Barcelona:
AMD's Next-Generation
Quad-Core Microprocessor


Ben Sander
AMD Principal Member of Technical Staff
Introducing “Barcelona”…
Native quad-core upgrade for 2007

Native Quad-Core Processor
To increase performance-per-watt efficiencies using the same Thermal Design Power.

Advanced Process Technology
65nm Silicon-on Insulator Process
Fast transistors with low power leakage to reduce power and heat.

Platform Compatibility
Socket and thermal compatible with “Socket F”.

Direct Connect Architecture
- Integrated memory controller designed for reduced memory latency and increased performance
  - Memory directly connected
- Provides fast CPU-to-CPU communication
  - CPUs directly connected
- Glueless SMP up to 8 sockets
Agenda: A Closer Look at the Barcelona Processor

Comprehensive Upgrades for SSE128
Expandable shared L3 cache
IPC-enhanced CPU cores
Virtualization Performance
Advanced Power Management
More delivered DRAM Bandwidth
### SSE 128 – What’s in it for the Customer?

**Commercial and Consumer Benefits**

<table>
<thead>
<tr>
<th>High Performance Technical Computing</th>
<th>Media Encode and Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oil and Gas Sims</td>
<td>HD Video</td>
</tr>
<tr>
<td>EDA</td>
<td>Audio</td>
</tr>
<tr>
<td>Financial Analysis</td>
<td>DVD Authoring</td>
</tr>
<tr>
<td>BioScience</td>
<td></td>
</tr>
</tbody>
</table>

**GOAL:**
Double vector SSE performance  
Both SSE Floating-point and SSE Packed Integer  
*Avoid creating bottlenecks in instruction or data delivery*
### Comprehensive Upgrades for SSE128

**Current Generation versus Next Generation**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Current Processor</th>
<th>“Barcelona”</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSE Exec Width</td>
<td>64</td>
<td>128 + SSE MOVs</td>
</tr>
<tr>
<td>Instruction Fetch Bandwidth</td>
<td>16 bytes/cycle</td>
<td>32 bytes/cycle + Unaligned Ld-Ops</td>
</tr>
<tr>
<td>Data Cache Bandwidth</td>
<td>2 x 64bit loads/cycle</td>
<td>2 x 128bit loads/cycle</td>
</tr>
<tr>
<td>L2/NB Bandwidth</td>
<td>64 bits/cycle</td>
<td>128 bits/cycle</td>
</tr>
<tr>
<td>FP Scheduler Depth</td>
<td>36 Dedicated x 64-bit ops</td>
<td>36 Dedicated x 128-bit ops</td>
</tr>
</tbody>
</table>

- Can perform SSE MOVs in the FP “store” pipe
  - Execute two generic SSE ops + SSE MOV each cycle (+ two 128-bit SSE loads)
- SSE Unaligned Load-Execute mode
  - Remove alignment requirements for SSE Id-op instructions
  - Eliminate awkward pairs of separate load and compute instructions
  - *Improve instruction packing and decoding efficiency*
CPU Core IPC Enhancements

- Advanced branch prediction
- 32B instruction fetch
- Sideband Stack Optimizer
- Out-of-order load execution
- TLB Optimizations
- Data-dependent divide latency
- More Fastpath instructions
  - CALL and RET-Imm instructions
  - Data movement between FP & INT
- Bit Manipulation extensions
  - LZCNT/POPCNT
- SSE extensions
  - EXTRQ/INSERTQ,
  - MOVNTSD/MOVNTSS
CPU Core IPC Enhancements

- Advanced branch prediction
- 32B instruction fetch
- Sideband Stack Optimizer
- Out-of-order load execution
- TLB Optimizations
- Data-dependent divide latency
- More Fastpath instructions
  - CALL and RET-Imm instructions
  - Data movement between FP & INT
- Bit Manipulation extensions
  - LZCNT/POPCNT
- SSE extensions
  - EXTRQ/INSERTQ,
  - MOVNTSD/MOVNTSS

- Dedicated 512-entry Indirect Predictor
- Double return stack size
- More branch history bits and improved branch hashing
CPU Core IPC Enhancements

- Advanced branch prediction
- **32B instruction fetch**
- Sideband Stack Optimizer
- Out-of-order load execution
- TLB Optimizations
- Data-dependent divide latency
- More Fastpath instructions
  - CALL and RET-Imm instructions
  - Data movement between FP & INT
- Bit Manipulation extensions
  - LZCNT/POPCNT
- SSE extensions
  - EXTRQ/INSERTQ,
  - MOVNTSD/MOVNTSS

- Benefits integer code too
- Reduced split-fetch instruction cases
CPU Core IPC Enhancements

- Advanced branch prediction
- 32B instruction fetch
- **Sideband Stack Optimizer**
  - Out-of-order load execution
  - TLB Optimizations
  - Data-dependent divide latency
  - More Fastpath instructions
    - CALL and RET-Imm instructions
    - Data movement between FP & INT
  - Bit Manipulation extensions
    - LZCNT/POPCNT
  - SSE extensions
    - EXTRQ/INSERTQ,
    - MOVNTSD/MOVNTSS
- Perform stack adjustments for PUSH/POP operations “on the side”
- Stack adjustments don’t occupy functional unit bandwidth
- Breaks serial dependence chains for consecutive PUSH/POP operations
CPU Core IPC Enhancements

- Advanced branch prediction
- 32B instruction fetch
- Sideband Stack Optimizer
- **Out-of-order load execution**
- TLB Optimizations
- Data-dependent divide latency
- More Fastpath instructions
  - CALL and RET-Imm instructions
  - Data movement between FP & INT
- Bit Manipulation extensions
  - LZCNT/POPCNT
- SSE extensions
  - EXTRQ/INSERTQ,
  - MOVNTSD/MOVNTSS

► **New technology allows load instructions to bypass:**
  - Other loads
  - Other stores which are known not to alias with the load

► **Significantly mitigates L2 cache latency**
CPU Core IPC Enhancements

- Advanced branch prediction
- 32B instruction fetch
- Sideband Stack Optimizer
- Out-of-order load execution

**TLB Optimizations**
- Data-dependent divide latency
- More Fastpath instructions
  - CALL and RET-Imm instructions
  - Data movement between FP & INT
- Bit Manipulation extensions
  - LZCINT/POPCINT
- SSE extensions
  - EXTRQ/INSERTQ,
  - MOVNTSD/MOVNTSS

- **Support for 1G pages**
- **48bit physical address**
- **Larger TLBs key for:**
  - Virtualized workloads
  - Large-footprint databases and transaction processing

- **DTLB**
  - Fully-associative 48-way TLB (4K, 2M, 1G)
  - Backed by L2 TLBs:
    - 512 x 4K, 128 x 2M

- **ITLB**
  - 16 x 2M entries
CPU Core IPC Enhancements

- Advanced branch prediction
- 32B instruction fetch
- Sideband Stack Optimizer
- Out-of-order load execution
- TLB Optimizations
- Data-dependent divide latency
- More Fastpath instructions
  - CALL and RET-Imm instructions
  - Data movement between FP & INT
- Bit Manipulation extensions
  - LZCNT/POPCNT
- SSE extensions
  - EXTRQ/INSERTQ,
  - MOVNTSD/MOVNTSS
DRAM Basics

- Complex access protocol:
  - ACT to load row into sense amp
  - READ column from sense amp
  - PRECHARGE to reset sense amp

- Efficient Access Requires:
  - Access different banks
    - 4-8 banks/chip
    - 1-4 chips/channel
  - Column locality
Trends in DRAM bandwidth

Improved Efficiency is the Answer

Higher per-socket bandwidth demands

Diverse streams increase conflicts

DRAM efficiency declining

We must improve *delivered* DRAM bandwidth
Delivering more DRAM bandwidth

- Independent DRAM controllers
- Optimized DRAM paging
- Re-architect NB for higher BW
- Write bursting
- DRAM prefetcher
- Core prefetchers

- Concurrency
- More DRAM banks
  - reduces page conflicts
- Longer burst length
  - improves command efficiency
Delivering more DRAM bandwidth

- Independent DRAM controllers

**Optimized DRAM paging**

- Re-architect NB for higher BW
- Write bursting
- DRAM prefetcher
- Core prefetchers

增加页面命中，减少页面冲突

*历史基于模式预测器*
Delivering more DRAM bandwidth

- Independent DRAM controllers
- Optimized DRAM paging
- Re-architect NB for higher bw
  - Increase buffer sizes
  - Optimize schedulers
  - Ready to support future DRAM technologies
- Write bursting
- DRAM prefetcher
- Core prefetchers
Delivering more DRAM bandwidth

- Independent DRAM controllers
- Optimized DRAM paging
- Re-architect NB for higher BW

- Write bursting
- Minimize Rd/Wr Turnaround
- DRAM prefetcher
- Core prefetchers
Delivering more DRAM bandwidth

- Independent DRAM controllers
- Optimized DRAM paging
- Re-architect NB for higher BW
- Write bursting

**DRAM prefetcher**

- Track positive and negative, unit and non-unit strides
- Dedicated buffer for prefetched data
- Aggressively fill idle DRAM cycles

- Core prefetchers
Delivering more DRAM bandwidth

- Independent DRAM controllers
- Optimized DRAM paging
- Re-architect NB for higher BW
- Write bursting
- DRAM prefetcher

**Core prefetchers**

- **DC Prefetcher** fills directly to L1 Cache
- **IC Prefetcher** more flexible
  - 2 outstanding requests to any address
**Balanced, Highly Efficient Cache Structure**

**Dedicated L1**
- Locality keeps most critical data in the L1 cache
- Lowest latency
- 2 loads per cycle

**Dedicated L2**
- Sized to accommodate the majority of working sets today
- Dedicated to eliminate conflicts common in shared caches
  - Better for Virtualization

**Shared L3 – NEW**
- Victim-cache architecture maximizes efficiency of cache hierarchy
- Fills from L3 leave likely shared lines in the L3
- Sharing-aware replacement policy
- Ready for expansion at the right time for customers
Virtualization Background

• Why virtualize?
  – Huge cost savings through consolidation
  – Disaster recovery – can move OSs to a new server.
  – Expect 40% of x86 servers will be virtualized in next 3 years!
  – Consumer applications as well

• AMD-V status
  – Hardware virtualization support available now
  – Simplifies software investment for Hypervisor development
  – Improved security via AMD-V Device Exclusion Vector
  – Better performance from AMD-V tagged TLBs
Virtualized Address Translation

- Virtualization adds a new level of address translation in the Hypervisor
  - Guest physical to Host physical

- Current Technology: Shadow Paging
  - Software-only approach to virtualized address translation
  - Complex and slow
Faster Virtualization Performance

- **Nested Paging (NP)**
  - Guest and Host page tables both exist in memory
    - *The processor walks both guest and host page tables*
  - Nested walk can have up to 24 memory accesses!
    - *Hardware caching accelerates the walk*
  - “Wire-to-wire” translations are cached in TLBs
  - NP eliminates Hypervisor cycles spent managing shadow pages
    - *As much as 75% of Hypervisor time*

- **Barcelona also reduces world-switch time by 25%**
  - World-switch time: round-trip to the Hypervisor and back
Advanced Power Management

- Separate CPU core and Northbridge power planes
  - Allow processors to reduce voltage while NB continues to run
    *Power savings*
  - Also can apply additional voltage to NB to raise the NB frequency
    *Performance boost in power-constrained platforms*

- Enhanced PowerNow!
  - Ability to *dynamically* and *individually* adjust core frequencies for improved power efficiency

```
        100% Workload
        50% Workload
          Idle
          Idle

45% Power State
```
Quad-core System Power

2P System

- 190 watts for processors
- 16 watts for chipset
- 35.2 watts for DDR2

- Direct Connect Savings:
  - *No external memory controller* – saves 25 watts
  - *No FBDIMM* – saves 48 watts

- System power is the metric that matters to our customers.
- Direct Connect helps reduce system power.
Summary

• “Barcelona”:
  - Native quad-core upgrade planned for 2007

• Processor Details
  - Comprehensive upgrades for SSE128
    Upgrade Execution, Instruction, and Data Bandwidth
  - IPC-enhanced core
    Boost core performance
  - More delivered DRAM bandwidth
    Even from the same DRAM technology
  - L3 Cache Architecture
    Shared and expandable
  - Virtualization performance
    Nested paging and faster world-switches
  - Advanced power management
    Optimize system power
Questions and Thank you!