

Course 205

Itanium® System Architecture Overview

A 2-day Course

With its massive on-chip resources, several parallel execution pipes, and support for large, fast caches the Itanium Architecture creates new opportunities and challenges for both hardware, system-level, and application-level developers. This course will bring you up-to-speed on Itanium's architectural features at both the processor and system levels as well as their impact on software development approaches.

You will benefit from this workshop if you

- Are evaluating Intel's Itanium processors for future designs
- Want to keep up-to-date with the latest features and capabilities in the Itanium processor family

You will learn

- What traditional bottlenecks the Itanium Architecture eliminates
- The innovative register structure of the Itanium family CPUs
- How to exploit the CPU's capabilities when developing system or application software
- The key features of the chipsets that support Itanium-class processors

Prerequisites

A technical background is expected, including a solid understanding of general computer architecture and terminology. Familiarity with the IA-32 hardware architecture at the server or workstation level is helpful but not required.

The training approach

- **Up to date information:** We update the materials before every event.
- **Straightforward explanations:** Technical concepts and terms are explained in English. You will walk away with a thorough understanding of what the Itanium Architecture brings to the table and how to exploit its capabilities.

Workshop topics

Itanium's solution to traditional CPU architecture bottlenecks

- EPIC: Explicitly Parallel Instruction Computing – No more sequential semantics
- Remove branches with predicated instructions
- Minimize impact of memory latency
- Decrease procedure call overhead
- Minimize loop optimization overhead
- Massive memory resources
- Massive register resources
- The Register Stack Engine (RSE)

The Itanium micro architecture

- Processor pipelines
- Processor block diagram
- Instruction processing
- Dispersal logic
- Execution units
- Pipeline control
- Exception handler
- Cache subsystem
 - Level 1 instruction and data caches
 - Level 2 and Level 3 cache
 - Cache coherency
 - ALAT and TLB
- IA-32 execution
- Memory addressing
 - System memory model
 - Virtual memory model
 - Memory alignment
- I/O addressing

Itanium Product Family (IPF) chipsets

- Itanium Tukwila – Socket- and chipset-compatible with Xeon
 - Integrated memory controller
 - QuickPath replaces the Front Side Bus
 - Intel standardized Northbridges and Southbridges
- The E8870 chipset
 - Block diagram
 - Chipset highlights
 - Scalable Node Controller (SNC)
 - Scalability Port Switch (SPS)
 - Server I/O Hub (SIOH)
 - PCI/PCI-X Hub (P64H2)
 - Interface Control Hub (ICH)
 - Firmware Hub (FWH)

A new way to boot

- Why change?
- The Itanium boot process
 - *Tiano – The Intel Platform Innovation Framework for EFI*
 - EFI – Extensible Firmware Interface
 - The operating system loader
- Runtime services & protocols
- Firmware address space
- Firmware Interface Table (FIT)
- New disk layout

Tools for integration and test

- Test Access Port – TAP
 - Interface
 - TAP registers
 - Instructions for Itanium TAP
 - Reset behavior
- Integration tools
 - In-target probe
 - Logical modeling tools

Itanium processor package

- Cartridge features
- Mechanical dimensions

Application level software

- System environment
 - Instruction set transitions
 - Compiler to processor communication
 - Control and data speculation
 - Predication
 - Branching
 - Register rotation
 - Floating-point architecture
- Execution environment
 - General registers
 - Floating point registers
 - Predicate registers
 - Branch registers
 - Application registers
 - User mask
 - Processor identification
- Application programming model
 - Integer computation
 - Compare instructions and predication
 - Memory address instructions
 - Data pre-fetch hints
 - Branch instructions

System level software

- Itanium system environment
 - System environment
 - Privilege levels
 - System state registers
 - Virtual memory support
 - Region registers
 - Translation Lookaside Buffer (TLB)
- Interruptions
 - Four types of defined interruptions
 - Interruption programming model
 - Interruption handling
- Register Stack Engine (RSE) and backing store
- Debugging and performance monitoring
- Context management

Available operating systems

- Unix
- Linux
- Windows
- VMS
- Others