

## Preface

[Lin Chao](#)

Editor

*Intel Technology Journal*

With the close of the year 1999, it is appropriate that we look forward to Intel's next generation architecture--the Intel Architecture (IA)-64. The IA-64 represents a significant shift for Intel architecture: it moves from 32-bits to 64-bits. Targeted for production in mid-2000, the Itanium™ processor will be the first IA-64 processor. One of Intel's key aims is to facilitate a transition of this magnitude. To this end, Intel and key industry suppliers are working to ensure that a complete set of "ingredients" is available for the IA-64 architecture. This includes operating systems, compilers, and application development tools that are "64-bit capable." In this issue of the *Intel Technology Journal*, you will learn about Intel's efforts in IA-64 software technology.

In the first papers, Intel's own IA-64 compiler efforts are discussed. The first paper gives an overview of Intel's production compiler, code named "Electron." We then move to the second and third papers where software development tools for the IA-64 architecture are discussed. As is often the case with a brand new architecture, software companies start developing software in advance of actual hardware. SoftSDV (a software-based system development vehicle) is a tool that simulates IA-64 hardware platforms in lieu of actually having the hardware. This tool assists engineers in porting commercial operating systems and applications to the IA-64. Intel's IA-64 Assembler is described in the fourth paper. The Assembler can simplify IA-64 assembly language programming.

Validation is a critical function in the testing of new circuits such as those on IA-64 silicon. The fourth paper describes the porting of the Linux\* and Mach\* operating systems and how they run on software simulators to exercise operating system related functionality in the IA-64 architecture.

The final two papers discuss the floating-point functions on the IA-64 architecture. Fast and accurate computation of transcendental functions (e.g. sin, cos, and exp) and the implementation of floating-point operations are also discussed.