



The Tile Processor: A 64-Core Multicore for Embedded Processing

Anant Agarwal
Tilera Corporation

HPEC 2007

Markets Demanding More Performance

Networking market

- Demand for high performance
 - Services being integrated in the infrastructure
 - Faster speeds 1Gbps » 2Gbps » 4Gbps » 10 Gbps
- Demand for more services
 - In-line L4 – L7 services, intelligence everywhere
 - Integration of video with networking



Security Appliances



Switches



Routers

Digital Multimedia market

- Demand for high performance
 - H.264 encoding for High Definition
 - Pre & post processing
- Demand for more services
 - VoD, video conferencing, transcoding, transrating



Video Conferencing



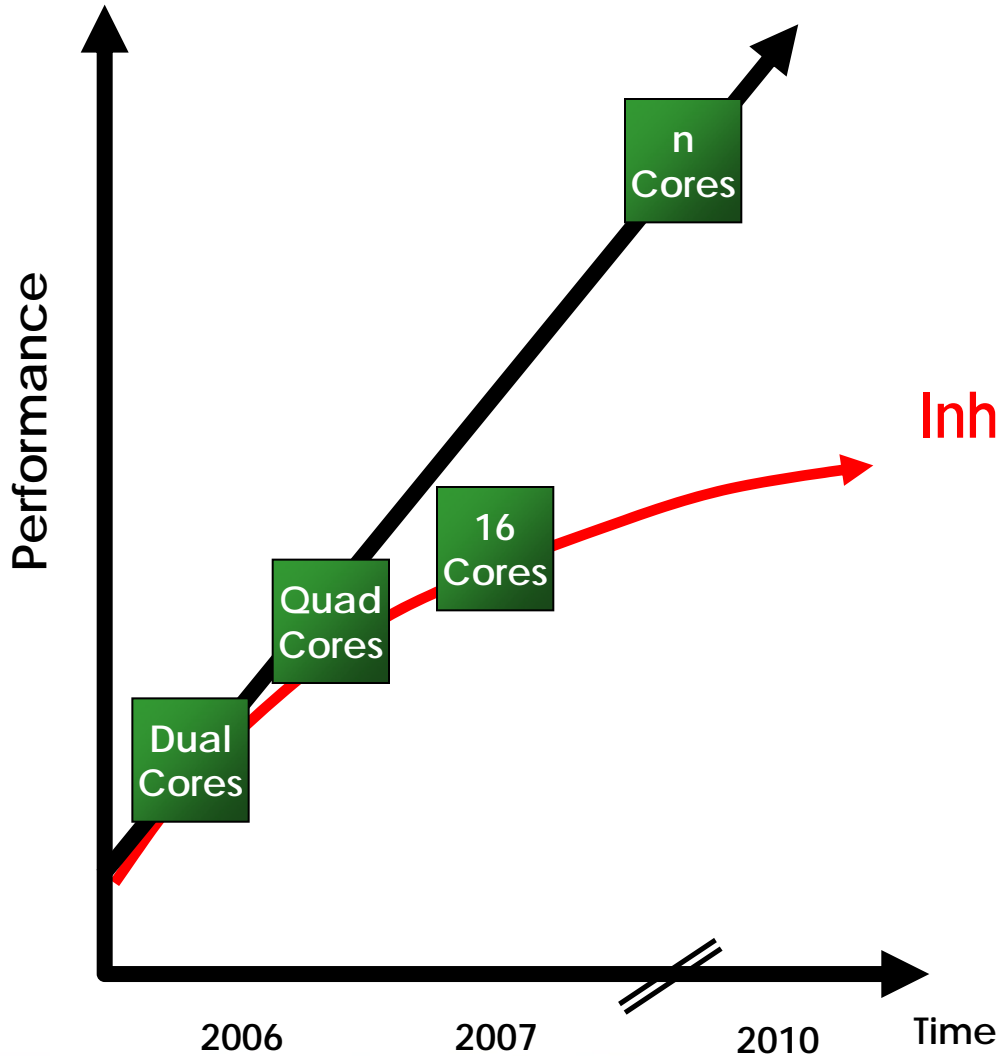
Cable & Broadcast



Surveillance DVR

... and with power efficiency and programming ease

Industry Aggressively Embracing Multicore

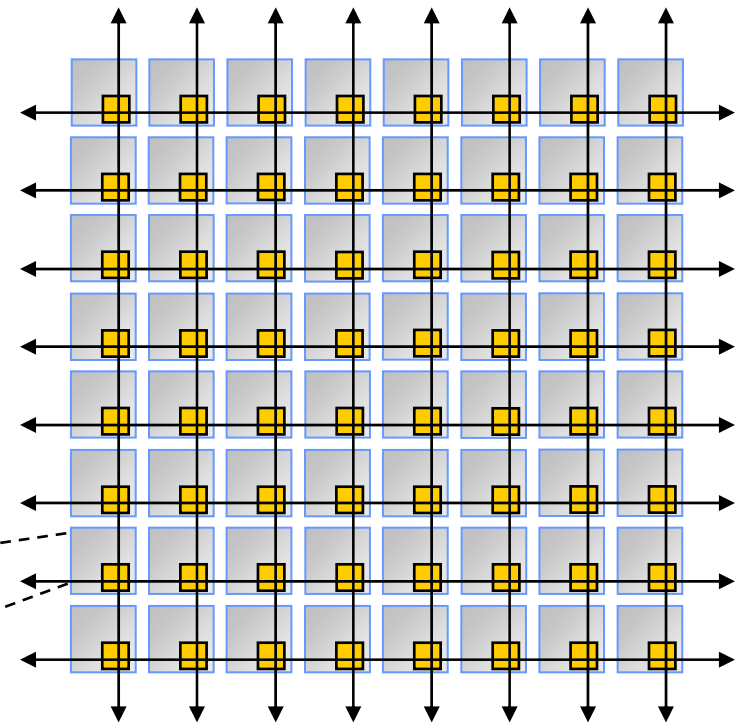
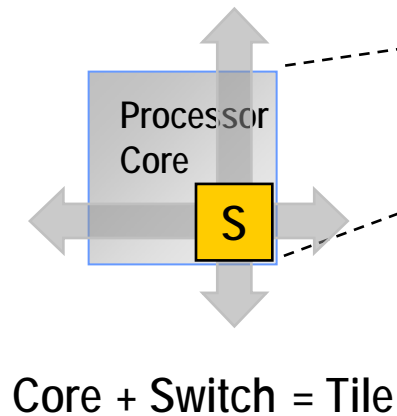


Inherent architectural bottlenecks:

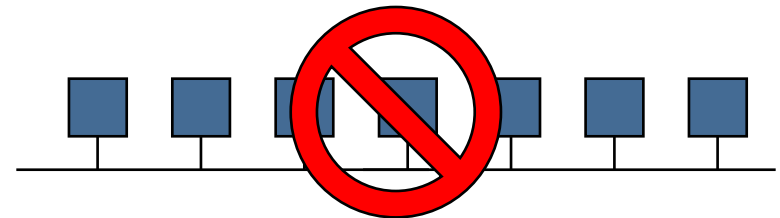
- No scalability
- Power inefficiency
- Primitive programming model

Tiled Multicore Closes the Performance Gap

- Cores connected by mesh network
- Unlike buses, meshes scale
- Resources are distributed
 - improved power efficiency
- Modular – easy to layout and verify



Current Bus Architecture



The TILE64™ Processor

Multicore Performance (90nm)

Number of tiles (general purpose cores)	64
On chip distributed cache	5 MB
Operations @ 750MHz (32, 16, 8 bit)	144-192-384 BOPS
On chip interconnect bandwidth	32 Terabits per second
Bisection bandwidth	2 Terabits per second

Power Efficiency

Power per tile	170 – 300 mW
Clock speed	600-1000 MHz

I/O and Memory Bandwidth

I/O bandwidth	40 Gbps
Main Memory bandwidth	200 Gbps

Programming

ANSI standard C
SMP Linux programming
Stream programming

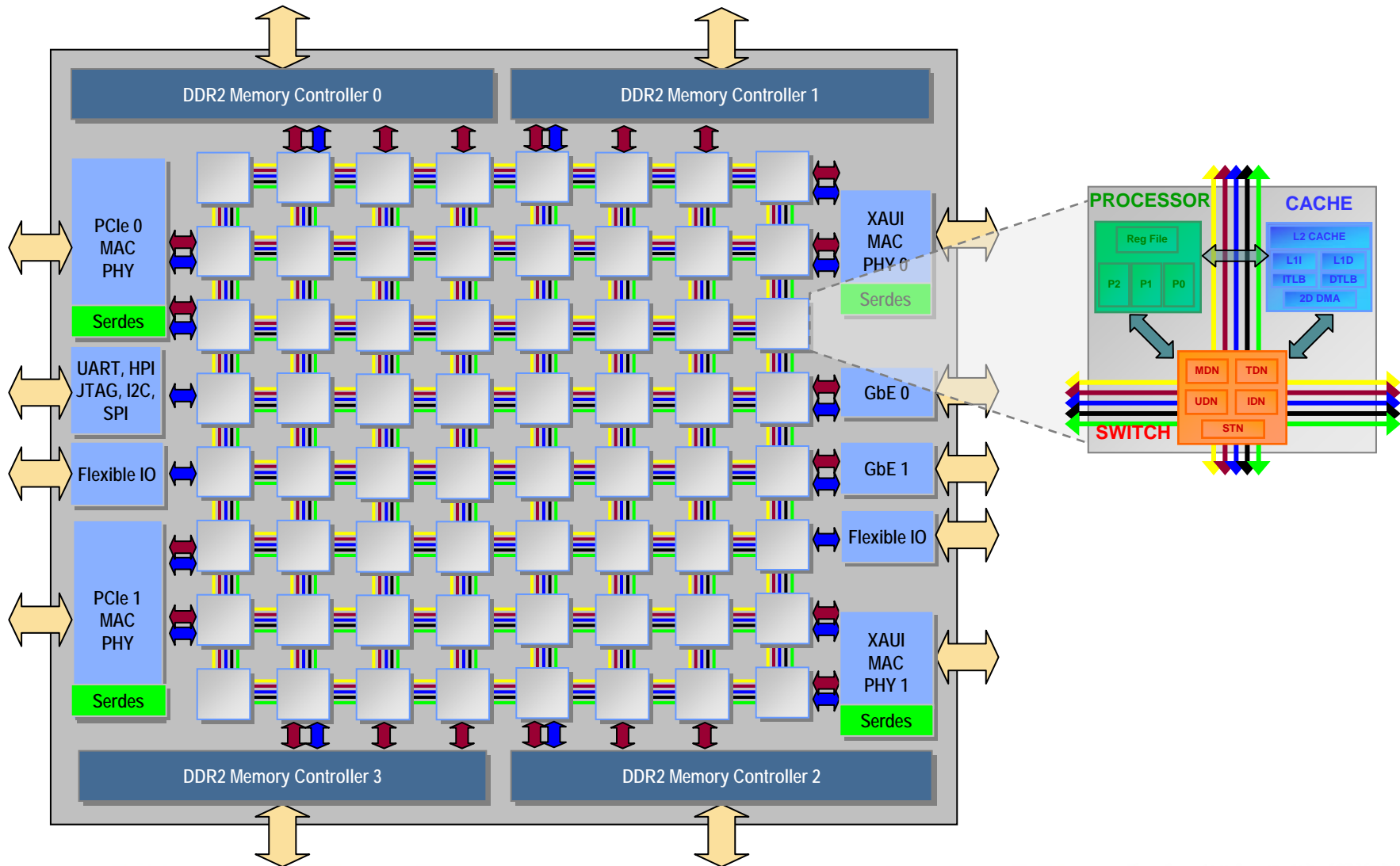


The TILE64 chip is shipping today



TILE64 Processor Block Diagram

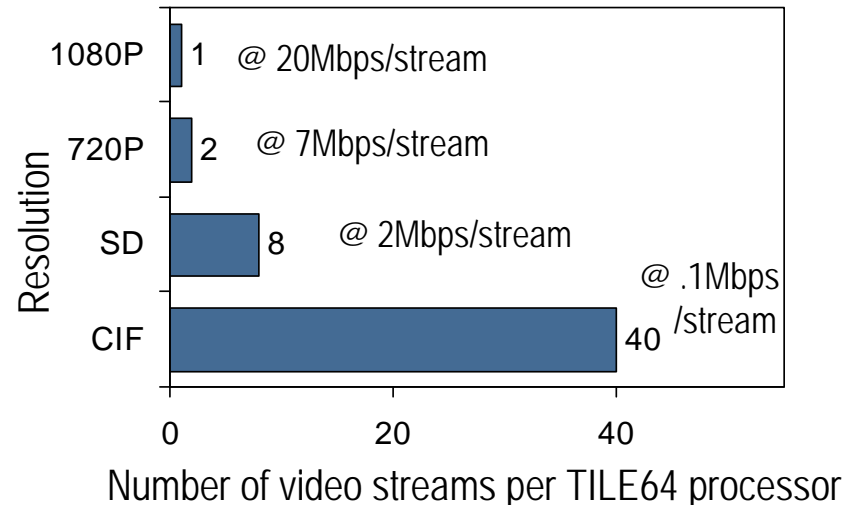
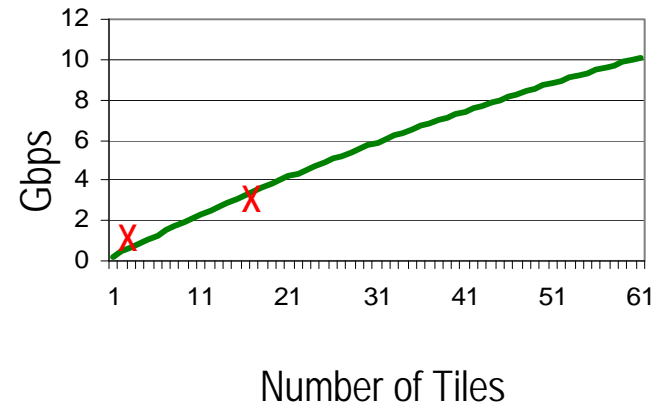
A Complete System on a Chip



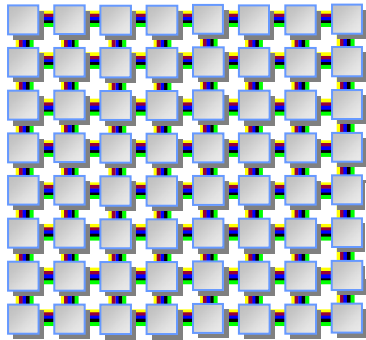
Performance in Networking and Video

- Performance in networking
 - 10Gbps of SNORT
 - Complete SNORT database
 - All SNORT pre-processors
 - Customer's real world data
 - Open source SNORT software base
- Performance in video
 - H.264 video encode
 - Encodes 40 CIF video streams @ 30fps
 - Encodes two 720p HD streams @ 30fps
 - PSNR of 35 or more
 - Open source X264 software base

Performance on a single TILE64 Processor
vs. other multicore solutions



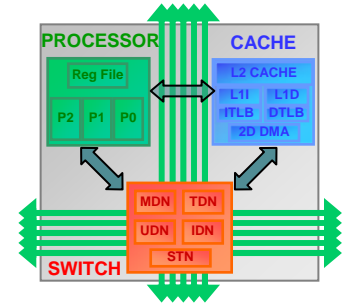
Key Innovations



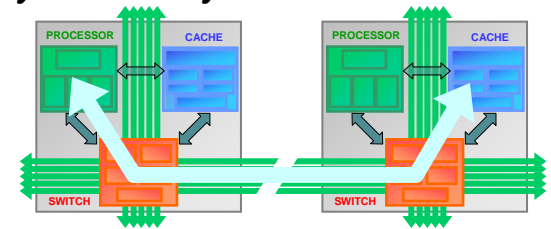
1. iMesh™ Network
– How to scale



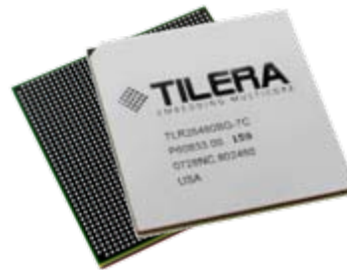
2. General purpose cores
– How to balance core size & number of cores



3. Multicore Coherent Cache
– How to obtain both cache capacity and locality



4. Multicore Hardwall™
– How to virtualize multicore

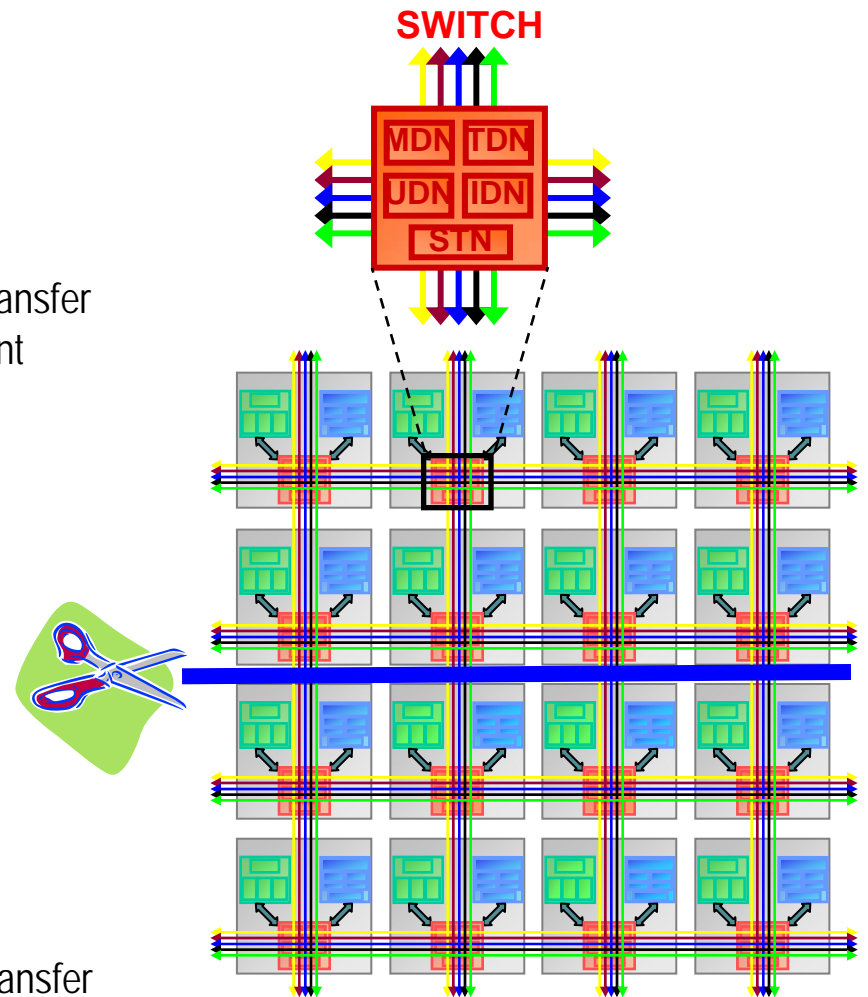


5. Multicore Development Environment
– How to program

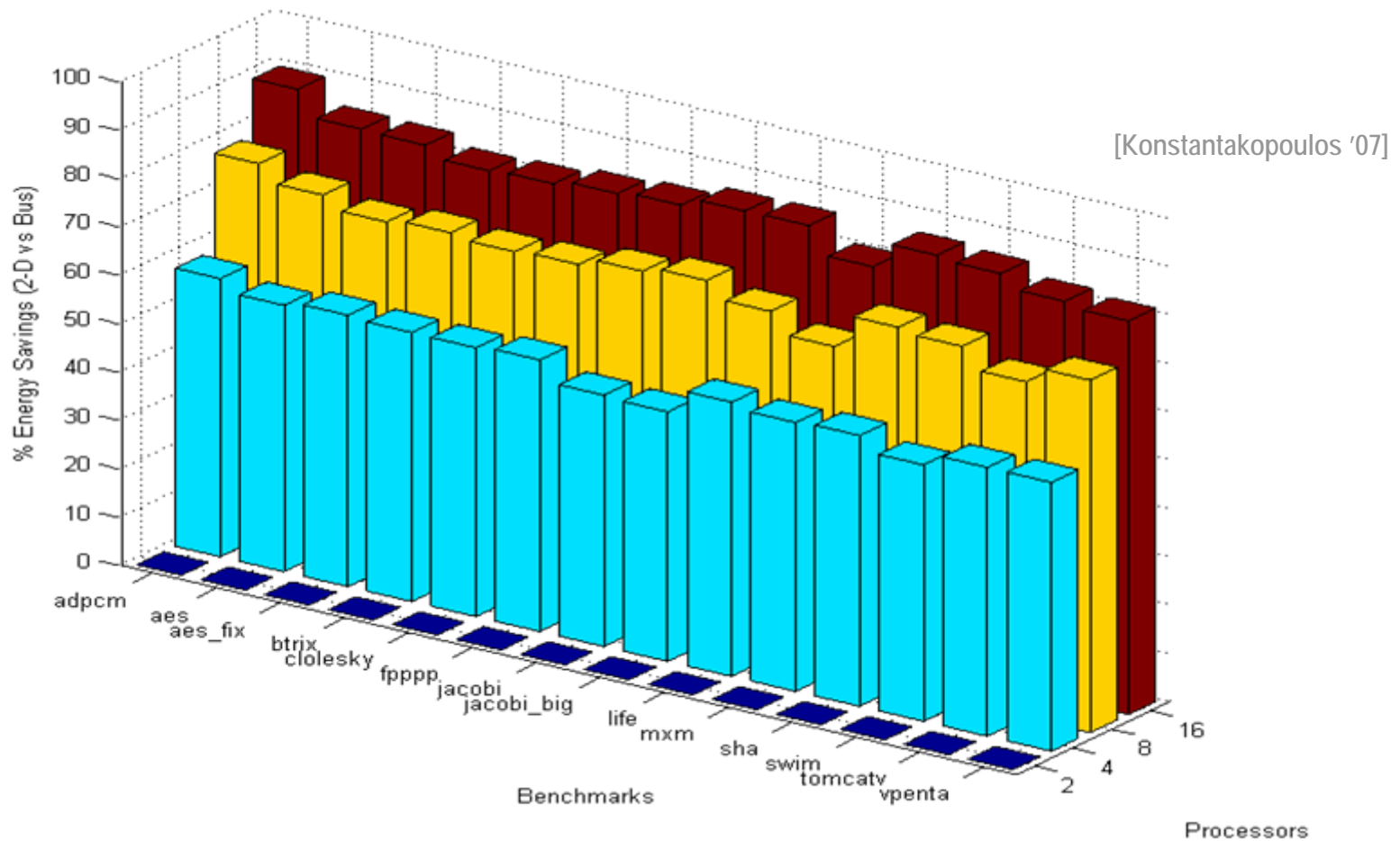


1- iMesh On-Chip Network Architecture

- 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-I/O data transfer
 - Packet switched, wormhole routed, point-to-point
 - Near-neighbor flow control
 - Dimension-ordered routing
- Performance
 - ASIC-like one cycle hop latency
 - 2 Tbps bisection bandwidth
 - 32 Tbps interconnect bandwidth
- 5 independent networks
 - One static, four dynamic
 - IDN – System and I/O
 - MDN – Cache misses, DMA, other memory
 - TDN – Tile to tile memory access
 - UDN, STN – User-level streaming and scalar transfer



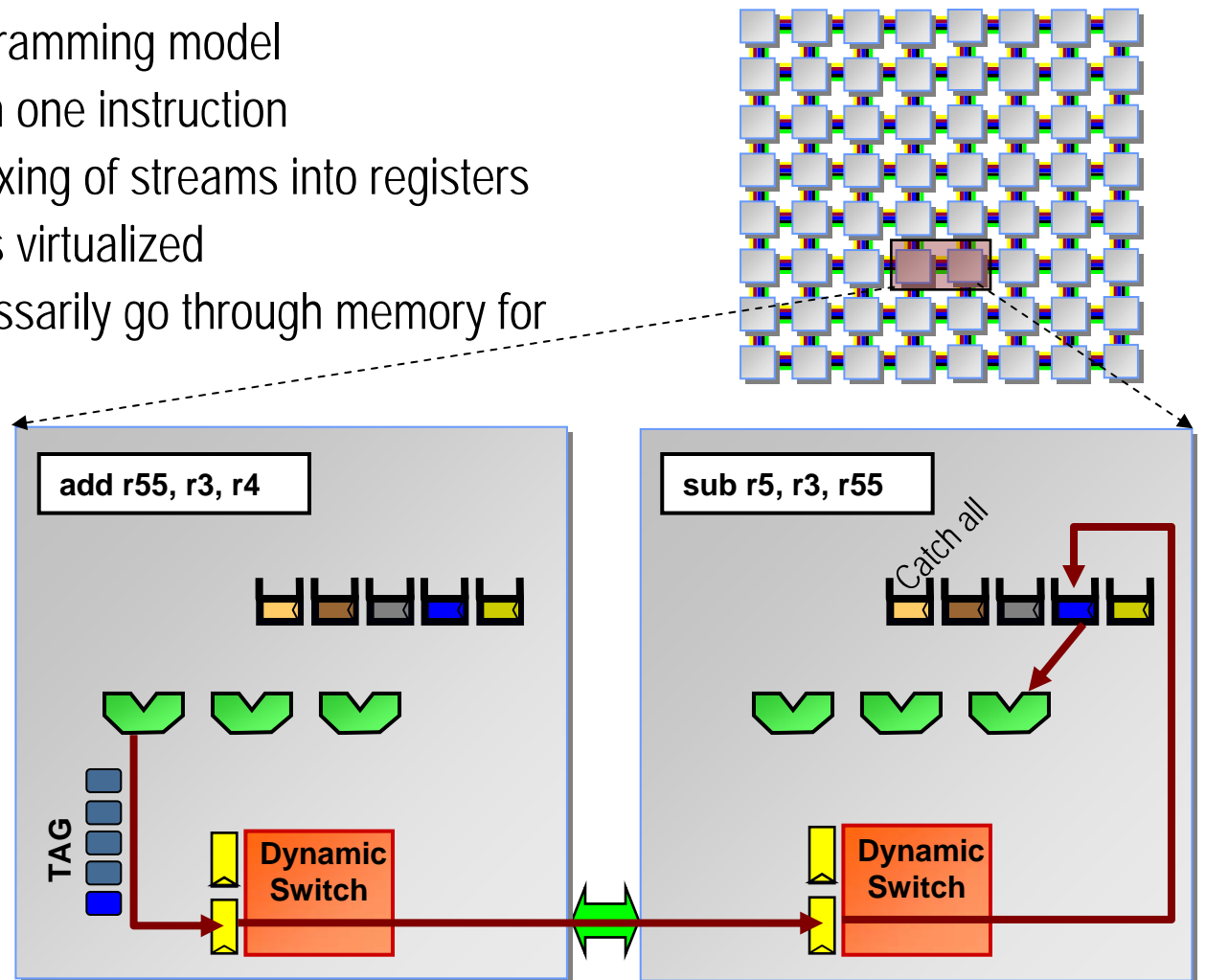
Meshes are Power Efficient



More than 80% power savings over buses

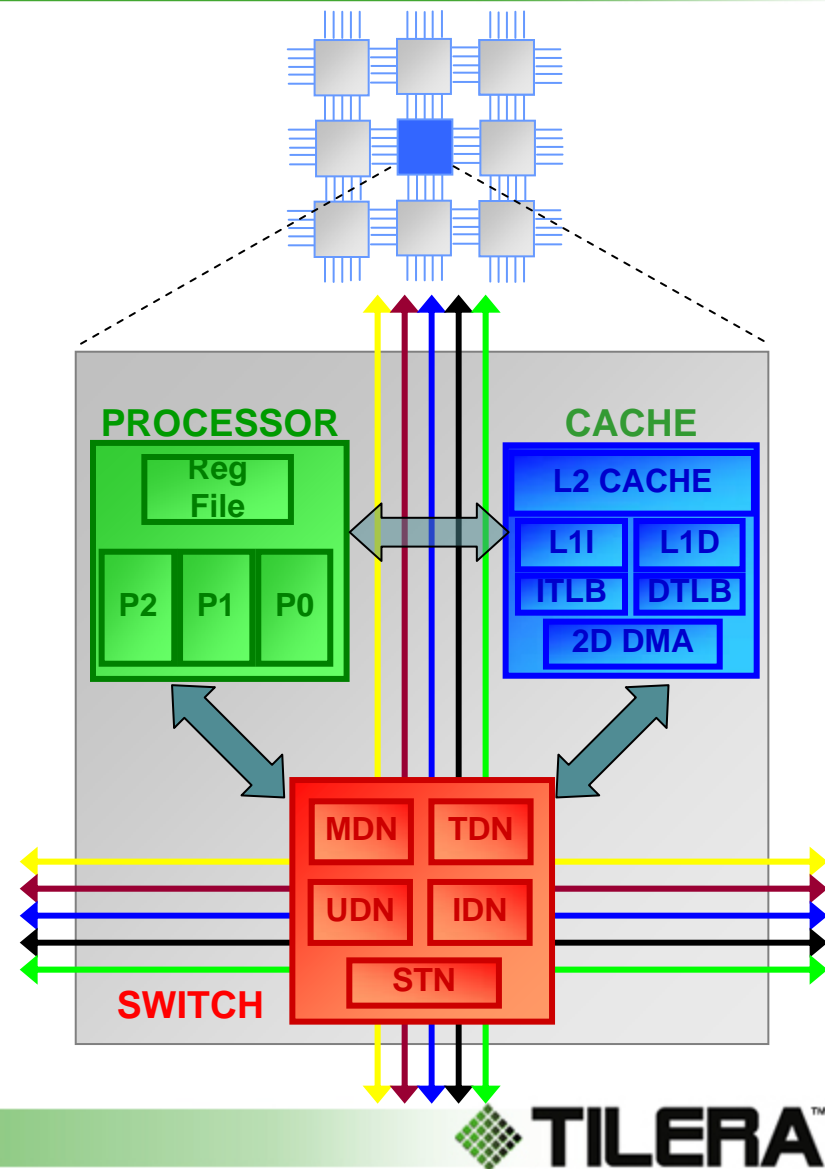
Direct User Access to Interconnect

- Enables stream programming model
- Compute and send in one instruction
- Automatic demultiplexing of streams into registers
- Number of streams is virtualized
- Streams do not necessarily go through memory for power efficiency



2- Full-Featured General Purpose Cores

- Processor
 - Homogeneous cores
 - 3-way VLIW CPU, 64-bit instruction size
 - SIMD instructions: 32, 16, and 8-bit ops
 - Instructions for video (e.g., SAD) and networking (e.g., hashing)
 - Protection and interrupts
- Memory
 - L1 cache: 8KB I, 8KB D, 1 cycle latency
 - L2 cache: 64KB unified, 7 cycle latency
 - Off-chip main memory, ~70 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
- 7 BOPS/watt



5- Multicore Software Tools and Programming

- Arguably biggest multicore challenge
- Multicore software tools challenge
 - Current tools are primitive – use single process based models
 - E.g., how do you single-step an app spread over many cores
 - Many multicore vendors do not even supply tools
- Multicore programming challenge
 - Key tension between getting up and running quickly using familiar models, while providing means to obtain full multicore performance
 - How do you program 100—1000 cores?
 - Intel Webinar likens threads to the “Assembly of parallel programming” – but familiar and still useful in the short term for small numbers of cores
 - Need a way to transition smoothly from today’s programming to tomorrow’s

Tilera's Approach to Multicore Tools: Spatial Views and Collectives

Grid view

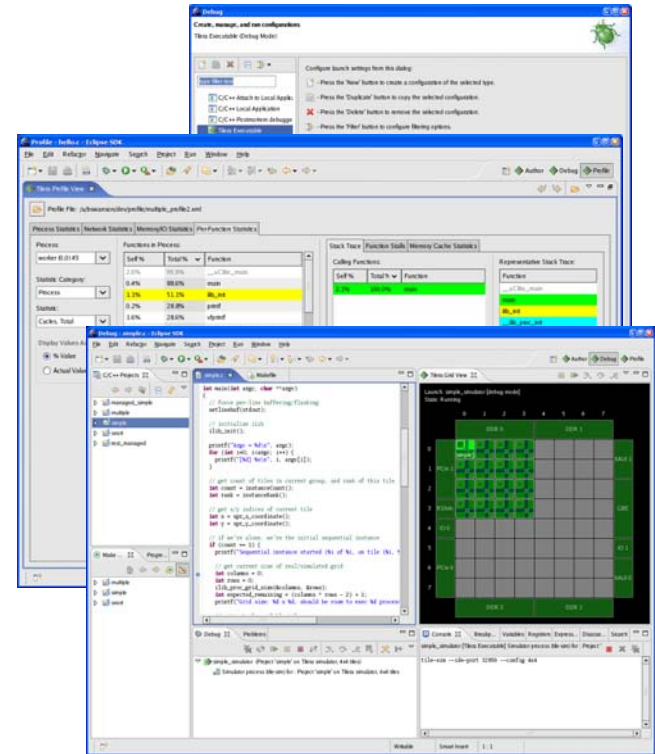
- Provides spatial view
- For selecting single process or region
- Eclipse based

Multicore Debugger

- GDB standard based -- familiar
- Aggregate control and state display
- Whole-application model for collective control
- Low skid breakpointing of all related processes

Multicore Profiler

- Collective stats
- Aggregate over selected tiles



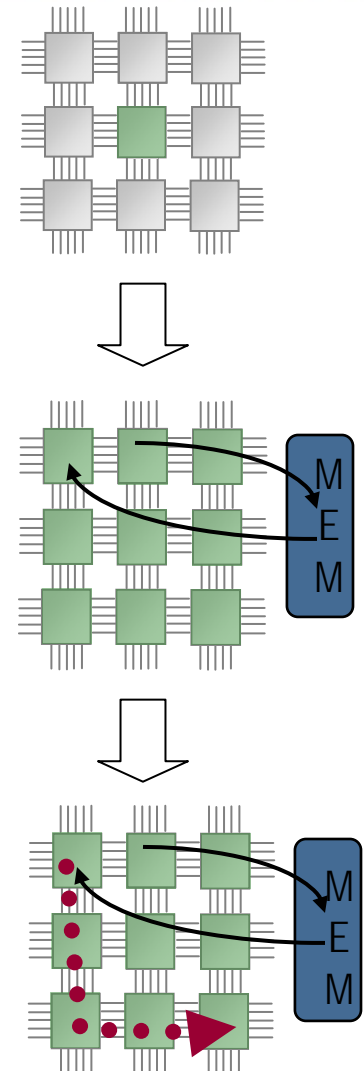
Gentle Slope Programming Model

Gentle slope programming philosophy

- Facilitates immediate results using off-the-shelf code
- Incremental steps to reach performance goals

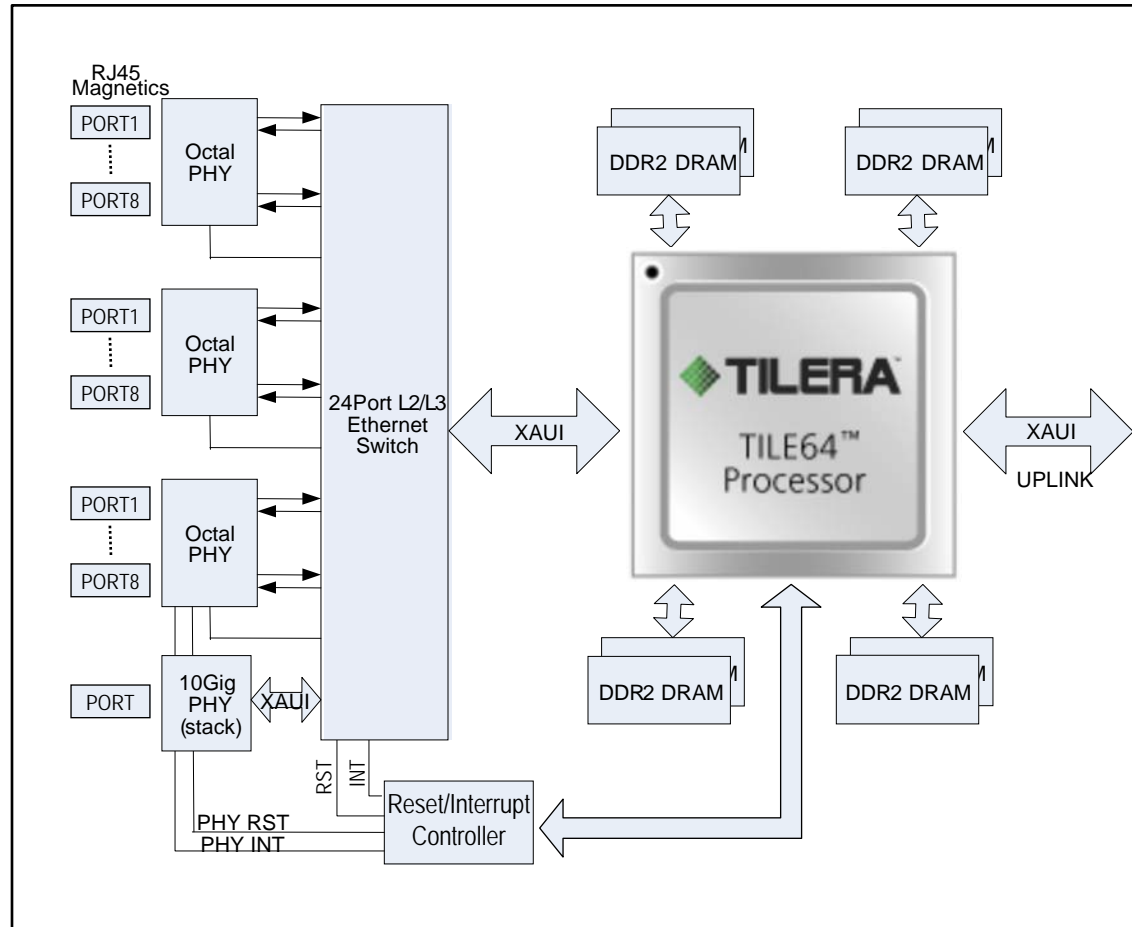
Three incremental steps

- Compile and run standard C applications on a single tile
- Run the program in parallel using standard SMP Linux models – pthreads or processes
- Use stream programming using iLib – a light-weight sockets-like API



Example System Design

Intelligent Switch Design



Summary

- Current multicores face software and scalability challenges
- iMesh network based Tile Processor scales to many cores
- Gentle slope programming offers:
 - Convenience of SMP Linux/threads programming model
 - Performance scalability through streaming channels
- TILE64 silicon, software tools, and applications deployed in customers' systems

Additional Information

PSNR: Peak signal to noise ratio
MDN: Memory dynamic network
UDN: User dynamic network
TDN: Tile dynamic network
IDN: I/O dynamic network
STN: Static network

The following are trademarks of Tiler Corporation: Tiler, the Tiler Logo, Tile Processor, TILE64, Embedding Multicore, Multicore Development Environment, Gentle Slope Programming, iLib, iMesh and Multicore Hardwall. All other trademarks and/or registered trademarks are the property of their respective owners.

© Copyright 2007 Tiler Corporation