



For Decision Makers in the Embedded and
Real-Time, Open Systems Computer Industry

PFU Systems

**Fast, Connected
Systems**

**STAY
SECURE**

Tech Focus

Embedded PCs

Big Picture Issues Drive
Embedded BIOS Choice

Big Picture Issues Drive Embedded BIOS Choice

While often overlooked, an embedded PC's BIOS has a huge influence in an application's overall cost and reliability.

by Kishan Jainandunsing, PFU Systems

The BIOS ranks as perhaps the most undervalued part of a PC/AT system. More attention tends to be focused on the operating system, application software and CPU performance. Such thinking is understandable in commercial PC systems, where the BIOS is ruled by the "WinTel" PC platform specifications and is relegated to an invisible commodity. However in embedded applications the BIOS is not predetermined solely by those platform specifications. Rather it's often a subset with several extensions to assure reliability, availability and serviceability, or "RAS".

The BIOS plays a fundamental role in providing RAS capabilities to embedded systems. Yet RAS functionality isn't something that's typically available as a standard product through BIOS and single board computer vendors. As such, systems designers must do the BIOS themselves or else get a custom BIOS from the BIOS or SBC vendor. Either way project managers can face a situation where BIOS licensing and development eat up an estimated 40% to 60% of total product development cost. That figure increases significantly when product updates and upgrades are taken into consideration. With that in mind, it's clear that the total-cost-of-ownership (TCO) over a product's or product line's lifetime is of equal or greater importance than the initial product development cost.

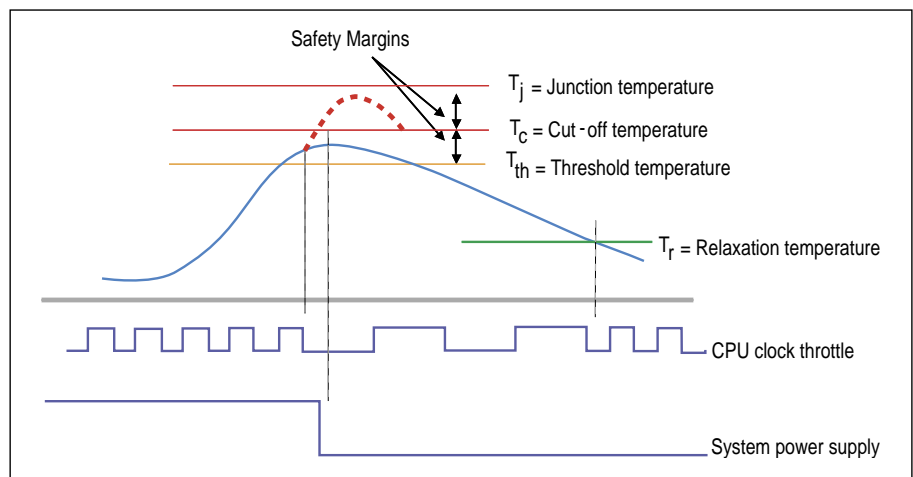


Figure 1 In embedded systems that use high-performance x86 processors, the BIOS can play a role in preventing CPUs from overheating as a result of fan failure, airflow blockage or ambient temperature rising beyond specifications. CPU temperature is monitored, reading a thermal diode via an SMBus, and the CPU clock is throttled to regulate the temperature, or ultimately shut down the system's power supply.

Basic BIOS Duties

To appreciate the larger scope of embedded BIOS issues, it's important to understand the basic duties of a BIOS. The BIOS, which stands for Basic Input, Output System, traditionally performs the following functions listed in order of activity: It initializes the chipset; tests and initializes system memory and peripheral device functionality; and boots an operating system. When an x86 system is powered up, the CPU executes the first instruction of the BIOS program.

Subsequently the BIOS will initialize chipset registers to enable timers, DMA controllers, etc.

Once a chipset is properly initialized, the BIOS continues to initialize and test other functions of the system, such as serial ports, parallel ports, IDE controllers, floppy disk drive controllers, L2 cache and system memory. It looks for extension ROM code to execute and further initialize peripheral device controllers, such as VGA, SCSI and so forth. Finally, the BIOS loads the boot-loader, a tiny program, from

the boot sector on the media that contains the operating system. At this point the BIOS passes control over to the boot-loader, which then executes and bootstraps the OS to execute further in system memory.

A memory area, typically called the “CMOS”, holds real-time clock date and time information, and BIOS default settings for drive configurations, serial and parallel port resource settings, and other peripherals. Default settings can be changed to allow flexible system configurations, without recompiling the BIOS program. CMOS memory is volatile and is typically backed up with a battery so that date and clock information and modifications to the CMOS remain intact after a system power-down.

System Tasks Enter the Game

Over the years designers have expanded the role of the BIOS support functions, such as system and power management, enhancing the manageability of a PC/AT system. The system management (SM) BIOS provides system configuration and status information, which other BIOS functions, the OS, applications and utilities may act upon. Examples of static system configuration information are memory type and size, and CPU type and speed. Examples of dynamically recorded information are CPU temperature and ECC memory bit error events.

Power management support was first added through the Advanced Power Management (APM) specification. The APM spec was later supplanted by the Advanced Configuration and Power Interface (ACPI) specification. Unlike APM, ACPI support in the BIOS gives the operating system the ability to orderly and selectively reduce power consumption in the system.

ACPI has a well-defined model for the CPU, chipset and PCI I/O chip power management configuration and configuration tables. It also defines motherboard device enumeration and configuration, and provides APIs for software to interface with ACPI. The role of an ACPI-enabled BIOS is to register all ACPI-capable devices with their respective configuration and power management data, for use by an ACPI-enabled OS, or by applications that are specifically written to talk to the ACPI set of APIs.

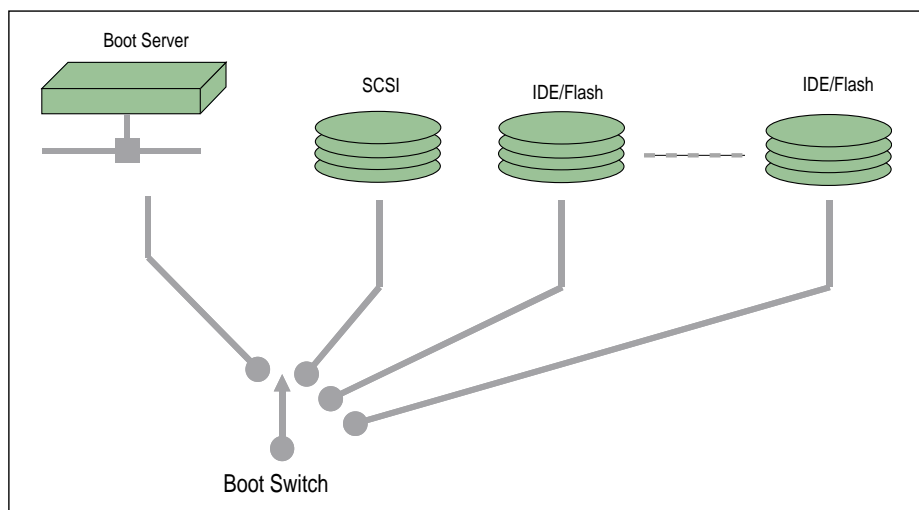


Figure 2 In embedded systems the OS boot media are vulnerable to mechanical or electrical failure and corruption. The BIOS can be geared to autonomously recover from a boot failure through a time-out mechanism, and cycle through the alternatives, such as hard disk drives, flash disks or a network-boot device.

Extensions for Embedded Apps

The BIOS in an embedded computer requires more than just system management capabilities. More often than not, embedded computers play a mission-critical role in their applications. As such, system reliability is an important requirement, extending beyond that which is accepted in the consumer and corporate computing space. OEM designers continually face the challenge of adding RAS capabilities to embedded systems that incorporate PC/AT host subsystems.

There are many areas that may compromise the reliability of an embedded system. Some of those are beyond the control of the BIOS. Among those areas that are within the BIOS’s realm are the CPU, system memory, OS boot media, CMOS area and system BIOS area. Each of these areas requires special safeguards against failure.

In embedded systems that use high-performance x86 processors, the CPU is vulnerable to overheating as a result of fan failure, airflow blockage or ambient temperature rise beyond specifications. The BIOS may play a role in safeguarding against any of those potential hazards. The CPU temperature is monitored, reading a thermal diode via an SMBus, and the CPU clock is throttled to regulate the temperature, or ultimately shut down the system’s power supply to prevent irreversible thermal damage (Figure 1).

Embedded systems that use large amounts of memory and high-density memory chips are vulnerable due to bit errors as a result of alpha and gamma particles. There is not much the BIOS can do here, except provide event log support for ECC memory. One solution is to deploy a memory controller with memory scrubbing and ChipKill Correct hardware support. Those features must be enabled in the BIOS. The memory controller may also support partitioning memory in hot standby and operational banks. In this case the BIOS is also required to enable this function.

ChipKill Correct and memory scrubbing are high-availability techniques that were developed by IBM for mainframes. ChipKill Correct detects dual-bit errors on the same memory access. Memory scrubbing detects and corrects bit errors during cycles in which the memory is not accessed.

Vulnerable OS Boot Area

In any embedded system the OS boot media are vulnerable to mechanical or electrical failure and corruption. A defect usually results in a “replacement-configuration-testing” cycle that can be time consuming and costly. The BIOS may implement strategies to autonomously recover from a boot failure through a time-out mechanism, and cycle through multiple alternatives, which may include hard disk drives, flash disks or even a network-boot device (Figure 2).

The battery used to back-up CMOS data is often the weakest link. Losing CMOS data means that the BIOS, at power-on, may cause the system to halt as it finds a corrupt CMOS area, requiring manual intervention to reload defaults to the CMOS memory. Embedded system designs typically avoid use of a CMOS backup battery and rely instead on the system power supply, since there is seldom a need to change default-operating conditions. Clock synchronization is handled differently in such cases. It is imperative that the BIOS handles recovery from a power failure without manual intervention to resolve CMOS checksum errors.

The system BIOS area may sustain corruption as well. In this case there is no simple manual intervention problem to solve. But rather more involved measures are required to enable the BIOS to recover autonomously from corruption. The BIOS may implement strategies to this effect, using alternative paths to continue the POST process. These alternatives may require additional NVRAM.

Cost of Ownership a Priority

Under pressure to lower unit cost, OEMs often overlook the total-cost-of-ownership of embedded products. It's easy to miss the fact that using off-the-shelf single board computers or subsystems without RAS functionality in the BIOS can increase development budgets, risks, and time-to-market. On top of the RAS functions further customization may be required, introducing further cost, risk and delay. When development costs can't be amortized over the entire product's lifecycle, the end-product unit cost is significantly increased. Costs drive up

even further when the run rate of the product is lower than expected and when chipset changes are required when the chipset is discontinued. Either situation can be a nightmare to embedded equipment OEMs.

To avoid those nightmares, it's best to leave BIOS issues in the hands of an SBC vendor that's savvy in such matters. OEM designers should invest the time to research a vendor's products, considering BIOS development and total-cost-of-ownership. Such time spent is well worth the investment when measured against the alternative of purchasing a BIOS license with chipset adaptation code and associated development costs. Those purchases can run upward of \$500,000 and can result in significant slips in schedule.

At first glance it may seem that requesting relatively minor BIOS customization from the board or subsystem vendor is more costly than in-house development. But a make-or-buy analysis quickly reveals that the time and expense of acquiring the BIOS license with chipset adaptation code, and associated development, favors a buy decision.

Leverage BIOS Costs Across Platforms

Selecting a board or subsystem vendor with off-the-shelf scalable products built on a common platform is equally important. OEM designers are then able to leverage off-the-shelf functionality and customizations across chipsets to guarantee long product life and amortization of development costs. This subtle aspect of a product is often overlooked in favor of unit cost and other factors.

For its part, PFU Systems designs its SOM products around fully scalable architectures, which strongly favor single chipset platforms. Where chipset changes are necessary, BIOS differences between chipsets are shielded as much as possible from the OEM. PFU Systems' SOM products are designed with off-the-shelf BIOS features tailored to the needs of embedded applications, inclusive of power management and high availability as described earlier. The company also provides BIOS customization services for its entire line of products.

Clearly the BIOS in an embedded x86 system plays a significant role in enhancing the reliability. Safeguarding the system from damage and autonomous recovery from a failure are important characteristics of an embedded system, and can be realized cost-effectively through a well-crafted BIOS. Such safeguards are not available in BIOS products designed for consumer and corporate PC markets, and a design from scratch is a challenging and costly undertaking.

Meanwhile chipset changes cause recurrent development costs, and drive up total-cost-of-ownership and ultimately, unit cost. OEM designers need to be aware of the risks taken when the BIOS aspects of a solution are not factored into the decision making process and when cost is an overriding factor. OEM designers must carefully select their embedded hardware suppliers, paying attention to off-the-shelf embedded BIOS features, product lines with scalable architectures and the ability to provide BIOS customizations. ■



Headquarters

PFU Systems, Inc.
3350 Scott Blvd., Bldg. 34-B
Santa Clara, CA 95054-3105
Phone: 408.327.1750
Fax: 408.327.1751

Midwest Sales Office

PFU Systems, Inc.
5929 Baker Road, Suite 440
Minnetonka, MN 55345-5956
Phone: 952.345.0085
Fax: 952.345.0090

Eastern and Int'l Sales Office

PFU Systems, Inc.
206 New Edition Court
Cary, NC 27511-4451
Phone: 919.319.0900
Fax: 919.319.3552

www.PFUsystems.com