
# ZFx86™

## Phoenix BIOS

## User's Supplement

## Version 1.06 Rev A

## November 2, 2001


P/N 9100-0066-03 Rev A

THIS DOCUMENT AND THE INFORMATION CONTAINED THEREIN IS PROVIDED "AS-IS" AND WITHOUT A WARRANTY OF ANY KIND. YOU, THE USER, ACCEPT FULL RESPONSIBILITY FOR PROPER USE OF THE MATERIAL. ZF MICRO DEVICES, INC. MAKES NO REPRESENTATIONS OR WARRANTIES THAT THIS DATA BOOK OR THE INFORMATION CONTAINED THERE-IN IS ERROR FREE OR THAT THE USE THEREOF WILL NOT INFRINGE ANY PATENTS, COPYRIGHT OR TRADEMARKS OF THIRD PARTIES. ZF MICRO DEVICES, INC. EXPLICITLY ASSUMES NO LIABILITY FOR ANY DAMAGES WHATSOEVER RELATING TO ITS USE.

**LIFE SUPPORT POLICY**

**ZF MICRO DEVICES** PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZF MICRO DEVICES, INC.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



ZFx86, FailSafe FailSafe Boot ROM, Z-tag ZF-Logic, InternetSafe, OEMmodule SCC, ZF SystemCard, ZF FlashDisk-SC, netDisplay, ZF 104Card, ZF SlotCard, and ZF Micro Devices logo are trademarks of ZF Micro Devices, Inc. Other brands and product names are trademarks of their respective owners.

ZF Micro Devices * 1052 Elwell Court, Palo Alto, CA 94303 * Tel: 650-965-3800 * Fax 650-965-4050

# 1. Introduction to the ZFx86 BIOS

The ZFx86 BIOS is the Phoenix 4.0 Revision 6 BIOS customized for the ZFx86™, and is licensed for use with the ZF Micro Devices, Inc. ZFx86 System-On-a-Chip.

This manual is a supplement to the "PhoenixBIOS™ 4.0 Revision 6 User's Manual" dated June 22, 2000 and is included with the ZFx86 BIOS Release Set version 1.06. It covers ZFx86 specific configuration settings and utilities used to manage the ZFx86 BIOS.

Certain Hypertext Links in this document take you either to the web, or to other ZF Micro Devices documents. For the document links to work, the PDF version of this document should be in the same directory as all the other .pdf files. On the ZFx86 Integrated Development System CD, all the PDF documents are in the subdirectory named \Documents.

See the PhoenixBIOS™ 4.0 Rev6 User Manual.PDF.

# 2. Features

The BIOS version 1.06 supports a load feature for custom PCI video BIOS binaries. Using the ZEB utility's main menu, you may combine the ZFx86 BIOS with your own PCI video BIOS. See *'Loading A PCI Video BIOS'* on page 25. The BIOS shadows the PCI video BIOS and treats it as if it were a standard PCI Extension ROM and initializes the matching embedded PCI Video chip.

NOTE: Insure that your PCI VIdeo BIOS uses a valid license.

In addition to the standard features documented in the PhoenixBIOS™ User's Manual, the ZFx86 BIOS includes these extended features important for embedded applications:

- ZFlash OS Loader Hook – enables operating systems such as Linux and VxWorks to boot from the same flash chip that contains the BIOS.
- ZFlash legacy ISA extension processor – allows user extension ROMs to be placed in the same flash device as BIOS
- Configuration settings that manage ZFx86 ZF Logic Memory and I/O Chip Selects for Disk On Chip, flash based extensions and custom I/O hardware
- Advanced Power Management 1.2 Functions
- Universal Serial BUS Host Controller and Legacy Configuration Settings
- Infrared support
- Watchdog Timer Function
- Remote Management from PC Host
- Resident Flash Disk Function

### 2.1. ZFx86 BIOS Set Contents

The zfx10600.ZIP compressed file contains the components of the ZFx86 BIOS Set software release version 1.06 (part number 9270-0012-01060). The ZFx86 BIOS Set contains the items in the following list:

- PhoenixBIOS 4.0 revision 6 User's Manual
- ZFx86 BIOS User's Manual Supplement (P/N 9100-0066-02)
- Routing ZFx86 Interrupts Application Note (P/N 9150-0015-00B)
- readme.txt – release notes text file
- readme.pdf – release notes in PDF format
- zfx10600.rom – Binary image file with ZF Micro Devices, Inc. splash screen[1] (P/N 9272-0100-010600)
- zfx10600.ron – Binary image file – with no splash screen (P/N 9272-0101-010600)
- zfx10600rom.bin – Z-tag Manager binary image file with splash screen (P/N 9272-0103-010600)
- testlogo.bmp – generic 640x480x256 color splash screen bit map graphic (P/N 9272-0112-00)
- amdflash.exe utility (P/N 9272-0106-01)
- zeb.exe – ZF Edit BIOS utility (P/N 9272-0111-000600)

## 3. Installation

The ZFx86 BIOS binary image files are 256 Kbytes in length. Any of these files may be placed in an ROM, EPROM or Flash Device that is chip selectable by the processor's reset vector, (0FFFFFFF0h) and chip select 0. Although the image is 256 Kbytes in this space, during initialization certain blocks are decompressed and/or discarded. The result is that a 128 Kbyte image loads into the shadow area of system memory at 0E0000h to 0FFFFFh. The BIOS image may be loaded into the Integrated Development System (IDS) AMD Flash using either the Z-tag Dongle or from a DOS prompt using the amdflash.exe Utility.

Optionally, external Programmers may be used to transfer the BIOS image into EPROM or Flash devices.

Find detailed instructions for installing user software in flash devices in "*Booting User Software From Flash Chips*" Software Note, Part # 9100-0067-00. Download this document from the ZF Micro Devices website: *http://www.zfmicro.com*

### 3.1. Using The amdflash.exe

AMDFLASH is a convenient way to update the BIOS in the Integrated Development System or another Target Board which has a 2 MB AMDFLASH chip installed. AMDFLASH supports only AMD flash chips, and requires that you have a working BIOS already installed (and thus can boot DOS).

Amdflash.exe always loads the BIOS image from a file called BIOS.ROM residing on a floppy or a hard disk. In the installation example below, we copy the BIOS files into a directory located on the C: Drive named AMDFLASH, and the files are being loaded from an A: floppy drive.

1. Start up screen containing the ZF Micro, Inc. logo.

To load the ZFx86 BIOS follow this procedure:

1. Use the COPY /B command to transfer the desired BIOS image to the BIOS.ROM file. This file must be in the same directory as amdflash.exe. For example, type:

   **C:\AMDFLASH> COPY /B A:zfx10600.rom BIOS.ROM**

   Always use the /B binary argument when you COPY from a DOS prompt.

2. Set the jumpers so that the ZFx86 boots from the AMDFLASH (the board may be set already – newer systems arrive as such). Jumper pins 4 and 6 of the FLASH CS CONF jumper JP7, and set BOOTSTRAPS switch S3 #12 to OFF. See Figure 1.

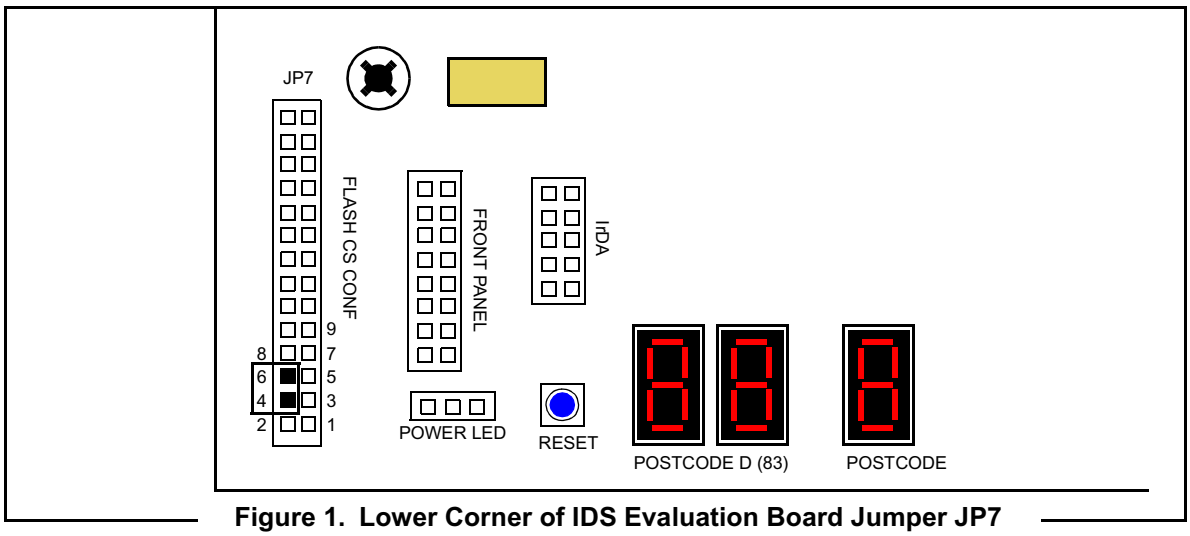



**Figure 1. Lower Corner of IDS Evaluation Board Jumper JP7**

Since the AMDFLASH utility runs from a DOS prompt, make sure you are able to boot using any working BIOS, or the GS from the ATMEL CHIP (if present), or a previous version of PhoenixBIOS from the AMD Flash.

Note that if you boot DOS from the GS/ATMEL, you need to change the FLASH CS CONF jumper JP7 from pins 3 and 5 to pins 4 and 6 (see Figure 1), and place S3 switch #12 to the right before executing the AMDFLASH utility.

3. Boot DOS. You may boot DOS from the floppy disk or the hard disk.

4. Copy the amdflash.exe program (contained in the ZIP file) to the DOS disk. For example, type:

   **C:\AMDFLASH> COPY /B A:amdflash.exe**

5. Copy the .BIN files into the same directory. For example, type:

   **C:\AMDFLASH> COPY /B A:zfx10600rom.bin**

6. Run "AMDFLASH 0".

   Status messages display as the system boots.[2] Do not change any jumpers unless you boot from the ATMEL Flash (see step 2 above). This version of the program places a copy of the BIOS.ROM in the IDS AMD flash.

2. If you have trouble running AMDFLASH 0 program, we recommend removing any ISA slot boards (except video). Currently, AMDFLASH is "hardwired" for a 2 MB Flash Chip.

### 3.2. Using the Dongle

Use the Z-tag Manager software application to load the BIOS image into the Z-tag Dongle. Then use the Dongle to load the BIOS on systems that support the Z-tag interface such as the IDS. See the examples in Chapter 4 of the *ZFx86 Integrated Development System Quick Start Guide*.[3]

The Dongle must contain two SEEPROMs. Dongles manufactured after January, 2001 contain two SEEPROMs. The Z-tag Manager provides an error message if the Dongle does not contain enough SEEPROM memory.

## 4. BIOS Setup

To start the BIOS Setup utility, press the [F2] function key during the boot up process. The PhoenixBIOS Setup Utility screen displays with the following selections across the top:

- Main
- Advanced
- Security
- Power
- Boot
- Exit

### 4.1. Main

The Main menu selections contain no changes with this release; therefore, that information is not duplicated in this supplement. For information specific to the Main menu, see the PhoenixBIOS™ 4.0 Rev6 User Manual.PDF.

### 4.2. Advanced

Figure 2 shows the Advanced menu selections.

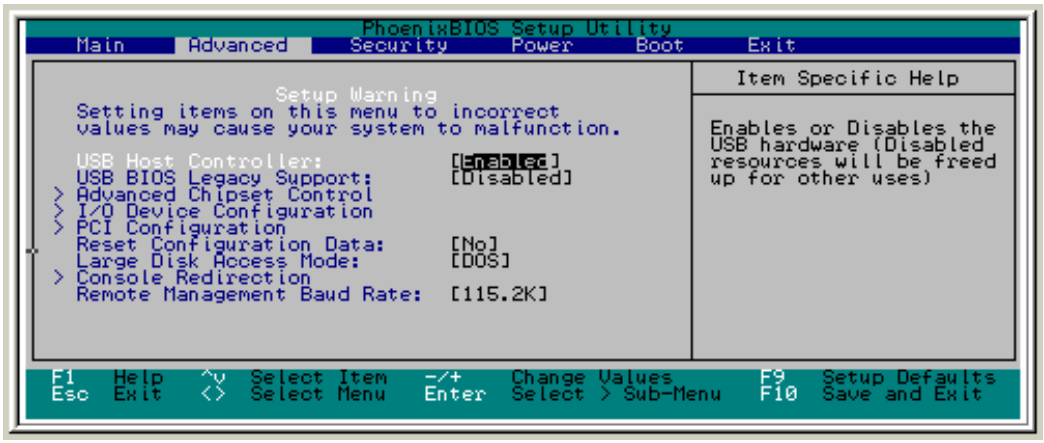



**Figure 2. Advanced Setup Screen**

3. The Dongle's flash programmer allows you to specify where in the flash to place the BIOS. Place the BIOS in the high addresses of the flash. Since the ZFx86 Phoenix BIOS is a 256K image, use starting address 1C0000 if you have a 2MB Flash, and C0000 if you have a 1 MB flash, and so on.

**Table 1: Advanced Setup Screen**

| Feature | Options | Description |
|---|---|---|
| USB Host controller | Disabled<br>Enabled | Enables or Disables the USB hardware. (Disabled resources are available for other uses.) Default setting is **Enabled**. |
| USB BIOS Legacy Support | Disabled<br>Enabled | Enables or Disables support for USB Keyboard and Mouse. (Enable for use with a non-USB aware Operating System such as DOS or UNIX.) Default setting is **Disabled**. |
| Advanced Chipset Control | | See *'Advanced Chipset Control'* on page 9. |
| I/O Device Configuration | | See *'I/O Device Configuration'* on page 12. |
| PCI Configuration | | See *'PCI Configuration'* on page 14. |
| Reset Configuration Data | No<br>Yes | Select 'Yes' if you want to clear the Extended System Configuration Data (ECSD) area. Default setting is **No**. |
| Secured Setup Configurations | No<br>Yes | Yes – Prevents a Plug and Play Operating System from changing system settings. Default setting is **No**. |
| Installed O/S | Other<br>Win95 | Select the operating system installed on your system that you will use most commonly. Default setting is **Other**.<br>**Note:** An incorrect setting causes some operating systems to display unexpected behavior. |
| Large Disk Access Mode | Other<br>DOS | For UNIX, Novel NetWare, or other operating systems, select **Other**.<br>If you are installing new software and the drive fails, change this selection and try again. Different operating systems require different representations of drive geometries. Default setting is **DOS**. |
| Console Redirection | | See *'Console Redirection'* on page 18 |
| Remote Management Baud Rate | 115.2K<br>57.6K<br>38.4K<br>28.8K<br>19.2K<br>14.4K<br>9600<br>2400 | Selects the baud rate used for serial remote configuration mode. Default setting is **115.2K**. |

## 4.3. Advanced Chipset Control

Figure 3 shows the Advanced Chipset Control submenu selections.

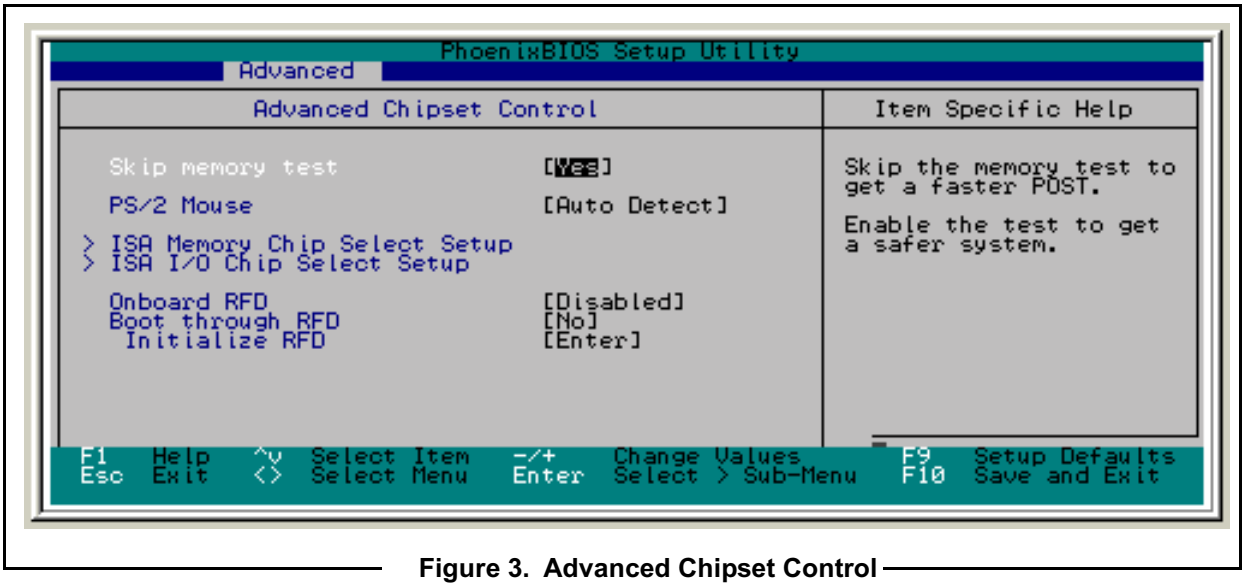



**Figure 3. Advanced Chipset Control**

**Table 2: Advanced Chipset Control**

| Feature | Options | Description |
|---|---|---|
| Skip memory test | No<br>Yes | Skip the memory test to get a faster POST. Enable the test to create a more stable system. |
| PS/2 Mouse | Disabled<br>Enabled<br>Auto Detect | Disabled – Prevents any installed PS/2 mouse from functioning, but frees up IRQ12.<br>Enabled – Forces the PS/2 mouse port to be enabled regardless if a mouse is present.<br>Auto Detect – Only enables the PS/2 mouse if present. Default setting is **Auto Detect**.<br>OS Controlled – Option displays only if the OS controls the mouse. |
| ISA Memory Chip Select Setup | | See *'ISA Memory Chip Select Setup'* on page 10. |
| ISA I/O Chip Select Setup | | See *'ISA I/O Chip Select Setup'* on page 11. |
| Onboard RFD | Disabled<br>To mem_cs1<br>To mem_cs2<br>To mem_cs3 | Selects whether the onboard flash disk is enabled. Default setting is **To mem_cs3**. |
| Boot Through RFD | No<br>Yes | Select Yes to boot from the RFD as floppy A:. Default setting is **No.** |

### 4.3.1. ISA Memory Chip Select Setup

The Memory Chip Selects provide initial values for the four memory windows which may be created using the ZF-logic built in the ZFx86 chip. For more information, see *'Understanding Memory Windows'* on page 26.

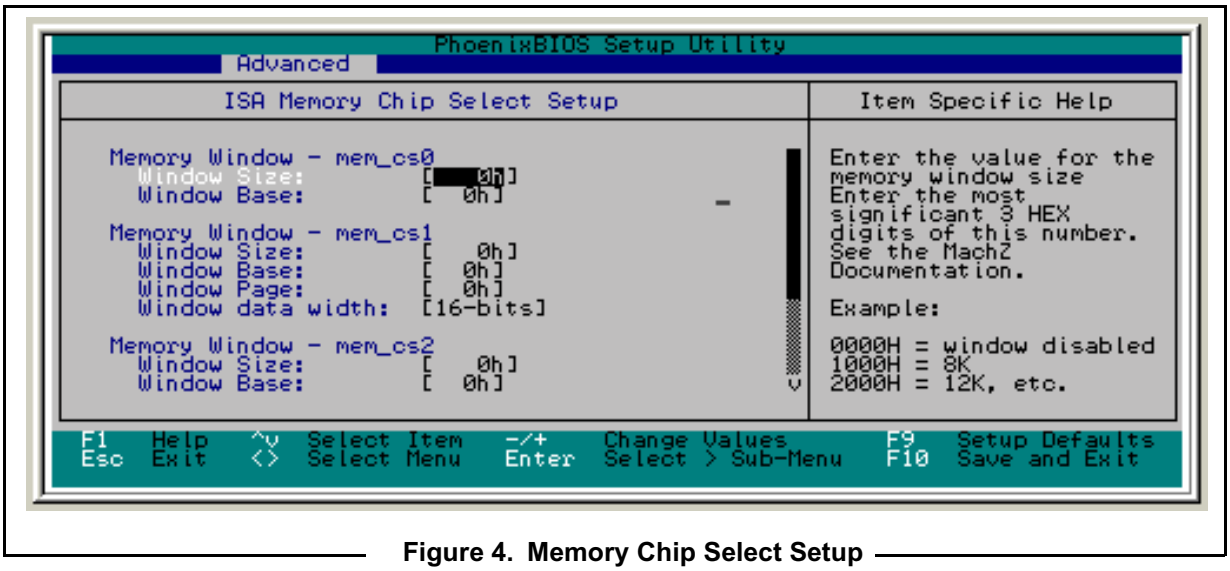



**Figure 4. Memory Chip Select Setup**

**Table 3: ISA Memory Chip Select Setup**

| Feature | Options | Description |
|---|---|---|
| Window SIZE | | Enter the value for the memory window size register. Acceptable range is from 000000H to FFF000H. Enter the hexadecimal value of the most significant 3 HEX digits of this number. See the ZFx86 documentation for additional information.<br>Example: 0000H = window = disabled, 1000H = 8K, 2000 = 12K, and so on. |
| Window BASE | | Enter the value for the memory window base register. Acceptable range is from 000000H to FFF000H. Enter the hexadecimal value of the most significant 3 HEX digits of this number.See the ZFx86 documentation for additional information. Example: 0C0000 = base address is 000C0000 (or C000:0) |
| Window PAGE | | PAGE = 1000000 – BASE + FLASHA. If Base = 0D0000, then set PAGE to F30000 so that D000:0 goes to address 0 in the flash. That is 1000000 – D0000 + 0. For D0000 to go to D0000 in the flash, set PAGE to 0, that is, 1000000 - D0000 + D0000 (you only specify 4 digits). |
| Read/Write Control | Read/Write<br>Read Only | Select the behavior of the memory range between read/write or read only access type. Default setting is **Read/Write**. |
| Window data width | 16-bits<br>8-bits | Select the window datapath width. Default setting is **8-bits**. |

### 4.3.2. ISA I/O Chip Select Setup

Figure 5 shows the ISA I/O Chip Select Setup menu selections.

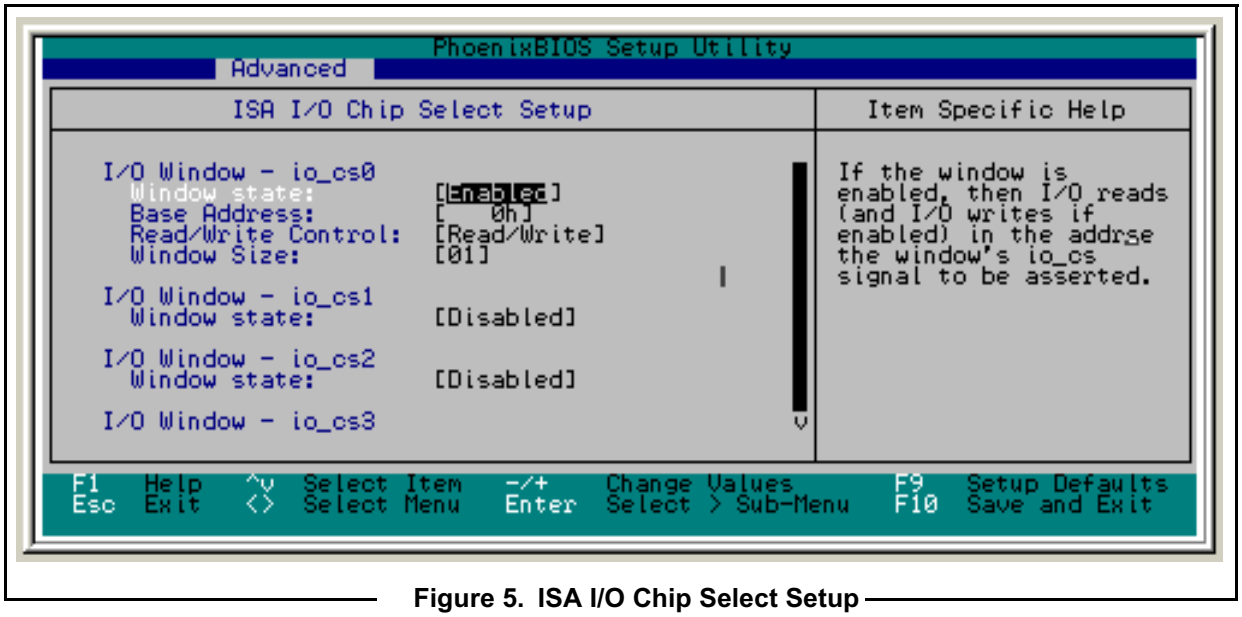



**Figure 5. ISA I/O Chip Select Setup**

**Table 4: ISA I/O Chip Select Setup**

| Feature | Options | Description |
|---|---|---|
| Window state | Enabled<br>Disabled | If the window is enabled, then I/O reads (and I/O writes if enabled) in the address range of Base to Base+Size –1 causes the window's io_cs signal to be asserted. Default setting is **Disabled**. |
| Base address | 0 - FFFF | Enter the base address for the I/O window. Default setting is **0**. |
| Read/Write Control | Read/Write<br>Read Only | Select the behavior of the I/O range between read/write or read only access type ports. Setting window to read only mode disables the IOW_N signal on ISA bus for I/O window address range.<br>Default setting is **Read/Write**. |
| Window data width | 8-bits<br>16-bits | Select the window datapath width. Default setting is **8-bits**. |
| Active level | Active Low<br>Active High | Select the level to assert on the io_cs pin. The ZFx86 asserts the selected level on the window's io_cs pin when the program accesses I/O in the window range of Base to Base+Size-1. Default setting is **Active Low**. |
| Window Size | 01 - 16 | Select the number of consecutive I/O address to decode starting from the I/O window base. Example, a value of 4 decodes 4 consecutive 8-bit I/O addresses, or 2 consecutive 16-bit addresses. Decode occurs on I/O read, and may occur on I/O write. Default setting is **01**. |

### 4.3.3. I/O Device Configuration

Figure 6 shows the I/O Device Configuration menu selections.

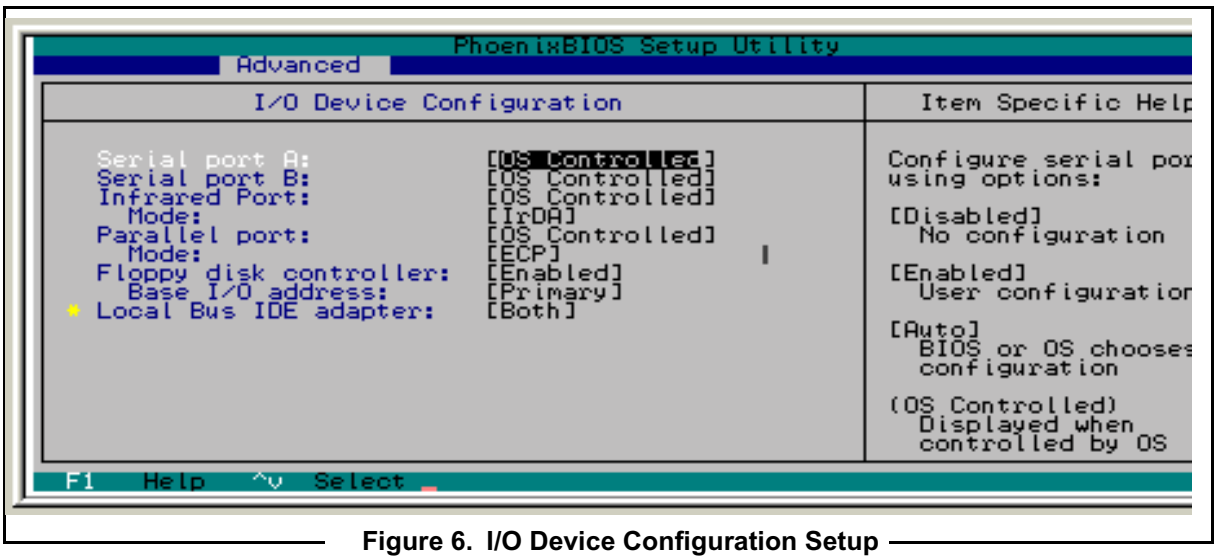



**Figure 6. I/O Device Configuration Setup**

**Table 5: I/O Device Configuration Setup**

| Feature | Options | Description |
|---|---|---|
| Serial Port A | Disabled<br>Enabled<br>Auto<br>OS Controlled | Configure Serial Port A using the following options:<br>Disabled – No configuration<br>Enabled – User Configuration<br>Auto – BIOS or OS chooses configuration<br>OS Controlled – Displays when port controlled by OS. Default setting. |
| Base I/O address | 3F8<br>2F8<br>3E8<br>2E8 | Set the base I/O address for Serial Port A. Default setting is **3F8.** |
| Interrupt | IRQ 3<br>IRQ 4 | Set the interrupt for serial port A. Default setting is **IRQ 4.** |
| Serial Port B | Disabled<br>Enabled<br>Auto<br>OS Controlled | Configure serial port B using the following options:<br>Disabled – No configuration<br>Enabled – User Configuration<br>Auto – BIOS or OS chooses configuration<br>OS Controlled – Displayed when controlled by OS. Default setting. |
| Base I/O address | 3F8<br>2F8<br>3E8<br>2E8 | Set the base I/O address for Serial Port B. Default setting is **2F8.** |
| Interrupt | IRQ 3<br>IRQ 4 | Set the interrupt for Serial Port B. Default setting is **IRQ 3.** |
| Infrared Port | Disabled<br>Enabled<br>Auto<br>OS Controlled | Configure Infrared port using the following options:<br>Disabled – No configuration<br>Enabled – User Configuration<br>Auto – BIOS or OS chooses configuration<br>OS Controlled – Displays when controlled by OS. Default setting. |
| Mode | IrDA<br>FIR | Set mode for Infrared port. Default setting is **IrDA.** |

**Table 5: I/O Device Configuration Setup (Continued)**

| Feature | Options | Description |
|---|---|---|
| Base I/O address | 3F8<br>3E8 | Select the base I/O address for Infrared port. Default setting is **3E8.** |
| Interrupt | IRQ 3<br>IRQ 5 | Select the interrupt for the Infrared port. Default setting is **IRQ 5.** |
| Parallel port | Disabled<br>Enabled<br>Auto<br>OS Controlled | Configure parallel port using options:<br>Disabled – No configuration<br>Enabled – User Configuration<br>Auto – BIOS or OS selects the configuration<br>OS Controlled – Displayed when controlled by OS. Default setting. |
| Mode | Output only<br>Bi-directional<br>EPP<br>ECP | Set the mode for the parallel port using the following options:<br>Output only, Bi-directional, EPP, and ECP. Default setting is **ECP.** |
| Base I/O address | 378<br>278<br>3BC | Set the base I/O address for the parallel port. Default setting is **378.** |
| Interrupt | IRQ 5<br>IRQ 7 | Set the interrupt for the parallel port. Default setting is **IRQ 7.** |
| DMA channel | DMA 1<br>DMA 3 | Set the DMA channel for the parallel port. Default setting is **DMA 1.** |
| Floppy disk controller | Disabled<br>Enabled<br>Auto | Configure using the following options:<br>Disabled – No configuration<br>Enabled – User Configuration. Default setting.<br>Auto – BIOS or OS chooses configuration<br>OS Controlled – Displayed when controlled by OS. |
| Base I/O address | Primary<br>Secondary | Set the base I/O address for the floppy disk controller using options.<br>Default setting is **Primary.** |
| Local Bus IDE adapter | Disabled<br>Primary<br>Secondary<br>Both | Enable the integrated local bus IDE adapter. Default setting is **Both.** |

### 4.3.4. PCI Configuration

Figure 7 shows the PCI Configuration submenu settings.

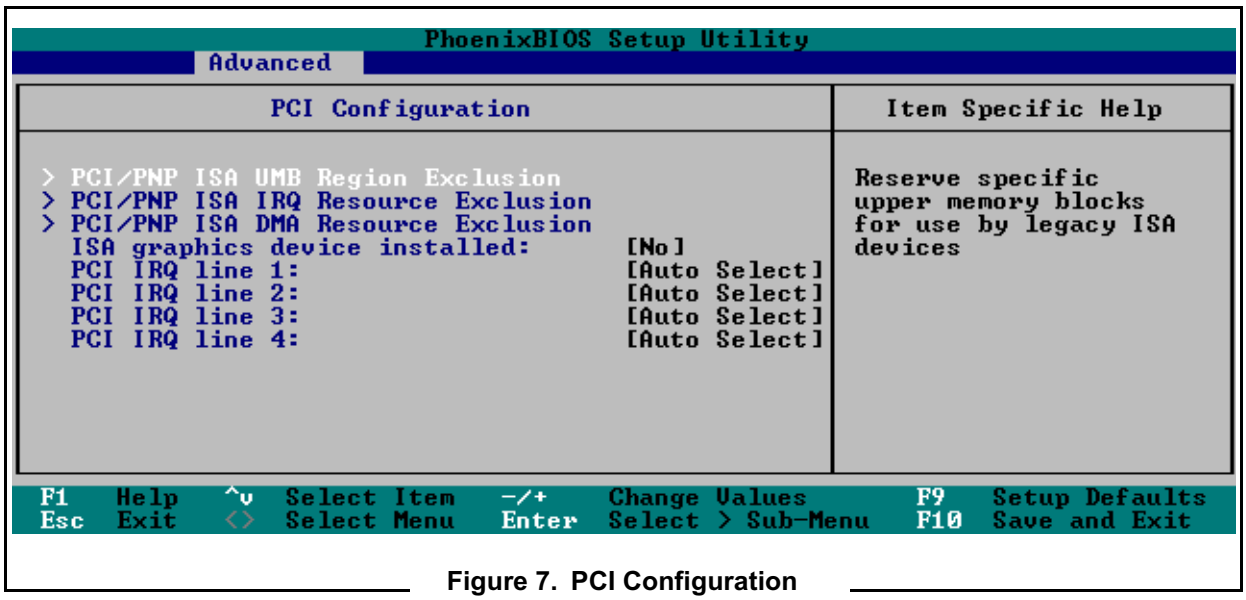



**Figure 7. PCI Configuration**

**Table 6: PCI Configuration**

| Feature | Options | Description |
|---|---|---|
| PCI/PNP ISA UMB Region Exclusion | | Reserve specific upper memory blocks for use by legacy ISA devices. See *'PCI/PNP ISA UMB Region Exclusion'* on page 15. |
| PCI/PNP ISA IRQ Resource Exclusion | | Reserve specific IRQ's for use by legacy ISA devices. See *'PCI/PNP ISA IRQ Resource Exclusion'* on page 16. |
| PCI/PNP ISA DMA Resource Exclusion | | Reserve specific DMA channels for use by legacy ISA devices. See *'PCI/PNP ISA DMA Resource Exclusion'* on page 17. |
| ISA graphics device installed: | No<br>Yes | Enable ISA (non-VGA) graphics device to access palette data in PCI VGA device. Default setting is **No**. |
| PCI IRQ line 1<br>PCI IRQ line 2<br>PCI IRQ line 3<br>PCI IRQ line 4 | Disabled<br>Auto Select<br>3<br>4<br>5<br>7<br>9<br>10<br>11<br>12<br>14<br>15 | PCI devices use hardware interrupts called IRQ's. A PCI device cannot use IRQ's already in use by ISA Enable. |

### 4.3.5. PCI/PNP ISA UMB Region Exclusion

Figure 8 shows the PCI/PNP ISA UMB Region Exclusion submenu selections. This menu may display with different values depending on your configuration .

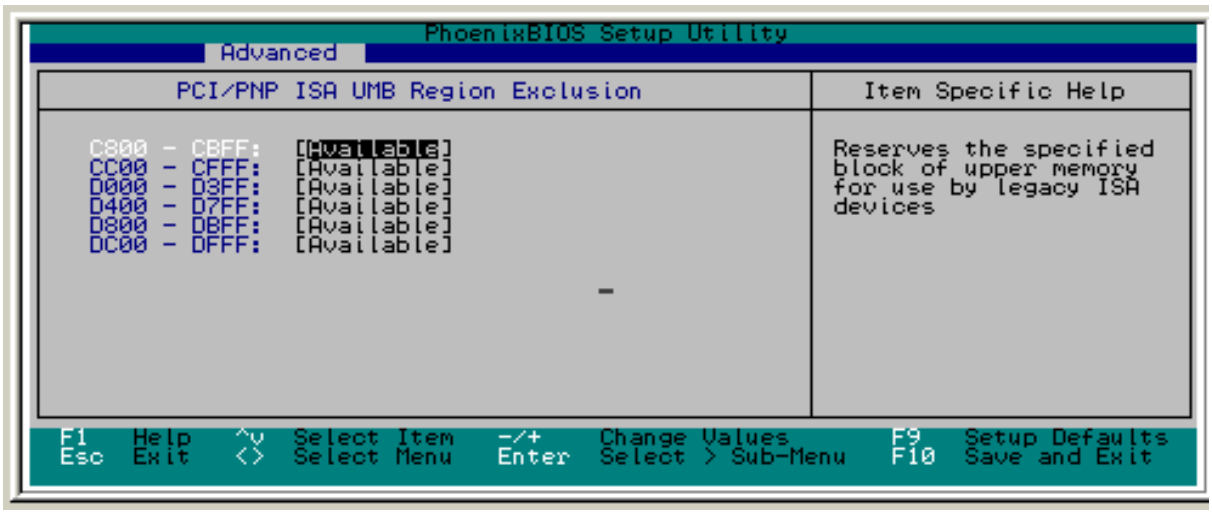



**Figure 8. PCI/PNP ISA UMB Region Exclusion**

**Table 7: PCI/PNP ISA UMB Region Exclusion**

| Feature | Options | Description |
|---|---|---|
| C800 - CBFF<br>CC00 - CFFF<br>D000 - D3FF<br>D400 - D7FF<br>D800 - DBFF<br>DC00 - DFFF | Available<br>Reserved | Reserves the specified block of upper memory for use by legacy ISA devices. Default setting is **Available**. |

### 4.3.6. PCI/PNP ISA IRQ Resource Exclusion

Figure 9 shows the PCI/PNP ISA IRQ Resource Exclusion submenu selections. This menu may display with different values depending on your configuration.

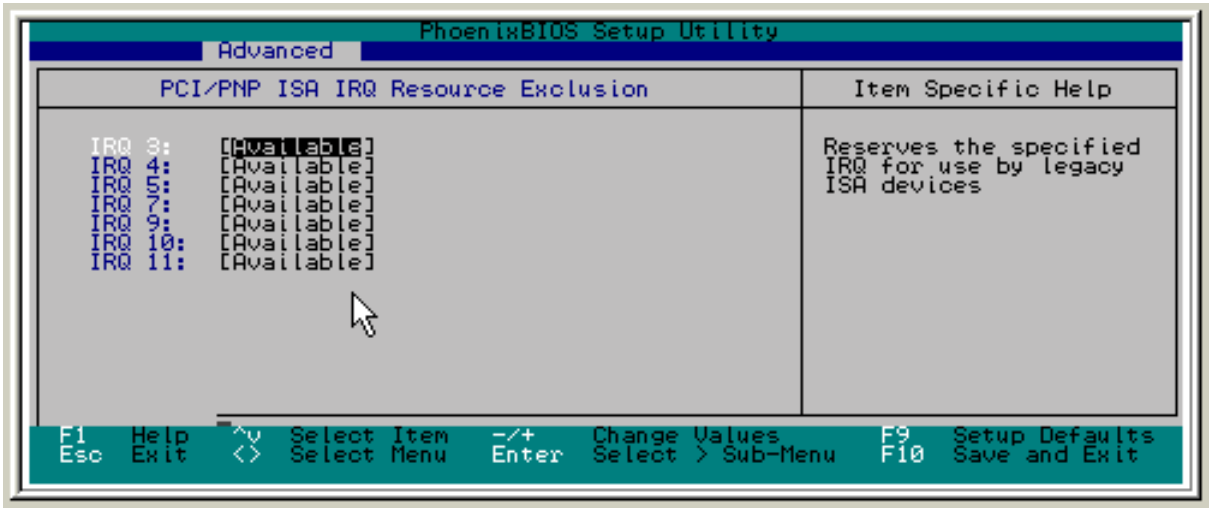



**Figure 9. PCI/PNP ISA IRQ Resource Exclusion**

**Table 8: PCI/PNP ISA IRQ Resource Exclusion**

| Feature | Options | Description |
|---|---|---|
| IRQ 3<br>IRQ 4<br>IRQ 5<br>IRQ 7<br>IRQ 9<br>IRQ 10<br>IRQ 11 | Available<br>Reserved | Reserves the specified IRQ for use by legacy ISA devices. Default setting is **Available**. |

### 4.3.7. PCI/PNP ISA DMA Resource Exclusion

Figure 10 shows the PCI/PNP ISA DMA Resource Exclusion submenu selections.

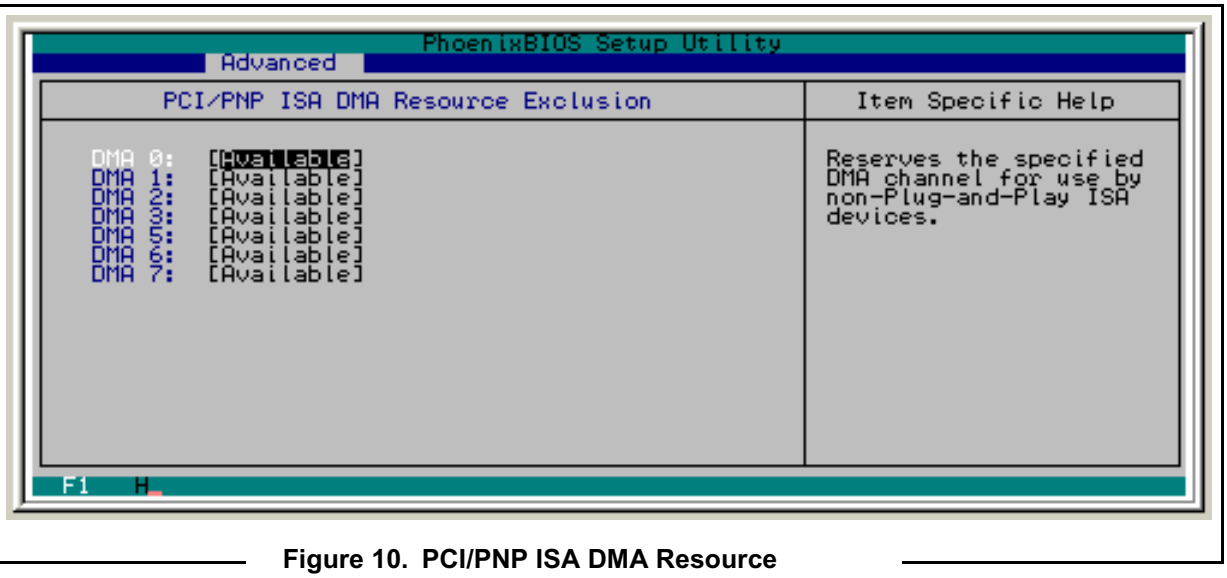



**Figure 10. PCI/PNP ISA DMA Resource**

**Table 9: PCI/PNP ISA DMA Resources Exclusion**

| Feature | Options | Description |
|---|---|---|
| DMA 0<br>DMA 1<br>DMA 2<br>DMA 3<br>DMA 4<br>DMA 5<br>DMA 6<br>DMA 7 | Available<br>Reserved | Reserves the specified DMA channel for use by non-Plug-and-Play ISA devices. Default setting is **Available**. |

### 4.3.8. Console Redirection

UCR (Universal Console Redirect) feature supports those embedded systems which do not use a keyboard or video monitor. The BIOS Binary Image file may be configured for this feature using the ZEB utility. See "Using the ZEB Editor with the IDS" on page 20.

A null modem cable connection is required between the ZFx86 COM A (COM1) and a PC/ANSI terminal or terminal emulator such as Procomm™ or Hyperterminal™. Text mode video operations are only supported during console redirect.

The operators at the other machine have the same controls (for example, using the "hot" keys and rebooting the system during POST). They also have the ability to enter SETUP and modify the settings stored in CMOS. The system-security features, however, cannot be circumvented if enabled.

The current limitation is that most non-DOS operating systems interfere with console redirection, disabling access by the other machine after POST. The reason for this is that UCR reserves the serial port for redirection use. The port chosen by the user in SETUP is removed from the BDA (BIOS Data Area). Any operating system that does **not** rely on the BDA to get information on the available serial ports, but instead, independently searches for them, will cause the connection to be lost after POST. This event is true for all applications, including mouse drivers.

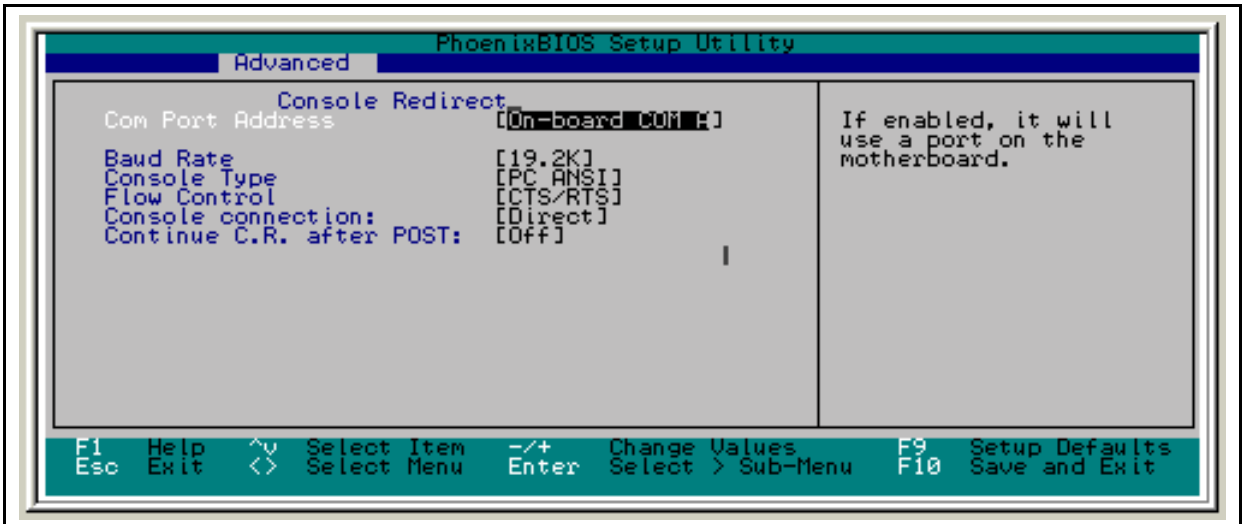



**Figure 11. Console Redirection**

**Table 10: Console Redirection**

| Feature | Options | Description |
|---|---|---|
| Com Port Address | Disabled<br>On-board COM A<br>On-board COM B | If enabled, ZFx86 uses a port on the motherboard. Default setting is **Disabled**. (Console redirection default is **COM A**.) |
| Baud Rate | 300<br>1200<br>2400<br>9600<br>19.2 K<br>38.4 K<br>57.6 K<br>115.2 K | Enables the specified baud rate. Default setting is **19.2K**. (Console redirection default is **19.2K**.) |

**Table 10: Console Redirection (Continued)**

| Feature | Options | Description |
|---|---|---|
| Console Type | PC ANSI<br>VT100 | Enables the specified console type. Default setting is **PC ANSI.** |
| Flow Control | None<br>XON/XOFF<br>CTS/RTS | Enables Flow Control. Default setting is **CTS/RTS.** |
| Console connection | Direct<br>Via modem | Indicates whether the console is connected directly to the system or a modem. Default setting is **Direct**. |
| Continue C.R. after POST | Off<br>On | Enables Console Redirection after OS loads. Default setting is **Off**. (Console redirection default is **On**.) |

### 4.4. Other Setup Screens

For information on the screens below, refer to PhoenixBIOS™ 4.0 Rev6 User Manual.PDF which is supplied on the ZF CD ROM.

- Security
- Power
- Boot
- Exit
- Help

## 5. ZFlash OS Loader

This feature allows Operating Systems, such as Linux or VxWorks to load and boot from the same flash device that holds the BIOS.

Prior to a boot attempt of standard media devices, the ZFx86 BIOS scans external Flash devices address blocks. These blocks are defined by the user configurable Memory Window Chip Selects and contain a standard legacy ISA extension ROM header. A checksum (modulo 100h) is performed, and if successful, the BIOS transfers the boot attempt to code beginning at byte 3 of the special header. When the transfer is executed, a signature parameter value of 'MORX' is passed in register EDX, allowing the user to authenticate the call.

Thus, user supplied external code may be installed in flash and can continue the boot function using its own algorithms. Details are available from ZF Micro Devices. Ask support for the "*Booting User Software From Flash Chips*" Software Note (P/N 9100-0067-00) and Loading Linux form Flash (P/N 9150-0017-00), or download these documents from the ZF Micro Devices website: *http://www.zfmicro.com*

# 6. Using the ZF Edit BIOS (ZEB) Utility

The ZF Edit BIOS utility, ZEB.EXE, allows you to establish custom default BIOS settings. The editor is ideal for those ZFx86 embedded systems with no battery backed CMOS storage, and allows additional debug flexibility when you bring up new designs. ZEB runs in either DOS or Windows.

Remember to disable your CMOS battery (if your system contains one) and to completely re-power the system each time you experiment with BIOS changes. The power reset guarantees that the BIOS reboot launches with the new defaults you selected.

This utility supports creating a debug version of the BIOS that outputs POST Codes on the serial port. See the ZEB Debug menu item.

## 6.1. Using the ZEB Editor with the IDS

Use ZEB to change the default CMOS BIOS settings. Remember to disable the CMOS battery by setting pins 2 and 3 to Clear. Refer to Figure 12 for the JP5 jumper location.

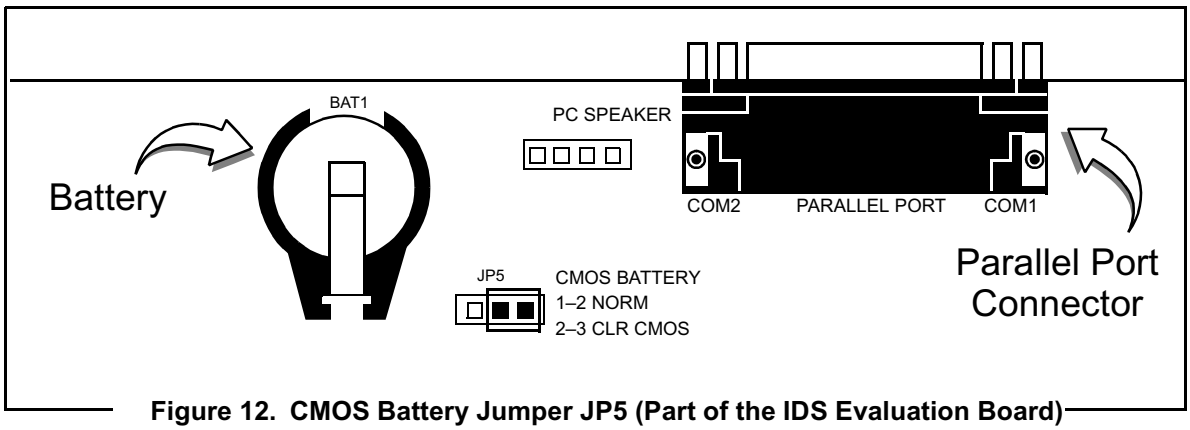



**Figure 12. CMOS Battery Jumper JP5 (Part of the IDS Evaluation Board)**

The ZEB menus contain information on how to change BIOS settings. The following procedure illustrates how to change the default Summary Window setting.

1. Start ZEB at a DOS prompt by typing the following:

   **ZEB <your_image_name.rom>**

   If you do not enter a BIOS Image name (<your_image_name.rom>), a listing of directories and files found in the launch directory displays. Use the arrow keys to select the BIOS Image you want to edit, and press Enter.

   Use the up and down arrow keys to select an item on the start up menu.

2. Select the **Edit Defaults** menu item, and press the Enter key. See Figure 13.

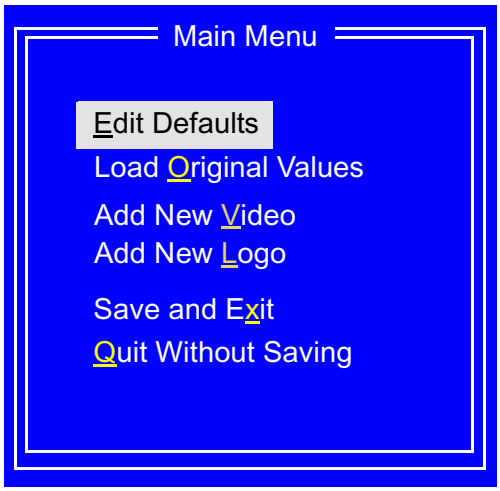



**Figure 13. ZEB Main Menu**

3. The ZEB utility displays seven top-level menu selections. Using the right and left arrow keys, select the **Main** menu item. See Figure 14.

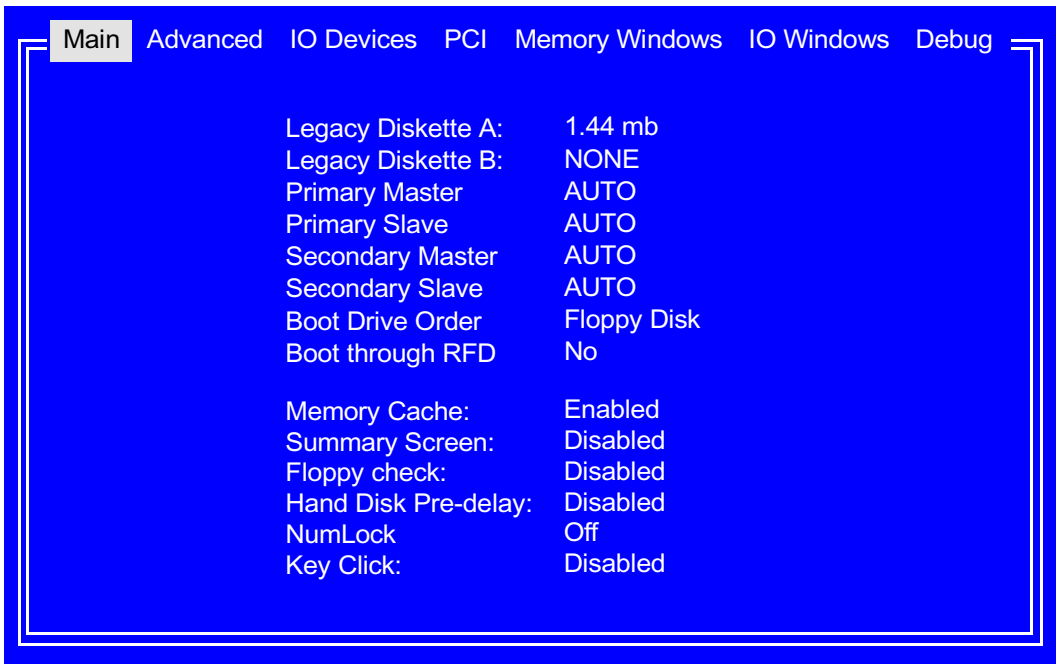



**Figure 14. ZEB Main Menu Selections**

4. Press Enter to access the Main menu.
   a. Use the down arrow to move down the menu selections until the **Summary Screen** item is selected.

b. Use the right and left arrow keys to toggle the Enabled or Disabled selection. See Figure 15.

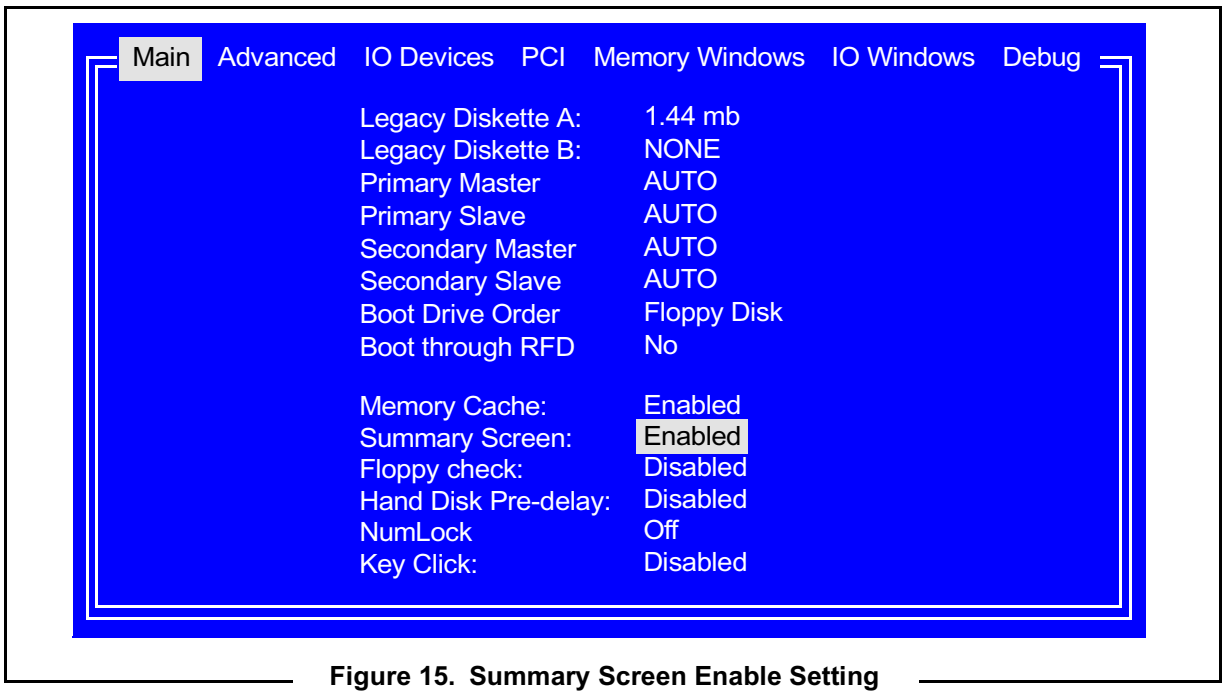



**Figure 15. Summary Screen Enable Setting**

c. With Enabled selected, press the **Escape** key to exit the Main menu.

5. Press the **Escape** key again to exit the ZEB Main menu.

a. Use the down arrow to select **Save and Exit**. See Figure 16.


**Figure 16. ZEB Main Menu**

b. Press **Enter**. The Save & Exit? command prompt displays.

c. Press the **Enter** key to save the changed BIOS file. See Figure 17.


**Figure 17. Save and Exit Menu**

d. Type a new file name, or type the original file name in the Output Filename: prompt as desired. Remember, file names may not exceed eight characters in length. See Figure 18.

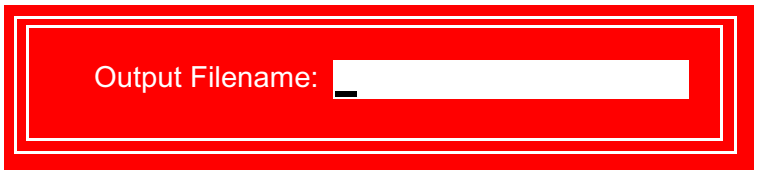



**Figure 18. Output Filename Menu**

e. Press the **Enter** key.

The ZEB utility exits, and returns you to the DOS prompt.

### 6.1.1. Loading A Custom Splash Screen Graphic

The ZEB utility allows you to add a custom splash screen graphic into your BIOS image file.

1. If you have exited the ZEB utility, relaunch it by typing the following at a DOS prompt:

   **ZEB <your_image_name.rom>**

   NOTE: You must load the zfx10600.**rom** image as the editable BIOS file, because ZEB substitutes the testlogo image or your new splash screen image with the default ZF Micro logo graphic. *Do not use the zfx10600.**ron** image file as it does not contain a splash screen graphic.*

2. Select the **Add New Logo** menu item from the Main Menu, and press the Enter key. See Figure 19.

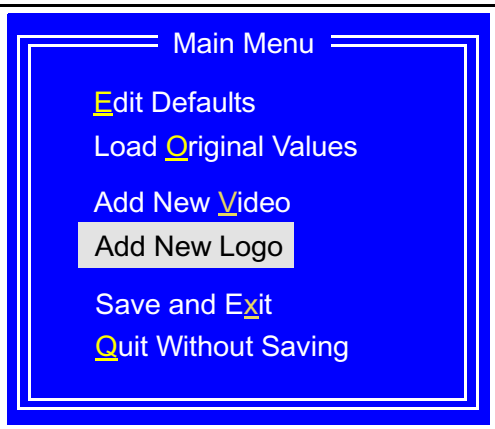



**Figure 19. ZEB Main Menu**

The ZEB utility displays the Logo filename screen. See Figure 20.

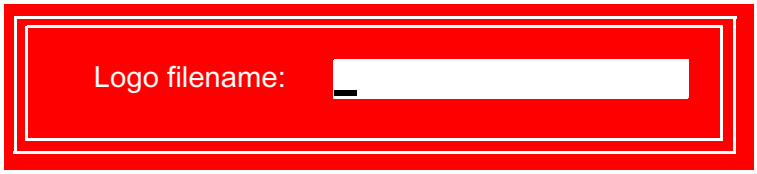



**Figure 20. Logo Filename Menu**

3. Enter your Logo filename or use the generic 640x480x256 color splash screen bit map graphic named testlogo.bmp included with this BIOS release.

NOTE: Your logo *must not* exceed the generic 640x480x256 color bit map size.

4. Press Enter. The Save & Exit? command prompt displays.
5. Press the **Enter** key to save the changed BIOS file. See Figure 21.


**Figure 21. Save and Exit Menu**

6. Type a new file name, or type the original file name in the Output Filename: prompt as desired. Remember, file names may not exceed eight characters in length. See Figure 22.

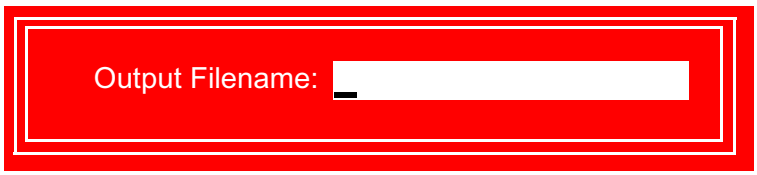



**Figure 22. Output Filename Menu**

7. Press the **Enter** key.

The ZEB utility exits, and returns you to the DOS prompt.

### 6.1.2. Copying <your_image.rom> to BIOS.ROM file

To verify that your BIOS changes are correct, copy your edited BIOS image to the AMDFLASH directory using the "/b" binary switch. Remember to rename your edited image to BIOS.ROM using the following copy command, for example:

**C:\AMDFLASH> COPY /B <your_image_name.rom> BIOS.ROM**

### 6.1.3. Removing AC-Power to the IDS

The IDS non-volatile RAM is battery backed, and although you have disabled the battery by jumpering JP5 pins 2 and 3 (see Figure 12), you must also re-power the IDS board to erase the original data in the non-volatile RAM.

### 6.1.4. Loading A PCI Video BIOS

ZEB and the ZFx86 BIOS provides an external load capability for a customer supplied PCI Video BIOS. A PCI Video BIOS binary image may be integrated permanently into your ZFx86 BIOS binary image. Note that it is your responsibility to ensure that the Video BIOS image matches your PCI video hardware, and also complies with all licensing and copyrights pertaining to the Video BIOS that you use.

The Zfx86 BIOS reserves space, up to 28 Kbytes, to accommodate a compressed custom Video BIOS. The ZEB editor automatically compresses the Video BIOS binary image when you integrate it into a revised ZFx86 BIOS binary image. Generally, an uncompressed Video BIOS of up to 48 Kbytes will compress and fit into the binary image's reserved space.

If you are concerned whether your video BIOS image will fit in this space, we suggest you pre-compress it using a utility such as PKZIP to verify that it compresses to 28 Kbytes or less.

NOTE: Do not attempt to load a previously compressed Video BIOS image using ZEB. ZEB always assumes the Video BIOS binary image is uncompressed.

1. If you have exited the ZEB utility, relaunch it by typing the following at a DOS prompt:

   **ZEB <your_image_name.rom>**

   NOTE: You may load either the zfx10600.**rom** or **.ron** image as the editable BIOS file.

2. Select the **Add New Video** menu item from the Main Menu, and press the Enter key. See Figure 23.

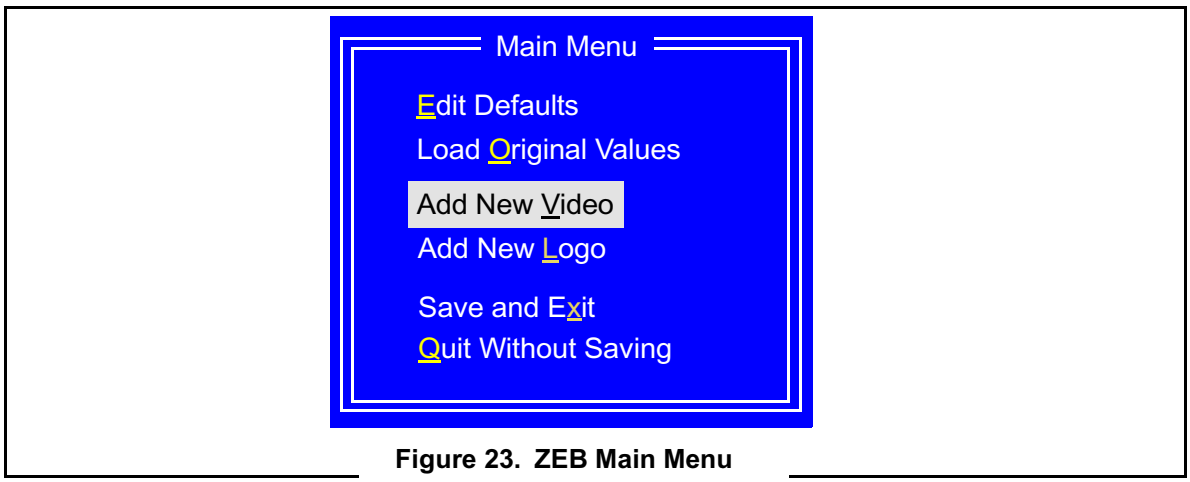



**Figure 23. ZEB Main Menu**

The ZEB utility displays the **Video filename** screen. See Figure 24.

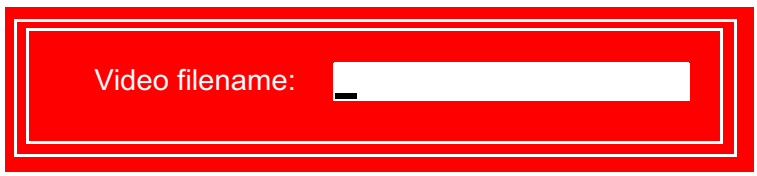



**Figure 24. Video Filename Menu**

3. Enter your Video filename. Note that file names may not exceed eight characters in length.

4. Press Enter. The **Save & Exit?** command prompt displays. See Figure 26.


**Figure 25. Save and Exit Menu**

5. Press the **Enter** key to save the changed ZFx86 BIOS file.
6. Type a new file name, or type the original file name in the **Output Filename:** prompt as desired. Remember, file names may not exceed eight characters in length. See Figure 26.

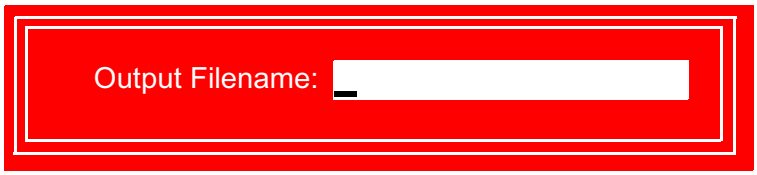



**Figure 26. Output Filename Menu**

7. Press the **Enter** key.

The ZEB utility exits, and returns you to the DOS prompt.

# 7. Technical Tips

This section discusses items and techniques particular to the effective use of the ZFx86 BIOS.

## 7.1. Understanding Memory Windows

Memory windows are a feature of the ZF Logic. Memory windows provide chip select and addressing for SRAM and Flash Chips external to the ZFx86 chip.

Memory windows benefit the user two ways:

- Allows interconnect to Flash and SRAM chips without extra glue logic
- Allows addressing of up to 64 MB of external Flash/SRAM (4 chip selects times 16 MB per chip select) through an aperture in the ISA address space called a memory window.

### 7.1.1. Memory Window Basics

XT computers contained a paged memory scheme that allows the XT to access memory above 1 MB. A small aperture (window) below 1 MB was used to address perhaps 1 to 8 MB of additional RAM through page switching.

The memory windows on the ZFx86 work in a similar manner. From a DOS prompt, run the program AMDFLASH on the ZFx86 and specify option P in the Main Menu screen. You see a printout similar to the following:

```
Phoenix ZFx86 BIOS Memory Window Setup Calculator
Window OK at C8000–CBFFFH
Window OK at CC000–CFFFFH
Window OK at D0000–D3FFFH
Window OK at D4000–D7FFFH
Window OK at D8000–DBFFFH
Window OK at DC000–DFFFFH
```

These numbers specify where to place the windows (apertures) in the memory space below 1 MB. Actually, it specifies memory in the range of C0000 through FFFFF which is not shadowed. The ZFx86 BIOS lives in the upper 128K of the first megabyte, that is, addresses E0000 through F0000. However, to allow the BIOS to run faster, rather than read it from 8-bit wide flash, the BIOS is copied into the typical 32-bit wide SDRAM. Then the SHADRC and SHADWC registers in the ZFx86 North Bridge hardware are set to enable read from the SDRAM and not to enable Write to the SDRAM. This results in memory reads directed to this address space that are also directed to the North Bridge SDRAM controller, which in turn is generally connected to high performance 32-bit wide SDRAM.

Addresses not shadowed, which are in the range C0000 to FFFFF, are directed to the ISA bus. Any addresses on the ISA bus, which are below E0000, may be used as memory windows.

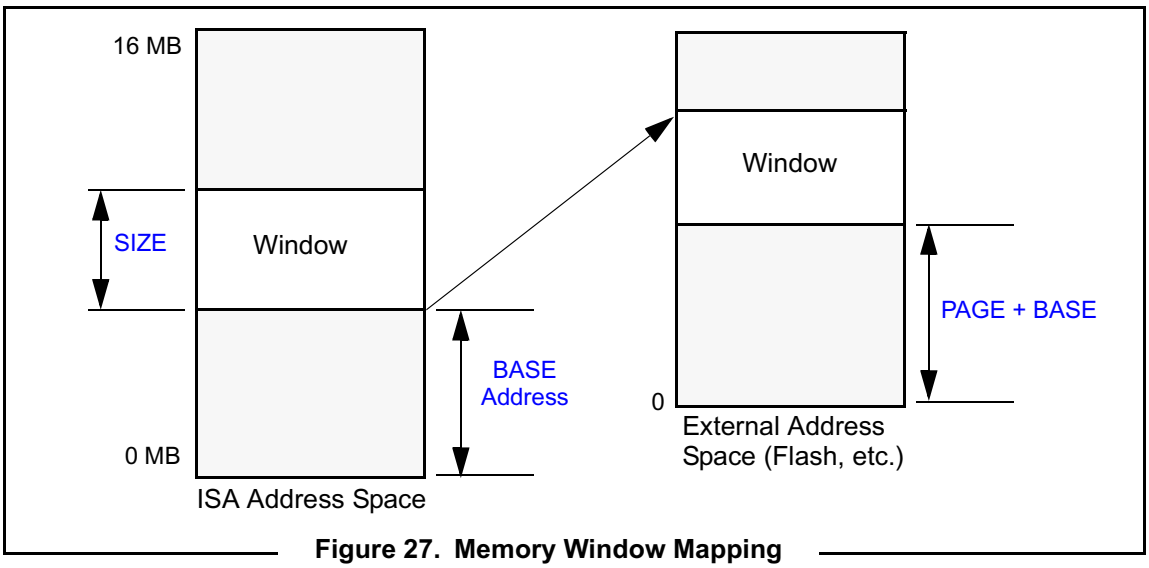



**Figure 27. Memory Window Mapping**

To use these memory windows to access external SRAM or FLASH chips, the ZFx86 allows you to activate one of four mem_csn pins (where n=0,1,2, or 3) when addresses between BASE and BASE+SIZE are encountered.

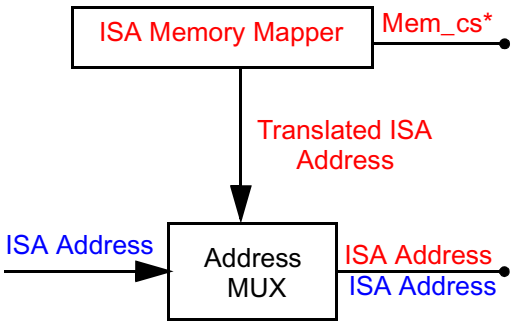



**Table 11: Memory Mapper Pins**

| Pin | Name* | Description |
|---|---|---|
| B04 | Mem_cs0 | ZF Logic Memory Mapper CS 0 |
| D05 | Mem_cs1 | ZF Logic Memory Mapper CS 1 |
| A03 | Mem_cs2 | ZF Logic Memory Mapper CS 2 |
| C04 | Mem_cs3 | ZF Logic Memory Mapper CS 3 |

**Figure 28. ISA Memory Map Pins**

Thus if you set base to D0000 and size to 64K (and those addresses were available), whenever you execute a memory read or a memory write in the range of D0000 to DFFFF you also generate a memory chip select. Which chip select would you get? Well, there are four register sets, one for each of the four mem_cs signals.

The address which comes out on the ISA bus is translated by the memory window hardware. In general, the question is this: how do I move the target window through my flash chip so that using the fixed window I have set up in the ZFx86 address space I can reference all of the data in the flash chip (up to 16 MB). This answer is embodied in a formula for setting the value of the PAGE register:

PAGE = 1000000 – BASE + FLASHA.

If BASE = 0D0000 then set PAGE to F30000 so that D000:0 goes to address 0 in the Flash. That is 1000000 – D0000 + 0.

For D0000 to go to D0000 in the flash, set PAGE to 0. That is, 1000000 – D0000 + D0000. (only specify 6 digits).

### 7.1.2. Using the BIOS to Set Initial Memory Window Positions

AMDFLASH contains a calculator to make the initial setup of your windows easier. Set the initial values of the BASE/SIZE/PAGE registers using exactly the same technique you use in your own software to reset or move these memory windows.

The operative part of the BASE/SIZE/PAGE registers is a 12 bit field (3 hex digits) out of a 32-bit register. The BASE/SIZE/PAGE registers are 32-bits wide, but only bits 23–12 are used (12 bits or 3 hex digits). The possible ranges of data written to these registers is thus 000000H – FFF000H.

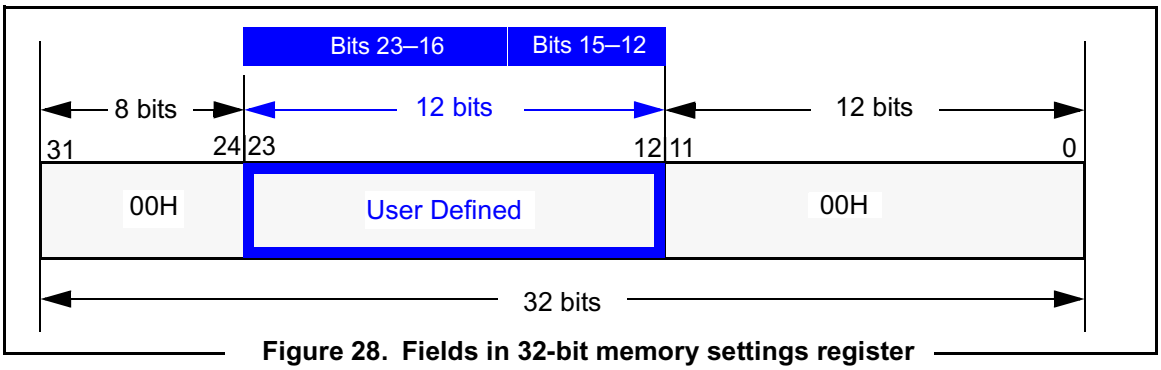



**Figure 28. Fields in 32-bit memory settings register**

Since each field is 12 bits wide, the actual data values are 000 to FFF hex (in the 12 bit field).

If you want the window to start at D0000 in the ZFx86 address space, set the BASE to 0D0000 hex, that is, 0D0 in the 12-bit field and in the BIOS settings for the Memory Window BASE.

Table 12 shows the values used to setup a Memory Window Chip Select that enables blocks placed at flash address 0 to be mapped to ISA addresses shown in the first column.

**Table 12: Sample Window Calculations**

| ISA Window Base | BASE (hex) | PAGE (hex) |
|---|---|---|
| C8000 | C8 | F38 |
| CC000 | CC | F34 |
| D0000 | D0 | F30 |
| D4000 | D4 | F26 |
| D8000 | D8 | F28 |
| DC000 | DC | F24 |

Calculate the window SIZE as BASE to BASE+SIZE where it is assumed that the right most 12 bits are implicitly FFF. Therefore, a size of 001000 hex would be 1FFF, thereby providing an 8K window. Note that you might think that 0 would provide a 4K window; however, we decode 0 internally to disable the window.

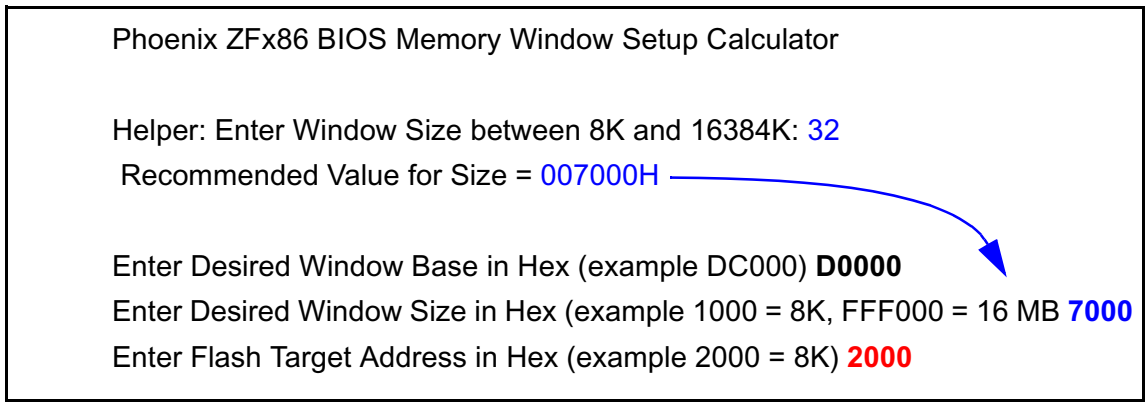
Treat the PAGE field as a signed number (if PAGE = –BASE then the target is the beginning of the flash), but because all 32 bits are not available, it becomes a bit tricky.

It is best to use the calculator built into AMDFLASH. An example of the printout and a copy of the source code in C follows.

### Sample Code for Initial Memory Window Positions

Create a **32K** window starting at **D0000** in the ZFx86 address space. The window initially points to an **offset 2000H** within the flash chip.

Example: Set the Page register to 00F32000H (or F32 in the ZFx86 Phoenix BIOS Memory Window Setup screen).




NOTE: In the following example, the program creates a 32K window starting at D0000H which points initially to offset 2000H in the flash using BASE = 0D0H, SIZE = 007H, and PAGE = F32H.

```
1     unsigned long ulBase, ulSize, ulTarget, ulPage, ulDesiredK;
2     printf ("\r\n\n\nPhoenix ZFx86 BIOS Memory Window Setup Calculator \r\n", uiWorkingCS);
3
4     printf ("\r\nHelper: Enter Window Size between 8 and 16384K: ");
5     ulDesiredK = uiScanInDecimal(0);
6
7     if ((ulDesiredK >= 8) && (ulDesiredK <= 16384))
8           printf ("\r\n Recommended Value for ""Size"" = %06lXH\r\n", (ulDesiredK-4) * 1024 );
9
10     printf ("\r\nEnter Desired Window Base in Hex (example DC000) ");
11        ulBase = ulScanInHex ();
12     printf ("\r\nEnter Desired Window Size in Hex (example 1000 = 8K, FFF000 = 16 MB ");
13        ulSize = ulScanInHex ();
14        printf ("\r\nEnter Flash Target Address in Hex (example 2000 = 8K) ");
15        ulTarget = ulScanInHex ();
16              ulBase = ulBase & 0xFFF000;
17        ulSize = ulSize & 0xFFF000 ;
18        ulTarget = ulTarget & 0xFFF000;
19        printf ("\r\nBase = %08lXH or %03lXH or %04ld decimal", ulBase, ulBase >> 12, ulBase
                    >>12);
20        if (ulSize == 0) printf ("\r\nWindow Disabled Size == 0");
21        else
22        printf ("\r\nSize = %08lXH or %03lXH or %04ld decimal for Size = %dK", ulSize, ulSize >> 12,
                    ulSize >>12, ((ulSize >> 12) + 1) * 4);
23        ulPage = (0x1000000 - ulBase + ulTarget) & 0xFFF000;
24        printf ("\r\nPage = %08lXH or %03lXH or %04ld decimal for Flash Offset = %08lXH\r\n\n",
25                  ulPage, ulPage >> 12, ulPage >>12, ulTarget);
```

## 7.2. Using the Watchdog TImer

The ZFx86's Watch Dog Timer function is implemented using the following interface:

INT 15H System Services Interrupt
Calling Parameters:

Register AH = C3H Enable/Disable Watchdog Timer
Register AL = 00 Disable Watchdog Timeout

Register AL = 01 Enable Watchdog Timeout
Register BX = 1 - 255 seconds

Return Parameters:

Register Flag CF (Carry Flag) = 0 = Operation Complete
Register Flag CF (Carry Flag) = 1 = Operation Failed

WDTON.EXE is a sample test program for WDT functions. WDTON.c is the source code written in MS C and may be used as a tutorial for developing user based WDT routines in Assembler or Other C language compilers.

As an example, set the IDS board's watchdog Jumper J2 to short pin 2 to 3. (WD OSC INT). Then, at a DOS prompt, run WDTON 10. The WDTON utility displays the time continuously for 10 seconds. The test keeps the ZFx86's watchdog timer from firing until the ten second interval. At that time the watchdog timer fires and the system resets.

**WDTON.C Sample Program**

```
6   // Tests WDT INT 15 Functions
7
8   #include <ctype.h>
9   #include <io.h>
10   #include <dos.h>
11   #include <stdlib.h>
12   #include <stdio.h>
13   #include <string.h>
14   #include <time.h>
15
16   void main(argc,argv)
17   int argc;
18   char *argv[];
19   {
20   char tbuffer[9];
21   union REGS inregs, outregs;
22   int delay_value=0;
23   int wait_value=0;
24   int start_secs = 0;
25   int stop_secs = 0;
26   time_t t1,t2;
27
28   setbuf(stdout,NULL);
```

```
29
30  if(argc !=2)                        // only two args allowed
31     { printf("\nError: \n");
32       printf("Format: WDTON <secs>  i.e WDTON 10"); exit(EXIT_FAILURE);}
33
34  delay_value = (abs(atoi(argv[1])));
35  wait_value = delay_value*2;
36  system("cls");
37  //_strtime (tbuffer);
38  printf("\nWatch Dog Timer Tickle Test\n");
39  printf("\nSystem will REBOOT in %2.2d seconds",delay_value);
40  printf("\nStart Time  %s\n", _strtime (tbuffer));
41
42  //start_secs = (atoi(&tbuffer[6]));
43  start_secs = (time(&t1));
44  //printf("Startsecs == %d\n", start_secs);
45  //stop_secs =+ (atoi(&tbuffer[6]));
46  //printf("Start Secs   %d\n", start_secs);
47
48  time(&t1);
49
50  while ((difftime(t2,t1) != delay_value+1))
51     {
52     printf("Tickling WDT %s\r", _strtime (tbuffer));
53     time(&t2);
54     }
55  printf("WDT FAILED   %s\n", _strtime (tbuffer));
56  printf("\n");
57  while ((difftime(t2,t1) < delay_value+7))
58     {
59     printf("Waiting 5 more seconds: %s\r", _strtime (tbuffer));
60     time(&t2);
61     }
62  outregs.x.ax = 0xc301;
63  outregs.x.bx = delay_value;
64  int86(0x15 ,&inregs, &outregs);
65  printf("\nINT 15 function C3 01 failed CF = %d\n", outregs.x.cflag);
66  }
```

## 7.3. POST Code Information

Although Power On Self Test (POST) errors and beep codes are documented in the "Phoenix BIOS 4.0 Revision 6 User's manual," this section contains the latest comprehensive POST and beep codes. Table 13 lists the POST codes in numerical order. However, POST code routines are run in a logical order and some routines may run more than once. To aid debugging of a bring-up board, we have provided Table 14, which shows the actual sequence of these routines relevant to the 1.06 version BIOS.

**Table 13: POST Code Descriptions**

| POST Code | Function Name | Beeps | Description |
|---|---|---|---|
| 00h | NULL | | Not implemented |
| 01h | IPMI INIT | | Intelligent Platform Management Interface (POST Code Displayed But No Other Operation Is Performed) |
| 02h | VERIFY REAL | | Verify Real Mode |
| 03h | DISABLE NMI | | Disable Non-Maskable Interrupt (NMI) |
| 04h | GET CPU TYPE | | Get CPU type |
| 06h | HW INIT | | Initialize system hardware |
| 07h | CS BIOS DESHAD | | Disable shadow and execute code from the ROM. |
| 08h | CS INIT | | Initialize chipset with initial POST values |
| 09h | SET IN POST | | Set IN POST flag |
| 0Ah | CPU INIT | | Initialize CPU registers |
| 0Bh | CPU CACHE ON | | Enable CPU cache |
| 0Ch | CACHE INIT | | Initialize caches to initial POST values |
| 0Eh | IO INIT | | Initialize I/O component |
| 0Fh | FDISK INIT | | Initialize the local bus IDE |
| 10h | PM INIT | | Initialize Power Management |
| 11h | REG INIT | | Load alternate registers with initial POST values |
| 12h | RESTORE CR0 | | Restore CPU control word during warm boot |
| 13H | PCI BM RESET | | Initialize PCI Bus Mastering devices |
| 14h | 8742 INIT | | Initialize keyboard controller |
| 16h | CHECKSUM | 1-2-2-3 | BIOS ROM checksum |
| 17h | PRE SIZE RAM | | Initialize cache before memory Auto size |
| 18h | TIMER INIT | | 8254 timer initialization |
| 1Ah | DMA INIT | | 8237 DMA controller initialization |
| 1Ch | RESET PIC | | Reset Programmable Interrupt Controller |
| 20h | REFRESH | 1-3-1-1 | Test DRAM refresh |
| 22h | 8742 TEST | 1-3-1-3 | Test 8742 Keyboard Controller |
| 24h | SET HUGE ES | | Set ES segment register to 4 GB |
| 26h | ENABLE A20 | | Not implemented |
| 28h | SIZE RAM | | Auto size DRAM |
| 29h | PMM INIT | | Initialize POST Memory Manager |
| 2Ah | ZERO BASE | | Clear 512 kB base RAM |
| 2Bh | ENH CMOS INIT | | Enhances CMOS feature initalization (POST Code Displayed But No Other Operation Is Performed) |
| 2Ch | ADDR TEST | 1-3-4-1 | RAM failure on address line xxxx* |
| 2Eh | BASERAML | 1-3-4-3 | RAM failure on data bits xxxx* of low byte of |
| 2Fh | PRE SYS SHADOW | | Enable cache before system BIOS shadow |

**Table 13: POST Code Descriptions (Continued)**

| POST Code | Function Name | Beeps | Description |
|---|---|---|---|
| 30h | BASERAMH | | Not implemented |
| 32h | COMPUTE SPEED | | Test CPU bus-clock frequency |
| 33h | PDM INIT | | Initialize Phoenix Dispatch Manager |
| 34h | CMOS TEST | | Not implemented |
| 35h | REG REINIT | | Not implemented |
| 36h | CHK SHUTDOWN | | Warm start shut down |
| 37h | CS REINIT | | Not implemented |
| 38h | SYS SHADOW | | Shadow system BIOS ROM |
| 39h | CACHE REINIT | | Not implemented |
| 3Ah | CACHE AUTO | | Auto size cache |
| 3Bh | DBGSRV INIT | | Debug Service feature initialization (POST Code Displayed But No Other Operation Is Performed) |
| 3Ch | ADV CS CONFIG | | Advanced configuration of chipset registers |
| 3Dh | ADV REG CONFIG | | Load alternate registers with CMOS values |
| 3Eh | READ HW | | Not implemented |
| 3Fh | ROMPILOT MEMORY | | Not implemented |
| 40h | SPEED | | Not implemented |
| 41h | ROMPILOT INIT | | Initialize extended memory for RomPilot |
| 42h | VECTOR INIT | | Initialize interrupt vectors |
| 44h | SET BIOS INT | | Not implemented |
| 45h | DEVICE INIT | | POST device initialization |
| 46h | COPYRIGHT | 2-1-2-3 | Check ROM copyright notice |
| 47h | I2O INIT | | Initialize I20 support |
| 48h | CONFIG | | Check video configuration against CMOS |
| 49h | PCI INIT | | Initialize PCI bus and devices |
| 4Ah | VIDEO | | Initialize all video adapters in system |
| 4Bh | QUIETBOOT START | | QuietBoot start |
| 4Ch | VID SHADOW | | Shadow video BIOS ROM |
| 4Eh | CR DISPLAY | | Display BIOS copyright notice |
| 4Fh | MULTBOOT INIT | | Initialize MultiBoot |
| 50h | CPU DISPLAY | | Display CPU type and speed |
| 51h | EISA INIT | | Initialize EISA board |
| 52h | KB TEST | | Test keyboard |
| 54h | KEY CLICK | | Set key click if enabled |
| 55h | USB INIT | | Enable USB devices |
| 56h | ENABLE KB | | Not implemented |
| 58h | HOT INT | 2-2-3-1 | Test for unexpected interrupts |
| 59h | PDS INIT | | Initialize POST display service |
| 5Ah | DISPLAY F2 | | "Display prompt ""Press F2 to enter SETUP""" |
| 5Bh | CPU CACHE OFF | | Disable CPU cache |
| 5Ch | MEMORY TEST | | Test RAM between 512 and 640 kB |
| 5Eh | BASE ADDR | | Not implemented |
| 60h | EXT MEMORY | | Test extended memory |
| 62h | EXT ADDR | | Test extended memory address lines |
| 64h | USERPATCH1 | | Jump to UserPatch1 |

**Table 13: POST Code Descriptions (Continued)**

| POST Code | Function Name | Beeps | Description |
|---|---|---|---|
| 66h | CACHE ADVNCD | | Configure advanced cache registers |
| 67h | MP INIT MIN | | Initialize Multi Processor APIC |
| 68h | CACHE CONFIG | | Enable external and CPU caches |
| 69h | PM SETUP SMM | | Setup System Management Mode (SMM) area |
| 6Ah | DISP CACHE | | Display external L2 cache size |
| 6Bh | CUST DFLT | | Load custom defaults |
| 6Ch | DISP SHADOWS | | Display shadow-area message |
| 6Eh | DISP NONDISP | | Display possible high address for UMB recovery |
| 70h | ERROR MSGS | | Display error messages |
| 72h | TEST CONFIG | | Check for configuration errors |
| 74h | RTC TEST | | Not implemented |
| 76h | KEYBOARD | | Check for keyboard errors |
| 7Ah | KEYLOCK | | Not implemented |
| 7Ch | HW INTS | | Set up hardware interrupt vectors |
| 7Dh | ISM INIT | | Initialize Intelligent System Monitoring |
| 7Eh | COPROC | | Initialize coprocessor if present |
| 80h | IO BEFORE | | Disable onboard Super I/O ports and IRQs |
| 81h | LATE DEVICE INIT | | Late POST device initialization |
| 82h | RS232 | | Detect and install external RS232 ports |
| 83h | FDISK CFG IDE CTRLR | | Configure non-MCD IDE controllers |
| 84h | LPT | | Detect and install external parallel ports |
| 85h | PCI PCC | | Initialize PC-compatible PnP ISA devices |
| 86h | IO AFTER | | Re-initialize onboard I/O ports. |
| 87h | MCD INIT | | Configure Motherboard Configurable Devices |
| 88h | BIOS INIT | | Initialize BIOS Data Area |
| 89h | ENABLE NMI | | Enable Non-Maskable Interrupts (NMIs) |
| 8Ah | INIT EXT BDA | | Initialize Extended BIOS Data Area |
| 8Bh | MOUSE | | Test and initialize PS/2 mouse |
| 8Ch | FLOPPY | | Initialize floppy controller |
| 8Eh | AUTOTYPE | | Not implemented |
| 8Fh | FDISK FAST PREINIT | | Determine number of ATA drives |
| 90h | FDISK | | Initialize hard-disk controllers |
| 91h | FDISK FAST INIT | | Initialize local-bus hard-disk controllers |
| 92h | USERPATCH2 | | Jump to UserPatch2 |
| 93h | MP INIT | | Build MPTABLE for multi-processor boards |
| 95h | CD | | Install CD ROM for boot |
| 96h | CLEAR HUGE ES | | Clear huge ES segment register |
| 97h | MP FIXUP | | Fix up Multi Processor table |
| 98h | ROM SCAN | 1-2 | "Search for option ROMs. One long, two short beeps on check sum failure." |
| 99h | FDISK CHECK SMART | | Check for SMART Drive |
| 9Ah | MISC SHADOW | | Shadow option ROMs |
| 9Bh | PMCPUSPEED | | Not implemented (see POST code 32h + 50h) |
| 9Ch | PM SETUP | | Set up Power Management |
| 9Dh | SECURITY INIT | | Initialize security engine |

**Table 13: POST Code Descriptions (Continued)**

| POST Code | Function Name | Beeps | Description |
|---|---|---|---|
| 9Eh | IRQS | | Enable hardware interrupts |
| 9Fh | FDISK FAST INIT2 | | Determine number of ATA and SCSI drives |
| A0h | TIME OF DAY | | Set time of day |
| A2h | KEYLOCK TEST | | Check key lock |
| A4h | KEY RATE | | Initialize typematic rate |
| A8h | ERASE F2 | | Erase F2 prompt |
| AAh | SCAN FOR F2 | | Scan for F2 key stroke |
| ACh | SETUP CHECK | | Enter SETUP |
| AEh | CLEAR BOOT | | Clear Boot flag |
| B0h | ERROR CHECK | | Check for errors |
| B1h | ROMPILOT UNLOAD | | Inform RomPilot about the end of POST. |
| B2h | POST DONE | | POST done - prepare to boot operating system |
| B3h | ENH CMOS STORE | | stores CMOS data |
| B4h | ONE BEEP | 1 | One short beep before boot |
| B5h | QUIETBOOT END | | Terminate QuietBoot |
| B6h | PASSWORD | | Check password |
| B7h | ACPI | | Initialize ACPI BIOS |
| B8h | SYSTEM INIT | | Not implemented |
| B9h | PREPARE BOOT | | Prepare Boot |
| BAh | DMI | | Initialize SMBIOS |
| BBh | INIT BCVS | | Initialize PnP Option ROMs |
| BCh | PARITY | | Clear parity checkers |
| BDh | BOOT MENU | | Display MultiBoot menu |
| BEh | CLEAR SCREEN | | Clear screen |
| BFh | CHK RMDR | | Check virus and backup reminders |
| C0h | INT19 | | Try to boot with INT 19 |
| C1h | PEM INIT | | Initialize POST Error Manager (PEM) |
| C2h | PEM LOG | | Initialize error logging |
| C3h | PEM DISPLAY | | Initialize error display function |
| C4h | PEM SYSER INIT | | Initialize system error handler |
| C5h | DUAL CMOS | | PnPnd dual CMOS |
| C6h | DOCK INIT | | Initialize note dock |
| C7h | DOCK INIT LATE | | Initialize note dock late |
| C8h | FORCE | | Force check |
| C9h | EXT CHECKSUM | | Extended checksum |
| CAh | SERIAL KEY | | Redirect Int 15h to enable remote keyboard |
| CBh | ROMRAM | | Redirect Int 13h to Memory Technologies Devices such as ROM, RAM, PCMCIA, and serial disk |
| CCh | SERIAL VID | | Redirect Int 10h to enable remote serial video |
| CDh | PCMATA | | Re-map I/O and memory for PCMCIA |
| CEh | PEN INIT | | Initialize digitizer and display message |
| CFh | XBDA FAIL | | Allocates space for Extended BIOS data area components |
| D1h | BIOS STACK INIT | | BIOS stack initalization |

The **Event #** displayed in Table 14 indicates the POST Code sequence order for the ZFx86 BIOS version 1.06. Note that not all available POST Codes are used in the Boot sequence, and this sequence may change with other BIOS versions.

**Table 14: Boot Sequence POST Code**

| Event # | POST Code | Function Name | Beeps | Description |
|---|---|---|---|---|
| 1 | C3h | PEM DISPLAY | | Initialize error display function |
| 2 | C2h | PEM LOG | | Initialize error logging |
| 3 | 02h | VERIFY REAL | | Verify Real Mode |
| 4 | C8h | FORCE | | Force check |
| 5 | C9h | EXT CHECKSUM | | Extended checksum |
| 6 | 24h | SET HUGE ES | | Set ES segment register to 4 GB |
| 7 | 04h | GET CPU TYPE | | Get CPU type |
| 8 | 06h | HW INIT | | Initialize system hardware |
| 9 | 18h | TIMER INIT | | 8254 timer initialization |
| 10 | 07h | CS BIOS DESHAD | | Disable shadow and execute code from the ROM. |
| 11 | 08h | CS INIT | | Initialize chipset with initial POST values |
| 12 | C4h | PEM SYSER INIT | | Initialize system error handler |
| 13 | 0Eh | IO INIT | | Initialize I/O component |
| 14 | 0Ch | CACHE INIT | | Initialize caches to initial POST values |
| 15 | 16h | CHECKSUM | 1-2-2-3 | BIOS ROM checksum |
| 16 | 17h | PRE SIZE RAM | | Initialize cache before memory Auto size |
| 17 | 28h | SIZE RAM | | Auto size DRAM |
| 18 | CBh | ROMRAM | | Redirect Int 13h to Memory Technologies Devices such as ROM, RAM, PCMCIA, and serial disk |
| 19 | 2Ah | ZERO BASE | | Clear 512 kB base RAM |
| 20 | 2Ch | ADDR TEST | 1-3-4-1 | RAM failure on address line xxxx* |
| 21 | 2Eh | BASERAML | 1-3-4-3 | RAM failure on data bits xxxx* of low byte of |
| 22 | 0Ah | CPU INIT | | Initialize CPU registers |
| 23 | 3Ah | CACHE AUTO | | Auto size cache |
| 24 | 2Fh | PRE SYS SHADOW | | Enable cache before system BIOS shadow |
| 25 | 38h | SYS SHADOW | | Shadow system BIOS ROM |
| 26 | 01h | IPMI INIT | | Intelligent Platform Management Interface (POST Code Displayed But No Other Operation Is Performed) |
| 27 | 20h | REFRESH | 1-3-1-1 | Test DRAM refresh |
| 28 | 29h | PMM INIT | | Initialize POST Memory Manager |
| 29 | 33h | PDM INIT | | Initialize Phoenix Dispatch Manager |
| 30 | 2Bh | ENH CMOS INIT | | Enhances CMOS feature initalization (POST Code Displayed But No Other Operation Is Performed) |
| 31 | C1h | PEM INIT | | Initialize POST Error Manager (PEM) |
| 32 | 3Bh | DBGSRV INIT | | Debug Service feature initialization (POST Code Displayed But No Other Operation Is Performed) |
| 33 | 09h | SET IN POST | | Set IN POST flag |
| 34 | CEh | PEN INIT | | Initialize digitizer and display message |
| 35 | 3Ah | CACHE AUTO | | Auto size cache |
| 36 | 0Bh | CPU CACHE ON | | Enable CPU cache |
| 37 | D1h | BIOS STACK INIT | | BIOS stack initalization |
| 38 | 0Fh | FDISK INIT | | Initialize the local bus IDE |

## Table 14: Boot Sequence POST Code (Continued)

| Event # | POST Code | Function Name | Beeps | Description |
|---|---|---|---|---|
| 39 | 10h | PM INIT | | Initialize Power Management |
| 40 | 14h | 8742 INIT | | Initialize keyboard controller |
| 41 | 1Ah | DMA INIT | | 8237 DMA controller initialization |
| 42 | 1Ch | RESET PIC | | Reset Programmable Interrupt Controller |
| 43 | 22h | 8742 TEST | 1-3-1-3 | Test 8742 Keyboard Controller |
| 44 | C8h | FORCE | | Force check |
| 45 | 32h | COMPUTE SPEED | | Test CPU bus-clock frequency |
| 46 | 67h | MP INIT MIN | | Initialize Multi Processor APIC |
| 47 | 69h | PM SETUP SMM | | Setup System Management Mode (SMM) area |
| 48 | 6Bh | CUST DFLT | | Load custom defaults |
| 49 | 3Ch | ADV CS CONFIG | | Advanced configuration of chipset registers |
| 50 | 3Dh | ADV REG CONFIG | | Load alternate registers with CMOS values |
| 51 | 42h | VECTOR INIT | | Initialize interrupt vectors |
| 52 | 46h | COPYRIGHT | 2-1-2-3 | Check ROM copyright notice |
| 53 | 45h | DEVICE INIT | | POST device initialization |
| 54 | 47h | I2O INIT | | Initialize I20 support |
| 55 | 49h | PCI INIT | | Initialize PCI bus and devices |
| 56 | C6h | DOCK INIT | | Initialize note dock |
| 57 | C5h | DUAL CMOS | | PnPnd dual CMOS |
| 58 | 48h | CONFIG | | Check video configuration against CMOS |
| 59 | 4Ah | VIDEO | | Initialize all video adapters in system |
| 60 | 4Ch | VID SHADOW | | Shadow video BIOS ROM |
| 61 | 24h | SET HUGE ES | | Set ES segment register to 4 GB |
| 62 | 59h | PDS INIT | | Initialize POST display service |
| 63 | CCh | SERIAL VID | | Redirect Int 10h to enable remote serial video |
| 64 | 55h | USB INIT | | Enable USB devices |
| 65 | 52h | KB TEST | | Test keyboard |
| 66 | 54h | KEY CLICK | | Set key click if enabled |
| 67 | 76h | KEYBOARD | | Check for keyboard errors |
| 68 | 58h | HOT INT | 2-2-3-1 | Test for unexpected interrupts |
| 69 | 3Fh | ROMPILOT MEMORY | | Not implemented |
| 70 | C4h | PEM SYSER INIT | | Initialize system error handler |
| 71 | 7Ch | HW INTS | | Initialize note dock late |
| 72 | 41h | ROMPILOT INIT | | Initialize extended memory for RomPilot |
| 73 | 4Bh | QUIETBOOT START | | QuietBoot start |
| 74 | C6h | DOCK INIT | | Initialize note dock |
| 75 | 4Eh | CR DISPLAY | | Display BIOS copyright notice |
| 76 | 50h | CPU DISPLAY | | Display CPU type and speed |
| 77 | C9h | EXT CHECKSUM | | Extended checksum |
| 78 | 51h | EISA INIT | | Initialize EISA board |
| 79 | 9Dh | SECURITY INIT | | Initialize security engine |
| 80 | 5Ah | DISPLAY F2 | | +Display prompt "Press F2 to enter SETUP" |
| 81 | 5Bh | CPU CACHE OFF | | Disable CPU cache |
| 82 | 5Ch | MEMORY TEST | | Test RAM between 512 and 640 kB |

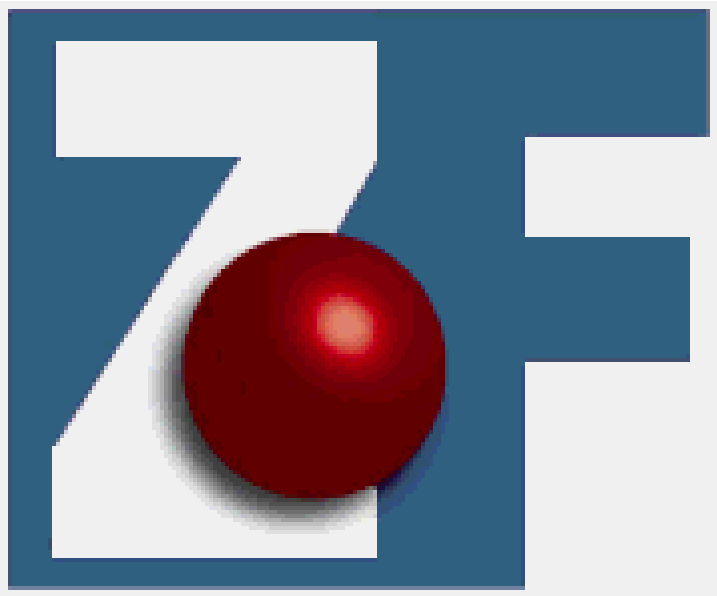
**Table 14: Boot Sequence POST Code (Continued)**

| Event # | POST Code | Function Name | Beeps | Description |
|---|---|---|---|---|
| 83 | 60h | EXT MEMORY | | Test extended memory |
| 84 | 62h | EXT ADDR | | Test extended memory address lines |
| 85 | 64h | USERPATCH1 | | Jump to UserPatch1 |
| 86 | 66h | CACHE ADVNCD | | Configure advanced cache registers |
| 87 | 68h | CACHE CONFIG | | Enable external and CPU caches |
| 88 | 6Ah | DISP CACHE | | Display external L2 cache size |
| 89 | 6Ch | DISP SHADOWS | | Display shadow-area message |
| 90 | CAh | SERIAL KEY | | Redirect Int 15h to enable remote keyboard |
| 91 | 70h | ERROR MSGS | | Display error messages |
| 92 | 72h | TEST CONFIG | | Check for configuration errors |
| 93 | 4Fh | MULTBOOT INIT | | Initialize MultiBoot |
| 94 | CDh | PCMATA | | Re-map I/O and memory for PCMCIA |
| 95 | 7Dh | ISM INIT | | Initialize Intelligent System Monitoring |
| 96 | 7Eh | COPROC | | Initialize coprocessor if present |
| 97 | C1h | PEM INIT | | Initialize POST Error Manager (PEM) |
| 98 | 80h | IO BEFORE | | Disable onboard Super I/O ports and IRQs |
| 99 | CAh | SERIAL KEY | | Redirect Int 15h to enable remote keyboard |
| 100 | 88h | BIOS INIT | | Initialize BIOS Data Area |
| 101 | 8Ah | INIT EXT BDA | | Initialize Extended BIOS Data Area |
| 102 | 81h | LATE DEVICE INIT | | Late POST device initialization |
| 103 | 87h | MCD INIT | | Configure Motherboard Configurable Devices |
| 104 | 85h | PCI PCC | | Initialize PC-compatible PnP ISA devices |
| 105 | 82h | RS232 | | Detect and install external RS232 ports |
| 106 | 84h | LPT | | Detect and install external parallel ports |
| 107 | 86h | IO AFTER | | Re-initialize onboard I/O ports. |
| 108 | 83h | FDISK CFG IDE CTRLR | | Configure non-MCD IDE controllers |
| 109 | CEh | PEN INIT | | Initialize error display function |
| 110 | 89h | ENABLE NMI | | Enable Non-Maskable Interrupts (NMIs) |
| 111 | 8Ch | FLOPPY | | Initialize floppy controller |
| 112 | CBh | ROMRAM | | Redirect Int 13h to Memory Technologies Devices such as ROM, RAM, PCMCIA, and serial disk |
| 113 | CDh | PCMATA | | Re-map I/O and memory for PCMCIA |
| 114 | 90h | FDISK | | Initialize hard-disk controllers |
| 115 | 8Bh | MOUSE | | Test and initialize PS/2 mouse |
| 116 | 95h | CD | | Install CD ROM for boot |
| 117 | 92h | USERPATCH2 | | Jump to UserPatch2 |
| 118 | B6h | PASSWORD | | Check password |
| 119 | 98h | ROM SCAN | 1-2 | Prepare Boot |
| 120 | 93h | MP INIT | | Build MPTABLE for multi-processor boards |
| 121 | 9Ch | PM SETUP | | Set up Power Management |
| 122 | C7h | DOCK INIT LATE | | Initialize note dock late |
| 123 | 9Eh | IRQS | | Enable hardware interrupts |
| 124 | A0h | TIME OF DAY | | Set time of day |
| 125 | A2h | KEYLOCK TEST | | Check key lock |
| 126 | CFh | XBDA FAIL | | Allocates space for Extended BIOS data area components |

## Table 14: Boot Sequence POST Code (Continued)

| Event # | POST Code | Function Name | Beeps | Description |
|---|---|---|---|---|
| 127 | A4h | KEY RATE | | Initialize typematic rate |
| 128 | C2h | PEM LOG | | Initialize error logging |
| 129 | BAh | DMI | | Initialize SMBIOS |
| 130 | A8h | ERASE F2 | | Erase F2 prompt |
| 131 | AAh | SCAN FOR F2 | | Scan for F2 key stroke |
| 132 | ACh | SETUP CHECK | | Enter SETUP |
| 133 | AEh | CLEAR BOOT | | Clear Boot flag |
| 134 | B0h | ERROR CHECK | | Check for errors |
| 135 | B2h | POST DONE | | POST done - prepare to boot operating system |
| 136 | B4h | ONE BEEP | 1 | One short beep before boot |
| 137 | B5h | QUIETBOOT END | | Terminate QuietBoot |
| 138 | C5h | DUAL CMOS | | PnPnd dual CMOS |
| 139 | BEh | CLEAR SCREEN | | Clear screen |
| 140 | B6h | PASSWORD | | Check password |
| 141 | BCh | PARITY | | Clear parity checkers |
| 142 | B7h | ACPI | | Initialize ACPI BIOS |
| 143 | 9Bh | PMCPUSPEED | | Not implemented (see POST code 32h + 50h) |
| 144 | BDh | BOOT MENU | | Display MultiBoot menu |
| 145 | BFh | CHK RMDR | | Check virus and backup reminders |
| 146 | C8h | FORCE | | Force check |
| 147 | 91h | FDISK FAST INIT | | Initialize local-bus hard-disk controllers |
| 148 | 9Fh | FDISK FAST INIT2 | | Determine number of ATA and SCSI drives |
| 149 | 97h | MP FIXUP | | Fix up Multi Processor table |
| 150 | 99h | FDISK CHECK SMART | | Check for SMART Drive |
| 151 | B1h | ROMPILOT UNLOAD | | Inform RomPilot about the end of POST. |
| 152 | C7h | DOCK INIT LATE | | Initialize note dock late |
| 153 | B3h | ENH CMOS STORE | | stores CMOS data |
| 154 | B9h | PREPARE BOOT | | Prepare Boot |
| 155 | CCh | SERIAL VID | | Redirect Int 10h to enable remote serial video |



**ZF Micro Devices, Inc.**

**1052 Elwell Court**

**Palo Alto, California 94303**

**(650) 965-3800 · Fax 965-4050**

**www.zfmicro.com**