The new AMD 6200 series CPU and its relevance to HPC

Binesh Majithia
Technical Pre-Sales
binesh.majithia@amd.com

Lee-Martin King
UK&I Business Development
lee-martin.king@amd.com
AMD in High-Performance Computing

CPUs and GPUs Drive Efficient Performance

<table>
<thead>
<tr>
<th>TOP500.ORG list</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPUs</strong></td>
</tr>
<tr>
<td>Oak Ridge &quot;Jaguar&quot;</td>
</tr>
<tr>
<td>DOE/NNSA/LANL/SNL – “Cielo”</td>
</tr>
<tr>
<td>The National Energy Research Scientific Computing Center (NERSC) – “Hopper”</td>
</tr>
<tr>
<td><strong>GPUs</strong></td>
</tr>
<tr>
<td>Nat’l Supercomputer Center, Tianjan &quot;Tianhe-1&quot;</td>
</tr>
</tbody>
</table>

Since 2005, AMD technology has powered more than 70 top-25 entries in the list of the world’s most powerful supercomputers

Source: www.top500.org
AMD Server Platform Strategy

4P/8P Platforms
~5% of Market*

2P Platforms
~75% of Market*

1P Platforms
~20% of Market*

Performance-per-watt and Expandability

Performance-per-watt and Expandability

Highly Energy Efficient and Cost Optimized

Low cost for dedicated web hosting

Today

2011

2012

AMD Opteron™ 6000 Series Platform

• 2/4 socket; 4 memory channels
• For high core density

AMD Opteron™ 4000 Series Platform

• 1/2 socket; 2 memory channels
• For low power per core

AMD Opteron™ 3000 Series Platform

• 1 socket; 2 memory channels
• For low cost per core

“Hydra”

“Bulldozer”

SR5600 Series Chipsets

Future Core

Future Product

AMD Opteron 6100 Series processor
8 and 12 cores

AMD Opteron 6200 Series processor
4, 8, 12 and 16 cores

AMD Opteron 4100 Series processor
4 and 6 cores

AMD Opteron 4200 Series processor
6 and 8 cores

AMD Opteron 3200 Series processor
4 and 8 cores

Future Product

Future Product

Future Product

Future Product

*AMD internal estimates of total server market as of Q3 2011
Driving HPC Performance Efficiency

Bulldozer Module - Advanced Performance/Watt
Leadership Multi-Threaded Micro-Architecture

Full Performance From Each Core
- Dedicated execution units per core
- No shared execution units as with SMT

High Frequency / Low-Power Design
- Core Performance Boost
  - "Boosts" frequency of cores when available power allows
- Power efficiency enhancements
  - Deeper core sleep states

Virtualization Enhancements
- Faster switching between VMs
- AMD-V extended migration support

Shared Double-sized FPU
- Amortizes very powerful 256-bit unit across both cores

Improved IPC
- Micro-architecture and ISA enhancements
  SSE4.1/4.2, AVX 1.0/1.1, SSSE3

Enhanced Systems Management
- Greater power management control via APML
Customer Requirements:
- Scalable performance
- Strong floating point performance
- High memory throughput
- More cores for highly threaded apps
- Wide range of technical instructions

**HPC**

- Linux OS
- Open64
- GCC
- PGI Compilers

---

**Superior Performance**

- 73GB/s memory throughput
- 73% more memory bandwidth than Intel
- Maximum cores per rack
- More FLOPs per sq. foot
- 33% lower cost per core

**Greatest FLOPs per Sq. Foot**

With almost twice the FLOPs per sq. ft. with AMD Opteron™ 6276 Series processors, it would take 2 racks of Intel Xeon 5670 racks to match AMD in density and performance.

---

1-3 See complete benchmark data on slides 35-37.
**Flex FP: More flexible technical processing**

*More performance and new instruction support*

- **Runs SSE and AVX simultaneously**
  - No processing penalty for simultaneous execution

- **Executes two SSE or AVX (128-bit) instructions simultaneously or one AVX (256-bit) instruction per Bulldozer module**
  - Wider range of FP processing capabilities than competition

- **Processes calculations in a single cycle using FMA4* and XOP instructions**
  - Executes more instructions in fewer cycles than competition

- **Uses dedicated floating point scheduler**
  - No waiting on integer scheduler to run instructions
  - Designed to be always available to schedule floating point operations

*FMAC can execute an FMA4 execution \((a=b+c*d)\) in one cycle vs. 2 cycles that would be required for FMA3 or standard SSE floating point calculation.
### ASSUMES latest updates/patches are installed*

<table>
<thead>
<tr>
<th>Enabled</th>
<th>Compatible</th>
<th>Not Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optimized to support some or all of “Bulldozer’s” new features</strong></td>
<td><strong>Will boot and run but not take advantage of “Bulldozer’s” new features outside of new instructions</strong></td>
<td><strong>Will not run on “Bulldozer” platforms and/or will not be supported by OSV</strong></td>
</tr>
<tr>
<td><strong>Includes new instruction support:</strong></td>
<td><strong>Includes new instruction support:</strong></td>
<td><strong>Does not support new instructions for either Bulldozer or Sandy Bridge:</strong></td>
</tr>
<tr>
<td>• Hyper-V Nex Gen (in development)</td>
<td>• Linux kernel 2.6.32 – 2.6.36</td>
<td>• Hyper-V R1</td>
</tr>
<tr>
<td>• Linux kernel 2.6.37 +</td>
<td>• Novell SLES 11 SP1</td>
<td>• Hyper-V R2, Hyper-V R2 SP1</td>
</tr>
<tr>
<td>• Novell SLES 11 SP2 Beta (includes Xen)</td>
<td>• RHEL 6.1</td>
<td>• Novell SLES 10 SP4 and higher</td>
</tr>
<tr>
<td>• RHEL 6.2 with KVM (in development)</td>
<td>• Ubuntu 10.10</td>
<td>• RHEL 5.7 (included KVM)</td>
</tr>
<tr>
<td>• Windows Server 2008 R2 SP1</td>
<td></td>
<td>• Solaris 10 – 10u8</td>
</tr>
<tr>
<td>• Windows 8 Server (in development)</td>
<td></td>
<td>• VMware ESX 3.5</td>
</tr>
<tr>
<td>• Xen 4.1</td>
<td></td>
<td>• VMware ESX 4.0 – 4.1u1</td>
</tr>
<tr>
<td>• Ubuntu 11.04 (w/ KVM)</td>
<td><strong>Will run but not necessarily provide performance uplift</strong></td>
<td>• Windows Server 2003 versions prior to R2 SP2</td>
</tr>
<tr>
<td>• VMware vSphere 5.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Versions in this category also include latest software advances

---

* Please note: For proper support of available features/processors, the latest updates/patches always needs to be installed
<table>
<thead>
<tr>
<th>Compiler</th>
<th>Status</th>
<th>SSSE3</th>
<th>SSE4.1-2</th>
<th>FMA4</th>
<th>XOP</th>
<th>Auto Generates Code</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCC 4.6.1</td>
<td>Available</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>GCC 4.4 is included in RHEL 6.0 distribution and should be updated to GCC 4.6.1 for optimized support</td>
</tr>
<tr>
<td>Microsoft Visual Studio 2010 SP1</td>
<td>Available</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>No</td>
<td>Supports new instructions but does not auto generate code</td>
</tr>
<tr>
<td>Open64 4.2.5</td>
<td>Available</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td><a href="http://developer.amd.com/open64">http://developer.amd.com/open64</a></td>
</tr>
<tr>
<td>Open64 4.5</td>
<td>Planned for Dec 2011</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>Will provide incremental performance and functionality improvements</td>
</tr>
<tr>
<td>PGI 11.9</td>
<td>Available</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>PGI Unified Binary™ technology combines into a single executable or object file code optimized for multiple AMD and Intel processors</td>
</tr>
<tr>
<td>ICC 12</td>
<td>Available</td>
<td>✓</td>
<td>(−mAVX flag)</td>
<td>No</td>
<td>✓</td>
<td></td>
<td>−mAVX is designed to run on any x86 processor, however the ICC runtime makes assumptions about cache line sizes and other parameters that causes code to fail on AMD processors</td>
</tr>
</tbody>
</table>
### THE NEW “BULLDOZER” INSTRUCTIONS

#### A CLOSER LOOK

<table>
<thead>
<tr>
<th>Instructions</th>
<th>Applications/Use Cases</th>
</tr>
</thead>
</table>
| **SSSE3, SSE4.1, SSE4.2 (AMD and Intel)** | • Video encoding and transcoding  
                          • Biometrics algorithms  
                          • Text-intensive applications |
| **AESNI PCLMULQDQ (AMD and Intel)** | • Application using AES encryption  
                          • Secure network transactions  
                          • Disk encryption (MSFT BitLocker)  
                          • Database encryption  
                          • Cloud security |
| **AVX (AMD and Intel)** | Floating point intensive applications:  
                          • Signal processing / Seismic  
                          • Multimedia  
                          • Scientific simulations  
                          • Financial analytics  
                          • 3D modeling |
| **FMA4 (AMD Unique)** | • Vector and matrix multiplications  
                          • Polynomial evaluations  
                          • Chemistry, physics, quantum mechanics and digital signal processing |
| **XOP (AMD Unique)**  | • Numeric applications  
                          • Multimedia applications  
                          • Algorithms used for audio/radio |

XOP and FMA4 instruction set extensions are AMD unique 128-bit and 256-bit instructions designed to:
- Improve performance by increasing the work per instruction
- Reduce the need to copy and move around register operands
- Allow for some new cases of automatic vectorization by compilers

---

For more details: AMD64 Architecture Programmer’s Manual Volume 6: 128-Bit and 256-Bit XOP and FMA4 Instructions

NEW “BULLDOZER” INSTRUCTIONS
USAGE RECOMMENDATIONS

Software using SSE instructions should be recompiled with AVX 128 and FMA4 compiler options (see Compiler Optimization Guide*) & linked to ACML 5.x libraries.

If software currently supports the new instructions that are common with Intel (SSSE3, SSE4.1/4.2, AES-NI, AVX):

- No recompile of code needed if the software only checks ISA feature bits
- Recompile needed if software also checks for CPU VENDOR (For Example: ICC generates code that checks for CPU VENDOR)

For software to support FMA4 or XOP (AMD-specific instructions):

- Rewritten to call new instructions
- OR-
- Recompiled with options to automatically generate code that uses these instructions
- OR-
- Linked to a library that offers support for these instructions

WHY DOES AMD SUPPORT A RANGE OF COMPILERS?

No one compiler services all of our target markets

<table>
<thead>
<tr>
<th>Compilers</th>
<th>Languages Supported</th>
<th>Processors Supported</th>
<th>Operation Systems Supported</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCC</td>
<td>C, C++, Fortran, Objective-C, Java, Ada, Go</td>
<td>Wide variety including: x86, AIX, SPARC, ARM</td>
<td>Wide variety including: Linux, Windows, Mac OS, Android, Solaris</td>
<td>Default compiler for Linux</td>
</tr>
<tr>
<td>Intel</td>
<td>C, C++, Fortran</td>
<td>Intel x86, Itanium</td>
<td>Linux, Windows, Mac OS</td>
<td>Performance compiler for Intel</td>
</tr>
<tr>
<td>Open64</td>
<td>C, C++, Fortran</td>
<td>AMD and Intel x86</td>
<td>Linux</td>
<td>Performance compiler for AMD</td>
</tr>
<tr>
<td>PGI</td>
<td>C, C++, Fortran</td>
<td>AMD, Intel x86, NVIDIA CUDA</td>
<td>Linux, Mac OS, Windows</td>
<td>Performance compiler for HPC</td>
</tr>
<tr>
<td>MSFT Visual Studio</td>
<td>C, C++, C#, Basic</td>
<td>AMD and Intel x86</td>
<td>Windows</td>
<td>Default compiler for Windows</td>
</tr>
</tbody>
</table>

- Default compilers are used to compile the kernel, some of the system software, and libraries for the OS
- Customers are often reluctant to change compilers
- Compilers used to generate high performance code are not necessarily the ones used for mainstream server applications
OPEN64 COMPILER | A CLOSER LOOK

Setting the “–march” (microarchitecture) flag will automatically optimize code for the target processor’s instruction set

<table>
<thead>
<tr>
<th>Open64 Settings</th>
<th>Processor Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>-march=bdver1</td>
<td>AMD Opteron™ 4200 and 6200 Series</td>
</tr>
<tr>
<td>-march=barcelona</td>
<td>AMD Opteron™ 13xx, 14xx, 23xx, 24xx, 83xx, 84xx, 4100, and 6200 Series</td>
</tr>
<tr>
<td>-march=any86</td>
<td>Any x86 processor</td>
</tr>
</tbody>
</table>

“Bulldozer” compiler optimizations enabled by –march=bdver1*

- Support for all new instructions (SSSE3, SSE4.1, SSE4.2, AVX, FMA, and XOP)
- Automatically selects instructions to improve performance (intrinsics and inline)
- Automatic calls to libM (math library) functions that use these new instructions
- Code generation tuned for microarchitecture, e.g. instruction latencies, cache sizes
- Adjusted to take advantage of the improved hardware prefetcher
- Improvements in code layout and alignment to take advantage of shared compute unit, e.g. “dispatch scheduling”

* Additional information: http://developer.amd.com/tools/open64/Documents/open64.html
### GCC COMPILER | A CLOSER LOOK

Setting the “–march” (microarchitecture) flag will automatically optimize code for the target processor’s instruction set

<table>
<thead>
<tr>
<th>Open64 Settings</th>
<th>Processor Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>-march=bdver1</td>
<td>AMD Opteron™ 4200 and 6200 Series</td>
</tr>
<tr>
<td>-march=amdfam10</td>
<td>AMD Opteron™ 13xx, 14xx, 23xx, 24xx, 83xx, 84xx, 4100, and 6200 Series</td>
</tr>
<tr>
<td>-march=generic</td>
<td>Any x86 processor</td>
</tr>
</tbody>
</table>

**“Bulldozer” compiler optimizations enabled by –march=bdver1**

- Support for all new instructions (SSSE3, SSE4.1, SSE4.2, AVX, FMA, and XOP)
- Automatically selects instructions to improve performance (intrinsics and inline)
- Scalar and vector libm calls available with AMD Libm
- Code generation tuned for microarchitecture, e.g. instruction latencies, cache sizes
- Memset/Memcpy inliner heuristics
- Defaults to 128-bit vectorization
- Improvements in code layout and alignment

“AMD OPTERON™ 4200 AND 6200 SERIES PROCESSORS LIBRARY SUPPORT

A library is a collection of pre-written code and subroutines

<table>
<thead>
<tr>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACML (AMD Core Math Library)</strong></td>
<td>Set of optimized and threaded math routines for HPC, scientific, engineering and related compute-intensive applications</td>
</tr>
<tr>
<td><strong>AMD LibM</strong></td>
<td>C library containing a collection of basic math functions optimized for x86-64 processors</td>
</tr>
<tr>
<td><strong>IPP (Intel Performance Primitives)</strong></td>
<td>Library of multicore-ready, optimized software functions for multimedia, data processing, and communications applications</td>
</tr>
</tbody>
</table>

ACML (AMD CORE MATH LIBRARY) | A CLOSER LOOK

- A full implementation of Level 1, 2 and 3 Basic Linear Algebra Subroutines (BLAS), with key routines optimized for high performance on AMD Opteron™ processors.
- A full suite of Linear Algebra (LAPACK) routines. As well as taking advantage of the highly-tuned BLAS kernels, a key set of LAPACK routines has been further optimized to achieve considerably higher performance than standard LAPACK implementations.
- A comprehensive suite of Fast Fourier Transforms (FFTs) in both single-, double-, single-complex and double-complex data types.
- Random Number Generators in both single- and double-precision.

Compiler Support
- Absoft Pro Fortran
- GFORTRAN
- Intel Fortran (Linux, Windows)
- NAG Fortran
- Open64
- PGI Fortran (Linux, Windows)

For more information on ACML, go to: http://developer.amd.com/libraries/acml/pages/default.aspx
### ACML SUPPORT | A CLOSER LOOK

<table>
<thead>
<tr>
<th></th>
<th>Linear Algebra</th>
<th>Fast Fourier Transforms (FFT)</th>
<th>Others</th>
<th>Compiler Support</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACML 5.0</strong></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td><strong>(Aug 2011)</strong></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>• SGEMM (single precision)</td>
<td>• Complex-to-Complex (C-C)</td>
<td>• Random Number Generators</td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>• DGEMM (double precision)</td>
<td>single precision FFTs</td>
<td>• AVX compiler switch for Fortran</td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>• L1 BLAS</td>
<td></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td><strong>ACML 5.1</strong></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td><strong>(Dec 2011)</strong></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>• CGEMM (complex single decision)</td>
<td>• Real-to-complex (R-C)</td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>• ZGEMM (complex double precision)</td>
<td>single precision FFTs</td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Double precision C-C and R-C FFTs</td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
</tbody>
</table>

For additional information on ACML, go to: [http://developer.amd.com/libraries/acml/pages/default.aspx](http://developer.amd.com/libraries/acml/pages/default.aspx)
| STARTING POINTS FOR APPLICATION OPTIMIZATION |

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Compiler</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended for SPECCPU, LINPACK, HPC Challenge</td>
<td>Novell SLES 11 SP1 or RHEL 6.1</td>
<td>Open64 4.2.5</td>
</tr>
<tr>
<td>Recommended for application development and benchmarks with gcc</td>
<td>Novell SLES 11 SP1 or RHEL 6.1</td>
<td>GCC 4.6</td>
</tr>
<tr>
<td>Recommended for HPC application code development</td>
<td>Novell SLES 11 SP1 or RHEL 6.1</td>
<td>Open64 4.25 or PGI 11.9</td>
</tr>
<tr>
<td>Recommend for integer code development for Windows</td>
<td>Windows Server 2008 RS SP1</td>
<td>Microsoft Visual Studio 2010 SP1</td>
</tr>
</tbody>
</table>

Recommendations are based on AMD evaluations, please evaluate for your specific workload.
Three Eras of Processor Performance

**Single-Core Era**
- **Enabled by:**
  - ✓ Moore’s Law
  - ✓ Voltage Scaling
  - ✓ Micro-Architecture
- **Constrained by:**
  - ✗ Power
  - ✗ Complexity

**Multi-Core Era**
- **Enabled by:**
  - ✓ Moore’s Law
  - ✓ Desire for Throughput
  - ✓ 20 years of SMP arch
- **Constrained by:**
  - ✗ Power
  - ✗ Parallel SW availability
  - ✗ Scalability

**Heterogeneous Systems Era**
- **Enabled by:**
  - ✓ Moore’s Law
  - ✓ Abundant data parallelism
  - ✓ Power efficient GPUs
- **Temporarily constrained by:**
  - ✗ Programming models
  - ✗ Communication overheads
GPU Compute Offload – 3 Phases

**Proprietary Drivers Era**
- “Early Adopter” programmers
- Exploit early programmable “shader cores” in the GPU
- Make your program look like “graphics” to the GPU
- CUDA, Brook+, etc

**Industry Standard Drivers Era**
- Expert programmers
- Good APIs for compute
- “C and C++ like”
- Multiple address spaces & explicit data movement

**Architected Era**
- Mainstream programmers
- GPU is a first class member of the platform architecture
- Full C++ support
- Single unified & coherent address space

- **OpenCL™ /DirectCompute Driver-based APIs**

**Fusion APUs and Features**
- Proprietary Drivers Era
- OpenCL™/DirectCompute Driver-based APIs

**Architecture Maturity & Programmer Accessibility**
- Poor
- Excellent

**Timeline**
- 2002 - 2008
- 2009 - 2011
- 2012 - 2020
A New Era of Processor Performance

Microprocessor Advancement

Single-Core Era

Multi-Core Era

Heterogeneous Systems Era

Heterogeneous Computing

System-level programmable

OpenCL™/DirectX® driver-based programs

Graphics driver-based programs

Homogeneous Computing

Throughput Performance

GPU

Microprocessor Advancement

Single-Core Era

Multi-Core Era

Heterogeneous Systems Era

Heterogeneous Computing

System-level programmable

OpenCL™/DirectX® driver-based programs

Graphics driver-based programs

Homogeneous Computing

Throughput Performance

GPU
AMD Fusion APUs Fill the Need

x86 CPU owns the Software World
- Windows®, MacOS and Linux® franchises
- Thousands of apps
- Established programming and memory model
- Mature tool chain
- Extensive backward compatibility for applications and OSs
- High barrier to entry

GPU Optimized for Modern Workloads
- Enormous parallel computing capacity
- Outstanding performance-per-watt-per-dollar
- Very efficient hardware threading
- SIMD architecture well matched to modern workloads: video, audio, graphics
AMD Fusion: Enabling Heterogeneous Computing in A Broader Set of Applications

Discrete architectures can deliver performance acceleration…

**CPUs**
- Single-threaded performance
- Efficient flow control

**Add GPUs**
- Parallel Data performance
- High performance per watt

But…
- Costly data movement means complex programming (optimal kernel sizes, hand-tuning, etc.)

**AMD Fusion**

All of the CPU and GPU advantages, plus:

+ No data movement bottleneck
+ No limits on memory size
+ Easier to program; broader application availability
  - Scientific and engineering
  - Sorting/searching
  - Real time Audio/video
  - Face/object recognition
  - Image processing
  - Physics and AI
DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREOF, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This presentation contains forward-looking statements concerning AMD and technology partner product offerings which are made pursuant to the safe harbor provisions of the Private Securities Litigation Reform Act of 1995. Forward-looking statements are commonly identified by words such as "would," "may," "expects," "believes," "plans," "intends," "strategy," "roadmaps," "projects" and other terms with similar meaning. Investors are cautioned that the forward-looking statements in this presentation are based on current beliefs, assumptions and expectations, speak only as of the date of this presentation and involve risks and uncertainties that could cause actual results to differ materially from current expectations.

ATTRIBUTION

© 2010 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, ATI, the ATI logo, Radeon and combinations thereof are trademarks of Advanced Micro Devices, Inc. Microsoft, Windows, and Windows Vista are registered trademarks of Microsoft Corporation in the United States and/or other jurisdictions. OpenCL is trademark of Apple Inc. used under license to the Khronos Group Inc. Other names are for informational purposes only and may be trademarks of their respective owners.