OpenSystems Publishing

SATELLITE PROCESSORS: THE SKY'S THE LIMIT

Military EMBEDDED SYSTEMS

The COTS Technology Authority

VOLUME 3 NUMBER 4
JULY/AUG 2007

IN THIS ISSUE:

WWW.MIL-EMBEDDED.COM

Trends in SDR, reconfigurable computing

Industry

Security

Military

Game Console

Storage Leader

Coming Soon!

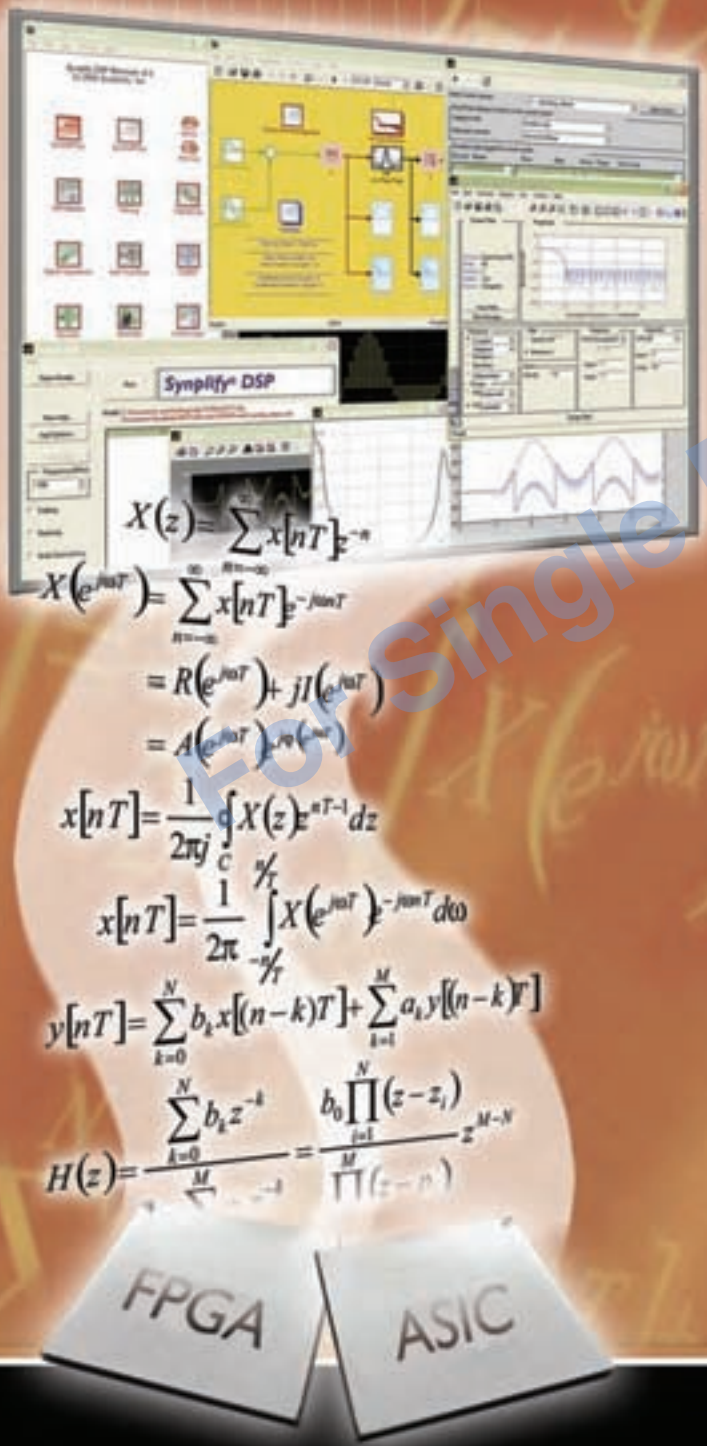
- SATA Interface
- Ultra DMA Support
- No Seek Error & No Noise
- Lower Power Consumption
- No Latency Delay
- Shock Resistant & Anti-vibration
- RoHS Compliant

PQI Corporation
Tel:(510)651-7281
Fax:(510)651-7240
Learn more at: www.pqimemory.com

pqi

Synplify® DSP ASIC Edition

A Breakthrough in DSP Design



Synplicity's Synplify DSP synthesis solution offers a fast, efficient way to implement DSP algorithms in silicon. By automating architectural optimizations like pipelining, resource sharing, and multi-channelization, engineers can save months of RTL coding, simplify design capture, speed up verification, and create technology-independent IP.

Synplify DSP Software Uniquely Offers:

- **Technology-independent DSP modeling library**
- **Comprehensive multi-rate and vector math support**
- **Fixed-point quantization and analysis tools**
- **Powerful DSP synthesis engine**
- **Architecturally optimized Verilog and VHDL implementation**
- **Target the latest ASIC technologies and FPGA devices**
- **Integrated support for standard ASIC design flows**
- **Memory extraction for flexible support of 3rd party memory vendors**

For more information on Synplicity's Synplify DSP solution and all of Synplicity's offerings, please visit our website at www.synplicity.com or contact info@synplicity.com

Synplicity®

Simply Better Results

Military EMBEDDED SYSTEMS

www.mil-embedded.com

July/August Volume 3 Number 4

COLUMNS

FEATURES

E-LETTER

www.mil-embedded.com/eletter

E-cast

IMS and IP Network
Aug. 15, 2 p.m. EDT

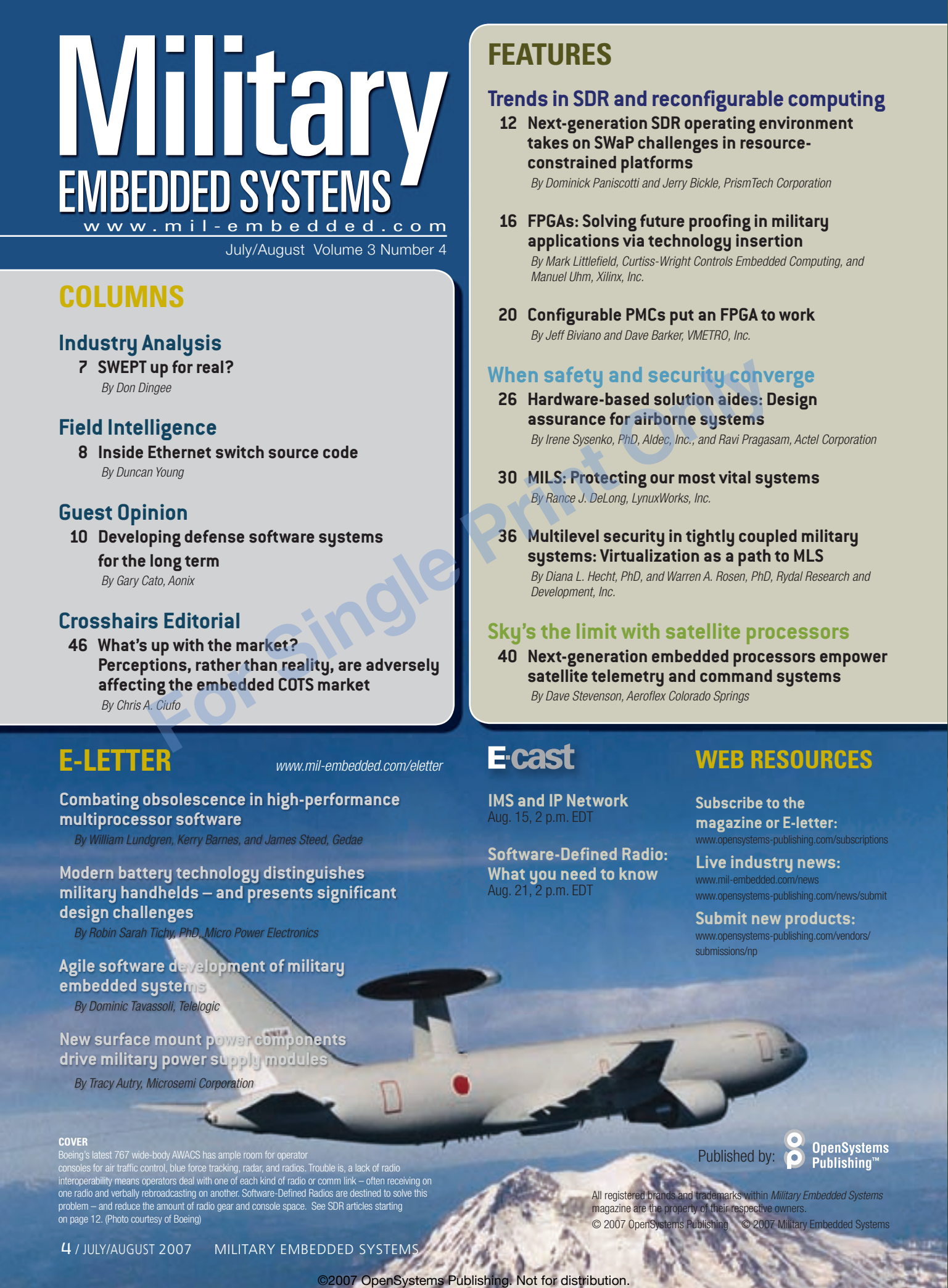
Software-Defined Radio: What you need to know
Aug. 21, 2 p.m. EDT

WEB RESOURCES

Subscribe to the magazine or E-letter:
www.opensystems-publishing.com/subscriptions

Live industry news:
www.mil-embedded.com/news
www.opensystems-publishing.com/news/submit

Submit new products:
www.opensystems-publishing.com/vendors/submissions/np



COVER
Boeing's latest 767 wide-body AWACS has ample room for operator consoles for air traffic control, blue force tracking, radar, and radios. Trouble is, a lack of radio interoperability means operators deal with one of each kind of radio or comm link – often receiving on one radio and verbally rebroadcasting on another. Software-Defined Radios are destined to solve this problem – and reduce the amount of radio gear and console space. See SDR articles starting on page 12. (Photo courtesy of Boeing)

Published by: OpenSystems Publishing™

The Engines of Innovation

Building on its storied legacy as the inventor of the microprocessor and microcontroller, today Intel offers one of the broadest lines of Intel® processors, chipsets, standards-based processor boards and software components for the embedded market segment. With Intel® multi-core processors serving as the building block foundation for industrial automation, interactive clients, automotive, communications and the newest in-vehicle infotainment applications, Intel remains the catalyst for innovation. The universal motor for electronics has now become the engine for change.

http://developer.intel.com/design/info/887.htm

OpenSystems Publishing

Advertising/Business Office
30233 Jefferson Avenue
St. Clair Shores, MI 48082
Tel: 586-415-6500 ■ Fax: 586-415-4882

Vice President Marketing & Sales
Patrick Hopper
phopper@opensystems-publishing.com

Business Manager
Karen Layman

Sales Group
Dennis Doyle, Senior Account Manager
ddoyle@opensystems-publishing.com

Tom Varcie, Senior Account Manager
tvarcie@opensystems-publishing.com

Doug Cordier, Account Manager
dcordier@opensystems-publishing.com

Andrea Stabile, Advertising/Marketing Coordinator
astabile@opensystems-publishing.com

Christine Long, E-marketing Manager
clong@opensystems-publishing.com

Regional Sales Managers
Phil Arndt – New England
parndt@opensystems-publishing.com

Richard Ayer – West Coast
rayer@opensystems-publishing.com

Barbara Quinlan – Midwest/Southwest
bquinlan@opensystems-publishing.com

Ron Taylor – East Coast/Mid Atlantic
rtaylor@opensystems-publishing.com

International Sales
Account Manager – Israel
daronovic@opensystems-publishing.com

Reprints and PDFs
Becky Mullaney – 717-399-1900, Ext. 166
mesreprints@opensystems-publishing.com

Advertiser Information

Military EMBEDDED SYSTEMS

AN OPENSYSTEMS PUBLICATION

Military and Aerospace Group

- DSP&FPGA Product Resource Guide
- DSP-FPGA.com
- DSP-FPGA.com E-letter
- Military Embedded Systems
- Military Embedded Systems E-letter
- Military Embedded Systems Resource Guide
- PC/104 & Small Form Factors
- PC/104 & Small Form Factors E-letter
- PC/104 & Small Form Factors Resource Guide
- VME and Critical Systems
- VME and Critical Systems E-letter
- VME and Critical Systems Resource Guide

Group Editorial Director Chris A. Ciufo
cciufo@opensystems-publishing.com

Senior Editor (columns) Terri Thorson
tthorson@opensystems-publishing.com

Assistant Editor Sharon Schnakenburg
sschnakenburg@opensystems-publishing.com

Copy Editor Robin DiPerna

European Representative Hermann Strass
hstrass@opensystems-publishing.com

Art Director Steph Sweet

Senior Web Developer Konrad Witte

Graphic Specialist David Diomede

Circulation/Office Manager Phyllis Thompson
subscriptions@opensystems-publishing.com

OpenSystems Publishing

Editorial/Production office:
16872 E. Ave of the Fountains, Ste 203, Fountain Hills, AZ 85268
Tel: 480-967-5581 ■ Fax: 480-837-6466
Website: www.opensystems-publishing.com

Publishers John Black, Michael Hopper, Wayne Kristoff

Vice President Editorial Rosemary Kristoff

Communications Group

Editorial Director Joe Pavlat
Assistant Managing Editor Anne Fisher
Senior Editor (columns) Terri Thorson
Technology Editor Curt Schwaderer
European Representative Hermann Strass

Embedded and Test & Analysis Group

Editorial Director Jerry Gipper
Editorial Director Don Dingee
Associate Editor Jennifer Hesse
Technical Editor Chad Lumsden
Special Projects Editor Bob Stasonis
European Representative Hermann Strass

ISSN: Print 1557-3222

Military Embedded Systems (USPS 019-288) is published six times a year (January/February, March/April, May/June, July/August, September/October, November/December) by OpenSystems Publishing LLC, 30233 Jefferson Avenue, St. Clair Shores, MI 48082.

Subscriptions are free to persons interested in the design or promotion of *Military Embedded Systems*. For others inside the US and Canada, subscriptions are $28/year. For 1st class delivery outside the US and Canada, subscriptions are $50/year (advance payment in US funds required).

Canada: Publication agreement number 40048627

Return address WDS, Station A PO Box 54, Windsor, ON N9A 615

POSTMASTER: Send address changes to *Military Embedded Systems*
16872 E. Ave of the Fountains, Ste 203, Fountain Hills, AZ 85268

SWEPT up for real?

By Don Dingee

I had no idea that we would be Slashdotted when we posted the Raytheon news release entitled *Raytheon Develops World's First Polymorphic Computer* on our Mil-Embedded.com RSS newsfeed March 20. I thought I would try to go deeper behind the scenes for the back story on this.

In 1999, Robert Graybill began his term as program manager at DARPA, laying out a vision for embedded computing. "As we look at future systems, they are anything but bounded and will need to evolve with each mission and with technology advances. Translated into computer architecture terms, this means the architecture must be able to support a broad spectrum of functionality by morphing on demand over time. At the same time, each unique mission's functionality, size, weight, energy, performance, and time requirements must still be satisfied, and hence the term *polymorphous computing*."

I like the SWEPT acronym Graybill proposed better than what we usually call *SWaP* (Size, Weight, and Power), as *power* can be a bit ambiguous.

In May 2005, Raytheon was awarded the contract for the Morphable Networked Micro-Architecture (MONARCH) project, working with the Information Sciences Institute (ISI) at the University of Southern California, with assistance from IBM, the Georgia Institute of Technology, and Mercury Computer Systems.

On March 20, we received the news of the first breathing prototypes of a MONARCH chip delivered from the IBM fab to Raytheon. MONARCH combines six RISC cores coupled into a Field Programmable Computing Array (FPCA), with a morphing interconnect to optimize memory access and routing between clusters, and two RapidIO ports for connecting industry-standard devices.

Jeff LaCoss, ISI principal architect, says: "The RISC engines are only a part of the MONARCH chip. Stream processing (such as FFTs) are handled by Raytheon's Field Programmable Computing Array (FPCA). The RISC ISA is an ISI invention developed for the DIVA Architecture, an earlier DARPA-funded project at ISI. The ISA is modeled on the MIPS R3000 ISA, and is also much like the Hennessey and Patterson DLX ISA."

The RISC engines and the stream processing FPCA can work closely together. According to LaCoss, the RISC processors have two modes of operation: "They can run in 32-bit mode like an R3000 with floating point. However, the chip can *morph* and assign resources from the stream processor to the RISC engines. These resources become a 256-bit WideWord data path that behaves similarly to a SIMD processor executing in place of the RISC 32-bit data path. That is, the RISC engine runs instructions for either 32- or 256-bit mode."

This gives MONARCH the horsepower for complex algorithms, and LaCoss gives an example. "A killer app for the WideWord is the corner-turn algorithm, something that conventional CPUs are terrible at. The WideWord can turn an 8 x 8 array of 32-bit objects in around 150 clock cycles, 8 memory reads, some permutations, and 8 memory writes. A conventional processor (such as an R3000 or Power Architecture) must do 64 reads to get the data, 64 writes to store the data, plus a whole bunch of addressing calculations. That's a lot of clocks and instructions. Caches help conventional processors (MONARCH has none) but don't cure the memory-access latency problems."

The RISC engines run an RTEMS-derivative RTOS written by Exogi under subcontract to ISI. Software tools include a production-quality C++ 4.1 compiler written for ISI by CodeSourcery. The assembler is an ISI tool based on GNU *gas*. There are cycle-accurate simulators for the RISC engines, the FPCA stream-processing fabric, and the chip as a whole.

Boutique processors developed for specific jobs should help optimize SWEPT – Size, Weight, Energy, Performance, and Time – to new levels.

Inside Ethernet switch source code

By Duncan Young

Many military platforms are adopting GbE as their backbone intraplatform network. The U.S. Navy has been using Ethernet for many years in combat systems, but many other types of platforms are also turning to Ethernet for its ease and economy of implementation, its wealth of hardware support, and its performance. Some of these new adopters are satisfying requirements for the Global Information Grids (GIGs) IPv6 mandate, but many are adopting Ethernet purely on its merits. Ethernet switches are essential elements of a network and are readily available in stand-alone or embeddable formats, covering the complete spectrum of environmental requirements from the benign office to the harshest military specifications.

The choice of network or switched fabric for a particular application is complex, and more than one type may be employed on a platform to achieve optimum performance. The characteristics of the primary contenders can be summarized:

» **Serial RapidIO –** High performance, high data integrity, low-latency switched fabric for interconnection of many ports within a chassis. Typically Serial RapidIO is used in multicomputing environments for DSP applications.

» **Fibre Channel –** Supports concurrent protocols such as SCSI and IP on the same network. It has flexible topologies and for small systems can be configured without a switch. The use of fibre at the physical level offers inherent protection from electromagnetic effects, making it suitable for military applications. The primary commercial use of Fibre Channel is in Storage Area Networks (SANs).

» **PCI Express –** High-performance, scalable, serial connection using PCI-like memory mapping; often used for connection to high-speed peripheral devices or sensors; similar to PCI as it has a single master that controls access.

» **Ethernet –** High-performance network with 1 Gbps readily available now, 10 Gbps being introduced; synonymous with the Internet and IP. It can be used as an interconnect at many levels from board to board, between chassis, between systems, as an intraplatform private network and, of course, extending to the Internet or the military's GIG.

Key to the performance and interoperability of switches is the onboard software, including protocol, management, and operating system. The functions of a switch are controlled by IEEE standards for the physical and signaling levels and by Request for Comments (RFCs) published by the Internet Engineering Task Force (IETF) for the switching, routing, and management of the switch. Vendors of switches often make use of off-the-shelf, real-time operating systems and commercially produced protocol software to provide a complete packaged product for their customers. However this is not always the ideal solution for long-running military programs where the customer may want to "freeze" at a particular revision of software and still receive support, or have the ability to pass on limited rights to the source code to their end customer.

When Ethernet is used in real-time intraplatform networks, there can be timing and response parameters that can be modified for performance or determinism just for that platform, requiring easy access to the source code for modification and its subsequent long-term support. An example of this is the Spanning Tree Protocol (STP), which is used to determine the optimum routing of messages through a network. When any new connection is made, it must first listen and learn the applicable routing of the network before establishing its normal operation. A root bridge (switch) is elected, and the STP then creates paths through the network from switch to switch, using the shortest routes in a tree structure. Without STP, this routing might create loops resulting in broadcast storms from switch to switch. STP can take a matter of minutes to establish new routing before a device can become active on the network. To address this delay, Rapid Spanning Tree Protocol (RSTP) was introduced to provide a more rapid

convergence of the tree, but there are specific cases in critical networks where yet faster reconvergence is needed.

An intraplatform network, such as might be implemented on an armored fighting vehicle, an Unmanned Aerial Vehicle (UAV), or a naval vessel's combat system, is essentially a closed private network. The network will have been designed with many redundant paths to provide enhanced survivability in the event of battle damage. Waiting many seconds for the RSTP to reestablish network routing if active paths are lost would be unacceptable, but within the closed network environment, it is possible to change the parameters of the RSTP outside its normal limits to provide much faster response to path failures. By reducing the time to detect failures and reestablish network routing, an intraplatform network can be tailored to suit the criticality requirements of a specific platform yet still retain the economy and ease of use that Ethernet offers.

Access to switch source code and the ability to make rapid changes to suit particular customer requirements are both essential ingredients of success in tailoring commercial technologies to suit the sometimes unique requirements of military programs. Recognizing this, GE Fanuc has introduced OpenWare, a switch management package configured from a mix of modules written in-house and from open source to provide the complete protocol, management, and operating system suite for their new range of NETernity Ethernet switches supporting both IPv4 and IPv6. The RM922C, a 24-port switch in 6U VME format, is shown in Figure 1.

Leveraging the best that the commercial technology base has to offer has many advantages for military programs. It translates into lower development costs, faster time-to-deployment, and less risk. However, there are times when the best needs to be better in order to meet the most stringent military requirements. Ethernet switches use tried and tested technology that can be improved on where safety, time-sensitivity, and fail-safe operation are more critical than their commercial counterparts. But this requires the access and capability to modify a switch's source code. Providing this will only serve to enhance Ethernet's adoption into a broader range of critical and strategic military platforms, many of which continue to use proprietary solutions today.

To learn more, e-mail Duncan at young.duncan1@btinternet.com.

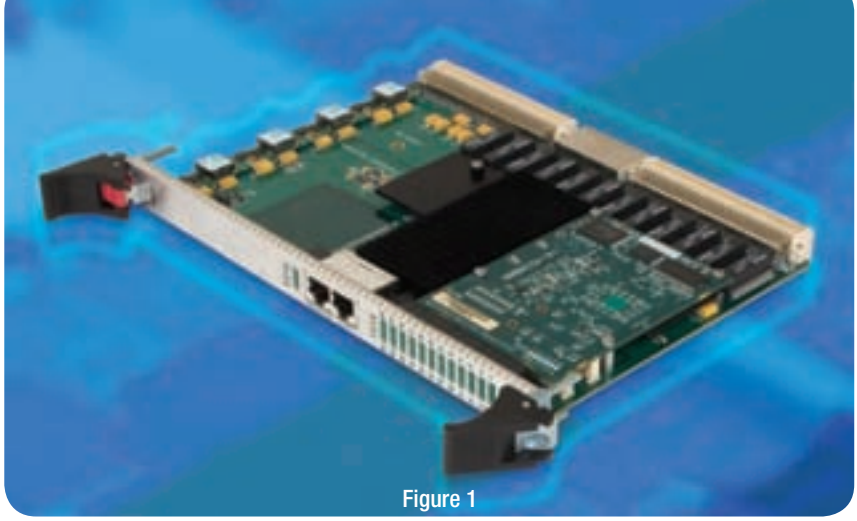


Figure 1

Guest opinion

Developing defense software systems for the long term

By Gary Cato

Military software built with obscure or unproductive programming languages has proven to be an expensive problem. The exploding complexity of software systems creates an imperative for techniques that not only focus on development productivity but will have the commercial popularity and longevity to avoid costly obsolescence that threatens military systems today. Military programs should increasingly look toward Java-based solutions both as a means to drive up productivity and to stave off obsolescence.

Mission-critical software's complexity continues to increase, keeping pace with hardware capacity expansion. Increasing software complexity is a fact of life that begs for technology solutions to manage it. Ironically, as software complexity has increased, so has the use of C++ (and C) in mission-critical development. C++ does not scale well as system complexity grows. There are a number of reasons for this, some within the language itself and others within the ecosystem surrounding the language.

Since the elimination of the Ada mandate nearly 10 years ago, a great deal of pressure within the DoD has resulted in more and more new programs using C++ in a belated chase for access to a more commercially successful language. The result has been a dangerous dumbing down of development for mission-critical software even as complexity has continued to increase.

The commercial success and technical superiority of Java provide a better path. The failures in modularity and scalability of C++, and its inherent high error rates and lack of safety, are absent with Java. While Java enjoys modularity, safety, and scalability similar to Ada, Java also enjoys commercial appeal and support surpassing even C++. Java is now used more broadly on desktop and enterprise applications than C++ and is finding increasing acceptance in embedded and mission-critical applications, thanks to a new generation of technologies friendly to the rigorous needs of resource-constrained systems: native compilation, predictable threading, real-time memory management, and high-efficiency standard libraries.

Java technology has been shown to result in a 2-3x improvement in programming productivity compared to C++, an increasingly vital factor in managing the escalating costs of mission-critical software development. Java also has been shown to result in up to 80 percent fewer errors than C++, containing the costs of integration and long-term maintenance in the face of growing software complexity.

Ada remains a great language. However, when the decision is made for a new program to follow a path other than Ada, C++ is a poor substitute that will be a persistent source of errors and high lifetime cost. Java is a vastly superior alternative to C++ that must be given serious consideration for all new mission-critical programs.

Gary Cato, director of strategic alliances at Aonix, can be reached at: gary.cato@aonix.com.

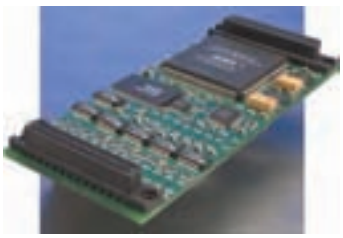
Acromag introduces affordable FPGA I/O. For ALL your projects.

As an engineer, your projects are unique, ever-changing, and budget-bound. That's why our new PMC modules give you an affordable solution to create custom I/O boards.

But if you thought FPGA computing was only for top-end applications, think again. Our PMCs are ideal for protocol conversion, simulation, in-circuit-testing, and much more.

So, why settle for generic I/O when you can design exactly what you need while staying in budget and reducing your time to market?

- Virtex®-II, Virtex-4, Acex®, and Cyclone®-II FPGAs
- Large DRAM buffers and dual-ported SRAM
- Conduction-cooled models available



Industry Pack FPGA I/O also available

Cost-effective custom I/O
Choose from a variety of I/O configurations:

- Digital I/O: TTL, CMOS, RS422, or LVDS I/O
- Analog I/O: 16-bit 100MHz A/D, 900KHz D/A

Faster time to market
Why waste precious time building a board from scratch? Our new FPGA modules let you process your I/O signals any way you want. Quickly.

Flexibility to meet unexpected challenges
Acromag FPGA I/O will help you bring your projects in on time and under budget. And with FPGAs, you'll be ready to adapt to all the inevitable changes.

Thinking about FPGA I/O?
Think flexible. Think affordable.
Think Acromag.

Call or visit our website today – for VME, CompactPCI, PMC, and Industry Pack solutions.

Acromag®
THE LEADER IN INDUSTRIAL I/O

www.acromagembedded.com
877-295-7088 or 248-295-0310

ANALOG I/O DIGITAL I/O SERIAL I/O FPGA COUNTER/TIMER QUADRATURE

Manufactured in Wixom, Michigan, USA

Next-generation SDR operating environment takes on SWaP challenges in resource-constrained platforms

By Dominick Paniscotti and Jerry Bickle

Traditional Operating Environments (OEs) are struggling today to meet the high data rate performance and stringent Size, Weight, and Power (SWaP) requirements of resource-constrained SDRs. To address these challenges, a next-generation COTS, standards-compliant SDR OE is needed – one that is supported by a finely tuned middleware, vertically integrated, and performance optimized.

First-generation standards-based SDR OEs – originally defined by Department of Defense (DoD) contractors on the Joint Tactical Radio System (JTRS) program (see sidebar) – have reached their performance ceiling. A new breed of OE is necessary to ensure that the benefits of SDR and the JTRS program are fully realized.

When the JTRS program kicked off several years ago, OEs were developed by building custom implementations of SCA-compliant Core Frameworks (CFs) on top of existing COTS middleware products. Given the multivendor nature of these initial JTRS OE implementations, the potential for end-to-end performance optimizations was limited. Further, specialized processors such as DSPs and FPGAs, commonly used in resource-constrained platforms, interfaced with these OEs through a high-overhead custom Hardware Abstraction Layer (HAL). The result: First-generation SDR OEs have been handcuffed performance-wise and struggle to meet increasingly high data rates and stringent SWaP requirements for resource-constrained, small form factor SDRs.

To pick up where original OEs have maxed out, next-generation COTS, standards-compliant SDR OEs need to be developed as vertically integrated, performance-optimized products. These new OEs must feature small memory footprints and have the ability to run on any mix of General Purpose Processor (GPP), DSP, and FPGA processor technologies. They also need to be supported by a finely tuned COTS and standards-based middleware and optimized for performance across the complete runtime SDR architecture. It is this mix of qualities that will ensure that SDR is viable across a wide range of resource-constrained platforms.

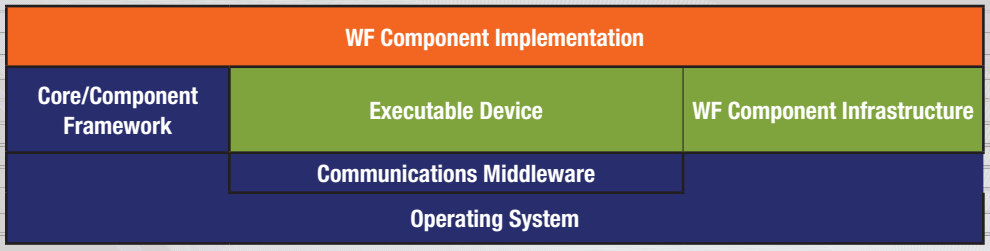
Middleware is a critical component

Middleware's role in next-generation OEs cannot be overstated. Middleware is "where the rubber meets the road" for SDR. At its very essence, middleware provides location-transparent communications between waveform components and enables these components to be located on different processors in an SDR. To achieve this task, it is crucial that it operates efficiently with both the underlying operating system and the CF.

JTRS OE

A standards-based SDR OE was defined by DoD contractors on the Joint Tactical Radio System (JTRS) program to address radio waveform interoperability requirements. This OE, specified by the JTRS Software Communications Architecture (SCA), comprises a Core Framework (CF), CORBA middleware, and an Operating System (OS) with associated board support packages, as illustrated.

The OE imposes design constraints on waveforms and other applications to provide increased portability of those applications from one radio platform to another. These design constraints include specified interfaces between the core framework and application software, as well as restrictions on operating system waveform usage. The SCA thus provides a development rule set focused on the detailed radio set, waveform and software development standards, and specifications that describe how to make the system designed to it interoperable. The core framework is an architectural concept defining the essential "core" set of open software interfaces and profiles that provide for the deployment, management, interconnection, and intercommunication of software application components in embedded, distributed-computing communication systems.



For Systems
that Demand
the Most

XG-5000K

Our most advanced FPGA Based PC/104 Plus Circuit Board

Features

- Xilinx Spartan 3, 5 Million Gate FPGA
- 264 User Configurable I/O
- 256 MByte of SRAM
- 32 MByte of Flash
- 6 MByte of Serial Page DataFlash
- CompactFlash Type 1 Connector
- Secondary Xilinx Spartan 3, 400,000 Gate FPGA
- 10/100Base-T Ethernet
- 2 RS232 Ports
- 25MHz FPGA Master Clock Source
- Secondary 0 - 25MHz user programmable DDS FPGA Master Clock Source
- Incorporates Xilinx's design revisioning technology and can retain onboard as many as 16 partial or 4 complete FPGA design BIT files
- Can be powered from the PC/104 connector or an external DC power supply
- PC/104 Plus Form Factor; ISA and PCI 33MHz Bus Interfaces
- Extended temperature -40C to +85C

Jacyl Technology is a Team Member under Lear Siegler Services, Inc for the US Governments CECOM Rapid Response Contract

JACYL TECHNOLOGY

Advancing Today's Technology into Tomorrow

www.jacyltechnology.com

However, when building their first-generation SDR OEs, radio manufacturers undertook the significant effort of building custom core frameworks and integrating them with operating systems and existing COTS CORBA middleware – as specified by the Software Communications Architecture[1][2] (SCA). Therefore, optimization potential of this multivendor, integrated solution was inherently limited, and has resulted in large memory footprints and suboptimal performance.

Compounding this middleware challenge is the fact that CORBA historically has been associated with large enterprise applications, not real-time embedded systems. It is only now that next-generation OEs can benefit from the significant effort that has been devoted in recent years within the middleware community to streamline CORBA for use in embedded, real-time applications.

For example, new OMG CORBA standards such as CORBA/e[3] have been introduced to address the most demanding requirements of performance-based embedded applications without forfeiting interoperability, portability, and platform independence. Designed to dramatically reduce the footprint and overhead of typical enterprise middleware, this new standard eliminates much of the dynamic and resource-consuming aspects of CORBA, thereby facilitating implementations that can meet the stringent requirements of real-time embedded systems such as SDRs.

The processor challenge

To meet the stringent SWaP requirements of resource-constrained platforms, specialized processors such as DSPs and FGPAs are often used in the design of an SDR. Therefore, the logical SDR architecture described earlier should map to a physical architecture consisting of these specialized processors as well as GPPs, with waveform components executing on and communicating across all three processor types.

As mentioned earlier, first-generation OE designers implemented adapters, or HAL, to enable waveform components implemented on these specialized processors to communicate with components implemented on GPPs, as shown in Figure 1. These adapters required the implementation of custom message protocols, which resulted in decreased waveform component portability due to the specialized interfaces, and decreased performance (latency and throughput) due to the processing overhead of these adapters. These HALs were proprietary in nature and not based upon open standards. Also, waveform developers were forced to implement custom message protocol handlers in addition to the waveform logic required to deliver radios to their customers. This handcrafting of message protocol handlers ultimately led to increases in development and maintenance costs.

Today, next-generation OEs employ an alternative approach, in which the use of standards-based, COTS communications middleware is extended across all processor types, as shown in Figure 2, bringing architectural consistency throughout the SDR.

For SCA-compliant SDRs, this architectural consistency had been achieved by utilizing the high-performance, small footprint CORBA ORBs available DSPs, FPGAs, and GPPs. Now, waveform components implemented on each of these processors may communicate with others without knowledge of their location and without handcrafting special protocols or adapters. This ubiquitous standard communication layer allows radio developers to retarget their software to lower-cost, higher-performance, lower-power platforms as processor technologies continue to mature.

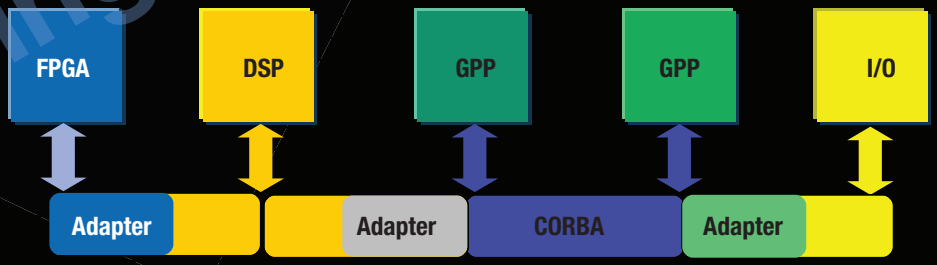


Figure 1

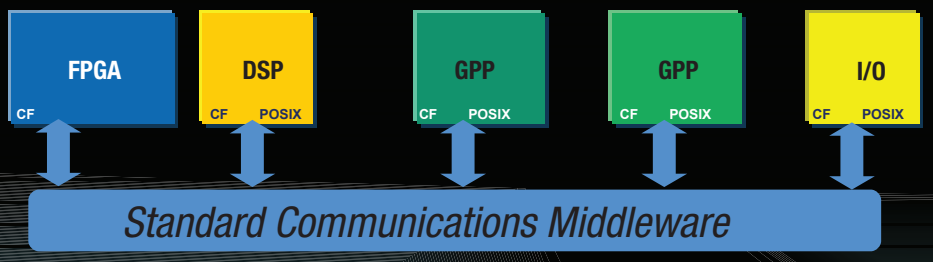


Figure 2

The ISV community and next-generation OEs

There has been much promising work within the Independent Software Vendor (ISV) community on developing next-generation OEs. ISVs are developing OE solutions that can maintain the full functionality of first-generation OEs, yet are not dependent on underlying operating systems or compilers to achieve small memory footprints. And maybe most appealing to system integrators and SDR developers, these next-generation platform-independent OEs provide a direct plug-and-play OE replacement for existing SDRs needing to reduce SWaP. They also provide SDR developers with a flexible, cost-effective COTS alternative to internally developed solutions.

To address DSP resource constraints, next-generation OEs leverage the availability of high-performance, COTS middleware specifically designed and optimized for DSPs. For example, PrismTech delivers a C language CORBA ORB utilizing 75 percent less memory than comparable C++ implementations. A reduction in memory footprint, when achieved without any sacrifices in performance or functionality, enables the deployment of SDR architectures on DSPs and allows a common SDR architecture to be used seamlessly across the increasingly typical processor types used in SDRs.

For FPGA-based components, hardware ORBs are used in second-generation OEs to enable CORBA-based communication between FPGAs, DSPs, and GPPs. Implemented as portable VHDL cores, these types of hardware ORBs can be used by radio developers with FPGAs or as part of an ASIC.

This is in contrast to first-generation SDR implementations that attempted to extend CORBA to FPGAs by executing a CORBA ORB on embedded processor cores within an FPGA. This approach not only failed to achieve the desired portability, consistent SDR architecture, and SDR SWaP goals, but it also decreased throughput and increased latency because adapters were still required within the FPGA.

Because of the architectural consistency of next-generation OEs, new radio platform configurations are available to address the stringent SWaP demands of resource-constrained systems. It is now feasible to implement an SDR-compliant, handheld radio using only DSPs, or just DSPs and FPGAs. Recent proposals from the DoD also request SDR platforms with precisely these processor configurations.

Another key technology utilized in next-generation OEs is the CORBA Extensible Transport Framework (ETF). This framework allows the development of standard and efficient protocols to support optimized communication between ORBs. Off-the-shelf ORBs typically provide TCP/IP transports that are not adequate for many real-time, embedded, constrained systems. ETF provides the mechanisms for adding other efficient, high-performance transports to take advantage of high-speed serial or parallel interfaces such as RapidIO and CompactPCI.

Next-generation OEs are available today

In summary, next-generation SDR OEs address SWaP concerns without constraining processor hardware choices while providing the benefits of a consistent SDR architecture.

Small footprint, high-performance OEs with static memory footprints of well under 1 MB are available today. This 3-10x reduction in size makes standards-based SDRs viable for even the smallest form factor applications. In addition, companies such as PrismTech are also offering the Integrated Circuit ORB (ICO) for FPGA and ASIC SDR platforms to provide end-to-end architectural consistency across all processor types in use today for SDR platforms.

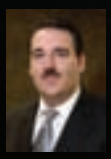
The new OMG software radio specification

Additional SWaP savings can be realized by SDR OEs based upon the new OMG software radio specification[4] and the flexibility inherent in this standard. The standard, which represents the SCA's commercial evolution, allows SDR developers to tailor their solutions by eliminating architectural features, as needed, to deliver only the SDR features required by their end users.

The availability of these next-generation OEs relieves radio manufacturers of the development and maintenance burden associated with custom OEs and significantly accelerates time to market for their SDR products. As a result, application developers using these architectures can now focus on delivering radio functionality rather than struggling to build proprietary middleware to allow communication between the disparate processors used within SDRs. The technological advances in next-generation OEs along with the commercial availability of optimized implementations for the military and commercial markets has made SDR an available reality today.

References

1. Software Communications Architecture 2.2, http://jtrs.spawar.navy.mil/sca/downloads.asp
2. Software Communications Architecture 2.2.2, http://jtrs.spawar.navy.mil/sca/downloads.asp
3. CORBA/e and RT CORBA, *www.omg.org/technology/documents/specialized_corba.htm*
4. OMG PIM and PSM for Software Radio Components, www.omg.org/technology/documents/profile_catalog.htm#UML_for_SWRadio



Dominick Paniscotti, *PrismTech's VP Engineering, SDR Products, has been involved in SDR design and development since 1998. He is a cofounder of the Object Management Group (OMG) software-based communication task force and actively supports the commercial standardization of SDR technologies via the SDR Forum. He earned his BSEE from Fairleigh Dickinson University.*

Jerry Bickle, *PrismTech's Chief Scientist, SDR Products, has been involved in SDR development and design since 1997. He and Dominick Paniscotti coauthored the JTRS Software SCA 2.2 specification, and Jerry has also served as a member of the JTRS Technical Architecture Group (TAG). He helped lead the SDR Forum's SCA adoption and the formation of the OMG software-based communication task force. He holds a BA in Computer Science and Mathematics and an MS in Secondary Education from Northern Illinois University.*

PrismTech Corporation
6 Lincoln Knoll Lane, Suite 100
Burlington, MA 01803
781-270-1177
info@prismtech.com
www.prismtech.com

FPGAs:
Solving future proofing in military applications via technology insertion

By Mark Littlefield and Manuel Uhm

Technology insertion using COTS FPGA-based products promises to deliver significant benefits but requires adequate planning up front. Success involves both hardware and software strategies. Following a common hardware model and minimizing connector pin changes are key to easing hardware technology insertion issues. Software problems can be mitigated through the use of highly abstracted APIs from one product generation to the next.

Over the past decade, the concept of "technology insertion" has become a sort of mantra for the Aerospace and Defense (A&D) community, promising an approach for upgrading fielded systems with the latest, most advanced technology with the least delay. The rapid pace of technological innovation – and subsequent obsolescence – has made technology insertion critical for the success of multiyear A&D development, test, and deployment projects. Unmanned Aerial Vehicles (UAVs), radar, and SIGnals INTelligence (SIGINT) are examples of sophisticated platforms that have benefited from technology insertion. To fully understand the challenges of technology insertion, including complexity and cost, it's useful to consider what is involved in successfully using this approach for upgrading legacy systems. Using FPGAs in a reconfigurable computing application provides a good example of the benefits and challenges entailed in making technology insertion work. This example also enables examination of some common issues associated with technology insertion in such applications, how COTS vendors can best help their customers to address those problems, and how the overall technology insertion problem can be significantly eased by proper planning.

Using FPGAs in technology insertion

The basic problem domains involved in technology insertion can be roughly categorized as:

- Hardware – How the equipment physically connects
- System software – How the operating environment, system libraries and utilities, drivers, and middleware provide an infrastructure for applications

Using FPGA technology in an embedded multicomputing system provides an interesting case example because it straddles both the hardware and system software domains. FPGAs fall into the hardware domain because they are physically integrated to hardware elements, and the developer must work very closely with these elements when designing an application. However, FPGAs present a software challenge as well since they must be programmed and often incorporate vendor or third-party supplied "blocks" or "IP." In addition, a general purpose processor using system library calls is typically used to configure the FPGAs and the commands they use to communicate with other processors and devices in the system. Such systems are often complex and heterogeneous as shown in Figure 1.

For technology insertion, the ideal scenario is a plug replacement module that requires no hardware or software changes. While historically uncommon for FPGA-based computing products, this scenario can also often be undesirable since state-of-the-art computing platforms can offer new, more advanced features that system integrators can use to their advantage. The goal for the system integrator must therefore be to maximize the benefits of technology insertion while minimizing the impact on the existing system. The degree of flexibility in the hardware and software domains can be directly affected by early design choices made by the system integrator and the products offered by COTS vendors.

The simplest approach for a COTS vendor to ease technology insertion hardware problems is to follow a common hardware model from generation to generation of platform and to minimize connector pin changes. Industry board and module standards such as VME, VPX-REDI (VITA 46/48), PMC (VITA 32), and XMC (VITA 42) address a large part of this problem by specifying fixed form factors and pin definitions for board I/O such as buses or switched fabric interconnects. Vendors can extend this model by maintaining pin footprints across product generations for such common interfaces as serial ports and Ethernet.

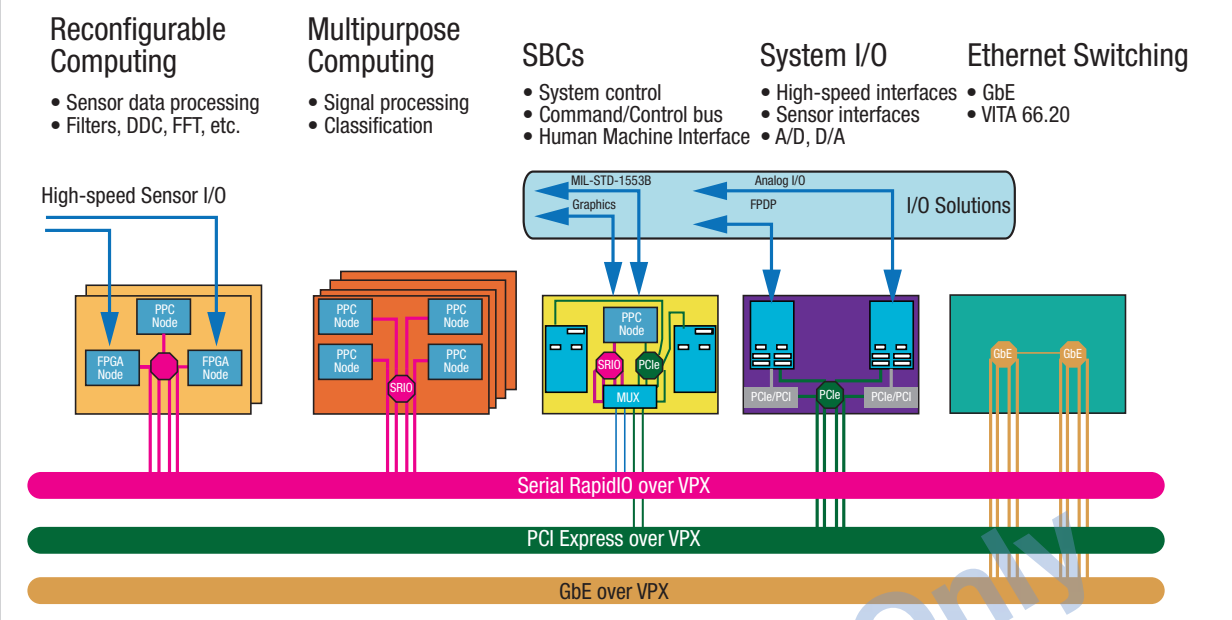


Figure 1

The same also holds true for FPGAs. A common, or at least similar, I/O footprint minimizes the need for radical redesign during a technology insertion project.

The importance of software

Although FPGA development is very tightly linked to the target component and platform hardware design, there is a lack of standardized or industry-accepted tools and frameworks to abstract this linkage. The result is that software can typically have an even greater effect on a technology insertion program than hardware. The software challenges can be somewhat mitigated on general purpose processors by using off-the-shelf operating systems such as VxWorks or Linux, supported with full-featured BSPs, communications middleware, and application frameworks. For COTS vendors of FPGA products, the challenge is to provide development tools to the integrator to ease technology insertion while maintaining the performance demanded by the application developer. A common approach is for the COTS vendor to develop a standard "wrapper" or gasket that essentially represents the static infrastructure, such as interfaces to A/Ds, memories, Ethernet, and so on. The blocks and infrastructure should be designed to support a set of commonly implemented use cases in the most efficient manner possible, so as to minimize the size of the wrapper. Ideally, such infrastructure would represent only 5 to 10 percent of the total die size of the FPGA.

Software support of FPGAs

No less important than the ease of integrating existing application code into a new FPGA platform for a technology insertion project is the integration of the FPGA-based application into a larger multicomputer system. General purpose computing elements are often closely tied to the system's FPGAs by performing various command and control functions such as DMA engine control. Subsequently, the ease of a technology insertion can be directly affected by how well a vendor can maintain APIs from one product generation to the next, which, in turn, is often linked to the level of abstraction in the API: The more abstraction, the less likely it is that the API will change between product generations. One of the key tasks of an external processor is the command and control of data movement both within the FPGA and between the FPGA and other general purpose processing nodes.

Making technology insertion real

There are many examples of a common hardware strategy from several COTS vendors that aid technology insertion. One example can be seen in Curtiss-Wright's latest-generation VPX-based board products. The three primary computing platforms in this product line – a dual 8641-based SBC, a quad 8641-based DSP engine, and the dual Xilinx Virtex-5 (Figure 2) FPGA-based CHAMP-FX2 (Figure 3) board – all share a set of common hardware design elements and were

Figure 2

Figure 3

specifically designed with common pin footprints for common I/O elements. For instance, the FPGA board not only shares the PowerPC node design elements with the two other VPX boards, but it also closely resembles its preceding VME version in its I/O and memory configuration.

The design of the VPX FPGA board and its VME predecessor took a balanced approach between the twin goals of maximum technology insertion value and minimum insertion effort. Their Continuum FXtools design kit provides a highly optimized set of IP blocks focused on the peripheral I/O and memories, with only a lightweight, scalable switching block and a few additional utility blocks to ease application integration. By providing a minimal but highly optimized infrastructure, FXtools abstracts those common I/O and memory objects without sacrificing performance. Designing with these IP blocks and adopting basic use cases can help ease future technology insertion.

To ease both the integration and technology insertion of FPGA-based computing elements, the boards' Continuum IPC communications middleware was extended to directly control DMA-based data transfers involving the FPGA node. The IPC software enables the application developer to create named buffer and data transfer objects that are globally visible to the system. Thus, any processor in the IPC system can create named buffers, attach to already created named buffers, and create data transfer objects between buffers, without having to resort to code-manipulating complex memory map translations or DMA command packet creation.

Hints for avoiding technology insertion headaches

While hardware and system software commonality across product generations is an important element in planning a successful technology insertion project, the application developer and system integrator can themselves play an important role in defining how difficult a future technology insertion project may be. For example, if the application developer bypasses vendor or OS APIs to directly manipulate hardware, the resulting code will be significantly less portable to future hardware platforms. However, there are a number of other ways in which an application developer or system integrator can structure their application/system to minimize technology insertion headaches.

The first rule of designing for later technology insertion should be: "Don't fight the system software or OS." Most software frameworks, such as a BSP, utility library, or communications middleware like Continuum IPC, are designed with one or more basic use cases or design patterns in mind. One of the easiest ways for a developer to introduce problems in a later technology insertion project is to bend the software framework to match their favorite design pattern. Doing so virtually ensures that a later developer will have to bend things in a different way simply to make it work. The same holds true for the FPGA IP developer. While implementation details and interfaces may shift from one product generation to the next, the basic design patterns usually do not. Using vendor-supplied IP blocks and data flows in the manner that the vendor originally intended helps minimize the amount of recoding needed for a technology insertion project.

Another beneficial technique that can be employed both for software and FPGA IP is to analyze the vendor-supplied APIs and block interfaces to identify those interfaces that are likely to shift in future products and to create an abstraction layer or "shim" between them and the application code. A helpful clue is that if an API or block interface closely mirrors the underlying hardware, it is likely to shift between product generations. While such layers can be seen as adding unnecessary code and performance delays to a system, often the performance impact is minimal while the benefits for a later technology insertion project can be enormous – changing a shim is much easier than changing multiple instances of an API call or IP block instantiation. It's important not to overuse this approach, however, as adding shims or abstraction layers complicates designs and can sometimes make debugging more difficult.

Planning for the future

Technology insertion can live up to its promise of future-proofing, but only if it is adequately planned for up front. Most COTS vendors have developed successive generations of products with this in mind. In order to capture this value, system integrators need to work closely with such vendors early in the design cycle.

Mark Littlefield *is product marketing manager for Curtiss-Wright Controls Embedded Computing's FPGA computing products. He has more than 15 years of experience in the embedded computing industry, first as an engineer developing robot vision systems for NASA, then later as a field applications engineer, technical program manager, and product manager. Mark has a BS and an MS in Control Systems Engineering from the University of West Florida.*

Curtiss-Wright Controls Embedded Computing
741-G Miller Drive SE
Leesburg, VA 20175
703-779-7800
mark.littlefield@curtisswright.com
www.cwcembedded.com

Manuel Uhm *is senior marketing manager for the DSP division at Xilinx, responsible for strategic marketing and development of division road maps for commercial wireless and defense. Prior to joining Xilinx in 2004, Manuel was responsible for marketing at Spectrum Signal Processing, Inc. He is currently co-chair of the markets committee of the SDR Forum. Manuel received his MBA from Simon Fraser University and studied Electrical Engineering at Queen's University.*

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
408-559-7778
manuel.uhm@xilinx.com
www.xilinx.com

DDC®

©Data Device Corporation

EXPERIENCE

25 Years of MIL-STD-1553 Service!

Toll Free: 1-800-DDC-5757 **www.ddc-web.com**

Configurable PMCs put an FPGA to work

By Jeff Biviano and Dave Barker

Reconfigurable computing isn't a novel concept, but the idea of a configurable PMC using an FPGA, swappable I/O adapter modules, and I/O-specific IP is a new wrinkle. These configurable PMCs can help reduce design time by providing a platform for adding I/O to a single board computer with PMC sites. We explore the key architectural points to consider in designing around the configurable PMC concept.

It's natural for engineers to spend much of their time thinking about the core processing capability of a system, often in the form of a single board computer. However, experienced designers know that getting the most out of a system means devoting much of their attention to I/O capability, especially in the case of high-speed I/O.

Perhaps a custom sensor for a surveillance system must be integrated, but a full custom I/O design is time prohibitive. Or, perhaps a C4ISR module requires increased A/D performance, but again a totally new A/D design would take too long. In both cases, the developers probably have other development and integration issues to address and don't have the time or resources to devote to developing new I/O cards.

Modular I/O subsystems designed around an FPGA can make a developer's life easier by reusing common components and tailoring the solution to fit application needs. An I/O subsystem can be based on any form factor, but there is strong argument for basing it on the industry-standard PMC form factor. According to Venture Development Corporation, 88 percent of single board computers sold in 2007 are projected to have PMC or XMC sites available. This is because of the success of the proven PMC form factor over its decade-plus life.

Architecture for configurable PMCs

Configuring a PMC for a variety of I/O roles logically involves three architectural elements (Figure 1):

- Adapting the I/O pinout to meet the physical interface requirement
- Providing the I/O data acquisition and processing functionality
- Getting data from the PMC into the host single board computer for further processing

Adapting the I/O pinout

To satisfy the needs of a different application or the changing requirements of the same application, a new PMC has to be built. However, if the PMC is designed from the start with an FPGA and the I/O interface is on a pluggable I/O adapter module, the only portion of the hardware that needs to be changed is the pluggable I/O adapter module (Figure 2); the base PMC module can be reused with the appropriate programming of the FPGA.

This modular approach has the advantage of flexibility to adapt a wide variety of I/O types with a single hardware architecture and the ability to evolve with changing requirements over time. It also cuts development cycle times since proven and optimized functional blocks such as memory interfaces and PCI/PCI-X connectivity are reused.

Pluggable I/O adapter modules can be designed for a variety of I/O functions; with an open interface, the possibility exists for custom user-designed I/O adapter modules for even more functions. These modules can contain both the necessary physical elements and functional circuitry such as a high-speed A/D converter to capture data from an I/O device. For instance, an I/O adapter module with dual 16-bit 125 MSps A/D channels operating simultaneously will produce 500 MBps of data. A PMC-X module with a 133 MHz 64-bit PCI-X link with approximately 1 GBps bandwidth is able to support these data rates.

I/O data acquisition and processing functionality

An FPGA on the PMC provides the developer with the flexibility of not only implementing the necessary I/O protocol but also the ability to implement application-specific processing and the performance to process very high data rates in real time. Building on the previous A/D example, if the application required two additional analog input channels, a quad 16-bit 125 MSps A/D I/O adapter module could be developed that would produce 1 GBps of data. This would most likely exceed the theoretical ~1 GBps of 133 MHz 64-bit PCI-X. However, if only a 5 MHz band of interest was desired in each channel, a digital down converter could be used to produce a downconverted sample rate of 12.5 MHz based on a sampling frequency of 2.5 times the 5 MHz bandwidth. The data coming out of the FPGA would be reduced to 100 MBps (12.5 MHz/sec x 16-bits/

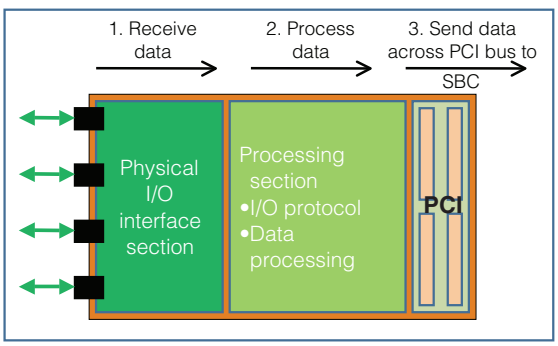


Figure 1

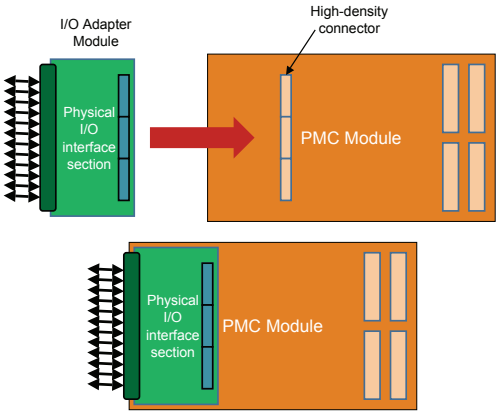


Figure 2

sample x 4 channels), which could easily be handled by the PCI-X link.

Another way to handle higher input data rates is to increase the capacity of the link between the PMC and the single board computer. PMC-X and XMC modules can increase the bandwidth over the 64-bit, 66 MHz PCI link of a PMC.

Data movement into host SBC

A PCI, PCI Express, Serial RapidIO, or PCI-X interface can be implemented with an ASIC. Alternatively, these capabilities can be implemented as an IP core in the FPGA. By choosing the right FPGA, a single FPGA can interface to the I/O, implement the protocol and preprocess the data, and include a PCI, PCI-X, Serial RapidIO, or PCI Express IP core to interface the PMC/XMC to the single board computer. With the host interface logic implemented in an FPGA, system throughput can be optimized by attention to elements such as buffering, interrupt handling, DMA, and other strategies to help data stream efficiently from the I/O subsystem into the core processing complex.

Utilizing this approach, it's difficult not to envision a single PMC with a family of pluggable I/O adapter modules spanning a range of high-speed functions all based on the same FPGA hardware and IP. An example implementation of this approach is the VMETRO PMC-FPGA05 family, with a Xilinx Virtex-5 and a series of off-the-shelf I/O adapter modules and IP for I/O including A/D, D/A, FPDP, LVDS, Camera Link, RS-485, and L-band digital receiver functions (Figure 3).

In this implementation, a 138-pin connector near the PMC front panel brings signal traces routed for use as either single-ended signals or differential pairs to the FPGA. The signals are grouped into two banks, with each bank independently configurable to 2.5 V or 3.3 V signaling. Having 138 configurable pins routed from an FPGA to an I/O adapter module connector provides the capability to support a wide range of I/O.

Configuring the FPGA

The ideal configurable FPGA platform provides a flexible I/O module interface that can be used to accommodate a multitude of I/O functions from analog conversion to interfacing with complex bus protocols. However, the primary focus for an application is the core processing algorithm. To allow application engineers to concentrate their efforts on the design of the algorithm, it is imperative that they have accompanying interface IP at their disposal. The IP needs to abstract away the intricacies of complex protocols, providing simple intuitive interfaces that can be easily integrated with the core processing algorithm.

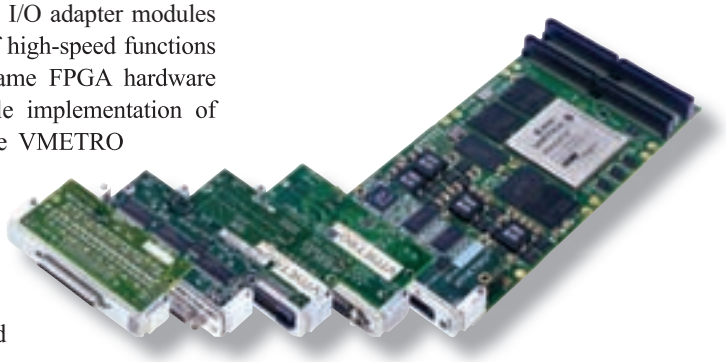


Figure 3

Interface IP can typically be broken down into two basic categories:

- Streaming interfaces where a continuous data stream is presented
- Addressable interfaces where data is stored or retrieved from a specified location

Providing a common method for interfacing to these two categories of interfaces allows an array of I/O options to be readily adapted to by an application programmer.

A look at the block diagram of the example implementation highlights three interfaces that are commonly present in a configurable FPGA platform (Figure 4).

I/O module interface

The I/O module interface most often represents sensor input and arrives in a streaming format. In the example block diagram, the I/O is a single channel of ADC input data. The associated IP interface takes the ADC samples and presents the user with a common streaming IP interface that can connect directly to the processing algorithm block or can be connected to FIFO with a compatible interface for crossing between the operating frequency of the ADC interface to the algorithm interface.

Memory interfaces

Memory interfaces are typically addressable in nature, but in some instances memory can be used to buffer a data stream. The example implementation shows two blocks of external memory with two associated styles of IP blocks. The first block provides user interfaces that can be used to address individual memory elements such as where memory is being used as a lookup table for a processing algorithm. The second IP block illustrated shows the memory interfaces with streaming interface blocks for the user to connect an input interface to one and an output to the other. Thus the user has a simple choice to connect the algorithm directly to the PCI-X interface or use the QDR as a buffer that can absorb data during instances where there is a delay or hiccup in transferring data across PCI-X to a general purpose processor.

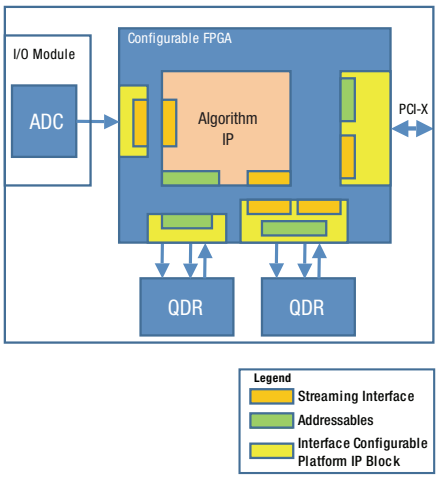


Figure 4

Common IP interface advantages

By creating a common method for interfacing to IP elements, the user is able to concentrate most of the coding effort on analysis of the different types of filtering and processing effects in the algorithm, as opposed to creating glue logic to interface to the varying bus protocols. Once common interfaces have been established, they can

be utilized by high-level FPGA tools such as Xilinx System Generator or Impulse C. Thus, the high-level tools can be used to create IP that more readily drops into a larger FPGA design.

Drivers and software

Once the FPGA design is established, it behaves identically to how any "hard" PMC design would. To integrate the configured FPGA platform into the target system, software needs to be established to communicate between the FPGA and the attached general purpose processor.

Platform software should take advantage of reusable IP and provide reusable drivers. As there are variances in the reconfigurable designs such as the number of DMA channels, or the location of register control interfaces connected to the PCI-X interface, the software needs to be able to adapt. One way for the software to automatically adapt to the changes in the hardware interface is to provide a feature table in hardware that can be referenced when the platform drivers are initialized. The combination of the configuration table in hardware and adaptable driver software from a plug-and-play interface can, for instance, dynamically instantiate the corresponding number of DMA drivers in software.

Established software APIs should include support for accessing the addressable interface connected to the PCI-X interface, accessing streaming interfaces connected to the DMA channels, as well as respond to asynchronous interrupt events from the configurable FPGA device.

Advanced design tools

High-level design tools are becoming more prevalent in the configurable FPGA industry. The tools abstract away the finite details of HDL languages, allowing designers to create their FPGA algorithms in more familiar and mathematically friendly environments such as The MathWorks Simulink environment and C.

Common IP interfaces aid in integrating such cores into larger designs. In addition, these high-level tools can perform hardware verification of the completed FPGA design by using configurable platform software to send the simulated data sets to real hardware while reading back the processed data results to ensure that the hardware design matches the simulation results.

Tapping the power of reuse

We have shown how using configurable PMCs with pluggable I/O adapter modules backed by an FPGA delivers tangible benefits for system designers. Developers can reuse functionality, tailor, or even customize I/O to meet specific needs and offload preprocessing functions using the FPGA's capability. Because of the highly integrated designs that can be achieving utilizing a modular FPGA-based PMC approach, faster, denser systems can be designed with a reduced SWaP footprint. Robust FPGA development and debug toolsets, along with off-the-shelf, proven, optimized IP blocks, help reduce design cycle times. Designing around a configurable PMC, whether developed in-house or a COTS product, is an approach that designers should strongly consider.

Jeff Biviano *is a senior systems engineer at VMETRO. He served in a systems engineering role for five years at Transtech DSP before its acquisition by VMETRO in 2004. Jeff has a BS in Computer Engineering Technology from the Rochester Institute of Technology.*

Dave Barker *is VP of market development for embedded solutions at VMETRO. Before joining VMETRO in 2005, Dave was marketing manager for VME products at the Motorola Computer Group and has worked in the industry for more than 25 years. Dave has a BS in Computer Science from the University of Pittsburgh and an MBA from the University of Phoenix.*

VMETRO, Inc.
1880 Dairy Ashford, Suite 400
Houston, TX 77077
281-584-0728
jbiviano@vmetro.com
dbarker@vmetro.com
www.vmetro.com

RTD Embedded Technologies, Inc.

"MIL Value for COTS prices"™

Geode cpuModules™

Pentium® M cpuModules™

8000 MIPS dspModules™

cpuModules™ −40 to +85°C

		Pentium® M				Intel® Celeron®						AMD Geode		
		CMX158886PX1400HR	CMD158886PX1400HR	CMX158886PX1400HR-BRG	CMD158886PX1400HR-BRG	CME147786CX400HR	CME147786CX650HR	CML147786CX400HR	CML147786CX650HR	CMX147786CX400HR	CMX147786CX650HR	CME26686CX333HR	CME27686CX333HR	CMC26686CX333HR
Bus	AT Expansion Bus			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PCI Universal Expansion Bus	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
	PCI Bus Masters	4	4	4	4	4	4	4	4	4	4		4	
	APIC (add'l PCI interrupts)	9	9	9	9	9	9	9	9	9	9			
CPU and BIOS	CPU Max Clock Rate (MHz)	1400	1400	1400	1400	400	650	400	650	400	650	333	333	333
	L2 Cache	2MB	2MB	2MB	2MB	256k	256k	256k	256k	256k	256k	16K	16k	16k
	Intel SpeedStep Technology	✓	✓	✓	✓									
	ACPI Power Mgmt	2.0	2.0	2.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0			
	Max Onboard DRAM (MB)	512	512	512	512	512	512	512	512	512	512	256	256	256
	RTD Enhanced Flash BIOS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Nonvolatile Configuration	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Quick Boot Option Installed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	USB Boot	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Peripherals	Watchdog Timer & RTC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IDE and Floppy Controllers	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ATA/IDE Disk Socket, 32 DIP	1	1	1	1	1	1	1	1	1	1	1	1	1
	Audio	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
	Digital Video	LVDS	LVDS	LVDS	LVDS			TTL	TTL	LVDS	LVDS	TTL	TTL	
	Analog Video	SVGA	SVGA	SVGA	SVGA	SVGA	SVGA	SVGA	SVGA	SVGA	SVGA	SVGA	SVGA	
	AT Keyboard/Utility Port	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PS/2 Mouse	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	USB Mouse/Keyboard	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
I/O	RS-232/422/485 Ports	2	1	2	1	2	2	2	2	2	2	2	2	2
	USB 2.0 Ports	2	4	2	4									
	USB Ports					2	2	2	2	2	2	2	2	2
	10/100Base-T Ethernet	1		1		1	1	1	1	1	1	1	1	
	ECP Parallel Port	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	aDIO(Advanced Digital I/O)	18	18	18	18	18	18	18	18	18	18	18	18	18
	multiPort(aDIO, ECP, FDC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SW	ROM-DOS Installed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	DOS, Windows, Linux	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

utilityModules™ −40 to +85°C

dspModules™
- Coprocessors
- Accelerators

Specialty I/O
- Pulse width modulator
- Incremental encoder
- Opto-isolated MOSFET

Frame Grabbers
- Single or multi-channel
- MPEG-2 compression

Video Controllers
- Analog VGA
- TTL and DVI panels

Communication Modules
- Copper or fiber Ethernet
- USB 2.0 and Firewire
- CAN Bus & CAN Spider
- Dual Synchronous Serial
- Quad Serial w/ Ethernet
- Octal PCI Serial

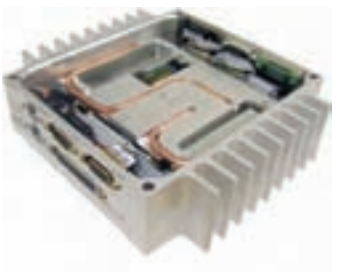
Wireless Telematics
- GSM, GSM-R, CDMA
- EDGE, GPRS, SMS
- GPS, Wi-Fi, Bluetooth

Motion Controllers
- DC motor controllers
- Synchro, resolver, LVDT

Power Supplies
- 50/75/83/88/100 Watts
- Wide input range
- ATX Power Supply
- UPS backup
- MIL-STD-704/461

Mass Storage
- 1.8/2.5" IDE & PCMCIA
- CompactFlash

IDAN™— Intelligent Data Acquisition Node
- Easily build your PC/104 system
- Rugged PC/104 stackable framed modules
- Quick interchangeability and expansion
- Structural heat sinks and heat pipes
- Optional cooling fins
- Milled aluminum frames
- Standard PC connectors
- Optional MIL-SPEC paint & shock mounts
- −40 to +85 °C



Full Product Line and Pricing Online

A Founder of the PC/104 Consortium • ISO9001:2001 Certified

HighRel PC/PCI-104 Modules and Systems

–40 to +85°C

Autonomous SmartCal™

Wireless Telematics

Frame Grabbers

dataModules® –40 to +85°C		Smart A/D		Analog I/O					Digital I/O						
		SDM7540HR	SDM8540HR	DM6210HR	DM6420HR	DM6430HR	DM7520HR	DM6620HR	DM6812HR	DM6814/16HR	DM6856HR	DM6888HR	DM6956HR	DM7820HR
Bus	AT Expansion Bus	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PCI Expansion Bus Master	✓	✓				✓							✓
	McBSP Serial Ports	✓	✓				✓							
Analog Input	Single-Ended Inputs	16	16	16	16	16	16							
	Differential Inputs	8	8		8	8	8							
	Max Throughput (kHz)	1250	1250	40	500	100	1250							
	Max Resolution (bits)	12	12	12	12	16	12							
	Input Ranges/Gains	3/7	3/7	3/1	3/4	1/4	3/6							
	Autonomous SmartCal	✓	✓											
	Data Marker Inputs	3	3		3		3							
Conversions	Channel-Gain Table	8k	8k		8k	8k	8k							
	Scan/Burst/Multi-Burst	✓	✓		✓	✓	✓							
	A/D FIFO Buffer	8k	8k		8k	8k	8k							
	Sample Counter	✓	✓		✓	✓	✓							
	DMA or PCI Bus Master	✓	✓		✓	✓	✓	✓						✓
	SyncBus	✓	✓				✓							
Digital I/O	Total Digital I/O	16	16	16	16	16	16	16	48	18/9	32	64	32	48
	Bit Programmable I/O	8	8		8	8	8	8	24	6/0				48
	Advanced Interrupts	2	2		2	2	2	2	2					2
	Input FIFO Buffer	8k	8k		8k	8k	8k							4M
	Opto-Isolated Inputs										16	48	16	
	Opto-Isolated Outputs										16	16		
	User Timer/Counters	3	3	3	2	3	3	3	3	3				10
	External Trigger	✓	✓		✓	✓	✓	✓	✓					✓
	Incr. Encoder/PWM									3/9				
	Relay Outputs												16	
Analog Out	Analog Outputs	2	2		2	2	2	4						
	Max Throughput (kHz)	200	200		200	100	200	200						
	Resolution (bits)	12	12		12	16	12	12						
	Output Ranges	4	4		3	1	4	4						
	D/A FIFO Buffer	8k	8k				8k	8k						

RTD FieldPads™

- Ruggedized, embedded computer systems
- User-specified CPU and PC/PCI-104 expansion
- Weathertight components
- Integrated 6.5-inch video panel, keyboard
- Heat pipes for high performance CPUs
- User-defined MIL connectors
- Internal and external battery packs
- Expand with any RTD PC/PCI-104 product

Industrial FieldPad™

Ideal for control and monitoring of processes on factory floors or industrial installations. Mounting flanges allow the unit to be installed on machinery or walls, enabling standard PC access in a rugged enclosure resistant to industrial environments.

Tactical FieldPad™

Designed for mobile and portable applications where the angled panel and ergonomic design allow for optimal viewing with flexible positioning. Data collection/downloading and information access accomplished through wired or wireless connections.

HiDAN™ and HiDAN*plus*™— HighRel Intelligent Data Acquisition Node

- HiDAN is a rugged, watertight enclosure for a stack of PC/104 modules
- HiDAN*plus* combines the modularity of IDAN with the environmental ruggedness of HiDAN
- Integrated tongue and groove O-ring for environmental sealing and EMI suppression
- Structural heat sinks and heat pipes
- Optional cooling fins
- Milled aluminum frames
- **Stackable signal raceway**
- Optional MIL-SPEC paint
- MIL I/O connectors
- Shock-mount optional
- **–40** to **+85** °C

www.rtd.com
Specifications, manuals, drivers, and plant tour

RTD Embedded Technologies, Inc.
103 Innovation Blvd • State College, PA 16803
T: 814-234-8087 • F: 814-234-5218

"Accessing the Analog World"®

Hardware-based solution aides:
Design assurance for airborne systems

By Irene Sysenko, PhD, and Ravi Pragasam

Photo courtesy of U.S. Air Force

As airborne systems increase in complexity to take advantage of the benefits that the latest technology can offer, the arduous task of validating the design for assurance per specifications set forth by governing bodies becomes monumental. Software tools offer limited features and do not always offer a solution that meets the required specification. A hardware solution in combination with software offers a more complete solution and helps to speed up the verification and validation process.

Today, in the era of multimillion gate designs, reusable IP, and Systems-on-Chip (SoCs), verification is considered the largest bottleneck in the design assurance process. Simultaneously, time to market continues to shrink as avionics designs have the same timing pressures as commercial applications. The avionics industry faces a test/cost problem where verification and testing represent a large part of the development expenses. In order to address the lack of complete verification, the hardware design life cycle should include new technologies that can help accelerate the design assurance process.

Three methodologies – hardware emulators, hardware accelerators, and FPGA prototypes – have emerged to the forefront to provide the highest performance of complete verification methodologies in the industry. As the level of design assurance increases to accommodate new industry guidelines, design verification approaches require special methods and tools. COTS hardware-software platforms directly address these challenges.

Hardware design assurance

More and more frequently, industry guidelines drive how systems are designed and verified. In the avionics industry, for example, the DO-254 document provides design assurance guidance for the development of "safe" airborne electronic hardware. The process involves multiple steps. First, system functions must be allocated to different hardware blocks at the system level. These blocks are then assigned corresponding system development assurance levels.

Figure 1 illustrates the relationships and interactions between three processes – hardware, software, and safety assessments – as a system requirement may result in the design assurance of multiple complex processes. For example, a hardware function that contains safety requirements involves both the safety assessment and the hardware design life-cycle processes.

It is important to note that the hardware design assurance level is associated with the five levels of safety criticality: Levels A through E (see sidebar, page 28). The DO-254 document notes that the hardware design assurance process and safety assess-

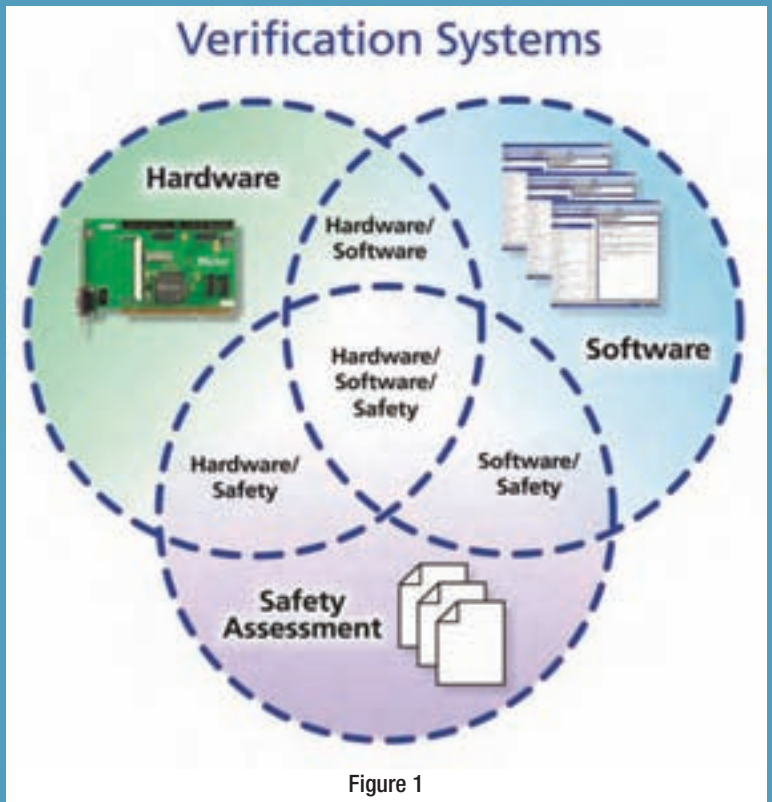


Figure 1

Elcard™ Wireless LAN Modules

Designed for Industrial and Professional Applications

PC/104+ WIB4xx dual WLAN module

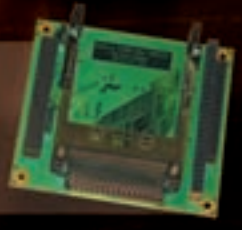
USB-attached WLAN

- PC/104+, USB, and PCI versions
- IEEE 802.11b/g/a/h WLAN standards
- At 2.4GHz up to 11 and 54/108Mbps bandwidths
- At 5 GHz up to 54/108Mbps bandwidths
- Dual antenna diversity
- Extended temp versions available (-40°C to 85°C and -20°C to 70°C operating)
- Rugged and shock resistant, high altitude operation
- Long term supply
- O/S support for Linux, Microsoft™ Windows™ XP/2000/NT/98SE/ME

Rugged Access Points Available

- Dual WLAN versions available (WIB400 series)
- Evaluation kits for easy start-up
- Ranges of 1 mile+ can be reached even at 100mW Tx power with our directional antennas
- Ranges of several miles can be reached with our power amps and special antennas
- WIB250 WLAN module provides dual band 802.11g/ 2.4GHz & 802.11a/5GHz with two antenna connectors

-40°C to +85°C Operating Temperature Range Versions Available



AIB220 PC/104+ Cardbus

- Cardbus/PCMCIA Adapter
- Dual Type I/II or single Type III
- Linux and Win9x/2K/XP support
- 3.3V and 5V card support
- TI PCI1420 chipset

Elcard

Elcard USA
10849 Kinghurst, Suite 105
Houston, Texas 77099
Toll Free: 800-688-4405
Phone: 281-568-4744
Fax: 281-568-4604
Email: sales@elcard-usa.com
Web: www.elcard-usa.com

Hardware Design Assurance Levels

Level A. Failure Condition Classification – *Catastrophic.* Hardware function whose failure or anomalous behavior would cause a failure of system function resulting in a catastrophic failure condition. Level A is most critical, with a classification for failure condition of "catastrophic," for example, "failure conditions that would prevent continued safe flight and landing." While effects on occupants are not defined for this level, fatal injury to many of the occupants would probably result.
Level B. Failure Condition Classification – *Hazardous/Severe-Major.* Hardware function whose failure or anomalous behavior would cause a failure of system function resulting in a hazardous/severe-major failure condition.
Level C. Failure Condition Classification – *Major.* Hardware function whose failure or anomalous behavior would cause a failure of system function resulting in a major failure condition.
Level D. Failure Condition Classification – *Minor.* Hardware function whose failure or anomalous behavior would cause a failure of system function resulting in a minor failure condition.
Level E. Failure Condition Classification – *No effect.* Hardware function whose failure or anomalous behavior would cause a failure of system function with no effect on operational capability.

ment should jointly determine compliance. The designated level for each function should demonstrate that an acceptable level of design assurance has been achieved.

Verification plan and design flow

Simulation can be used to analyze the impact of hardware parameter variations, thereby reducing design errors that can compromise safety and thus build an overall confidence in the design. Per the DO-254 specification, simulation is an important design analysis tool, both for design operation visualization and high-level functional verification.

The Aldec-Actel solution offers three stages of design verification support: RTL simulation to verify design functionality, gate-level simulation to check timing and catch errors introduced during synthesis and place and route, and verification of the design in hardware. For all three simulation runs, designers can utilize the same test bench or a set of *golden vectors* as shown in Figure 2.

The results of each simulation stage are stored, compared, and verified. The resulting data then establishes functional completeness and correctness of the hardware design. The verification process is complete if all simulation stages confirm the same data. If this occurs, the verification process and data can be included in the DO-254 certification documents such as the Hardware Verification Plan and Hardware Verification Data.

RTL simulation

At the RTL simulation level, an extensive test suite that consists of different test benches is developed to support all aspects of the design specification. In many cases, the same test bench can be used with different input data to simulate design functionality. Test benches can be based on traditional HDL languages, such as Verilog or VHDL, and assertion checkers need to be included. All simulation activity must be performed within the logic simulator environment. One means of design checking involves the use of simulation results as a waveform file. Subsequently, the waveform file can be utilized to compare results from other levels of design verification. After successful verification, design sources can be synthesized to receive the post-synthesis netlist.

Gate-level simulation

To achieve gate-level simulation, the aforementioned test bench suite can be used again. As before, all verification activity is done using a software simulator. All gate-level, nontiming simulation results are compared against the RTL simulation using waveform files. Since automatic waveform comparison is not applicable for timing simulation, special checkpoints are put in place to compare corresponding simulation results.

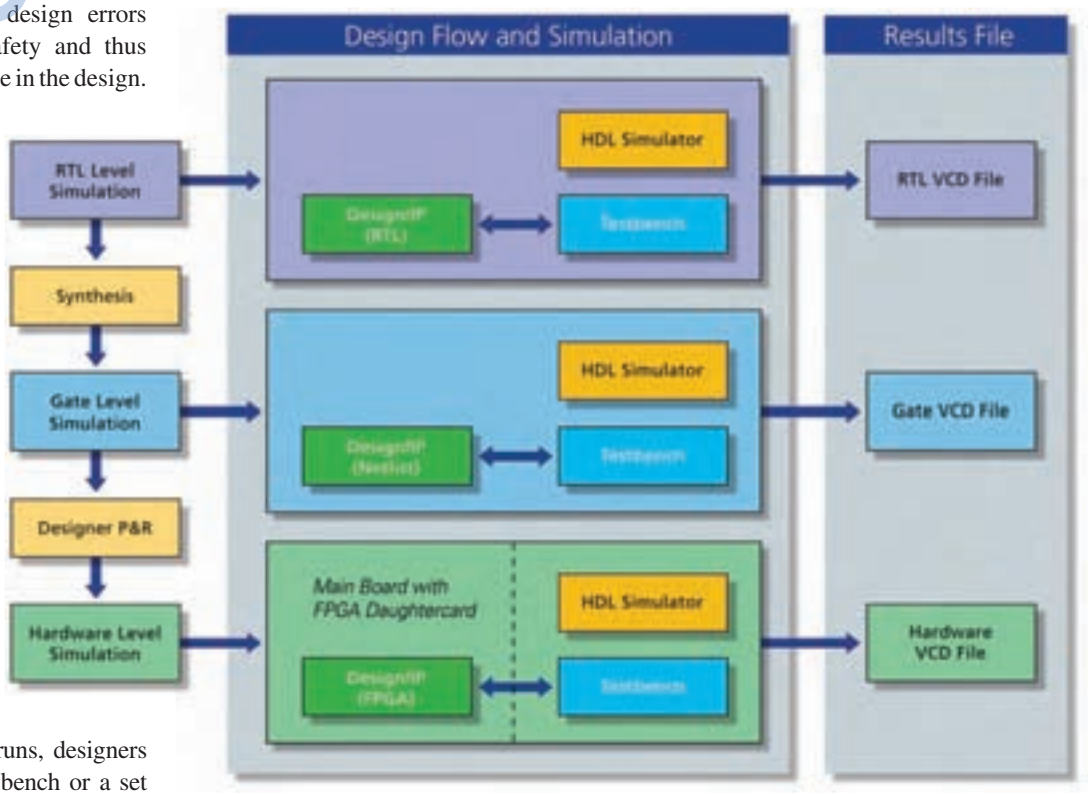


Figure 2

Hardware-level simulation

Hardware simulation is used to accomplish two major goals. The first is the achievement of functional design verification in real hardware. Because simulation performance is an order of magnitude faster in hardware than in software, the second goal is validation of the design by running more test cases.

Using Hardware Embedded Simulation (HES) technology, the designer can start design verification in the actual hardware immediately following initial RTL validation. HES is a hybrid software-hardware simulation platform that is driven by software that functions as a design test bench executed by a logic simulator. HES technology is designed to implement the Design Under Test (DUT) in a reconfigurable hardware device, such as an FPGA, and then verify that the design functions in actual hardware.

The FPGA communicates on an event basis with the remaining sections of the test bench resident in the logic simulator. The HES technology then utilizes the same set of test benches and generates a signal waveform file that will need to match the one from RTL simulation. Another advantage of HES simulation is that the DUT interface signal data can be stored and used later as simulation input vectors. Recording both the input and output interface data allows not only DUT testing but also comparison of the output with the set of golden vectors.

Design prototyping

The design prototyping phase proves the logic operates as close to or equal to the application's target speed. As it is one of the final stages of the design verification, this phase requires the most resources to set up and execute.

FPGAs are ideal for this design phase and offer the benefits of programmability that can accommodate design changes with little impact to the design verification phase. Two types of challenges can be encountered at this stage. The first challenge is the process of setting up the design in the hardware target, such as reprogrammable flash-based FPGAs. These single-chip solutions offer several advantages that make them ideal for design assurance. Unlike an SRAM FPGA with an external boot ROM, flash-based FPGAs are highly secure and immune to neutrons. Alternatively, SRAM FPGAs are very susceptible to neutrons and cause configuration upsets – not well-matched for stringent DO-254 requirements. Configuration upsets in SRAM FPGAs can cause the functionality of the FPGA to change. This demonstrates poor design assurance, but more importantly, these changes can result in low safety assessments and grave consequences, particularly in the avionics industry.

The design is first targeted to a high-density FPGA for initial prototyping. If a smaller size device is used, the design may have to be partitioned across multiple FPGAs using commercially available software, adding a layer of complexity to the design assurance process. Initial testing of the prototype FPGA can be accomplished using the test vectors generated during HES simulation as inputs. Special input/output interface logic has to be designed into the FPGA to ensure the synchronization of the hardware with the generated output from the software application. FIFO memory blocks onboard the FPGA can be used as the input/output interface using two interrupt lines. A request is then made to the software application to fill in the input buffer or to empty an output buffer. The output vectors generated are converted into a standard waveform format and used to compare the results with the outputs from earlier verification stages. To achieve further testing after debugging the FPGA under test, the design can be exercised further with different test vectors or real-time data.

DO-254 verification made easier

Along with traditional design verification methodologies, such as RTL and gate-level simulation, HES can be used successfully to test the design in hardware, using a simulator and then later using a real-time high-speed clock. The delta cycle accurate behavior can be validated using off-the-shelf standard tools, such as a waveform viewer or list viewer.

These methods allow the customer to meet the hardware verification plan specified in design assurance guidance for airborne electronic hardware. There are procedures, methods, and standards to be applied to achieve hardware verification compliance with a set of established standards such as DO-254. The Aldec-Actel Design Assurance HES and prototyping methodology for verification and validation of airborne systems provides a complete solution to meet airborne system safety requirements as specified by the DO-254 objectives.

Irene Sysenko *has worked as a research engineer at Aldec Inc. since 2005. Prior to that, she spent two years at Aldec, Inc. working as an R&D engineer, designing processor-based simulation systems. Before joining Aldec, Irene worked as a professor at Kharkov National University of Radio-Electronics, Ukraine, teaching digital design testing and diagnostics. She holds a PhD in Computer Engineering from Kharkov National University of Radio-Electronics.*

Aldec, Inc.
2260 Corporate Circle
Henderson, NV, 89074
702-990-4400, Ext. 254
irenes@aldec.com
www.aldec.com

Ravi Pragasam *is a senior marketing manager in the military and aerospace product marketing team at Actel Corporation. He has worked in the FPGA industry for more than 15 years and has served in a variety of roles including marketing and applications engineering. He has two MS degrees in Electrical and Computer Engineering from Kansas State University and College of Engineering, India.*

Actel Corporation
2061 Stierlin Court
Mt. View, CA 94043-4655
650-318-4773
ravi.pragasam@actel.com
www.actel.com

MILS: Protecting our most vital systems

By Rance J. DeLong

Photo courtesy of U.S. Army

Military systems increasingly involve safety- and security-critical matters, requiring a new approach to designing high-threat, high-asset-value systems. The emerging Multiple Independent Levels of Security (MILS) paradigm provides a modular, flexible, and trustworthy foundation for national security and critical infrastructure.

Commercial products and approaches to security do not meet the requirements of contemporary network-centric "systems of systems" in high-threat, high-asset-value environments. The MILS paradigm[1] – based on a new breed of commercial security products including high-assurance separation kernels and middleware – applies venerable, established principles alongside recent advancements in microprocessors, computer security, avionics safety, software engineering, and formal methods.

Critical trade-offs

The unrelenting growth of embedded controls, information processing, and communications in military systems has caused a massive spike in demand for computer power in deployed systems. The challenge of meeting such demands within the feasible constraints of Space, Weight, and Power (SWaP) will never be fully resolved, but consolidating functions on powerful microprocessors helps ease SWaP constraints.

The C4ISR trend is gravitating toward increasing connectivity and increasing need for controlled sharing; coalitions are formed, redefined, and dissolved. Information must be shared and analyzed at speeds dictated by tactical constraints. Systems must protect valuable information assets and be robust against serious threats.

Systems increasingly involve safety- and security-critical considerations. Military vehicles, ships, and aircraft serve as weapons platforms and intelligence conduits, and onboard computer systems are becoming more and more integral to vehicle operation. Diverse requirements for different kinds of systems are being streamlined into combined requirements that must be met by a single system.

Safety and security background

Well-established yet distinct traditions for the construction of dependable safety- and security-critical systems require a degree of assurance that far surpasses what "best commercial practice" provides.

The safety of commercial airborne systems is subject to SAE Aerospace Recommended Practice as interpreted by RTCA DO-178B[2] requirements, and corresponding safety standards exist for military aircraft. The DO-178B Level A (the most stringent level) can be characterized as technically conservative because it applies conventional process and testing practices, albeit very thoroughly and conscientiously. DO-178B does provide an escape clause for alternative methods, such as formal methods, as long as they achieve the same objectives as the ones prescribed. This is rumored to be much more explicit in the awaited DO-178C, scheduled for release in late 2008.

The development history of high-assurance secure systems is long, if sparse, and can be characterized as technically progressive because it has applied the best available methods. Confidence in the trustworthiness of secure systems has typically been sought through formal methods. In fact, security and formal methods grew up together in the 1970s when much advancement in formal methods was motivated – and funded – by security projects.

Since the 1960s, security projects have recognized the need for security to be designed and implemented at the low-

est levels: the operating system and the hardware mechanisms that support it. Representatives of early secure operating system developments include ADEPT-50, the Multics security enhancements, the UCLA Data Secure Unix Kernel, the Kernelized Secure Operating System (KSOS), the Secure Communications Processor (SCOMP), the Provably Secure Operating System (PSOS), Multinet Gateway, BLACKER, the Boeing MLS LAN, and GEMSOS.

These systems often incorporated the security policy-enforcing mechanisms in a security kernel, typically a general purpose, heavyweight operating system. The policy enforced was a mandatory access control policy, usually a version of the Bell-LaPadula Model (BLP)[3], also known as Multi-Level Security (MLS). BLP attempts to formally describe the familiar practice of classifying information, assigning clearances to individuals, and granting or denying access on the basis of classification, clearance, and mode of access.

One limitation of these systems is that practical, operational considerations lead to the need for trusted processes that require special privileges granted by the security kernel in order to perform their functions. The security kernel taken with such non-kernel security-related software comprised the Trusted Computing Base (TCB).

Assurance background

The primary barrier to providing a convincing argument about the trustworthiness of a TCB is complexity. The security kernel and other TCB components typically comprise large, complex, monolithic objects. To perform rigorous and complete analysis of such objects using formal methods was beyond state-of-the-art 25 years ago and is arguably so even today. Rigorous and complete analysis or formal methods refer to using specifications written in languages that have formal semantics and an associated proof system so that analysis and proofs are automated (or human directed), objective, repeatable, and logically sound. Such methods can only be applied to well-structured objects of limited complexity.

Consequently, one of the tenets of MILS is to decompose a system into a collection of reusable components, each of which is small enough to be rigorously analyzed for correctness and/or security properties.

Because MILS is intended to meet both safety and security standards, it would be tempting to apply all of the processes recommended for security and safety standards. This, however, would result in an excessive burden and cost of process. Rather than applying the union of the processes, defining a single process that would satisfy the union of the two standards' objectives is recommended.

The National Computer Security Center (NCSC), formed at the NSA in 1981, published the Trusted Computer System Evaluation Criteria (TCSEC)[4] to provide a standard for the requirements for secure systems and the measurement of systems intended to meet those requirements. The standards represented in the TCSEC evolved through a series of renditions including the Federal Criteria in the United

States and the Information Technology Security Evaluation Criteria in the United Kingdom and Europe. Appearing first in 1996, since 1998 the Common Criteria (CC)[5] has provided a broadly accepted international standard (ISO/IEC 15408). The TCSEC (a.k.a. Orange Book) designated systems according to D, C1, C2, B1, B2, B3, and A1. The CC designates systems according to Evaluation Assurance Levels (EAL) 1 through 7.

Precursors to MILS

Spurred by the NCSC and TCSEC, a host of computer vendors commercially produced trusted MLS operating systems from the mid-1980s through the 1990s. These systems are, for the most part, only medium assurance, that is, B1 according to the TCSEC or EAL 4 according to the CC. Ironically, since such systems can only be evaluated to medium assurance, they do not meet accreditation requirements to be deployed in the environments where they would actually be used to protect and separate classified data. Instead, EAL 5 through EAL 7 are required, depending upon the threat environment and the value of the assets.

Microkernels date back to the 1980s, originally serving as the basis for early experiments in factoring operating system functionality into a minimal kernel supporting highly modular services. They typically performed poorly compared to monolithic operating systems on the microprocessors of the day.

Virtual Machine Monitors (VMMs) date back to the 1972 IBM VM/370. Traditionally, a VMM creates a virtual environment indistinguishable from the bare hardware an operating system may run on without modification. A VMM is not a separation kernel, and vice versa. A VMM enforces a policy of isolation, while a separation kernel additionally enforces a policy of information flow control. A separation kernel could be constructed with VMM properties, provided appropriate hardware support is available.

Enter MILS: the separation kernel and MILS middleware

A separation kernel, first proposed by John Rushby[6], program director for formal methods and dependable systems at SRI International, works with the protection mechanisms provided by the underlying microprocessor hardware to enforce with a very high degree of assurance the primitive policies of isolation and information flow control – the prerequisite guarantees needed for the construction of software reference validation mechanisms that enforce higher-level policies such as MLS. The MILS paradigm depends explicitly on Saltzer's and Schroeder's[7] Principle of Least Privilege and Principle of Complete Mediation, enforced within the separation kernel and supported by the separation kernel for higher levels of the system design. The security requirements for a separation kernel are set forth in the Separation Kernel Protection Profile (SKPP)[8].

When the separation kernel was first conceived, microprocessor features and performance were not adequate to implement complex systems while paying the security tax for robust isolation provided by a separation kernel. As recently as 10 years ago, 10,000 partition switches per second would have left little, if any, of a processor's cycles available for applications. Today, a processor can perform 60,000 partition switches per second and still have more than 95 percent of its cycles available to applications. Quantitative improvement in processor speed has enabled MILS' qualitatively different approach to security, putting separation kernels squarely in the sweet spot of the perennial performance/security trade-off.

Most of the services provided by conventional operating systems are pushed out of the separation kernel into other high-assurance components referred to as MILS middleware.

The separation kernel and MILS middleware subsystems must be sufficiently simple to enable rigorous analysis of the properties of each. The MILS paradigm calls for each high-assurance subsystem to be decomposed into as many elements as necessary to facilitate that analysis by delineating the role and constraints on each element. Complex high-assurance systems can, in turn, be constructed from MILS components by building upon the functionality and security properties of those components. Figure 1 shows an MLS system constructed in the MILS style.

LynxSecure Separation Kernel

LynuxWorks, in collaboration with SRI International, is developing a high-assurance separation kernel and an integrated formal development approach for MILS systems. The project aims to provide a high-assurance integrated development environment that will enable experienced engineers, though not experts in formal methods, to use this secure separation kernel to develop high-assurance products and systems. The companies are

MILS-style MLS System Stack

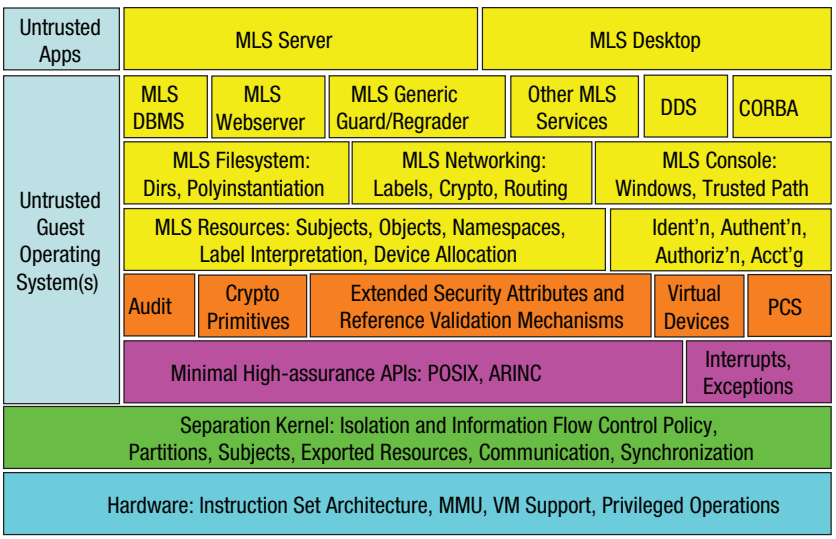


Figure 1

also helping lead the MILS community through The Open Group's Real-Time Embedded Systems (RTES) Forum (www.opengroup.org/rtforum).

The LynxSecure Separation Kernel (Figure 2) fully implements the SKPP and will be certified at the highest levels of security and safety: CC EAL 7+ and DO-178B Level A. Using Intel Virtualization Technology as a platform for its first release, this kernel will create virtual machines able to run heavyweight operating systems such as Microsoft Windows as guest operating systems without modification. Hardware virtualization support, now appearing in commodity microprocessors, makes it possible to provide virtual machines with a minimum performance impact.

The LynxSecure Separation Kernel also includes a high-assurance runtime interface, a lightweight guest operating system with a simple, formally specified and verified API that facilitates the construction of high-assurance applications.

Unprecedented vendor cooperation

One surprising outcome of the MILS initiative is that it has encouraged cooperation and collaboration from competitors that usually go out of their way to avoid each other. Because MILS components don't come from a single source, each part is provided by a company specializing in that particular technology. This requires an extraordinary level of cooperation among competitors to achieve products that can not only interoperate but do so securely.

As a testament to this cooperation, four MILS component vendors and a major system integrator presented an integrated live demonstration of MILS capability and interoperability at the meeting of The Open Group in Washington, D.C. in April 2006.

Within The Open Group's RTES Forum, the MILS community is developing a coherent set of community standards in the form of protection profiles based on the Common Criteria (CC). There, MILS stakeholders are working in concert to achieve a common vision of the MILS architecture and to streamline the process of developing the many needed protection profiles and other standards.

Future of MILS

The MILS effort is breaking new ground in the area of high-assurance security and functional composition. Each part of a high-assurance system must meet a precise set of constraints for the whole to meet system-level security requirements and functionality. The RTES Forum is working toward a unifying MILS integration framework.

The military's Global Information Grid (GIG) envisioned by strategic planners levies challenging requirements for information assurance: a dynamic, highly connected environment that demands unprecedented robustness, flexibility, and agility. The degree of security robustness required in a dynamic, highly connected network architecture cannot be overestimated. MILS promises to provide this kind of assurance and flexibility at every level, from sensor and control processors to communications devices to workstations and servers.

MILS shows potential for meeting the requirements of high-assurance security with new, high-assurance commercial products. MILS components exhibit the robustness needed for key security-enforcing GIG mechanisms.

Some significant defense programs have already committed to MILS, and

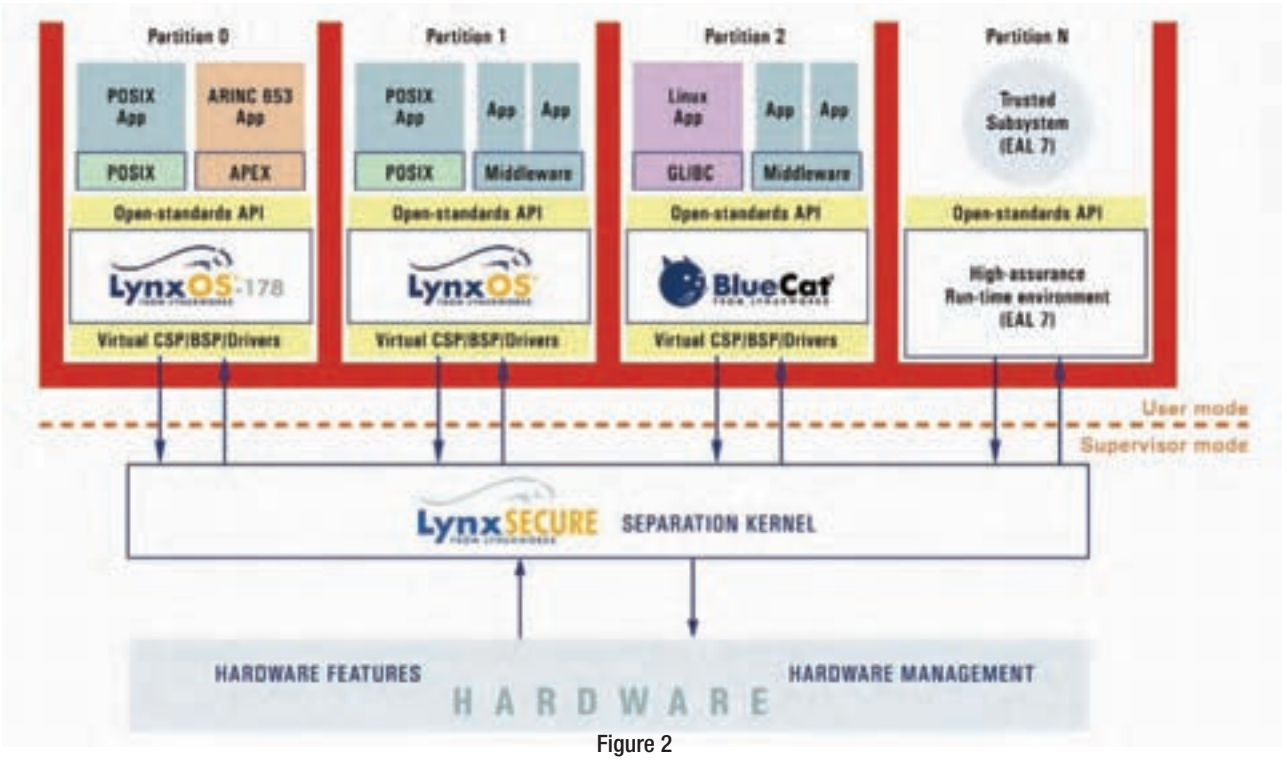


Figure 2

many others are prime candidates. MILS component vendors are broadcasting the message that they are committed and that MILS technology is coming.

Acknowledgments

The author acknowledges colleagues Ed Mooring of LynuxWorks and John Rushby of SRI International for providing the critical mass of experience and know-how that has been applied to this work on the LynxSecure Separation Kernel and the methodology and tools for high-assurance development.

References

1. W. M. Vanfleet, R. W. Beckwith, B. Calloni, J. A. Luke, C. Taylor, and G. Uchenick. MILS: architecture for high assurance embedded computing. CrossTalk, 18:12–16, August 2005.
2. Requirements and Technical Concepts for Aviation, Washington, DC. DO-178B: Software Considerations in Airborne Systems and Equipment Certification, December 1992.
3. D. E. Bell and L. J. LaPadula. Secure computer system: Unified exposition and Multics interpretation. Technical Report ESD-TR-75-306, Mitre Corporation, Bedford, MA, March 1975.
4. Department of Defense. Department of Defense Trusted Computer System Evaluation Criteria, December 1985. DOD 5200.28-STD.
5. Common Criteria for Information Technology Security Evaluation, September 2006. Version 3.1, CCMB-2006-09-001, 002, 003.
6. John Rushby. The design and verification of secure systems. In Eighth ACM Symposium on Operating System Principles, pages 12–21, December 1981.
7. J. H. Saltzer and M. D. Schroeder. The protection of information in computer systems. In Proceedings of the IEEE, volume 63, pages 1278–1308, September 1975.
8. Information Assurance Directorate, National Security Agency. U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness, October 2006. Version 1.1.

Rance J. DeLong *is staff scientist for security and assurance at LynuxWorks, Inc., and an adjunct lecturer at the Center for Advanced Study and Practice of Information Assurance at Santa Clara University. He has 28 years of security product development experience including the Kernelized Secure Operating System, the Provably Secure Operating System, and Sun Microsystems' Trusted Solaris. Rance has a BS in Physics/Mathematics, a BA in Philosophy from Moravian College, and has completed extensive postgraduate work at Lehigh and Stanford universities.*

LynuxWorks, Inc.
855 Embedded Way
San Jose, CA 95138
408-979-3900
www.lynuxworks.com

Annapolis Micro Systems

The FPGA Systems Performance Leader!

FOPEN **Radar Systems** **Software Defined Radio** **FLIR**
SIGINT **ELINT** **Digital Receivers** **Recording Systems**

Annapolis
Micro Systems

High Performance Signal Processing in Scalable FPGA Computing

Above and Beyond -----

FPGA Acceleration

190 Admiral Cochrane Drive, Suite 130, Annapolis, Maryland 21401
wfinfo@annapmicro.com (410) 841-2514 www.annapmicro.com

Photo courtesy of USN

Multilevel security in tightly coupled military systems: Virtualization as a path to MLS

By Diana L. Hecht, PhD, and Warren A. Rosen, PhD

The authors describe the similarities between virtualization and Multi-Level Security (MLS) systems and provide an example of how both can be supported in a COTS network protocol by placing security labels in network packet headers and specialized processing structures built into switches.

Security is a major cost driver in military systems and is of particular concern when using commercial network protocols in the military environment. MLS systems may be employed to reduce the amount of application software that must be secured and certified at the highest level, avoiding the expense and complexity associated with maintaining the entire system at the highest level of security. The major problem with MLS systems is that the limited demand for them (primarily military and government agencies) does not compensate for the high costs of development and certification. Virtualization, on the other hand, is rapidly becoming a standard technology capable of supporting security while also providing scalability, flexibility, increased utilization, lower cost, and availability for today's IT infrastructure.

It should be noted that MLS differs from Multiple Independent Levels of Security (MILS). MILS is a layered architecture consisting of a layer directly above the hardware known as the *separation kernel*. The next layer, directly above the kernel is known as *middleware* and runs in user mode and performs the tasks traditionally handled by the OS (memory allocation, I/O drivers, and so on). It also provides services to extend the scope of the separation kernel allowing for intersystem communication. The uppermost layer is where the user applications run.

We describe how virtualization can be used as a path to a multi-level secure system and how network security can be provided for such a system in a military environment. We first explain the similarities between MLS and virtualized systems and then, as an example, show how virtualization and security can be supported within the RapidIO protocol. We provide an extension to the existing RapidIO protocol through the overloading of existing fields in a particular packet header format in order to provide a security label designation.

Features of MLS and virtualization

Security is an increasing concern in military and other mission-critical computing environments. The major security risk that exists in current approaches to general purpose computing arises from sharing resources among the various processes/applications running in the computer. Typical military systems support a number of applications running concurrently that create or use information of different security levels or classifications. In such cases, it is extremely important to separate information/resources with different security levels and to prevent any leakage of information among the various application processes that are running at different security clearance levels (top secret, sensitive, unclassified, and so forth).

MLS systems and virtualized systems are similar in nature in terms of their goal of providing strong isolation of their supported components (application/memory partitions in the case of MLS systems and virtual machines in the case of virtualized systems). An MLS system is one that has system resources at more than one security level and is able to prevent users from accessing resources for which they lack authorization. Applications are compartmentalized so that they can be prevented from interacting with one another. Security labels or tags are assigned to resources (memory partition, application data) when they are created or allocated to indicate the security level required for access. The operating system uses the security label to allow only authorized processes to access the data or other resources.

Virtualization provides a layer of abstraction between the physical hardware and the operating system and/or user applications. It also allows multiple virtual machines, possibly with different operating systems, to run in isolation, side-by-side on the same physical machine. A Virtual Machine Manager (VMM) or hypervisor is a small kernel that resides just above the hardware level and manages the virtual machines and their access to system resources.

MLS systems and virtualized systems differ in the degree of isolation and containment they provide (Table 1). MLS systems can consolidate data of different sensitivities onto a single computer or system and regulate access to the data by ensuring that data of a certain sensitivity level can only be accessed by authorized users. In order to enforce appropriate user access to data, processes and their assigned resources are compartmentalized so that they can be prevented from interacting with one another. The entire system works under a single operating system that provides mechanisms for keeping processes of differing security levels from interacting or interfering with each other. Virtualized systems provide a framework under which a number of

Comparison of MLS and virtualization

	MLS	Virtualization
Similarities		
Goal	Allows multiple users with different security clearance to share physical machine but prevents unauthorized access to data or system resources	Allows optimum utilization of physical machine by switching between virtual machines and managing shared resources
Isolation	Provided by MLS operating system	Provided by virtual machine monitor
Differences		
Unit of isolation	Partitions within a machine	Between virtual machines
I/O separation	Makes use of security label	Makes use of Device-ID
Market	Primarily used in military systems	Becoming widespread in IT markets

Table 1

virtual machines (each with their own operating system and address space) can coexist on the same physical machine.

I/O (network interface) for MLS/ Virtualization

For the purposes of this article, we are interested in a method for providing isolation of I/O data (primarily traffic on the interconnection network of a distributed system). Both virtualized and MLS systems currently provide support for isolation of I/O data. Virtualized I/O mechanisms are beginning to emerge from vendors such as AMD and Intel. AMD's solution is to offer an I/O Memory Management Unit (IOMMU) that provides management of DMA accesses involving both access permission verification and physical-to-virtual address translation (based on which virtual machine the I/O device is assigned to). A Device-ID is used to access the data structures necessary to verify access rights to the data and to access the relevant address translation page tables. Intel's solution is based on a similar approach using the VT-d architecture and the Device-ID to access relevant data structures. Traditionally, MLS systems support isolation of network data via labeled networks. Security labels are placed in the network packets, and a monitor or guard mechanism is used in the network interfaces or switches to manage/restrict packet travel through the network and delivery to the destination.

One of the main differences between the domain protection provided in the I/O virtualization architecture and the protection required by the MLS implementation is that current virtualized systems control access to data based on the I/O device requesting access (using a Device-ID). Meanwhile, MLS systems control access to data based on the memory partition or process within a node that makes the request (using a security label). Since different partitions may have unique security levels, it is not sufficient for access to a partition in the destination node to be granted solely based on the device/node ID. The decision of whether to grant access to a partition must be based both on the device/node ID and the security level of the individual partition within the node that generated the request.

RapidIO and MLS/Virtualization

To show how virtualization can be used to provide multilevel security in a COTS network, we used the RapidIO protocol as an example. RapidIO is becoming increasingly popular in military systems due to its high throughput, low latency, small footprint, low power, robustness, and support for a wide variety of features such as encapsulation.

A promising approach to adding security within the RapidIO protocol is to place security labels in the Class of Service field of the Type-9 packet header. This provides switches with the information necessary to determine whether a packet should be routed through to the output port/link indicated by the destination address. The Type-9 packet format is the Data Streaming transaction format presented in the RapidIO Interconnect Specification Part 10: Data Streaming Logical Specification, which provides support for segmentation and reassembly, encapsulation, and Class of Service designation. Rydal's proposed MLS extensions encapsulate RapidIO Logical I/O layer packets inside a Type-9 packet, thereby providing support for security

in existing systems using basic I/O transactions described in the RapidIO Interconnect Specification Part 1: Input/Output Logical Specification (RapidIO.org).

When a process resident on a particular node requests a read/write access to a memory address located on another node, the RapidIO packet is created, and the security label assigned to the partition from which the request originated is placed in the Class of Service field in the packet header. The input port of the switch connected to an endpoint node receives the RapidIO packet and uses the destination ID to determine the security levels that are allowed at the destination node. The major data structure used for this task is the Node Partition Table, shown in Figure 1. The Node Partition Table is indexed by the destination ID, and each entry in the table is a 32-bit mask indicating all of the partition security levels that may access that node. This table is used by the switch processing logic to cross-check an incoming request of a particular security level with the allowable security levels for accesses to the destination node. If a read/write request is issued by a partition in the source node that does not have the authority to access the requested partition in the destination node, the request is not allowed to proceed through the switch but is instead handled as a security violation error condition.

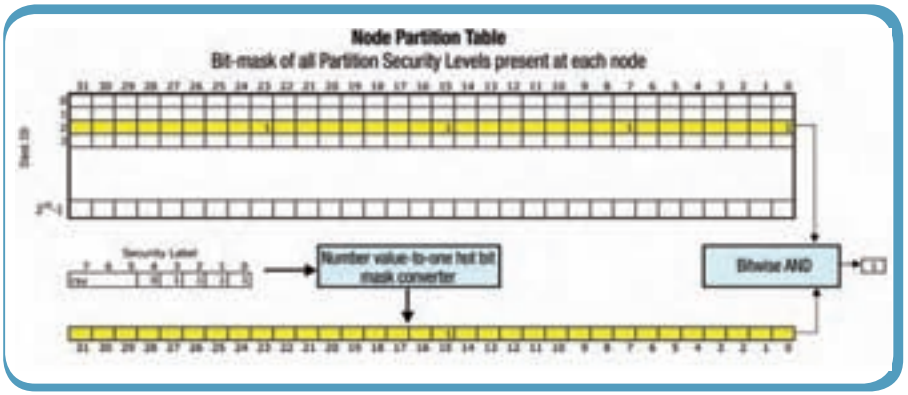


Figure 1

If source and destination nodes are trusted, the source node can be relied upon to correctly generate the security label from the memory address for the read/write operation and the destination node can be relied upon to correctly examine the data request, memory address, SourceID, and security level to verify that the access should be granted. If the OS of the destination node is not trusted, the switch must be able to determine the partition to which the memory address belongs and compare it to the security label in the switch before allowing the packet to pass through the link to the node's input port. Figure 2 illustrates the two cases.

To make use of the RapidIO-MLS extensions in one of the virtualized systems described earlier, the I/O virtualization architecture implementation (IOMMU) must receive as the Device-ID a descriptor that can be used to indicate the security level of the partition (rather than the device) that initiated the request. Support for MLS on current virtualized systems can be provided by passing the secu-

rity label to the IOMMU in place of a Device-ID. Providing a security label in the Type-9 packet header allows RapidIO the ability to support both MLS security and virtualization, thereby allowing a much larger number of applications to take advantage of the powerful features provided by RapidIO.

RapidIO provides path to support both MLS and virtualization

We have described the similarities and differences between MLS and virtualized systems and shown how virtualization and security can be supported within the RapidIO protocol. By building on the similarities between MLS and virtualization, a network can be designed to support systems based on an MLS operating system or virtualized systems. We have presented an example of one approach to this within the RapidIO network protocol. The changes to the protocol packet structure are minimal, providing a great deal of flexibility for any combination of traditional, secure, or virtualized system structure.

***Dr. Diana L. Hecht** is senior research engineer at Rydal Research and Development, Inc. Currently she works in the area of advanced network and signal-processing technology. Diana is involved in a number of research and development efforts aimed at high-performance signal processing for military applications and led the design team that developed Rydal's FPGA-based RapidIO switch. She has also participated in research involving network switch design sponsored by the Office of Naval Research through a subcontract from Rydal Research. Diana holds a PhD in Computer Engineering from Drexel University.*

***Dr. Warren A. Rosen,** president, founded Rydal Research and Development, Inc. in 1998 for the purpose of carrying out research and development of advanced networking and signal-processing technologies. Prior to that, he worked at the Naval Air Warfare Center, Aircraft Division in Warminster, Pennsylvania, where he established an optical communications laboratory for development and characterization of optical components, systems, and protocols for high-performance avionics data networks. He has conducted research sponsored by the National Security Agency, National Science Foundation, the National Oceanic and Atmospheric Administration, DARPA, the Office of Naval Research, and the Missile Defense Agency. He holds four U.S. patents in computer networking and signal processing. He earned his PhD in Physics from Temple University.*

Rydal Research and Development, Inc.
1523 Noble Road
Rydal, PA 19046
215-886-5678
diana@rydalresearch.com
warren@rydalresearch.com
www.rydalresearch.com

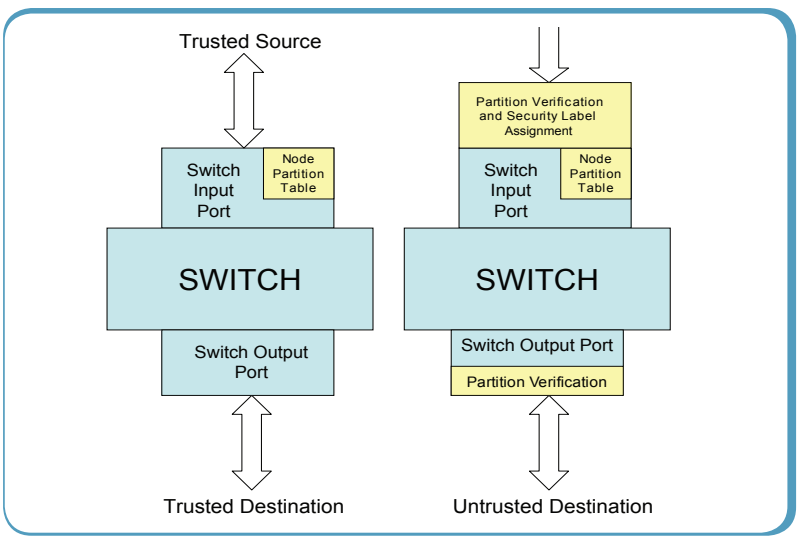
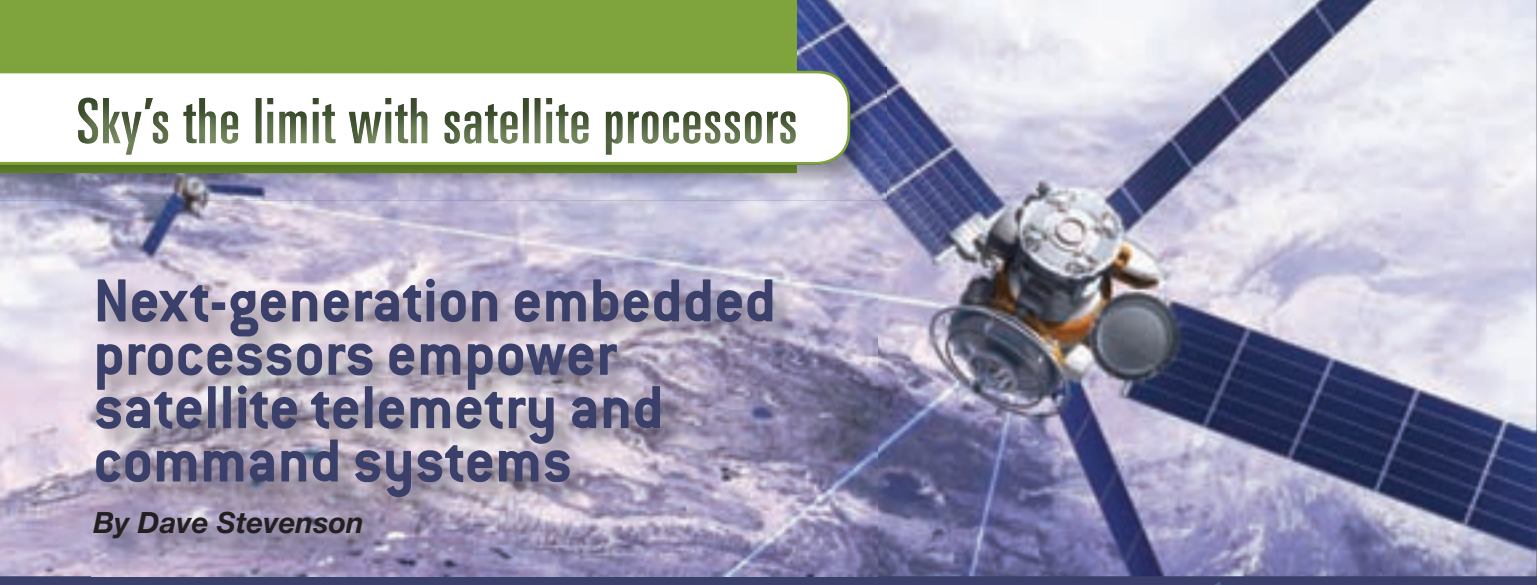


Figure 2



Next-generation embedded processors empower satellite telemetry and command systems

By Dave Stevenson

Future satellite applications will place a greater demand on telemetry and command systems for increased intelligence and performance. One method is an embedded processor card that can be used to implement a standard telemetry and command satellite system.

Among the challenges facing satellite manufacturers today are mechanical reliability, reduced power consumption, shrinking monetary budgets, and increased demands for on-satellite data processing. Among radiation-hardened electronics, fault-tolerant processors play a key role. Identifying a set of solutions – from mechanical chassis to digital electronics – that fits within a given set of specifications is a problem facing satellite designers and integrators. Modern technology is meeting the challenges of high-density space applications via radiation performance, power, mechanical weight and form factor, standardized software development tools, and next-generation System-on-Chip (SoC) designs.

A brief history of satellite evolution

In the early 1990s, NASA began outlining plans and initiating directives for a new, smaller spacecraft technology through smaller, faster, cheaper satellites. NASA recognized that smaller systems have some distinct advantages over large, costly spacecraft. A small spacecraft is pound-for-pound less expensive to produce and more tolerant of schedule and funding changes than a larger, more costly spacecraft. Small spacecraft are also less dependent on space-shuttle-sized launch vehicles to achieve orbit.

One key factor driving NASA's smaller, cheaper, faster mantra was that the organization needed to launch more satellites and experiments using fewer program dollars. In the spirit of NASA's 1992 small satellite initiatives, space component manufacturers today strive to minimize risk, size, cost, and power, while pushing for the maximum performance allowable for a space-borne, high-reliability device.

Commercial influences on space electronics evolution

Many new technological advancements can be attributed to forces in the commercial and space markets. The commercial cellular phone market developed the SoC device as a solution to provide smaller, more power-efficient single IC devices that incorporated a mix of functions common to the telecommunications industry. The impact SoC technology has had in the cellular phone market is readily known by anyone that owned a five pound brick-sized cellular phone in the early 1990's and now totes around one of the sleek, credit-card-sized phones offered today.

Radiation and Power Requirements

Device Type	Total Ionizing Dose (TID)	Single Event Induced Latch-up (SEL)	Single Event Induced Effects (SEU)	Weight	Performance/Power
Commercial SoC	Poor TID induced device high leakage current failure	Poor SEL induced high-current device failure	Poor performance causing processor corruption of memory and instructions	**Excellent** 2-grams low weight, plastic package	**Excellent** typically 15 mW/MHz
Space SoC	Designed to meet at least 100 krads	Requires immunity to greater than 100 MeV @ 125 °C	Requires design mitigation techniques to eliminate susceptibility	**Good** 18-grams most requirements dictate ceramic sealed packages	**Excellent** performance of space devices achieve slightly less power/MHz
Space board-level product	Designed to meet at least 100 krads	All board components used must have immunity to greater than 100 MeV @ 125 °C	Design implements mitigation techniques to eliminate susceptibility	**Poor**/greater than 1,500 grams	**Poor**/ 100 mW/MHz more components require additional power

Table 1

On the other hand, space instrument and control system developers experimenting with SoC architectures are faced with additional challenges not seen in the commercial market segment. Although SoC technology is readily available with many configurations that a satellite developer can theoretically use directly, using commercial technology in a space environment is not feasible because of the radiation exposure that occurs outside the protection of the earth's atmosphere. Table 1 outlines a collection of decision factors satellite designers must consider during space-borne component development.

Compilation of a satellite SoC device

SoC devices typically contain a processor, a set of support functions such as timers and interrupt controllers, as well as interface functions for external bus operations. Figure 1 illustrates an SoC device that incorporates common satellite bus interfaces, a memory controller, a fault-tolerant 32-bit SPARC, and associated Debug Support Unit (DSU) integrated with an industry standard Advanced Microprocessor Bus Architecture (AMBA) bus.

Radiation performance in space

As stated earlier, one critical point to consider when selecting a component for use in a space environment is its radiation performance. The requirement to survive a multiyear space mission necessitates the use of radiation-hardened electronics in most satellite and space vehicles. The heart of any satellite control system is a radiation-hard processor designed with fault-tolerance as a goal, not an afterthought.

When selecting a component that is to be used in a radiation environment, always check the data sheet for assurances that the component will not have adverse operating effects when exposed to heavy ions or total dose particles creating a system latch-up (high-current condition), or performance degradation during the satellite's operating life. A fault-tolerant processor will also reliably operate through harsh radiation conditions without generating illegal instruction cycles or corrupting program memory. The processor should be designed for operation in a space environment and should include the functionality to detect and correct Single Event Effects (SEEs) in all on-chip RAM memories (Table 2).

Minimizing power consumption

In addition to the criteria listed in Table 2, the space-borne SoC should implement a sleep mode halting the pipeline and caches until an interrupt occurs, as well as clock controls to disable individual clocks to unused peripheral functions; this is an efficient way to minimize power consumption when the application is idle and should be accomplished without additional tool-specific software support.

SoC System SEE Criteria

SoC Element	Single Event Effects	Single Event Latch-up
Internal Memory	Capability to detect and correct SEE errors	Capability to operate through >100 MeV @ 125 °C
Processor Instruction Execution	Designed with fault tolerance creating total immunity to SEE	Capability to operate through >100 MeV @ 125 °C
Internal Registers	Capability to detect and correct SEE errors	Capability to operate through >100 MeV @ 125 °C

Table 2

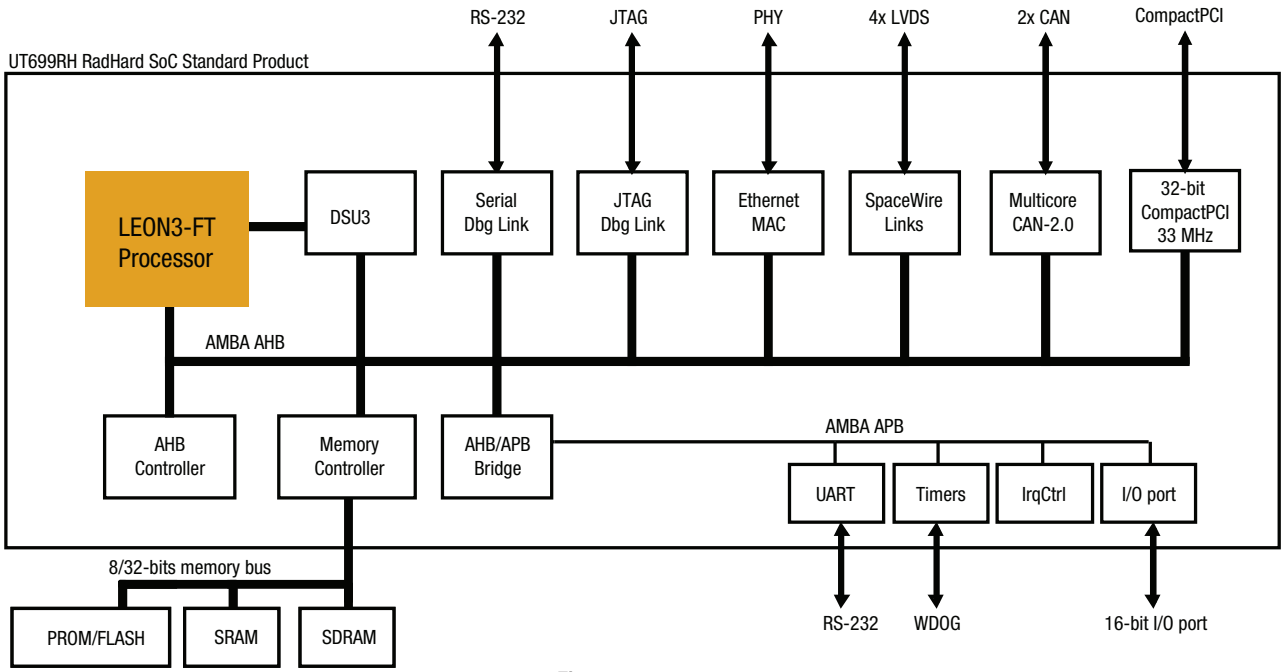


Figure 1

The benefits achieved when using an SoC architecture include lower power consumption, higher performance, and higher reliability compared to a board-level solution. In addition to increased hardware efficiency, the modularity of the SoC architecture creates a natural platform for modular software development. These SoC technology advantages are readily apparent by studying a simple example application. Figure 2 illustrates a generic box-level Telemetry and Command (T&C) unit and a corresponding SoC solution that incorporates all the box-level functions into a single integrated circuit.

The T&C black box comprises four individual boards integrated through a backplane and weighing approximately 7 pounds or 3,200 grams. Each board in the T&C performs independent transactions to gather telemetry and command an attitude control system through a variety of bus types. The total power budget for the T&C box is approximately 9 W (Table 3). Note that most of the black box power is consumed through the interboard communications and not the spacecraft links. In contrast, the SoC device performs the same system transactions with less power and weight, approximately 3 W at 100 MHz and 18 grams respectively.

A key to weight control

Another factor linked to the production of a reliable smaller, cheaper satellite is the mechanical weight and form factor. Compressing the area and mechanical features of a box-level component into a single device will not only dramatically reduce the weight, but increases overall reliability. To demonstrate, mechanical reliability includes assessment of mechanical dynamic loads within the electronic box; the calculation of these loads includes the Printed Wiring Boards (PWBs), the box backplane assembly, and each mechanical connection that connects one board to another. Therefore, a reliability assessment for the example box includes at least six mechanical failure points between two boards as illustrated in Figure 3. Combining all the functions from both boards into a single SoC integrated circuit significantly reduces the board and backplane mechanical failure sites from six to one failure point because the entire system now resides on a single board.

SoC software development support

A final decision factor with regard to selecting an SoC is processor support by third-party software development tools. Since

Weight and Power

	Weight	Power
Black Box	3,200 grams	9 W
SoC	18 grams	3 W
Net Savings	**3,182 grams**	**6 W**

Table 3

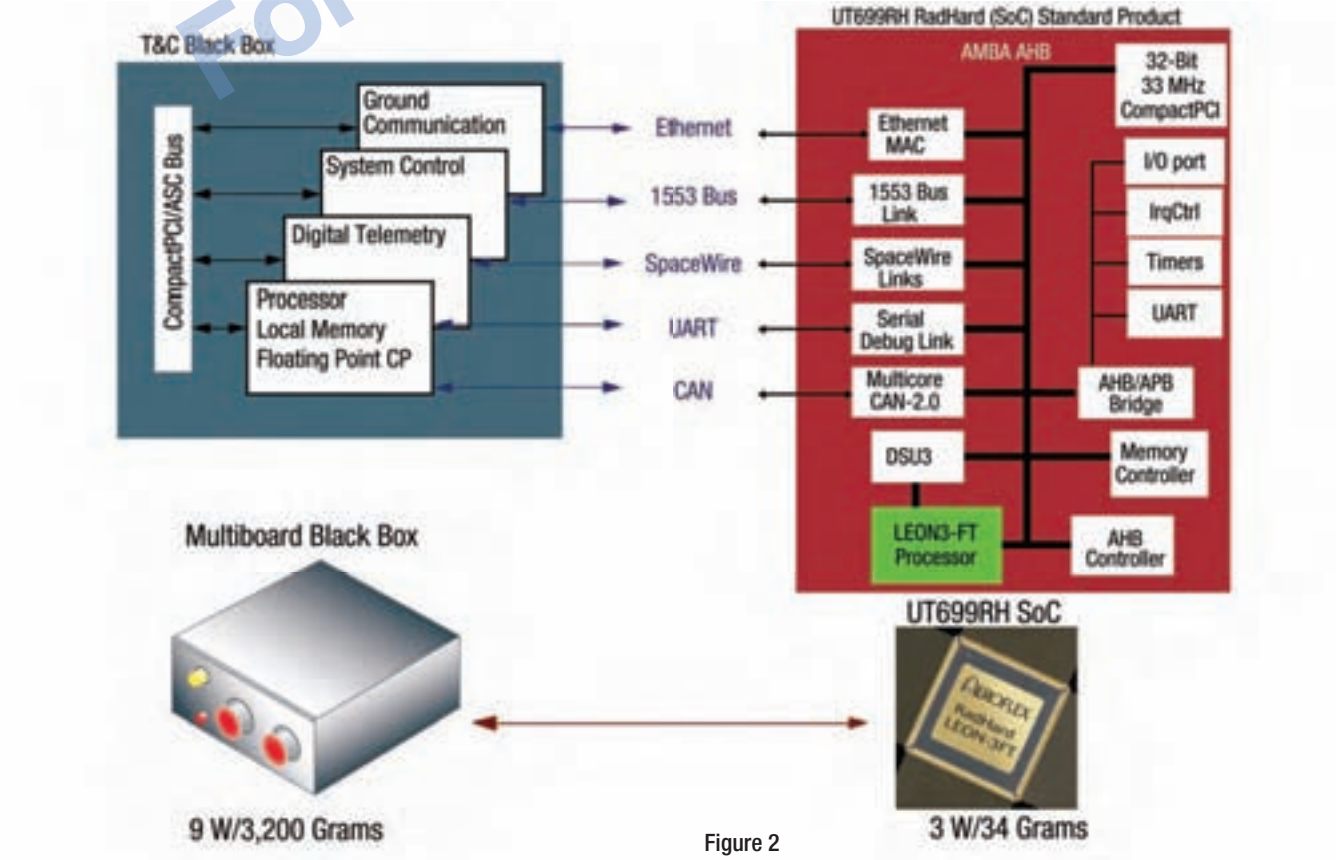


Figure 2

numerous functions (processor and peripherals) reside in a single SoC circuit, the developer has limited visibility of the device's interworkings. Magnifying this issue is the fact that qualified satellite flight equipment is scrutinized through rigorous testing to validate the hardware and software elements of the system. Table 4 highlights a few of the issues a designer should consider before settling on an SoC solution.

A System-on-Chip device with a set of features that seamlessly integrates to third-party tool suites as listed in Table 4 will provide enough functionality to satisfy most flight-qualified satellite software requirements while enabling SoC test and evaluation with common ground test equipment.

Next-gen processor-based applications in space

In conclusion, since much time and energy are expended to test and qualify a satellite system, software support and debug capability must be a key decision factor when choosing SoC components. The next generation of processor-based applications for space will be centered around SoC solutions that support the hardware, software, and radiation capabilities described. An embedded processor coupled to an SoC device architecture will be a key player in future satellite systems, accelerating development cycles through modularity while improving performance and reliability.

Dave Stevenson *is systems engineering group manager with Aeroflex Colorado Springs. His experience spans satellite telemetry and command subsystem design and new product business development. Dave is a licensed Professional Engineer in Electrical Engineering for the State of Colorado. He earned his BSEE from the University of Colorado and an MBA from the University of Denver.*

Aeroflex Colorado Springs
4350 Centennial Blvd.
Colorado Springs, CO 80907
719-594-8238
dave.stevenson@aeroflex.com
www.aeroflex.com

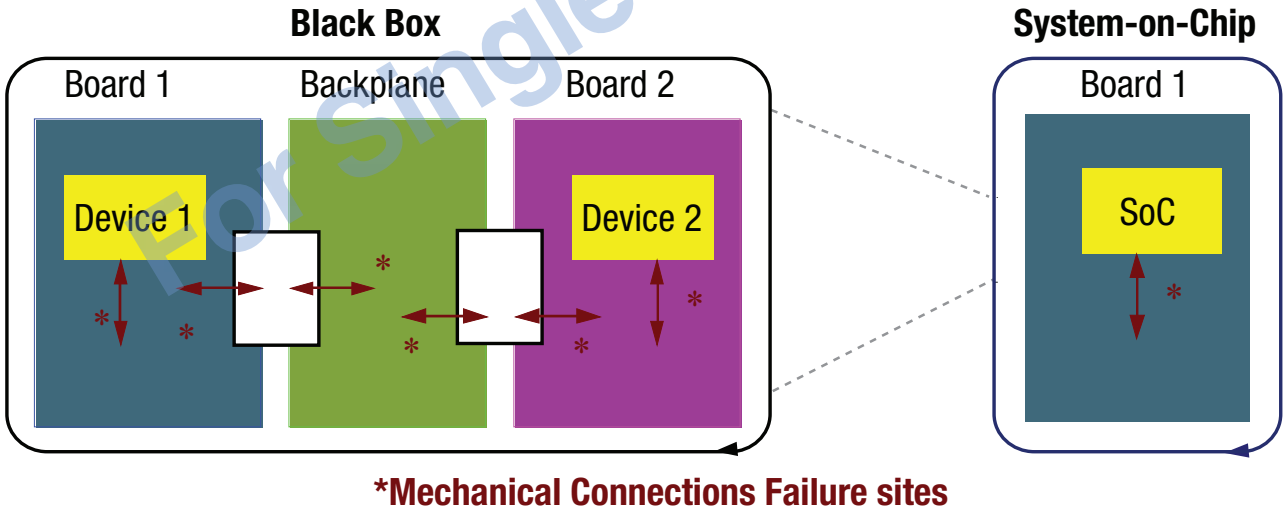


Figure 3

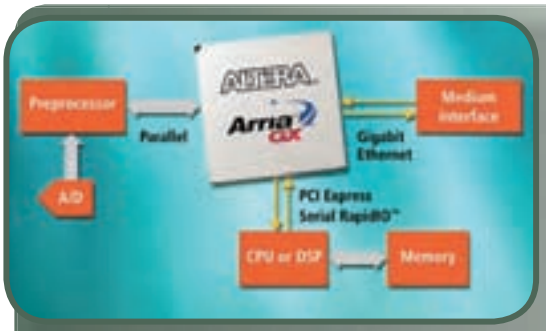
Software Support Criteria

Software	Hardware Debug
Support by third-party, industry standard development tool suites and compilers	Support for nonintrusive hardware debug through a standard interface port (Ethernet, RS-232, and others)
Real-time operating support by third-party, industry standard development tool suites and compilers	Watch-point set support for address and/or instruction execution
Software development environment support by a variety of platforms and operating systems	Breakpoint set based on address and/or instruction execution allowing trace and single stepping of the processor execution
	Full access to all SoC memory and register elements

Table 4

Editor's Choice Products

Low-cost FPGA bridges fabrics, custom I/O



Spotting a trend in I/O bridging, Altera has introduced their Arria GX FPGA family along with the latest version of their Quartus II (version 7.1) development software. The company discovered that every embedded system – from medical imaging device or industrial PLC to military radar or sonar – always has some sort of custom I/O. This I/O talks to the system-specific sensor, comms channel, or proprietary hardware. But elsewhere in the system, designers are routinely adding PCI Express, GbE, or Serial RapidIO.

Voila! The Arria GX is a transceiver-based FPGA ranging from 21,580 to 90,220 Logic Elements (LEs), with up to 4.5 Mb memory and sporting 12 transceivers. Supporting two speeds of 1.25 and 2.5 Gbps, the device can realize PCIe (x1 and x4), GbE, and Serial RapidIO (1x and 4x) protocols. Relying on the company's proven Stratix II GX transceiver technology in 90 nm (meaning: *not* "bleeding edge"), the Arria GX is intended to be low cost. A 50K LE device sells for $50 in 25,000 unit volume. For defense applications – in either commercial or extended temperature – the Arria GX is ideal when full-featured FPGAs are either overkill or simply cost too much.

Altera Corporation
www.altera.com
RSC# 33364

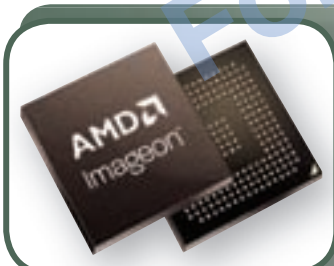
Inexpensive FPGA for DSP

Keeping with our low-cost Editor's Choice theme this edition, FPGA heavyweight Xilinx is one step closer to putting an FPGA in a cell phone with their Spartan-3A DSP. Sharp-eyed readers will wonder why the defense industry should care about FPGAs in cell phones, but the answer is that when FPGAs are *cheap* enough and *low power* enough for cell phones, then they'll be even more useful in increasingly cost-sensitive (relatively speaking) and portable military systems.

Boasting more than 20 GMACS for under $30, with a claimed 50 percent less dynamic power than other comparable reconfigurable DSP devices, the Spartan-3A DSP "platform" family uses a new Xilinx XtremeDSP slice that can be interconnected in creative ways on-chip. The highest-performing family member cranks 2,200 Gbps memory bandwidth, and the chip's DSP48A slices can realize wide math functions, DSP filters, and complex arithmetic – all at reduced power. The Spartan-3A DSP platform has up to 53,712 logic cells, 2268 Kb of block RAM, and 373 Kb of distributed RAM. Of course, Xilinx's development tools such as System Generator for DSP and AccelDSP synthesis have been updated for the new family.

Xilinx
www.xilinx.com/dsp
RSC# 33365

Mobile multimedia



So what does tomorrow's Future Force Warrior have in common with the next-generation UAV? They'll both be equipped with advanced image processing gear, and they'll have to do it on a strict power budget. Designed for portable, multimedia applications such as handheld consumer game consoles and video players, the AMD Imageon 2298, 2294, and 2192 media processors might be just the eye-openers needed for tomorrow's military multimedia systems.

If you envision 3G-enabled cell phones with full-motion video, you've captured the right target applications for these chips. They support DVD-quality video, image stabilization, high-performance audio, and a color-rich TV output display. They can support up to 12 Mpixel cameras, native hardware vector graphics, and onboard resolution scaling. The devices talk directly to a host CPU and (cellular) baseband system; they can also directly drive LCD screens up to WVGA *and* TV-type monitors. Additional inputs accepted include a digital video broadcast from a tuner and an SD card for storage. Although designed for portable consumer applications, the Imageon series certainly has a home on the battlefield of tomorrow.

Advanced Micro Devices
www.amd.com
RSC# 33494

BatterySmart makes CPUs sleepy

InHand Electronics' BatterySmart software has been shipping in select portable devices used by the military for a couple of years, so the product isn't exactly "new." But the company recently received U.S. Patent No. 7,111,179 for "Method and Apparatus for Optimizing Performance and Battery Life of Electronic Devices Based on System and Application Parameters." We think this honor is indicative of the uniqueness of BatterySmart dynamic power management software, so we are awarding it a belated Editor's Choice award. The product first gained recognition in handheld PDA-like computers deployed with U.S. troops and based upon InHand's *Elf* and *Fingertip* small form factor modules, which wowed us at the time of their introduction.

Designed to work with contemporary Intel StrongARM CPUs and their various power management states (called *idle modes*), BatterySmart can dynamically adjust the CPU clock speed based upon the operating bandwidth requirements of a handheld device. Additionally, the software provides several user-controlled interfaces that allow designers – or the users themselves – to understand how peripherals such as LCDs, PCMCIA cards, or flash cards are impacting battery life. An API in the software allows developers to tune the code to balance platform performance and I/O needs as a function of battery energy. Overall, dramatic power savings are possible with BatterySmart.

InHand Electronics
www.inhand.com
RSC# 33495

Editor's Choice Products are drawn from OSP's product database and press releases. Vendors may add their new products to our website at www.opensystems-publishing.com/vendors/submissions/np/ and submit press releases at www.opensystems-publishing.com/news/submit. OSP reserves the right to publish products based on editors' discretion alone, and does not guarantee publication of any product entries.

Critical Technical Information

Product Announcements

Leading Vendors

All at These Timely Events!



"For the first time since Motorola introduced the VMEbus 25 years ago, we believe another standard has emerged that could change the landscape of embedded computing across all industry segments."

– Paul Virgo, Marketing Director, Motorola, Inc.
Embedded Communications Computing

advanced TCA® summit E·U·R·O·P·E

advanced TCA® summit

September 18–19, 2007
Disney Newport Bay Club Hotel
Paris, France
euroATCAsummit.com

October 16–18, 2007
Santa Clara Convention Center
Santa Clara, California
advancedTCAsummit.com

Learn how to create standards-based communications solutions, make equipment more flexible and maintainable, reduce equipment costs and cut development time.

Topics: get the latest on AdvancedTCA, MicroTCA and AdvancedMC; designing for triple-play networks; high-availability systems; market research; interoperability; rugged systems; power and cooling; shelf management; and applications.

Who Should Attend: communications and embedded systems developers; telecom equipment manufacturers; solution providers; consultants; engineers and engineering managers; telecom hardware and software specialists; venture capitalists; analysts; marketing and PR personnel.

For Early-Bird Discounts Register Online Today!

Conference ConCepts, Inc.

What's up with the market?

Perceptions, rather than reality, are adversely affecting the embedded COTS market

By Chris A. Ciufo

According to the government's Office of Management and Budget – based upon information provided by the Bush Administration – the DoD has had incredible success in the Global War On Terror (GWOT). U.S. armed forces and their allies have liberated 50 million people in Iraq and Afghanistan and trained more than 215,000 Iraqi and 82,000 Afghan security forces – moving those countries closer to economic and civilian security. We've also deployed countless new technologies to the battlefield, from stronger and lighter body armor, to armed UAVs and UCAVs, to dramatically improved situational awareness, C4ISR command chain decision making, and *golden hour* medical advancements.

Overall, the numbers are astoundingly favorable regardless of how you look at it. From a military materiel, technology, and lethality standpoint, technology has indeed delivered on its promise of a *force multiplier* while increasing war fighter *survivability.* Moreover, the DoD budget is near an all-time high in real dollars, and FY07 should wind up at least 7 percent over FY06 to approximately $439.3 billion[1] and a whopping 48 percent over 2001.

But to those of us in the embedded industry "living and dying" on the basis of contract awards and *Science and Technology (S&T)* R&D spending, things don't feel so good. There's a sense that the market is slowing and getting ready to change, despite the top-down *Green Book* program-by-program numbers. As a result, big prime contractors are belaying RFQs and POs, schedules are sliding to the right, and vendors are slowing down their own R&D efforts, component purchasing and hiring decisions, and even advertising and trade show spending. **What the heck is going on?**

1 DoD budgets – look closer. Operational and Maintenance (O&M) expenses are *huge*. President Bush did the right thing by raising service members' pay by about 25 percent, enhancing special pays and bonuses to improve recruiting and retention, improving personnel housing and education benefits, and reorganizing the Army ground forces to improve mobility. The Air Force is using C-17s to ferry in supplies and fuel, reducing casualties from moving materiel by truck into Baghdad but doubling or tripling transport costs. We're wearing out Cobra, Huey, and Apache rotor blades in the fine desert sand, overstressing HMMWV engines and requiring more frequent replacement, and keeping Navy and Air Force assets flying 24/7 to provide suppression and RECON. This burns fuel and increases maintenance costs. For instance, in FY07 the number of Predator orbits has nearly doubled to 21 to provide sustained 24-hour surveillance. All of these O&M expenses mean less money is available for nonessential program spending[2], so we've seen F-22, F-35, DDG1000, FCS, JTRS, and countless others slide out, or smaller programs may go on life support. So while the topline DoD budget is up, a larger portion is being spent on nontechnology line items. Ergo, less money to spend on tech.

2 Decreased optimism. Poll after poll shows a majority of Americans no longer support operations in Iraq, and are becoming increasingly cranky about it. Additionally, the era of ENRON and Worldcom scandals has weighed on the buying public. According to *FORTUNE* magazine, a recent Gallup poll finds that "confidence in 'big business' among U.S. adults is very low." The number of Americans with no health insurance has increased during the past five years alone, and health care costs are rising at 10-15 percent per year. Total government debt held by the public was $4.8 billion in FY06 and is forecast to peak at $5.4 billion in FY 2011, while the federal surplus has turned to deficit and exceeds 30 percent (public debt as a percentage of GDP) until 2012 per CBO. The housing boom has cooled and many markets have seen inventory go from *hours on the market* to *months on the market;* as we went to press, consumer 30-year fixed mortgage rates hit their highest in 3 years. And though unemployment remains thankfully low in the near-record 4.5 percent range, in embedded tech the amount of offshoring to China, India, Russia, and other places has steadily increased. Chances are every one of us knows someone whose job was replaced. In short: Defense buyers, embedded tech company CEOs, and regular tech workers feel less optimistic about the future. Human nature is to withdraw and protect what you have, avoiding risk and uncertainty. This translates into lower R&D spending, fewer POs for CapEx and component inventories, and ordering "just enough" boards to meet customer backlog.

3 Washington political change. During mid-term elections last year, both houses of Congress fell under Democrat control. Business leaders – the same ones who drive both big and small companies in the DoD and embedded systems industries – are fully aware that a long-term change in party leadership could have serious ramifications. Companies are already girding for change, just in case. Investment is being funneled into more "green" efforts such as ethanol and energy conservation; large defense contractors have long been at the forefront of beating swords into plowshares when the defense climate changed. In 2008, if the Republicans lost control of the White House, DoD spending would likely (eventually) start to decrease. These real possibilities are also changing today's perceptions of the market, artificially slowing it down and creating caution among buyers.

There are other factors, of course. Add them to these three biggies and the top-down view of the embedded COTS market raises a caution flag. In effect, perception might become reality.

Chris A. Ciufo, Group Editorial Director
cciufo@opensystems-publishing.com.

[1]That's base budget – Plus-ups, bridges, riders, and supplementals make the actual number extremely variable until the year ends.
[2]"Essential" is anything that assists the war fighter in his or her *right-now* mission. That includes urban warfare such as IED discovery, verbal translation, or improved survivability.

GE Fanuc
Embedded Systems

Leading the way in Digital Receiver Technology.

An uncompromising commitment to be the best.

Delivering the best solution means not making compromises. Not comp romising performance – when you could have the industry-leading sustained performance of the ICS-8552B or ICS-8554D. Not compromising reliability – when you could have the world-beating expertise of GE Fanuc in developing rugged solutions.

Not compromising choice – when the ICS-8552B and ICS-8554D provide the best in ADC, DDC and FPGA technology for software defined radio, military communications, radar and signal intelligence applications. And not compromising your budget when you can choose the optimum solution for either development or deployment.

At GE Fanuc Embedded Systems, our commitment is to offer you our best in performance, reliability and selection. And on that we will not compromise.

www.gefanucembedded.com

ICS-8552B

ICS-8554D

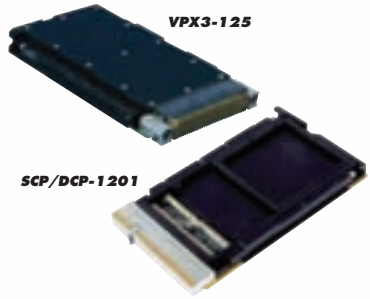
GE FANUC

RADSTONE EMBEDDED COMPUTING

Now a part of GE Fanuc Embedded Systems

IN THE HEAT OF BATTLE WE'LL HELP YOU KEEP YOUR COOL.

Temperature extremes on the battlefield can be brutal. They play havoc with the thermal limits of today's high-performance electronics and make aerospace and defense system integrators fight a two-front challenge – maximizing performance while beating the heat. We can help. We're experts in power management and thermal design. From intelligent component selection process to innovative, patented board and system cooling technologies, we knock-out heat so you can take full advantage of today's cutting-edge processing power.



Our new 3U VPX and CompactPCI Single Board Computers utilize ultra-low power processors from P.A. Semi™ and Intel® to maximize performance-per-watt in weight and space-constrained environments.

CURTISS WRIGHT Controls
Embedded Computing

www.cwcembedded.com

POWER MANAGEMENT... *ABOVE & BEYOND*