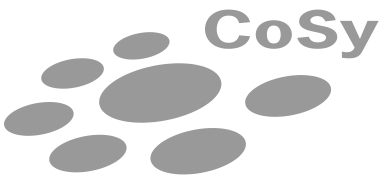

CoSy Embedded C Emulation

DSP Application Development on a Workstation & HW/SW Co-Design

White Paper

ACE Associated Compiler Experts bv

Version:	2008.4
Date:	March 26, 2008
Status:	release
Confidentiality:	public
Reference:	CoSy-8054-emulator

CoSy Embedded C Emulation

by ACE Associated Compiler Experts bv.

Printed in the Netherlands, March 26, 2008.

Using CoSy for DSP Application Development

The CoSy® compiler development system from ACE Associated Compiler Experts bv generates high performance compilers for an astounding range of architectures. CoSy is modular, flexible and comes with professional support from a stable team of leading compiler experts. Originally focused on the DSP market, ACE understands the specific needs of DSP developers and offers the ISO ratified Embedded C extension to C to address those requirements. By working closely with the industry, ACE continues to refine its product offerings to provide the most advanced compiler development system.

CoSy offers DSP developers the opportunity to access the market faster using its emulation capability. Embedded C emulation lets application development teams develop arithmetically accurate DSP applications on standard workstations. It can be applied when the target architecture has not yet fully crystallized, but it is also just more convenient than development on a simulator.

Why Embedded C

As DSPs continue to invade our daily lives, the contest to develop faster, more efficient products is key. DSP developers long for a standardized, efficient, high-level language that offers ease of use as well as portability. While assembly language programming delivers the performance required, it is tedious, difficult and non-portable. Naturally, most DSP developers will turn to assembly code to tweak the performance, but the bulk of the code has to come from another source.

C is the obvious language to turn to. It offers the ease of use and portability DSP developers require. Its pedal to the metal nature implies performance. However, some key elements are missing, making it difficult to efficiently implement for DSPs. ACE, along with members of the DSP industry, has developed a unique set of extensions that offers developers the ability to use the language they love with great success.

Embedded C is the ISO ratified extension to C for use in DSP application development. The Embedded C extensions to C include support for a number of DSP specific features:

- Fixed point data types allow the developer to use fixed point types that vary in size, rounding and saturation. Even accumulator types, that typically only live inside the multiply-accumulate functional unit to deal with overflow, are fully supported.
- Flexible memory support. When programming a DSP, it is crucial to have explicit control over the use of memories, such as X and Y, internal and external. These are especially important to algorithm performance, so the opportunity to control where variables are placed and how they are used and accessed delivers a better product.
- CoSy additionally offers circular buffers. Circular buffers provide a way to cycle through a buffer of data and return to the front of the buffer again once the end of the buffer is reached without incurring a cost penalty. If it is supported in the hardware, developers want to use it.

Embedded C offers DSP developers the opportunity to write their application in a portable manner. Regardless of the architecture used today, it can be recompiled for a different processor in the future. The investment into application development is preserved for the next generation product with minimal work.

The full definition of Embedded C can be found in the ISO Technical Report on Embedded C (TR 18037) that was ratified in 2004. The `www.embedded-c.org` website is dedicated to track tools, commercial and research developments related to Embedded C.

The Versatility of Embedded C Emulation

In a typical DSP flow, an application is developed in MatLab or C++. It initially runs on a workstation and is then painstakingly morphed into a combination of C and assembly that is specific for the target DSP architecture. The morphed code does not run on the workstation any more because of its use of target specific extensions and assembly. While such flow is typical, it is definitely not optimal. It requires maintaining multiple sets of code and mutual updates with each iteration.

This non-optimal flow is necessary because standard C does not have the performance critical arithmetic types, notably fixed-point data types, that are necessary in optimized DSP applications.

Embedded C solves the first half of the equation, providing the time-critical extensions that are necessary to make the code run efficiently on the target architecture. But what about the developer? As these extensions are not available on the typical workstation processor, Embedded C is not straightforwardly mapped to it.

This is where CoSy's emulation technology comes in. A CoSy "emulation compiler" can be generated by providing just two descriptions: one of the exact, bit-accurate, arithmetic properties of the target DSP processor's data types, and one of the data types available on the workstation's host-processor, which typically lacks fixed-point data types. With just these two descriptions, CoSy's emulation technology generates a compiler, including the necessary libraries, that maps the target processor arithmetic to the host processor arithmetic.

Using this emulation compiler to compile Embedded C code targeted for the DSP processor creates an executable that runs on the workstation *but does all the arithmetic exactly as if it was running on the DSP*. If the target is a 24-bit DSP, all the arithmetic will be done with exactly those 24 bits of precision, even though the workstation offers plenty more bits.

As a result, Embedded C emulation simplifies the design flow significantly, reducing both the headaches associated with maintaining two sets of code, as well as increasing market opportunity by reducing development time. CoSy's Embedded C emulation allows to develop, test and debug DSP application algorithms on the PC or workstation without having access to the actual processor or even a simulator. And when the actual target code needs to be produced, the same code compiles directly using CoSy's full Embedded C compilation technology.

HW/SW Co-Design

But it does not end here. CoSy's emulation technology offers a unique opportunity for architecture exploration. Since the actual processor is not required to test and debug applications, it is easy to try different architecture and application combinations. How many bits of precision are needed to run the application at the required accuracy? To determine the optimal implementation, just generate a number of emulation compilers by specifying word lengths, saturation and overflow behavior. Taking these results into the architecture design can save significantly in the cost of hardware and power. Thus, critical architecture parameters can be investigated before committing to the detailed hardware design investment. No other compiler development system on the market offers this capability.

In addition to architecture exploration, the emulation capability creates a tight feedback loop between hardware and software developers. The application domain can give feedback to the architecture group while the processor is still being developed, identifying features that could be added or deleted to improve application performance. Hardware/Software co-design has never been more true.

CoSy for DSP compilation: The Only Solution

ACE understands the needs of DSP developers. Our customers are successful because we work together to deliver the tools that they need. We believe Embedded C is the current wave in DSP design. Check out CoSy at `www.ace.nl` today and see for yourself how easy DSP compiler development can be.