Microsemi Fusion™ FPGAs Supporting Intelligent Peripheral Management Interface (IPMI) Applications

Introduction

The IPMI specification includes two elements: 1) a server management protocol and 2) an architectural specification for system management, primarily for server applications. It provides three levels of integration: v1, v1.5, and v2.0. Each level provides more complex interfaces and alerting capabilities. IPMI is designed to be modular so that each level does not carry undue cost burden and drives intelligence to the appropriate level. An example of this intelligence is using the system's host processor to perform a task, instead of a microcontroller, if it can be done more flexibly and economically with the processor. The IPMI protocol and interfaces allow system administrators to interrogate a server for potential issues while the system or individual baseboard is up or down. IPMI does not require application for the baseboard management controller (BMC) to provide data for diagnosing system hardware issues or readiness.

IPMI has been designed to integrate new management levels, extending functionality without requiring that existing capabilities be reimplemented or redesigned. This extends to design for "economic incorporation" of changes in the population and to the implementation of the baseboard and remote sensors. IPMI is meant to minimize the impact on hardware and software when sensor population or sensor hardware interfaces are changed. This is designing to maximize "self configurability" in system management software (i.e., "Plug and Play"). The OEM implementation of the IPMI architecture should scale from entry through enterprise and data center class server systems. With this in mind, the architecture should be adaptable from single-board and single-chassis through multi-board and multi-chassis systems. An OEM's low-end solution should be a proper functional subset of higher-end solutions.

IPMI provides clean points for OEM extension and integration. The specification reserves command numbers, sensor numbers, and other types of numbers for OEM extension to provide product differentiation. Microsemi Fusion FPGAs provide IPMI, a method of implementation and flexibility that cannot be provided by conventional microcontroller methods. Microsemi Fusion FPGA allows the FPGA firmware and application software to be updated and reconfigured for expanded capabilities as the OEM develops them.

Implementation

Illustrated in Figure 1 on page 2, IPMI defines multiple functions and interfaces, some mandatory and others optional, via the BMC. IPMI requires a few mandatory functions: the IPM device (internal BMC function), a system interface, a system data record (SDR) repository, a watchdog timer, an event receiver, a system event log (SEL) interface, a field replaceable unit (FRU) inventory, an initialization agent, and internal event generation. An optional function is the Intelligent Platform Management Bus (IPMB) interface, but this is typically implemented with at least one connection that must provide a path from the system interface through the BMC to the IPMB connection. This application note focuses on the requirements for a BMC with integrated NMC functions and satellite management controller (SMC) to be compliant with the IPMI specification, v1.5 and above, using the features of Microsemi Fusion FPGAs. This application note will also show the value of the Fusion FPGA and Microsemi DirectCore intellectual property, while leaving room for individual OEM customization, such as sensors, the PCI management bus interface, serial messaging, bridging support, and LAN capabilities.

Looking at IPMI from a logical perspective, you can see in Figure 1 on page 2 that the functional blocks are divided into their respective interfaces or tasks, with many points of commonality.

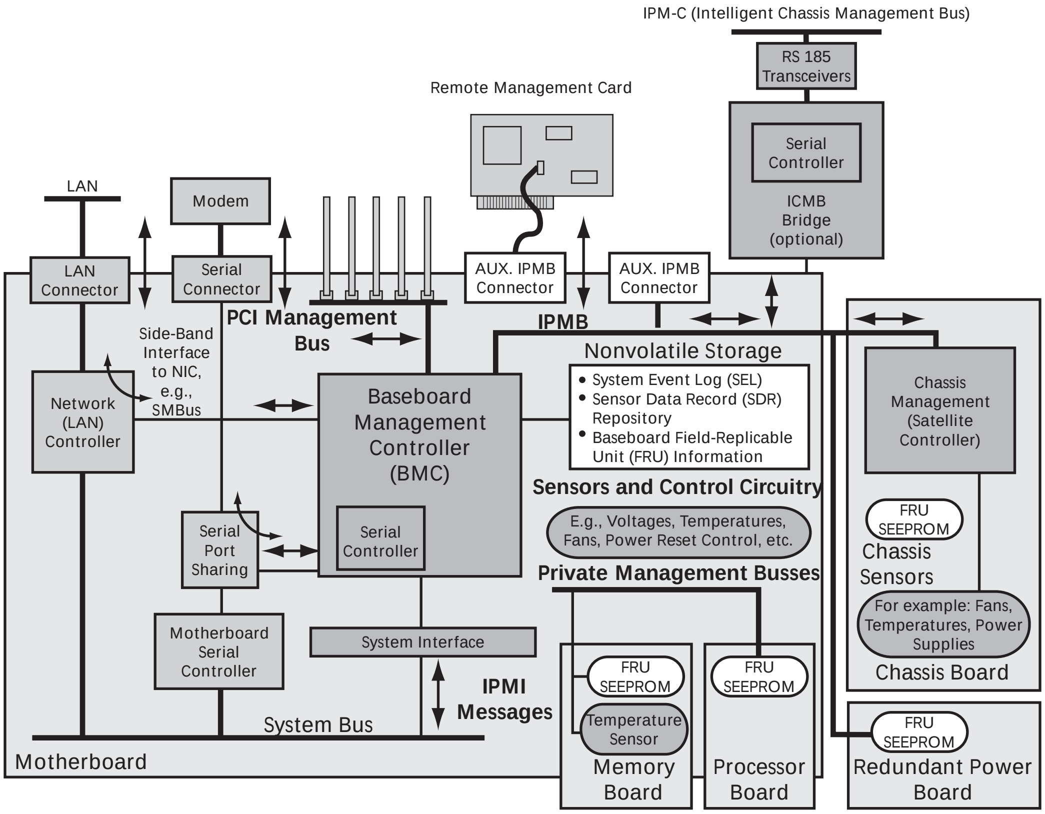

***Figure 1* • IPMI Block Diagram**

In Figure 2, the example IPMI implementation uses three hardware components: in this case two intelligent satellite management platforms and a BMC. In combination with the system software from the primary application's implementation, this makes for a powerful diagnostic system for system administrators. The IPMI protocol and interfaces allow system administrators to interrogate a server for potential issues while the system is up or down, since the application is a prerequisite for the BMC to provide data for diagnosing system readiness and potential issues.

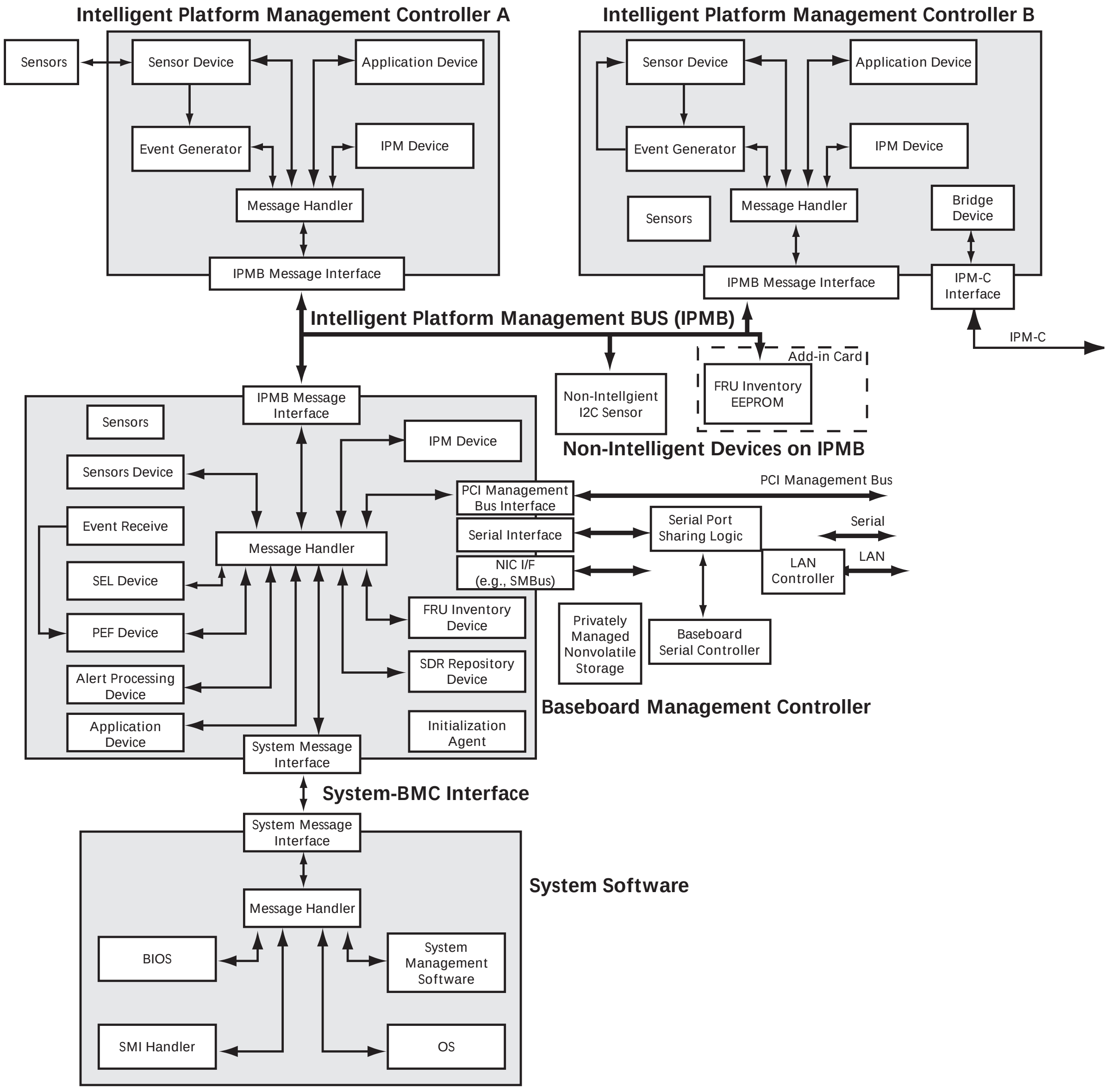

Figure 2 • **Intelligent Platform Management Logical Device**

Baseboard Management Controller (BMC)

The Baseboard Management Controller (BMC) system block is the primary component in the IPMI specification. This system block is also the hardware device for which Microsemi Fusion FPGAs can provide the greatest flexibility and reuse for required functions and optional interfaces without reinventing the hardware or software for each generation of OEM system. Figure 2 on page 3 shows all of the BMC function blocks required for an IPMI implementation. The following is a description of the various blocks:

- IPM Device – This represents the basic intelligent device that responds to the platform sensor and event interface messages. All devices on the IPMB are expected to respond to the mandatory "IPM Device" commands. Other management controllers that communicate via IPMI messages to the overall system are also considered IPM devices.
- Sensor Device – This block provides the command interface to one or more sensors. Sensor devices provide a set of commands for discovering, configuring, and accessing sensors.
- SDR Repository Device – The SDR device is a logical management device that provides the interface to the SDRs for the system. It also provides a set of commands for managing the SEL.
- SEL Device – This is a logical management device that provides the interface to the SEL for the system. The SEL device provides a set of commands for managing the SEL.
- FRU Inventory Device – Provides the interface to a particular module's FRU inventory information. There will typically be one set of FRU inventory information for each major module in the system.
- Event Receiver Device (ERD) – Accepts and acknowledges event request messages (ERMs). The ERD then passes the ERM to the SEL device for logging. The medium is IPMB.
- Event Generator Device (EGD) – Represents the functionality that is used to deliver event messages to the ERD. The EGD includes commands to allow configuration of event message delivery.
- Application Device – Physical instantiation of an IPM device will implement some device-specific functionality that falls outside the standard sensor and event functions.
- PEF Device – Function associated with comparing an event message against a set of selectable "event filters" and generating a selectable action on a match.
- Alerting Processing Device – Functions associated with queuing up and processing alerts, and alert policies that determine to which destinations an alert will be sent.
- Chassis Device – Functions associated with recovery control actions, such as power-on/off, power cycle, reset, diagnostic interrupt, chassis identification indicator, and system boot.
- Message Handler – Functions associated with configuration and operation of message authentication and routing, both internal to the BMC and among the different interfaces to the BMC.
- IPMB can also define the IPM-C "bridge device".

From the above blocks, the BMC primary function initializes the management of the system and monitors and controls traffic from other management controllers/sensors on the IPMB interface to the system software through the system interface. As a higher system-level function, the BMC can take the appropriate action by alerting the system of potential issues or IPMI events. If this capability is not required, the BMC can simply record those events in a nonvolatile memory for extraction, when requested, to either an external system interface, an IPM-C interface, a v2.0 interface, or a higher level management application through the system interface.

Network Management Controller (NMC)

Network management controllers (NMC) are typically industry standard discrete LAN controllers that are dedicated to the BMC. Depending upon the specific IPMI application, however, the LAN controller may be shared for general system use. A typical NMC is connected to the BMC and system bus via separate PCI interfaces, as shown in Figure 1 on page 2. An integrated BMC includes an NMC as a processor peripheral and will be discussed later (also shown in Figure 6 on page 11). The integrated BMC removes the need for a PCI interface, but allows for one if connection of the LAN controller to the system bus is necessary.

Satellite Management Controller (SMC)

Satellite management controllers (SMC) are considerably different when compared to their BMC counterparts. They are smaller, OEM customized, and dedicated to a series of specific common tasks. A typical SMC consists of a microcontroller (usually 8 or 16 bits) sometimes with analog, an IPMB and/or IPM-C interface, memory, and sensors. An SMC that resides in the same chassis as the BMC is connected to the BMC via the IPMB interface, whereas an SMC residing in another chassis or connecting to a BMC in another chassis is interfaced via the IPM-C.

BMC/SMC Messaging

The IPMI specification is extremely flexible concerning the quantity and interfaces allowed, since the communication is based on a set format of IPMI messaging, which uses a request/response protocol. Since the format is defined, IPMI facilitates multi-master operation on busses such as the IPMB and ICMB, allowing messages to be interleaved and allowing multiple management controllers to intercommunicate directly on the bus. The request portion of the protocol is based on a request from the source that is in the form of a command in a set format. These commands are grouped into functional command sets using a field called the network function code. They are functionally grouped to make it easier to organize and manage the assignment and allocation of commands.

All IPMI request messages have the following:

- Network function
- Command
- Optional data
- Completion code field

Since the format is universal, all devices in the IPMI network respond according to their address.

Fusion in IPMI

Microsemi Fusion devices are well suited for these functions based on the ability of Fusion FPGAs to be configured and reconfigured in-system for specific application and sensing requirements. Fusion has many capabilities for sensing, storing, and message handling, but a key component is the scalability and integration factor, which cannot be provided by other solutions using a currently installed device. The following are examples of how Fusion can solve many of the IPMI requirements.

Analog Functions

A BMC/SMC does not necessarily need to be able to measure mixed signals, but with the integrated analog capabilities of Microsemi Fusion FPGAs, measuring system voltages, currents, and temperatures is made much easier. It also allows the designer the flexibility to add more measurement nodes for expansion without the addition of unanticipated components that may be needed in other implementations. Fusion allows the architect to use the Fusion device as a platform block that can be reimplemented in other systems without the worry of missing a required signal. To handle these functions, the Fusion device contains a number of analog input and output pins. These pins are arranged in groups, called Analog Quads. As shown in Figure 3 on page 6, Analog Quads then feed an analog multiplexer (MUX), which in turn connects to an internal sample and hold circuit (S/H) and finally to a 12-bit analog to digital converter (ADC).

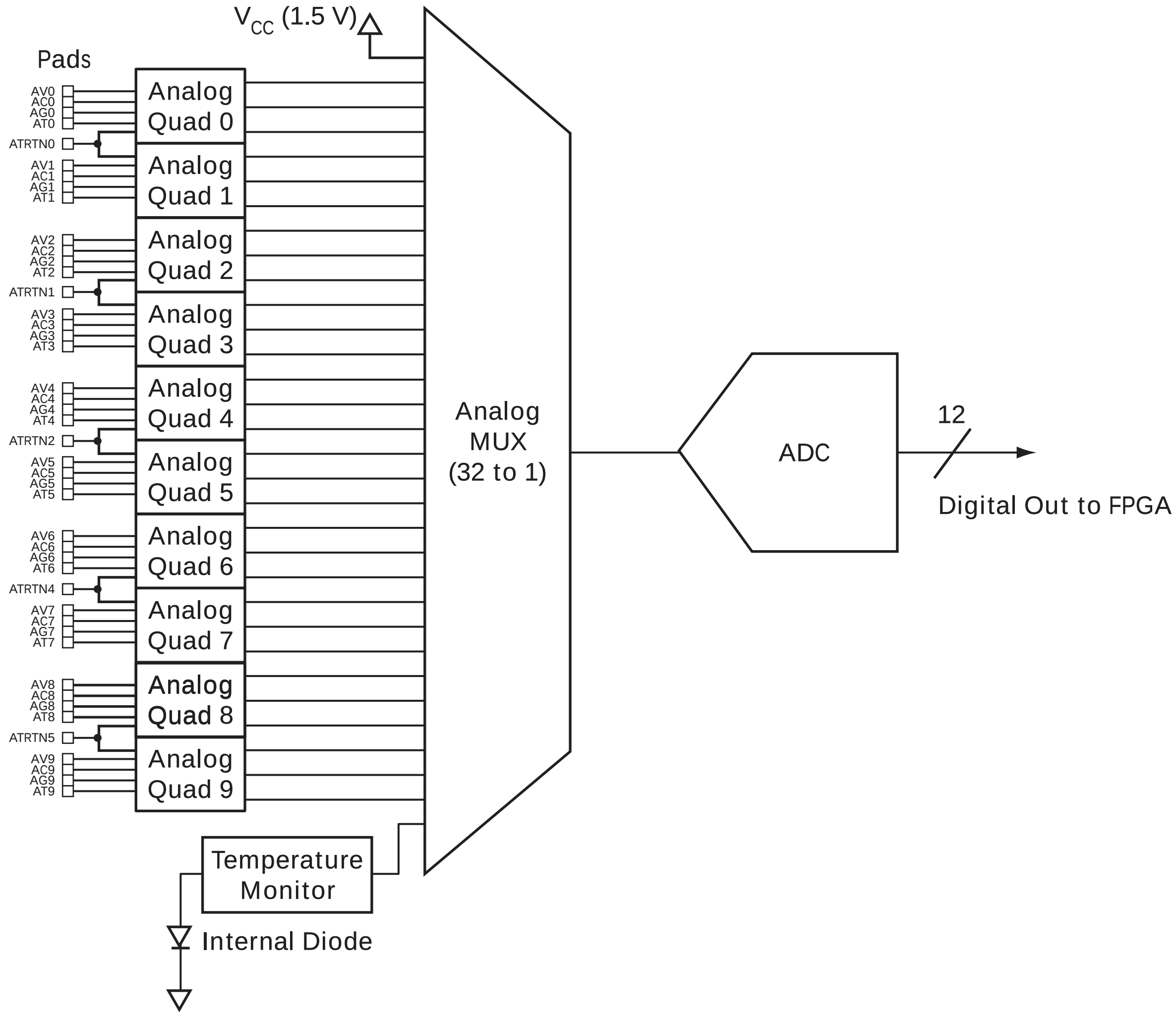

Figure 3 • **Fusion Analog Quad, MUX, S/H, and ADC Structure**

Each Analog Quad can be configured to support voltage, current, and temperature measurements. Any of the three inputs can be configured to provide a single-ended voltage measurement. Alternatively, the AV and AC inputs can be used as a differential pair to measure the voltage drop across an external current monitoring resistor, and AT can be configured to supply a small oscillating current to an external diode to measure ambient temperature.

Voltage Monitor

The maximum voltage tolerance of the analog inputs is ±12 V with respect to circuit ground. Each input has a built-in prescale amplifier that can be configured to provide 8-, 10-, or 12-bit sampling resolution over input voltage ranges from zero to 0.125, 0.250, 0.50, 1.0, 2.0, 4.0, 8.0, or 16.0 V. Additionally, the inputs are capable of inverting negative voltages with respect to ground into positive-sampling signals.

Measuring mixed voltages will be required for most motherboard applications. The ability to scale input voltages eliminates external components and improves precision for voltage measurements. For a typical motherboard application, the designer will be expected to measure and control at least the following voltages: +3.3 VDC, +5 VDC, +12 VDC, –12 VDC and +5 VSB. With specialty components finding their way into applications, the voltage and current monitoring requirements could well exceed 8–10 voltage signals. The 12-bit ADC in Microsemi Fusion FPGAs provides voltage measurements with 4 mV resolution and unadjusted accuracy of 25 mV, well within the limits required for level thresholds concerning alert generation to the application software.

Current Monitor

For current measurements and monitoring, a differential amplifier measures the minute voltage drop across an external current sensing resistor. This amplifier has a fixed gain of 10X, so a voltage drop of 0.256 V corresponds to a 2.56 V input to the sample and hold circuit. Assuming that the internal 2.56 V reference is used, this means that a 0.025 Ω resistor can measure up to 10 A of current with 2 mA resolution and total unadjusted error of 12 mA. Note that for current measurements, AV must be at a greater voltage level than AC, so bidirectional current sensing requires two sense resistors and two Analog Quads, connected as shown in Figure 4.

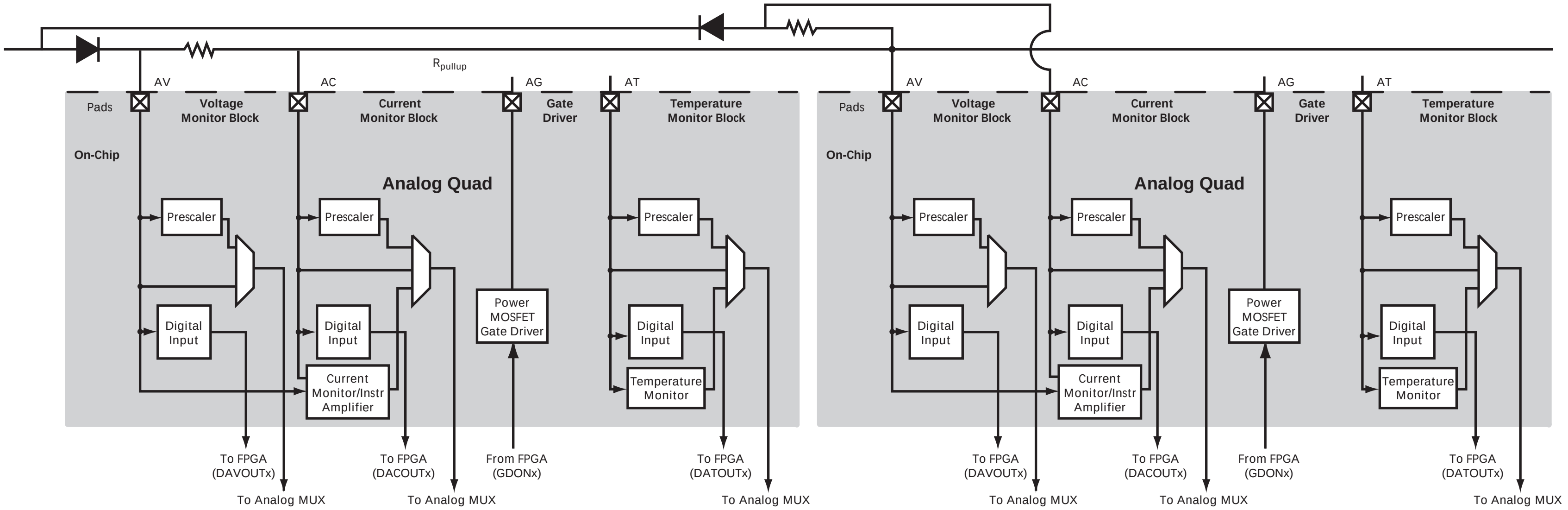

Figure 4 • **Bidirectional Current Sensing Using Two Fusion Analog Quads**

Temperature Monitor

Temperature is easily measured using an inexpensive bipolar silicon transistor connected to the AT input of an Analog Quad. When configured to monitor temperature, the Analog Quad provides two precisely generated strobe currents to the diode connected transistor and then measures the difference between the resulting forward voltages. This method provides automatic temperature measurements accurate to within 2°C.

In addition to the analog inputs, each Analog Quad contains a MOSFET gate driver. These drivers have programmable drive strength and are designed to pull the gate of a power MOSFET to ground. These drivers allow the BMC to control external MOSFET switches and trim inputs of power supplies or fans that are usually needed in end applications where IPMI is implemented.

Unlike most off-the-shelf devices used for IPMI, Fusion also includes two on-board oscillators: an RC oscillator and a crystal oscillator.

The RC oscillator provides a 100 MHz time base accurate to within 1 ppm. This eliminates the need for an external crystal oscillator. This time base is used for several internal functions of the device and can also be used for user logic with clock conditioning through a built-in phase-locked loop (PLL). The PLL can provide a 50% duty cycle, M/N frequency scaling, and programmable phase shifts for up to three internal clocks, based on a single input clock frequency. Coupled through a No-glitch MUX (NGMUX), this unique feature allows you to conserve power in the power-off state by switching to a significantly slower processing clock without external components.

The crystal oscillator can be used as the clock for the real-time counter (RTC) inside the Fusion device. The oscillator and the RTC are powered along with the I/O ring from a 3.3 V power rail and are always active as long as power is applied to the device.

For IPMI applications, the RTC can be used as a watchdog timer for power-on conditions or can be used as a real-time clock for timestamping of sensor readings into the SEL. Additionally, this circuitry includes an activity detector that can be used to wake up the device whenever an external event, such as a remote boot, a power button press, or bus activity, is detected.

Finally, the Fusion analog block contains an on-board voltage regulator to provide the 1.5 V core voltage for the Fusion device. Running the core of the device at 1.5 V cuts active power dissipation by half. This regulator has an internal shutdown mechanism associated with the RTC circuit. The shutdown circuit allows the FPGA logic to initiate "standby mode" by shutting down the regulator. Then the RTC can wake up the device by starting up the regulator when either an external event occurs or the RTC count reaches a programmed value. The on-board regulator also allows the designer to design the system based on a single voltage power rail of a nominal +3.3 V.

Digital Functions

To provide maximum flexibility for Fusion designs, the FPGA logic has complete access and control over the entire analog block of the device. Analog Quad configuration, multiplexer selection, settling time, ADC reference source, ADC clock frequency, and ADC resolution can all be directly controlled via a simple 8-bit interface available to user logic in the FPGA logic gates.

Included in the FPGA fabric for the device are various numbers of programmable logic tiles (gates or registers), synchronous dual-port SRAM blocks, and programmable I/O pins (Figure 5 on page 9). Any combination of structured logic, state machines, embedded processor cores, and various interface logic can be implemented to control battery monitoring, charging, and discharging functions.

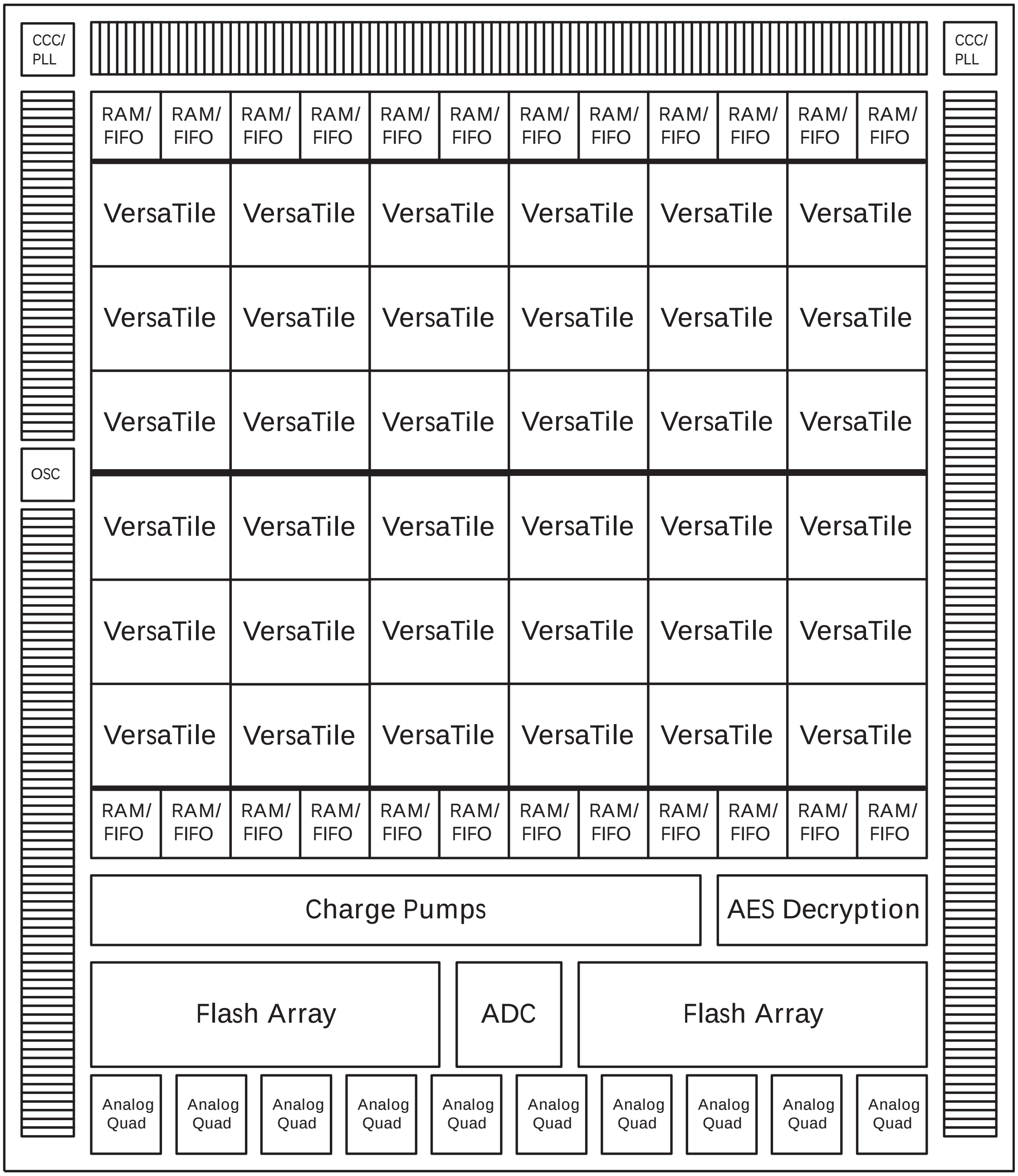

***Figure 5* • Fusion FPGA Fabric Architecture**

Digital logic in the FPGA fabric can be used to implement the required system interface, the wake-up initialization logic, the shutdown state saving logic, and various state machines for SEL interfacing. An embedded processor to sequence ADC conversions and perform the various calculations necessary for alert processing to the higher-level system on potential issues or system parameters may also be implemented. Further, FPGA logic can support unique design features, such as a PCI management bus interface, a serial/modem interface, and serial port sharing. These additional features are all provided in one package, giving the designer a powerful and flexible device to meet all needs in a compact form factor that will not be found in off-the-shelf, low-end microcontrollers.

Embedded Flash Memory Block

Another unique feature of the Fusion device is embedded Flash memory blocks. Each block contains 256 kbytes of nonvolatile memory space and the associated buffering and logic required to easily read, modify, and rewrite the memory contents from the FPGA fabric.

For an IPMI application, the Embedded Flash memory is used to store the BMC/SMC FRU data that can be downloaded from a write-protected FlashROM built into the device. This embedded Flash capability allows the BMC/SMC implementation to scan the system for FRU inventory data to be stored in the embedded Flash for easy access by the higher-level system. The embedded Flash memory is also used for the SDR repository and SEL, which is usually implemented with external bulk Flash memory. The last requirement for the embedded Flash memory is to store microcode instructions for an embedded processor to carry out the message handling and supervisory requirements of the BMC/SMC.

Nonvolatility, Reprogrammability, and Security

Nonvolatility is a key feature of the Fusion device that makes it suitable for IPMI applications. The Fusion device is truly "live at power-up." The Flash-based FPGA fabric retains its programmed state even when power is removed (sleep or power-down modes) and is ready for action immediately when core voltage reaches the turn-on threshold. This means that the Fusion FPGA can reliably control MOSFET states during the power-up ramp and can respond to external hardware commands and begin active monitoring within microseconds of a wake-up event.

Complete reprogrammability of the FPGA fabric and embedded Flash memory blocks provides great flexibility and easy modification during initial product development or later field updates. The Fusion device supports programming and verification of either FPGA fabric or embedded Flash memory contents via a standard JTAG interface, and advanced security features help prevent reverse engineering or tampering with the design.

To protect the contents of the FPGA fabric, a 128-bit AES encryption key can be individually programmed into each device. Once this key is programmed into the device and enabled, attempts to access the device to erase, verify, or program either the FPGA fabric or the embedded Flash memory must be properly encrypted with this key. If no future modifications are to be allowed, you can choose to take a more drastic step and throw away the key. In this case, it is no longer possible to erase, verify, or reprogram the FPGA fabric, and access to the embedded Flash memory may be limited to the internal FPGA logic interface.

Microsemi DirectCore IP in IPMI

Microsemi offers a majority of the cores required to implement either BMC, NMC, or SMC IPMI functions with or without an integrated NMC, in conjunction with Microsemi Fusion or ProASIC®3 products. The cores can be classified into four main categories: communications, processing, analog, and memory interface. Each category is discussed in the following sections, and an example Fusion-enabled IPMI system detailing what Microsemi mixed-signal FPGA can compile in a single chip is provided (Figure 6 on page 11). The datasheets for Microsemi DirectCore IP can be found at

http://www.microsemi.com/products/fpga-soc/design-resources/ip-cores.

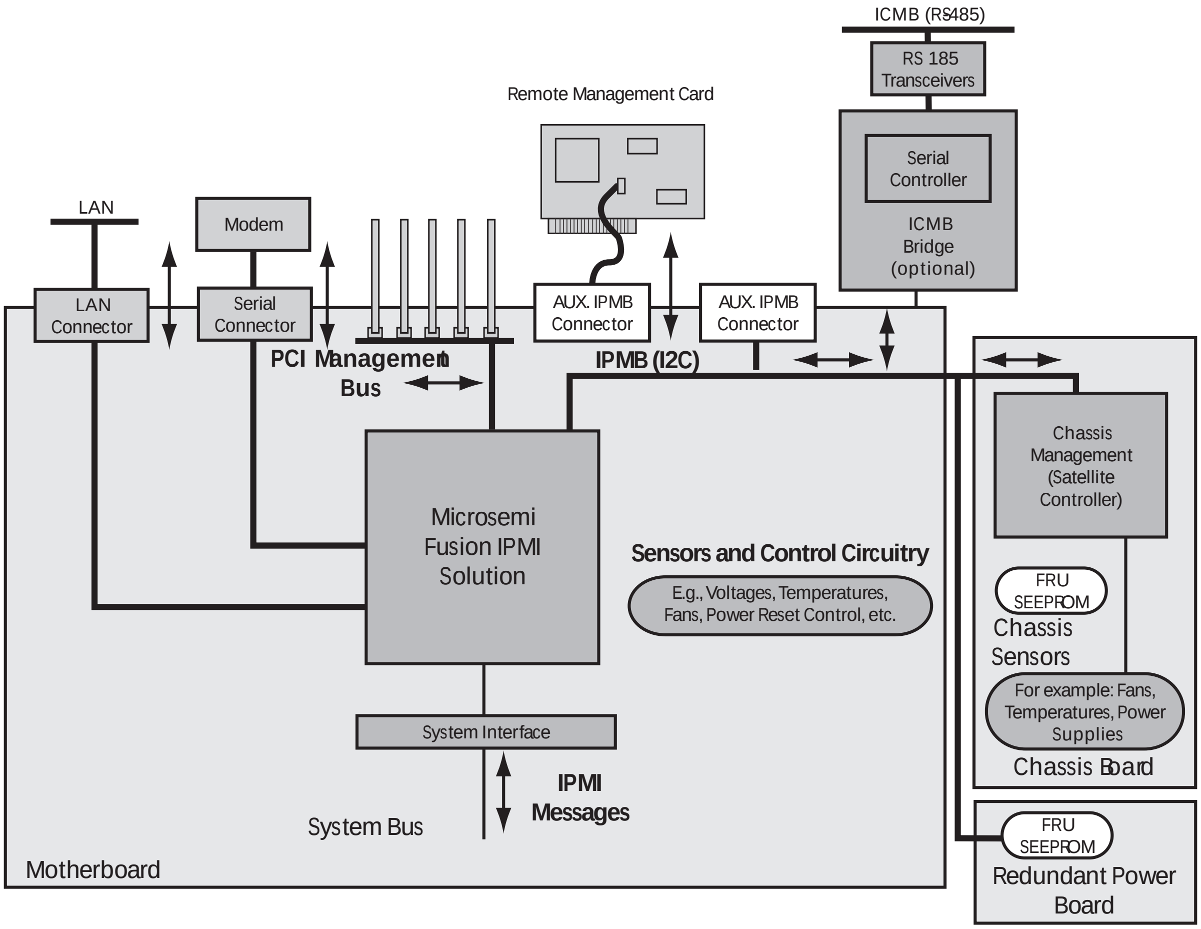

Figure 6 • **Fusion-Enabled IPMI Block Diagram**

Communications

The communications category consists of the Microsemi Core10/100, CoreI2C, CorePCIF, CoreUART, and CoreSMBus DirectCores. Core10/100 IP can be used to replace the baseboard network management controller connected directly to the soft-core processor and treated as a processor peripheral, which eliminates the need for the a PCI interface. The baseboard's IPMB / IPM-C interface can be implemented utilizing CoreSMBus. If additional interfaces are needed (i.e., PCI, an RS-232 compatible serial interface, or additional SMBus interfaces) CorePCIF, CoreUART, and/or CoreSMBus modules may be added to the design.

Processing

The Microsemi soft core processor library includes CoreMP7 (an FPGA-optimized ARM7TDMI-S™ implementation), Core8051 (a single-cycle derivative of industry-standard 8051 architecture) with the traditional SFR interface, and Core8051s with an advanced peripheral bus (APB) interface.

Memory Interface

CoreCFI and CoreFMEE both provide direct access to the internal Flash memory found on the Fusion PSC. CoreCFI provides a subset of the common Flash interface between the Fusion Flash memory and user logic. CoreFMEE is a Flash memory endurance extender, which mimics a serial EEPROM. With a user-defined number of Flash memory pages, CoreFMEE emulates industry standard serial EEPROMs while extending the write endurance of the memory by using a circular buffer. Both CoreCFI and CoreFMEE may be used to implement the SEL, SDR, and FRU memories found on the BMC and SMC platforms.

Analog Interface

CoreAI and CorePWM provide an analog interface between user logic and the Fusion PSC analog block. CorePWM provides pulse width modulation generation for up to eight different channels. Both cores may be operated either in a standalone mode or as a peripheral of the soft-core processors. Using CoreAI, the designer has access to all of the Fusion analog voltage inputs as well as the real-time clock and gate drivers. The combination of the two cores would allow the designer to generate a pulse-train with which to drive the gate drivers. This could be used for driving external devices such as a system fan.

The cores discussed in this section support baseboard and network management as well as satellite control in IPMI and are listed below:

- Remote Communications: Core10/100, CoreI2C, CorePCIF, CoreUART, and CoreSMBus
- Processing: Core8051s and CoreMP7
- Analog and Memory Interfaces: CoreFMEE, CoreAI, CorePWM, and CoreCFI

Fusion-Enabled IPMI System

Figure 6 on page 11 details the modifications of Figure 1 on page 2 proposed in the previous section by incorporating Microsemi DirectCore IP into the Fusion PSC as the baseboard management controller. Figure 7 illustrates a Fusion-enabled SMC for the Microsemi advanced mezzanine card. Fusion-enabled IPMI systems can be built today with the existing Fusion silicon and DirectCore IP. This DirectCore IP will be imported into CoreConsole in Q4, so that customers may easily build a custom IPMI semiconductor IP platform. Additionally, the IPMI protocol stacks for the MicroTCA power module enhanced module management controller and the Advanced Mezzanine Card module management controller interfaces will be available in Q4 and are discussed in the next section.

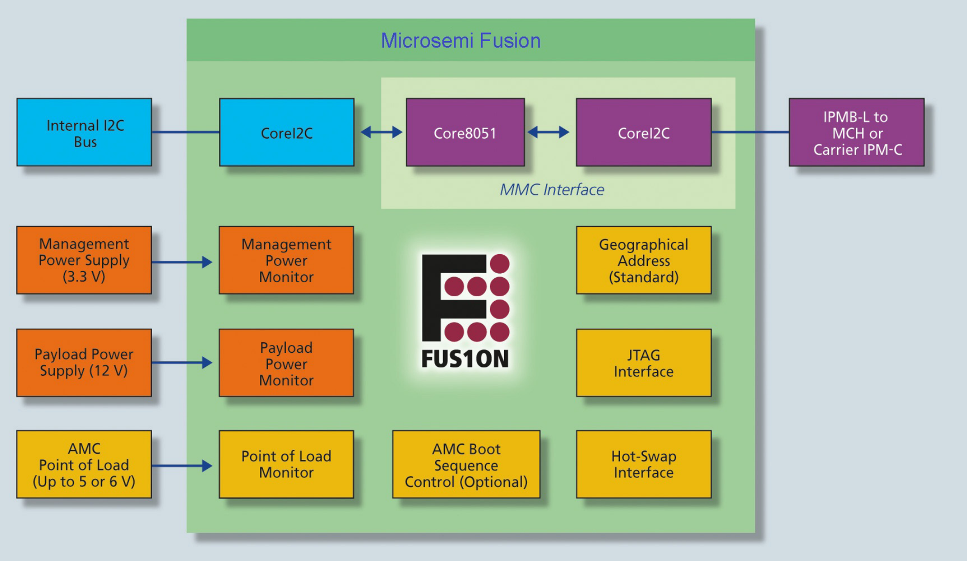
Microsemi Fusion
Internal I2C Bus
CoreI2C
Core8051
CoreI2C
IPMB-L to MCH or Carrier IPM-C
MMC Interface
Management Power Supply (3.3 V)
Management Power Monitor
FUS1ON
Geographical Address (Standard)
Payload Power Supply (12 V)
Payload Power Monitor
JTAG Interface
AMC Point of Load (Up to 5 or 6 V)
Point of Load Monitor
AMC Boot Sequence Control (Optional)
Hot-Swap Interface

Figure 7 • **Fusion-Enabled Advanced Mezzanine Card**

Summary

Microsemi Fusion FPGAs combine all of the features needed to implement an IPMI BMC, NMC, or satellite controller. Measurements of analog voltage, current, and temperature; nonvolatile storage for SEL, FRU inventory, and processor code; and flexible digital logic gates for state machines and embedded processor functions are fused together into a full-featured, single-chip solution that leaves plenty of room for creativity and innovation. Fusion enables the OEM and designers the flexibility to implement additional features to differentiate their next generation of product without sacrificing time to market or cost constraints.

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.