



Larrabee: A Many-Core x86 Architecture for Visual Computing

**Doug Carmean
Larrabee Chief Architect**

Hot Chips 20

August 26, 2008



Agenda

Architecture Convergence

Larrabee Architecture

Graphics Pipeline

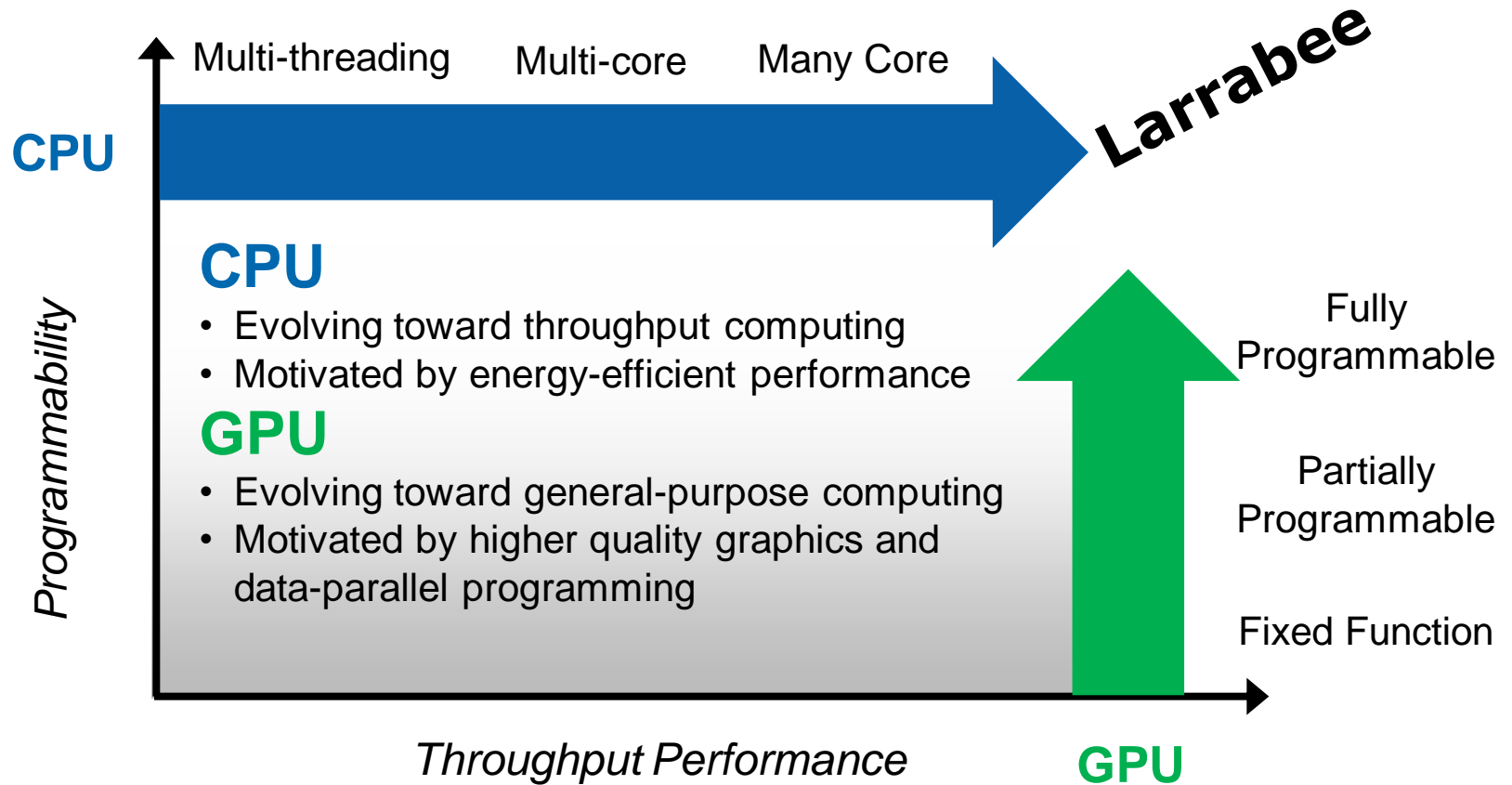
Agenda

Architecture Convergence

Larrabee Architecture

Graphics Pipeline

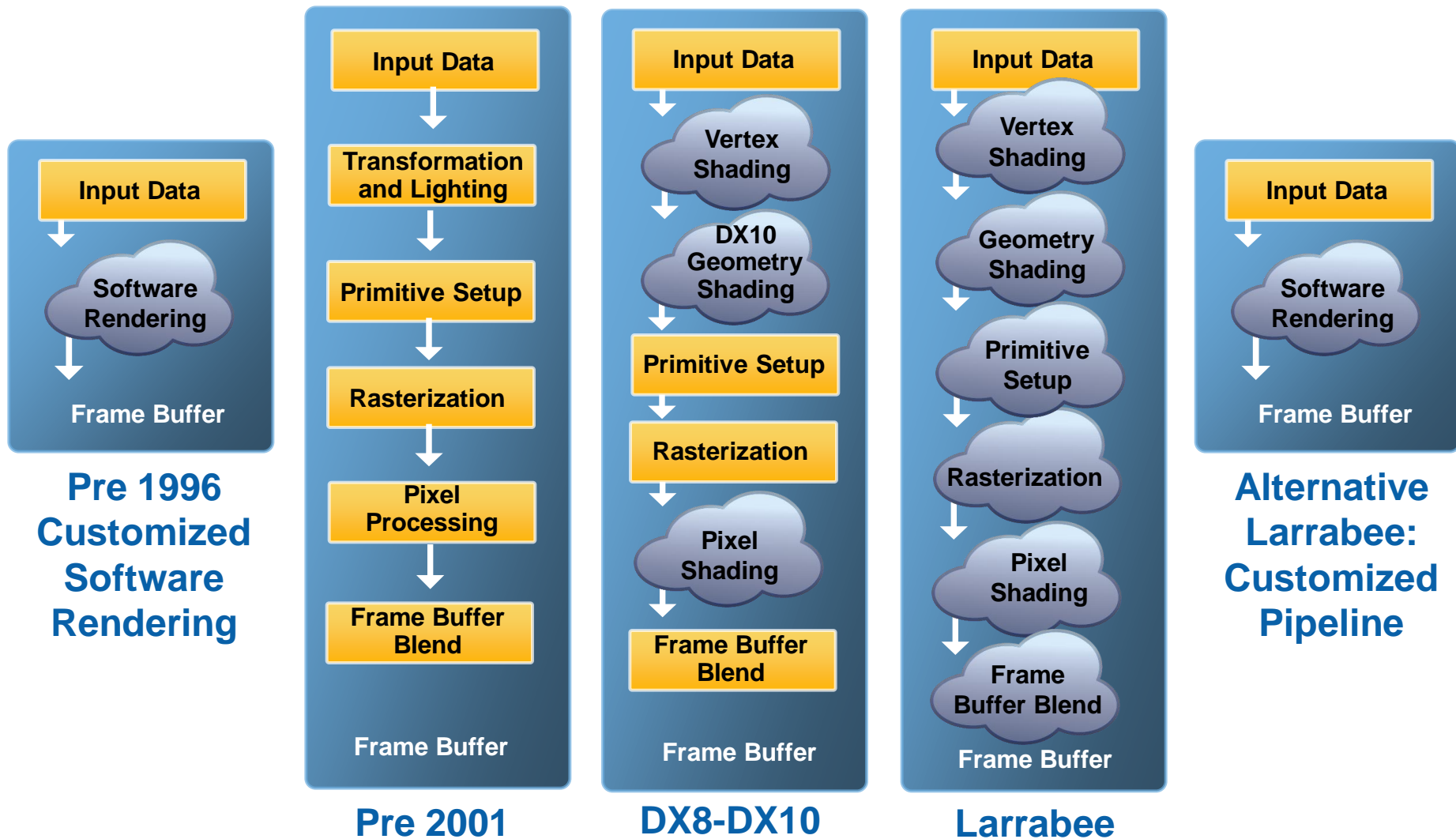
Architecture Convergence



Larrabee: CPU programmability with GPU throughput



Graphics Rendering Pipelines



Larrabee \Lar*a*bee"\, n. [from Intel]

1: a general, programmer-friendly, architecture combining data level and thread level parallelism optimized for high performance throughput applications

2: a high performance visual computing device

Agenda

Architecture Convergence

Larrabee Architecture

Graphics Pipeline

Cores for Throughput Tasks

Design experiment: not a real 10-core chip!

CPU design experiment:
*specify a throughput-
optimized processor with
same area and power of a
standard dual core CPU.*

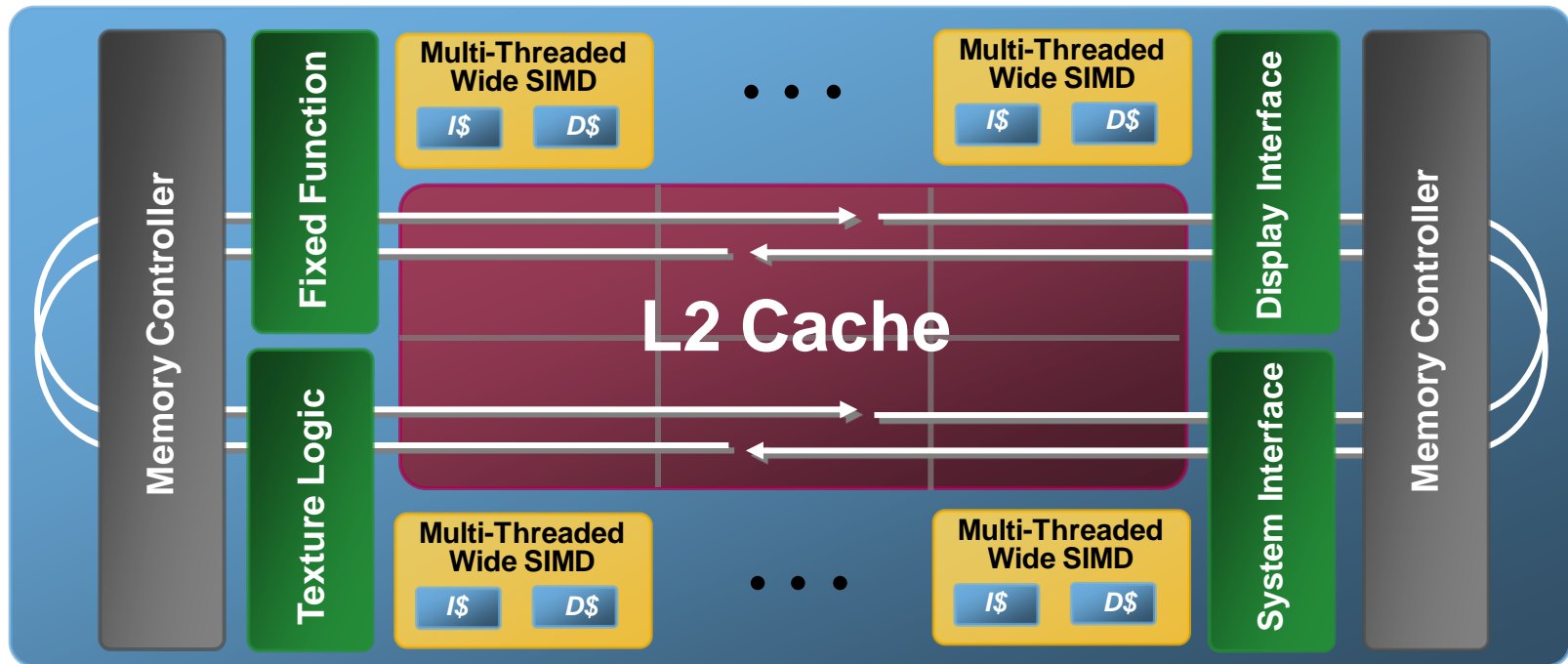
# CPU cores	2 out of order	10 in-order
Instructions per issue	8 per clock	2 per clock
VPU lanes per core	4-wide SSE	16-wide
L2 cache size	4 MB	4 MB
Single-stream	4 per clock	2 per clock
Vector throughput	16 per clock	160 per clock

10 times the peak throughput per clock

**Peak vector throughput for given power and area.
Ideal for graphics & other throughput applications.**

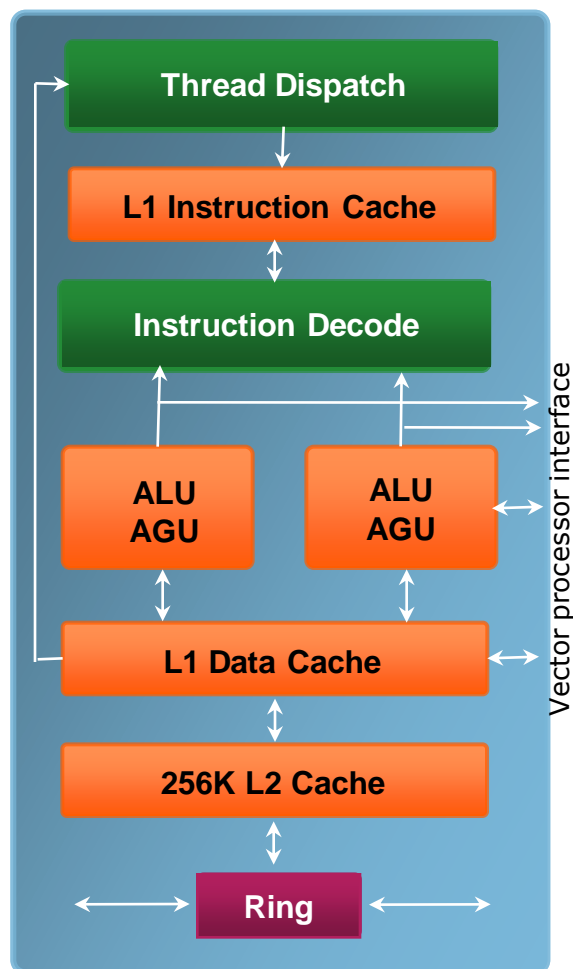


Larrabee Block Diagram



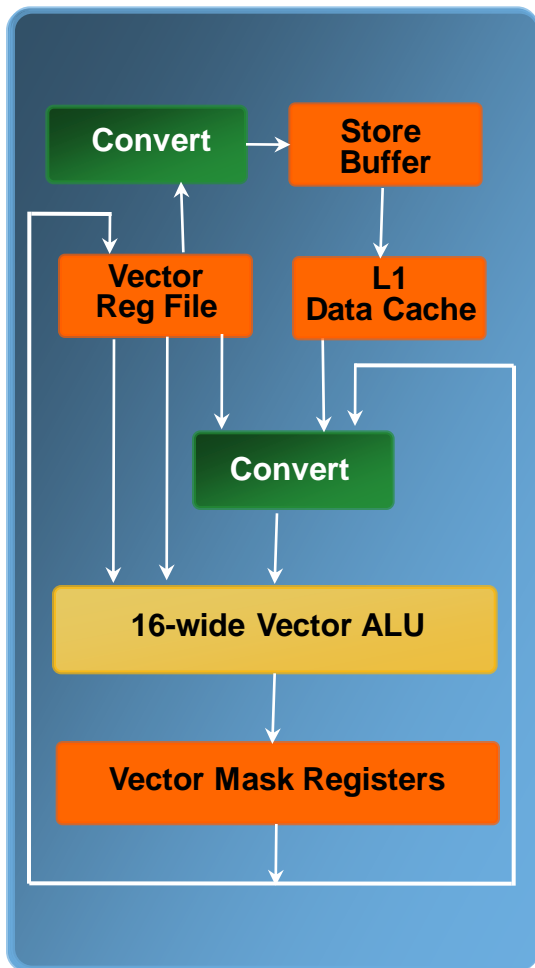
- IA Cores communicate on a wide ring bus
 - Fast access to memory and fixed function blocks
 - Fast access for cache coherency
- L2 cache is partitioned among the cores
 - Provides high aggregate bandwidth
 - Allows data replication & sharing

Larrabee IA Core



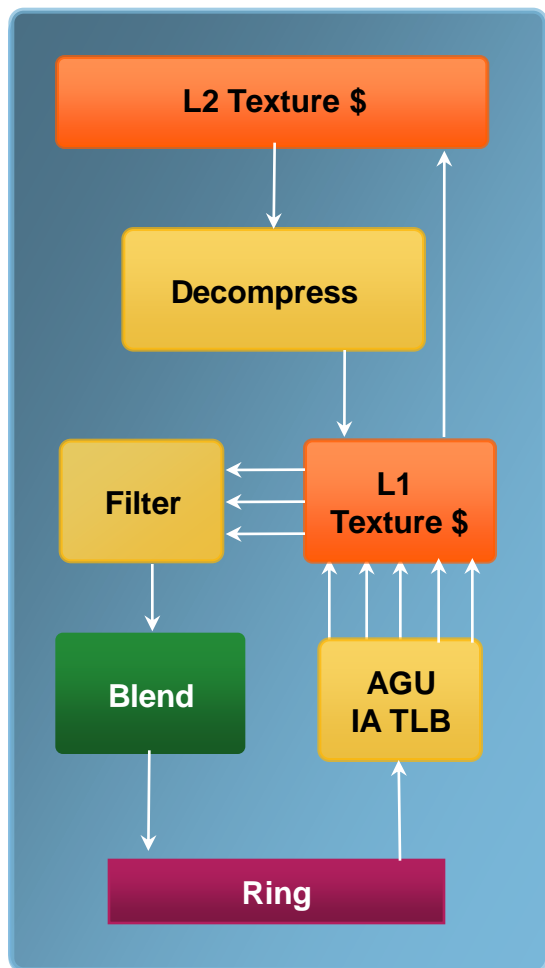
- Separate scalar and vector units with separate registers
- In-order IA scalar core
- Vector unit: 16 32-bit ops/clock
- Short execution pipelines
- Fast access from L1 cache
- Direct connection to subset of the L2 cache
- Prefetch instructions load L1 and L2 caches

Vector Processing Unit



- Vector complete instruction set
 - Scatter/gather for vector load/store
 - Mask registers select lanes to write, which allows data-parallel flow control
 - Can map a separate execution kernel to each vector lane
- Vector instructions support
 - Fast, wide read from L1 cache
 - Numeric type conversion and data replication while reading from memory
 - Rearrange the lanes on register read
 - Fused multiply add (three arguments)
 - Int32, Float32 and Float64 data

Texture Sampler



- Fixed function texture sampler
 - Typical texture operations, including decompression, anisotropic filtering
 - Core communication via the L2 cache
 - Supports virtual address translation using IA page formats
- Fixed function vs. software
 - Texture filtering needs specialized data access to unaligned 2x2 blocks of pixels
 - Filtering is optimized for 8-bit color
 - Code would take 12x longer for filtering or 40x longer if texture decompression is required

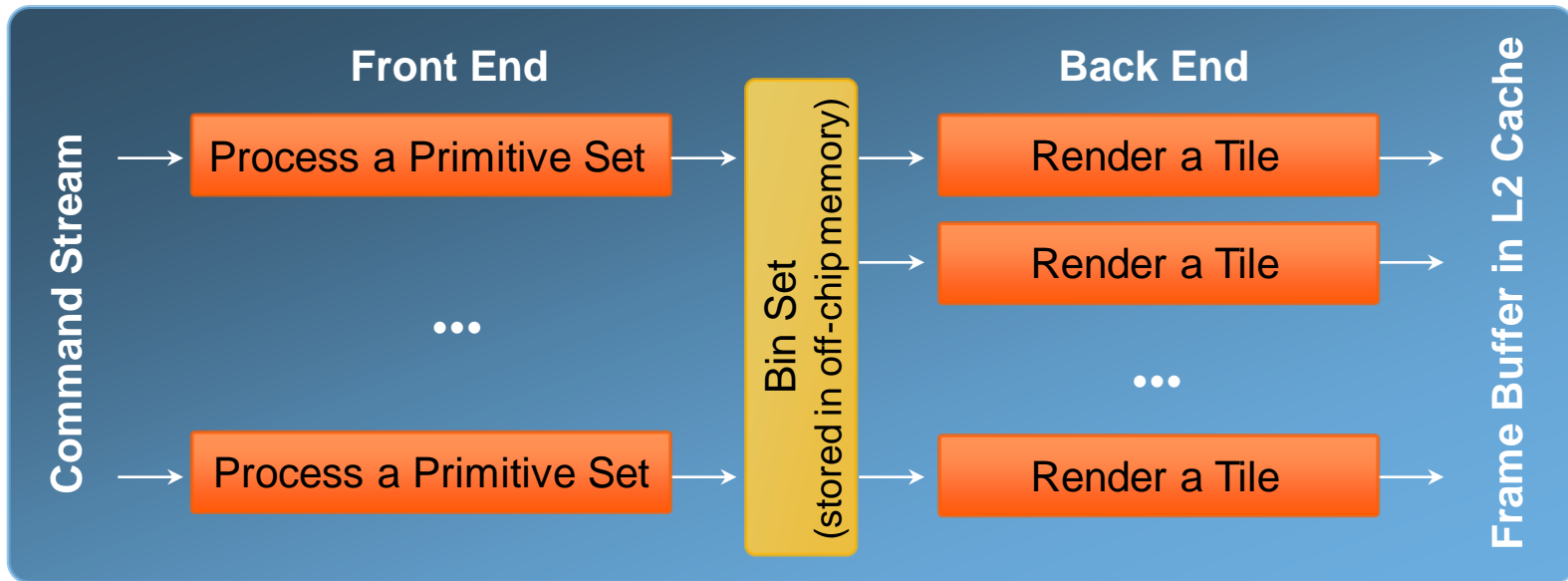
Agenda

Architecture Convergence

Larrabee Architecture

Graphics Pipeline

Larrabee's Binning Renderer



- Goals of a Binning Renderer:

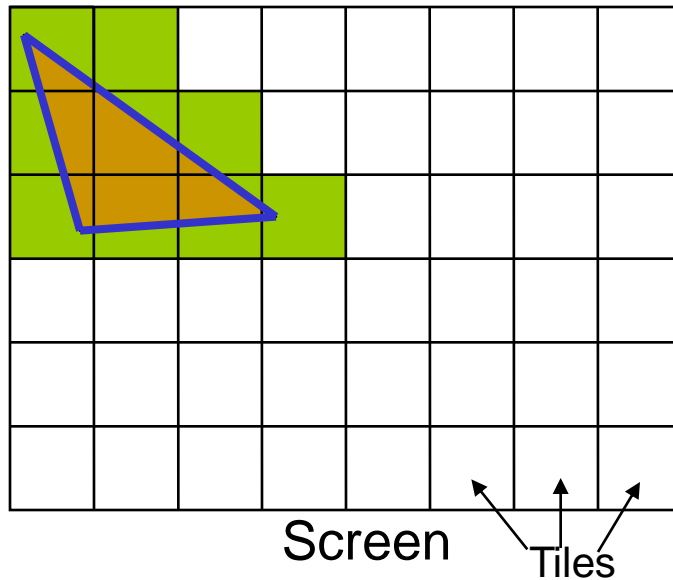
1. Parallelism

- a) Provide independent work queues for cores/threads
- b) Significantly reduce synchronization points

2. Bandwidth Efficiency

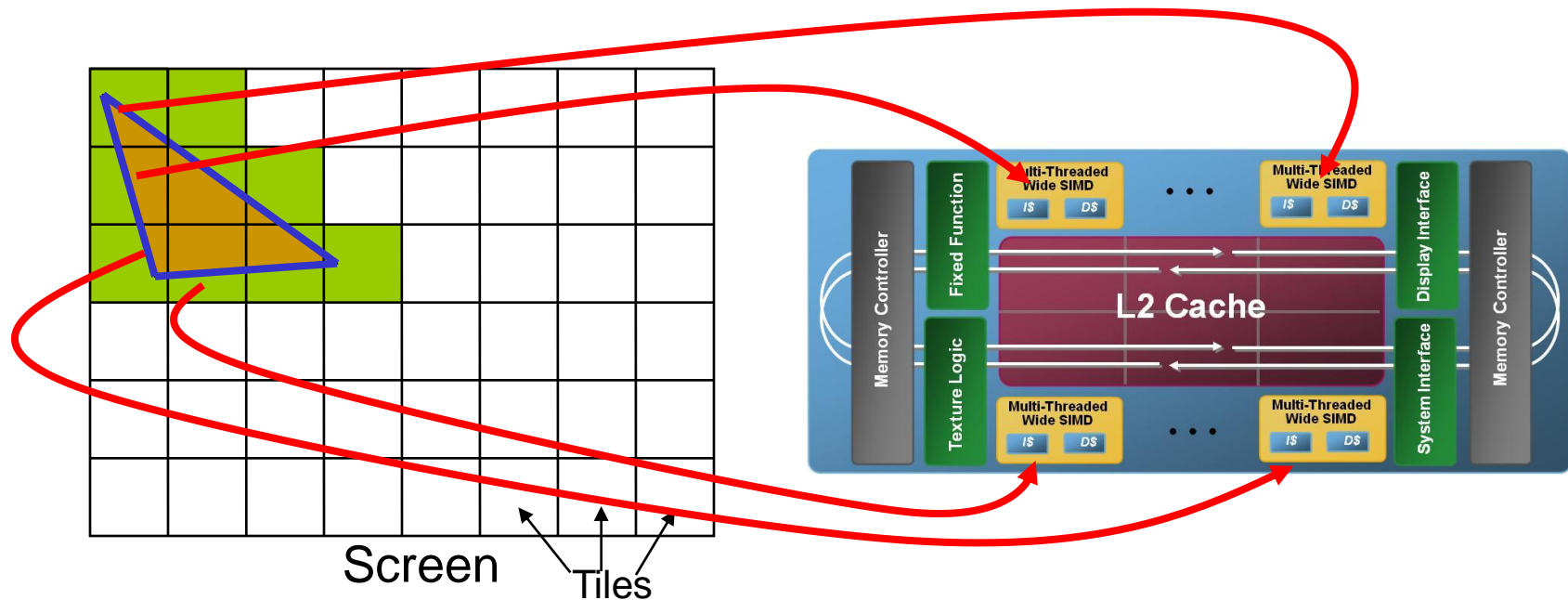
- a) Utilize on-die caches
- b) Reduce off-die bandwidth pressure

Larrabee's Binning Renderer



Screen is tiled and tiles processed concurrently
Software buffers (pixel, depth) also tiled
Tremendous bandwidth to tiled buffers

Larrabee's Binning Renderer



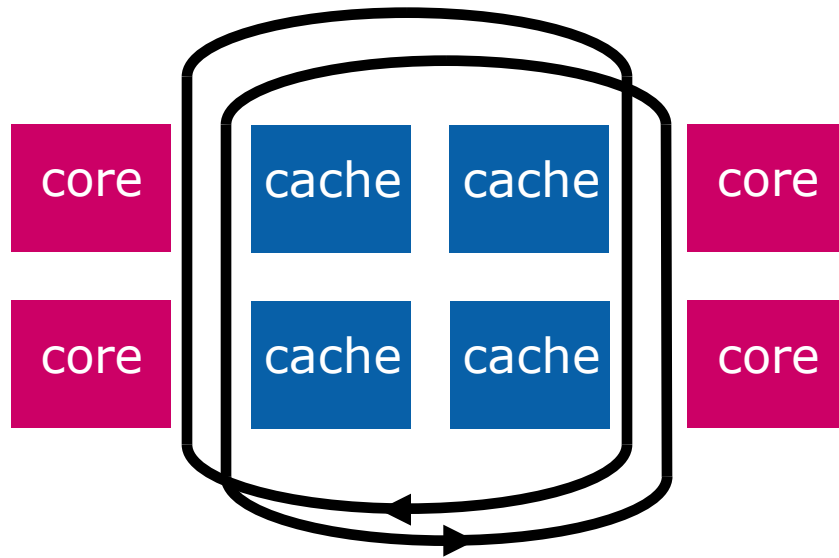
Screen is tiled and tiles processed concurrently
Software buffers (pixel, depth) also tiled
Tremendous bandwidth to tiled buffers

Agenda

Bandwidth Implications

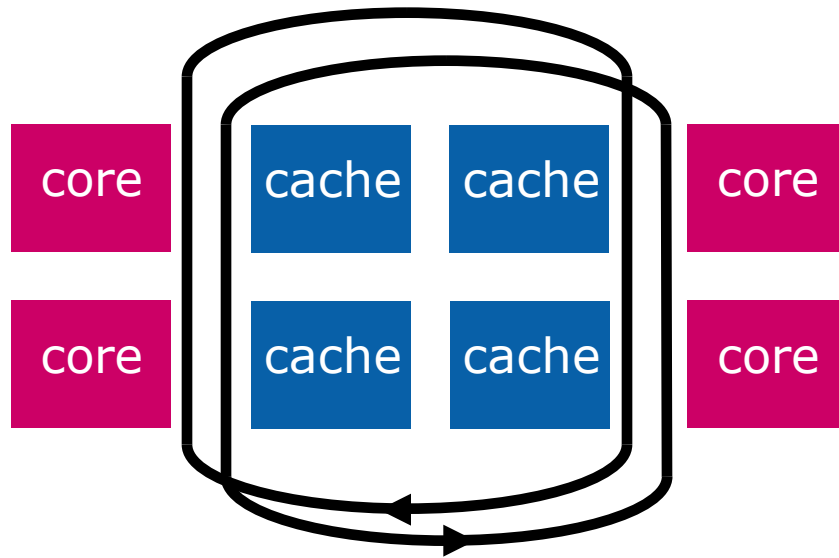


Server Style Interconnect



Bi-direction ring: physically sits over cache
Banked, logically unified ring cache
Straightforward, elegant architecture
Tremendous aggregate ring bandwidth

Server Style Interconnect: 3D Graphics



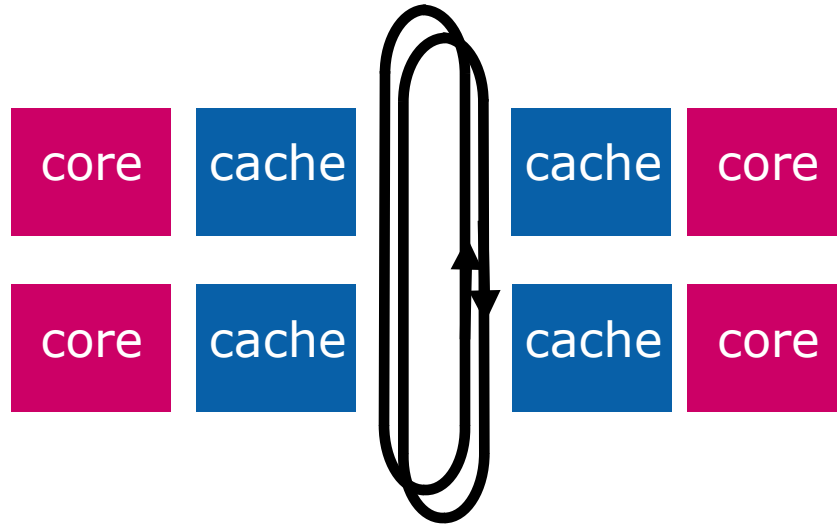
3D graphic tile buffers reside in ring cache

Tiles don't fit in cores L1\$

Tremendous BW to tiled buffers

Tile buffer BW requirement => heavy ring traffic

Larrabee Ring



Access ring on private L2\$ miss

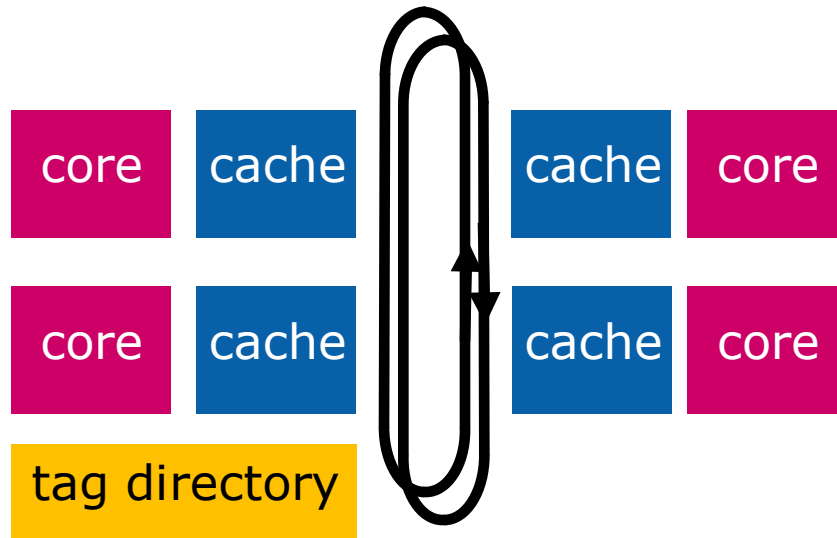
Tile buffer accesses do not travel on ring

However, memory coherence implies

L2\$ miss requires snooping of all other L2\$

RFO transaction needs to traverse ring

Larrabee Ring with Central Tag Directory



Central tag directory (td) reduces coherence traffic

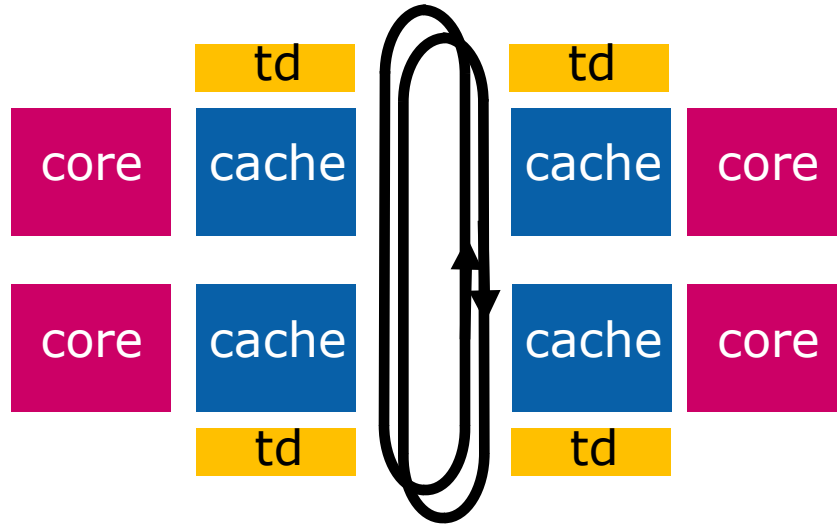
- Copy of all L2\$ tags

- Tracks MESI state

However, hot spot at the tag directory

- Tag directory BW doesn't scale with cores

Larrabee Ring: Distributed Tag Directory



Distributed tag directory (td)

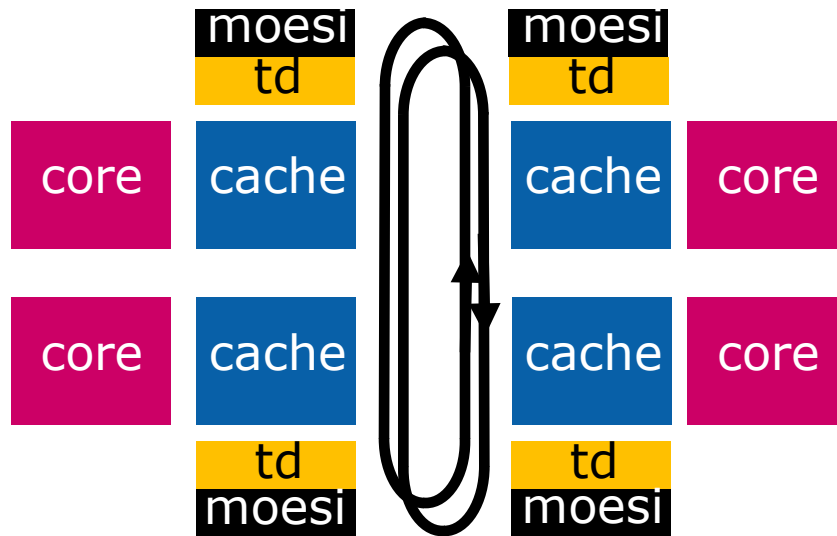
Address hash to determine specific tag directory

Solves hot spot

However, cache-cache transfer require global ordering

Introduces serialization

Larrabee Ring: Enhanced Coherency

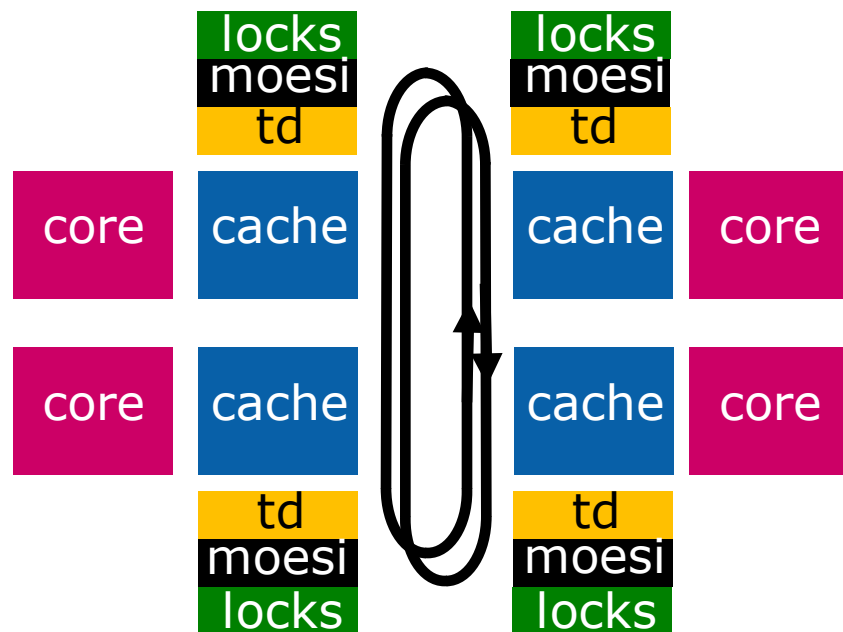


Change from MESI to MOESI

Cache-cache ownership transfer on die

However, contended locks still hop around ring

Larrabee Ring: Future Enhancement



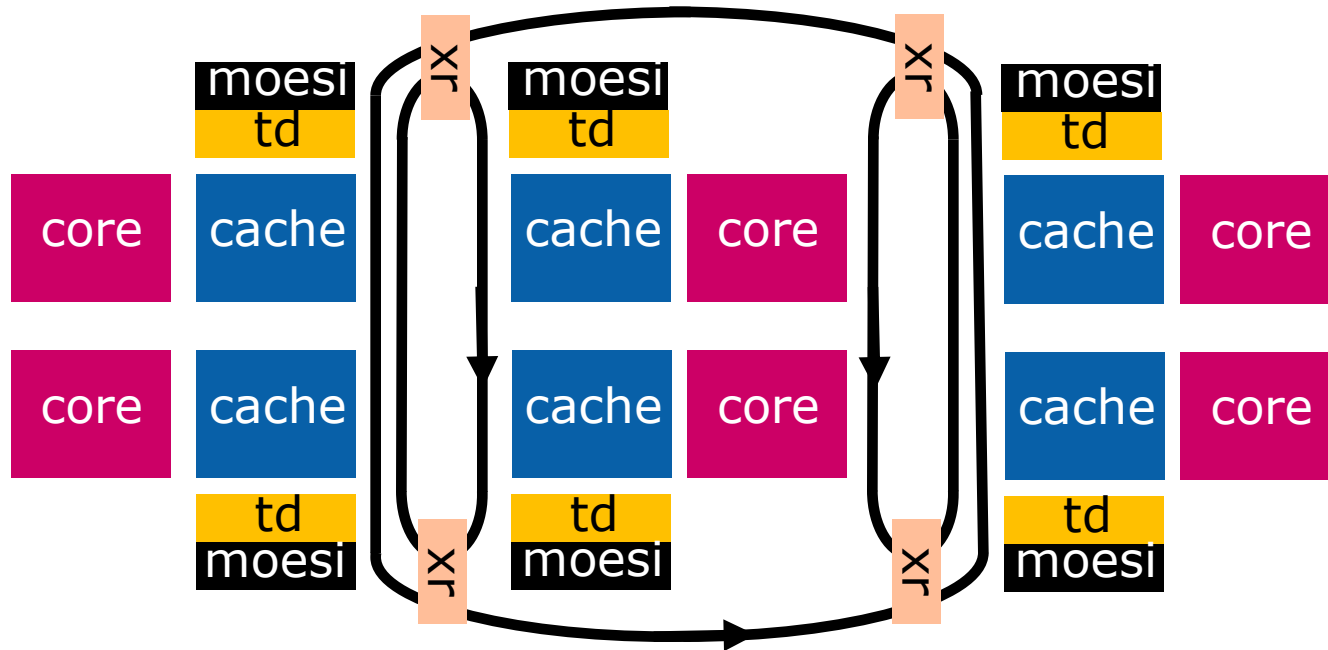
Lock cache holds lines containing contended locks

Locked inc/dec/cmp use ALU in lock cache

Big speedup (>10x) on contended locks

Contended lock throughput @ 1 per clock

Larrabee Ring: Scalability

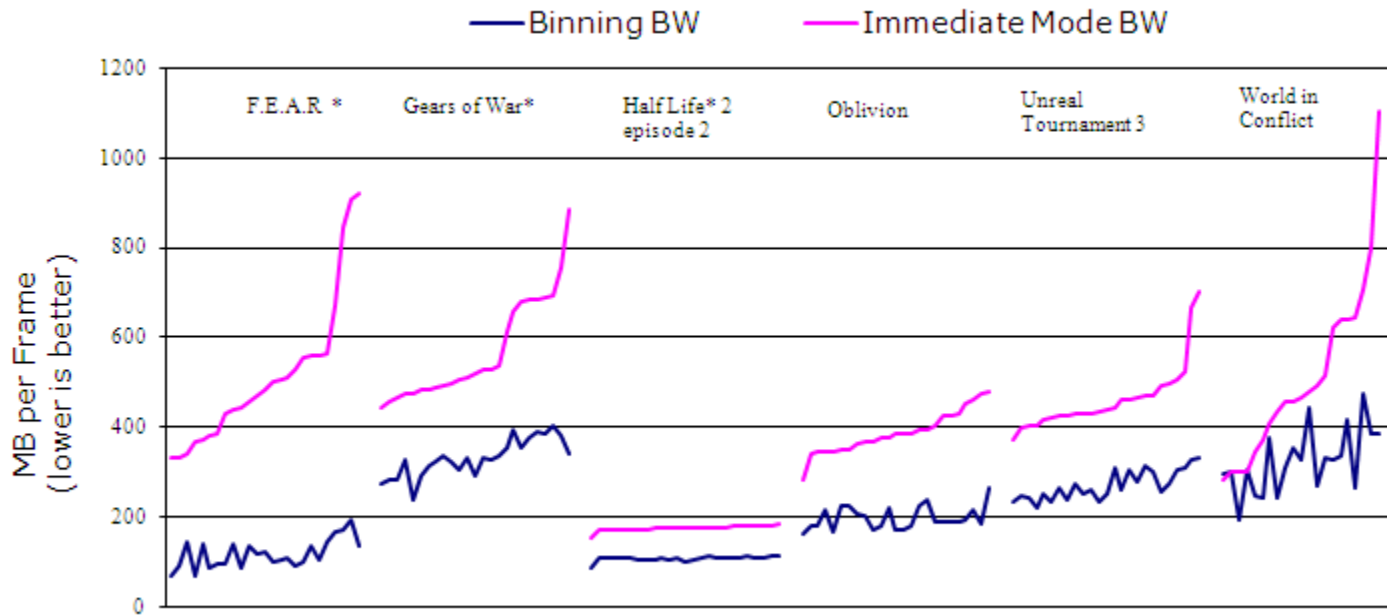


Large core count configurations need higher bandwidth

Required flexibility for physical topology

Xring: 3 separate rings with efficient ring crossing

Bandwidth Results



- Binning mode

- Includes bin reads & writes
- Reads/writes each pixel once due to tiles in the L2 cache

- Immediate mode

- Assumes perfect HeirZ cull
- Assumes 1MB each for compressed depth, color and stencil buffers



Summary

- Each Larrabee core is a complete IA core
 - Context switching & pre-emptive multi-tasking
 - Virtual memory and page swapping
 - Fully coherent caches at all levels of the hierarchy
- Efficient inter-block communication
 - Ring bus for full inter-processor communication
 - Low latency high bandwidth L1 and L2 caches
 - Fast synchronization between cores and caches
- Elegant use of fixed function logic
 - No backend blender between cores and memory
 - No rasterization logic between vertex and pixel stages
 - Result: flexible load balancing & general functionality

Larrabee: a general throughput computing architecture



Questions?



Legal Disclaimer

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.
- Intel may make changes to specifications and product descriptions at any time, without notice.
- All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.
- Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Larrabee and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user
- Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.
- Intel, Intel Inside and the Intel logo are trademarks of Intel Corporation in the United States and other countries.
- *Other names and brands may be claimed as the property of others.
- Copyright © 2008 Intel Corporation.



Risk Factors

This presentation contains forward-looking statements that involve a number of risks and uncertainties. These statements do not reflect the potential impact of any mergers, acquisitions, divestitures, investments or other similar transactions that may be completed in the future. The information presented is accurate only as of today's date and will not be updated. In addition to any factors discussed in the presentation, the important factors that could cause actual results to differ materially include the following: Demand could be different from Intel's expectations due to factors including changes in business and economic conditions, including conditions in the credit market that could affect consumer confidence; customer acceptance of Intel's and competitors' products; changes in customer order patterns, including order cancellations; and changes in the level of inventory at customers. Intel's results could be affected by the timing of closing of acquisitions and divestitures. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Revenue and the gross margin percentage are affected by the timing of new Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel's response to such actions; Intel's ability to respond quickly to technological developments and to incorporate new features into its products; and the availability of sufficient supply of components from suppliers to meet demand. The gross margin percentage could vary significantly from expectations based on changes in revenue levels; product mix and pricing; capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; excess or obsolete inventory; manufacturing yields; changes in unit costs; impairments of long-lived assets, including manufacturing, assembly/test and intangible assets; and the timing and execution of the manufacturing ramp and associated costs, including start-up costs. Expenses, particularly certain marketing and compensation expenses, vary depending on the level of demand for Intel's products, the level of revenue and profits, and impairments of long-lived assets. Intel is in the midst of a structure and efficiency program that is resulting in several actions that could have an impact on expected expense levels and gross margin. Intel's results could be impacted by adverse economic, social, political and physical/infrastructure conditions in the countries in which Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. A detailed discussion of these and other factors that could affect Intel's results is included in Intel's SEC filings, including the report on Form 10-Q for the quarter ended June 28, 2008.

