MB968

8-port (Haswell + C226)
Networking Motherboard

USER'S MANUAL

Version 1.2

Acknowledgments

Table of Contents

Introduction

Product Description

The MB968 networking motherboard is based on the latest Intel® C226 chipset. The platform supports LGA1150 Haswell processors. Four DDR3 UDIMM sockets allows up to 32GB system memory.

The motherboard supports a total of eight Ethernet ports with the port 5 to port8 supporting Bypass function. It also has two fast SATA III 6Gbps ports, two USB 3.0 ports, two USB 2.0 ports and one USB 2.0 for Mini PCI-e. MB968 utilizes the dramatic increase in performance provided this Intel's latest cutting-edge technology. Measuring 355mm x 185mm, MB968 features Intel® Active Management Technology 8.0, offers eight GbE LAN ports, two fast 6Gbps SATA ports and two USB3.0 ports.

MB968 Features

- Supports Intel® Shark Bay DT LGA1150 Haswell processors
- Supports Intel® Lynx Point C226 PCH chipset
- Four DDR3 UDIMM for maximum 32GB total, 1600MHz, Unbuffered
- Intel CPU integrated graphics
- Intel® Clarkville I217LM GbE PHY for Eth1, No Bypass
  Eth2~4: Intel® Springville I210-AT, No Bypass
  Eth5~6: Intel® Springville I210-AT, supports Bypass
  Eth7~8: Intel® Springville I210-AT, supports Bypass
- Two SATA III (6.0Gb/s)
- One cableless 2.5" HDD (use IP332)
- Optional 3.5" HDD x1, 2.5" HDD x2

Specifications

Product Name	MB968
Processor	• Support for Intel® Shark Bay DT LGA1150 Haswell processors • TDP = 35W ~ 95W (DC / QC)
CPU Speed	TBD
Cache Size	Up to 8MB
Chipset	Intel® Lynx Point C226 PCH Package =23 mm x 22 mm , 0.65 mm ball pitch
BIOS	AMI BIOS
Memory	• Four DDR3 UDIMM total for 32GB max memory (4Gb chip support) • Support DDR3 at 1.5V • Dual channel DDR3 up to 1600 MHz • Unbuffered • ECC or non-ECC
Video	• Intel® CPU integrated graphics • IBASE VGA4 pin header on board
Network Controller	• Eth1: Intel® Clarkville I217LM GbE PHY, 6mm x 6mm, QFN48 with iAMT 9.0 supporting. No Bypass • Eth2~4: Intel® Springville I210-AT. No Bypass. • Eth5~6: Intel® Springville I210-AT.. Support Bypass. • Eth7~8: Intel® Springville I210-AT.. Support Bypass.
SATA Ports	• Two SATA III (6.0Gb/s), 7-pin SATA Blue connector • One for Front PCI-e Golden Finger #1 (to IP332) • Two for mSATA (Mini PCI-e) • One for CF Card
Fan Connector	4-pin smart fan connectors: • Three for CPU Fan (Smart Fan reference to CPU temperature in BIOS) • One for System Fans (Smart Fan reference to CPU temperature in BIOS))
Compact Flash	Marvell 88SA8052 SATA to PATA for Compact Flash type II
Digital IO	4 in & 4 out, 2x5 pin-header
Front Panel LED	#1 LED: Power (Green = Power On, Off= No Power) #2 LED: Bypass or HDD (Jumper Select) Bypass: Green = LAN 5-6 or 7-8 Bypass, Off = LAN Normal #3 LED: Status (GPIO control, Yellow / Red)

<table>
<tr><td>USB Ports</td><td>

Two USB 3.0 + 2.0 ports at front panel
One USB 2.0 for Mini PCI-e
Six USB 2.0 pin headers (pitch 2.54)

</td></tr>
<tr><td>Network Bypass</td><td>

Two segment hardware Bypass (Eth5 & 6; Eth7 & 8)
Bypass mode selection in BIOS

</td></tr>
<tr><td>LPC I/O</td><td>Fintek F81866AD-I (128-pin LQFP [14mm x 14 mm])

COM1: RJ-45 Console x1
COM2~4: RS-232 [2x5] Pin Header Onboard x3
Hardware monitors
Fan Connector x4
Digital IO 4in & 4 Out

</td></tr>
<tr><td>Smart Fan Control</td><td>The active temperature may be adjusted based on system thermal test result. All reference to CPU temperature.
<table>
<tr><td></td><td>Active Temperature</td><td>Tolerance</td><td>Default Smart Fan Enable or Disable</td></tr>
<tr><td>CPU Fan</td><td>50</td><td>+/- 5</td><td>Enable</td></tr>
<tr><td>System Fan</td><td>50</td><td>+/- 5</td><td>Enable</td></tr>
</table>
</td></tr>
<tr><td>RTC</td><td>Intel C226 built-in RTC with on-board lithium battery & holder</td></tr>
<tr><td>Expansion Slot (CPU PEG port)</td><td>CPU jumper setting PEG port to following configurations for golden fingers & riser cards
<table>
<tr><td>Configuration</td><td>Golden Finger #1 Compatible Riser Cards</td><td>Golden Finger #2 Compatible Riser Cards</td></tr>
<tr><td>1</td><td>IP332</td><td>IP331</td></tr>
<tr><td>2</td><td>IP333</td><td>IP333</td></tr>
</table>
</td></tr>
<tr><td>Expansion Interface</td><td>

Mini PCI-e Socket x1 (m-SATA compatible)
Mini PCI-e Socket x1 (support m-SATA only)

</td></tr>
<tr><td>Front Panel Buttons & Connector</td><td>

Two RJ-45 1x4 connectors for Eth1~4 & 5~6
USB 3.0 x2
RJ-45 (for console, COM1)
Three LEDs for Power, Bypass or HDD & Status
Factory Mode Restore Reset Switch

</td></tr>
<tr><td>Rear I/O interface</td><td>

PSU AC inlet
1x or 2x Slot (Depend on product SKU)

</td></tr>
<tr><td>Jumper / Pin Header / Switch</td><td>

AT or ATX mode selection jumper
ATX mode power on / off pin header
Power on LED pin header
HDD active LED pin header
System Reset pin header
Clear CMOS
Clear ME RTC
Golden finger (1x16 or 2x8) switch jumper

</td></tr>
</table>

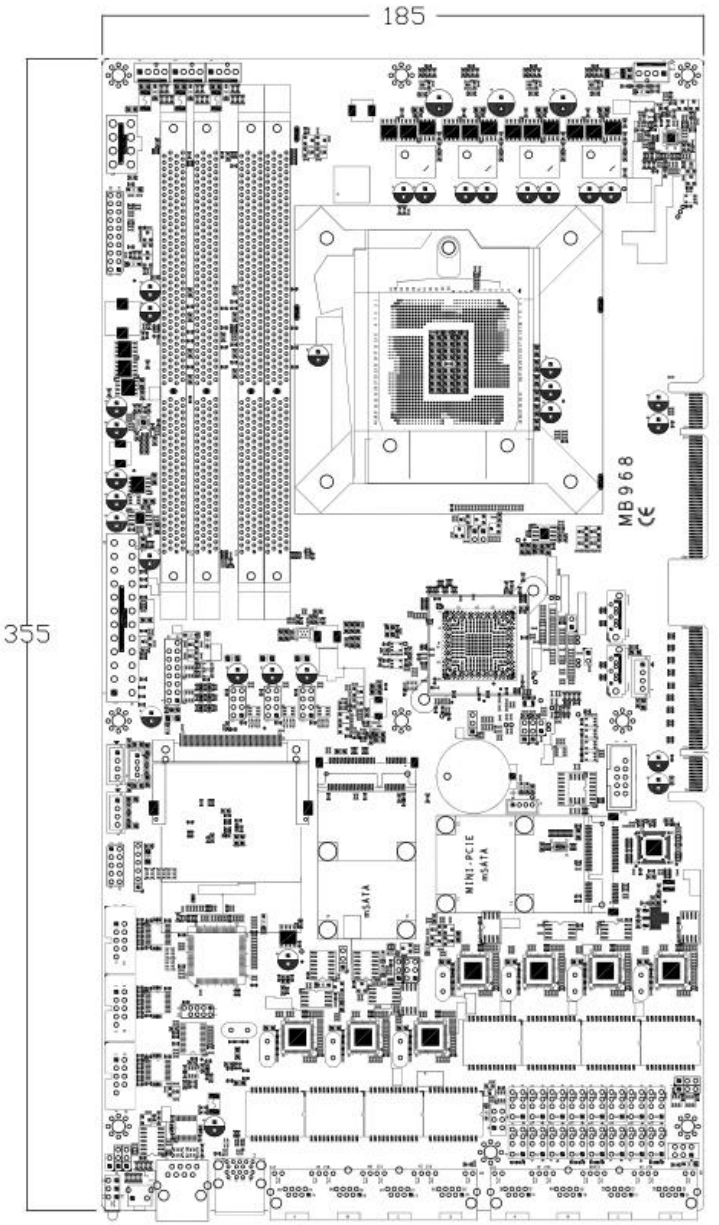
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
TPM	Nuvoton WPCT210AA0WX
Operating Temperature	0°C ~ 45°C
Storage Temperature	-20°C ~ 70°C
Operational Humidity	5% ~ 95% Relative Humidity (non-condensing)
RoHS Compliant	Yes
Board Size	355 x 185mm, FR-4, 1.6mm thickness
Compatible Cards	• IBP161: 4-port Realtek GbE LAN Card • IBP162: 2-port Intel 10GbE Fiber LAN Card • IBP163: 2+2 ports Intel I350-AM2 Fiber + Copper LAN Card • IBP164: Crypto acceleration Card • IBP165: 4-port Intel® I210-AT GbE LAN Card • IBP167: 8-port Intel® I350-AM4 GbE LAN Card • IP331: PCI-e 1-to-1 Riser Card • IP332: PCI-e Adapter Card (with 2.5” HDD Interface) • IP333: PCI-e 2-to-2 Riser Card

Checklist

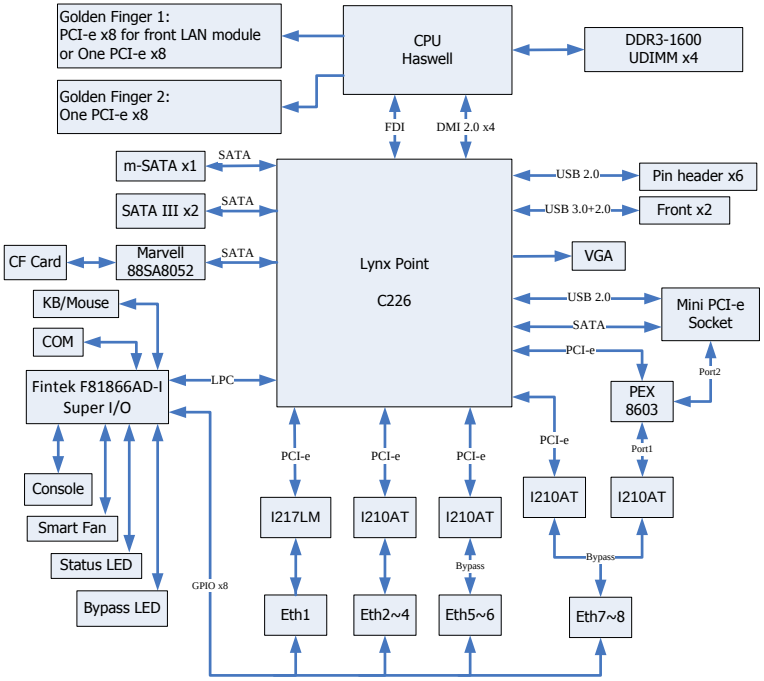
Your MB968 package should include the items listed below.

- MB968 motherboard
- Driver DVD

Board Dimensions



Block Diagram



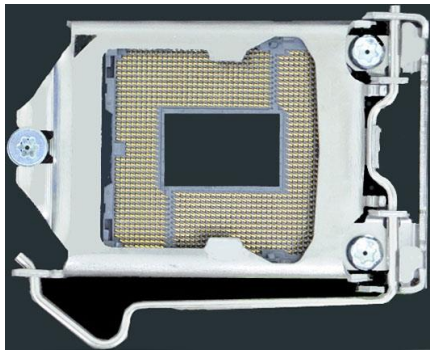
Installations

This section provides information on how to use the jumpers and connectors on the MB968 in order to set up a workable system. The topics covered are:

Installing the CPU

The MB968 board supports an LGA1150 Socket (shown below) for Intel Clarkdale processors.

To install the CPU, unlock first the socket by pressing the lever sideways, then lift it up to a 90-degree. Then, position the CPU above the socket such that the CPU corner aligns with the gold triangle matching the socket corner with a small triangle. Carefully insert the CPU into the socket and push down the lever to secure the CPU. Then, install the heat sink and fan.



***NOTE**: Ensure that the CPU heat sink and the CPU top surface are in total contact to avoid CPU overheating problem that would cause your system to hang or be unstable.*

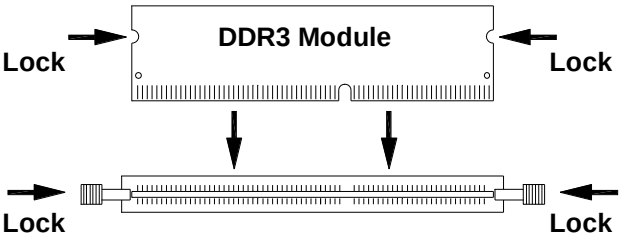
Installing the Memory

The MB968 board supports four DDR3 memory socket for a maximum total memory of 32GB in DDR3 DIMM memory type.

Installing and Removing Memory Modules

To install the DDR3 modules, locate the memory slot on the board and perform the following steps:

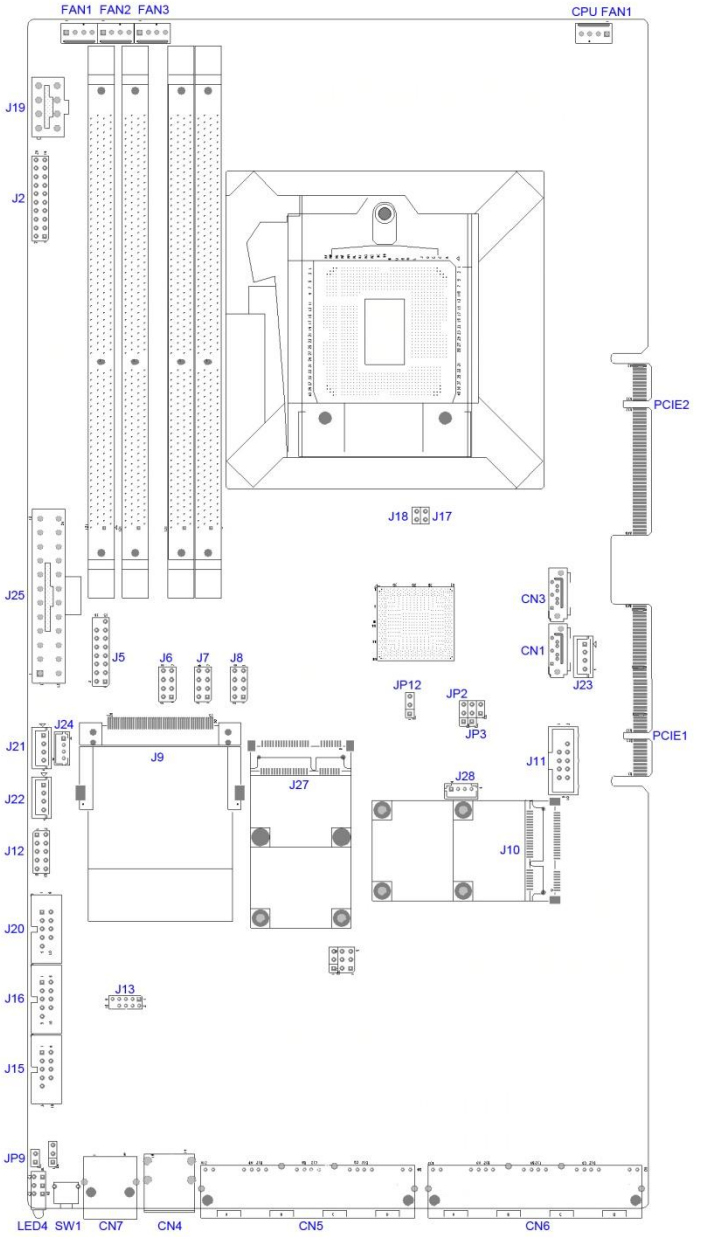
1. Hold the DDR3 module so that the key of the DDR3 module aligned with that on the memory slot.
2. Gently push the DDR3 module in an upright position until the clips of the slot close to hold the DDR3 module in place when the DDR3 module touches the bottom of the slot.
3. To remove the DDR3 module, press the clips with both hands.

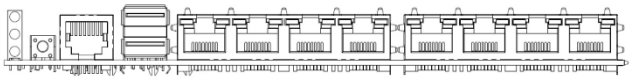


Setting the Jumpers

Jumpers are used on MB968 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 MB968 and their respective functions.

Jumper Locations on MB968





Jumper Settings on MB968

JP2: Clear CMOS Setting

JP2	Setting
	Normal
	Clear CMOS

JP3: Clear ME Setting

JP3	Setting
	Normal
	Clear ME

JP9: AT / ATX Mode Setting

JP9	Setting
	ATX
	AT

JP12: BIOS Flash Security Setting

JP12	Setting
	Normal
	For BIOS Update

JP15: LED Function Selection

JP15	Setting
	HDD Activate
	Bypass Activate

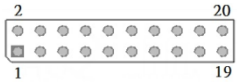
J17, J18: PCIE Config Setting

J18	J17	Setting
		2 x 8 for Golden Finger PCIE1 & PCIE2
		1x16 for Golden Finger PCIE2

Connectors on MB968

J2: System Function Connector

J2 provides connectors for system indicators that provide light indication of the computer activities and switches to change the computer status. J2 is a 20-pin header that provides interfaces for the following functions



Pin 2, 4, 6, 8: Speaker

This connector provides an interface to a speaker for audio tone generation. An 8-ohm speaker is recommended.

Pin #	Signal Name
2	SPEAKER
4	NC
6	GND
8	+5V

Pin 1, 3, 5: Power LED

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

Pin #	Signal Name
1	+5V
3	NC
5	GND

Pin 13, 14: ATX Power ON Switch

This 2-pin connector is an "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
13	GND
14	Power_ON

Pin 17, 18: 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.

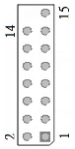
Pin #	Signal Name
17	GND
18	PM_SYSRST#

Pins 19, 20: HDD LED

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
19	+3.3V
20	-HDD_LED

J5: VGA Connectors

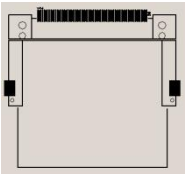


Signal Name	Pin #	Pin #	Signal Name
VGA_R	1	2	VGA_PWR
VGA_G	3	4	GND
VGA_B	5	6	NC
NC	7	8	VGADDCDATA
GND	9	10	HSYNC
GND	11	12	VSYNC
GND	13	14	VGADDCCLK
GND	15		

J6, J7, J8: USB6~USB11 Ports

Signal Name	Pin #	Pin #	Signal Name
+5V	1	2	GND
D-	3	4	D+
D+	5	6	D-
GND	7	8	+5V

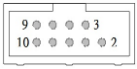
J9: Compact Flash Socket



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

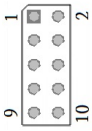
J10: Mini PCI- E / mSATA Socket

J11: SPI Debug Port



Signal Name	Pin #	Pin #	Signal Name
		2	NC
SPI_CS#0	3	4	+3.3V
SPI_SO	5	6	SPI0_HOLD#
SPI0_WP#	7	8	SPI_CLK
GND	9	10	SPI_SI

J12: Digital IO 4-IN / 4-OUT Connector

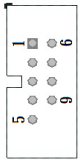


Signal Name	Pin #	Pin #	Signal Name
GND	1	2	+5V
OUT3	3	4	OUT1
OUT2	5	6	OUT0
IN3	7	8	IN1
IN2	9	10	IN0

J13: 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		

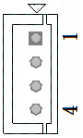
J15, J16, J20: Serial Port (COM1~COM3)



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

J19, J25: ATX Power Connector

J21, J22, J23: Power Connector, Pitch 2.54mm



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

J24: Power Connector, Pitch 2.0mm

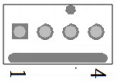
Pin #	Signal Name
1	+5V
2	Ground
3	Ground
4	+12V

J27: mSATA Socket

CN1, CN3: HDD Serial ATA Connector

CPU_FAN1: CPU Fan Connector

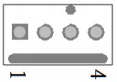
CPU_FAN1 is a 4-pin header for the CPU fan.
The fan must be 12V (Max. 1A).



Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection
4	Rotation control

FAN1, FAN2, FAN3: System Fan Connectors

FAN1, FAN2, FAN3 is a 4-pin header for system fans.
The fan must be 12V (Max. 1A).



Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection
4	Rotation control

LED4: Status LED

A1 & C1 : Status LED
A2 & C2 : Bypass or HDD status LED
A3 & C3 : Power LED

Signal Name	Pin #	Pin #	Signal Name
SIO_GPIO33	A1	C1	SIO_GPIO32
+5 V	A2	C2	JP15 Selection
+3.3 V	A3	C3	GND

SW1: Software reset button

I/O base :
Read IO 0x1C00 and set bit 7 to “1” (Enable GPIO function)
Read IO 0x1C04 and set bit 7 to “1” (GPIO act as GPI)
Read IO 0x1C0C and set check bit 7 (Control Pin)

Signal Name	Pin #	Pin #	Signal Name
GND	1	2	PCH GPIO7

Digital I/O Sample Configuration

Filename：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 "F81865.H"

#define BIT0 0x01
#define BIT1 0x02
#define BIT2 0x04
#define BIT3 0x08
#define BIT4 0x10
#define BIT5 0x20
#define BIT6 0x40
#define BIT7 0x80

//---------------------------------------------------------------------------
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;
	unsigned char DIO;

	printf("Fintek 81865/81866 digital I/O test program\n");

	SIO = Init_F81865();
	if (SIO == 0)
	{
		printf("Can not detect Fintek 81865/81866, program abort.\n");
		return(1);
	}//if (SIO == 0)

	Dio5Initial();
/*
	//for GPIO50..57
	Dio5SetDirection(0xF0);	//GP50..53 = input, GP54..57=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);
*/

	//for GPIO50..57
	Dio5SetDirection(0xF0);	//GP50..53 = input, GP54..57=output

	Dio5SetOutput(0x00);				//clear
//	DIO = Dio5GetInput() & 0x0F;
```

```
        Dio5SetOutput(0x00);                    //clear
        DIO = Dio5GetInput() & 0x0F;
        if (DIO != 0x0A)
        {
                printf("The Fintek 81865 digital IO abnormal, abort.\n");
                return(1);
        }//if (DIO != 0x0A)

        Dio5SetOutput(0xA0);                    //clr# is high
        Dio5SetOutput(0xF0);                    //clk and clr# is high
        Dio5SetOutput(0xA0);                    //clr# is high

        DIO = Dio5GetInput() & 0x0F;
        if (DIO != 0x05)
        {
                printf("The Fintek 81865 digital IO abnormal, abort.\n");
                return(1);
        }
        printf("!!! Pass !!!\n");
        return 0;
}
//-----------------------------------------------------------------------------
void Dio5Initial(void)
{
        unsigned char ucBuf;

        //switch GPIO multi-function pin for gpio 50~57

//gpio53~57 UR5_FULL_EN(bit1), clear UR6_FULL_EN(bit3)
        //set UR5_FULL_EN,should set UR_GP_PROG_EN = 1 (reg26,bit0) first

        ucBuf = Get_F81865_Reg(0x26);
        ucBuf |= BIT0;
        Set_F81865_Reg(0x26, ucBuf);
        //set UR5_FULL_EN(bit1), clear UR6_FULL_EN(bit3)
        ucBuf = Get_F81865_Reg(0x2A);
        ucBuf &= ~BIT3;//clear bit 3,
        ucBuf |= BIT1;//set bit 1,
        Set_F81865_Reg(0x2a, ucBuf);

//GPIO51 ~ GPIO52
        //clear UR6_ALT_EN(bit5),  IR_ALT_EN(bit4),set FDC_GP_EN(bit3)

//GPIO50
//set FDC_GP_EN(bit3), clear  RTS6_ALT_EN(RTS6_2_ALT_EN)(bit6)

        ucBuf = Get_F81865_Reg(0x2A);
        ucBuf &= ~(BIT4+BIT5+BIT6);     //clear UR6_ALT_EN(bit5),  IR_ALT_EN(bit4),
RTS6_ALT_EN(RTS6_2_ALT_EN)(bit6)
        Set_F81865_Reg(0x2a, ucBuf);
        //set FDC_GP_EN(bit3), should clear UR_GP_PROG_EN (reg26,bit0) first
        ucBuf = Get_F81865_Reg(0x26);
        ucBuf &= ~BIT0;
        Set_F81865_Reg(0x26, ucBuf);//clear UR_GP_PROG_EN = 0 (reg26,bit0)

        ucBuf = Get_F81865_Reg(0x2A);
        ucBuf |= BIT3;          //set FDC_GP_EN(bit3),
        Set_F81865_Reg(0x2a, ucBuf);


        Set_F81865_LD(0x06);                                    //switch to logic device 6

        //enable the GP5 group
        ucBuf = Get_F81865_Reg(0x30);
        ucBuf |= 0x01;
        Set_F81865_Reg(0x30, ucBuf);

        Set_F81865_Reg(0xA0, 0x00);                             //define as input mode
        Set_F81865_Reg(0xA3, 0xFF);                             //push pull mode
}
//-----------------------------------------------------------------------------
```

```
void Dio5SetOutput(unsigned char NewData)
{
	Set_F81865_LD(0x06);                                    //switch to logic device 6
	Set_F81865_Reg(0xA1, NewData);
}
//-----------------------------------------------------------------------------
unsigned char Dio5GetInput(void)
{
	unsigned char result;

	Set_F81865_LD(0x06);                                    //switch to logic device 6
	result = Get_F81865_Reg(0xA2);
	return (result);
}
//-----------------------------------------------------------------------------
void Dio5SetDirection(unsigned char NewData)
{
	//NewData : 1 for input, 0 for output
	Set_F81865_LD(0x06);                                    //switch to logic device 6
	Set_F81865_Reg(0xA0, NewData);
}
//-----------------------------------------------------------------------------
unsigned char Dio5GetDirection(void)
{
	unsigned char result;

	Set_F81865_LD(0x06);                                    //switch to logic device 6
	result = Get_F81865_Reg(0xA0);
	return (result);
}
//-----------------------------------------------------------------------------
```

Filename：81865.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 "F81865.H"
#include <dos.h>
//-----------------------------------------------------------------------------
unsigned int F81865_BASE;
void Unlock_F81865 (void);
void Lock_F81865 (void);
//-----------------------------------------------------------------------------
unsigned int Init_F81865(void)
{
	unsigned int result;
	unsigned char ucDid;

	F81865_BASE = 0x4E;
	result = F81865_BASE;

	ucDid = Get_F81865_Reg(0x20);
	if (ucDid == 0x07||ucDid == 0x10)                       //Fintek 81865/66
	{	goto Init_Finish;	}

	F81865_BASE = 0x2E;
	result = F81865_BASE;

	ucDid = Get_F81865_Reg(0x20);
	if (ucDid == 0x07||ucDid == 0x10)                       //Fintek 81865/66
	{	goto Init_Finish;	}

	F81865_BASE = 0x00;
	result = F81865_BASE;

Init_Finish:
```

```
	return (result);
}
//---------------------------------------------------------------------------
void Unlock_F81865 (void)
{
	outportb(F81865_INDEX_PORT, F81865_UNLOCK);
	outportb(F81865_INDEX_PORT, F81865_UNLOCK);
}
//---------------------------------------------------------------------------
void Lock_F81865 (void)
{
	outportb(F81865_INDEX_PORT, F81865_LOCK);
}
//---------------------------------------------------------------------------
void Set_F81865_LD( unsigned char LD)
{
	Unlock_F81865();
	outportb(F81865_INDEX_PORT, F81865_REG_LD);
	outportb(F81865_DATA_PORT, LD);
	Lock_F81865();
}
//---------------------------------------------------------------------------
void Set_F81865_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_F81865();
	outportb(F81865_INDEX_PORT, REG);
	outportb(F81865_DATA_PORT, DATA);
	Lock_F81865();
}
//---------------------------------------------------------------------------
unsigned char Get_F81865_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_F81865();
	outportb(F81865_INDEX_PORT, REG);
	Result = inportb(F81865_DATA_PORT);
	Lock_F81865();
	return Result;
}
//---------------------------------------------------------------------------
```

Filename：81865.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 __F81865_H
#define __F81865_H                    1
//---------------------------------------------------------------------------
#define       F81865_INDEX_PORT           (F81865_BASE)
#define       F81865_DATA_PORT            (F81865_BASE+1)
//---------------------------------------------------------------------------
#define       F81865_REG_LD               0x07
//---------------------------------------------------------------------------
#define F81865_UNLOCK             0x87
#define       F81865_LOCK                       0xAA
//---------------------------------------------------------------------------
unsigned int Init_F81865(void);
void Set_F81865_LD( unsigned char);
void Set_F81865_Reg( unsigned char, unsigned char);
unsigned char Get_F81865_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif //__F81865_H
```

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:

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.

```
//---------------------------------------------------------------------------
//
// 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 "F81866.H"
//---------------------------------------------------------------------------
int main (int argc, char *argv[]);
void EnableWDT(int);
void DisableWDT(void);
//---------------------------------------------------------------------------
int main (int argc, char *argv[])
{
        unsigned char bBuf;
        unsigned char bTime;
        char **endptr;

        char SIO;

        printf("Fintek 81866 watch dog program\n");

        SIO = Init_F81866();
        if (SIO == 0)
        {
                printf("Can not detect Fintek 81866, program abort.\n");
                return(1);
        }//if (SIO == 0)

        if (argc != 2)
        {
                printf(" Parameter incorrect!!\n");
                return (1);
        }

        bTime = strtol (argv[1], endptr, 10);
        printf("System will reset after %d seconds\n", bTime);

        if (bTime)
        {       EnableWDT(bTime); }
        else
        {       DisableWDT();       }
```

```
	return 0;
}

//---------------------------------------------------------------------------
void EnableWDT(int interval)
{
	unsigned char bBuf;

	bBuf = Get_F81866_Reg(0x2B);
	bBuf &= (~0x20);
	Set_F81866_Reg(0x2B, bBuf);			//Enable WDTO

	Set_F81866_LD(0x07);				//switch to logic device 7
	Set_F81866_Reg(0x30, 0x01);			//enable timer

	bBuf = Get_F81866_Reg(0xF5);
	bBuf &= (~0x0F);
	bBuf |= 0x52;
	Set_F81866_Reg(0xF5, bBuf);			//count mode is second

	Set_F81866_Reg(0xF6, interval);		//set timer

	bBuf = Get_F81866_Reg(0xFA);
	bBuf |= 0x01;
	Set_F81866_Reg(0xFA, bBuf);			//enable WDTO output

	bBuf = Get_F81866_Reg(0xF5);
	bBuf |= 0x20;
	Set_F81866_Reg(0xF5, bBuf);			//start counting
}
//---------------------------------------------------------------------------
void DisableWDT(void)
{
	unsigned char bBuf;

	Set_F81866_LD(0x07);				//switch to logic device 7

	bBuf = Get_F81866_Reg(0xFA);
	bBuf &= ~0x01;
	Set_F81866_Reg(0xFA, bBuf);			//disable WDTO output

	bBuf = Get_F81866_Reg(0xF5);
	bBuf &= ~0x20;
	bBuf |= 0x40;
	Set_F81866_Reg(0xF5, bBuf);			//disable WDT
}
//---------------------------------------------------------------------------
//---------------------------------------------------------------------------
//
// 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 "F81866.H"
#include <dos.h>
//---------------------------------------------------------------------------
unsigned int F81866_BASE;
void Unlock_F81866 (void);
void Lock_F81866 (void);
//---------------------------------------------------------------------------
unsigned int Init_F81866(void)
{
	unsigned int result;
	unsigned char ucDid;

	F81866_BASE = 0x4E;
	result = F81866_BASE;
```

```
        ucDid = Get_F81866_Reg(0x20);
        if (ucDid == 0x07)                                      //Fintek 81866
        {       goto Init_Finish;       }

        F81866_BASE = 0x2E;
        result = F81866_BASE;
        ucDid = Get_F81866_Reg(0x20);
        if (ucDid == 0x07)                                      //Fintek 81866
        {       goto Init_Finish;       }

        F81866_BASE = 0x00;
        result = F81866_BASE;

Init_Finish:
        return (result);
}
//-----------------------------------------------------------------------------
void Unlock_F81866 (void)
{
        outportb(F81866_INDEX_PORT, F81866_UNLOCK);
        outportb(F81866_INDEX_PORT, F81866_UNLOCK);
}
//-----------------------------------------------------------------------------
void Lock_F81866 (void)
{
        outportb(F81866_INDEX_PORT, F81866_LOCK);
}
//-----------------------------------------------------------------------------
void Set_F81866_LD( unsigned char LD)
{
        Unlock_F81866();
        outportb(F81866_INDEX_PORT, F81866_REG_LD);
        outportb(F81866_DATA_PORT, LD);
        Lock_F81866();
}
//-----------------------------------------------------------------------------
void Set_F81866_Reg( unsigned char REG, unsigned char DATA)
{
        Unlock_F81866();
        outportb(F81866_INDEX_PORT, REG);
        outportb(F81866_DATA_PORT, DATA);
        Lock_F81866();
}
//-----------------------------------------------------------------------------
unsigned char Get_F81866_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_F81866();
        outportb(F81866_INDEX_PORT, REG);
        Result = inportb(F81866_DATA_PORT);
        Lock_F81866();
        return Result;
}
//-----------------------------------------------------------------------------
//-----------------------------------------------------------------------------
//
// 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 __F81866_H
#define __F81866_H                     1
//-----------------------------------------------------------------------------
#define         F81866_INDEX_PORT               (F81866_BASE)
#define         F81866_DATA_PORT                (F81866_BASE+1)
//-----------------------------------------------------------------------------
#define         F81866_REG_LD                   0x07
//-----------------------------------------------------------------------------
```

```
#define F81866_UNLOCK                   0x87
#define         F81866_LOCK                             0xAA
//---------------------------------------------------------------------------
unsigned int Init_F81866(void);
void Set_F81866_LD( unsigned char);
void Set_F81866_Reg( unsigned char, unsigned char);
unsigned char Get_F81866_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif //__F81866_H
```

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:

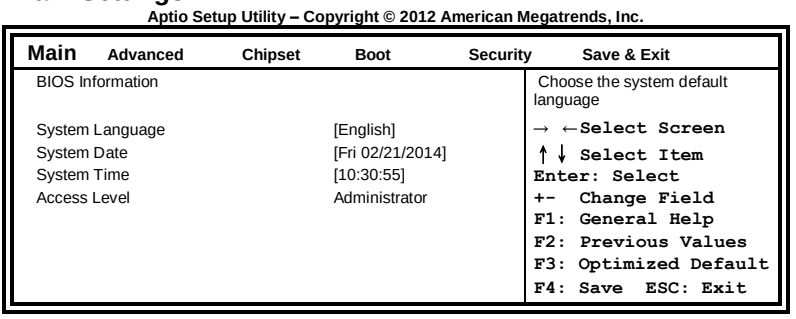
```
Press <DEL> / <F2> to Enter Setup
```

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.*

Main Settings



System Language

Choose the system default language.

System Date

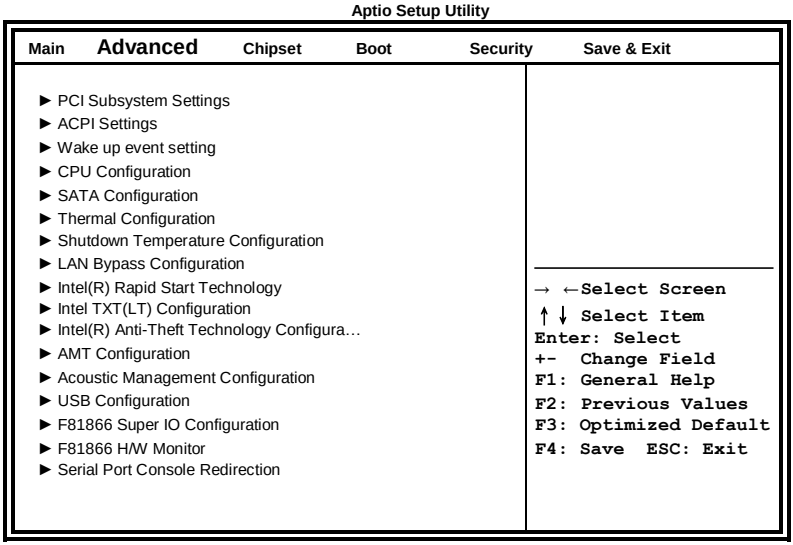
Set the Date. Use Tab to switch between Data elements.

System Time

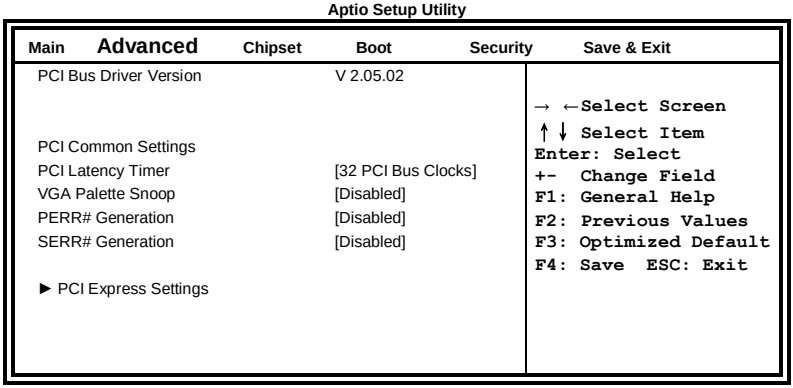
Set the Time. Use Tab to switch between Data elements.

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.



PCI Subsystem Settings



PCI Latency Timer

Value to be programmed into PCI Latency Timer Register.

VGA Palette Snoop

Enables or disables VGA Palette Registers Snooping.

PERR# Generation

Enables or disables PCI device to generate PERR#.

SERR# Generation
Enables or disables PCI device to generate SERR#.

PCI Express Settings
Change PCI Express devices settings.

PCI Express Settings

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
PCI Express Device Register Settings					
Relaxed Ordering			[Disabled]		
Extended Tag			[Disabled]		
No Snoop			[Enabled]		
Maximum Payload			[Auto]		→ ←Select Screen
Maximum Read Request			[Auto]		↑↓ Select Item
					Enter: Select
PCI Express Link Register Settings					+- Change Field
ASPM Support			[Disabled]		F1: General Help
WARNING: Enabling ASPM may cause some PCI-E devices to fail					F2: Previous Values
					F3: Optimized Default
Extended Synch			[Disabled]		F4: Save ESC: Exit
Link Training Retry			[5]		
Link Training Timeout			100		
Unpopulated Links			[Keep Link ON]		
Restore PCIE Register			[Disabled]		

Relaxed Ordering
Enables or disables PCI Express Device Relaxed Ordering.

Extended Tag
If ENABLED allows device to use 8-bit Tag field as a requester.

No Snoop
Enables or disables PCI Express Device No Snoop option.

Maximum Payload
Set Maximum Payload of PCI Express Device or allow System BIOS to select the value.

Maximum Read Request
Set Maximum Read Request Size of PCI Express Device or allow System BIOS to select the value.

ASPM Support

Set the ASPM Level: Force L0s – Force all links to L0s State: AUTO – BIOS auto configure : DISABLE – Disables ASPM.

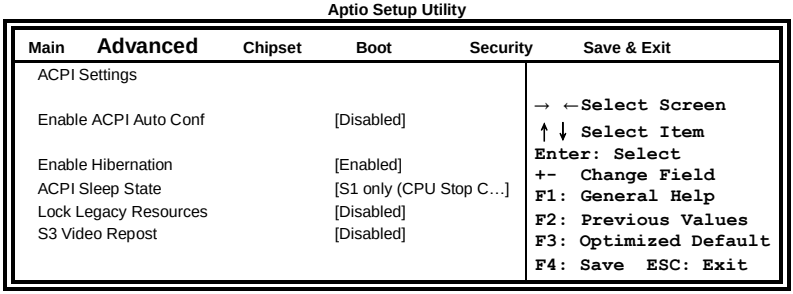
Extended Synch
If ENABLED allows generation of Extended Synchronization patterns.

Link Training Retry
Defines number of Retry Attempts software will take to retrain the link if previous training attempt was unsuccessful.

Link Training Timeout
Defines number of Microseconds software will wait before polling 'Link Training' bit in Link Status register. Value range from 10 to 1000 uS.

Unpopulated Links
In order to save power, software will disable unpopulated PCI Express links, if this option set to 'Disable Link'.

ACPI Settings



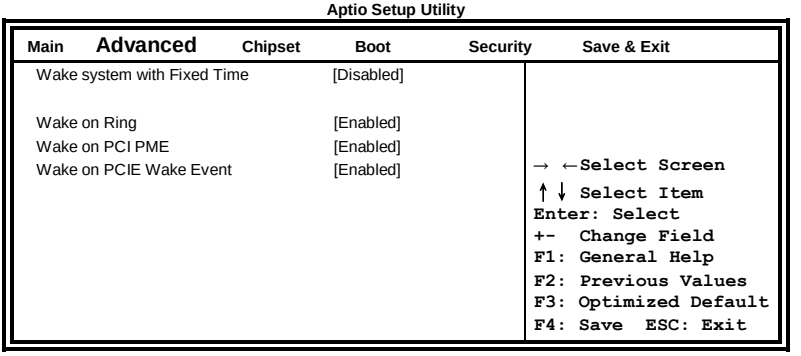
Enable Hibernation
Enables or Disables System ability to Hibernate (OS/S4 Sleep State). This option may be not effective with some OS.

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

Lock Legacy Resources
Enabled or Disabled Lock of Legacy Resources.

S3 Video Repost
Enable or disable S3 Video Repost.

Wake up event settings



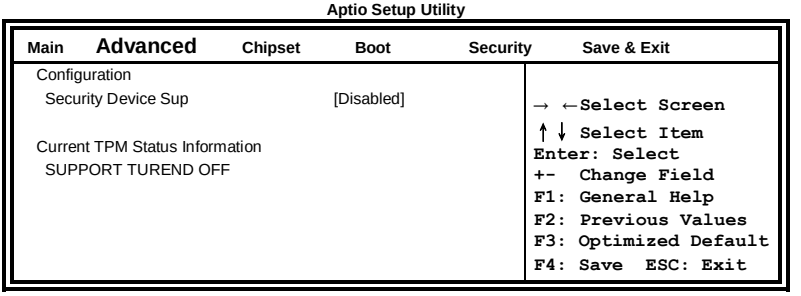
Wake system with Fixed Time

Enables or Disables System wake on alarm event. When enabled, System will wake on the hr::min:: sec specified.

Wake on PCIE PME Wake Event

The options are Disabled and Enabled.

Trusted Computing

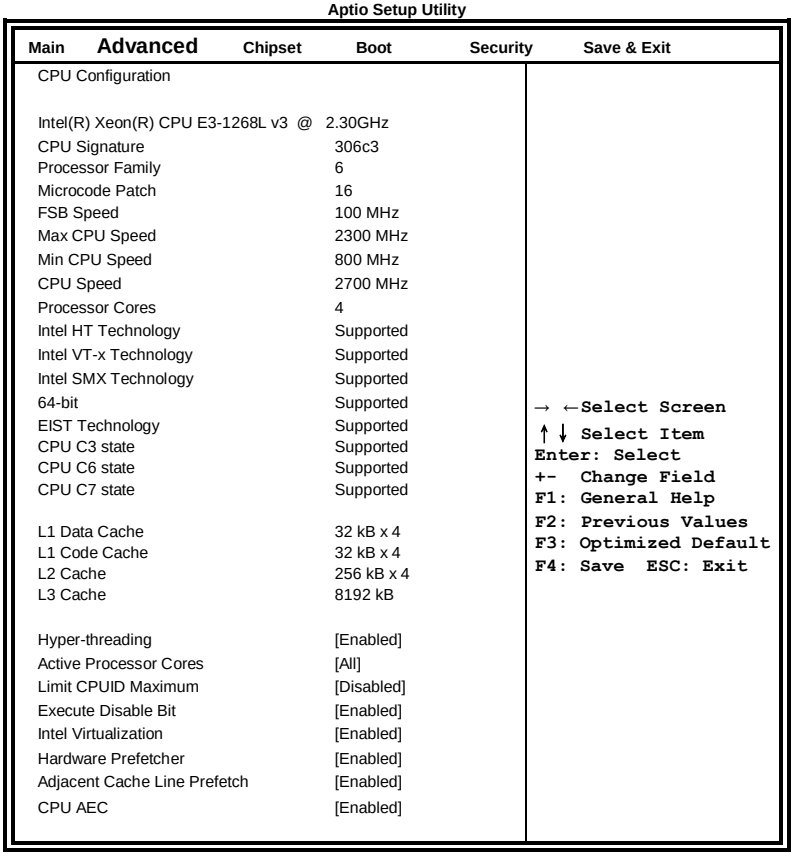


Security Device Support

Enables or disables BIOS support for security device. O.S. will not show Security Device. TCG EFI protocol and INT1A interface will not be available.

CPU Configuration

This section shows the CPU configuration parameters.



Hyper-threading

Enabled for Windows XP and Linux (OS optimized for Hyper-Threading Technology) and Disabled for other OS (OS not optimized for Hyper-Threading Technology). When Disabled, only one thread per enabled core is enabled.

Active Processor Cores

Number of cores to enable in each processor package.

Limit CPUID Maximum

Disabled for Windows XP.

Execute Disable Bit

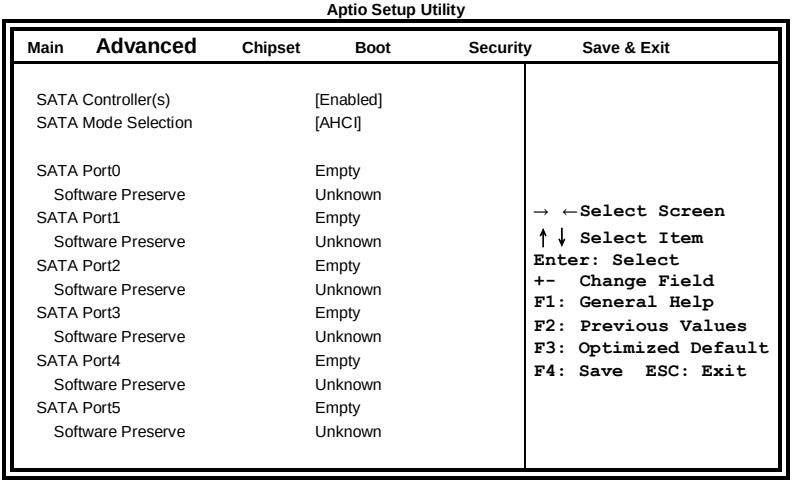
XD can prevent certain classes of malicious buffer overflow attacks when combined with a supporting OS (Windows Server 2003 SP1, Windows XP SP2, SuSE Linux 9.2, Re33dHat Enterprise 3 Update 3.)

Intel Virtualization Technology
When enabled, a VMM can utilize the additional hardware capabilities provided by Vanderpool Technology.

Hardware Prefetcher
To turn on/off the Mid level Cache (L2) streamer Prefetcher.

Adjacent Cache Line Prefetch
To turn on/off prefetching of adjacent cache lines.

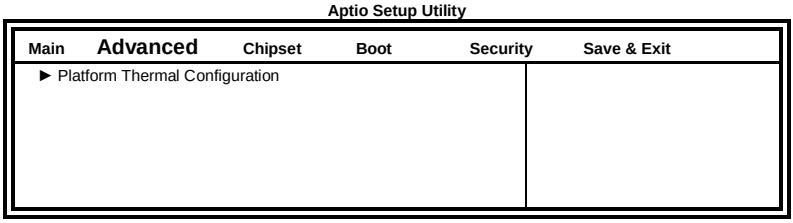
SATA Configuration
SATA Devices Configuration.



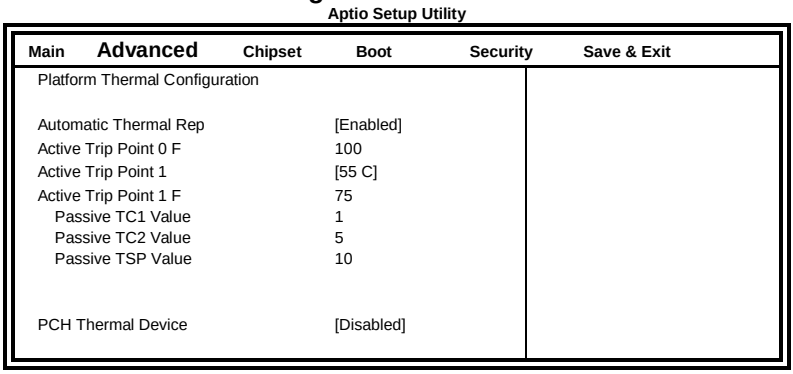
SATA Controller(s)
Enable / Disable Serial ATA Controller.

SATA Mode Selection
(1) IDE Mode.
(2) AHCI Mode.
(3) RAID Mode.

Thermal Configuration



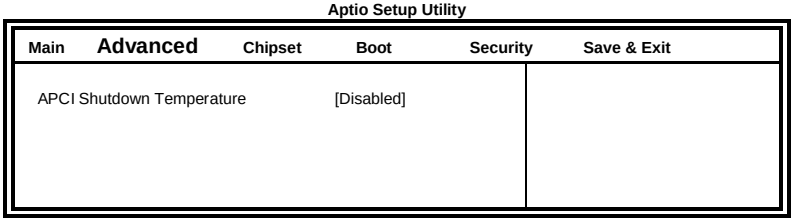
Platform Thermal Configuration



Automatic Thermal Reporting

Configure CRT, PSV and ACO automatically based on values recommended in BWG's thermal reporting for thermal management settings. Set to Disable for manual configuration.

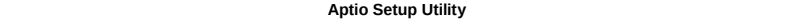
Shutdown Temperature Configuration



ACPI Shutdown Temperature

Set function Disabled or 70/75/80/85/90/95/100 ℃

LAN Bypass Configuration



Main	Advanced	Chipset	Boot	Security	Save & Exit
LAN Bypass Configuration					
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

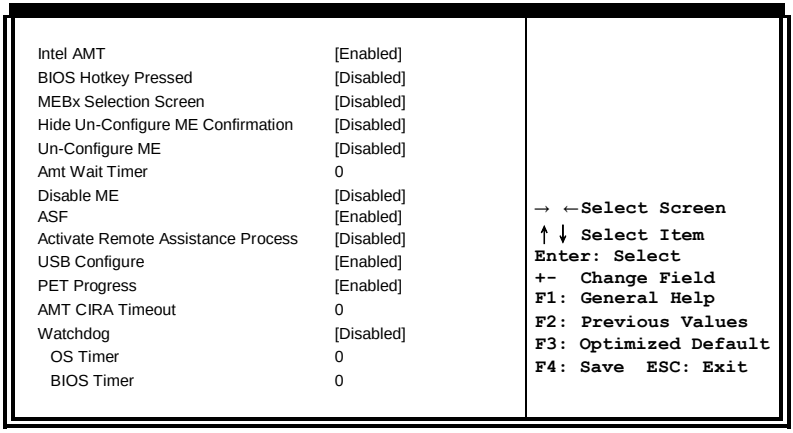
Main	Advanced	Chipset	Boot	Security	Save & Exit
LAN Bypass Configuration					
Bypass Quick Setting		[Custom Define]			
WDT Reset Signal		[Disabled]			
WDT Bypass Setting					
LAN5 LAN6 Bypass		[Normal]			
LAN7 LAN8 Bypass		[Normal]			
Ext LAN1 LAN2 Bypass		[Normal]			
Ext LAN3 LAN4 Bypass		[Normal]			
System OFF Bypass Setting					
LAN5 LAN6 Bypass		[Normal]			
LAN7 LAN8 Bypass		[Normal]			
Ext LAN1 LAN2 Bypass		[Normal]			
Ext LAN3 LAN4 Bypass		[Normal]			

Note: "Ext LAN Bypass" items only appear when extended IBASE LAN module card installed.

AMT Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit



AMT Configuration
Options are Enabled and Disabled.
Note: iAMT H/W is always enabled. This option just controls the BIOS extension execution. If enabled, this requires additional firmware in the SPI device.

Unconfigure ME
Perform AMT/ME unconfigure without password operation.

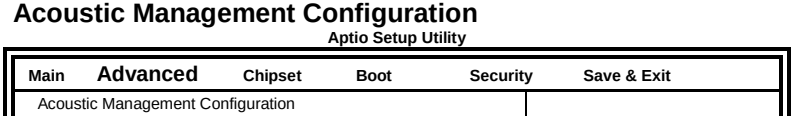
Amt Wait Timer
Set timer to wait before sending ASF_GET_BOOT_OPTIONS.

Activate Remote Assistance Process
Trigger CIRA boot.

PET Progress
User can Enable/Disable PET Events progress to receive PET events or not.

Watchdog Timer
Enable/Disable Watchdog Timer.

Acoustic Management Configuration



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

Smart fan function Enable or Disable.

USB Configuration

Aptio Setup Utility

Main **Advanced** Chipset Boot Security Save & Exit		
USB Configuration		
USB Devices: 1 Keyboard, 1 Mouse, 2 Hubs		
Legacy USB Support	[Enabled]	→ ←Select Screen ↑ ↓ Select Item Enter: Select +- Change Field F1: General Help F2: Previous Values F3: Optimized Default F4: Save ESC: Exit
USB3.0 Support	[Enabled]	
XHCI Hand-off	[Enabled]	
EHCI Hand-off	[Disabled]	
Port 60/64 Emulation	[Enabled]	
USB hardware delays and time-outs:		
USB Transfer time-out	[20 sec]	
Device reset tine-out	[20 sec]	
Device power-up delay	[Auto]	

Legacy USB Support

Enables Legacy USB support.
AUTO option disables legacy support if no USB devices are connected.
DISABLE option will keep USB devices available only for EFI applications.

USB3.0 Support

Enable/Disable USB3.0 (XHCI) Controller support.

XHCI Hand-off

This is a workaround for OSes without XHCI hand-off support. The XHCI ownership change should be claimed by XHCI driver.

EHCI Hand-off

Enabled/Disabled. This is a workaround for OSes without EHCI hand-off support. The EHCI ownership change should be claimed by EHCI driver.

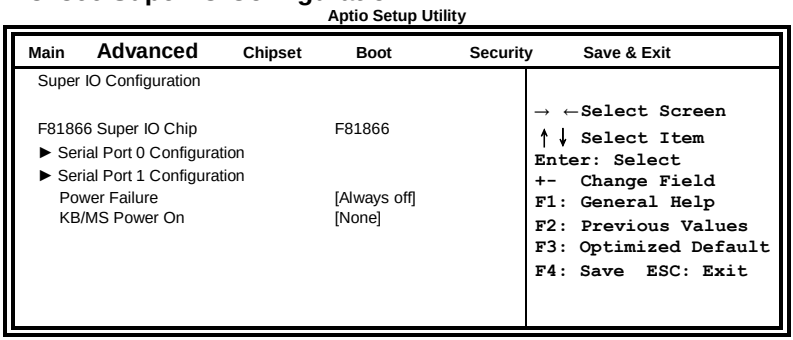
Port 64/60 Emulation
Enables I/O port 60h/64h emulation support. This should be enabled for the complete USB keyboard legacy support for non-USB aware OSes.

USB Transfer time-out
The time-out value for Control, Bulk, and Interrupt transfers.

Device reset tine-out
USB mass Storage device start Unit command time-out.

Device power-up delay
Maximum time the device will take before it properly reports itself to the Host Controller. 'Auto' uses default value: for a Root port it is 100ms, for a Hub port the delay is taken from Hub descriptor.

F81866 Super IO Configuration



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

F81866 H/W Monitor

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
PC Health Status					
Fan1 smart fan control		[50 C]			
Fan2 smart fan control		[50 C]			
Fan3 smart fan control		[Disabled]			
System temperature1		+41 C			
System temperature2		+38 C			
System temperature3		+37 C			
FAN1 Speed		1545 RPM			
FAN2 Speed		1550 RPM			
FAN3 Speed		1546 RPM			
VIN1		+1.776 V			
VIN2		+5.171 V			
VIN3		+12.408 V			
VSB5V		+5.016 V			
VCC3V		+3.392 V			
VSB3V		+3.392 V			
VBAT		+3.264 V			

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

Temperatures/Voltages

These fields are the parameters of the hardware monitoring function feature of the motherboard. The values are read-only values as monitored by the system and show the PC health status.

Fan1/Fan2/Fan3 Smart Fan Control

This field enables or disables the smart fan feature. At a certain temperature, the fan starts turning. Once the temperature drops to a certain level, it stops turning again.

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.

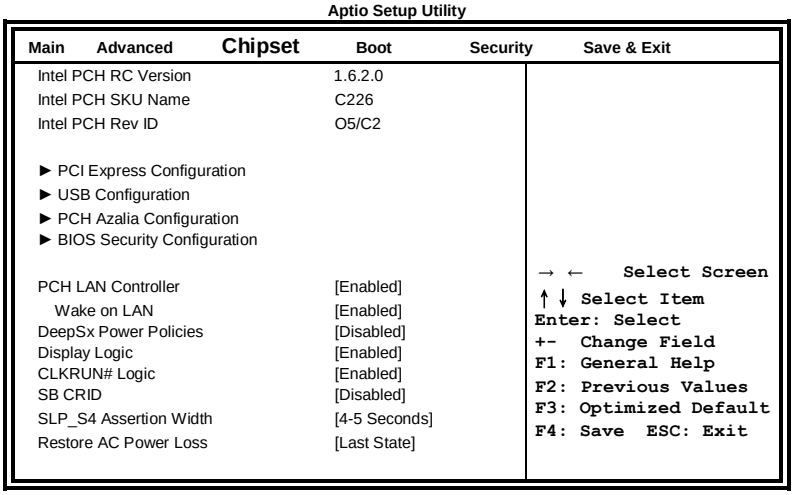
Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
► PCH-IO Configuration					
► System Agent (SA) Configuration					

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

PCH-IO Configuration

This section allows you to configure the North Bridge Chipset.



PCH LAN Controller

Enable or disable onboard NIC.

Wake on LAN

Enable or disable integrated LAN to wake the system. (The Wake On LAN cannot be disabled if ME is on at Sx state.)

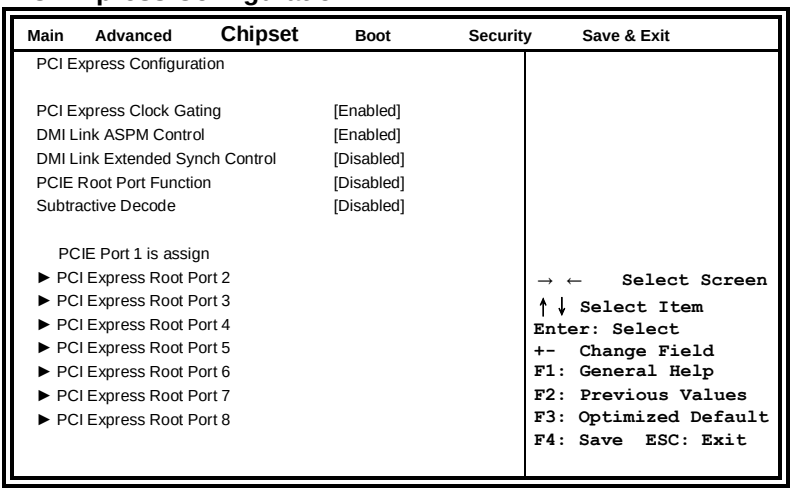
SLP_S4 Assertion Width

Select a minimum assertion width of the SLP_S4# signal.

Restore AC Power Loss

Select AC power state when power is re-applied after a power failure.

PCI Express Configuration



PCI Express Clock Gating

Enable or disable PCI Express Clock Gating for each root port.

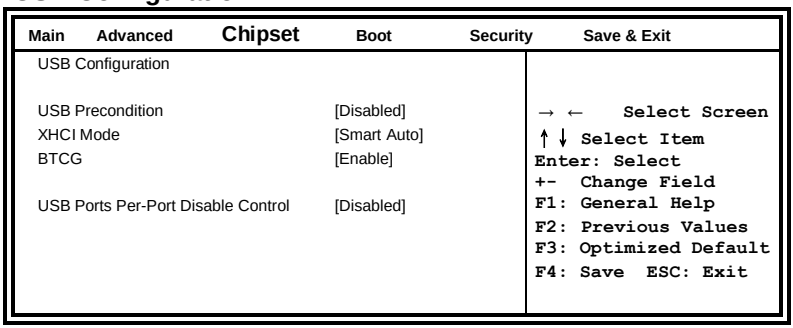
DMI Link ASPM Control

The control of Active State Power Management on both NB side and SB side of the DMI link.

PCIE Root Port Function

Enable or disable PCI express Root Port function swapping.

USB Configuration



USB Precondition

Precondition work on USB host controller and root ports for faster enumeration.

xHCI Mode

Mode of operation of xHCI controller

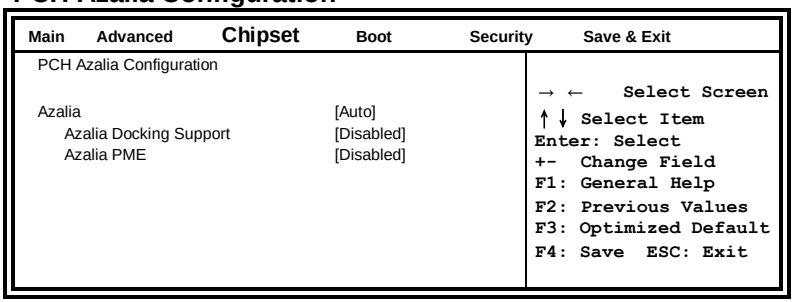
BTCG

Enable or disable trunk clock gating.

USB Ports Per-Port Disable Control

Control each of the USB ports (0~13) disabling.

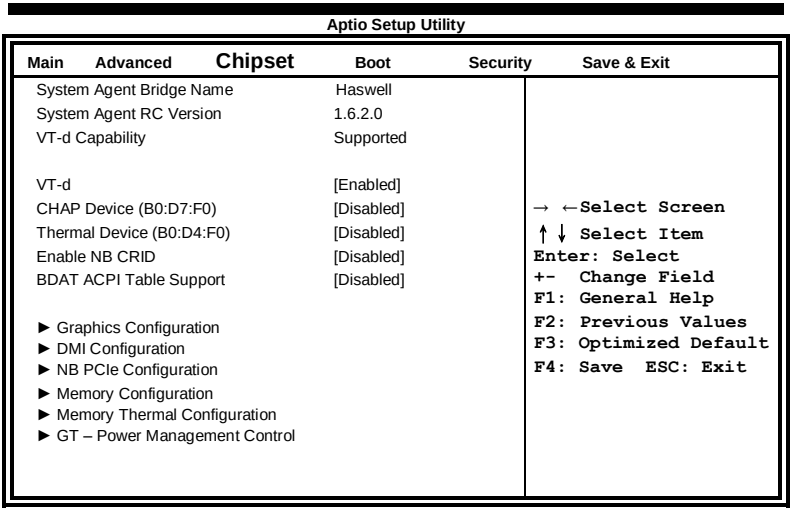
PCH Azalia Configuration



Azalia

Control Detection of the Azalia device.
Disabled = Azalia will unconditionally disabled.
Enabled Azalia will be unconditionally enabled.
Auto = Azalia will enabled if present, disabled otherwise.

System Agent (SA) Configuration



VT-d

Check to enable VT-d function on MCH.

Enable NB CRID

Enable or disable NB CRID WorkAround.

Graphics Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit
Graphics Configuration					
IGFX VBIOS Version			2164		
IGfx Frequency			700 MHz		
Graphics Turbo IMON			31		
Primary Display			[Auto]		→ ←Select Screen
Internal Graphics			[Auto]		↑↓ Select Item
GTT Size			[2MB]		Enter: Select
Aperture Size			[256MB]		+- Change Field
DVMT Pre-Allocated			[32M]		F1: General Help
DVMT Total Gfx Mode			[256M]		F2: Previous Values
Gfx Low Power Mode			[Enabled]		F3: Optimized Default
Graphics Performance			[Disabled]		F4: Save ESC: Exit
► LCD Control					

Primary Display

Select which of IGFX/PEG/PCI graphics device should be primary display or select SG for switchable Gfx.

Internal Graphics

Keep IGD enabled based on the setup options.

DVMT Pre-Allocated

Select DVMT 5.0 Pre-Allocated (Fixed) graphics memory size used by the internal graphics device.

DVMT Total Gfx Mem

Select DVMT 5.0 total graphics memory size used by the internal graphics device.

Gfx Low Power Mode

This option is applicable for SFF only.

Primary IGFX Boot Display (LCD Control)

Select the Video Device that will be activated during POST. This has no effect if external graphics present. Secondary booty display selection will appear based on your selection. VGA modes will be supported only on primary display.

Memory Configuration

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

Memory Information		
Memory Frequency	1600 MHz	
Total Memory	32768 MB (DDR3)	
DIMM#0	8192 MB (DDR3)	
DIMM#1	8192 MB (DDR3)	→ ←Select Screen
DIMM#2	8192 MB (DDR3)	↑↓ Select Item
DIMM#3	8192 MB (DDR3)	Enter: Select
CAS Latency (tCL)	11	+- Change Field
Minimum delay time		F1: General Help
CAS to RAS (tRCDmin)	11	F2: Previous Values
Row Precharge (tRPmin)	11	F3: Optimized Default
Active to Precharge (tRASmin)	28	F4: Save ESC: Exit
XMP Profile 1	Not Supported	
XMP Profile 2	Not Supported	

Boot Settings

Aptio Setup Utility

Main	Advanced	Chipset	Boot	Security	Save & Exit

Boot Configuration		
Setup Prompt Timeout	1	
Bootup NumLock State	[On]	
Quiet Boot	[Disabled]	
Fast Boot	[Disabled]	
Boot mode select	[LEGACY]	→ ←Select Screen
FIXED BOOT ORDER Priorities		↑↓ Select Item
Boot Option #1	[Hard Disk]	Enter: Select
Boot Option #2	[CD/DVD]	+- Change Field
Boot Option #3	[USB Hard Disk]	F1: General Help
Boot Option #4	[USB CD/DVD]	F2: Previous Values
Boot Option #5	[USB Key]	F3: Optimized Default
Boot Option #6	[USB Floppy]	F4: Save ESC: Exit
Boot Option #7	[Network]	
► CSM16 parameters		

Setup Prompt Timeout

Number of seconds to wait for setup activation key.
65535(0xFFFF) means indefinite waiting.

Bootup NumLock State

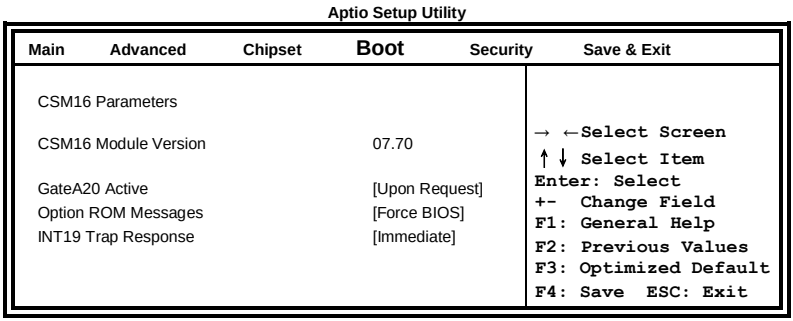
Select the keyboard NumLock state.

Quiet Boot
Enables/Disables Quiet Boot option.

Fast Boot
Enables/Disables boot with initialization of a minimal set of devices required to launch active boot option. Has no effect for BBS boot options.

Boot Option Priorities
Sets the system boot order.

CSM16 parameters
This section allows you to configure the boot settings.



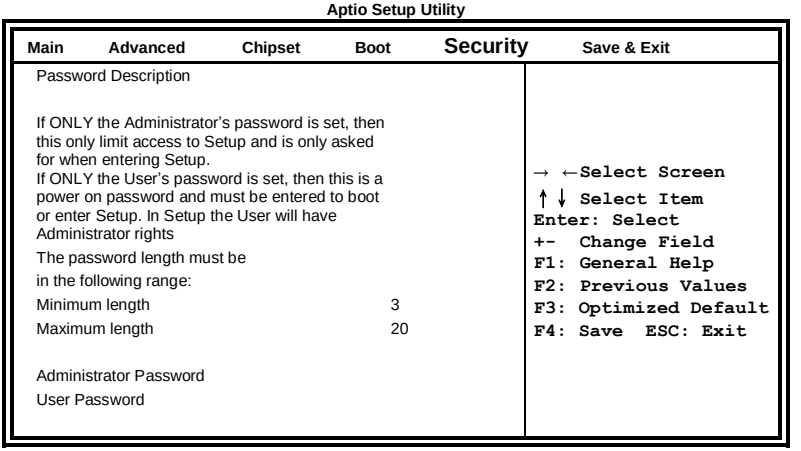
GateA20 Active
UPON REQUEST: GA20 can be disabled using BIOS services
ALWAYS: do not allow disabling GA20; this option is useful when any RT code is executed above 1MB.

Option ROM Messages
Set display mode for Option ROM

INT19 Trap Response
BIOS reaction on INT19 trapping by option ROM:
IMMEDIATE: execute the trap right away.
POSTPONED: execute the trap during legacy boot.

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.



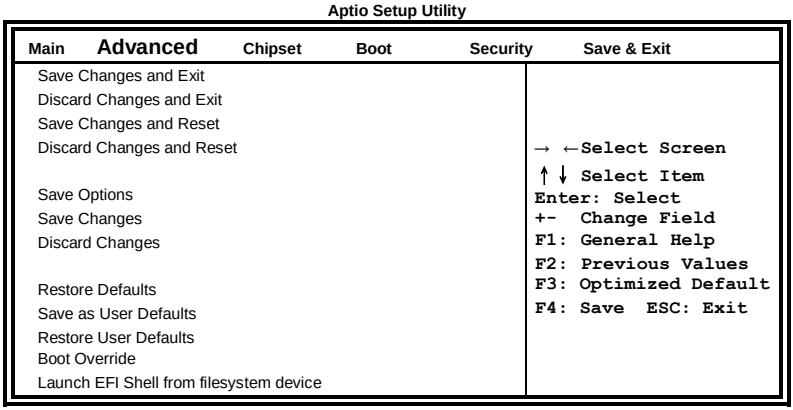
Administrator Password

Set Setup Administrator Password.

User Password

Set User Password.

Save & Exit Settings



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.

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 DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series 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.

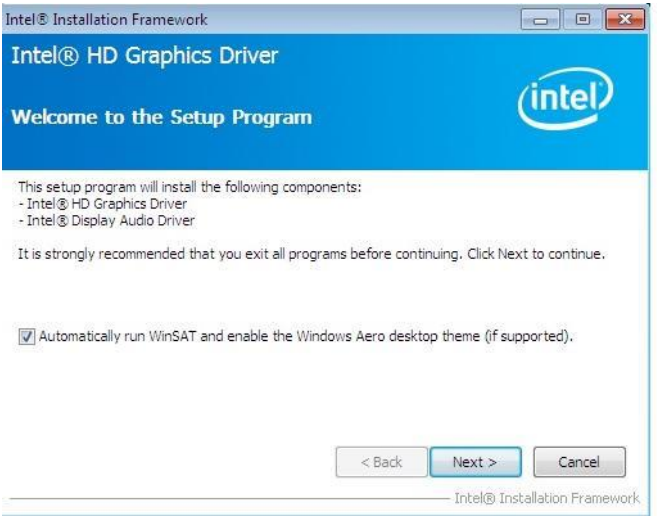
6. The Setup process is now complete. Click ***Finish*** to restart the computer and for changes to take effect.

VGA Drivers Installation

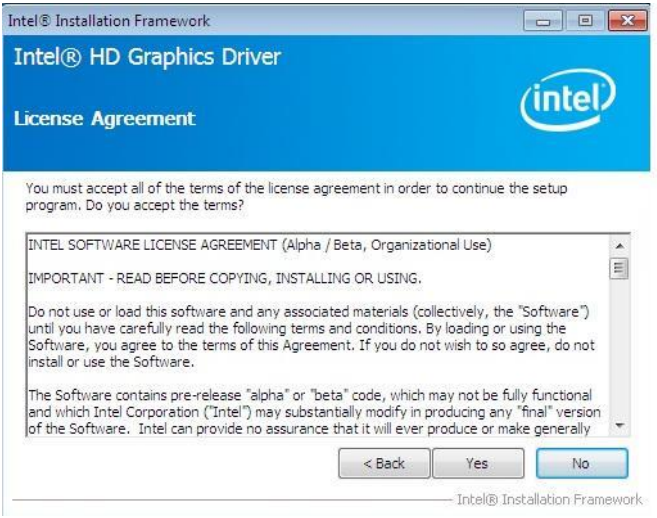
1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***.

2. Click ***Intel(R) Core(TM) i3/i5/i7 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 screen shown below, click ***Install*** to continue.

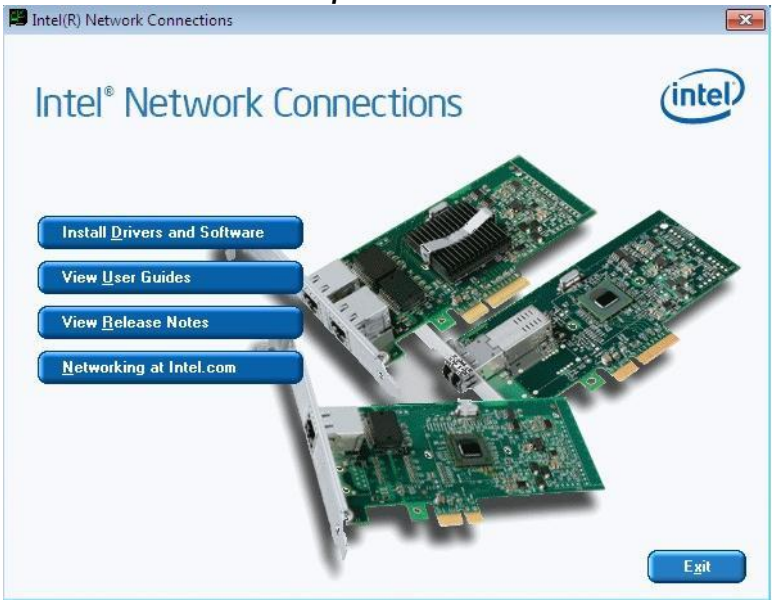
6. Setup complete. Click ***Finish*** to restart the computer and for changes to take effect.

LAN Drivers Installation

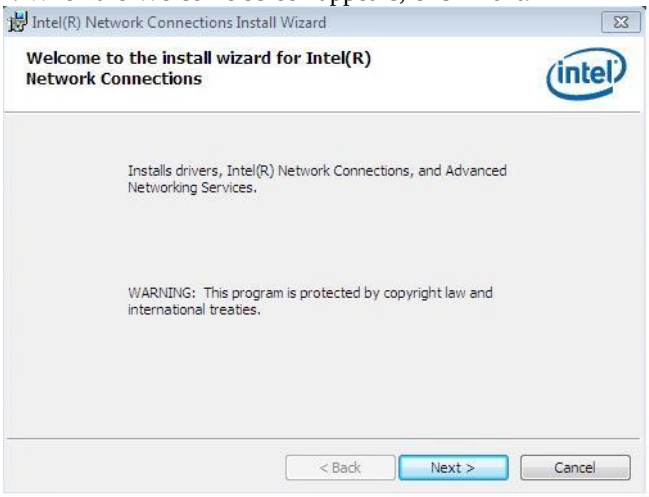
1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***.

2. Click ***Intel(R) PRO LAN Network Driver.***

3. Click ***Install Drivers and Software.***

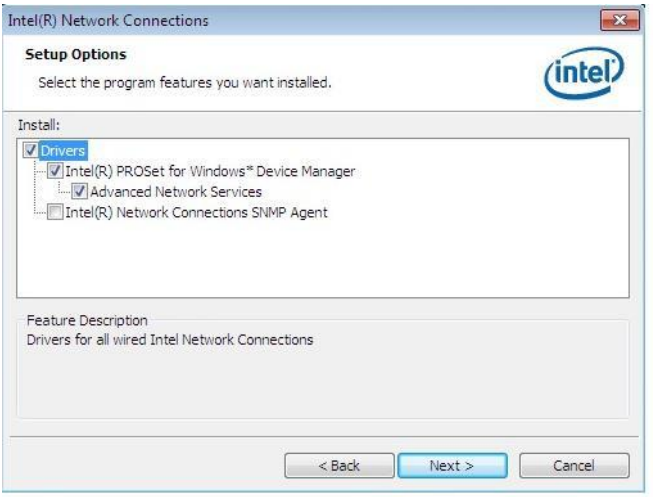


4. When the Welcome screen appears, click ***Next***.

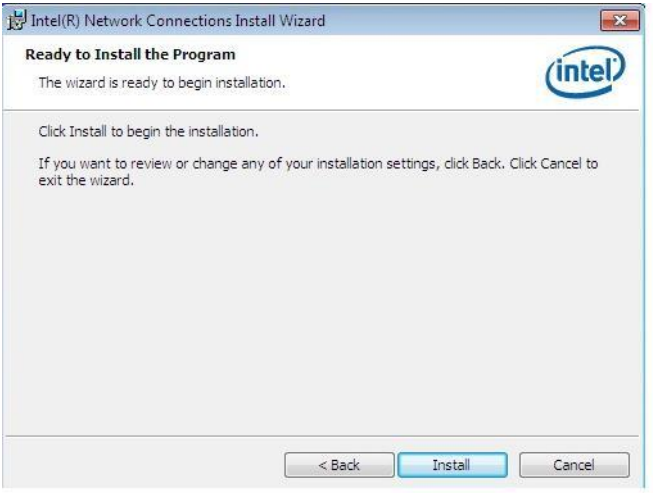


5. Click ***Next*** to to agree with the license agreement.

6. Click the checkbox for **Drivers** in the Setup Options screen to select it and click **Next** to continue.



7. The wizard is ready to begin installation. Click ***Install*** to begin the installation.



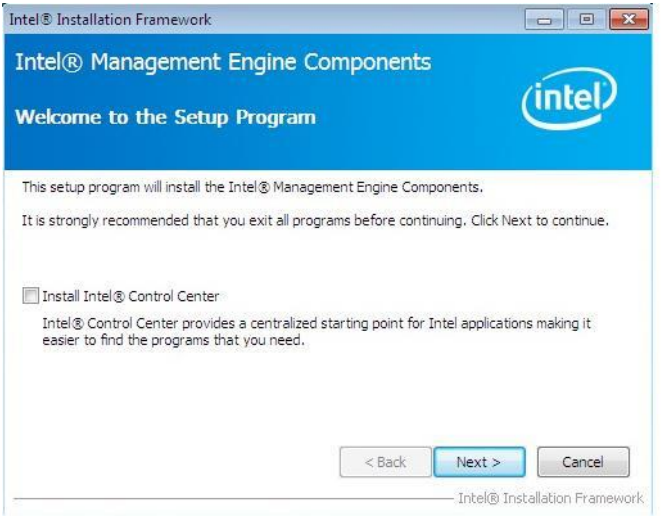
8. When InstallShield Wizard is complete, click ***Finish***.

Intel® Management Engine Interface

Follow the steps below to install the Intel Management Engine.

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers*** and then ***Intel(R) AMT 9.0 Drivers***.

2. When the Welcome screen to the InstallShield Wizard for Intel® Management Engine Components, click the checkbox for **Install Intel® Control Center** & click ***Next***.



3. Click ***Yes*** to to agree with the license agreement.

4. When the Setup Progress screen appears, click ***Next***. Then, click ***Finish*** when the setup progress has been successfully installed.

Intel® USB 3.0 Drivers

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) 8 Series Chipset Drivers***.

2. Click ***Intel(R) USB 3.0 Drivers.***

3. When the Welcome screen to the InstallShield Wizard for Intel® USB 3.0 eXtensible Host Controller Driver, click ***Next***.

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® USB 3.0 eXtensible Host Controller Driver.

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

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	System Timer
060h - 064h	Keyboard Controller
070h - 07Fh	Real Time Clock
080h - 09Fh	DMA Controller #2
0A0h - 0BFh	Interrupt Controller #2
0C0h - 0DFh	DMA Controller #3
0F0h	Clear Math Coprocessor Busy Signal
0F1h	Reset Math Coprocessor
3B0h- 3DFh	Graphics adapter Controller
3F8h - 3FFh	Serial Port #0 (COM1)
2F8h - 2FFh	Serial Port #1 (COM2)
3E8h – 3EFh	Serial Port #2 (COM3)
2E8h – 2EFh	Serial Port #3 (COM4)
3E8h - 3EFh	Serial Port #4 (COM5)
2E8h - 2EFh	Serial Port #5 (COM6)
360h - 36Fh	Network Ports

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
IRQ0	System Timer Output
IRQ1	Keyboard
IRQ4	Serial Port #1
IRQ3	Serial Port #2
IRQ5	Serial Port #3
IRQ11	Serial Port #4
IRQ8	Real Time Clock
IRQ14	Primary IDE
IRQ15	Secondary IDE

C. 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	Eth5, 6	Read / Write	Enable / Disable LAN Bypass function when the system is power On. 1 = Enable LAN Bypass function 0 = Disable LAN Bypass function
1	Eth7, 8		
2	Expansion LAN Card Eth1, 2		
3	Expansion LAN Card Eth3, 4		

CR 0x24 : System-Off Bypass Register

Attribute : Read / Write
Reset default : 0x0000

Bit	LAN Port#	Read / Write	Description
0	Eth5, 6	Read / Write	Enable / Disable LAN Bypass function when the system is power Off. 1 = Enable LAN Bypass function 0 = Disable LAN Bypass function
1	Eth7, 8		
2	Expansion LAN Card Eth1, 2		
3	Expansion LAN Card Eth3, 4		

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	Eth5, 6	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.
1	Eth7, 8		
2	Expansion LAN Card Eth1, 2		
3	Expansion LAN Card Eth3, 4		

CR 0x2A : Watchdog (WDT) Bypass Register

Attribute : Read / Write
Reset default : 0x0000

Bit	LAN Port#	Read / Write	Description
0	Eth5, 6	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
1	Eth7, 8		
2	Expansion LAN Card Eth1, 2		

3	Expansion LAN Card Eth3, 4		**CR 0x28**", it will follow below setting: 1 = Enable LAN Bypass function 0 = Disable LAN Bypass function

CR 0x2C : Card Plugged and Bypass Status Register

Attribute : Read

Reset default : N/A

Bit	LAN Port#	Read / Write	Description
0	Eth5, 6	Read	1 = The expansion slot is populated & support LAN Bypass function 0 = The expansion slot is empty or does not support LAN Bypass function
1	Eth7, 8		
2	Expansion LAN Card Eth1, 2		
3	Expansion LAN Card Eth3, 4		