

Preface

[Lin Chao](#)

Editor

Intel Technology Journal

Welcome to 1999, and the 1st Quarter issue of the Intel Technology Journal. The focus of this issue is Intel's current CAD tools, the tools used to design Intel's microprocessors. In the early 1960s, the process of designing the first integrated circuit was entirely manual. As one might guess, this was very expensive, laborious, and error-prone. However, in the 1970's, software CAD tools came to the rescue and became an integral part of the design of complex microprocessors produced by Intel.

These complex CAD tools are the focus of this issue. The first paper describes the architectural direction of Nike, Intel's next generation CAD tool suite, which supports code sharing to improve development efficiency, tool quality, and maintainability. As microprocessors approach deep sub-micron dimensions, the impact of physical design must be considered early in the circuit design phase to prevent costly re-designs. The second paper discusses Intel's FUB Circuit Design Environment combining circuit design with physical layout planning.

Datapath design (from RTL to layout) can take more than 60% of a project's human resources. In the third paper, a new design workflow is proposed along with a set of tools that will further automate the design process. The hope is that this proposed workflow will spur both academia and industry to tackle the problem of more automated datapath design tools. The fourth paper looks at the challenge Intel faces in making tools run on UNIX* and Windows NT* operating systems running on Intel and non-Intel architecture platforms. Tools and methods for formal verification at gate-level description are also discussed in the fifth paper. An example of formal functional verification of gate-level floating-point designs against IEEE-level specifications is presented. Structural and functional testing tradeoffs are described in the final paper. Again we feel that there are opportunities for the industry to provide more robust and scalable solutions for defect-based testing.