

Microarchitectural Power Analysis for CPU Power/Performance Optimization

George Z. N. Cai

Chee How Lim

*MS JF3-202
Intel Corp.
Hillsboro, OR 97124*

*MS RA1-311
Intel Corp.
Hillsboro, OR 97124*

{chee.how.lim, george.cai}@intel.com

Abstract

This paper presents a technique called microarchitectural power analysis (MPA) for microprocessors. This technique consists of dynamic, static, and multivariate power analysis. It provides the runtime power behaviors and average power consumption analysis, as well as multivariate correlation based power/performance optimization for microprocessors. The SimpleScalar microprocessor simulator is selected as the simulation engine. The default configuration of the SimpleScalar architecture is used as the baseline architectural configuration. The power analysis can be implemented at full-chip level, unit level, and functional block level. Detail dynamic power behaviors can be captured at program runtime. The static and multivariate analysis is the post simulation analysis. The results of power analysis show (1) how an architectural configuration affect the dynamic and static power behaviors; (2) how the architectural performance and maximal clock frequency be affected by the worst-case power consumption, supply current variation, aggressive clock gating impact, and dynamic thermal distribution; and (3) how to optimize the power/performance at functional block level, unit level, and full-chip. Today, high clock frequencies and advanced deep sub-micron processing technologies have created a new dimension in CPU architecture and design: power/performance

optimization. This paper provides the technique that systematically evaluates a new microprocessor architecture in terms of best power/performance optimization.

Introduction

Challenges on Power Efficient Architecture

Rapid increases in clock frequencies and number of transistors per die have dramatically elevated the overall power demands for high-performance microprocessors. High power consumption directly affects the primary goals of the high-performance microprocessor design: sustainable highest clock frequency and deliverable performance. With semiconductor manufacturing technology quickly advancing, transistor scaling will be soon reaching its physical limits. Transistor leakage will become one of the largest power consumption factors in future high-performance microprocessors. The difficulty of future voltage scaling raises great challenges to microprocessors used in mobile and small form-factor desktop computers because of the high power consumption.

Power consumption limitations have impacts on how to architect and design a microprocessor, such as speculation implementation, on-chip system integration, multiple frequency implementation, and multiple voltage domain techniques. Since power consumption is directly related to the clock frequency, the worst-case power constraint can also limit the maximum clock frequency supported. Therefore, microprocessor performance can be restricted due to power consumption issues. In other words, there is a new architectural dimension in the microprocessor architecture: power/performance and power/cost optimizations. One example is the use of branch prediction confidence to

improve speculation accuracy that reduces wasted power consumption [6,7]. The branch prediction confidence may also have the potential to enhance the performance of microprocessors by wisely sharing resources and reducing bus congestion. Therefore, a power-efficient architecture may benefit both power consumption reduction and performance enhancement.

The power limitation of high performance mobile microprocessors is already critical to its design. In addition to extra heat removal costs, high power consumption in mobile microprocessors also reduces the battery lifetime. Hence, a mobile computing system quality and reliability could be affected by its high power consumption. To avoid the microprocessor from overheating, especially on the application programs with high power consumption characteristics, we must exploit new architectures and low-power techniques, which are beyond the traditional clock gating method to further reduce microprocessor power consumption. A power-efficient architecture has a unique architectural value in addition to the absolute performance gain. It will benefit microprocessors in terms of the maximal clock frequency and worst thermal heat reduction, especially in the fastest growing microprocessor market segments: mobile computing and small form factor (slim design) desktop computer markets.

Power Analysis and Power/Performance Optimization

In traditional microprocessor design flow, power analysis and reduction have been addressed in circuit and manufacturing fields. To increase effectiveness, it is necessary to address power consumption issues from the architecture-level in addition to the traditional efforts. To date, power modeling and analysis have focussed mainly in

embedded microprocessors, such as DSP and simple embedded microcontrollers [8]. In this paper, a comprehensive microarchitecture power analysis (MPA) technique is introduced. This technique provides dynamic, static, and multivariate power analysis capability to address power issues in general-purpose microprocessors.

Dynamic power analysis is used for microprocessor runtime power analysis. Over the last few years, dynamic power analysis has become more and more important due to the widely used aggressive clock gating designs and rapidly increasing clock frequencies. This is because when power demands fluctuate excessively, inductive effects will cause circuit performance degradation or malfunction. However, most dynamic power analysis has been in the circuit and manufacturing fields. Microarchitecture analysis tools or techniques must detect if there are dynamic power violations on a given microarchitecture by monitoring cycle-to-cycle power consumption.

Static power analysis is a popular method to estimate average power consumption. It collects the microprocessor architecture total activity statistics during the execution period. By converting those collected architectural activity statistics to the physical activity statistics, the average power consumption can be calculated from those physical activity statistics with physical design parameters, including power densities. The advantage of the static power analysis is that it provides average power consumption for a broad range of applications. It shows basic power behaviors of a given architecture with its target applications, and this is meaningful for battery-life estimation. However, static power analysis conceals the runtime power behaviors of a given microarchitecture. Therefore, it may not reveal the problem caused by the worst-case power consumption.

Multivariate analysis is for power/performance optimization [5]. It uses all measurement data from static and dynamic power analysis as well as the circuit design variables from the design database. It focuses on the correlation among functional block level power consumption, full-chip power consumption, and benchmark performance. To achieve minimum power consumption in a high performance microprocessor, multivariable analysis targets two fundamental questions: (1) How do we identify targets for power reduction within microprocessor architecture and design (there are hundreds of functional blocks within a high performance microprocessor, each being implemented with different circuit design styles). It basically asks where the power is heavily consumed and why. It challenges the entire microprocessor architecture and micro-architecture from the power consumption's point of view. (2) How do we reduce power consumption at the identified architecture and design targets with no performance impact. It involves new tradeoffs in the microarchitecture between the performance and power consumption. Traditional microprocessor architecture tradeoff decisions are only based on cost/performance analysis. These decisions are good for microprocessor performance. They may or may not be good for power savings. As power/performance optimization becomes a new dimension for microprocessor design, power consumption requirement becomes critical. The architecture and design tradeoff criteria and priority for the mobile and small form factor PC microprocessors are very different as compared to traditional microprocessors for desktop computers. The conventional microprocessor architecture and design analysis cannot provide enough information to make optimal design decisions for power efficient microprocessors.

Related Work

There have been quite a few papers and books on low power design or low power VLSI methods [1,2]. They focused on the low power circuit implementations. Though the microarchitecture impacts were briefly discussed, there is no architecture-level power simulation, estimation, and analysis in those papers. There were several excellent papers on analytic power modeling [2,8]. However, there is no general-purpose CPU microarchitecture detail in those models. Those models emphasized on transistor-level dynamic characteristics. There are several reports on embedded microprocessor and DSP RTL level power modeling and simulation [8]. The microarchitectures were specifically for DSP. It does not provide the capability to run SPEC, Winstone, Sysmark and other general-purpose microprocessor benchmarks. The papers from Dirk Grunwald's research group had for the first time addressed speculation accuracy and its power consumption impact. Power/performance tradeoffs were also discussed in the papers [6,7]. Unfortunately, there were no architecture-level dynamic power model and functional block level power behavior analyses.

Dynamic Microarchitecture Power Analysis

The dynamic microarchitecture power analysis is a runtime power behavior analysis based on architectural simulations. This activity-sensitive power analysis contains two parts: a cycle-based microprocessor performance simulator, and a dynamic power model. The cycle-based performance simulator provides architectural activity information of a given microprocessor as well as the simulation engine. Based on the given microprocessor architecture, the fundamental design decision must be made for the dynamic power model, such as architectural partition for design. Each design unit

consists of a number of functional blocks. Each functional block is implemented by certain circuit styles with a selected VLSI layout floor plan. The number of transistors and the type of circuits will occupy different silicon area. Different circuit design styles will achieve different propagation delays and consume different amounts of power. Therefore, the given microarchitecture and circuit delay requirement can be used to track the potential power consumption.

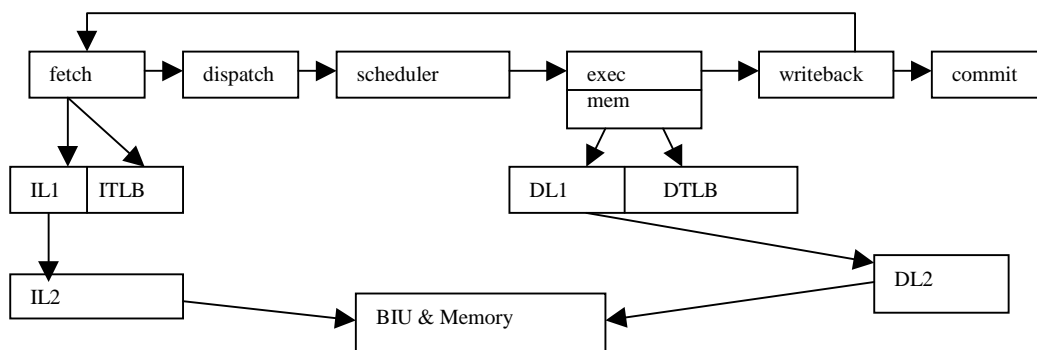


Figure 1 SimpleScalar Architecture Pipeline and Partition

The SimpleScalar architecture pipeline [9] can be viewed as the above diagram. The logic partition of this pipeline is in Figure 2.

Dynamic power monitors are parts of the dynamic power model. They are integrated into each functional block within the architectural performance simulator. These monitors work interactively with the architectural performance simulator to produce the runtime power consumption information. The dynamic power model will carefully translate the power monitor information into effective active factors for each functional block. The dynamic power model also consists of each functional block area, effective active area, active power density, and inactive power density. The dynamic power analysis will collect the number of di/dt threshold violations, the number of the power threshold

violations, the maximum di/dt , and the maximum power consumption at each functional block and the full-chip level. All those information is very useful for selecting a microarchitecture, making the optimal design decisions for cost/performance and power/performance.

In this paper, we selected the SimpleScalar simulator as our basic simulation engine because it has been widely used in microprocessor research community and represents a typical modern microprocessor architecture. As in the case of real microprocessor design, we first partitioned the SimpleScalar microprocessor architecture into 6 design units: fetch unit, decode unit, scheduler, execution unit, memory unit, and bus interface unit [9]. As Figure 2 shows, there are 34 functional blocks that have relatively independent architectural functions and design requirements. The corresponding power consumption are very different, such as the power consumption of data cache array is different to the power consumption of integer execution unit adder. The functional block design requirements provide the basis of logic design and circuit implementation. We can estimate the dynamic power consumption of a functional block from its logic design and circuit implementation. For example, SimpleScalar's memory unit can be partitioned into the following functional blocks as indicated on Table 1.

| Function block name | Basic architecture functions |
|---------------------|------------------------------------|
| MEPMHUDLOG | Page missing handler logic |
| METLBINCAC | Instruction TLB |
| MEIL1NCAC | L1 instruction cache data array |
| MEIL1NTAG | L1 instruction cache tag array |
| MEIL1NLOG | L1 instruction cache control logic |
| METLBDACAC | Data TLB |
| MEDL1DACAC | L1 data cache data array |
| MEDL1DATAG | L1 data cache tag array |
| MEDL1DALOG | L1 data cache control logic |
| MEUL2IDCAC | L2 cache data array |
| MEU2IDTAG | L2 cache tag array |
| MEUL2IDLOG | L2 cache control logic |

Table 1 Functional Blocks of Memory Unit for Power Modeling

The table 1 shows the list of functional blocks that were created in the dynamic power model as the memory unit based on the simple scalar architectural simulator. They consist of the data array, tag array, and control logic of Instruction TLB, Data TLB, L1, and L2 caches.

To effectively describe the dynamic power behaviors, we introduce the following terms:

- ***di/dt threshold*** (DT): di/dt is the supply current difference during a unit time. It is a function of a microprocessor execution time and the circuit supply current at a given microprocessor functional block. The di/dt threshold is a given measurement to setup the limit of the maximum supply current change within a given period of time.

When microprocessor logic is in the transition from one state to another state, the circuit implementation to achieve the state transition must have its required current supply. Otherwise, the circuit can not implement the defined logic function correctly within the given time period. High-performance microprocessors make this requirement critical due to its very fast state transition

to support fast core clock frequencies. Multiple frequency domain implementation and aggressive clock gating methods may generate a huge spike on the supply current. These design choices require di/dt measurement as a significant criterion when evaluating a new microarchitecture.

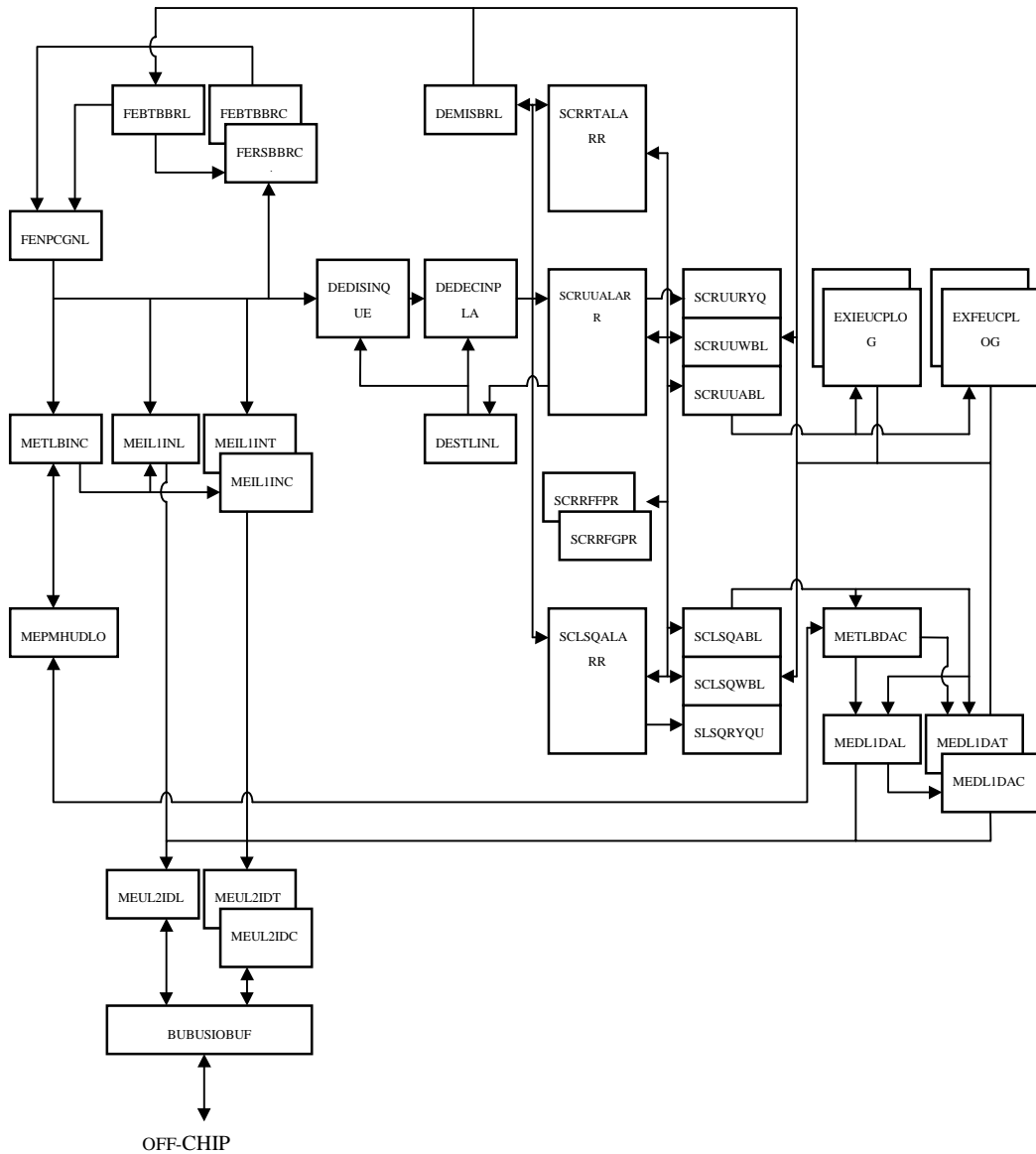


Figure 2 Partition of SimpleScalar Architecture for Dynamic Power Model

The simple scalar architecture was partitioned as 6 units and 34 functional blocks. They are fetch unit, decode unit, scheduler, execution unit, memory unit, and bus interface unit. Each functional block represents relatively independent architectural functions and design requirements.

- ***Power threshold*** (PT): The CPU dynamic power consumption is a function of CPU execution time, microarchitecture, process technology, and runtime instruction streams. The power threshold is a given measurement to setup the limit of the maximum power consumption during a microprocessor execution.

When a microprocessor executes different instruction streams, the power consumption of a given functional block or the entire chip can be very different. The power threshold violation within the microprocessor can directly affect the maximum microprocessor frequency, chip thermal distribution, and packaging. The silicon may not work correctly as a result of overheating if the power threshold violation exists. Therefore, microarchitecture must avoid power threshold violations.

- ***Dynamic power monitor*** (DPM): It consists of a group of runtime counters and procedure calls. These counters and procedure calls interactively work with the architectural simulator to record the total number of di/dt threshold (DT) and power threshold (PT) violations. The dynamic power monitor can also record the maximum di/dt and power consumption. The violation distribution is an option for the dynamic power monitors. The dynamic power monitor in a functional block captures all architectural activities regarding to the power consumption at the runtime.

- ***Effective activity factor*** (EAF): It contains the numbers of performance parameters and the dynamic power monitor variables. The effective activity factor normalizes the power consumption effects from its dynamic power monitor counts, performance activities counts, affected areas, and power densities. It represents the actual architectural activities that may have impact on the power consumption. It is one of the most important power model parameters.

The effective active factor considers all possible activities based on the fact that power sensitive activities and performance sensitive activities may not be same. For example, the access of a virtual addressed data cache and its corresponding data TLB may occur at the same time. The DTLB access is a speculative access. If the virtual data cache access is hit, the speculative DTLB access will be either abort or cancelled. These cancelled DTLB accesses do not have any significance to the microprocessor performance simulator. However, it does have the real impact on the power consumption. Therefore, the dynamic power model must count any DTLB access because any DTLB access does activate the bank selection logic and tag logic circuit, and it does consume the power.

- ***Effective area*** (EA): It may consist of the numbers of circuit area within a given function block layout area. These circuits may have different characteristics and power consumption.

For example, the static circuits and dynamic circuits may have very different power consumption. The clock distribution tree within a functional block can be spread every where. It is hard to estimate the actual area of the clock distribution within a functional block. However, the clock distribution does consume the power.

- ***Active power density*** (APD): It is a measurement of the power consumption per unit area within a functional block during execution. The active power density is one of the most important parameters for active power consumption estimation. It can be obtained from either the circuit power analysis or existing silicon measurement or other methods. If an active power density has multiple circuit design style involved, such as dynamic datapath circuit and static decoding logic, the active power density provides an estimation of the average power density among all involved circuits with careful calibration on each involved circuit.
- ***Inactive power density*** (IPD): It is a measurement of the power consumption per unit area within a functional block when it is inactive. The inactive power density is one of the most important parameters for the power estimation. It can be obtained from the circuit power analysis, existing silicon measurement, semiconductor process profile, or other methods. It consists of subthreshold leakage and other leakage current caused by low transistor threshold voltage. With current aggressive transistor scaling, the gate oxide of the transistor will

soon reach its physical limit. The leakage current is going to be a major factor in full-chip power consumption in near future.

The dynamic power consumption of a functional block at the execution time t may be calculated by the following equation.

$$P(t) = \sum_{k=0}^{k=c} \left\{ \left[\sum_{i=0}^{i=n} \alpha_{(k,i)} X_{(k,i)} Y_{(k,i)} \right] + \left[\sum_{j=0}^{j=m} \beta_{(k,j)} Y_{(k,j)} \right] \right\}$$

Here P is the dynamic power consumption at the execution time t . c is the total number of functional blocks within a CPU design. n is the total number of active effective area within a functional block at the execution time t . m is the total number of inactive effective area within a functional block at the execution time t . The index k, i, j , represent the functional block index, the active effective area index, and the inactive effective area index, respectively. α is the active power density of i th active effective area within the functional block k ; β is the inactive power density of j th inactive effective area within the functional block k . X is the effective active factor. Y is the effective area within a functional block.

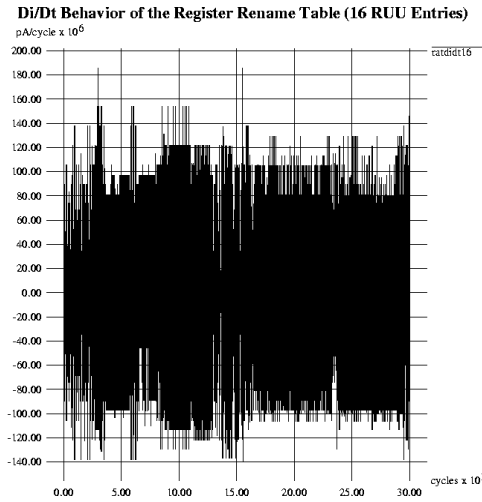


Figure 3 Di/Dt Behaviors of a Given RRT

This is a traditional RRT with capability to allocate 16 RUU and 8 LSQ entries. There are five fields on each entry. The di/dt behaviors of the RRT were recorded at the initial 30,000 cycle interval during SPEC95 gcc execution.

Figure 3 shows the di/dt behaviors of Register Rename Table (16 RUU entries) during initial 30,000 cycle interval of SPEC95 gcc execution. If the value of di/dt threshold, DT,

is 200 μA , then there is no violation. If the value of di/dt threshold is 160 μA , then there will be 2 violations. We also see many large di/dt swings during the initial 2,000 cycles. This represents rapid fluctuations in supply current demands by the circuits. The di/dt is observed to be relatively stable after 16,000 cycles.

Figure 4 shows the dynamic power behaviors of Register Rename Table. We can see that there are huge power variations. The power consumption of RRT is relatively stable after 7,000 cycles. If we set the dynamic power threshold to be 190 μW , there is no dynamic power violation. If we set the dynamic power threshold at 140 μW , there are 15 dynamic power violations. The dynamic power consumption may not be identical to the di/dt changes because the dynamic power change is not rapid.

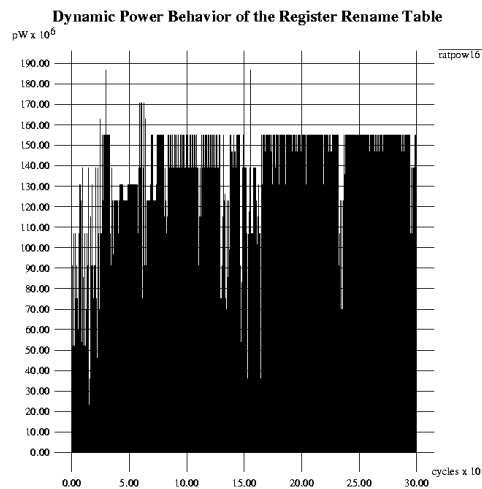


Figure 4 Dynamic Power Behaviors of a Given RRT

This is a traditional RRT with capability to allocate 16 RUU and 8 LSQ entries. There are five fields on each entry. The dynamic power behaviors of the RRT were recorded at the initial 30,000 cycle interval during SPEC95 gcc execution.

When the RUU entries are reconfigured from 16 to 32, we can see that the di/dt variation is getting large at the beginning 10,000 cycles (refer to Figure 5). The di/dt stays in the same range as di/dt for the 16 entry RUU. However, the average di/dt value is larger than one of the 16 entry RUU. It means the average current variation is larger in the configuration of the 32 RUU entry microarchitecture than one at the 16 RUU entry architecture.

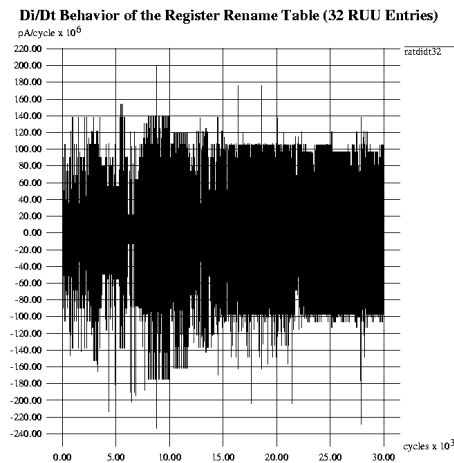


Figure 5 Dynamic Behaviors Comparison of a Given RRT

The dynamic behaviors are changed due to the RUU microarchitectural changes. The RRT dynamic behaviors were recorded at the initial 30,000 cycle interval during SPEC95 gcc execution.

The full-chip supply current and power consumption behaviors will be decided by the supply current and power consumption of all functional blocks. If most blocks have the large di/dt swing at the same period of time, the full-chip will have huge di/dt swing. Then the implementation of this architecture will be very difficult if not impossible. If the architecture can reduce the risk of the dynamic power violation, then the full-chip microprocessor may have a high clock frequency implementation. It means higher

microprocessor performance can be achieved as compared to one that has many potential dynamic power violations.

Static Microarchitecture Power Analysis

The static microarchitecture power analysis is a method to estimate the average power consumption of a microprocessor under different workloads. The analysis is able to provide the full-chip as well as functional block power consumption. This static power estimation provides an overview of a given microarchitectural power consumption characteristics, and it is especially useful when the workloads have high power characteristics. The workload can be categorized as office applications, games and multimedia applications, scientific computational applications, SPEC benchmarks, etc. Static power analysis can be very useful for microarchitecture-related power/performance optimization. The static power analysis can be viewed as a special case of the dynamic power analysis. Since the static power consumption is the average power consumption, it can be obtained by adding all dynamic power consumption together and divided by the execution time.

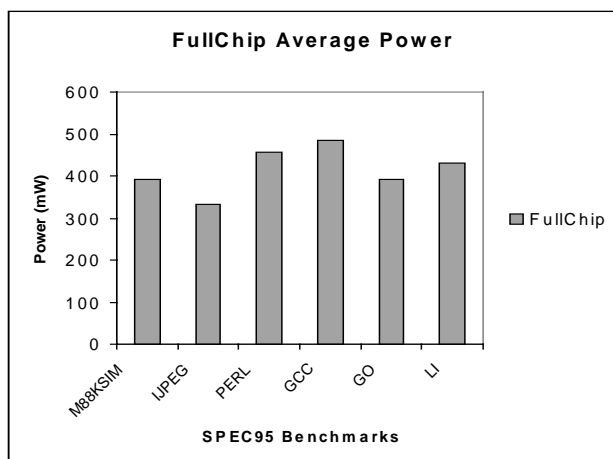


Figure 6 Full-Chip Static Power Consumption

The full-chip average power consumption changed when the microprocessor executes different workloads. The above indicates 6 SPEC95 benchmark average power consumption. Each benchmark ran 5 million instructions on the baseline SimpleScalar simulator.

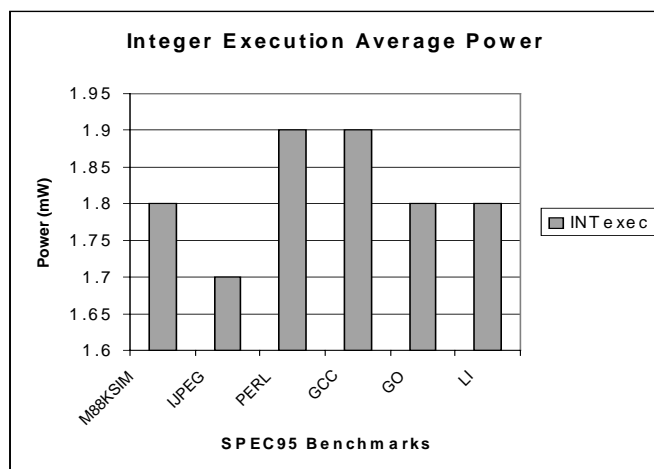


Figure 7 Integer Execution Unit Static Power Consumption

The average power consumption of the integer execution unit varied when the microprocessor executes different workloads. The average power consumption of 6 SPEC95 benchmarks show at the above. Each benchmark ran 5 million instructions on the baseline SimpleScalar simulator.

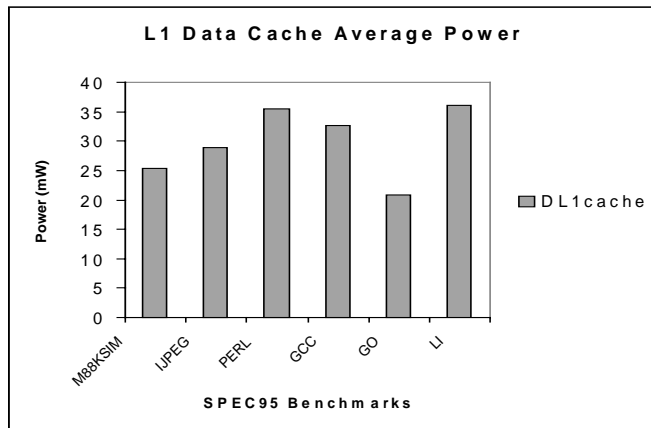


Figure 8 L1 Data Cache Static Power Consumption

The average power consumption of L1 data cache (4 way 16KB Write-through) varied when the microprocessor executes different workloads. The average power consumption of 6 SPEC95 benchmarks show at the above. Each benchmark ran 5 million instructions on the baseline Simplescalar simulator.

Multivariate Analysis and Optimization

Multivariate power/performance analysis helps microprocessor architects and designers to identify the possible power reduction targets within complex microprocessor architectures. It also provides an easy and quick way to predict potential power consumption and performance impact due to correspond architectural and design changes. The power consumption and performance prediction is used to keep the architectural modifications on the correct direction before complex detailed microprocessor power/performance simulations carry out.

| Functional block name: A | Full-chip power correlation (correlation coefficient) | Performance correlation (correlation coefficient) | Unit power correlation (correlation coefficient) | Correlation with variables from interfaced functional block [correlation coefficient (percentage of interface signals)] |
|------------------------------|---|---|--|---|
| Microarchitecture (option 0) | 0.95 | 0.20 | 0.94 | >0.80 (7%); >0.50 (10%); >0.20 (60%); >0.0 (100%) |
| Circuit Implementation A | Reduce 2% full-chip power | No performance change | Reduce 10% block power | No significant changes on correlation of interface blocks |
| Circuit Implementation B | Reduce 3% full-chip power | -1% average performance change | Reduce 20% block power | No significant changes on correlation of interface blocks |
| Microarchitecture (option 1) | 0.93 | 0.89 | 0.94 | >0.80 (20%); >0.50 (20%); >0.20 (25%); >0.0 (100%) |
| Circuit Implementation C | Increase 1% full-chip power | Increase 0.5% performance | Increase 5% block power | >0.80 (30%); >0.50 (30%); >0.20 (35%); >0.0 (100%) |
| Circuit Implementation D | Increase 2% full-chip power | -1% average performance change | Increase 20% block power | No significant changes on correlation of interface blocks |

Table 2 A Functional Block Level Multivariate Analysis

This is a typical functional block level multivariate analysis. The proposed microarchitecture and potential circuit implementation are evaluated with the correlation to the full-chip power consumption, functional block power consumption, average performance, and interfaced functional block impacts. Here the correlation is from 1 (strongest positive correlation) to -1 (strongest negative correlation). From the table, the microarchitecture (option 0) with its circuit implementation A is the best power/performance design among all microarchitecture and their circuit implementations.

The multivariate analysis uses the results of the dynamic and static microarchitectural power analysis as the inputs. The correlations of multiple microprocessor performance and power variables are analyzed. We use the SimpleScalar simulator to collect a group of architectural performance data and their corresponding power consumptions. The multivariate analysis is not only interested in their central tendency and variation of each individual performance and power consumption variables. It is also interested in assessment of the association among the performance and the power consumption, such as a full-chip performance and power consumption, full-chip performance and functional block power consumption. Mathematically, the relationships are measured by correlation

coefficients, which can have values between -1 and 1 , inclusively. A strong relationship is indicated by a large positive or negative correlation coefficients while a weak relationship is indicated by a correlation coefficient that is close to 0 . The correlation coefficient by itself cannot determine if an architectural parameter has a high power consumption or performance. Instead, it only indicates that an architectural performance variable is a tight grip on either the full-chip or functional block power consumption.

The correlation coefficients of those variables are not systematically controlled by architects or designers, but by complex factors that need not be precisely determined. In other words, the multivariate coefficient values represent the fact of power consumption and performance from compounded multiple effects on complex microprocessor architectural and design factors. This is one of the reasons that multivariate power/performance analysis is different from other power and performance analysis methods.

The multivariate power/performance analysis does not assume that any particular performance variable has a causal logic relation with power consumption at full-chip level. By observing how the performance and power consumption vary with each other in a set of experiments, multivariate analysis tends to have the effect of summarizing complex relationships among variables without deriving all possible complex equations. However, it does not establish a causal relation between the power consumption and performance variables. In the multivariate power/performance analysis, we do not make any causal logic relation assumption during the correlation analysis. This assumption reflects that the power consumption and performance of a microprocessor is related to many architectural and design issues. The simple causal logic relation may not always fit

in a real microprocessor design. The correlation studies could be the common, inexpensive, and valuable power/performance studies in general.

Conclusion and Future Research

The paper presents a systematic analysis method for power/performance optimization. Based on the SimpleScalar architectural simulator, this paper showed how to create a dynamic microarchitecture power simulator. The dynamic power simulator can capture the microprocessor architecture activities and the corresponding dynamic and static power behaviors. The dynamic power analysis shows the runtime microprocessor power behaviors, especially on the supply current variation and dynamic power variations. The static power analysis provides the average power consumption crossing benchmarks with millions instructions. The power analysis can provide meaningful data from the functional block level to the full-chip level. The performance and power simulation results are used for the multivariate analysis. The performance and power consumption correlation is evaluated for power/performance optimization.

Power/performance is a new architecture and design dimension for microprocessors in addition to conventional cost/performance tradeoffs. With rapid increase in microprocessor clock frequency and aggressive transistor scaling, power/performance optimization becomes an important and significant architecture issue. With finite heat removal capacity, higher power consumption and higher di/dt variations translate to either higher cost or lower maximum clock frequency.

We plan to improve the interface flexibility and accuracy of the microarchitectural power simulator based on the SimpleScalar simulator. We also plan to use the power analysis to exploit new energy-efficient architectures, such as reduction of speculation power consumption. We hope the power simulator and power/performance analysis has a contribution on developing new energy efficient microprocessor architectures.

Acknowledgement

The authors like to give our special appreciation to Todd Austin for many useful discussions on how to build a dynamic power simulator based on the SimpleScalar simulator. The authors would like to thank Tosaku Nakanishi and Greg Taylor for their support from the beginning of the project. We thank the encouragement from professors W. Robert Daasch, Brad Calder, Dean Tullsen, and Trevor Mudge. We would also like to thank Steve Gunther, Doug Carmean, Micah Barany, John Hall and many of our colleagues for their pioneering research and work on power/performance optimization at Intel Mobile and Handheld Product Group, Logic Technology Development, and Performance Microprocessor Division.

Reference

- [1] Gary K. Yeap, "Practical Low Power Digital VLSI Design", Kluwer Academic Publishers Group, 101 Philip drive, Assinippi Park, Norwell, Massachusetts 02061, 1998.
- [2] Paul Landman and Jan Rabaey, "Activity-Sensitive Architectural Power Analysis", IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems, 1996.
- [3] Steven J. E. Wilton and Norman P. Jouppi, "An Enhanced Access and Cycle Time Model for On-chip Caches", Digital Western Research lab WRL Research Report 93/5. Compaq Western Research Laboratory, 250 University Avenue, Palo Alto, California 94301. July 1994.
- [5] George Cai, etc, "Multivariate Analysis for high performance mobile processors", the proceedings of ISCA 1st Power driven microarchitecture workshop, Barcelona, Spain, June 1998.
- [6] Dirk Grunwald, Artur Klauser, Srilatha Manne and Andrew Pleszkun, "Confidence Estimation for Speculation Control", the proceedings of 25th International Symposium on Computer Architecture. Bacerlona, Spain, 1998.
- [7]. Srilatha Manne, Artur Klauser and Dirk Grunwald, "Pipeline Gating: Speculation Control For Energy Reduction", the proceedings of 25th International Symposium on

Computer Architecture. Bacerlona, Spain, 1998.

[8] Ben Klass, Don Thomas, Herman Schmitt, David Nagle, "Modeling Inter-Instruction Energy Effects in a Digital Signal Processor", the proceedings of ISCA 1st Power driven microarchitecture workshop, Barcelona, Spain, June 1998.

[9] D. Burger and T. M. Austin, "Evaluating Future Microprocessors: the SimpleScalar Tool Set," Technical Report #1342, University of Wisconsin, June 1997.

[10] Intel Celeron Processor Datasheet, Chapter 4 Thermal Specifications and Design Considerations, Intel Corp. 2200 Mission College Blvd. Santa Clara, California 95052, 1999.