

A Temperature-Aware Power Estimation Methodology

Madhu Saravana Sibi Govindan, Stephen Keckler, Sani Nassif*, Emrah Acar*

Computer Architecture and Technology Laboratory
Department of Computer Sciences
University of Texas at Austin
sibi, skeckler@cs.utexas.edu

*IBM Austin Research Laboratory, Austin
nassif,emrah@us.ibm.com

September 2, 2008

Department of Computer Sciences
Technical Report TR-2007-43
The University of Texas at Austin

Abstract

Reducing power consumption, improving designer productivity and mitigating thermal effects are grand challenges for future CMOS-based designs in the nanometer regime [1]. Solving these challenges requires a power estimation methodology that is temperature aware and simple, fast and accurate. In this paper, we present such a power estimation methodology that utilizes data from different levels of modeling abstraction and is applicable to both current and future processors. Our methodology leverages design data from the gate-level model and activity factors from the structural RTL model and refines the initial power estimates based on a thermal and power grid model. We demonstrate our methodology using a SOC-style, tiled, general purpose, chip multiprocessor implemented at 130nm and provide scaled-down estimates at 90nm, 65nm, 45nm and 32nm technologies.

1 Introduction

Increasing power consumption in modern day microprocessors not only affects battery life in mobile platforms, but also increases packaging and cooling costs in desktop and server platforms. Therefore, designing power-efficient processors has been a main focus of research in industry and academia. Efficient and accurate power estimation methodologies are required for designing power-efficient processors.

Power estimation can be performed with various models of a design: high-level analytical models, C-based architectural models, structural RTL (Register Transfer Level) models, gate-level models with and without layout data and circuit-level models. Estimating power using each of these models has its own advantages and disadvantages. For example, power estimation using a C-based model is useful for design-space exploration. The process is fast and designers can estimate activity factors (switching activity for various elements in the design) from realistic, long running workloads. However, such C models lack accurate design information such as the capacitance data. On the other hand, a detailed transistor model of the design combined with accurate circuit simulation gives more accurate power estimates. But, circuit-level and gate-level simulations are prohibitively slow. Therefore, a power estimation methodology that utilizes the right set of data from the right model or abstraction level is desirable. For example, design data can be obtained from a gate-level netlist and activity factors of realistic workloads can come from a more abstract model. In this paper, we present one such methodology which leverages design data from a gate-level netlist, but uses activity factors from a structural RTL model.

Power dissipation of a chip is closely related to and is affected by the temperature of the chip. For example, dynamic power dissipation increases the temperature of the chip, which in turn increases leakage power dissipation. This is due to a non-linear dependence of leakage power on temperature. So, it is essential for any power estimation methodology to include temperature in its estimates. Our methodology generates temperature-independent power estimates first, which are later refined by an IBM tool to generate temperature-aware power estimates.

Our methodology can also be used to study how the power estimates of a current generation processor scales when the same design is scaled to future process technologies. This is made feasible because the methodology leverages low-level data from a gate-level netlist and floor-planning information. We also present a simple methodology for such power scaling studies. We discuss constant transistor scaling where the number of transistors in the design is kept constant when scaling.

We make the following contributions in this paper:

- We present a simple and fast power estimation methodology that utilizes design data from a gate-level model of the design and activity factors from a structural RTL model. We also refine the power estimates that are initially temperature-independent to generate temperature-aware estimates.
- We also present a simple power scaling methodology to study how the power consumption of a chip changes with process technologies.
- Finally, we present a case study by applying the above methodologies to an SOC-style, tiled, ASIC-designed, chip multiprocessor that was implemented at 130nm. We present the power estimates at 130nm along with scaled-down estimates for 90nm, 65nm, 45nm and 32nm.

The rest of the paper is organized as follows: section 2 discusses related work. Section 3 explains the methodology for temperature-independent power estimation, temperature-based refinement of those estimates and power scaling studies. Section 4 presents a case study by applying these methodologies to TRIPS processor [2], an SOC-style, tiled, chip multiprocessor. Some of the limitations of this study and future enhancements are listed in section 5. Section 6 concludes the paper.

2 Related Work

There are several power estimation techniques in the literature. Bernacchia et al. [3] and Gupta et al. [4] suggest analytical approaches to high-level power estimation. These methods use analytical models in combination with characterization of various circuits to estimate power. These methods are fast, but they require characterization of the circuits. Wattch [5], an extension to SimpleScalar, was developed for analyzing the power trade-offs between different micro-architectural configurations and studying the effects of compiler optimizations on power. Simplepower [6] is a cycle-accurate architectural-level power simulator, which simulates an in-order 5-stage pipeline. These tools use a C-based model of the design and power models to estimate power. These C models are useful during early design-space exploration and can provide activity factor estimates for long running workloads.

Najm lists several structural RTL power estimation techniques in [7]. Tools like Primepower [8] perform power estimation both at the structural RTL and gate levels. Power Theatre [9], and PowerCompiler [8] perform power estimation and power-optimized synthesis of a design. Other circuit simulation tools like Nanosim, HSPICE [8] and SPICE [10] can also be used for low-level power estimation. The advantage of these tools is the accuracy of the estimates. Given the right set of parameters, these estimates are very close to the actual power values. However, many of these tools require detailed transistor models of the design and gate level or circuit simulations, which are very slow. Moreover, performing circuit simulation or gate-level simulation of a full chip is extremely difficult, often intractable.

Many of the above-mentioned methods do not refine their power estimates based on a thermal model. However, tools like Hotspot [11] can be used with Wattch or SimpleScalar to add a thermal model of the design. As mentioned before, our methodology gathers accurate design data from a gate-level netlist and activity factors from a structural RTL model. Also, our methodology refines the initial temperature-independent power estimates based on a thermal and power grid model to generate temperature-aware estimates. The thermal model includes a model of the heat-sink and the chip package, whereas the power grid model is obtained from floor-planning information.

3 Methodology

This section explains our methodology for temperature-aware power estimation and power scaling studies.

3.1 Temperature-Aware Power Estimation

The flowchart in figure 1 shows the overall methodology. There are two steps in the methodology. The first step generates power estimates for various micro-architectural blocks of the design. In the second step, the power estimates are provided as input to an IBM tool called LAVA [12], which provides refined, temperature-aware power estimates. Power estimates from step I leverage data from a gate-level netlist of the design, the technology library and a methodology for estimating the activity factors. Step II uses data from a thermal model, a power grid model and floor-planning information to provide refined results.

3.1.1 Step I: Power Estimation

The total power consumption is estimated as the sum of dynamic and leakage power.

Dynamic Power Estimation: Dynamic power depends on the total capacitance, the activity factor (average number of toggles per cycle), the supply voltage and the clock frequency of the design. The value of supply voltage is obtained from the technology documentation. Designers have an early estimate of the targeted frequency, which gets refined during the late stages of the design. Moreover, chips typically have multiple voltage/frequency settings and dynamically switch among those settings. This work assumes a specific voltage/frequency setting for its power estimates.

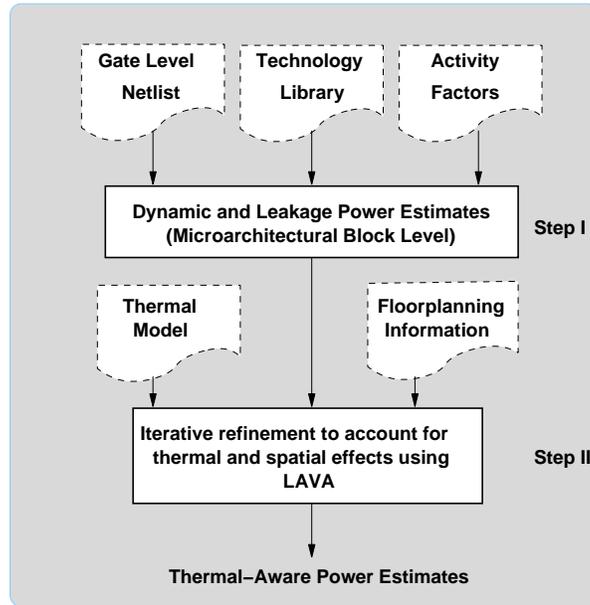


Figure 1: Integrated Power estimation methodology

Capacitance: The major components of a chip that contribute to dynamic power dissipation are clock tree (buffers and interconnect), logic (gates and latches), array structures (SRAMs and register arrays), interconnect and I/O drivers. The capacitance of all the logic (gates and latches) is estimated from a gate-level netlist and the technology library. The fraction of the logic capacitance that switches every clock cycle (for e.g., clock inputs to latches) is distinguished from the logic capacitance that does not switch every cycle. The capacitance of all the I/O drivers, the capacitance of clock buffers, regular array structures like SRAM arrays, register files are also calculated from the technology library. Stroobandt et al. [13] suggested a methodology for estimating the average interconnect length of the modules in a design. This methodology is based on Rent’s rule, which relates the number of pins of a module to the average number of interconnects in that module. Using Stroobandt’s methodology, we estimate the average interconnect length of all the micro-architectural blocks. Then, we determine the number of interconnects in the block (from the gate-level netlist). This is multiplied by the average interconnect length to get the total interconnect length. The total interconnect length is used to estimate the total interconnect capacitance of the block.

Activity Factors: The flowchart in figure 2 shows the methodology [8] for estimating activity factors of various micro-architectural blocks. The block-level functional verification infrastructure is used for estimating activity factors. This is a randomized verification infrastructure that tests all the functionalities of a given micro-architectural block. A structural RTL simulation tool monitors activity factors of all objects in a list of objects (created during synthesis) and produces activity factors of all primary inputs, hierarchical ports and sequential element outputs. This activity factor information from the structural RTL model is propagated to a gate-level netlist. We use a synopsys tool to perform this propagation and to provide the gate-level activity factors. We repeat this process multiple times (2 to 5 times) with different input sets and measure the average activity factor of each micro-architectural block. While this methodology is less accurate than performing a gate-level simulation, it is much faster than gate-level or circuit-level simulation.

Leakage Power: Sub-threshold and gate-tunneling leakage currents are the most dominant sources of leakage current [12]. We estimate leakage power using the following formula:

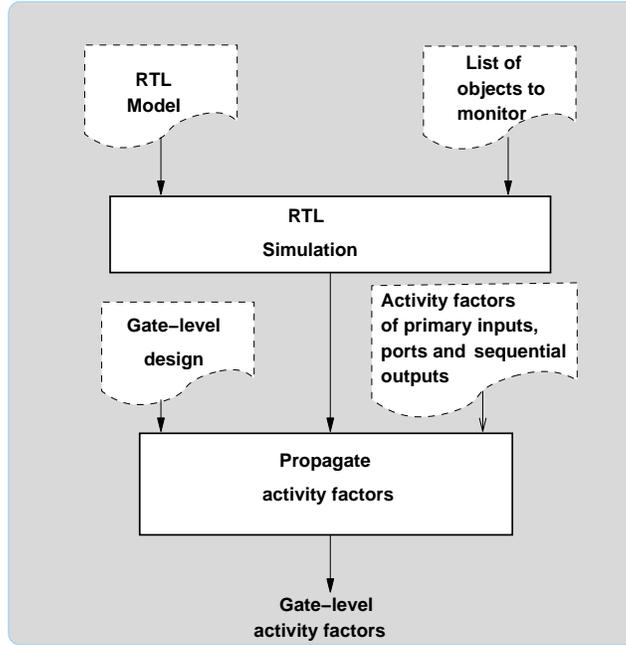


Figure 2: Estimation of gate-level activity factors.

$$P_L = V_{dd} * [(W_p + W_n) * [I_{sub} + L * J_{gate}]] \quad (1)$$

where V_{dd} is the supply voltage, W_p and W_n are the total PFET and NFET widths in the design, I_{sub} is the sub-threshold leakage current per unit width, L is the minimum gate length of the design, and J_{gate} is the current density. The PFET and NFET widths of the cells are specified in the technology library. Using the list of cells present in the design, the total PFET and NFET widths in a micro-architectural block is calculated. Section 3.2 explains the estimation of I_{sub} and J_{gate} .

3.1.2 Step II: Modeling Temperature, Leakage and Voltage dependence

We use the tool LAVA to model the dependence among temperature, leakage and voltage. LAVA accepts initial dynamic and leakage power estimates for all micro-architectural blocks. Because dynamic power depends directly on V_{dd} and leakage power depends on temperature and V_{dd} , LAVA refines the initial power estimates provided to it until the values converge. LAVA uses state-of-the-art numerical algorithms (iterative Algebraic Multi-Grid - AMG) to calculate the full-chip v_{dd} and temperature profiles. AMG solves power grids with multi-million nodes very soon because of its hierarchical nature. LAVA also estimates the temperature and voltage gradients across the chip. LAVA could also be used for hot-spot and thermal run-away analysis.

LAVA requires floor-planning data including the location and area of the micro-architectural blocks, the location of signal I/O, V_{dd} and Ground pads on the chip and metal layers of the power grid. The thermal resistance of the package and the heat-sink are obtained from their respective documentation. Also, since most of the heat flow is in the upward direction (towards the heat-sink), we did not model the heat flow in the downward direction (towards the motherboard). This is similar to the model used by Chun Ku et al. [14]. The thermal resistance of the C4 pads is set to a large value to model this effect.

3.2 Power Scaling Studies

This section explains our methodology for power scaling studies. As mentioned before, we perform a constant-transistor scaling study where the number of transistors is held constant as the design is scaled to future process technologies. This study is useful when the same processor design is scaled to a future technology.

Capacitance Scaling: We estimate how the gate capacitance scales across technologies and use the results to scale dynamic power. The other forms of parasitic capacitance of logic scale differently. However, we assume that all forms of logic capacitance scale at the same rate. We perform HSPICE simulations using the Predictive Technology Model (PTM) [15] to estimate how the gate capacitance scales. A current source is attached to a minimum-sized inverter and the voltage at the input node of the inverter is plotted against time. The slope of the curve dv/dt is estimated and the gate capacitance is found using the equation:

$$C = I/(dv/dt) \quad (2)$$

This experiment is repeated for various technologies using the PTM model files and the ratio of gate capacitance across technologies is found.

Interconnect Scaling: We pessimistically assume that the interconnect capacitance does not scale at all in future technologies, but the interconnect length scales at the same rate as transistors.

Leakage Current Scaling: Leakage current, especially gate-leakage current is predicted to be a serious problem in future technologies [16]. We use the sub-threshold current values predicted by Zhao et al. [15] for future technologies. We use the gate-oxide thickness values from [15] and the work that relates gate-oxide thickness to gate-leakage density [17] to estimate gate-leakage densities. Table 5 tabulates the actual values used. We use the above estimates for gate capacitance, interconnect capacitance and leakage current to scale the power estimates to future technologies.

4 Case Study

This section presents a case study by applying our methodology to the TRIPS processor, a general purpose, chip multiprocessor designed at 130nm. The power estimates at 130nm are presented along with the scaled power estimates at 90nm, 65nm, 45nm and 32nm.

4.1 TRIPS Processor

Based on prior work on Grid Processor Architectures [18, 19] and Non-Uniform Cache Architectures (NUCA) [20], the TRIPS prototype processor was designed and implemented at 130nm IBM ASIC process with about 170 million transistors. It is an SOC-style, chip multiprocessor with two processor cores and 1 MB of reconfigurable L2 NUCA cache. The TRIPS processor has a highly distributed micro-architecture consisting of many replicated blocks called “tiles”. Refer to [18, 19, 2] for an elaborate discussion of the TRIPS micro-architecture. This study estimates the power of a baseline TRIPS processor without any power optimizations like clock gating. Estimating the effects of such power optimizations is intended as future work. However, the above methodology and power model could be used for that study too.

4.2 Integrated Power Estimates

The various parameters needed for power estimation were obtained as described in section 3.1 and are listed in Table 1. Table 2 shows the average activity factors for various micro-architectural blocks in the design. Table 3 compares the power estimates for various blocks in the TRIPS processor provided by Step I and Step II of our methodology. The blocks that require most attention during power optimizations can be identified by such

Parameter	Value
f	533MHz
v_{dd}	1.6 volts
α	See table 2
Heat-sink thermal resistance	0.32 <i>Celsius/Watt</i>
C4 thermal resistance	100000 (Infinity)
Heat-sink temperature	50°C
C4 temperature	68°C

Table 1: Table showing the various parameters used for estimation

Micro-architectural Block	Activity Factor
Control	0.080
Execution	0.073
Register File	0.082
L2 Cache	0.079
On-Chip Network Router	0.099
L1 I-cache	0.068
L1 D-cache	0.057
Others	0.067

Table 2: Activity Factors for various micro-architectural blocks of the TRIPS processor

categorization. Figure 3 shows the estimates provided by Step I (Column 2 of Table 3) divided into various categories like clock tree, by logic etc.

Table 4 shows the Step I estimates and Step II estimates for the total dynamic and leakage power consumption at 130nm. From these results, we observe that the Step II estimates are different from the Step I estimates (about 5%). LAVA refines the leakage power estimates based on the temperature of the chip. The dynamic power estimates are also refined by LAVA based on the V_{dd} drops in the power grid. The combined effect of these refinements is that estimates of Step II are less than that of Step I. We also note that the temperature variation is only within 0.3% across the chip and that the voltage variation is only within 3% across the entire chip.

4.3 Results of Scalability Study

The power estimates of the TRIPS chip at 130nm were scaled to various technologies like 90nm, 65nm, 45nm and 32nm using the methodology described in section 3.2.

We use the same α values for the TRIPS micro-architectural blocks across the technologies (see Table 2). Typically, the clock frequency of the design increases when scaled to future technologies, but a constant clock frequency is assumed in this study. However, the same methodology can be used even when α and “f” values change. Table 5 presents the ratio of gate capacitances for future technologies. For e.g., if gate capacitance is 1 unit at 130nm, it is 0.33 units at 65nm. The values of sub-threshold and gate-leakage current were obtained as mentioned in section 3.2. The PFET/NFET widths and lengths of the transistors are assumed to scale as per the scaling factor. The v_{dd} values for various technologies are obtained from the ITRS road-map projections [1].

Table 6 shows the results of the scaling studies (estimates are from Step I). These results show that the same

Micro-architectural Block	Step I estimates	Step II estimates (LAVA)
Control	1.16	1.12
Execution	22.37	21.21
Register File	2.24	2.12
L2 Cache	10.45	10.09
On-Chip Network Router	6.69	6.25
L1 I-cache	1.09	1.04
L1 D-cache	8.92	8.56
Others	5.25	5.11
Total	58.17	55.51

Table 3: Total Power Consumption in Watts at 130nm: Split into various blocks in the TRIPS chip.

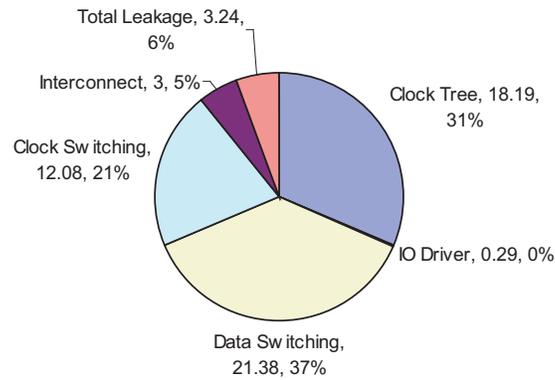


Figure 3: Power estimates at 130nm (Watts) split into various categories

Category	Step I	Step II
Total Dynamic	54.93	52.27
Total Leakage	3.24	3.23
Total Power	58.17	55.50

Table 4: Total Dynamic and Leakage power consumption in Watts at 130nm provided by Step I and Step II of our methodology

Tech nm	V_{dd} volts	C_g ratio No unit	I_{sub} <i>nA/um</i>	J_{gate} <i>A/cm²</i>
130	1.6	1.0	30.0	0.06
90	1.2	0.55	50.0	0.50
65	1.0	0.33	70.0	1.0
45	0.9	0.20	100.0	5.0
32	0.8	0.12	150.0	12.0

Table 5: V_{dd} and threshold voltage values at various technologies

Type	130nm	90nm	65nm	45nm	32nm
Dyn	54.93	26.55	10.17	4.97	2.64
Leak	3.24	6.59	7.82	17.92	26.36
Total	58.17	33.14	17.99	22.90	29.00

Table 6: Scaling study results

design consumes less dynamic power at future technologies. This result is intuitive because of the reduction in V_{dd} and gate capacitance values. The dynamic power consumption reduces by 52% on an average every generation. On the other hand, the leakage power consumption for the same design increases in future technologies by 75% on an average. Starting from 45nm, leakage power starts to dominate the dynamic power consumption (assuming that there are no power optimizations). At 32nm, the leakage power is 91% of the total power consumption.

5 Limitations and future work

The activity factors are estimated using structural RTL simulations. In the ideal case, the activity factors must be estimated using a higher abstraction level like a C-based model. This level of abstraction would be the most useful to estimate activity factors for SPEC like workloads.

This paper uses a simplified scaling model for interconnects and parasitic capacitances other than gate capacitance. Better interconnect scaling models are needed in the future. Moreover, this paper studies constant-transistor scaling as opposed constant-area scaling. Constant-area scaling requires a design-space exploration to decide how to use the extra chip area. This is an interesting future work. Modeling of dynamic voltage/frequency scaling is also intended as future work.

It is also useful to compare the power estimates from different abstraction levels and to analyze the sources of inaccuracies at each level. This paper is a step in that direction because it provides reasonably accurate power estimates at the gate-level. We intend to estimate power at abstraction levels like structural RTL and C models and compare them with the current results.

6 Conclusion

In this paper, we presented a simple power estimation methodology that leveraged design data from a gate-level netlist and activity factors from a structural RTL model. We also refined the initial temperature-independent power estimates to generate temperature-aware power estimates. Next, we presented a methodology for studying how power estimates of a design scale across process technologies. Finally, we presented a case study by applying our methodologies to an SOC-based, tiled, chip multiprocessor and presented the results.

References

- [1] S. I. Association, “The international technology roadmap for semiconductors(ITRS),” 2005.
- [2] D. Burger, S. Keckler, K. McKinley, M. Dahlin, L. John, C. Lin, C. Moore, J. Burrill, R. McDonald, and W. Yoder, “Scaling to the end of silicon with EDGE architectures,” *IEEE Computer*, vol. 37, no. 7, pp. 44–55, July 2004.
- [3] G. Bernacchia and M. C. Papaefthymiou, “Analytical macromodeling for high-level power estimation,” in *ICCAD '99: Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design*. Piscataway, NJ, USA: IEEE Press, 1999, pp. 280–283.
- [4] S. Gupta and F. N. Najm, “Analytical model for high level power modeling of combinational and sequential circuits,” in *VOLTA '99: Proceedings of the IEEE Alessandro Volta Memorial Workshop on Low-Power Design*. Washington, DC, USA: IEEE Computer Society, 1999, p. 164.
- [5] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: a framework for architectural-level power analysis and optimizations,” in *ISCA '00: Proceedings of the 27th annual international symposium on Computer architecture*. New York, NY, USA: ACM Press, 2000, pp. 83–94.
- [6] W. Ye, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, “The design and use of simplepower: a cycle-accurate energy estimation tool,” in *DAC '00: Proceedings of the 37th conference on Design automation*. New York, NY, USA: ACM Press, 2000, pp. 340–345.
- [7] F. N. Najm, “A Survey of Power Estimation Techniques in VLSI Circuits,” *IEEE Transactions on Very Large Scale Integrated Systems*, vol. 2, no. 4, pp. 446–455, 1994.
- [8] Synopsys, Inc., “Synopsys products.” [Online]. Available: https://solvnet.synopsys.com/dow_search
- [9] Sequence Design, Inc., “Power theatre: Low power design and power analysis for nanometer system-on-chip design.”
- [10] EECS Department of the University of California at Berkeley., “Spice.” [Online]. Available: <http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE/>
- [11] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, “Temperature-aware microarchitecture: Modeling and implementation,” *ACM Trans. Archit. Code Optim.*, vol. 1, no. 1, pp. 94–125, 2004.
- [12] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, “Full chip leakage estimation considering power supply and temperature variations,” in *ISLPED '03: Proceedings of the 2003 international symposium on Low power electronics and design*. New York, NY, USA: ACM Press, 2003, pp. 78–83.
- [13] D. Stroobandt, H. V. Marck, and J. V. Campenhout, “An accurate interconnection length estimation for computer logic,” 1996. [Online]. Available: citeseer.ist.psu.edu/stroobandt96accurate.html
- [14] J. C. Ku, S. Ozdemir, G. Memik, and Y. Ismail, “Thermal management of on-chip caches through power density minimization,” in *MICRO 38: Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 283–293.
- [15] W. Zhao and Y. Cao, “New generation of predictive technology model for sub-45nm design exploration,” in *ISQED '06: Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED'06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 585–590.

- [16] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, no. 4, pp. 23–29, 1999.
- [17] M. G. Chris Bowen, Gerhard Klimeck and D. Chapman, "Dopant fluctuations and quantum effects in sub-0.1um cmos," 1997. [Online]. Available: http://www.cfdrc.com/nemo/pubs/isdrs_html/isdrs_html.html
- [18] R. Nagarajan, K. Sankaralingam, D. Burger, and S. W. Keckler, "A design space evaluation of grid processor architectures," in *Proceedings of the 34th Annual ACM/IEEE International Symposium on Microarchitecture*, December 2001, pp. 40–51.
- [19] K. Sankaralingam, R. Nagarajan, H. Liu, C. Kim, J. Huh, D. Burger, S. W. Keckler, and C. R. Moore, "Exploiting ILP, TLP, and DLP with the polymorphous TRIPS architecture," in *Proceedings of the 30th Annual International Symposium on Computer Architecture*, June 2003, pp. 422–433.
- [20] C. Kim, D. Burger, and S. W. Keckler, "An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches," in *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*. New York, NY, USA: ACM Press, 2002, pp. 211–222.