The Exascale Challenge

Shekhar Borkar
Intel Corp.
October 13, 2011

This research was, in part, funded by the U.S. Government. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.
Outline

Exascale performance goals
Major challenges
Potential solutions
Paradigm shift
Summary
Performance Roadmap

- GFLOP: 1.0E+00, 1.0E+02, 1.0E+04, 1.0E+06, 1.0E+08
- Technologies: Mega, Giga, Tera, Peta, Exa
- Devices: Client, Hand-held

12 Years to Tera, 11 Years to Peta, 10 Years to Exa
From Giga to Exa, via Tera & Peta

- **Relative Tr Performance**
  - 1986: G
  - 1996: Tera (30X)
  - 2006: Peta
  - 2016: Exa

- **Transistor Performance**
  - 1986: G
  - 1996: Tera (36X, 4,000X Concurrency)
  - 2006: Peta
  - 2016: Exa (2.5M X)

- **Relative Energy/Op**
  - 1986: 5V
  - 1996: Tera
  - 2006: Peta

- **Power**
  - 1986: G
  - 1996: Tera (80X)
  - 2006: Peta
  - 2016: Exa (1M X, 4,000X)
Building Terascale Today

Decode and control
Translations
...etc
Power supply losses
Cooling...etc

~3KW
Disk

100W
Com

100W
Memory

150W
Compute

2.5KW

10TB disk @ 1TB/disk @10W

100pJ com per FLOP

0.1B/Byte @ 1.5nJ per Byte

5W

~3W

~5W

~20W

KW for Tera, MW for Peta, GW for Exa?
The UHPC* Program & Goals

*DARPA, Ubiquitous HPC Program

20MW, Exa

20KW, Peta

<table>
<thead>
<tr>
<th>Operation</th>
<th>Approx Energy Today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction Execution</td>
<td>5-10 nJ</td>
</tr>
<tr>
<td>FP operation</td>
<td>200 pJ</td>
</tr>
<tr>
<td>Byte read from cache</td>
<td>10-20 pJ</td>
</tr>
<tr>
<td>Byte read from DRAM</td>
<td>1.5 nJ</td>
</tr>
<tr>
<td>Byte over IC fabric</td>
<td>5 pJ/hop—250 pJ+</td>
</tr>
</tbody>
</table>

20 pJ/Flop (system)
Scaling Assumptions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Transistor density</td>
<td>1.75</td>
<td>1.75</td>
<td>1.75</td>
<td>1.75</td>
<td>1.75</td>
<td>1.75</td>
<td>1.75</td>
</tr>
<tr>
<td>Frequency scaling</td>
<td>15%</td>
<td>10%</td>
<td>8%</td>
<td>5%</td>
<td>4%</td>
<td>3%</td>
<td>2%</td>
</tr>
<tr>
<td>Vdd scaling</td>
<td>-10%</td>
<td>-7.5%</td>
<td>-5%</td>
<td>-2.5%</td>
<td>-1.5%</td>
<td>-1%</td>
<td>-0.5%</td>
</tr>
<tr>
<td>Dimension &amp; Capacitance</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>SD Leakage scaling/micron</td>
<td>1X Optimistic to 1.43X Pessimistic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

45nm Core + Local Memory
- DP FP Add, Multiply Integer Core, RF Router (50%)
- Memory 0.35MB (50%)
- 6mm², 3.5GHz, 7GF, 1.2W

8nm Core + Local Memory
- DP FP Add, Multiply Integer Core, RF Router 0.17mm² (50%)
- Memory 0.35MB 0.17mm² (50%)
- ~0.6mm
- 0.34mm², 4.6GHz, 9.2GF, 0.24 to 0.46W
2018, 8nm technology node

<table>
<thead>
<tr>
<th>Core Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores/Module</td>
<td>1150</td>
</tr>
<tr>
<td>Total Local Memory</td>
<td>400 MB</td>
</tr>
<tr>
<td>Frequency</td>
<td>4.61 GHz</td>
</tr>
<tr>
<td>Peak performance</td>
<td>10.6 TF</td>
</tr>
<tr>
<td>Power</td>
<td>300 - 600W</td>
</tr>
<tr>
<td>Energy efficiency</td>
<td>34 - 18 GF/Watt</td>
</tr>
</tbody>
</table>

30-60 MW for Exascale
Processor Node

Peak performance | 10.6 TF
Total DRAM Capacity | 512GB
Total DRAM BW | 1TB/s (0.1B/FLOP)
DRAM Power | 800 W*
Total Power | 1100 - 1400W
Energy efficiency | 9.5 - 8 GF/Watt

110-140 MW for Exascale

*Assumes 5% Vdd scaling each technology generation
140 pJ energy consumed per accessed bit
Node Power Breakdown

10 TF, ~ 1KW

- Aggressive voltage scaling
- Hierarchical heterogeneous topologies
- Efficient signaling Repartitioning

- Compute
- Fabric
- DRAM
Voltage Scaling

When designed to voltage scale

Energy Efficiency

Freg

Total Power

Leakage

Normalized

Vdd (Normal)
Near Threshold Logic

65nm CMOS, 50°C

320mV

Max. Frequency (MHz)

Energy Efficiency (GOPS/Watt)

Active Leakage Power (mW)

H. Kaul et al, 16.6: ISSCC08
Energy Efficiency with Vdd Scaling

- Vdd
- 0.7x
- 0.5x

~3X Compute energy efficiency with Vdd Scaling
Memory & Storage Technologies

- **Energy/bit (Pico Joule)**
- **Capacity (G Bit)**
- **Cost/bit (Pico $)**

**Technologies:**
- SRAM
- DRAM
- NAND, PCM
- Disk

(Endurance issue)
Traditional DRAM

- Activates many pages
- Lots of reads and writes (refresh)
- Small amount of read data is used
- Requires small number of pins

New DRAM architecture

- Activates few pages
- Read and write (refresh) what is needed
- All read data is used
- Requires large number of IO’s (3D)

Energy cost today: ~150 pJ/bit
3D Integration of DRAM

- Thin Logic and DRAM die
- Through silicon vias
- Power delivery through logic die
- Energy efficient, high speed IO to logic buffer
- Detailed interface signals to DRAMs created on the logic die

The most promising solution for energy efficient BW
Communication Energy

Energy/bit (pJ) vs. Interconnect Distance (cm)

- On Die
- Chip to chip
- Board to Board
- Between cabinets
On-die Mesh Interconnect

- 45nm 20mm: 70 Cores
- 32nm 20mm: 123 Cores
- 22nm 20mm: 214 Cores
- 16nm 20mm: 375 Cores

Packet Switched Mesh
16B=128 bit each direction
0.4mm @ 1.5u pitch
192GB/s Bisection BW

<table>
<thead>
<tr>
<th>Tech</th>
<th>Core (mm)</th>
<th>Port size (mm)</th>
<th>Bisection BW GB/sec@3GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>65nm</td>
<td>5</td>
<td>0.4</td>
<td>192</td>
</tr>
<tr>
<td>45nm</td>
<td>3.5</td>
<td>0.4</td>
<td>272</td>
</tr>
<tr>
<td>32nm</td>
<td>2.5</td>
<td>0.4</td>
<td>384</td>
</tr>
<tr>
<td>22nm</td>
<td>1.8</td>
<td>0.4</td>
<td>543</td>
</tr>
<tr>
<td>16nm</td>
<td>1.3</td>
<td>0.4</td>
<td>768</td>
</tr>
</tbody>
</table>
1. Network power too high (27MW for EFLOP)
2. Worse if link width scales up each generation
3. Cache coherency mechanism is complex

27 pJ/Flop on-die communication energy

Traditional Homogeneous Network
0.08 pJ/bit/switch
0.04 pJ/mm (wire)
Assume Byte/Flop
The 80-Core TFLOP Chip (2006)

- 12.64mm I/O Area
- 21.72mm I/O Area
- single tile 1.5mm
- 2.0mm

- 8 X 10 Mesh
- 32 bit links
- 320 GB/sec bisection BW @ 5 GHz
Power Breakdown
80 Core TFLOP Chip

- Clock dist.: 11%
- Dual FPMACs: 36%
- Router + Links: 28%
- IMEM + DMEM: 21%
- 10-port RF: 4%
48 Core Single Chip Cloud (2009)

- 2 Core clusters in 6 X 4 Mesh (why not 6 x 8?)
- 128 bit links
- 256 GB/sec bisection BW @ 2 GHz
Power Breakdown
48 Core SCC Chip
Packet Switched Interconnect

1. Router acts like STOP signs—adds latency
2. Each hop consumes power (unnecessary)
Mesh—Retrospective

Bus: Good at board level, does not extend well
- Transmission line issues: loss and signal integrity, limited frequency
- Width is limited by pins and board area
- Broadcast, simple to implement

Point to point busses: fast signaling over longer distance
- Board level, between boards, and racks
- High frequency, narrow links
- 1D Ring, 2D Mesh and Torus to reduce latency
- Higher complexity and latency in each node

Hence, emergence of packet switched network

But, pt-to-pt packet switched network on a chip?
Bus—The Other Extreme...

**Issues:**
- Slow, < 300MHz
- Shared, limited scalability?

**Solutions:**
- Repeaters to increase freq
- Wide busses for bandwidth
- Multiple busses for scalability

**Benefits:**
- Power?
- Simpler cache coherency

Move away from frequency, embrace parallelism
Repeate Bus (Circuit Switched)

**Arbitration:**
- Each cycle for the next cycle
- Decision visible to all nodes

**Repeaters:**
- Align repeater direction
- No driving contention

Assume:
- 10mm die,
- 1.5u bus pitch
- 50ps repeater delay

<table>
<thead>
<tr>
<th></th>
<th>Core (mm)</th>
<th>Bus Seg Delay (ps)</th>
<th>Max Bus Freq (GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>65nm</td>
<td>5</td>
<td>195</td>
<td>2.2</td>
</tr>
<tr>
<td>45nm</td>
<td>3.5</td>
<td>99</td>
<td>2</td>
</tr>
<tr>
<td>32nm</td>
<td>2.5</td>
<td>51</td>
<td>1.8</td>
</tr>
<tr>
<td>22nm</td>
<td>1.8</td>
<td>26</td>
<td>1.5</td>
</tr>
<tr>
<td>16nm</td>
<td>1.3</td>
<td>13</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Anders et al, A 2.9Tb/s 8W 64-Core Circuit-switched Network-on-Chip in 45nm CMOS, ESSCIRC 2008
Anders et al, A 4.1Tb/s Bisection-Bandwidth 560Gb/s/W Streaming Circuit-Switched 8×8 Mesh Network-on-Chip in 45nm CMOS, ISSCC 2008
Circuit-switched NoC eliminates intra-route data storage
Packet-switching used only for channel requests
⇒ High bandwidth and energy efficiency (1.6 to 0.6 pJ/bit)

Anders et al, A 4.1Tb/s Bisection-Bandwidth 560Gb/s/W Streaming Circuit-Switched 8×8 Mesh Network-on-Chip in 45nm CMOS, ISSCC 2008
Hierarchical & Heterogeneous

Bus to connect over short distances

Hierarchy of Busses

Or hierarchical circuit and packet switched networks
Tapered, but over-provisioned bandwidth. Pay (energy) as you go (afford).
But wait, what about Optical?

**Tx Power**

\[ P_{\text{min}} = 5.5 \text{ mW/Gbps} @ 10 \text{ Gb/s x 4} \]

65nm

**Rx Frontend Power**

**Transceiver Power**

**E/O Power Overheads**

- Most of power overhead comes from laser current
- Low voltage (~1 V), low current (< 10 mA) needed to achieve power efficiency less than 10 mW/Gbps

Source: Hirotaka Tamure (Fujitsu), ISSCC 08 Workshop on HS Transceivers
Data Locality

Core-to-core Communication on the chip: 
\(~10\text{pJ per Byte}\)

Chip to memory Communication:
\(~1.5\text{nJ per Byte}\)
\(~150\text{pJ per Byte}\)

Chip to chip Communication:
\(~100\text{pJ per Byte}\)

Data movement is expensive—keep it local
(1) Core to core, (2) Chip-to-chip, (3) Memory
Impact of Exploding Parallelism

Almost flat because Vdd close to Vt

4X increase in the number of cores (Parallelism)

Increased communication and related energy

Increased HW, and unreliability

1. Strike a balance between Com & Computation
2. Resiliency (Gradual, Intermittent, Permanent faults)
## Road to Unreliability?

<table>
<thead>
<tr>
<th>From Peta to Exa</th>
<th>Reliability Issues</th>
</tr>
</thead>
</table>
| 1,000X parallelism                    | More hardware for something to go wrong  
>1,000X intermittent faults due to soft errors                                  |
| Aggressive Vcc scaling to reduce power/energy | Gradual faults due to increased variations  
More susceptible to Vcc droops (noise)  
More susceptible to dynamic temp variations  
Exacerbates intermittent faults—soft errors                                     |
| Deeply scaled technologies           | Aging related faults  
Lack of burn-in?  
Variability increases dramatically                                               |

**Resiliency will be the corner-stone**
Resiliency

<table>
<thead>
<tr>
<th>Faults</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permanent faults</td>
<td>Stuck-at 0 &amp; 1</td>
</tr>
<tr>
<td>Gradual faults</td>
<td>Variability</td>
</tr>
<tr>
<td></td>
<td>Temperature</td>
</tr>
<tr>
<td>Intermittent faults</td>
<td>Soft errors</td>
</tr>
<tr>
<td></td>
<td>Voltage droops</td>
</tr>
<tr>
<td>Aging faults</td>
<td>Degradation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Faults cause errors (data &amp; control)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datapath errors</td>
</tr>
<tr>
<td>Silent data corruption</td>
</tr>
<tr>
<td>Control errors</td>
</tr>
</tbody>
</table>

Minimal overhead for resiliency

Error detection  
Fault isolation  
Fault confinement  
Reconfiguration  
Recovery & Adapt
# Needs a Paradigm Shift

## Past and present priorities—

<table>
<thead>
<tr>
<th>Single thread performance</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming productivity</td>
<td>Legacy, compatibility Architecture features for productivity</td>
</tr>
<tr>
<td>Constraints</td>
<td>(1) Cost</td>
</tr>
<tr>
<td></td>
<td>(2) Reasonable Power/Energy</td>
</tr>
</tbody>
</table>

## Future priorities—

<table>
<thead>
<tr>
<th>Throughput performance</th>
<th>Parallelism, application specific HW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power/Energy</td>
<td>Architecture features for energy Simplicity</td>
</tr>
<tr>
<td>Constraints</td>
<td>(1) Programming productivity</td>
</tr>
<tr>
<td></td>
<td>(2) Cost</td>
</tr>
</tbody>
</table>

**Evaluate each (old) architecture feature with new priorities**
Number of design rules increase
Complexity increases even faster
Exponentially increasing mask cost
Exponentially increasing cost of a design error
Design must function and in production on the first pass

Source: Synopsys Inc.
Source: David Abercrombie, Mentor Graphics
Toshiba’s Experiment
Streaming Processing Unit (Cell)

IBM’s Full Custom

Toshiba’s Synthesized

65nm process technology, 8 metal layers

30% smaller than full custom design, 28% shorter wires

Ueno, et al; A Design Methodology Realizing an Over GHz Synthesizable Streaming Processing Unit; VLSI Circuit Symposium 2007, 5-4
Future of Custom Design

Why do we do custom design?

1. Higher Frequency (Interconnect engineering), &
2. Higher density (smaller die)

In the future, frequency will not be limited by interconnect RC
Design rules have become complex, and becoming worse
Design mistake will be a lot more expensive

Claims:

Custom design will not give frequency benefit
Custom design will not give die size benefit

*Hard macros will be short lived!*
Platform, System on a Chip

Complex logic cores will become uninteresting
It’s the platform and the system that will matter
Logic cores will be *standard cells* with different flavors

Each cell is a soft macro
Fully validated
Design synthesized for each process technology
Put together as a system on a chip

**Challenges:**
Correct by construction
Validation of the system
Exascale Technology for Mainstream

Most of the energy budget for Legacy cores and adequate cache

Challenge: Use remaining budget with cache power density for logic

1. Aggressive Vdd scaling to Vt+Δ (Near Threshold Logic)
2. Fine grain system level power management
Summary

Von-Neumann computing & CMOS technology (nothing else in sight)

Voltage scaling to reduce power and energy
  • Explodes parallelism
  • Cost of communication vs computation—critical balance
  • Resiliency to combat side-effects and unreliability

Programming system for extreme parallelism

System software to harmonize all of the above