Abstract—This paper describes an integrated network-on-chip architecture containing 80 tiles arranged as an 8x10 2-D array of floating-point cores and packet-switched routers, both designed to operate at 4 GHz. Each tile has two pipelined single-precision floating-point multiply accumulators (FPMAC) which feature a single-cycle accumulation loop for high throughput. The on-chip 2-D mesh network provides a bisection bandwidth of 2 Terabits/s. The 15-FO4 design employs mesochronous clocking, fine-grained clock gating, dynamic sleep transistors, and body-bias techniques. In a 65-nm eight-metal CMOS process, the 275 mm² custom design contains 100 M transistors. The fully functional first silicon prototype is designed to deliver over 1.0 TFLOPS of average performance while dissipating 97 W at 4.27 GHz and 1.07 V supply.

Index Terms—CMOS digital integrated circuits, crossbar router and network-on-chip (NoC), floating-point unit, interconnection, leakage reduction, MAC, multiply-accumulate.

I. INTRODUCTION

T HE scaling of MOS transistors into the nanometer regime opens the possibility for creating large scalable Network-on-Chip (NoC) architectures [1] containing hundreds of integrated processing elements with on-chip communication. NoC architectures, with structured on-chip networks are emerging as a scalable and modular solution to global communications within large systems-on-chip. The basic concept is to replace today’s shared buses with on-chip packet-switched interconnection networks [2]. NoC architectures use layered protocols and packet-switched networks which consist of on-chip routers, links, and well defined network interfaces. As shown in Fig. 1, the NoC architecture basic building block is the “network tile”. The tiles are connected to an on-chip network that routes packets between them. Each tile may consist of one or more compute cores and include logic responsible for routing and forwarding the packets, based on the routing policy of the network. The structured network wiring of such a NoC design gives well-controlled electrical parameters that simplifies timing and allows the use of high-performance circuits to reduce latency and increase bandwidth. Recent tile-based chip multiprocessors include the RAW [3], TRIPS [4], and ASAP [5] projects. These tiled architectures show promise for greater integration, high performance, good scalability and potentially high energy efficiency.

With the increasing demand for interconnect bandwidth, on-chip networks are taking up a substantial portion of system power budget. The 16-tile MIT RAW on-chip network consumes 36% of total chip power, with each router dissipating 40% of individual tile power [6]. The routers and the links of the Alpha 21364 microprocessor consume about 20% of the total chip power. With on-chip communication consuming a significant portion of the chip power and area budgets, there is a compelling need for compact, low power routers. At the same time, while applications dictate the choice of the compute core, the advent of multimedia applications, such as three-dimensional (3-D) graphics and signal processing, places stronger demands for self-contained, low-latency floating-point processors with increased throughput. A computational fabric built using these optimized building blocks is expected to provide high levels of performance in an energy efficient manner. This paper describes design details of an integrated 80-tile NoC architecture implemented in a 65-nm process technology. The prototype is designed to deliver over 1.0 TFLOPS of average performance while dissipating less than 100 W.

The remainder of the paper is organized as follows. Section II gives an architectural overview of the 80-tile NoC and describes the key building blocks. The section also explains the FPMAC unit pipeline and design optimizations used to accomplish single-cycle accumulation. Router architecture details, NoC communication protocol and packet formats are also described. Section III describes chip implementation details, including the high-speed mesochronous clock distribution network used in this design. Details of the circuits used for leakage power management in both logic and memory blocks are also discussed. Section IV presents the chip measurement results. Section V
concludes by summarizing the NoC architecture along with key performance and power numbers.

II. NOC ARCHITECTURE

The NoC architecture (Fig. 2) contains 80 tiles arranged as an 8x10 2-D mesh network that is designed to operate at 4 GHz [7]. Each tile consists of a processing engine (PE) connected to a 5-port router with mesochronous interfaces (MSINT), which forwards packets between the tiles. The 80-tile on-chip network enables a bisection bandwidth of 2 Terabits/s. The PE contains two independent fully pipelined single-precision floating-point multiply-accumulator (FPMAC) units, 3 KB single-cycle instruction memory (IMEM), and 2 KB of data memory (DMEM). A 96-bit Very Long Instruction Word (VLIW) encodes up to eight operations per cycle. With a 10-port (6-read, 4-write) register file, the architecture allows scheduling to both FPMACs, simultaneous DMEM load and stores, packet send/receive from mesh network, program control, and dynamic sleep instructions. A router interface block (RIB) handles packet encapsulation between the PE and router. The fully symmetric architecture allows any PE to send (receive) instruction and data packets to (from) any other tile. The 15 fan-out-of-4 (FO4) design uses a balanced core and router pipeline with critical stages employing performance setting semi-dynamic flip-flops. In addition, a scalable low power mesochronous clock distribution is employed in a 65-nm eight-metal CMOS process that enables high integration and single-chip realization of the teraFLOPS processor.

A. FPMAC Architecture

The nine-stage pipelined FPMAC architecture (Fig. 3) uses a single-cycle accumulate algorithm [8] with base 32 and internal carry-save arithmetic with delayed addition. The FPMAC contains a fully pipelined multiplier unit (pipe stages $S_1-S_3$), and a single-cycle accumulation loop ($S_4$), followed by pipelined addition and normalization units ($S_5-S_8$). Operands A and B are 32-bit inputs in IEEE-754 single-precision format [9]. The design is capable of sustained pipelined performance of one FPMAC instruction every 250 ps. The multiplier is designed using a Wallace tree of 4-2 carry-save adders. The well-matched delays of each Wallace tree stage allow for highly efficient pipelining ($S_1-S_3$). Four Wallace tree stages are used to compress the partial product bits to a sum and carry pair. Notice that the multiplier does not use a carry propagate adder at the final stage. Instead, the multiplier retains the output in carry-save format and converts the result to base 32 (at stage $S_3$), prior to accumulation. In an effort to achieve fast single-cycle accumulation, we first analyzed each of the critical operations involved in conventional FPUs with the intent of eliminating, reducing or deferring the logic operations inside the accumulate loop and identified the following three optimizations [8].

1) The accumulator (stage $S_5$) retains the multiplier output in carry-save format and uses an array of 4-2 carry save adders to “accumulate” the result in an intermediate format. This removes the need for a carry-propagate adder in the critical path.

2) Accumulation is performed in base 32 system, converting the expensive variable shifters in the accumulate loop to constant shifters.

3) The costly normalization step is moved outside the accumulate loop, where the accumulation result in carry-save is added (stage $S_7$), the sum normalized (stage $S_7$) and converted back to base 2 (stage $S_8$).

These optimizations allow accumulation to be implemented in just 15 FO4 stages. This approach also reduces the latency of dependent FPMAC instructions and enables a sustained multiply-add result (2 FLOPS) every cycle. Careful pipeline re-balancing allows removal of 3 pipe-stages resulting in a 25% latency improvement over work in [8]. The dual FPMACs in each PE provide 16 GFLOPS of aggregate performance and are critical to achieving the goal of teraFLOPS performance.
B. Instruction Set

The architecture defines a 96-bit VLIW which allows a maximum of up to eight operations to be issued every cycle. The instructions fall into one of the six categories (Table I): Instruction issue to both floating-point units, simultaneous data memory load and stores, packet send/receive via the on-die mesh network, program control using jump and branch instructions, synchronization primitives for data transfer between PEs and dynamic sleep instructions. The data path between the DMEM and the register file supports transfer of two 32-bit data words per cycle on each load (or store) instruction. The register file issues four 32-bit data words to the dual FPMACs per cycle, while retiring two 32-bit results every cycle. The synchronization instructions aid with data transfer between tiles and allow the PE to stall while waiting for data (WFD) to arrive. To aid with power management, the architecture provides special instructions for dynamic sleep and wake of each PE, including independent sleep control of each floating-point unit inside the PE. The architecture allows any PE to issue sleep packets to any other tile or wake it up for processing tasks. With the exception of FPU instructions which have a pipelined latency of nine cycles, most instructions execute in 1–2 cycles.

C. NoC Packet Format

Fig. 4 describes the NoC packet structure and routing protocol. The on-chip 2-D mesh topology utilizes a 5-port router based on wormhole switching, where each port is subdivided into “FLITs” or “Flow control unITs”. Each FLIT contains six control signals and 32 data bits. The packet header (FLIT_0) allows for a flexible source-directed routing scheme, where a 3-bit destination ID field (DID) specifies the router exit port. This field is updated at each hop. Flow control and buffer management between routers is debit-based using almost-full bits, which the receiver queue signals via two flow control bits when its buffers reach a specified threshold. Each header FLIT supports a maximum of 10 hops. A chained header (CH) bit in the packet provides support for larger number of hops. Processing engine control information including sleep and wakeup control bits are specified in the FLIT_1 that follows the header FLIT. The minimum packet size required by the protocol is two FLITs. The router architecture places no restriction on the maximum packet size.

D. Router Architecture

A 4 GHz five-port two-lane pipelined packet-switched router core (Fig. 5) with phase-tolerant mesochronous links forms the key communication fabric for the 80-tile NoC architecture. Each port has two 39-bit unidirectional point-to-point links. The input-buffered wormhole-switched router [18] uses two logical lanes (lane 0–1) for dead-lock free routing and a fully
non-blocking crossbar switch with a total bandwidth of 80 GB/s (32 bits x 4 GHz x 5 ports). Each lane has a 16 FLIT queue, arbiter and flow control logic. The router uses a 5-stage pipeline with a two stage round-robin arbitration scheme that first binds an input port to an output port in each lane and then selects a pending FLIT from one of the two lanes. A shared data path architecture allows crossbar switch re-use across both lanes on a per-FLIT basis. The router links implement a mesochronous interface with first-in-first-out (FIFO) based synchronization at the receiver.

The router core features a double-pumped crossbar switch [10] to mitigate crossbar interconnect routing area. The schematic in Fig. 6(a) shows the 36-bit crossbar data bus double-pumped at the fourth pipe-stage of the router by interleaving alternate data bits using dual edge-triggered flip-flops, reducing crossbar area by 50%. In addition, the proposed router architecture shares the crossbar switch across both lanes on an individual FLIT basis. Combined application of both ideas enables a compact 0.34 mm² design, resulting in a 34% reduction in router layout area as shown in Fig. 6(b), 26% fewer devices, 13% improvement in average power and one cycle latency reduction (from 6 to 5 cycles) over the router design in [11] when ported and compared in the same 65-nm process [12]. Results from comparison are summarized in Table II.

### E. Mesochronous Communication

The 2-mm-long point-to-point unidirectional router links implement a phase-tolerant mesochronous interface (Fig. 7). Four of the five router links are source synchronous, each providing a strobe (Tx_clk) with 38 bits of data. To reduce active power, Tx_clk is driven at half the clock rate. A 4-deep circular FIFO, built using transparent latches captures data on both edges of the delayed link strobe at the receiver. The strobe delay and duty-cycle can be digitally programmed using the on-chip scan chain. A synchronizer circuit sets the latency between the FIFO write and read pointers to 1 or 2 cycles at each port, depending on the phase of the arriving strobe with respect to the local clock. A more aggressive low-latency setting reduces the synchronization penalty by one cycle. The interface includes the first stage of the router pipeline.

### F. Router Interface Block (RIB)

The RIB is responsible for message-passing and aids with synchronization of data transfer between the tiles and with power management at the PE level. Incoming 38-bit-wide FLITs are buffered in a 16-entry queue, where demultiplexing based on the lane-ID and framing to 64 bits for data packets (DMEM) and 96 bits for instruction packets (IMEM) are accomplished. The buffering is required during program execution since DMEM stores from the 10-port register file have priority over data packets received by the RIB. The unit decodes FLIT_1 (Fig. 4) of an incoming instruction packet and generates several PE control signals. This allows the PE
Fig. 6. (a) Double-pumped crossbar switch schematic. (b) Area benefit over work in [11].

Fig. 7. Phase-tolerant mesochronous interface and timing diagram.

Fig. 8. Semi-dynamic flip-flop (SDFF) schematic.

III. DESIGN DETAILS

To allow 4 GHz operation, the entire core is designed using hand-optimized data path macros. CMOS static gates are used to implement most of the logic. However, critical registers in the FPMAC and router logic utilize implicit-pulsed semi-dynamic flip-flops (SDFF) [13], [14]. The SDFF (Fig. 8) has a dynamic master stage coupled to a pseudo-static slave stage. The FPMAC accumulator register is built using data-inverting

to start execution (REN) at a specified IMEM address (PCA) and is enabled by the new program counter (NPC) bit. After receiving a full data packet, the RIB generates a break signal to continue execution, if the IMEM is in a stalled (WFD) mode. Upon receipt of a sleep packet via the mesh network, the RIB unit can also dynamically put the entire PE to sleep or wake it up for processing tasks on demand.
rising edge-triggered SDFFs with synchronous reset and enable. The negative setup time of the flip-flop is taken advantage of in the critical path. 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, properties which are desirable in high-performance digital designs.

The chip uses a scalable global mesochronous clocking technique, which allows for clock phase-insensitive communication across tiles and synchronous operation within each tile. The on-chip phase-locked loop (PLL) output [Fig. 9(a)] is routed using horizontal M8 and vertical M7 spines. Each spine consists of differential clocks for low duty-cycle variation along the worst-case clock route of 26 mm. An opamp at each tile converts the differential clocks to a single ended clock with a 50% duty cycle prior to distributing the clock across the tile using a balanced H-tree. This clock distribution scales well as tiles are added or removed. The worst case simulated global duty-cycle variation is 3 ps and local clock skew within the tile is 4 ps. Fig. 9(b) shows simulated clock arrival times for all 80 tiles at 4 GHz operation. Note that multiple cycles are required for the global clock to propagate to all 80 tiles. The systematic clock skews inherent in the distribution help spread peak currents due to simultaneous clock switching over the entire cycle.

Fine-grained clock gating [Fig. 9(a)], sleep transistor and body bias circuits [15] are used to reduce active and standby leakage power, which are controlled at full-chip, tile-slice, and individual tile levels based on workload. Each tile is partitioned into 21 smaller sleep regions with dynamic control of individual blocks in PE and router units based on instruction type. The router is partitioned into 10 smaller sleep regions with control of individual router ports, depending on network traffic patterns. The design uses nMOS sleep transistors to reduce frequency penalty and area overhead. Fig. 10 shows the router and on-die network power management scheme. The enable signals gate the clock to each port, MSINT and the links. In addition, the enable signals also activate the nMOS sleep transistors in the nMOS sleep device in the register-file is sized to provide a 4.3X reduction in array leakage power with a 4% frequency impact. The global clock buffer feeding the router is finally gated at the tile level based on port activity.

Each FPMAC implements unregulated sleep transistors with no data retention [Fig. 11(a)]. A 6-cycle pipelined wakeup sequence largely mitigates current spikes over single-cycle re-activation scheme, while allowing floating point unit execution to start one-cycle into wakeup. A faster 3-cycle fast-wake option is also supported. On the other hand, memory arrays use a regulated active clamped sleep transistor circuit [Fig. 11(b)] that ensures data retention and minimizes standby leakage power [16]. The closed-loop opamp configuration ensures that the virtual ground voltage ($V_{SSV}$) is no greater than a $V_{REF}$ input voltage under PVT variations. $V_{REF}$ is set based on memory cell standby $V_{MIN}$ voltage. The average sleep transistor area overhead is 5.4% with a 4% frequency penalty. About 90% of FPU logic and 74% of each PE is sleep-enabled. In addition,
forward body bias can be externally applied to all nMOS devices during active mode to increase the operating frequency and reverse body bias can be applied during idle mode for further leakage savings.

IV. EXPERIMENTAL RESULTS

The teraFLOPS processor is fabricated in a 65-nm process technology [12] with a 1.2-nm gate-oxide thickness, nickel salicide for lower resistance and a second-generation strained silicon technology. The interconnect uses eight copper layers and a low-k carbon-doped oxide ($k = 2.9$) inter-layer dielectric. The functional blocks of the chip and individual tile are identified in the die photographs in Fig. 12. The 275 mm$^2$ fully custom design contains 100 million transistors. Using a fully-tiled approach, each 3 mm$^2$ tile is drawn complete with C4 bumps, power, global clock and signal routing, which are seamlessly
arrayed by abutment. Each tile contains 1.2 million transistors with the processing engine accounting for 1 million (83%) and the router 17% of the total tile device count. Decoupling capacitors occupy about 20% of the total logic area. The chip has three independent voltage regions, one for the tiles, a separate supply for the PLL and a third one for the I/O circuits. Test and debug features include a TAP controller and full-scan support for all memory blocks on chip.

The evaluation board with the packaged chip is shown in Fig. 13. The die has 8390 C4 solder bumps, arrayed with a single uniform bump pitch across the entire die. The chip-level power distribution consists of a uniform M8-M7 grid aligned with the C4 power and ground bump array. The package is a 66 mm × 66 mm flip-chip LGA (land grid array) and includes an integrated heat spreader. The package has a 14-layer stack-up (5-4-5) to meet the various power planes and signal requirements and has a total of 1248 pins, out of which 343 are signal pins. Decoupling capacitors are mounted on the land-side of the package as shown in Fig. 13(b). A PC running custom software is used to apply test vectors and observe results through the on-chip scan chain. First silicon has been validated to be fully functional.

Frequency versus power supply on a typical part is shown in Fig. 14. Silicon chip measurements at a case temperature of 80°C demonstrates chip maximum frequency ($F_{\text{MAX}}$) of 1 GHz at 670 mV and 3.16 GHz at 950 mV, with frequency increasing to 5.1 GHz at 1.2 V and 5.67 GHz at 1.35 V. With all 80 tiles ($N = 80$) actively performing single precision block-matrix operations, the chip achieves a peak performance of 0.32 TFLOPS (670 mV), 1.0 TFLOPS (950 mV), 1.63 TFLOPS (1.2 V) and 1.81 TFLOPS (1.35 V).

Several application kernels have been mapped to the design and the performance is summarized in Table III. The table shows the single-precision floating point operation count, the number of active tiles ($N$) and the average performance in
TFLOPS for each application, reported as a percentage of the peak performance achievable with the design. In each case, task mapping was hand optimized and communication was overlapped with computation as much as possible to increase efficiency. The stencil code solves a steady-state 2-D heat diffusion equation with periodic boundary conditions on left and right boundaries of a rectilinear grid, and prescribed temperatures on top and bottom boundaries. For the stencil kernel, chip measurements indicate an average performance of 1.0 TFLOPS at 4.27 GHz and 1.07 V supply with 358K floating-point operations, achieving 73.3% of the peak performance. This data is particularly impressive because the execution is entirely overlapped with local loads and stores and communication between neighboring tiles. The SGEMM matrix multiplication code operates on two 100 x 100 matrices with 2.63 million floating-point operations, corresponding to an average performance of 0.51 TFLOPS. It is important to note that the read bandwidth from local data memory limits the performance to half the peak rate. The spreadsheet kernel applies reductions to tables of data consisting of pairs of values and weights. For each table the weighted sum of each row and each column is computed. A 64-point 2-D FFT which implements the Cooley–Tukey algorithm [17] using 64 tiles has also been successfully mapped to the design with an average performance of 27.3 GFLOPS. It first computes 8-point FFTs in each tile, which in turn passes results to 63 other tiles for the 2-D FFT computation. The complex communication pattern results in high overhead and low efficiency.

Fig. 15 shows the total chip power dissipation with the active and leakage power components separated as a function of frequency and power supply with case temperature maintained at 80°C. We report measured power for the stencil application kernel, since it is the most computationally intensive. The chip power consumption ranges from 15.6 W at 670 mW to 230 W at 1.35 V. With all 80 tiles actively executing stencil code the chip achieves 1.0 TFLOPS of average performance at 4.27 GHz and 1.07 V supply with a total power dissipation of 97 W. The total power consumed increases to 230 W at 1.35 V and 5.67 GHz operation, delivering 1.33 TFLOPS of average performance. Fig. 16 plots the measured energy efficiency in GFLOPS/W for the stencil application with power supply and frequency scaling. As expected, the chip energy efficiency increases with power supply reduction, from 5.8 GFLOPS/W at 1.35 V supply to 10.5 GFLOPS/W at the 1.0 TFLOPS goal to a maximum of 19.4 GFLOPS/W at 750 mV supply. Below 750 mV, the chip $V_{MAX}$ degrades faster than power saved by lowering tile supply voltage, resulting in overall performance reduction and consequent drop in the processor energy efficiency. The chip provides up to 394 GFLOPS of aggregate performance at 750 mV with a measured total power dissipation of just 20 W. Fig. 17 presents the estimated power breakdown at the tile and router levels, which is simulated at 4 GHz, 1.2 V supply and at 110°C. The processing engine with the dual FPMACs, instruction and data memory, and the register file accounts for 61% of the total tile power [Fig. 17(a)]. The communication power is significant at 28% of the tile power and the synchronous tile-level clock distribution accounts for 11% of the total. Fig. 17(b) shows a more detailed tile to tile communication power breakdown, which includes the router, mesochronous interfaces and links. Clocking power is the largest component, accounting for 33% of the communication power. The input queues on both lanes and data path circuits is the second major component dissipating 22% of the communication power.

Fig. 18 shows the output differential and single ended clock waveforms measured at the farthest clock buffer from the phase-locked loop (PLL) at a frequency of 5 GHz. Notice that the duty cycles of the clocks are close to 50%. Fig. 19 plots the global clock distribution power as a function of frequency and power supply. This is the switching power dissipated in the clock spines from the PLL to the opamp at the center of all 80 tiles. Measured silicon data at 80°C shows that the power...
is 80 mW at 0.8 V and 1.7 GHz frequency, increasing by 10X to 800 mW at 1 V and 3.8 GHz. The global clock distribution power is 2 W at 1.2 V and 5.1 GHz and accounts for just 1.3% of the total chip power. Fig. 20 plots the chip leakage power as a percentage of the total power with all 80 processing engines and routers awake and with all the clocks disabled. Measurements show that the worst-case leakage power in active mode varies from a minimum of 9.6% to a maximum of 15.7% of the total power when measured over the power supply range of 670 mV to 1.35 V. In sleep mode, the nMOS sleep transistors are turned off, reducing chip leakage by 2X, while preserving the logic state in all memory arrays. Fig. 21 shows the active and leakage power reduction due to a combination of selective router port activation, clock gating and sleep transistor techniques described in Section III. Measured at 1.2 V, 80°C and 5.1 GHz operation, the 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 with all ports and global clock buffers feeding the router disabled is 126 mW. This number includes power dissipated in the router, MSINT and the links.

V. CONCLUSION

In this paper, we have presented an 80-tile high-performance NoC architecture implemented in a 65-nm process technology. The prototype contains 160 lower-latency FPMAC cores
and features a single-cycle accumulator architecture for high throughput. Each tile also contains a fast and compact router operating at core speed where the 80 tiles are interconnected using a 2-D mesh topology providing a high bisection bandwidth of over 2 Terabits/s. The design uses a combination of micro-architecture, logic, circuits and a 65-nm process to reach target performance. Silicon operates over a wide voltage and frequency range, and delivers teraFLOPS performance with high power efficiency. For the most computationally intensive application kernel, the chip achieves an average performance of 1.0 TFLOPS, while dissipating 97 W at 4.27 GHz and 1.07 V supply, corresponding to an energy efficiency of 10.5 GFLOPS/W. Average performance scales to 1.33 TFLOPS at a maximum operational frequency of 5.67 GHz and 1.35 V supply. These results demonstrate the feasibility for high-performance and energy-efficient building blocks for peta-scale computing in the near future.

ACKNOWLEDGMENT

The authors would like to thank V. De, Prof. A. Alvandpour, D. Somasekhar, D. Jenkins, P. Aseron, J. Collias, B. Nefcy, P. Iyer, S. Venkataraman, S. Saha, M. Haycock, J. Schutz, and J. Rattner for help, encouragement, and support; T. Mattson, R. Wijngaart, and M. Frumkin from SSG and ARL teams at Intel for assistance with mapping the kernels to the design; the LTD and ATD teams for PLL and package design and assembly; the entire mask design team for chip layout; and the reviewers for their useful remarks.

REFERENCES

Gregory Ruhl (M’07) received the B.S. degree in computer engineering and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 1998 and 1999, respectively.

He joined Intel Corporation, Hillsboro, OR, in 1999 as a part of the Rotation Engineering Program where he worked on the PCI-X DO switch. Gigabit Ethernet validation, and individual circuit research projects. After completing the REP program, he joined Intel’s Circuits Research Lab where he has been working on design, research and validation on a variety of topics ranging from SRAMs and signaling to terascale computing.

Saurabh Dighe (M’05) received the B.E. degree in electronics engineering from University of Mumbai, India, in 2001, and the M.S. degree in computer engineering from University of Minnesota, Minneapolis, in 2003. His work focused on computer architecture and the design, modeling and simulation of digital systems.

He was with Intel Corporation, Santa Clara, CA, working on front-end logic and validation methodologies for the Itanium processor and the Core processor design team. Currently, he is with Intel Corporation’s Circuit Research Labs, Microprocessor Technology Labs in Hillsboro, OR and is a member of the prototype design team involved in the definition, implementation and validation of future terascale computing technologies.

Howard Wilson was born in Chicago, IL, in 1957. He received the B.S. degree in electrical engineering from Southern Illinois University, Carbondale, in 1979.

From 1979 to 1984 he worked at Rockwell-Collins, Cedar Rapids, IA, where he designed navigation and electronic flight display systems. From 1984 to 1991, he worked at National Semiconductor, Santa Clara, CA, designing telecom components for ISDN. With Intel since 1992, he is currently a member of the Circuits Research Laboratory, Hillsboro, OR, engaged in a variety of advanced prototype design activities.

James Tschanz (M’97) received the B.S. degree in computer engineering and the M.S. degree in electrical engineering from the University of Illinois at Urbana-Champaign in 1997 and 1999, respectively.

Since 1999, he has been a Circuits Researcher with the Intel Circuit Research Lab, Hillsboro, OR. His research interests include low-power digital circuits, design techniques, and methods for tolerating parameter variations. He is also an Adjunct Faculty Member at the Oregon Graduate Institute, Beaverton, OR, where he teaches digital VLSI design.

David Finan received the A.S. degree in electronic engineering technology from Portland Community College, Portland, OR, in 1989.


Arvind Singh (M’05) received the B.Tech. degree in electronics and communication engineering from the Institute of Engineering and Technology, Lucknow, India, in 2000, and the M.Tech. degree in VLSI design from the Indian Institute of Technology, Delhi, India, in 2001. He holds the University Gold Medal for his B.Tech. degree.

He is a Research Scientist at the Microprocessor Technology Labs, Intel Bangalore, India, working on scalable on-die interconnect fabric for terascale research and prototyping. His research interests include network-on-chip research and prototyping. His research interests include low-power digital circuits, many core processors and advanced prototyping.

Sailendra Jain received the B.E. degree in electronics engineering from the Devi Ahilya Vishwavidyalaya, Indore, India, in 1999, and the M.Tech. degree in electrical engineering from the IIT, Madras, India, in 2001.

Since 2004, he has been with Intel, Bangalore Design Lab, India. His work includes research and advanced prototyping in the areas of low-power high-performance digital circuits and physical design of multi-million transistors chips. He has coauthored two papers in these areas.

Vasantha Erraguntla (M’92) received the Bachelors degree in electrical engineering from Osmania University, India, and the Masters degree in computer engineering from University of Louisiana, Lafayette.

She joined Intel in Hillsboro, OR, in 1991 and worked on the high-speed router technology for the Intel Teraflop machine (ASCI Red). For over 10 years, she was engaged in a variety of advanced prototype design activities at Intel Laboratories, implementing and validating research ideas in the areas of high-performance and low-power circuits and high-speed signaling. She relocated to India in June 2004 to start up the Bangalore Design Lab to help facilitate circuit research and prototype development. She has co-authored seven papers and has two patents issued and four pending.


In 1992, he joined Intel Corporation’s Supercomputer Systems Division, where he worked on the research and design of clock distribution systems for massively parallel supercomputers. He worked on the system clock design for the ASCI Red TeraFlop Supercomputer (Intel, DOE and Sandia). From 1996 to 2000, he worked as a Signal Integrity Engineer in Intel Corporation’s Micro-
processor Division, focusing on clock distribution for Intel microprocessor package and motherboard designs. In 2001, he joined the circuit research area of Intel Labs. He has been working on physical prototype design and measurement systems for high-speed IO and terascale microprocessor research.

Yatin Hoskote (M’96) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, and the M.S. and Ph.D. degrees in computer engineering from the University of Texas at Austin.

He joined Intel in 1995 as a member of Strategic CAD Labs doing research in verification technologies. He is currently a Principal Engineer in the Advanced Prototype design team in Intel’s Microprocessor Technology Lab working on next-generation network-on-chip technologies.

Dr. Hoskote received the Best Paper Award at the Design Automation Conference in 1999 and an Intel Achievement Award in 2006. He is Chair of the program committee for 2007 High Level Design Validation and Test Workshop and a guest editor for IEEE Design & Test Special Issue on Multicore Interconnects.

Nitin Borkar received the M.Sc. degree in physics from the University of Bombay, India, in 1982, and the M.S.E.E. degree from Louisiana State University in 1985.

He joined Intel Corporation in Portland, OR, in 1986. He worked on the design of the i860 family of embedded microcontrollers. In 1990, he joined the i486DX2 microprocessor design team and led the design and the performance verification program. After successful completion of the i486DX2 development, he worked on high-speed signaling technology for the Teraflop machine. He now leads the prototype design team in the Circuit Research Laboratory, developing novel technologies in the high-performance low-power circuit areas and applying those towards terascale computing research.

Shekhar Borkar (M’97) was born in Mumbai, India. He received the B.S. and M.S. degrees in physics from the University of Bombay, India, in 1979, and the M.S. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1981.

He is an Intel Fellow and Director of the Circuit Research Laboratories at Intel Corporation, Hillsboro, OR. He joined Intel in 1981, where he has worked on the design of the 8051 family of microcontrollers, iWarp multi-computer, and high-speed signaling technology for Intel supercomputers. He is an adjunct member of the faculty of the Oregon Graduate Institute, Beaverton. He has published 10 articles and holds 11 patents.