FUNCTIONAL UNITS POWER GATING IN SMT PROCESSORS

Houman Homayoun[1], Kin F. Li[1] and Setareh Rafatirad[2]
[1]Department of Electrical and Computer Engineering, University of Victoria, {homayoun,kinli}@uvic.ca
[2]Department of Computer Science, Tehran-Azad University of Technology, rafatirad@azad.ac.ir

ABSTRACT

Power consumption has emerged as a primary concern in processor design constraints. Power-aware techniques are being applied at all levels of circuit and system design. These techniques aim at reducing power or energy dissipation in all types of computer equipments while meeting a desired throughput. At the architectural level, Power-aware design has been an active area of research in the last decade for superscalar processors. Simultaneous multithreading processor (SMT), introduced as a complementary architecture to superscalar to increase throughput, has received less attention in the context of low-power design techniques. In SMT processors functional units are one of the major power consumers. In this paper we first study the opportunity for reducing the power consumption of functional units. Our results show that functional units are idle for a significant portion of the total execution cycle. Then we reuse and evaluate a microarchitectural technique to reduce functional unit power through power gating which has been recently proposed for superscalar processors. We show that in SMT processors, this technique can reduce floating point unit power considerably while maintaining performance.

1. INTRODUCTION

Power dissipation in a CMOS circuit can be classified as dynamic or static. Dynamic power dissipation is the result of switching activity while static power dissipation is due to leakage current. Subthreshold leakage, gate oxide tunneling, drain-induced barrier lowering (DIBL), gate induced drain leakage, hot carrier effects, reverse-biased PN junctions, and punchthrough currents [7,8] are various sources of leakage current. Among them the subthreshold leakage is considered to be an important contributor. Subthreshold leakage current flows from drain to source even when the transistor is off (we refer to this state as idle). A conventional effective technique to reduce subthreshold leakage is to block the supply voltage from reaching the transistor which is referred to as power gating.

Recent studies in superscalar processors [1] have shown that functional units are idle for a significant portion of execution time. Hu et al. have shown that functional units are idle for more than 60% of total execution cycles across SPEC2K benchmarks for a superscalar processor. In this paper we investigate the opportunity for power gating the functional units in SMT processors. Moreover, we apply Hu's technique to SMT processors and evaluate its effectiveness to reduce power consumed in functional units while maintaining performance.

2. POWER GATING

Figure 1 shows the transition states in power gating. In cycle C1 the power gating signal is received. The structure responds to the signal by pushing down the supply voltage to zero. This requires some cycles, as shown. After the source voltage is fully discharged, the structure goes into deep sleep mode in which it consumes virtually no power. At cycle C2 the structure receives a wakeup signal. It responds to this signal by pulling up the supply voltage to Vcc. This again requires some extra cycles. In our work, the transitions from active to sleep (sleep) and from sleep to active mode (wakeup) are defined as overhead_cycles. According to past studies [1], power gating the computational logic would save negligible power for the

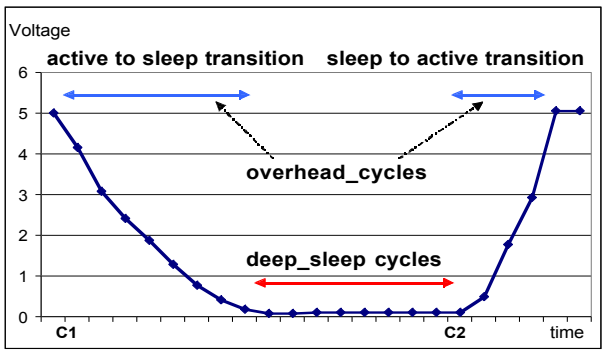

Figure 1: Transition state in power gating.

0-7803-9195-0/05/$20.00 ©2005 IEEE.

overhead_cycles intervals. We refer to the rest of the transition cycles as deep_sleep cycles in which the power gated structure is in fully discharge state and consumes virtually no power.

In figure 2 we present how power gating is achieved using a header transistor to block supply voltage from reaching a functional unit. The power gate detection circuit decides when it is appropriate to turn off the voltage supply. Once the sleep signal is generated, and after a transition period, the Vcc signal will be blocked from reaching the functional unit. As it appears, the extra hardware cost for power gating is a single transistor and a detection circuit. Accordingly the main complexity of power gating is determined by the detection circuit.

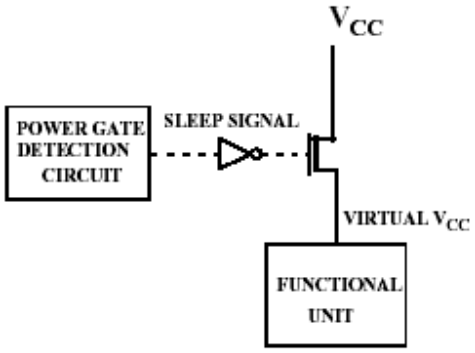

Figure 2: Schematic showing major blocks exploited in power gating.

3. MOTIVATION

Recently, several manufacturers announce small scale SMT processors. To model this, our baseline SMT processor comprises two thread contexts capable of running two benchmarks simultaneously. We define power gating opportunity as the percentage of total execution cycles a functional unit is in deep_sleep cycles state (figure 1) in which it dissipates no power. In figure 3 we report the power gating opportunity for various functional units for a multiprogram workload of a subset of the SPEC CPU2000 benchmark suit [9]. For each thread in a workload we simulated 200 million instructions after skipping the initial 200 million instructions. We report

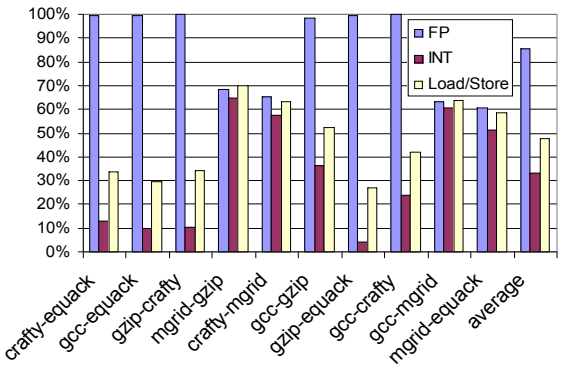

Figure 3: Functional units power gating opportunity when overhead cycle is zero.

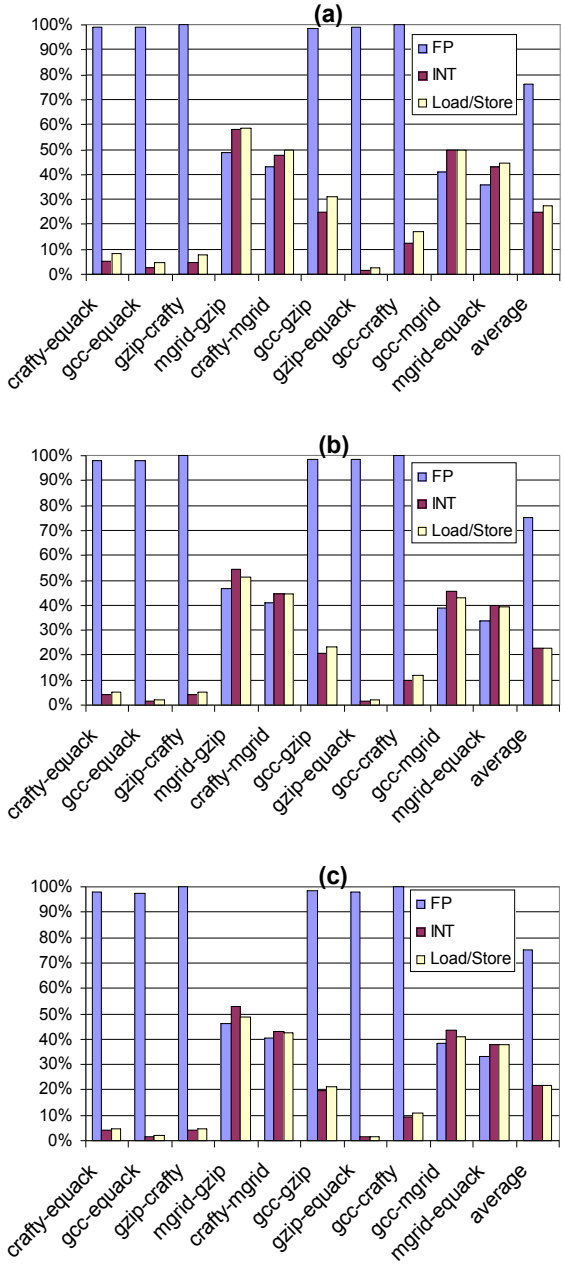

Figure 4: Functional units power gating opportunity for a) 6 b) 14 and c) 20 overhead_cycles.

the results when there is no overhead_cycles. Zero overhead_cycles shows the upper-bound opportunity for power gating the functional units. This in fact shows the percentage of total execution cycles a functional unit is in idle state while running a program. Across all workloads, the floating point unit is idle for a considerable percentage of the total execution cycles (around 85% on average). Average idle period is least for the integer unit (33% of the total execution cycles).

To provide insight into how variation in overhead_cycles impacts power gating opportunity, in figures 4(a), 4(b) and 4(c) we also report for the same processor when overhead_cycles is 6, 14 and 20 respectively. As expected, increasing the overhead_cycles reduces the chances of power gating. Integer ALU and load/store unit do not have much opportunity for power gating. This is particularly the case for 20 overhead_cycles where they can be power gated for only 21% of the execution cycles. For different overhead_cycles, floating point unit has the highest opportunity for power gating (more than 70% of the execution time).

We conclude from figure 3 that there is motivating opportunity in SMT processors to exploit idle times and to power gate floating point unit to reduce power dissipation. However, identifying idle times early enough is a challenging problem. Moreover, reactivating the gated execution units soon enough is critical since stalling instruction execution could come with a performance penalty.

4. MICROARCHITECTURAL TECHNIQUE TO POWER GATE FUNCTIONAL UNITS

Several microarchitectural techniques such as power-gating [2], dual threshold domino circuit [3], input vector control [4], body-bias control [5] and time-based power gating [1] have been proposed to reduce the power of idle logics in superscalar processors. Among them, only time-based power gating [1] targets functional units. It attempts to reduce functional units' leakage power through the detection of their long idle time. Once such long consecutive idle cycles are detected, the corresponding functional unit is turned off until it is requested again. In this section we reuse and evaluate this technique in SMT processors.

4.1. Time-Based Power Gating in SMT processors

In time-based power gating we monitor the functional units and power gate them once a long enough idle cycle is detected. We refer to this consecutive idle cycle as idle_detect. Hu et al. [1] have shown that for a wide issue superscalar processor this technique can put a floating point unit to sleep for up to 28% of the execution cycles at a performance loss of 2%. Using the outcome of branch predictor they have shown that it is possible to gate-off fixed point functional units for 40% of the execution cycles with the same performance loss. In this work we investigate the efficiency of this technique for simultaneous multithreaded processors.

As shown, in SMT processors, integer unit and load/store unit power gate savings are not considerable as compared to the power gate savings in floating point functional unit. This is particularly true when the overhead_cycle increases to 20 cycles. This shows that integer and load/store units are more frequently in use than the floating point functional unit. As a performance loss is associated with each wakeup of a power gated structure, it is not worthwhile to power gate integer and load/store units. Accordingly we do not power gate these two units to keep performance loss at a low level.

5. METHODOLOGY AND RESULTS

In this section we report our analysis framework. For the microarchitectural simulation, we used a modified version of SMTSIM 2.0 alpha [6]. The base processor model is detailed in Table I. To evaluate time-based power gating we report performance and percentage of total execution cycles when a functional unit is gated-off. We assume power gating overhead_cycle to be 10 cycles.

Table 1: Base processor configuration.

Pipeline	9 stages
Fetch Policy	8 instructions/cycle, up to 2 threads
Functional Units	6 integer, 3 floating point
Instruction Queues	64-entry integer and floating point queues
Renaming Registers	100 integer and floating point
Retirement Bandwidth	12 instructions/cycle
Branch Predictor	Hybrid Predictor
BTB	4KB entries, 4-way set associative
I-Cache	256KB, 2-way set associative, single ported
D-Cache	256KB, 2-way set associative, dual ported
L2 cache	16 MB, direct mapped, 20-cycle latency, fully pipelined
Memory bus	128 bits wide, 4-cycle latency

Figure 5 shows the percentage of cycles the floating point unit is power gated when idle_detect threshold is set to 50 cycles. In the same figure we report performance degradation when this technique is applied. As shown, on

average, floating point unit can be power gated for more than 70% of total execution cycles with a small performance degradation (around 1% on average). Increasing the idle_detect threshold reduces the power gate savings. On the other hand, decreasing the idle_detect threshold increases the performance cost. With an idle_detect threshold sets to fifty cycles there is considerable power savings and at the same time negligible performance degradation.

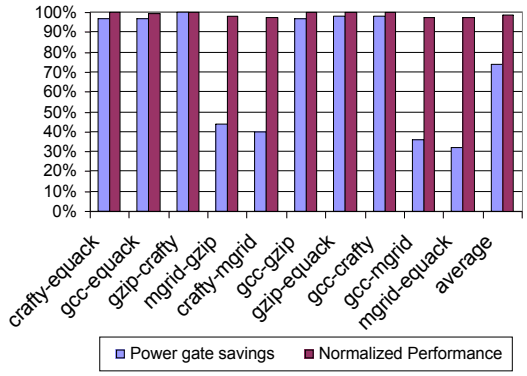

Figure 5: Power savings and performance degradation when power gate technique is applied to floating point functional unit.

6. CONCLUSION

In this paper we analyzed how power gating could be exploited in SMT processors. In particular we investigated how variations in timing overhead and design parameters impact power gating opportunity. Our study shows that it is possible to power gate floating point unit while maintaining performance. We also show that increasing the overhead cycles decreases integer and load/store units power savings significantly. Future studies will examine approaches to improve power-efficiency of power gating techniques for all functional units in SMT processor. The effect of overhead_cycle variation on performance loss and power savings will also make an interesting study.

7. REFERENCES

[1] Zhigang Hu, et al, "Microarchitectural Techniques for PowerGating of Execution Units." ISLPED (2004).
[2] Powel, M., Yang, S., Falsafi, B., Roy, K., and Vijaykumar, T. Gated, "Vdd: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories." ISLPED (2000).
[3] Kao, J., and Chandrakasan, A., "Dual-Threshold Voltage Techniques for Low-Power Digital Circuits." IEEE Journal of Solid State Circuits 35 (2000).
[4] Johnson, M., Somasekhar, D., Chiou, L., and Roy, K., "Leakage Control With Efficient Use of Transistor Stacks in Single Threshold CMOS." IEEE Transactions on VLSI Systems 10 (2002).
[5] Duarte, D., Tsai, Y. F., Vijaykrishnan, N., and Irwin, M. J., "Evaluating Run-Time Techniques for Leakage Power Reduction." ASPDAC (2002).
[6] Dean Tullsen, Susan Eggers, and Henry Levy, "Simultaneous Multithreading: Maximizing On-Chip Parallelism." ISCA (1995).
[7] Mamidipaka, M., Khouri, K., Dutt, N. and Abadir, M., "Leakage Power Estimation in SRAMs," CECS Technical Report #03-32, Center for Embedded Computer Systems, University of California Irvine. (2003).
[8] Tsividis, Y. P., "Operation and Modeling of the MOS Transistor." McGraw-Hill Book Company (1988).
[9] Standard Performance Evaluation Corporation: SPEC CPU 2000 V1.2 at www.spec.org/cpu2000