LESSONS LEARNED FROM THE 80-CORE TERA-SCALE RESEARCH PROCESSOR

Contributors

Saurabh Dighe
Intel Corporation

Sriram Vangal
Intel Corporation

Nitin Borkar
Intel Corporation

Shekhar Borkar
Intel Corporation

Index Words

TeraFLOPS
Many-core
80-tile
Network-on-chip
2D Mesh Network

Abstract

Sustained tera-scale-level performance within an affordable power envelope is made possible by an energy-efficient, power-managed simple core, and by a packet-switched, two-dimensional mesh network on a chip. From our research, we learned that (1) the network consumes almost a third of the total power, clearly indicating the need for a new approach, (2) fine-grained power management and low-power design techniques enable peak energy efficiency of 19.4 GFLOPS/Watt and a 2X reduction in standby leakage power, and (3) the tiled design methodology quadruples design productivity without compromising design quality.

Introduction

Intel's Tera-scale Research Computing Program [1] lays out a vision for future computing platforms and underscores the need for tera-scale performance. We envision hundreds of networked cores running complex parallel applications under a highly constrained energy budget. Consequently, one of the important research areas in this initiative is to develop a scalable tera-scale processor architecture that can address the needs of our future platforms. The Teraflops Research Processor is a key first step in this direction. Focusing on some of the vital ingredients of a tera-scale architecture: a power optimized core, a scalable on-chip interconnect, and a modular global clocking solution, we established the following research goals for our project:

- Achieve teraFLOPS performance under 100 W.
- Prototype a high-performance and scalable on-chip interconnect.
- Demonstrate an energy-efficient architecture with fine-grained power management.
- Develop design methodologies for network-on-chip architectures (NoC).

Our intent in this article is to focus on key lessons we learned from the research prototype. We present our findings in a structured format. We first provide an overview of the chip and briefly describe key building blocks. We then highlight the novel design techniques implemented on the chip and the tiled design approach. Next, we summarize measured silicon results. Finally, we discuss the pros and cons of certain design decisions, including our recommendations for future tera-scale platforms.

Architecture Overview

Rapid advancement in semiconductor process technology and a quest for increased energy efficiency have fueled the popularity of multi-core and NoC architectures [2]. The teraFLOPS research processor contains 80 tiles arranged as an 8 x 10, 2-D mesh network, shown in Figure 1. Each tile consists of a processing engine (PE) connected to a 5-port router with mesochronous interfaces (MSINT), which forwards packets between the tiles. More detailed information on the chip architecture and interconnect can be found in [3, 4].

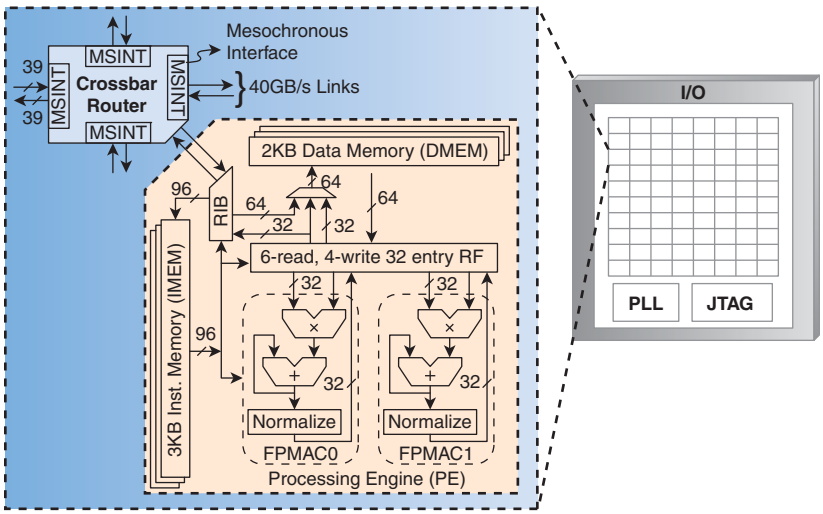


Figure 1: 80-core Processor Tile Architecture
Source: Intel Corporation, 2009

Processing Engine

The PE contains two independent fully-pipelined, single-precision, floating-point multiply-accumulator (FPMAC) units capable of providing an aggregate performance of 20 GFLOPS. The key to achieving this high performance is a fast, single cycle, accumulation algorithm [5], developed by analyzing each of the critical operations involved in conventional floating point units (FPUs) with the intent of eliminating, reducing, or deferring the amount of logic operations inside the accumulate loop.

We came up with the following three optimizations. First, the accumulator retains the multiplier output in carry-save format and uses an array of 4-2 carry-save adders to accumulate the results in an intermediate format. This removes the need for a carry-propagate adder in the critical path. Second, accumulation is performed in base 32, converting expensive variable shifters in the accumulate loop to constant shifters. Third, we moved the costly normalization step outside the accumulate loop, where the accumulation result in carry-save is added, and the sum is normalized and converted back to base 2. These optimizations allow accumulation to be implemented in just fifteen FO4 stages. This approach also reduces the latency of dependent FPMAC instructions and enables a sustained multiply-add result (2FLOPS) every cycle. Moving to 64-bit arithmetic results in wider mantissa for increased throughput.

"The PE contains two independent fully-pipelined, single-precision units capable of providing an aggregate performance of 20 GFLOPS."

"The 80-tile, on-chip network is a 2D mesh that provides a bisection bandwidth of 2 Terabits/s."

The PE includes a 3-KB, single-cycle, instruction memory (IMEM) and 2KB of data memory (DMEM). This amounts to a total distributed on-die memory of 400 KB. The capacity of the local memory was enough to support blocked execution of a select few LAPACK kernels. With a 10-port (6-read, 4-write) register file, we allow scheduling to both FPMACs, simultaneous DMEM load/store, and packet send/receive from the mesh network. A router interface block (RIB) handles packet encapsulation between the PE and router.

On-chip Interconnect

The 80-tile, on-chip network is a 2D mesh that provides a bisection bandwidth of 2 Terabits/s. The key communication block for the NoC is a 5-port, pipelined, packet-switched router with two virtual lanes (see Figure 2) capable of operating at 5 GHz [6] at a nominal supply of 1.2 V. It has a 6-cycle latency or 1.2 ns/hop at 5 GHz. It connects to each of its neighbors and the PE by using phase-tolerant mesochronous links that can deliver data at 20 GBytes/sec. The network uses a source-directed routing scheme, based on wormhole switching, that has two virtual lanes for dead-lock free routing and an on-off scheme, by using almost-full signals for flow control. The width of the links and router frequency were chosen to transfer a single precision FPU operand at high speed and approximately 1 ns/hop latency.

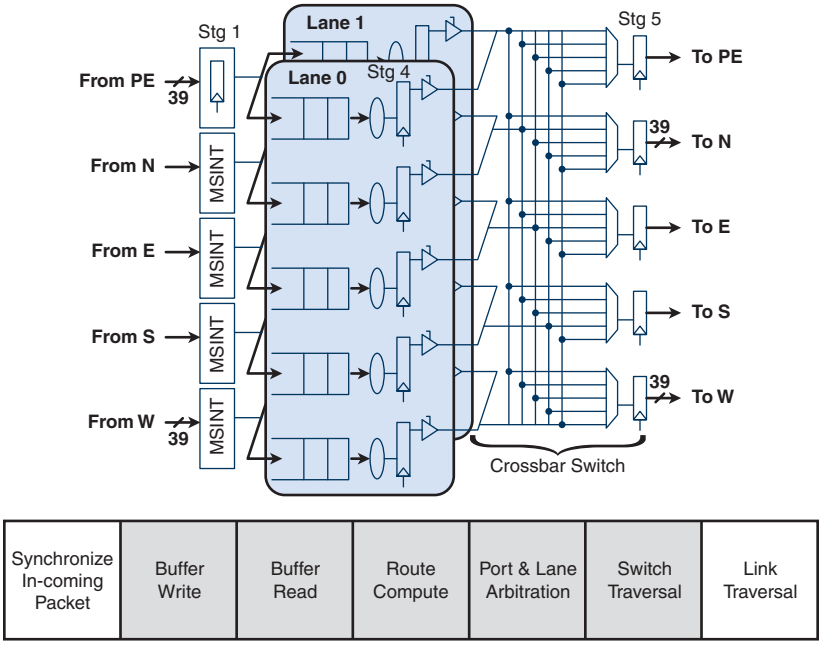


Figure 2: 5-port Two-lane Shared Crossbar Router Architecture
Source: Intel Corporation, 2009

Instruction Set and Programming Model

We define a 96-bit Very Long Instruction Word (VLIW) that allows a maximum of up to eight operations to be issued every cycle. The instructions fall into one of five categories: instruction issue to both floating-point units, simultaneous data memory load and stores, packet send/receive via the on-die mesh network, program control that uses jump and branch instructions, and synchronization primitives for data transfer between PEs. With the exception of FPU instructions, which have a pipeline latency of nine cycles, most other instructions execute in one to two cycles.

"We define a 96-bit Very Long Instruction Word (VLIW) that allows a maximum of up to eight operations to be issued every cycle."

To aid with power management, the instruction set includes support for dynamic sleep and wakeup of each floating-point unit. The architecture allows any PE to issue sleep packets to any other tile or to wake it up for processing tasks.

The architecture supports a message-passing programming model by providing special instructions to exchange messages to coordinate execution and share data [7]. The fully symmetric architecture allows any PE to send or receive instructions and data packets to or from any other tile.

Novel Circuit and Design Techniques

We used several circuit techniques to achieve high performance, low power, and a short design cycle. The fifteen FO4 design uses a balanced core and router pipeline, with critical stages employing performance-setting, semi-dynamic flip-flops. In addition, a robust scalable mesochronous clock distribution is employed in a 65-nanometer, 8-metal CMOS process that enables high integration and single-chip realization of the teraFLOP processor.

"With the exception of FPU instructions, which have a pipeline latency of nine cycles, most other instructions execute in one to two cycles."

Circuit Design Style

To enable a 5-GHz operation, we designed the entire core by using hand-optimized datapath macros. For quick turnaround we used CMOS static gates to implement most of the logic. However, critical registers in the FPMAC and at the router crossbar output utilize implicit-pulsed, semi-dynamic flip-flops (SDFF) [8, 9], which have a dynamic master stage coupled with a pseudostatic slave stage. When compared to a conventional static, master-slave flip-flop, SDFF provides both shorter latency and the capability of incorporating logic functions, with minimum delay penalty, each of which are desirable properties in high-performance digital designs. However, pulsed flip-flops have several important disadvantages. The worst-case hold time of this flip-flop can exceed clock-to-output delay because of pulse width variations across process, voltage, and temperature conditions. Therefore, pulsed flip flops must be carefully designed to avoid failures due to min-delay violations.

"To enable a 5-GHz operation, we designed the entire core by using hand-optimized datapath macros."

"We used fine-grained clock gating and sleep transistor circuits to reduce active and standby leakage power, which are controlled at full-chip, tile-slice, and individual tile levels, based on workload."

Fine grain Power Management

To achieve the goal of demonstrating teraFLOPS performance below 100 watts of power, we had to adopt and combine various power-saving features and use innovative power-management technologies. To this end, we used fine-grained clock gating and sleep transistor circuits [10] to reduce active and standby leakage power, which are controlled at full-chip, tile-slice, and individual tile levels, based on workload. Figure 3 shows clock and power gating in the FPMAC, router, and instruction/data memories. Approximately 90 percent of FPU logic and 74 percent of each PE is sleep enabled. Each tile is partitioned into twenty-one smaller sleep regions, and dynamic control of individual blocks is based on instruction type. Each FPMAC can be controlled through NAP/WAKE instructions. The router is partitioned into ten smaller sleep regions, and control of individual router ports depends on network traffic patterns. We inserted sleep transistors in the register file cells without impacting area too much. An additional track had to be used to route the sleep signal. Special attention was paid to sleep-non-sleep interfaces, and intelligent data gating at flip-flop boundaries ensured additional firewall circuits were not required.

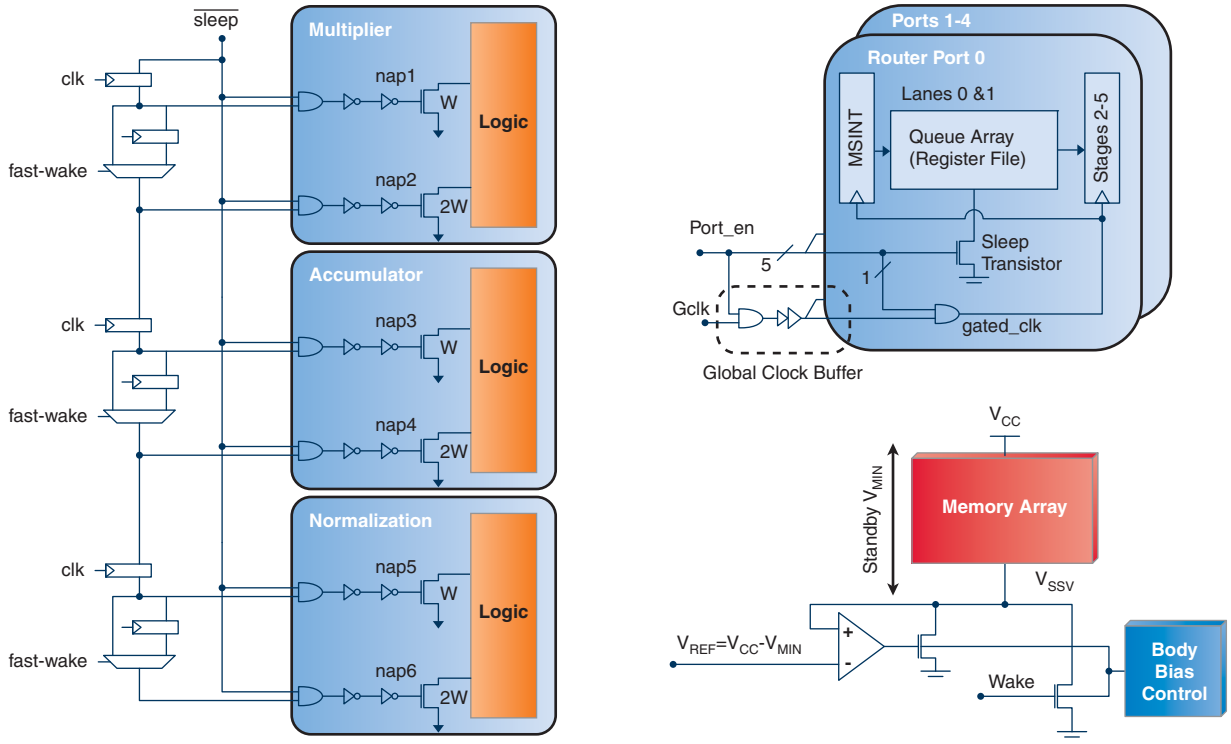


Figure 3: Fine-grain Power Management in the Tile
Source: Intel Corporation, 2009

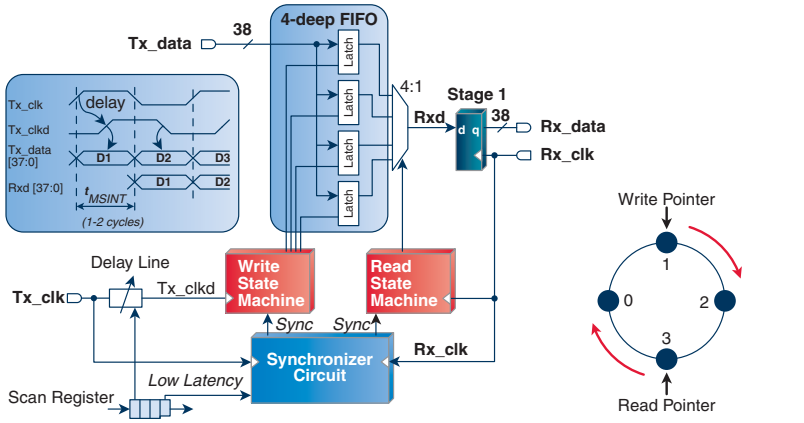


Figure 4: Phase-tolerant Mesochronous Interface
Source: Intel Corporation, 2009

Mesochronous Clocking

The chip uses a scalable global mesochronous clocking technique, that allows for clock-phase-insensitive communication across tiles and for synchronous operation within each tile. The on-chip PLL output is routed by using horizontal (Metal-8) and vertical (Metal-7) spines. Each spine consists of differential clocks for low duty-cycle variation along the worst-case clock route of 26 mm. An op-amp at each tile converts the differential clock inputs to a single-ended clock with a 50 percent duty cycle, prior to distribution, by using an H-tree. The 2-mm long point-to-point, unidirectional router links implement a phase-tolerant, mesochronous interface as shown in Figure 4. This allows clock-phase-insensitive communication across tiles and enables a scalable, on-die communication fabric that simplifies global clock distribution.

Double-pumped Crossbar

The crossbar switch area increases as a square function $O(n^2)$ of the total number of I/O ports and the number of bits per port. Consequently, the crossbar can dominate a large percentage of the area. To alleviate this problem, we double pump the crossbar data buses by interleaving alternate data bits as shown in Figure 5. We use dual-edge triggered flip-flops to do this; thereby, effectively reducing by half the crossbar hardware cost.

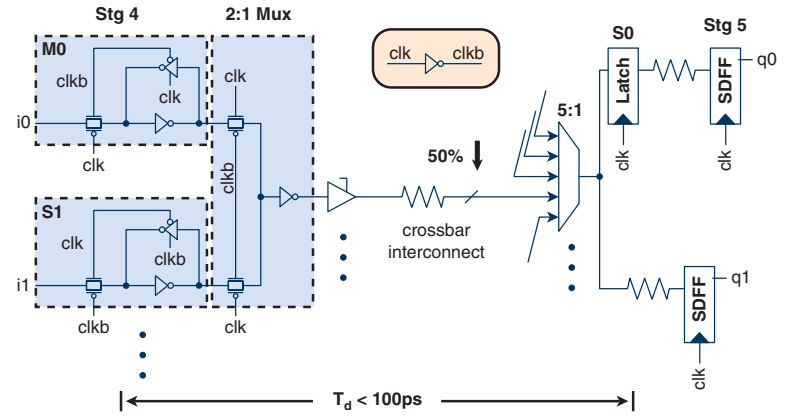


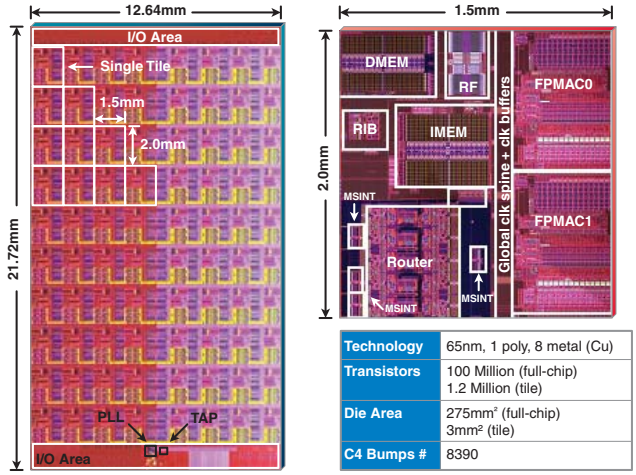
Figure 5: Double-pumped Crossbar
Source: Intel Corporation, 2009

"Each tile is completely self-contained, including power bumps, power tracks, and global clock routing."

Tiled-design Methodology

While implementing the teraFLOPS processor, we followed a "tiled design methodology" where each tile is completely self-contained, including power bumps, power tracks, and global clock routing. This design enabled us to seamlessly array all tiles at the top level, by simply using abutment. This methodology enabled rapid completion of a fully custom design with less than 400 person-months of effort.

Results



Technology	65nm, 1 poly, 8 metal (Cu)
Transistors	100 Million (full-chip) 1.2 Million (tile)
Die Area	275mm² (full-chip) 3mm² (tile)
C4 Bumps #	8390

Figure 6: Full-chip and Tile Micrograph and Characteristics
Source: Intel Corporation, 2009

We fabricated the teraFLOPS processor in 65-nm process technology. The die photographs in Figure 6 identify the chip's functional blocks and individual tiles. The 275-mm², fully custom design contains 100 million transistors. The chip supports a wide dynamic range of operation; namely, 1 GHz at 670 mV up to 5.67 GHz at 1.35 V, as shown in Figure 7. Increased performance with higher voltage and frequency can be achieved at the cost of power. As we scale Vcc/frequency, power consumption ranges from 15.6 W to 230 W as shown in Figure 8. Fine-grain sleep transistors limit the leakage power from 9.6 percent to 15.6 percent of the total power. With all 80 tiles actively performing single-precision, block-matrix operations, the chip achieves a peak performance of 1.0 TFLOPS at 3.16 GHz while dissipating 97 W. By reducing voltage, and by operating close to the threshold voltage of the transistor, energy efficiency for the stencil application can be improved from 5.8 GFLOPS per Watt to a maximum of 19.4 GFLOPS per Watt as shown in Figure 9.

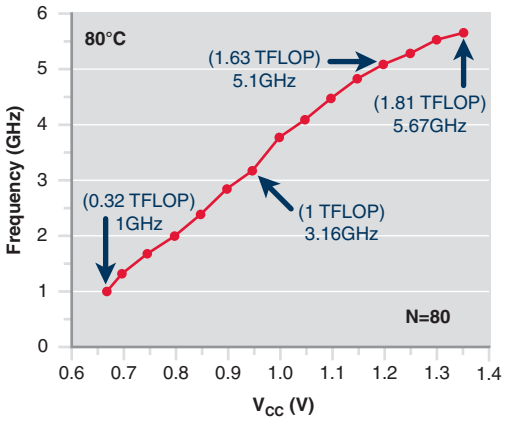


Figure 7: Measured Chip Fmax and Peak Performance
Source: Intel Corporation, 2009

The measured global clock distribution power is 2 W at 1.2 V and 5.1 GHz operation, and it accounts for just 1.3 percent of the total chip power. At the tile level, power breakdown shows that the dual FPMACs account for 36 percent of total power, the router and links account for 28 percent, the IMEM and DMEM account for 21 percent, the tile-level synchronous clock distribution accounts for 11 percent, and the multi-ported register file accounts for 4 percent. In sleep mode, the nMOS sleep transistors are turned off, reducing chip leakage by 2X, while preserving the logic state in all memory arrays. Total network power per tile can be lowered from a maximum of 924 mW with all router ports active to 126 mW, resulting in a 7.3X reduction. The network leakage power per tile when all ports and global clock buffers to the router are disabled is 126 mW. This number includes power dissipated in the router, MSINT, and in the links.

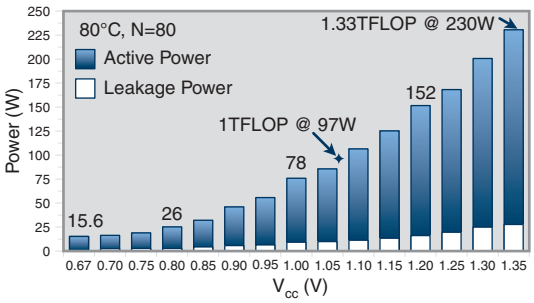


Figure 8: Measured Power Versus Vcc for Stencil Application
Source: Intel Corporation, 2009

Discussion and Tradeoffs

The goal of achieving teraFLOPS performance under 100 W entails studying the traditional tradeoffs between performance, power, and die size, but equally important are looking at issues such as multi-generation scalability, modular design/validation, and support for parallel programming models.

Today's general-purpose cores are capable of performance in the order of tens of GFLOPS. However, achieving teraFLOPS performance with these cores on the current process technology is prohibitive, from an area and power perspective. Our work corroborates that a computational fabric built by using programmable, special-purpose cores provides high levels of performance in an energy-efficient manner. Power-optimized fast computation hardware, simple decoded VLIW instruction words, and low-power memories ensure that a large percentage of the energy consumed goes towards computing FLOPS. While architecting the core we were aware of the importance of balancing data memory bandwidth with compute/communication bandwidth, which entailed adding a single cycle, 6-read, 4-write register file. As data transfer on chip costs significant energy, larger caches will be required to keep the data local. Maintaining coherency across many cores is a significant challenge as well. Hardware costs and increased coherency traffic on the mesh will pose hurdles for completely hardware-based coherent systems. Instead, future tera-scale processors will explore message-passing architectures. Special on-die, message-passing hardware is very efficient for core-to-core communication, making software-based coherency with hardware assists a viable solution for the future. In addition to support for message passing, another enhancement that proved important is the ability to overlap compute and communication. A core can directly transfer instructions/data into the local memory of another core without interrupting the other core. This resulted in improved FPMAC utilization with fewer idle cycles and enabled performance numbers that were close to the maximum achievable.

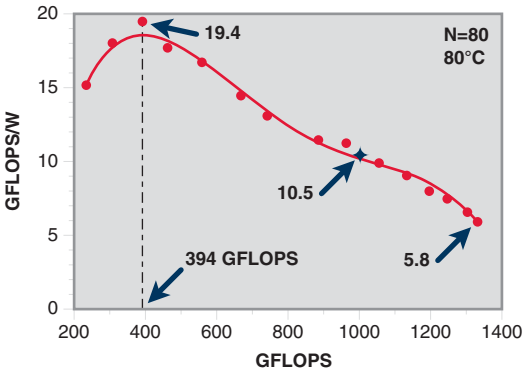


Figure 9: Measured Chip Energy Efficiency for Stencil Application
Source: Intel Corporation, 2009

With increasing demand for interconnect bandwidth, on-chip networks are taking up a substantial portion of the system's power budget. The router on our teraFLOPS processor consumes 28 percent of tile power. Our goal for a compelling solution is to use compact low-power routers that consume less than 10 percent of the chip power and die budget. At the same time they must deliver high, on-die bisection bandwidth and low latency. Techniques, such as speculation and bypass, are well known, but they add to the power consumption and are therefore undesirable. Future routers would also need to incorporate extensive fine-grained, power-management techniques to enable dynamic operation that adapts to differing traffic patterns. Heterogeneous NoCs [11], that allocate resources as needed, and circuit-switched networks [12, 13] are promising approaches. Traffic patterns and bandwidth requirements are going to dictate on-die network architectures for the future. Hybrid approaches to on-die networks can save communication power by utilizing fewer fully-connected crossbar routers at the expense of reduced bandwidth. Instead of one router per core in each tile, we could amortize the power/area of the router by having two or more cores on a shared bus connected to the local port of the router in each tile.

"The router on our teraFLOPS processor consumes 28 percent of tile power. Our goal for a compelling solution is to use compact low-power routers that consume less than 10 percent of the chip power."

The two popular clocking techniques for on-die networks are 1) a completely synchronous system with closely matched skews, and 2) a globally asynchronous, locally synchronous system with handshaking signals for data transfer (GALS). Synchronous systems are the simplest to implement and are well understood, but they can consume significant power for high-frequency clock distribution. With increased with-in die variation, matching skews across large dies is becoming difficult, which also results in excessive timing guard bands. GALS suffers from area overhead, due to additional hand-shaking circuits, lack of mature design tools, and increased design complexity. The mesochronous clocking scheme tries to address these problems by distributing a single frequency clock without the overhead of matching clock skews. This causes phase differences between clocks, distributed to individual routers that need to be accounted for by synchronization circuitry in the data paths. This technique scales well as tiles are added or removed. Multiple cycles are required for the global clock to propagate to all 80 tiles; this systematic skew inherent in the distribution helps spread peak currents because of simultaneous clock switching. To support mesochronous or phase-tolerant communication across tiles, we pay a synchronization latency penalty for the benefit of a lightweight global clock distribution. The area and power overhead of the synchronizers can be significant for wide links. It is important to understand these tradeoffs before abandoning a synchronous implementation in favor of mesochronous clocking.

Tera-scale computing platforms need to be efficient to meet the energy constraints of future data centers. We employed per-tile, fine-grained power management with clock and power gating. By exposing WAKE/NAP instructions to software, we could put FPMACs to sleep during idle windows. This enabled us to reach an energy efficiency of 19.4 GFLOPS/Watt. In stark contrast, a 3-GHz, general-purpose CPU provides an energy efficiency of 0.07 GFLOPS/Watt. As different applications with different compute/communication profiles and performance requirements are invoked over time, the optimal number of cores and Vcc/frequency to achieve maximum energy efficiency varies. Hence, to further improve workload efficiency, we recommend dynamic voltage frequency scaling with independent voltage and frequency islands for future tera-scale processors.

"We reached an energy efficiency of 19.4 GFLOPS/Watt. In stark contrast, a 3-GHz, general-purpose CPU provides an energy efficiency of 0.07 GFLOPS/Watt."

To operate across a wide dynamic voltage range it is important to implement circuits with robust static CMOS logic that operate at low voltages. Operating close to threshold voltage of the transistor increases energy efficiency; however, contention circuits in register files and small signal arrays typically limit the lowest operating voltage (Vccmin) of a processor. It is critical for tera-scale processors to operate at the lowest energy point, and this makes research in Vccmin-lowering techniques a vital part of the tera-scale research agenda. Designs should also be optimized for power with extensive usage of low-leakage transistors and selective usage of nominal transistors in critical paths. It is important to strike a balance between delay penalty and leakage savings during device-type selection for sleep transistors. We chose to utilize nominal devices for 5-GHz operation with a 2X leakage savings.

As we integrate more cores on a single die, adopting a scalable design methodology is critical for design convergence, validation, product segmentation, and time-to-market. The proposed tiled design methodology enabled faster convergence in timing verification and physical design. Global wires that do not scale well with technology could be avoided. We ensured the tiles were small, completely self-contained, and could be assembled by abutment. This also ensures uniform metal/via density that helps in manufacturability and yield. Consequently, we achieved high levels of integration with a small design team and low overhead. Pre/post-silicon debug effort was greatly reduced with first silicon stepping fully functional. In addition, a standardized communication fabric with a predefined interface combined with a tiled design approach provides the flexibility of integrating any number of homogenous or heterogeneous cores and facilitates product segmentation.

"The tiles were small, completely self-contained, and could be assembled by abutment."

"Tera-flop performance is possible within a mainstream power envelope."

Conclusion

Tera-flop performance is possible within a mainstream power envelope. Careful co-design at architecture, logic, circuit, and physical design levels pays off, with silicon achieving an average performance of 1 TFLOP at 97 W and a peak power efficiency of 19.4 GFLOPS/Watt. Tile-based methodology fulfilled its promise, and the design was done with half the team in half the time. Communication power accounts for almost one-third of the total power, highlighting the need for further research in low-power, scalable networks that can satisfy the requirements of a tera-scale platform. On a final note, to be able to successfully exploit the computing capability of a tera-scale processor, research into parallel programming is vital.

References

[1] J. Held, J. Bautista, and S. Koehl. "From a few core to many: A tera-scale computing research overview." 2006. Available at *http://www.intel.com*

[2] W.J. Dally and B. Towles. "Route Packets, Not Wires: On-Chip Interconnection Net-works." In *Proceedings 38th Design Automation Conference* (DAC 01), ACM Press, 2001, pages 681-689.

[3] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, C. Roberts, V. Erraguntla, Y. Hoskote, N. Borkar, and S. Borkar. "An 80-Tile Sub-100W TeraFLOPS Processor in 65-nm CMOS." *IEEE Journal of Solid-State Circuits,* vol. 43, pages 29–41, January 2008.

[4] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. "A 5GHz Mesh Interconnect for a Teraflops Processor." *IEEE Micro,* Vol. 27, pages 51-61, 2007.

[5] S. Vangal, Y. Hoskote, N. Borkar and A. Alvandpour. "A 6.2-GFlops Floating-Point Multiply-Accumulator with Conditional Normalization." *IEEE Journal of Solid-State Circuits,* pages 2314–2323, Oct., 2006.

[6] S. Vangal, A. Singh, J. Howard, S. Dighe, N. Borkar, A. Alvandpour. "A 5.1GHz 0.34mm^2 Router for Network-on-Chip Applications." *IEEE Symposium on VLSI Circuits,* 2007. 14-16, June 2007.

[7] T. Mattson, R. Van der Wijngaart, M. Frumkin. "Programming the Intel 80-core network-on-a-chip terascale processor." In *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing,* November 15-21, 2008, Austin, Texas.

[8] F. Klass. "Semi-Dynamic and Dynamic Flip-Flops with Embedded Logic." *1998 Symposium on VLSI Circuits, Digest of Technical Papers,* pages 108–109, 1998.

[9] J. Tschanz, S. Narendra, Z. Chen, S. Borkar, M. Sachdev, V. De. "Comparative Delay and Energy of Single Edge-Triggered & Dual Edge-Triggered Pulsed Flip-Flops for High-Performance Microprocessors." *ISLPED,* pages 147-151, 2001.

[10] J. Tschanz, S. Narendra, Y. Ye, B. Bloechel, S. Borkar and V. De. "Dynamic sleep transistor and body bias for active leakage power control of microprocessors." *IEEE Journal of Solid-State Circuits,* pages 1838–1845, Nov. 2003.

[11] K. Rijpkema et al. "Trade-Offs in the Design of a Router with Both Guaranteed and Best-Effort Services for Networks on Chip." In *IEEE Proceedings On Computers and Digital Techniques,* vol. 150, no. 5, 2003, pages 294-302.

[12] P. Wolkette et al. " An Energy-Efficient Reconfigurable Circuit-Switched Network-on-Chip." In *Proceedings IEEE International Parallel and Distributed Symposium, (IPDS 05), IEEE CS Press,* 2005, pages 155a.

[13] M. Anders, H. Kaul, M. Hansson, R. Krishnamurthy, S. Borkar. "A 2.9Tb/s 8W 64-core circuit-switched network-on-chip in 45nm CMOS." *34th European Solid-State Circuits Conference,* 2008. *ESSCIRC,* pages 182-185, 15-19 Sept. 2008.

Acknowledgements

We thank the entire Advanced Microprocessor Research team at Intel for flawless execution of the chip, we thank Tim Mattson and Rob Van Der Wijngaart for workloads, and Joe Schutz, Greg Taylor, Matt Haycock, and Justin Rattner for guidance and encouragement.

Copyright