1052 Elwell Court, Palo Alto, CA 94303 Tel: 800.683.5943 www.zfmicro.com

Using ZFx86 BIOS Extension ROMs

The ZFx86 is an “embedded-centric” BIOS. It contains features beyond those normally associated with traditional personal computers and notebook systems, aimed specifically at embedded applications.

One feature, discussed in this note, targets the ZFx86 System-On-Chip built-in external chip select logic. This chip select logic allows designers to place Extension ROMs in the same Flash device that contains the system BIOS.

A valuable part of the x86 legacy, Extension ROMs provide an architected way to introduce custom code and data into the BIOS. This extensibility gives the embedded designer the ability to control and assign system resources in the embedded device. The BIOS scans the Extension ROMs into the Flash after POST (Power On Self Test) and before the operating system boots under these conditions:

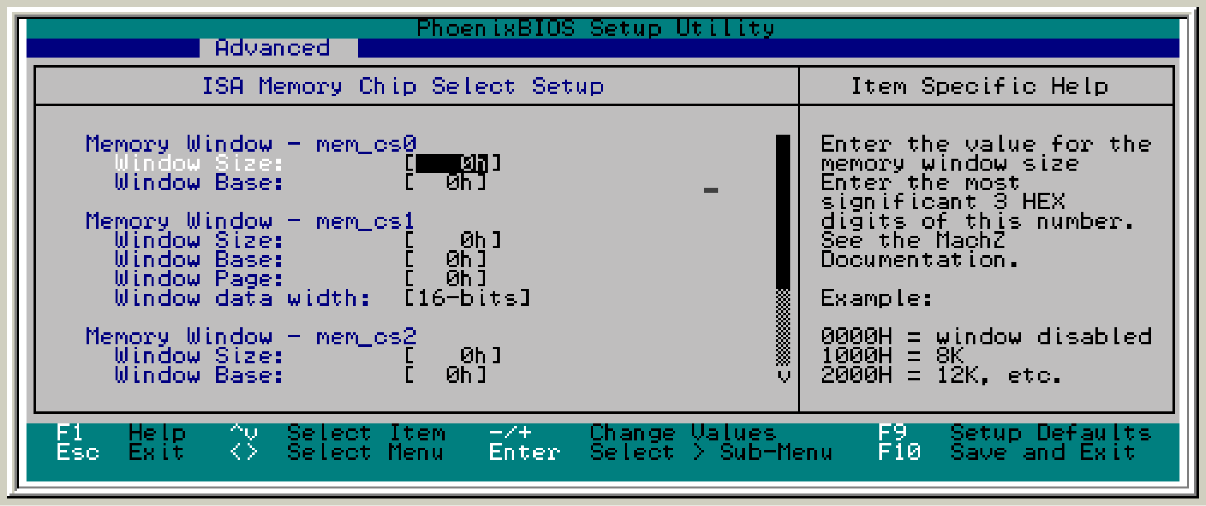
- If the Extension ROM module contains a block signature in its header of 055h and 0AAh in bytes zero and one, the ROM module progress.
- If the 55/AA signature is found, the system performs a checksum according to the length found in byte two (specified as the number of 512 byte blocks that comprise the module).
- Finally, if the checksum is correct, the BIOS transfers control to byte three, the far entry point of the Extension ROM module. The Module exits back to BIOS by executing a far return instruction. The scan takes place in System ROM linear memory space: 0xC0000 through 0xDffffh

In addition to the standard BIOS ROM Extension scan described above, the ZFx86 BIOS incorporates a secondary scan that occurs after POST and under control of the Memory Chip Select Window settings.

The ZFx86's on-chip logic supports external chip selects that the designer maps to either System I/O or Memory Space. Use the PhoenixBIOS Setup Utility **Advanced > ISA Memory Chip Select Setup** menu to set the following chip selects:

- Map any one of four chip-selects to any 8 or 16-bit input/output port.
- Or map an additional four chip-selects to system ROM space as Extension ROMs.

Figure 1 shows the PhoenixBIOS Chip Select Setup menu configuration settings used to map Extension ROMs.



```
                        PhoenixBIOS Setup Utility
            Advanced
        ISA Memory Chip Select Setup              Item Specific Help

   Memory Window - mem_cs0                      Enter the value for the
     Window Size:           [   0h]             memory window size
     Window Base:           [  0h]              Enter the most
                                                significant 3 HEX
   Memory Window - mem_cs1                      digits of this number.
     Window Size:           [   0h]             See the MachZ
     Window Base:           [  0h]              Documentation.
     Window Page:           [  0h]
     Window data width:     [16-bits]           Example:

   Memory Window - mem_cs2                      0000H = window disabled
     Window Size:           [   0h]             1000H = 8K
     Window Base:           [  0h]              2000H = 12K, etc.

 F1   Help  ^v  Select Item   -/+    Change Values          F9   Setup Defaults
 Esc  Exit  <>  Select Menu   Enter  Select > Sub-Menu      F10  Save and Exit
```

Figure 1. ISA Memory Chip Select Setup Menu

Use the code and data contained in an Extension ROM for a variety of tasks including device initialization procedures called once during BIOS boot, or an operational function called as needed, at OS run time. For example:

- OS load from Flash
- Custom Video BIOS
- Pre-OS Diagnostics
- Low level Part of an I/O Subsystem

Extension ROMs may provide any type of solution imaginable. The rest of this note uses a self-paced tutorial illustrating how you write, build, install, and run a custom Extension ROM using a simple example.

Most Extension ROM code is written in x86 Assembler Language. You may use Standard C Language as long as there are no direct or indirect system calls, because the Operating System is not initialized when the Extension Rom is called. The example uses the x86 Macro Assembly Language (MASM).

Writing The Extension ROM

The Extension ROM example below accomplishes the following things:

- Displays the value 0x11111111 on the ZF IDS 32-bit POST code port
- Displays the value for a few seconds
- Exits and allows the system BIOS to continue booting

The source code example is as follows:

```
; build and test a Extension ROM program
; build the ROM using MASM compiler:
; ml /Zm /AT ext01.asm
; create the checksum value using romext.exe (Download this executable file
; from the ZF Micro Devices’ website)
; romext ext01.asm

        .486p

code            segment use16
                assume cs:code

; extension-rom header

                db      55h
                db      0AAh
                db      00h      ; ext rom size in 512 byte blocks calculated
                                 ; by ROMEXT.EXE which also pads and
                                 ; checksums this file
                jmp     start


start:

        mov     ax,cs
        mov     ds,ax
        mov     es,ax

                mov     eax,011111111h  ; Output test value to LCD's
                out     80h,eax
                mov     ecx,3FFFFh      ; delay a bit
                loopd   $


                retf
code            ends
                 end
```

Compiling the Extension ROM

1. Compile the Extension ROM under DOS using the MASM (MicroSoft Version 6.1d) compiler. Type the following command in the DOS directory that contains the ext01.asm file created from the sorce code sample found on page 3:

   ```
   ML /Zm /AT ext01.asm
   ```

2. Use the utility ROMEXT.EXE to calculate and build the checksum byte.

   Note: Download the ROMEXT.exe utility from the ZF Micro Devices' web site. See the "*Using BIOS ExtensionROM.zip*" file: *http://www.zfmicro.com.*

   Type this command:

   ```
   ROMEXT ext01.com
   ```

   The output file created using this utility is always named *ROMEXT.ROM*.

3. Rename the ROMEXT.ROM file to EXT01.ROM

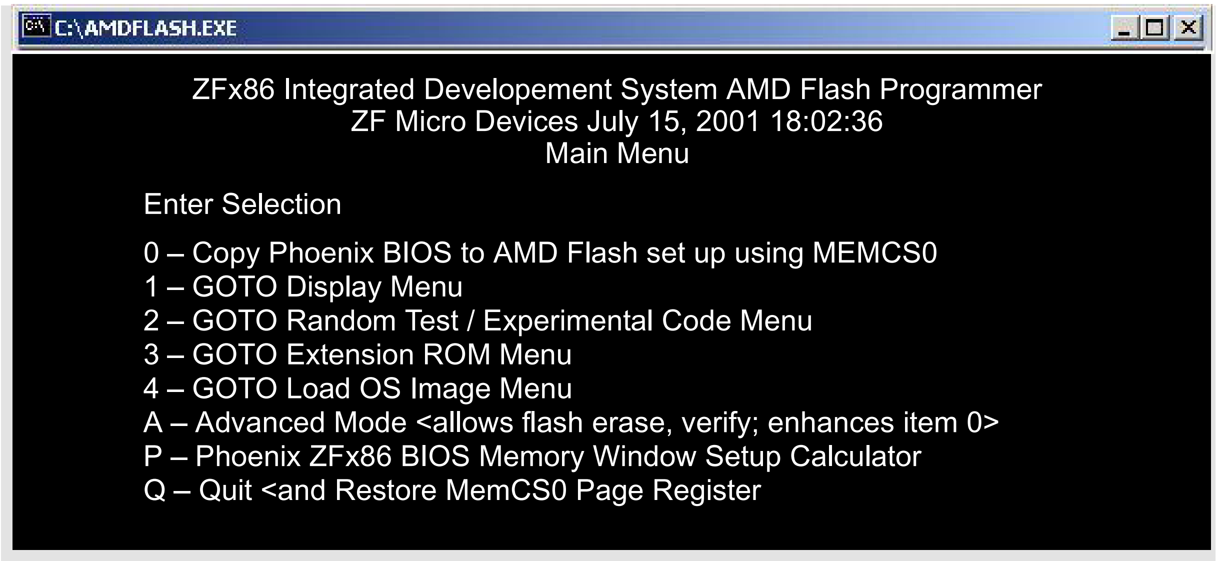
Installing the Extension ROM Using AMDFLASH.EXE

Use the AMDFLASH.EXE utility to place the file EXT01.ROM into the AMD Flash soldered down on the IDS. In this procedure, you place the Extension ROM in the bottom of the Flash using a sequence of AMDFLASH.EXE commands.

1. Move to the directory that contains the AMDFLASH.EXE and launch it by typing:

   ```
   AMDFLASH
   ```

   The AMDFLASH.EXE utility displays. See Figure 2.



```
      ZFx86 Integrated Developement System AMD Flash Programmer
                 ZF Micro Devices July 15, 2001 18:02:36
                              Main Menu

Enter Selection

0 – Copy Phoenix BIOS to AMD Flash set up using MEMCS0
1 – GOTO Display Menu
2 – GOTO Random Test / Experimental Code Menu
3 – GOTO Extension ROM Menu
4 – GOTO Load OS Image Menu
A – Advanced Mode <allows flash erase, verify; enhances item 0>
P – Phoenix ZFx86 BIOS Memory Window Setup Calculator
Q – Quit <and Restore MemCS0 Page Register
```

Figure 2. AMDFLASH.EXE Main Menu

2. Select item A – **Advanced Mode** <allows flash erase, verify; enhances item 0>

   The AMDFLASH Main Menu screen updates.

3. Select item **8** – Erase AMD Sectors

4. If you wish to erase a specific memory sector, enter that memory sector value, or press **Enter** (CR = Carriage Return) to erase all memory on the Flash chip.

5. At the "**Enter Sector to Clear (1C000, c000 as an example) or AAA to quit**" prompt, type **0** and press **Enter**.

6. Type **AAAA** to quit the Erase AMD Sectors selection.

Writing The EXT01.ROM To The Flash Chip

1. From the AMDFLASH Main Menu, select item **3** – GOTO eXtension ROM Menu

2. Select item **1** – Setup to Copy eXtension ROMs into FLASH

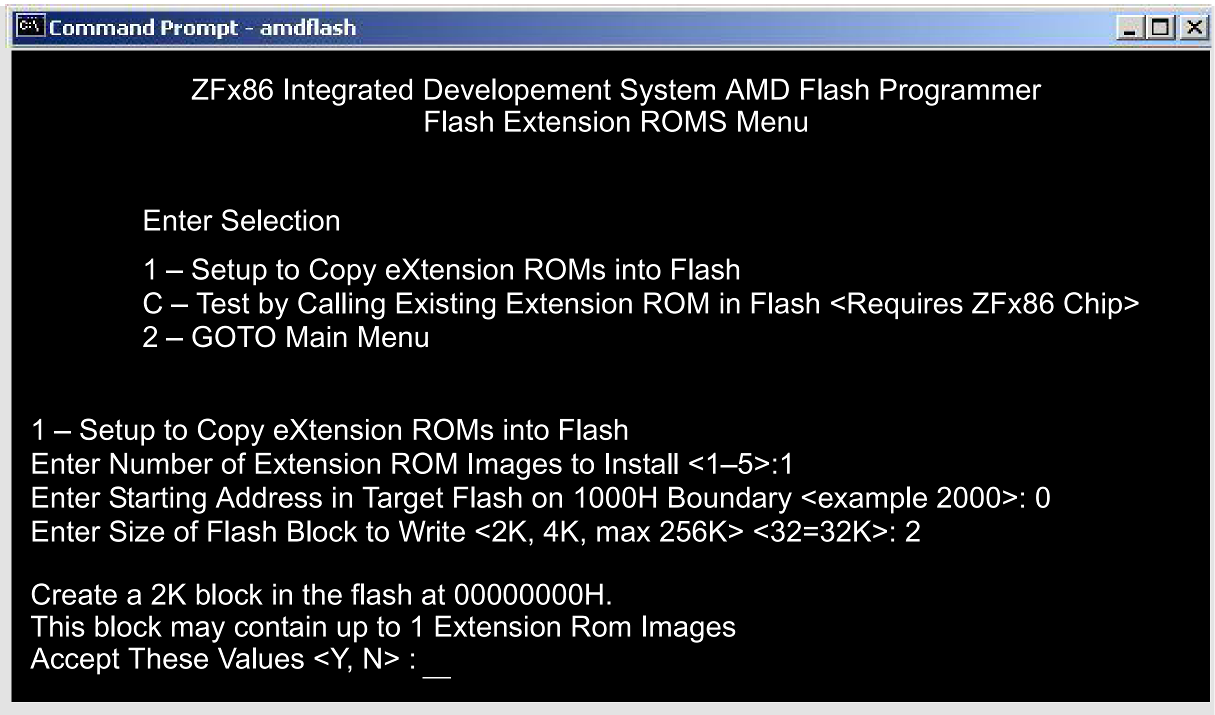
   The AMDFLASH screen updates with "Enter Number of eXtension ROMs to Install <1–5>

3. Type **1**, and press **Enter**.

4. In the "Enter Starting Address in Target Flash on 1000H boundry <example 2000>: type **0** and press **Enter**.

5. In the "Enter Size of Flash Block to Write <2K, 4K, Max 256K> <32=32K>: type **2** and press **Enter**.

   The AMDFLASH screen updates and should appear as in Figure 3.



```
ZFx86 Integrated Developement System AMD Flash Programmer
Flash Extension ROMS Menu


Enter Selection

1 – Setup to Copy eXtension ROMs into Flash
C – Test by Calling Existing Extension ROM in Flash <Requires ZFx86 Chip>
2 – GOTO Main Menu


1 – Setup to Copy eXtension ROMs into Flash
Enter Number of Extension ROM Images to Install <1–5>:1
Enter Starting Address in Target Flash on 1000H Boundary <example 2000>: 0
Enter Size of Flash Block to Write <2K, 4K, max 256K> <32=32K>: 2

Create a 2K block in the flash at 00000000H.
This block may contain up to 1 Extension Rom Images
Accept These Values <Y, N> : __
```

Figure 3. Setup Extension ROM Into Flash Menu

6. In the “Accept These Values” prompt, type **Y** for yes.
7. The AMDFLASH menu updates and requests that you “Press any key to continue”. Do so.
8. When the menu displays the “2 – Add First Extension ROM to RAM Flash Buffer” menu item, type **2**.
9. At the “Enter the Name of the Extension ROM File, such as VIDEO.ROM: type **EXT01.ROM** and press **Enter**.
10. The AMDFLASH menu updates and requests that you “Press any key to continue”. Do so.
11. When the menu displays the “3 – Transfer eXtension ROMS to Physical Flash” menu item, type **3**.
12. The AMDFLASH menu updates and requests that you “Press any key to continue”. Do so.
13. Type **Q** to exit the Flash Extension ROMS Menu.
14. Type **Q** to exit the AMDFLASH Main Menu.

Setting The Memory Window Chip Select Values

Once you write the EXT01.ROM binary image to the Flash, set the Memory Window Chip Select using the BIOS Configuration menu. Follow this procedure:

1. Press the Reset button on the target system.
2. After the splash screen displays and after the keyboard LEDs flash, press the [F2] key to enter the BIOS Setup Utility.
3. In the BIOS Setup Utility, select the **Advanced > ISA Memory Window Chip Select Setup** menu item.
4. Set Memory Window – mem_cs0 to the following values:
    - Window Size: 1
    - Window Base: 0D0
    - Window Page: F30
5. Exit the BIOS Setup Untility, and save the mem_cs0 settings.

Testing the Extension ROM

1. Reboot the target system, and monitor the IDS POST Code LED display.
2. After the normal POST codes display on the LEDs, you should see 11111111 for a few seconds.
3. Then the system boot continues normally.