

Military Vehicle Upgrades and Modernization

## Implementation Standard Suits Needs of Military System Designs

As military electronic systems get more complex, the process of doing system debug and verification become ominous. The IEEE Nexus 5001 standard provides a framework that smoothes the way.

Neal Stollon, CTO  
HDL Dynamics

As military and aerospace electronic systems become more complex, new challenges become prevalent. Effectively locating and resolving problems becomes more complex with the increased computing density and complexities of multiprocessor systems. Extensive functional and stress testing of both new system designs and mature systems are also increasingly complex analysis tasks. Better understanding of component and subsystems operations in a variety of environments and over time is important for system qualification, calibration, upgrading, and safety-critical design.

The time and effort related to finding errors and bugs is an increasing amount of the overall system verification budget and typically increases in direct proportion to the complexity of a given system and application. Therefore, improving the understanding of computing systems at every level of integration for a range of operations is critical to successful system development and deployment. Detailed functional visibility into the system operation is one of the most important tools an engineer can have for optimizing and maintaining the system.

### NEXUS 5001 Implementation Classes

	NEXUS 5001	Features
Basic run control	Static debugging Breakpoints	Single step Set breakpoints and watchpoints Two breakpoints minimum Device identification Static memory and I/O access
Instruction Trace Watchpoints	Watchpoints Ownership Trace Program Trace	All Class 1 features Monitor process ownership in Real-Time program tracing
Data Trace Read/write Access	Data Trace Real-time read/write Transfers	All Class 2 features Access memory and I/O in real time Real-Time data tracing
Memory and Port Substitution	Memory Substitution Port Replacement	All Class 3 features Start traces on watchpoint occurrence program execution from Nexus port

Figure 1

The IEEE 5001 standard roughly groups features in terms of four classes of increasing complexity.

An effective method to address these requirements is by monitoring and controlling the system hardware to

discover and debug errors. Adding instrumentation to a design enables better functional observability and control

of systems to address more subtle issues related to integration and environment. A standards-based approach, IEEE-ISTO 5001 (aka Nexus 5001), is a robust and effective mechanism to implement embedded instruments. Nexus 5001 as a standard provides a consistent framework for a spectrum of analysis capabilities, over an increasing range of interfaces, which address support and debug of complex systems.

## The Nexus 5001-2012 Standard

The 2012 Nexus 5001 specification release adds support for 1149.7 and Aurora SerDes standard interfaces. These provide significant systems advantages in implementing more comprehensive debug instrumentation environments. IEEE 1149.7 is an extension to the 1149.1 JTAG standard that defines 2-wire JTAG interfaces as well as parallel chip level data interfaces (as opposed to 1149.1, which typically uses daisy chained serial data interfaces between chips and/or requires external logic for switching signals between multiple devices without daisy-chaining). 1149.7 also supports advanced test and debug features such as Custom Data Transport and Background Data Transport modes, not available in 1149.1, which improve JTAG data transfer flexibility and throughput.

Aurora Gigabit SerDes instrumentation interface, based on the widely used Aurora link-layer protocol, allows for multiple channels to move trace and other bandwidth-intensive data across point-to-point serial links. Aurora is a scalable low-latency protocol that was originally developed for logic-constrained FPGA implementations, which provides a transparent and flexible framing interface to high-speed serial links.

These standard interfaces allow for the introduction of Nexus 5001 instrumentation into systems with pin and wire limited applications such as mobile devices as well as complex networking and computing systems that require large amount of debug data to provide a useful view of system operations.

Additional instrumentation interfaces may also be custom implemented based on Nexus commands. Nexus 5001 supports memory accesses to both on and off chip memory elements, transferring information to local memory for transfer and capture using other peripheral interfaces in a system. It also allows the import of specific data from memory for use in debug configurations. As an advanced example, debug-specific instructions (imported under Nexus 5001 control) may be loaded as a substitute for normal instruction control. This allows the export of trace and other debug information under processor control.

Whether at a board or a chip level, systems contain diverse subsystem components, typically including both processors and dedicated logic. System problems may be due to hardware or software, or a combination of hardware, software and external factors. Problems may be masked or obscured by levels of board and system unit hierarchy making detection and resolution difficult. Historically, the process of systems debugging relies on a combination of system level analysis, detailed (RTL or gate) simulation and instrumentation-based analysis.

For large and complex systems verification, system level approaches may be limited in ability to simulate and analyze the required functional details of large systems. Where standalone hardware or software errors may be commonly found using various simulation verification ap-

### A Basic Nexus 5001 Architecture

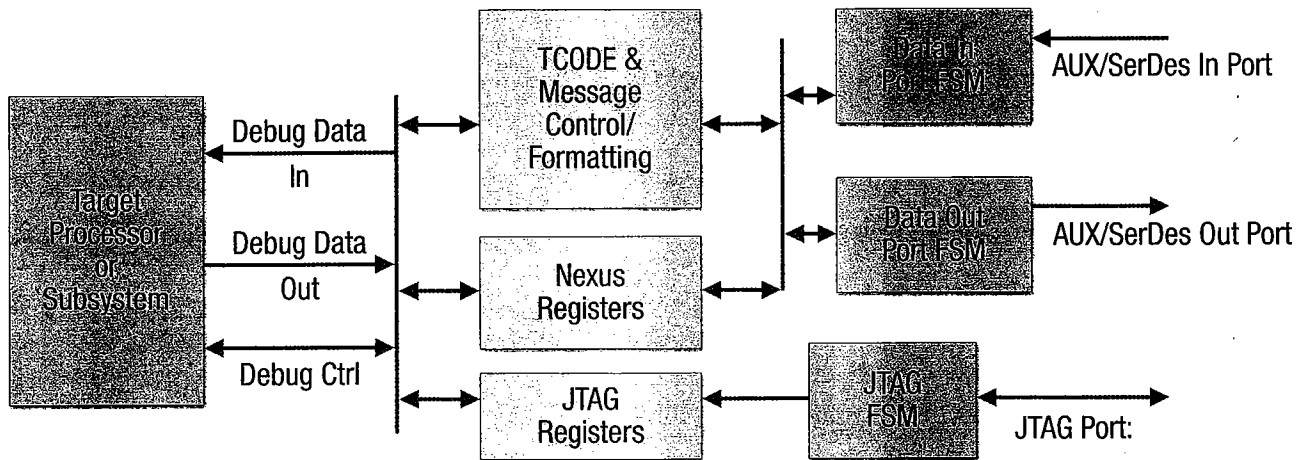


Figure 2

This high level overview shows the basic elements of Nexus 5001 architecture. Key to the concept is that information is transferred as packet messages consisting of fields that include information being transferred as well as information about the information.

proaches, more complex interacting errors may be beyond the ability to model and simulate effectively. Interactions of hardware and software in a real application environment can be complex, subtle and can increase with field upgrades, changes in environment, loss of calibration and other issues associated with extended use. Instrumentation-based analysis provides tools that can be used both during prototyping and in the field over the system lifecycle for identifying and rooting out of system problems.

For military and aerospace systems, having external instruments available in the field is often not feasible. In addition, external instrument-based analysis has only limited I/O for access to operations embedded in the board or chip and capturing information related to subtle or infrequent errors. The automotive industry, facing similar challenges, has adopted instrumentation approaches to improve analysis capabilities. For essential functions such as engine and drive trains, most U.S. automotive systems use Nexus 5001-based instrumentation as a

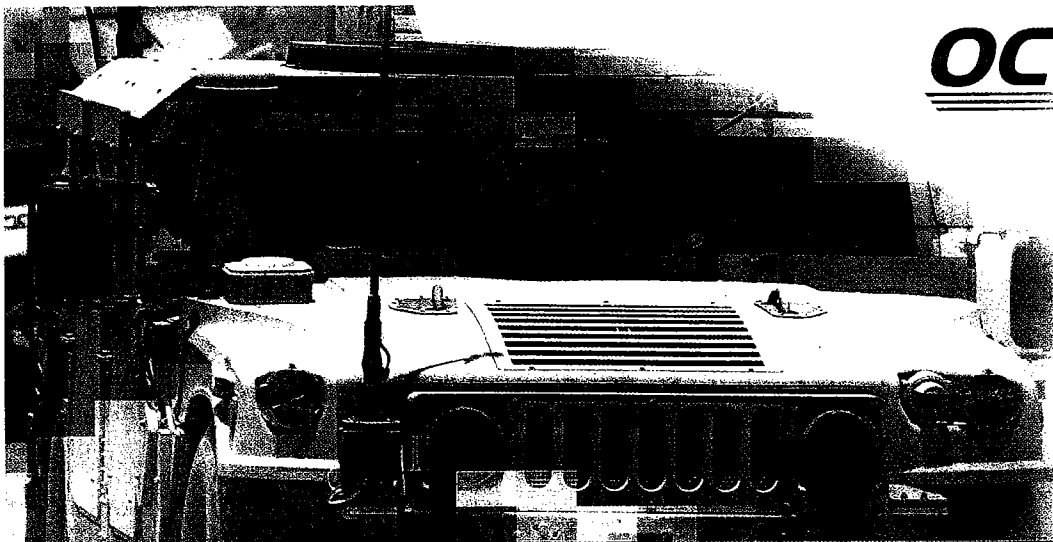
standard method for both analysis and calibration, which may be used in the lab and in the field.

### What Is Nexus 5001?

As military and aerospace electronics systems do not typically mandate standardization in instrumentation as part of requirements, many customized embedded debug features and implementations have been developed. In particular, instrumentation, while included in many diverse systems, may rely on ad hoc solutions developed over the years. These increase the complexity and logistics of system level debug. Other industries, in particular automotive electronics, have recognized this issue, with the result being the initial development (in 1999) of a suite of standardized debug instrumentation architectures, and their subsequent adoption as the IEEE 5001 standard, also known as Nexus 5001. Nexus 5001 has developed over the last decade as a standard-based instrumentation system that supports debug of multi and heterogeneous

processing systems with capabilities to address debug of many complex applications in computing, automotive electronics, telecom and so on. The same approach is very well suited to military and aerospace system designs.

Nexus 5001 was developed to address the limitations and concerns on instrumentation in complex systems. Nexus 5001 defines a modular framework, with features such as packet-based messages, standard and user defined instructions, and user definable fields on many instructions. Standard instructions include industry proven and best in class features for instruction and data trace and data transfer, monitoring, breakpoints and run control for system debug, and importing of data for calibration and port replacement operations. Since debug-related concerns and tradeoffs vary, ranging from the increased budgeted logic required for effective system debug to adequate bandwidth and resources to debug complex architectures and systems, the IEEE 5001 standard roughly groups these features in terms of four classes of



# OCTAGON

SYSTEMS



## COMPUTERS IN MOTION

**CUSTOM • COMPATIBLE • RELIABLE**

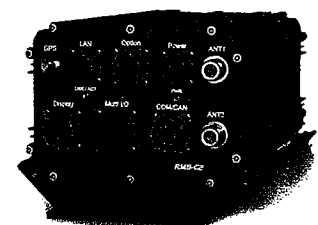
Hardware Solutions for Extreme Applications

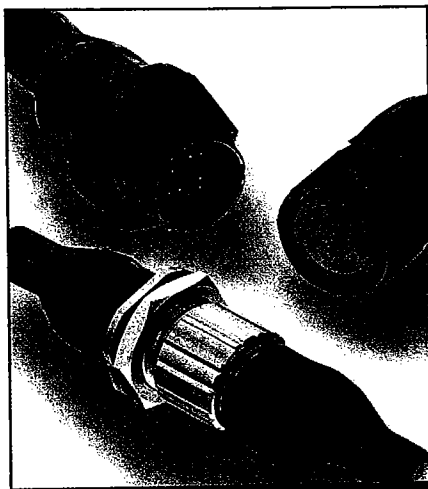
Designed for Rapid Customization

Designed & Manufactured in the USA

ISO 9001 Quality Management System

Over 50 Million Field Hours





**CeeLok FAS-T Connector**

Rugged, Field Terminable,  
10 Gb/s Ethernet I/O Connector

- High-speed data rates - 10 Gb/s
- Crimp-Snap contacts for easy termination and field repairability
- Small form factor connector - Shell size 8
- Integral backshell provides low cost, low weight strain relief and 360° EMI shielding
- Ruggedized for harsh environments

TheFutureUnleashed.com



AEROSPACE,  
DEFENSE & MARINE

**A System Level Nexus 5001 Configuration**

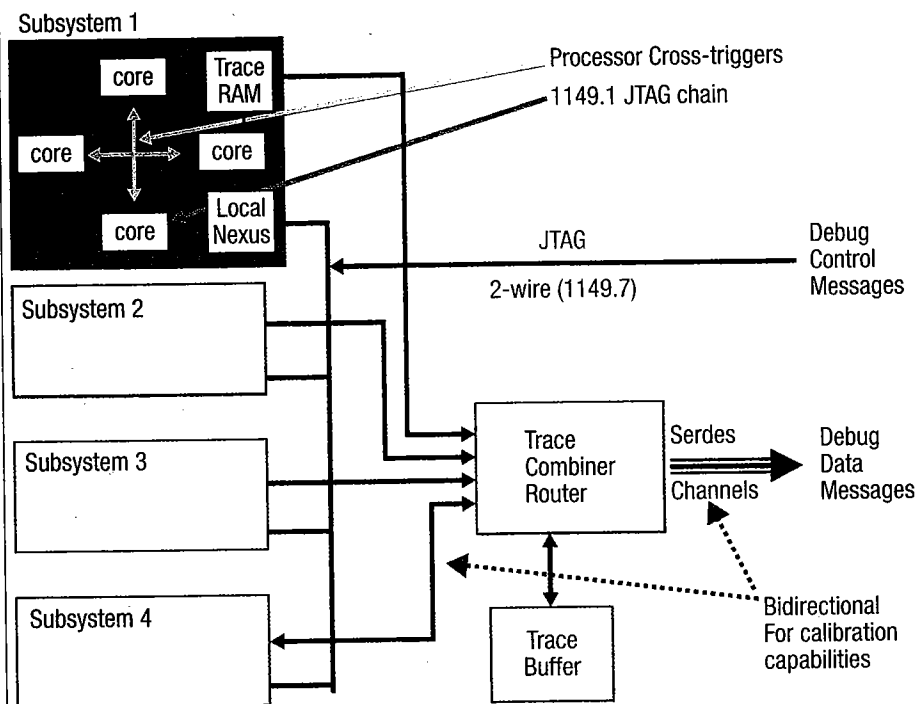


Figure 3

This example implementation uses 1149.7 and Aurora Gigabit SerDes in a system level configuration.

increasing complexity, which are summarized in Figure 1. These classes are guidelines, and not all system debug features will or need to fit into a given class. A comprehensive definition of the various services and features, along with other topics in this section, is discussed in the IEEE 5001 Nexus specification.

**Packet-Based Debug-centric Protocol**

At its core and one of the differentiating features of Nexus 5001 is that it defines a packet-based debug-centric protocol. This allows a variety of debug operations to be defined for different subsystems that can be supported by a single debug core. The debug core can have assorted instrumentation functions that are accessed by standard commands. Nexus 5001 defines standard commands for identification of subsystems, loading and access of on

chip data, including memory accesses, and a diverse set of processor instruction and data trace capabilities. Nexus 5001 also allows creation of user defined instrumentation logic, functions and commands that can be used along with the standard commands. This allows both the ability to debug using a pre-defined and proven instrumentation environment and to implement custom or application-specific instrumentation solutions that would then co-exist with the Nexus 5001 defined and implemented infrastructure.

A high level overview of the Nexus 5001 architecture is shown in Figure 2, Key to the Nexus 5001 concept, information is transferred as packet messages consisting of fields that include information being transferred as well as information about the information. Each message is essentially self-contained and includes packet level fields of information for or

from tools about the source or destination, type of information provided, the register location, timing relative to other information and so on. The source or destination of this information can be a set of Nexus registers, some of which are fully defined and others of which are open for vendor or user-specific applications. The type of debug message is defined through a TCODE header field in each packet. The Nexus 5001 specification de-

fines standardized TCODEs, which range from general purpose register accesses to application-specific trace operations.

### Multicore with Nexus 5001

Nexus 5001 architecture for multicore applications allows several different implementations of a debug interface to be instantiated. Specific instruments can be implemented as required for a given subsystem using Nexus 5001-defined

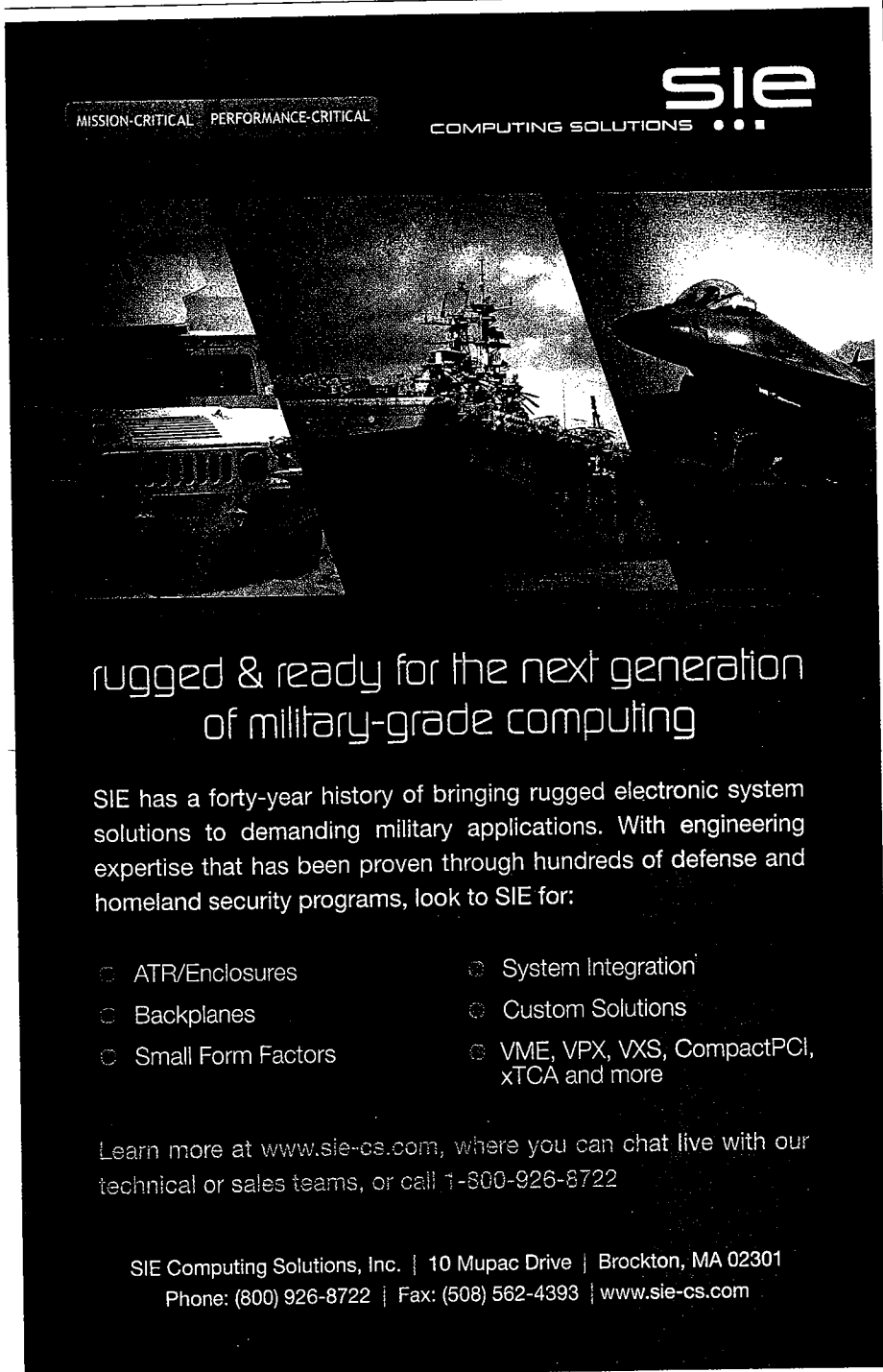
interfacing guidelines for passing information between the subsystem and the Nexus interfaces and registers. Standard Nexus 5001 commands typically use Nexus-defined registers. Custom or application-specific instruments can implement user defined registers, which may be mapped to a reserved space or a memory map. Since the transfer of information is message based, a variety of scheduling and transfer of messages between the Nexus 5001 interfaces and different cores are possible.

Alternate debug approaches such as instrumented software require that the application software is modified in order to capture debug information. This can introduce changes in processor flow that may change performance or mask errors. Nexus instrumentation incorporates on-chip trace and control logic that operates in a parallel background mode to the processor, which allows capture of debug information from hardware operations running at full speed. As such, Nexus instrumentation typically does not reduce processor performance and is unlikely to introduce unexpected changes or latencies that can mask a problem.

Previous revisions of Nexus 5001, in common with other types of debug instruments, support 1149.1 JTAG and streaming parallel trace ports. New to the 2012 Nexus 5001 specification release, 1149.7 and Aurora SerDes standard interfaces are supported, which provide significant systems advantages in implementing more comprehensive debug instrumentation environments. An example using 1149.7 and Aurora Gigabit SerDes in a system level configuration is shown in Figure 3. The Sidebar "The Nexus 5001-2012 Standard" describes the details of the 2012 enhancements.

### Why Use Nexus 5001?

Debug instrumentation is valuable during the design process as both an extension of the verification process and over a system's life cycle as a method of exploring and debugging those operational scenarios that are not too realistically possible with simulation-based tools. The key consideration of including instrumentation in a design is that for a



**MISSION-CRITICAL PERFORMANCE-CRITICAL**

**SIE**  
COMPUTING SOLUTIONS

**rugged & ready for the next generation of military-grade computing**

SIE has a forty-year history of bringing rugged electronic system solutions to demanding military applications. With engineering expertise that has been proven through hundreds of defense and homeland security programs, look to SIE for:

- ATR/Enclosures
- Backplanes
- Small Form Factors
- System Integration
- Custom Solutions
- VME, VPX, VXS, CompactPCI, xTCA and more

Learn more at [www.sie-cs.com](http://www.sie-cs.com), where you can chat live with our technical or sales teams, or call 1-800-926-8722

SIE Computing Solutions, Inc. | 10 Mupac Drive | Brockton, MA 02301  
Phone: (800) 926-8722 | Fax: (508) 562-4393 | [www.sie-cs.com](http://www.sie-cs.com)

modest investment in hardware design and logic, it allows for real-time testing of hardware and software both under conditions being simulated and in real-life environments that may be too complex or costly to include in a simulation.

Components with debug resources, such as processors and FPGAs, work differently, have different data formats and are not trivial to integrate. Nexus 5001 provides a common debug framework

that can be used with both diverse processor and logic systems. The ability to comprehensively debug systems that include different devices and subsystems is important to the design validation and verification concepts that are central to the DO-254 and 17B initiatives.

Nexus 5001 provides a best in class standards-based approach to the problems of debug of complex and diverse systems. It has application at board level

as well as chip levels of integration to discover and address problems and root causes of issues using real-time trace and analysis features, as well as supporting calibration and BIST operations. Nexus 5001 provides a comprehensive standard set of debug options for processor and multiprocessor systems along with features to easily integrate in custom and user defined instruments. Nexus 5001 use has been proven on over 15 different processor architectures and has been used extensively in reliability critical applications such as automotive engine and power train electronics.

### More Interface Options

Nexus 5001 has supported industry standard 1149.1 JTAG and parallel trace and calibration ports since its initial release in 1999. With the release of IEEE 5001-2012, Nexus 5001 also supports emerging debug interfaces such as IEEE 1149.7, which allows new JTAG features such as 2-wire interfaces and parallel chip and board level debug configurations. This allows subsystems to be interconnected for debug purposes with a minimal pin interface at the chip, board and systems levels.

IEEE 5001-2012 also specifies high-speed trace using industry standard SerDes interfaces, allowing improved real-time trace bandwidth needed to support multicore architectures. Nexus 5001 is supported by an industry organization, the 5001 Nexus Forum, which provides resources for both technical and business support of the IEEE 5001 standard. In complex environments where system reliability and accessibility are critical, Nexus 5001 in conjunction with other debug instrumentation solutions provide real advantages in helping to observe, control and understand operations of complex computing systems. ■■

*Neal Stollon is chairman of the 5001 Nexus Forum, which provides industry support for Nexus.*

HDL Dynamics  
(972) 458-9625  
Dallas, TX 75248  
[www.hldynamics.com].

**DON'T BE A VICTIM**

EVEN WHEN YOUR LEGACY BOARD IS DEEMED EOL, GIVE YOUR PROJECT A JUMP START WITH THE I7 INTEL POWERED XVME-6300 AND KEEP YOUR SYSTEM ALIVE.

- ANY VME LEGACY SYSTEM (INCLUDING 5V ONLY CHASSIS)
- HARDWARE BYTE SWAP
- 3 OR 5 ROW VME CONNECTORS, WITH OR WITHOUT PD
- EXTENSIVE I/O OPTIONS

VME • VPX • CPCI • PMC • COM EXPRESS

**XEMBEDDED**  
www.xembedded.com  
734-975-0577