

# The Challenges of New Software for a New Architecture

By [Richard Wirt](#),

Intel Fellow and Director, MicroComputer Software Labs  
Intel Corp.

In this Q4, 1999 issue of the *Intel Technology Journal*, we look at some of the software efforts that have gone into bringing out the new IA-64 architecture. Early in the development of the IA-64 architecture, we set very aggressive goals for the software compilers, floating point performance, and simulation environment.

Over the years, the compiler has become a very important factor in contributing to the utilization of a processor's architecture. The Intel IA-64 compiler, code named Electron, was considered part of the IA-64 architecture. We had compiler architects working side by side with CPU architects. This was very important since the IA-64 is a new architecture that exploits many new concepts, and the microarchitecture depends on the compiler to manage many of the resource dependencies. Not only has this compiler served to bring up many of the initial operating systems and applications, it has also served to help other third party compiler vendors to understand how to generate code for this new architecture. The compiler has met the challenge of pushing the boundaries and of achieving our initial architecture goals.

Being a new architecture, the IA-64 architecture provided Intel with the opportunity to take another look at the floating point on the IA-64 architecture. We set a goal of making it faster, making it fully IEEE compliant, and of achieving a near 0.5 units in the last place of precision for the

transcendental functions libraries. The width of the architecture allowed us to take another look at the traditional algorithms. Since divide and square root are executed in software in the Itanium™ processor implementation of the IA-64 architecture, we have also formally proven the algorithms used.

Perhaps the biggest challenge was to simultaneously bring up and debug a new architecture, new chipsets, new compilers, and new versions of the operating systems. To do this, we set our sights on building a full system-level simulator that functionally behaved faithfully at the register and I/O ports levels as the first platforms were being built. Additionally, we wanted to be able to add performance simulators for the CPU, caches, and chipsets. The software simulator became known as the SoftSDV. It met its requirements to be functionally equivalent to the first software development vehicles manufactured for use by the ISVs and OS vendors. Well before first silicon was available for the Itanium processor, we were running the firmware, all the major operating systems, and many major applications on the SoftSDV. This served its purpose very well, as we were able to run the same binaries on the real SDVs, using new silicon, within hours of their availability from Intel's manufacturing fabs.

With this issue of the *Intel Technology Journal*, we hope you will get a better insight into the software behind Intel's new 64-bit

architecture and gain an appreciation of the outstanding efforts of the many people on the software teams that contributed to the IA-64 software technologies.

Copyright © Intel Corporation 1999. This publication was downloaded from <http://www.intel.com/>.

Legal notices at <http://www.intel.com/sites/corporate/tradmarks.htm>