MB839-1C
MB839-4C

USER'S MANUAL

Version A1

Acknowledgments

Award is a registered trademark of Award Software International, Inc.

PS/2 is a trademark of International Business Machines Corporation.

Intel® and **Atom**™ and Celeron® are registered trademark of Award Software International, Inc. registered trademark and trademark of Intel Corporation.

Microsoft Windows is a registered trademark of Microsoft Corporation.

Winbond is a registered trademark of Winbond Electronics Corporation.

All other product names or trademarks are properties of their respective owners.

Table of Contents

This page is intentionally left blank.

Introduction

Product Description

The MB839 is a small footprint single board computer that is configured with the Intel Celeron® J1900 or Atom^{TM} E3815 SoC.

This 5.25-inch SBC provides greater flexibility for developers of embedded computing solutions. It is ideally suited for rugged and compact designs as in Internet devices and applications in automation, industrial control, data acquisition, thin client and other embedded PC applications.

One DDR3 SO-DIMM on board implements up to 8GB of system memory. Four Gigabit LAN ports are also available.

Advanced connectivity and expansion interfaces are provided by one CF socket, one SATA-III, one Mini PCI-e and two USB 2.0 ports.

MB839-4C Features

- Intel Celeron® J1900 at 2.0GHz processor
- Four Intel® Gigabit LAN ports
- DDR3L SO-DIMM x1, up to 8GB
- Mini PCI-E half size slot x1
- Compact Flash socket
- LAN Bypass Enable/Disable pre-setting by BIOS on Eth 3 & 4

MB839-1C Features

- Intel Atom^{TM} E3815 at 1.46GHz processor
- Four Intel® Gigabit LAN ports
- DDR3L SO-DIMM x1, up to 8GB
- Mini PCI-E half size slot x1
- Compact Flash socket
- LAN Bypass Enable/Disable pre-setting by BIOS on Eth 3 & 4

Checklist

Your MB839 package should include the items listed below.

- The MB839-1C or MB839-4C embedded board
- Drivers DVD containing chipset drivers and flash memory utility
- Cables are optional.

MB839 Specifications

Product Name	MB839-1C or MB839-4C
Form Factor	5.25" Disk Size SBC
CPU Type Operating Frequency	Intel "Bay Trail" Processor, 22nm Bulk MB839-4C: Celeron J1900 = 2.0 GHz [TDP= 10W], Quad Core MB839-1C: Atom E3815 = 1.46 GHz [TDP= 5W], Single Core
BIOS	AMI BIOS w/ACPI
Ethernet controller	Intel I211-AT PCI Express Gigabit ethernet controller x4
Memory	CPU on-die memory controller supporting up to 8GB One DDR3L SO-DIMM socket, Non-ECC, unbuffered
LAN	• Console: RS-232 @ RJ45 • Eth1, 2, 3 & 4: Intel I211-AT @ RJ45 with LED
Network Bypass	One segment hardware Bypass (Eth3 & 4) Control by GPIO / Watchdog
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
Storage	• Onboard CF Socket x1 • 7-pin SATA connector for 2.5" HDD x1 (optional HDD kit)
Rear Panel	• Cylindrical (Tip) Connector DC +12V inlet with Screw Lock • Power On / Off Switch • Optional opening for Wireless LAN antenna • RJ45 x1 for Console • RJ45 with LED x4 for GLAN • USB 2.0 x2
Front Panel	• LED: Power (Green) / Alarm (Red) / Status (Yellow) • LAN Speed LED (Yellow / Green) x4 • LAN Link / Act LED (Green) x4 • Factory Mode Restore Reset Switch (GPIO control)
USB 2.0	USB 2.0 x4 • External x2 • [2x4] Pin header Onboard x1 • Mini PCI-e Socket x1
Video	VGA pin header onboard x1
LPC I/O	Nuvoton NCT5523D: • RS-232 [2x5] Pin Header Onboard x1 • RJ45 Console x1 • Hardware monitors • Fan Connector x1
Internal I/O Headers	• 4-pin Smart Fan Connector x1 • 2-pin header for DC-in (12V) x1 • Keyboard + Mouse ([1x6] Pin Header) x1
Expansion Interface	• Mini PCI Socket x1 • Mini PCI-e Socket x1 (USB Signal Only)
Power Supply	Full range 40W Adapter / 12V (Optional)
Dimensions	203(W) x 149.5(D) mm
Operation Temperature	0 ~ 60 ˚C (32 ~ 140 ˚F)
Storage Temperature	-20 ~ 80 ˚C (-4 ~ 176 ˚F)

Board Dimensions
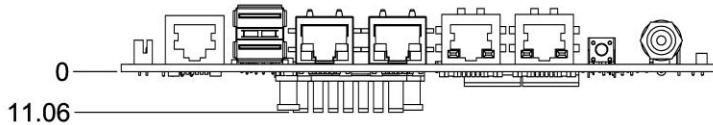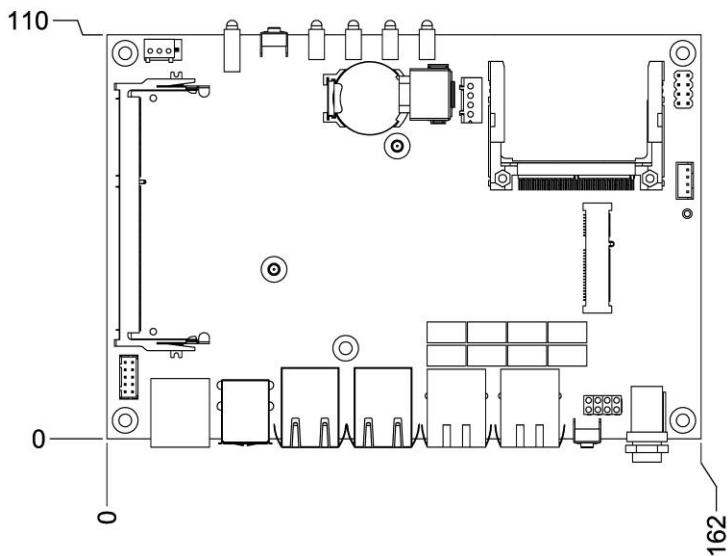
unit: mm

Installations

This section provides information on how to use the jumpers and connectors on theMB839 in order to set up a workable system. The topics covered are:

Installing the Memory

The MB839 board supports one DDR3 memory socket that can support up to 8GB memory, DDR3L (w/o ECC function).

Installing and Removing Memory Modules

To install the DDR3L module, locate the memory slot on the board and perform the following steps:

1. Hold the DDR3L module so that the key of the DDR3L module aligns with that on the memory slot. Insert the module into the socket at a slight angle (approximately 30 degrees). Note that the socket and module are both keyed, which means that the module can be installed only in one direction.

2. To seat the memory module into the socket, apply firm and even pressure to each end of the module until you feel it slip down into the socket.

3. With the module properly seated in the socket, rotate the module downward. Continue pressing downward until the clips at each end lock into position.

4. To remove the DDR3L module, press the clips with both hands.
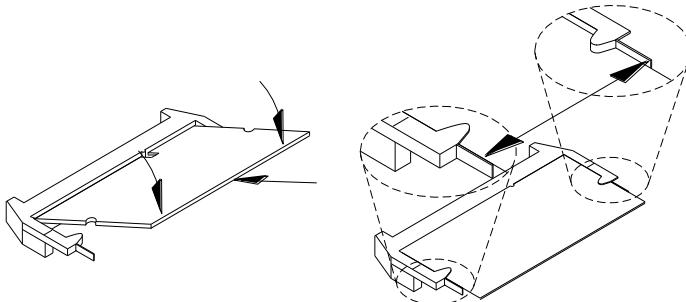
Setting the Jumpers and Connectors

Jumpers are used on MB839 to select various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your needs. The following lists the connectors on MB839 and their respective functions.

Jumper & Connector Location on MB839-1C / MB839-4C

JP1: MicroSD Power select
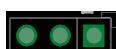
Pin #	Signal Name
1	+V1.8V
2	+VSDIO_IN
3	+V3.3V

JP2: BIOS Flash Security Setting

Pin #	Signal Name
1	GND
2	I2S_2_TXD

JP3: Clear CMOS Setting

JP2	Setting
	Normal
	Clear CMOS

JP4: Clear ME Setting

JP3	Setting
□ □ □ 1 2 3	Normal
□ □ □ 1 2 3	Clear ME

JP5: VGA Connectors

Signal Name	Pin #	Pin #	Signal Name
VGA_R	1	2	VGA_PWR
VGA_G	3	4	GND
VGA_B	5	6	NC
NC	7	8	VGADDCDATA
GND	9	10	HSYNC
GND	11	12	VSYNC
GND	13	14	VGADDCCLK
GND	15		

JP6: LPC Debug Port

Signal Name	Pin #	Pin #	Signal Name
LPC_AD0	1	2	SIO_PLTRST#
LPC_AD1	3	4	LPC_FRAME#
LPC_AD2	5	6	+3.3V
LPC_AD3	7	8	Ground
LPC_CLK	9		

JP7: SPI Debug Port

Signal Name	Pin #	Pin #	Signal Name
NC	1	2	NC
SPI_CS#0	3	4	+1.8V
SPI_SO	5	6	SPI0_HOLD#
SPI0_WP#	7	8	SPI_CLK
GND	9	10	SPI_SI

JP12: System Function Connector

JP12 provides connectors for system indicators that provide light indication of the computer activities and switches to change the computer status. JP12 is a 8-pin header that provides interfaces for the following functions

Pin 1,2: Power LED

The power LED indicates the status of the main power switch.

Pin #	Signal Name
1	+5V
2	GND

Pin 3,4: ATX Power ON Switch

Pin 3 & 4 are for "ATX Power Supply On/Off Switch" on the system that connects to the power switch on the case. When pressed, the power switch will force the system to power on. When pressed again, it will force the system to power off.

Pin #	Signal Name
3	Power_ON
4	GND

Pin 5, 6: Hard Disk Drive LED Connector

This connector connects to the hard drive activity LED on control panel. This LED will flash when the HDD is being accessed.

Pin #	Signal Name
5	+3.3V
6	-HDD_LED

Pin 7, 8: Reset Switch

The reset switch allows the user to reset the system without turning the main power switch off and then on again. Orientation is not required when making a connection to this header.

Pin #	Signal Name
7	PM_SYSRST#
8	GND

J3: MCU Update (Factory use only)

J4: Serial Port (COM2)

Signal Name	Pin #	Pin #	Signal Name
DCD#	1	6	DSR#
SIN	2	7	RTS#
SOUT	3	8	CTS#
DTR#	4	9	RI#
GND	5		

J7: SATA Power Connector

Pin #	Signal Name
1	+5V
2	Ground
3	Ground
4	+12V

J8 : USB 2.0 Pin Header
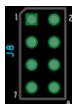
Signal Name	Pin #	Pin #	Signal Name
NC	1	2	GND
NC	3	4	D+
NC	5	6	D-
GND	7	8	+5V

J10: AT_12V Connector
DC-in internal connector supports +12V.

Pin #	Signal Name
1	+12V
2	Ground

Remarks: Do not connect CN12 and J10 at the same time.

FAN1: System Fan Power Connector

Pin header for system fan. The fan must be 12V (Max. 1A).

Pin #	Signal Name
1	Ground
2	+12V
3	Rotation Control

LED1, LED2, LED3, LED4: LAN Port Link, Active LED

LED5: Status LED

A1 & C1 : Status LED
A2 & C2 : Bypass LED
A3 & C3 : Power LED
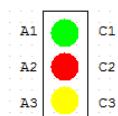
Signal Name	Pin #	Pin #	Signal Name
SIO_GP27	A1	C1	SIO_GP26
ALARM_R	A2	C2	BY_LED_R
PWR_R	A3	C3	GND

Index port: 4E
Data port: 4F
Device: 07
F5h ➜ bit5, bit6 (Control pin)

SW3: Software reset button

Signal Name	Pin #	Pin #	Signal Name
GND	1	2	Intel SoC GPIO7

IO Base:
Read memory 0 x fed0e238 and set bit 1 to "1" as (GPI),
    set bit 1 to "0" as (GPO)
Read memory 0 x fed0e238 and check the bit 0 (Control Pin)
Note: SW3 is controlled by GPIO only.

CN1: MicroSD Connector

CN2: Console Port (COM1)
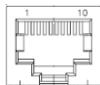
Pin #	Signal Name (RS-232)
1	RTS, Request to send
2	DTR, Data terminal ready
3	TXD, Transmit data
4	Ground
5	Ground
6	RXD, Receive data
7	DSR, Data set ready
8	CTS, Clear to send

CN3: Serial ATA Port

CN4: USB2.0 Ports

CN5, CN6, CN7, CN8: LAN 1G / 100M Port

CN9: SO-DIMM DDR3 Socket

CN10: Compact Flash Connector
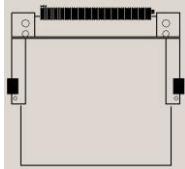
Note: CF card supports IDE mode only.
If CF card applied, please set the SATA configuration to "IDE mode" in BIOS.

CN11: Mini PCI- E Connector (Half size)

CN12 : DC Power Jack (+12V only)
Note: CN12 and J10 cannot be connected at the same time.

BIOS Setup

This chapter describes the different settings available in the AMI BIOS that comes with the board. The topics covered in this chapter are as follows:

BIOS Introduction

The BIOS (Basic Input/Output System) installed in your computer system's ROM supports Intel processors. The BIOS provides critical low-level support for a standard device such as disk drives, serial ports and parallel ports. It also password protection as well as special support for detailed fine-tuning of the chipset controlling the entire system.

BIOS Setup

The BIOS provides a Setup utility program for specifying the system configurations and settings. The BIOS ROM of the system stores the Setup utility. When you turn on the computer, the BIOS is immediately activated. Pressing the <Del> key immediately allows you to enter the Setup utility. If you are a little bit late pressing the <Del> key, POST (Power On Self Test) will continue with its test routines, thus preventing you from invoking the Setup. If you still wish to enter Setup, restart the system by pressing the "Reset" button or simultaneously pressing the <Ctrl>, <Alt> and <Delete> keys. You can also restart by turning the system Off and back On again. The following message will appear on the screen:

```
Press  <DEL>  to  Enter  Setup
```

In general, you press the arrow keys to highlight items, <Enter> to select, the <PgUp> and <PgDn> keys to change entries, <F1> for help and <Esc> to quit.

When you enter the Setup utility, the Main Menu screen will appear on the screen. The Main Menu allows you to select from various setup functions and exit choices.

Warning: *It is strongly recommended that you avoid making any changes to the chipset defaults. These defaults have been carefully chosen by both AMI and your system manufacturer to provide the absolute maximum performance and reliability. Changing the defaults could cause the system to become unstable and crash in some cases.*

System Date

Set the Date. Use Tab to switch between Data elements.

System Time

Set the Time. Use Tab to switch between Data elements.

Advanced Settings

This section allows you to configure and improve your system and allows you to set up some system features according to your preference.

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

OnBoard LAN PXE Rom   [Disabled]
► ACPI Settings
► Super IO Configuration
► H/W Monitor
► LAN Bypass Configuration
► Serial Port Console Redirection
► CPU Configuration
► PPM Configuration
► IDE Configuration
► SDIO Configuration

```
→ ←Select Screen
↑↓ Select Item
Enter: Select
+-  Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save  ESC: Exit
```

OnBoard LAN PXE Rom
Controls the execution of UEFI and Legacy PEX OpROM.

ACPI Settings

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

ACPI Settings

Enable ACPI Auto Configuration   [Disabled]

Enable Hibernation   [Enabled]
ACPI Sleep State   S3 (Suspend to RAM)

```
→ ←Select Screen
↑↓ Select Item
Enter: Select
+-  Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save  ESC: Exit
```

Enabled ACPI Auto Configuration
Enables or Disables BIOS ACPI Auto Configuration.

Enable Hibernation
Enables or Disables System ability to Hibernate (OS/S4 Sleep State). This option may be not effective with some OS.

ACPI Sleep State
Select the highest ACPI sleep state the system will enter, when the SUSPEND button is pressed.

Super IO Configuration

Set Parameters of Serial Ports. User can Enable/Disable the serial port and Select an optimal settings for the Super IO Device.

H/W Monitor

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
PC Health Status					
Smart Fan Function		[50]			
System temperature		+33 C			
CPU temperature		+37 C			
System Fan Speed		3600 RPM			
VCORE		+0.840 V			
+1.35V		+1.368 V			→ ←Select Screen
AVCC		+3.388 V			↑↓ Select Item
VSB3		+3.376V			Enter: Select
VCC3V		+3.388 V			+- Change Field
VBAT		+3.280 V			F1: General Help
					F2: Previous Values
CPU Shutdown Temperature		[Disabled]			F3: Optimized Default
					F4: Save ESC: Exit

Smart Fan Function

50 ℃

60 ℃

70 ℃

80 ℃

90 ℃

CPU Shutdown Temperature

The default setting is disabled.

Ethernet Bypass Configuration

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

LAN Bypass Configuration

Firmware Information
Signature   ETH-BYPASS
Version   2.1

Bypass Quick Setting   [Normal]

Bypass Quick Setting

Set LAN Bypass to Normal, Bypass, Firewall or Custom Define Mode

Normal mode: All LAN ports in NORMAL. When Watchdog monitor system hangs, software will initiates a system reboot.
Bypass mode: All LAN ports in BYPASS during power-off or watchdog initiates Bypass. System will not reboot.
Firewall mode: All LAN ports in BYPASS until software change it to NORMAL under OS. When watchdog monitors system hang, software will initiates a system reboot.
Custom Define mode: Customer defines watchdog reset, watchdog Bypass and power-off Bypass settings.

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

LAN Bypass Configuration

Bypass Quick Setting   [Custom Define]

Watchdog Reset Signal   [Disabled]

WDT Bypass Setting
LAN3 LAN4 Bypass   [Normal]

System OFF Bypass Setting
LAN3 LAN4 Bypass   [Normal]

Serial Port Console Redirection

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

COM0

Console Redirection	[Enabled]	→ ←Select Screen
►Console Redirection Setting		↑↓ Select Item
		Enter: Select
Serial Port for Out-of Band Management /		+- Change Field
Windows Emergency Management Services (EMS)		F1: General Help
		F2: Previous Values
Console Redirection	[Disabled]	F3: Optimized Default
►Console Redirection Settings		F4: Save ESC: Exit

Console Redirection Setting

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

COM0

Console Redirection Settings

		→ ←Select Screen
Terminal Type	[VT100]	↑↓ Select Item
Bits per second	[115200]	Enter: Select
Data Bits	[8]	+- Change Field
Parity	[None]	F1: General Help
Stop Bits	[1]	F2: Previous Values
Flow Control	[None]	F3: Optimized Default
VT-UTF8 Combo Key Support	[Enabled]	F4: Save ESC: Exit
Recorder Mode	[Disabled]	
Resolution 100x31	[Disabled]	
Legacy OS Redirection Resolution	[80x24]	
Putty KeyPad	[VT100]	
Redirection After BIOS POST	[Always Enable]	

CPU Configuration

This section shows the CPU configuration parameters.

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

Socket 0 CPU Information

Intel(R) Celeron(R) CPU J1900 @ 1.99 GHz

Processor Speed	1990 MHz	→ ←Select Screen
Processor Cores	4	↑↓ Select Item
Intel HT Technology	Not Supported	Enter: Select
Intel VT-X Technology	Supported	+- Change Field
		F1: General Help
L1 Data Cache	24 kB x 4	F2: Previous Values
L2 Data Cache	32 kB x 4	F3: Optimized Default
L2 Cache	1024 kB x2	F4: Save ESC: Exit

PPM Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

PPM Configuration

EIST   [Enabled]

→ ←Select Screen
↑↓ Select Item
Enter: Select
+- Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save ESC: Exit

IDE Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

IDE Configuration

Serial-ATA (SATA)   [Enable]

SATA Speed Support   [Gen2]
SATA Mode   [IDE Mode]

Serial-STA Port 0   [Enable]
SATA Port0 HotPlug   [Disabled]

Serial-STA Port 1   [Enable]
SATA Port1 HotPlug   [Disabled]

SATA Port0
TRANSCEND   (4.0GB)

SATA Port1
SAMSUNG 470 Se   (64.0GB)

→ ←Select Screen
↑↓ Select Item
Enter: Select
+- Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save ESC: Exit

SATA Controller(s)
Enable / Disable Serial ATA Controller.

Configure SATA as
(1) IDE Mode.
(2) AHCI Mode.

SDIO Configuration

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

SDIO Configuration

SDIO Access Mode   [Auto]

```
→ ←Select Screen
↑↓ Select Item
Enter: Select
+-  Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save  ESC: Exit
```

Auto: Access SD device in DMA mode if controller supports it,
       otherwise in PIO mode.

DMA: Access SE device in DMA mode.

PIO: Access SD device in PIO mode.

Chipset Settings

This section allows you to configure and improve your system and allows you to set up some system features according to your preference.

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

► North Bridge
► South Bridge

```
→ ←Select Screen
↑↓ Select Item
Enter: Select
+-  Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save  ESC: Exit
```

North Bridge

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

Memory Information

Total Memory   2048 MB (LPDDR3)

Memory Slot0   2048 MB (LPDDR3)

```
→ ←  Select Screen
↑↓ Select Item
Enter: Select
+-  Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save  ESC: Exit
```

South Bridge

Main	Advanced	Chipset	Boot	Security	Save & Exit
					→ ← Select Screen
Restore AC Power Loss	[Power On]				↑↓ Select Item
					Enter: Select
					+- Change Field
					F1: General Help
					F2: Previous Values
					F3: Optimized Default
					F4: Save ESC: Exit

Power On: motherboard will Power ON when DC power source plugged.
Power Off: motherboard will Power ON when power switch (SW2)
        pushed.

Security Settings

This section allows you to configure and improve your system and allows you to set up some system features according to your preference.

Aptio Setup Utility

Main	Advanced	Chipset	Security	Boot	Save & Exit
Password Description					
If ONLY the Administrator's password is set, then this only limits access to Setup and is only asked for when entering Setup. If ONLY the User's password is set, then this is a power on password and must be entered to boot or enter Setup. In Setup the User will have Administrator rights. The password length must be in the following range:					→ ←Select Screen
					↑↓ Select Item
Minimum length		3			Enter: Select
Maximum length		20			+- Change Field
					F1: General Help
Administrator Password					F2: Previous Values
User Password					F3: Optimized Default
					F4: Save ESC: Exit
HDD Security Configuration:					
HDD0: TRANSCEND					
HDD1: SAMSUNG 470					

Administrator Password

Set Setup Administrator Password.

User Password

Set User Password.

Boot Settings

Aptio Setup Utility

Main	Advanced	Chipset	**Boot**	Security	Save & Exit

Boot Configuration

Setup Prompt Timeout   10

Bootup NumLock State   [On]

Quiet Boot   [Enabled]

Fast Boot   [Disabled]

Boot Option Priorities

Boot Option #1   [PATA PM: TRANS..]

Boot Option #2   [Disabled]

Hard Drive BBS Priorities

```
→ ←Select Screen
↑↓ Select Item
Enter: Select
+-  Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save  ESC: Exit
```

Boot Option Priorities
Sets the system boot order.

Hard Drive BBS Priorities
Sets the SATA devices boot order.

Save & Exit Settings

Aptio Setup Utility

Main	**Advanced**	Chipset	Boot	Security	Save & Exit

Save Changes and Exit

Discard Changes and Exit

Save Changes and Reset

Discard Changes and Reset

Save Options

Save Changes

Discard Changes

Restore Defaults

Save as User Defaults

Restore User Defaults

Boot Override

```
→ ←Select Screen
↑↓ Select Item
Enter: Select
+-  Change Field
F1: General Help
F2: Previous Values
F3: Optimized Default
F4: Save  ESC: Exit
```

Save Changes and Exit
Exit system setup after saving the changes.

Discard Changes and Exit

Exit system setup without saving any changes.

Save Changes and Reset

Reset the system after saving the changes.

Discard Changes and Reset

Reset system setup without saving any changes.

Save Changes

Save Changes done so far to any of the setup options.

Discard Changes

Discard Changes done so far to any of the setup options.

Restore Defaults

Restore/Load Defaults values for all the setup options.

Save as User Defaults

Save the changes done so far as User Defaults.

Restore User Defaults

Restore the User Defaults to all the setup options.

Boot Override

Pressing ENTER causes the system to enter the OS.

Drivers Installation

This section describes the installation procedures for software and drivers. The software and drivers are included with the motherboard. If you find the items missing, please contact the vendor where you made the purchase. The contents of this section include the following:

IMPORTANT NOTE:
After installing your Windows operating system, you must install first the Intel Chipset Software Installation Utility before proceeding with the drivers installation.

Intel Chipset Software Installation Utility

The Intel Chipset Drivers should be installed first before the software drivers to enable Plug & Play INF support for Intel chipset components. Follow the instructions below to complete the installation.

1. Insert the disc that comes with the board. Click *Intel* and then *Intel(R) Cedarview Chipset Drivers*.

2. Click *Intel(R) Chipset Software Installation Utility*.

3. When the Welcome screen to the Intel® Chipset Device Software appears, click *Next* to continue.

4. Click *Yes* to accept the software license agreement and proceed with the installation process.

5. On the Readme File Information screen, click *Next* to continue the installation.

6. The Setup process is now complete. Click *Finish* to restart the computer and for changes to take effect.

VGA Drivers Installation

1. Insert the disc that comes with the board. Click *Intel* and then *Intel(R) Cedarview Chipset Drivers*.
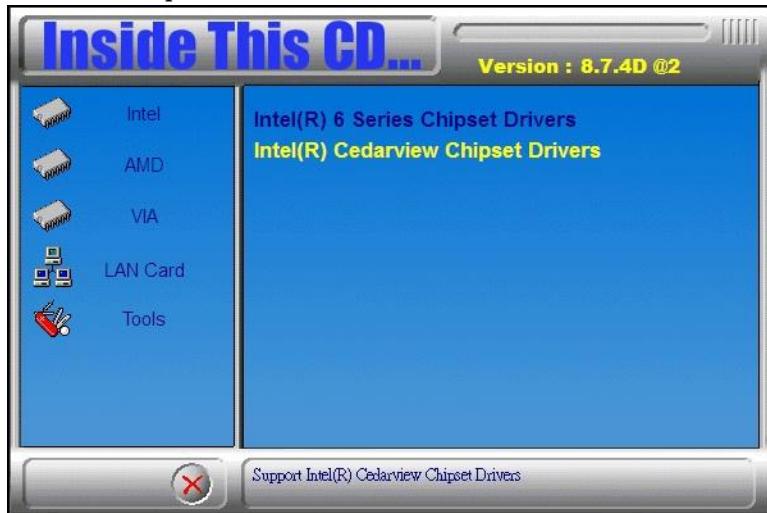
2. Click *Intel(R) Cedarview Graphics Driver.*
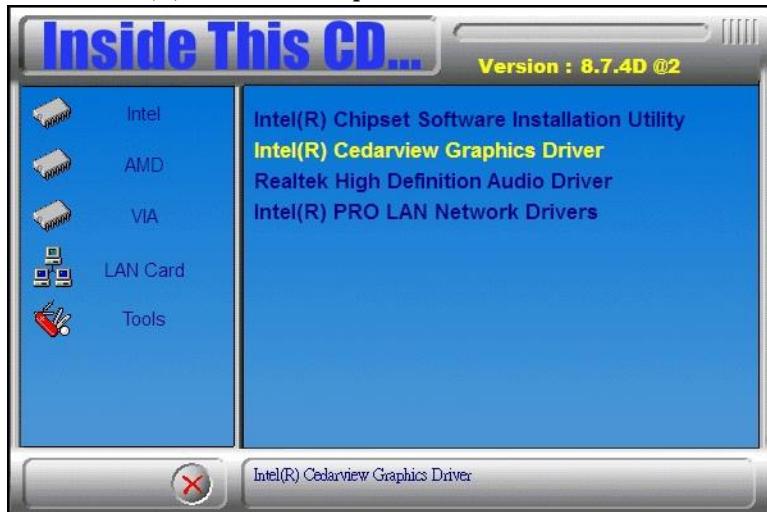
3. When the Welcome screen appears, click *Next* to continue.

4. Click *Yes* to to agree with the license agreement and continue the installation.

5. On the Readme File Information screen, click *Next* to continue the installation of the Intel® Graphics Media Accelerator Driver.

6. On Setup Progress screen, click *Next* to continue.

7. Setup complete. Click *Finish* to restart the computer and for changes to take effect.

LAN Drivers Installation

1. Insert the disc that comes with the board. Click *LAN Card* and then *Intel LAN Cedarview Drivers*.

2. On the next screen, click *Install Drivers and Software*.
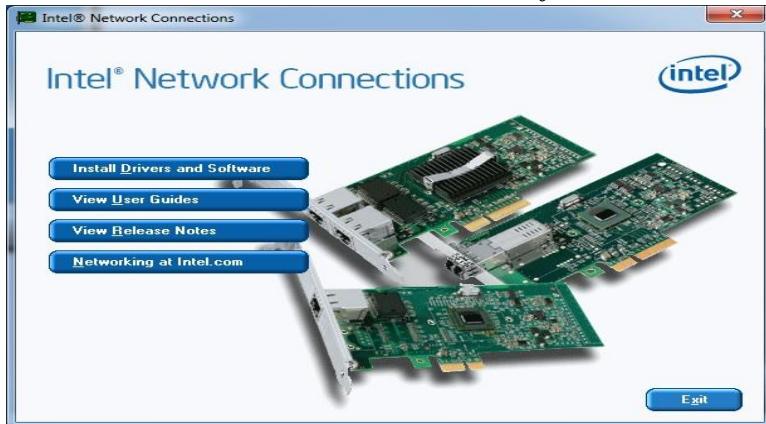
3. In the Welcome screen, click *Next*. On the next screen, click *Yes* to to agree with the license agreement.

4. Agree with the license agreement and click *Next*.

5. Click the checkbox for **Drivers** in the Setup Options screen to select it and click **Next** to continue.
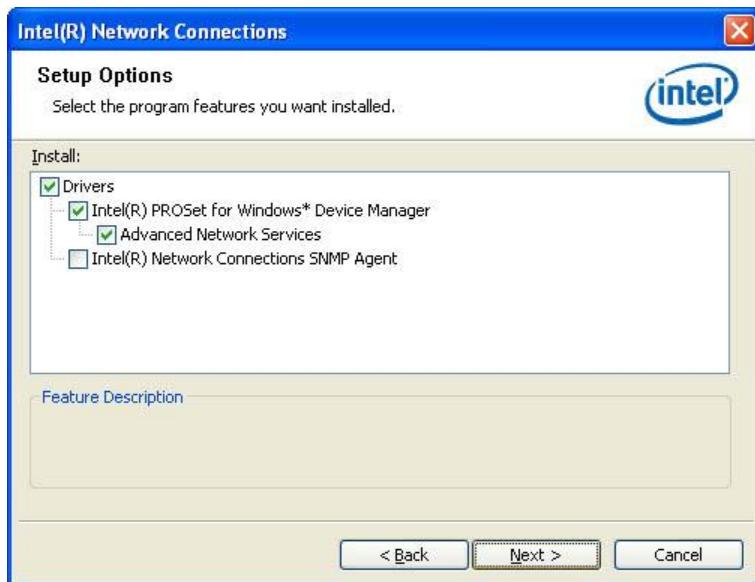
6. The wizard is ready to begin installation. Click *Install* to begin the installation.

7. When InstallShield Wizard is complete, click *Finish*.

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
000h - 01Fh	DMA Controller #1
020h - 03Fh	Interrupt Controller #1
040h - 05Fh	Timer
060h - 06Fh	Keyboard Controller
070h - 07Fh	Real Time Clock, NMI
080h - 09Fh	DMA Page Register
0A0h - 0BFh	Interrupt Controller #2
0C0h - 0DFh	DMA Controller #2
0F0h	Clear Math Coprocessor Busy Signal
0F1h	Reset Math Coprocessor
1F0h - 1F7h	IDE Interface
2F8h - 2FFh	Serial Port #2(COM2)
2B0h- 2DFh	Graphics adapter Controller
360h - 36Fh	Network Ports
3F8h - 3FFh	Serial Port #1(COM1)

B. 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 PS/2 Keyboard
IRQ 3	Communications Port (COM2)
IRQ 4	Communications Port (COM1)
IRQ 5	Communications Port (COM3)
IRQ 7	Communications Port (COM4)
IRQ 8	System CMOS/real time clock
IRQ 11	Intel(R) N10/ICH7 Family SMBus Controller - 27DA
IRQ 12	Microsoft PS/2 Mouse
IRQ 13	Numeric data processor
IRQ 16	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CB
IRQ 18	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CA
IRQ 19	Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
IRQ 19	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C9
IRQ 22	High Definition Audio Controller
IRQ 23	Intel(R) N10/ICH7 Family USB2 Enhanced Host Controller - 27CC
IRQ 23	Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C8
IRQ 4294967290	Intel(R) 82583V Gigabit Network Connection
IRQ 4294967291	Intel(R) 82583V Gigabit Network Connection #2
IRQ 4294967292	Intel(R) Graphics Media Accelerator 3600 Series
IRQ 4294967293	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D2
IRQ 4294967294	Intel(R) N10/ICH7 Family PCI Express Root Port - 27D0

C. Watchdog Timer Configuration

The WDT is used to generate a variety of output signals after a user programmable count. The WDT is suitable for use in the prevention of system lock-up, such as when software becomes trapped in a deadlock. Under these sorts of circumstances, the timer will count to zero and the selected outputs will be driven. Under normal circumstance, the user will restart the WDT at regular intervals before the timer counts to zero.

SAMPLE CODE:

File of the NCT5523D.H

```
//-----------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//-----------------------------------------------------------------------
#ifndef __NCT5523D_H
#define __NCT5523D_H            1
//-----------------------------------------------------------------------
#define     NCT5523D_INDEX_PORT        (NCT5523D_BASE)
#define     NCT5523D_DATA_PORT         (NCT5523D_BASE+1)
//-----------------------------------------------------------------------
#define     NCT5523D_REG_LD            0x07
//-----------------------------------------------------------------------
#define NCT5523D_UNLOCK           0x87
#define     NCT5523D_LOCK              0xAA
//-----------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//-----------------------------------------------------------------------
#endif//__NCT5523D_H
```

File of the MAIN.CPP.

```
//-------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//-------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "NCT5523D.H"
//-------------------------------------------------------------------------
int main (void);

void WDTInitial(void);
void WDTEnable(unsigned char);
void WDTDisable(void);


//-------------------------------------------------------------------------
int main (void)
{
      char SIO;

      SIO = Init_NCT5523D();
      if (SIO == 0)
      {
            printf("Can not detect Nuvoton NCT5523D, program abort.\n");
            return(1);
      }

      WDTInitial();

      WDTEnable(10);

      WDTDisable();

      return 0;
}
//-------------------------------------------------------------------------
void WDTInitial(void)
{
      unsigned char bBuf;
      Set_NCT5523D_LD(0x08);                              //switch to logic device 8
      bBuf = Get_NCT5523D_Reg(0x30);
      bBuf &= (~0x01);
      Set_NCT5523D_Reg(0x30, bBuf);                       //Enable WDTO
}
//-------------------------------------------------------------------------
```

```
void WDTEnable(unsigned char NewInterval)
{
      unsigned char bBuf;

      Set_NCT5523D_LD(0x08);                                    //switch to logic device 8
      Set_NCT5523D_Reg(0x30, 0x01);                  //enable timer

      bBuf = Get_NCT5523D_Reg(0xF0);
      bBuf &= (~0x08);
      Set_NCT5523D_Reg(0xF0, bBuf);                   //count mode is second

      Set_NCT5523D_Reg(0xF1, NewInterval);        //set timer
}
//--------------------------------------------------------------------------
void WDTDisable(void)
{
      Set_NCT5523D_LD(0x08);                                    //switch to logic device 8
      Set_NCT5523D_Reg(0xF1, 0x00);                    //clear watchdog timer
      Set_NCT5523D_Reg(0x30, 0x00);                    //watchdog disabled
}
//--------------------------------------------------------------------------
```

File of the NCT5523D.CPP

```cpp
//-------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//-------------------------------------------------------------------------
#include "NCT5523D.H"
#include <dos.h>
//-------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//-------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
      unsigned int result;
      unsigned char ucDid;

      NCT5523D_BASE = 0x4E;
      result = NCT5523D_BASE;

      ucDid = Get_NCT5523D_Reg(0x20);
      if (ucDid == 0xC4)                            //NCT5523D??
      {      goto Init_Finish;    }

      NCT5523D_BASE = 0x2E;
      result = NCT5523D_BASE;

      ucDid = Get_NCT5523D_Reg(0x20);
      if (ucDid == 0xC4)                            //NCT5523D??
      {      goto Init_Finish;    }

      NCT5523D_BASE = 0x00;
      result = NCT5523D_BASE;

Init_Finish:
      return (result);
}
//-------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
      outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
      outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//-------------------------------------------------------------------------
void Lock_NCT5523D (void)
{
      outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//-------------------------------------------------------------------------
```

```
void Set_NCT5523D_LD( unsigned char LD)
{
      Unlock_NCT5523D();
      outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
      outportb(NCT5523D_DATA_PORT, LD);
      Lock_NCT5523D();
}
//-----------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
      Unlock_NCT5523D();
      outportb(NCT5523D_INDEX_PORT, REG);
      outportb(NCT5523D_DATA_PORT, DATA);
      Lock_NCT5523D();
}
//-----------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
      unsigned char Result;
      Unlock_NCT5523D();
      outportb(NCT5523D_INDEX_PORT, REG);
      Result = inportb(NCT5523D_DATA_PORT);
      Lock_NCT5523D();
      return Result;
}
//---------------------------------------------------------------------
```

D. Digital I/O Sample Code

File of the NCT5523D.H
```
//-------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//-------------------------------------------------------------------------
#ifndef __NCT5523D_H
#define __NCT5523D_H              1
//-------------------------------------------------------------------------
#define     NCT5523D_INDEX_PORT          (NCT5523D_BASE)
#define     NCT5523D_DATA_PORT           (NCT5523D_BASE+1)
//-------------------------------------------------------------------------
#define     NCT5523D_REG_LD              0x07
//-------------------------------------------------------------------------
#define NCT5523D_UNLOCK              0x87
#define     NCT5523D_LOCK                0xAA
//-------------------------------------------------------------------------
unsigned int Init_NCT5523D(void);
void Set_NCT5523D_LD( unsigned char);
void Set_NCT5523D_Reg( unsigned char, unsigned char);
unsigned char Get_NCT5523D_Reg( unsigned char);
//-------------------------------------------------------------------------
#endif//__NCT5523D_H
```

File of the MAIN.CPP

```cpp
//-------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//-------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "NCT5523D.H"
//-------------------------------------------------------------------------
int main (void);

void Dio5Initial(void);
void Dio5SetOutput(unsigned char);
unsigned char Dio5GetInput(void);
void Dio5SetDirection(unsigned char);
unsigned char Dio5GetDirection(void);
//-------------------------------------------------------------------------
int main (void)
{
        char SIO;

        SIO = Init_NCT5523D();
        if (SIO == 0)
        {
                printf("Can not detect Nuvoton NCT5523D, program abort.\n");
                return(1);
        }

        Dio5Initial();

        //for GPIO20..27
        Dio5SetDirection(0x0F); //GP20..23 = input, GP24..27=output
        printf("Current DIO direction = 0x%X\n", Dio5GetDirection());

        printf("Current DIO status = 0x%X\n", Dio5GetInput());

        printf("Set DIO output to high\n");
        Dio5SetOutput(0x0F);

        printf("Set DIO output to low\n");
        Dio5SetOutput(0x00);

        return 0;
}
//-------------------------------------------------------------------------
```

```
void Dio5Initial(void)
{
      unsigned char ucBuf;

  ucBuf = Get_NCT5523D_Reg(0x1C);
  ucBuf &= ~0x02;
  Set_NCT5523D_Reg(0x1C, ucBuf);

      Set_NCT5523D_LD(0x07);                              //switch to logic device 7
      //enable the GP2 group
      ucBuf = Get_NCT5523D_Reg(0x30);
      ucBuf |= 0x04;
      Set_NCT5523D_Reg(0x30, ucBuf);
}
//-------------------------------------------------------------------------
void Dio5SetOutput(unsigned char NewData)
{
      Set_NCT5523D_LD(0x07);                              //switch to logic device 7
      Set_NCT5523D_Reg(0xE1, NewData);
}
//-------------------------------------------------------------------------
unsigned char Dio5GetInput(void)
{
      unsigned char result;

      Set_NCT5523D_LD(0x07);                              //switch to logic device 7
      result = Get_NCT5523D_Reg(0xE1);
      return (result);
}
//-------------------------------------------------------------------------
void Dio5SetDirection(unsigned char NewData)
{
      //NewData : 1 for input, 0 for output
      Set_NCT5523D_LD(0x07);                              //switch to logic device 7
      Set_NCT5523D_Reg(0xE8, NewData);
}
//-------------------------------------------------------------------------
unsigned char Dio5GetDirection(void)
{
      unsigned char result;

      Set_NCT5523D_LD(0x07);                              //switch to logic device 7
      result = Get_NCT5523D_Reg(0xE8);
      return (result);
}
//-------------------------------------------------------------------------
```

File of the NCT5523D.CPP

```
//-------------------------------------------------------------------------
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//-------------------------------------------------------------------------
#include "NCT5523D.H"
#include <dos.h>
//-------------------------------------------------------------------------
unsigned int NCT5523D_BASE;
void Unlock_NCT5523D (void);
void Lock_NCT5523D (void);
//-------------------------------------------------------------------------
unsigned int Init_NCT5523D(void)
{
      unsigned int result;
      unsigned char ucDid;

      NCT5523D_BASE = 0x4E;
      result = NCT5523D_BASE;

      ucDid = Get_NCT5523D_Reg(0x20);
      if (ucDid == 0xC4)                               //NCT5523D??
      {      goto Init_Finish;    }

      NCT5523D_BASE = 0x2E;
      result = NCT5523D_BASE;

      ucDid = Get_NCT5523D_Reg(0x20);
      if (ucDid == 0xC4)                               //NCT5523D??
      {      goto Init_Finish;    }

      NCT5523D_BASE = 0x00;
      result = NCT5523D_BASE;

Init_Finish:
      return (result);
}
//-------------------------------------------------------------------------
void Unlock_NCT5523D (void)
{
      outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
      outportb(NCT5523D_INDEX_PORT, NCT5523D_UNLOCK);
}
//-------------------------------------------------------------------------
void Lock_NCT5523D (void)
{
      outportb(NCT5523D_INDEX_PORT, NCT5523D_LOCK);
}
//-------------------------------------------------------------------------
```

```
void Set_NCT5523D_LD( unsigned char LD)
{
      Unlock_NCT5523D();
      outportb(NCT5523D_INDEX_PORT, NCT5523D_REG_LD);
      outportb(NCT5523D_DATA_PORT, LD);
      Lock_NCT5523D();
}
//----------------------------------------------------------------------------
void Set_NCT5523D_Reg( unsigned char REG, unsigned char DATA)
{
      Unlock_NCT5523D();
      outportb(NCT5523D_INDEX_PORT, REG);
      outportb(NCT5523D_DATA_PORT, DATA);
      Lock_NCT5523D();
}
//----------------------------------------------------------------------------
unsigned char Get_NCT5523D_Reg(unsigned char REG)
{
      unsigned char Result;
      Unlock_NCT5523D();
      outportb(NCT5523D_INDEX_PORT, REG);
      Result = inportb(NCT5523D_DATA_PORT);
      Lock_NCT5523D();
      return Result;
}
//----------------------------------------------------------------------------
```

E. Register of the LAN Bypass Controller

To fulfill the varied requests on LAN Bypass controller, IBASE provide the smart LAN Bypass controller. User can define the Bypass function behavior when the system is power-on, power-off and WDT signal is asserted.

The controller is behind the SMBus controller. The I^2C address is listed as below:

	I^2C Address (8bit)	Remark
1st Controller	0x68	
2nd Controller	0x6A	Optional

CR 0x22 : System-On Bypass Register

Attribute : Read / Write
Reset default : 0x0000

Bit	LAN Port#	Read / Write	Description
0	Eth3, 4	Read / Write	Enable / Disable LAN Bypass function when the system is power On. 1 = Enable LAN Bypass function 0 = Disable LAN Bypass function

CR 0x24 : System-Off Bypass Register

Attribute : Read / Write
Reset default : 0x0000

Bit	LAN Port#	Read / Write	Description
0	Eth3, 4	Read / Write	Enable / Disable LAN Bypass function when the system is power Off. 1 = Enable LAN Bypass function 0 = Disable LAN Bypass function

CR 0x26 : Watchdog (WDT) WDT_IN# Signal Control Register

Attribute : Read / Write

Reset default : 0x00

Bit	Read / Write	Description
0	Read / Write	**WDT_OUT# Generator** The capacity use the WDT to reset the system 1 = Generate 100ms pulse to reset signal when WDT signal is asserted. 0 = Ignore the WDT signal.
1	Read / Write	**WDT LAN Bypass Enable** The capacity use the WDT to set LAN Bypass function 1 = Enable LAN Bypass function when the WDT signal is asserted. **CR 0x28** and **CR 0x2A** will be available if this bit is set to "1". 0 = Disable WDT LAN Bypass function.
2 ~ 7		Reserved

CR 0x28 : Watchdog (WDT) Bypass Control Register

Attribute : Read / Write

Reset default : 0x0000

Bit	LAN Port#	Read / Write	Description
0	Eth3, 4	Read / Write	Enable / Disable WDT Bypass function for each LAN port. 1 = Follow the setting in "WDT Bypass Register **CR 0x2A**" when the WDT signal is asserted. 0 = Ignore to control the bypass when the WDT is asserted.

CR 0x2A : Watchdog (WDT) Bypass Register

Attribute : Read / Write

Reset default : 0x0000

Bit	LAN Port#	Read / Write	Description
0	Eth3, 4	Read / Write	The function works when the bit in **CR 0x28** is "1". It controls LAN Bypass function should be Enabled / Disabled when the WDT signal is asserted. If the bit is set to "1" in "WDT Bypass Control Register **CR 0x28**", it will follow below setting: 1 = Enable LAN Bypass function 0 = Disable LAN Bypass function