Realizing a Power Efficient, Easy to Program Manycore: The Tile Processor

Anant Agarwal
Tilera, MIT
Multicore is Everywhere

Laptop

Automobiles

Cellphones

Datacenters/Clouds

TVs

Desktop

Network Switches

Telepresence
Parallel Computing No Longer Province of Rocket Scientists

The computing world is ready for radical change.
Tilera Announced the TILE-Gx100 in Oct `09

- 100 general-purpose cores
- Runs SMP Linux
- Standard programming
- 1.25GHz – 1.5GHz
- Full 64-bit processors
- 32 MBytes total cache
- 546 Gbps memory BW
- 200 Tbps iMesh BW
- 80-120 Gbps packet I/O
- 80 Gbps PCIe I/O
- Wire-speed packet engine
  - 120Mpps
- MiCA engines:
  - 40 Gbps crypto
  - 20 Gbps compress
But first, let’s rewind to 1996
Research Vision to Commercial Product

The opportunity
1B Transistors in 2007

1996

A blank slate

1997

MIT Raw
16 cores

2002

The future?

100B transistors

2018

TileGx100
100 cores

2010

Tile Processor
64 cores

2007
The Opportunity

1996...

20MIPS cpu in 1987

Few thousand gates
The Opportunity

The billion transistor chip of 2007
How to Fritter Away Opportunity

- Does not scale
- Burns a lot of power

Diagram:
- Caches
- Control
- 100 ported RegFil and RR
- More resolution buffers, control
Key Multicore Challenges: The 3 P’s

- **Performance challenge**
  - How to scale from 1 to 1000 cores -- the number of cores is the new MegaHertz

- **Power efficiency challenge**
  - Performance per watt is the new metric – power efficiency trumps instruction-set (ISA) compatibility

- **Programming challenge**
  - How to distinguish between a processor and a paper weight
Our Early Raw Proposal

Got parallelism?
Take Inspiration from ASICs

ASICs have high performance and low power
- Custom-routed, short wires
- Lots of ALUs, registers, memories – huge on-chip parallelism

But how to build a programmable chip?
Replace Long Wires with Routed Interconnect

[IEEE Computer ’97]
16-Way ALU Clump → Distributed ALUs
Distributed ALUs, Routed Bypass Network

Scalar Operand Network (SON) [TPDS 2005]
From a Large Centralized Cache...
...to a Distributed Shared Cache
Distributed Everything + Routed Interconnect → Tiled Multicore

Each tile is a processor, so programmable
On-Chip Interconnect Routes Messages

For distributed cache access, off-chip misses, I/O, user-level messages
Tiled Multicore Captures ASIC Benefits and is Programmable

- Scales to large numbers of cores
- Modular – design and verify 1 tile
- Power efficient
  - Short wires plus locality opts – $CV^2f$
  - Chandrakasan effect, more cores at lower freq and voltage – $CV^2f$

Core + Switch = Tile

Current Bus Architecture
“Raw” Die Photo

...2002

16 tiles, 425MHz, 18 Watts (vpenta) IBM 0.18 micron process

[ISCA 2004]
Research Vision to Commercial Product

1996

A blank slate

1B Transistors in 2007

1997

The opportunity

1996

MIT Raw
16 cores

2002

The future?

TILE-Gx100
100 cores

2010

100B transistors

2018

Tile Processor
64 cores

2007
Why Do We Care?
Markets Demanding More Performance at Lower Power

**Cloud**
- Demands higher performance density and low power

**Wireless infrastructure market**
- Demands higher throughput, more services and low power

**Networking market**
- Demands higher performance, better security and services

**Digital multimedia market**
- Newer algorithms (e.g., H.264) for higher compression demand more performance
The Tile Processor is a System-on-a-Chip

Performance

<table>
<thead>
<tr>
<th></th>
<th>TILEPro64</th>
</tr>
</thead>
<tbody>
<tr>
<td># of cores</td>
<td>64</td>
</tr>
<tr>
<td>On-chip cache (MB)</td>
<td>5.6</td>
</tr>
<tr>
<td>Cache coherency</td>
<td>Yes w/ DDC</td>
</tr>
<tr>
<td>Operations (16/32-bit BOPS)</td>
<td>221/166</td>
</tr>
<tr>
<td>On chip bandwidth (Terabit/s)</td>
<td>38</td>
</tr>
<tr>
<td>Clock speed (MHz)</td>
<td>700, 866</td>
</tr>
</tbody>
</table>

Power

<table>
<thead>
<tr>
<th></th>
<th>TILEPro64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical power -5 device (W)</td>
<td>N/A</td>
</tr>
<tr>
<td>Typical power -7 device (W)</td>
<td>17-21</td>
</tr>
<tr>
<td>Typical power -9 device (W)</td>
<td>27-34</td>
</tr>
</tbody>
</table>

I/O and Memory

<table>
<thead>
<tr>
<th></th>
<th>TILEPro64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet bandwidth</td>
<td>2 XAUI, 2GbE</td>
</tr>
<tr>
<td>PCIe interfaces</td>
<td>2 x 4-lanes</td>
</tr>
<tr>
<td>Flexible I/O pins</td>
<td>64</td>
</tr>
<tr>
<td>DDR2 bandwidth (peak Gbps)</td>
<td>200</td>
</tr>
</tbody>
</table>

TILEPro36 also available

Tilera development systems
Remember the 3 P’s?

- **Performance challenge**
  - How to scale from 1 to 1000 cores -- the number of cores is the new MegaHertz

- **Power efficiency challenge**
  - Performance per watt is the new metric – power efficiency trumps instruction-set (ISA) compatibility

- **Programming challenge**
  - How to distinguish between a processor and a toaster
Key Innovations

1. General purpose cores
   – Standard OS and programming

2. iMesh™ Network
   – How to scale and be energy efficient

3. Multicore Dynamic Distributed Cache
   – How to achieve cache coherence and run standard software

4. Multicore Hardwall™
   – How to virtualize multicore

5. Multicore Development Environment
   – How to program
1 - Full-Featured General Cores Enable Throughput Oriented Computing and Standard Languages

- **Processor**
  - Each core is a complete computer
  - 3-way VLIW CPU
  - Designed for low power – 200mW per core
  - SIMD instructions: 32, 16, and 8-bit ops
  - Instructions for video (e.g., SAD) and networking
  - Protection and interrupts
  - Single core performance roughly the same as a modern MIPS or ARM core

- **Memory**
  - L1 cache: 8KB I, 8KB D, 1 cycle latency
  - L2 cache: 64KB unified, 7 cycle latency
  - 32-bit virtual address space per process
  - 64-bit physical address space
  - Instruction and data TLBs
  - Cache integrated 2D DMA engine

- **Switch in each tile**

- **Runs SMP Linux**
- **Runs off-the-shelf open-source C/C++ programs**
2- iMesh On-Chip Network

- **Distributed resources**
  - 2D Mesh peer-to-peer tile networks
  - 5 independent networks
  - Each with 32-bit channels, full duplex
  - Tile-to-memory, tile-to-tile, and tile-to-IO data transfer
  - Packet switched, wormhole routed, point-to-point
  - Near-neighbour flow control, dimension-ordered routing

- **Performance and energy efficiency**
  - ASIC-like one cycle hop latency
  - 2 Tbps bisection bandwidth
  - 32 Tbps interconnect bandwidth
  - Low power

- **6 independent networks**
  - One static, four dynamic
  - IDN – System and I/O
  - MDN – Cache misses, DMA, other memory
  - TDN, VDN – Tile to tile memory access and coherence
  - UDN, STN – User-level streaming and scalar transfer

Achieves scalability and power efficiency
Meshes are Power Efficient

More than 80% power savings over buses

[Konstantakopoulos ‘07]
3 – Coherent On-Chip Cache System Enables Standard Programming

- **Distributed cache**
  - Each tile has local L1 and L2 cache
  - Aggregate of L2 serves as a globally shared L3

- **Dynamic Distributed Cache (DDC™)**
  - Hardware based cache coherence
  - Hardware tracks sharers, invalidates stale copies
  - One or multiple coherency domains
  - Dedicated network to manage coherency

- **Coherent direct-to-cache I/O**
  - Header/packet delivered directly to tile caches
  - Cache coherent delivery
  - Significant DRAM bandwidth and latency reduction
4 – Multicore Hardwall™ Technology for Virtualization and Protection in Cloud Environments

The virtualization and protection challenge

- Multicores need to run multiple OS’s and applications in cloud environments
- OS’s must be protected from each other
- I/O and other shared resources must be virtualized

Multicore Hardwall technology

- Protects applications and OS by prohibiting unwanted interactions
- Configurable to include one or many tiles in a protected area
- Supported by Tilera hypervisor running on all the tiles
Configurable Fine Grain Protection

Full Stack Linux with Hypervisor

TLB Access  DMA Engine  “User” Network  I/O Network

Key:
0 – User Code
1 – OS
2 – Hypervisor or PAL
3 – Hypervisor Debugger
5 – Standard Tools and Software Stack

Multicore Development Environment

**Standards-based tools**

- Standard programming
  - SMP Linux 2.6.26
  - ANSI C/C++
  - pthreads

- Integrated tools
  - SGI compiler
  - Standard gdb gprof
  - Eclipse IDE

- Innovative tools
  - Multicore debug
  - Multicore profile

**Standard application stack**

- **Application layer**
  - Open source apps
  - Standard C/C++ libs

- **Operating System layer**
  - 64-way SMP Linux
  - Zero Overhead Linux
  - Bare metal environment

- **Hypervisor layer**
  - Virtualizes hardware
  - I/O devices drivers
  - Load balancer

---

© 2009 Copyright Tilera Corporation. All Rights Reserved.
Multiple Software Environments to Meet Diverse Needs of Embedded and Cloud Systems

- **Standard SMP Linux**
  - Standard Linux environment with processes and threads
  - Ideal for applications and control plane code requiring operating system services
  - Open source applications work out of the box

- **Standard SMP Linux with Zero Overhead Linux (ZOL)**
  - Zero Linux overhead (Eliminates OS interrupts, timer ticks, etc.)
  - Transparent to programmer - no software change required
  - For high performance data-plane applications not requiring OS services

- **Bare metal environment**
  - Full control of the hardware on up to 64 tiles
  - No operating system or hypervisor layers
  - For embedded applications requiring fine grain control of memory, and IO

- **Hybrid environment**
  - Using 2 or all three of the above models
  - Each environment can be run on one or more Tiles
  - Ideal for customers aggregating data plane and control plane code on one chip
Parallel Programming using Standard Models

- 64-way SMP Linux
  - Single system image across all tiles
- Standard pthreads API
  - pthread_create()
  - Shared memory model by default
  - Synchronize using mutexes and locks
- Standard Linux processes
  - fork(), exec()
  - Separate address space
  - Share memory: mmap(), mspaces
  - Communicate: Pipes / local sockets
- Gentle slope programming optimizations using Linux extensions
  - Control memory location and distribution
  - Control thread scheduling and location
- New models and further optimizations using TMC library (Tile Multicore Components)
Scaling Up: TileGx100 Announced in Oct `09

- 100 general-purpose cores
- Runs SMP Linux
- Standard programming
- 1.25GHz – 1.5GHz
- Full 64-bit processors
- 32 MBytes total cache
- 546 Gbps memory BW
- 200 Tbps iMesh BW
- 80-120 Gbps packet I/O
- 80 Gbps PCIe I/O
- Wire-speed packet engine
  - 120Mpps
- MiCA engines:
  - 40 Gbps crypto
  - 20 Gbps compress
Scaling Down: TILEGx16™ also Announced

- 16 Processor Cores
- 1.0 & 1.25 GHz speeds
- Full 64-bit processors
- 5.2 MBytes total cache
- 200 Gbps memory BW
- 20 Tbps iMesh BW
- 24 Gbps total packet I/O
  - 2 ports 10GbE (XAUI)
  - 12 ports 1GbE (SGMII)
- 32 Gbps PCIe I/O
- Wire-speed packet engine
  - 30Mpps
- MiCA engine:
  - 10 Gbps crypto
  - 5 Gbps compress &
  5 Gbps decompress
- Midrange 36 core part also announced
Vision for the Future

Corollary of Moore’s Law: The number of cores will double every 18 months

The ‘core’ is the logic gate of the 21st century