



# INTRODUCING THE ITANIUM PROCESSORS

**John Crawford**  
Intel

..... The press and the technical community have generated much excitement and speculation about the IA-64 instruction set and the Itanium processor. Intel and Hewlett-Packard have rolled out (one instruction at a time) the instruction set. Intel is rolling out (one transistor at a time) the Itanium processor and other platform components. This special issue provides the broad technical community with a comprehensive introduction to the concepts and mechanisms that

form the basis of the IA-64 instruction set and the related Itanium processor products.

These articles were written well in advance of the product introduction for the Itanium processor. Consequently, the articles focus on describing *what* will be provided as these concepts and products come to market rather than *how well* the products and concepts work. Our intention is to provide grounding in the products and concepts to *IEEE Micro* readers.

## History

My involvement with the IA-64 instruction set architecture started in December 1993 when HP and Intel met to discuss the opportunity to collaborate on high-end microprocessors. I was chartered to lead a group of Intel architects, software experts, and chip designers to evaluate the technical merits of concepts HP might bring to such a partnership, and to present our technical capabilities to a corresponding group of HP experts for their evaluation.

During the first half of 1994, these two teams met and exchanged information on our 64-bit directions and capabilities. To provide protection of our mutual intellectual property, the teams agreed to establish a neutral site at an HP training office in Santa Clara, roughly midway between the Intel site in Santa Clara, the HP site in Cupertino, and HP Labs in Palo Alto. A conference room was dedicated to the task, with two large filing cabinets, one assigned to each company. All of our notes, along with any printed material received from the other company, were stored in these cabinets and locked when we left the room. All we could leave with was what was in our heads. If HP and Intel decided not to form a joint activity, the agreement was to destroy the contents of the cabinets. There were no corresponding restrictions placed on our use of the residual information content in our brains.

Corresponding teams met to determine the business merits of such a partnership, and the merits of cooperation in other technical areas. The investigations went well, and on 6 June 1994, HP and Intel announced formation of a joint research activity to develop a 64-bit instruction set architecture that would become the basis for a line of Intel microprocessors. At that time I was asked to lead an HP-Intel team to jointly define the instruction set architecture.

This architecture team was formed from both Intel and HP with computer architects, compiler and operating system software experts, and chip designers. Over the next year or two the joint team produced a series of specifications that reflected the best concepts, judgment, and data from both companies and external researchers. (Schlansker and Rau recount the HP history.<sup>1</sup>) Functional and performance simulators were built, and a proto-

type compiler based on the University of Illinois Impact compiler system<sup>2</sup> was targeted at the evolving instruction set spec.

The joint instruction set development preceded a massive effort at Intel to prepare all of the products needed to successfully launch a new computer architecture. Intel started first one, then several, processor designs to implement the IA-64 instruction set. Compiler and operating system software development projects replaced prototype software development, and application software development began. As this is published, Intel is preparing to bring to market its biggest product line since the launch of the 386 processor in 1985. Certainly, the next several years will continue to be interesting for all of us involved in Intel's Itanium processor products.

## EPIC and ILP

The IA-64 instruction set is based on a set of concepts that we describe as EPIC, for Explicitly Parallel Instruction Computing. Our belief is that EPIC is the next advance beyond RISC that's needed to keep on the performance treadmill defined by Moore's law of doubling performance every 18 months, or an annual growth rate of 1.6 times. Improvements in the underlying silicon technology yield about a 1.2-times annual improvement rate via faster silicon devices. The rest must be made up with improvements in circuit design and in parallel execution, by overlapping the execution of more instructions through deeper pipelining (to enable a higher clock rate) and/or executing more instructions in parallel. This second kind of parallelism is known as instruction-level parallelism, or ILP, and is measured as the number of instructions executed each clock cycle (IPC).

Our belief is that the EPIC techniques will enable us to stay on the curve of increasing levels of ILP.

## Hardware-compiler interface

EPIC is based on the premise that the compiler has much better visibility into program execution than does the hardware. Certainly the compiler can look at a much larger window, and it has more time for analysis (seconds versus nanoseconds). What it's missing is knowledge of individual dynamic events such as individual branch taken/not taken,

and cache hit/miss, although there may be statistical aggregate information on these events if profile data is available.

There are three main tenets of an EPIC architecture. It provides

- mechanisms to enable the compiler to arrange the computation efficiently based on its global knowledge,
- sufficient resources such as registers and functional units to perform multiple operations in parallel, and to store the “inventory” of intermediate results, and
- instruction formats that let the compiler communicate to the hardware the key information it has gleaned from the program as it’s compiled.

Each of these three key EPIC characteristics are embodied in the IA-64 instruction set and are described in the first article, “Introducing the IA-64 Architecture.” This article describes the key concepts and mechanisms provided in the instruction set at a high level, along with the main motivations behind the inclusion of these mechanisms. The focus of this article is on the instruction set as seen by all programs (both applications and the operating system). Virtual memory support in IA-64 includes better support for sharing (up to and including support for a single address space) that we expect to happen with the large increase in addressing that comes with a 64-bit architecture.

“The Itanium Processor Microarchitecture” describes the internal structure of the first processor to implement the IA-64 instruction set. It emphasizes how the processor supports EPIC constructs such as predication, control and data speculation, and multiway branches.

The compiler is a key component of performance in an Itanium processor-based system. “The Intel IA-64 Compiler Code Generator” describes the structure and considerations for the code generation phase of a compiler that targets the IA-64 instruction set from the point of view of how the generated code makes use of EPIC constructs.

“The AzusA 16-Way Itanium Server” describes the server system developed by NEC for enterprise-level computing applications. The article discusses the system architecture and the partitioning and clustering

structure of this large system with a remote and local latency ratio of 1.5, and RAS features to support enterprise applications.

“High Availability and Reliability in the Itanium Processor” describes a set of capabilities for detecting, containing, reporting, and recovering from hardware failures in processor and external buses. These features are important in the enterprise-computing application area that is a key target of the initial Itanium processor systems.

**W**e’ve scheduled more articles for upcoming issues of *IEEE Micro*.

---

### References

1. M.S. Schlansker and B.R. Rau, “EPIC: Explicitly Parallel Instruction Computing,” *Computer*, Feb. 2000, pp. 37-45.
2. W.W. Hwu et al, “Compiler Technology for Future Microprocessors,” *IEEE Proc.*, Vol. 83, No. 12, Dec. 1995, pp. 1625-1640.

**John Crawford** is an Intel Fellow who has worked at Intel for 23 years, 5 as a software developer and 18 as a computer architect. Prior to managing the joint Intel-HP team that defined the IA-64 instruction set, he was the chief architect of the 386 and 486 microprocessors and comanaged the Pentium processor development. Crawford earned a BS degree from Brown and an MS degree from the University of North Carolina, both in computer science. He received the ACM-IEEE Eckert-Mauchly award in 1995 and the IEEE Ernst Weber Engineering Leadership Recognition in 1996.

Direct comments about this special issue to John H. Crawford, Intel Corp., SC12-304, 2200 Mission College Blvd., Santa Clara, CA 95052-8119; john.h.crawford@intel.com.