Wide Operating Temperature

+85ºC

-40ºC

COM-746E

Wide Range Temperature
COM Express Type 2 CPU Module

User's Manual

Version 1.1

+85ºC

-40ºC

2011.10

This page is intentionally left blank.

Contents

Chapter 1

Introduction

1.1 Copyright Notice

All Rights Reserved.

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.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

1.2 Declaration of Conformity

CE

The CE symbol on your product indicates that it is in compliance with the directives of the Union European (EU). A Certificate of Compliance is available by contacting Technical Support.

This product has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from ARBOR. Please contact your local supplier for ordering information.

This product has passed the CE test for environmental specifications. 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.

Warning

This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

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.

NOTE:

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

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%)).

SVHC / REACH

To minimize the environmental impact and take more responsibility to the earth we live, Arbor hereby confirms all products comply with the restriction of SVHC (Substances of Very High Concern) in (EC) 1907/2006 (REACH --Registration, Evaluation, Authorization, and Restriction of Chemicals) regulated by the European Union.

All substances listed in SVHC < 0.1 % by weight (1000 ppm)

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.

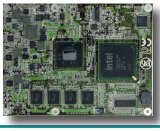
Vendors disclaim all other warranties, either expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the hardware, the accompanying product's manual(s) and written materials, and any accompanying hardware. This limited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

Packing List

Before you begin installing your single board, please make sure that the following materials have been shipped:



1 x COM-746E 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

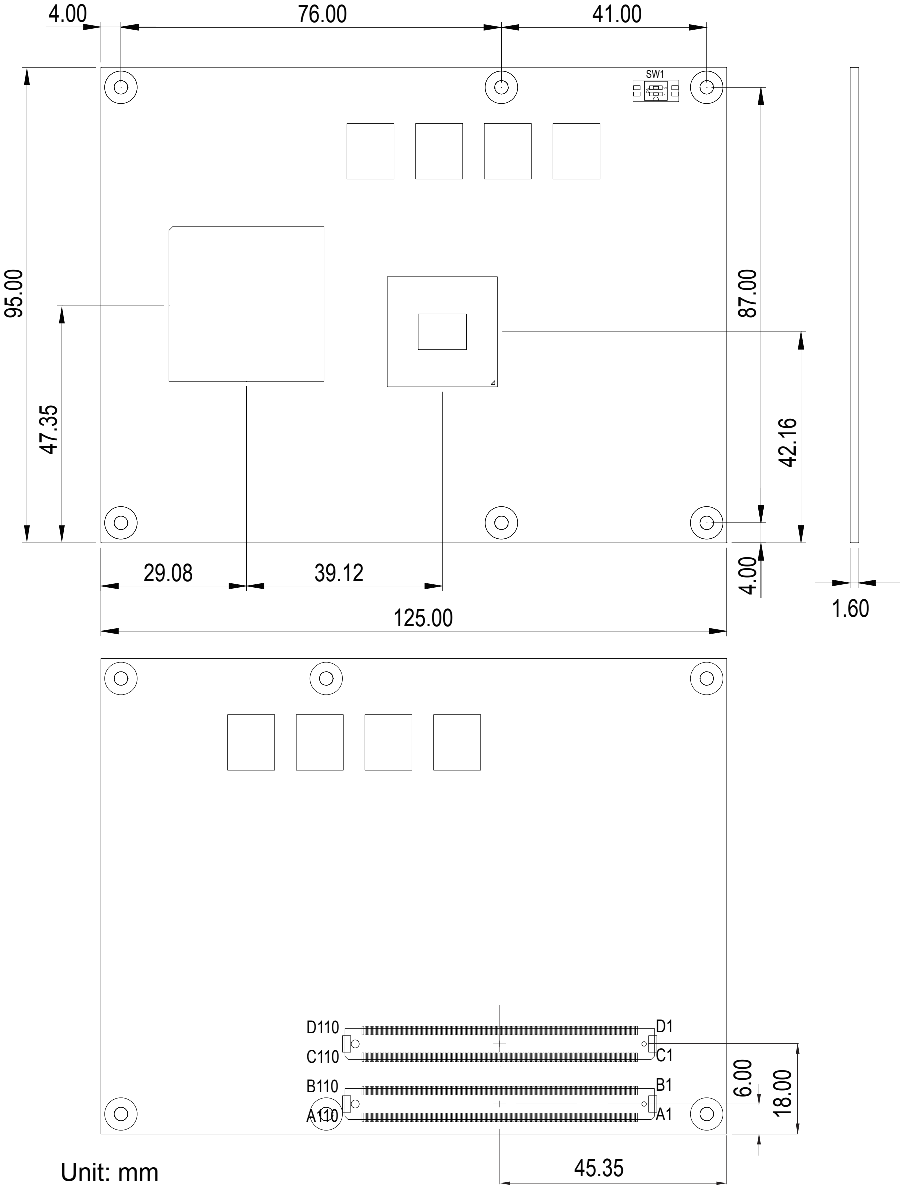
Ordering Information

COM-746E	Intel® Atom™ N455 COM Exress CPU module
HS-0746-F1	Heat Spreader (95 x 125 x 18mm)
PBE-1700 R1.1	COM Express Type 2 evaluation board in ATX form factor
CBK-04-1700-00	Cable kit 1 x SATA cable 1 x COM port cable 1 x FDD cable 1 x IDE cable

1.10 Specifications

Form Factor	COM Express Type 2 CPU Module
CPU	Intel Atom™ N455 at 1.66GHz processor
Chipset	Intel® ICH8M
System Memory	Soldered onboard 2GB DDR3 SDRAM
VGA/ LCD Controller	Intel® Graphics Media Accelerator 3150 graphics core w/ Analog RGB/ Single Channel 18-bit LVDS (Dual independent displays)
Ethernet controller	1 x Intel 82574L PCIe Gigabit Ethernet
BIOS	AMI PnP Flash BIOS
Serial ATA	3 x Serial ATA ports w/ 300MB/s HDD transfer rate
IDE Interface	1 x IDE (Ultra DMA 100/66/33), supports 2 IDE devices
Universal Serial Bus	8 x USB 2.0 ports
LCD	Single Channel 18-bit LVDS
Expansion Interface	5 x PCIe x1 lanes 4 x PCI masters LPC interface
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



Unit: mm

This page is intentionally left blank.

Chapter 2

Installation

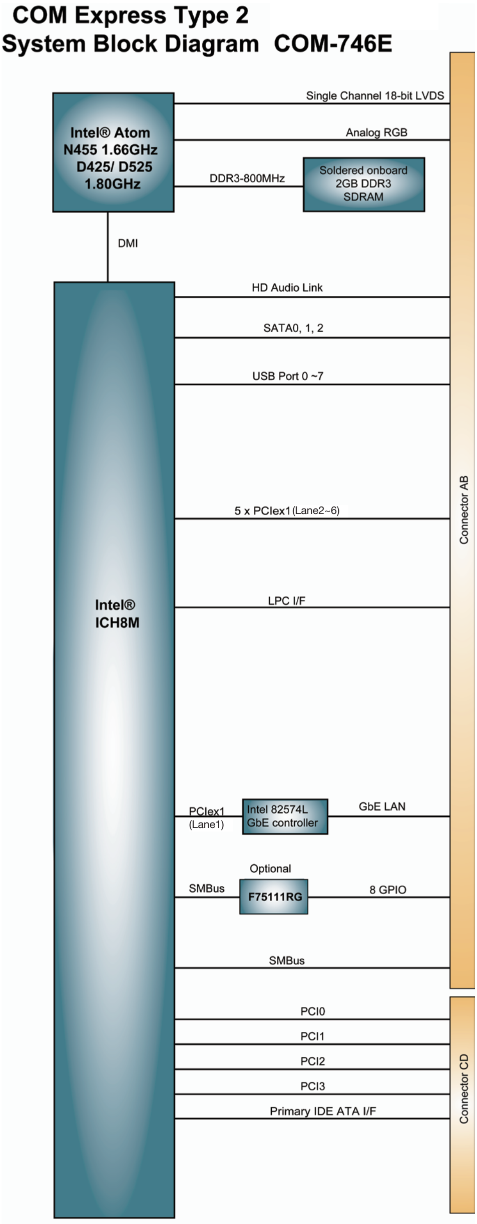
2.1 What is “COM Express” ?

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:

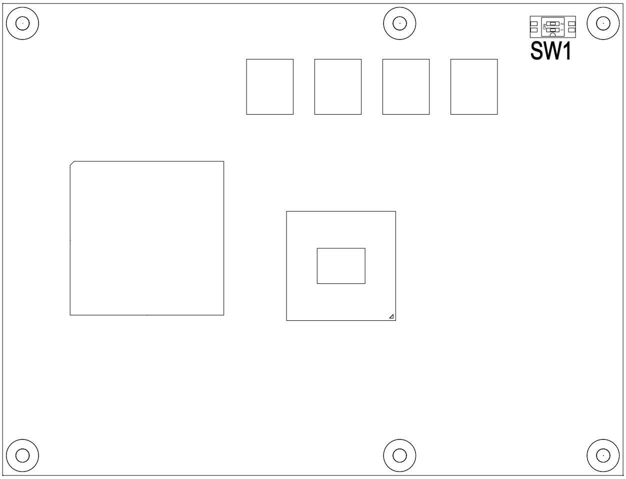
- Retail & Advertising
- Medical
- Test & Measurement
- Gaming & Entertainment
- Industrial & Automation
- Military & Government
- Security

2.2 Block Diagram

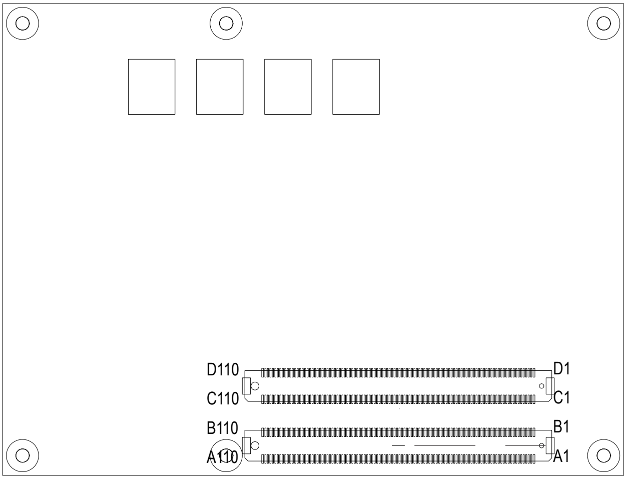


2.3 Jumpers and Connectors

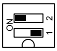
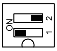
(Top)



(Bottom)



SW1: AT/ATX Power mode selection

	Power Mode Selection
	AT Mode
	ATX Mode (Default)

2.4 COM Express AB Connector (bottom side)

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	N/C	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	SATA1_TX+	SATA0_TX+	A16
B17	SATA1_TX-	SATA0_TX-	A17
B18	N/C	N/C	A18
B19	SATA1_RX+	SATA0_RX+	A19
B20	SATA1_RX-	SATA0_RX-	A20
B21	GND	GND	A21
B22	N/C	SATA2_TX+	A22
B23	N/C	SATA2_TX-	A23
B24	PWR_OK	N/C	A24
B25	N/C	SATA2_RX+	A25
B26	N/C	SATA2_RX-	A26
B27	WDT	N/C	A27
B28	N/C	ATA_ACT#	A28
B29	N/C	AC_SYNC	A29
B30	AC_SDIN0	AC_RST#	A30
B31	GND	GND	A31
B32	SPKR	AC_BITCLK	A32
B33	N/C	AC_SDOUT	A33
B34	N/C	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	N/C	N/C	A54
B55	PCIE_RX4+	PCIE_TX4+	A55

B56	PCIE_RX4-	PCIE_TX4-	A56
B57	N/C	GND	A57
B58	PCIE_RX3+	PCIE_TX3+	A58
B59	PCIE_RX3-	PCIE_TX3-	A59
B60	GND	GND	A60
B61	PCIE_RX2+	PCIE_TX2+	A61
B62	PCIE_RX2-	PCIE_TX2-	A62
B63	N/C	N/C	A63
B64	PCIE_RX1+	PCIE_TX1+	A64
B65	PCIE_RX1-	PCIE_TX1-	A65
B66	WAKE0#	GND	A66
B67	N/C	N/C	A67
B68	PCIE_RX0+	PCIE_TX0+	A68
B69	PCIE_RX0-	PCIE_TX0-	A69
B70	GND	GND	A70
B71	N/C	LVDS_A0+	A71
B72	N/C	LVDS_A0-	A72
B73	N/C	LVDS_A1+	A73
B74	N/C	LVDS_A1-	A74
B75	N/C	LVDS_A2+	A75
B76	N/C	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	N/C	LVDS_A_CK+	A81
B82	N/C	LVDS_A_CK-	A82
B83	CKLVDS_BKLT_CTRL	N/C	A83
B84	VCC_5V_SBY	N/C	A84
B85	VCC_5V_SBY	N/C	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	A91
B92	VGA_BLU	RSVD	A92
B93	VGA_HSYNC	N/C	A93
B94	VGA_VSYNC	RSVD	A94
B95	VGA_I2C_CK	RSVD	A95
B96	VGA_I2C_DAT	GND	A96
B97	N/C	VCC_12V	A97
B98	N/C	VCC_12V	A98
B99	N/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

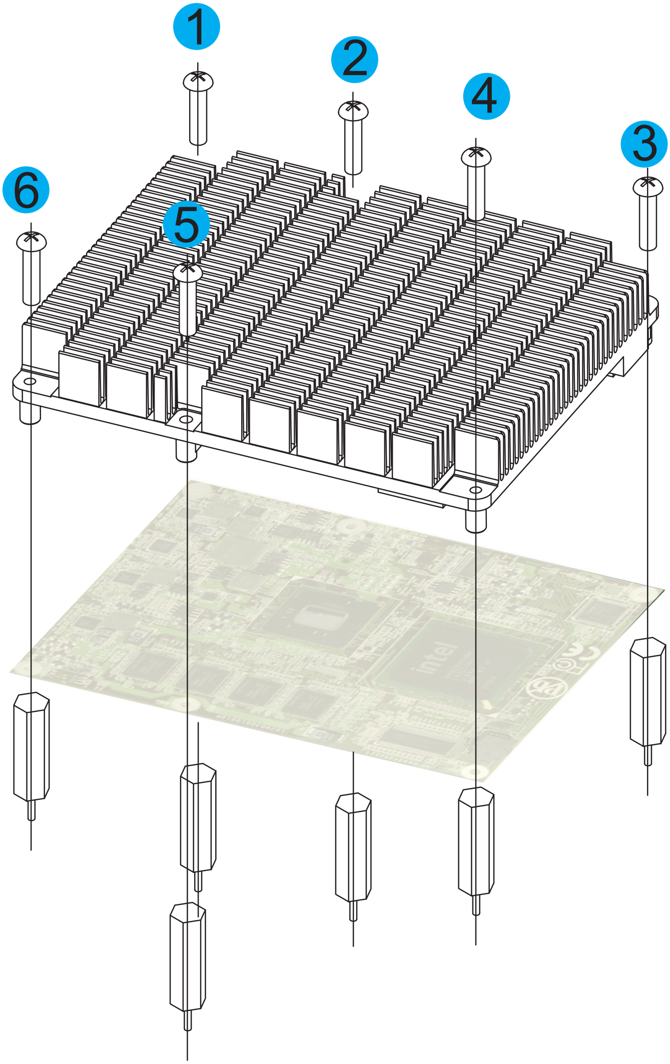
2.5 COM Express CD Connector (bottom side)

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	*N/C*	*PCI_AD31*	C48
D49	*PCI_M66EN*	*PCI_IRQA#*	C49
D50	*PCI_CLK*	*PCI_IRQB#*	C50
D51	*GND*	*GND (FIXED)*	C51
D52	*N/C*	*N/C*	C52
D53	*N/C*	*N/C*	C53
D54	*N/C*	*N/C*	C54
D55	*N/C*	*N/C*	C55

D56	*N/C*	*N/C*	C56
D57	*N/C*	*N/C*	C57
D58	*N/C*	*N/C*	C58
D59	*N/C*	*N/C*	C59
D60	*GND*	*GND*	C60
D61	*N/C*	*N/C*	C61
D62	*N/C*	*N/C*	C62
D63	*RSVD*	*RSVD*	C63
D64	*RSVD*	*RSVD*	C64
D65	*N/C*	*N/C*	C65
D66	*N/C*	*N/C*	C66
D67	*GND*	*RSVD*	C67
D68	*N/C*	*N/C*	C68
D69	*N/C*	*N/C*	C69
D70	*GND*	*GND*	C70
D71	*N/C*	*N/C*	C71
D72	*N/C*	*N/C*	C72
D73	*N/C*	*N/C*	C73
D74	*N/C*	*N/C*	C74
D75	*N/C*	*N/C*	C75
D76	*GND*	*GND*	C76
D77	*N/C*	*RSVD*	C77
D78	*N/C*	*N/C*	C78
D79	*N/C*	*N/C*	C79
D80	*GND*	*GND*	C80
D81	*N/C*	*N/C*	C81
D82	*N/C*	*N/C*	C82
D83	*RSVD*	*RSVD*	C83
D84	*GND*	*GND*	C84
D85	*N/C*	*N/C*	C85
D86	*N/C*	*N/C*	C86
D87	*GND*	*GND*	C87
D88	*N/C*	*N/C*	C88
D89	*N/C*	*N/C*	C89
D90	*GND*	*GND*	C90
D91	*N/C*	*N/C*	C91
D92	*N/C*	*N/C*	C92
D93	*GND*	*GND*	C93
D94	*N/C*	*N/C*	C94
D95	*N/C*	*N/C*	C95
D96	*GND*	*GND*	C96
D97	*N/C*	*RSVD*	C97
D98	*N/C*	*N/C*	C98
D99	*N/C*	*N/C*	C99
D100	*GND*	*GND*	C100
D101	*N/C*	*N/C*	C101
D102	*N/C*	*N/C*	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 Heatsink Installation

1. Put the heatsink and screw it on in the direction shown in the figure below.
2. Insert six screws downwards into the holes and screw them tightly.



2.7 The Installation Paths of CD Driver

Windows 2000 & XP

Driver	Path
CHIPSET	\CHIPSET\INF 9.11
LAN	\ETHERNET\INTEL\82574IT\WINXP_32_155 \ETHERNET\INTEL\82574IT\WINXP_64_155
VGA	\GRAPHICS\INTEL_2K_XP_32\5182
AUDIO	\AUDIO\REALTEK_HD\WIN2K_XP_x86x64_R252

Windows 7

Driver	Path
CHIPSET	\CHIPSET\INF 9.11
LAN	\ETHERNET\INTEL\82574IT\WIN7_32 \ETHERNET\INTEL\82574IT\WIN7_64
VGA	\GRAPHICS\INTEL_WIN7_32\2230 \GRAPHICS\INTEL_WIN7_64\2214
AUDIO	\AUDIO\REALTEK_HD\Win7_R252

Chapter 3

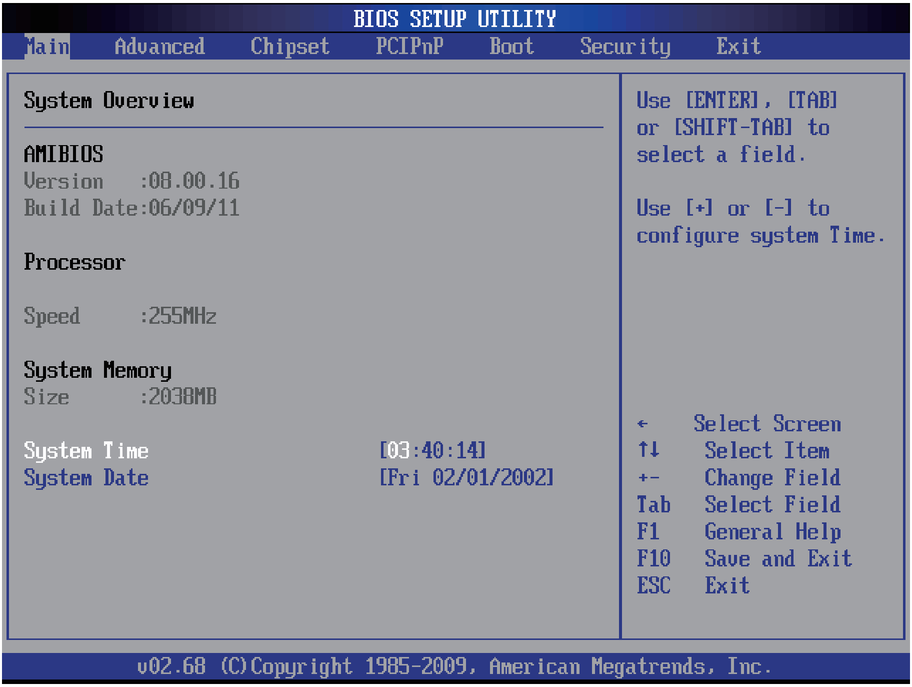
BIOS

3.1 BIOS Main Setup

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

When you turn on the computer, the AMI BIOS is immediately activated. To enter the BIOS SETUP UTILILTY, press **“Delete”** once the power is turned on.
When the computer is shut down, the battery on the motherboard supplies the power for BIOS RAM.

The **Main Setup** screen lists the following information
System Overview
BIOS Version: displays the current version information of the BIOS
Build Date: the date that the BIOS version was made/updated
Processor (auto-detected if installed)
Speed: displays the processor speed
System Memory (auto-detected if installed)
Size: lists the memory size information



```
System Overview                                    Use [ENTER], [TAB]
                                                   or [SHIFT-TAB] to
AMIBIOS                                            select a field.
Version    :08.00.16
Build Date:06/09/11                                Use [+] or [-] to
                                                   configure system Time.
Processor

Speed      :255MHz

System Memory
Size       :2038MB
                                                   ←    Select Screen
System Time                [03:40:14]              ↑↓    Select Item
System Date                [Fri 02/01/2002]        +-    Change Field
                                                   Tab   Select Field
                                                   F1    General Help
                                                   F10   Save and Exit
                                                   ESC   Exit

        v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

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

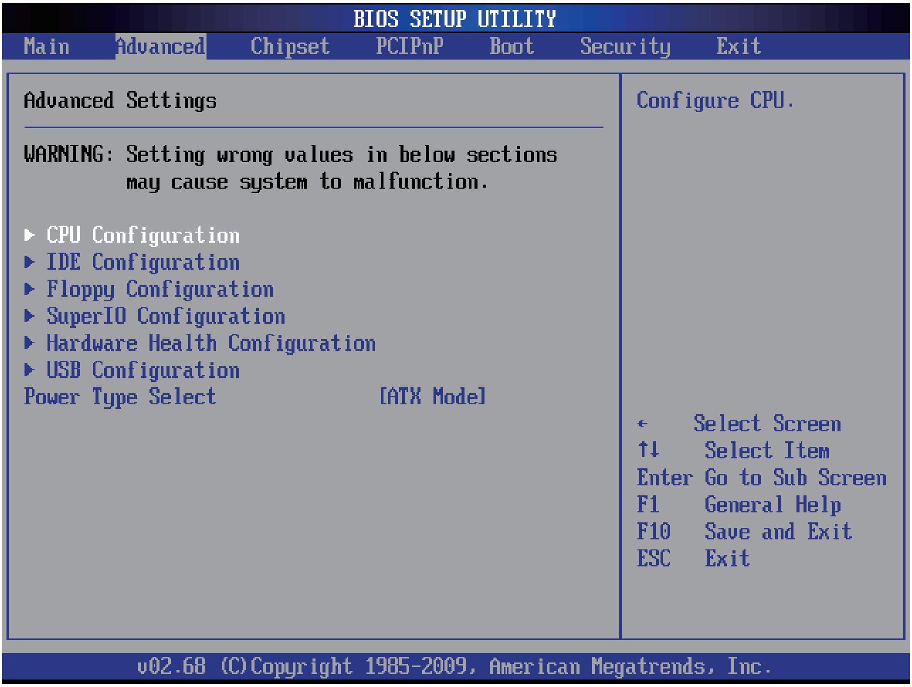
Key Commands

BIOS Setup Utility is mainly a key-based navigation interface. Please refer to the following key command instructions for navigation process.

← →	Move to highlight a particular configuration screen from the top menu bar / Move to highlight items on the screen
↓ ↑	Move to highlight previous/next item
Enter	Select and access a setup item/field
Esc	On the Main Menu – Quit the setup and not save changes into CMOS (a message screen will display and ask you to select "OK" or "Cancel" for exiting and discarding changes. Use "←" and "→" to select and press "Enter" to confirm) On the Sub Menu – Exit current page and return to main menu
Page Up / +	Increase the numeric value on a selected setup item / make change
Page Down / -	Decrease the numeric value on a selected setup item / make change
F1	Activate "General Help" screen
F10	Save the changes that have been made in the setup and exit. (a message screen will display and ask you to select "OK" or "Cancel" for exiting and saving changes. Use "←" and "→" to select and press "Enter" to confirm)

3.2 Advanced Settings

The “Advanced” screen provides the setting options to configure CPU, IDE, Floppy, SuperIO, Hardware Health and USB. You can use “←” and “→” keys to select “Advanced” and use the “↓” and “↑” to select a setup item.

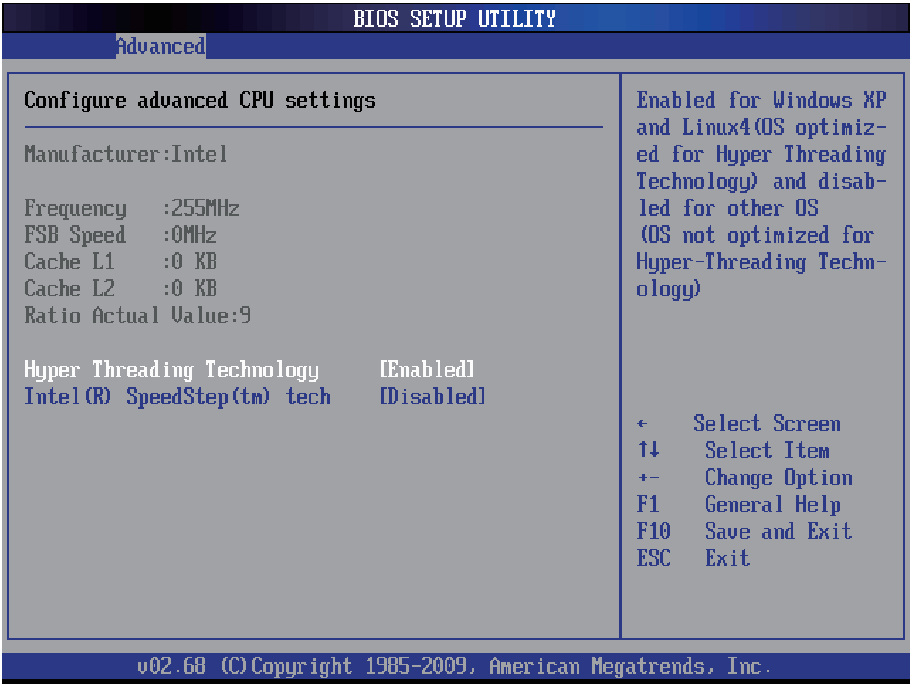


```
                         BIOS SETUP UTILITY
 Main    Advanced   Chipset   PCIPnP   Boot   Security   Exit

 Advanced Settings                                       Configure CPU.

 WARNING: Setting wrong values in below sections
          may cause system to malfunction.

 ▶ CPU Configuration
 ▶ IDE Configuration
 ▶ Floppy Configuration
 ▶ SuperIO Configuration
 ▶ Hardware Health Configuration
 ▶ USB Configuration
 Power Type Select               [ATX Mode]
                                                         ←     Select Screen
                                                         ↑↓    Select Item
                                                         Enter Go to Sub Screen
                                                         F1    General Help
                                                         F10   Save and Exit
                                                         ESC   Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Note: *please pay attention to the “WARNING” part at the left frame before you decide to configure any setting of an item.*

3.2.1 CPU Configuration

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



```
                         BIOS SETUP UTILITY
          Advanced

 Configure advanced CPU settings                 Enabled for Windows XP
                                                 and Linux4(OS optimiz-
 Manufacturer:Intel                              ed for Hyper Threading
                                                 Technology) and disab-
 Frequency    :255MHz                            led for other OS
 FSB Speed    :0MHz                              (OS not optimized for
 Cache L1     :0 KB                              Hyper-Threading Techn-
 Cache L2     :0 KB                              ology)
 Ratio Actual Value:9

 Hyper Threading Technology      [Enabled]
 Intel(R) SpeedStep(tm) tech     [Disabled]
                                                 ←    Select Screen
                                                 ↑↓    Select Item
                                                 +-    Change Option
                                                 F1    General Help
                                                 F10   Save and Exit
                                                 ESC   Exit

           v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Hyper Threading Technology

If enabled, your processor supports Hyper-Threading Technology.
The choice: Disabled, Enabled (Default).

Intel® SpeedStep™ tech

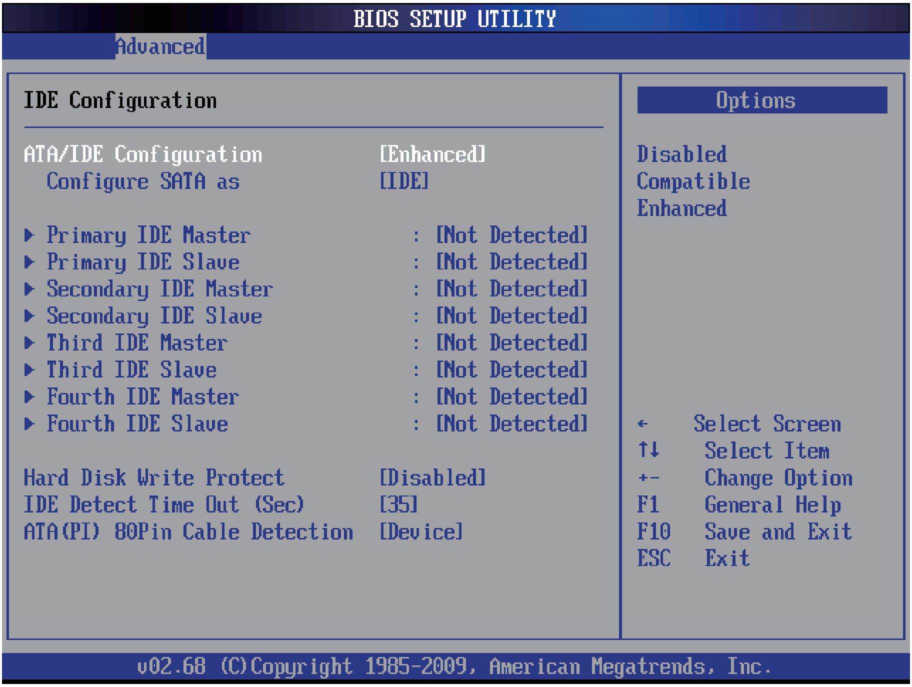
Maximum: CPU speed is set to maximum.
Minimum: CPU speed is set to minimum.
Automatic: CPU speed controlled by Operating system.
Disabled: Default CPU speed.

3.2.2 IDE Configuration

Select the “IDE Configuration” to configure the IDE settings. When an item is selected, there is a status description appearing at the right. You can use “Page Up/+” and “Page Down/-” keys to change the value of a selected item.

Primary IDE Master/Slave

Select one of the IDE devices to configure it. Press <Enter> to access its the sub menu.



```
                              BIOS SETUP UTILITY
          Advanced
 IDE Configuration                                       Options

 ATA/IDE Configuration            [Enhanced]             Disabled
   Configure SATA as              [IDE]                  Compatible
                                                         Enhanced
 ▶ Primary IDE Master               : [Not Detected]
 ▶ Primary IDE Slave                : [Not Detected]
 ▶ Secondary IDE Master             : [Not Detected]
 ▶ Secondary IDE Slave              : [Not Detected]
 ▶ Third IDE Master                 : [Not Detected]
 ▶ Third IDE Slave                  : [Not Detected]
 ▶ Fourth IDE Master                : [Not Detected]
 ▶ Fourth IDE Slave                 : [Not Detected]     ←     Select Screen
                                                         ↑↓     Select Item
 Hard Disk Write Protect          [Disabled]             +-     Change Option
 IDE Detect Time Out (Sec)        [35]                   F1     General Help
 ATA(PI) 80Pin Cable Detection    [Device]               F10    Save and Exit
                                                         ESC    Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

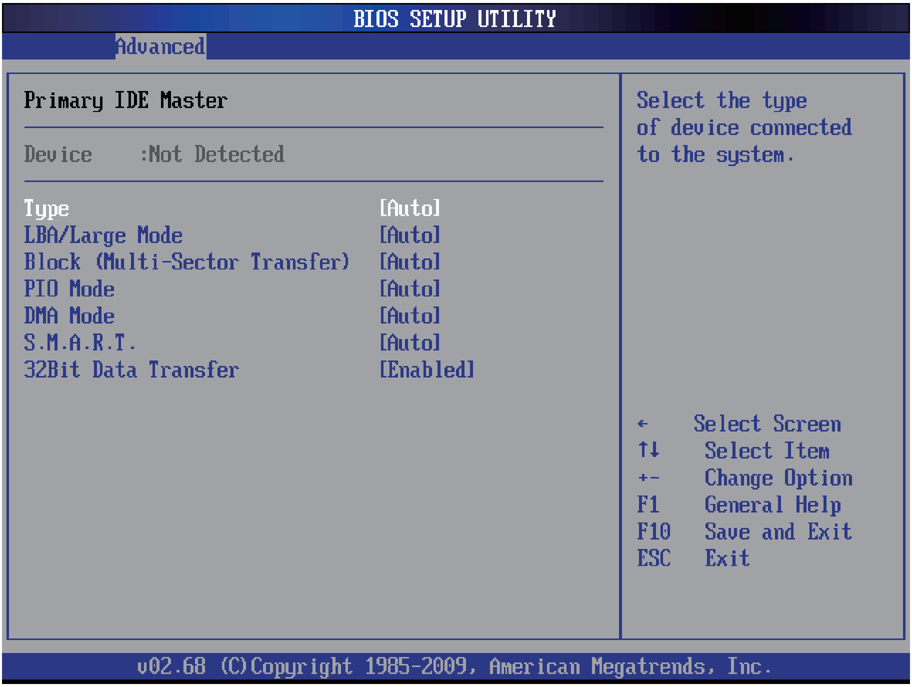
ATA/IDE Configuration

Use this item to specify the integrated IDE controller.
The choice: Disabled, Compatible, Enhanced

Configure SATA as

The choice: IDE Mode, AHCI Mode

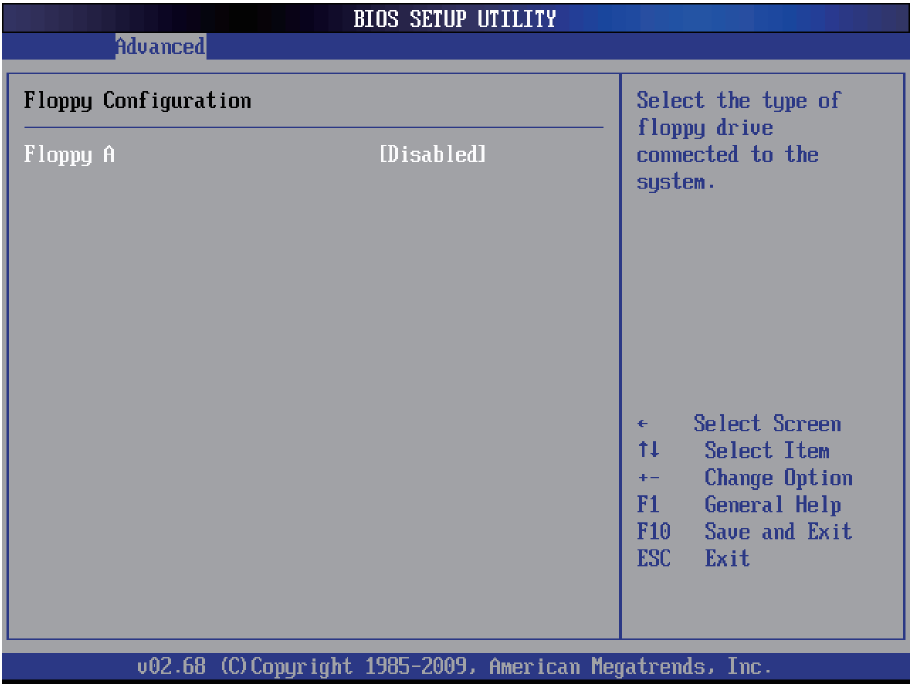
Primary IDE Master



```
                         BIOS SETUP UTILITY
          Advanced

 Primary IDE Master                                  Select the type
                                                     of device connected
 Device     :Not Detected                            to the system.

 Type                             [Auto]
 LBA/Large Mode                   [Auto]
 Block (Multi-Sector Transfer)    [Auto]
 PIO Mode                         [Auto]
 DMA Mode                         [Auto]
 S.M.A.R.T.                       [Auto]
 32Bit Data Transfer              [Enabled]

                                                     ←     Select Screen
                                                     ↑↓    Select Item
                                                     +-    Change Option
                                                     F1    General Help
                                                     F10   Save and Exit
                                                     ESC   Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Type: the type of devices.
LBA / Large Mode: LBA (Logical Block Addressing) is a method of addressing data on a disk drive. The maximum is 137 GB. You can set "Auto" (auto-detect or) or "Disabled".
Block (Multi-Sector Transfer): sets block sector transfer timing options.
PIO Mode: sets the IDE PIO (Programmable I/O) timing options.
DMA: configures the DMA options.
S.M.A.R.T.: sets "Auto", "Enable" or "Disable" for Self-Monitoring Analysis and Reporting Technology (S.M.A.R.T.) to predict impending drive failure.
32Bit Data Transfer: enables or disables 32-bit data transfer. The default is "Enabled".

3.2.3 Floppy Configuration



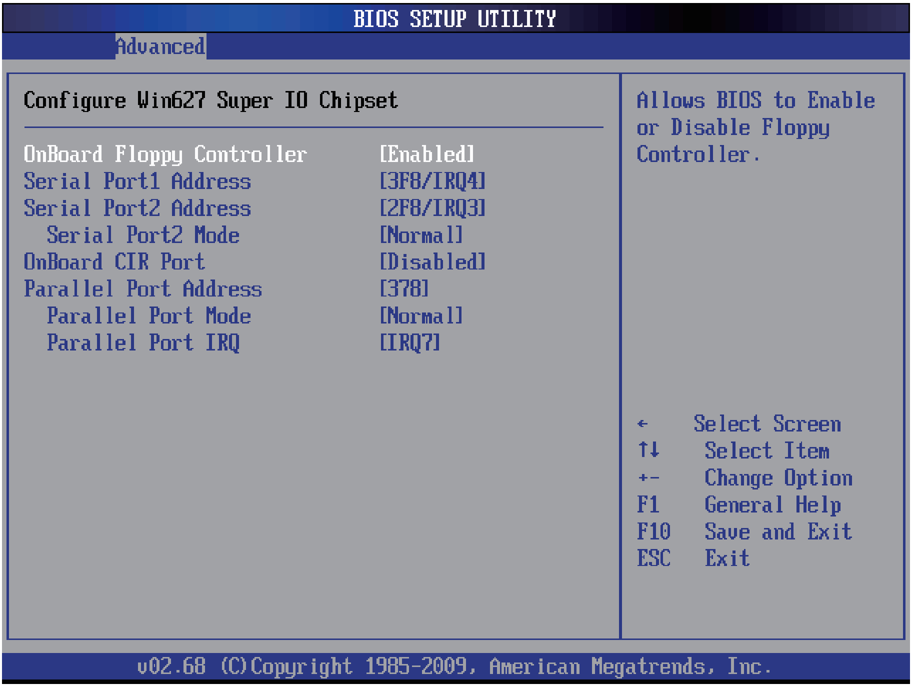
```
                         BIOS SETUP UTILITY
        Advanced
 Floppy Configuration                          Select the type of
                                               floppy drive
 Floppy A                  [Disabled]          connected to the
                                               system.

                                               ←    Select Screen
                                               ↑↓    Select Item
                                               +-    Change Option
                                               F1    General Help
                                               F10   Save and Exit
                                               ESC   Exit
         v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

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

Use “Super IO Configuration” to specify address and modes for Serial Port and Parallel Port.



```
                              BIOS SETUP UTILITY
        Advanced

 Configure Win627 Super IO Chipset                  Allows BIOS to Enable
                                                    or Disable Floppy
 OnBoard Floppy Controller        [Enabled]         Controller.
 Serial Port1 Address             [3F8/IRQ4]
 Serial Port2 Address             [2F8/IRQ3]
   Serial Port2 Mode              [Normal]
 OnBoard CIR Port                 [Disabled]
 Parallel Port Address            [378]
   Parallel Port Mode             [Normal]
   Parallel Port IRQ              [IRQ7]

                                                    ←     Select Screen
                                                    ↑↓    Select Item
                                                    +-    Change Option
                                                    F1    General Help
                                                    F10   Save and Exit
                                                    ESC   Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

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.

OnBoard CIR Port

Use this item to enable or disable support for onboard CIR port.

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

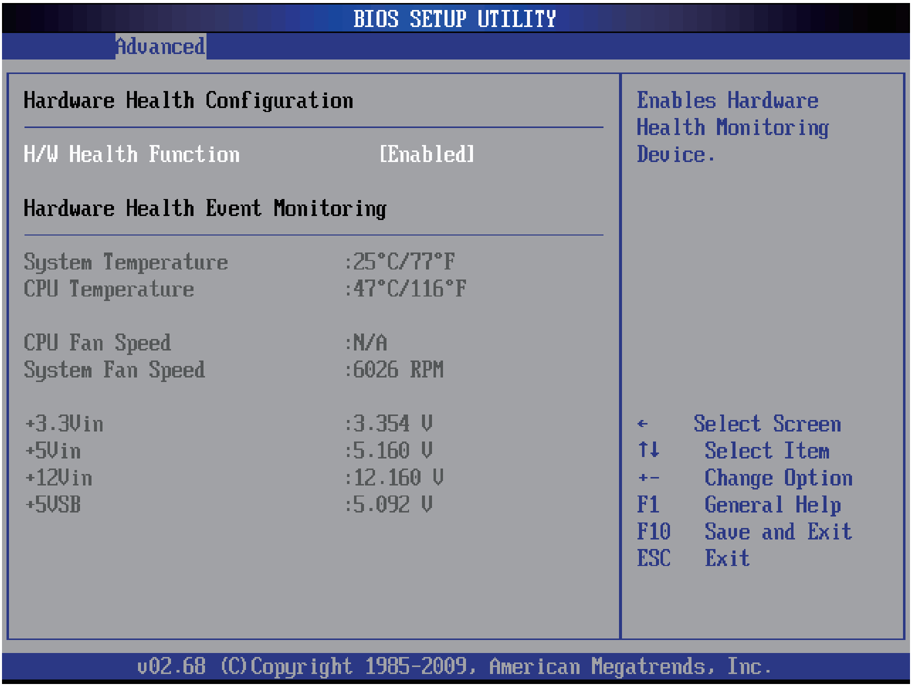
Parallel Port IRQ

Select an interrupt for the parallel port.
The choice:

IRQ5
IRQ7

3.2.5 Hardware Health Configuration

The “Hardware Health Configuration” lists out the temperature and voltage information that is being monitored. The default for “H/W Health Function” is “Enabled”.



```
                           BIOS SETUP UTILITY
          Advanced
 Hardware Health Configuration                       Enables Hardware
                                                      Health Monitoring
 H/W Health Function               [Enabled]          Device.

 Hardware Health Event Monitoring

 System Temperature             :25°C/77°F
 CPU Temperature                :47°C/116°F

 CPU Fan Speed                  :N/A
 System Fan Speed               :6026 RPM

 +3.3Vin                        :3.354 V              ←     Select Screen
 +5Vin                          :5.160 V              ↑↓     Select Item
 +12Vin                         :12.160 V             +-     Change Option
 +5VSB                          :5.092 V              F1     General Help
                                                      F10    Save and Exit
                                                      ESC    Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

System Temperature

Displays the currently monitored system temperature.

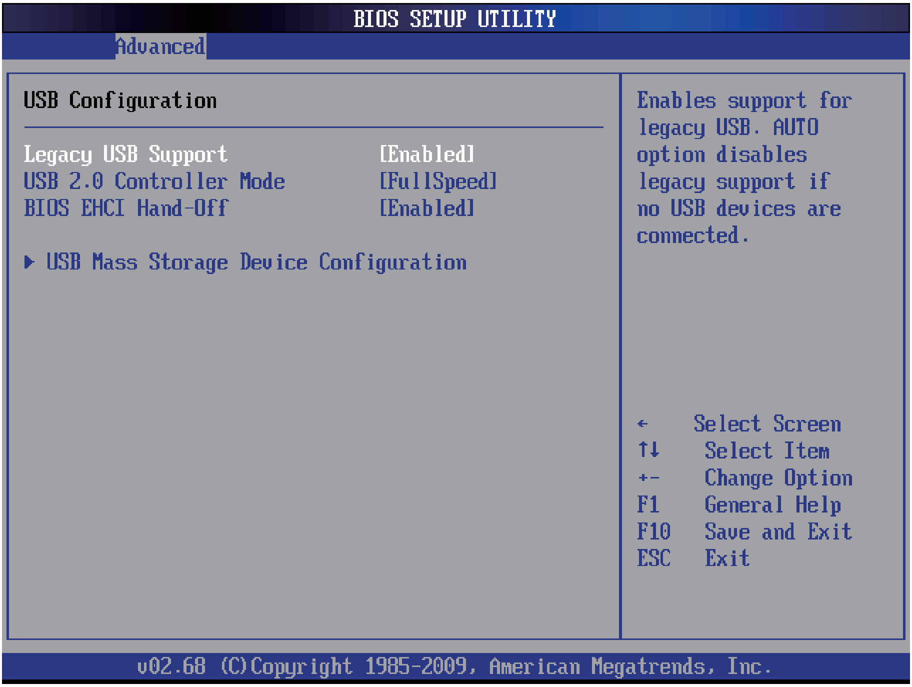
CPU Temperature

Displays the currently monitored CPU temperature.

+3.3Vin / +5Vin / +12Vin

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

3.2.6 USB Configuration



```
                              BIOS SETUP UTILITY
         Advanced
 USB Configuration                                    Enables support for
                                                      legacy USB. AUTO
 Legacy USB Support              [Enabled]            option disables
 USB 2.0 Controller Mode         [FullSpeed]          legacy support if
 BIOS EHCI Hand-Off              [Enabled]            no USB devices are
                                                      connected.
 ▶ USB Mass Storage Device Configuration

                                                      ←    Select Screen
                                                      ↑↓    Select Item
                                                      +-    Change Option
                                                      F1    General Help
                                                      F10   Save and Exit
                                                      ESC   Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Legacy USB Support

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

USB 2.0 Controller Mode

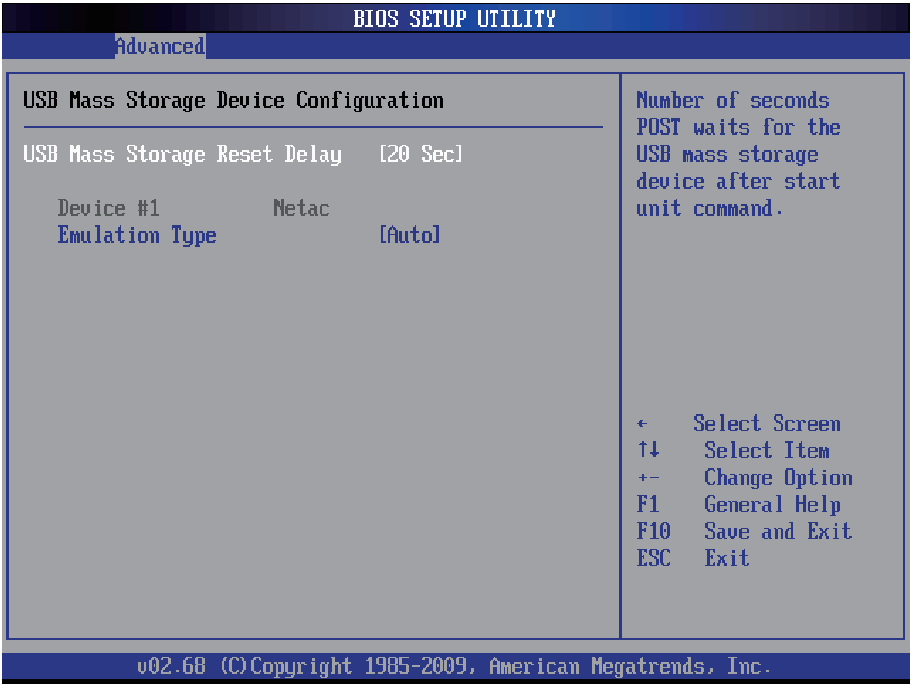
Configures the USB 2.0 controller in High Speed (480Mbps) or Full Speed (12MBPS).

BIOS EHCI Hand-Off

Enabled: enables the EHCI Hand-Off function by BIOS
Disabled: disables the EHCI Hand-Off function by BIOS
Note: *this setting option allows you to enable EHCI Hand Off if your computer operating system does not support it.*
EHCI is the abbreviation for Enhanced Host Controller Interface which is necessary for high speed USB operation.

USB Mass Storage Device Configuration

USB Mass Storage Reset Delay:
Number of seconds POST (Power-On Self-Test) waits for the USB mass storage device after starting unit command.



```
                              BIOS SETUP UTILITY
         Advanced

 USB Mass Storage Device Configuration         Number of seconds
                                               POST waits for the
 USB Mass Storage Reset Delay   [20 Sec]       USB mass storage
                                               device after start
    Device #1          Netac                   unit command.
    Emulation Type              [Auto]

                                               ←     Select Screen
                                               ↑↓    Select Item
                                               +-    Change Option
                                               F1    General Help
                                               F10   Save and Exit
                                               ESC   Exit

         v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Emulation Type

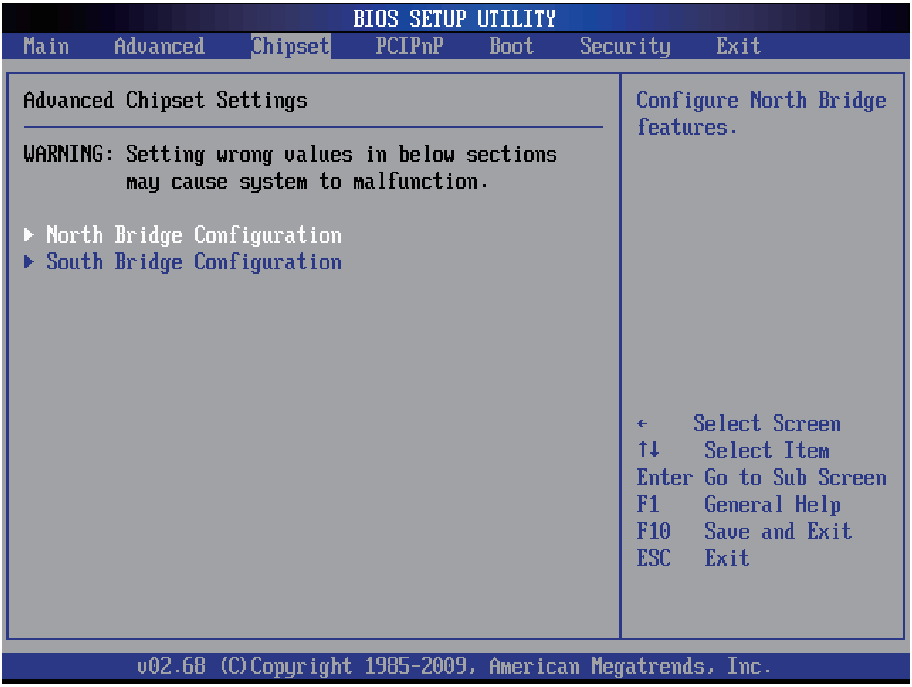
Sets the value for the system to select the emulation type for USB devices. In general, options include “Auto”, “FDD” and “HDD” (HDD stands for Hard Disk Drive, while FDD is also known as 3 1/2 floppy).
Please keep in mind that options such as “FDD” might not always be available as some computers are not built with this type of connectors.

Note:

If “Auto” is selected, USB device with storage less than 530MB will be emulated as Floppy and remain as hard drive. Forced FDD option can be used to force a HDD formatted drive to “BOOT” as FDD (for example, ZIP drive)

3.3 Chipset Setting

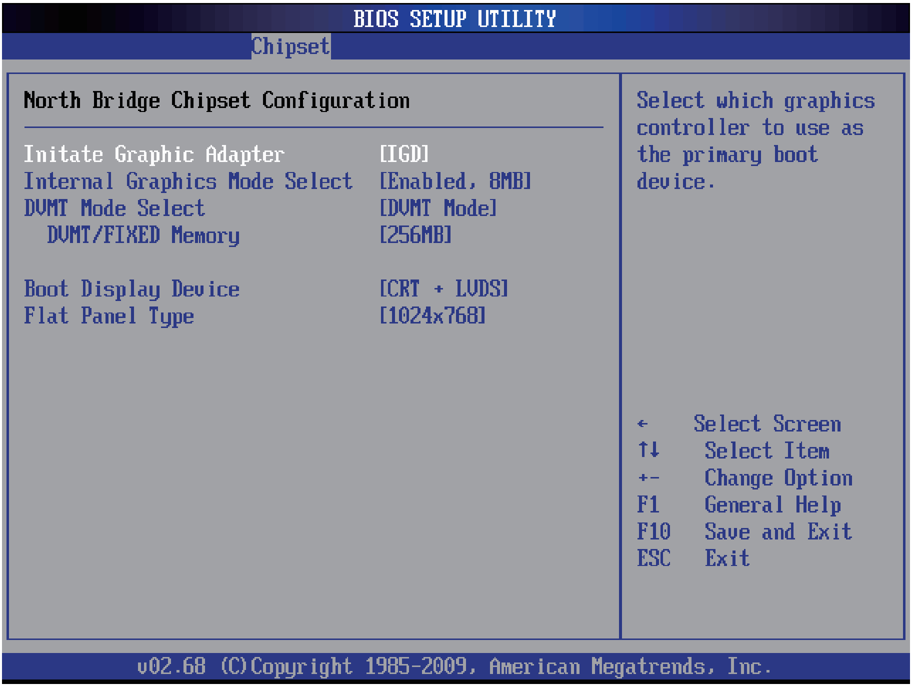
Select “Chipset” to access to “North Bridge Configuration” and “South Bridge Configuration”. You can enter the sub menu of the two configuration options.



```
                         BIOS SETUP UTILITY
  Main     Advanced     Chipset     PCIPnP     Boot     Security     Exit

Advanced Chipset Settings                           Configure North Bridge
                                                    features.
WARNING: Setting wrong values in below sections
         may cause system to malfunction.

▶ North Bridge Configuration
▶ South Bridge Configuration

                                                    ←     Select Screen
                                                    ↑↓     Select Item
                                                    Enter Go to Sub Screen
                                                    F1     General Help
                                                    F10    Save and Exit
                                                    ESC    Exit

            v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Note: *please pay attention to the “WARNING” part at the left frame before you decide to configure any setting of an item.*

3.3.1 North Bridge Chipset Configuration



```
                              BIOS SETUP UTILITY
                    Chipset

 North Bridge Chipset Configuration                   Select which graphics
 ─────────────────────────────────────────────        controller to use as
 Initate Graphic Adapter          [IGD]               the primary boot
 Internal Graphics Mode Select    [Enabled, 8MB]      device.
 DVMT Mode Select                 [DVMT Mode]
   DVMT/FIXED Memory              [256MB]

 Boot Display Device              [CRT + LVDS]
 Flat Panel Type                  [1024x768]




                                                      ←    Select Screen
                                                      ↑↓    Select Item
                                                      +-    Change Option
                                                      F1    General Help
                                                      F10   Save and Exit
                                                      ESC   Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Initate Graphic Adapter: select which graphics controller to use as the primary boot device.

Integrated Graphics Mode Select: when set as “Enabled”, you can select the size of system memory that can be used for the integrated graphic device.

DVMT Mode Select: This item allows you to select the DVMT mode.
The choice: FIXED, DVMT, BOTH.

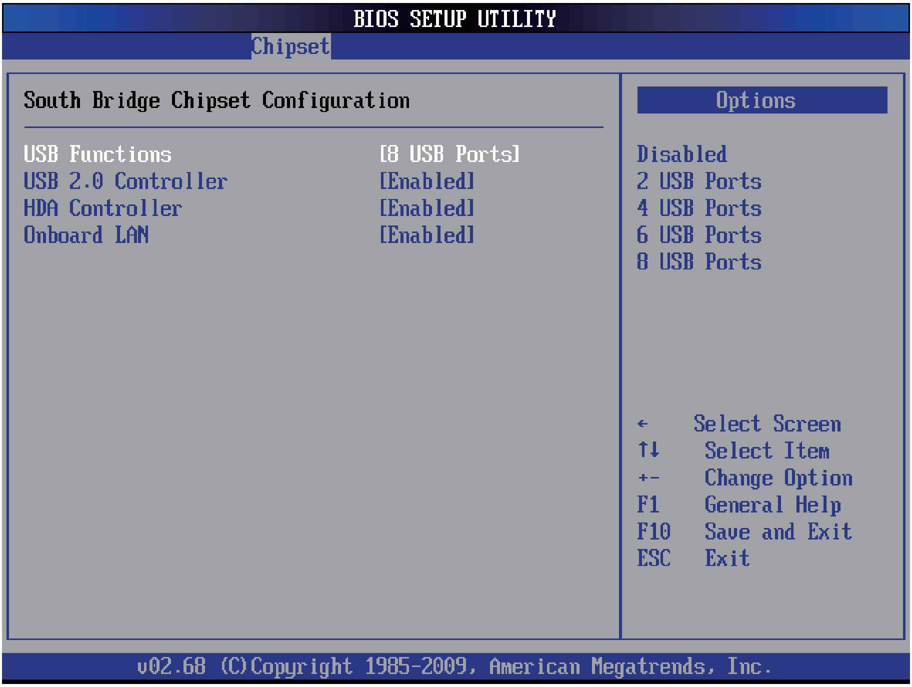
DVMT/FIXED Memory: This item allows you to select the DVMT or FIXED memory size.

Boot Display Device: boot setting for the display device connected to the computer, such as “External CRT” monitor.

Flat Panel Type: the resolution types of the connected flat panel display device.

3.3.2 South Bridge Chipset Configuration

Normally, the south bridge controls the basic I/O functions, such as USB. This screen allows you to access to the configurations of the IOs.



```
                         BIOS SETUP UTILITY
                 Chipset
 South Bridge Chipset Configuration                   Options
 ---------------------------------------------
 USB Functions                  [8 USB Ports]         Disabled
 USB 2.0 Controller             [Enabled]             2 USB Ports
 HDA Controller                 [Enabled]             4 USB Ports
 Onboard LAN                    [Enabled]             6 USB Ports
                                                      8 USB Ports

                                                      ←     Select Screen
                                                      ↑↓     Select Item
                                                      +-     Change Option
                                                      F1     General Help
                                                      F10    Save and Exit
                                                      ESC    Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

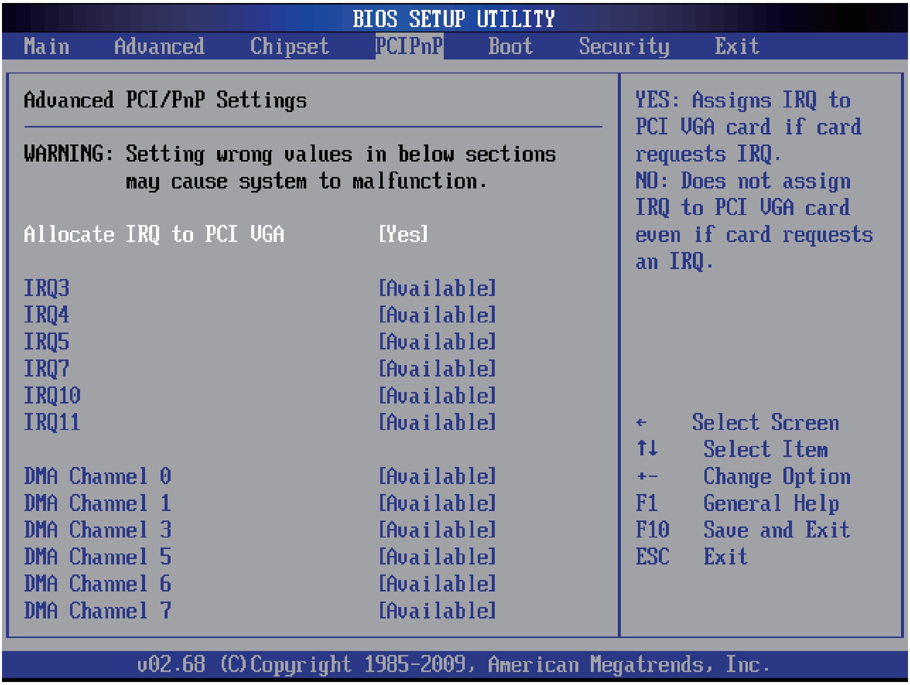
USB Functions: selects the number of USB ports to be enabled.

USB 2.0 Controller: if your computer has USB 2.0 ports, please choose “Enabled” to activate the USB 2.0 ports. The default is “Enabled”.

HDA Controller: this item allows you to select the chipset family to support High Definition Audio Controller.
The Choice: Enabled, Disabled.

Onboard LAN: Select “Enabled” if your system has a LAN device installed on the system board and you wish to use it.
The Choice: Enabled, Disabled.

3.4 Advanced PCI/PnP Settings



```
                              BIOS SETUP UTILITY
 Main     Advanced     Chipset     PCIPnP     Boot     Security     Exit

 Advanced PCI/PnP Settings                              YES: Assigns IRQ to
                                                        PCI VGA card if card
 WARNING: Setting wrong values in below sections        requests IRQ.
          may cause system to malfunction.              NO: Does not assign
                                                        IRQ to PCI VGA card
 Allocate IRQ to PCI VGA          [Yes]                 even if card requests
                                                        an IRQ.
 IRQ3                             [Available]
 IRQ4                             [Available]
 IRQ5                             [Available]
 IRQ7                             [Available]
 IRQ10                            [Available]
 IRQ11                            [Available]           ←     Select Screen
                                                        ↑↓     Select Item
 DMA Channel 0                    [Available]           +-     Change Option
 DMA Channel 1                    [Available]           F1     General Help
 DMA Channel 3                    [Available]           F10    Save and Exit
 DMA Channel 5                    [Available]           ESC    Exit
 DMA Channel 6                    [Available]
 DMA Channel 7                    [Available]

            v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Allocate IRQ to PCI VGA

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

IRQ3 - IRQ11

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

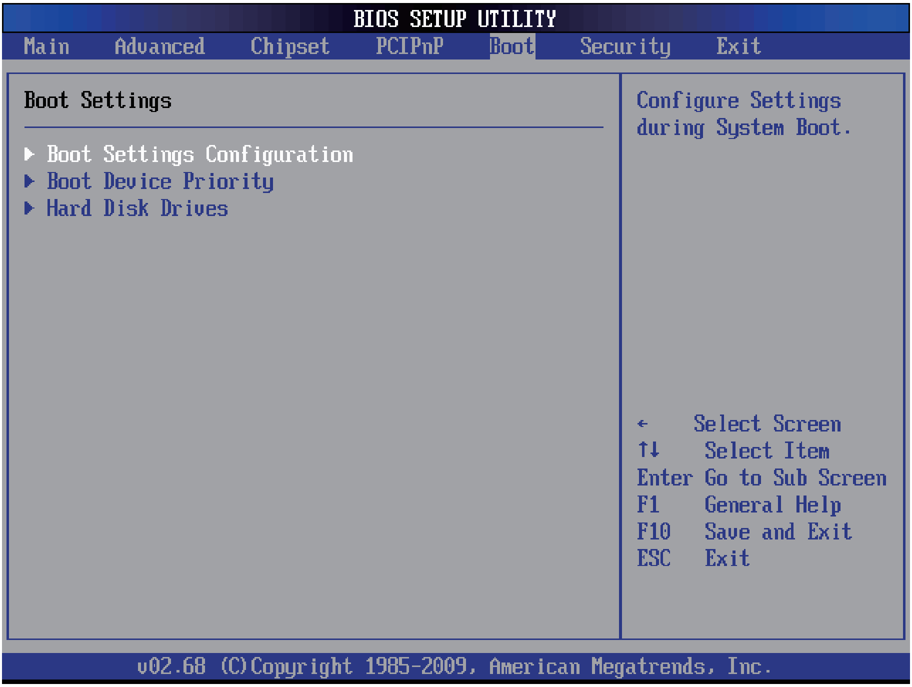
DMA Channel 0 - DMA Channel 7

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

Note: *please pay attention to the “WARNING” part at the left frame before you decide to configure any setting of an item.*

3.5 Boot Setting

The “Boot” screen provides the access to configure the settings for system boot.



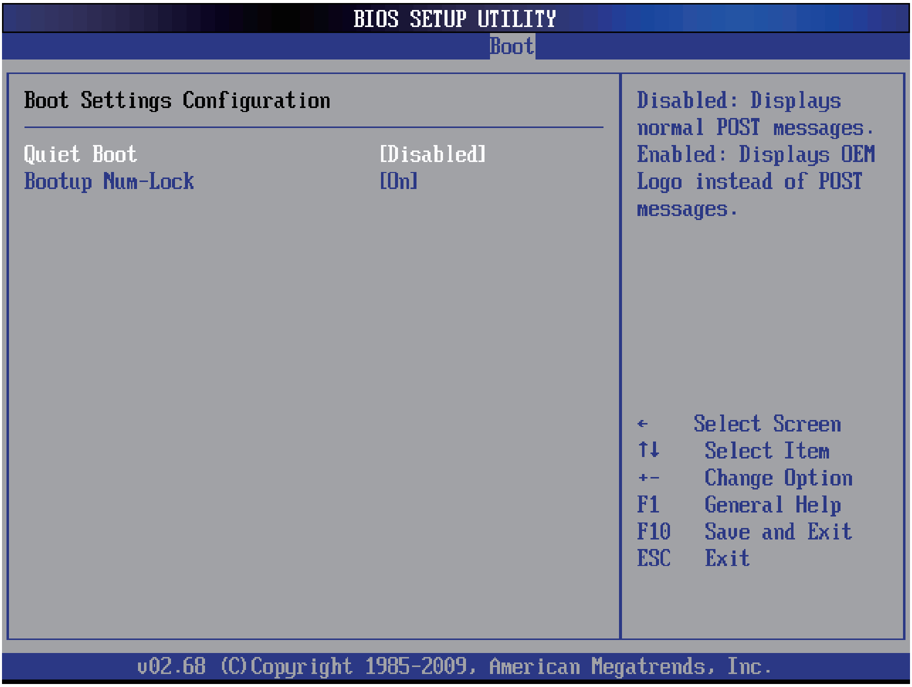
```
                           BIOS SETUP UTILITY
 Main    Advanced    Chipset    PCIPnP    Boot    Security    Exit

 Boot Settings                                  Configure Settings
                                                during System Boot.
 ▶ Boot Settings Configuration
 ▶ Boot Device Priority
 ▶ Hard Disk Drives




                                                ←     Select Screen
                                                ↑↓     Select Item
                                                Enter Go to Sub Screen
                                                F1     General Help
                                                F10    Save and Exit
                                                ESC    Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Boot Setting Configuration: enter the sub menu for boot setting.

Boot Device Priority: access to the sub menu for boot device priority.

Hard Disk Drives: Press Enter and it shows Bootable and Hard Disk drives.

3.5.1 Boot Setting Configuration

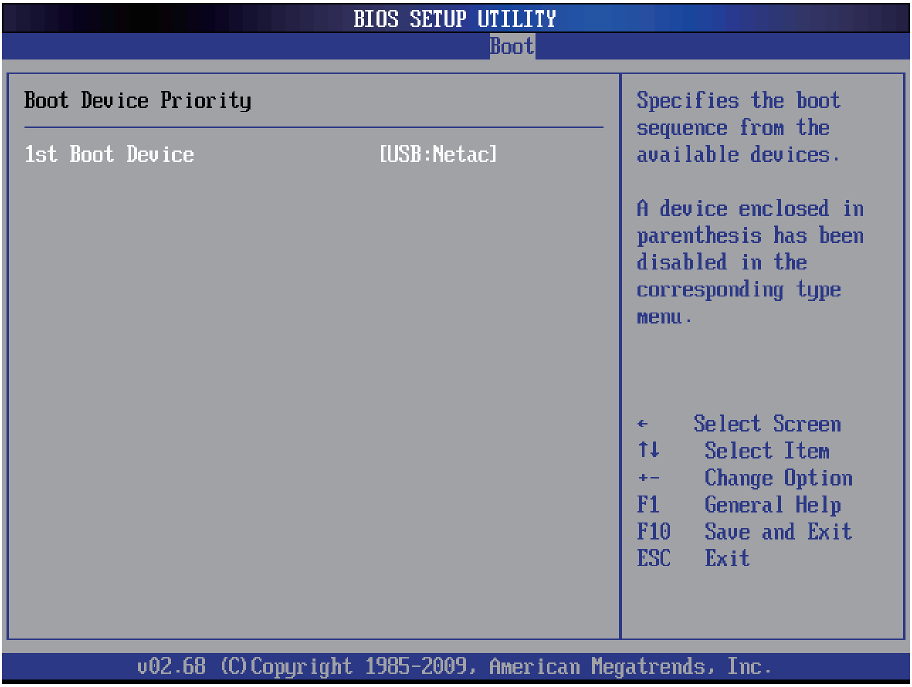


```
                              BIOS SETUP UTILITY
                                                Boot
 Boot Settings Configuration                      Disabled: Displays
                                                  normal POST messages.
 Quiet Boot                    [Disabled]         Enabled: Displays OEM
 Bootup Num-Lock               [On]               Logo instead of POST
                                                  messages.

                                                  ←    Select Screen
                                                  ↑↓    Select Item
                                                  +-    Change Option
                                                  F1    General Help
                                                  F10   Save and Exit
                                                  ESC   Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Quiet Boot: displays normal POST messages when it's selected as "Disabled". When it is set as "Enabled", OEM messages will be displayed instead of POST messages. The default is "Disabled".

Bootup Num-Lock: modifies Number Lock setting when the system boots up. Select "On" to automatically enable the Number Lock on keyboard when the system is booting up.

3.5.2 Boot Device Priority

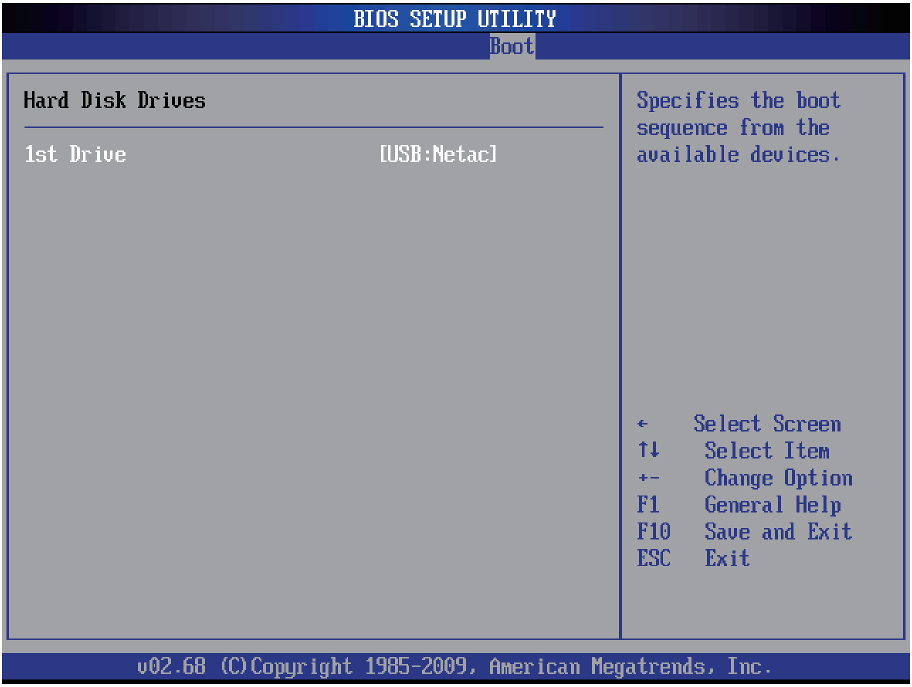


```
                              BIOS SETUP UTILITY
                                     Boot
 Boot Device Priority                              Specifies the boot
                                                   sequence from the
 1st Boot Device               [USB:Netac]         available devices.

                                                   A device enclosed in
                                                   parenthesis has been
                                                   disabled in the
                                                   corresponding type
                                                   menu.

                                                   ←    Select Screen
                                                   ↑↓    Select Item
                                                   +-    Change Option
                                                   F1    General Help
                                                   F10   Save and Exit
                                                   ESC   Exit

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

1st Boot Device

Select which devices to be booted according to the priority order of available devices.

3.5.3 Hard Disk Drives



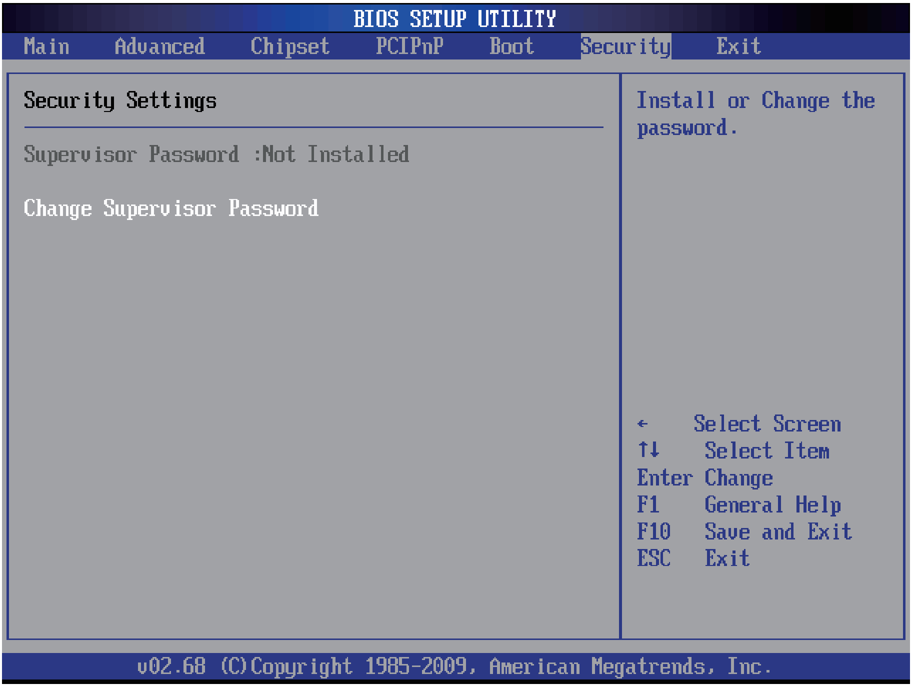
```
                              BIOS SETUP UTILITY
                                                Boot
Hard Disk Drives                                        Specifies the boot
                                                        sequence from the
1st Drive                    [USB:Netac]                available devices.

                                                        ←    Select Screen
                                                        ↑↓   Select Item
                                                        +-   Change Option
                                                        F1   General Help
                                                        F10  Save and Exit
                                                        ESC  Exit
        v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

1st Drive

Select which drives to be booted according to the priority order of available drives.

3.6 Security Setting

The “Security Settings” screen allows you to set password.

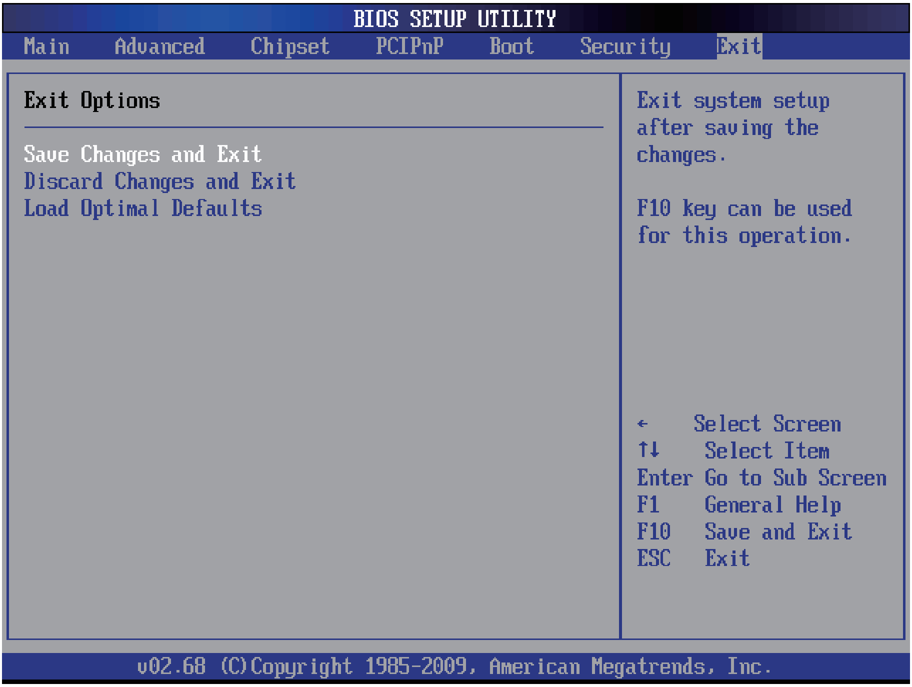


```
                         BIOS SETUP UTILITY
 Main    Advanced    Chipset    PCIPnP    Boot    Security    Exit

 Security Settings                                  Install or Change the
                                                    password.
 Supervisor Password :Not Installed

 Change Supervisor Password




                                                    ←    Select Screen
                                                    ↑↓    Select Item
                                                    Enter Change
                                                    F1    General Help
                                                    F10   Save and Exit
                                                    ESC   Exit

        v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Change Supervisor Password: the default is “Not Installed”, but you can change the Supervisor Password and then it will appear “Installed”. Please always remember your password or else you will have to reset the whole system.

3.7 Exit Setting

Select “Exit” to set exit options, save changes or load default values.



```
Exit Options                       Exit system setup
                                   after saving the
Save Changes and Exit              changes.
Discard Changes and Exit
Load Optimal Defaults              F10 key can be used
                                   for this operation.

                                   ←     Select Screen
                                   ↑↓     Select Item
                                   Enter Go to Sub Screen
                                   F1     General Help
                                   F10    Save and Exit
                                   ESC    Exit

       v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

Save Changes and Exit

When you press “Enter” on this option, a message described as the one below will appear:

“Save configuration changes and exit setup?”

Pressing <OK> stores the configuration changes made in BIOS in CMOS menu - a special section of memory that stays on after you turn your system off, and then exit. The next time you boot your system up, the new configured system values will take place.

Note: *you can also press <F10> to enable this operation.*

Discard Changes and Exit

Exit system setup without saving any changes.
You can also press <ESC> to activate this function.

Load Optimal Defaults

When you press <Enter> on this option, a message dialog box will appear asking for your confirmation:

Load Optimal Defaults?
[OK]   [Cancel]

Press [OK] to load the BIOS Optimal Default values for all the setup options.

You can also press <F9> key to enable 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.

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.

Address	Device Description
0x00000000-0x00000CF7	PCI bus
0x00000000-0x00000CF7	Direct memory access controller
0x00000010-0x0000001F	Motherboard resources
0x00000020-0x00000021	Programmable interrupt controller
0x00000022-0x0000003F	Motherboard resources
0x00000040-0x00000043	System timer
0x00000044-0x0000005F	Motherboard resources
0x00000060-0x00000060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
0x00000061-0x00000061	System speaker
0x00000062-0x00000063	Motherboard resources
0x00000064-0x00000064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
0x00000065-0x0000006F	Motherboard resources
0x00000070-0x00000071	System CMOS/real time clock
0x00000072-0x0000007F	Motherboard resources
0x00000080-0x00000080	Motherboard resources
0x00000081-0x00000083	Direct memory access controller
0x00000084-0x00000086	Motherboard resources
0x00000087-0x00000087	Direct memory access controller
0x00000088-0x00000088	Motherboard resources
0x00000089-0x0000008B	Direct memory access controller
0x0000008C-0x0000008E	Motherboard resources
0x0000008F-0x0000008F	Direct memory access controller
0x00000090-0x0000009F	Motherboard resources
0x000000A0-0x000000A1	Programmable interrupt controller
0x000000A2-0x000000BF	Motherboard resources

0x000000C0-0x000000DF	Direct memory access controller
0x000000E0-0x000000EF	Motherboard resources
0x000000F0-0x000000FF	Numeric data processor
0x000001F0-0x000001F7	Primary IDE Channel
0x00000274-0x00000277	ISAPNP Read Data Port
0x00000279-0x00000279	ISAPNP Read Data Port
0x000002F8-0x000002FF	Communications Port (COM2)
0x00000378-0x0000037F	Printer Port (LPT1)
0x000003B0-0x000003BB	Intel(R) Graphics Media Accelerator 3150
0x000003C0-0x000003DF	Intel(R) Graphics Media Accelerator 3150
0x000003F6-0x000003F6	Primary IDE Channel
0x000003F8-0x000003FF	Communications Port (COM1)
0x00000400-0x0000041F	Intel(R) ICH8 Family SMBus Controller - 283E
0x000004D0-0x000004D1	Motherboard resources
0x00000500-0x0000053F	Motherboard resources
0x00000800-0x0000087F	Motherboard resources
0x00000A00-0x00000A0F	Motherboard resources
0x00000A79-0x00000A79	ISAPNP Read Data Port
0x00000D00-0x0000FFFF	PCI bus
0x0000C400-0x0000C407	Intel(R) Graphics Media Accelerator 3150
0x0000C480-0x0000C49F	Standard Universal PCI to USB Host Controller
0x0000C800-0x0000C81F	Intel(R) ICH8 Family USB Universal Host Controller - 2832
0x0000C880-0x0000C89F	Intel(R) ICH8 Family USB Universal Host Controller - 2831
0x0000CC00-0x0000CC1F	Intel(R) ICH8 Family USB Universal Host Controller - 2830
0x0000D080-0x0000D08F	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
0x0000D400-0x0000D40F	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828

0x0000D480-0x0000D483	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
0x0000D800-0x0000D807	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
0x0000D880-0x0000D883	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
0x0000DC00-0x0000DC07	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
0x0000E000-0x0000EFFF	Intel(R) ICH8 Family PCI Express Root Port 6 - 2849
0x0000EC00-0x0000EC1F	Intel(R) 82574L Gigabit Network Connection
0x0000FFA0-0x0000FFAF	Intel(R) ICH8M Ultra ATA Storage Controllers - 2850

Appendix B: BIOS Memory Map

Address	Device Description
0xF0000000-0xFED8FFFF	PCI bus
0xFE900000-0xFE97FFFF	Intel(R) Graphics Media Accelerator 3150
0xD0000000-0xDFFFFFFF	Intel(R) Graphics Media Accelerator 3150
0xFE800000-0xFE8FFFFF	Intel(R) Graphics Media Accelerator 3150
0xFE780000-0xFE7FFFFF	Intel(R) Graphics Media Accelerator 3150
0xFE9F8000-0xFE9FBFFF	Microsoft UAA Bus Driver for High Definition Audio
0xFEA00000-0xFEBFFFFF	Intel(R) ICH8 Family PCI Express Root Port 6 - 2849
0xFEAE0000-0xFEAFFFFF	Intel(R) 82574L Gigabit Network Connection
0xFEB00000-0xFEBFFFFF	Intel(R) 82574L Gigabit Network Connection
0xFEADC000-0xFEADFFFF	Intel(R) 82574L Gigabit Network Connection
0xFE9FF800-0xFE9FFBFF	Intel(R) ICH8 Family USB2 Enhanced Host Controller - 2836
0xFED1C000-0xFED1FFFF	Motherboard resources
0xFED20000-0xFED3FFFF	Motherboard resources

0xFED40000-0xFED8FFFF	Motherboard resources
0xFED00000-0xFED003FF	High precision event timer
0xFFB00000-0xFFBFFFFF	Intel(R) 82802 Firmware Hub Device
0xFFF00000-0xFFFFFFFF	Intel(R) 82802 Firmware Hub Device
0xFFC00000-0xFFEFFFFF	Motherboard resources
0xFEC00000-0xFEC00FFF	Motherboard resources
0xFEE00000-0xFEE00FFF	Motherboard resources
0xFE9FFC00-0xFE9FFCFF	Intel(R) ICH8 Family SMBus Controller - 283E
0xFED14000-0xFED19FFF	System board
0xFED90000-0xFED93FFF	System board
0xFED90000-0xFED93FFF	System board
0xE0000000-0xEFFFFFFF	Motherboard resources
0x0000-0x9FFFF	System board
0xA0000-0xBFFFF	PCI bus
0xA0000-0xBFFFF	Intel(R) Graphics Media Accelerator 3150
0xC0000-0xCFFFF	System board
0xD0000-0xDFFFF	PCI bus
0xE0000-0xFFFFF	System board
0x100000-0x7F6FFFFF	System board
0x7F700000-0xDFFFFFFF	PCI bus

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 0	System timer
IRQ 1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
IRQ 3	Communications Port (COM2)
IRQ 4	Communications Port (COM1)
IRQ 8	System CMOS/real time clock

IRQ 9	Microsoft ACPI-Compliant System
IRQ 11	Intel(R) ICH8 Family SMBus Controller - 283E
IRQ 12	Microsoft PS/2 Mouse
IRQ 13	Numeric data processor
IRQ 14	Primary IDE Channel
IRQ 16	Intel(R) Graphics Media Accelerator 3150
IRQ 16	Standard Universal PCI to USB Host Controller
IRQ 17	Intel(R) 82574L Gigabit Network Connection
IRQ 18	Intel(R) ICH8 Family USB Universal Host Controller - 2832
IRQ 18	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
IRQ 19	Intel(R) ICH8 Family USB Universal Host Controller - 2831
IRQ 21	Microsoft UAA Bus Driver for High Definition Audio
IRQ 22	Intel(R) ICH8 Family PCI Express Root Port 1 - 283F
IRQ 23	Intel(R) ICH8 Family PCI Express Root Port 6 - 2849
IRQ 23	Intel(R) ICH8 Family USB Universal Host Controller - 2830
IRQ 23	Intel(R) ICH8 Family USB2 Enhanced Host Controller - 2836

Appendix D: Digital I/O Setting

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

C Language Code

```
//==== History ====//
//compile by TCPP 3.0
//R00     5/18/2010          1st modify

//#include “ring1726.h”
#include <stdio.h>
#include <dos.h>
#include <conio.h>

#define EC_CMD_Port                    0x6C
#define  EC_DATA_Port    0x68

unsigned long Process_686C_Command_Write(unsigned long m_ECCMD, unsigned
long m_ECDATA);
```

```
unsigned long Process_686C_Command_Read(unsigned long m_ECCMD );
unsigned long ECU_Write_686C_RAM_BYTE( unsigned long
ECUMemAddr,unsigned long ECUMemData );
unsigned long ECU_Read_686C_RAM_BYTE( unsigned long ECUMemAddr );
unsigned char SMB_Byte_READ(int SMPORT, int DeviceID, int REG_INDEX);
void SMB_Byte_WRITE(int SMPORT, int DeviceID, int REG_INDEX, int REG_DATA);



char APName[]=      “\t\tEmETX-i2903+PBE1700) DIO Testing Program\n”
                    “\t==========================================\n” ;

char APHelp[]=      “\n - Pass ‘A’ key for inver state of DIO GP1x”
                                        “\n - Pass ‘S’ key for inver state of DIO GP2x”
                                        “\n - Pass ‘D’ key for inver state of DIO GP3x”
                    “\n - Pass ‘Esc’ key for Exit”
                    “\n” ;

void main(void){
          char getkey = 0;
//        char DIOSTS=0;
//        char tempJ=0;
//        char tempA=0;
          unsigned char GP2xVal,GP3xVal,GP1xVal;
          int SMB_PORT_AD  = 0x400;
          //--int SMB_DEVICE_ADD = 0x9C;  /*75111R’s Add=6eh */
          int SMB_DEVICE_ADD = 0x6E;      /*75111R’s Add=6eh */

          clrscr();  //clear screen
          printf(APName);
          printf(APHelp);

          //pg DIO as output
          //0:input 1:Output
/*        Index 10, GPIO1x Output pin control          */
          SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x10,0xff);
          delay(10);
/*        Index 20, GPIO2x Output pin control          */
//poweron defalult 0x00::::  SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_
ADD,0x20,0x00); //pg as Input
          SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x20,0xff);
/*        Index 40, GPIO3x Output pin control          */
          SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x40,0x0f);
          delay(10);
```

```
            //pg DIO default LOW
/*          Index 11, GPIO1x Output Data value           */
            SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x11,0x00);
            GP1xVal = 0;
            delay(10);

/*          Index 21, GPIO2x Output Data value           */
            SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x21,0x00);
            GP2xVal = 0;
            delay(10);

/*          Index 41, GPIO3x Output Data value           */
            SMB_Byte_WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x41,0x00);
            GP3xVal = 0;

            gotoxy(1,9);
            //printf("DIO Status: Low \n");

            do{
                        if (getkey != 27){
                                    while (!kbhit());
                                    getkey = getch();
                                    switch (getkey){
                                                case 'D':
                                                case 'd':
                                                            if (GP3xVal == 0)
                                                            {
                                                                        GP3xVal = 1; //DIO all
high
                                                                        //pg DIO high
                                                                        SMB_Byte_
WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x41,0x0f);

                                                                        gotoxy(1,10);
                                                                        printf("GP3x Status:
LED OFF\n");

                                                            }
                                                            else
                                                            {
                                                                        GP3xVal = 0; //DIO all
low
                                                                        //pg DIO LOW
                                                                        SMB_Byte_
WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x41,0x00);
```

```
                            gotoxy(1,10);
                            printf("GP3x Status:
LED ON \n");
                        }

                        break;
                case 'A':
                case 'a':
                        if (GP1xVal == 0)
                        {
                            GP1xVal = 1; //DIO all
high
                            //pg DIO high
                            SMB_Byte_
WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x11,0xff);

                            gotoxy(1,8);
                            printf("GP1x Status:
LED OFF\n");

                        }
                        else
                        {
                            GP1xVal = 0; //DIO all
low
                            //pg DIO LOW
                            SMB_Byte_
WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x11,0x00);

                            gotoxy(1,8);
                            printf("GP1x Status:
LED ON \n");
                        }
                    break;
                case 'S':
                case 's':
                        if (GP2xVal == 0)
                        {
                            GP2xVal = 1; //DIO all
high
                            //pg DIO high
                            SMB_Byte_
WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x21,0xff);
```

```
                                                                            gotoxy(1,9);
                                                                            printf("GP2x Status:
LED OFF\n");

                                                                  }
                                                                  else
                                                                  {
                                                                            GP2xVal = 0; //DIO all
low
                                                                            //pg DIO LOW
                                                                            SMB_Byte_
WRITE(SMB_PORT_AD,SMB_DEVICE_ADD,0x21,0x00);

                                                                            gotoxy(1,9);
                                                                            printf("GP2x Status:
LED ON \n");
                                                                  }
                                                        break;
                                              default:
                                                        break;
                                    };
                                    //-printf( "Input: [%c]      ", getkey);  //DEBUG
                          };
          }while (getkey != 27); //ESC ascii==27
          //pg all DIO as Input
}


unsigned long Process_686C_Command_Write(unsigned long m_ECCMD, unsigned
long m_ECDATA)
{
 //-----------------------------------------------------------------------
 int     i,temp;
 unsigned long m_OutBuf;
 //-----------------------------------------------------------------------
 m_OutBuf=inportb(0x6C);
 if ( ( m_OutBuf&0x00000003) > 0 )
   {
     // temp=inportb(0x68);
     return 0xFFFFFFFF;
    }

 outport(0x6C,m_ECCMD);
 for ( i=0; i<=4000; i++ )
 {
    m_OutBuf=inportb(0x6C);
    if ( ( m_OutBuf&0x00000002) == 0 )  break;
 }
  if ( i < 3999 )
     {
       outport(0x68,m_ECDATA);
        for ( i=0; i<=4000; i++ )
        {
          m_OutBuf=inportb(0x6C);
          if ( ( m_OutBuf&0x00000002) == 0 )
                {    return 0x00000000;  }
         }
     }

 if ( i > 3999 )   m_OutBuf=inportb(0x68);
 return 0xFFFFFFFF;
}
//---------------------------------------------------------------------------
unsigned long Process_686C_Command_Read(unsigned long m_ECCMD )
{
int i,temp;
 unsigned long m_OutBuf,m_InBuf;
 m_OutBuf=inportb(0x6C);
if ( ( m_OutBuf&0x00000003) > 0 )
    {
       temp=inportb(0x68);
       return 0xFFFFFFFF;

     }
 m_InBuf = m_ECCMD;
 outport(0x6C,m_InBuf);
 for ( i=0; i<=3500; i++ )
 {
    m_OutBuf=inportb(0x6C);
   if ( ( m_OutBuf&0x00000001) > 0 )
    {
      temp=inportb(0x68);
      temp= (temp & 0x000000FF ) ;
      return temp;
     // break;
    }
 }
 if ( i > 3499 )
    {
    temp=inportb(0x68);
    return 0xFFFFFFFF;
     }
 return 0xFFFFFFFF;
 }

//-------------------------------------------------------------------------
 unsigned long ECU_Read_686C_RAM_BYTE( unsigned long ECUMemAddr )
{
  unsigned long uDATA1,uDATA2,ECRamAddrH,ECRamAddrL;
  ECRamAddrL=ECUMemAddr%256;   ECRamAddrH=ECUMemAddr/256;
  //
  uDATA1=Process_686C_Command_Write(0x000000A3, ECRamAddrH );
  if ( uDATA1==0xFFFFFFFF ) { return 0xFFFFFFFF; }
  //
  uDATA1=Process_686C_Command_Write(0x000000A2, ECRamAddrL );
  if ( uDATA1==0xFFFFFFFF ) { return 0xFFFFFFFF; }
  //
  uDATA1=Process_686C_Command_Read( 0x000000A4 );
  if ( uDATA1 > 0x000000FF ) { return 0xFFFFFFFF; }
  uDATA2=Process_686C_Command_Read( 0x000000A4 );
  if ( uDATA2 > 0x000000FF ) { return 0xFFFFFFFF; }
  if (uDATA1==uDATA2) return uDATA1;
  else return  0xFFFFFFFF;
}
//-------------------------------------------------------------------------
 unsigned long ECU_Write_686C_RAM_BYTE( unsigned long
ECUMemAddr,unsigned long ECUMemData )
{
  unsigned long uDATA, RD_DATA, ECRamAddrH, ECRamAddrL;
  ECRamAddrL=ECUMemAddr%256;   ECRamAddrH=ECUMemAddr/256;
  //
  uDATA=Process_686C_Command_Write(0x000000A3, ECRamAddrH );
  if ( uDATA==0xFFFFFFFF ) { return 0xFFFFFFFF;}
  //
  uDATA=Process_686C_Command_Write(0x000000A2, ECRamAddrL );
  if ( uDATA==0xFFFFFFFF ) { return 0xFFFFFFFF;}
  //
  uDATA=Process_686C_Command_Write(0x000000A5, ECUMemData );
  if ( uDATA==0xFFFFFFFF ) { return 0xFFFFFFFF;}
  //
  return  0x00000000;
}
//-------------------------------------------------------------------------
```

```
unsigned char SMB_Byte_READ(int SMPORT, int DeviceID, int REG_INDEX)
{
        unsigned char SMB_R;
        outportb(SMPORT+02, 0x00);      /* clear */
        outportb(SMPORT+00, 0xff);      /* clear */
        delay(10);
        outportb(SMPORT+04, DeviceID+1);        /* clear */
        outportb(SMPORT+03, REG_INDEX);         /* clear */
        outportb(SMPORT+02, 0x48);      /* read_byte */
        delay(10);
        //printf(“ %02x “,inportb(SMPORT+05));
        SMB_R= inportb(SMPORT+05);
        return SMB_R;
}

void 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);          /* clear */
        outportb(SMPORT+03, REG_INDEX);         /* clear */
        outportb(SMPORT+05, REG_DATA);          /* read_byte */
        outportb(SMPORT+02, 0x48);      /* read_byte */
/*      delay(10);
        printf(“ %02x “,inportb(SMPORT+05)); */
}
```

This page is intentionally left blank.