



*Accelerating with Many-cores &
Special Purpose Hardware*

Ajay Bhatt

Intel Fellow

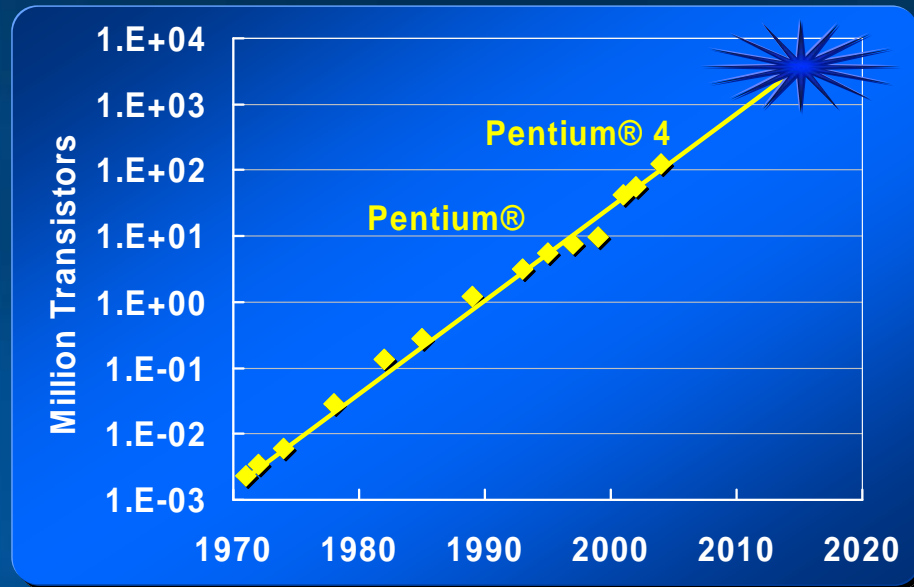
Digital Enterprise Group

August 27, 2007

Agenda

- **Motivation**
- **Accelerating with Multi core CPU**
- **Accelerating the Platform**
- **Summary**

Historical Driving Forces



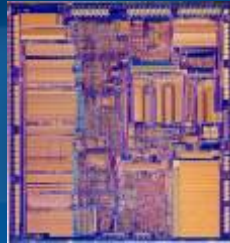
1971

4004 Processor
2300 Transistors



1978

8008 Processor
IBM PC



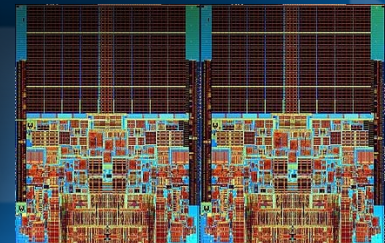
1985

i386 Processor
32-bit



1993

Pentium Processor
3.1M transistors



2006

561M Transistors

1993

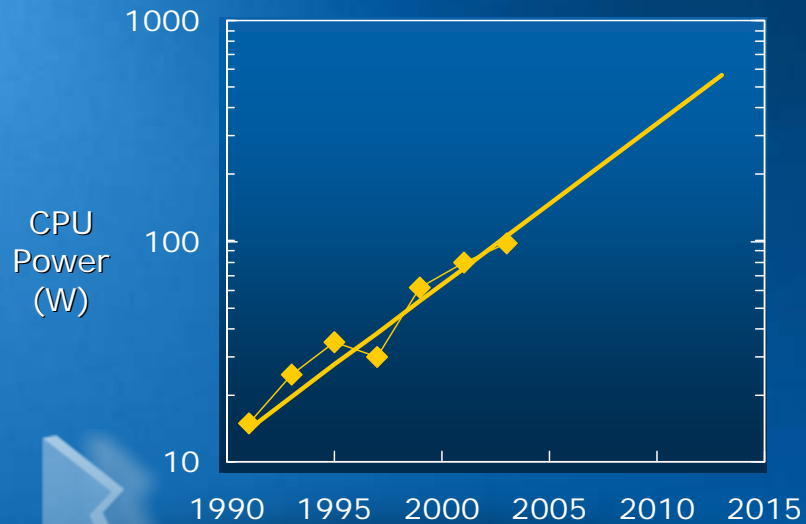
2003



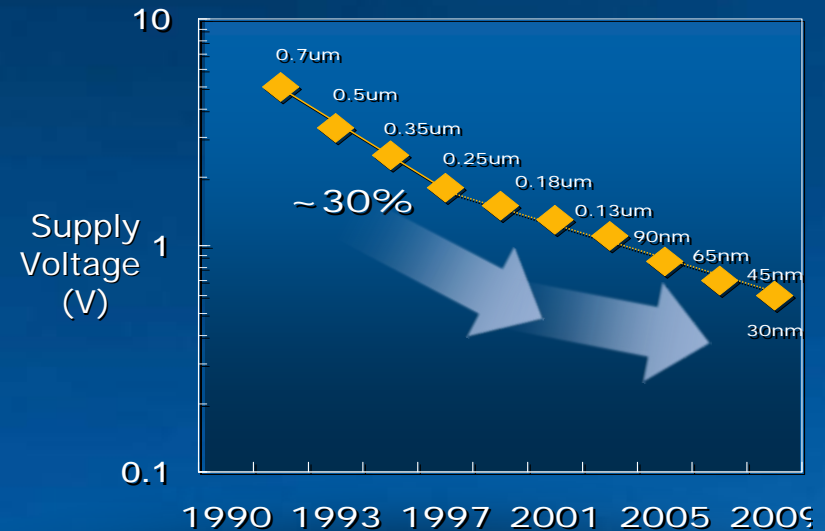
The GHz Era

Power: A Major Concern

Power Limitations

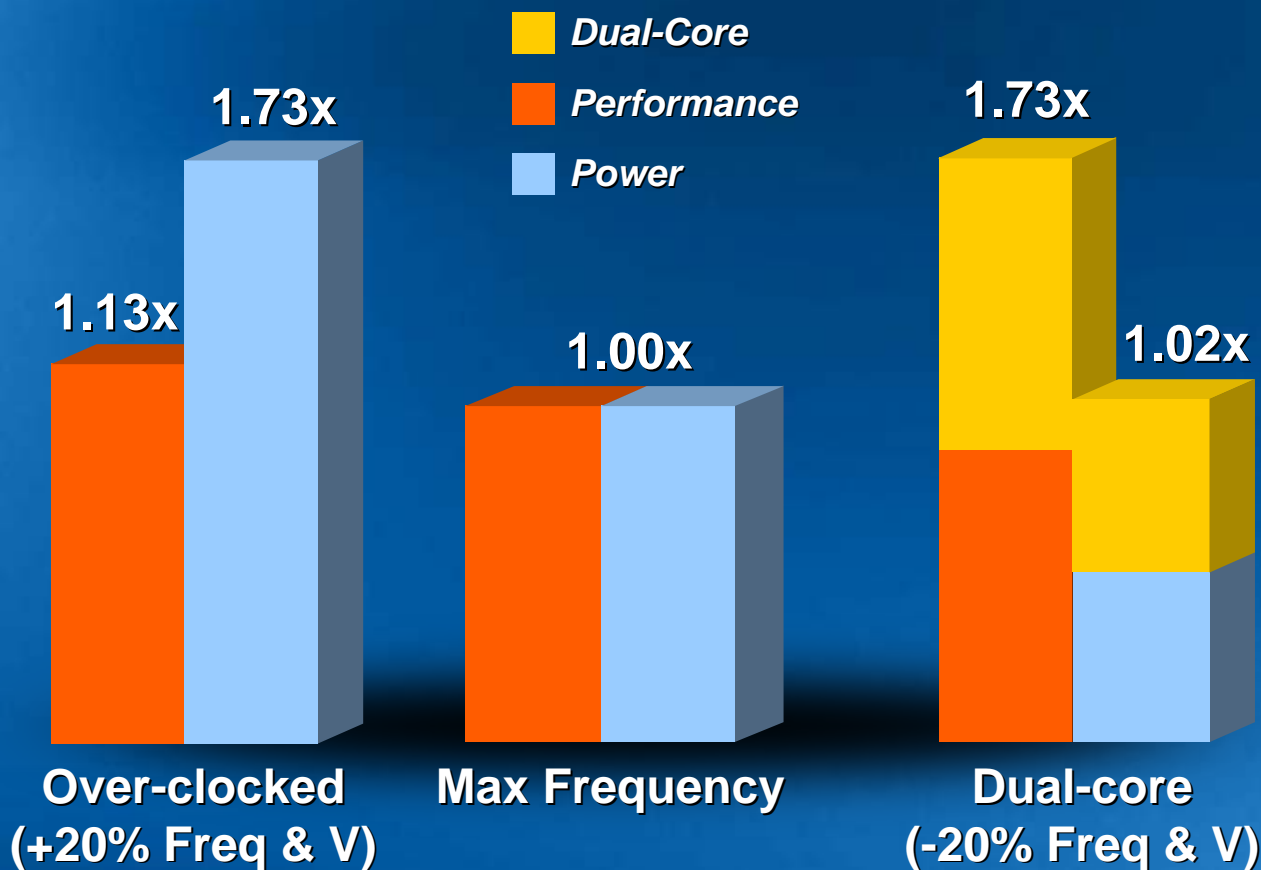


Diminishing Voltage Scaling



Power = Capacitance x Voltage² x Frequency
also
Power ~ Voltage³

Multi-Core Energy-Efficient Performance



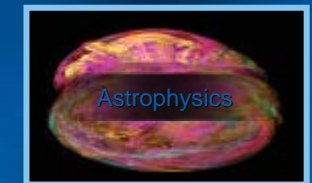
Relative single-core frequency and Vcc

SRC: Intel Internal Measurements, date of data and system configurations



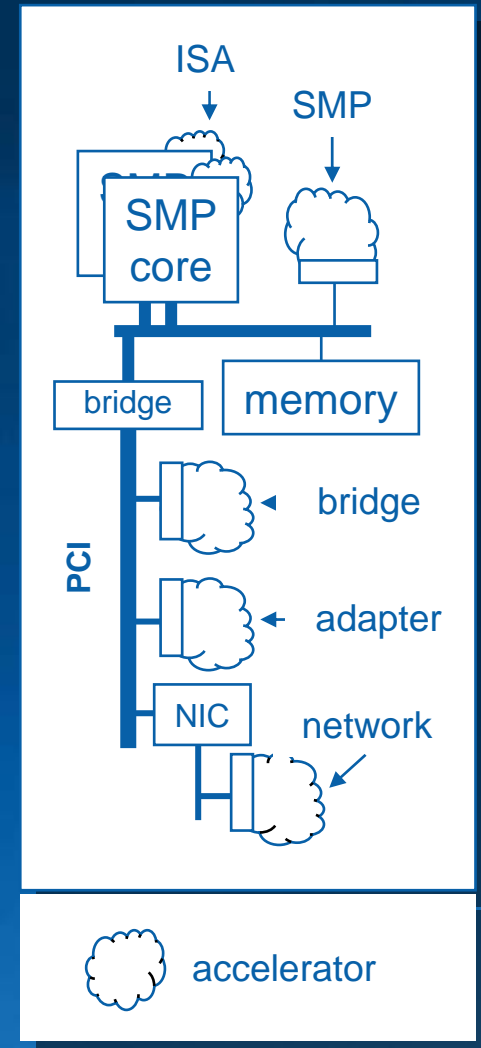
Moving beyond Productivity Workloads

- Rigid body game physics
- Fluid simulation
- Portfolio management
- Text mining
- Signal / image processing primitives
- Derivative pricing suite
- Stochastic optimization suite
- Partitioning structure collision tests
- Dense and Sparse matrix primitives



Accelerators in the Platform

- Accelerator: device optimized to enhance the performance or functionality of a computing system.
- Vertical solution: a performs a complete customer function without a additional server or client system. Example: game console.
- Appliance: a vertical solution with substantially reduced management & programming requirements.



Source : IBM and Intel

**Accelerators = performance, programmability
& precision at lower power and cost**

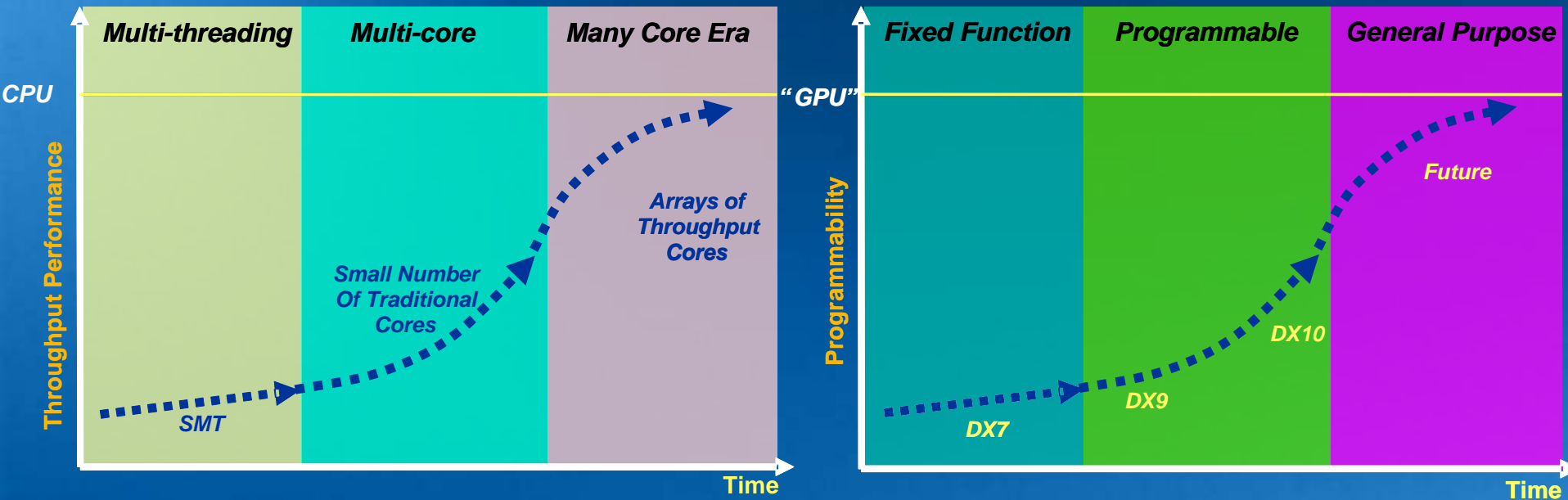
Computing Evolution

CPU:

- Evolving toward throughput computing
- Motivated by energy-efficient performance

Accelerators:

- Evolving toward general-purpose computing
- E.g. Motivated by higher quality graphics and general purpose computing



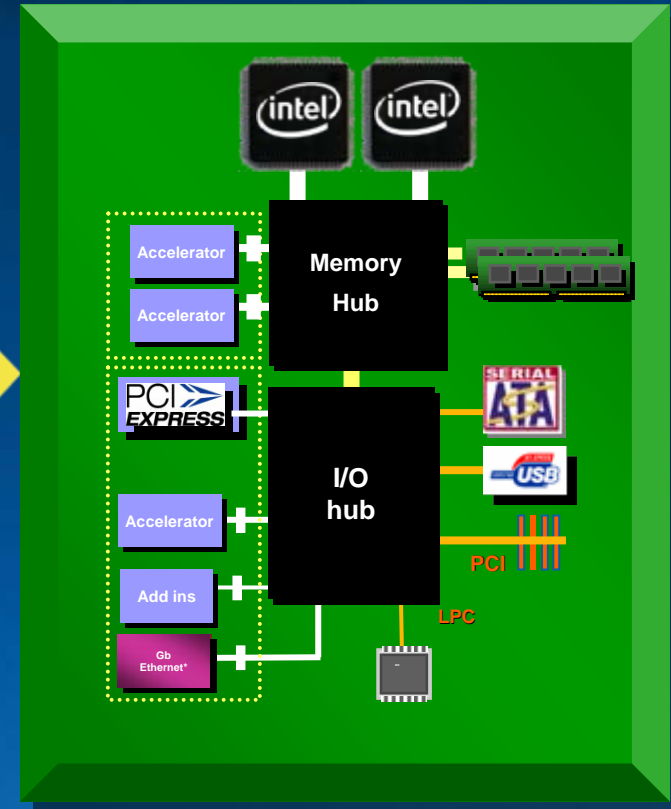
Innovation with Acceleration

Intel® Architecture with Multi Core

- General purpose Scalability
- Economies of Scale

Intel® Architecture with Accelerators

- Special Purpose Performance
 - Geneseo PCIe extensions
 - QuickAssist Technology



Unified Architecture for Multi Cores & Accelerators

Moore's Law

In 1965, Intel co-founder Gordon Moore predicted that the number of transistors on a piece of silicon would double every couple of years—an insight later dubbed “Moore’s Law.” His prediction has held true, as ever-shrinking transistor sizes have allowed exponential growth in the number of transistors on a single chip.

Moore’s Law is now a benchmark for the electronics industry, and Intel applies its principles to help people to play, learn and work. Whole new ways for the company has come about as a result of Moore’s Law.

... Accelerating with Multi core CPU

... Accelerating with *platform*



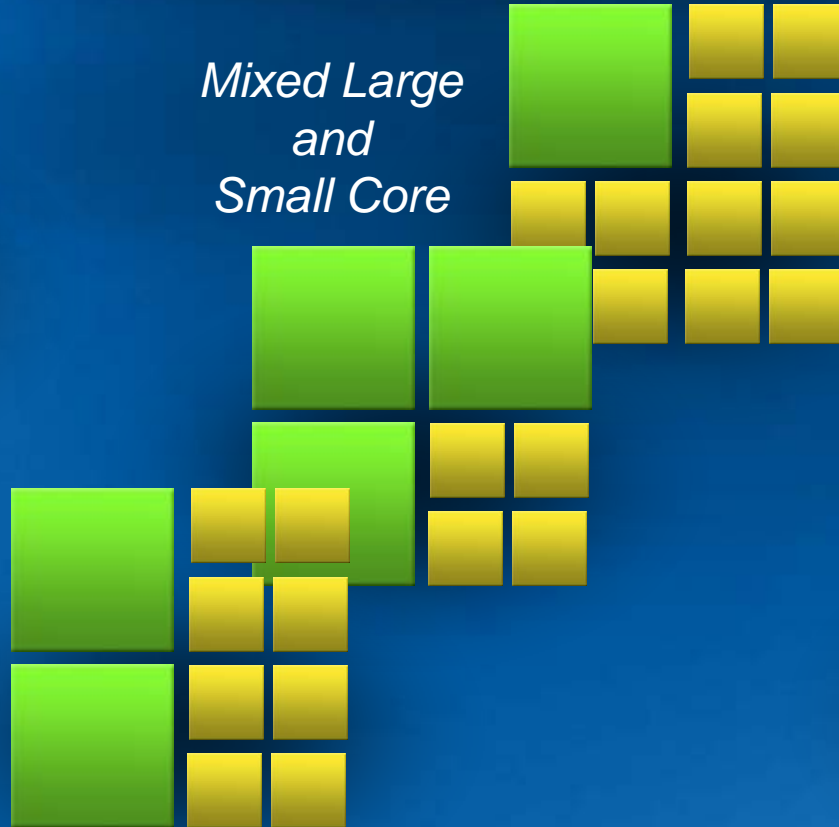


Multi-core and Many-cores

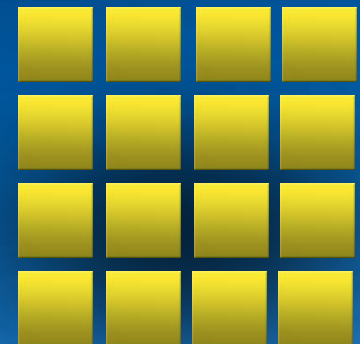
All Large Core



Mixed Large and Small Core



All Small Core



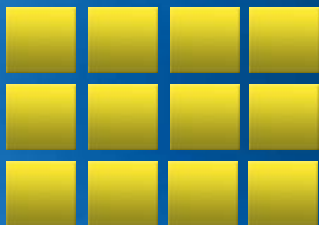
Energy Efficient Performance with Multi-threaded Cores

Note: the above pictures don't represent any current or future Intel products

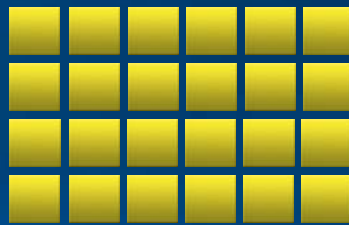


Increasing Throughput through Parallelism

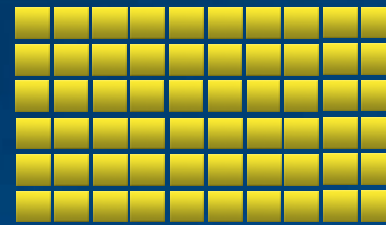
16 Cores



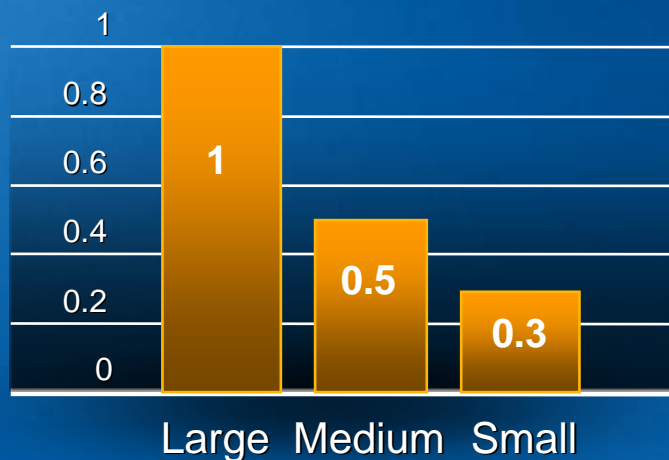
64 Cores



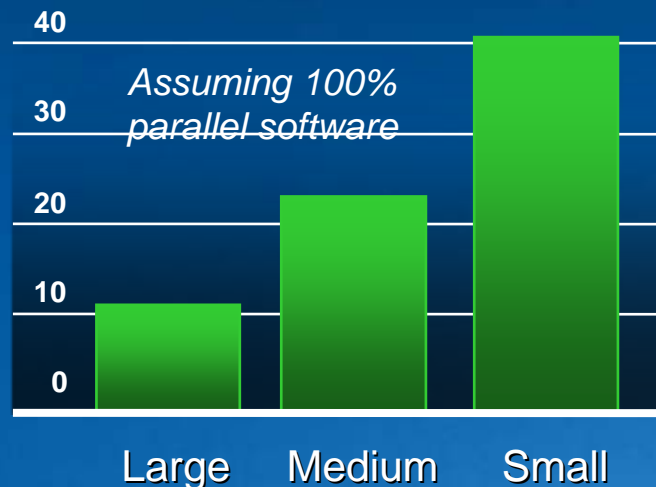
256 Cores



Single Core Performance
Relative Performance



System Performance

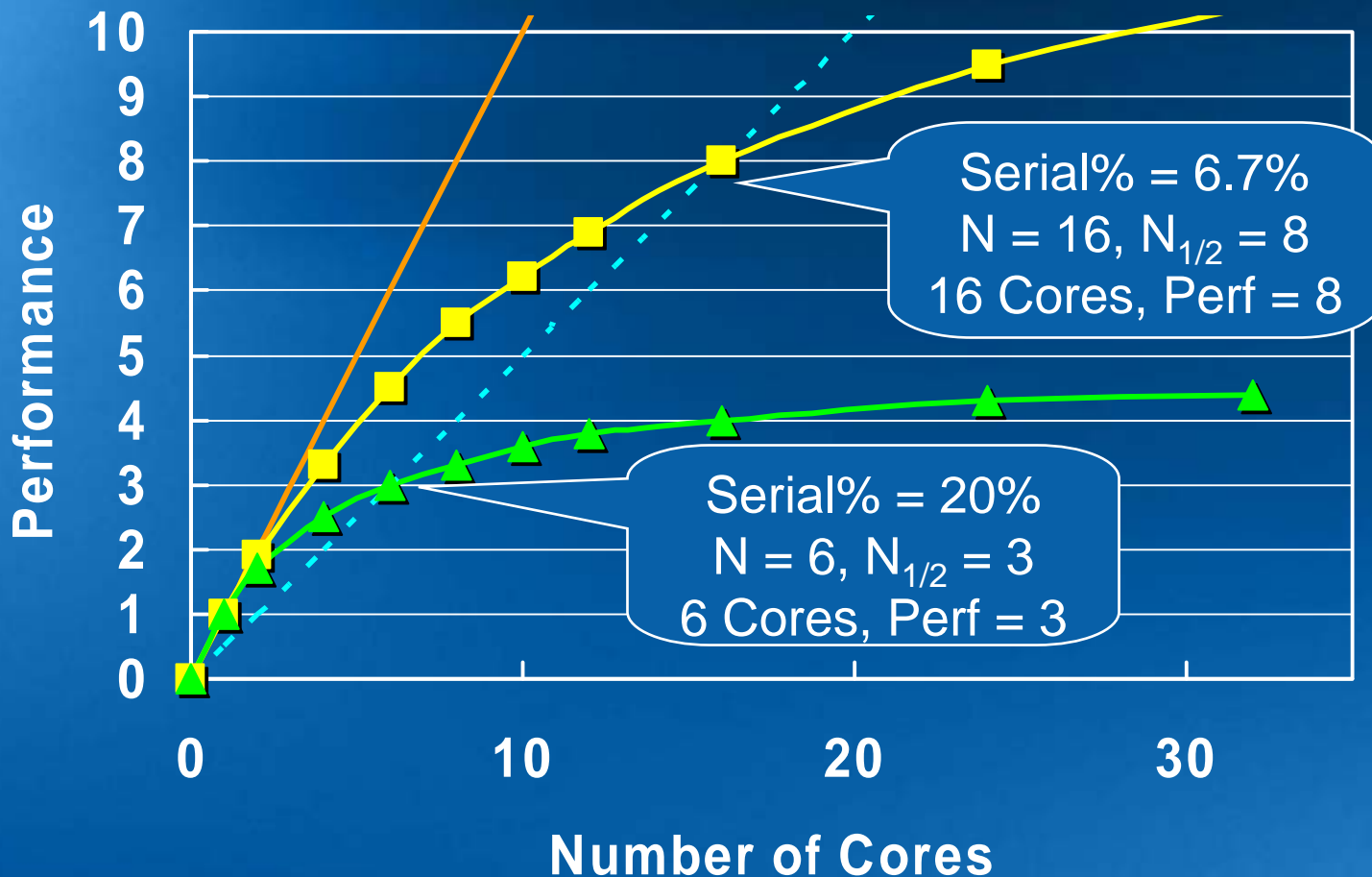


Throughput optimized (Thread Level Parallelism) performance is more power efficient and applies to interesting applications



Performance Scaling

Amdahl's Law: $\text{Parallel Speedup} = 1 / (\text{Serial}\% + (1 - \text{Serial}\%) / N)$

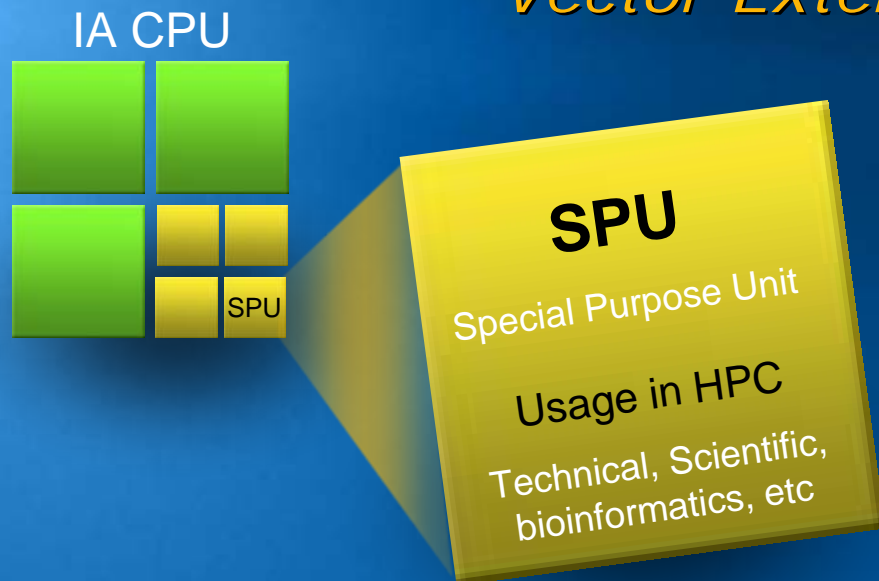


Software Parallelization is Key to the Success



Many-core with Special Purpose Hardware

Vector Extension Example

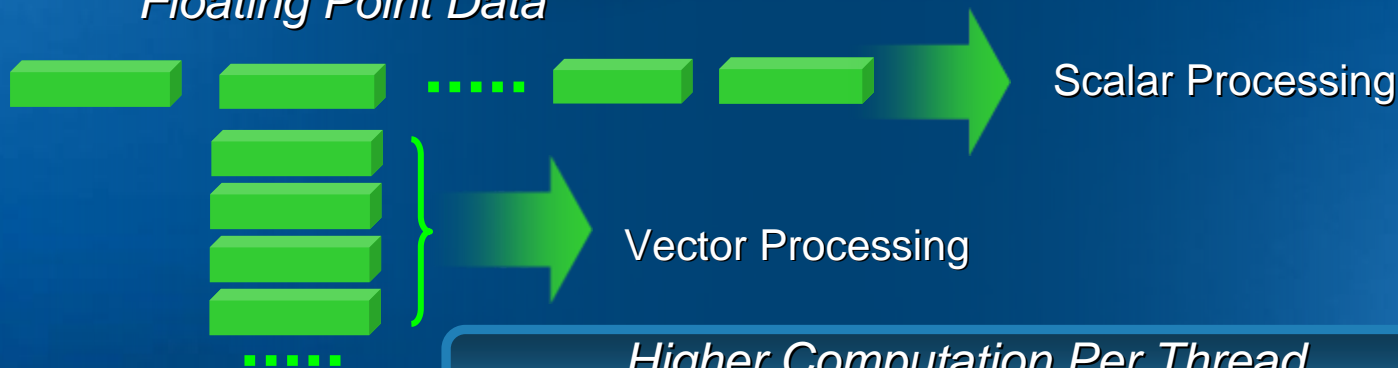


Vector Processor Unit (VPU)

SIMD Vector Advantages:

- Best energy efficiency for data parallel apps
- Efficient programming environment

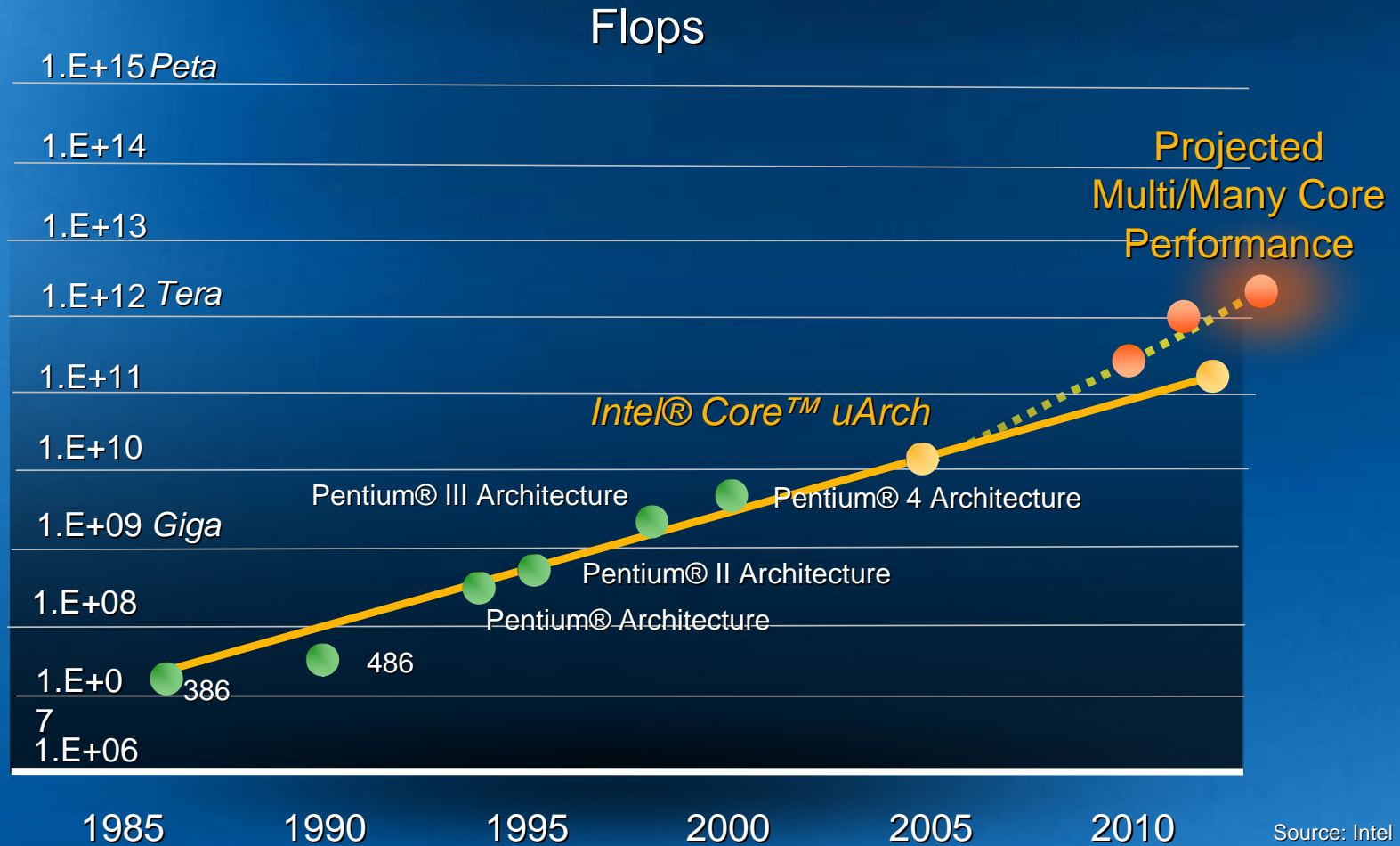
Floating Point Data



*Higher Computation Per Thread
With Fewer Threads*



Where Are We Heading with Many Core?

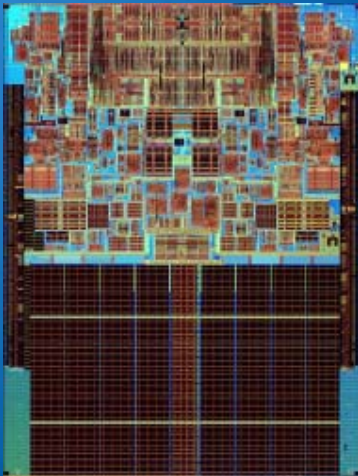


All timeframes, dates, and products are subject to change without further notification"

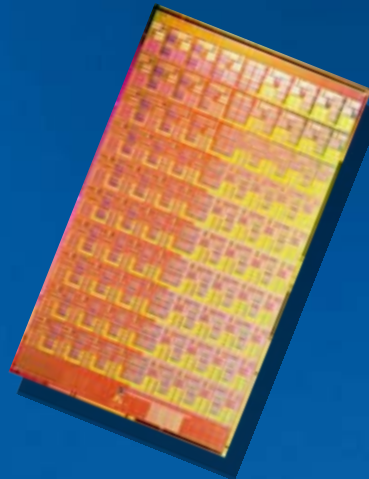


Summary of Acceleration with CPUs

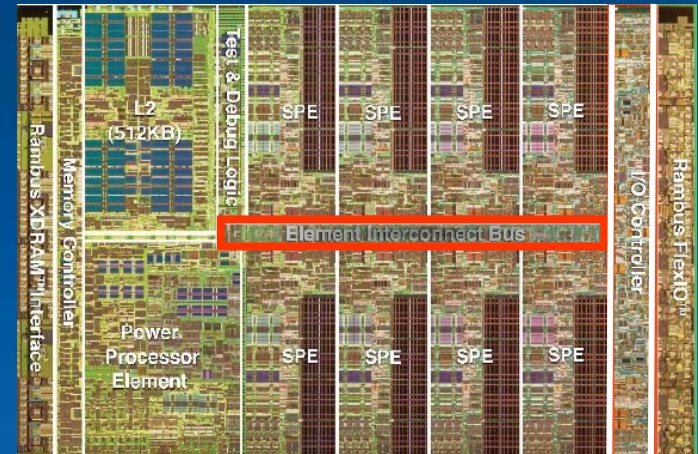
- Gigascale integration continues as per Moore's Law
- Power consumption is becoming a major concern
- Multiple cores on-die are becoming the standard to deliver performance at reasonable power



Intel® Core™ 2 Duo



Intel Research Chip: 80 core Polaris



Source: CELL presentation at Hotchips 17, 2005

The future will very likely include both ILP optimized cores and TLP optimized cores



Moore's Law

In 1965, Intel co-founder Gordon Moore predicted that the number of transistors on a piece of silicon would double every two years—an insight later dubbed "Moore's Law." His prediction has held true, as ever-shrinking transistor sizes have allowed exponential growth in the number of transistors on a single chip.

Moore's Law is now a benchmark for the electronics industry, and Intel applies its principles to... Whole new ways for people to play, learn... the company has... have come about... e's Law.

... Accelerating with Multi core CPU

... Accelerating with *platform*



Growing Interest in Add-on Accelerators

Math-Intensive Transactions

Financial, scientific, economic models

Attribute: Expansive number-crunching



Clearspeed, DRC, HPC Applications

Visualization and Media Processing

Graphics, video, speech

Attribute: Real-time response



Nvidia, Ageia, Media Processing

Next Generation I/O Devices

Networking, Storage, Infiniband

Attribute: High Bandwidth

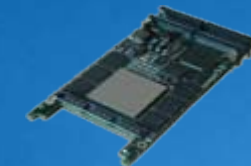


Intel, Broadcom, Mellanox,

Embedded Content Processing

Data mining, encryption, compression, XML

Attribute: Fixed algorithmic functions



SGI, CIG, Cisco



*Other names, Trademarks and brands may be claimed as the property of others

Sample of Industry Acceleration Activity

XML, Java	XML Processing	Actional, Cisco, Conformative, DataPower, Reactivity, Sarvega, Tarari
	Java Virtual Machine	Azul
Networks: Storage, Cluster, LAN	TCP/IP + Ethernet	Adaptec, Allied Telesyn, Amasso, Brocade, Chelsio, Cisco(Topspin), Citrix, Crescendo Networks, Enigma Semiconductor, Infrant, NetEffect, NextIO, Nortel, Precision I/O, Silverback, Sensory Networks, Tehuti, Toplayer, Voltaire
	InfiniBand	Cisco (Topspin), Mellanox, SilverStorm, Voltaire
Application Acceleration		(e), Radware
Security, Privacy, Rights Management		theroptics, Cisco, ase, Lucent, nCipher, Sun, Tarai, Vormetric
Real-time Analytics		
Collaboration & Information Mgmt		
High Performance Computing		Voltaire, ...
Intelligent Storage Network	Storage Virtualization	Acopia, Brocade, EMC, HP, Hitachi, Index Engines, IBM, NeoPath, Sun, Troika,
	Storage Services	
Accelerator Technology	System-on-Chip	Arteris S.A., Bay Microsystems, Broadcom, Cavium, Freescale Semiconductor, Infineon, LSI Logic, Rapport (Kilocore), Raza Microelectronics, STMicroelectronics, Teja, Tensilica
	ASIC	Advanced Architectures, Britestream, Cavium, Critical Blue, Elixent, Forte, Freescale Semiconductor (Seaway Networks), IP Fabrics, LSI Logic, Mellanox, nCipher, Propulsion Networks, STMicroelectronics, Xelerated

This is Complex

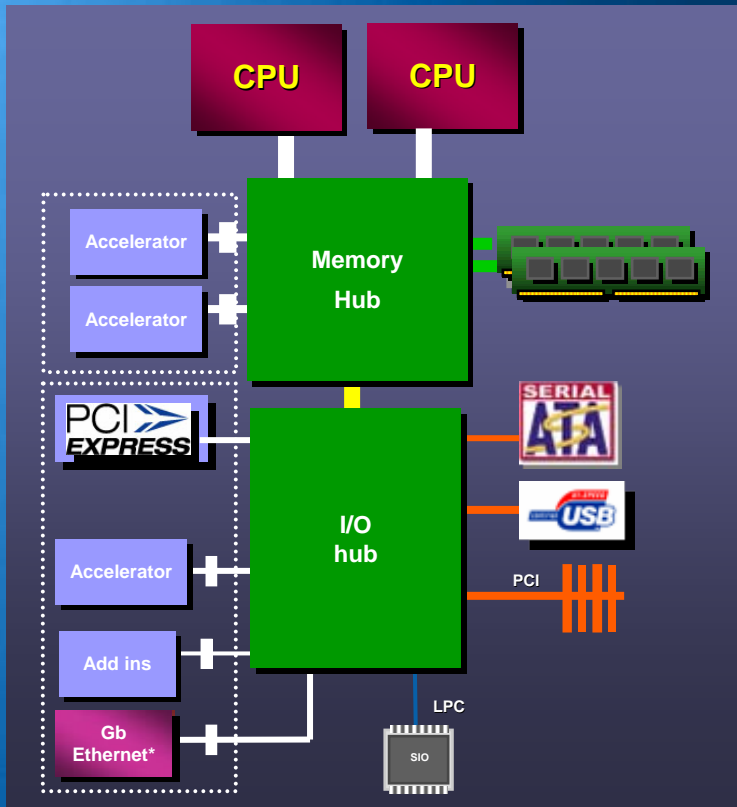
The Accelerators are Specialized to the application

- Have no common attach point
- Have no common programming model
- Have no common architecture

Source : IBM and Intel

*Other names, Trademarks and brands may be claimed as the property of others

System bottleneck for Accelerators



- **Interface Performance**
 - Reduce hardware overhead
 - Configuration and error handling
 - Status and synchronization
 - Virtualization and power management
- **Programming Model and Tools**
 - Ease of programming
 - Portability across accelerator vendors
 - Consistent programming model for both discrete, integrated & CPU based solutions
 - Reduce software overhead
- **Scheduling & Memory mgmt.**
 - Memory Buffer allocation policy
 - Virtual Address Space

Geneseo

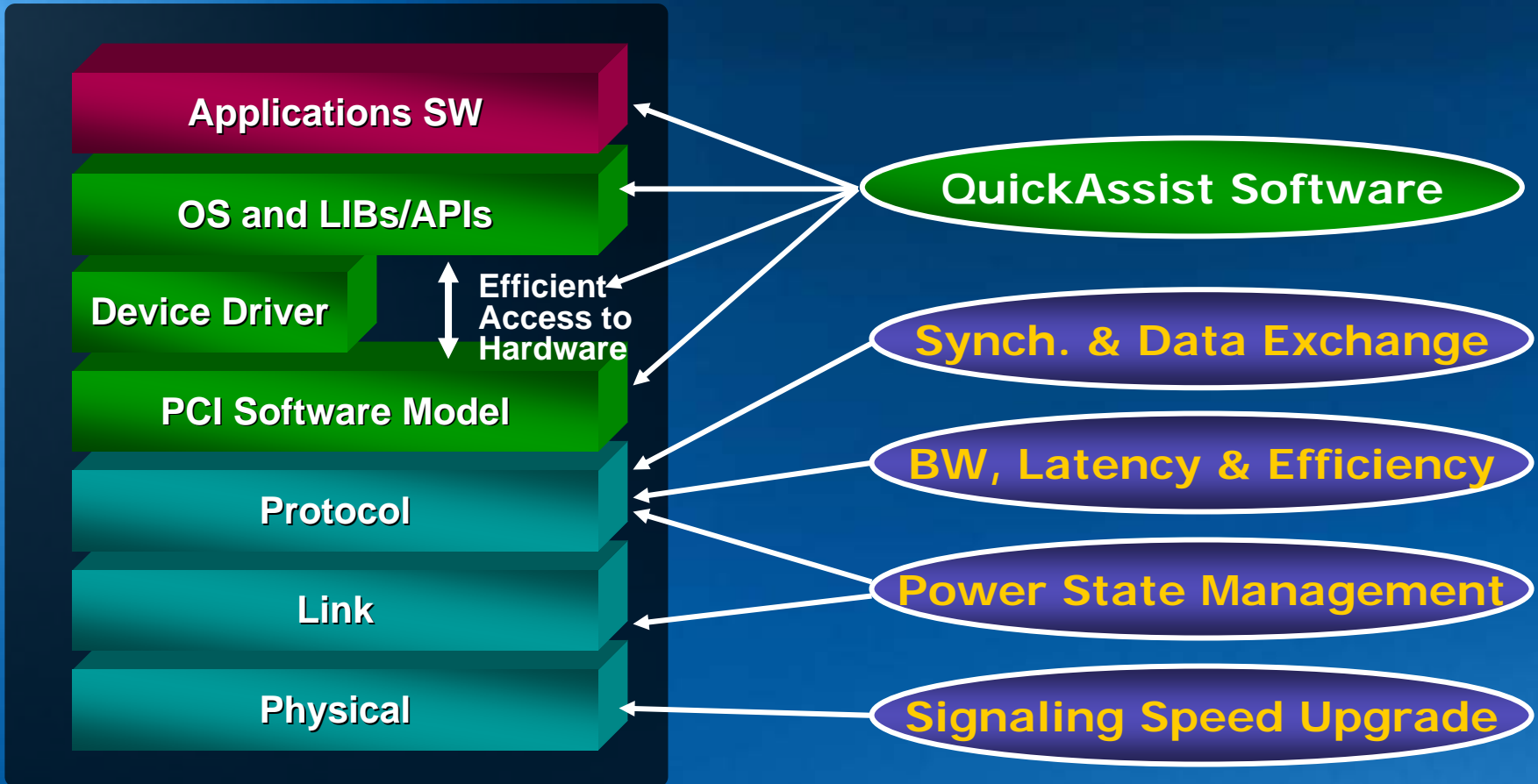
QuickAssist

Software & Platform latencies are significantly higher than physical IO latency for most accelerators

Taxonomy of Accelerators

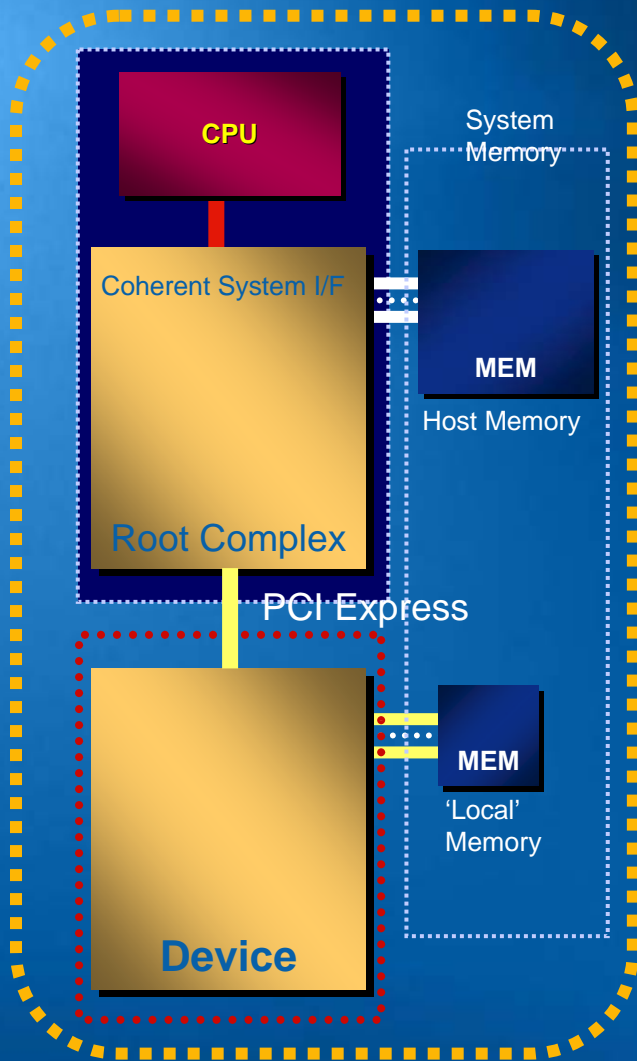
<u>Accelerator</u>	<u>HW Interface</u>	<u>SW Interface</u>	<u>Key Attributes</u>
Non Sequencer Reconfigurable e.g.: Nallatech	CPU Interface Geneseo / PCI Express	Driver function call	<ul style="list-style-type: none"> ▪ FPGA Based ▪ Functional flexibility ▪ Faster turn around ▪ Targeted Acceleration
Non Sequencer Fixed function e.g. Crypto, X87, MPEG4	Geneseo/PCI Express, On-Die	Driver function call	<ul style="list-style-type: none"> ▪ ASIC/Semi Custom ▪ Lower cost ▪ Higher performance ▪ Coarse Grain Interface
Sequencer e.g. GPUs, Cell, ClearSpeed	CPU Interface, Geneseo / PCI Express, On-die	Direct Programming	<ul style="list-style-type: none"> ▪ ASIC/Semi Custom/Custom ▪ Native ISA ▪ Fine/Coarse Grain Interface ▪ Programmable ▪ Higher performance / watt

Geneseo Layered Architecture



**Enhance PCI Express Layered Architecture
for Application Accelerators**

Geneseo: PCIe Extensions Proposed



Latency Reduction

- Data Re-use Hints – Mechanism for efficient and lower latency access
- Prefetch - Mechanism to lower access latency
- Ordering – Transaction level attribute/hint to optimize ordering within RC and memory subsystem
- Pause/Resume - Mechanism to interrupt low priority transmissions

Increased Throughput

- Signaling/Encoding Scheme – B/W efficiency improvement through replacement of 8b/10b encoding, most likely w/scrambling scheme
- Packet header overhead improvements - New semantics for transferring payloads

Software Model Improvements

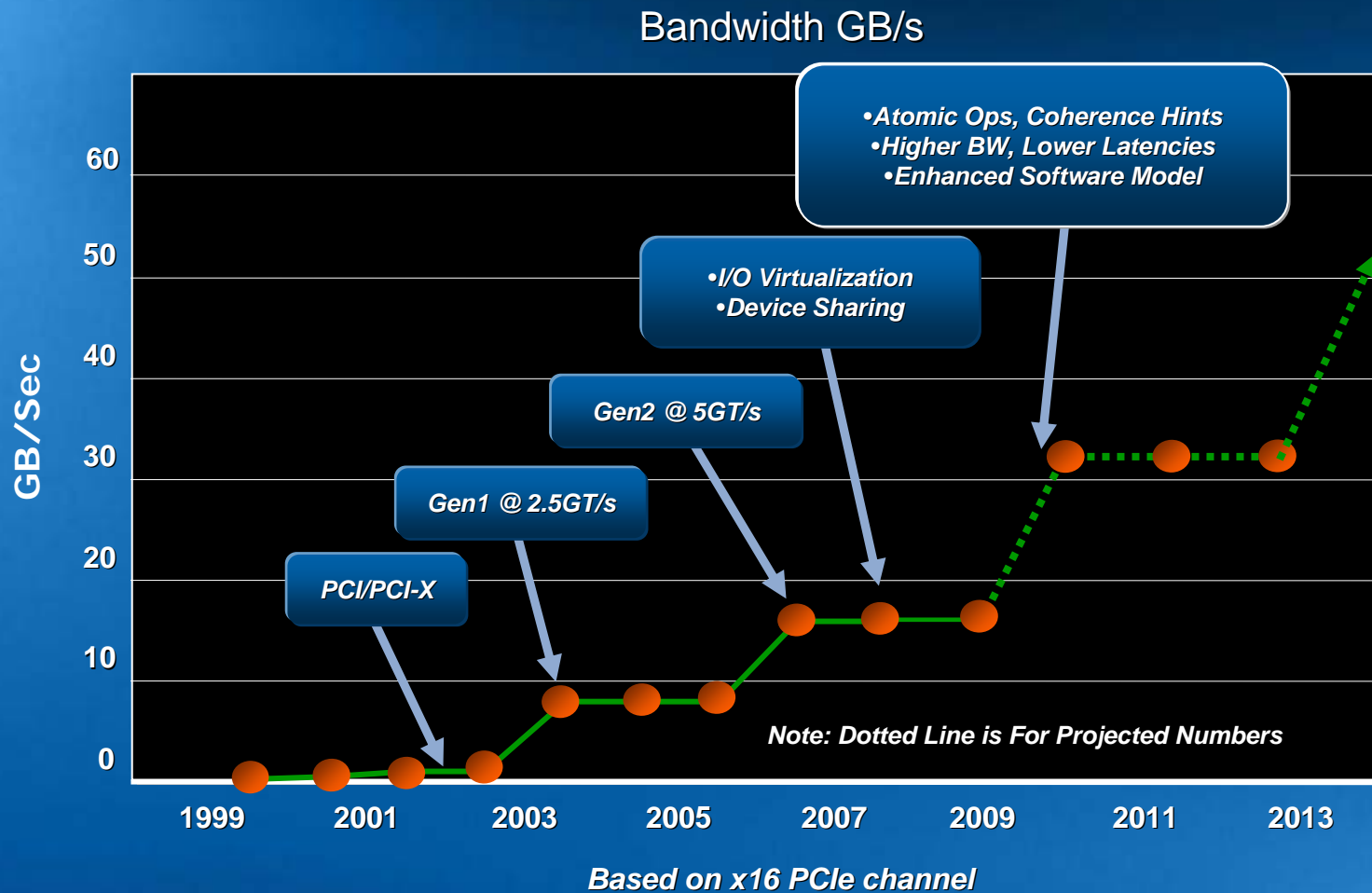
- Atomic Read-Modify-Write – Mechanisms to reduce synchronization overhead
- Shared memory support via I/O MMU

Power Management

- Dynamic Power Allocation - Support for dynamic performance/power operational modes through standard configuration mechanism

PCI E to Geneseo Transition

Performance, Bandwidth and Functionality



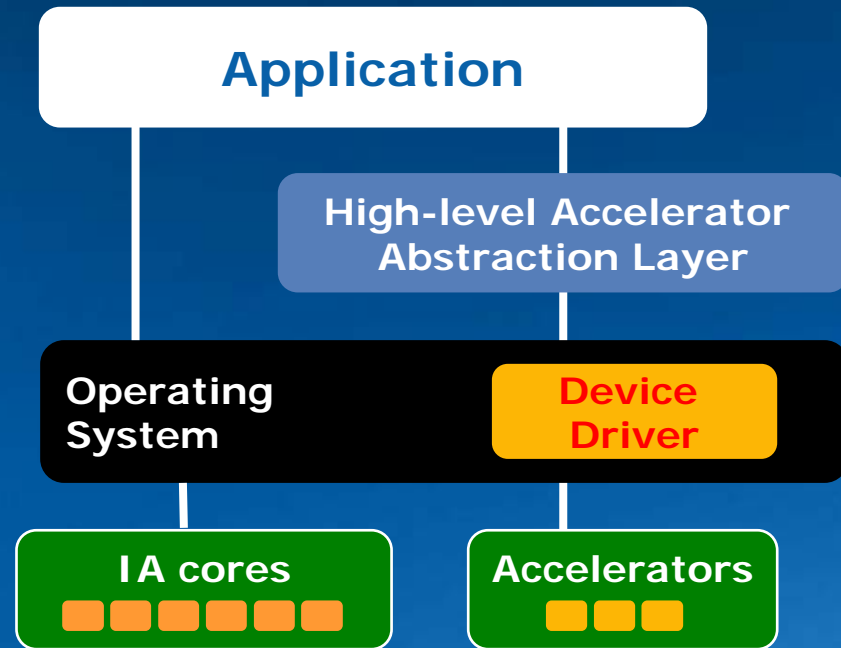
Tracking Moore's Law

Challenge: Programming Environment

- Programmers are getting used to multi threaded environment due to multiple IA cores
- Programmer cannot use the IA programming environment for accelerators.

Accelerators today rely on custom drivers bound tightly to the OS.

The software layers between the app & accelerator present significant programming overhead

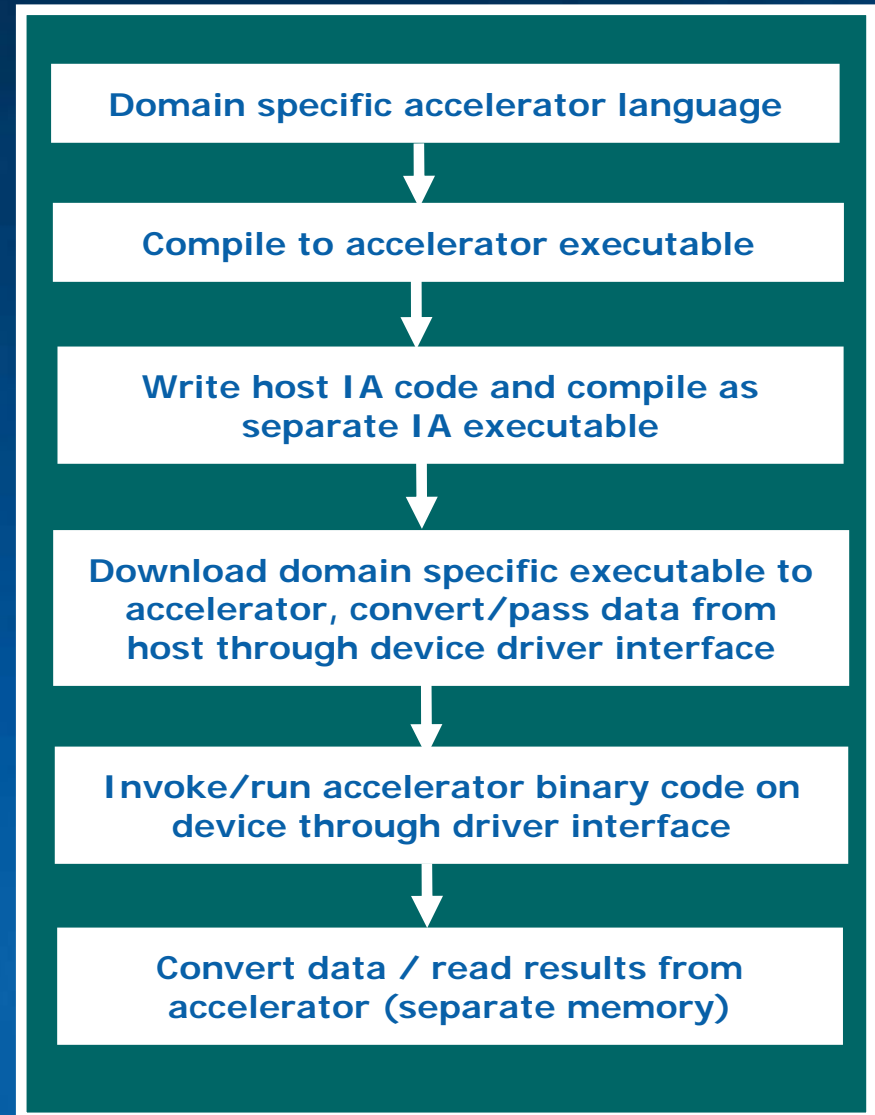


Programming environment today is Accelerator specific & cumbersome



Current Programming Environment Example

- Compiler
 - Code for accelerator isolated and compiled as accelerator specific binary using custom tools
 - Host code for manually loading / executing accelerator compiled separately as IA binary
- Runtime
 - Explicit management of data movement and data set size restrictions
 - Accelerator specific driver interfaces for loading, executing and scheduling device
 - Limited extensibility across multiple heterogeneous targets

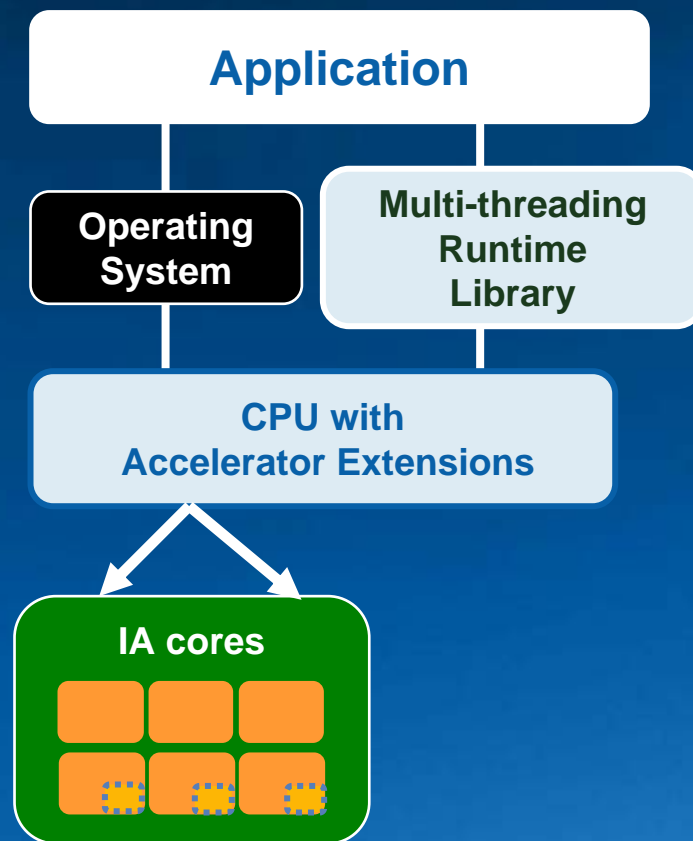


Current Programming Models

Function Call Interface	Direct Programming (Domain Specific)
<ul style="list-style-type: none">• Suitable for <u>fixed function</u> acceleration• Specific functions exposed to programmer in a familiar API environment • Examples: Crypto, IPP, CPM, CAPI, MPEG, ZLIB, ... • Data converted to avoid shared address space requirements • Data sets need to be flattened along with parameter marshalling • Data set needs to be sliced due to lack of pageable memory	<ul style="list-style-type: none">• Suitable for accelerating <u>arbitrary</u> algorithms• Unfamiliar / cumbersome programming environment very specific to particular IHV or device class • Examples: CUDA, DirectX, CTM, ... • Programmer must convert address spaces from host to device for code / data • Data sets need to be flattened along with parameter marshalling • Data set needs to be sliced due to lack of pageable memory

A Concept: IA “look ‘n feel” for Accelerators

- A method to give accelerators an IA “look ‘n feel”
 - E.g. Important for sequencers to provide complete solution, in addition to device side tools
- Run-time library with corresponding OS support makes an accelerator appear to be a processor functional unit (e.g. SSE+MMX).
- Accelerators would even **share virtual memory with IA cores**
- Run-time intelligently distributes work between IA cores and accelerators



Need: consistent Architecture and Programming Environment for a Heterogeneous Multi-core Multithreaded System”

Need Comprehensive Software Solutions

Visualization of applications and the system

Architectural Analysis



Highly optimizing compilers delivering scalable solutions

Introduce Threads & MPI



Detect latent programming to address unique challenges

Confidence / Correctness

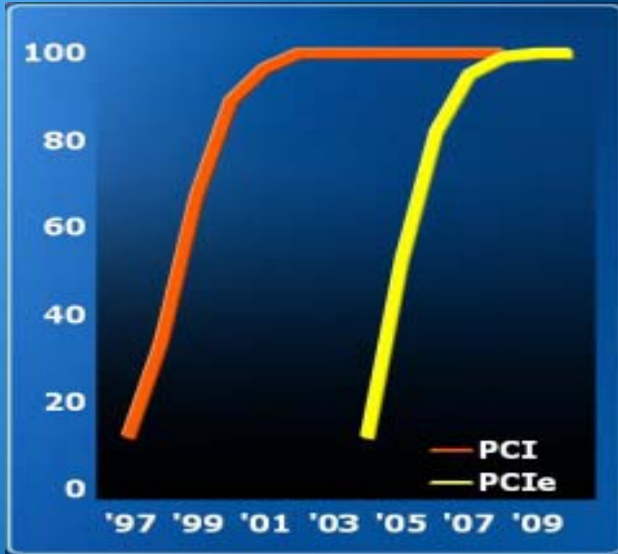


Tune for performance and scalability

Optimize / Tune



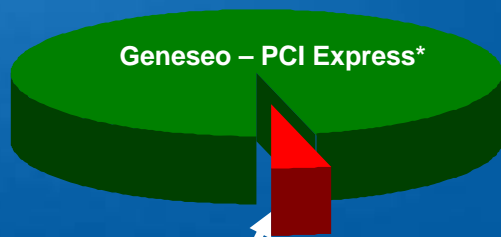
Accelerators Focus



- Open Ubiquitous Standards Based Approach
PCIe* Gen1, PCIe* Gen2, and Geneseo
- Enable third party FSB-FPGA Modules – targeted for FSI, Oil and Gas, Life Sciences, Digital Health, etc.
FSB-FPGA Modules Targeted 4Q07/1Q08



- Intel® QuickAssist Technology that seamlessly allows the SW to access CPU & accelerators across various applications.



Source : Intel Internal

Tightly Coupled

Focus on Open Standards Based Attach Strategy

*Other names and brands may be claimed as the property of others

Summary

- Moore's Law to continue for a foreseeable future
- Multiple cores on-die are becoming the standard to deliver performance at reasonable power
- Intel working to improve accelerator interface via open industry standard extensions to PCI Express* (Geneseo)
- Ease of use challenges and software issues remain
- Need applications and tools that can exploit multiple threads