AXIOMTEK

CEM880

4th Generation Intel® Core™ Processor COM Express™ Type 6 Basic Module

User's Manual

RoHS

Disclaimers

This manual has been carefully checked and believed to contain accurate information. Axiomtek Co., Ltd. assumes no responsibility for any infringements of patents or any third party's rights, and any liability arising from such use.

Axiomtek does not warrant or assume any legal liability or responsibility for the accuracy, completeness or usefulness of any information in this document. Axiomtek does not make any commitment to update the information in this manual.

Axiomtek reserves the right to change or revise this document and/or product at any time without notice.

No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Axiomtek Co., Ltd.

CAUTION

If you replace wrong batteries, it causes the danger of explosion. It is recommended by the manufacturer that you follow the manufacturer's instructions to only replace the same or equivalent type of battery, and dispose of used ones.

December 2014, Version A2

Printed in Taiwan

ESD Precautions

Computer boards have integrated circuits sensitive to static electricity. To prevent chipsets from electrostatic discharge damage, please take care of the following jobs with precautions:

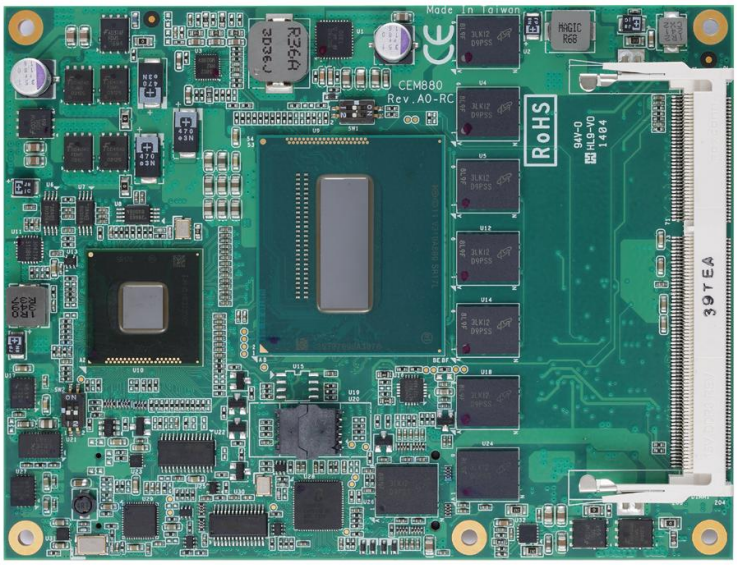
- Do not remove boards or integrated circuits from their anti-static packaging until you are ready to install them.
- Before holding the board or integrated circuit, touch an unpainted portion of the system unit chassis for a few seconds. It discharges static electricity from your body.
- Wear a wrist-grounding strap, available from most electronic component stores, when handling boards and components.

Trademarks Acknowledgments

Table of Contents

This page is intentionally left blank.

Chapter 1
Introduction



The CEM880 is a new COM Express™ Type 6 Basic Module to support BGA type quad/dual core 4th generation Intel® Core™ i7/ i5/ i3/ Celeron® processors. It integrates Intel® PCH QM87 chipset which supports the most updated high speed I/Os like PCI-Express Gen 3 at 8GT/s, SuperSpeed USB 3.0 at 5Gb/s, and SATA-600 at 6Gb/s. The CEM880 does fully comply with COM Express™ Type 6 specification. It provides 24 Lanes of PCI-Express, Gigabit Ethernet, HD audio interface, VGA/LVDS LCD and 3 configurable DDI for more flexible digital display options.

1.1 Features

- Intel® 4th generation Core™ i7/ i5/ i3/ Celeron® BGA processors
- Intel® PCH QM87
- 1st bank is onboard DDR3L with memory up to 4GB (optional). 2nd bank is one 204-pin DDR3L SO-DIMM supporting up to 8GB memory capacity.
- Support 24 Lanes of PCI-Express – 8 Lanes support Gen 2 at 5GT/s and 16 Lanes support Gen 3 at 8GT/s (Lane 8 is occupied by Intel® Giga LAN).
- 4 SATA-600
- 8 USB 2.0 ports
- 4 USB 3.0 ports
- TPM v1.2

1.2 Specifications

- **CPU**
  - Intel® 4th generation Core™ i7/ i5/ i3/ Celeron® BGA processors.

- **Chipset**
  - Intel® QM87 Express chipset.

- **BIOS**
  - American Megatrends Inc. BIOS.
  - 64Mbit SPI Flash, DMI, Plug and Play.
  - RPL/PXE Ethernet Boot ROM, customized default saving features, LPC-free supported, uses SPI type Flash memory.

- **System Memory**
  - One 204-pin DDR3L 1333/1066MHz SO-DIMM slot with maximum memory capacity up to 8GB.
  - Onboard DDR3L 1333/1066MHz memory supports maximum capacity up to 4GB (optional).

- **TPM**
  - Trusted Platform Module compatible with TPM1.2 Main and PC Client specification based on Intel LPC Bus Interface.

- **Expansion Interface**
  - One PCI-Express x16 (Gen 3) for discrete graphics or general purpose PCI-Express (2 x8 or 1 x8 with 2 x4).
  - Seven PCI-Express x1 (Lanes 1/2/3/4/5/6/7 are free); Lane 8 is occupied by GbE.

- **USB Interface**
  - Eight USB ports comply with USB Spec. Rev. 2.0.
  - Four USB ports comply with USB Spec. Rev. 3.0.

- **SATA Interface**
  - Four SATA 6Gb/s ports supported through COM Express™ connector.

- **Graphics**
  - Integrated in processor HD graphics 4600 max. frequency up to 1GHz.
  - CRT interface supports up to 1920x1200.
  - 18/24-bit dual channel LVDS interface.
  - Three DDI ports support HDMI/DVI/DisplayPort.

- **Ethernet**
  - One 1000/100/10 Base-T provided by Intel® I217 with integrated boot ROM.

- **Audio**
  - Integrated on Intel® PCH QM87.

- **Power Management**
  - ACPI (Advanced Configuration and Power Interface).

- **Form Factor**
  - Basic module 125mm x 95mm.

1.3 Utilities Supported

- Intel® QM87 utility and driver
- Graphics driver
- Ethernet utility and driver
- ME driver
- USB 3.0 driver

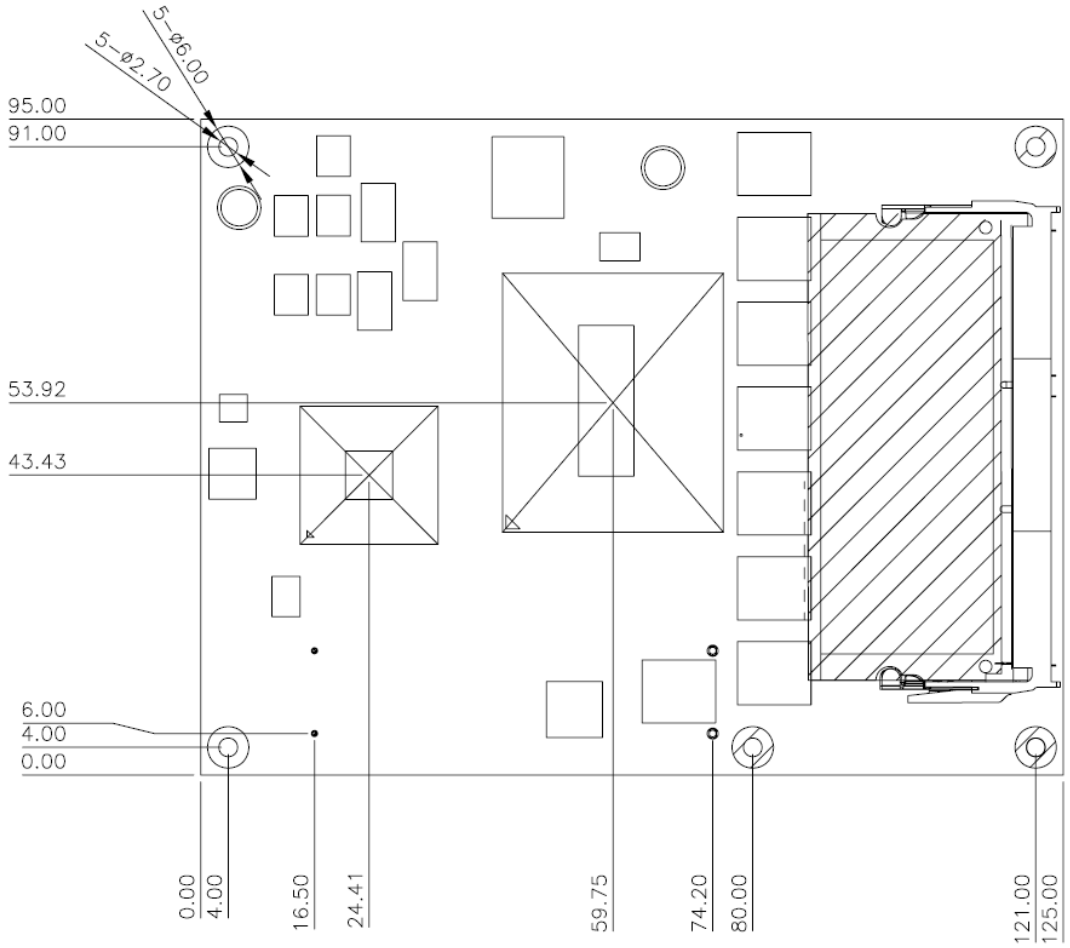
Note

All specifications and images are subject to change without notice.

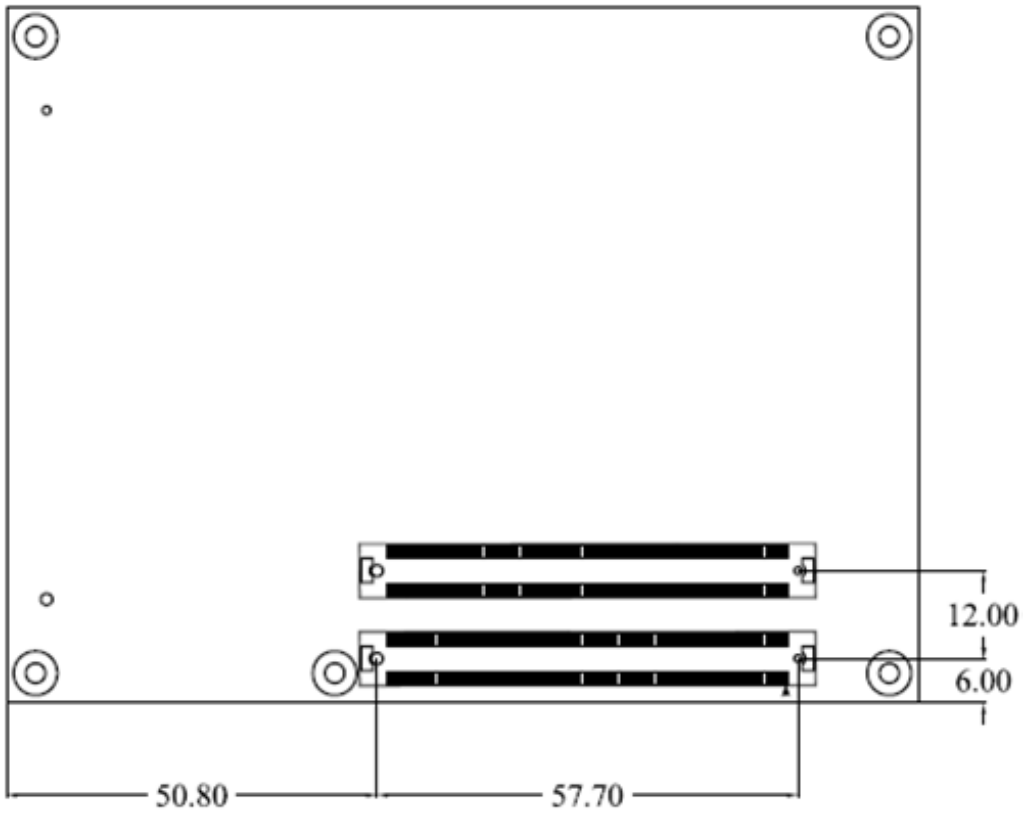
This page is intentionally left blank.

Chapter 2
Module and Pin Assignments

2.1 Module Dimensions and Fixing Holes

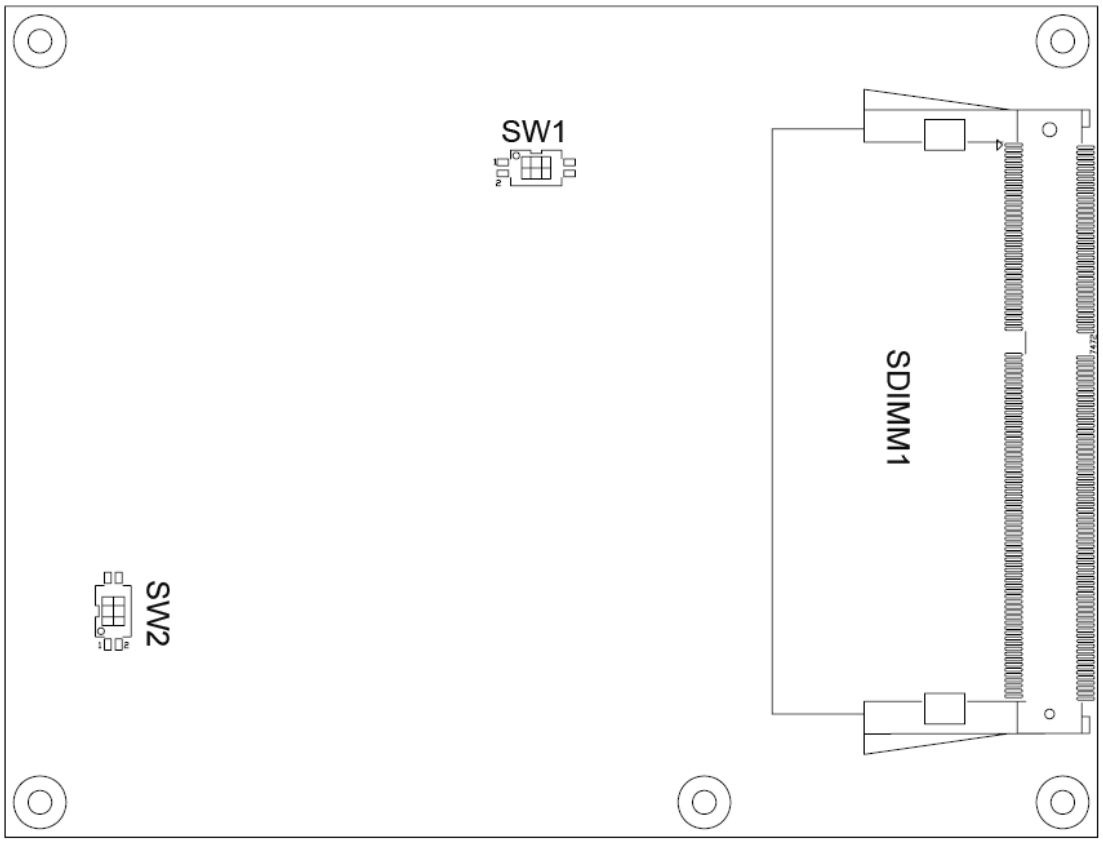


Top View

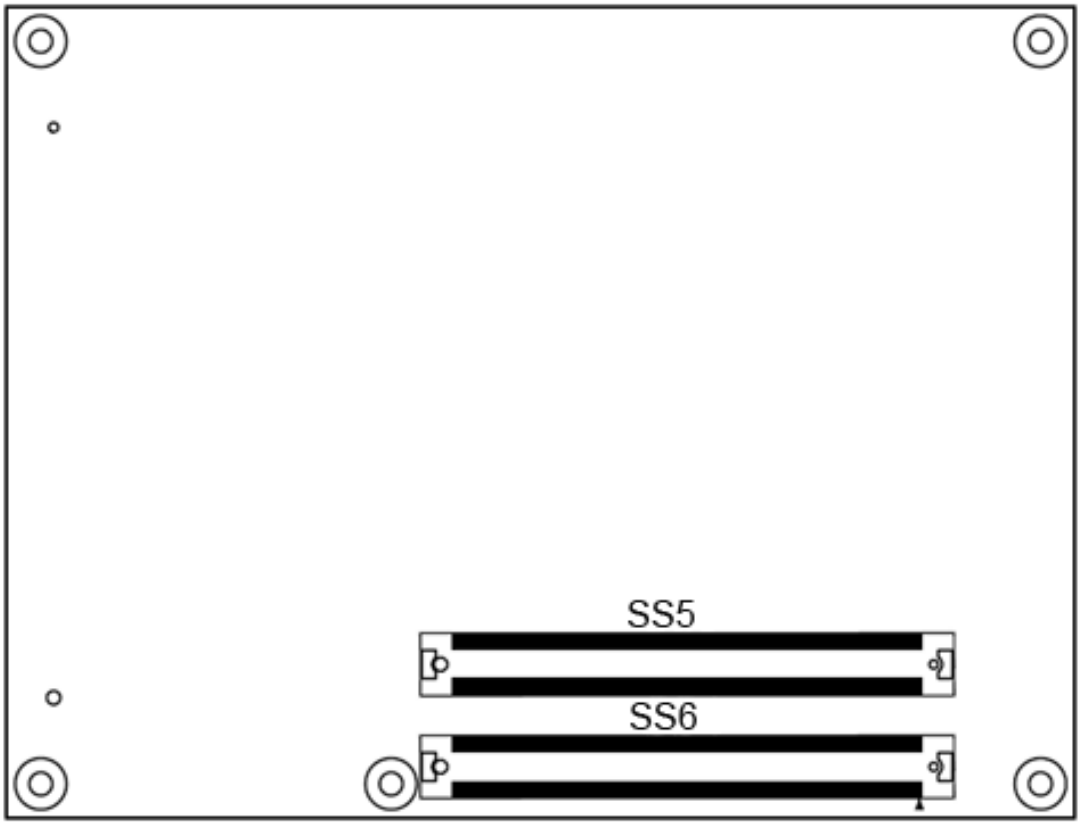


Bottom View

2.2 Module Layout



Top View



Bottom View

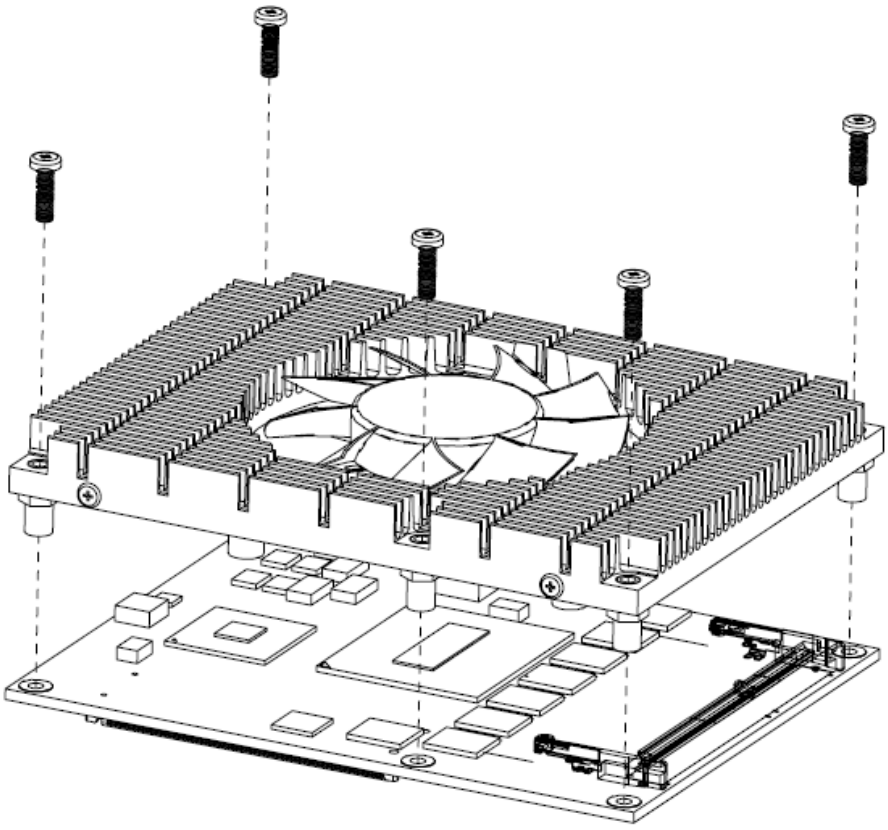
2.3 Installing Heatsink

For thermal dissipation, a heatsink enables the CEM880's components to dissipate heat efficiently. All heat generating components are thermally conducted to the heatsink in order to avoid hot spots. Below images illustrate how to install the heat spreader.

1. There is a protective plastic covering on the thermal pads. This must be removed before the heatsinkr can be mounted.

2. Each heatsink is designed for a specific CEM module. The thermal pads on the heatsink are designed to make contact with the necessary components on the CEM module. When mounting the heatsink you must make sure that the thermal pads on the heatsink make complete contact (no space between thermal pad and component) with the corresponding components on the CEM module. This is especially critical for CEM modules that have higher CPU speeds (for example 1.0GHz or more) to ensure that the heatsink acts as a proper thermal interface for cooling solutions.

3. This CPU module has five assembly holes for installing heat spreader plate. Use the five screws to secure the heat spreader plate to the CEM880. Be careful not to over-tighten the screws.



2.4 Switch Settings

Properly configure switch settings on the CEM880 to meet your application purpose. Below you can find a summary table of all switches and onboard default settings.

Note

Once the default switch setting needs to be changed, please do it under power-off condition.

Switch	Description	Setting
SW1	PCI-Express Bifurcation Setting Default: One x16 PCI-Express	SW1-1 OFF, SW1-2 OFF
SW2	Auto Power On Default: Disable	SW2-1 OFF
	Restore BIOS Optimal Defaults Default: Normal Operation	SW2-2 OFF

2.4.1 PCI-Express Bifurcation Setting (SW1)

The SW1 is for PCI-Express bifurcation setting. See table below for detailed information.

Function	Setting
Select one x8 and two x4 PCI-Express	SW1-1 ON, SW1-2 ON
Select two x8 PCI-Express	SW1-1 ON, SW1-2 OFF
Reserved	SW1-1 OFF, SW1-2 ON
Select one x16 PCI-Express (Default)	SW1-1 OFF, SW1-2 OFF

2.4.2 Auto Power On and Restore BIOS Optimal Defaults (SW2)

If dip1 of SW2 (SW2-1) is enabled for power input, the system will be automatically power on without pressing soft power button. If this jumper is disabled for power input, it is necessary to manually press soft power button to power on the system.

The dip2 of SW2 (SW2-2) is for restoring BIOS default status. Flip SW2-2 to ON position for a few seconds then flip it back to OFF position. Doing this procedure can restore BIOS optimal defaults.

Function	Setting
Disable auto power on (Default)	SW2-1 OFF
Enable auto power on	SW2-1 ON
Normal operation (Default)	SW2-2 OFF
Restore BIOS optimal defaults	SW2-2 ON

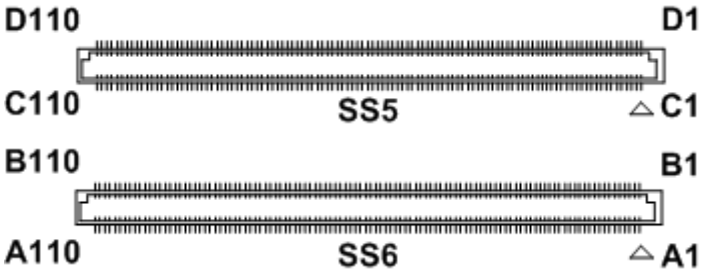
2.5 Connectors

Signals go to the other parts of the system through connectors. Loose or improper connection might cause problems, please make sure all connectors are properly and firmly connected. Here is a summary table which shows connectors on the hardware.

Connector	Description
SDIMM1	DDR3L SO-DIMM Connector
SS5	COM Express™ Connector
SS6	COM Express™ Connector

2.5.1 COM Express™ Connectors (SS5 and SS6)

Below table shows the pin assignments for the 220-pin COM Express™ connectors.



Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	GND (FIXED)	B1	GND (FIXED)	C1	GND (FIXED)	D1	GND (FIXED)
A2	GBE0_MDI3-	B2	GBE0_ACT#	C2	GND (FIXED)	D2	GND (FIXED)
A3	GBE0_MDI3+	B3	LPC_FRAME#	C3	USB_SSRX0-	D3	USB_SSTX0-
A4	GBE0_LINK100#	B4	LPC_AD0	C4	USB_SSRX0+	D4	USB_SSTX0+
A5	GBE0_LINK1000#	B5	LPC_AD1	C5	GND (FIXED)	D5	GND (FIXED)
A6	GBE0_MDI2-	B6	LPC_AD2	C6	USB_SSRX1-	D6	USB_SSTX1-
A7	GBE0_MDI2+	B7	LPC_AD3	C7	USB_SSRX1+	D7	USB_SSTX1+
A8	GBE0_LINK#	B8	LPC_DRQ0#	C8	GND (FIXED)	D8	GND (FIXED)
A9	GBE0_MDI1-	B9	LPC_DRQ1#	C9	USB_SSRX2-	D9	USB_SSTX2-
A10	GBE0_MDI1+	B10	LPC_CLK	C10	USB_SSRX2+	D10	USB_SSTX2+
A11	GND (FIXED)	B11	GND (FIXED)	C11	GND (FIXED)	D11	GND (FIXED)
A12	GBE0_MDI0-	B12	PWRBTN#	C12	USB_SSRX3-	D12	USB_SSTX3-
A13	GBE0_MDI0+	B13	SMB_CK	C13	USB_SSRX3+	D13	USB_SSTX3+
A14	GBE0_CTREF	B14	SMB_DAT	C14	GND (FIXED)	D14	GND (FIXED)
A15	SUS_S3#	B15	SMB_ALERT#	C15	N.C.	D15	DDI1_CTRLCLK_AUX+
A16	SATA0_TX+	B16	SATA1_TX+	C16	N.C.	D16	DDI1_CTRLDATA_AUX-
A17	SATA0_TX-	B17	SATA1_TX-	C17	N.C.	D17	N.C.
A18	SUS_S4#	B18	SUS_STAT#	C18	N.C.	D18	N.C.
A19	SATA0_RX+	B19	SATA1_RX+	C19	PCIE_RX6+	D19	PCIE_TX6+
A20	SATA0_RX-	B20	SATA1_RX-	C20	PCIE_RX6-	D20	PCIE_TX6-
A21	GND (FIXED)	B21	GND (FIXED)	C21	GND (FIXED)	D21	GND (FIXED)
A22	SATA2_TX+	B22	SATA3_TX+	C22	N.C.	D22	N.C.
A23	SATA2_TX-	B23	SATA3_TX-	C23	N.C.	D23	N.C.
A24	SUS_S5#	B24	N.C.	C24	DDI1_HPD	D24	N.C.
A25	SATA2_RX+	B25	SATA3_RX+	C25	N.C.	D25	N.C.
A26	SATA2_RX-	B26	SATA3_RX-	C26	N.C.	D26	DDI1_PAIR0+
A27	BATLOW#	B27	WDT	C27	N.C.	D27	DDI1_PAIR0-
A28	(S)ATA_ACT#	B28	AC/HDA_SDIN2	C28	N.C.	D28	N.C.
A29	AC/HDA_SYNC	B29	AC/HDA_SDIN1	C29	N.C.	D29	DDI1_PAIR1+
A30	AC/HDA_RST#	B30	AC/HDA_SDIN0	C30	N.C.	D30	DDI1_PAIR1-
A31	GND (FIXED)	B31	GND (FIXED)	C31	GND (FIXED)	D31	GND (FIXED)
A32	AC/HDA_BITCLK	B32	SPKR	C32	DDI2_CTRLCLK_AUX+	D32	DDI1_PAIR2+
A33	AC/HDA_SDOUT	B33	N.C.	C33	DDI2_CTRLDATA_AUX-	D33	DDI1_PAIR2-
A34	BIOS_DISABLE#	B34	N.C.	C34	DDI2_DDC_AUX_SEL	D34	DDI1_DDC_AUX_SEL
A35	THRMTRIP#	B35	THRM#	C35	N.C.	D35	N.C.
A36	USB6-	B36	USB7-	C36	DDI3_CTRLCLK_AUX+	D36	DDI1_PAIR3+
A37	USB6+	B37	USB7+	C37	DDI3_CTRLDATA_AUX-	D37	DDI1_PAIR3-
A38	USB_6_7_OC#	B38	USB_4_5_OC#	C38	DDI3_DDC_AUX_SEL	D38	N.C.
A39	USB4-	B39	USB5-	C39	DDI3_PAIR0+	D39	DDI2_PAIR0+
A40	USB4+	B40	USB5+	C40	DDI3_PAIR0-	D40	DDI2_PAIR0-
A41	GND (FIXED)	B41	GND (FIXED)	C41	GND (FIXED)	D41	GND (FIXED)
A42	USB2-	B42	USB3-	C42	DDI3_PAIR1+	D42	DDI2_PAIR1+
A43	USB2+	B43	USB3+	C43	DDI3_PAIR1-	D43	DDI2_PAIR1-
A44	USB_2_3_OC#	B44	USB_0_1_OC#	C44	DDI3_HPD	D44	DDI2_HPD
A45	USB0-	B45	USB1-	C45	N.C.	D45	N.C.
A46	USB0+	B46	USB1+	C46	DDI3_PAIR2+	D46	DDI2_PAIR2+
A47	VCC_RTC	B47	EXCD1_PERST#	C47	DDI3_PAIR2-	D47	DDI2_PAIR2-
A48	EXCD0_PERST#	B48	EXCD1_CPPE#	C48	N.C.	D48	N.C.
A49	EXCD0_CPPE#	B49	SYS_RESET#	C49	DDI3_PAIR3+	D49	DDI2_PAIR3+
A50	LPC_SERIRQ	B50	CB_RESET#	C50	DDI3_PAIR3-	D50	DDI2_PAIR3-
A51	GND (FIXED)	B51	GND (FIXED)	C51	GND (FIXED)	D51	GND (FIXED)
A52	PCIE_TX5+	B52	PCIE_RX5+	C52	PEG_RX0+	D52	PEG_TX0+
A53	PCIE_TX5-	B53	PCIE_RX5-	C53	PEG_RX0-	D53	PEG_TX0-
A54	GPI0	B54	GPO1	C54	TYPE0#	D54	PEG_LANE_RV#
A55	PCIE_TX4+	B55	PCIE_RX4+	C55	PEG_RX1+	D55	PEG_TX1+

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A56	PCIE_TX4-	B56	PCIE_RX4-	C56	PEG_RX1-	D56	PEG_TX1-
A57	GND	B57	GPO2	C57	TYPE1#	D57	TYPE2#
A58	PCIE_TX3+	B58	PCIE_RX3+	C58	PEG_RX2+	D58	PEG_TX2+
A59	PCIE_TX3-	B59	PCIE_RX3-	C59	PEG_RX2-	D59	PEG_TX2-
A60	GND (FIXED)	B60	GND (FIXED)	C60	GND (FIXED)	D60	GND (FIXED)
A61	PCIE_TX2+	B61	PCIE_RX2+	C61	PEG_RX3+	D61	PEG_TX3+
A62	PCIE_TX2-	B62	PCIE_RX2-	C62	PEG_RX3-	D62	PEG_TX3-
A63	GPI1	B63	GPO3	C63	N.C.	D63	N.C.
A64	PCIE_TX1+	B64	PCIE_RX1+	C64	N.C.	D64	N.C.
A65	PCIE_TX1-	B65	PCIE_RX1-	C65	PEG_RX4+	D65	PEG_TX4+
A66	GND	B66	WAKE0#	C66	PEG_RX4-	D66	PEG_TX4-
A67	GPI2	B67	WAKE1#	C67	N.C.	D67	GND
A68	PCIE_TX0+	B68	PCIE_RX0+	C68	PEG_RX5+	D68	PEG_TX5+
A69	PCIE_TX0-	B69	PCIE_RX0-	C69	PEG_RX5-	D69	PEG_TX5-
A70	GND(FIXED)	B70	GND(FIXED)	C70	GND(FIXED)	D70	GND(FIXED)
A71	LVDS_A0+	B71	LVDS_B0+	C71	PEG_RX6+	D71	PEG_TX6+
A72	LVDS_A0-	B72	LVDS_B0-	C72	PEG_RX6-	D72	PEG_TX6-
A73	LVDS_A1+	B73	LVDS_B1+	C73	GND(FIXED)	D73	SDVO_CLK
A74	LVDS_A1-	B74	LVDS_B1-	C74	PEG_RX7+	D74	PEG_TX7+
A75	LVDS_A2+	B75	LVDS_B2+	C75	PEG_RX7-	D75	PEG_TX7-
A76	LVDS_A2-	B76	LVDS_B2-	C76	GND	D76	GND
A77	LVDS_VDD_EN	B77	LVDS_B3+	C77	N.C.	D77	N.C.
A78	LVDS_A3+	B78	LVDS_B3-	C78	PEG_RX8+	D78	PEG_TX8+
A79	LVDS_A3-	B79	LVDS_BKLT_EN	C79	PEG_RX8-	D79	PEG_TX8-
A80	GND(FIXED)	B80	GND(FIXED)	C80	GND(FIXED)	D80	GND(FIXED)
A81	LVDS_A_CK+	B81	LVDS_B_CK+	C81	PEG_RX9+	D81	PEG_TX9+
A82	LVDS_A_CK-	B82	LVDS_B_CK-	C82	PEG_RX9-	D82	PEG_TX9-
A83	LVDS_I2C_CK	B83	LVDS_BKLT_CTRL	C83	N.C.	D83	N.C.
A84	LVDS_I2C_DAT	B84	VCC_5V_SBY	C84	GND	D84	GND
A85	GPI3	B85	VCC_5V_SBY	C85	PEG_RX10+	D85	PEG_TX10+
A86	N.C.	B86	VCC_5V_SBY	C86	PEG_RX10-	D86	PEG_TX10-
A87	N.C.	B87	VCC_5V_SBY	C87	GND	D87	GND
A88	PCIE0_CK_REF+	B88	BIOS_DIS1	C88	PEG_RX11+	D88	PEG_TX11+
A89	PCIE0_CK_REF-	B89	VGA_RED	C89	PEG_RX11-	D89	PEG_TX11-
A90	GND (FIXED)	B90	GND (FIXED)	C90	GND (FIXED)	D90	GND (FIXED)
A91	SPI_POWER	B91	VGA_GRN	C91	PEG_RX12+	D91	PEG_TX12+
A92	SPI_MISO	B92	VGA_BLU	C92	PEG_RX12-	D92	PEG_TX12-
A93	GPO0	B93	VGA_HSYNC	C93	GND	D93	GND
A94	SPI_CLK	B94	VGA_VSYNC	C94	PEG_RX13+	D94	PEG_TX13+
A95	SPI_MOSI	B95	VGA_I2C_CK	C95	PEG_RX13-	D95	PEG_TX13-
A96	TPM_PP	B96	VGA_I2C_DAT	C96	GND	D96	GND
A97	N.C.	B97	SPI_CS#	C97	N.C.	D97	RSVD
A98	N.C.	B98	N.C.	C98	PEG_RX14+	D98	PEG_TX14+
A99	N.C.	B99	N.C.	C99	PEG_RX14-	D99	PEG_TX14-
A100	GND (FIXED)	B100	GND (FIXED)	C100	GND (FIXED)	D100	GND (FIXED)
A101	N.C.	B101	FAN_PWMOUT	C101	PEG_RX15+	D101	PEG_TX15+
A102	N.C.	B102	FAN_TACHIN	C102	PEG_RX15-	D102	PEG_TX15-
A103	LID#	B103	SLEEP#	C103	GND	D103	GND
A104	VCC_12V	B104	VCC_12V	C104	VCC_12V	D104	VCC_12V
A105	VCC_12V	B105	VCC_12V	C105	VCC_12V	D105	VCC_12V
A106	VCC_12V	B106	VCC_12V	C106	VCC_12V	D106	VCC_12V
A107	VCC_12V	B107	VCC_12V	C107	VCC_12V	D107	VCC_12V
A108	VCC_12V	B108	VCC_12V	C108	VCC_12V	D108	VCC_12V
A109	VCC_12V	B109	VCC_12V	C109	VCC_12V	D109	VCC_12V
A110	GND (FIXED)	B110	GND (FIXED)	C110	GND (FIXED)	D110	GND (FIXED)

This page is intentionally left blank.

Chapter 3
Hardware Description

3.1 Microprocessor

The CEM880 supports Intel® Core™ i7/ i5/ i3 and Celeron® processors, which enables your system to operate under Windows® 7, Windows® 8 and Linux environments. The system performance depends on the microprocessor. You must install the heatsink or cooler carefully and properly to prevent damage.

3.2 BIOS

The CEM880 uses AMI Plug and Play BIOS with a single 64Mbit SPI Flash.

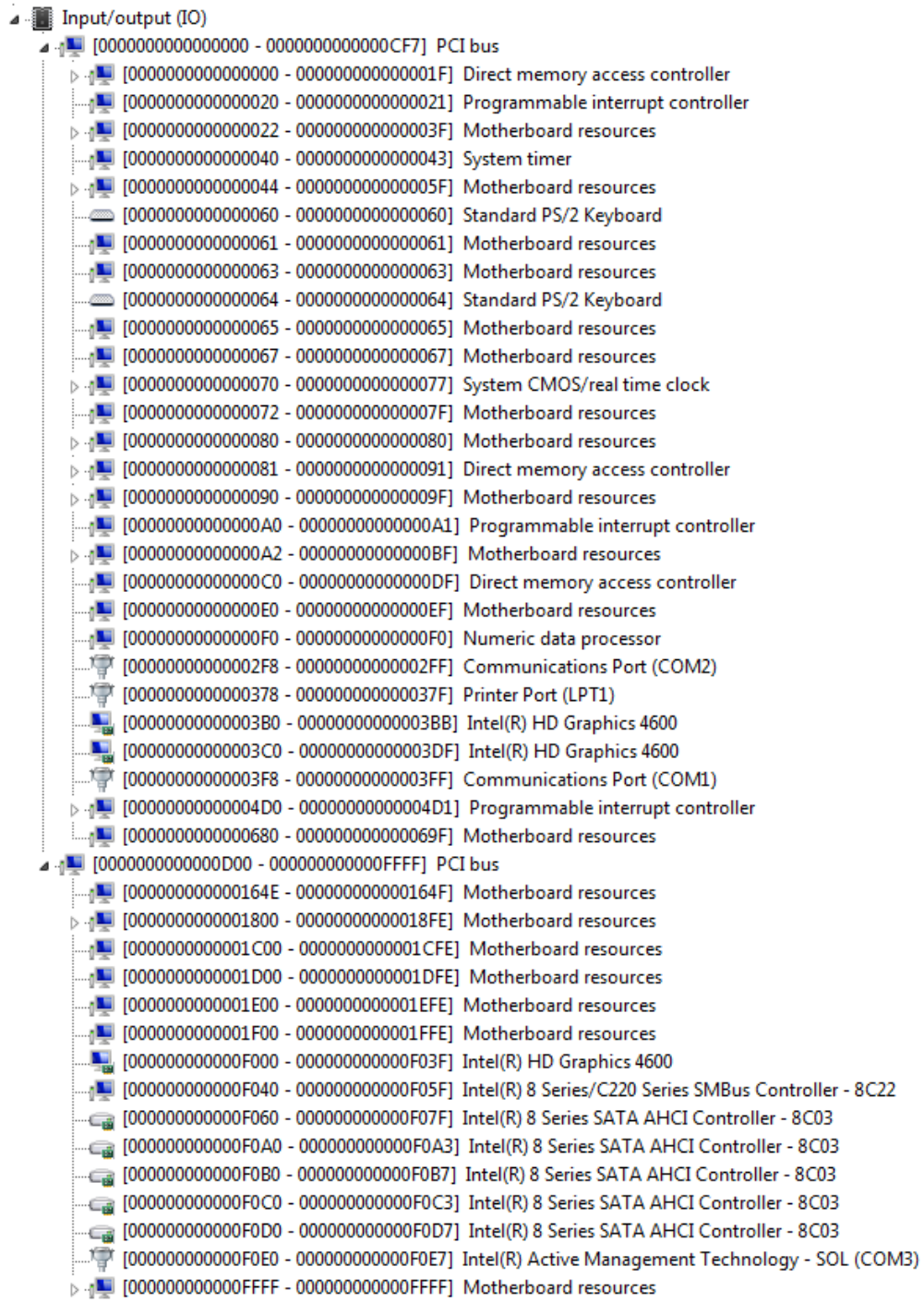
3.3 System Memory

The CEM880 supports DDR3L onboard memory which maximum capacity can be up to 4GB (optional) and one 204-pin DDR3L 1333/1066MHz SO-DIMM socket for maximum memory capacity up to 8GB DDR3L SDRAMs. The memory module can come in sizes of 1GB, 2GB, 4GB and 8GB.

3.4 I/O Port Address Map

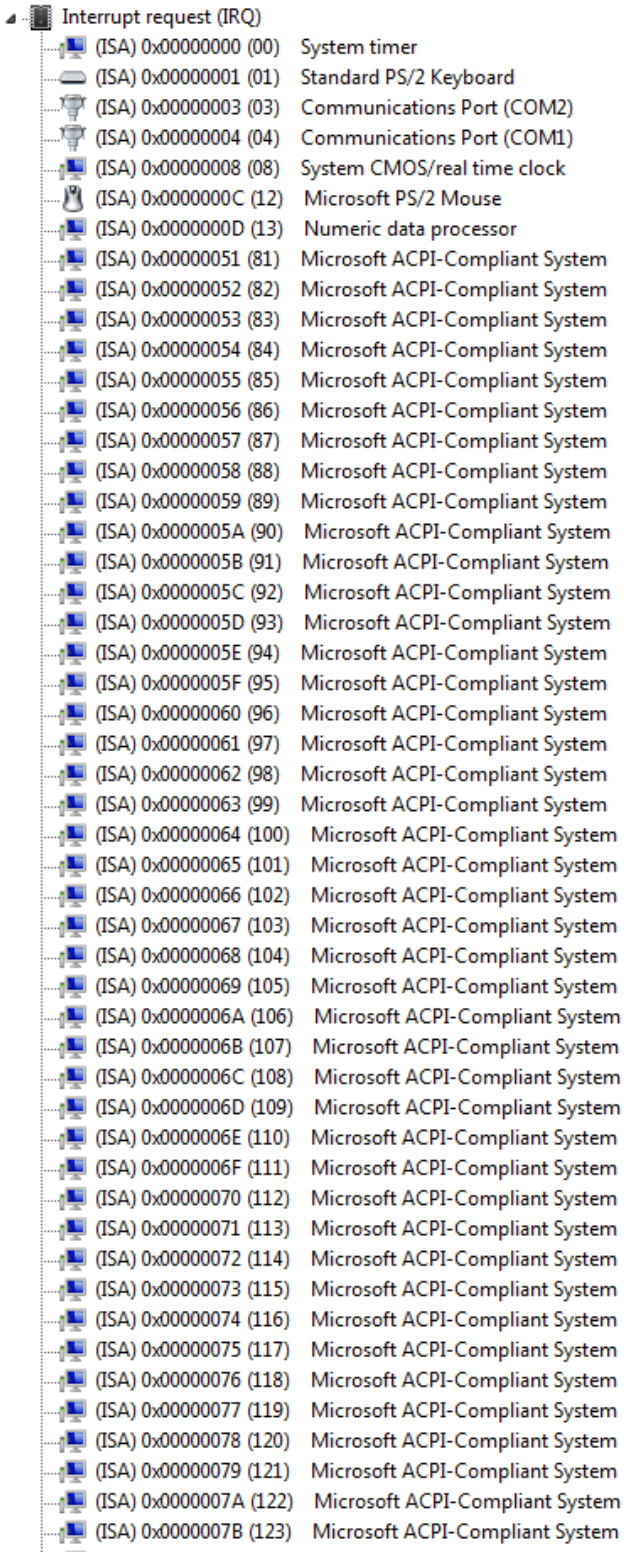
The Intel® Core™ i7/i5/i3 and Celeron® processors communicate via I/O ports. Total 1KB port addresses are available for assigning to other devices via I/O expansion cards.

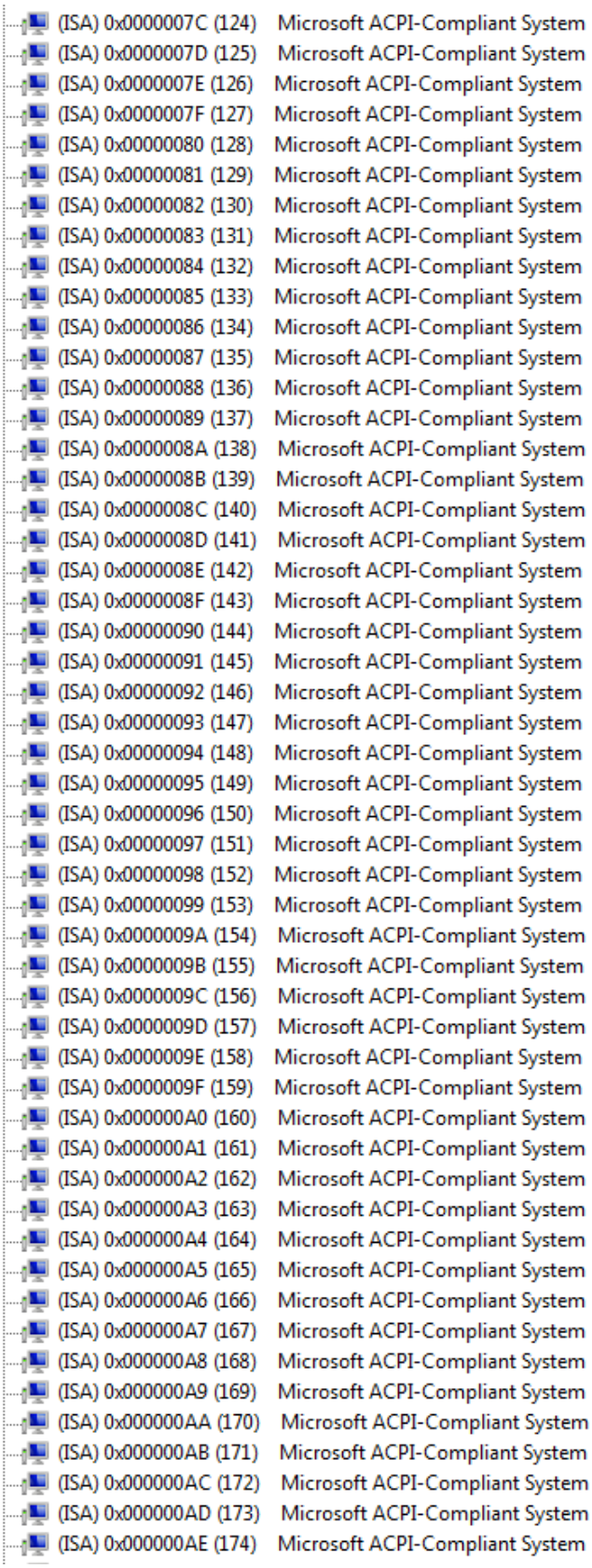
The I/O port addresses (with CEB94006 baseboard under Windows® 7) are as follows:

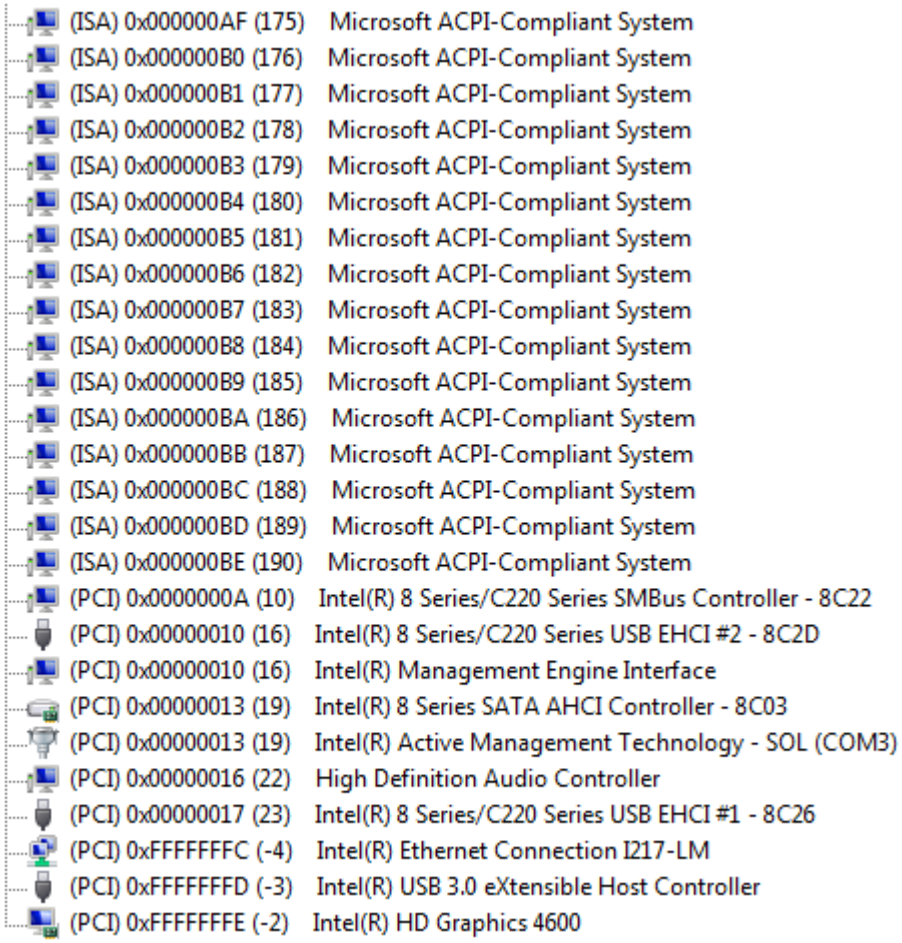


3.5 Interrupt Controller (IRQ) Map

The interrupt controller (IRQ) mapping list (with CEB94006 baseboard under Windows® 7) is shown as follows:

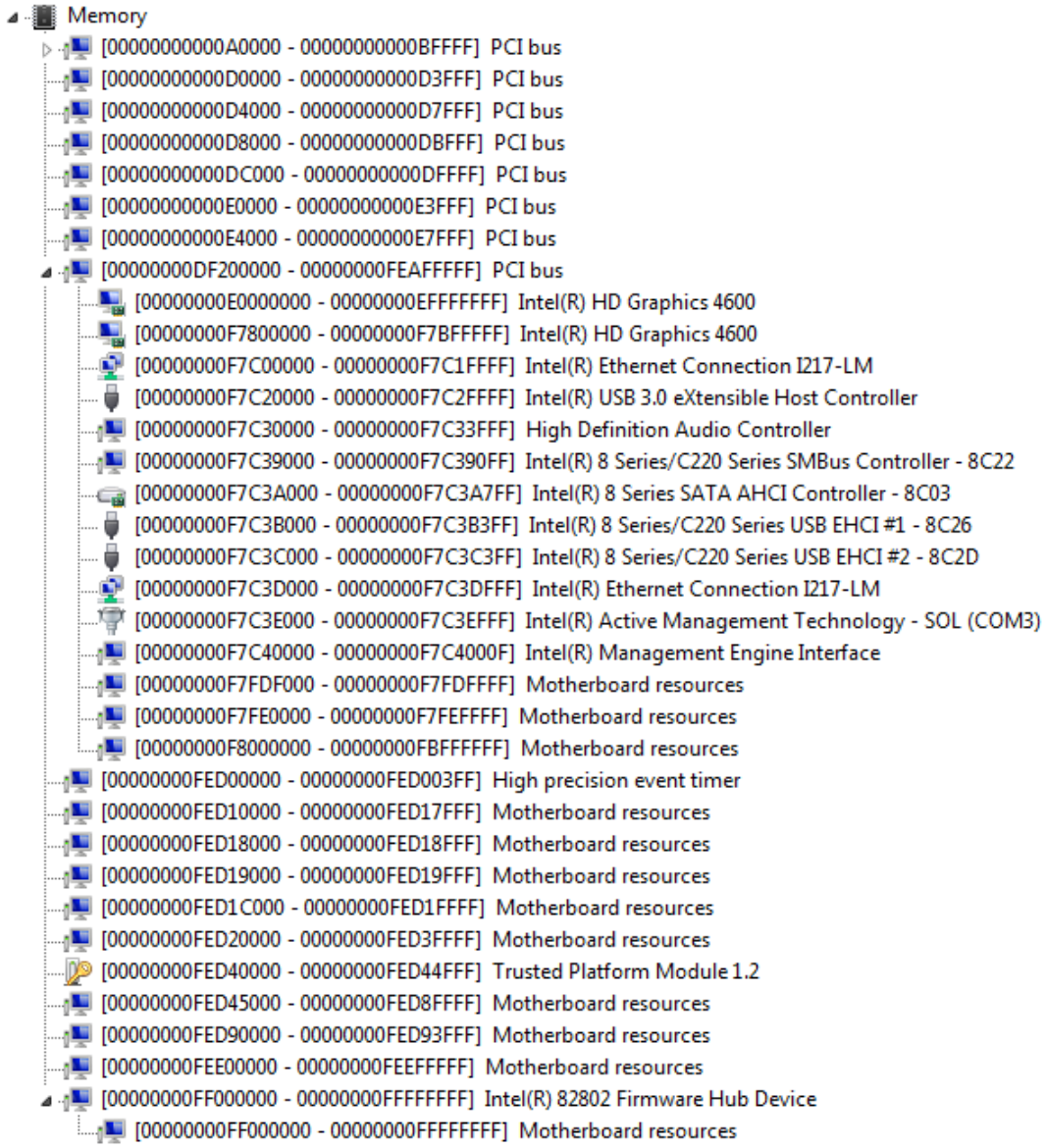



(ISA) 0x0000007C (124) Microsoft ACPI-Compliant System
(ISA) 0x0000007D (125) Microsoft ACPI-Compliant System
(ISA) 0x0000007E (126) Microsoft ACPI-Compliant System
(ISA) 0x0000007F (127) Microsoft ACPI-Compliant System
(ISA) 0x00000080 (128) Microsoft ACPI-Compliant System
(ISA) 0x00000081 (129) Microsoft ACPI-Compliant System
(ISA) 0x00000082 (130) Microsoft ACPI-Compliant System
(ISA) 0x00000083 (131) Microsoft ACPI-Compliant System
(ISA) 0x00000084 (132) Microsoft ACPI-Compliant System
(ISA) 0x00000085 (133) Microsoft ACPI-Compliant System
(ISA) 0x00000086 (134) Microsoft ACPI-Compliant System
(ISA) 0x00000087 (135) Microsoft ACPI-Compliant System
(ISA) 0x00000088 (136) Microsoft ACPI-Compliant System
(ISA) 0x00000089 (137) Microsoft ACPI-Compliant System
(ISA) 0x0000008A (138) Microsoft ACPI-Compliant System
(ISA) 0x0000008B (139) Microsoft ACPI-Compliant System
(ISA) 0x0000008C (140) Microsoft ACPI-Compliant System
(ISA) 0x0000008D (141) Microsoft ACPI-Compliant System
(ISA) 0x0000008E (142) Microsoft ACPI-Compliant System
(ISA) 0x0000008F (143) Microsoft ACPI-Compliant System
(ISA) 0x00000090 (144) Microsoft ACPI-Compliant System
(ISA) 0x00000091 (145) Microsoft ACPI-Compliant System
(ISA) 0x00000092 (146) Microsoft ACPI-Compliant System
(ISA) 0x00000093 (147) Microsoft ACPI-Compliant System
(ISA) 0x00000094 (148) Microsoft ACPI-Compliant System
(ISA) 0x00000095 (149) Microsoft ACPI-Compliant System
(ISA) 0x00000096 (150) Microsoft ACPI-Compliant System
(ISA) 0x00000097 (151) Microsoft ACPI-Compliant System
(ISA) 0x00000098 (152) Microsoft ACPI-Compliant System
(ISA) 0x00000099 (153) Microsoft ACPI-Compliant System
(ISA) 0x0000009A (154) Microsoft ACPI-Compliant System
(ISA) 0x0000009B (155) Microsoft ACPI-Compliant System
(ISA) 0x0000009C (156) Microsoft ACPI-Compliant System
(ISA) 0x0000009D (157) Microsoft ACPI-Compliant System
(ISA) 0x0000009E (158) Microsoft ACPI-Compliant System
(ISA) 0x0000009F (159) Microsoft ACPI-Compliant System
(ISA) 0x000000A0 (160) Microsoft ACPI-Compliant System
(ISA) 0x000000A1 (161) Microsoft ACPI-Compliant System
(ISA) 0x000000A2 (162) Microsoft ACPI-Compliant System
(ISA) 0x000000A3 (163) Microsoft ACPI-Compliant System
(ISA) 0x000000A4 (164) Microsoft ACPI-Compliant System
(ISA) 0x000000A5 (165) Microsoft ACPI-Compliant System
(ISA) 0x000000A6 (166) Microsoft ACPI-Compliant System
(ISA) 0x000000A7 (167) Microsoft ACPI-Compliant System
(ISA) 0x000000A8 (168) Microsoft ACPI-Compliant System
(ISA) 0x000000A9 (169) Microsoft ACPI-Compliant System
(ISA) 0x000000AA (170) Microsoft ACPI-Compliant System
(ISA) 0x000000AB (171) Microsoft ACPI-Compliant System
(ISA) 0x000000AC (172) Microsoft ACPI-Compliant System
(ISA) 0x000000AD (173) Microsoft ACPI-Compliant System
(ISA) 0x000000AE (174) Microsoft ACPI-Compliant System


(ISA) 0x000000AF (175) Microsoft ACPI-Compliant System
(ISA) 0x000000B0 (176) Microsoft ACPI-Compliant System
(ISA) 0x000000B1 (177) Microsoft ACPI-Compliant System
(ISA) 0x000000B2 (178) Microsoft ACPI-Compliant System
(ISA) 0x000000B3 (179) Microsoft ACPI-Compliant System
(ISA) 0x000000B4 (180) Microsoft ACPI-Compliant System
(ISA) 0x000000B5 (181) Microsoft ACPI-Compliant System
(ISA) 0x000000B6 (182) Microsoft ACPI-Compliant System
(ISA) 0x000000B7 (183) Microsoft ACPI-Compliant System
(ISA) 0x000000B8 (184) Microsoft ACPI-Compliant System
(ISA) 0x000000B9 (185) Microsoft ACPI-Compliant System
(ISA) 0x000000BA (186) Microsoft ACPI-Compliant System
(ISA) 0x000000BB (187) Microsoft ACPI-Compliant System
(ISA) 0x000000BC (188) Microsoft ACPI-Compliant System
(ISA) 0x000000BD (189) Microsoft ACPI-Compliant System
(ISA) 0x000000BE (190) Microsoft ACPI-Compliant System
(PCI) 0x0000000A (10) Intel(R) 8 Series/C220 Series SMBus Controller - 8C22
(PCI) 0x00000010 (16) Intel(R) 8 Series/C220 Series USB EHCI #2 - 8C2D
(PCI) 0x00000010 (16) Intel(R) Management Engine Interface
(PCI) 0x00000013 (19) Intel(R) 8 Series SATA AHCI Controller - 8C03
(PCI) 0x00000013 (19) Intel(R) Active Management Technology - SOL (COM3)
(PCI) 0x00000016 (22) High Definition Audio Controller
(PCI) 0x00000017 (23) Intel(R) 8 Series/C220 Series USB EHCI #1 - 8C26
(PCI) 0xFFFFFFFC (-4) Intel(R) Ethernet Connection I217-LM
(PCI) 0xFFFFFFFD (-3) Intel(R) USB 3.0 eXtensible Host Controller
(PCI) 0xFFFFFFFE (-2) Intel(R) HD Graphics 4600

3.6 Memory Map

The memory (with CEB94006 baseboard under Windows® 7) mapping list is shown as follows:



Chapter 4
AMI BIOS Setup Utility

The AMI UEFI BIOS provides users with a built-in setup program to modify basic system configuration. All configured parameters are stored in a flash chip to save the setup information whenever the power is turned off. This chapter provides users with detailed description about how to set up basic system configuration through the AMI BIOS setup utility.

4.1 Starting

To enter the setup screens, follow the steps below:

1. Turn on the computer and press the <Del> key immediately.
2. After you press the <Del> key, the main BIOS setup menu displays. You can access the other setup screens from the main BIOS setup menu, such as the Advanced and Chipset menus.

Note

If your computer cannot boot after making and saving system changes with BIOS setup, you can restore BIOS optimal defaults by setting SW2-2 (see section 2.4.2).

It is strongly recommended that you should avoid changing the chipset's defaults. Both AMI and your system manufacturer have carefully set up these defaults that provide the best performance and reliability.

4.2 Navigation Keys

The BIOS setup/utility uses a key-based navigation system called hot keys. Most of the BIOS setup utility hot keys can be used at any time during the setup navigation process. These keys include <F1>, <F2>, <Enter>, <ESC>, <Arrow> keys, and so on.

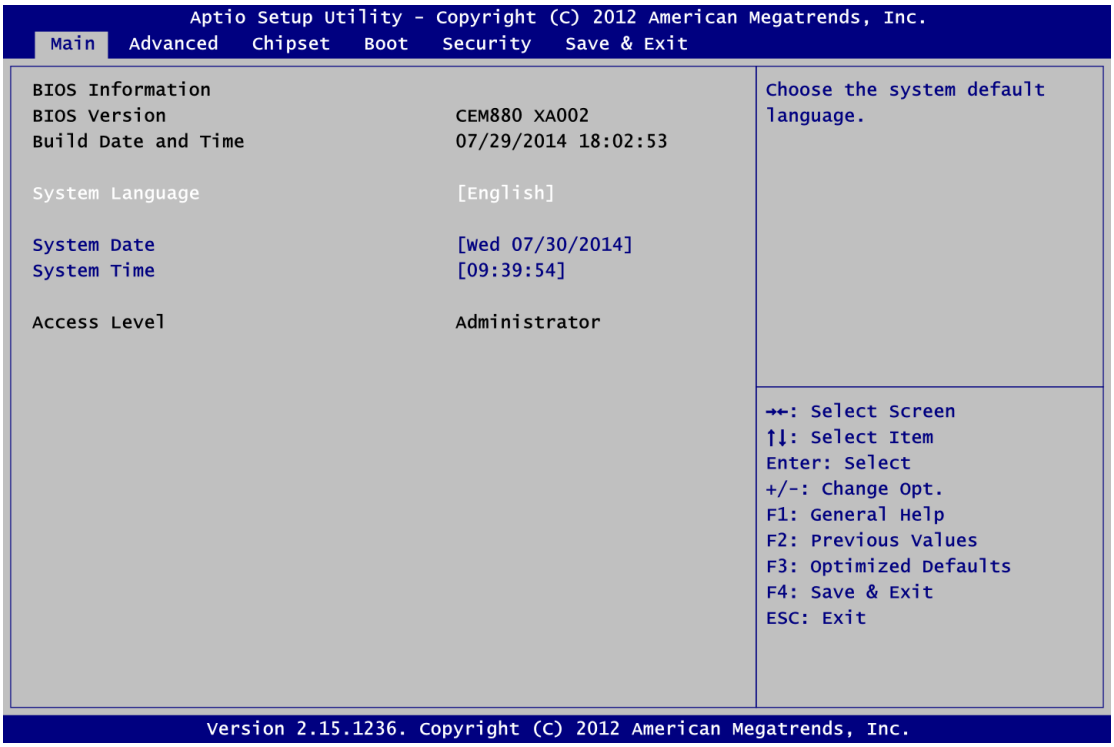
Note

Some of the navigation keys differ from one screen to another.

Hot Keys	Description
→← Left/Right	The Left and Right <Arrow> keys allow you to select a setup screen.
↑↓ Up/Down	The Up and Down <Arrow> keys allow you to select a setup screen or sub-screen.
+– Plus/Minus	The Plus and Minus <Arrow> keys allow you to change the field value of a particular setup item.
Tab	The <Tab> key allows you to select setup fields.
F1	The <F1> key allows you to display the General Help screen.
F2	The <F2> key allows you to Load Previous Values.
F3	The <F3> key allows you to Load Optimized Defaults.
F4	The <F4> key allows you to save any changes you have made and exit Setup. Press the <F4> key to save your changes.
Esc	The <Esc> key allows you to discard any changes you have made and exit the Setup. Press the <Esc> key to exit the setup without saving your changes.
Enter	The <Enter> key allows you to display or change the setup option listed for a particular setup item. The <Enter> key can also allow you to display the setup sub- screens.

4.3 Main Menu

When you first enter the setup utility, you will enter the Main setup screen. You can always return to the Main setup screen by selecting the Main tab. System Time/Date can be set up as described below. The Main BIOS setup screen is shown below.



BIOS Information
Display BIOS information.

System Language
Use this item to choose the system default language.

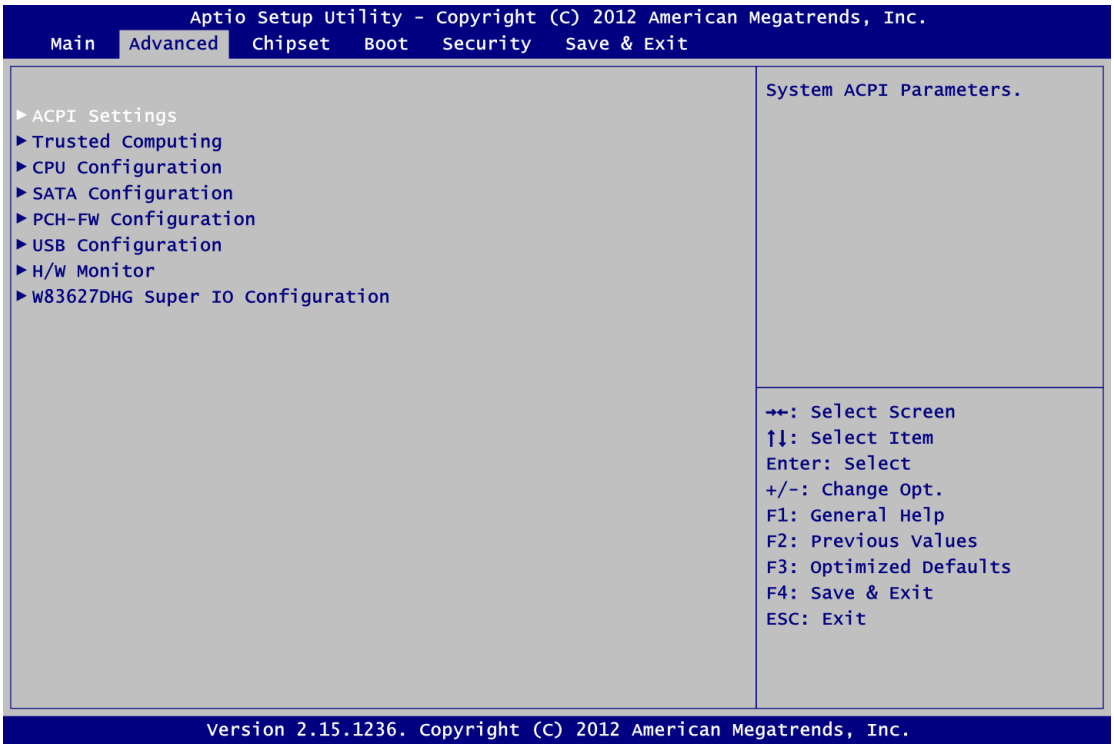
System Date/Time
Use this option to change the system time and date. Highlight System Time or System Date using the <Arrow> keys. Enter new values through the keyboard. Press the <Tab> key or the <Arrow> keys to move between fields. The date must be entered in MM/DD/YY format. The time is entered in HH:MM:SS format.

4.4 Advanced Menu

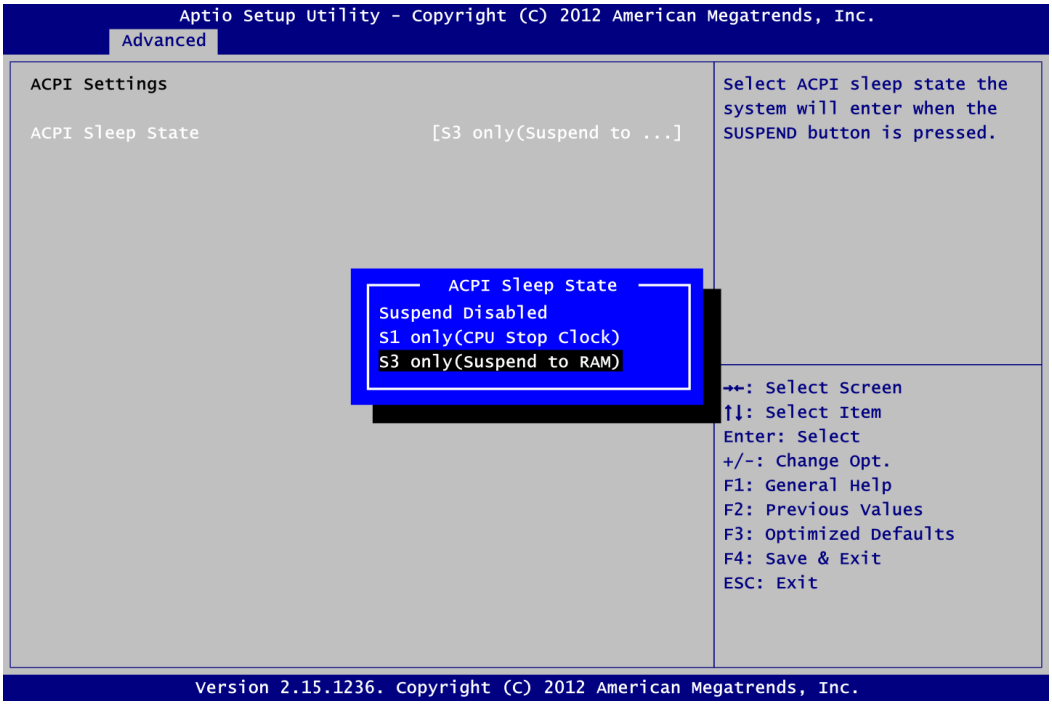
The Advanced menu also allows users to set configuration of the CPU and other system devices. You can select any of the items in the left frame of the screen to go to the sub menus:

- ▶ ACPI Settings
- ▶ Trusted Computing
- ▶ CPU Configuration
- ▶ SATA Configuration
- ▶ PCH-FW Configuration
- ▶ USB Configuration
- ▶ H/W Monitor
- ▶ W83627DHG Super IO Configuration

For items marked with "▶", please press <Enter> for more options.

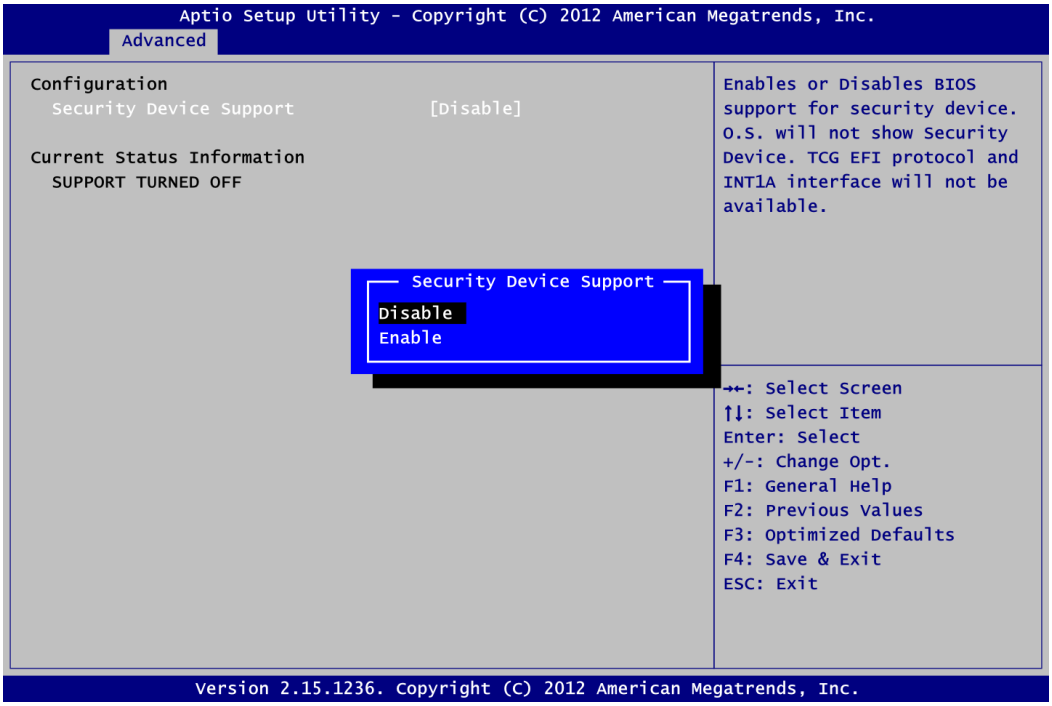


- **ACPI Settings**
  You can use this screen to select options for the ACPI configuration, and change the value of the selected option. A description of the selected item appears on the right side of the screen.



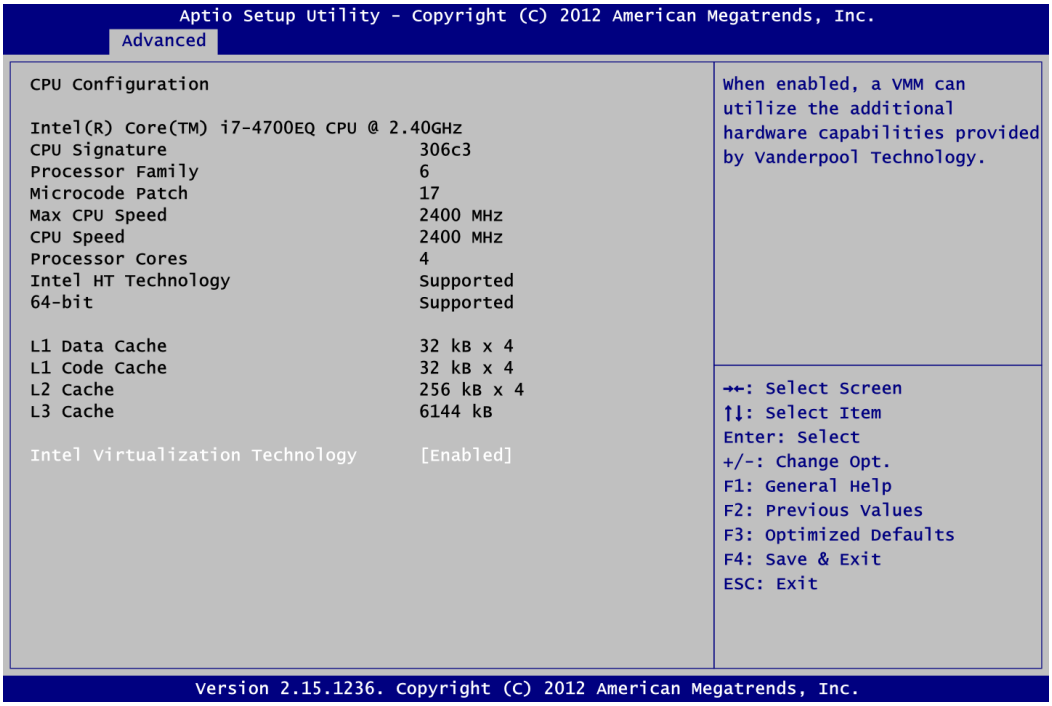
ACPI Sleep State
Select the ACPI (Advanced Configuration and Power Interface) sleep state. Configuration options are Suspend Disabled, S1 only (CPU Stop Clock) and S3 only (Suspend to RAM). The S3 only (Suspend to RAM) option selects ACPI sleep state the system will enter when suspend button is pressed.

- **Trusted Computing**
  You can use this screen for TPM (Trusted Platform Module) configuration. It also shows current TPM status information.



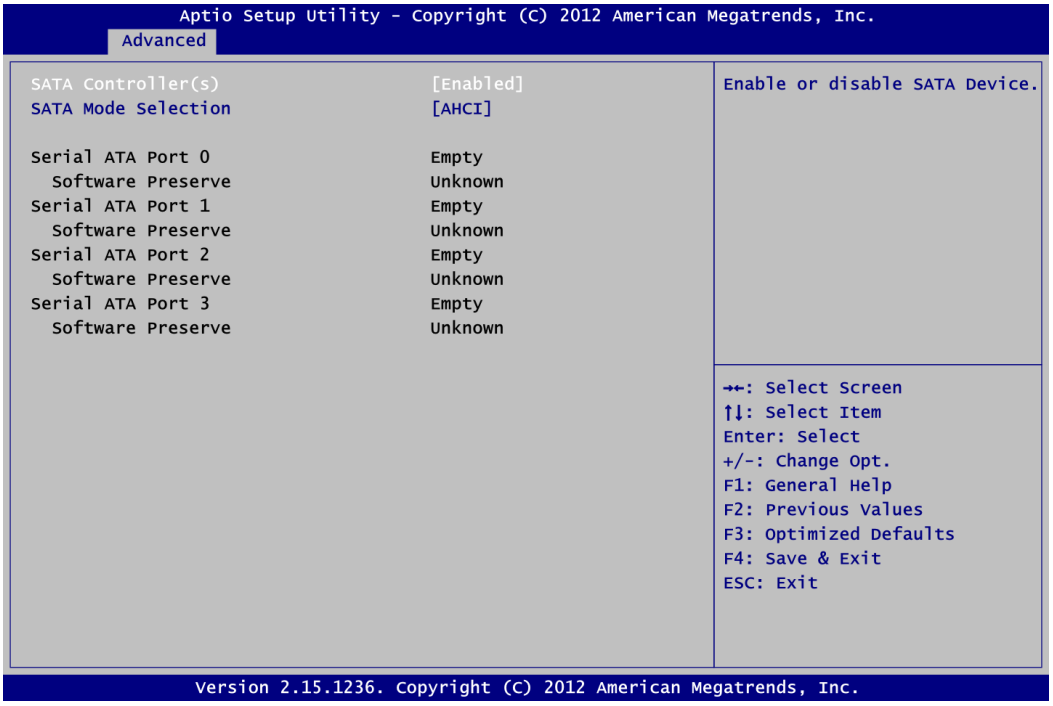
Security Device Support
Enable or disable BIOS support for security device. Operating system will not show security device.

- **CPU Configuration**
  This screen shows the CPU Configuration, and you can change the value of the selected option.



Intel Virtualization Technology
Enable or disable Intel Virtualization Technology. When enabled, a VMM can utilize the additional hardware capabilities. It allows a platform to run multiple operating systems and applications independently, hence enabling a computer system to work as several virtual systems.

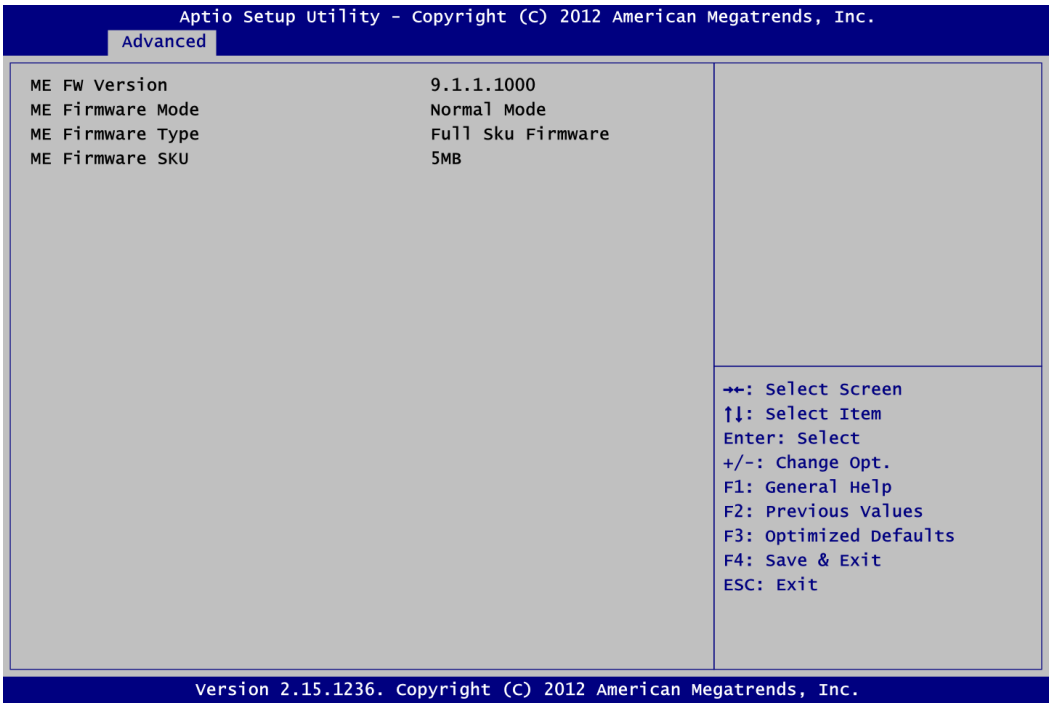
- **SATA Configuration**
  In the SATA Configuration menu, you can see the currently installed hardware in the SATA ports. During system boot up, the BIOS automatically detects the presence of SATA devices.



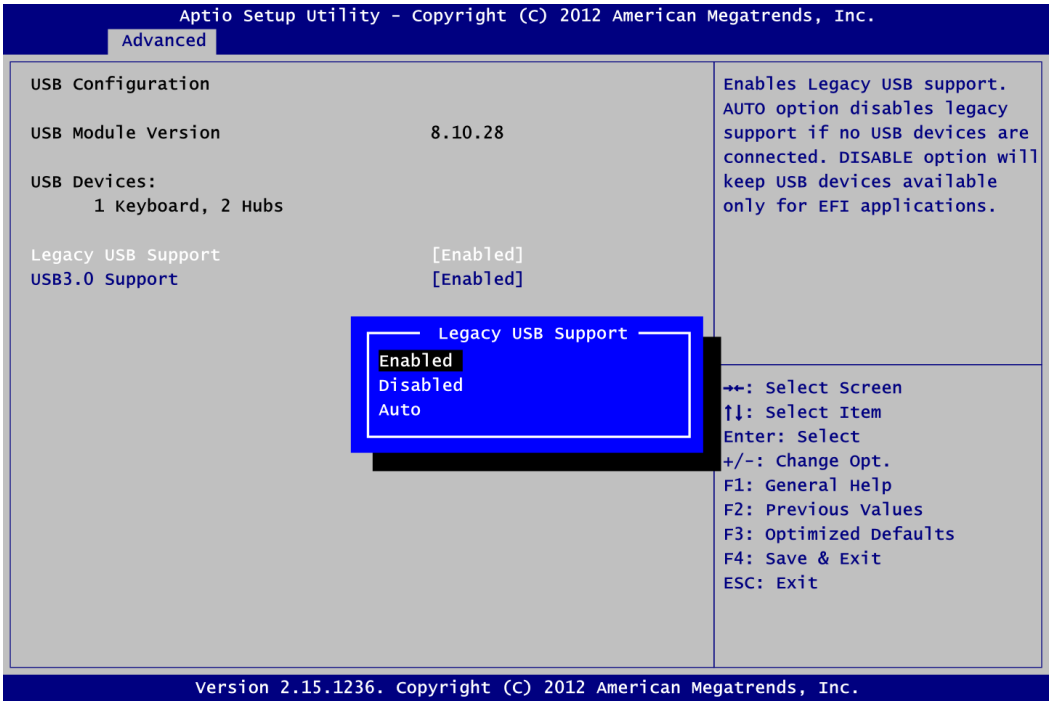
SATA Controller(s)
Enable or disable SATA device.

SATA Mode Selection
Determine how SATA controller(s) operate. Operation modes are IDE Mode, AHCI Mode and RAID Mode.

- **PCH-FW Configuration**
  This screen displays ME Firmware information.



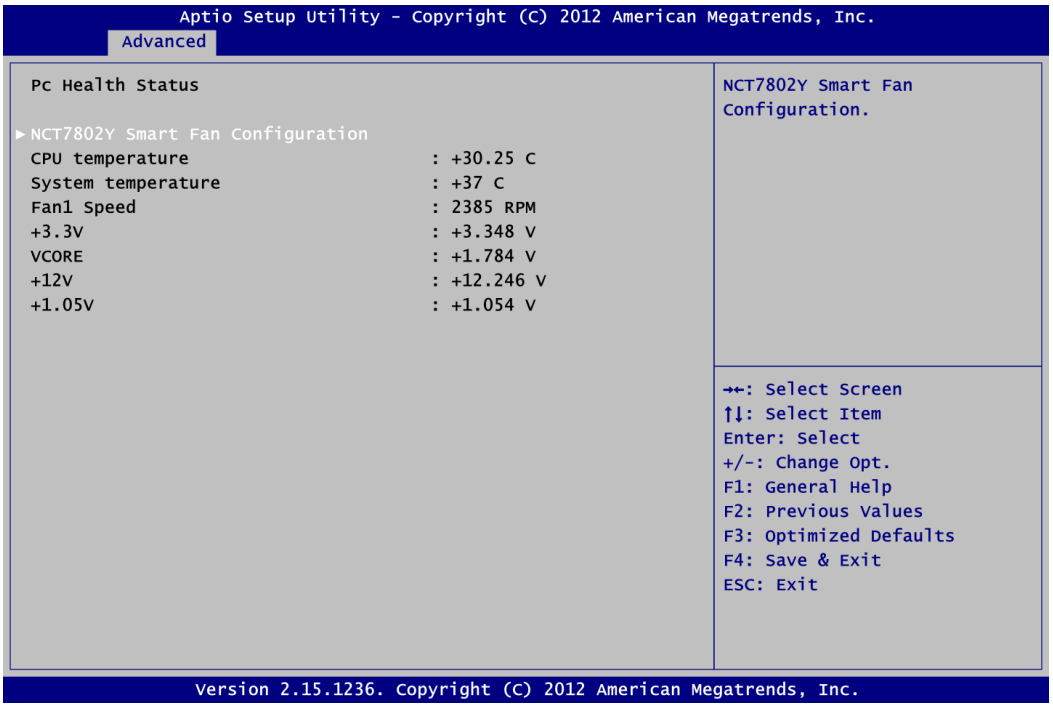
- **USB Configuration**
  You can use this screen to select options for the USB Configuration, and change the value of the selected option. A description of the selected item appears on the right side of the screen.



Legacy USB Support
Select legacy support for USB devices. The default setting is Enabled. Auto option disables legacy support if no USB devices are connected. Disable option will keep USB devices available only for EFI applications.

USB 3.0 Support
Use this item to enable or disable support for USB 3.0.

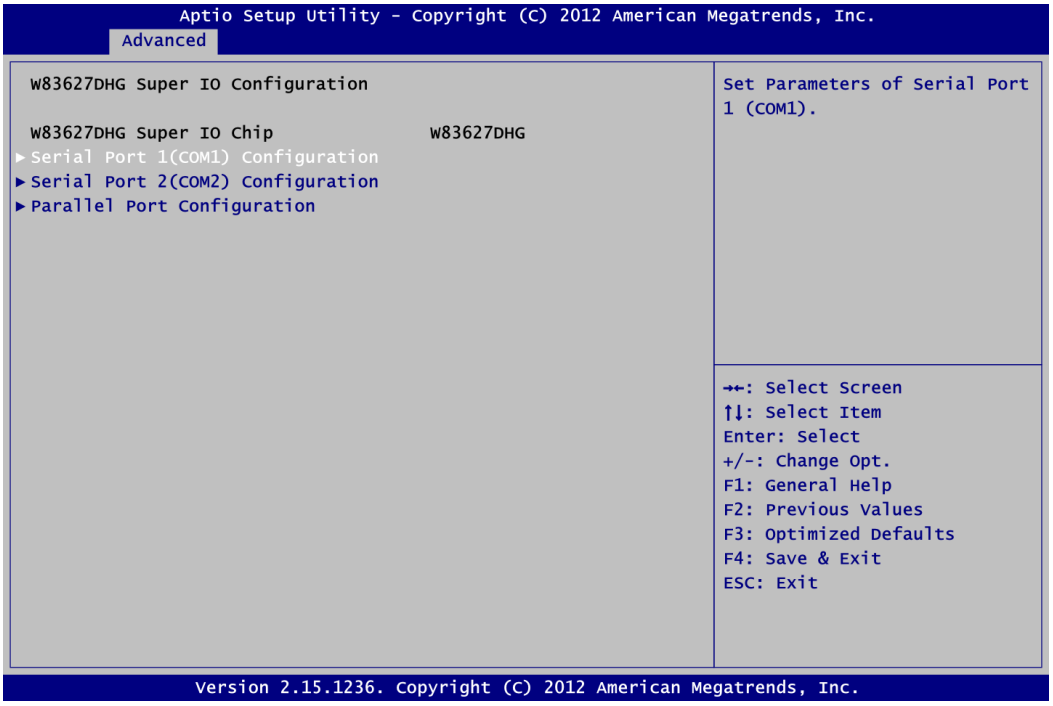
- **H/W Monitor**
  This screen is for Smart Fan configuration and hardware health status monitoring.



This screen displays the temperature of system and CPU, cooling fan speed in RPM and system voltages (+3.3V, VCORE, +12V and +1.05V).

NCT7802Y Smart Fan Configuration
This option is for Smart Fan mode configuration.

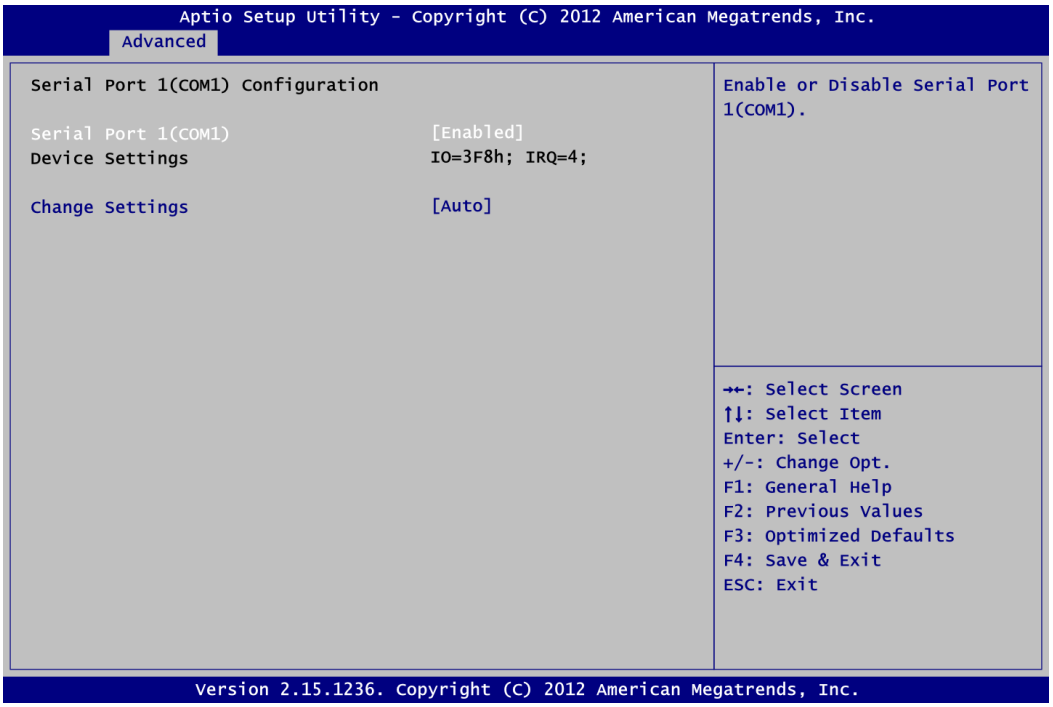
- **W83627DHG Super IO Configuration**
  You can use this screen to select options for the Super IO Configuration, and change the value of the selected option. A description of the selected item appears on the right side of the screen. For items marked with "►", please press <Enter> for more options.



Serial Port 1(COM1)~2(COM2) Configuration
Set parameters of serial port 1 (COM1) ~ 2 (COM2).

Parallel Port Configuration
Set parameters of parallel port.

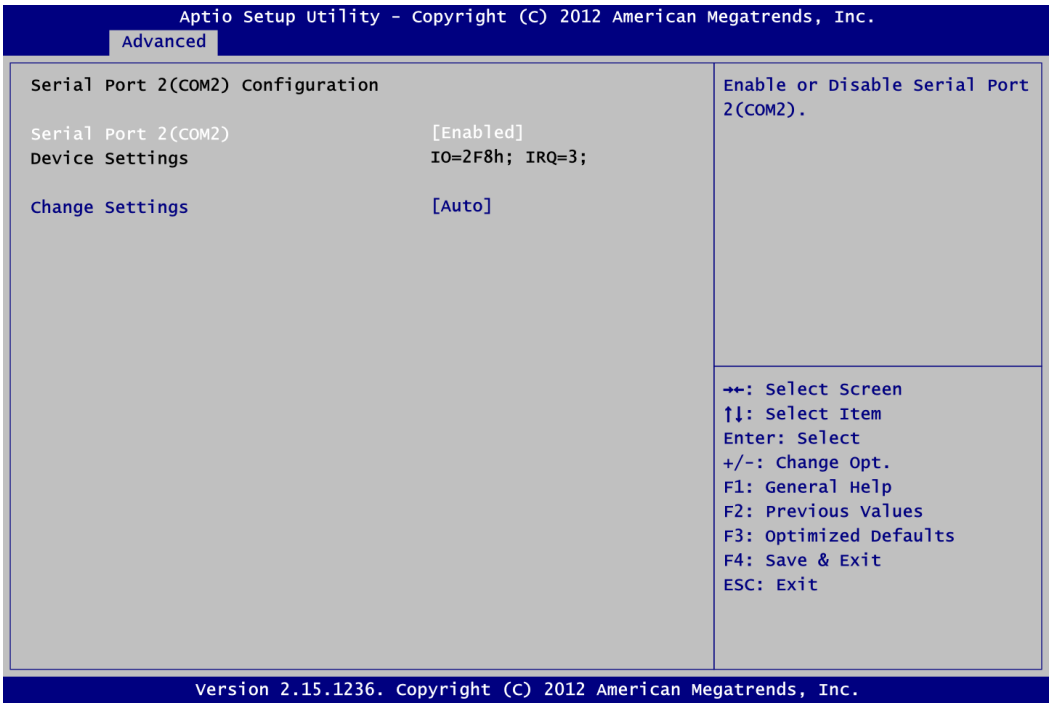
- **Serial Port 1(COM1) Configuration**



Serial Port 1(COM1)
Enable or disable serial port 1 (COM1). The optimal setting for base I/O address is 3F8h and for interrupt request line is IRQ4.

Change Settings
Change the serial port settings.

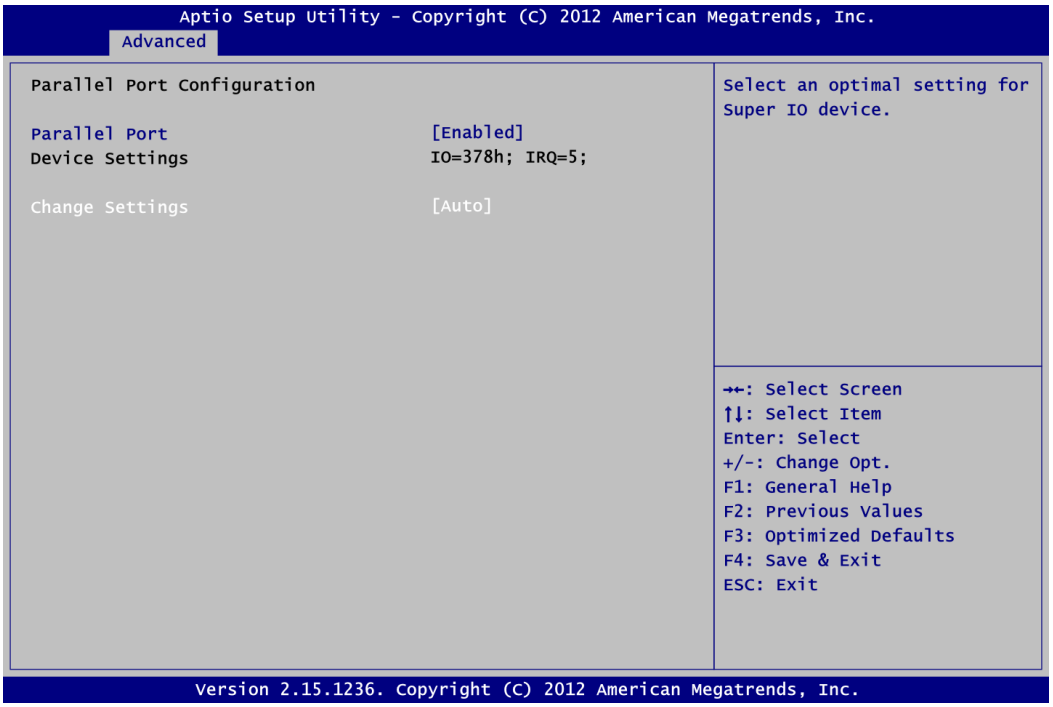
- **Serial Port 2(COM2) Configuration**



Serial Port 2(COM2)
Enable or disable serial port 2 (COM2). The optimal setting for base I/O address is 2F8h and for interrupt request address is IRQ3.

Change Settings
Change the serial port settings.

- **Parallel Port Configuration**



Parallel Port
Enable or disable parallel port (LPT). The optimal setting for base I/O address is 378h and for interrupt request line is IRQ5.

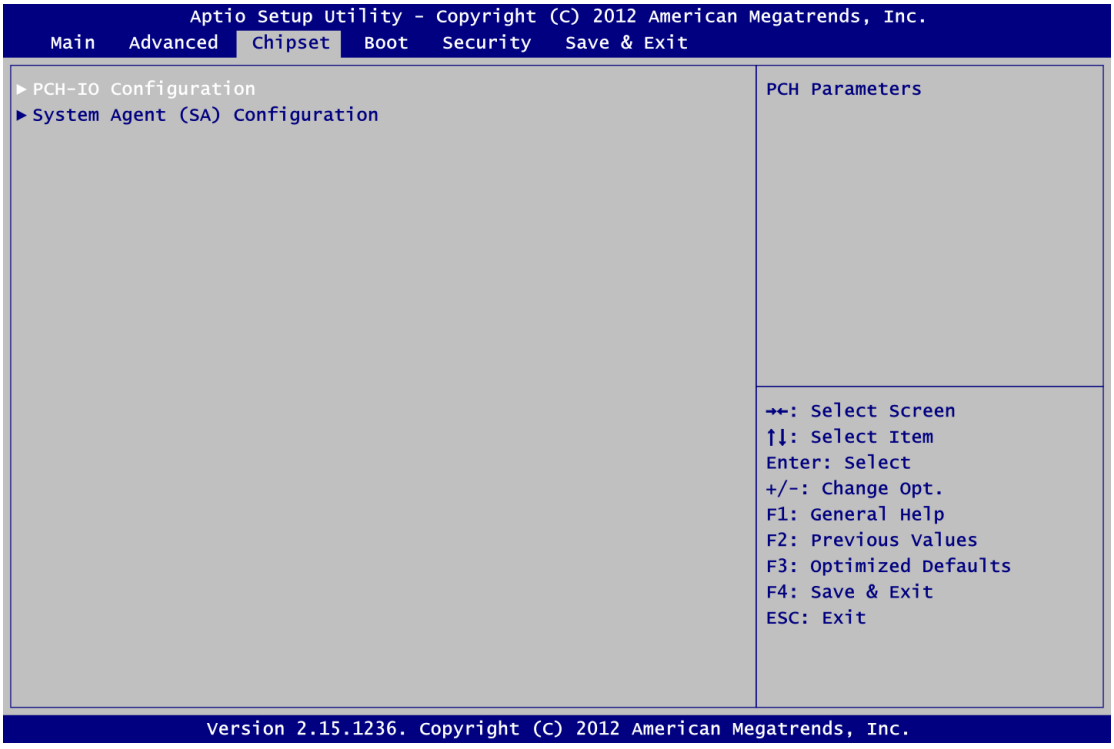
Change Settings
Change the parallel port settings.

4.5 Chipset Menu

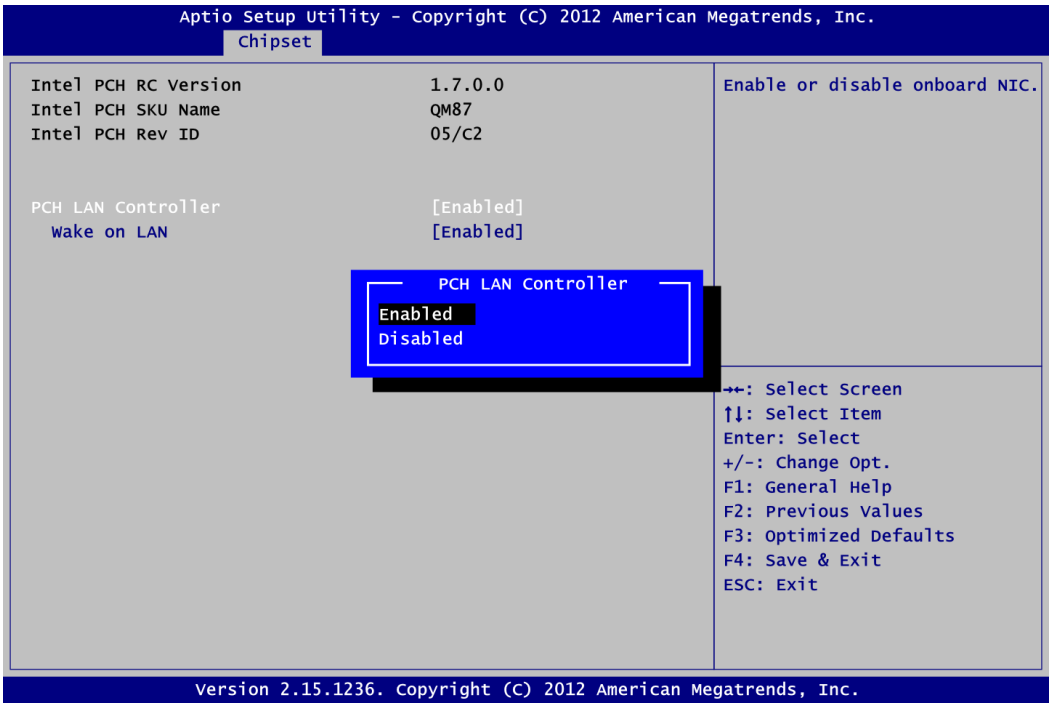
The Chipset menu allows users to change the advanced chipset settings. You can select any of the items in the left frame of the screen to go to the sub menus:

- ► PCH-IO Configuration
- ► System Agent (SA) Configuration

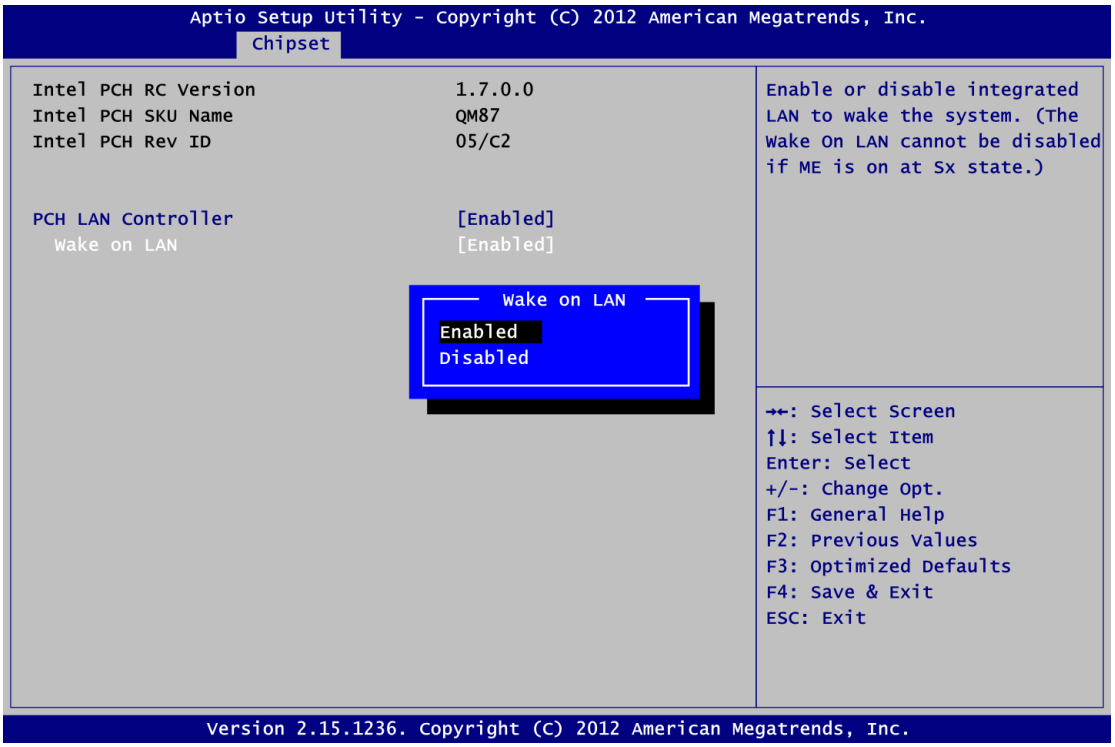
For items marked with "►", please press <Enter> for more options.



- **PCH-IO Configuration**

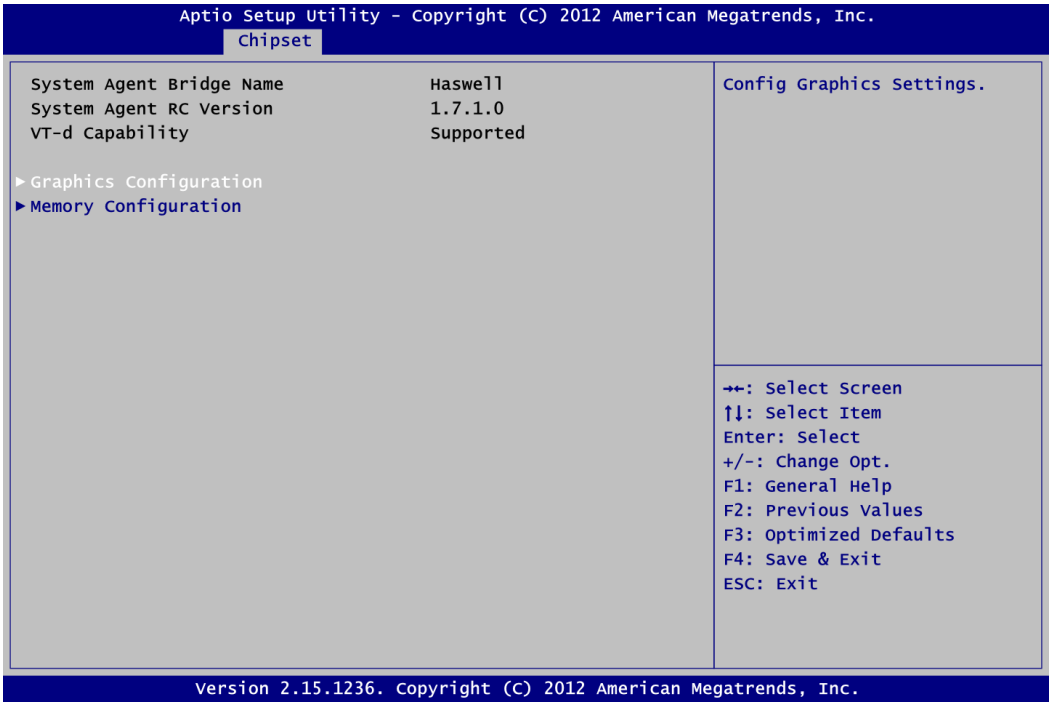


PCH LAN Controller
Enable or disable onboard PCH LAN controller.



Wake on LAN
Enable or disable integrated LAN to wake the system.

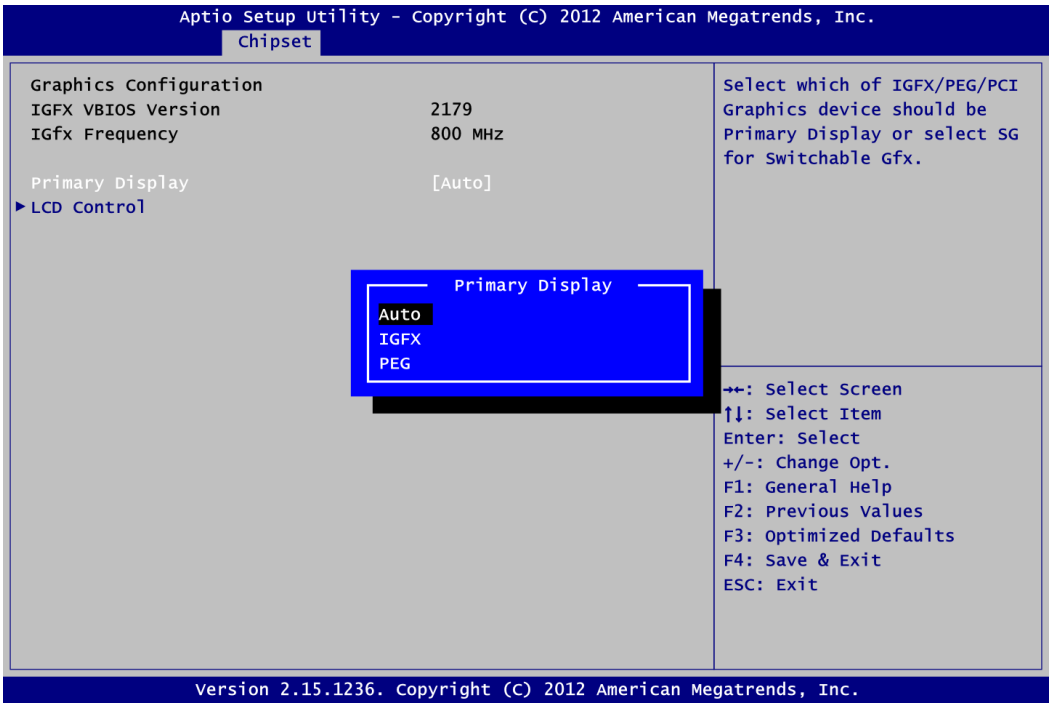
- **System Agent (SA) Configuration**
  This screen allows users to configure System Agent (SA) parameters. For items marked with "►", please press <Enter> for more options.



Graphics Configuration
Open sub menu for parameters related to graphics configuration.

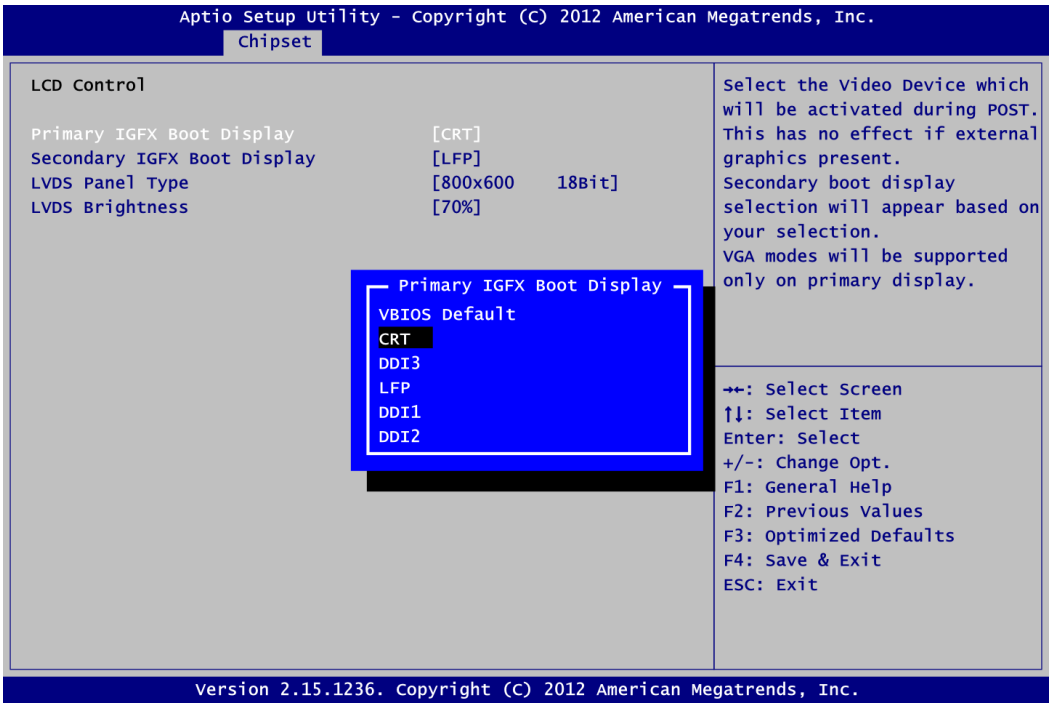
Memory Configuration
Open sub menu for information related to system memory.

- **Graphics Configuration**

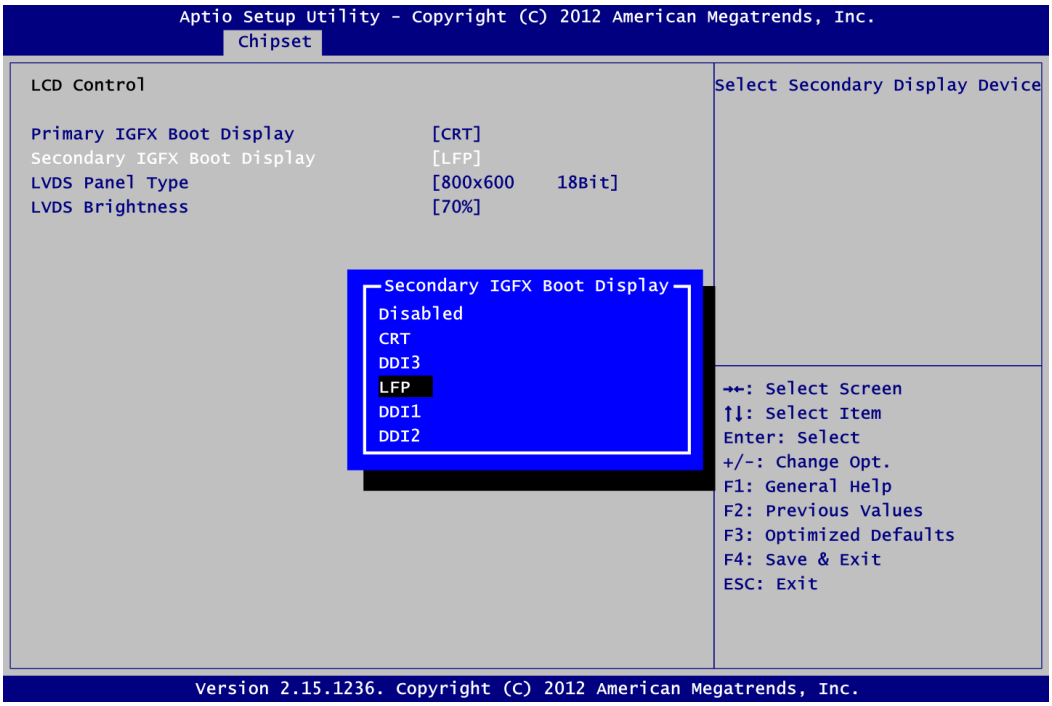


Primary Display
Select which graphics device should be primary display. Configuration options are Auto, IGFX and PEG.

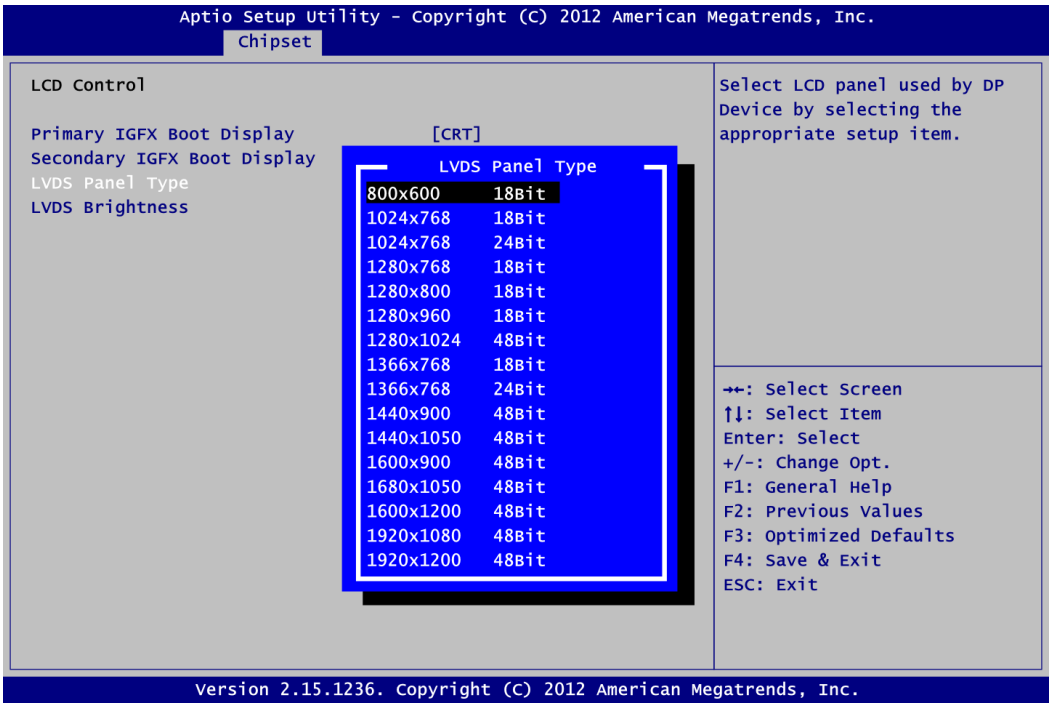
LCD Control
This item allows you to select LCD panel control options. Please press <Enter> to go to the sub menus.



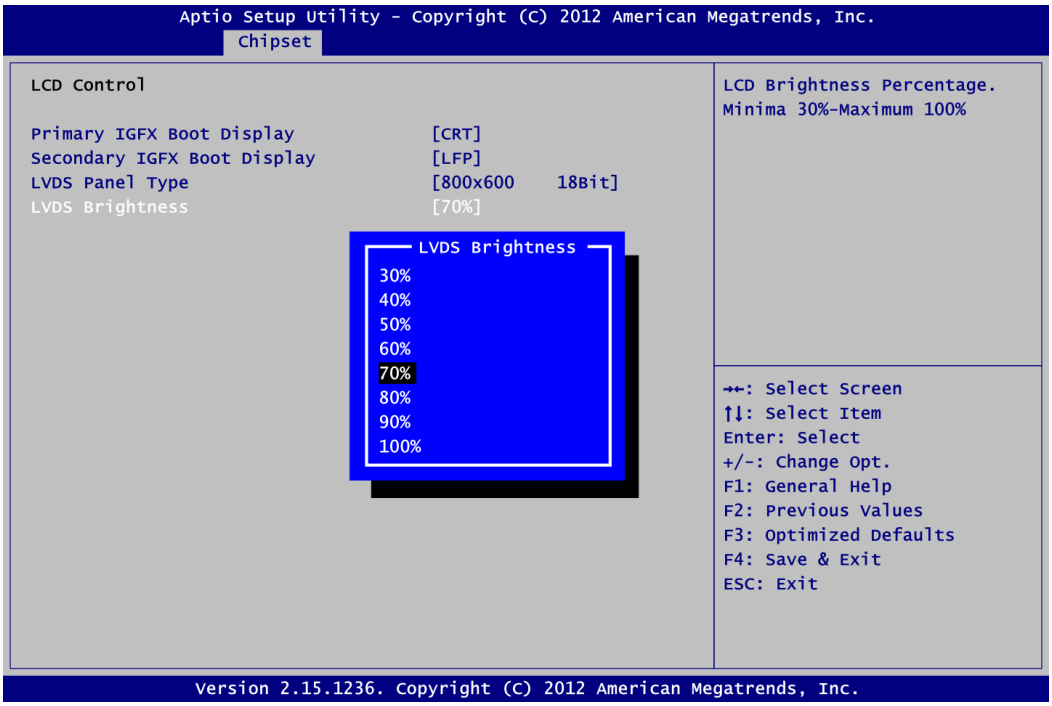
Primary IGFX Boot Display
Select the video device which will be activated during POST (Power-On Self Test).



Secondary IGFX Boot Display
Select the secondary IGFX boot display.

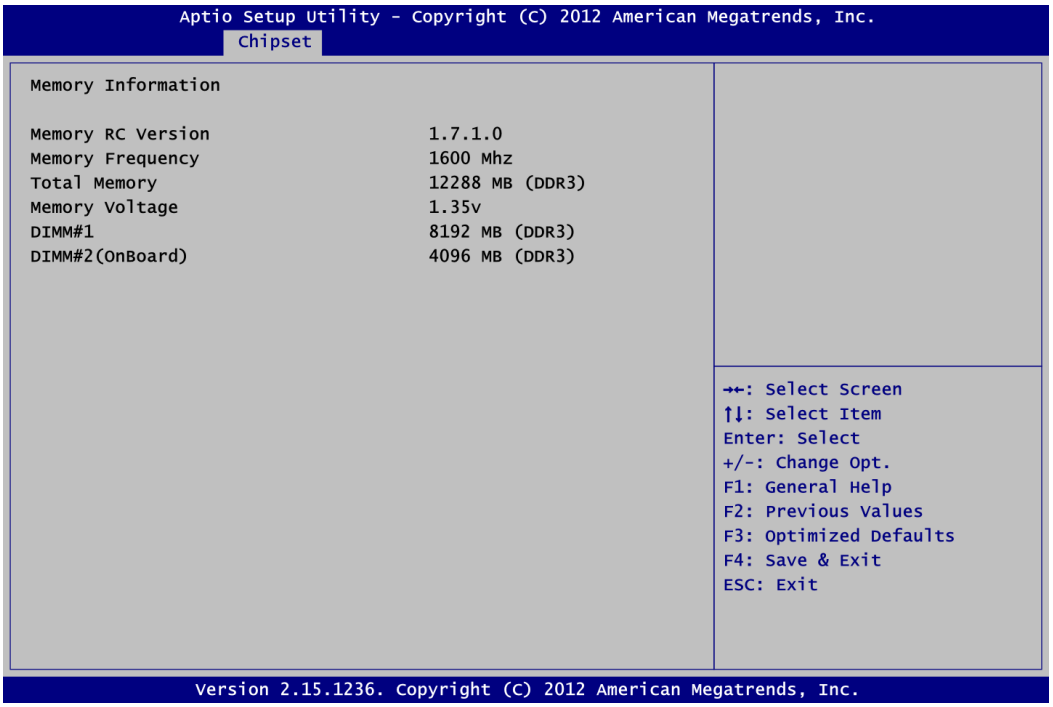


LVDS Panel Type
Select LVDS panel resolution.



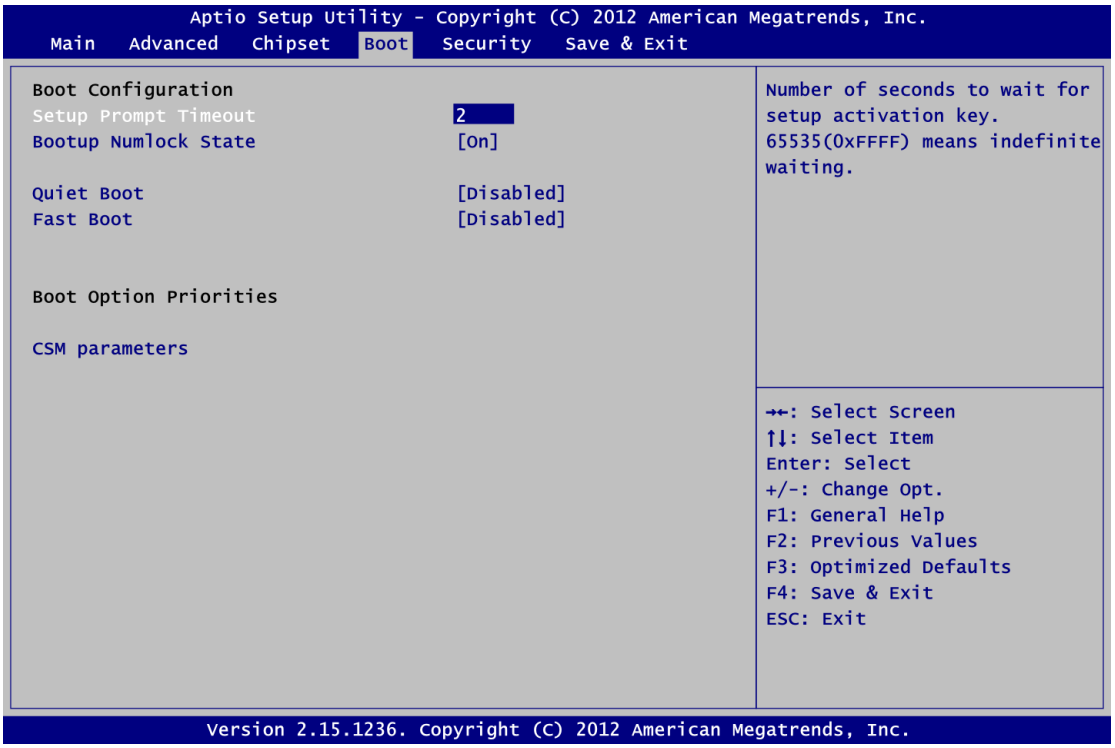
LVDS Brightness
Select the brightness of LVDS panel ranging from 30% to 100%. The default setting is 70%.

- **Memory Configuration**
  This screen shows the system memory information.



4.6 Boot Menu

The Boot menu allows users to change boot options of the system.



- **Setup Prompt Timeout**
  Number of seconds to wait for setup activation key. 65535(0xFFFF) means indefinite waiting.

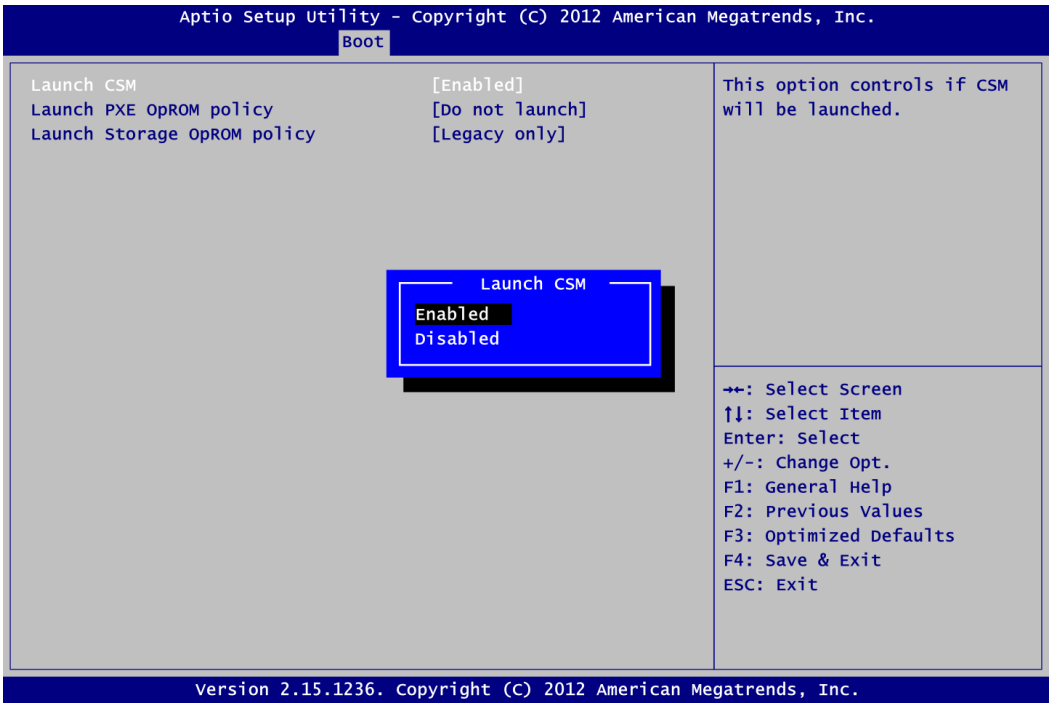
- **Bootup NumLock State**
  Use this item to select the power-on state for the keyboard NumLock.

- **Quiet Boot**
  Select to display either POST output messages or a splash screen during boot-up.

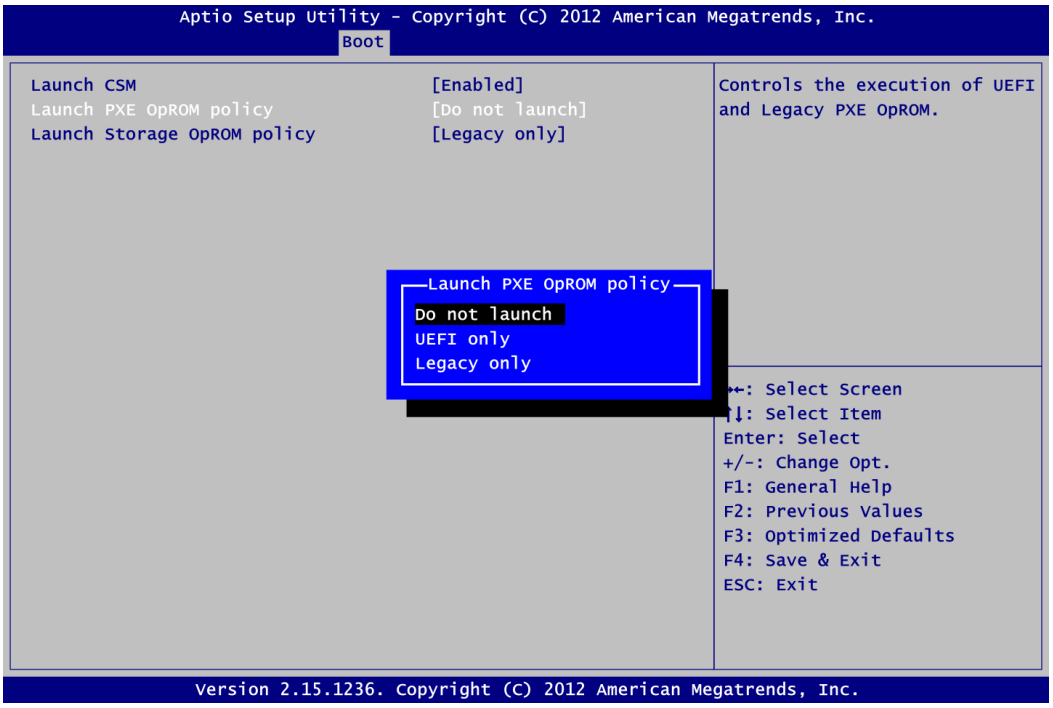
- **Fast Boot**
  Enable or disable fast boot feature. BIOS skips some certain steps to decrease time needed for booting up.

- **Boot Option Priorities**
  These are settings for boot priority. Specify the boot device priority sequence from the available devices.

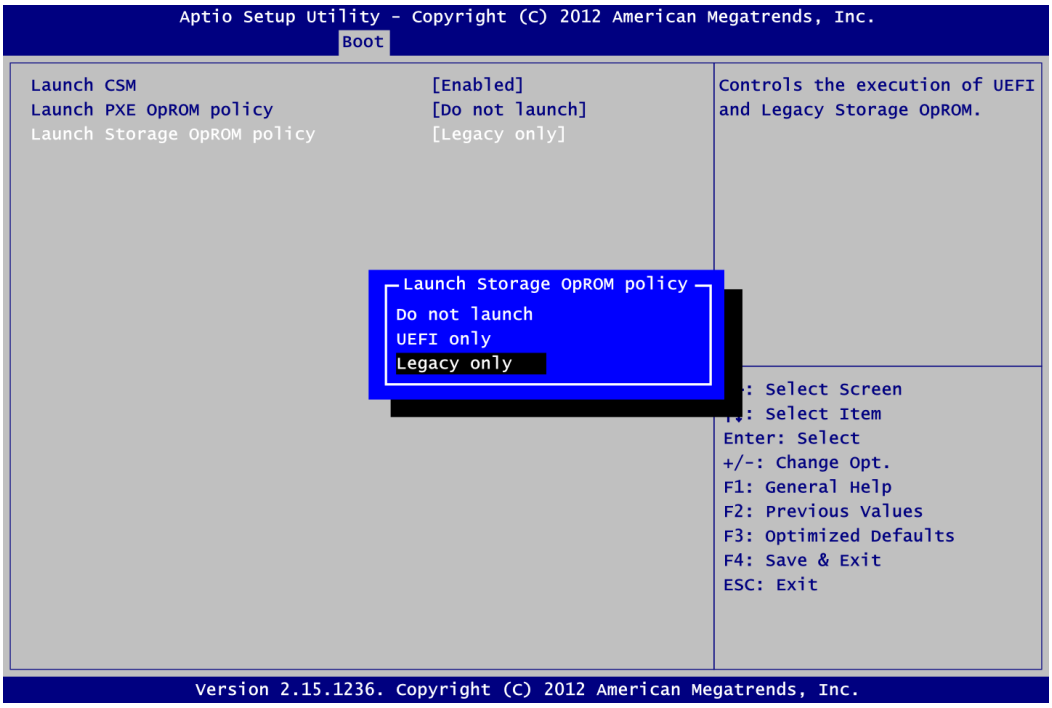
- **CSM Parameters**



Launch CSM
Enable or disable launch CSM (Compatibility Support Module).



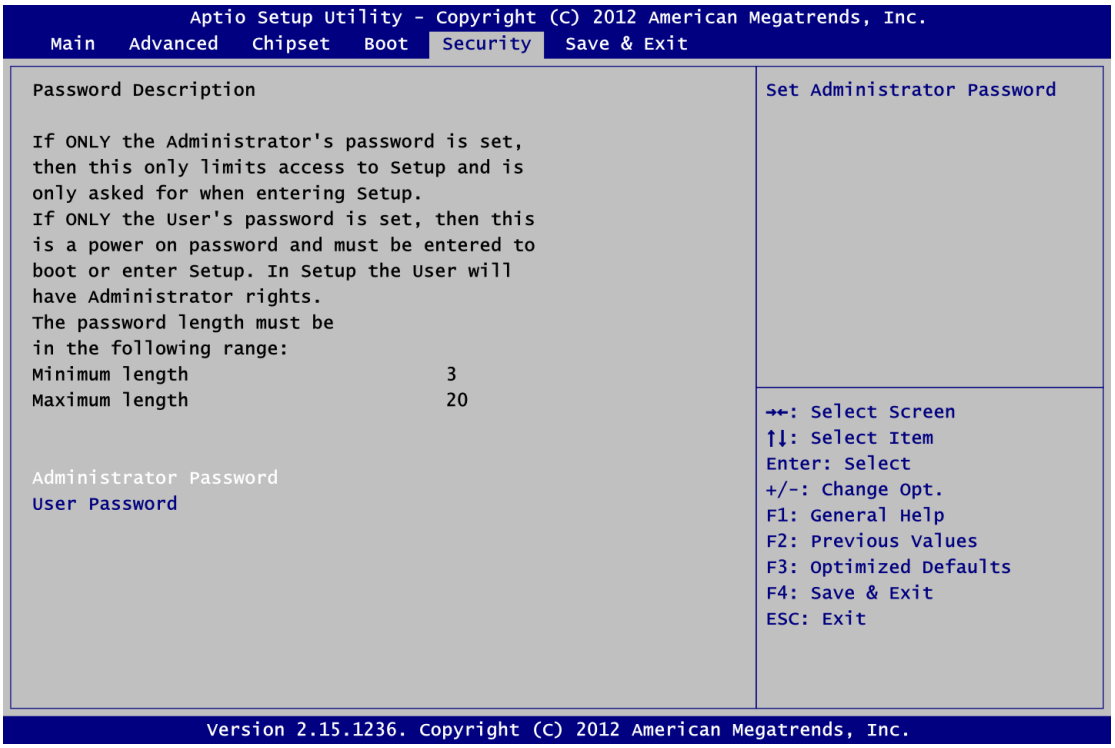
Launch PXE OpROM policy
Control the execution of UEFI and Legacy PXE OpROM.



Launch Storage OpROM policy
Control the execution of UEFI and Legacy Storage OpROM.

4.7 Security Menu

The Security menu allows users to change the security settings for the system.

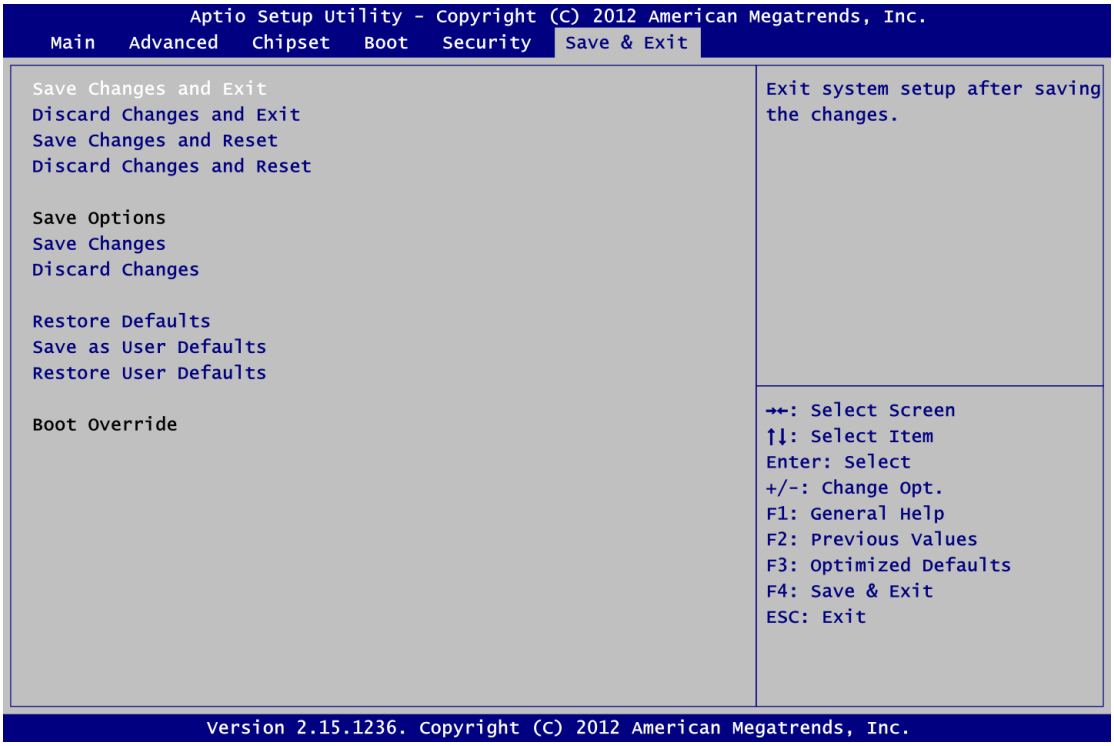


- **Administrator Password**
  Set administrator password.

- **User Password**
  Set user password.

4.8 Save & Exit Menu

The Save & Exit menu allows users to load your system configuration with optimal or fail-safe default values.



- **Save Changes and Exit**
  When you have completed the system configuration changes, select this option to leave Setup and return to Main Menu. Select Save Changes and Exit from the Save & Exit menu and press <Enter>. Select Yes to save changes and exit.

- **Discard Changes and Exit**
  Select this option to quit Setup without making any permanent changes to the system configuration and return to Main Menu. Select Discard Changes and Exit from the Save & Exit menu and press <Enter>. Select Yes to discard changes and exit.

- **Save Changes and Reset**
  When you have completed the system configuration changes, select this option to leave Setup and reboot the computer so the new system configuration parameters can take effect. Select Save Changes and Reset from the Save & Exit menu and press <Enter>. Select Yes to save changes and reset.

- **Discard Changes and Reset**
  Select this option to quit Setup without making any permanent changes to the system configuration and reboot the computer. Select Discard Changes and Reset from the Save & Exit menu and press <Enter>. Select Yes to discard changes and reset.

- **Save Changes**
  When you have completed the system configuration changes, select this option to save changes. Select Save Changes from the Save & Exit menu and press <Enter>. Select Yes to save changes.

- **Discard Changes**
  Select this option to quit Setup without making any permanent changes to the system configuration. Select Discard Changes from the Save & Exit menu and press <Enter>. Select Yes to discard changes.

- **Restore Defaults**
  It automatically sets all Setup options to a complete set of default settings when you select this option. Select Restore Defaults from the Save & Exit menu and press <Enter>.

- **Save as User Defaults**
  Select this option to save system configuration changes done so far as User Defaults. Select Save as User Defaults from the Save & Exit menu and press <Enter>.

- **Restore User Defaults**
  It automatically sets all Setup options to a complete set of User Defaults when you select this option. Select Restore User Defaults from the Save & Exit menu and press <Enter>.

- **Boot Override**
  Select a drive to immediately boot that device regardless of the current boot order.

Appendix A
Watchdog Timer and GPIO

About Watchdog Timer

Software stability is major issue in most application. Some embedded systems are not watched by human for 24 hours. It is usually too slow to wait for someone to reboot when computer hangs. The systems need to be able to reset automatically when things go wrong. The watchdog timer gives us solution.

The watchdog timer is a counter that triggers a system reset when it counts down to zero from a preset value. The software starts counter with an initial value and must reset it periodically. If the counter ever reaches zero which means the software has crashed, the system will reboot.

About GPIO

The onboard GPIO (general input and output) has 8 bits (GPI0~3 and GPO0~3). In default, all pins are pulled high with +3.3V level (according to main power). The BIOS default settings are 4 inputs and 4 outputs where all of these pins are set to 1. Use these GPIO signals to control cash drawers and sense warning signals from an Uninterrupted Power System (UPS), or perform store security control.

Sample Program

Programming sample code (from CEM880 FINTEK F75111R):

```
/*---------------- INCLUDE FILE ----------------------*/
#include <stdio.h>
#include <conio.h>
#include <bios.h>

#define UCHAR   unsigned char
#define UINT    unsigned int
#define SMIOBASE      0xF040
/******************
       SMIOBASE can get from PCI device Bus-0,Device-31, Function-3
       Register 20h~23h, the value is the IO base address.
*******************/
#define SM_REG         (SMIOBASE+3)
#define SM_ADDR        (SMIOBASE+4)
#define SM_DATA        (SMIOBASE+5)
#define SM_CMD         (SMIOBASE+2)
#define SM_STATUS      (SMIOBASE+0)
#define SM_byteAccess 0x48
#define Device_Addr    0x6E  // depend on hardware designed Low:0x9C, High:0x6E
UCHAR _read_smbus(UCHAR ,UCHAR);
void _write_smbus(UCHAR,UCHAR, UCHAR);
#define F75111_CHIPID           0x0003
#define F75111_VENDORID         0x3419
#define WDTOUT10_CntlReg1       0x34
#define WDTOUT10_2S_bit         0x04    //bit 2
#define WDTOUT10_OINV_bit       0x02    //bit 1
#define WDTOUT10_Status_bit     0x01    //bit 0 , write 1 for clear status
#define WDTOUT10_CntlReg2       0x35
#define WDTOUT10_Enable_bit     0x80    //bit 7
#define WDTOUT10_PTIME          0x7f    //bit 0~6
```

```
/* ------------------ MAIN PROGRAM -------------------*/
main()
{
UCHAR   xch,xch2;
UINT    chipid=0, vendorid=0;
       //Check the Chip ID information
       xch=_read_smbus(Device_Addr,0x5a);     //Chip ID 1
       xch2=_read_smbus(Device_Addr,0x5b);    //Chip ID 2
       chipid=((UINT)xch2 << 8) + (UINT)xch;
       xch=_read_smbus(Device_Addr,0x5d);     //Vendor ID 1
       xch2=_read_smbus(Device_Addr,0x5E);    //Vendor ID 2
       vendorid=((UINT)xch2 << 8) + (UINT)xch;
       if (F75111_CHIPID != chipid || F75111_VENDORID != vendorid )
         { printf("!!! Not found F75111 chip !!!\n");
           exit(-1);
         }
       printf("=== Found F75111 chip ===\n");
       printf("=== GPIO Output Test ===\n");

       //set GPIO3x  direction
       printf("Set F75111 GPIO3x pin is output direction\n");
       _write_smbus(Device_Addr,0x40,0x0f);   //GPIO3x Output direction

       //set GPIO3x output level or plus
       printf("Set F75111 GPIO3x pin is output Level\n");
       _write_smbus(Device_Addr,0x43,0);      //GPIO3x Level control

       //write GPIO3x data
       printf("Write GPIO3x data is 0x0A\n");
       printf("GPO0=0, GPO1=1, GPO2=0, GPO3=1\n");
       _write_smbus(Device_Addr,0x41,0x0a);
       printf("Please check the GPO level and hit any key to continue\n");
       getch();

       //write GPIO3x data
       printf("Write GPIO3x data is 0x05\n");
       printf("GPO0=1, GPO1=0, GPO2=1, GPO3=0\n");
       _write_smbus(Device_Addr,0x41,0x05);
       printf("Please check the GPO level and hit any key to continue\n");
       getch();


       //set GPIO10,11,12 used
       printf("=== GPIO Input Test ===\n");
       printf("Set F75111 GPIO1x is used GPIO function\n");
       xch=_read_smbus(Device_Addr,0x03);
       xch &= 0xE0 ;
       _write_smbus(Device_Addr,0x03,xch);    //set Pin GPIO10/11/12 used
       _write_smbus(Device_Addr,0x04,0);      //set Pin GPIO1x used

       printf("Set F75111 GPIO10,11,12,13 is input function\n");
       _write_smbus(Device_Addr,0x10,0x00);   //set GPIO1x input direction

       printf("Set F75111 GPIO10,11,12,13 is Level mode\n");
       _write_smbus(Device_Addr,0x13,0x00);   //set GPIO1x Level Control

       xch=_read_smbus(Device_Addr,0x12);     //read GPIO1x Status
       printf("Read the GPI0,1,2,3 input data is %02X\n",xch);
       printf("Please Change the GPIx input and hit any key to continue\n");
       getch();
       xch=_read_smbus(Device_Addr,0x12);     //read GPIO1x Status
       printf("Read the GPI0,1,2,3 input data is %02X\n",xch);
       printf("Please Change the GPIx input and hit any key to continue\n");
       getch();
       xch=_read_smbus(Device_Addr,0x12);     //read GPIO1x Status
       printf("Read the GPI0,1,2,3 input data is %02X\n",xch);
       /*******************************************/
       printf("===== WatchDogTimer Test =====\n");
       printf("Set WDTOUT10 pin used\n");
       _write_smbus(Device_Addr,0x01,0x20);   //Pin1 config

       printf("Set WDTOUT10 output 100ms plus, normal level\n");
       _write_smbus(Device_Addr,WDTOUT10_CntlReg1,0);      //WDT10 control

       printf("Set WDTOUT10 Time 10 seconds and enable WDT\n");
       _write_smbus(Device_Addr,WDTOUT10_CntlReg2,0x8A);   //WDT10 control
```

```
        printf("Please hit any key in period of 10 seconds\n");
        getch();
        _write_smbus(Device_Addr,WDTOUT10_CntlReg2,0x8A);   //WDT10 control
        printf("The sytsem will reset when the 10 seconds times out\n");
}

void _write_smbus(UCHAR xAddr,UCHAR xReg, UCHAR xData)
{
        while (1)
        { if (_check_smbus_busy()==0) break;
        }
        outp(SM_REG, xReg);
        xdelay();             //because the CPU too fast,delay for IO
        outp(SM_ADDR, xAddr);
        xdelay();             //because the CPU too fast,delay for IO
        outp(SM_DATA, xData);
        xdelay();             //because the CPU too fast,delay for IO
        outp(SM_CMD, SM_byteAccess);
        xdelay();             //because the CPU too fast,delay for IO
        outp(SM_STATUS, 02); //clear interrupt status
}

UCHAR _read_smbus(UCHAR xAddr,UCHAR xReg)
{
UCHAR   xch,xch2;
        while (1)
        { if (_check_smbus_busy()==0) break;
        }
        outp(SM_REG, xReg);
        xdelay();             //because the CPU too fast,delay for IO
        outp(SM_ADDR, xAddr+1);
        xdelay();             //because the CPU too fast,delay for IO
        outp(SM_CMD, SM_byteAccess);
        xdelay();             //because the CPU too fast,delay for IO
        while (1)
        { if (_check_smbus_busy()==0) break;
        }
        xch=inp(SM_DATA);
        xdelay();             //because the CPU too fast,delay for IO
        outp(SM_STATUS, 2);  //clear interrupt status
        return xch;
}

_check_smbus_busy(void)
{
UCHAR   xch;
        xch=inp(SM_STATUS);
        if (xch & 0x02 ) outp(SM_STATUS, 2);   //clear interrupt status
        if ( xch & 0x02 ) return 1;
        if ( xch & 0x01 ) return 1;
        return 0;
}

xdelay()
{
int     xxi,xxj,xxk=0;
        for (xxi=0 ; xxi< 0x1000 ; xxi++) {
                for (xxj=0 ; xxj < 0x100 ; xxj++) {
                        xxk++;
                }
        }
}
```