Intel's inside ... everything
CPU does Windows, DSP, and UAS/UAVs

Also:
Hot software: Multicore and open source

Military
EMBEDDED SYSTEMS

January/February 2011 Volume 7 Number 1

EVENTS
http://events.opensystemsmedia.com

Embedded World 2011
March 1-3, 2011 • Nürnberg, Germany
www.embedded-world.eu

ESC Silicon Valley
May 2-5, 2011 • San Jose, CA
http://esc-sv09.techinsightsevents.com

E-CAST
http://ecast.opensystemsmedia.com

Simplifying the Complexities of Multicore Processors with COTS Single Board Computer Solutions
February 17, 2011 • 12pm MST
Presented by: Freescale Semiconductor, Emerson Network Power, Eurotech Inc.

WEB RESOURCES

Subscribe to the magazine or E-letter
Live industry news • Submit new products
http://submit.opensystemsmedia.com

White papers:
Read: http://whitepapers.opensystemsmedia.com
Submit: http://submit.opensystemsmedia.com

ON THE COVER:
Increasingly, UAS payloads like those of this MQ-9 Reaper capture more video, include more sensors, and are prime candidates to let Intel Core processors crunch DSP algorithms. See article on page 26. Photo provided by General Atomics Aeronautical Systems, Inc.

Published by: **OpenSystems media.**

ENVIROINK™
The inks used to print the body of this publication contain a minimum of 20%, by weight, renewable resources.

WWW.MIL-EMBEDDED.COM

Field Intelligence

Mezzanines continue to evolve

By Duncan Young

The ability to customize the functionality of off-the-shelf embedded computing modules by the addition of mezzanines has become an essential element in the battle to get the most from the least space at an affordable price. Mezzanine standards abound, one of the most common being "IEEE1386.1, PMC/XMC," which can be used with many types of architectures such as VMEbus, VPX (VITA 46), and CompactPCI, as well as the desktop PC using PCI carrier cards.

The PMC standard was introduced more than 15 years ago and offers an ideal amount of real estate. It has been continuously improved with greater connectivity, an extended PCI bus, better power dissipation and, most recently, support for high-speed serial signaling such as PCI Express between mezzanine and base card. Continuous improvement has yielded a quadrupling of the PMC's power dissipation capability compared to the original IEEE1386.1 specification, as more functionality and performance get packed into its limited space.

PMC applications

There are many different military applications for PMC modules, ranging from flash storage, avionics bus interfaces, and general-purpose I/O, to Network Interface Controllers (NICs) and switches to sensor processors. Some of the most challenging of these PMC applications are sensor front-end processors using either an FPGA or the emerging GPGPU-based processing arrays. Using these types of devices pushes the boundaries of power dissipation. Typically, such a sensor processing PMC incorporating, for example, a Xilinx Virtex-5 FPGA and high-speed analog I/O will dissipate almost 30 W. These types of front-end processors are often used as digital down-converters, video processors, or radar pre-processors. With the addition of analog O/P, a complete Electronic Support Measures (ESM) subsystem can be configured on a single PMC module, in many cases able to operate without runtime processor intervention.

Multiple PMC/XMC modules

Because PMC/XMC modules offer such efficient use of space, subsystems often get configured with more mezzanine modules than can be fitted to just one SBC. A 3U-sized SBC is limited to one mezzanine slot, whereas 6U accommodates two. Carrier cards provide the means of adding further mezzanine slots within a chassis, but must be connected to the SBC via PCI, PCI-X, or PCI Express, generally using the SBC's PMC/XMC site to make this connection. In a VPX environment, adding further XMC modules has been simplified. They can be plugged directly onto a passive carrier card into any backplane slot supporting PCI Express. However, where PCI Express is not so readily available, other configurations and permutations might require PCI bridges, for example, to support a PCI Express mezzanine in a PCI-X environment.

The myriad embedded products vendors supporting PMC/XMC mezzanines recognize the ways that the original specification has been stretched to accommodate today's levels of performance and functionality. This is reflected in their SBC designs, which account for the additional power needs and provide well-managed signal routing and noise containment through the use of best practices. This is not always the case in the prototyping laboratory where critical PMC/XMC modules might be incorporated into a desktop PC environment using a PCI carrier.

Compared to a rack-mounted modular system, a desktop's PCI and PCI-X slots can be electrically noisy and lack sufficient power or heat management to support leading-edge PMC/XMC modules. In this case, the PCI carrier card can be vital to remedying these shortcomings. One viable example of this is the soon-to-be-announced short form factor PCI Express carrier card from GE Intelligent Platforms (Figure 1) supporting Gen2 x8 PCI Express to the host and including cooling for high-power PMC or XMC modules.

Figure 1 | Short form factor PMC carrier card for PCI applications from GE Intelligent Platforms

New mezzanine proposals

Just as VPX provides a step increase in capability compared to its VMEbus forebear, significant evolution of mezzanine standards will be needed to maintain future technology leadership. A VITA 71/Rugged Mezzanine working group has been formed to evaluate and create standards for a new generation of mezzanines for embedded computing users. Based on the considerable industry experience gained with the PMC/XMC standards, significant advances can be expected:

- Increased interoperability among vendors
- Improved connectors with signaling rates to 12 GHz
- More efficient space utilization by reducing connector sizes and keep-out areas
- Fewer power rails
- Greater power dissipation
- A wider range of sizes

The standardization of mezzanine specifications will continue to be challenging as SBC designers need to anticipate a multiplicity of mezzanine vendors plus future growth in capability and performance. The result can be overspecification or the provision of many unused resources to meet every eventuality. VITA 71 needs to address these issues to level the playing field, yet at the same time leaving scope for the continued innovation and evolution so essential to a standard's long-term future.

To learn more, e-mail Duncan at duncan_young1@sky.com.

RTD Embedded Technologies, Inc.

Modularity at its Best.

Deployed System

New Functionality

Easy Upgrade

Above: **PCI/104-Express** Intel® Core™ 2 Duo IDAN® showing how easy it is to upgrade, repair or re-configure RTD's modular, rugged systems based on mission requirements.

Below: a sample of RTD's **PCI Express** offering. RTD proudly leads the industry in **PCI/104-Express** selection and development.

Intel® Core™ 2 Duo cpuModule™	Intel® Atom™ E6xx cpuModule™	High-Speed Digital I/O	Isolated Digital I/O	88W Power Supply

SATA Drive Carrier	Dual-Slot Mini PCIe	5-Port Ethernet Switch	Dual Gigabit Ethernet	PCI Express to PCI Bridge

DESIGNED AND MADE IN THE UNITED STATES
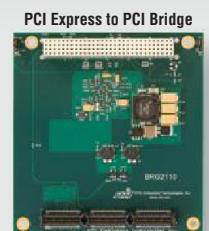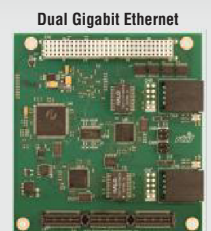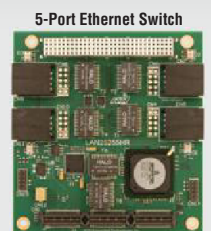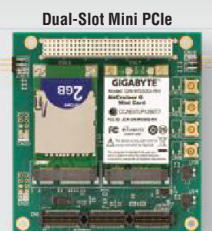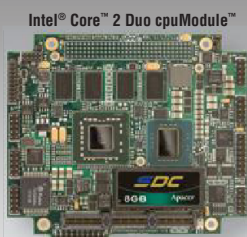
The Leading Source for PCI Express
Design, Engineering, Manufacturing & Tech Support

www.rtd.com   AS9100 and ISO 9001 Certified   **sales@rtd.com**

Mil Tech Insider

Open standards for digital video drive common infrastructure

By Steve Edwards

The complexity of deployed embedded military systems increases as more video sensors are added to ground and airborne platforms, delivering ever-increasing amounts of video data to be processed, viewed, and archived. As the variety of video sensors used in military radar and signal processing applications continues to multiply, many of which feature incompatible input requirements, the result has been increased system complexity. One key area of complexity is the cabling and configuration needed to distribute multispectral data within these sensor-to-video-/-display/-recorder embedded systems.

Today's military platforms often have upwards of dozens of sensors, making the real-time distribution of video data a real challenge. The good news is that open standards are underway and gaining traction, such as the new DEFSTAN 00-82 standard in the UK. This trend is helping to drive industry use of standard media, such as 10 GbE to transport data, which will reduce system complexity and promote system interoperability.

The open standards approach

A standard approach to digital video connectivity will help military suppliers provide compatible subsystems that support a common infrastructure. Emerging standards for digital video interoperability support this approach. One example, as mentioned, is a well advanced, recently published standard in the UK: DEF STAN 00-82, which establishes protocols for video digital streaming. This standard is being mandated by the UK Ministry of Defence's (MoD's) Generic Vehicle Architecture standard (DEF STAN 23-09), which is overseen by a standards consortium. Curtiss-Wright Controls Embedded Computing (CWCEC) recently demonstrated implementations of digital video streaming and recording over a 10 GbE fabric using DEF STAN 00-82.

A typical video distribution system architecture comprises the front-end interfaces to the sensors, which accept data from legacy analog systems (RGB or composite video) and newer interfaces such as HD-SDI and 3G-SDI. It can also accept streamed video data from network-compatible sensors that generate data onto a digital network, such as GbE Vision, or systems built on top of Real-time Transport Protocol (RTP), as invoked by DEF STAN 00-82. Multiple video streams are then sent through a rugged Ethernet switch and multiplexed to create the 10 GbE fabric of the video network within the platform.

The platform's video processing and data storage equipment, such as CWCEC's Sentric2 digital video recorder, connect to the 10 GbE fabric to extract multiple video feeds from the multiplexed stream (Figure 1). The appropriate image processing can then be performed on the video data, such as multispectral image fusion, image registration and stabilization, image enhancement, and image stitching that combines multiple sensor views to provide an outside view of the platform.

Figure 1 | The Sentric2 High-Definition Video Recorder System captures, compresses, and stores multiple channels of high-resolution analog or digital video, composite TV, network video, and audio in digital formats.

Supporting heterogeneous protocols

By using an open standard for transporting multispectral data over a single network, the system designer is also able to eliminate the need to translate different video protocols. The results are a unified transport medium and a unified vocabulary for controlling the various video sources and the format of the video streams.

The best advantage of using a common digital video streaming format is that it eliminates the need to perform protocol conversion, which can be costly for high-bandwidth video data. A better method is for the sensor to speak the same language as the equipment receiving the video signal. There are two aspects to video protocols: data and control. It is especially important that the equipment that wants to subscribe to video streams from specific sensors be able to speak a control language that those sensors understand. The control language should be sufficiently generic that multiple manufacturers can conform to that standard with their own specialized equipment.

Eliminating dedicated cabling

Typically, individual cabling is used to connect each video sensor to the data-concentration point. The common infrastructure approach eliminates the need for multiple cables in a confined space, supporting transmission over a standard network of HD sensor video, simultaneous digital video, and metadata recording. It also enables multiple video streams to be concentrated at one or more processing units to undergo advanced image-fusion techniques.

Keeping up with sensor proliferation

Attaining video sensor data interoperability with open standards and a common infrastructure network will help COTS vendors keep up with the growth of sensors on military platforms. The real-time battlefield data these sensors provide saves lives and supports critical missions. Reducing cabling and configuration complexity and providing a high-bandwidth common architecture for distributing video sensor data will enable onboard processors to take full advantage of the growing amounts of video sensor data being made available to today's warfighter.

To learn more, e-mail Steve at Steve.Edwards@curtisswright.com.

1 GHz PC/104 SBC
Supports Networking and Communications

Hardened for harsh, rugged environments, the low power PCM-VDX-2 PC/104 single board computer is ideal for Mil/COTS applications. It is packed with serial, USB, and Ethernet ports, plus GPIO. With its diverse range of functions, most applications will not require additional I/O cards; however, it has Mini PCI and PC/104 connectors for specialty I/O expansion.

- Fanless, low power 1GHz Vortex86DX processor
- PC/104 Bus compliant form factor (90 x 96 mm)
- 512MB of soldered-down onboard DRAM
- 1MB of battery-backed SRAM
- CompactFlash socket
- Optional 512MB onboard SSD flashdisk
- Full-featured I/O includes:
  - Two 10/100 Mbps Ethernet ports
  - Four USB 2.0 ports
  - ESD protection on LAN, USB, and serial ports
  - Four serial RS-232/422/485 ports
  - 16 lines of general purpose I/O
  - PATA, LPT, PS/2 KYBD and Mouse controller
  - Mini PCI and PC/104 expansion connectors
  - WDT, RTC, status LEDs, and beeper
- Extended temperature -40°C to +85°C operation
- Runs Linux, DOS, and other x86-compatible operating systems
- Downloadable drivers available
- Responsive and knowledgeable technical support

Understanding long-term product availability is a critical issue for Mil/COTS customers, the PCM-VDX-2 is offered beyond 2017.

Contact us for additional product information and pricing. Our factory application engineers look forward to working with you.

Ask about our 30-day product evaluation.
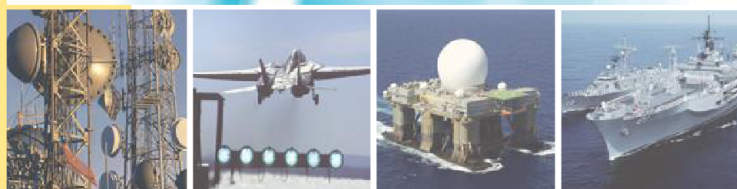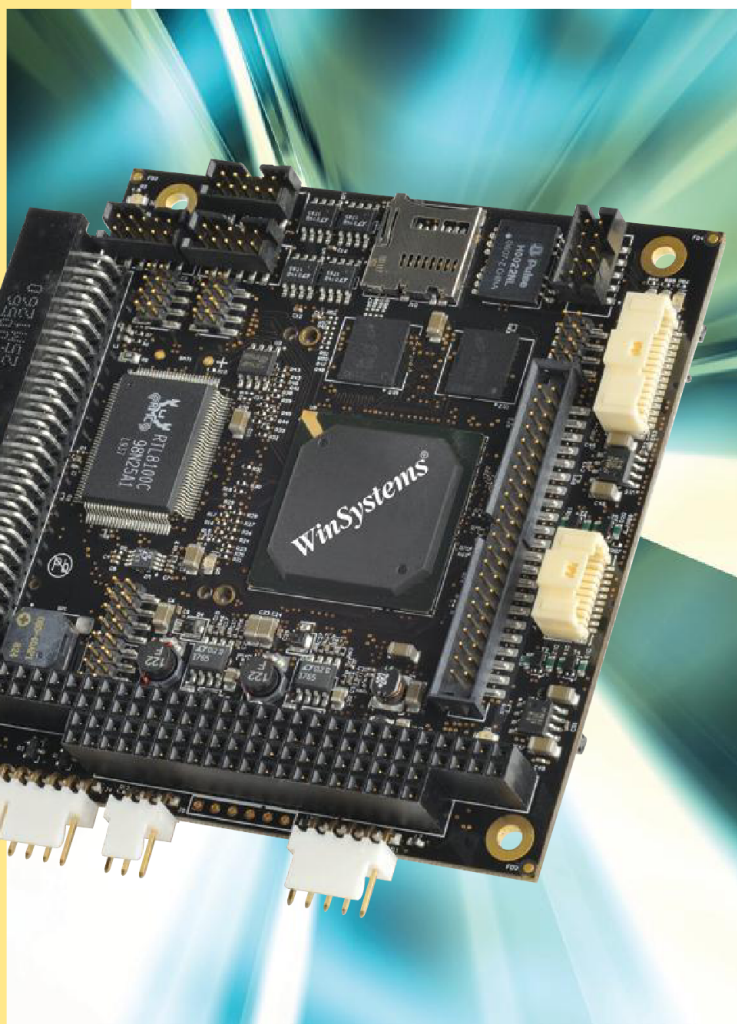
Legacy Software Migration

By Charles Rush

Migrating legacy middleware to multicore environments

Multicore microprocessors in embedded defense systems are becoming ubiquitous. Embedded systems have specialized requirements, such as heat dissipation and battery life, that restrict the ability to simply crank up the clock speed of a processor. Embedded microprocessor designers grappling with these physical limitations have turned to multicore architectures as a preferred method for continuing geometric performance improvements. Multicore ARM and PowerPC chips are already available, and Intel is pushing its low-power multicore processors into embedded designs. For defense system architects updating legacy systems with new capabilities, incorporating a multicore processor can require an analysis and development effort that extends well beyond the traditional software port.

Unlike the performance increases from boosting microprocessor clock speeds, accessing the performance potential of a multicore processor often requires engineering development efforts to a parallelize software execution. Along with efforts of identifying, expressing, and implementing parallelization, most approaches to parallel software require the explicit management of communications and shared resources. Selecting a set of software components that already addresses these concerns can greatly simplify the transition to a multicore microprocessor architecture.

In particular, the critical but invisible middleware that provides the communication framework for many embedded systems can become a significant asset in simplifying the architecture and design of a multicore system. Software engineers in pursuit of multicore performance often overlook the influence middleware has, and how the proper choice of middleware can significantly improve performance.

How middleware simplifies defense systems architecture

Middleware frameworks evolved to help software developers overcome the inherent complexity of programming for distributed systems. The complexity resulted from the heterogeneity of networks, hardware, operating systems, and programming languages present in many systems. The Common Object Request Broker Architecture (CORBA) is a standards-based middleware that communicates among operating systems, multiple languages (C++, Ada, and Java), microprocessors (x86, PowerPC, ARM, MIPS, and so on), and data transports (Ethernet, PCI Express, shared memory, RapidIO, InfiniBand, and so on). CORBA has been used in defense applications for years because of its flexibility and interoperability, resulting in millions of lines of CORBA code in today's defense systems. It can be found in defense applications ranging from command and control to weapons systems to air traffic control systems.

A CORBA application is architected as a set of communicating objects, each able to provide services or utilize the services provided by another object. The CORBA middleware handles all of the details of communication between objects. The application is unaware of where an object is physically hosted or how messages are transported among objects. All CORBA object communications are candidates for parallel execution. During a system update, parallel execution can be increased by re-deploying CORBA objects in new processes across several microprocessors or moving them to new multicore microprocessors without requiring application refactoring.

Internal tests show that applications using multicore-enabled middleware can achieve a linear increase in performance as each core is added without rewriting code.

How middleware affects performance

Middleware frameworks can create performance bottlenecks stemming from their design and implementation. Some of the ways that middleware can be a bottleneck include:

■ Protecting critical sections
■ Resource usage
■ Networking

Locks are necessary to keep critical sections of the application or system resources protected from being used by more than one execution thread at a time. However, poorly designed middleware uses locks very inefficiently, for example, too many locks are created or code segments locked are too large. Efficient middleware minimizes the number of locks and ensures that each thread holds the lock for the minimum amount of time.

Middleware can also consume excess resources. For example, if the middleware opens too many file descriptors that identify system components such as communications sockets, it can consume too many resources. This causes the operating system to slow or even halt application execution. Again, efficient middleware will consume minimal resources.

A key component of middleware is the data network transport services that read and write data to the network. How well these services exploit the power of today's multicore processors plays a major role in determining an application's performance.

Issues to consider include the speed of each transport and whether the middleware can support concurrent processing across a network interface. It is also important to determine if middleware can use different physical network transports such as PCI, FireWire, shared memory, TCP/IP, RapidIO, ATM, and InfiniBand. A well-designed middleware will include a pluggable transport architecture that places no limit on the number or types of transport that can be used.

Multicore performance simplified with efficient middleware

As critical as the middleware communication framework is, it is often overlooked as an opportunity for improving application performance. Most legacy middleware is unable to take full advantage of today's multicore processors and can hobble application performance despite careful upgrades to other portions of the application infrastructure. Replacing legacy middleware implementation with multicore-capable middleware, such as the ORB*express*, is a low-cost, low-risk option that can simplify system architecture, enable reuse of legacy source code with a minimum amount of additional programming, and enable increased system performance.

Charles Rush *is Senior VP of the Middleware Technologies Group at Objective Interface Systems, Inc. He can be contacted at charles.rush@ois.com.*

Daily Briefing: News Snippets

By Sharon Hess, Assistant Managing Editor

Data must-haves: Proven reliability, fast transmission

While advanced, battlefield-savvy technology is paramount, what good is it without the ability to transmit data through it – as reliably and quickly as possible? Accordingly, Northrop Grumman continues to ensure reliability via its recent successful second flight test phase completion pertaining to the U.S. Army's Multi-Role Tactical Common Data Link (MR-TCDL) system. The 14-flight test series comprised a NASA-ER2 aircraft (Figure 1) and a Gulfstream II aircraft, both of which featured integrated MR-TCDL. The two aircraft were digitally linked together as well as to specified ground entry points. The testing outcome indicated that MR-TCDL provides data transmission – reliably – in excess of 200 Mbps between aircraft separated by more than 270 nautical miles and between ground networks and more than one aircraft. MR-TCDL additionally facilitates fast extension and connection of wireless and terrestrial wired networks. L-3 Communications Systems is MR-TCDL's builder.

Figure 1 | A NASA-ER2 aircraft was one of two aircraft participating in Northrop Grumman's recent MR-TCDL testing. Photo courtesy of NASA

DARPA contract boosts jamming capabilities

DARPA has enlisted BAE Systems National Security Solutions to incarnate "counter-adaptive wireless communication threats" technologies, per a recent $8 million contract between the two entities. The technology will fall under the Behavioral Learning for Adaptive Electronic Warfare (BLADE) program umbrella, entailing new techniques and novel algorithms to arm DoD electronic warfare systems with the capability to automatically jam battlefield RF threats – and within appropriate time frames. Contract work will occur at BAE's Burlington, MA; Piscataway, NJ; and Nashua, NH sites and is slated for completion by May 2012.

Navy subs/ships ride the COTS wave

In the wake of seemingly endless defense spending cuts, the U.S. Navy's move toward suiting up or refitting its surface ship and submarine systems with open architectures is becoming reality. The evidence: A recent Phase III Small Business Innovative Research (SBIR) contract modification of $16 million to The Consulting Network, Inc. The modification covers "required services … related to open architecture concepts" such as hardware and software development and integration, and COTS products procurement for myriad surface ships and Virginia-class/other subs. Services and wares under the contract modification will be proffered when requested by the Navy, with the goal of readying future versions and legacy flavors of Navy systems for Global Information Grid (GIG) and FORCEnet compliance. Contract work is slated for fulfillment by this September. The contracting activity is the Naval Sea Systems Command in Washington, D.C.

Sensitive mil apps will claim their power

"Sensitive military electronic applications" might soon have a new life force to sustain them, according to a recent contract between the United States Air Force Research Laboratory (AFRL) and City Labs, Inc. The almost $1 million contract specifies that City Labs incarnates long-life batteries for possible use in radar systems, UAVs or drones, computers, aircraft, and sensors (Figure 2). The City Labs battery, based on tritium, can withstand temps of -50 °C to +150 °C, along with extreme altitude and vibration. Used for illuminating Exit signs on commercial aircraft and in commercial buildings, theaters, and schools, tritium is a hydrogen-spawned radioactive isotope. Hence, batteries based on it are touted to last as long as 20 years.
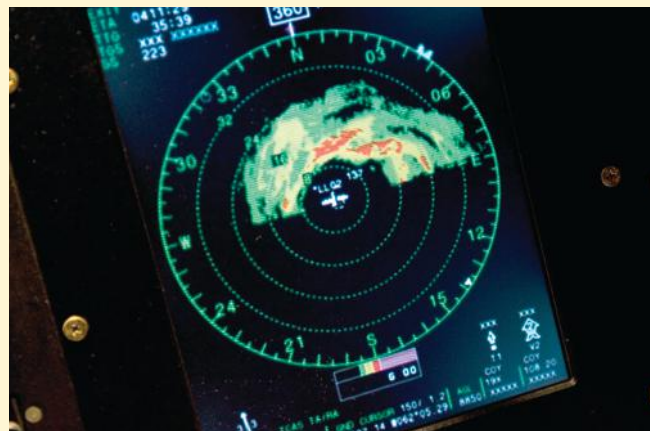
Figure 2 | Radar systems are one type of "sensitive military electronic application" set to benefit from a recent contract between the AFRL and City Labs, Inc. for long-life batteries.

Super Hornets fly in early

Four ahead-of-delivery-schedule iterations of Boeing's F/A-18F Super Hornet multirole aircraft recently touched down on Australian soil for assimilation by the Royal Australian Airforce (RAAF) (Figure 3). This latest delivery brings the RAAF's total F/A-18F Super Hornet ownership to 15, with 9 more on order. The early delivery of the aircraft to Base Amberley facilitated RAAF's ability to achieve IOC [Initial Operating Capability] faster, as the RAAF transitions from the classic Hornet and F-111. Though all 15 F/A-18F Super Hornets delivered to the RAAF sport Raytheon's APG-79 Active Electronically Scanned Array (AESA) radar, 3 of the 4 recently delivered iterations were "prewired for potential conversion to electronic attack capability," as will be all remaining RAAF Super Hornets, Boeing reports.

Figure 3 | Boeing recently delivered four more iterations of its F/A-18F Super Hornet multirole aircraft to the Royal Australian Airforce, ahead of schedule. Photo courtesy of the Australian Department of Defence

Unmanned vehicle thwarts sub threats

When a system is autonomous, technicians and other military personnel are free to use their time for other pressing matters. A recent $2 million DARPA contract with prime Science Applications International Corporation (SAIC) supports this paradigm, stipulating that SAIC develops an autonomous, unmanned surface vessel concept. Dubbed the Anti-Submarine Warfare (ASW) Continuous Trail Unmanned Vessel (ACTUV) Phase I, the unmanned vessel program's purpose is to aid in thwarting modern threat submarines. The primary goal: to produce an unmanned surface vessel that sustains ongoing, autonomous tracking of threat submarines. Potential missions include undersea warfare, enforcing "maritime rules of the road," and continuous at-sea operations. Five other companies will assist SAIC in Phase I, which spans the contract's six-month duration.

For consideration in Daily Briefings, submit your press releases at http://submit.opensystemsmedia.com. Submission does not guarantee inclusion.

Figure 4 | A recent U.S. Army/General Dynamics Land Systems contract provides for conversion of 42 Abrams M1A2 battle tanks into an M1A2S variant for the Kingdom of Saudi Arabia. Photo courtesy of U.S. Army

Abrams tank gets away with new configuration

The trusty Abrams M1A2 battle tank is about to undergo a metamorphosis, per a recent U.S. Army contract with General Dynamics Land Systems for $37 million (Figure 4). Accordingly, 42 of the tanks will be re-engineered to sport a new M1A2S configuration for delivery to the Kingdom of Saudi Arabia. The M1A2S variant is reportedly equipped with additional lethality-increasing capabilities and renders reduced obsolescence. General Dynamics is slated to fulfill work under the contract at its Lima, Ohio locale by September 2012. The contracting activity is the U.S. Army TACOM LCMC.

Technology duo benefits from Navy contract

The mention of one-half of any famous pair brings to mind the other half … Antony and Cleopatra, Romeo and Juliet, and certainly for myriad years: MH-60R and MH-60S. Case in point (regarding the latter duo, anyway …): A recent $72 million contract between the U.S. Navy and Lockheed Martin Mission Systems and Sensors in Owego, NY, stipulates that Lockheed Martin will provide 24 mission avionics systems plus common cockpits for MH-60R (Figure 5). Additionally, MH-60S will benefit from the 18 common cockpits afforded by the contract. Both of the multimission, maritime helicopters' end-of-life components are also covered under the contract. The contracting activity is the Naval Air Systems Command, and work completion is anticipated by December.

Figure 5 | The oft-paired MH-60R and MH-60S maritime helicopters will benefit from a recent $72 million contract between the U.S. Navy and Lockheed Martin. Lockheed Martin photo

Enhancing application performance on multicore systems

By John Blevins

U.S. Air Force photo by Senior Airman Debbie Lockhart

In a simple view of the world, an application running on a multicore system should run at least as fast as the same application runs on a single-core system with the same CPU power. Unfortunately, in practice, that is not the case. However, by implementing application parallelism and using an SMP OS and an embedded hypervisor, one can solve the challenge and realize dramatic performance improvements.

Let's face it. The Multicore Era is upon us. How did we get here? For years, processor manufacturers delivered on the Moore's Law promise of doubling CPU performance every couple years by increasing the number of transistors, increasing clock rates, and increasing instruction-level parallelism. We saw clock rates go from 1 GHz in the year 2000 to 2 GHz in 2001 and finally to 3 GHz in 2002, where we seem to have hit a clock-speed brick wall.

The combination of increasing power demands and rising chip temperatures seems to have put the brakes on the clock speed race. One of the surest signs was Apple's switch from the PowerPC to the Intel Architecture. Promises of 3 GHz PowerPC G5 processors in Apple laptops never materialized due to power and heat problems. Processor manufacturers quickly realized that to keep doubling performance, they needed a new trick. That new trick was to add multiple cores to a chip. Now Mac computers with 12 cores at 2.93 GHz exist, and coincidentally, dual- and quad-core single board computers and systems are very prevalent in the military embedded realm today.

However, if the software running in the system is not optimized for multicore,

there can be degradation in performance when migrating from a single-core based system. The higher transistor density in a multicore CPU does not generally translate to an increase in speed on applications that are not parallelized. Additionally, SMP OS and embedded hypervisor RTOS technologies can aid optimum performance in single-core to multicore migration. But first we will examine the issues of synchronization, concurrency, and scheduling.

Synchronization and concurrency overhead kill performance

A real-time embedded SMP operating system must maintain deterministic scheduling and interrupt response as well as respond rapidly to interrupts and high-priority tasks. This job becomes substantially more difficult and time consuming in a multicore CPU. For example, when multiple CPUs are active, data may be accessed by more than one of them simultaneously, adding a new level of concurrency issues for the OS to deal with. This requires additional mechanisms for concurrency control. Because of the increased concurrency in multicore systems, locking and synchronization mechanisms are more complex and take more CPU time than in single-core systems.

Generally, on a single-core system, a critical section of code can be protected from interrupts by simply disabling pre-emption, doing the necessary work, and re-enabling preemption. Enabling and disabling preemption can be as simple as storing a value in a variable. In a multicore system, each core has its own unique set of interrupts, so disabling pre-emption does not make a lot of sense, since code on the other cores could still execute the critical section. Instead some form of locking needs to be introduced. In LynxOS, a deterministic hard real-time OS from LynuxWorks, this is done with Kernel Spinlocks and requires the use of special hardware mechanisms such as locked bus cycles. These mechanisms are considerably more complex than the simple memory accesses a single-core system uses, and this complexity adds to overall performance degradation.

In addition, when code on one core is in a critical section, code on other cores is blocked waiting for the code to finish the critical section. If the locks are coarse-grained, it is possible that several cores could be idle because they are unable to schedule any useful work.

The synchronization and concurrency overhead incurred on multicore systems

is most visible at the operating system software level, but is also apparent in multi-threaded applications that rely heavily on constructs like condition variables, semaphores, and message queues. Operating systems compiled to support multiple cores are typically about 10 percent slower on a single-core system than the same operating system compiled to support a single core.

Scheduling threads across multiple cores

The scheduling algorithms play a key part in harnessing the power of multiple cores and can cause performance issues if not implemented carefully. Typical scheduling algorithms maintain a per-CPU queue of threads that are ready to run and allocate CPU time based on this queue. However, in a real-time system, it is critical to preserve real-time determinism, so the scheduling approach is different. The scheduling happens on a global basis where the highest-priority thread runs on the first available CPU. However, this may lead to higher levels of cache misses. This can be addressed by using design optimizations in real-time thread scheduling.

One such design optimization, known as *processor affinity*, allows applications to request an "affinity" to a processor core. In this case, the operating system schedules the applications on the preferred processor core, as long as it does not affect overall system scheduling. A more rigid form of processor affinity is *processor binding,* where the task is always scheduled on the same processor core. However, this approach in RTOSs may lead to priority inversions. Operating system design should accommodate considerations such as processor affinity

without degrading real-time determinism and responsiveness. In the context of a real-time operating system, other key factors such as priority scheduling and interrupt latency should be preserved in multicore architectures.

An SMP-enabled real-time operating system must schedule tasks dynamically and transparently between processors to efficiently balance workloads using available processors. It optimizes the support of load balancing on multiple cores along with preserving the key elements of real-time latency and determinism. If the operating system "bounces" the application from core to core, the application will take additional Translation Lookaside Buffer (TLB) and cache misses, reducing performance. On the other hand, if the application is "pinned" to a core, there may be enough additional demand placed on that core to slow down the application, compared to running it on a single core.

Taking full advantage of multicore processors

To maximize multicore performance, application parallelism, SMP-enabled OSs, and embedded hypervisor technologies should be explored.

Application parallelism maximizes CPU utilization

All applications should be carefully examined for opportunities to parallelize the tasks. In parallel computing, an application is broken down into threads that execute independently on separate cores (Figure 1). Application parallelism is dependent on the ratio of computation to communication overhead. The computation is the amount of time the CPU spends executing application code. The

communication overhead is the amount of time that the OS spends in communicating between cores. In a typical multicore architecture, the communication overhead indicates how often messages are sent between different cores. The more threads an application has, the higher the chances that they are scheduled on different cores, which in turn increases the communication overhead.

Each type of system has different characteristics, but when optimizing application parallelism to maximize performance, there are broadly two types of application parallelism that can be used:

1. *Coarse-grained parallelism* is characterized by large tasks, single threaded and low communication overhead. In this case, the ratio of computation to communication overhead is high. This indicates that the communication overhead is lower than computation time, thereby yielding better multicore performance.

2. *Fine-grained parallelism* is characterized by small tasks, multithreaded and high communication overhead. In this case, the ratio of computation to communication overhead is low. This indicates that the communication overhead is higher than computation time, thereby yielding lower multicore performance.

Applications that are CPU-bound can exploit the full power of multicore architectures since they are coarse-grained, while memory-bound or I/O-bound applications (fine-grained) may need to be optimized to avoid the bottlenecks that arise due to the communication overhead in symmetric multiprocessing architectures.
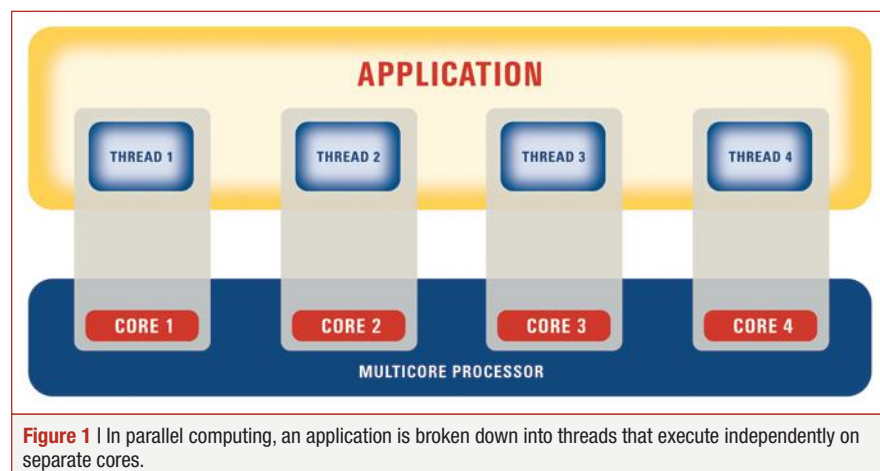
Figure 1 | In parallel computing, an application is broken down into threads that execute independently on separate cores.

POSIX-based OSs provide a rich environment of threading functionality to make it easy for developers to implement parallelism in their applications. Developers must consider the design trade-offs of using multithreading versus non-multithreading to harness the power of multiple processor cores. In some instances, applications may perform better on a single-core system.

Multicore optimization with SMP OS and hypervisor technology

Another approach to multicore optimization centers around choosing an
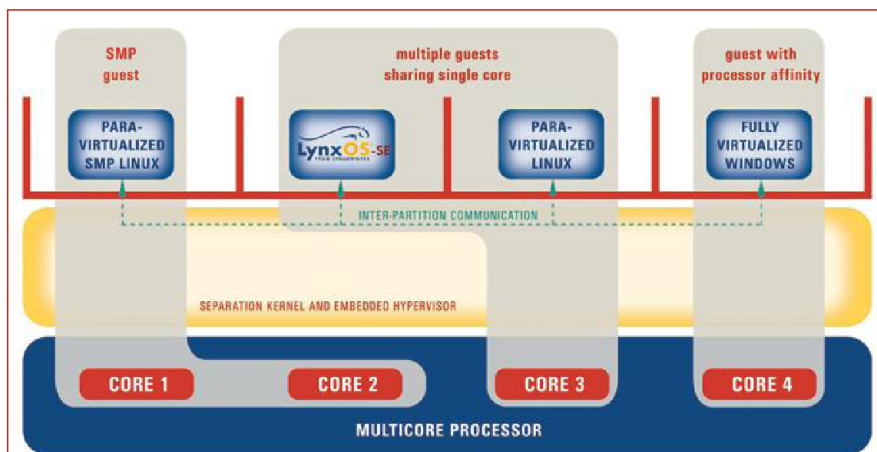
Figure 2 | A Type 1 hypervisor runs directly on the hardware and has complete control of the platform.

appropriate OS. An SMP-enabled OS can help add concurrency to an application by balancing the threads running on multiple CPUs and maintaining a deterministic hard real-time performance level.

But what if you could get even more control over how the OS runs on the multicore CPU? A new trend emerging in multicore environments is the use of a small hypervisor operating system, which abstracts the capabilities of hardware and allows multiple heterogeneous operating system instances to run on a single hardware platform. A Type 1 hypervisor, such as LynxSecure from LynuxWorks (Figure 2), runs directly on the hardware and has complete control of the platform, providing superior utilization of processor resources. In the SMP-enabled hypervisor, a single copy of the hypervisor can allow a single guest operating system to utilize multiple cores. The same hypervisor can enable AMP by allocating a single guest operating system to a unique core. This can be extended to allow AMP and SMP on the same platform through judicious allocation of guest operating systems on single or multiple cores, thereby increasing processor utilization significantly.
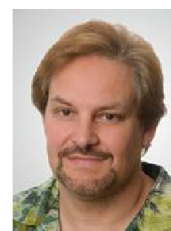
John Blevins is the Director of Product Marketing and Tools Development at LynuxWorks, with more than 25 years of software experience in the embedded industry. Contact him at jb@lnxw.com.

LynuxWorks
408-979-3900
www.lynuxworks.com

Mission
Control the sea

Critical
Cutting-edge reliability

Royal Navy Astute Class nuclear-powered attack submarine.

Thales' periscope provides a 360° scan of the surface above with minimal risk of detection.

Wind River embedded solutions deliver the breakthrough dependability and performance essential to innovation.

To control the sea, a submarine depends on remaining invisible. But when designing and building a sub, visibility is critical. That's why Thales partnered with Wind River to create a breakthrough in periscope design for the Royal Navy's new Astute-Class submarine.

Relying upon the proven innovation, reliability and performance of our VxWorks RTOS platform, Thales developed a state-of-the-art optronic imaging system that provides stable, high-resolution views in the world's most demanding conditions.

It's the kind of teamwork and support that's made Wind River a trusted leading provider of advanced embedded solutions for aerospace and defense.

To see how Wind River can help you innovate with confidence, download our Mission Critical Toolkit at www.windriver.com/missioncritical/security.

WIND RIVER

Effective use of open source in embedded applications

By Espen Bøch

Very specialized and often in-house designed solutions using highly proprietary hardware and software have traditionally dominated the embedded industry. The massive focus on cost reductions and the increased use of COTS products have led to lower hardware costs, but the cost of software remains high. However, if used and managed in the right way, open source software projects can help Military and Aerospace (M&A) system developers complete projects on time and within budget while improving the overall quality and stability of the resulting product.

U.S. Air Force photo by Airman 1st Class Corey Hook

The military embedded industry is in the middle of a massive change of implementation strategy. For many years, military programs were more or less exempt from commercial consideration. Both politicians and the public in general have set focus on the cost of military programs. Moore's law and increased use of COTS products have contributed to lower hardware cost, but software has not experienced a similar cost reduction path.

Can open source software help in getting more out of limited project budgets? According to a recently released VDC report on embedded operating systems[1], the use of Linux in embedded projects shows an annual growth rate of about 50 percent, while new projects using non-Linux operating systems show a steady state. The most interesting observation is that among the projects using Linux, about 80 percent use free public Linux distributions. However, the study also shows that this 80 percent includes the projects with the longest implementation times, indicating that the use of free software does not necessarily reduce the total project cost. Thus, open source projects must be properly planned and managed.

When the benefits of open source software are utilized, the project not only completes on time and within budget, but also results in a more robust and flexible product. The challenges and benefits of open source in M&A applications are discussed, and an application example is presented.

Free open source versus commercial distributions

Even though the software is open source, it does not necessarily have to be free. What are the benefits of choosing a commercial open source distribution over a free alternative? The commercial Linux providers argue for better support, an integrated tool chain, and feature enhancements. Are they right?

Support is a major concern in all projects – and the most compelling argument for choosing a commercial solution. Most organizations have experienced poor support and recognize the cost of this. Unfortunately, even for commercial alternatives, responsiveness is not always a given. It is often hard to get access to the development engineers. The best one can hope for is for the issue to be resolved

in the next official release, which can be months away. For free distributions with an active user group, support questions most often get responses immediately with initial suggestions to workarounds and fixes. Oftentimes a fix is available within hours.

The next argument is: "Commercial alternatives offer an integrated tool chain with cross-compilers, ICE, debuggers, memory management tools, and so on." This is mostly true, but the number of free alternatives increases every day, and the quality of the tools continues to improve.

The area where commercial Linux distributions have the most to offer is feature enhancements. No community-based Linux distributions offer good real-time support currently. Further, DO-178 safety certification, EAL Common Criteria, and MILS are all examples of requirements not likely to be implemented as part of a free Linux distribution. In fact, it is not even clear that any Linux distribution will ever meet these most stringent criteria. For applications targeting these standards, the commercial offerings (Linux or hard RTOS) will continue to be the best bet.

Challenges in open source projects

Given that the choice has been made to go for a free open source alternative, either for the operating system or the application, which challenges must one expect to encounter? How can the project be managed to have the best chances of success?

Project developers must take into account that Open Source Components (OSCs) are to be used and construct the project plan accordingly. Sufficient time must be allocated for selection, evaluation, and verification of OSC.

In his article "Mission-critical development with open source software: Lessons learned," Jeff Norris at JPL[2] describes the most critical factors to consider when designing with Open Source Software (OSS): functionality, maturity and longevity, and verification.

Functionality

The search for suitable open source components can be challenging. Projects where there is a good dialogue between the end users and the development team have the best chances of success. In many cases, an open source component has functionality close to what the end customer wants, but it is not an exact match. Either the requirements can be changed slightly or the open source community can be utilized to implement the missing features. If an acceptable match cannot be found, the alternative is to revert to in-house design for the specific component.

Projects using OSCs are particularly well suited for agile project management, where the project is run in small iterations, adding open source components one by one to build the required functionality. Each iteration starts with a brief analysis of the alternative components. The analysis results in selection of one component that is used to implement a few of the most critical use cases. If testing reveals that the resulting functionality does not meet the project requirements, the next iteration is started with another component. Once selected and properly evaluated, the new functionality can be presented to the customer for approval, and the process is repeated for the next component.

Maturity and longevity

The quality of free open source components is a concern and is often used as an argument to utilize a commercial alternative. The activity in user forums and mailing lists gives a good indication.

How well established is the open source software project? Consider how much time the development team has invested, release history, community activity, bug database, functionality backlog, and so on. Does the project have a test plan? What is the quality of the test plan? How responsive is the development team to feature requests and issue reports? Some open source communities, like SourceForge.net, assign a project ranking, which is a good indicator of the overall project activity. Look for a large and diverse development team. Watch out for single developers or small project communities. These will represent a risk when their priorities change.

Verification

Incorporate the open source component into the project's overall test strategy. Open source components often include detailed unit test and component-level tests far exceeding those common in commercial alternatives. These tests – combined with large, active user groups constantly utilizing, testing, and verifying the code – result in overall increased system reliability. Most in-house designs do not have external review, and many do not even have internal reviews.

Open source developers know that their code will be scrutinized by a number of other community contributors. One of the main motivators for contributing is to get recognition for good and well-written code. In most cases, the result is more readable and better-documented code than what can be found in commercial alternatives. An open source component selection checklist is imperative to the process (Table 1).

Application example: Ultra-small rugged NAS

A real-world example of the process described is Galleon Embedded Computing's recently announced Ultra Small Rugged NAS product. The requirements were nearly met by several open source NAS alternatives. An Intel hardware platform was chosen since most open source projects are developed for Intel Architecture. A Web search revealed a number of open source projects that were close to meeting the requirements: FreeNAS (BSD based), CryptoNAS and NASLite-M2 (Linux, commercial based), Turnkey Linux NAS, Gluster and OpenFiler (Linux, community based), to name a few. After a brief survey of the most promising alternatives, a more thorough evaluation was done of the two candidates that most closely met the requirements specification: FreeNAS and OpenFiler.

The two candidates are based on FreeBSD and Linux, respectively, and both alternatives met the specification with respect to functionality. SourceForge.net front-page statistics for the two projects as of December 1, 2010 indicate that FreeNAS has a much bigger user group with almost 30,000 weekly downloads totaling more than 1 million downloads in 2010. This compares to 300,000 downloads for OpenFiler, which is still sufficient to satisfy the criteria of a large and diverse user community.

Project parameter	Checklist
Functionality	• Specification compliance • Responsiveness to feature requests • Ease of integration with existing features
Maturity	• Community project ranking • User forum activity • Release history • Functionality backlog • Bug database
Longevity	• Size of project community • Responsiveness to issue reports • Community diversity
Verification	• Project test plan exists • Quality of test plan • Verification history • Bug database • Revision history

Table 1 | Open source component selection checklist

Reviewing the OpenFiler backlog, it became evident that few of the issues had been addressed recently, with the latest release based on a Linux core several generations old. The latest update was almost five months old at the time of review. The FreeNAS project, on the other hand, had frequent updates, a very active user forum, and good responsiveness to a few test inquiries. The many contributors have a diverse background, and the project has a good ranking with 95 percent of the 700-plus users providing "thumbs up" feedback. A review of the negative feedback did not reveal any issues considered critical to the project.

Using an agile project process based on the aforementioned analysis, developers started the project by using FreeNAS in the first three iterations. These iterations focused on testing the file-system performance on solid-state disks, testing the network performance through four 1 Gb and two 10 GbE connections, and testing the functionality of the Web GUI. All test results showed that FreeNAS complied with the project requirements. Even so, to ensure the best possible performance of the final product, two more iterations were conducted to test the file system and network performance of OpenFiler for comparison. The recommended component selection process is depicted in Figure 1.
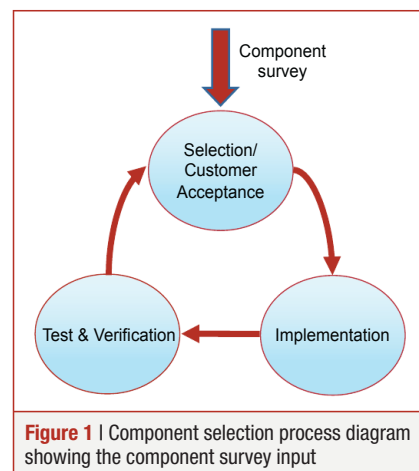
Figure 1 | Component selection process diagram showing the component survey input

After reviewing the results of the performance tests for both alternatives, FreeNAS was chosen as the optimal solution for this project. Building the required functionality using in-house designed software would have taken years to complete, and would never have reached the same level of robustness and stability as the selected solution. Using FreeNAS, in less than three months the basic NAS functionality was ready, and the design team could start implementing specialized features not available in the standard distribution, such as GPS timeserver synchronization.

References:
[1] www.vdcresearch.com
[2] Jeff Norris, "Mission-critical development with open source software: Lessons learned," IEEE Software, January 2004

Espen Bøch is VP Sales & Marketing at Galleon Embedded Computing. He has more than 15 years of experience including design and project and program management at VMETRO and Curtiss-Wright Controls. He can be contacted at espen@galleonembedded.com.

Galleon Embedded Computing
0047 4777 3133
www.galleonembedded.com

Intel Architecture enables digital signal processing

By Peter Thompson and Peter Carlston

▶▶ ▶▶

Historically, system designers have tended not to consider Intel's offering for DSP applications. With the advent of the latest architectural innovations, that seems set to change.

Military and aerospace applications have an insatiable need for processing power, but often have hard power limits. Many viable technologies exist, including DSP chips, FPGAs, and others, but in recent years developers have become accustomed to the ease of programming that comes with the use of general purpose processors, typically supplemented by integrated vector processing engines. Now, with a number of architectural features that are introduced in the latest Intel Core i7 processor products, there is a convergence of ubiquitous availability, ease of programming, and raw GFLOPS performance that makes a compelling case to consider Intel architecture chips for the most challenging signal- and image-processing requirements that mil/aero has to offer.

Considering the road map

Traditionally, the x86 family was not particularly adept at processing streams of Floating Point (FP) data. In fact, the original Intel 8086 and successors did not have an inherent FP capability. That was left to an add-on dedicated Intel 8087 FP chip or slow software emulation. By the time the Intel Pentium processor era arrived, things had changed somewhat with the introduction of the MultiMedia eXtension (MMX technology) instructions and execution unit. This was primarily targeted at encoding and decoding various audio and video streams, but with a little ingenuity could be used for other signal-processing tasks, albeit with a limit to integer operation. Over time, MMX technology begat Intel Streaming SIMD Extensions (Intel SSE), which extended the instruction set to be ever more DSP-friendly, including floating point support.

Along the way, a non-x86 device – the Intel i860 – emerged. This was designed as a graphics device, but an embedded industry that was starved for performance discovered that it was actually pretty darn good at DSP. Unfortunately, it was a niche product for Intel when compared with the mass markets that the x86 line served, and eventually, after a couple of generations, it fell by the wayside.

Fast-forward to late 2010/early 2011, and Intel is starting to ship the second generation of the Intel Core i7 processor line, which implements a microarchitecture codenamed "Sandy Bridge." Once again, the SIMD floating-point unit has been updated – to Intel Advanced Vector Extensions (Intel AVX) (see Sidebar 1). New architectural features and migration tools make these devices worthy of consideration for DSP applications.

New architectural features boost DSP performance

The most apparent benefit of Intel AVX is doubled vector-pipeline width. The execution unit and associated register file are both now 256 bits wide, increased from the 128 bits of previous generations. This alone can account for nearly a doubling of floating point vector performance. The execution unit uses Single Instruc-

tion Multiple Data (SIMD) operation to increase throughput beyond what has become an effective wall in terms of what is viable by simply increasing clock rates and reducing die geometries. Clock rates above 2-3 GHz see diminishing returns for the power consumed, and as geometries migrate to 32 nm and smaller, inefficiencies resulting from leakage become more significant. A 256-bit vector pipe-

line can execute eight single precision (32-bit) FP operations concurrently (the same instruction, but on eight different sets of data points), compared with four on SSE implementations. In some cases, two instructions can be executed per cycle, such as when doing a multiply and add at the same time, thus allowing 16 operations concurrently (versus eight with SSE). In reality, each operation takes

multiple cycles, but by employing pipelining, once a startup penalty has been paid, results are available on every cycle. This SIMD operation maps naturally to many DSP algorithms as they tend to feature the required data parallelism.

There will be members of the new Intel Core i7 processor family with two and four cores featuring the Sandy Bridge architecture. Each core has its own Intel VX unit. This means that a quad-core version can potentially execute 64 single precision FP operations every clock cycle. Contrast this with the 16 operations per cycle on first-generation, dual-core i7 devices.

New instructions such as broadcasts and masked loads and stores enable better utilization of the available FLOPS. Changes to the memory unit allow for two read requests and one write of 16 bytes each in one cycle. This is a key feature in keeping the execution unit fed with data and avoiding the pipeline stalls that can severely impact performance. These features serve to close the efficiency gap that existed between AltiVec and SSE in

some DSP algorithms that required data reorganization for efficiency. Figure 1 shows some of these features.

Hence, the key elements in deploying these processors in mil/aero signal- and image-processing applications include:

■ The availability of BGA devices, which allow the components to be soldered down, rather than using socketed devices that can fail under high levels of shock and vibration

■ Intel's seven-year life cycle for parts on the embedded platform road map

Harnessing DSP application performance

So what does all this mean for performance? In an attempt to answer that question, several DSP applications have been run on previous generation i7 class devices codenamed "Arrandale" and on Sandy Bridge. When comparing performance using a single core, with both classes operating at the same clock rate, an increase in performance of approximately 2x has been demonstrated with Sandy Bridge. This illustrates both the theoretical speedup that would be expected from doubling the SIMD unit width, and
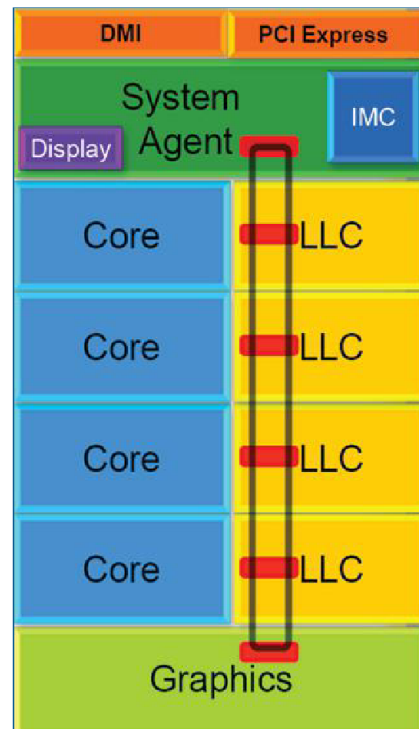
Figure 1 | The microarchitecture of Intel's Sandy Bridge lends itself well to DSP applications.

also the balance of memory access needed to achieve that increase.

All this performance is great, but how can it be exploited for application domains such as radar, SIGINT, ELINT, and so on? At the lowest level and highest complexity is programming the Intel AVX unit using primitives that can be called from C or other high-level languages. While no more complex than the assembly code programming that many programmers cut their teeth on in the past, getting good performance at this level is not a trivial task. Many factors must be understood and factored in when coding to avoid pipeline stalls and resource contention.

Compilers offer some help. Several already have Intel AVX support, coupled with varying degrees of automatic vectorization. Source code is analyzed, and where possible, for-loops are mapped to Intel AVX SIMD operations. This can be a help with dusty-deck code, but in reality there are many impediments that can preclude effectiveness, as the compiler must always err on the side of caution. Where ambiguity exists over loop iterators, for instance, the compiler cannot make assumptions that it cannot verify, so it will always generate lower-performance, guaranteed-correct code.

Math libraries offer a good alternative. Intel Integrated Performance Primitives (Intel IPP) and Intel Math Kernel Library (Intel MKL) are highly tuned for Intel AVX. Algorithm coverage is broad, and performance is hard to beat. However, in some eyes, they suffer from being seen as proprietary Application Programming Interfaces (APIs). Also, support for operating systems tends to stick with the mainstream of Windows and Linux. Support for real-time operating systems common in the embedded world is currently lacking, for the most part.

As an alternative, many programs turn to more open standard libraries, such as the Vector Signal- and Image-Processing Library (VSIPL) sponsored by DARPA as a cross-platform, cross-vendor API, and VSIPL++, its C++ sibling. These libraries can help isolate applications from underlying architectures. For example, many applications written for the AltiVec SIMD unit on Freescale devices can be migrated to Intel AVX on Sandy Bridge with a recompile. Under the covers, VSIPL may be implemented with handcrafted Intel AVX code, or may take advantage of Intel IPP/Intel MKL, or may be a mixture of both. In any case, the end result is highly tuned performance at the application level without the coder having to understand the low-level nuances of the chip. They can instead focus on their domain of knowledge.

I7 today and tomorrow

Since the introduction of SSE, Intel processors have been used for signal- and image-processing applications. With the introduction of AVX, this adoption seems set to accelerate. The benefits that this architecture brings to raw performance and to GFLOPS/Watt metrics are immediately apparent. Additionally, given Intel's track record on die shrinking new architectures (tick-tock development), this can be expected to continue. GE Intelligent Platforms, a member of the Intel Embedded Alliance, has many Intel i7 products shipping today (Figure 2) and will be introducing new products as soon as the new Sandy Bridge architecture processors are released.
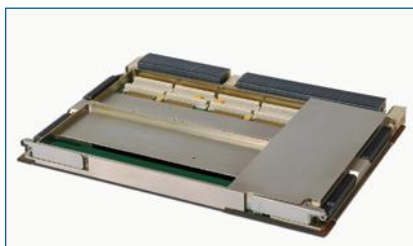
Figure 2 | GE's SBC622 is a 6U OpenVPX single board computer featuring the Core i7 processor.

Peter Thompson is Director of Applications, GE Intelligent Platforms. He can be contacted at peter.thompson@ge.com.

GE Intelligent Platforms
978-437-1477
www.ge-ip.com

Peter Carlston is Platform Architect, Embedded Computing Division, at Intel Corporation.

Intel Corporation
www.intel.com
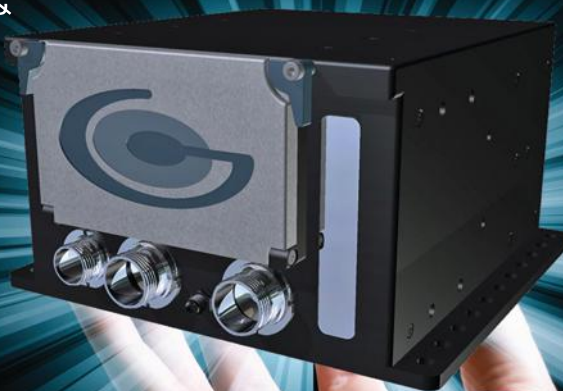
2011: Top technologies for the warfighter

By Chris A. Ciufo, Editor

Coming to the military soon is the same technology empowering the civilian consumer market as it migrates from desktops to Netbooks and Tablets. Meanwhile, TV becomes an interactive part of "any available screen" fed by computing power stored in the Internet cloud. Oh, and that Verizon or AT&T iPhone in your pocket? Yep: The DoD's gotta have one of those, too. Tie it all together with wireless networks and embedded security, and that becomes "Warfighter, 2015." That is, today's consumer tech will be deployed in military platforms well before 2015. Here are some of my predictions for the most likely "gotta have it" technologies for tomorrow's embedded military systems.

1. Intel second-generation Core CPUs

At January 2011's CES in Las Vegas, Intel rolled out their second-generation Core family, with additional performance improvements including more cores, higher frequencies, and dramatically improved graphics. This author was ready to buy a new laptop with *last* year's Cadillac model Core i7 dual-core CPU. The plan now is to wait on hardware shipping with the 45 W, 2.30 GHz Core i7 2820QM with 8/4 (cores/threads) and 3.4 GHz Turbo Boost Technology 2.0, Intel AES/TXT/vPro, and Advanced Vector Extensions. (Figure 1 shows **Intel's 2nd Gen Core CPU lineup**).

All New 2nd Gen Intel® Core™ Mobile Processor (SV) Lineup						
Brand	Core i5	Core i5	Core i7	Core i7	Core i7	Core i7
Processor Number	i5-2520M	i5-2540M	i7-2620M	i7-2720QM	i7-2820QM	i7-2920XM
Price	$225	$266	$346	$378	$568	$1096
TDP	35W	35W	35W	45W	45W	55W
Cores/ Threads	2/4	2/4	2/4	4/8	4/8	4/8
CPU Base Freq (GHz)	2.50	2.80	2.70	2.20	2.30	2.50
Intel® Turbo Boost Technology 2.0 Max Frequency (GHz) SC	3.20	3.30	3.40	3.30	3.40	3.50
DC	3.00	3.10	3.20	3.20	3.30	3.40
QC	N/A	N/A	N/A	3.00	3.10	3.20
DDR3 (MHz)	1333MHz	1333MHz	1333MHz	1600MHz	1600MHz	1600MHz
L3 Cache	3MB	3MB	4MB	6MB	8MB	8MB
Intel® HD Graphics 3000	Yes	Yes	Yes	Yes	Yes	Yes
Gfx Base Render Frequency	650MHz	650MHz	650MHz	650MHz	650MHz	650MHz
Graphics Max Dynamic Frequency	1300MHz	1300MHz	1300MHz	1300MHz	1300MHz	1300MHz
Intel® Hyper-threading Technology	Yes	Yes	Yes	Yes	Yes	Yes
Intel® Advanced Vector Extensions (AVX)	Yes	Yes	Yes	Yes	Yes	Yes
Intel® Quick Sync Video	Yes	Yes	Yes	Yes	Yes	Yes
Intel® AES/TXT/vPro Technology	Yes	Yes	Yes	Yes	Yes	Yes
Intel® Virtualization Technology	Yes	Yes	Yes	Yes	Yes	Yes
Package	rPGA/BGA	rPGA/BGA	rPGA/BGA	rPGA/ BGA-1224*	rPGA/ BGA-1224*	rPGA

Information updated January 4, 2011
For updated processor pricing http://www.intc.com/priceList.cfm
For processor specifications http://ark.intel.com/

Figure 1 | Intel's 2nd Gen Core CPU lineup (SV laptop family shown)

Loosely translated, this particular CPU is one down from Intel's best Mobile (SV) processor family, although lower-power (LV/ULV) versions exist with weaker specs. Gen 2 is the Sandy Bridge architecture and instruction set that builds on last year's Nehalem series. Turbo Boost 2.0 ramps up the clock speed above max power and schedules un- or under-used cores and

cycles. AES/TXT (Trusted Execution Technology) are built into Intel hardware to facilitate security, and vPro provides remote management and security. Meanwhile, AVX renders a floating-point oriented 256-bit instruction set extension to SSE that I'm convinced will give Freescale's AltiVec serious headaches.

How does all this *really* apply to defense applications? All that cool consumer multimedia gear is applicable to C4ISR, simulation and training, and secure red/black DoD platforms.

2. Rugged shoeboxes

A perennial favorite, the military has moved from conduction-cooled, open-standard, backplane-equipped ATR boxes to purpose-built, rugged "shoeboxes." Recent examples include Kontron's Cobalt with 38999 connectors, 50,000-foot operational environment, and small "boxed PC" fanless design. No one cares what format the cards are inside; military designers only want the right performance to fit in the available envelope. In Cobalt's case, it's a Computer-on-Module (COM) design, with common I/O in the baseboard and the CPU flavor of the month on top. In Kontron's case (no pun), it's a Core 2 Duo or Atom. But the box can dissipate close to 100 W with derating, says Product Manager David O'Mara, which puts a 2nd Gen Core i3 or i5 at 17-20 W within the realm of possibility.

Destined for UAS/UAV programs, these rugged shoeboxes are cheap, optimize SWaP, and focus on machine-to-machine computing with the office-like features of Intel vPro, GbE, and extra graphics muscle via NVIDIA GPUs or AMD Fusion microcontrollers. If a PC, graphics engine, or network controller is all that's needed, then a rugged shoebox is likely to be the preferred choice over a VME, CompactPCI, or OpenVPX ATR-style chassis.

3. Cloud computing

Though the X-Windows phenomenon of the 1990s failed, the promise of a cheap, local, lightweight processor and graphics engine with an OS supported by heavyweight distant processing is a reality today. The Internet offers fast enough pipes to most desktops, with even 3G cellular speeds at nearly 1 Mbps and future 4G well in excess of that. To the desktop, wired connections exceed 5 Mbps and top out at double digits for fiber-to-the-home such as Verizon's FIOS. Software companies have responded with more processing horsepower in the browser itself; Google's Chrome includes advanced Java, while Microsoft's latest IE includes support for patented HTML5 and H.264 decoding. In civilian tech, a *thin client* is key to saving money and easing IT management. And the DoD faces the same challenges. Rugged, full-spec laptops like those from Panasonic or GD Itronix are expensive, while cheaper Netbooks or even

---

[1] There's no obvious P/N differentiation between Gen 1 and Gen 2 CPUs except for the "2xxx" nomenclature. Previous Core CPUs didn't have the "2" prefix.

Android Tablets may do the trick of displaying recon video or moving maps, filing action reports, or requesting call-for-fire support. Says virtualization expert Bill Parker of Logicallis, a company selling Virtual Desktop Interface (VDI) software, VDI offers hard-cost savings, management ease, centralized back-ups, regulatory compliance, and productivity. Siamak Farah, CEO of cloud services company InfoStreet, has observed some of the same desktop and enterprise trends, but disagrees that putting resources in the cloud requires a VDI-type model. In fact, he predicts that cloud computing will be mainstream in the civilian world due to the advent of smartphones or Tablets running Google's Android or Apple's iOS. Software developers are likely to create more "Apps" (what he calls *mini software*) that will become Web-available. Apple's App Store for the Mac OS (different from the iPhone) was followed quickly by Microsoft's own announced but unnamed at presstime "App Store." And the company's own Steve Balmer predicted in April 2010 that "90 percent" of Microsoft's customers would be entirely cloud-based.

The thin client, cloud computing model – run on secure DoD networks such as the Navy's NMCI – is likely to prevail for a handful of dismounted and many command post and data center computer platforms. Heavyweight embedded computers will still control mission and fire control boxes, along with sensor processors, but PC-like platforms will increasingly become thin clients, supported by processors and software in the DoD cloud.

4. Google
At 83 percent worldwide market share in search, according to analyst firm iSuppli, Google's server farms' compute horsepower rivals that of Amazon's Elastic Compute Cloud (EC2). From Google Docs to Google Earth, Google lists 47 products or categories on its website, and many of these are finding – or could find – their way into deployed military systems. For instance, Picasa's facial recognition software organizes your photo collection by a person's name. Can that same technology be applied to find insurgents in Global Hawk video feeds? Google Earth's zoom capability and image enhancement rely on state-of-the-art algorithms, as does Google Maps' ability to download the appropriate chunks of database information to display maps, street views, or traffic-plus-roadway information in a mash-up. Google Voice technology provides excellent-quality VoIP packet processing, and Google's server farms employ adaptive processing and echo cancellation that rivals some DoD contractors' dedicated packet processing engines.

So although Google gives away their stuff for free, some of their products and technologies have incredible utility in military applications. Just wait until Google offers to provide enterprise-quality *secure* networks.

Honorable mentions
The following honorable-mention technologies (in no particular order) have definite military applicability and are already deployed on the battlefield or soon will be:

5. Cisco's Radio Aware Routing
Part of Cisco's Mobile Ready Net and offered on the company's 3200 and 5940 (3U CompactPCI, air- and conduction-cooled) deployed routers, the concept of *ad hoc* routing applies cognitive radio techniques to route data across whatever battlefield assets are available. Radio Aware Routing (Figure 2) is part of RFC5578, PPP over Ethernet (PPPoE) Extensions for Credit Flow and Link Metrics.
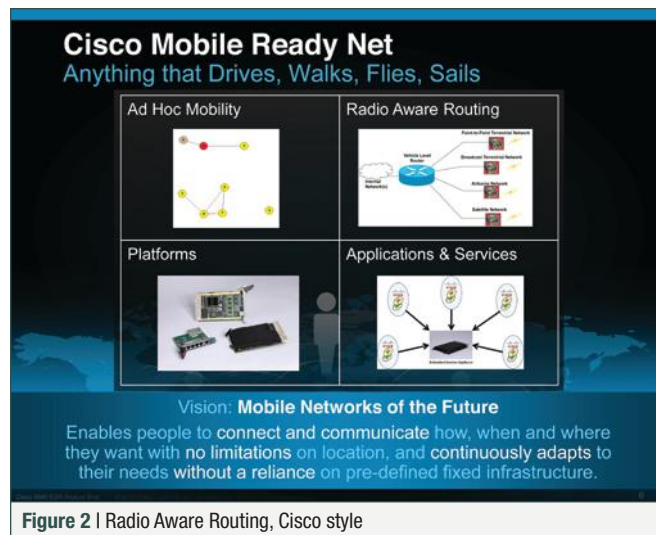
Figure 2 | Radio Aware Routing, Cisco style

6. Wind River's Linux Secure
This is the first Common Criteria EAL 4+ security-certified Linux distro, applicable for better-than-desktop Linux DoD systems. If Linux is the OS of choice, Linux Secure is worth a look.

7. Automated video analysis software, from vendors such as Kitware, IvySys, and others
These heavy-duty programs rely on beefy processors to scrub through video (and other) data looking for preprogrammed elements or events. The intent is to reduce operator workload and errors when scanning hours of UAS video.

8. Tablet PCs
All the rage at the 2010 and 2011 CES shows, all major laptops vendors – and many new players, too – hope to capitalize on Apple's iPad's success. Most of the new Tablets won't be in production until the 2011 holiday season. Wikipedia, ZDNet, and myriad other consumer media sources have their own *Tablet Roundup* Web pages. Meanwhile, Apple's current iPad is showing up in military technology demonstrators, and iPad Apps are in the B&D stage at several primes.

9. Xilinx/AutoESL synthesis tool
The notable prediction here is not the FPGA itself, rather, it's the shift in FPGA design from HDL to high-level languages. As I predicted in 2010, Xilinx sees value in programming in C/C++ as a way to get more people designing with FPGAs. As we went to press, Xilinx purchased AutoESL Design Technology, their partner in programming tools. The *AutoPilot* tool will "enable the company to deliver the benefits of programmable platforms to a broader base of companies." What does all this mean to the defense industry? Design and deployment just got a whole lot faster.

10. OpenVPX
So Mercury Computer Systems' 2009-initiated, outside VITA, OpenVPX Industry Working Group produced the spec later ratified by VITA and ANSI as *VITA 65*. Today, OpenVPX is showing up in wares from compact rugged systems to DSP engines, SBCs, PCIe cables, chassis, power supplies, and just about everything else VPX you can name. In light of OpenVPX's original goal of defining interoperable VPX system-level specs leading to interoperable VPX LRUs, OpenVPX's modern-day benefits are plenteous: fewer frustrated VPX systems engineers, more rugged, and more (2-Level-Maintenance) technologies on the battlefield. Bring it on: "Warfighter, 2015."

Domain-specific property checking with advanced static analysis

By Paul Anderson

Advanced static analysis tools are well known for being good at finding generic defects in programs. They can also be extended by end users to find domain-specific errors unique to a particular application, thanks to customized property checkers.

Advanced static analysis tools use sophisticated whole-program, path-sensitive techniques to find deep semantic problems including safety defects and security vulnerabilities. Several of the defects identified by these tools can be found in the CWE/SANS Top 25 most dangerous programming errors (http://cwe.mitre.org/top25), which lists the most important bugs to avoid in order to reduce security risk.

Specifically, out of the box, advanced static analysis tools can find generic programming defects. A typical generic report from a static analysis tool might show warnings such as uninitialized variables, buffer overruns, unreachable conditionals, and unreachable calls, among many others; however, not all security vulnerabilities or safety bugs are caused by instances of generic problems. Many defects are unique to a particular application.

An often underappreciated aspect of these tools is that they are extensible, so they can often be configured or programmed to find violations of domain-specific rules, too. So if a development team has its own internal rules for using a proprietary API, or requires programmers to use a particular idiom, then it is often possible to write a checker that signals violations of those rules. Hence, programmers can, often with little programming effort, dramatically increase the value they get from the tools.

A common use case is when a bug occurs and triggers a long and painful diagnosis and debugging period. Once the defect has been located, the wise action is to first find other places in the code where the error was repeated, then to take steps to detect the bug if it reoccurs. A custom check to be run on both existing code and new code can be an effective, low-cost way to achieve this. How these tools work and how their checkers can be customized is described using code snippets from CodeSonar, though the principles can be utilized with other advanced static analysis tools, too.

How advanced static analysis works

To understand the different ways in which extensions can be written, it first helps to know what static analysis is, and how static analysis tools work under the hood.

Static analysis tools work much like compilers. They take source code as input, which they then parse and convert to an Intermediate Representation (IR). A block diagram of the architecture of an advanced static analysis tool is shown in Figure 1. Whereas a compiler would use the IR to generate object code, static analysis tools retain the IR, and checkers are usually implemented by traversing or querying the IR, looking for particular properties or patterns that indicate defects.
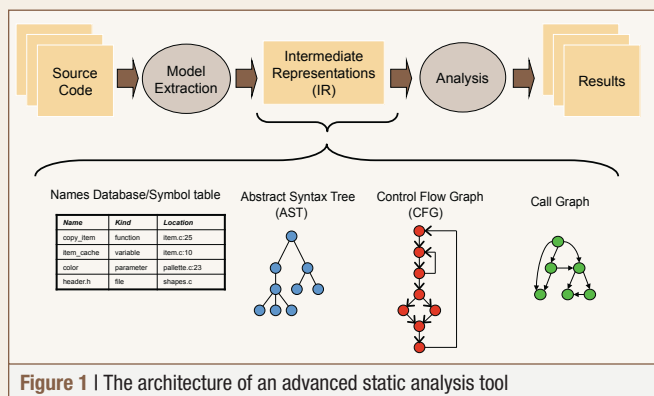
Figure 1 | The architecture of an advanced static analysis tool

The advanced tools get their power from sophisticated symbolic execution techniques that explore paths through the control-flow graph. These algorithms keep track of the abstract state of the program and know how to use that state to exclude consideration of infeasible paths.

Checkers for domain-specific properties can access these representations and exploit the analysis algorithms in various ways.

Custom checkers

How does an end user author a custom property checker? This depends strongly on the nature of the property, so several mechanisms are available:

■ Existing checkers can be extended by adding directives to a configuration file.

- The user can add annotations to their code that instruct the analysis to look for certain properties. These annotations can be done on the side in an aspect-oriented fashion if users do not wish to perturb the source code.
- An API allows users access to all of the intermediate representations. Typically, a visitor pattern is used that allows extensions to exploit traversals the analysis is already doing.

Let's take a closer look at these mechanisms.

Configuration files

These advanced static analysis tools implement dozens of checkers. Often, a user needs a checker that is only slightly different from a built-in one, and many of them have been designed to be extensible. One class of checkers finds functions whose use is forbidden. For example, the C library function **gets** is notoriously insecure. The checker is implemented by a phase that finds references to function names, and then matches them against a set of regular expressions. It is a simple matter to add additional regular expressions to this set by adding lines to a configuration file.

Code annotations

The second way to write checkers is to add annotations to the code.

Suppose, for example, that there is an internal function named **foo** that takes a single integer parameter, and that a potential security vulnerability is introduced when this parameter is -1. A check for this case could be implemented by adding some code to the body of **foo** as follows:

```
void foo(int x)
{
#ifdef __CSURF__
    csonar_trigger(x, "==", -1,
            "Dangerous call to foo()");
#endif __CSURF__

    ...
}
```

The **#ifdef** construct ensures that this new code is not seen by the regular compiler. However, when this code is analyzed by the tool, the call to **csonar_trigger** is seen. Thus, this call is never actually executed, but is instead interpreted by the tool as a directive to the underlying analysis engine. If the analysis concludes that the trigger condition may be satisfied, then it will issue a warning with the given message.

Most programmers prefer not to clutter their code with such annotations, so there is an alternative way to implement this kind of check that allows it to be written in a separate file. This approach is also appropriate when the source code for **foo** is not available, such as when it is in a third-party library. To do this, the author of the checker would write a replacement function as follows:

```
void csonar_replace_foo(int x)
{
    csonar_trigger(x, "==", -1,
            "Dangerous call to foo()");
    foo();
}
```

When the analysis sees the definition of **csonar_replace_foo**, it treats all calls in the code to **foo** (except the one inside **csonar_replace_foo**) as if they were calls to **csonar_replace_foo** instead.

This trigger idiom is good for checking temporal properties, particularly sequences of function calls. Suppose there is a rule that says that **bar** should never be called while **foo** is executing. A check might be implemented as follows:

```
static int foo_is_executing = 0;
void csonar_replace_foo(int x) {
    foo_is_executing = 1;
    foo(x);
    foo_is_executing = 0;
}
void csonar_replace_bar(void) {
    csonar_trigger(foo_is_executing,
        "==", 1,
        "Call to bar from foo");
    bar();
}
```

Note that a global state variable is used to record whether or not **foo** is active. Before entry to **foo** it is set to one, and then reset to zero after **foo** returns. This variable is then checked in the trigger in **bar**, and if set to one, a warning will be issued.

The previous example shows how a global property might be checked. The same mechanism can be used to write checks for properties of individual objects. Attributes can be attached to objects to track their state.

This approach allows users to author static checks almost as if they were writing dynamic checks. The API for this kind of check is small, and the language is C, so there is a gentle learning curve. This simplicity is deceptive: The technique can be used to implement quite sophisticated checks.

API for intermediate representations
The final way to implement a custom check is to use the API that gives access to the underlying representations (the IR). One company used this API to find violations of a custom idiom for handling hardware errors.

This API can be used for other program analysis tasks, too. For example, a medical device company uses one tool primarily to not only identify potential tasking problems in their code, but also to emit information that allows them to interactively explore properties of stack configurations. Other applications include gathering custom metrics on the code and generating input to program visualization tools.

Many checks can be written using a visitor pattern – a function invoked as a callback on every IR element of a given type. Visitors can be defined for files, identifiers, subprograms, and nodes in the control-flow graph and the syntax tree. The interface to a visitor allows for pattern matching against the construct.

CodeSonar has a special kind of visitor, which is invoked during the path exploration of the control-flow graph. This allows the checker author to write sophisticated checks that leverage the built-in path-sensitive program analysis capabilities of the tool. A key aspect of this kind of checker is that it uses the part of the analysis that eliminates the exploration of infeasible paths, which automatically reduces the probability of false positive results.

Increasing software quality
Advanced static analysis tools have become essential tools for software developers because they are capable of finding serious security and safety defects. Authoring custom checkers is a low-cost way to multiply the effectiveness of these tools.

Paul Anderson is VP of Engineering at GrammaTech, Inc. He received his B.Sc. from Kings College, University of London and his Ph.D. from City University London. He has worked on software analysis and transformation tools for 20 years. He has helped organizations such as NASA, the FAA, Lockheed Martin, and Boeing apply automated code analysis to improve the quality and security of their software. He can be contacted at paul@grammatech.com.

GrammaTech, Inc.
607 273-7340
www.grammatech.com

10G SERIES RECORD & PLAYBACK SYSTEMS
ULTRA-FAST, SCALABLE & 10 GIGABIT NETWORK POWERED

Synchronized (Phase-Coherent) Recording & Playback of Multiple Wideband Radio Signals

DTA-2300

4 X 10 GbE Data Links

10G/R4
- 8 or 16 IF Inputs & Outputs
- 38.4 TBytes Storage Capacity
- 2.4 GBytes/s Sustained Record/Playback Rate
- Optional DTA-3200 (20MHz to 6GHz) tunable RF Front End

4 x DTA-5000

DTA-2300

2 X 10 GbE Data Links

2 x DTA-5000

10G/R2
- 4 or 8 IF Inputs & Outputs
- 19.2 TBytes Storage Capacity
- 1.2 GBytes/s Sustained Record/Playback Rate
- Optional DTA-3200 (20MHz to 6GHz) tunable RF Front End

DTA-2300

10 GbE

1 x DTA-5000

10G/R1
- 2 or 4 IF Inputs & Outputs
- 9.6 TBytes Storage Capacity
- 0.6 GBytes/s Sustained Record/Playback Rate
- Optional DTA-3200 (20MHz to 6GHz) tunable RF Front End

(User PC)
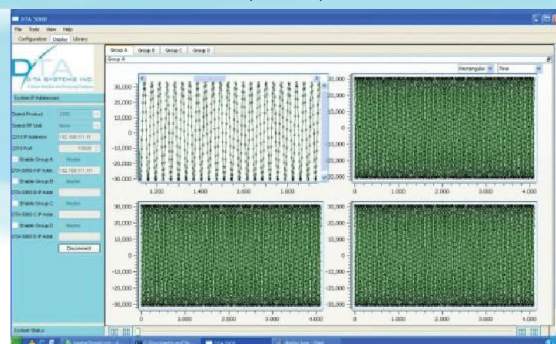
Comprehensive GUI for Failsafe Operation & Control of All System Components. Time, FFT and Waterfall Displays for Monitoring Data Quality During Record or Playback.

With 16-bit 130MHz ADCs & 500MHz DACs and programmable DDCs and DUCs, the 10G Series Systems offer high-precision and large IF Bandwidths. They are fully-integrated for immediate deployment.

1-877-382-3222
SALES@D-TA.COM

Download 10G Series Technical Presentation
WWW.D-TA.COM

NOTHING ELSE COMES CLOSE

D-TA
D-TA SYSTEMS INC.
A Sensor Interface and Processing Company

SENSOR PROCESSORS THAT DRASTICALLY REDUCE DEPLOYMENT TIME AND COSTS

Justifiably taboo:
Avoiding malloc()/free() APIs in military/aerospace embedded code

By Steve Graves

Want to start a lively – even contentious – discussion among programmers? Just ask, "Is it safe to use dynamic memory allocation?"

Popularized in C/C++, dynamic allocation eases development by doling out system memory to application processes as needed at runtime and retrieving the memory when it is no longer needed.

But dynamic allocation is widely considered taboo in safety-critical embedded software. The use of the C runtime library's malloc() and free() APIs, which do the grunt work of dynamic allocation, can introduce disastrous side effects such as memory leaks or fragmentation. Further, malloc() can exhibit wildly unpredictable performance and become a bottleneck in multithreaded programs on multicore systems. Due to its risk, dynamic memory allocation is forbidden, under the DO-178B standard, in safety-critical embedded avionics code.

Developers across the embedded industry seem to react viscerally to the topic. In a recent Internet technical group discussion, the question, "Do you use dynamic memory allocation in your embedded design?" garnered an astonishing 77 responses, typified by "generally, it is considered a violation of best practices" for fault-tolerant systems, and "if the requirements include 'five-nines' (99.999 percent uptime) reliability, hard real-time, or small memory footprint, the answer is 'never.'" Job seekers, take note: One consulting

engineer's interview strategy "is to gently probe the prospective employee on their dynamic memory allocation usage in real-time apps. If they have no problem with it, they are not going to be hired."

A better strategy – both for code safety and job interview success – is to replace the standard (default) allocators in safety-critical code with customized memory allocation functions that more closely match specific allocation scenarios. The following discussion describes two such custom memory managers: a stack-based allocator and a thread-local allocator. Another way to rid an application of malloc() and free() – and thus gain better performance, stability, and predictability – is to replace code based on standard allocation functions with off-the-shelf software that incorporates custom allocators. Use of an In-Memory Database System (IMDS) is discussed as an example of this "buy rather than build" approach.

It's standard, but is it the best?

Why are standard (dynamic) memory managers a poor choice for mission-critical code? Typically they are based on list allocator algorithms that organize the memory pool into contiguous locations (free holes) in a singly linked list. The allocator then "walks" this chain looking for the appropriate hole to meet a request.

List allocators are the quintessential general-purpose function: They do a pretty good job of allocating and deallocating memory across a wide range of situations – but "pretty good" is not good enough in mission- or safety-critical systems.

Stack-based algorithm: Allocate and rewind memory

Certain application scenarios call for allocating many short-lived objects, then freeing them all at once. A stack-based allocator (not be confused with the application call stack) is one type of custom allocator that works well here. With this algorithm, each allocation returns the address of the current position of the stack pointer and advances the pointer by the amount of the request (Figure 1). When memory is no longer needed, the stack pointer is rewound. Processing overhead is reduced because there is no

Figure 1 | A custom stack-based allocator
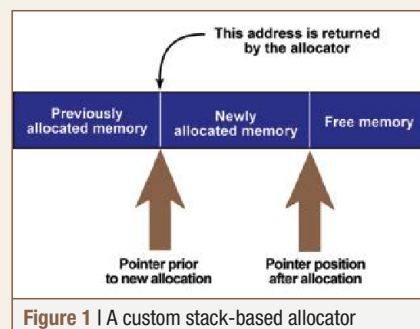
chain of pointers to manage, nor are there any allocation sizes or free holes to track. This approach is safer, too: A memory leak can't be accidentally introduced through improper deallocation because the application does not have to track specific allocations.

The overhead eliminated by using a stack-based allocator versus a standard list allocator increases as the application continues to run. When memory is deallocated in random order, the list allocator often needs to add both a pointer and a size value to its chain (this is called *fragmentation*), so that pointers and size values represent an ever-larger percentage of total heap size. So the list allocator's overhead (the amount of meta-data that must be managed and the likelihood of having to walk further to find a suitable free hole) grows as the application continues to run. (With the stack-based allocator, all chunks allocated from a point in time are returned to the heap in one action, avoiding fragmentation.)

The multithreaded, multicore allocation challenge

The default malloc() and free() functions that are controlled by a mutex are often to blame when multithreaded applications bog down on multiprocessor hardware. Threads using these allocators can cause locking conflicts, and the OS resolves these in part via performance-draining context switches. A custom thread-local allocator avoids conflicts by assigning a specific memory pool to each thread. The thread's allocation is performed from this block without interfering with other threads' requests, thus enhancing performance and predictability. When a thread allocator runs out of memory, some other allocator can assign it another block if the system allows it. The thread-local allocator uses a *Pending Request List* or PRL for each thread to coordinate the release of memory blocks that are freed by a thread other than the one that performed the original allocation. Memory that is allocated and deallocated by the same thread requires no coordination, and therefore no lock conflicts occur.

In short, problems are avoided in safety-critical code by removing memory management responsibility from malloc() and free() and assigning it to the application, which uses custom allocators that mesh with specific application tasks. The custom allocator sets aside a buffer for the exclusive use of that task, usually during boot-up, and satisfies memory allocation requests from it. If the buffer memory

runs low, the application is informed and can free up memory inside the buffer. Or it can find more memory elsewhere to devote to the task. Exhausting the memory in this dedicated pool has no impact on other parts of the system. Custom allocators that might be chosen include the ones discussed, as well as bitmap allocators, block allocators, and others.

Allocation via third-party application

The benefits of custom memory allocators can also be harnessed by integrating third-party software that uses them. IMDSs are a good candidate to benefit from custom allocators, because they are designed expressly to manage application objects

in RAM. Figure 2 illustrates allocation/deallocation using malloc() and free().

```
Structure definition:

typedef struct _Sensor {
    unsigned int s_id;
    char sub_nbr [16];
}Sensor_t, *Sensor_p;

Function:

foo()
{
Sensor_p Sensor;

// instantiate a Sensor structure
Sensor = (Sensor_p) malloc(sizeof(Sensor_t);

// use it…
Sensor-> s_id = 99999;
strcpy(Sensor-> sub_nbr, "something");

// destroy it when you're done
free(Sensor);
}
```

Figure 2 | Memory allocation using malloc() and free()

Figure 3 shows the same process using McObject's eXtremeDB, an IMDS that incorporates custom allocators, including stack-based and thread-local. At the start of Figure 2, a C program defines a structure, declares a pointer to an instance of that structure, and allocates memory for it via malloc().

The programmer using the IMDS defines classes in a database schema file, which is processed (via a special compiler) to produce a .C file, as well as a .H file that contains type definitions and function prototypes.

If the program that uses malloc/free is multithreaded and threads will share the Sensor object, the developer must implement concurrency control. With an IMDS, concurrency is managed automatically via transactions. Figure 3 shows how a transaction begins (mco_trans_start) and gets a transaction handle.

Calling Sensor_new() claims some of the memory pool dedicated to the IMDS for a new Sensor object. (In a military/aerospace application, a sensor object could represent anything from optical sensors for tracking missile targets to biosensors for defense in chemical warfare or motion sensors to aid in navigating an aircraft.) Sensor_new() returns a handle to the database object, through which the object's values can be written and/or read. In contrast, the C program works directly with the structure's fields, creating the need for concurrent access controls in a multithreaded application.

When the C program finishes using the Sensor structure, free() returns memory to the heap. When the code with the IMDS finishes, the space in the database is relinquished, the transaction ended, and the memory used for the sensor object returned to the dedicated memory pool.

The eXtremeDB IMDS can run low on memory, but this would generate a "database full" error message that can be dealt with by the application. In contrast, memory fragmentation and leakage caused by malloc() and free() could destabilize the entire system. The IMDS offers a mechanism that works "behind the scenes" to allocate and free memory with greater efficiency and flexibility from using multiple underlying allocator types, avoiding the riskiness inherent in malloc() and free().

Custom memory managers, though, are not particular to an IMDS. For example, off-the-shelf code to manage sensor networks is well-suited to a type of custom memory manager called a *block allocator*. While discrete sensor values are not known in advance, these values' size is fixed and known (like a 4-byte timestamp and an 8-byte value), and the block allocator excels in parceling out memory chunks of a predefined size. The stack-based allocator is useful for any computing requirement that can be divided into a first stage in which all the memory is needed, and a second stage in which all the memory is no longer needed. Any program that has to parse some input stream fits this description. For example,

```
Class definition:

class Sensor (
    uint4      s_id;
    char<16>  sub_nbr;
);

Function:

foo()
{
MCO_RET rc;
Sensor theSensor;
mco_trans_h t;// transaction handle

// begin a transaction
rc = mco_trans_start(db,
            MCO_READ_WRITE,
            MCO_TRANS_FOREGROUND,
            &t);

// instantiate a Sensor object
// in the in-memory database
rc = Sensor_new( &t, &theSensor );

// use it...
rc = Sensor_s_id_put( );
rc = Sensor_sub_nbr_put( );

// destroy it when you're done
rc = Sensor_delete( &theSensor );
rc = mco_trans_commit( t );
}
```

Figure 3 | Memory allocation using an in-memory database system

a communication surveillance program might parse a stream of text (spoken words), build a tree of tokens (words or phrases), then perform some post-processing on it. That post-processing could be deciding whether a given word or phrase is relevant within its context.

In fact, it is hard to think of an application type that would not benefit from memory management that is geared to its specific allocation patterns and challenges. Of course, customizing memory management adds another consideration to the already complex task of software development. But software engineers entering the safety-critical arena know the demands and stakes are higher than in consumer or business application development. Writing code that avoids dynamic memory allocation and instead uses one or more custom memory managers is less convenient. But it adds safety and stability, and that's a trade-off engineers of safety-critical systems should embrace.

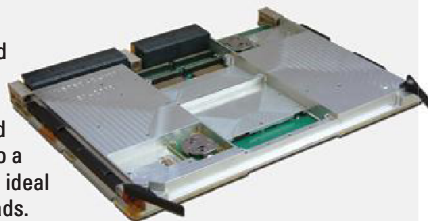*Steve Graves is cofounder and CEO of McObject, a provider of embedded Database Management System (DBMS) software. He can be contacted at steve.graves@mcobject.com.*

McObject
425-888-8505 • www.mcobject.com

Stroke of Genius

Now Shipping the NEW Cobalt Series of Virtex-6 Modules!

Unleash your inner genius with Pentek's new Cobalt™ Virtex-6 XMC modules. Built to optimize high-performance, wideband, communications, SIGINT, radar, beamforming and telemetry applications, the high-speed modules include:

- A/Ds with sampling ranges from 10 MHz to 1 GHz
- 800 MHz D/As, DUCs & Multiband DDCs
- Dedicated memory for each I/O stream
- Intelligent chaining DMA engines
- Secondary serial gigabit interface
- Multichannel, multiboard synchronization
- ReadyFlow® Board Support Libraries
- GateFlow® FPGA Design Kit & Installed IP
- VPX, XMC, PCI, PCIe, cPCI, VME/VXS, rugged XMC

With extensive resources and new enhancements in DSP engines, block RAM, logic and clocking technology, the Virtex-6 family represents the industry's most advanced FPGAs.

Call 201-818-5900 or go to www.pentek.com/go/mescobalt for your FREE *Putting FPGAs to Work in Software Radio Handbook*, technical data-sheets and price quotations.
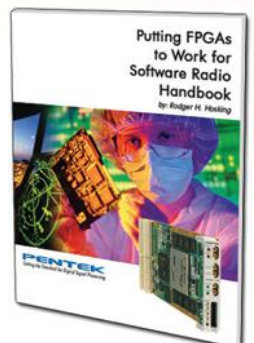
Now Available in VPX!

COTS & Rugged Versions

PENTEK
Setting the Standard for Digital Signal Processing

Pentek, Inc., One Park Way, Upper Saddle River, NJ 07458 Phone: 201.818.5900 Fax: 201.818.5904 e-mail:info@pentek.com www.pentek.com
Worldwide Distribution & Support, Copyright © 2010, Pentek, Inc. Pentek, Cobalt and GateFlow are trademarks of Pentek, Inc.
Other trademarks are properties of their respective owners.

Editor's Choice Products

Use 15 boards or one IC?

Size, Weight, and Power (SWaP) reduction is imperative on the military scene, and National Semiconductor's Analog Front-End (AFE) Integrated Circuits (ICs) are helping: Just one of these configurable AFE ICs for sensor systems is touted to handle the same load as 25 components strapped onto 15 boards. Not only that, sensor system designs that formerly took weeks to design now only take a few minutes when the AFE ICs, along with National Semiconductor's WEBENCH—Sensor AFE Designer bench-top evaluation tool, are used.

Here's how it works: WEBENCH and the AFE ICs allow engineers to choose a sensor, then design and configure their ware. Next, configuration data is downloaded and sent directly to a sensor AFE, and voila! Either the LMP91000 or LMP90100 AFE ICs can be used in this scenario. LMP90100 is a multichannel, 24-bit sensor AFE suited to high-performance transducer and transmitter applications and renders an average draw of less than 0.7 mA within the -40 °C to +125 °C temp range. In contrast, the LMP91000 is a low-power, configurable potentiostat offering a sensor-to-ADC integrated signal path, ideal for two-terminal oxygen sensors and gas and micro-power chemical sensing applications. Its 10 microAmps average voltage/output gain make it ideal for 4 mA to 20 mA transmitter applications and battery-operated systems.

National Semiconductor • www.mil-embedded.com/p47216 • www.national.com

Chip Scale Atomic Clock gives quartz oscillators the boot

Attention all quartz-based oscillators: There's a new oscillator in town, boasting the accuracy, SWaP, and reduced power consumption of atomic clock technology. Indeed, Symmetricom's QUANTUM SA.45s Chip Scale Atomic Clock (CSAC) reportedly renders two orders of magnitude better timekeeping than its quartz predecessors including Temperature Compensated Crystal Oscillators (TCXOs) and Oven-Controlled Crystal Oscillators (OCXOs): less than 0.1 seconds' error during a human lifespan. The atomic QUANTUM SA.45s CSAC weighs a miniscule 35 grams, has a volume of 16 cc, and consumes only 115 mW power, which the company says is "1/40th or less the power consumption of existing atomic oscillators."

Highly suited to underwater exploration and portable applications where continuous receipt of GPS signals is impossible, this CSAC's main mission is to provide timing stability and fast signal reacquisition. It can be tucked inside man-pack radios, UAVs, geophysical sensors, dismounted IED jammers, and GPS units themselves. Though Symmetricom's military-savvy "Option 002" variant operates at -40 °C to +85 °C, the company also vends an "Option 002" commercial-application flavor, affording an operating temp of -10 °C to + 70 °C. Interestingly, the CSAC program was originally DARPA initiated.

Symmetricom, Inc. • www.mil-embedded.com/p47217 • www.symmetricom.com

New PMC is out of this world

Space missions aren't conducted merely for entertainment. Rather, reliable information gathering and transmission are the focus, and Aitech's S750 rad-hard Layer 2 GbE switch PCI PMC paves the way for smooth data transfer in space. Specifically, the PMC enables Ethernet data to move between the host CPU and GbE NIC endpoints. It can receive and send raw data packets and IPv4 User Datagram Protocol (UDP). And with key function redundancy and onboard memory courtesy of a rad-hard FPGA, the PMC is prepared to fly into Geostationary Earth Orbit (GEO), the Mars terrestrial realm, or Low Earth Orbit (LEO). And its 10 krad (Si) Total Ionizing Dose (TID) and less than 8 W power consumption help. (Aitech can ramp up the radiation hardness levels if requested.)

Married to Aitech's 3U CompactPCI PowerPC-based S950 SBC, the S750 PMC enables the user to activate managed-switch functions including myriad power-saving choices, port management, and GbE packet transfers that are either dynamically or statically configurable. The union of PMC and SBC harnesses S950's SDRAM redundancy and onboard flash along with S750's Triple Modular Redundancy (TMR) — resulting in low latency and high levels of data integrity during transmission.

Aitech Defense Systems • www.mil-embedded.com/p47220 • www.rugged.com

Build your own system? Wait – it's already done

Rather than face the rigors of component selection, design, test, and qualification when building a system from the ground up, there's another way: GE Intelligent Platforms' CRS-C2P-3CC1 and CRS-C3P-3CB1 off-the-shelf rugged systems are application-ready, prevalidated, 2-and 3-slot CompactPCI systems tucked inside a 3U chassis. They are vended in either baseplate-cooled or convection-cooled style.

Residing in the box is a Freescale MPC7448 processor speeding along at 1.4 GHz. And don't forget the memory: 256 MB flash and 512 MB RAM. The fast-to-deployment systems — highly suitable for unmanned/manned ground vehicles, UAVS, and launch vehicles — include the VxWorks RTOS. A plethora of I/O is provided, too, including the likes of: discrete I/O, MIL-STD-1553, USB, serial, Ethernet, and ARINC 429. Rugged-deployment qualified, the CRS-C2P-3CC1 weighs 11 pounds and measures 3.96" x 7.15" x 9.03" (sans connectors), while the CRS-C3P-3CB1 tips the scale at 9 lbs and measures 5.60" x 4.25" x 8.76" (again, sans the connectors). Sounds like a good deal to us.

GE Intelligent Platforms • www.mil-embedded.com/p47218 • www.ge-ip.com

Module wins the SDR development battle

SDR development isn't a task for the faint of heart. The good news, though, is that Pentek's Model 71690 L-Band RF tuner/digital digitizer module is streamlining the L-band SDR development platform proposition. The only things Model 71690 needs are a host system (this can be something as simple as a personal computer) and an antenna. Thereafter, the XMC module can churn away, facilitating decoding, demodulation, decryption, and receipt of any digital L-band comms. This task, which sounds a lot easier now, is accomplished as the XMC processes signals straight from a low-noise block connection provided by the antenna.

Cutting-edge Virtex-6 FPGAs power the module, thus any of five variants can be used: LX365T, LX240T, LX130T, SX475T, or SX315T. Meanwhile, fast random access apps benefit from the up to 32 MB QDRII+ SRAM, and apps requiring deep memory can use the up to 2 GB DDR3 SDRAM provided. Not only that, the tuning reference for the RF subsystem can simply consist of either an optional onboard crystal oscillator or an external reference signal. Additionally, Virtex-6 FPGA software IP renders both custom and turnkey functions.

Pentek • www.mil-embedded.com/p47219 • www.pentek.com

PDA gives smartphones a run for their money

While smartphones are all the rage in the consumer arena, mobile PDAs are extremely popular on the battlefield. Handheld US is stepping into the picture with its Nautiz X3 ruggedized PDA, which measures 150 x 67 x 23 mm and tips the scale at less than 260 grams. Complying with MIL-STD-810G and rated to IP65, the PDA withstands temps of -20 °C to +60 °C and survives drops from 1.8 meters. Its compass, altimeter, built-in GPS, and G-sensor make it ideal for field ops, as does its 2.8" outdoor-viewing-optimized touchscreen. The PDA also runs from a 3300 mAh Li-Ion battery capable of rendering a full workday's worth of viewing without needing recharging.

But what's inside counts, too. Powered by an X-Scale processor at 806 MHz, Nautiz X3 comes with 512 MB flash memory and 256 MB RAM. And the PDA offers several handy features: a 3-megapixel camera with LED flash/autofocus, Bluetooth, WLAN, and 3G cellular. For imagery, a 1D laser scanner is provided as standard, and a 2D imager option is also available. These features are enabled via the Windows Mobile 6.5 Professional operating system. All in all, the PDA is designed to go "beyond a smartphone," the company says.

Handheld US • www.mil-embedded.com/p47222 • www.handheldgroup.com

Legacy x86 code reaches modern standards

As new certification standards arise in the avionics, industrial control, and other industries, legacy x86 applications are required to recertify. Thus, LDRA's x86 assembler flavor of its LDRA tool suite "enables certification for legacy applications that would not otherwise be certifiable," the company says. Compatible with any x86 assembler variant, the LDRA implementation proffers assembly coverage such as bitmap coverage, gain assembly static analysis, and report/artifact creation. These capabilities are needed because legacy apps often revolve around handcoded assembly, lack the entire rendition of high-level code, or might have BIOS that is board-specific and uncertifiable.

But that's not all. Other situations where the LDRA x86 syntax-supporting tool suite might come in handy include: 1) MC/DC scenarios, where high-level language coverage analysis is problematic; 2) compiled programs, with the possibility of instrumenting and disassembling; and 3) handcoded assembly programs that feature Board Suport Packages (BSPs). But there is hope: The LDRA implementation is able to nix the prospects of (and therefore, the expenses of) new-code writing, testing, and verification. The LDRA tool suite includes the LDRA Testbed, TBrun, TBreq, TBvision, Embed-X, TBsecure, TBevolve, TBsafe, and DO-178B Tool Qual Pack products.

**LDRA • www.mil-embedded.com/p47223 • www.ldra.com**

JTRS-developed PMC speeds implementation

Having been tried, tested, and proven to cut JTRS SDR waveform development time, 4DSP Inc.'s AD350 PMC expansion card is vended publicly these days. It is a good fit for MIMO testbed systems and wireless base stations, and it offers prototyping for multi-output and multi-input antenna transmitter system implementation. AD350 can also regulate amplitude voltages in synchrotron systems. So we're wondering: Does the PMC's effectiveness come from the pair of Altera Cyclone III FPGAs it houses, or from its dual-channel 14-bit A/D input? Or maybe it's the 150 MSps maximum sampling rate via its 16-bit, dual-channel, 500 Msps D/A output converter? Who knows. But this thing has so darned much capability that it's certainly at the head of its proverbial class.

Here's more: The PMC's channels are connected to the front panel, while *input* analog signals utilize efficient AC coupling and wideband transformers at 4.5 MHz to 650 MHz; that serves to lessen phase noise and amplitude distortions. Meanwhile, *output* analog channels are similarly coupled, but in the 4.5 MHz to 3 GHz range. A front-panel, single-ended SMA connection facilitates an external trigger. (The SMA connection is simpatico with TTL GPS receivers producing 1pps.) Additionally, users can take advantage of the PMC's flexibility by choosing an onboard or external clock, with the goal of cutting D/A and A/D SNR phase noise degradation and jitter.

4DSP Inc. • www.mil-embedded.com/p43452 • www.4dsp.com

Crosshairs Editorial
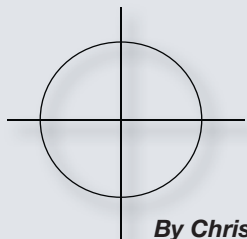
"For want of a nail, the shoe was lost." Forget about shoes; warfighters need portable power

By Chris A. Ciufo, Editor

I recently attended AFCEA's 20th annual West 2011 ("West") convention held in San Diego. The Marine Corps also held their annual Marine West at Camp Pendleton just north of San Diego during the West event. Where West had men in dark suits wearing badges that said "Lockheed Martin" or "Raytheon," Marine West had Marines and two-star generals confidently strolling in combat boots and BDUs. West was held in the air-conditioned convention center downtown, and Marine West was in a mammoth plastic tent outside in 80-degree weather. The attitudes of the attendees differed, too. Biz Dev execs schmoozed PMs and PEOs in San Diego, while buzz-topped former grunts and COs at Camp Pendleton swapped war stories with barely boot-camp graduates and battle-hardened Marines.

Yet despite the overwhelming differences between these venues, I noted lockstep similarities in the attitudes and the displayed equipment addressing warfighters' needs. For instance, even though West (co-sponsored by the U.S. Naval Institute and the proximate Naval Base San Diego) had a decidedly naval focus, the emphasis remained on systems that were rapidly deployable, effective, and portable.

Portability brings unique challenges. Assuming cost is no object – but it always is – it's SWaP that drives the design. At both events, row after row showed COTS platforms that were easily mounted on MRAPs or HMMWVs, man-packed, or tucked inside robot drones. On display at Marine West in the QinetiQ booth was an epaulette-mounted (shoulder) device about the size of a sandwich. Portable enough, for sure. A coiled cord connected to a wrist-worn controller with a small LCD display to complete the Ears/SWATS Shoulder Worn Acoustic Targeting System. Relying on a 10-12 hour battery, the system uses microphones to pinpoint gunshots fired at warfighters.

Funded out of PEO Soldier's SWATS program and part of the Army's Individual Gunfire Detection System – and I suspect inspired by several DARPA gunshot location systems dating back to the 1990s – the 1 W, 1-pound system forms a 25 m "bubble" around a soldier to provide relative bearing and absolute position of firing combatants. Compass, GPS, DSP, and sensors all work in lockstep to provide visual or audible cues before the next round finds its mark with deadly results. While highly effective and exceeding the ORD requirement, it seems to me that the 12-hour (max) life of the battery is too short for the typical mission and forces warfighters to carry extra batteries. Portability suffers as the need for batteries grows.

Batteries add weight that could be used for ammo, MREs, or water. QinetiQ is already working on Gen 2, which will be smaller but battery life is similar. If money was no object, an SoC or ASIC a la a cell phone design would save the day. At "nearly $10,000 each," the Army's 13,000 order is hardly high volume and certainly not in the realm of custom cell phone ICs. Interestingly, Raytheon BBN Technologies has the Boomerang Warrior-X, a similar portable soldier-worn system with excellent under 7.5 degree bearing error and +20 percent range error, but still a 12-hour battery life. Vehicle-mounted versions of both systems exist to protect convoys, with virtually unlimited on-board power. That makes them sort-of portable.

But once away from a vehicle and dismounted, portable power is nearly as important as a weapon, because the entire mission often resides on transmitting images, video, and sensor or voice data *back* to base camp. Whether on recon or lasing a target, the portable equipment still needs juice. About 10 battery and power converter companies were present showcasing rugged designs, most of which relied on the COTS Li-Ion or civilian Eveready-style batteries. And even though

the U.S. government is pouring money in electric vehicle battery research, batteries have changed little in the past five years and retain existing pros and cons. The exhibitors at House of Batteries, for instance, reminded me that the FAA won't transport Li-Ion on commercial flights anymore due to the risk of fire. And fuel cell batteries – a cell phone "hopeful" a few years ago – may off-gas flammable material that's bad news on a battlefield.

But there's hope for portable power. The batteries are often good enough, while the system power requirements keep dropping. Besides multicore, deep-sleep CPUs from ARM, Intel, and others, and mundane things like high-efficiency DC-DC converters, there's also LED lighting contributing to saving portable energy. On the latter, the more output lumens per consumed watt, the more energy available for the rest of the system. LED-backlit LCDs, panel lights, and local spot lamps save energy and extend battery life. And during the day, desert geographies like the Middle East can use portable solar panels like those from Iris Technology Corporation's SPACES system.

The plug-and-go simplicity of this system, along with solar cell portability, caught my attention. The nearly $11 million in USMC contracts for the Solar Portable Alternative Communications Energy System bought 1,650 modules that accept, among other sources, vehicle power, BA-XX90 and zinc-air batteries, and rollable solar panels. Two 12-3 VDC outputs are available to charge BA-XX90 batteries[1], power radios, and even USB dongles for GPS, cell phone, and laptop.

Every little bit helps. From solar cells to lower-power electronics, saving power and extending battery life in portable systems are crucial to the success of many missions. Judging by the number of COTS vendors focusing on portable power, no shoes will be lost for lack of a nail. Or battery.

---

[1] Clearly you wouldn't use a battery to charge the *equivalent* battery – you'd just swap it out. Instead, SPACES can convert energy from one battery type to another at 96 percent efficiency when field ops demand flexibility with on-hand equipment.

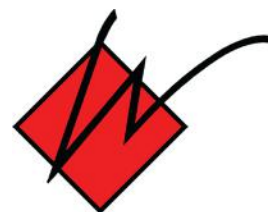# WILDSTAR 4 / 5 / 6
Family Architecture

Analog Product Guide

Above and Beyond

Commercial and Government Market Leader in High Speed Digital Signal Processing on FPGA Based Reconfigurable Computing Boards

Celebrating
Our 27th
Anniversary
1982 - 2009

Annapolis Micro Systems, Inc.

Annapolis Product Line

- Choose from a Full Line of Totally Integrated Hardware, Software and Tools
  - Compatible with all WILDSTAR^{TM} 4/5/6 Main Boards
  (VME64x, VXS, VPX, PCI-X, PCI-Express, and IBM BladeCenter)
- Build the highest performance applications in the smallest amount of time
- VHDL Models, Examples and Full CoreFire^{TM} Support Available
- Use most Standard Operating Systems - Windows, Linux, VxWorks, etc.
- Growing Number of I/O Standard Options, including multiple A/D and D/A boards,10G Infiniband, 10G Ethernet, 40/100G Ethernet, Serial FPDP, FibreChannel 2 and 4, etc.

Analog Mezzanine I/O Cards

- Use the industry's best performing ADCs and DACs
  - Best ENOB, SFDR, IMD Performance and Analog Input Bandwidth
  - Ultra Low Skew and Jitter Clock Distribution
  - Universal 50 Ohm Low Jitter Clock Input
  - High Precision Trigger Input with Fs Precision and Configurable Voltage Levels
- Synchronize ADCs and DACs on Multiple Mezzanine Cards to the Same Sample Clock
- Use WILDSTAR^{TM} Clock/Trigger Distribution Cards for additional Synchronization
- Shields prevent noise from coupling between channels and into analog front end
- Converter Temperature Sensing on many cards

Commercial Off the Shelf (COTS) Hardware/Software

- Save Time and Money - Use COTS Technology
- State of the Art Expert Hardware Design
- Robust, Fully Tested, High Performance Products
- Ensure Reliability

Benefits of Using CoreFire^{TM} FPGA Application Builder

- Achieve High Performance
- Save Processing Time and Chip Space
- Ensure Reliability
- Thousands of Parameterized Modules
  - Complex and Real FFTs, FIR Filters, Comb Filters
  - Digital Phase Locked Loop
  - Linear Feedback Shift Register
  - Includes Floating Point and Array Types
- Save Months of Development Time
- Dramatically Reduce Time-to-Market
- Nearly Seamless Technology Refresh Across FPGA and WILDSTAR Families
- Realize Repeatable Results
- Gain Company-Wide Standardization

The Annapolis Tradition

The highly experienced team of engineers at Annapolis Micro Systems, Inc. provides exceptional
Analog to Digital and Digital to Analog Products
with very low noise floors, very low spur levels and very flat bandwidth.

The Leaders in providing user friendly FPGA performance, we are famous internationally for the high
quality of our products and for our unparalleled dedication to ensuring
that the customers' applications succeed.

We offer training and incomparable special application development support,
as well as more conventional customer support.

WILDSTAR™ 4 / 5 / 6 Analog Conversion Products

Analog to Digital I/O Mezzanine Cards							
I/O Mezzanine Card	Sample Freq (MSps)	# of Chan	Resolution in Bits	Input Bandwidth (3dB Cutoff)	Onboard Oscillator	Chip Part Number	Mfg
Quad 130 MSps	130	4	16	550 MHz	Yes	LTC2208	Linear
Quad 160 MSps	160	4	16	550 MHz	Yes	LTC2209	Linear
Quad 170 MSps	170	4	16	730 MHz	Yes	ADS5484	TI
Quad 180 MSps	180	4	16	550 MHz	Yes	LTC2209	Linear
Quad 200 MSps	200	4	16	730 MHz	Yes	ADS5485	TI
Quad 250 MSps	250	4	13	850 MHz	No	ADS5444	TI
Quad 400 MSps	400	4	14	1.1 GHz	No	ADS5474	TI
Quad 500 MSps	500	4	12	1.8 GHz	No	ADS5463	TI
Quad 550 MSps	550	4	12	1.8 GHz	No	ADS54RF63	TI
Dual 1.0 GSps	1000	2	12	2.1 GHz	Yes		
1.5 GSps	1500	1	8	2.6 GHz	Yes	MAX108	Maxim
1.5 GSps	1500	1	10	2.5 GHz	No	AT84AS003	e2v
2.0 GSps	2000	1	10	2.5 GHz	No	AT84AS004	e2v
2.2 GSps	2200	1	8	2.5 GHz	Yes	MAX109	Maxim
Single 5.0 GSps or Dual 2.5 GSps	5000/ 2500	1 / 2	8	2.5 GHz	Yes	EV8AQ160	e2v
Digital to Analog I/O Mezzanine Cards							
I/O Mezzanine Card	Sample Freq (MSps)	# of Chan	Resolution in Bits	Output Bandwidth (3dB Cutoff)	Onboard Oscillator	Chip Part Number	Mfg
Quad 600 MSps	600	4	16	1.0 GHz	Yes	MAX5891	Maxim
Dual 1.5 GSps	1500	2	12	2.0 GHz	Yes	MAX5859	Maxim
Dual 2.3 GSps	2300	2	12	2.0 GHz	Yes	MAX19692	Maxim
Dual 4.0 GSps	4000	2	12	2.0 GHz	Yes	MAX19693	Maxim

Quad 130 / 160 / 170 / 180 / 200 MSps 16 Bit A/D Converter

Clock and Input Specifications

- External Clock Frequency   32 - ADC_{max} MHz
- Onboard Oscillator Jitter   0.5ps, 1ps RMS Max
  (12KHz – 80MHz)

For 130/160/180 ADCs

- 3dB Frequency Response   2 MHz – 550 MHz
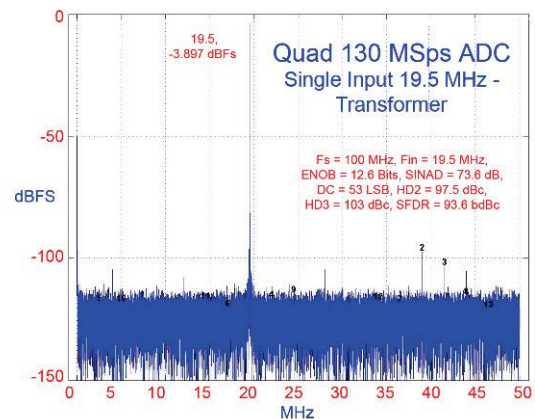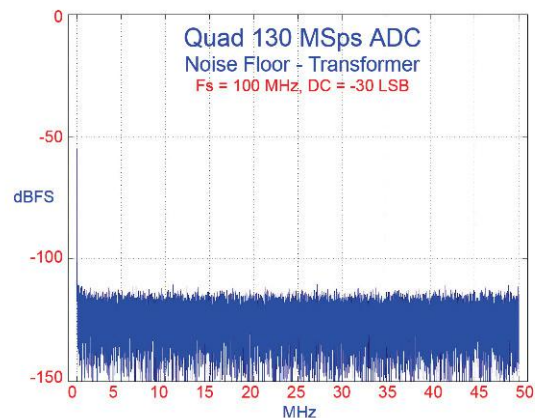- Full-Scale Input Amplitude   +11.8 dBm   +/- 0.25dBm

For 170/200 ADCs

- 3dB Frequency Response   2 MHz – 730 MHz
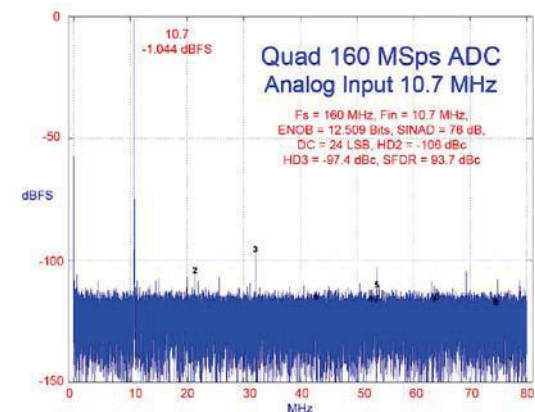
Quad 130 Performance

Transformer input, Fs = 100 MHz, Fin = 19.5 MHz

- Effective Number of Bits   12.6 bits
- Spur Free Dynamic Range   92 dB
- Crosstalk Isolation   > 115 dB

Quad 160 Performance

Transformer input, Fs = 160 MHz, Fin = 10.7 MHz

- Effective Number of Bits   12.5 bits
- Spur Free Dynamic Range   94 dB

Transformer input, Fs = 160 MHz, Fin = 61.44 MHz

- Effective Number of Bits   11.7 bits
- Spur Free Dynamic Range   81 dB

Transformer input, Fs = 160 MHz, Fin = 91 MHz

- Effective Number of Bits   11.6 bits
- Spur Free Dynamic Range   79 dB

Quad 200 Preliminary Performance

Transformer input, Fs = 200 MHz, Fin = 70 MHz

- Effective Number of Bits   11.95 bits
- Spur Free Dynamic Range   87 dB

Quad 130 MSps ADC Noise Floor - Transformer
Fs = 100 MHz, DC = -30 LSB

Quad 130 MSps ADC Single Input 19.5 MHz - Transformer
Fs = 100 MHz, Fin = 19.5 MHz, ENOB = 12.6 Bits, SINAD = 73.6 dB, DC = 53 LSB, HD2 = 97.5 dBc, HD3 = 103 dBc, SFDR = 93.6 bdBc

Quad 160 MSps ADC Analog Input 10.7 MHz
Fs = 160 MHz, Fin = 10.7 MHz, ENOB = 12.509 Bits, SINAD = 76 dB, DC = 24 LSB, HD2 = -106 dBc, HD3 = -97.4 dBc, SFDR = 93.7 dBc

Quad 130 / 160 / 170 / 180 / 200 MSps 16 Bit A/D Converter

Features

- Four Analog to Digital Converters of one of the types:
  - LTC2208   130 MSps   16 bits
  - LTC2209   160 MSps   16 bits
  - ADS5484   170 MSps   16 bits
  - LTC2209   180 MSps   16 bits
  - ADS5485   200 MSps   16 bits
- Six SMA Front Panel Connectors
  - Four 50-Ohm Analog Inputs
  - High Precision Trigger Input with Fs Precision
  - Universal Single Ended 50 ohm Clock Input
- Ultra Low Skew and Jitter Clock Distribution
- Independent Software Control for each ADC
  - Programmable Gain Analog Inputs
  - Control ADC Dither to Increase SFDR
- Main Board PCLK Sourcing Capability
- Manufacturing Options:
  - Input Type 130/160/180: Op-Amp (DC) or Transformer (AC)
  - Input Type 170/200: Op-Amp Narrow, OP-Amp Standard, Transformer 1:1, Transformer 4:1
  - Internal Clock (Onboard Oscillator)
  - Connector to stack additional I/O mezzanine cards
- High Precision Trigger Input: 1.65V LVPECL, 2.5V LVPECL, 3.3V LVPECL
- Use External Trigger to Synchronize Multiple ADCs

Quad 160 MSps 16 Bits - Input Bandwidth
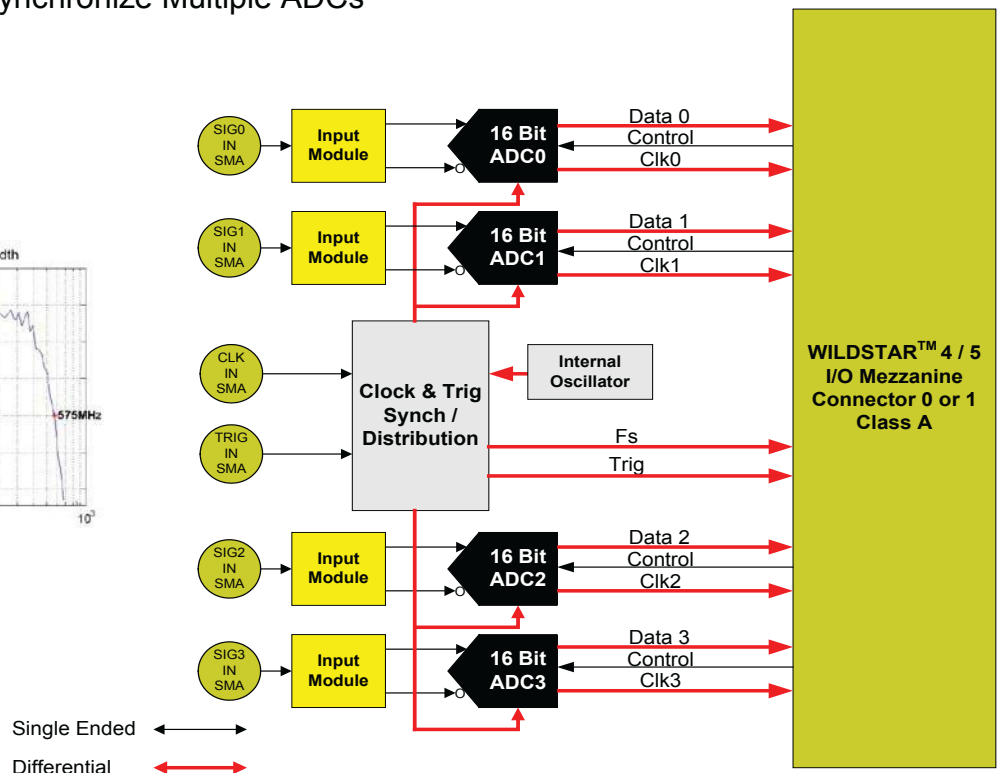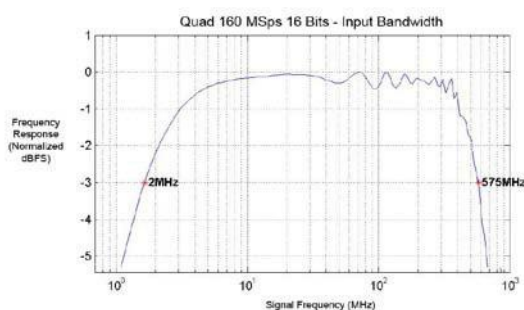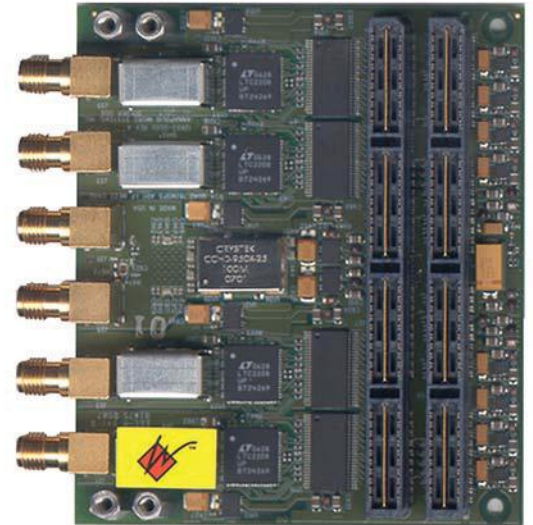
Quad 250 / 400 / 500 / 550 MSps A/D Converter

Clock Specifications

- External Clock Frequency   64 - ADC$_{max}$ MHz

Quad 250 MSps 13 Bits

Balun input
- Crosstalk Isolation   > 110 dB
- 3dB Frequency Response   0.400 – 850 MHz
- Full-Scale Input Amplitude   +9.4 dBm +/- 0.25dBm
  (at 10.7MHz)

Balun input, Fs = 250 MHz, Fin = 91 MHz
- Effective Number of Bits   11.0 bits
- Spur Free Dynamic Range   76 dB
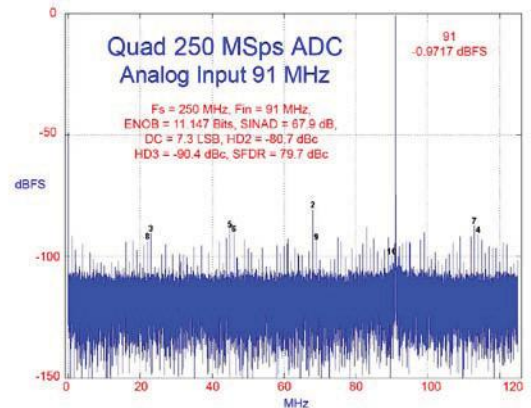
Quad 400 MSps 14 Bits

Balun input
- Crosstalk Isolation   > 110 dB
- 3dB Frequency Response   0.275 – 1150 MHz
- Full-Scale Input Amplitude   +8.9 dBm +/- 0.25dBm
  (at 10.7MHz)

Balun input, Fs = 400 MHz, Fin = 10.7 MHz
- Effective Number of Bits   11.35 bits
- Spur Free Dynamic Range   88 dB

Balun input, Fs = 400 MHz, Fin = 170 MHz
- Effective Number of Bits   11.2 bits
- Spur Free Dynamic Range   80 dB
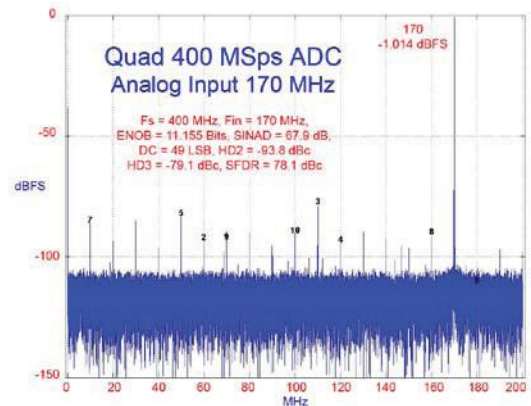
Quad 500 MSps 12 Bits

Balun input
- Crosstalk Isolation   > 100 dB
- 3dB Frequency Response   0.250 – 1850 MHz
- Full-Scale Input Amplitude   +9.4 dBm +/- 0.25dBm
  (at 10.7MHz)

Balun input, Fs = 500 MHz, Fin = 10.7 MHz
- Effective Number of Bits   10.55 bits
- Spur Free Dynamic Range   85 dB

Balun input, Fs = 500 MHz, Fin = 170 MHz
- Effective Number of Bits   10.45 bits
- Spur Free Dynamic Range   75 dB
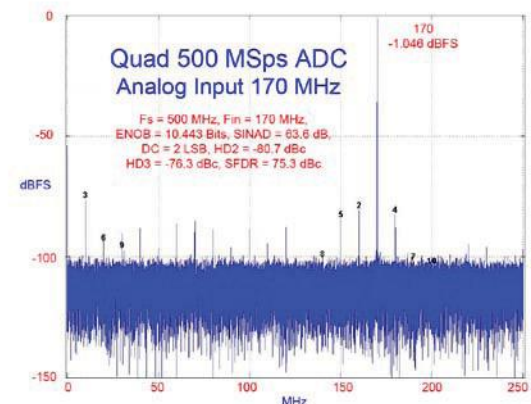
Quad 250 / 400 / 500 / 550 MSps A/D Converter

Features

- Four Analog to Digital Converters of one of the types:
  - ADS5444   250 MSps   13 bits
  - ADS5474   400 MSps   14 bits
  - ADS5463   500 MSps   12 bits
  - ADS54RF63   550 MSps   12 bits
- Six SMA Front Panel Connectors
  - Four 50-Ohm Analog Inputs
  - High Precision Trigger Input with Fs Precision
  - Universal Single Ended 50 ohm Clock Input
- Independent Software Control for each ADC
  - Programmable Gain Analog Inputs
  - Control ADC Dither to Increase SFDR
- Ultra Low Skew and Jitter Clock Distribution
- Main Board PCLK Sourcing Capability
- Manufacturing Options:
  - Input Type: Op-Amp for DC Content or Balun
- High Precision Trigger Input: 1.65V LVPECL, 2.5V LVPECL, 3.3V LVPECL
- Integrated Thermal Solution
- Use External Trigger to Synchronize Multiple ADCs

Quad 250 MSps 13 Bits - Input Bandwidth

Quad 400 MSps 14 Bits - Input Bandwidth
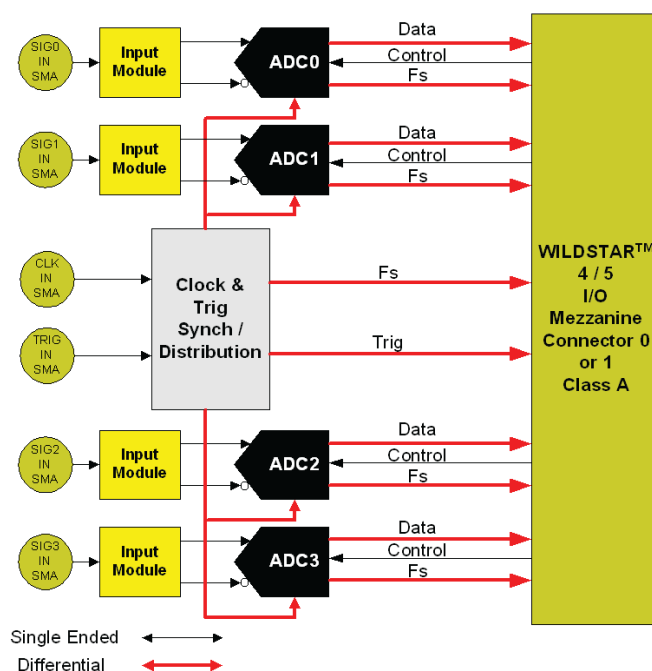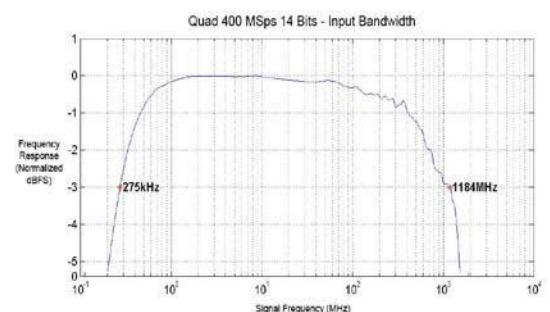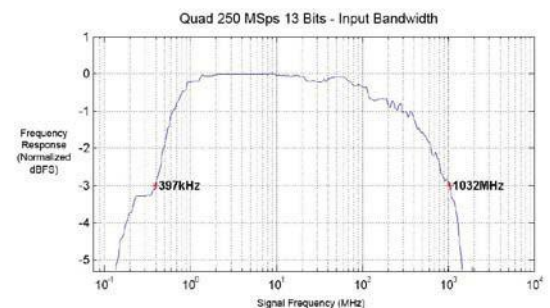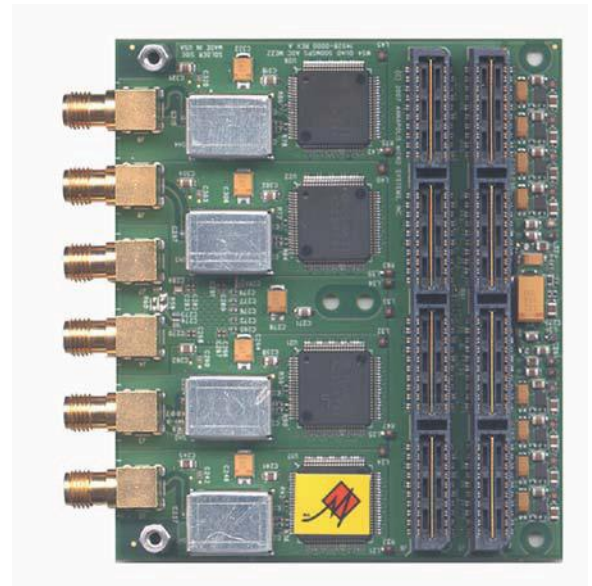
1.5 GSps 8 Bit A/D Converter

Performance

Balun input
- 3dB Frequency Response   64KHz - 2600 MHz
- Full-Scale Input Amplitude   -2.2 dBm +/- 0.5dBm

Op-amp input
- 3dB Frequency Response   DC - 2600 MHz

Balun input, Fs = 1500 MHz, Fin = 273 MHz
- Effective Number of Bits   7.6 bits
- SINAD   46 dB
- Spur Free Dynamic Range   63 dB
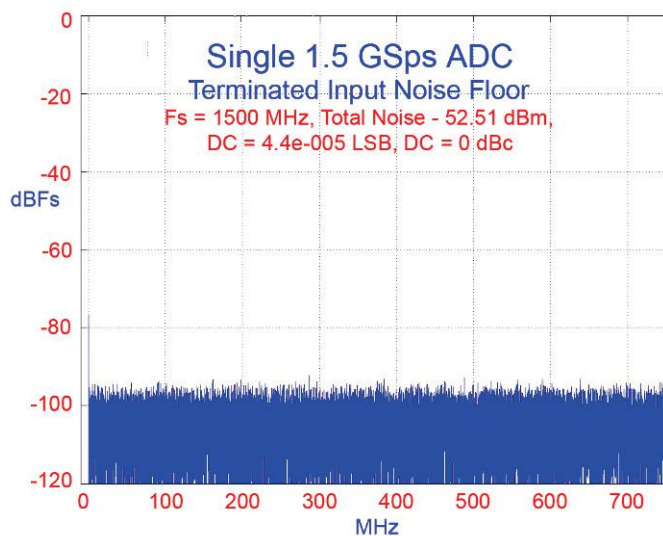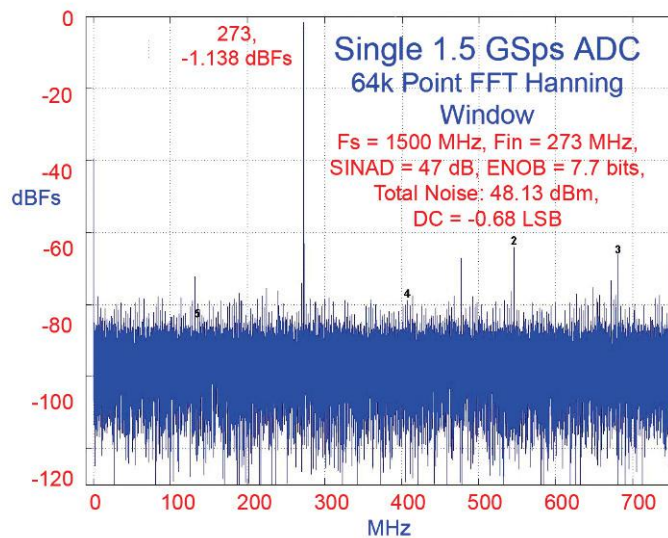
1.5 GSps 8 Bit A/D Converter

Features

- Single MAX108 8 bit Analog to Digital Converter
- Four SMA Front Panel Connectors
  - One Analog Input
  - High Precision Trigger Input with Fs Precision
  - Single Ended 50 ohm Clock Input or Differential 1.65V LVPECL Clock Source
- Optional internal oscillator
- Analog Input: DC or AC Coupled
- High Precision Trigger Input: 1.65V LVPECL, 2.5V LVPECL, 3.3V LVPECL
- Integrated Thermal Solution
- Adjustable Gain, Phase and DC Offset on ADC
- Independent Software Control
  - Programmable Gain Analog Inputs
  - Control ADC Dither to Increase SFDR
  - Variable DC Offset
- Use External Trigger to Synchronize Multiple ADC Cards
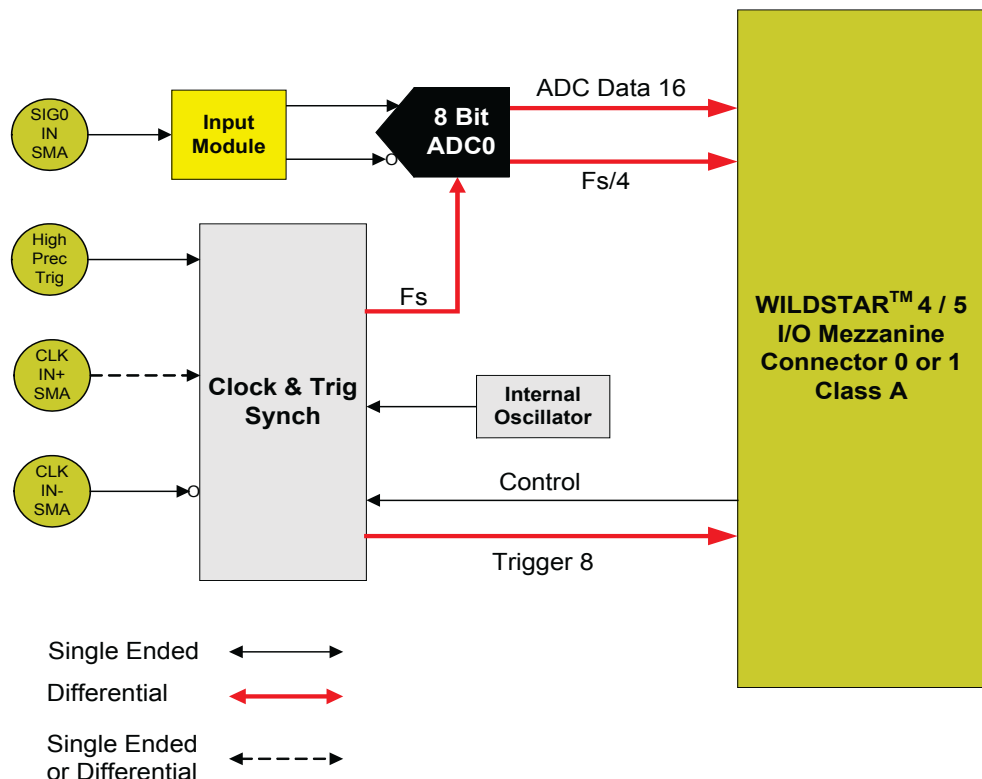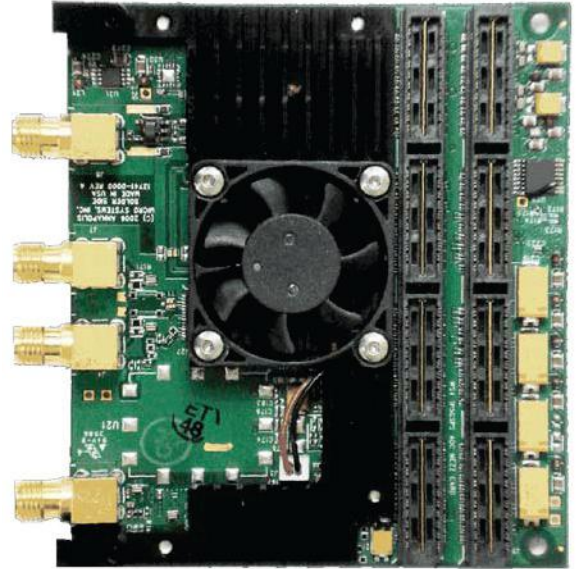
1.5 / 2.0 GSps 10 Bit A/D Converter

Input and Clock Specifications

Balun input

- 3dB Frequency Response   57 KHz – 2500 MHz
- Full-Scale Input Amplitude   -4 dBm +/- 0.25dBm (at 10.7MHz)
- External Clock Frequency   256 - ADC_{max} MHz

1.5 GSps 10-bit

Balun input, Fs = 1500 MHz, Fin = 10 MHz

- Effective Number of Bits   7.7 bits
- Spur Free Dynamic Range   52 dBc

Balun input, Fs = 1500 MHz, Fin = 748 MHz

- Effective Number of Bits   7.6 bits
- Spur Free Dynamic Range   52.5 dBc

2.0 GSps 10-bit

Balun input, Fs = 2000 MHz, Fin = 10.7 MHz

- Effective Number of Bits   7.8 bits
- Spur Free Dynamic Range   55 dBc

Balun input, Fs = 2000 MHz, Fin = 748 MHz

- Effective Number of Bits   7.7 bits
- Spur Free Dynamic Range   55 dBc

1.5 GSps ADC Analog Input 10.0 MHz
Fs = 1500 MHz, Fin = 10 MHz, ENOB = 7.7695 Bits, SINAD = 47.5 dB, DC = 0.75 LSB, HD2 = -54.4 dBc HD3 = -58.1 dBc, SFDR = 53.4 dBc

2.0 GSps ADC Analog Input 10.7 MHz
Fs = 2000 MHz, Fin = 10.71 MHz, ENOB = 7.8975 Bits, SINAD = 48.3 dB, DC = 7.1 LSB, HD2 = -65.9 dBc HD3 = -67 dBc, SFDR = 57 dBc

2.0 GSps 10 Bits - Input Bandwidth
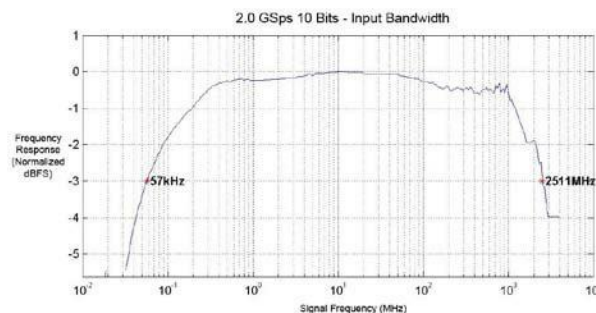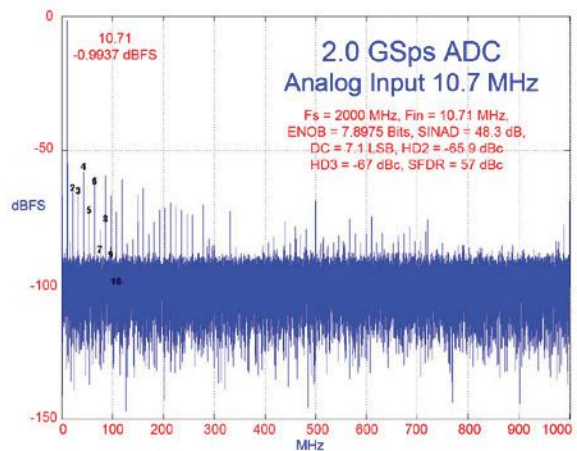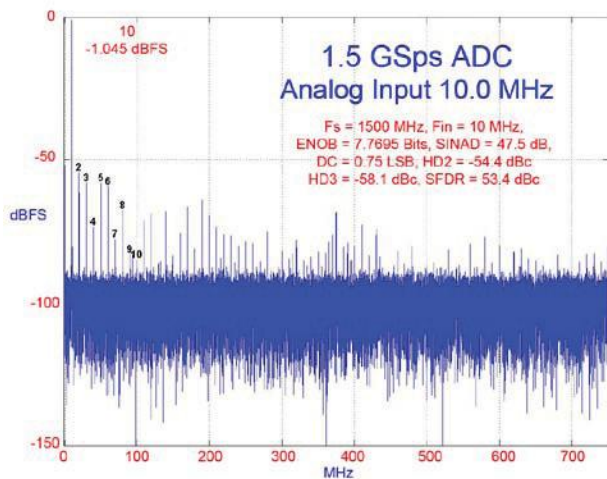
1.5 / 2.0 GSps 10 Bit A/D Converter

Features

- One Analog Convertor
  - AT84AS003   1.5 GSps   10 bits
  - AT84AS004   2.0 GSps   10 bits
- Four SMA Front Panel Connectors
  - One Analog Input
  - High Precision Trigger Input with Fs Precision
  - Single Ended 50 ohm Clock Input or Differential 1.65V LVPECL Clock Source
- Analog Input: DC or AC Coupled
- High Precision Trigger Input: 1.65V LVPECL, 2.5V LVPECL, 3.3V LVPECL
- Integrated Thermal Solution
- Software Controlled Gain and Phase on ADC Input
- Use External Trigger to Synchronize Multiple ADC Cards
- Source Main Board PCLK with Divided Version of Sample Clock
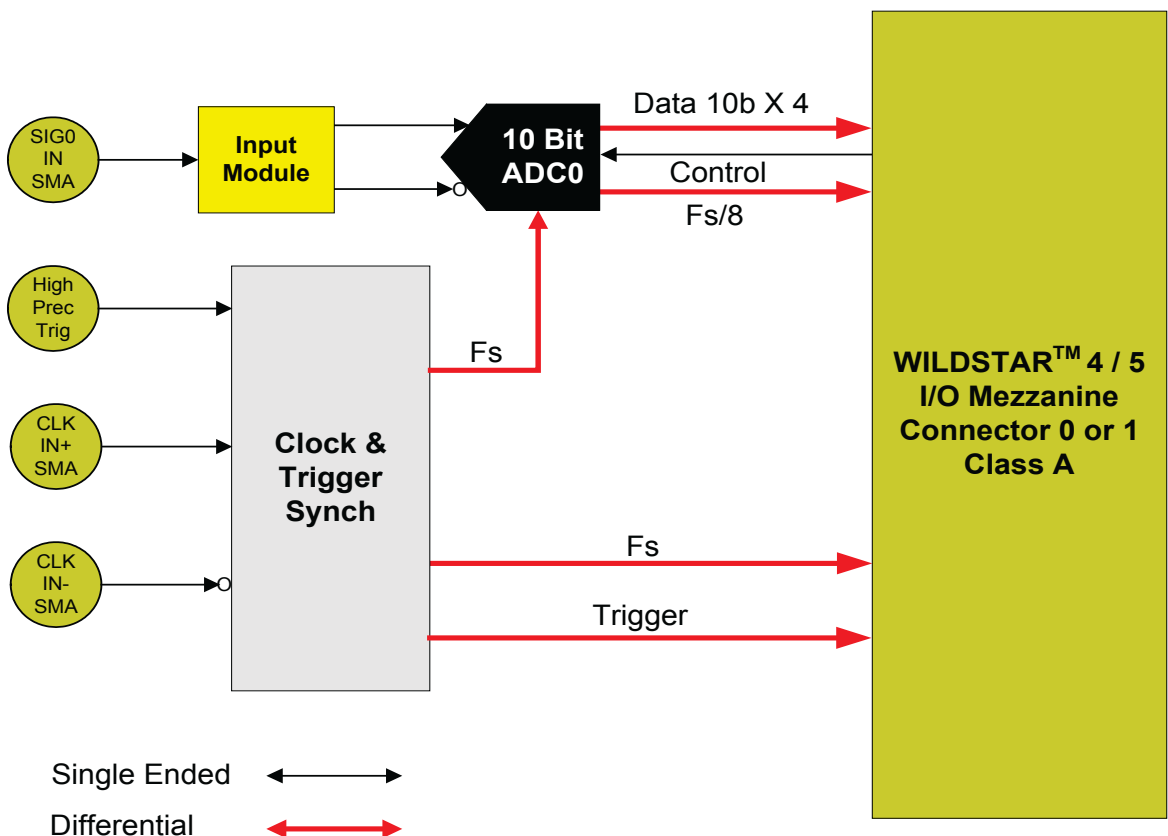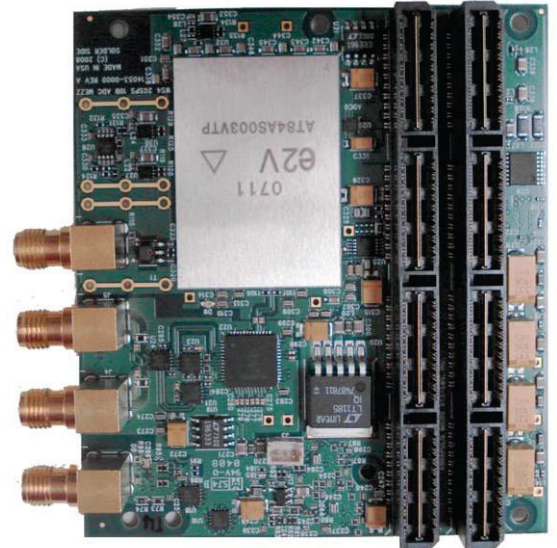
2.2 GSps 8 Bit A/D Converter

Performance

Fs = 2200 MHz, Fin = 10.7 MHz

- Effective Number of Bits   7.3 bits
- Spur Free Dynamic Range   64 dB
- 3dB Frequency Response   100 KHz - 2500 MHz
- External Clock Frequency   256 MHz - 2500 MHz
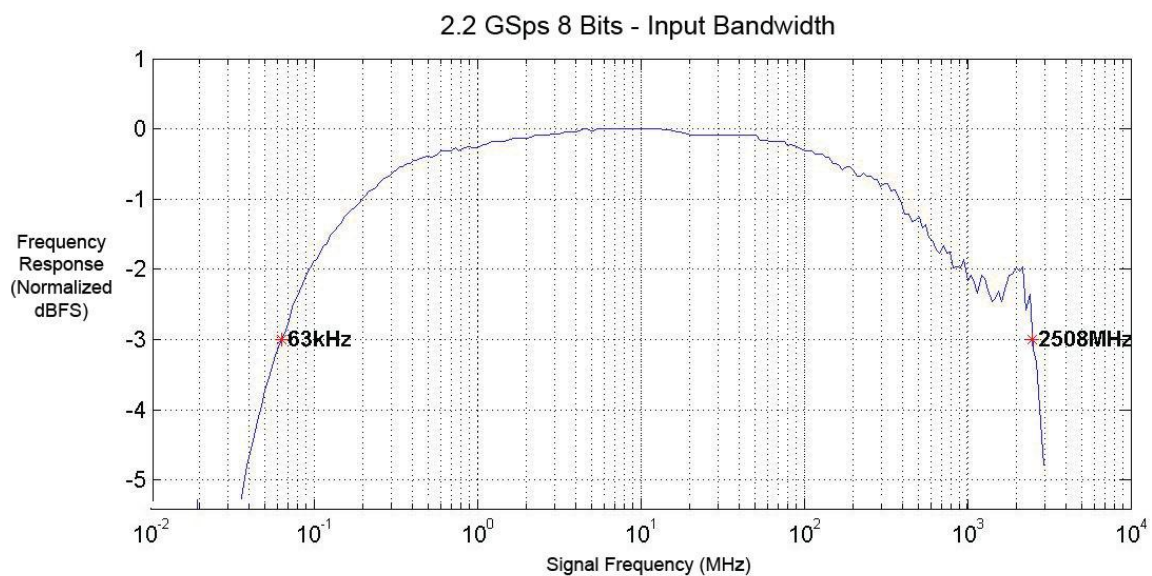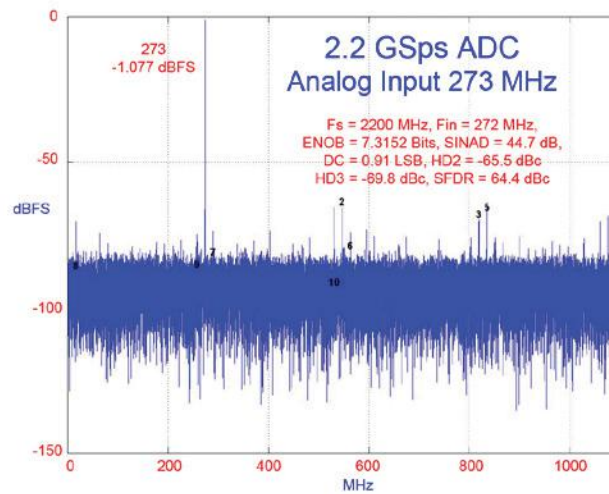
2.2 GSps 8 Bit A/D Converter

Features

- Single MAX109 8 bit Analog to Digital Converter
- Four SMA Front Panel Connectors
  - One Analog Input
  - High Precision Trigger Input with Fs Precision
  - Single Ended 50 ohm Clock Input or Differential 1.65V LVPECL Clock Source
- Optional PLL based Internal Oscillator
- Analog Input: DC or AC Coupled
- High Precision Trigger Input - 1.65V LVPECL, 2.5V LVPECL, 3.3V LVPECL
- Integrated Thermal Solution
- Software Controlled Gain, Phase and DC Offset on ADC
- Ultra Low Jitter PLL Based Internal Clock
- Use External Trigger to Synchronize Multiple ADC Cards
- Source Main Board PCLK with Divided Version of Sample Clock
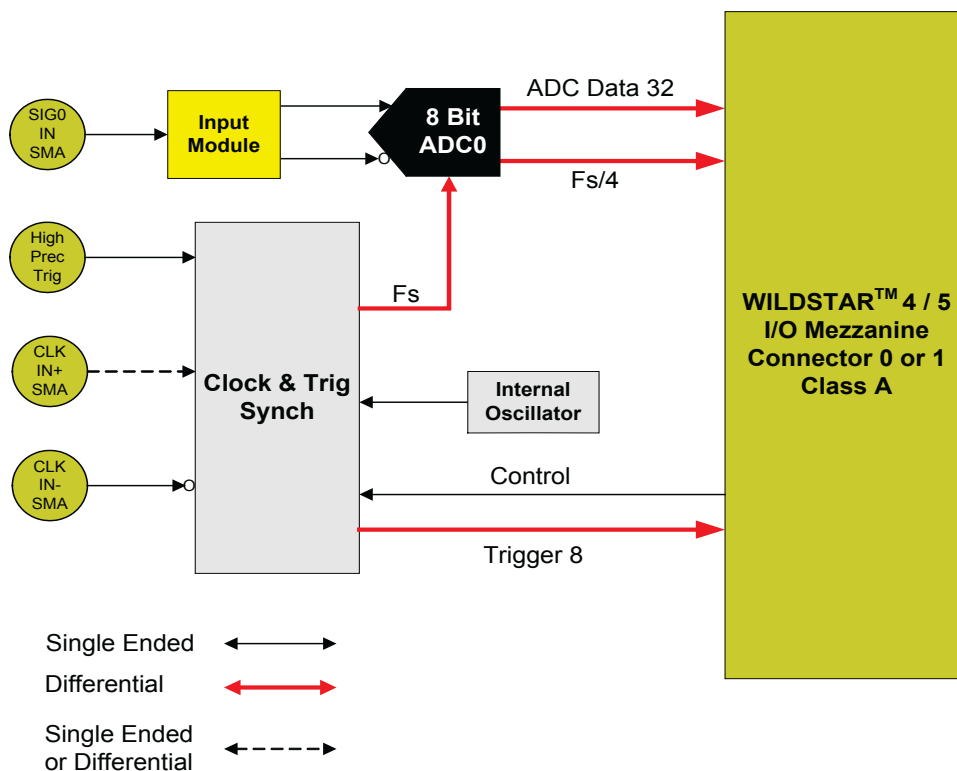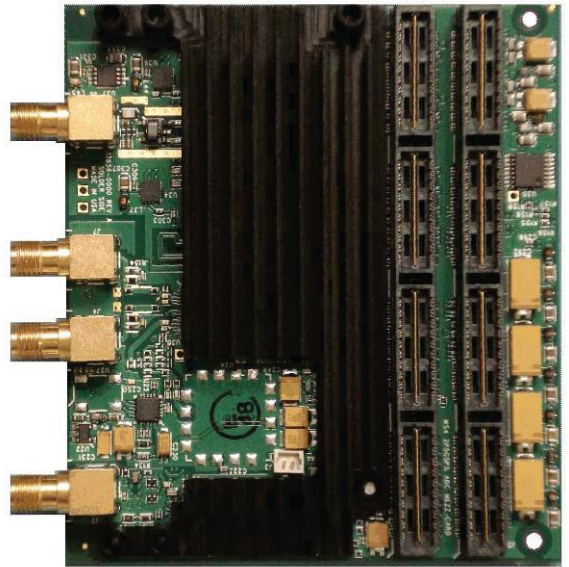
Single and/or Dual 2.5 / 5.0 GSps 8 Bit A/D Converter

Preliminary Performance

5.0 GSps Operation

Fs = 5000 MHz, Fin = 100 MHz
- Effective Number of Bits   7.1 bits
- Spur Free Dynamic Range   49 dBc

2.5 GSps Operation

Fs = 2500 MHz, Fin = 100 MHz
- Effective Number of Bits   7.3 bits
- Spur Free Dynamic Range   51 dBc
- 3dB Frequency Response - Software Controlled
  - 600 MHz
  - 800 MHz
 - 1.5 GHz
  - 2.5 GHz

Single and/or Dual 2.5 / 5.0 GSps 8 Bit A/D Converter

Features

- Single EV8AQ160 8 bit Analog to Digital Converter
- Runtime selectable input channels:
  - One channel at 5.0 Gsps
  - Two channels at 2.5 Gsps
- Five SMA Front Panel Connectors
  - Two Analog Inputs
  - High Precision Trigger Input with Fs Precision
  - Single Ended 50 ohm Clock Input or Differential 1.65V LVPECL Clock Source
- Optional PLL based Internal Oscillator
- Analog Input: DC or AC Coupled
- High Precision Trigger Input - 1.65V LVPECL, 2.5V LVPECL, 3.3V LVPECL
- Integrated Thermal Solution
- Adjustable Gain, Phase and DC Offset on ADC
- Ultra Low Jitter VCO/PLL Based Internal Clock
- Software Controlled Input Bandwidth: 600 MHz, 800 MHz, 1.5 GHz or 2.5 GHz
- Source Main Board PCLK with Divided Version of Sample Clock
- Use External Trigger to Synchronize Multiple ADC Cards
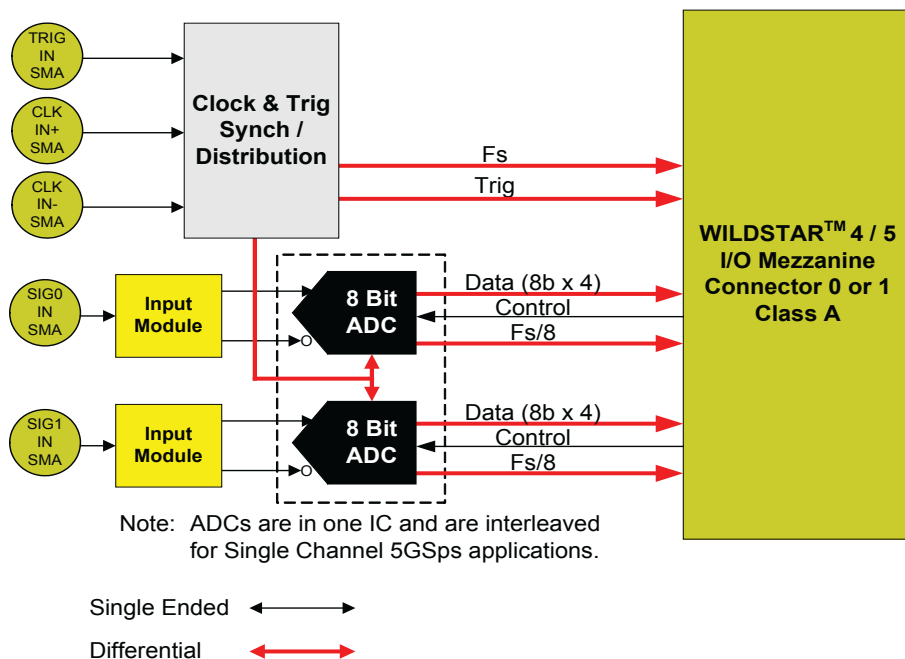
Quad 600 MSps 16 Bit D/A Converter

Performance

Channel 3 600MHZ
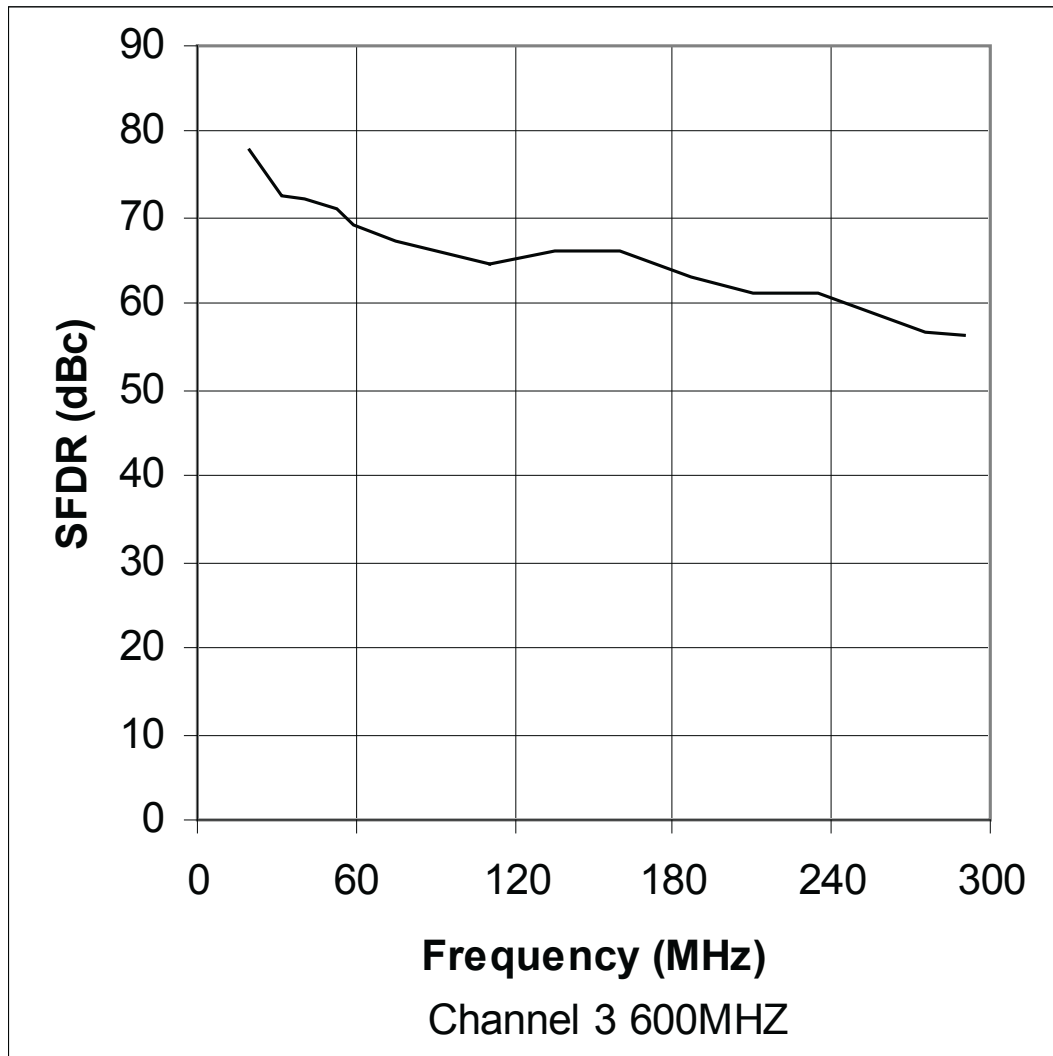
Quad 600 MSps 16 Bit D/A Converter

Features

- Four MAX5891 16 bit Digital to Analog Converters
- Six SMA Front Panel Connectors
  - Four Single Ended 50 ohm DAC Outputs
  - High Precision Trigger Input with Fs Precision
  - Universal Single Ended 50 ohm Clock Input
- Internal or External DAC Clocking Option
- Ultra Low Skew and Jitter Clock Distribution
- Main Board PCLK Sourcing Capability
- High Precision Trigger Input Manufacturing Option:
    1.65V LVPECL, 2.5V LVPECL, 3.3V LVPECL
- Excellent SFDR and IMD Performance
  - SFDR = 80 dBc at F_{OUT} = 30 MHz (to Nyquist)
  - SFDR = 71 dBc at F_{OUT} = 130 MHz (to Nyquist)
  - IMD = -95dBc at F_{OUT} = 30 MHz
  - IMD = -70dBc at F_{OUT} = 130 MHz
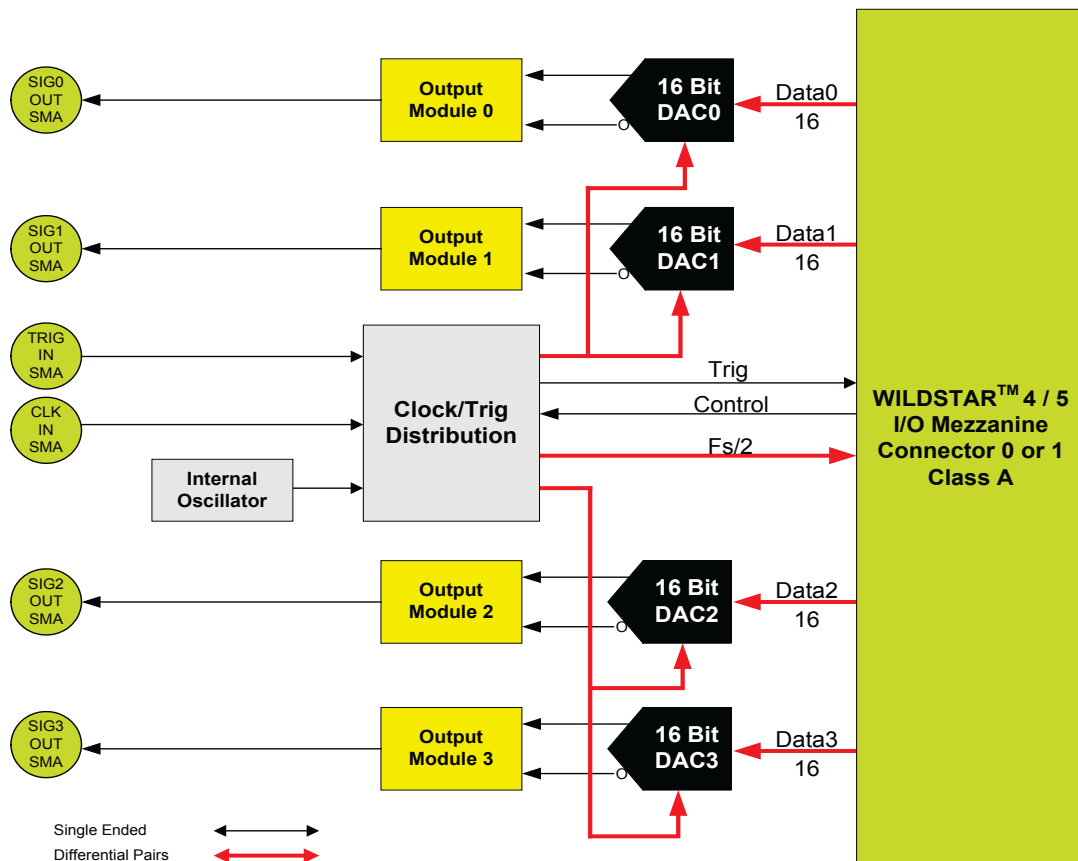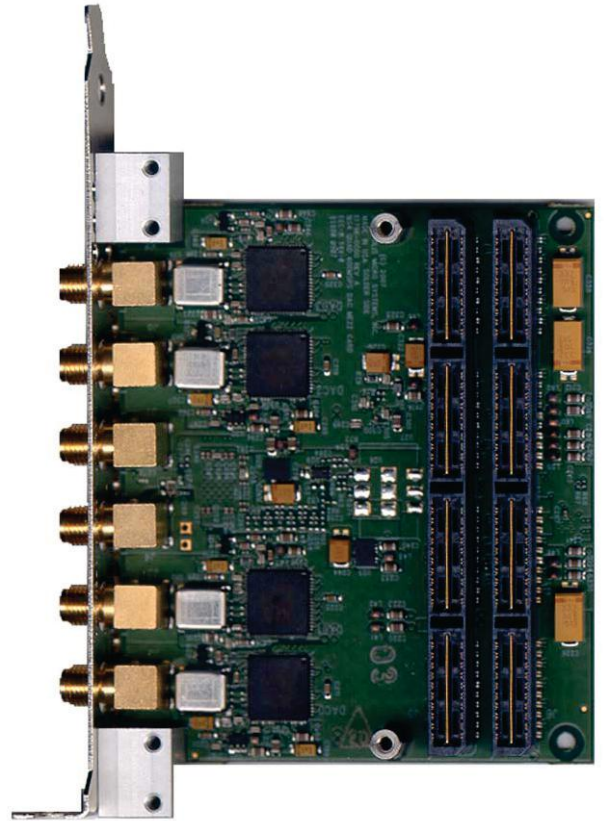
Dual 1.5 / 2.3 / 4.0 GSps 12 Bit D/A Converter
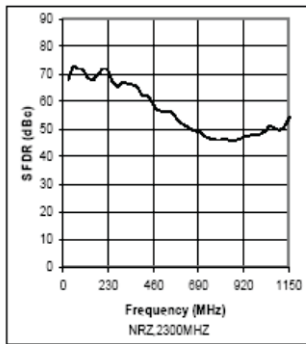
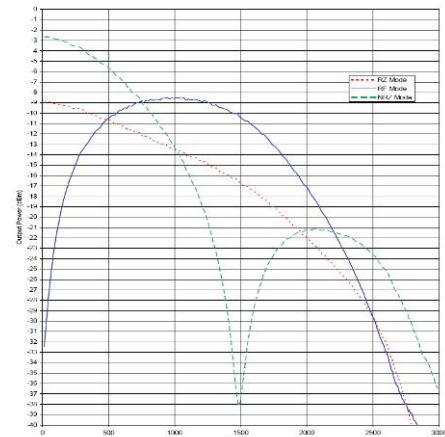

Figure 6-15: SFDR versus f_{out}, Fs=2300MHz, NRZ Mode
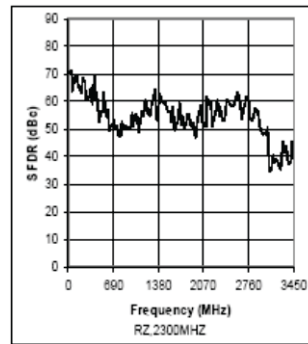
Figure 6-16: SFDR versus f_{out}, Fs=2300MHz, RZ Mode
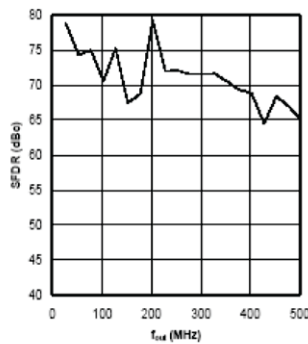

Full Scale Output Power at Fs = 1500 MHz
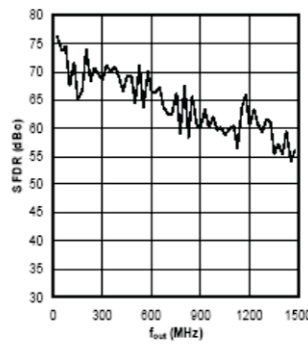

Figure 6-5: SFDR versus f_{out}, Fs=1000MHz, NRZ Mode
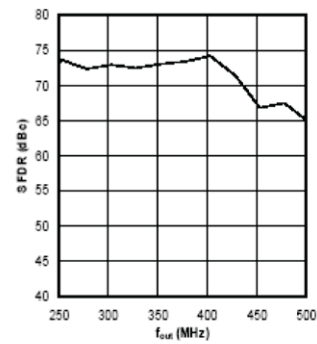

Figure 6-6: SFDR versus f_{out}, Fs=1000MHz, RZ Mode
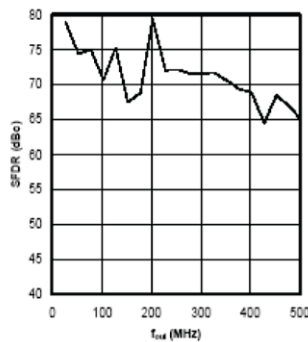

Figure 6-7: SFDR versus f_{out}, Fs=1000MHz, RF Mode
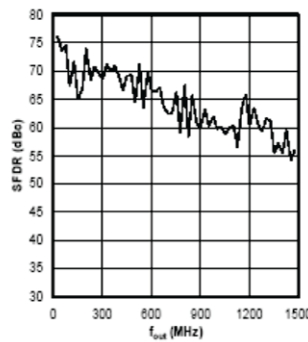

Figure 6-8: SFDR versus f_{out}, Fs=1500MHz, NRZ Mode
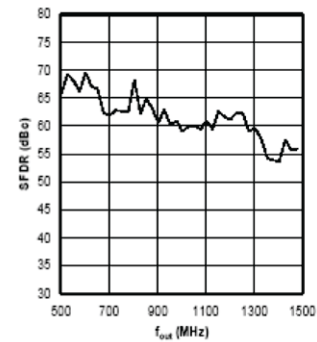


Figure 6-9: SFDR versus f_{out}, Fs=1500MHz, RZ Mode

Figure 6-10: SFDR versus f_{out}, Fs=1500MHz, RF Mode

Dual 1.5 / 2.3 / 4.0 GSps 12 Bit D/A Converter

Features

- One or Two Digital to Analog Converters of the type:
  - MAX5859   1500 MSps   12-bits
  - MAX19692   2300 MSps   12-bits
  - MAX19693   4000 MSps   12-bits
- Five SMA Front Panel Connectors
  - Two Single Ended DAC Outputs
  - High Precision Trigger Input with Fs Precision
  - Universal Single Ended or Differential 50 ohm Clock Input
- Internal or External DAC Clocking Option
- Ultra Low Skew and Jitter Clock Distribution
- Main Board PCLK Sourcing Capability
- High Precision Trigger Input Manufacturing Option: 1.65V LVPECL, 2.5V LVPECL, 3.3V LVPECL
- Excellent Gain Flatness in First 3 Nyquist Zones
- Multiple DAC Output Modes: RF, RZ and NRZ
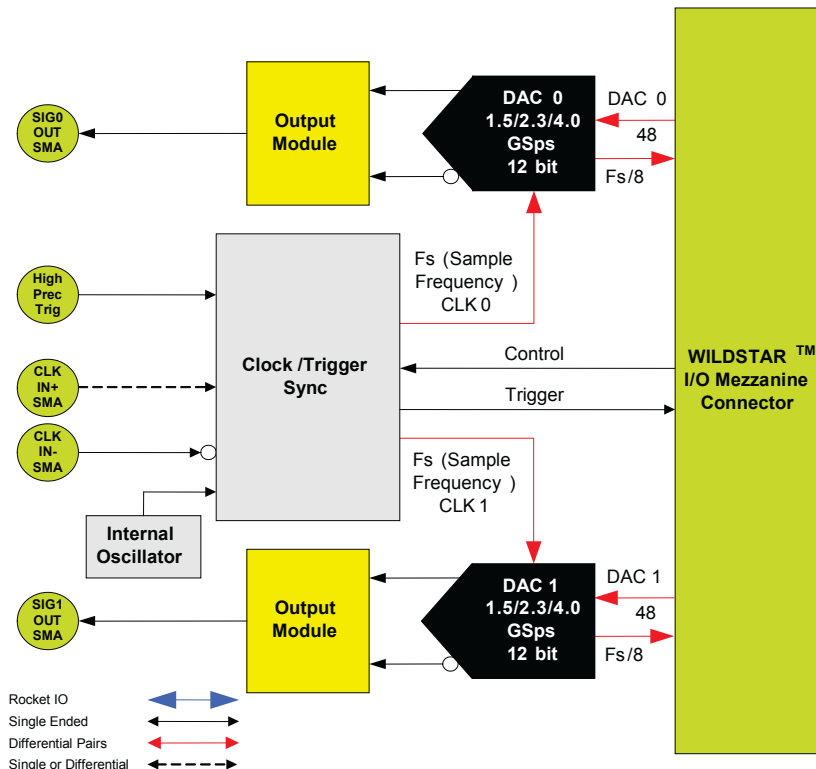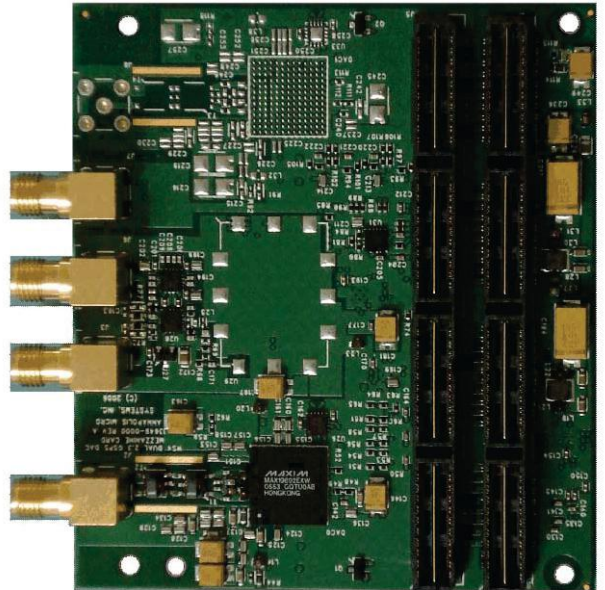
CoreFire^{TM} FPGA Application Builder
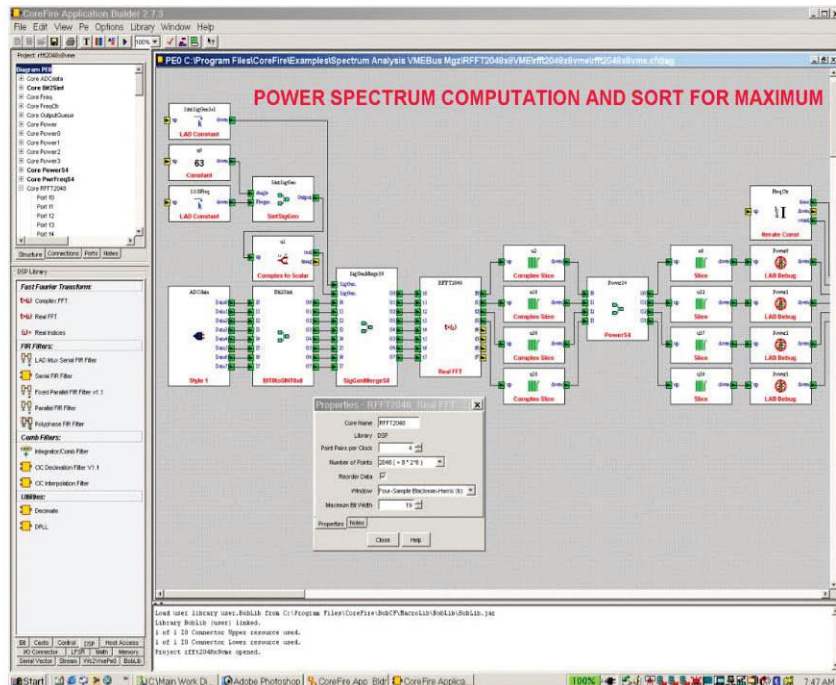
Build High Performance Applications
for Adaptive Computing Boards

Features

- Work from High Level, Data Flow Concept of Design
- Graphical User Interface Design Entry
- Thousands of Tested, Optimized CoreFire^{TM} Intellectual Property Cores
- Modules Automatically Handle Synchronization, Clocks, and Other Low Level Hardware Signals
- Board Support Packages Incorporate Hardware Details of Chips and Boards - Invisible to Users
- Hardware in the Loop Debug

Benefits

- Complete COTS Solution
- Save Time to Market
- Reduce Development Cost
- Reduce Risk
- Easy to Learn
- Works with Proven COTS Boards
- Concentrate on Solving Your Problem - Not on Low Level Hardware Details
- Very Easy to Modify Designs
- WILD^{TM} Solutions Outperform the Competition

Warranties, Updates, Customer Support and Classes

- WILD ^{TM} Boards Come with a 1 Year Hardware Warranty and 1 Year Soft/Firmware Update/Hotline Support
- Extended Hardware Warranty and Soft/Firmware Update/Hotline Support Available
- Hotline Support includes Access to Hotline Personnel, Via EMail, Phone and/or Fax, from 8:30AM - 5:30PM EST, Monday - Friday, except holidays
- Call Sales for class descriptions and schedules