the transfer speed of first generation SATA drives. The SATA drives can be configured in a RAID 0, RAID 1 or RAID 10 configuration.

Pin	Description
1	GND
2	TX+
3	TX-
4	GND
5	RX-
6	RX+
7	GND

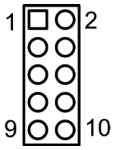


DIO1: Digital I/O Connector (4)

DIO1 is a 8-bit Digital Input / Output connector.

Connector type: 2.54 mm pitch 2x5-pin headers.

Pin	Description	Pin	Description
1	DIO1	2	DIO2
3	DIO3	4	DIO4
5	DIO5	6	DIO6
7	DIO7	8	DIO8
9	+5V	10	GND

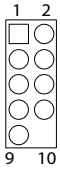


USB1, USB2: USB Connectors (7, 8)

The ITX-i45M2 CPU board on board supports three headers USB1 and USB2 that can connect up to six high-speed (Data transfers at 480Mb/s), full-speed (Data transfers at 12Mb/s) or low-speed (Data transfers at 1.5Mb/s) USB devices.

Connector type: 2.54mm 2x5-pin headers

Pin	Description	Pin	Description
1	+5V	2	+5V
3	USBD-	4	USBD-
5	USBD+	6	USBD+
7	GND	8	GND
9	GND	10	N/C (Key)



SYSF1: Systen Fan Connector (10)

SYSF1 is 3-pin headers for the system fan. The fan must be a +12V fan.

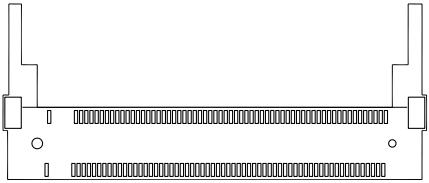
Pin	Description
1	GND
2	+12V
3	FAN_Detect

PCI1: 32-bit PCI Slot (11)

Pin	Description	Pin	Description
B1	-12V	A1	TRST
B2	TCK	A2	+12V
B3	GND	A3	TMS
B4	TDO	A4	TDI
B5	+5V	A5	+5V
B6	+5V	A6	INTA#
B7	INTB#	A7	INTC#
B8	INTD#	A8	+5V
B9	PRSTN1	A9	RSVD
B10	RSVD	A10	+5V
B11	PRSTN2	A11	RSVD
B12	GND	A12	GND
B13	GND	A13	GND
B14	RSVD	A14	3.3V_AUX
B15	GND	A15	RST#
B16	CLK	A16	+5V
B17	GND	A17	GNT#
B18	REQ#	A18	GND
B19	+5V	A19	PME#
B20	AD31	A20	AD30
B21	AD29	A21	+3.3V
B22	GND	A22	AD28
B23	AD27	A23	AD26
B24	AD25	A24	GND
B25	+3.3V	A25	AD24
B26	C/BE3#	A26	IDSEL
B27	AD23	A27	+3.3V
B28	GND	A28	AD22
B29	AD21	A29	AD20
B30	AD19	A30	GND
B31	+3.3V	A31	AD18
B32	AD17	A32	AD46
B33	C/BE2#	A33	+3.3V
B34	GND	A34	FRAME#

B35	IRDY#	A35	GND
B36	+3.3V	A36	TRDY#
B37	DEVSEL#	A37	GND
B38	GND	A38	STOP#
B39	LOCK#	A39	+3.3V
B40	PERR#	A40	SDONE
B41	+3.3V	A41	SBO#
B42	SERR#	A42	GND
B43	+3.3V	A43	PAR
B44	C/BE1#	A44	AD15
B45	AD14	A45	+3.3V
B46	GND	A46	AD13
B47	AD12	A47	AD11
B48	AD10	A48	GND
B49	GND	A49	AD9
B52	AD6	A52	C/BE0#
B53	AD7	A53	+3.3V
B54	+3.3V	A54	AD6
B55	AD5	A55	AD4
B56	AD3	A56	GND
B57	GND	A57	AD2
B58	AD1	A58	AD0
B59	+5V	A59	+5V
B60	ACK64#	A60	REQ64#
B61	+5V	A61	+5V
B62	+5V	A62	+5V

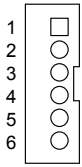
MINIPCI1: MiniPCI socket (12)



TV1: TV-out Connector (13)

The TV out connector is for output to a television.
Connector type: 2.00mm pitch 1x6-pin box wafer connector

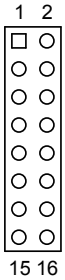
Composite Video			
1	CVBS	2	GND
3	Unused	4	GND
5	Unused	6	GND
S-Video			
1	Unused	2	GND
3	Luminance	4	GND
5	Chrominance	6	GND



JFRT1: Switches and Indicators (14)

It provides connectors for system indicators that provides light indication of the computer activities and switches to change the computer status.
Connector type: 2.54 mm pitch 2x8-pin headers

Pin	Description	Pin	Description
1	Power LED+	2	PWRBTN-
3	GND	4	PWRBTN+
5	GND	6	RESET+
7	HDD LED+	8	RESET-
9	HDD LED-	10	SPEAKER+
11	SMBCLK	12	SPEAKER+
13	SMBDATA	14	SPEAKER-
15	GND	16	SPEAKER-



PLED: Power LED Connector, pin 1-3.

This 2-pin connector connects to the case-mounted power LED. Power LED can be indicated when the CPU card is on or off. And keyboard lock can be used to disable the keyboard function so the PC will not respond by any input.

HLED: HDD LED Connector, pin 7-9.

This 2-pin connector connects to the case-mounted HDD LED to indicate hard disk activity.

SM Bus: SM Bus connector, pin 11, 13, 15.

PWRBTN: ATX soft power switch, pin 2-4.

This 2-pin connector connects to the case-mounted Power button.

RES: Reset Button, pin 6-8.

This 2-pin connector connects to the case-mounted reset switch and is used to reboot the system.

SPK: External Speaker, pin 10, 12, 14, 16.

This 4-pin connector connects to the case-mounted speaker.

CON1: RS-422/ 485 Connector (16)

Connector type: 2.00mm pitch 1x4-pin box wafer connector

Pin	RS-422	RS-485
1	TX+	Data+
2	TX-	Data-
3	RX+	N/C
4	RX-	N/C

INV1: LCD Inverter Connector (17)

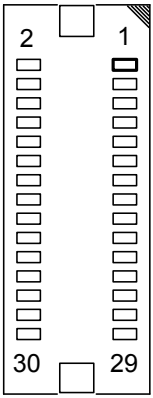
Connector type: 2.00mm pitch 1x5-pin box wafer connector.

Pin	Description
1	+12V
2	GND
3	Backlight on/off
4	N/C
5	GND

LVDS1: LVDS LCD Connector (18)

The LVDS connector supports 24-bit LVDS.
VDD could be selected by JVLCD1 in +5V or +3.3V.
Connector type: DF-13-30DP-1.25V

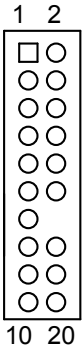
Pin	Description	Pin	Description
2	VDD	1	VDD
4	TX2CLK+	3	TX1CLK+
6	TX2CLK-	5	TX1CLK-
8	GND	7	GND
10	TX2D0+	9	TX1D0+
12	TX2D0-	11	TX1D0-
14	GND	13	GND
16	TX2D1+	15	TX1D1+
18	TX2D1-	17	TX1D1-
20	GND	19	GND
22	TX2D2+	21	TX1D2+
24	TX2D2-	23	TX1D2-
26	GND	25	GND
28	TX2D3+	27	TX1D3+
30	TX2D3-	29	TX1D3-



LPT1: Parallel Port Connector (20)

Connector type: 2.00 pitch 2x13-pin headers.
Shared with Floppy, LPT1 can be configured as a connector floppy disk drive interface through BIOS setup.

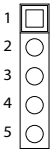
Pin	Description	Pin	Description
1	STROBE	14	AFD
2	PTD0	15	ERROR
3	PTD1	16	INIT
4	PTD2	17	SLIN
5	PTD3	18	GND
6	PTD4	19	GND
7	PTD5	20	GND
8	PTD6	21	GND
9	PTD7	22	GND
10	ACK	23	GND
11	BUSY	24	GND
12	PE	25	GND
13	SELECT	26	N/C



IR1: Infrared Connector (21)

Connector type: 2.54mm pitch 1x5-pin headers

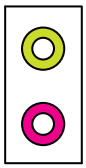
Pin	Description
1	+5V
2	N/C
3	IRRX
4	GND
5	IRTX



The IR connector can be configured to support wireless infrared module, user can transfer files to or from notebooks, PDA and printers.
Install infrared module onto IrDA connector and enable infrared function from BIOS setup and make sure to have correct orientation when you plug onto IrDA connector.

AUDIO1: HD AUDIO connector (22)

Connector type: double stacked audio jacks (Stereo ø3.50).



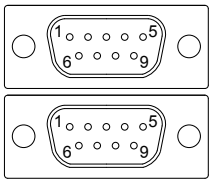
Green: Line Out

Pink: Mic

COM1, COM2: Serial Port Connectors (23)

Connector type: Double stacked D-Sub 9-pin male.

Pin	Description	Pin	Description
6	DSR#	1	DCD#
7	RTS#	2	RXD
8	CTS#	3	TXD
9	RI#	4	DTR#
		5	GND



LAN1, LAN2: RJ-45 + double stacked USB connectors (25, 26)

LAN1 and LAN2 support one Ethernet and two USB 2.0 connectors with 480Mb/s.

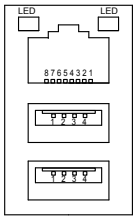
Connector type: RJ-45 + double stacked USB type A.

LAN (RJ-45)

Pin	Desc.	Pin	Desc.
1	MDI0+	5	MDI2+
2	MDI0-	6	MDI2-
3	MDI1+	7	MDI3+
4	MDI1-	8	MDI3-

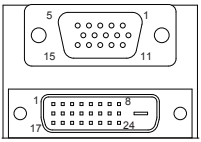
USB (USB type A connector)

Pin	Desc.
1	+5V
2	USB-
3	USB+
4	GND



VGA1: Analog RGB & DVI-D Connectors (27)

Connector type: VGA: D-Sub 15-pin female.
DVI-D: DVI-D female.



Analog RGB

DVI-D

Analog RGB Connector

Pin	Description	Pin	Description	Pin	Description
1	RED	6	GND	11	N/C
2	GREEN	7	GND	12	VDDAT
3	BLUE	8	GND	13	HSYNC
4	N/C	9	+5V	14	VSYNC
5	GND	10	GND	15	VDCLK

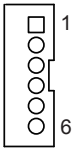
DVI-D Connector

Pin	Description	Pin	Description	Pin	Description
1	TMDS Data 2-	9	TMDS Data 1-	17	TMDS Data 0-
2	TMDS Data 2+	10	TMDS Data 1+	18	TMDS Data 0+
3	TMDS Data 2/4 shield	11	TMDS Data 1/3 shield	19	TMDS Data 0/5 shield
4	TMDS Data 4-	12	TMDS Data 3-	20	TMDS Data 5-
5	TMDS Data 4+	13	TMDS Data 3+	21	TMDS Data 5+
6	DDC clock	14	+5V	22	TMDS Data clock shield
7	DDC data	15	GND	23	TMDS clock+
8	Analog vertical sync	16	Hot plug detect	24	TMDS clock-

EKB1: External Keyboard & Mouse Connector (28)

Connector type: 2.54mm pitch 1x6-pin box wafer connector

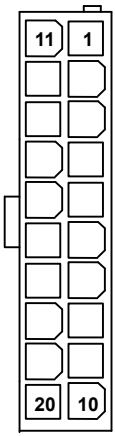
Pin	Description
1	KB Data
2	GND
3	MS Data
4	KB Clock
5	+5V
6	MS Clock



PW1: ATX Power Supply Connector (29)

The ATX power supply has a single lead connector with a clip on one side of the plastic housing. There is only one way to plug the lead into the ATX power connector. Press the lead connector down until the clip snaps into place and secures the lead onto the connector.

Pin	Description	Pin	Description
11	+3.3V	1	+3.3V
12	-12V	2	+3.3V
13	GND	3	GND
14	PS-ON	4	+5V
15	GND	5	GND
16	GND	6	+5V
17	GND	7	GND
18	-5V	8	PW-OK
19	+5V	9	+5VSB
20	+5V	10	+12V



Warning

Incorrect installation of the power supply could result in serious damage to the mainboard and connected peripherals. Make sure the power supply is unplugged from the AC outlet before connecting the leads from the power supply.

2.3 The Installation Paths of CD Driver

Windows 2000 & XP

Driver	Path
AUDIO	\AUDIO\REALTEK_HD\WINDOWS_R198
CHIPSET	\CHIPSET\INTEL\INF 9
LAN	\ETHERNET\REALTEK\8111B_WIN5698
VGA	\GRAPHICS\INTEL_2K_XP_32\1436 \GRAPHICS\INTEL_2K_XP_64\1436

Windows Vista

Driver	Path
AUDIO	\AUDIO\REALTEK_HD\Vista_R198
CHIPSET	\CHIPSET\INTEL\INF 9
LAN	\ETHERNET\REALTEK\Vista_6206_0619
VGA	\GRAPHICS\INTEL_VISTA_32\1598 \GRAPHICS\INTEL_VISTA_64\1598

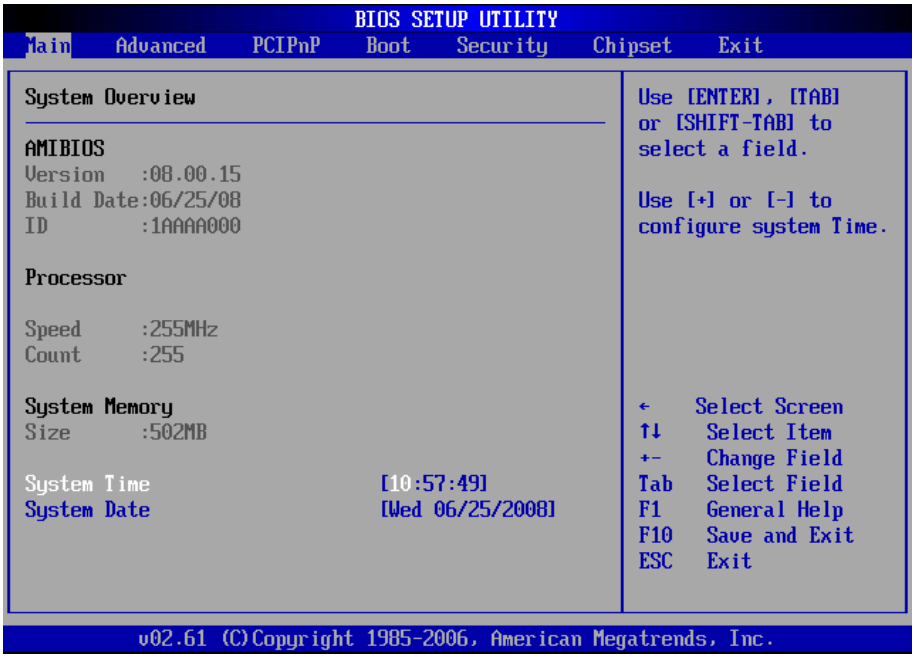
Chapter 3

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 allows you to select several configuration options. 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

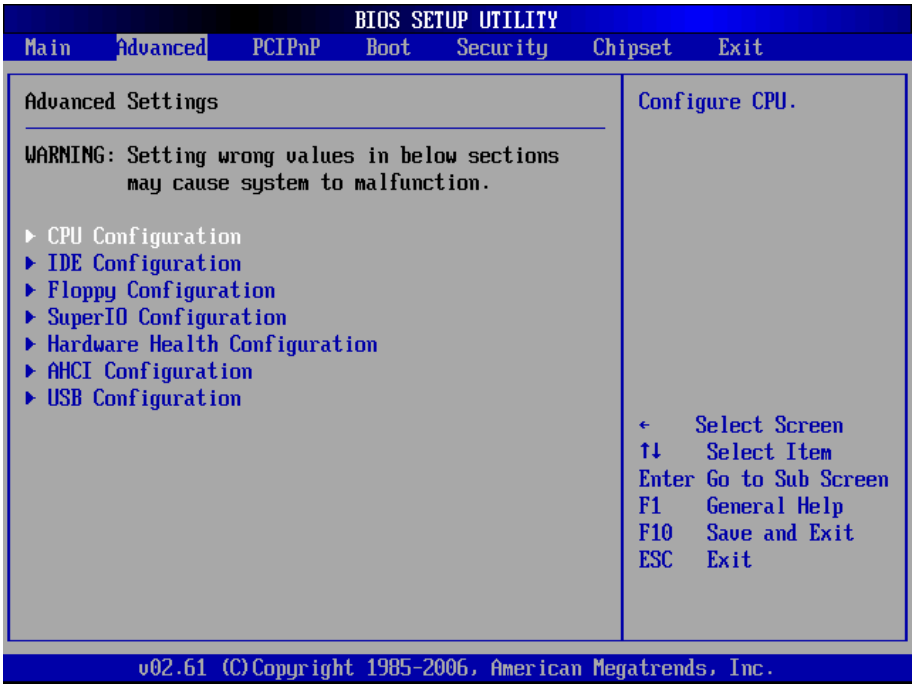
System Date

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

The date format is:

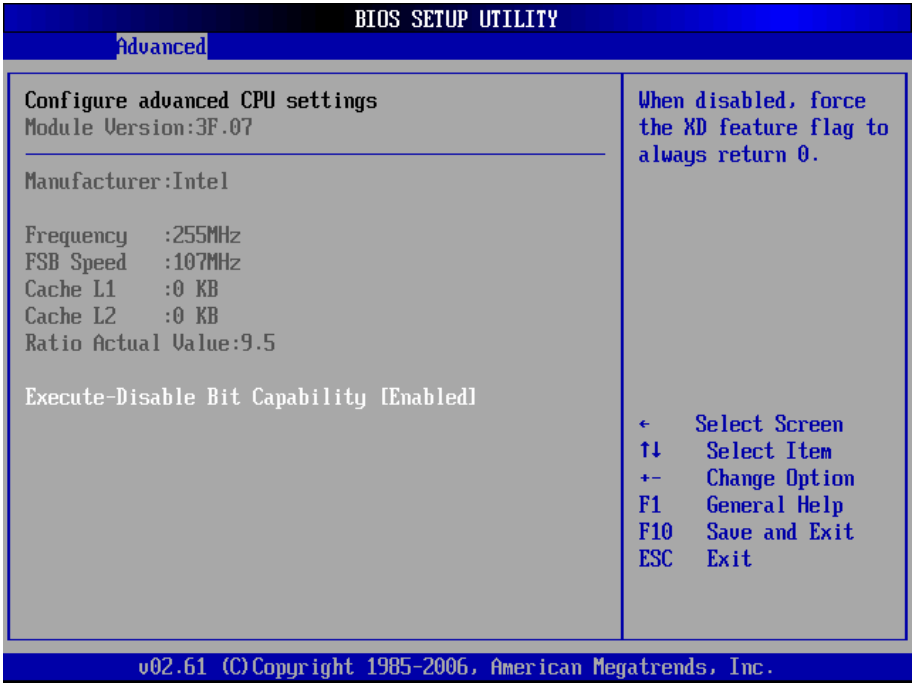
Day : Sun to Sat
Month : 1 to 12
Date : 1 to 31
Year : 1999 to 2099

3.2 Advanced Settings



3.2.1 CPU Configuration

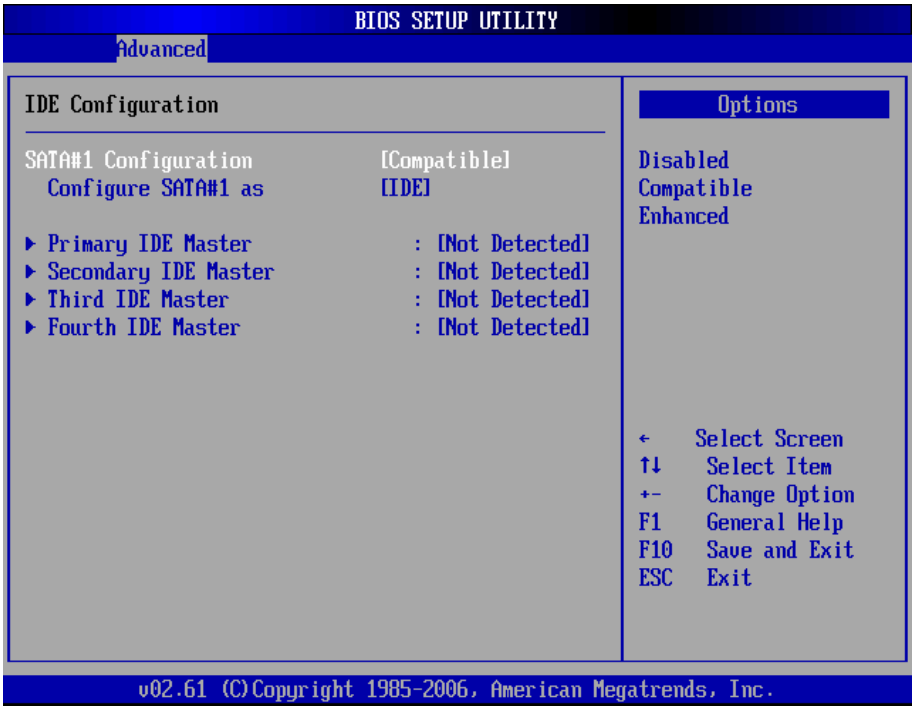
The CPU Configuration setup screen varies depending on the installed processor.



Execute Disable Bit

When disabled, force the SD feature flag to always return 0.

3.2.2 IDE Configuration



SATA#1 Configuration

Enable - Enable SATA configuration.
Disabled - Disable SATA configuration

Configure SATA#1 as

This BIOS feature controls the SATA controller's operating mode.
There are two available modes - IDE and RAID. When set to:

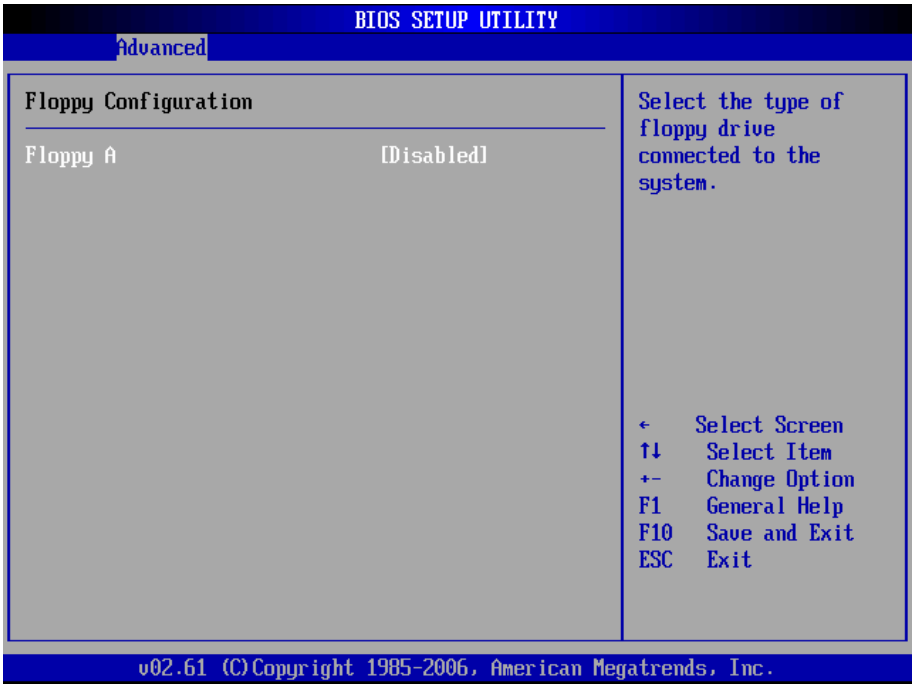
RAID - the SATA controller ***enables*** its RAID and AHCI functions when the computer boots up.

IDE - the SATA controller ***disables*** its RAID and AHCI functions when the computer boots up.

Primary/Secondary/Third/Fourth IDE Master/Slave

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

3.2.3 Floppy Configuration

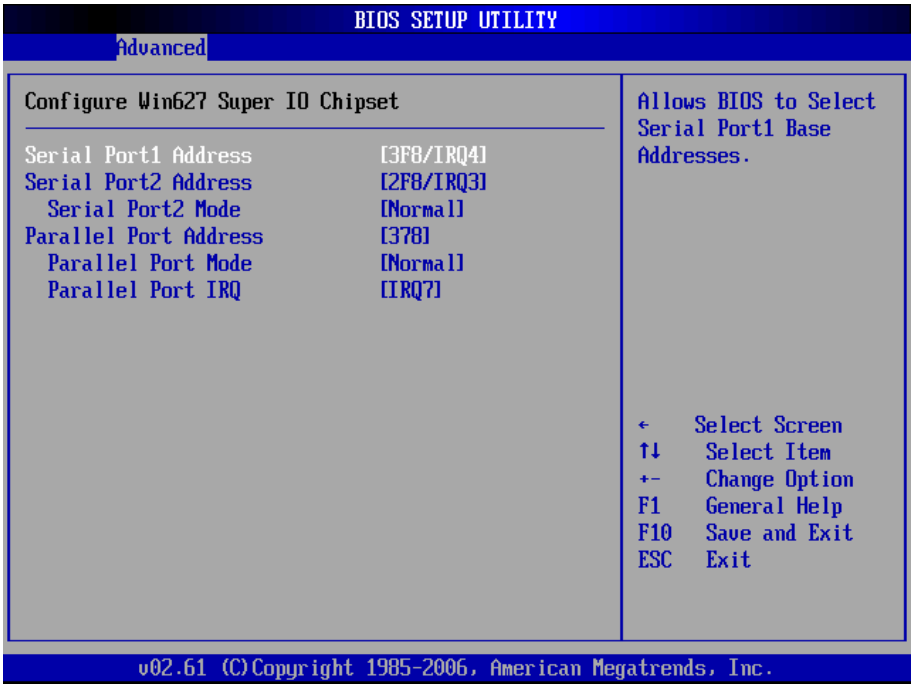


Select the type of floppy disk drive installed in your system.
The choice:

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

Floppy 3: The choice are Disabled or Enabled.

3.2.4 Super IO Configuration



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 your hardware and software both support one of the other available modes.
The choice:

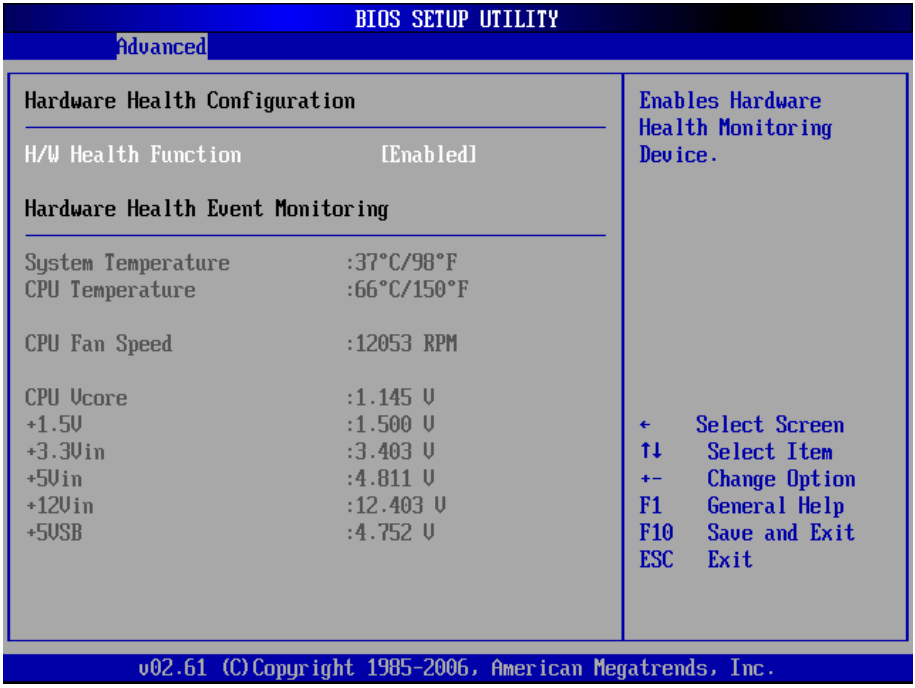
- SPP
- EPP
- ECP
- ECP + EPP
- Normal

Parallel Port IRQ

Select an interrupt for the parallel port.
The choice:

- IRQ5
- IRQ7

3.2.5 Hardware Health Configuration



System/ CPU Temperature

Show you the current System / CPU fan temperature.

CPU Fan Speed

Show you the current CPU Fan operating speed.

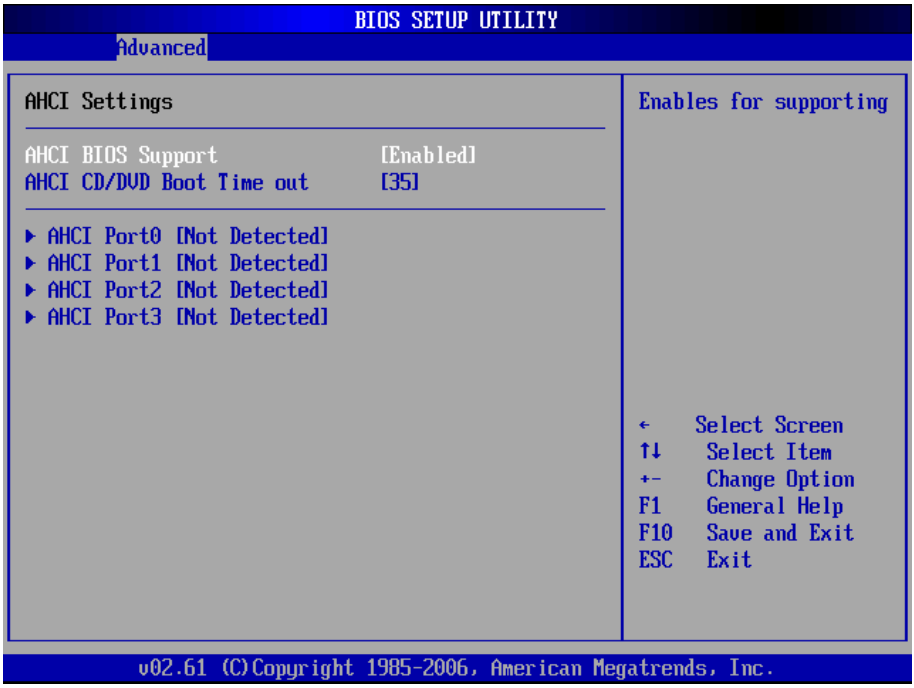
Vcore

Show you the voltage level of CPU (Vcore).

+1.5V / +3.3Vin / +5Vin / +12Vin / 5VSB

Show you the voltage level of the +1.5V, +3.3Vin, +5Vin, +12Vin and +5V standby.

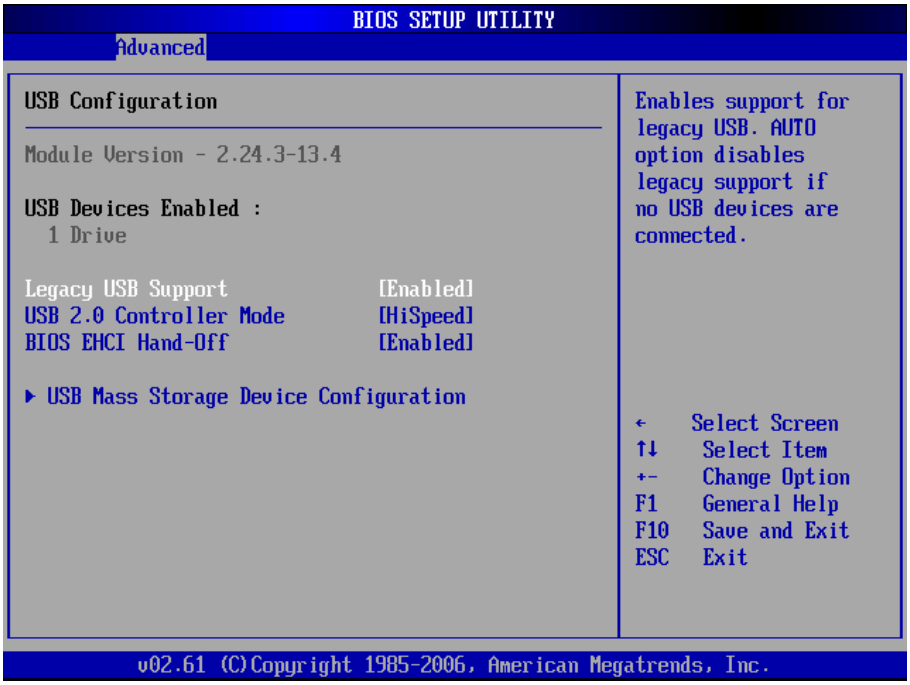
3.2.6 AHCI Configuration



AHCI Port 0 / Port 1 / Port 2/ Port 3

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

3.2.7 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

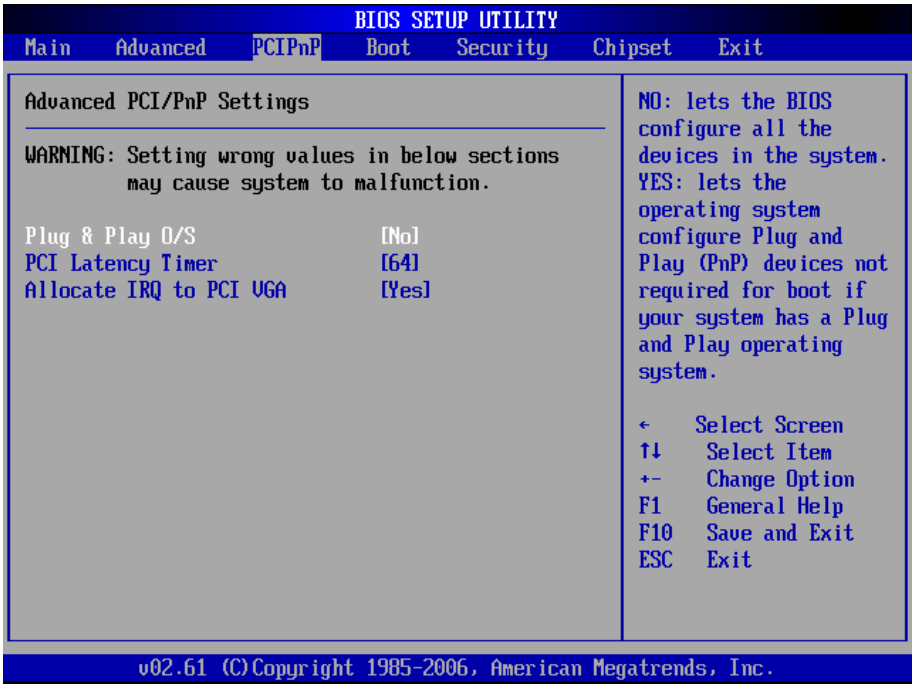
Configures the USB 2.0 controller in High Speed (480Mbps) or Full Speed (12MBPS).

BIOS EHCI Hand-Off

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

USB Mass Storage Device Configuration

3.3 Advanced PCI/PnP Settings



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.

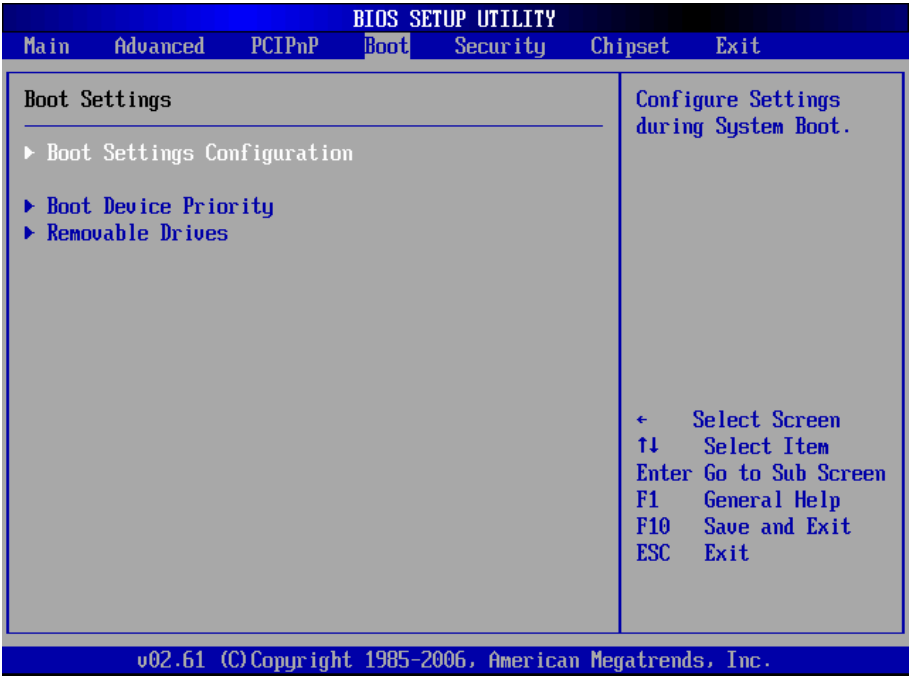
PCI Latency Timer

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.

3.4 Boot Settings



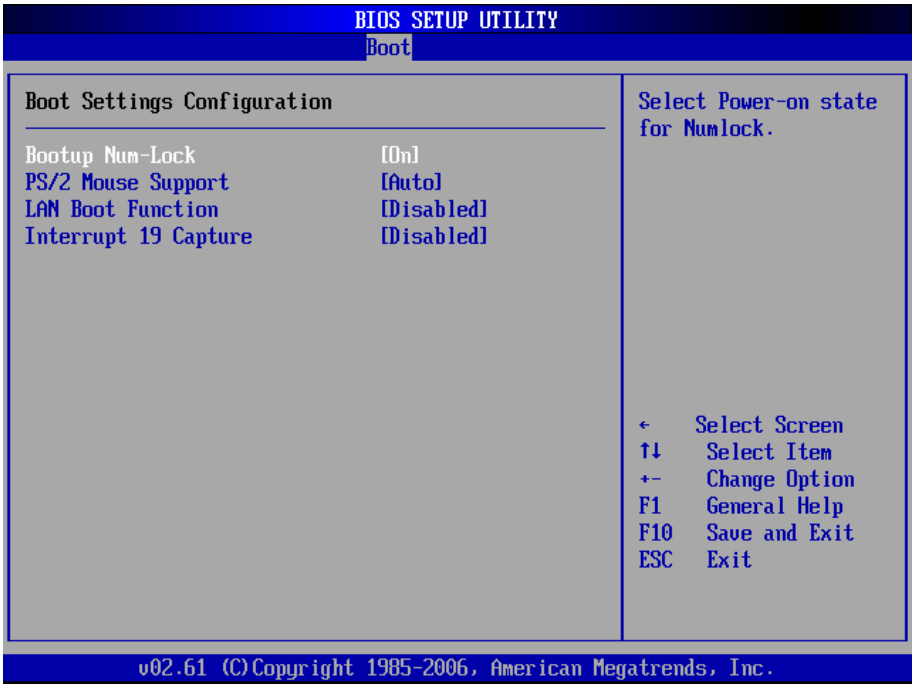
Boot Device Priority

Press Enter and it shows Bootable add-in devices.

Removable Drives

Press Enter and it shows Bootable and Removable drives.

3.4.1 Boot Settings Configuration



Bootup Num-Lock

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

PS/2 Mouse Support

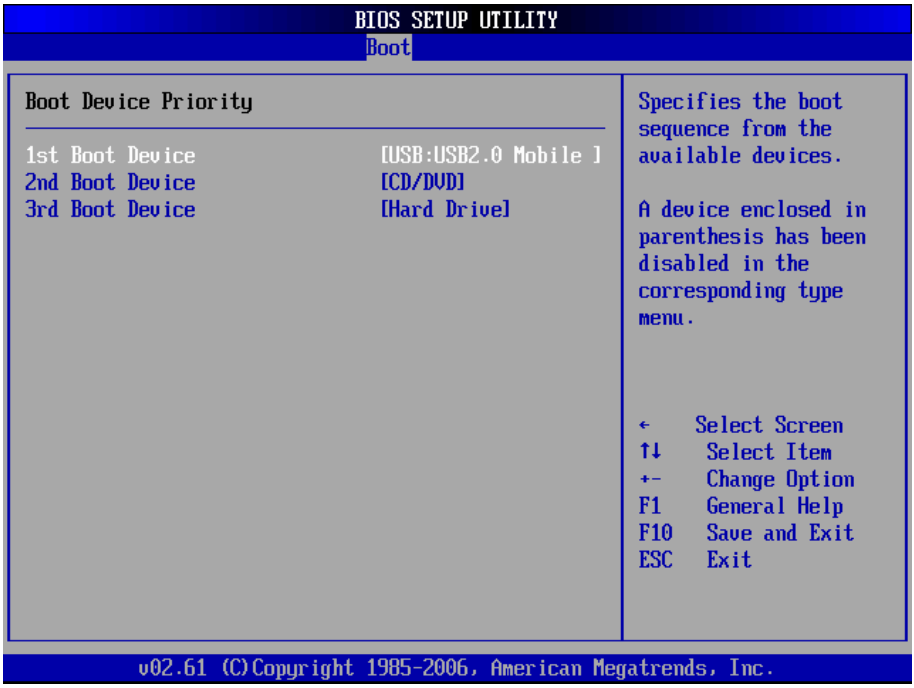
Interrupt 19 capture

Enabled: Allows option ROMs to trap interrupt 19. This is required by some PCI cards that provide a ROM based setup utility.

LAN Boot Function

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

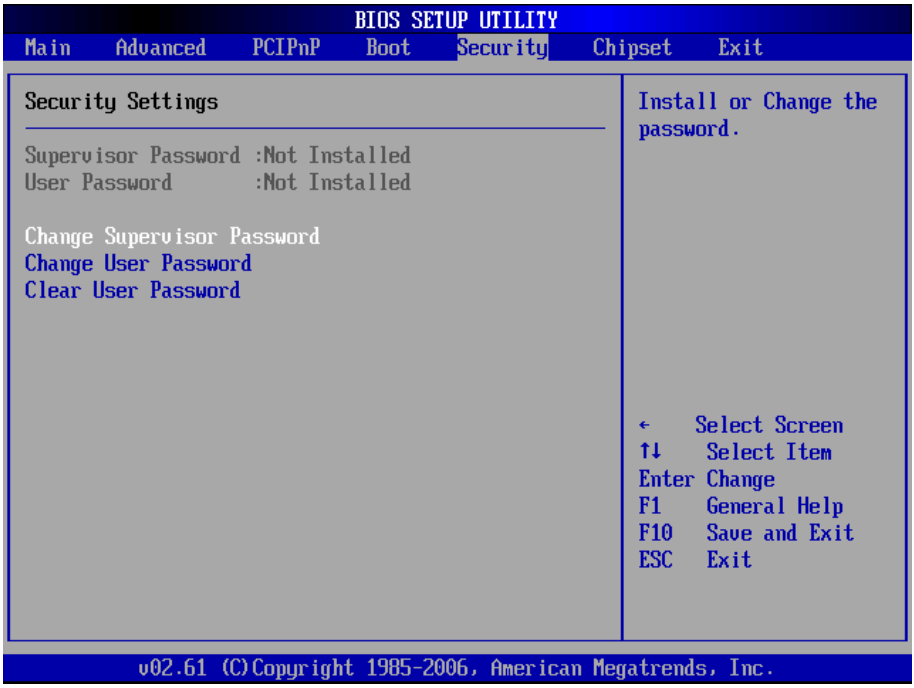
3.4.2 Boot device Priority



1st/ 2nd/ 3rd Boot Device

Specifies the boot sequence form the available devices.
A device enclosed in parenthesis has been disabled in the corresponding type menu.

3.5 Security



Supervisor Password & User Password

You can set either supervisor or user password, or both of them. 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.

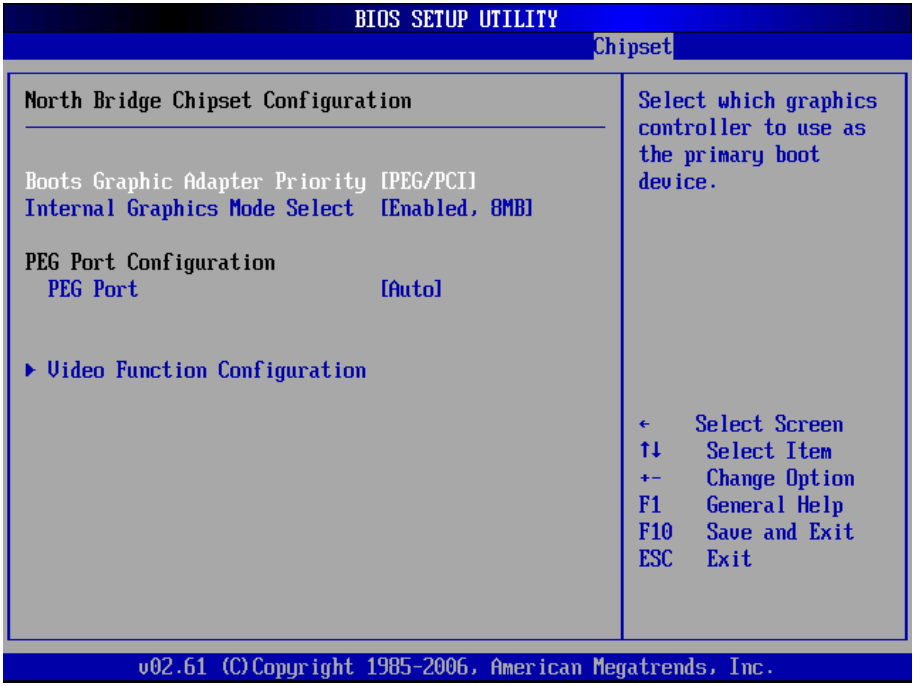
When a password has been enabled, you will be prompted to enter it every time you try to enter Setup. This prevents an unauthorized person from changing any part of your system configuration.

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.

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

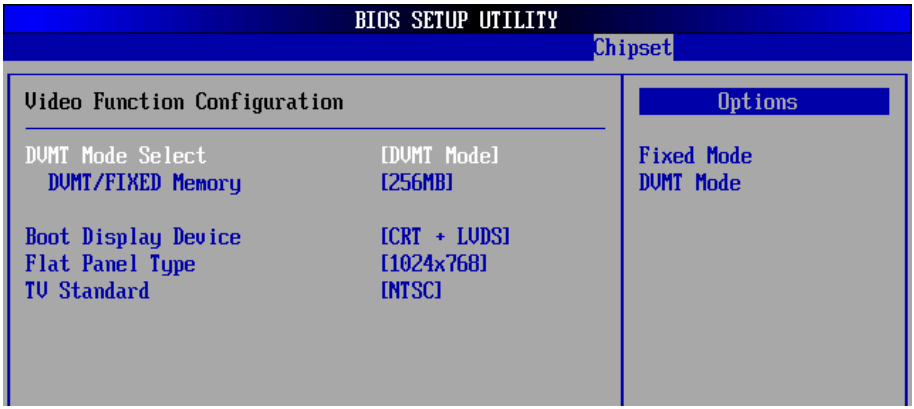
Select the amount of system memory used by the Internal graphics device.

PEG Port Configuration

This item allows you to control the PEG or on-chip VGA.

The Choice: Auto, Disabled.

Video Function Configuration



DVMT Mode

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

Boot Display Device

Setting: CRT, LVDS, CRT+ LVDS (Default).

Flat Panel Type

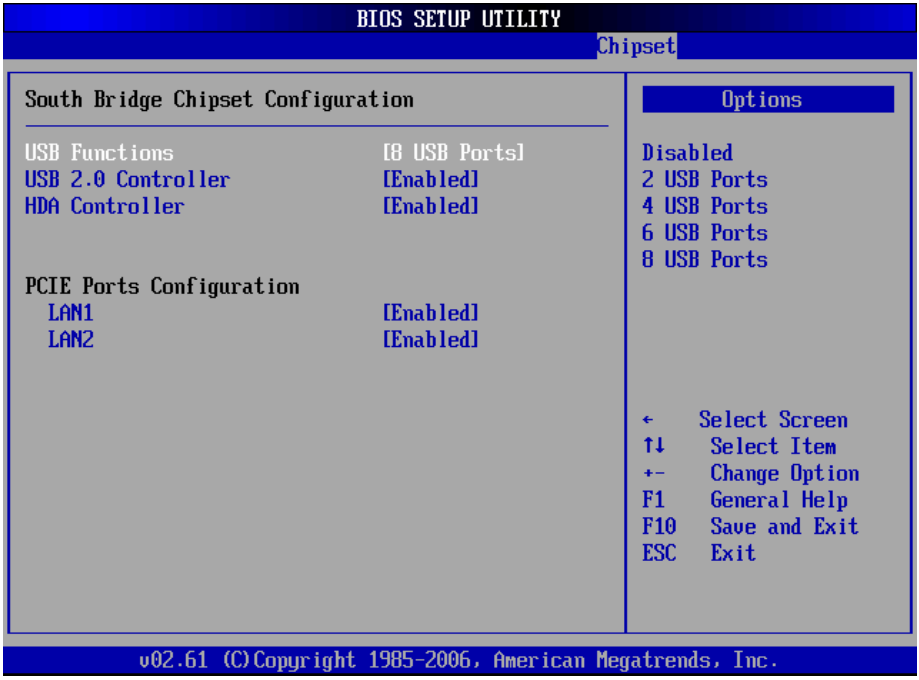
It allows you to select the Flat Panel type as below ---

Setting:
640x480
800x600
1024x768 (Default)
1280x1024
1400x1050
1400x1050
1600x1200
1280x768
1680x1050
1920x1200

TV Standard

Setting: NTSC (Default), PAL.

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

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.

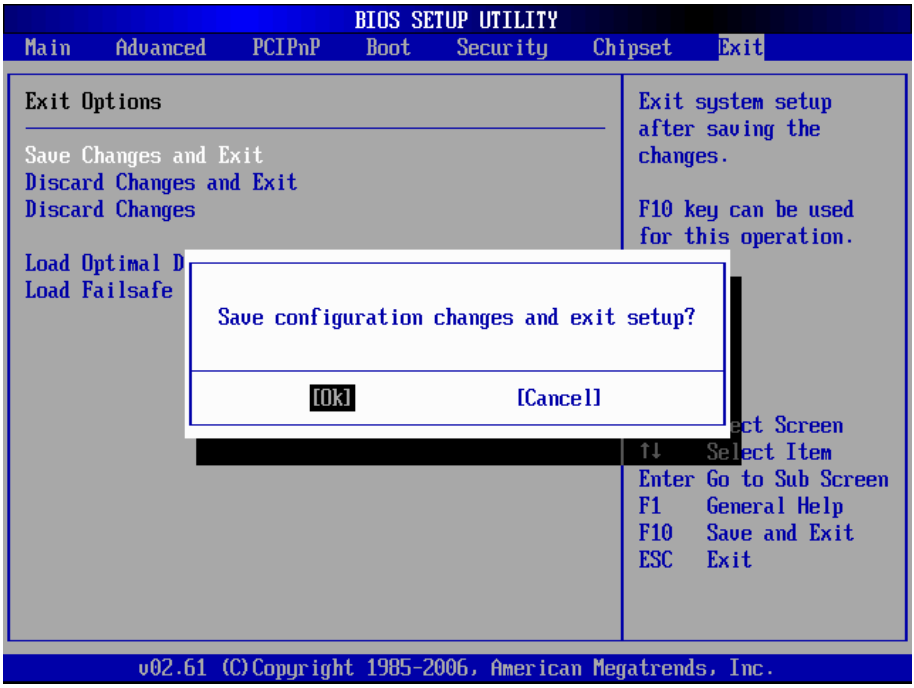
Onboard Giga LAN1 / LAN2

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

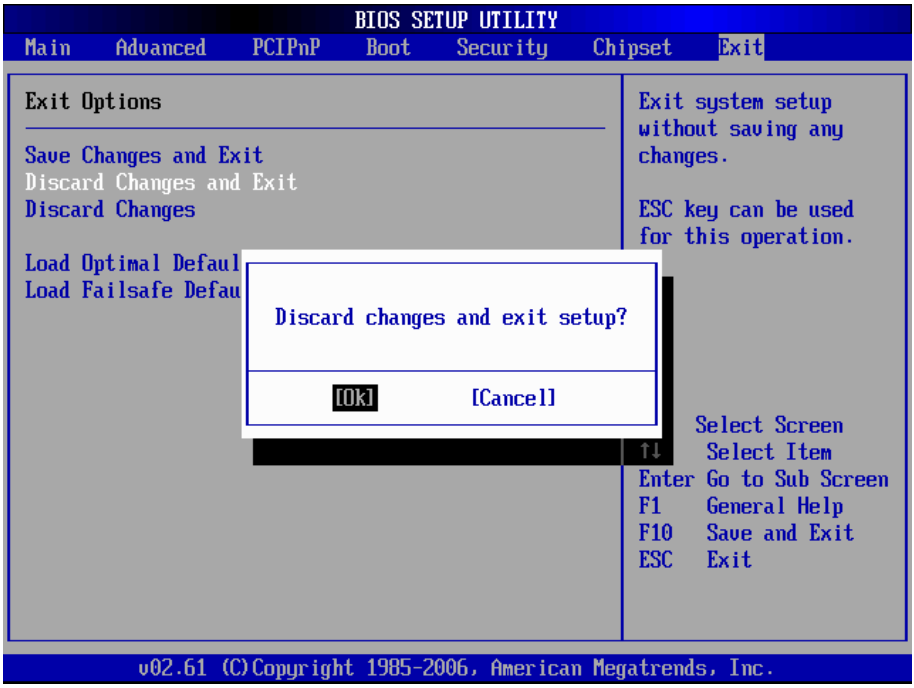


Pressing <Enter> on this item asks for confirmation:

Save configuration changes and exit setup?

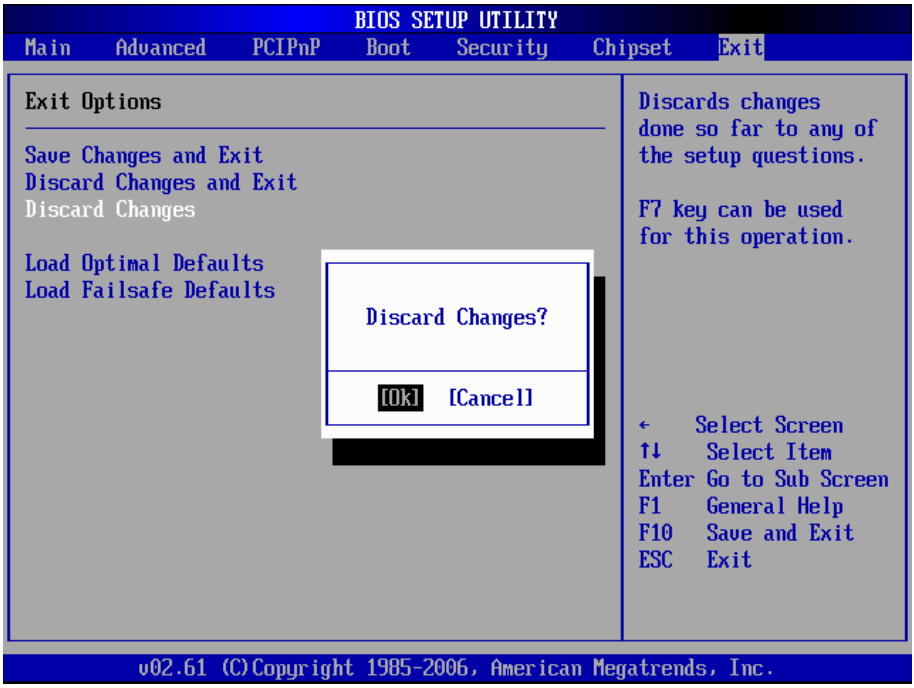
Pressing <OK> stores the selection made in the menus in CMOS - a special section of memory that stays on after you turn your system off. The next time you boot your computer, the BIOS configures your system according to the Setup selections stored in CMOS. After saving the values the system is restarted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

Discard Changes and Exit



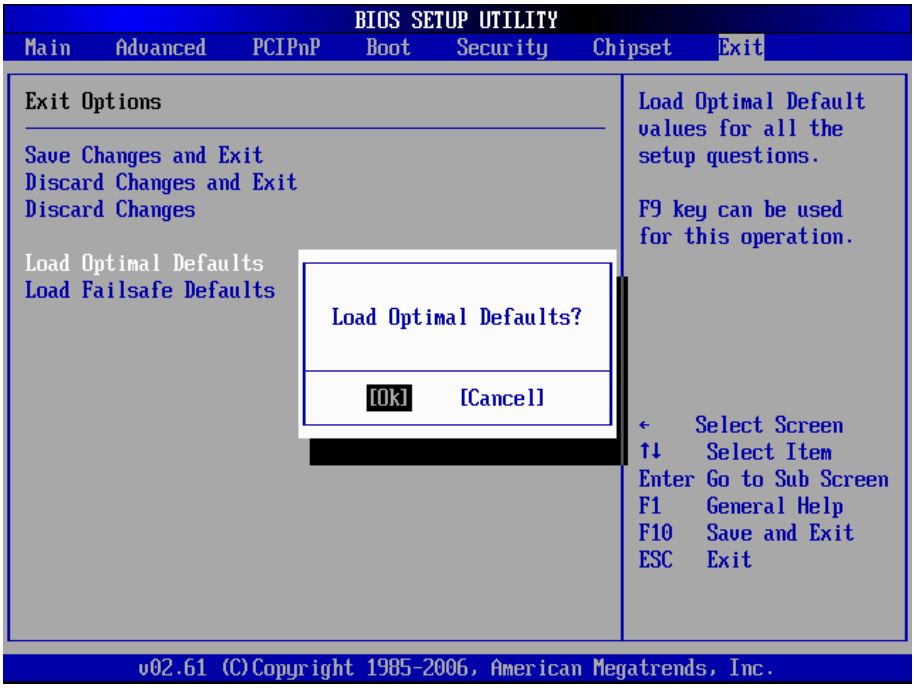
Exit system setup without saving any changes.
<ESC> key can be used for this operation.

Discard Changes



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

Load Optimal Defaults



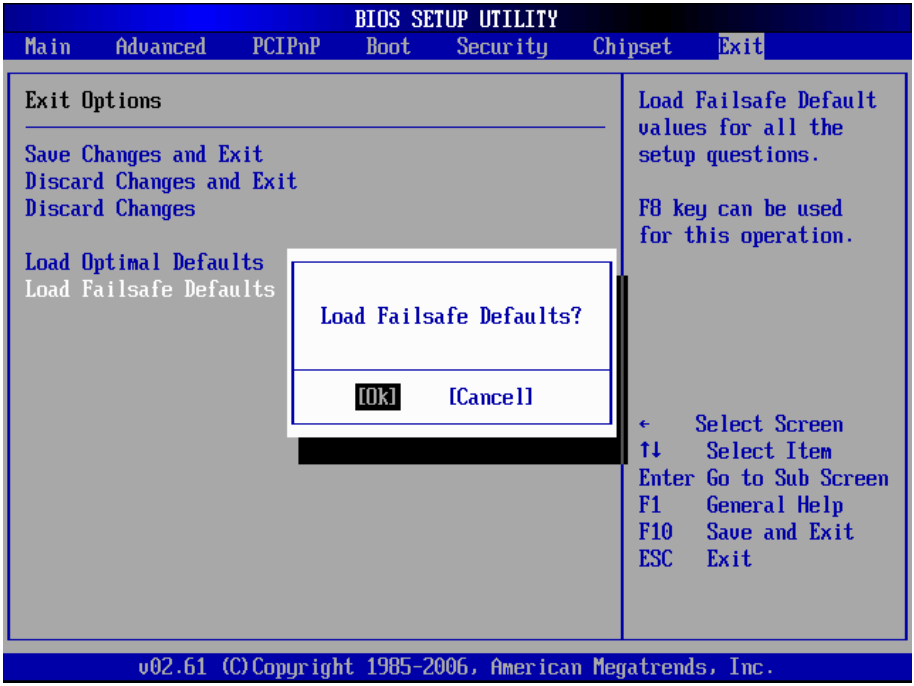
When you press <Enter> on this item, you get a confirmation dialog box with a message:

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 reseat 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 cirtical 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 deisabled. Perfrom keyboard controller BAT test. Save power-on CPUID value in scretch 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 not executed, start memory refresh and do memory sizing in Boot block code. Do additional chipset initialization. Re-enabled 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. Copies compressed boot block code to memory in right segments. Copies BIOS from ROM to RAM for faster access. Perfroms main BIOS checksum and updates recovery status accordingly.

D6	Both key sequence and OEM specific method is checked to determine if BIOS recovery is forced. If BIOS recovery is necessary, control flows tocheckpoint E0. Seed *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 in memory.
D8	The Tuntime 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 POS (ExecutePOSTKernel). See *POST Code Checkpoints* section of document for more information.
DC	System is saking 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 be 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. L2 cache is enabled.
E9	Set up floppy controller and data. Attempt to red 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 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 theuser 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/Disalbe NMI as selected.
90	Initialization of system management interrupt by invoking all handlers.
A1	Lian-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 userinput 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 buses. 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); 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.

Chapter 4

Appendix

4.1 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.

Address	Device Description
00000000 - 0000000F	DMA Controller
00000080 - 0000009F	DMA Controller
000000C0 - 000000DF	DMA Controller
00000020 - 00000021	Programmable Interrupt Controller
000000A0 - 000000A1	Programmable Interrupt Controller
00000040 - 00000043	System Timer
00000044 - 00000047	System Timer
00000060 - 00000064	Keyboard Controller
00000070 - 00000073	System CMOS/Real Time Clock
000000F0 - 000000FF	Math Co-processor
000001F0 - 000001F7	Primary IDE
00000274 - 00000277	ISAPNP Read Data Port
00000279, 00000A79	ISAPNP Configuration
000002E8 - 000002E7	Communications Port (COM4, If use)
000002F8 - 000002FF	Communications Port (COM2, If use)
00000378 - 0000037A	Parallel Port (If use)
000003B0 - 000003BF	MDA/MGA
000003C0 - 000003CF	EGA/VGA
000003D4 - 000003D9	CGA CRT register
000003E8 - 000003EF	Communications Port (COM3, If use)
000003F0 - 000003F5	Floppy Diskette
000003F6 - 000003F6	Primary IDE
000003F7 - 000003F7	Communications Port (COM1, If use)
00000400 - 0000041F	South Bridge SMB
00000480 - 0000041F	South Bridge GPIO
000004D0 - 000004D1	IRQ Edge/Level Control Ports

00000800 - 0000087F	ACPI
00000A00 - 00000A07	PME
00000A10 - 00000A17	Hardware Monitor
00000A20 - 00000A27	Digital I/O
00000A30 - 00000A37	SFIF
00000CF8	PCI Configuration Address
00000CFC	PCI Configuration Data
00004700 - 0000470B	TPM (If use)

4.2 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 0	System Timer
IRQ 1	Keyboard Controller
IRQ 2	VGA and Link to Secondary PIC
IRQ 3	Communications Port (COM2/ COM4)
IRQ 4	Communications Port (COM1/ COM3)
IRQ 5	PCI Device
IRQ 6	Standard Floppy Disk Controller
IRQ 7	Parallel Port
IRQ 8	System CMOS/real time clock
IRQ 9	Microsoft ACPI-Compliant System
IRQ 10	PCI Device
IRQ 11	PCI Device
IRQ 12	PS/2 Compatible Mouse
IRQ 13	FPU Exception
IRQ 14	PCI Device
IRQ 15	PCI Device

4.3 BIOS memory mapping

Address	Device Description
00000h - 9FFFFh	DOS Kernel Area
A0000h, BFFFFh	EGA and VGA Video Buffer (128KB)
C00000h - CFFFFh	EGA/VGA ROM
D0000h - DFFFFh	Adaptor ROM
E00000h - FFFFFh	System BIOS
EFD40000h - FED44FFFFh	TPM (If use)

4.4 Watchdog Timer (WDT) Setting

WDT is widely used for industry application to monitoring the activity of CPU. Application software depends on its requirement to trigger WDT with adequate timer setting. Before WDT time out, the functional normal system will reload the WDT. The WDT never time out for a normal system. Then, WDT will time out and reset the system automatically to avoid abnormal operation.

This board supports 255 levels watchdog timer by software programming. Below are the source codes written in assembly & C, please take them for WDT application examples.

Assembly Code

```
;-- Initial W83627hf --
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 87h
        out     DX, AX          ;
        out     DX, AX          ; initial W83627HF start
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 2Bh
        out     DX, AL          ; Select CR2B
        mov     AL, 00h
        inc     DX
        out     DX, AL          ; Set CR2B bit 4=0, PIN89=WDTO
;--
```

```
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 07h
        out     DX, AL          ; Point to Logical Device Selector
        mov     AL, 08h
        inc     DX
        out     DX, AL          ; Select Logical Device 8
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 30h
        out     DX, AL          ; select CR30
        mov     AL, 01h
        inc     DX
        out     DX, AL          ; update CR30 to 01h
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 0F0h
        out     DX, AL          ; select CRF0
        mov     AL, 00h
        inc     DX
        out     DX, AL          ; set CRF0=00h, output
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 0F5h
        out     DX, AL          ; select CRF5, WDT Timer unit
        mov     AL, 00h         ; bit2 =0 ->second ; bit2 =1 -> minute
        inc     DX
        out     DX, AL          ; update CRF5 bit2 to 00h
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 0F6h
        out     DX, AL          ; select CRF6, WDT Timer
        mov     AL, 05h
        inc     DX
        out     DX, AL          ; update CRF6 to 5 unit
;---
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, AAh
        out     DX, AX
;-- end
```

C language Code

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

/*-----     routing, sub-routing -----*/

void main()

{
          outportb(0x2e, 0x87);       /* initial IO port twice */
          outportb(0x2e, 0x87);

          outportb(0x2e, 0x2B);       /* select CR2B */
          outportb(0x2e+1, 0x00);     /* update CR2B bit4 to 00h */
                                      /* Set PIN89 as WDTO */

          outportb(0x2e, 0x07);       /* point to logical device selector */
          outportb(0x2e+1, 0x08);     /* select logical device 8 */
          outportb(0x2e, 0x30);       /* select CR30 */
          outportb(0x2e+1, 0x01);     /* update CR30 to 01h */
          outportb(0x2e, 0xf0);       /* select CRF0 */
          outportb(0x2e+1, 0x00);     /* update CRF0 to 00h */
          outportb(0x2e, 0xf5);       /* select CRF5 to set timer unit */
          outportb(0x2e+1, 0x00);     /* update CRF5 bit2, 0:sec; 1:Min. */
          outportb(0x2e, 0xF6);       /* select CRF6 */
          outportb(0x2e+1, 0x05);     /* update CRF6 to 05h (5 sec) */

          outportb(0x2e, 0xAA);       /* stop program W83627HF, Exit */
}
```

4.5 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.