IA-64 & x86-64 ISA’s

Competing specifications for the 64-bit microprocessor market

2 February 2005

/mtu/ee5970/f02/btd
Divergence approaches

• The x86 ISA has been the dominant ISA for many years
  - INTEL, AMD, Cyrix, Transmeta, Via
  - The ISA has been augmented repeatedly through the generations {386 extensions, MMX, SSE}
  - New ISA’s have failed {Alpha, i960, etc} due to compatibility issues

• Both Intel & AMD are designing “next generation” processors with a new ISA
  - Intel : Itanium (IA-64) available since 1H01
  - AMD : Hammer (x86-64) available since 2H03
AMD Processor Roadmap
AMD Processor roadmap

[Diagram showing AMD processor roadmap with various models and release dates]
Intel Processor Roadmap
Intel Roadmap
Why 64-bits

• 4 GBytes of memory corresponds to a 32-bit address
  ■ If a program desires to address more than 4GBytes of memory, the addressing must be extended beyond 32 bits

• Increased precision
  ■ Floating point values >64 bits: 64 bit functional units

• More Parallel Data
  ■ DSP / Multi-Media Vectors have a 128-bit wide data path (SSE2)
Fundamental Differences

• IA-64 is a clean break from previous ISA’s – but it will still execute x86 instructions (in emulated i.e. slow mode)
  ■ Adds functionality to increase processor performance
  ■ Makes the job of the hardware designer easier

• x86-64 is philosophically (some would say pathologically) another extension to the ages-old x86 architecture.
  ■ Has a single execution pipe for both legacy x86 and new x86-64 code
  ■ Extends address and data registers (x86 does NOT really have GPRs) 64 bits
RISC vs. CISC

- The performance differential between CISC & RISC has been reducing since the debate began
- x86 (CISC) processors are essentially RISC at their core with translation in decode
- Updated graph would probably show x86 (CISC) with higher perf than any RISC

- Used as a motivation why x86-64 (CISC) need not be radically overhauled
X86-64 Intro

- Opteron (Sledgehammer) – Server Market
- Athlon64 (Clawhammer) – Consumer Market
- AMD is making the Hammer range backwardly compatible with current x86 code.
  - adding 8 new 64-bit (now 16) general purpose registers (GPRs)
  - 64-bit versions of the original 8 x86 GPRs.
  - Has SSE & SSE2 support with 8 new (128-bit) SSE2 registers
  - Increased memory addresses ability for large processor intensive software.
64-bit flat virtual addressing

• x86-64 "long -mode" supports 64-bits of virtual address space, more memory than current computers (or super-computers)

• Opteron has
  ■ 48-bits of virtual address space
  ■ 40-bits of physical address space

• A single design platform for consumer and workstation level processor offerings
Wider GPR's to 64-bits

• The x86-64 supports 64-bit integers so very large numbers can be handled accurately & quickly with Legacy mode support for x86 mode: 16 & 32-bit operating systems.

• Long mode support allows 64-bit operating system to execute 16 & 32-bit software. 64-bit Long mode for advanced 64-bit operation via x86-64 extensions.

• 64-bit Virtual address space.
x86-64 Registers

- Adds 8 new (much needed) GPR
- SSE are DSP/Graphics operations for vectors of low-precision values
- Most registers only available in new 64-bit Long mode
x86-64 Operating Modes

- Three operating modes
- (2) modes are directly backward compatible
- (1) mode will continue to execute old operating systems
- Three modes enable performance improvement with existing hardware as new operating systems and software are developed and installed
Opteron System design

- Significant Changes to Bus Topology
- Uses HyperTransport Protocol
- DRAM controller on the processor die
AMD

• SledgeHammer codename became OPTERON Trademark

• ClawHammer codename became Athlon64 Trademark

• Systems were Demo’ed as early as Feb 27, 2002
IA - 64

• IA-64 is a 64-bit architecture—a first for Intel. In IA-64 designs, instructions are scheduled by the compiler, not by the hardware.

• Much of the logic that groups, schedules, and tracks instructions is not needed thus simplifying the circuitry and promising to improve performance.

• EPIC – Explicitly Parallel Instruction set Computing
  Not CISC, Not RISC – but very much like VLIW
The Compiler’s Goal

• Generate code to take advantage of Instruction Level Parallelism
  ■ The whole point of EPIC is to generate code that can be executed in parallel
  ■ Because of this, a compiler that simply “works” does not put out code that runs at average or reasonable speeds
  ■ The processor optimizations have been moved to become a compiler responsibility
  ■ The compiler makes an incredible difference in the performance of the processor
  ■ This is bleeding edge compiler technology
  ■ Operating systems and applications compiled on a “functional” compiler rather than a fully optimized compiler will probably be slower than current versions on current 32bit CPUs.
IA-64 Strategies

• Extracting parallelism is difficult
  - Existing architectures contain limitations that prevent sufficient parallelism on in-order implementations

• Strategy
  - Allow the compiler to exploit parallelism by removing static scheduling barriers (control and data speculation)
  - Enable wider machines through large register files, static dependence specification, static resource allocation
Instruction Bundling

- Uses a form of VLIW architecture
- Referred to as EPIC – Explicitly Parallel Instruction set Computing
- Three Instructions are combined into a 128-bit instruction bundle

- Facilitates Parallel Instructions.
- Instructions are executed in groups
Processor State

• Directly accessible CPU state
  ■ 128 x 65-bit General registers (GR)
  ■ 128 x 82-bit Floating-point registers (FR)
  ■ 64 x 1-bit Predicate registers (PR)
  ■ 8 x 64-bit Branch registers (BR)

• Indirectly accessible CPU state
  ■ Current Frame Marker (CFM)
  ■ Instruction Pointer (IP)

• Control and Status registers
  ■ 19 Application registers (AR)
  ■ User Mask (UM)
  ■ CPU Identifiers (CPUID)
  ■ Performance Monitors (PMC, PMD)

• Memory
Predication

• **Branches interrupt control flow/scheduling**
  - Mispredictions limit performance
  - Even with perfect branch prediction, small basic blocks of code cannot fully utilize wide machines

• **Strategies**
  - Allow compiler to eliminate branches (and increase basic block size) with predication
  - Reduce the number and duration of branch mispredicts by using compiler generated branch hints
  - Allow compiler to schedule more than one branch per clock - multiway branch
Predication Concepts

• Branching causes difficult to handle effects
  ■ Istream changes (reduces fetching efficiency)
  ■ Requires branch prediction hardware
  ■ Requires execution of branch instructions
  ■ Potential branch mispredictions

• IA-64 provides predication
  ■ Allows some branches to be moved
  ■ Allows some types of safe code motion beyond branches
  ■ Basis for branch architecture and conditional execution
Predication Example

- The Branch is eliminated and parallel (speculative) instructions are generated
  - Increases fetch efficiency
  - Eliminates Control Hazard
Parallel Compares

- Parallel compares allow compound conditionals to be executed in a single instruction group.
- Example:
  ```c
  if ( a && b && c ) { ... }
  ```
- Assembly:
  ```asm
  cmp.ne p1 = rA,0
  cmp.ne.and p1 = rB,0
  cmp.ne.and p1 = rC,0
  ```
• Predication registers can significantly reduce the control dependencies in applications
Prefetching

• 65\textsuperscript{th} bit of GPR’s represents NaT
  ■ NaT: Not-a-Thing
    ■ Represents an exception in speculative state
  ■ NaN: Not-a-Number
    ■ Represents +/- infinity & indeterminate values in IEEE 754 Floating Point std.

• Loads can be hoisted out of basic block & up the instruction stream
  ■ Architecturally knowledgeable Compiler can guess the appropriate number of cycles of prefetch
  ■ With the 65\textsuperscript{th} bit representing speculative exception even USE of a value can be hoisted
Value Speculation

- Enables Prefetching – but has far more uses
Rotating Register File

- **Procedure calls interrupt scheduling/control flow**
  - Software modularity is standard
  - Call overhead from saving/restoring registers

- **Strategy**
  - Provide special support for software modularity
  - Reduce procedure call/return overhead
    - Register Stack
    - Register Stack Engine (RSE)
Rotating Registers

- 128 Integer registers and 128 Floating Point registers
- 32 Static registers and 96 Rotating registers.
  - Similar to SPARC register windows
- Amount of rotating registers is programmable.
- Uses register renaming to implement the rotating registers.
The Register Stack

- Procedures are stored on the register stack.
- Each procedure frame overlaps to simplify parameter passing between routines.
The RSE is used for two main functions

- The automation of register saves and restores in hardware across procedure calls.
  - Maps the stack to memory.
- Its ability to be used to make use of unused memory bandwidth for fill and spill operations.
Software Pipelining

• Rotating Register Bases
  ■ Circular Buffer of General and FP Registers
  ■ Loop Branches Decrement both RRBs
  ■ Makes data transfer between stages transparent \( \Rightarrow \)
    same virtual registers are used in each loop iteration

• Advantages
  ■ Traditionally performed through loop unrolling
  ■ Lower overhead, less code, increased regularity
  ■ Especially useful for integer code with small number of
    loop iterations
Software Pipelining

- Software Pipelining
  - Overlapping Loop Iterations
Floating Point Architecture

- The 82-bit IA-64 floating point architecture enables a lot higher precision and range of values.
  - Smaller rounding error
  - Iterative calculations converge faster
Floating Point New Features

• Combined Multiply and Add operations improve performance.
• With 128 Registers, a lot more than its predecessor architectures, the IA-64 enables greater resource availability.
32-bit Compatibility

- Itanium directly executes IA-32 binary code
- Seamless Architecture allows full Itanium performance

![Diagram showing compatibility flow, including steps like Compatibility Fetch & Decode, IA-32 Dynamic Scheduler, Shared I-Cache, Shared Execution Core, IA-32 Retirement & Exceptions.]

2 February 2005
## IA-64/ Itanium™ Features

- **Explicit Parallelism**: compiler / hardware synergy
- **Predication/Speculation**: predicates, parallel compares, speculative insts and checks
- **Register Model**: large register file, rotating registers, register stack engine
- **Floating Point Architecture**: extended precision, 128 regs FMA, SIMD
- **Multimedia Architecture**: parallel arithmetic, parallel shift, data arrangement instructions
- **Memory Management**: 64-bit addressing, speculation, memory hierarchy control
- **Compatibility**: full binary compatibility with existing IA-32 in hardware

## Function

- Enables compiler to “express” parallelism, hardware to “exploit” it
- Enhances ILP by overcoming traditional barriers (branches/ stores), hides memory latency
- Able to optimize for scalar and object oriented applications
- High performance 3D graphics and scientific analysis
- Improves calculation throughput for multimedia data
- Manages large amounts of memory, efficiently organizes data from / to memory
- Existing software runs seamlessly

## Benefits

- Maximizes headroom for the future
- Achieves higher performance where traditional architectures can’t
- World-class performance for complex applications
- Enables more complex scientific analysis & Faster DCC/rendering
- Efficient delivery of rich Web content
- Increased architecture & system scalability
- Preserves investment in existing software

---

**IA-64: Enabling new levels of performance**

[Intel Labs]

2 February 2005

(mtusion/ee5970/f02/btd)
• Itanium has been available in machines since 2001
• Low volumes sold due to performance less than Pentium 4 on x86 applications
• Next generation IA-64 (McKinley) ITANİUM/2 promises higher performance
• Can an implementation demonstrate the advantage of ISA enhancement promises?
The AMD Opteron™ processor 200 Series

Server Benchmark Performance

<table>
<thead>
<tr>
<th>SPECint®_rate</th>
<th>AMD Opteron 246</th>
<th>Itanium 2 1.5GHz</th>
<th>Xeon 3.06GHz 1MB L3</th>
<th>AMD Opteron 244</th>
<th>Xeon 3.06GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPECfp®_rate</td>
<td>AMD Opteron 246</td>
<td>Itanium 2 1.5GHz</td>
<td>Xeon 3.06GHz 1MB L3</td>
<td>AMD Opteron 244</td>
<td>Xeon 3.06GHz</td>
</tr>
<tr>
<td>SPECjbb2000</td>
<td>Itanium 2 1.5GHz</td>
<td>AMD Opteron 246</td>
<td>Xeon 3.06GHz 1MB L3</td>
<td>AMD Opteron 244</td>
<td>Xeon 3.06GHz</td>
</tr>
<tr>
<td>MMB2</td>
<td>AMD Opteron 244</td>
<td>Xeon 3.06GHz</td>
<td>Itanium 2</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

## The AMD Opteron™ processor 800 Series

### Server Benchmark Performance

**SPECint®_rate**

<table>
<thead>
<tr>
<th>Processor</th>
<th>SPECint®_rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itanium 2 1.5GHz</td>
<td>60.0</td>
</tr>
<tr>
<td>AMD Opteron 846**</td>
<td>56.6</td>
</tr>
<tr>
<td>AMD Opteron 844</td>
<td>48.5</td>
</tr>
<tr>
<td>Xeon MP 2.8GHz</td>
<td>47.4</td>
</tr>
<tr>
<td>Xeon MP 2.0GHz</td>
<td>34.7</td>
</tr>
</tbody>
</table>

**SPECfp®_rate**

<table>
<thead>
<tr>
<th>Processor</th>
<th>SPECfp®_rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itanium 2 1.5GHz</td>
<td>82.7</td>
</tr>
<tr>
<td>AMD Opteron 846**</td>
<td>51.3</td>
</tr>
<tr>
<td>AMD Opteron 844</td>
<td>49.2</td>
</tr>
<tr>
<td>Xeon MP 2.8GHz</td>
<td>25.4</td>
</tr>
<tr>
<td>Xeon MP 2.0GHz</td>
<td>20.2</td>
</tr>
</tbody>
</table>

**SPECjbb2000**

<table>
<thead>
<tr>
<th>Processor</th>
<th>SPECjbb2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itanium 2 1.5GHz</td>
<td>116466</td>
</tr>
<tr>
<td>AMD Opteron 846**</td>
<td>108900</td>
</tr>
<tr>
<td>AMD Opteron 844</td>
<td>94405</td>
</tr>
<tr>
<td>AMD Opteron 844</td>
<td>90737</td>
</tr>
</tbody>
</table>

**SPECweb®99_SSL**

<table>
<thead>
<tr>
<th>Processor</th>
<th>SPECweb®99_SSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itanium 2 1.5GHz</td>
<td>3702</td>
</tr>
<tr>
<td>AMD Opteron 846**</td>
<td>3498</td>
</tr>
<tr>
<td>AMD Opteron 844</td>
<td>2174</td>
</tr>
</tbody>
</table>

**MMB2**

<table>
<thead>
<tr>
<th>Processor</th>
<th>MMB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xeon MP 2.8GHz</td>
<td>16528</td>
</tr>
<tr>
<td>AMD Opteron 844</td>
<td>15520</td>
</tr>
</tbody>
</table>

**TPC-C**

<table>
<thead>
<tr>
<th>Processor</th>
<th>TPC-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itanium 2 1.5GHz</td>
<td>131,640</td>
</tr>
<tr>
<td>Xeon MP 2.8GHz</td>
<td>84,595</td>
</tr>
<tr>
<td>AMD Opteron 844</td>
<td>82,226</td>
</tr>
</tbody>
</table>

© 2003 Advanced Micro Devices. AMD, the AMD Arrow Logo, AMD Opteron and any combinations thereof are trademarks of Advanced Micro Devices. SPEC and the benchmark names SPECfp, SPECint and SPECweb are registered trademarks of the Standard Performance Evaluation Corp. Competitive numbers shown reflect results published on www.spec.org as of August 5, 2003. For the latest SPEC results visit http://www.spec.org. For full MMB2 results visit http://www.microsoft.com/exchange/techinfo/planning/2000/PerfScal.asp. TPC-C data is current as of 8/5/03 and includes previously published TPC results. TPC-C data obtained from publicly available information and is subject to change without notice. For more information visit www.tpc.org.

2 February 2005
Comments

• IA-64 is a unique ground-up ISA
  ■ Will x86 performance suffer due to IA-64 core?
  ■ Many unique ground-up ISAs have failed due to “installed code base”

• X86-64 is the next phase in a saga, with baggage from each phase
  ■ 8080—>8088—>8086—>’268—>386—>MMX: SSE: 3d!NOW—>586—>x86-64
  ■ Will x86-64 enable competitive performance without the parallelism enhancements in IA-64

• Other factors
  ■ Cost, Power Consumption, Market Arrival, etc
The latest offerings in the Pentium 4 family now support AMD's x86-64 architecture, even though Intel is not willing to admit it very openly, by using cryptic names like EM64T and (gasp) IA-32e. Intel's FAQ admits their implementation is basically compatible with x86-64, except for the minor differences that have always set Athlons and P4s apart. It's about time Intel jumped on AMD's bandwagon, since its homegrown 64-bit architecture seems not to be doing very well.
References

• IA - 64 ‘An Overview’
  • www-ist.massey.ac.nz/~crjessho/comp_arch/html/IA64.ppt

• Hot Chips 11 – 1999

Reading Assignments:

• "AMD: x86-64 Technology White Paper"

• Itanium Processor Microarchitecture