Adopting the Banked Register File Scheme for Better Performance and Less Leakage

Hyung Beom Jang, Eui-Young Chung, and Sung Woo Chung

ABSTRACT—Excessively high temperature deteriorates the reliability and increases the leakage power consumption of microprocessors. The register file, known as one of the hottest functional units in microprocessors, incurs frequent dynamic thermal management operations for thermal control. In this letter, we adopt the banked register file scheme, which was originally proposed to reduce dynamic power consumption. By simply modifying the register file structure, the temperature in the register file was reduced dramatically, resulting in 13.37% performance improvement and 10.49% total processor leakage reduction.

Keywords—Microprocessor, banked register file, DTM, dynamic thermal management, leakage, computer architecture.

I. Introduction

Current design trends require multi-functional convergence with fast data processing. To keep pace with these requirements, transistors are packed even more tightly. Consequently, on-chip power densities are exponentially increased, leading to excessive temperature. Among the components in computer systems, microprocessors are the hottest since they generally operate at the highest frequency. This excessive temperature of the microprocessor, if uncontrolled, causes serious malfunctions related with reliability. To control the excessive temperature, additional architectural thermal management is necessary. Dynamic thermal management (DTM) techniques [1], such as dynamic voltage frequency scaling (DVFS) and fetch toggling, periodically monitor temperature at runtime and dynamically reduce heat dissipation as soon as the temperature exceeds a pre-defined DTM trigger threshold. However, these techniques inevitably deteriorate the performance of the microprocessor.

The register file is known as one of the hottest functional units in the microprocessor. To tackle the high temperature of the register file, DTM operations have been used, resulting in performance degradation. To reduce heat dissipation while minimizing the performance degradation, we adopt the banked integer register file scheme [2], which was originally proposed to reduce dynamic power consumption. The banked integer register file scheme is reported to reduce the dynamic energy by 38.6% [2] and the peak temperature of the integer register file by 19°C [3]. In this letter, we compare the impact of the lower heat dissipation of the banked integer register file on performance to that of the non-banked integer register file. We also examine the benefits of the banked integer register file scheme in relation to leakage power consumption. As far as we know, this is the first study that evaluates the impact of the banked register file on performance and leakage considering heat dissipation.

II. Banked Register File Structure

Generally, in a high-performance superscalar microprocessor, access to the register file is one of the timing-critical paths in determining the clock frequency. As the issue width and instruction window size are increased for better performance, the number of ports and the size of the register file also should be increased. However, increasing the numbers of ports and the size of the register file could be a dominant delay factor. To overcome these problems, the banked integer register file scheme can be adopted to reduce the access time by 20% while reducing instructions per cycle (IPC) by less than 5%, though

Manuscript received Feb. 14, 2008; revised May 19, 2008; accepted June 23, 2008.

This work was supported by the Second Brain Korea 21 Project and Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R01-2007-000-20750-0).

Hyung Beom Jang (phone: + 82 2 3290 3571, email: kuphy01@korea.ac.kr) and Sung Woo Chung (phone: +82 2 3290 3194, email: swchung@korea.ac.kr) are with the Division of Computer and Communication Engineering, Korea University, Seoul, Rep. of Korea.

Eui-Young Chung (email: eychung@yonsei.ac.kr) is with the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Rep. of Korea.

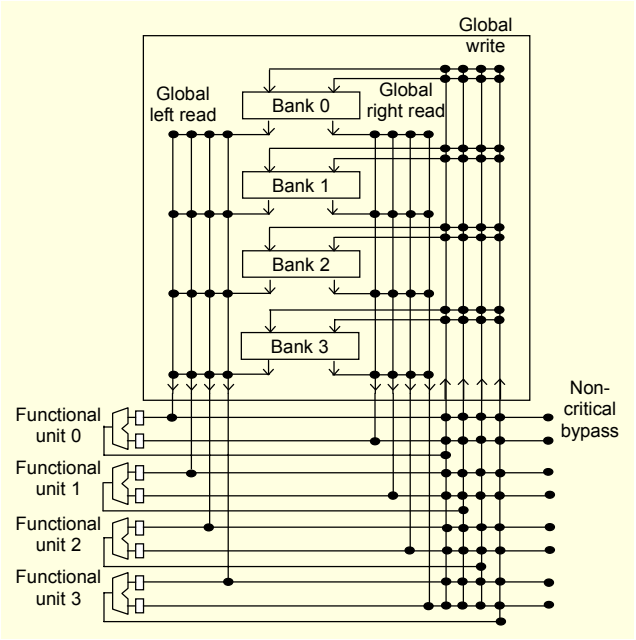

Fig. 1. Structure of the banked integer register file scheme [2].

the register file area is increased by 11% [2]. As shown in Fig. 1, the register file provides a total of eight global read ports and four global write ports using four interleaved register banks. In addition to these global ports, each bank has two local read ports and two local write ports connected to each other by the bit-line. This structure simplifies the local-global port crossbar by connecting one local port on each bank to the global left operand buses and the other to the global right operand buses. This allows any instruction to get both operands from the same bank in one cycle. The register file partitioned with several sub-word banks handles the inevitable bank conflicts without compromising cycle time or adding excessive complexity by containing wakeup-select loop.

III. Simulation Environments

We first compare the temperature of the banked integer register file [2] to that of the conventional non-banked register file. Then, we evaluate the performance of microprocessors that have a different register file structure when the following DTM techniques are used: local toggling (LTOG), which is also called fetch toggling, and dynamic voltage scaling (DVS). We set the DTM trigger threshold temperature to 90°C. Finally, we evaluate the total processor leakage power consumption, when the banked register file is adopted.

For processor simulations, we used the SimpleScalar [4] simulation toolset. The processor configuration is the same as that used in [1], and the Alpha 21264 floorplan is used. Due to the additional circuitry overhead, the area of the banked register file is larger than that of the conventional register file [2]. However, for conservative evaluation, we did not consider this area increase. Wattch [5], SimpleScalar, and HotLeakage [6] were also incorporated into our processor simulator to evaluate the dynamic and leakage power consumption of each functional unit in the microprocessor. We used HotSpot [1], which was validated against a commercial microprocessor, to convert the power trace into temperature at the architectural level. For simulations, we selected eight applications (crafty, gcc, gzip, mcf, parser, perlbmk, twolf, and vpr) from the SPEC2000 benchmarks suite [7].

IV. Simulation Results

Figure 2 shows the peak temperature of six units located near the integer register file (including the integer register file), assuming that no DTM technique is used. As shown in Fig. 2, the banked integer register file is cooler than the non-banked integer register file by as much as 12.8°C on average. When DTM techniques are applied, the banked register file invokes far fewer DTM operations, which may result in performance improvement. Note that the other functional units do not cause any DTM operation since their peak temperatures are below 90°C. In addition, the temperature of the units adjacent to the integer register file is also reduced by 1°C to 2°C since heat dissipation from the integer register file to the adjacent units is reduced.

Table 1 describes the performance improvement of the banked integer register file. The performance improvement with the banked integer register file is 13.37% on average. The lowered temperature of the banked integer register file reduces the number of cases exceeding a pre-defined DTM trigger threshold, which reduces the number of DTM operations. Consequently, the reduced number of DTM operations leads to performance improvement.

Table 2 shows the leakage reduction rate of the functional units near the integer register file including the integer register file and the total processor leakage reduction rate. The banked integer register file reduces the leakage power of the integer register file by as much as 56.36% on average. Since leakage power is dependent on temperature, the decrease in temperature caused by the reduced dynamic power in the banked integer register file leads to the leakage reduction in the banked integer register file. Additionally, the reduction in dynamic power consumption also leads to reduced heat dissipation to the adjacent functional unit, which results in 1% to 5% leakage reduction of the adjacent functional units. Thus, the total processor leakage power consumption is reduced by 10.48% on average by adoption of the banked register file scheme.

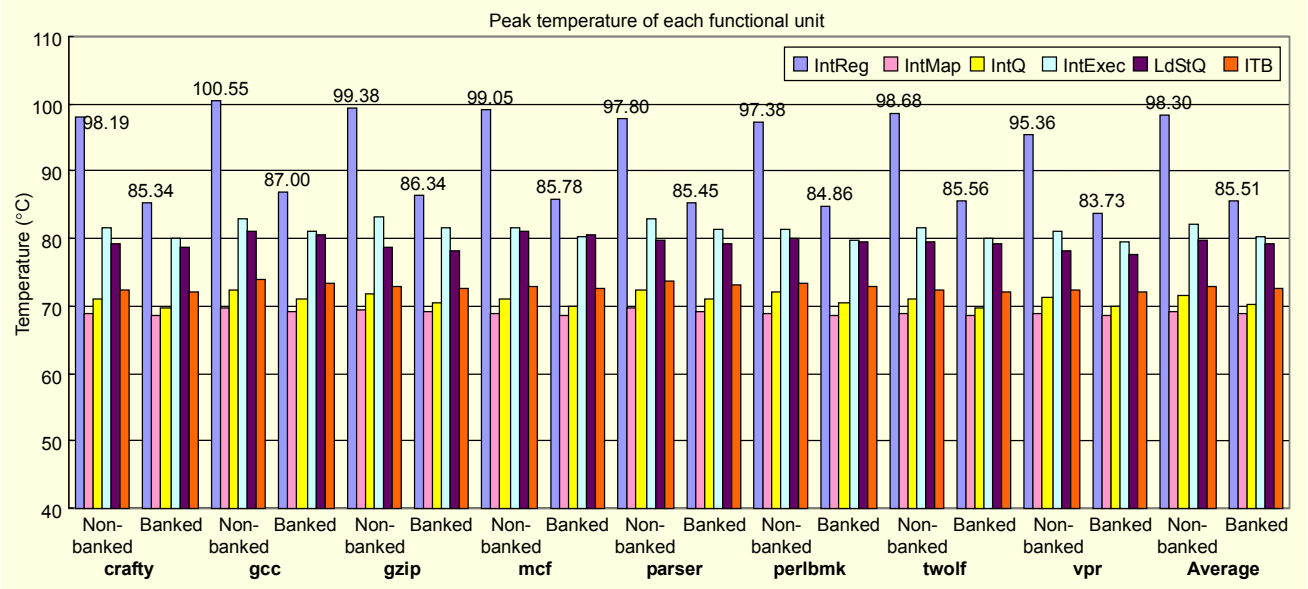

Fig. 2. Temperature comparison between the non-banked register file and the banked register file (without DTM techniques).

Table 1. Performance improvement with DTM techniques.

Application	Non-banked (cycles)	Banked (cycles)	Performance improvement (%)
crafty	190,711,859	161,586,745	15.27
gcc	220,363,620	193,721,210	12.09
gzip	206,822,730	174,664,358	15.55
mcf	162,334,899	133,973,548	17.47
parser	243,746,767	212,881,372	12.66
perlbmk	236,811,226	214,767,216	9.31
twolf	190,275,773	160,208,578	15.80
vpr	250,744,348	228,706,557	8.79
Average			13.37

Table 2. Leakage reduction.

Application	IntReg (%)	IntMap (%)	IntQ (%)	IntExec (%)	LdStQ (%)	ITB (%)	Total leakage (%)
crafty	56.20	0.77	3.33	4.26	1.30	0.82	10.55
gcc	56.95	0.96	3.87	4.94	1.61	1.03	10.83
gzip	56.25	0.91	3.65	4.66	1.59	0.99	11.03
mcf	56.43	0.68	3.16	4.05	1.12	0.72	10.71
parser	55.96	0.97	3.74	4.77	1.68	1.04	9.94
perlbmk	56.87	0.87	3.57	4.44	1.48	0.88	10.23
twolf	56.49	0.69	3.24	4.11	1.17	0.73	10.65
vpr	55.60	0.93	3.55	4.45	1.61	0.98	9.71
Average	56.37	0.84	3.51	4.46	1.43	0.89	10.49

V. Conclusion

In this letter, we adopted the banked integer register file to improve performance and to reduce leakage power consumption by decreasing the temperature of the integer register file. The simulation results show that the banked register file significantly reduces heat dissipation, resulting in an average performance improvement of 13.37%. The lowered temperature also reduces total processor leakage power by an average of 10.49%.

References

[1] K. Skadron et al., "Temperature-Aware Microarchitecture: Modeling and Implementation," *ACM Trans. Architecture and Code Optimization*, vol. 1, no. 1, Mar. 2004, pp. 94-125.

[2] J.H. Tseng and K. Asanović, "Banked Multiported Register Files for High-Frequency Superscalar Processors," *Proc. 30th ISCA Conf.*, June 2003, pp. 62-71.

[3] H.B. Jang, J.H. Kong, and S.W. Chung, "Adopting the Banked Register Files for Temperature-Aware Microprocessors," *Proc. ITC-CSCC Conf.*, July 2007.

[4] T. Austin, E. Larson, and D. Ernst. "Simplescalar: An Infrastructure for Computer System Modeling," *Proc. IEEE Computer Magazine*, vol. 35, Feb. 2002, pp. 59-67.

[5] D. Brooks, V. Tiwari, and M. Martonoshi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," *Proc. 27th ISCA Conf.*, June 2000, pp. 83-94.

[6] Y. Zhang et al., "HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects," *Tech. Report CS-2003-05*, Univ. of Virginia, Dept. of Computer Science, Mar. 2003.

[7] Standard Performance Evaluation Corp., http://www.spec.org/.