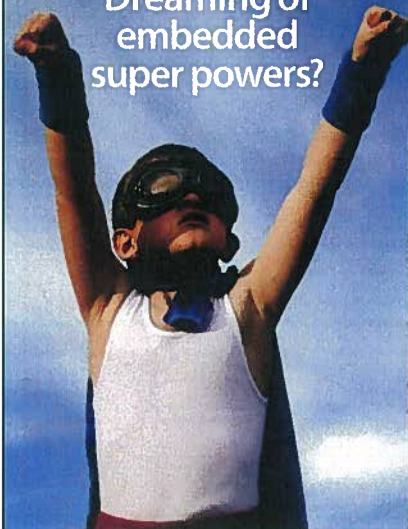


Dreaming of
embedded
super powers?



These boards
don't kid around.



Super strong
with high-performance ARM
cores and Pentium processors.



Super small
with boards 1/4 the size
of the 104[™] footprint.



Super cool
with extreme temperature ranges
and low-power operation.



Made in the USA
Single Board Computers • Microcontrollers • I/O Boards
PC/104 • EPIC • StackableUSB™ and more
Contact us at 818-244-4600 or online.

Distributing a Simulink® Model Across Multiple Heterogeneous Processors

Commonly available COTS solutions feature high-speed interconnected processing elements with ever decreasing geometries and ever increasing complexity. Modular COTS solutions can offer the designer an array of choice in terms of processors, interconnect, form factor, and memory. Designers are adopting these multiprocessor solutions because they can integrate both fixed and floating point digital signal processors (DSP), field programmable gate arrays (FPGA), and general purpose processors (GPPs) to enable the rapid prototyping and deployment of platforms that can accelerate time-to-market, be dynamically reconfigured in the field, and support a variety of high-end signal processing applications.

The 'rub' is that developments in signal processing hardware outpace developments in design tools and design methodologies. The resulting design gap has created a productivity and system-level-to-implementation challenge for engineers.

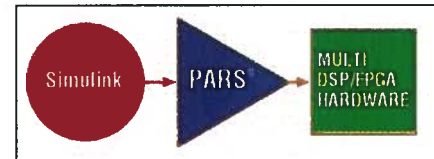
PARS

SundanceDSP created PARS (Parallel Application from Rapid Simulation) in response to technical and performance requirements set out by the US Navy. It led to the development of a communication solution code named "DSP8080-AIMM" (Altitude Interference Mitigation Module), which is a complete narrow band digital beam former system for communicating in hostile environments. PARS is a companion toolbox for The MathWorks™ Simulink® that enables designers to generate multiprocessor DSP/ FPGA applications from a single Simulink® model. PARS enables the engineer to partition a design into several tasks and assign them to multiple processors. Using PARS, a task is effectively a subsystem in a Simulink® model and any number of tasks can be assigned to a processor in the system, whether they be floating point DSP, fixed point DSP, FPGA, or host processor.

Inter-processor communication and synchronization code is built into the single application file and testbench that PARS creates from the Simulink® model.

The auto generated application code and testbench also contain all of the booting information for the entire multiprocessor platform. Expressly designed for ease-of-use, PARS also provides Hardware-In-the-Loop (HIL) capability to test the application, on the hardware, in real time.

By using PARS, designers are freed from having to create handwritten code for each processor target and then match it back to their Simulink® model. Fine grain code reporting, via the PARS



PARS capabilities at a glance.

profiler, enables designers to optimize the distribution of their model across the processors. It informs them about numbers of cycles and processing time for any selected task, be it on a DSP or FPGA. A single, deadlock-free application can be created that runs on all the target processors from one boot sequence allowing different flavors of the model under test to be quickly evaluated and benchmarked. For example, in a radar development, where different types of channel, antenna, modulation, and frequency information are required for different radar types, a developer can easily and quickly create different test applications directly from the model to experiment with.

Multiprocessor Model

The underlying model used by PARS for multiprocessing is based on communicating sequential processes (CSP). In this model, a computing system is a collection of concurrently active sequential processes that can communicate with each other over channels. A channel can transfer messages from one process to exactly one other process. A channel can carry messages in one direction only; if communication in both directions between two processes is required, two channels must be used. Both the sender and receiver agree on the size of

message being transmitted and the channels are blocking with a sending or receiving thread waiting until the thread at the other end of the channel attempts to communicate.

The System-Level to Implementation Gap

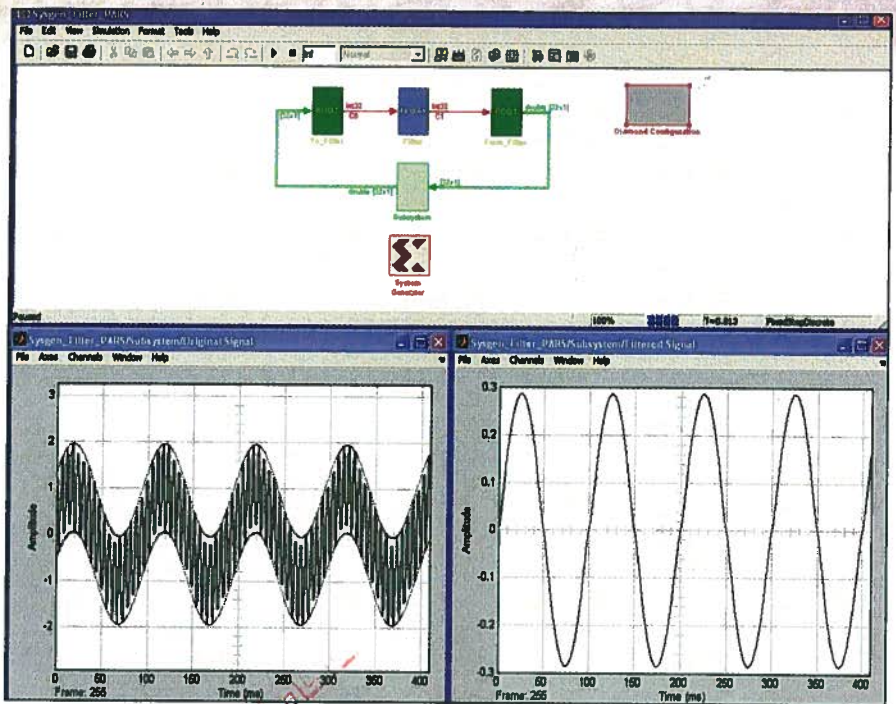
A key driver for the development of PARS was to provide a union between the system level model and the physical implementation. In typical multiprocessor development approaches requirements are captured, models created, and the developer hand codes the application in the preferred languages for the target processors; in the case of the advanced multiprocessing platform this can involve Assembly, C and C++, and the VHDL or Verilog hardware description languages (HDLs). Manual coding can be error prone and time consuming, require very different skill sets and engineering competencies (that are also critical to optimize the design partition), and as the refinement process iterates, the implementation can often move away from the original model. The corollary is greater difficulty in assessing the effect the differences will have on other components of the design at the system level.

This was one of the challenges faced by a radio design team at the Oak Ridge National Labs (ORNL). They selected PARS as a key element of their tool flow and design methodology to develop a software defined radar environment simulator (RES). The RES being developed by ORNL is responsible for presenting to the radar under test a variety of operator-defined scenarios that include radar returns from multiple simultaneous targets/objects and clutter sources

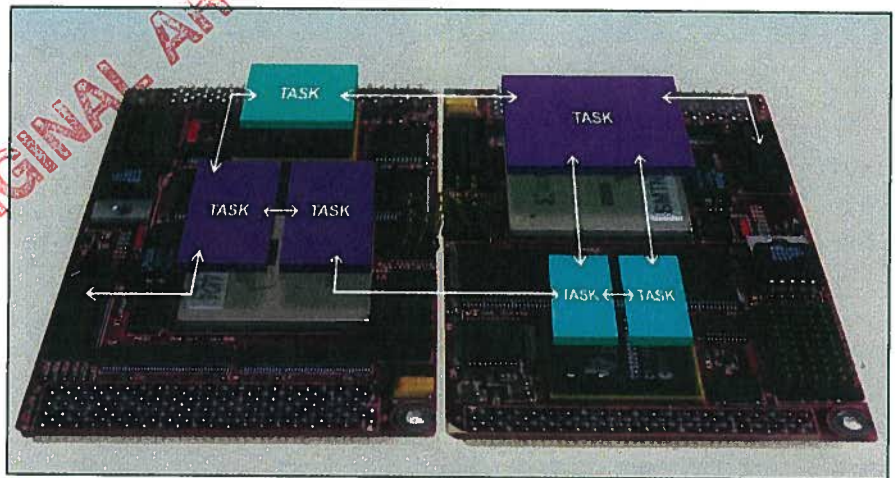
ORNL's Model-Driven Approach

The ORNL team adopted a model-driven design approach to deliver a system view of all their RES code/components, a seamless testbench, requirements traceability, and automatic code generation, among others.

Located in a VME chassis, for the RES project they selected multiprocessing solutions that consisted of: a DSP417 PMC/XMC multiprocessor module featuring dual fixed-point TI DSPs; an SMT374 dual floating-point TI DSP; an SMT318-SX55 module with two Virtex4 FX55 FPGAs; and an SMT351T module with a Virtex5 FX95 FPGA and 1GB of DDR2 memory. All the system components communicate via the Sundance High-Speed Bus (SHB) and Sundance



Sample PARS model and its simulation window.



PARS generated application tasks for target DSP/FPGA hardware.

Communication (SCom) core. An additional FPGA module from Vmetro featuring a Virtex II Pro and a Serial FPDP core was integrated into the system using SCom. SCom offers user defined, 2,4,8,16, or 32 bits wide, multi-port unidirectional inter-processor communication and delivers up to 800MB/s for a 32bit wide connection.

Once the ORNL team had designed and verified their application model in Simulink® they used PARS to aggregate and assign subsystems to the desired processor elements. Using PARS, multiple tasks/subsystems may be assigned to a single DSP or FPGA and the developers were able to quickly design and simulate platform-independent models and

then automatically generate code to rapidly prototype on the multi-DSP, multi-FPGA hardware.

Another critical element in the ORNL RES design was the ability to create and access multiple large look-up-tables. PARS MEM blocks allow the designer to manipulate large amounts of data in static arrays at run-time. The blocks are implementable on both DSP and FPGA tasks and during user task operation, and the protocol parser is able to access the contents of memory arrays. With added calls within the user task loop, it is also able to synchronize access to the static memory arrays. The generated code for PARS MEM block is automatically synchronized with code generated

Distributing a Simulink® Model

by Simulink® for the model. By utilizing PARS MEM block, the ORNL team can download/upload a section of memory inside a DSP or FPGA at any time, irrespective of the state of the underlying task. For example, the memory can be uploaded when the task is waiting for input, processing, or waiting for output.

Addressing an element of the design automation 'Holy Grail', PARS provided the ORNL team with automatic code generation for the final system. This gave the developers the greatest functionality in the shortest time. Critically,

PARS enabled the generation of code that delivered real-time signal processing performance that could be implemented within the computation resource defined by ORNL.

PARS auto-generates the final code by calling target specific tools, such as RTW-EC, Xilinx® System Generator, HDL Coder™, and Code Composer Studio™ (CCS). RTWEC generates C and C++ code optimized for embedded systems from Simulink®, Stateflow®, and embedded MATLAB® models. Simulink® HDL Coder™ generates bit-true, cycle-accurate

HDL (VHDL and Verilog) from Simulink®, Stateflow®, and embedded MATLAB®. CCS generates C and C++ code optimized for Texas Instruments DSPs. The automatic code generation requires that a restricted subset of available libraries be used, and this is supported by a rapidly increasing suite of available libraries.

For code compilation, loading and implementation, PARS compiles the single application by invoking 3L Diamond (that includes Diamond DSP and Diamond FPGA). Diamond FPGA encapsulates FPGA cores as tasks within its process flow, making the interfaces between FPGA and DSP tasks completely transparent. The Diamond configurator then uses the PARS generated configuration file and tasks to map the tasks onto processing elements and task communications channels on physical connections.

Squaring the circle on ORNL's requirements, PARS also supports on-hardware task performance profiling and is capable of profiling multiple parallel tasks during runtime, whether those tasks are assigned to a DSP or FPGA. On demand, PARS generates suitable code to extract low-level timing information during run time. This timing information is then sent to the host computer via a synchronized channel so that transmission does not affect the timing of the tasks. The resulting performance data is then benchmarked and baselined. The functional accuracy of the complete Simulink® model was initially verified on a host processor with the understanding that this instantiation of the model would not meet the real-time performance requirements.

Conclusion

ORNL's efforts are a testament to how vendors and customers can work together to integrate tool-flows that yield end-to-end design methodologies. By balancing design automation with designer control, fine grain reporting and tool interoperability, PARS has narrowed the design gap in model driven design for high-end signal processing implemented on COTS multiprocessing systems. Ongoing development and extended capabilities could make PARS an important part of a 'developer's toolbox when distributing a Simulink® model across multiple heterogeneous processors.

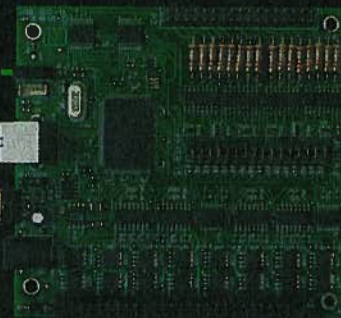
This article was written by Dr. Nory Nakhrae, Ph.D., CEO, Sundance DSP Inc. (Reno, NV). For more information, contact Dr. Nakhrae at nory.n@sundancedsp.com, or visit <http://info.hotims.com/22926-400>.

USB Embedded I/O Solutions Stackable or Unstackable USB

USB/104® Embedded OEM Series

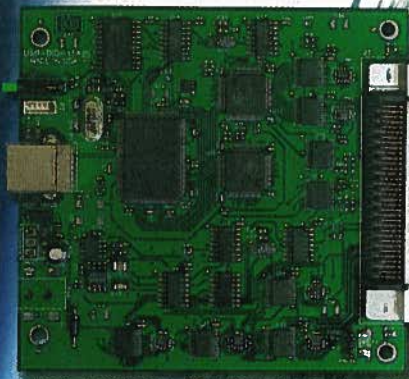
- Revolutionary USB/104® Form Factor for Embedded Applications
- USB Connector Features High Retention Design
- PC/104 Module Size and Mounting Compatibility
- Extended Temperature and Custom Options Available
- Choose From a Wide Variety of Analog, Digital, Serial, and Relay I/O

16-Bit Analog
Input 64-Channels
500kHz



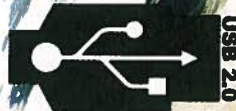
Isolated Digital I/O,
16 Inputs and 16
Solid-State Relay
Outputs

Digital I/O,
Sustained 16 MB/s
With 80MB/s Bursts



ACCES I/O Products' PC/104 size
embedded USB boards for OEM
data acquisition and control.

OEM System SPACE Flexibility
with dozens of USB/104® I/O
modules to choose from and
extended temperature options.
Explore the Possibilities!



Saving Space,
The Final Frontier

ACCES
I/O PRODUCTS, INC.

The source for all your I/O needs
To learn more about our Embedded USB/104® I/O boards visit
www.accesio.com
or call 800 326 1649. Come visit us at
10623 Roselle Street San Diego CA 92121

