

Exploiting Narrow-Width Values for Thermal-Aware Register File Designs

Shuai Wang, Jie Hu, and Sotirios G. Ziavras
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ 07102, USA
Email: sw63.jhu,ziavras@njit.edu

Sung Woo Chung
Division of Computer and Communication Engineering
Korea University
Seoul 136-713, Korea
Email: swchung@korea.ac.kr

Abstract—Localized heating-up creates thermal hotspots across the chip, with the integer register file ranked as the hottest unit in high-performance microprocessors. In this paper, we perform a detailed study on the thermal behavior of a low-power value-aware register file (VARF) that is subjected to internal fine-grain hotspots. To further optimize its thermal behavior, we propose and evaluate three thermal-aware control schemes, thermal sensor (TS), access counter (AC), and register-id (ID) based, to balance the access activity and thus the temperature across different partitions in the VARF. The simulation results using SPEC CINT2000 benchmarks show that the register-id controlled VARF (ID-VARF) scheme achieves optimized thermal behavior at minimum cost as compared to the other schemes. We further evaluate the performance impact of the thermal-aware VARF design with the dynamic thermal management (DTM). The experimental results show that the ID-VARF can improve the performance by 26.1% and 7.2% over the conventional register file and the original VARF design, respectively.

I. INTRODUCTION

In recent years, mainstream microprocessors have been experiencing dramatic performance improvement driven by the continuous technology scaling and innovative architectures. In the meantime, the slower scaling of the supply voltage compared to the rapidly increasing clock frequency and on-chip transistor integration density has led to an alarming situation of extremely high on-chip power density [1]. The high on-chip power density is the direct cause of the steep increase of the chip temperature in these mainstream high-performance microprocessors. Notice that the very high chip temperature not only demands costly cooling systems and packaging techniques [2], but also results in many other adversities such as exponential increase in leakage current [3], performance degradation due to reduced carrier mobility [4], increased vulnerability to soft errors [5], and reduced semiconductor device lifetime reliability [6]. Thus, on-chip temperature management and optimization are of paramount importance to the design of next generation high-performance microprocessors.

Among major on-chip components, the integer register file has traditionally been recognized as the hottest unit when running most general purpose applications [7]. This is mainly due to the high number of accesses per cycle to the register file and its multiported design in conventional superscalar microprocessors. In general, the integer register file exhibits

extremely high power density. Notice that the localized heating up is much faster than heat diffusing across the entire chip, which results in thermal hotspots within on-chip components with high power densities. Thus, the thermal situation in the integer register file may represent the worst scenario for designing the cooling systems and the packaging techniques. Consequently, the integer register file is among the first-class candidates for microprocessor thermal management and optimizations. Given the fact that heat dissipation in semiconductor circuits is induced by power consumption, a straightforward response to on-chip thermal issues is to control its power consumption. In this paper, we focus on the impact of both static and dynamic microarchitectural designs, exemplified by the low-power value-aware register file (VARF) [8][9][10], on managing and optimizing the on-chip temperature in high-performance microprocessors.

It is a general belief that the temperature does not have a strong correlation with the *power consumption*. Instead, the *power density* is the direct cause of the localized heating up [7][11]. From the perspective of power and power density optimization, we evaluate the original value-aware register file (VARF) design that exploits the majority narrow-width register values, i.e., data values that can be represented by fewer bits than the machine data width, and its thermal behavior. With both significantly reduced power consumption and power density in the VARF, the overall register file temperature is reduced. In order to further optimize the thermal behavior within the VARF, we propose and compare three thermal-aware control schemes to manage the storage and access of the narrow-width data in the register file: thermal sensor (TS) based, access counter (AC) based, and register-id (ID) based control schemes. Our simulation results indicate that the register-id controlled VARF (ID-VARF) achieves the same level of thermal optimization (i.e., reduced temperature and balanced thermal distribution across different partitions) in the register file while minimizing the overhead. When applying the dynamic thermal management (DTM), the ID-VARF achieves a 26.1% and 7.2% performance improvement over the conventional register file and the original VARF design, respectively.

The rest of the paper is organized as follows. In the next section, we discuss related work in low-power register file designs and thermal management techniques. In Section III,

we provide detailed designs of thermal-aware VARFs. We present our experimental setup and results in Section IV. Section V concludes this work.

II. RELATED WORK

Large multiported register files for instruction-level parallelism (ILP) exploitation in modern wide-issue dynamically scheduled superscalar microprocessors have led to increased access latency and per access power consumption. Most previous research on the register file has focused on developing scalable register file architectures in terms of performance, power, and area, such as hierarchical register files [12][13][14][15], clustered register files [16][17][18][19], virtual register files [20][21], physical register reuse/unification [22][23][24][25], banked register files [26][27], port-reduced register files with bypass-hint schemes [28][27], and exploiting narrow-width register values [29][30][31][8][9], for next generation high-performance and low power microprocessors.

Despite the wealth of efforts in designing scalable register files, few works have been conducted to investigate or optimize the thermal behavior of the register file, which is ranked as the hottest unit in high-performance microprocessors [7]. However, thermal-aware microarchitectures and dynamic thermal management techniques have been proposed and studied for years [32][33][34][35][7]. For fine-grain thermal optimization, John et. al. provided a detailed study of the thermal behavior in the level-one data cache and further proposed thermal-aware subarraying schemes in [36]. In [11], Ku et. al. proposed two thermal-aware cache architectures, namely the power density-minimized architecture (PMA) and block permutation schemes (BPS), for cacheline level thermal management in set-associative caches.

Powell et. al. [37] proposed to mitigate the power density by balancing the utilization of register file copies. Their scheme assumes two or multiple register file copies in the processor. In [38], a thermal herding scheme was proposed to locate the hottest partition of the register file on the top die closest to the heat sink to reduce the temperature. Their scheme is targeting at the thermal-aware design in the 3D stacked processors. A compiler-based register reallocation scheme was proposed in [39] to balance the register file power density and reduce the peak temperature. As a software approach, it requires compiler support. Further, similar to [37], this scheme is not a low power design rather to balance the temperature across the two banks or the entire register file. Consequently, after the thermal balancing, the register file is still among the hottest units and needs further thermal reduction. Our thermal-aware VARF not only reduces the power consumption thus the average temperature in the register file, but also mitigates the power density to balance the temperature in the register file.

III. VALUE-AWARE REGISTER FILE (VARF) FOR THERMAL OPTIMIZATION

A. Value-Aware Register File (VARF)

In high-performance 64-bit microprocessors, many generated register values during the execution do not require the full

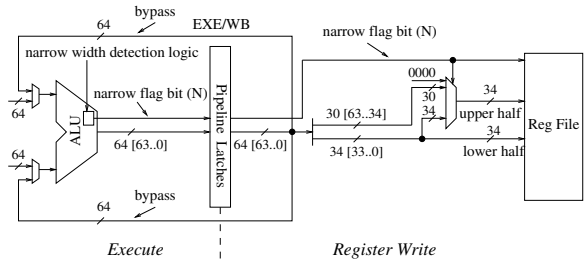


Fig. 1. Augmented datapath with width detection and value duplication.

width of 64 bits. Values that can be represented by significantly less than 64 bits are generally referred to as narrow-width values. The presence of narrow-width values has been well studied and exploited for performance and power optimizations [40][41][42][43][44][45]. Our experimental results show that on the average 97% of the produced integer register values can be represented by no more than 34 bits.

The design of the value-aware register file (VARF) exploits the fact that the access to the large number of leading zeros (ones) in narrow-width values can be avoided/disabled to significantly reduce the per register file access power consumption, while the original 64-bit value can be simply restored by using the existing sign extension logic at the inputs of ALUs. Due to the dominant narrow-width values written to or read from the register file, the overall register file power consumption can be dramatically reduced in VARF, which should also effectively bring down the high temperature in the register file. Controlling the bitline accesses in the register file exactly according to the bit width of the value, i.e., only activating bitlines containing no leading zeros (or ones), is neither necessary nor efficient due to the complexity of the required access control. Instead, we classify register values into two categories: 34-bit narrow-width values and 64-bit regular values. Figure 1 shows the block diagram of the datapath augmented with a duplication logic controlled by the narrow flag bit (N). The output data to be written into the register file is augmented to 68 bits and equally divided into two halves, the upper 34-bit half and the lower 34-bit half. The lower half is directly from bits[33..0]. If the data is detected as a 34-bit narrow-width value, the upper half is duplicated from bits[33..0]. If it is a regular value, we pad bits[63..34] with four padding bits (0000) to form the upper half. The narrow flag bit (N) is used to control the multiplexer for selecting the duplicated or the padded value. To capture narrow-width values, we extract the internal signals from the existing leading-0/1 detection logic within the functional units [46] (in order to minimize its timing overhead in deeply pipelined designs at new technology generations [47]). Furthermore, as shown in Figure 1, we put the datapath augmentation in the Register Write stage and it can be overlapped with the register decoding phase.

In the VARF, the register file is also partitioned into two halves. Each half contains 34-bit data and 1 narrow-width flag bit. Figure 2 shows the schematic diagram of the partitioned

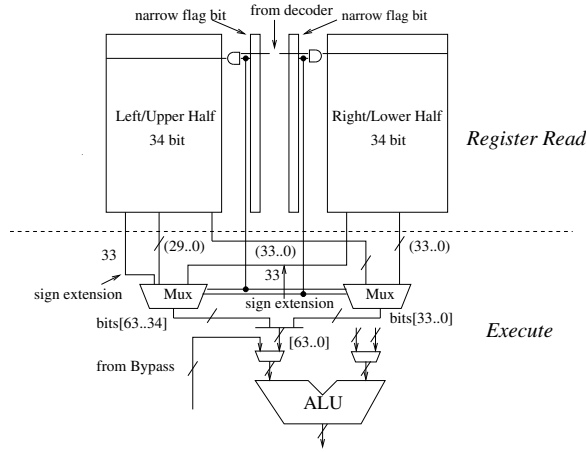


Fig. 2. The schematic diagram of the partitioned VARF.

VARF. During a register file write, the narrow-width data will be stored in one half of the VARF selected by the control logic of the chosen scheme and the other half will be gated by this flag bit. During a register file read, after precharging the bitlines, the wordline of the half with leading zeros (or ones) is immediately gated by the flag bit whose output is directly fed to the AND gate as shown in Figure 2. The power consumption is thus reduced by accessing only one half of the entire register file for these narrow-width values. Therefore, the temperature of the register file can be substantially reduced. However, the additional two multiplexers of each read port shown in Figure 2 may increase the register access time. Thus, we place them in the Execute stage to maintain the cycle time of the processor, since the critical path of the processor is usually not in the Execute stage, but in the Issue or Register Read stage. Furthermore, they can be implemented by expanding some existing logic in the ALU such as sign extension and operand shift [31]. Therefore, the hardware cost of the additional multiplexers can be minimized.

B. Optimizing the Thermal Behavior in VARFs

The original solution of the VARF is to store all the narrow-width data in one half of the register file, e.g. the right half. The left half is gated for power savings. We refer to it as the basic scheme of the VARF. The advantages of this scheme are its easy implementation in hardware and simple control. However, one of the major disadvantages of this scheme is that the power density of the right half will become much higher than that for the left half, which can make the right half a new hotspot among the on-chip components.

Targeting at power reduction, the original VARF design is not capable of eliminating the hotspots in the register file. This is mainly due to the extremely unbalanced access activities in the left and right halves of the VARF. A straightforward solution for thermal optimization in VARF is to evenly distribute the register file accesses between the two halves. To support this, we first propose a control scheme based on the temperature readings from the deployed thermal sensors (TS) in the two halves. We assume that the thermal sensor is located

in the middle of each half of the VARF and provides cycle-level temperature readings. At the pipeline writeback stage, a detected narrow-width value will trigger the control logic such that the temperature of the left and right halves of the VARF are compared and the narrow-width value is written into the half with the lower temperature. A regular value will write both halves. If the data is written into one half of the VARF, the corresponding flag bit will be set to one. Otherwise, the flag bit will be reset to zero. Notice that here we are only controlling the register file write operations. The register file read is consequently controlled by the two flag bits.

Note that thermal sensors are themselves power-hungry circuitry and their accuracy is limited by their size [7]. Therefore, to avoid high cost thermal sensors, we propose a second control scheme based on the use of the access counters (ACs). Each half in the VARF maintains its own access counter. Upon the register file write of a narrow-width value, a comparison of the two access counters determines which half the value should be written into. The flag bits will be updated accordingly. In our simulation, we assumed an infinite counter. In a real implementation, the counters need to be reset periodically to avoid saturation. If a small counter is adopted, it may introduce inaccuracy into the control. However, the overhead of a large counter is noticeable.

Both the thermal sensor and access counter based schemes are dynamic schemes that require feedbacks of the current processor state. To further reduce the cost of these dynamic designs, we propose a third control scheme that writes the narrow-width data into one of the two halves based on the physical id (ID) of the renaming register. If the id is an even number, we store the narrow-width data into the right half. If the id is odd, the narrow-width data is stored into the left half. When the narrow-width data is stored in the right half, as shown in Figure 2, bits[33..0] of the readout data are from the right half and bits[63..34] are restored by sign-extending the right half data. The left half of the register file is gated by the flag bit for power savings. When the narrow-width data is stored in the left half, it will be steered to bits[33..0] of the readout data and from bits[63..34] by its sign-extension. When the data is a regular value, due to the augmented datapath shown in Figure 1, the right half of the register file has bits[33..0] of the regular data and the lower 30 bits (excluding the upper 4 padding bits of the 34-bit readout) of the left half form the rest of the 30 bits in the regular data. When the two flag bits are 11, the data represents a regular value. When the two flag bits are 01 or 10, the data is a narrow-width value and is stored in the half with the 1 in its flag bit. Notice that this is a static scheme which is easy to implement compared to the previous two dynamic schemes.

IV. EXPERIMENTAL EVALUATION

A. Experimental Setup

We derive our simulators from SimpleScalar V3.0 [48] with Wattch [49], HotSpot [7] and HotLeakage [50] incorporated, to model a contemporary high-performance eight-issue micro-processor similar to Alpha 21464 [51]. Figure 3 shows the

TABLE I
PARAMETERS FOR THE SIMULATED PROCESSOR.

Processor Core	
Int/FP issue queue	128 entries
Load/Store Queue	256 entries
Active list (ACL)	512 entries
Int/FP Register File	512/512 registers
Datapath width	8 instructions per cycle
Function Units	8 IALU, 2 IMULT/IDIV, 4 FALU 2 FMULT/FDIV/FSQRT, 4 MemPorts
Branch Predictor	Alpha 21264 tournament predictor with 4K meta-table 2048-entry, 2-way BTB, and 32-entry RAS
Memory Hierarchy	
L1 I/DCache	64KB, 2 ways, 64B blocks, 2 cycle latency
L2 UCache	4MB, 8 ways, 128B blocks, 12 cycle latency
Memory	225 cycles first chunk, 12 cycles rest
TLB	fully-assoc., 128 entries
HotSpot Parameters	
Technology	70nm
Supply Voltage	0.9V
Clock Frequency	5.6Ghz
Ambient Air Temperature	45°C
Package Thermal Resistance	0.8K/W
Die size	11.5mm x 11.5mm

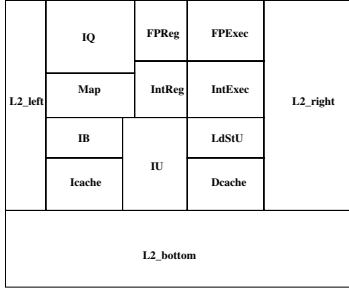


Fig. 3. The floorplan of the simulated processor

floorplan of our simulated processor which is scaled from [51]. Table I shows the detailed configuration for the simulated microprocessor.

For experimental evaluation, we use the SPEC CINT2000 suite compiled for the Alpha instruction set architecture with “peak” tuning. We use the reference input sets for this study. Each benchmark is first fast-forwarded to its early single simulation point (gap uses the standard single simulation point) specified by SimPoint [52]. Then, we simulate the next 100 million instructions in detail. For temperature profiling, each benchmark was run several times to reach a converged steady temperature.

B. Experimental Results and Analysis

Table II shows the steady temperature, an average across the SPEC CINT2000 benchmarks, of major components in the simulated processor with a conventional monolithic register file, the Base model. From Table II, the integer register file is the hottest on-chip unit with a temperature about 3 degrees higher than the second hottest unit and about 24 degrees higher than the coolest one. These results further confirm that the thermal control of the integer register file is of critical importance.

For comparison purposes, we model a coarse level VARF that treats the partitioned value-aware register file as one

TABLE II
AVERAGE STEADY TEMPERATURE OF MAJOR PROCESSOR COMPONENTS IN THE SIMULATED PROCESSOR WITH A Base INTEGER REGISTER FILE.

Block	Steady T (K)	Block	Steady T (K)
L2left	342.8	LdStU	357.5
L2bottom	336.8	Map	346.2
L2right	338.1	IQ	350.9
Icache	348.6	FPReg	349.9
Dcache	355.2	IntReg	360.6
IB	344.0	FPExec	346.6
IU	346.5	IntExec	349.8

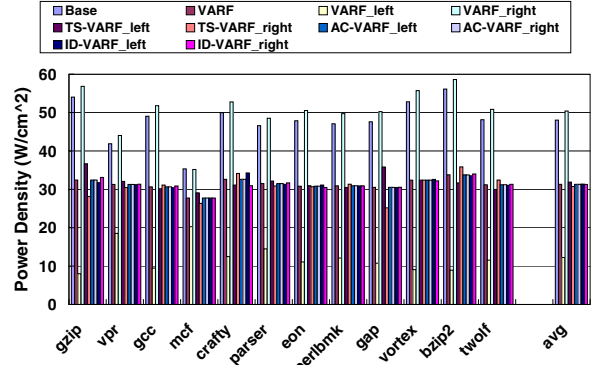


Fig. 4. Power density comparison among Base, VARF, TS-VARF, AC-VARF, and ID-VARF register files.

single block. To study the thermal behavior within VARF, we also model the VARF temperature at the partition level, with VARF_left referring to the left partition and VARF_right to the right partition. With the three proposed control schemes for thermal optimization within VARF, we further derive our thermal-sensor controlled VARF (TS-VARF), access-counter controlled VARF (AC-VARF), and register-id controlled VARF (ID-VARF) schemes. The temperature of these three VARFs is also modeled at the partition level.

Table III provides the register file access distribution (including reads and writes) between the left and right halves of different VARFs. The results show that i) a dominating 97% of the register file accesses are served by the right half in the original VARF, ii) with the proposed control schemes, register file accesses (for both reads and writes) are almost evenly distributed between the two halves, which indicates that the write control scheme is very effective in controlling the register file access activities.

Figure 4 compares the average power density in different register files. At the coarse level, the register-file-wide power density is reduced by 34.2% in the VARF register file. However, at the partition level, the power density of the right half

	Write_right	Read_right	Write_left	Read_left
VARF	0.35	0.62	0.01	0.02
TS-VARF	0.17	0.31	0.19	0.33
AC-VARF	0.18	0.32	0.18	0.32
ID-VARF	0.18	0.32	0.18	0.32

TABLE III
THE ACCESS DISTRIBUTION OF THE VARF, TS-VARF, AC-VARF, AND ID-VARF REGISTER FILES.

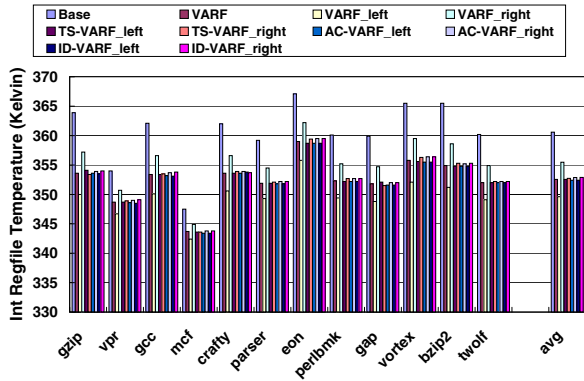


Fig. 5. Steady temperature comparison among Base, VARF, TS-VARF, AC-VARF, and ID-VARF register files.

(VARF_right) is very high, even higher than that of the Base register file, while the left half (VARF_left) has a quite low power density. This is because all the narrow-width data is residing in the right half and there is more power consumed by the additional data and tag bits. The power density is almost the same in the left and right halves in AC-VARF and ID-VARF, which means that the two control schemes divide the power quite evenly between the two partitions. In the TS-VARF scheme, the power density in the left half is slightly higher than that for the right half. Notice that the power overhead of thermal sensors in TS-VARF or access counters in AC-VARF is not included in this comparison. Although CMOS thermal sensors suffer from substantial errors, we optimistically assumed ideal sensors for TS-VARF without considering sensor errors. In the VARF thermal model, narrow-flag bits and padding bits are modeled as additional bitlines in the register file.

The steady temperature comparison of different register files is shown in Figure 5. Basically, the thermal behavior in the register files should be similar to the power density results. The VARF achieves about an 8-degree temperature reduction in the register file. The temperature difference between the two halves of the VARF is about 5.8 degrees, with the higher temperature in the right half, which could be a new hotspot on the chip. Note that the heat diffusion among neighboring blocks also plays an important role in the thermal behavior, which explains the temperature - power density disparity among the Base, VARF_left, and VARF_right. The TS-VARF achieves the best thermal behavior among the various schemes. The temperature difference between the left and right halves is only 0.2 degree. In the meantime the results show that the AC-VARF and ID-VARF also demonstrate good thermal behavior with 0.46 and 0.48 degree temperature difference between the two halves. Notice that all three thermal control schemes have successfully reduced the register file temperature below 353K (80°C).

Modeled by Cacti 4.2 [53], our ID-VARF shows a 3% increase in the register file access latency. To further study its performance impact, we evaluate ID-VARF in the presence of dynamic thermal management (DTM) using the DVFS

(dynamic voltage and frequency scaling) scheme. In the low power mode, voltage is scaled down to 0.7V (78% of the nominal) and frequency to 4.0GHz (71% of the nominal). A 10 μ s voltage/frequency scaling overhead and a thermal emergency threshold at 356 \pm 1K (83 \pm 1°C) are assumed. Figure 6 shows that with DTM, our thermal-aware ID-VARF achieves a 26.1% and 7.2% performance improvement over the conventional register file design and the original VARF design, respectively, which is sufficient to offset its access latency overhead. ID-VARF's Cacti model also shows an area overhead of 7%, compared to a conventional register file design. If this 7% area overhead is taken into account, our thermal-aware VARF further reduces its temperature by 0.57 degrees. Note that these results do not include the overhead of the multiplexers at each port since they are moved to the Execute stage.

V. CONCLUSIONS

In this paper, we propose to exploit the majority narrow-width data to optimize the thermal behavior of the integer register file, which has been widely identified as the hottest unit in high-performance microprocessors. Due to the extremely unbalanced access activities between its partitions, an original value-aware register file (VARF) design may introduce new hotspots in the most frequently accessed partitions. To further optimize the thermal behavior within the VARF, we propose and evaluate three thermal-aware control schemes to distribute and balance the access activities across the partitions. Our experimental results indicate that the simple register-id controlled VARF (ID-VARF) scheme achieves the same level of thermal optimization as other schemes (i.e., lower and balanced temperature across different partitions), at minimum cost. To further evaluate the performance impact of the proposed thermal-aware VARF design, we adopt the dynamic thermal management (DTM) to regulate the chip temperature. The ID-VARF achieves a 26.1% and 7.2% performance improvement over the conventional register file and the original VARF design.

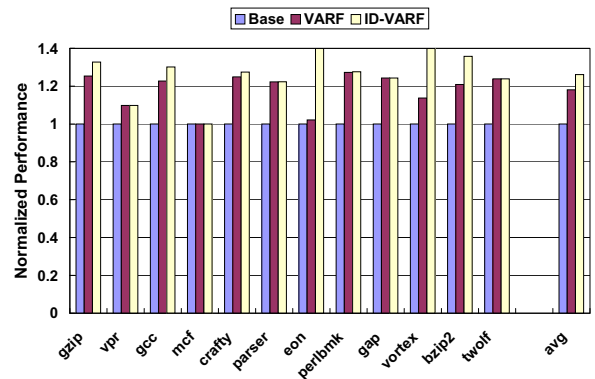


Fig. 6. Performance Improvement of thermal-aware ID-VARF design with the DTM compared to the Base and original VARF register file designs.

REFERENCES

- [1] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, no. 4, pp. 23–29, 1999.
- [2] V. Tiwari *et al.*, "Reducing power in high-performance microprocessors," in *35th Design Automation Conference*, 1998.
- [3] N. S. Kim *et al.*, "Leakage current: Moore's law meets static power," *Computer*, no. 12, pp. 68–75, Dec. 2003.
- [4] T. T. Mnatsakanov, M. E. Levinshtein, L. I. Pomortseva, and S. N. Yurkov, "Carrier mobility model for simulation of sic-based electronic devices," *Semiconductor Science and Technology*, vol. 17, no. 9, pp. 974–977, 2002.
- [5] R. Aitken and B. Hold, "Modeling soft-error susceptibility for ip blocks," in *11th IEEE International On-Line Testing Symposium*, 2005, pp. 70–73.
- [6] R. Viswanath, V. Wakharkar, A. Watwe, and V. Lebonheur, "Thermal performance challenges from silicon to systems," *Intel Technology Journal*, vol. Q3, 2000.
- [7] K. Skadron *et al.*, "Temperature-aware microarchitecture," in *Proceedings of the 30th annual international symposium on Computer architecture*, San Diego, California, 2003, pp. 2–13.
- [8] O. Ergin, "Exploiting narrow values for energy efficiency in the register files of superscalar microprocessors," in *Proc. of 16th International Workshop on Power and Timing Modeling, Optimization and Simulation*, 2006, pp. 477–485.
- [9] S. Wang, H. Yang, J. Hu, and S. G. Ziavras, "Asymmetrically banked value-aware register files," in *Proc. of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2007)*, 2007, pp. 363–368.
- [10] S. Wang, H. Yang, J. Hu, and S. G. Ziavras, "Asymmetrically banked value-aware register files for low energy and high performance," *Microprocessors and Microsystems*, vol. 32, no. 3, pp. 171–182, May 2008.
- [11] J. C. Ku *et al.*, "Thermal management of on-chip caches through power density minimization," in *Proc. Micro-38*, Nov. 2005, pp. 283–293.
- [12] J. A. Swensen and Y. N. Patt, "Hierarchical registers for scientific computers," in *Proc. of ICS'88*, 1988, pp. 346–354.
- [13] R. Balasubramonian, S. Dwarkadas, and D. H. Albonesi, "Reducing the complexity of the register file in dynamic superscalar processors," in *Proceedings of Micro-34*, December 2001, pp. 237–248.
- [14] E. Borch, E. Tune, S. Manne, and J. Emer, "Loose loops sink chips," in *Proc. of HPCA-8*, February 2002, pp. 270–281.
- [15] J. A. Butts and G. S. Sohi, "Use-based register caching with decoupled indexing," in *Proceedings of the 31st annual international symposium on Computer architecture*, 2004, pp. 302–313.
- [16] R. E. Kessler, "The alpha 21264 microprocessor," *IEEE Micro*, vol. 19, no. 2, pp. 24–36, March-April 1999.
- [17] A. Capitanio, N. Dutt, and A. Nicolau, "Partitioned register files for vlws: a preliminary analysis of tradeoffs," in *Proc. of Micro-25*, 1992, pp. 292–300.
- [18] K. I. Farkas, P. Chow, N. P. Jouppi, and Z. Vranesic, "The multiclustor architecture: reducing cycle time through partitioning," in *Proc. Micro-30*, 1997, pp. 149–159.
- [19] R. Canal, J.-M. Parcerisa, and A. Gonzalez, "Dynamic cluster assignment mechanisms," in *Proc. of HPCA-06*, 2000, pp. 132–142.
- [20] A. Gonzalez, J. Gonzalez, and M. Valero, "Virtual-physical registers," in *Proc. of HPCA-4*, 1998, pp. 175–184.
- [21] T. Monreal, A. Gonzalez, M. Valero, J. Gonzalez, and V. Vinals, "Delaying physical register allocation through virtual-physical registers," in *Proc. of Micro-32*, 1999, pp. 186–192.
- [22] S. Jourdan, R. Ronen, M. Bekerman, B. Shomar, and A. Yoaz, "A novel renaming scheme to exploit value temporal locality through physical register reuse and unification," in *Proc. Micro-31*, 1998, pp. 216–225.
- [23] J. F. Martinez, J. Renau, M. C. Huang, M. Prvulovic, and J. Torrellas, "Cherry: checkpointed early resource recycling in out-of-order microprocessors," in *Proc. fo Micro-35*, 2002, pp. 3–14.
- [24] S. Balakrishnan and G. Sohi, "Exploiting value locality in physical register files," in *Proc. of Micro-36*, 2003, pp. 265–276.
- [25] R. Gonzalez, A. Cristal, D. Ortega, A. Veidenbaum, and M. Valero, "A content aware integer register file organization," in *Proceedings of the 31st Annual International Symposium on Computer Architecture*, June 2004.
- [26] S. Wallace and N. Bagherzadeh, "A scalable register file architecture for dynamically scheduled processors," in *Proceedings of the 1996 Conference on Parallel Architectures and Compilation Techniques*, 1996, p. 179.
- [27] J. Tseng and K. Asanovic, "Banked multiported register files for high-frequency superscalar microprocessors," in *30th International Symposium on Computer Architecture (ISCA-30)*, San Diego, CA, June 2003.
- [28] I. Park, M. Powell, and T. Vijaykumar, "Reducing register ports for higher speed and lower energy," in *Proc. of Micro-35*, Dec. 2002.
- [29] M. H. Lipasti, B. R. Mestan, and E. Gunadi, "Physical register inlining," in *Proceedings of the 31st Annual International Symposium on Computer Architecture*, June 2004, pp. 325–335.
- [30] O. Ergin *et al.*, "Register packing: Exploiting narrow-width operands for reducing register file pressure," in *Proc. of MICRO-37*, Portland, OR, 2004, pp. 304–315.
- [31] M. Kondo and H. Nakamura, "A small, fast and low-power register file by bit-partitioning," in *Proc. of HPCA-11*, 2005, pp. 40–49.
- [32] H. Sanchez *et al.*, "Thermal management system for high performance PowerPC microprocessors," in *Proceedings of COMPCON 97*, San Jose California, 1997.
- [33] S. H. Gunther, F. Binns, D. M. Carmean, and J. C. Hall, "Managing the impact of increasing microprocessor power consumption," *Intel Technology Journal*, vol. Q1, 2001.
- [34] C. H. Lim, W. R. Daasch, and G. Cai, "A thermal-aware superscalar microprocessor," in *Proceedings. International Symposium on Quality Electronic Design*, 2002, pp. 517–522.
- [35] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proc. of HPCA'01*, 2001.
- [36] J. K. John, J. S. Hu, and S. G. Ziavras, "Optimizing the thermal behavior of subarrayed data caches," in *Proc. of ICCD-2005*, 2005, pp. 625–630.
- [37] M. D. Powell, E. Schuchman, and T. N. Vijaykumar, "Balancing resource utilization to mitigate power density in processor pipelines," in *Proc. of Micro-38*, 2005, pp. 294–304.
- [38] K. Puttaswamy and G. H. Loh, "Thermal herding: Microarchitecture techniques for controlling hotspots in high-performance 3d-integrated processors," in *Proc. of HPCA-13*, 2007, pp. 193–204.
- [39] X. Zhou, C. Yu, and P. Petrov, "Compiler-driven register re-assignment for register file power-density and temperature reduction," in *Proceedings of the 45th annual conference on Design automation*, 2008, pp. 750–753.
- [40] D. Brooks and M. Martonosi, "Dynamically exploiting narrow width operands to improve processor power and performance," in *Proc. of HPCA-5*, 1999.
- [41] G. H. Loh, "Exploiting data-width locality to increase superscalar execution bandwidth," in *Proc. of Micro-35*, November 18–22 2002, pp. 395–405.
- [42] A. Aggarwal and M. Franklin, "Energy efficient asymmetrically ported register files," in *Proc. of ICCD'03*, 2003, pp. 2–7.
- [43] L. Villa, M. Zhang, and K. Asanovic, "Dynamic zero compression for cache energy reduction," in *Proc. of Micro-33*, 2000, pp. 214–220.
- [44] R. Canal, A. Gonzalez, and J. E. Smith, "Very low power pipelines using significance compression," in *Proc. of Micro-33*, 2000, pp. 181–190.
- [45] R. Canal, A. Gonzalez, and J. E. Smith, "Value compression for efficient computation," in *Proceedings of the Euro-Par 2005*, 2005, pp. 519–529.
- [46] D. R. Lutz and D. N. Jayasimha, "Early zero detection," in *Proc. of ICCD'96*, 1996, pp. 545–550.
- [47] M. S. Hrishikesh *et al.*, "The optimal logic depth per pipeline stage is 6 to 8 fo4 inverter delays," in *Proceedings of the 29th Annual International Symposium on Computer Architecture*, 2002, pp. 14–24.
- [48] D. Burger and T. M. Austin, "The simplescalar tool set, version 2.0," Computer Sciences Department, University of Wisconsin, Tech. Rep. 1342, 1997.
- [49] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," in *Proc. International Symposium on Computer Architecture*, 2000.
- [50] Y. Zhang *et al.*, "Hotleakage: A temperature-aware model of subthreshold and gate leakage for architects," Dept. of Computer Science, Univ. of Virginia, Tech. Rep., 2003.
- [51] R. P. Preston *et al.*, "Design of an 8-issue superscalar risc microprocessor with simultaneous multithreading," in *Proc. IEEE International Solid-State Circuits Conference*, 2002.
- [52] T. Sherwood *et al.*, "Automatically characterizing large scale program behavior," in *Proc. of ASPLOS X*, October 2002.
- [53] D. Tarjan, S. Thoziyoor, and N. P. Jouppi, "Cacti 4.0," HP Laboratories, Tech. Rep., 2006.