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

Using ZFx86 FailSafe™ Technology To Implement Unattended Remote Flash Recovery

Simple, reliable, low-cost, and automatic are terms that sound too good to be true, but they describe ZF Micro Devices' solution for embedded applications that lose their critical firmware.

The ZFx86 contains FailSafe logic that can activate custom written pre-BIOS functions using its built-in extensible BUR (Boot Up ROM) and Z-tag™ Interface. This permits an embedded application to exploit designs that automate fault detection and recovery with minimum hardware components. Automatic recovery means unattended recovery which contributes significantly to a particular design's robustness.

The ZFx86 provides on-chip logic that enables customer applications to exploit robustness such as more effective fault tolerance at a lower BOM than our competition. Fewer components contribute to a smaller BOM cost as well as increased reliability. See Figure 1 for the ZFx86's functional block diagram.

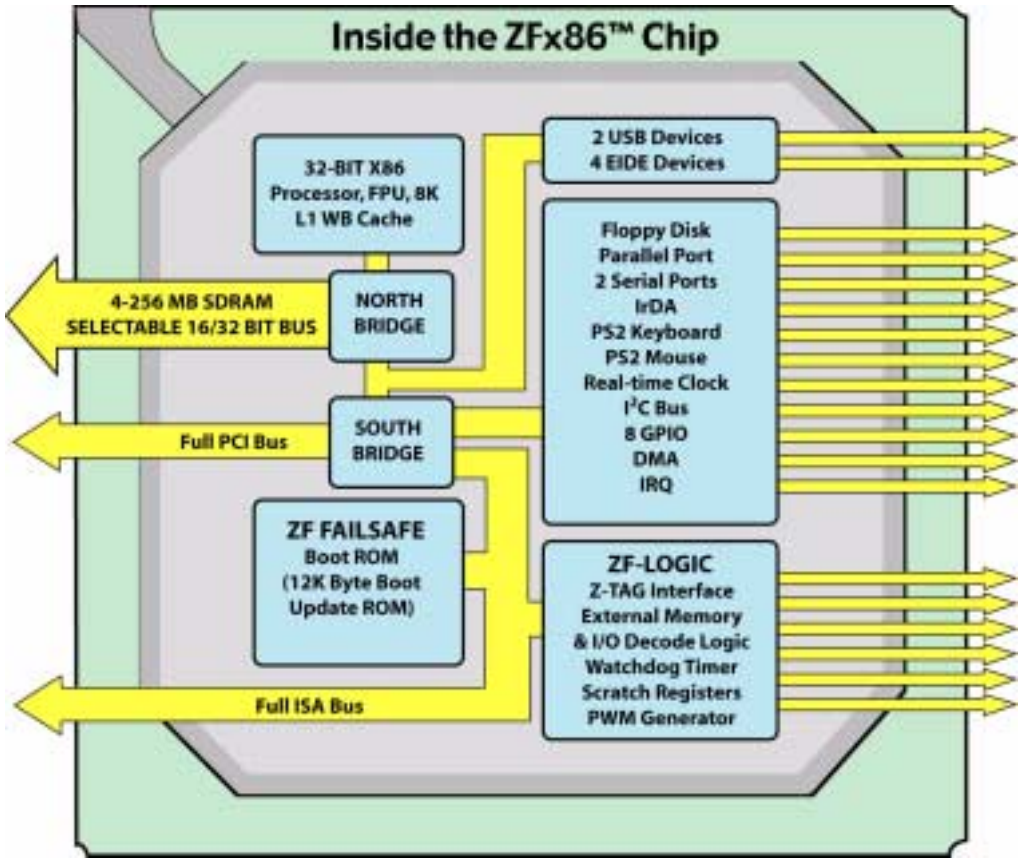


Figure 1. Inside the ZFx86

This list shows the ZFx86 internal components that support FailSafe:

- BUR (Boot Up ROM)
- Scratch and Bootstrap Registers
- Dual Stage Watchdog Timer
- I/O and Memory Chip Selects
  - IO_CS
  - MEM_CS
- Z-tag Interface
- PWM (Pulse Width Modulation) Generator

See Figure 2 for the ZFx86's FailSafe functional block diagram.

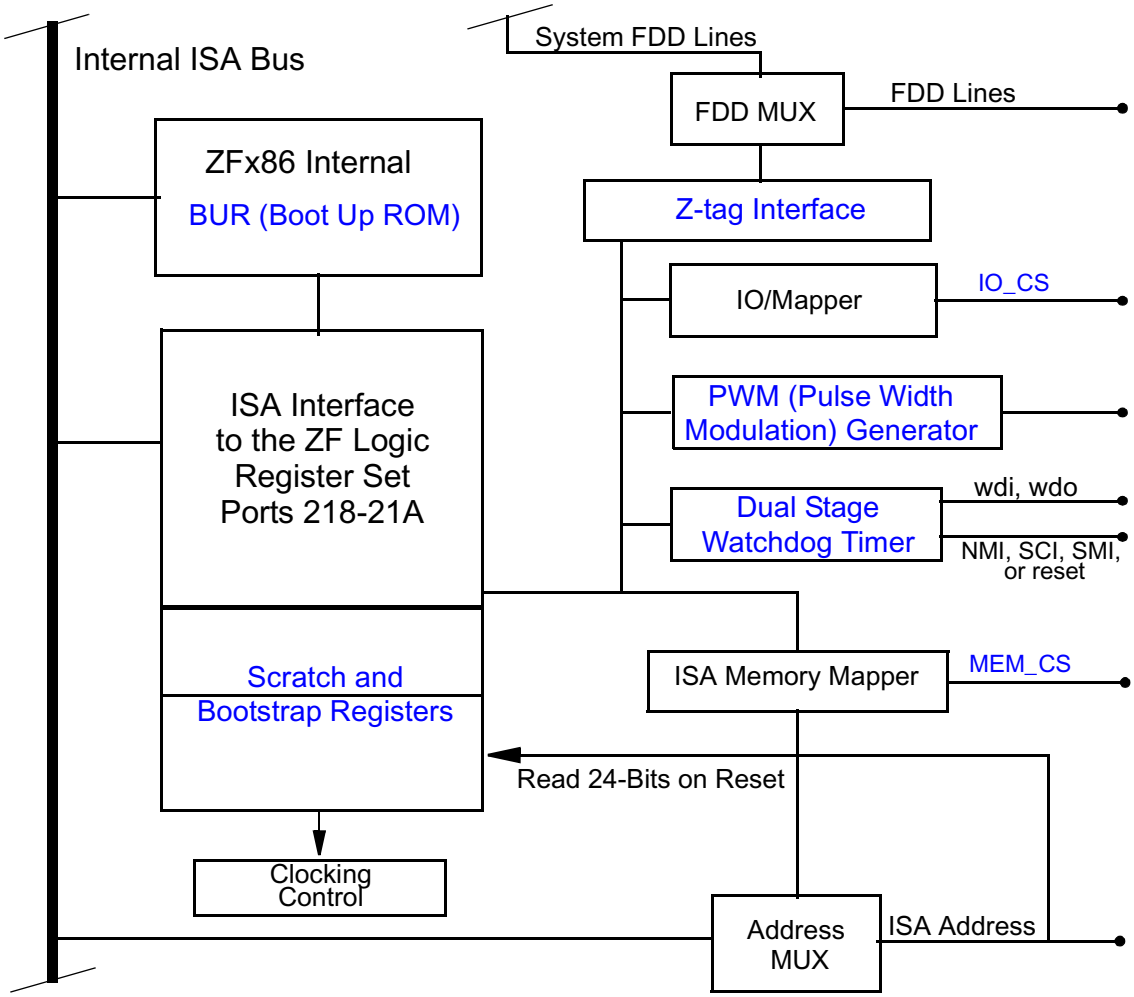


Figure 2. ZF FailSafe Logic Block Diagram

Collectively, these functions provide the power to implement the “ZF FailSafe” technology. ZF FailSafe gives the embedded designer the ability to add simple schemes such as Flash corruption detection and recovery, or much more sophisticated fault tolerant techniques, to their applications.

For example, many PC BIOS manufacturers offer on-board Flash firmware updates via the internet. These updates contain warnings that say, "If this procedure fails during data transfer, your system may become inoperable. You may need to obtain pre-programmed Flash directly from the factory…", or something similar. That's because the Flash update-software runs on the target system, and if corruption occurs during the data transfer or the Flash update, a system reboot is impossible. This corruption can be caused by a poor or noisy network/phone line connection, a power disruption, or some other transmission problem.

BUR uses a special internal RAM for extensible code and read/write data. Therefore, external memory is not required.

Because, BUR and BUR-resources such as FailSafe static-RAM are chip-resident, BUR is always available on the target system. It cannot be corrupted by an executing application, or during a brownout or noisy power conditions, also it cannot be corrupted during faulty data transfer over a network. Unexpected bugs or malicious code in operating systems, device drivers, or application software cannot overwrite BUR, because it is implemented as ROM inside the ZFx86 chip.

Enabling application bootstrap strap option (SA23) ensures that BUR always boots first during either a power-on reset or a warm reset.

Sample IDS Implementation

A sample demonstration, included here, shows how to implement the ZF FailSafe technology using our Integrated Development System (IDS) board. This board contains a soldered down 2M x 8-Bit AMD Flash Memory (Am29F016) chip. The sample implementation places the Phoenix based ZFx86-system BIOS into the Flash’s top 256K memory bytes.

Use Switches S2 and S3, on the IDS board, to access the ZF Logic bootstrap options. Enabling bootstrap SA23 causes the ZFx86 internal BUR to boot first during a cold or warm reset. The SA23 bootstrap selection is found on Switch S3, Key 8. See Figure 3.

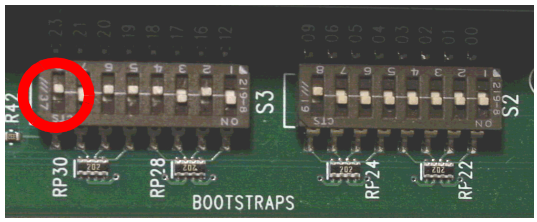
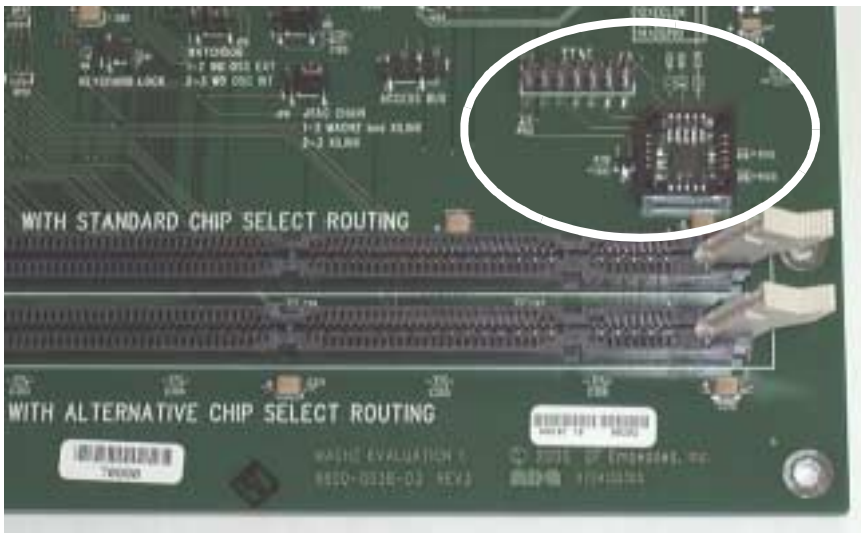


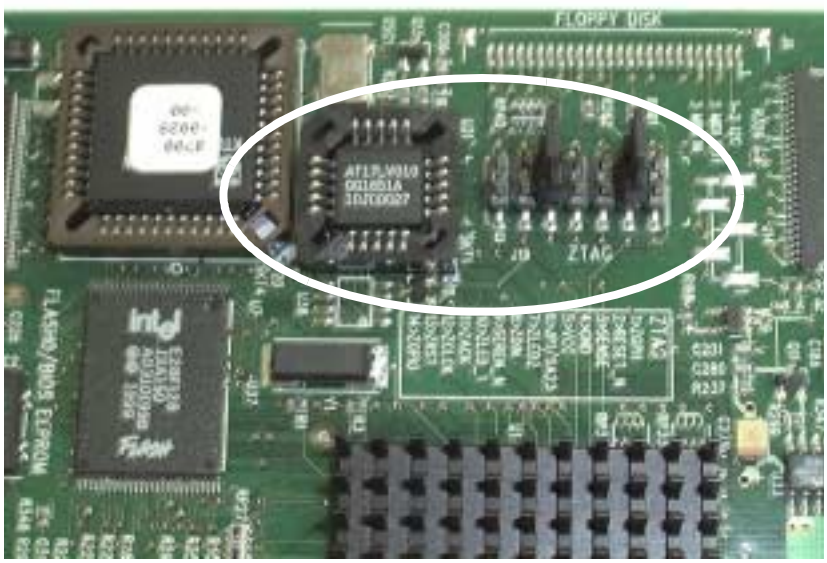
Figure 3. Bootstrap SA23 Switch Location

Install two jumpers across pins 5-6 and 11-12 on the Z-tag interface header J19. This causes BUR to execute a custom BUR Extension code that resides in an external SEEPROM located on-board the IDS (Figure 4) or the zPortPC (Figure 5).



Z-Tag Header and SEEPROM Chip

Figure 4. IDS' Z-tag Interface And The ATmel Chip Socket



Z-Tag Header and SEEPROM Chip

Figure 5. zPortPC's Z-tag Interface And The ATmel Chip

As an option, a Z-tag “Memory” dongle containing SEEPROMs may be plugged into the Z-tag header, and the BIOS downloaded to the IDS board. We call this the “Memory dongle” procedure. See Figure 6.

Figure 6. “Memory” Dongle Connected To The IDS Board

In the sample program, we created a BUR Extension (BUREXT01.COM) that performs a checksum on the embedded BIOS binary image located at the top of the AMD Flash memory.

If the checksum fails, indicating a corrupted Flash image, the extension "pings" the ZFx86’s internal COM1 Serial port. In response to the ping, a remote terminal may transmit a new Flash image directly to the IDS’ Serial port.

Automatic Flash Recovery Demonstration

The remainder of this document demonstrates how one may automatically recover from a system BIOS corruption. In this demonstration, we use the “Dongle” method mentioned previously to implement the FailSafe code. The following items are required or supplied with this document set:

- AMDFLASH.EXE utility (P/N 9272-0106-01) used to erase the Flash device on the development system board.
- BUREXT01.ROM – the BUR extension that performs a consistency check on the Flash and downloads the new Flash code during the recovery process. Run this program on any ZFx86 based design that incorporates the Z-tag interface, including the IDS or the zPortPC reference designs.
- BUREXT01.BIN – the Z-tag manager usable format of the BUREXT01.ROM file. This is loaded into the dongle by the Z-tag Manager utility.

- ZFx10600.ROM – ZFx86 BIOS release version 1.06
- LoadBIOS.com – Host resident software that performs the download using Ymodem protocol.
- Reset.ROM –
- RUN.bat – host batch file used to enable all of the host resident software used in the recovery process.
- ZFx86trm – terminal emulation program (executed by the RUN.BAT file)
- ZF Dongle-01 – the ZFx86 “Memory” dongle containing one or two SEEPROMs as pictured in Figure 6. (Part Number 9410-0021-01)
- PC Notebook or similar PC
- COM1-to-COM1 **null modem** cable

Note: To simplify this sample demo, we use a notebook PC (acting as a “tiny” server) to deliver the new BIOS image via the COM1 Serial output lines. However, you could establish the connection using a MODEM and POTS (Plain Old Telephone System).

Flash Recovery Procedure

Follow these steps to complete the Flash recovery demonstration using the ZF’s memory Dongle-01:

1. Copy the demo files found in the ZFx86 Flash Recovery Demo.zip file to your PC Notebook or similar PC. We recommend unzipping all items to a single folder.
2. Use the Memory Dongle-01 on the target system and boot the target system using this procedure:
   a. Use the Z-tag Manager utility, and load the BUREXT01.ROM file into the memory Dongle-1.

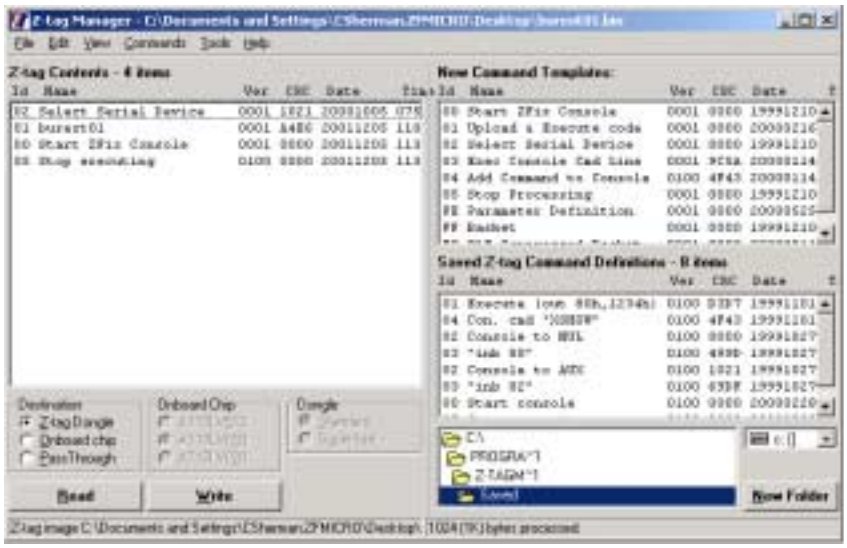


Figure 7. BUREXT01.BIN Loaded Into Z-tag Manager

b. Attach the memory Dongle-1 to the target system's Z-tag header.

3. "Corrupt" the BIOS by running AMDFLASH.EXE on the target system.
   a. Launch the AMDFLASH.exe on the host system by typing **amdflash**
   b. From the AMDFLASH Main menu, select **A** – Advanced Mode
   c. In the Advanced Mode menu, select **8** – Erase AMD Sectors
   d. To erase All Sectors, type **A**
   e. The AMDFLASH utility erases all sectors. The screen displays the erasure process.
   f. Exit the AMDFLASH utility by entering **Q**
4. Verify that the BIOS is corrupted by attempting a warm or cold reset of the target system.

   The target system does not respond in any way.
5. Connect the PC notebook's Serial Port to the target system's Serial Port using a null modem (COM1-to-COM1) cable. See the table below for the cable's wiring specifications.

IDS System			Target System		
Signal	**DB-9**		**DB-9**	**DB-25**	**Signal**
DCD	1		1	8	DCD
RXD	2		2	3	RXD
TXD	3		3	2	TXD
DTR	4		4	20	DTR
GND	5		5	1,7	GND
DSR	6		6	6	DSR
RTS	7		7	4	RTS
CTS	8		8	5	CTS
RI	9		9		RI

6. On the host, execute RUN.BAT and follow the on-screen instructions. The RUN.BAT launches a terminal transfer program in a DOS window.
7. Reset the target system. This causes BUR to execute BUREXT01.ROM in the Dongle-1. BUREXT01.ROM checks the Flash BIOS image, and if intact, branches to it; otherwise, it "pings" COM1 waiting for a new Flash image as discussed below.
8. The host system transmits the ZFx10600.ROM image into the target system.
9. Upon completing the download (when the counter reaches completion), the system reboots automatically. Observe the reboot on the POST code LED's on the IDS board.

The BUREXT01.ROM, loaded into the dongle, executes through the on-chip BUR by resetting the target system. It's in effect "phoning home" requesting a new BIOS. The host system reacts to the "phone call" by transmitting Ymodem packets that contain a fresh BIOS.

Notice in the transmission, when the final packet is transmitted to the target, it automatically reboots using the new BIOS. The problem is resolved, unattended.