McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures

Sheng Li[†‡], Jung Ho Ahn[§‡], Richard D. Strong[¶], Jay B. Brockman[†], Dean M. Tullsen[¶], Norman P. Jouppi[‡]
[†]University of Notre Dame, [‡]Hewlett-Packard Labs,
[§]Seoul National University, [¶]University of California, San Diego
[†]{sli2, jbb}@nd.edu, [‡]norm.jouppi@hp.com, [§]gajh@snu.ac.kr, [¶]{rstrong, tullsen}@cs.ucsd.edu

ABSTRACT

This paper introduces McPAT, an integrated power, area, and timing modeling framework that supports comprehensive design space exploration for multicore and manycore processor configurations ranging from 90nm to 22nm and beyond. At the microarchitectural level, McPAT includes models for the fundamental components of a chip multiprocessor, including in-order and out-of-order processor cores, networks-on-chip, shared caches, integrated memory controllers, and multiple-domain clocking. At the circuit and technology levels, McPAT supports critical-path timing modeling, area modeling, and dynamic, short-circuit, and leakage power modeling for each of the device types forecast in the ITRS roadmap including bulk CMOS, SOI, and double-gate transistors. McPAT has a flexible XML interface to facilitate its use with many performance simulators.

Combined with a performance simulator, McPAT enables architects to consistently quantify the cost of new ideas and assess tradeoffs of different architectures using new metrics like energy-delay-area^2 product (EDA^{2}P) and energy-delay-area product (EDAP). This paper explores the interconnect options of future manycore processors by varying the degree of clustering over generations of process technologies. Clustering will bring interesting tradeoffs between area and performance because the interconnects needed to group cores into clusters incur area overhead, but many applications can make good use of them due to synergies of cache sharing. Combining power, area, and timing results of McPAT with performance simulation of PARSEC benchmarks at the 22nm technology node for both common in-order and out-of-order manycore designs shows that when die cost is not taken into account clustering 8 cores together gives the best energy-delay product, whereas when cost is taken into account configuring clusters with 4 cores gives the best EDA^{2}P and EDAP.

Categories and Subject Descriptors

C.0 [**Computer Systems Organizations**]: GENERAL

MICRO'09, December 12–16, 2009, New York, NY, USA.

General Terms

Performance, Verification

1. INTRODUCTION

It has always been true in this community that tools both limit and drive research directions. Wattch [8], first presented in 2000, has been such a tool, enabling a tremendous surge in power-related architecture research. However, several factors drive the need for new tools to address changes in architecture and technology. This includes the need to accurately model multicore and manycore architectures, the need to model and evaluate power, area, and timing simultaneously, the need to accurately model all sources of power dissipation, and the need to accurately scale circuit models into deep-submicron technologies. This paper introduces a new power, area, and timing modeling framework called McPAT (**M**ulti**c**ore **P**ower, **A**rea, and **T**iming), which addresses these challenges.

McPAT advances the state of the art in several directions compared to Wattch, which is the current standard for power research. First, McPAT is an *integrated* power, area, and timing modeling framework that enables architects to use new metrics combining performance with both power and area such as energy-delay-area^2 product (EDA^{2}P) and energy-delay-area product (EDAP), which are useful to quantify the cost of new architectural ideas. McPAT specifies the low-level design parameters of regular components (e.g. interconnects, caches, and other array-based structures) based on high-level constraints (clock rate and optimization target) given by a user, ensuring the user is always modeling a reasonable design. This approach enables the user, if they choose, to ignore many of the low-level details of the circuits being modeled.

Second, McPAT models more than just dynamic power, which is critical in deep-submicron technologies since static power has become comparable to dynamic power [38]. McPAT models all three types of power dissipation—dynamic, static, and short-circuit power—to give a complete view of the power envelope of multicore processors.

Third, McPAT provides a complete, integrated solution for multithreaded and multicore/manycore processor power. Some researchers have combined Wattch's core power model with a router power model [18], but even that is an incomplete solution. Today's multicores are complex systems of cores, caches, interconnects, memory controllers, multiple-domain clocking, and other components. McPAT models the power of most of the important parts of multicore processors,

including all of the components listed above. Modeling multithreaded processors using Wattch is also hard since it does not model resource-sharing/partitioning schemes nor hardware implementation overhead in multithreaded processors. Wattch models out-of-order (OOO) processors based on the synthetic RUU model of SimpleScalar [9]. McPAT supports more detailed and realistic models based on existing OOO processors. McPAT can model both a reservation-station model and a physical-register-file model based on real architectures, including the Intel P6 [16] and Netburst [15].

Fourth, McPAT handles technologies that can no longer be modeled by the linear scaling assumptions used by Wattch. The simple linear scaling principles are no longer valid because device scaling has become highly non-linear in the deep-submicron era. McPAT uses technology projections from ITRS [38] for dynamic, static, and short-circuit power; as a result, this tool will naturally evolve with ITRS even beyond the end of the current road map.

The remainder of this paper is organized as follows. After discussing related work in Section 2, we describe the overall structure of McPAT in Section 3. Section 4 discusses the hierarchical, integrated model of power, area, and timing. It also presents the validation results. In Section 5, we combine McPAT with performance simulators and explore the interconnect options of future manycore processors by varying the degree of clustering over generations of process technologies. We conclude in Section 6.

2. RELATED WORK

CACTI [41] was the first tool to address the need for rapid power, area, and timing estimates for computer architecture research, focusing on RAM-based structures. The most recent release of the tool supports SRAM and DRAM based caches as well as plain memory arrays. It uses device models based on the industry-standard ITRS roadmap [38], using MASTAR [38] to calculate device parameters at different technology nodes. CACTI uses the method of logical effort to size transistors. It contains optimization features that enable the tool to find a configuration with minimal power consumption, given constraints on area and timing.

The complexity-effective approach [33] was one of the first attempts to use analytic models to obtain rapid estimates for processor timing, focusing on control, issue, selection, and bypass logic. Using generic circuit models for pipeline stages, it estimates the RC delay for each stage and determines the critical path.

Wattch [8] is a widely-used processor power estimation tool. Wattch calculates dynamic power dissipation from switching events obtained from an architectural simulation and capacitance models of components of the microarchitecture. For array structures, Wattch uses capacitance models from CACTI, and for the pipeline it uses models from [33]. When modeling out-of-order processors, Wattch uses the synthetic RUU model that is tightly coupled to the SimpleScalar simulator [9]. Wattch has enabled the computer architecture research community to explore power-efficient design options, as technology has progressed; however, limitations of Wattch have become apparent. First, Wattch models power without considering timing and area. Second, Wattch only models dynamic power consumption; the HotLeakage package [45] partially addressed this deficiency by adding models for subthreshold leakage. Third, Wattch uses simple linear scaling models based on 0.8μm technology that are inaccurate to make predictions for current and future deep-submicron technology nodes.

Orion [18] is a tool for modeling power in networks-on-chip (NoC). Version 2.0 includes models for area, dynamic power, and gate leakage, but does not consider short-circuit power or timing. It uses repeated wire models for interconnect, as well as device parameters for future technology nodes obtained from the ITRS roadmap using MASTAR and other methods. Kumar et al. provide further details on NoC layouts that take chip floorplans into consideration [22].

3. MCPAT: OVERVIEW AND OPERATION

McPAT is the first *integrated* power, area, and timing modeling framework for multithreaded and multicore/manycore processors. It is designed to work with a variety of performance simulators (and thermal simulators, etc.) over many technology generations. McPAT allows the user to specify low-level configuration details. It also provides default values when the user chooses to only specify high-level architectural parameters.

Figure 1 is a block diagram of the McPAT framework. Rather than being hardwired to a particular simulator, McPAT uses an XML-based interface with the performance simulator. This interface allows both the specification of the static microarchitecture configuration parameters and the passing of dynamic activity statistics generated by the performance simulator. McPAT can also send runtime power dissipation back to the performance simulator through the XML-based interface, so that the performance simulator can react to power (or even temperature) data. This approach makes McPAT very flexible and easily ported to other performance simulators. McPAT runs separately from a simulator and only reads performance statistics from it. Performance simulator overhead is minor – only the possible addition of some performance counters. Since McPAT provides complete hierarchical models from the architecture to the technology level, the XML interface also contains circuit implementation style and technology parameters that are specific to a particular target processor. Examples are array types, crossbar types, and the CMOS technology generation with associated voltage and device types.

The key components of McPAT are (1) the hierarchical power, area, and timing models described in Section 4, (2) the optimizer for determining circuit-level implementations, and (3) the internal chip representation that drives the analysis of power, area, and timing. Most of the parameters in the internal chip representation, such as cache capacity and core issue width, are directly set by the input parameters.

McPAT's hierarchical structure allows it to model structures at a very low level, and yet still allows an architect to focus on the high-level configuration. The optimizer determines unspecified parameters in the internal chip representation, focusing on two major regular structures: interconnects and arrays. For example, the user can specify the frequency and bisection bandwidth of the network-on-chip, the capacity and the associativity of caches, or the number of cache banks, while letting the tool determine the implementation details such as the choice of metal planes, the effective signal wiring pitch for the interconnect, or the length of wordlines and bitlines of the cache bank. These optimizations lessen the burden on the architect to figure out every detail, and

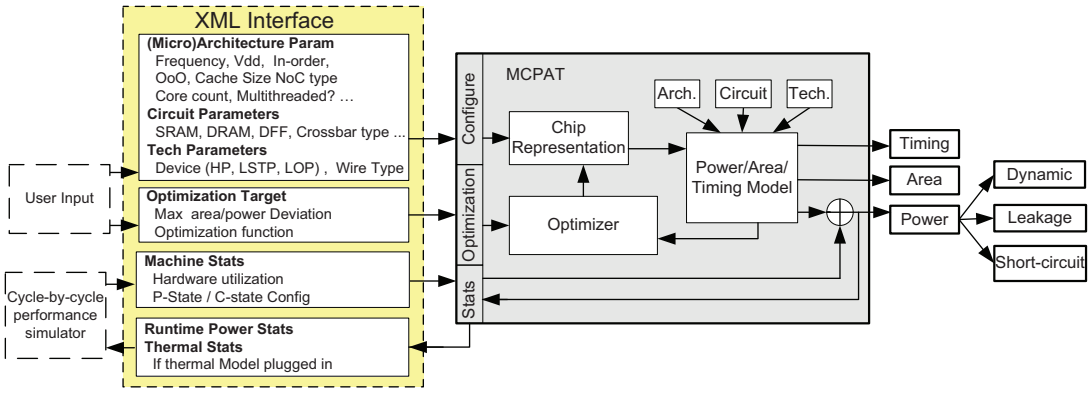


Figure 1: Block diagram of the McPAT framework.

significantly lowers the learning curve to use the tool. Users always have the flexibility to turn off these features and set the circuit-level implementation parameters by themselves.

The main focus of our tool is accurate power and area modeling, and a target clock rate is used as a design constraint. The user of McPAT specifies the target clock frequency, the area and power deviation, the optimization function, and other architectural/circuit/technology parameters. The optimization space that McPAT explores can be huge, especially when there are many unspecified parameters. McPAT does intelligent and extensive search of the design space. For each processor component, McPAT optimizes the circuit-level structure to satisfy the timing constraint. Then, if the resulting power or area is not within the allowed deviation of the best value found so far, the configuration is discarded. Finally, among the configurations satisfying the power and area deviation, McPAT applies the optimization function to report the final power and area values. The module power, area, and timing models together with the final chip representation generated by the optimizer are used to compute the final chip-area, timing, and peak power. The peak power of individual units and the machine utilization statistics (activity factor) are used to calculate the final runtime power dissipation.

Another distinguishing feature of McPAT is its ability to model advanced power management techniques, such as the P- and C-state [30] power management of modern processors. After calling McPAT to finish initialization, a performance simulator can pass statistical information and invoke McPAT anytime during the simulation. McPAT will then calculate the corresponding power dissipation for the particular period and send it back to the performance simulator when required. This allows the simulator to react to simulated power or thermal sensors (assuming a temperature model is attached to the backend), by changing voltage and frequency settings, or invoking one of multiple power-saving states on idle circuit blocks. This allows the architect to use the framework to model the full range of power management alternatives.

4. MODELING FRAMEWORKS AND VALIDATION OF MCPAT

In order to model the power, area, and timing of a multicore processor, McPAT takes an *integrated* and *hierarchical* approach. It is integrated in that McPAT models power, area, and timing *simultaneously*. Because of this McPAT is able to ensure that the results are mutually consistent from an electrical standpoint. It is hierarchical in that it decomposes the models into three levels: architectural, circuit, and technology. This provides users with the flexibility to model a broad range of possible multicore configurations across several implementation technology generations. Taken together, this integrated and hierarchical approach enables the user to paint a comprehensive picture of a design space, exploring tradeoffs between design and technology choices in terms of power, area, and timing.

4.1 Power, Area, and Timing Models

Power Modeling: Power dissipation of CMOS circuits has three main components: dynamic, short-circuit, and leakage power. Circuits dissipate dynamic power when they charge and discharge the capacitive loads to switch states. Dynamic power is proportional to the total load capacitance, the supply voltage, the voltage swing during switching, the clock frequency, and the activity factor. We calculate the load capacitance of a module by decomposing it into basic circuit blocks, and using analytic models for each block with appropriately sized devices. We calculate the activity factor using access statistics from architectural simulation together with circuit properties. Switching circuits also dissipate short-circuit power due to a momentary short through the pull-up and pull-down devices. We compute the short-circuit power using the equations derived in the work by Nose et al. [32] that predicts trends for short-circuit power. If the ratio of the threshold voltage to the supply voltage shrinks, short-circuit power becomes more significant. It is typically about 10% of the total dynamic power, however it can be as high as 25% of the dynamic power in some cases [32].

Transistors in circuits leak, dissipating static power. Leakage current depends on the width of the transistors and the local state of the devices. There are two leakage mechanisms. Subthreshold leakage occurs because a small current passes between the source and drain of off-state transistors. Gate leakage is the current leaking through the gate terminal, and varies greatly with the state of the device. We determine the unit leakage current using MASTAR [38] and Intel's data [4].

Timing Modeling: Like the power model, McPAT's timing model breaks the system down into components and stages. While the power model requires only the capacitance to compute dynamic power, the timing model uses both resistance and capacitance to compute RC delays, us-

ing a methodology similar to CACTI [41] and the work by Palacharla et al. [33], with significant changes in implementation described later in this section. McPAT determines the achievable clock frequency of a processor from the delays of its components along the critical path.

Area Modeling: McPAT takes the analytical methodology described in CACTI to model basic logic gates and regular structures, including memory arrays (e.g., RAM, CAM (content addressable memory), and DFFs (D flip-flop)), interconnects (e.g., router and link), and regular logic (e.g., decoder and dependency-checking unit). An algorithmic approach does not work well for complex structures that have custom layouts, such as ALUs. For these, currently McPAT takes an empirical modeling approach [12, 36] which uses curve fitting to build a parameterizable numerical model for area from published information on existing processor designs, and then scales the area for different target technologies.

4.2 Hierarchical Modeling Framework

McPAT's integrated power, area, and timing models are organized in a three-level hierarchy, as illustrated in Figure 2. This is the first modeling framework which completely models a multicore/manycore processor from the architecture to the technology level. On the architectural level, a multicore processor is decomposed into major architectural components such as cores, NoCs, caches, memory controllers, and clocking. On the circuit level, the architectural building blocks are mapped into four basic circuit structures: hierarchical wires, arrays, complex logic, and clocking networks. On the technology level, data from the ITRS roadmap [38] is used to calculate the physical parameters of devices and wires, such as unit resistance, capacitance, and current densities.

4.2.1 Multicore Architecture Level Modeling

The architecture level represents the building blocks of a multicore processor system. Below, we provide an overview of the models for these high-level blocks and how they are mapped to the circuit level.

Core: A core can be divided into several main units: an instruction fetch unit (IFU), an execution unit (EXU), a load and store unit (LSU), and an out-of-order (OOO) issue/dispatch unit for an OOO processor. Each of them can be further divided into hardware structures. For example, the EXU may contain ALUs, FPUs, bypass logic, and register files. In our hierarchical framework, the ALU and FPU are mapped to the complex logic model at the circuit level. Bypass logic can be mapped to a combination of the wire and logic models, while register files can be mapped to the array model. McPAT supports detailed and realistic models that are based on existing high-performance OOO processors. We greatly extend the basic analytical models in Palacharla, et al.'s work [33] to support both the reservation-station-based (data-capture scheduler) architectures such as the Intel P6 architecture [16] and the physical-register-file-based (non-data-capture scheduler) architectures such as the Intel Netburst [15] and the DEC Alpha architecture [19]. McPAT supports both RAM- and CAM-based renaming logic which can be found in the Intel and Alpha architectures.

McPAT also models the power, area, and timing of multithreaded processors, whether in-order (e.g., Sun Niagara) or out-of-order (e.g., Intel Nehalem). Since McPAT already contains models for each of the base processors, multithreading support is included by modeling the sharing and duplication of hardware resources, as well as the extra hardware overhead. McPAT models multithreaded architectures based on designs of the Niagara processors [20, 31], Intel hyperthreading technology [21], and early research in SMT architecture [42].

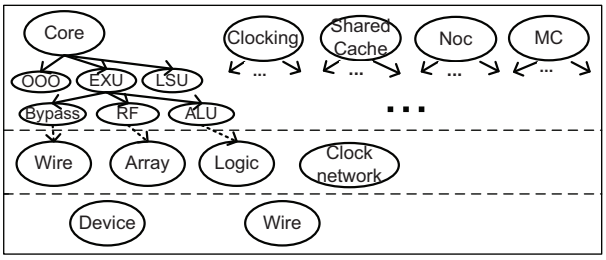


Figure 2: Modeling methodology of McPAT.

NoC: A NoC has two main components: signal links and routers. For signal links, we use hierarchical wires, as described in Section 4.2.2. We use the same analytical approach used in modeling cores to model routers: breaking the routers into basic building blocks such as flit buffers, arbiters, and crossbars; then building analytical models for each building block. Unlike Orion 2 [18] that only models area and power, McPAT models power, area and timing. McPAT is the first modeling tool supporting a double-pumped crossbar [43], which reduces die area for on-chip interconnect intensive designs.

On-chip caches: McPAT supports both shared and private caches. It models coherent caches by modeling the directory storage associated with each cache bank. Depending on the architecture, a directory can be mapped to CAM structures at the circuit level as in Niagara processors [20,31] or normal cache structures as in the Alpha 21364 [17].

Memory controller: McPAT is the first modeling framework to model on-chip memory controllers. Memory controller modeling follows a design from Denali [10] that contains three main hardware structures: 1) the front-end engine responsible for rescheduling the memory requests, 2) the transaction processing engine that has the logic and sequencer to generate the command, address, and data signal, and 3) the physical interface (PHY) that serves as an actual channel of the memory controller for communicating off-chip to memory. The front-end engine is modeled using CAM and RAM structures. We empirically model PHY and the transaction processing engine according to published data from Rambus [23] and AMD [3].

Clocking: Clocking circuitry has two main parts: the phase-locked loop (PLL) with fractional dividers to generate the clock signals for multiple clock domains, and the clock distribution network to route the clock signals. McPAT uses an empirical model for the power of a PLL and fractional divider, based on scaling published results from Sun and Intel [2, 37]. The clock distribution network can be directly mapped to the clock network model at the circuit level.

4.2.2 Circuit Level Modeling

Hierarchical repeated wires: Hierarchical repeated wires are used to model both local and global on-chip wires. Performance of wires is governed by two important parameters: resistance and capacitance. We model short wires us-

ing a one-section π-RC model [41]. Wires scale more slowly than gates with respect to RC delays, and unbuffered wires cannot keep up with the improved transistor delay. For long wires, we use a buffered wire model [41], where we optimize sizing parameters for energy-delay product rather than only for speed.

We assume multiple metal planes for local, intermediate, and global interconnect, each with different wire pitches and aspect ratios. The assignment of signals to wiring planes plays a key role in determining power, area, and timing characteristics. McPAT's optimizer automatically assigns wires to planes to achieve specified objectives by trying different metal layers and varying effective wiring pitches. Latches are also inserted and modeled when necessary to satisfy the target clock rate.

Arrays: McPAT includes models for three basic array models at the circuit level: RAM-, CAM-, and DFF-based arrays. These structures are modeled based on CACTI [41], with several important extensions. First, McPAT reimplements the CAM model from CACTI to more accurately reflect its multi-banked/ported structure and also adds a write operation. Second, McPAT adds a detailed DFF array model. Finally, McPAT adds gate leakage and improved timing models for arrays.

Logic: McPAT employs three different schemes for modeling logic blocks, depending on the complexity of the block. For highly regular blocks with predictable structures, such as memories or networks, McPAT uses the algorithmic approach of CACTI [41]. For structures that are less regular but can still be parameterized, such as thread selection or decoding logic, McPAT uses analytic models similar to those in [33] but modeled after existing processors from Intel, AMD, and Sun. Finally, for highly customized blocks such as functional units, McPAT uses empirical models based on published data for existing designs scaled to different technologies. For example, the ALU and FPU models are based on actual designs by Intel [29] and Sun [24].

Clock distribution network: A clock distribution network is responsible for routing clock signals of different frequencies to clock domains, with drops to individual circuit blocks. This network is a special case of a hierarchical repeated wire network, but has strict timing requirements with large fanout loads spanning the entire chip. It consumes a significant fraction of total chip power [11]. We represent a clock distribution network using a separate circuit model that has three distinct levels: global, domain, and local. We assume an H-tree topology for global-level and domain-level networks and a grid topology for the local networks as shown in both the Niagara processors [20,31] and the Intel Itanium processors [27]. NAND gates are used at all final clock heads to enable clock gating.

4.2.3 Technology Level Modeling

McPAT uses MASTAR [38] to derive device parameters from the ITRS 2007 roadmap [38]. The current implementation of McPAT includes data for the 90nm, 65nm, 45nm, 32nm, and 22nm technology nodes, which covers the ITRS roadmap through 2016. The ITRS assumes that planar bulk CMOS devices will reach practical scaling limits at 36nm, at which point the technology will switch to silicon-on-insulator (SOI). Below 25nm, the ITRS predicts that SOI will reach its limits and that double-gate devices will be the only option. McPAT captures each of these options, making it scalable with the ITRS roadmap.

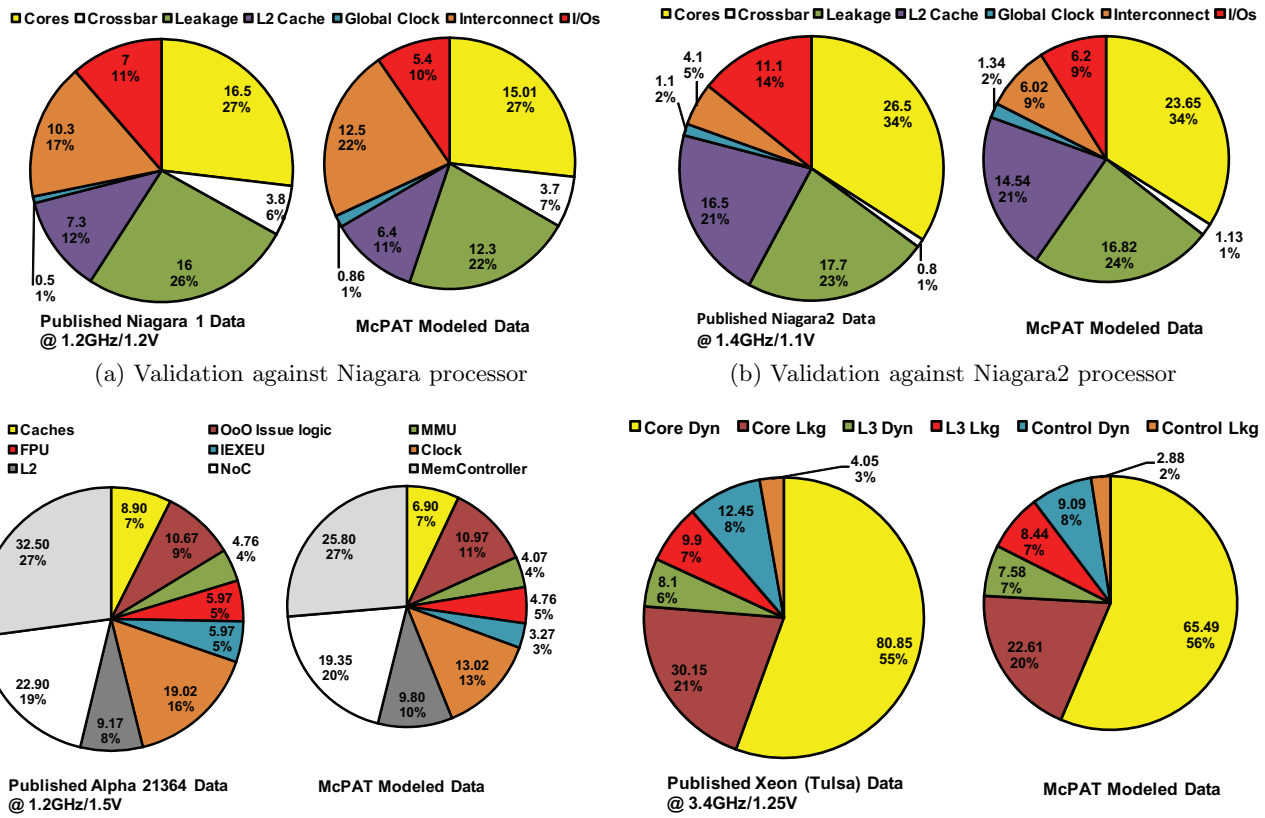
4.3 Modeling Power-saving Techniques

McPAT models two major power saving techniques: clock gating to reduce dynamic power and power-saving states (a.k.a. sleep states or C-states) to reduce static power. McPAT models the actual clock gating buffer circuitry at the distribution network heads, rather than using a heuristic approach as Wattch [8] did. McPAT is the first major architectural modeling tool to include power-saving states, which are used in many modern multicore processors such as the Intel Core 2 Duo [30]. Our framework models power-saving states with multiple sleep modes as described in [1]. The circuitry uses footer devices and adjusts their gate bias to achieve four operating modes: Active, Sleep, Dream, and Snore. Having different sleep modes allows tradeoffs between the power savings and the wakeup overhead with respect to wakeup power and delay. For example, dream mode can save 50% more static power than sleep mode, but at the expense of twice the wakeup delay and three times the wakeup energy. The user can specify power-saving modes for various components. As mentioned in Section 3, the modeling of C-states and the flexible XML-based interface enable McPAT to model advanced power management alternatives when required by the performance simulator.

4.4 Validation

The primary focus of McPAT is accurate power and area modeling at the architectural level when timing is given as a main design constraint. Both *relative* and *absolute* accuracy are important for architectural level power modeling. Relative accuracy means that changes in modeled power as a result of architecture modifications should reflect on a relative scale the changes one would see in a real design. While relative accuracy is critical, absolute accuracy is also important if one wants to compare results against thermal design power (TDP) limits, or to put power savings in the core in the context of the whole processor or whole system. The relative accuracy of McPAT ensures that the relative power weights of different components of a chip have been correctly modeled. The absolute magnitude accuracy of McPAT means that the power numbers for individual components and total power are evaluated correctly.

We compare the output of McPAT against published data for the 90nm Niagara processor [24] running at 1.2GHz with a 1.2V power supply, the 65nm Niagara2 processor [31] running at 1.4GHz with a 1.1V power supply, the 65nm Xeon processor [37] running at 3.4GHz with a 1.25V power supply, and the 180nm Alpha 21364 processor [17] running at 1.2GHz with a 1.5V power supply. These include in-order and out-of-order processors as well as single-threaded and multithreaded processors in the validation targets. Thus, the validations stress McPAT in a comprehensive and detailed way. The comparisons against Niagara and Niagara2 processors also test our ability to cross technology generations and retain accuracy. The configurations for the validations are based on published data on the target processors in [17,24,31,37,40], including target clock rate, working temperature, and architectural parameters. The target clock cycle time is used as the upper bound for the timing constraint that is used in McPAT to determine the basic circuit properties and must be satisfied before other optimizations and



(a) Validation against Niagara processor

(b) Validation against Niagara2 processor

(c) Validation against Alpha 21364 processor

(d) Validation against Xeon Tulsa processor

Figure 3: McPAT validation results. The numbers in all charts are the reported and modeled power numbers of the components. The percentages denote the ratios of the component power to the total power. There are miscellaneous components such as SOC logic and I/O that McPAT does not model in detail because their structures are unknown. Therefore, 61.4W out of the total 63W, 77.9W out of the total 84W, 119.8W out of the total 125W, and 145.5W out of the total 150W are modeled for Niagara, Niagara2, Alpha 21364, and Xeon Tulsa, respectively.

trade-offs can be applied. Therefore, the generated results must match the clock rate of the actual target processor. Because timing (target clock rate) is already considered when computing and optimizing power and area, only power and area validation results are shown in this section.

Detailed validation exercises can be found in the McPAT tech report [25]; here we simply summarize the validation results. Figure 3 shows these results. Unfortunately, the best available power numbers we have for these processors are for peak power rather than average power. Fortunately, McPAT can also output peak power numbers based on maximum activity factors. Results show that modeled power numbers track the published numbers well. For the Niagara processor, the absolute power numbers for cores and crossbars generated by McPAT match very well with the published data. Over all seven components, the average difference in absolute power between the modeled power numbers and published Niagara data is just 1.47W, for an average error per component of 23%. That number seems high, but the two big contributors are clock power (72% error, but a small contributor to total power) and leakage power (23% error). Both are significantly more accurate for the Niagara2. For Niagara2, the average error is 1.87W (26%), but by far the biggest contributor (4.9W error) is the I/O power. This arises because Niagara2 is a complicated SOC with different types of I/Os including memory, PCI-e, and 10Gb Ethernet [31] while McPAT only models on-chip memory controllers/channels as I/Os. If we were validating average power instead of peak power, this difference would shrink given the expected relative activity factors of those components. The modeled power of the OOO issue logic, a key component of the OOO core in the Alpha 21364 processor, is very close to the reported power with only 2.78% difference. Although there are no detailed power breakdowns of both the core and uncore parts of Xeon Tulsa, the modeled bulk power of core and uncore comes close to reported data, at -20.63% and -11% respectively.

Table 1 shows the comparison of total power and area for validations against the target processors. Differences between the total peak power generated by McPAT and reported data are 10.84%, 17.02%, 21.68%, and 22.61% for Niagara, Niagara2, Alpha 21364, and Xeon Tulsa, respectively. It is worth noting that these differences include the unknown parts that we do not model in detail. Chip-to-chip power variation in recent microprocessor designs [7] is also comparable to the magnitude of the power validation errors reported in Table 1. The modeled area numbers also track the published numbers well as shown in Table 1. The error is higher for Niagara2 because it has more I/O components, as mentioned above, which are not modeled in McPAT. Given the generic nature of McPAT, we consider these errors on both power and area acceptable.

Processor	Published total Power and Area	McPAT Results	% McPAT error
Niagara	63 W / 378 mm^2	56.17 W / 295 mm^2	-10.84 / -21.8
Niagara2	84 W / 342 mm^2	69.70 W / 248 mm^2	-17.02 / -27.3
Alpha 21364	125 W / 396 mm^2	97.9 W / 324 mm^2	-21.68 / -18.2
Xeon Tulsa	150 W / 435 mm^2	116.08 W / 362 mm^2	-22.61 / -16.7

Table 1: Validation results of McPAT with regard to total power and area of target processors.

5. SCALING AND CLUSTERING TRADEOFFS IN MANYCORE PROCESSORS

We illustrate the utilization of McPAT by applying it to scaling and clustering tradeoffs in a manycore architecture. We evaluate the following aspects of this architecture: (1) the scaling trends of power, area, and timing of the proposed manycore architecture, and (2) the benefits of organizing cores into clusters with local interconnect. We evaluate this architecture across five technologies—90, 65, 45, 32, and 22nm—which covers years 2004 to 2016 of the ITRS roadmap.

5.1 Experimental Setup

Figure 4 shows the manycore architecture we assume targeting future high throughput computing. It consists of multiple clusters connected by a 2D-mesh on-chip network. A cluster has one or more multithreaded Niagara-like [20] cores and a multi-banked L2 cache. Each core has up to 4 active threads and 32KB 4-way set-associative L1 instruction and data caches. All cores in a cluster share a multi-banked 16-way set-associative L2 cache. The number of L2 banks equals the number of cores per cluster. The size of an L2 bank is 256KB. All caches have 64B cache lines. A crossbar is used to connect cores and L2 cache banks for intra-cluster communications. A two-level hierarchical directory-based MESI protocol is used for cache coherency. Within a cluster, the L2 cache is inclusive and filters coherency traffic between L1 caches and directories. Between clusters, a cache directory is implemented by using directory caches that are associated to the on-chip memory controllers, similar to the implementation in the Alpha 21364 processor. The 2D-mesh networks have a data width of 256 bits. We use minimal dimension-order routing and two virtual channels per physical port. Each virtual channel has a 32-deep flit input buffer. Double-pumped crossbars [43] are used in the routers to reduce the die area. Routers in the networks have a local port that connects the hub of a cluster as well as ports that connect the neighboring routers.

Table 2 shows the parameters of the manycore architecture at each technology generation. We start from a conservative 2.0GHz clock rate at 90nm technology, which is about the average of that of the Niagara processor and Intel processors fabricated in 90nm processes. The intrinsic speed of high-performance transistors increases by 17% per year according to the ITRS. However, increasing clock frequency at this pace will lead to unmanageable chip power density. Moreover, unlike Intel's approach of changing micro-architectures of their processors during technology scaling, we increase the number of cores and memory controllers aggressively, while keeping the same micro-architecture for new generations. Therefore, we increase the clock frequency conservatively by around 15% every generation. We also start from a conservative die size around 200mm^2 at 90nm technology and use McPAT to optimize power, area, and timing. The results show that four cores can be placed within the specified area at 90nm. Then, we double the number of cores for each generation. It is difficult to increase the number of memory controllers and channels linearly with the increased core count because of the limited pin count of the chip [38]. We assume that the number of memory controllers grows proportional to the square root of the cluster count since the bandwidth of each controller also increases over time. The memory channels are shared by all clusters through the on-chip network and placed at the edge of the chip to minimize the routing overhead as shown in Figure 4. As shown in Table 2, we also scale the bandwidth of main memory based on the expected availability of major DIMM products at each technology node.

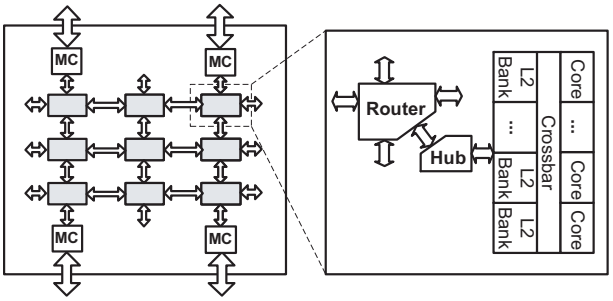


Figure 4: Manycore system architecture. MCs refer to memory controllers.

We developed a manycore simulation infrastructure where a timing simulator and a functional simulator are decoupled, as in GEMS [28]. We modified a user-level thread library [34] in the Pin [26] binary instrumentation tool to support more pthread APIs, and use it as a functional simulator to run applications. In-order cores, caches, directories, on-chip networks, and memory channels are modeled in an event-driven timing simulator, which controls the execution flow of a program running in the functional simulator and effectively operates as a thread scheduler.

SPLASH-2 [44], PARSEC [5], and SPEC CPU2006 [14] benchmark suites are used for experiments. The number of threads spawned in a multithreaded workload is the same as the number of hardware threads so each thread is statically mapped to a hardware thread. We use all SPLASH-2 applications and 5 of the PARSEC applications (only canneal, streamcluster, blackscholes, fluidanimate, and swaptions currently run on our infrastructure.) The simlarge dataset is used for PARSEC applications while the datasets used for SPLASH-2 applications are summarized in Table 3. For each application, the same dataset is used throughout all process generations. We use the SPEC CPU2006 benchmark suite to measure the system performance on consolidated workloads. The benchmark suite consists of integer (CINT) and floating-point (CFP) benchmarks, all of which are single-threaded. We pick 12 applications from both

Parameters	90nm	65nm	45nm	32nm	22nm
Clock rate (GHz)	2.0	2.3	2.7	3.0	3.5
The number of cores	4	8	16	32	64
The number of memory controllers	2	3	4	6	8
Memory capacity per channel (GB)	2	4	4	8	8
Main memory type	DDR2-667	DDR3-800	DDR3-1066	DDR3-1333	DDR3-1600

Table 2: Parameters of the manycore architecture across technology generations. Each memory controller has one memory channel.

CINT and CFP, and make 3 groups each, 4 applications per group, based on their main-memory bandwidth demand [14] as shown in Table 3. We find the representative simulation phases of each application and their weights using Simpoint 3.0 [39]. Each hardware thread runs a simulation phase and the number of instances per phase is proportional to its weight. We skip the initialization phases of each workload and simulate 2 billion instructions unless it finishes earlier.

5.2 Overview of Area and Power

Table 4 shows the area and maximum power of the proposed architecture with four cores per cluster across five technology generations. Core area varies from 43% to 62% of total die size across technologies. Core area scales worse than uncore area since uncore components, especially L2 caches, crossbars, and routers, are more regular and easier to scale across generations than cores. According to McPAT's area modeling results, the double-pumped crossbars reduce the area of intra-cluster crossbars and 2D mesh routers by 54.1% and 35.6% respectively, compared to the single-pumped crossbar implementations. Although we save significant area on interconnects within and between clusters, uncore components still occupy a big portion of total die area.

Short-circuit power is around 10% of the total dynamic power, with fluctuations within 3.1% across all the technology generations. The main reason for the stable short-circuit power is that we use ITRS technology models that have stable V_{th} to V_{dd} ratios. Cores burn about half of the total maximum dynamic power across generations. Gate leakage is an important component in 90nm and 65nm technology, being 37.6% of the total leakage power at 65nm technology. Hi-k metal gate transistors [4] are introduced at 45nm, which reduces the gate leakage by more than 90%. SOI technology and double gate (DG) devices that are used at 32nm and 22nm technology also help to keep the subthreshold leakage under control.

Table 5 shows the die areas of various manycore architectures when the number of cores per cluster is varied from 1 to 8 at the 22nm technology node. Since the total number of cores is fixed at 64, NoC size decreases as the number of cores per cluster increases. We keep the same NoC bisection bandwidth on all configurations. The 1 core per cluster design is the smallest in size because it does not need crossbars between cores and L2 caches. Even though it needs more routers to connect clusters, the size of each router is smaller since we keep the same bisection bandwidth. When the size of a mesh network is not square, its bisection bandwidth is limited by a cut through its smaller dimension. So we need the same link width for 8×4 and 4×4 networks. That is why the die size of the 2 core per cluster design is very close to that of the 4 core per cluster design, even though the latter has much bigger crossbars in the clusters.

	NoC size	Die size (mm^2)
1 core per cluster	8×8	239.1
2 cores per cluster	4×8	246.3
4 cores per cluster	4×4	250.6
8 cores per cluster	2×4	278.6

Table 5: NoC sizes and die areas of the manycore architecture at 22nm. The NoC bisection bandwidth is kept constant across configurations.

5.3 Performance and Efficiency Tradeoffs in Technology Scaling and Clustering

Because McPAT provides an integrated power, area, and timing model, when combined with performance data, chip multiprocessors can be analyzed using several metrics previously unavailable in architecture studies. We think energy-delay-area^2 product (EDA^{2}P) and energy-delay-area product (EDAP) are particularly interesting metrics. These metrics include both an operational cost component (energy) as well as a capital cost component (area). Although the die yield is proportional to the fourth power of the area [35], in practice due to good die yield and when combined with die per wafer, die cost is roughly proportional to the square of the area [13]. So when designing and manufacturing a chip multiprocessor, we believe EDA^{2}P is a good way of including chip cost into the optimization process. However, other fixed system costs such as memory and I/O reduce the overall system cost dependence on chip multiprocessor cost. Thus we believe EDAP could be a more useful metric for chip multiprocessors at the system level than EDA^{2}P. Hence, a chip vendor may favor EDA^{2}P, while a system vendor could prefer EDAP. Finally, given McPAT's area models, another interesting metric is power density: chip power divided by area. Cooling a microprocessor becomes substantially more difficult as power density increases, and this can add significant capital cost for more advanced packaging.

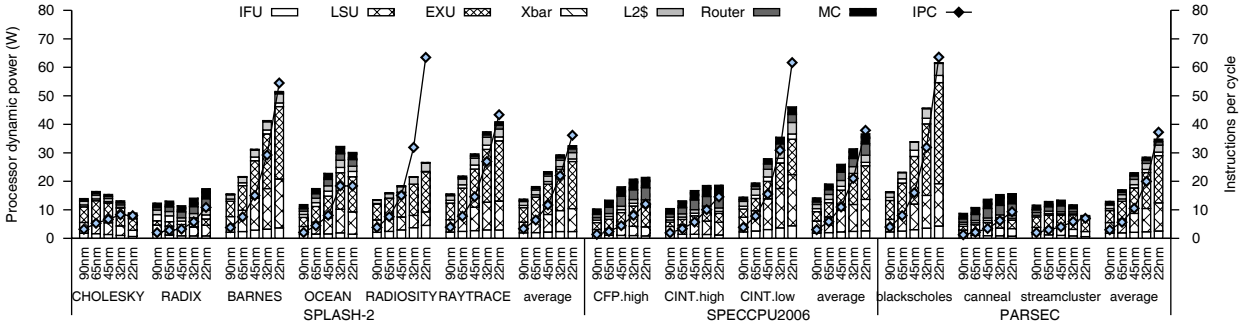
Figure 5 shows power, instructions per cycle (IPC), system energy-delay product (EDP), and EDAP of the 5 system configurations where the technology nodes are changed from 90nm to 22nm. These are all 4-core/cluster configurations. Applications are grouped by three benchmark suites, and in each group there are applications which are not listed due to space limitations, but they are included when average values are computed. Figure 5(a) shows the IPC and the dynamic power of 5 configurations. Applications such as RADIOSITY, CINT.low, and blackscholes are not limited by main memory bandwidth and scale close to linearly with the number of cores in the system. The IPCs of RADIX, CFP.high,

SPLASH-2				SPEC CPU2006	
Application	Dataset	Application	Dataset	Set	Applications
Barnes	16K particles	Cholesky	tk17.O	CINT	
FFT	1024K points	Radiosity	room	high	429.mcf, 462.libquantum, 471.omnetpp, 473.astar
FMM	16K particles	Raytrace	car	med	403.gcc, 445.gobmk, 464.h264ref, 483.xalancbmk
LU	512×512 matrix	Volrend	head	low	400.perlbench, 401.bzip2, 456.hmmer, 458.sjeng
Ocean	258×258 grids			CFP	
Radix	8M integers			high	433.milc, 450.soplex, 459.GemsFDTD, 470.lbm
Water-Sp	4K molecules			med	410.bwaves, 434.zeusmp, 437.leslie3d, 481.wrf
				low	436.cactusADM, 447.dealII, 454.calculix, 482.sphinx3

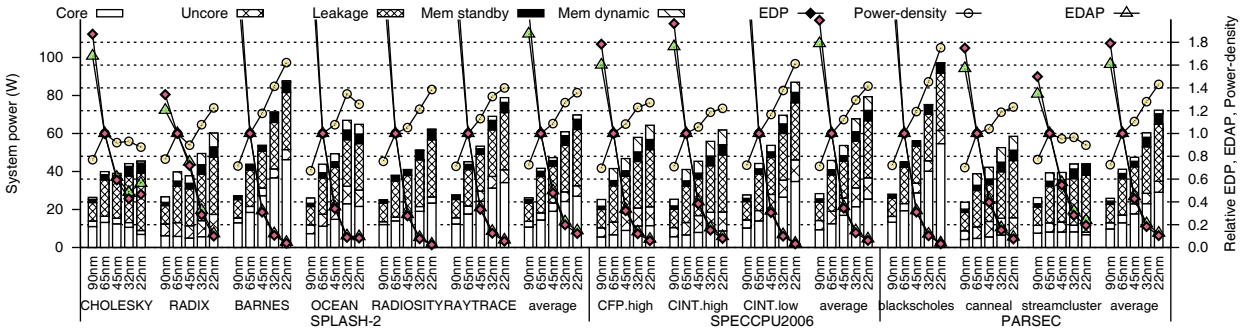
Table 3: SPLASH-2 datasets and SPEC 2006 application mixes for high, med, and low memory bandwidth.

	90nm	65nm	45nm	32nm	22nm
Core area (mm^2)	81.9	96.4	113.4	133.5	157.1
Uncore area (mm^2)	104.3	111.3	102.7	101.6	93.5
Die area (mm^2)	186.3	207.7	216.2	235.1	250.6
Max core dynamic power (W)	24.1	30.7	41.7	48.3	56.4
Max uncore dynamic power (W)	20.6	36.1	45.9	54.5	61.8
Total subthreshold leakage (W)	6.5	11.2	17.6	21.5	25.8
Total gate leakage (W)	2.6	6.7	0.7	1.6	2.5
Chip max power (W)	53.8	84.8	106.0	125.9	146.7

Table 4: Area and maximum power of configurations with 4 cores per cluster across technology generations.



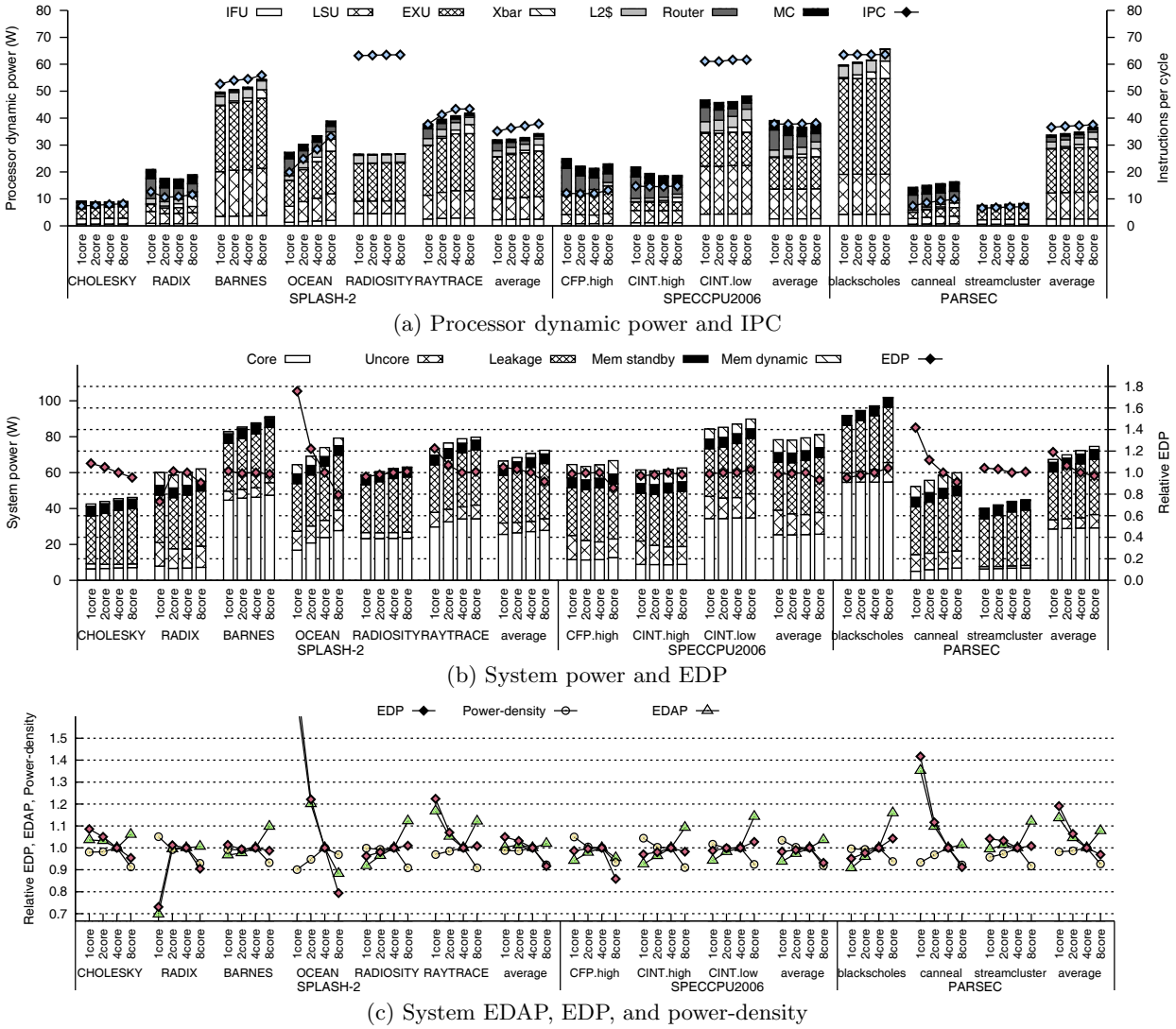
(a) Processor dynamic power and IPC



(b) System power, EDP, and EDAP

Figure 5: Power, power-density, IPC, EDP, and EDAP of the manycore systems while the technology nodes are changed from 90nm to 22nm. For each suite, there are applications which are not listed due to space limitations, but they are included when average values are computed.

CINT.high, and canneal improve relatively slowly since they are limited by main memory bandwidth, which scales worse than computation power. CHOLESKY and OCEAN are the applications with limited IPC scaling because of the insufficient parallelism within applications or small datasets, which in turn leads to the decrease of power density over technology generations. Figure 5(b) shows the system power breakdown, the power density, the EDP, and the EDAP of 5 configurations. The power density, the EDP and the EDAP values are normalized to the values of the 65nm configuration. Many of the applications have inflection points in power density at 65 nm because gate leakage is reduced by



(a) Processor dynamic power and IPC

(b) System power and EDP

(c) System EDAP, EDP, and power-density

Figure 6: Power, power-density, IPC, EDP, and EDAP of the manycore systems while the number of cores per cluster is changed from 1 to 8 on the 22nm technology node. For each suite, there are applications which are not listed due to space limitations, but they are included when average values are computed.

more than 90% from 65nm to 45nm. On many applications, the EDP values improve rapidly as process technology improves. But on CHOLESKY and OCEAN the EDP stops improving after the 32nm process due to limited IPC scaling. The EDAP values scale in a similar way to the EDP values, but change less since the die area increases as shown in Table 4.

In Figure 6, the number of cores per cluster is varied from 1 to 8 on the 22nm technology node. The layout of Figure 6 is the same as that of Figure 5, but here the power density, the EDP, and the EDAP values are normalized to the 4-core per cluster configuration. Figure 6(a) shows that throughout the applications, the intra-cluster crossbar power (Xbar) increases and inter-cluster power decreases as the number of cores per cluster increases. This is expected because the size and power of a crossbar scales super-linearly with the number of cores per cluster. The effects of this clustering on IPC depend heavily on applications. As more cores are grouped into a cluster, the size of the L2 cache that a core can access increases even though more cores share the multi-banked cache, so if multiple cores in a cluster share data (cores use a shared L2 cache synergistically), then an L2 cache can retain more of the combined working sets, hence lowering L2 misses. On OCEAN, RAYTRACE, and canneal, the IPC increases noticeably because of this synergistic cache sharing effect. As a result, the EDP of these applications improves as the number of cores per cluster increases, as shown in Figure 6(b). In contrast, there are applications like RADIOSITY, CINT.low, and blackscholes whose performance is not affected by cache sharing. On these applications, the EDP gets worse as more cores are grouped since the intra-cluster crossbar power increases.

On average, clustering more cores together improves the system energy-delay product. However, if we take the area of the processors into account, the benefits of cache sharing are negated, especially when the number of cores per cluster is 8. In that configuration, the system energy-delay-area product is worse than the configuration with 4 cores per cluster on all benchmark suites on average (Figure 6(c)). RADIX and CFP.high are two applications showing interesting perfor-

mance characteristics, which in turn is reflected in the EDP. On RADIX, when the number of cores per cluster changes from 1 to 2, the L2 miss percentage increases noticeably because cache lines in the L2 caches are evicted more frequently, meaning that cores in a cluster interfere with each other. On CFP.high, cache sharing does not affect performance when the number of cores per cluster is changed from 1 to 4, but the L2 miss rate drops considerably as 8 cores are grouped into a cluster. This is because with fewer cores per cluster, there are noticeable miss rate fluctuations between L2 caches, which are mostly attenuated when 8 cores share an L2 cache.

By using McPAT together with M5 [6], another cycle-accurate performance simulator, we also studied clustering trade-offs of manycore processors with 16 OOO cores at 22nm running at 3.5GHz. Each OOO core is similar to the single-threaded four-issue Alpha 21264 processor but with 32KB 4 way set-associative instruction and data caches. The total capacity of on-chip L2 caches of the OOO manycore is as same as the in-order manycore. The L2 caches are shared within a cluster and coherent among different clusters. The OOO manycore architecture (with fewer, larger cores) can provide similar peak performance and occupy similar die area to the previously described in-order manycore architecture. Specifically, both architectures' ideal IPCs are 64, and the area of the OOO manycore is about 3.5% larger than the in-order manycore when using the same number of cores per cluster. The same subset of PARSEC benchmark suite is used for the OOO simulations.

Figure 7 shows that clustering trade-offs with OOO cores have similar, but magnified, trends as when using in-order cores. For both the in-order and OOO cores running PARSEC benchmarks, if manycore die cost is not taken into account 8-core clusters provide the best EDP. However, 4-core clusters are best for both in-order and OOO cores when manycore die cost is taken into account by using EDA^{2}P and EDAP.

6. CONCLUSION

McPAT is the first tool to integrate power, area, and timing models for a complete processor, including components needed to model multithreaded and multicore/manycore systems. Unlike prior tools which were tightly integrated with specific performance simulators, McPAT uses an XML interface to decouple the architectural simulator from the power, area, and timing analysis, so that it can be readily used with a variety of simulators.

McPAT's power models account for dynamic, subthreshold leakage, gate leakage, and short-circuit power. It is also the first processor power modeling environment to support clock gating and power management schemes with multiple power-saving states. Like CACTI and Orion, McPAT uses MASTAR [38] to calculate device models based on the ITRS roadmap [38], giving it the ability to model not only bulk CMOS transistors, but also SOI and double-gate devices that will become increasingly important at the 32 nm technology node and beyond. Finally, as CACTI did for memory structures, McPAT includes the ability to determine power- and/or area-optimal configurations for array structures and interconnects, given specified targets for the timing value and optimization function. Validation shows reasonable agreement between McPAT's predictions and published data for both in-order and out-of-order processors.

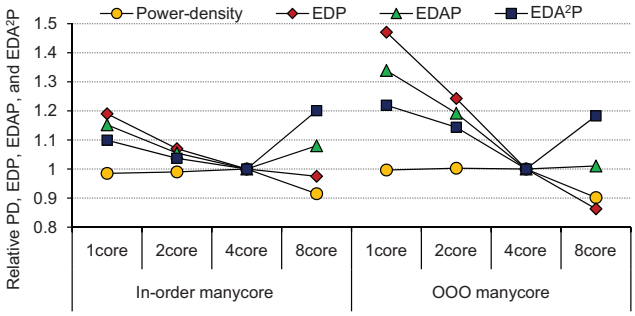


Figure 7: Averaged power-density, EDP, EDAP, and EDA^{2}P of both in-order and OOO manycore architectures at the 22nm technology node running PARSEC benchmarks. Total numbers of cores/threads are 64/256 and 16/16 for in-order and OOO processors, respectively. The number of cores per cluster is changed from 1 to 8 for both in-order and OOO processors.

By providing these capabilities, McPAT supports architects in exploring a broad design space for future multicore and manycore systems. Furthermore, metrics such as energy-delay-area^2 product (EDA^{2}P) and energy-delay-area product (EDAP) that include die cost can now be used. By combining power, area, and timing results of McPAT with performance simulation, we explored the interconnect options of future manycore processors by varying the degree of core clustering over several generations of process technologies. At the 22nm technology node when running PARSEC benchmarks for manycores built from both in-order and out-of-order cores, we found that when cost is not taken into account, clusters of 8 cores provide the best EDP, but when cost is included clusters of 4 cores provide the best EDA^{2}P and EDAP.

7. ACKNOWLEDGMENTS

The authors would like to thank Victor Zyuban and Shyamkumar Thoziyoor at IBM for answering our questions on circuit implementation and the anonymous reviewers for their constructive comments.

8. REFERENCES

[1] K. Agarwal, H. Deogun, D. Sylvester, and K. Nowka, "Power Gating with Multiple Sleep Modes," *ISQED*, 2006.

[2] H.-T. Ahn and D. Allstot, "A Low-jitter 1.9-V CMOS PLL for UltraSPARC Microprocessor Applications," *JSSC*, vol. 35, no. 3, pp. 450–454, 2000.

[3] AMD, "AMD Opteron Processor Benchmarking for Clustered Systems," *AMD WhitePaper*, 2003.

[4] C. Auth, *et al.*, "45nm High-k+Metal Gate Strain-Ehanced Transistors," *Intel Technology Journal*, vol. 12, 2008.

[5] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC Benchmark Suite: Characterization and Architectural Implications," in *PACT*, 2008.

[6] N. L. Binkert, *et al.*, "The M5 Simulator: Modeling Networked Systems," *IEEE Micro*, vol. 26, no. 4, pp. 52–60, 2006.

[7] S. Borkar, *et al.*, "Parameter Variations and Impact on Circuits and Microarchitecture," in *DAC*, 2003.

[8] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," in *ISCA*, 2000.

[9] D. Burger and T. M. Austin, "The simplescalar tool set, version 2.0," *SIGARCH Comput. Archit. News*, vol. 25, no. 3, 1997.

[10] Denali, "Using Configurable Memory Controller Design IP with Encounter RTL Complier," *Cadence CDNLive!*, 2007.

[11] M. K. Gowan, L. L. Biro, and D. B. Jackson, "Power Considerations in the Design of the Alpha 21264 Microprocessor," in *DAC*, 1998.

[12] S. Gupta, S. Keckler, and D. Burger, "Technology Independent Area and Delay Estimates for Microprocessor Building Blocks," UT Austin, Department of Computer Science," Tech. Rep., 2000.

[13] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach. 4th Edition*, 2007.

[14] J. L. Henning, "Performance Counters and Development of SPEC CPU2006," *Computer Architecture News*, vol. 35, no. 1, 2007.

[15] G. Hinton, *et al.*, "The Microarchitecture of the Pentium 4 Processor," *Intel Technology Journal*, vol. 1, 2001.

[16] Intel, "P6 Family of Processors Hardware Developer's Manual," *Intel White Paper*, 1998.

[17] A. Jain, *et al.*, "A 1.2 GHz Alpha Microprocessor with 44.8 GB/s Chip Pin Bandwidth," in *ISSCC*, 2001.

[18] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," in *DATE*, 2009.

[19] R. E. Kessler, "The Alpha 21264 Microprocessor," *IEEE Micro*, vol. 19, no. 2, 1999.

[20] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-Way Multithreaded Sparc Processor," *IEEE Micro*, vol. 25, no. 2, 2005.

[21] D. Koufaty and D. T. Marr, "Hyperthreading Technology in the Netburst Microarchitecture," *IEEE Micro*, vol. 23, no. 2, 2003.

[22] R. Kumar, V. Zyuban, and D. M. Tullsen, "Interconnections in Multi-Core Architectures: Understanding Mechanisms, Overheads and Scaling," in *ISCA*, 2005.

[23] H. Lee, *et al.*, "A 16Gb/s/link, 64GB/s Bidirectional Asymmetric Memory Interface," *JSSC*, vol. 44, no. 4, 2009.

[24] A. S. Leon, K. W. Tam, J. L. Shin, D. Weisner, and F. Schumacher, "A Power-Efficient High-Throughput 32-Thread SPARC Processor," *JSSC*, vol. 42, 2007.

[25] S. Li, J. Ahn, J. B. Brockman, and N. P. Jouppi, "McPAT 1.0: An Integrated Power, Area, and Timing Modeling Framework for Multicore Architecture," HP Labs, Tech. Rep. HPL-2009-206.

[26] C.-K. Luk, *et al.*, "Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation," in *PLDI*, Jun 2005.

[27] P. Mahoney, E. Fetzer, B. Doyle, and S. Naffziger, "Clock Distribution on a Dual-Core Multi-Threaded Itanium Family Processor," in *ISSCC*, 2005.

[28] M. M. K. Martin, *et al.*, "Multifacet's General Execution-driven Multiprocessor Simulator (GEMS) Toolset," *SIGARCH Computer Architecture News*, vol. 33, no. 4, 2005.

[29] S. Mathew, *et al.*, "A 4-GHz 300-mW 64-bit Integer Execution ALU with Dual Supply Voltages in 90-nm CMOS," *JSSC*, vol. 40, no. 1, 2005.

[30] A. Naveh, *et al.*, "Power and Thermal Management in the Intel Core Duo Processor," *Intel Technology Journal*, vol. 10, pp. 109–122, 2006.

[31] U. Nawathe, *et al.*, "Implementation of an 8-Core, 64-Thread, Power-Efficient SPARC Server on a Chip," *JSSC*, vol. 43, no. 1, 2008.

[32] K. Nose and T. Sakurai, "Analysis and Future Trend of Short-circuit Power," *IEEE TCAD*, vol. 19, no. 9, 2000.

[33] S. Palacharla, N. P. Jouppi, and J. E. Smith, "Complexity-Effective Superscalar Processors," in *ISCA*, 1997.

[34] H. Pan, K. Asanović, R. Cohn, and C.-K. Luk, "Controlling Program Execution through Binary Instrumentation," *Computer Architecture News*, vol. 33, no. 5, 2005.

[35] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective; 2nd ed.*, 2003.

[36] A. F. Rodrigues, "Parametric Sizing for Processors," Sandia National Laboratories," Tech. Rep., 2007.

[37] S. Rusu, S. Tam, H. Muljono, D. Ayers, and J. Chang, "A Dual-Core Multi-Threaded Xeon Processor with 16MB L3 Cache," in *ISSCC*, 2006.

[38] Semiconductor Industries Association, "Model for Assessment of CMOS Technologies and Roadmaps (MASTAR)," 2007, http://www.itrs.net/models.html.

[39] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder, "Automatically Characterizing Large Scale Program Behavior," in *ASPLOS*, Oct 2002.

[40] Sun Microsystems, "OpenSPARC," http://www.opensparc.net.

[41] S. Thoziyoor, J. Ahn, M. Monchiero, J. Brockman, and N. Jouppi, "A Comprehensive Memory Modeling Tool and its Application to the Design and Analysis of Future Memory Hierarchies," in *ISCA*, 2008.

[42] D. M. Tullsen, *et al.*, "Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor," in *ISCA*, 1996.

[43] S. Vangal, N. Borkar, and A. Alvandpour, "A Six-port 57GB/s Double-pumped Nonblocking Router Core," in *VLSI*, June 2005.

[44] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: Characterization and Methodological Considerations," in *ISCA*, 1995.

[45] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, "HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects," University of Virgina, Department of Computer Science," Tech. Rep., 2003.