IPSJ Transactions on System LSI Design Methodology Vol. 4 182–192 (Aug. 2011)

Regular Paper

Design and Implementation Fine-grained Power Gating on Microprocessor Functional Units

Zhao Lei,[†1,†2] Daisuke Ikebuchi,[†1] Kimiyoshi Usami,[†3]
Mitaro Namiki,[†4] Masaaki Kondo,[†2]
Hiroshi Nakamura[†5] and Hideharu Amano[†1]

In this paper, we present a prototype MIPS R3000 processor, which integrates the fine-grained power gating technique into its functional units. To reduce the leakage power consumption, functional units, such as multiplier and divider can be power-gated individually according to the workload of the execution program. The prototype chip – Geyser-1 has been implemented with Fujitsu's 65 nm CMOS technology; and to facilitate the design process with fine-grained power gating, a fully automated design flow has also been proposed. Comprehensive real-chip evaluations have been performed to verify the leakage reduction efficiency. According the evaluation results with benchmark programs, the fine-grained power gating can reduce the power of the processor by 5% at 25˚C and 23% at 80˚C.

1. Introduction

Leakage power consumption has become a major design constraint in recent microprocessors. In CMOS technology, leakage power arises from the imperfect nature of transistors, where "leak" currents constantly flow from the power supply to the ground even in the off-state. Since leakage power increases exponentially with scaled threshold voltage, there has been a three to five times leakage raise per technology-generation [1]. Recently, leakage power constitutes 20–40% of the power budget of microprocessors [2], and its reduction techniques are indispensable in current and future process technologies.

Power Gating (PG) is one of the most effective leakage-reduction techniques, with which circuit blocks are not connected directly to the power grid but through power switches. To reduce the leakage power, the connection between idle circuit blocks and their power supply can be temporarily cut off by turning off power switches. In recent commercial microprocessors, core-level PG [3] has been implemented by inserting power switches between the global power grid and the power ring of processor cores. When the operating system knows the idle state of a processor core may last for certain times, the core, which can be either the CPU-core [4] or other heterogeneous cores [5], will be put into the sleep mode by shutting off its power supply. Although such a core-based PG control scheme is straightforward and easy to be applied, it misses leakage-saving opportunities when a portion of intra-core components are in the idle state. Moreover, the wakeup latency of core-level PG, which is the time needed to fully restore the power of a sleep core, is in the order of micro-second. That implies, with core-level PG, only conservative PG control schemes can be applied when a long idle time of the processor core is detected.

In contrast, PG control schemes which aggressively power on/off functional units within a processor core have also been studied [6)–8)]. By exploiting PG opportunities at a finer granularity, these schemes usually achieve better leakage reduction effects than the core-based PG scheme. However, PG is a non-ideal technique, and aggressively powering on/off functional units may incur unaffordable penalties on both performance and power consumption. Hu, et al. [6] analyze the costs involved in power-gating functional units and present an analytical model of the Break-Even Time (BET), which is the minimum time a functional unit should remain in sleep mode such that the saved leakage energy can compensate the dynamic energy overhead caused by powering on/off the unit. If a sleep event is shorter than BET, PG consumes more power instead of saving. To avoid such short-term sleep events, in the same paper, they have proposed a time-based PG control policy and a branch-guided policy. Youssef, et al. [7] have further exploited PG opportunities by tracking the behavior of the executing program across different time segments and predicting the length of idle periods of functional units; and Lungu, et al. [8] have proposed a scheme to guarantee the quality of PG with a successful monitor. In addition, compiler-based

[†1] Keio University
[†2] Graduate School of Information Systems, University of Electro-Communications
[†3] Shibaura Institute of Technology
[†4] Tokyo University of Agriculture and Technology
[†5] The University of Tokyo

schemes[9)–11)], which employ static code analysis or dynamic profiling to identify the time period when a functional unit is not used, have also been proposed.

However, all above papers miss the corresponding circuit-level techniques. Since functional units are power-gated at runtime, high-speed fine-grained PG techniques that can better exploit leakage-saving opportunities spatially and temporally are indispensable.

In our previous works[12),13)], we have presented a framework to implement the fine-grained PG on microprocessor functional units by integrating circuit-level, architecture-level, and system software techniques. At circuit-level, we have proposed a fine-grained PG technique, which has nano-second order wakeup latency and can be implemented at arbitrary granularity. At architecture-level, a PG control scheme, which keeps a functional unit active only when being used, has been applied. In addition, BET-aware PG control schemes, which are guided by the system software (compiler and operating system) have also been proposed to achieve maximum leakage reduction effects. The power evaluation comes from a piecewise-linear power model which is based on the transistor-level simulation of each functional unit. While the evaluation result is reliable regarding the power reduction effects on each functional unit, it ignores the impact of PG on the whole processor, in terms of both functionality and power consumption. In another word, the feasibility of proposed methodology has not been proved.

In this paper, we apply the proposed framework to a real-chip implementation[⋆1]. The feasibility and power reduction efficiency of the proposed methodology have been verified with real-silicon evaluation. The contributions of the paper can be summarized as follows:

- We have proposed a fully automated design flow of fine-grained PG. Compared with our previous work, no manual edit on layout is needed. Moreover, we have improved the design methodology, and the area overheads of functional units have been reduced from 41% to 9.2%.
- We have implemented a MIPS R3000 prototype chip with Fujitsu's 65 nm CMOS technology. The prototype chip integrates the fine-grained PG technique into its functional units, which can be power-gated individually at runtime. To the best of our knowledge, this is the first chip which provides fine-grained PG control schemes on functional units.
- The feasibility of the proposed design flow has been verified with real-silicon evaluations. Furthermore, comprehensive real-chip evaluations have been executed to measure leakage-saving results and other important parameters (BET, wakeup latency and so on). We have measured these parameters at different temperatures by using a thermal chamber, and the obtained results are more reliable than those obtained from simulations.

The rest of this paper is organized as follows. Section 2 introduces the fine-grained PG design methodology; and the PG control schemes on functional units are presented in Section 3. In Section 4, we illustrate the implementation of the prototype chip; and the real-chip evaluation results are shown in Section 5. Section 6 is conclusion and the future work.

2. The Fine-grained PG Methodology

In this section, we illustrate the fine-grained PG technique which is integrated into the functional units design in this work to reduce leakage power at runtime. In addition, a fully automated design flow will be presented.

2.1 Fine-grained PG

Unlike the conventional PG, fine-grained PG employed in this work requires a set of special cells each of which has its own virtual ground (VGND) line (such a set of cells is referred as PG-cells in this paper). As shown in **Fig. 1**, VGND lines from several cells are connected through a shared high-Vth footer power switch to the real GND. If the sleep signal is set on, the logic block is put into the sleep mode by cutting off the connection between VGND and GND. In this case, the VGND is charged up to a voltage near the VDD, and the leakage current is reduced consequently. When the power switch is turned on, parasitic capacitance on the VGND line is discharged through the power switch, and a certain amount of time (wakeup latency) is required to wait the voltage on VGND to become stable.

The wakeup latency is affected by physical parameters such as power switch size and VGND capacitance. In order to fine tune these parameters, we used a Locally-Shared Virtual ground (LSV) scheme[15)] shown in **Fig. 2**. With this

[⋆1] This work is based on Ref. 14)

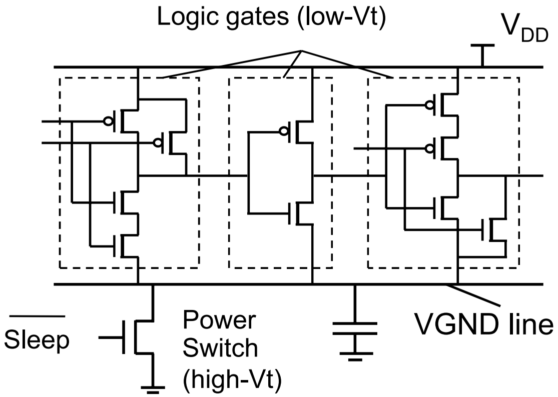

Fig. 1 Out line of power gating circuit.

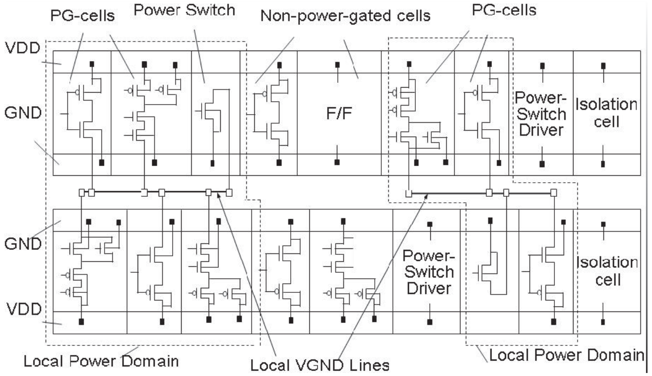

Fig. 2 A VGND architecture for a fine grain PG.

scheme, the entire PG target is partitioned into smaller local power domains, and the local VGND line and the power switch are shared only within a local power domain. Although power switches are controlled by a unique control signal for each PG target, the size of power switches can be tuned independently, implying that the IR-drop on the VGND line can be managed easily by selecting appropriate-sized power switches for each local power domain. Furthermore, since existing ground rail in the PG-cells is used as the real ground, permanent power networks are reachable throughout the PG target; thus, non-power-gated cells such as flip-flops, clock buffers, repeaters, power switch drivers, and isolation cells can be distributed arbitrarily among local power domains without incurring power-routing. Furthermore, an in-rush current suppression mechanism [16], which skews the wakeup timing of each local domain by simply down-sizing a portion of the leaf drives of the power-switch-driver tree, has also been adopted with the LSV scheme.

Compared with the UPF-based methodology [17], the LSV scheme can control the size and the number of local power domains by taking into account the given requirement on wakeup latency. As a result, the wakeup latency of fine-grained PG is typically less than a few nano-seconds (we will confirm this in Section 5). Moreover, there is no constraints on the placement of power switches and non-power-gated cells as the UPF-based methodology, and the power integrity issues caused by power-routing can be minimized. Compared with the cell-based PG methodology [18], which integrated a power switch in each primitive cell, our scheme has less area overheads. Compared with the ring-based PG [18], the IR-drop target can be managed easily with smaller number power switches, and the non-power-gated cells can be placed within a PG target with our methodology.

2.2 The Design Flow of Fine-grained PG

To facilitate the design process with fine-grained PG, we newly developed a fully automated design flow. As shown in **Fig. 3**, the design flow can be explained as follows:

1) A set of PG-cells is generated, generated cells include GDS file and the timing library.

2) An RTL model of a PG target with sleep-control signals is designed.

3) The RTL model is synthesized by using Synopsys Design Compiler.

4) Isolation cells are inserted to all the output ports of the synthesized netlist in order to prevent the propagation of floating output values when the PG target is in the sleep mode.

5) The netlist with the isolation cells is placed by using Synopsys Astro.

6) The local power domain is partitioned, local VGND lines are formed, and the power switches are inserted between the VGND and GND lines by using Sequence Design's CoolPower [19].

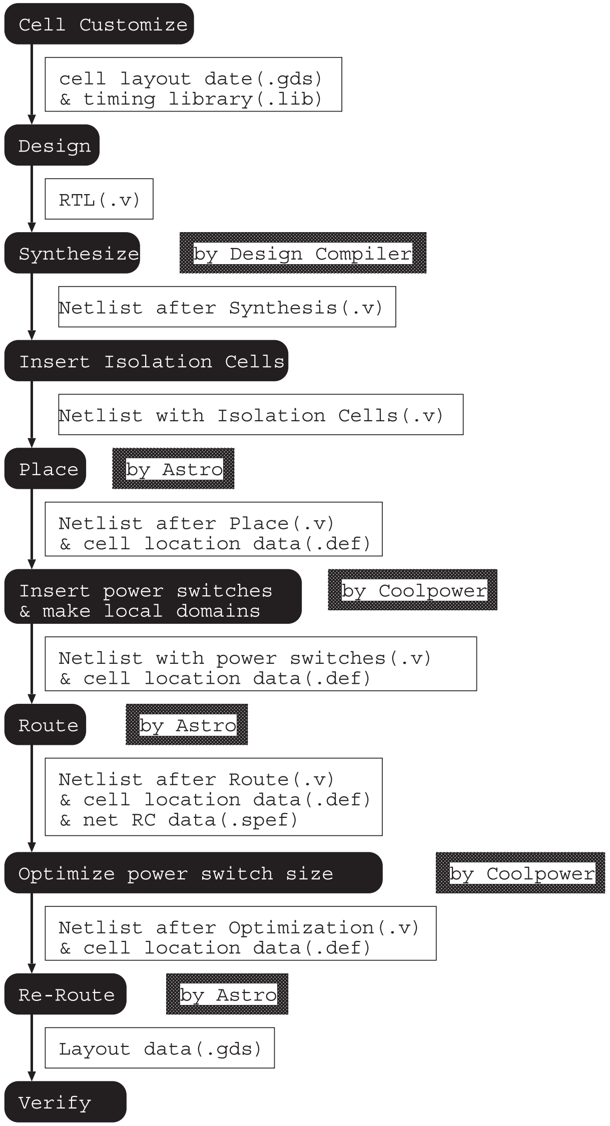

Fig. 3 Design flow.

7) The netlist with the power switches is routed by using Synopsys Astro.

8) The previous two steps are performed again for the purpose of VGND optimization, power switch sizing, and routing.

In the end of the design flow, the GDS file of a PG target will be generated. Since PG-cells can coexist with common cells in a row, the conventional timing-driven placement and routing methodology can be used. Furthermore, this flow is fully automated, and the additional design complexity for the fine-grained PG is small.

3. Runtime PG Control Schemes on Functional Units

In this section, we propose PG control schemes which can dynamically power on/off functional units in response to workloads of programs currently running on the processor. Here, a widely used 32-bit embedded processor, MIPS R3000[20], is selected as the target processor. As shown in **Fig. 4**, MIPS R3000 provides a standard five-stage pipeline structure, consisting of Instruction Fetch (IF), Instruction Decode (ID), Execution (EX), Memory Access (MEM), and Write Back (WB). To apply the fine-grained PG technique to MIPS R3000, we select following units as the PG targets:

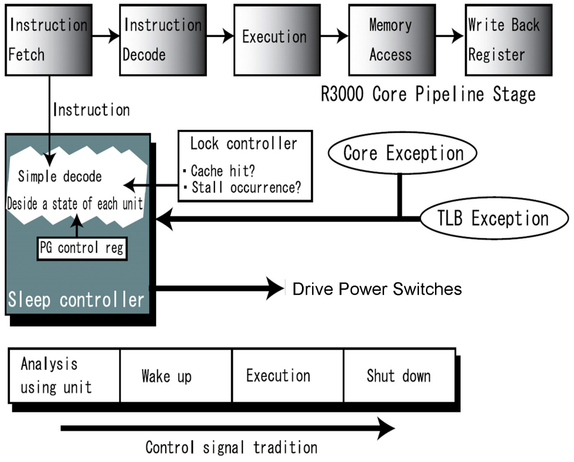

Fig. 4 Sleep control by the fetched instruction.

- CLU (Common arithmetic and Logic Unit)
  A general computational unit for addition and subtraction operations. It can be put into the sleep mode when branch, NOP, or memory access instructions without address calculation are fetched.
- Shifter
  A barrel shifter which can shift a data word by a specified number of bits in one clock cycle. Since the shifter occupies the considerable area but is not so frequently used, it is implemented as an individual unit.
- Multiplier
  A 32-bit multiplier which takes 4 clock cycles for each multiplication operation. If upper 16-bit of either operand is all-0, the upper part of the multiplier can be put into the sleep mode.
- Divider
  A 32-bit Divider which takes 10 clock cycles for each Division operation.

Note that, functional units occupy about 55% of the area of the processor core (without on-chip caches and TLBs), and their usage is easy to be identified based on the fetched instructions. Moreover, all of them are implemented with combinational circuits, implying that state retention techniques are not required. Thus, only 1-bit sleep signal is needed to control the mode of each unit.

3.1 Fundamental PG Control Policy

A fundamental PG control policy tries to put each functional unit into the sleep mode right after finishing its operation. As shown in Fig. 4, the mode of function units is controlled by sleep signals that are generated from a dedicated sleep controller. When an instruction is fetched in the IF stage, the sleep controller checks the fetched instruction and judges which unit is to be used. Here, the target working frequency of the processor is set to 100 MHz, and we assume the wakeup latency is 1 clock cycle (10 ns). To hide the wakeup latency, a simple decoder is provided in the IF stage to detect which functional unit will be used by the currently fetched instruction. As soon as a functional unit is detected, a sleep signal will be generated and sent to the unit immediately. Thus, when the instruction reaches the EX stage, the required unit had already been fully powered up, and no performance detriment will be introduced by the fundamental policy.

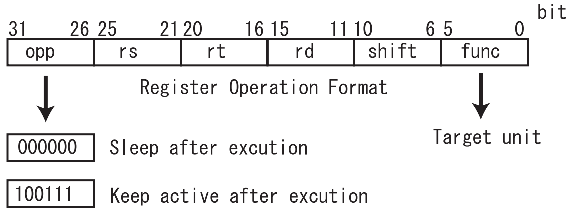

Fig. 5 Non-PG instructions.

When an instruction is fetched in the IF stage, the decoder checks the upper most 6 bits of the instruction, and judges whether the instruction executes a R-Type operation (ROP) [20]. If so, the functional unit to be used can be identified by the last 6 bits of the instruction. Otherwise, extra judgments are needed to decide whether it is an I-Type instruction which uses CLU for address-calculation.

3.2 BET-aware PG Control

The leakage reduction effects of the fundamental PG control policy are sensitive to the BET in that functional units are switched between the sleep mode and the active mode frequently. Since extra power consumption is induced by powering on/off functional units as well as the sleep controller, the fundamental policy has a risk to increase the power consumption instead of saving. For example, when multiplication operations are executed iteratively with a small interval, the multiplier will be woken up soon after its shut-off. If the sleep time of multiplier is less than the BET, the mode-transition overheads of PG will increases the total power consumption. In this work, we employ the compiler to detect instructions that may cause small sleep intervals (smaller than BET); and mode-transition overheads can be eliminated by keeping functional units active after its operation. For this purpose, we introduce a set of non-PG instructions.

Figure 5 shows an example of non-PG instructions. In MIPS R3000 ISA, when the most upper 6 bits (ope-code) is all-0, the instruction performs computational operation; and the type of operation is determined by the last 6 bits. Here, we used “100111”, which is not defined in the original ISA, as the upper 6 bits to indicate non-PG instructions. After executing a non-PG instruction, the corresponding functional unit will not be power-gated, but kept in the active state. Such an active state will be kept until the another instruction, which use the same functional unit but with all-0 ope-code, is executed. Thus, by replacing small-interval instructions with non-PG instructions, the overhead of power gating can be avoided.

Furthermore, leakage power is sensitive to the temperature, that is, it increases exponentially as the temperature rises. BET is also influenced by the temperature but in the opposite direction – BET becomes shorter at higher temperatures (we confirmed this in Section 5). Such a characteristic can be exploited to achieve better leakage reduction effects. When the chip is working at a low temperature, PG policies that power on/off functional units conservatively should be used to avoid mode-transition overheads; while more aggressive policies should be adopted in higher temperatures. In this work, we have implemented two different PG control policies that can be changed dynamically according to the chip temperature: (1) fundamental, runtime PG policy (as mentioned in Section 3.1), (2) units never going to the sleep mode. These policies are applied based on the value stored in a PG policy register, which can be written only in the kernel mode; and the operating system decides which policy should be used according to the information from an on-chip leakage monitor.

4. Geyser-1 Prototype Chip

To demonstrate the feasibility of the fine-grained PG technique and prove the leakage reduction effects of the PG control scheme presented in Section 3, a prototype chip, Geyser-1, has been implemented.

4.1 Design Policy

Fine-grained PG complicates the power grid design during layout. For this reason, our first prototype chip, Geyser-0[12),13)], failed to work due to unexpected power-rail shorts. To simplify the back-end design, the second prototype chip, Geyser-1, has been designed with following policies: (1) only the CPU core is implemented on a chip. Caches and TLBs provided in Geyser-0 are moved off-chip. (2) The design flow is improved so that no manual edit on the layout is needed. (3) 65 nm Fujitsu's high-Vth CMOS process is used instead of 90 nm standard process used in Geyser-0.

The decision to move caches off chip has serious impacts on the back-end design. Because of the pin-limitation problem, a part of address/data signals must be multiplexed. Additional delay of such multiplexers, long wires and I/O buffer to access off-chip caches severely degrades the operating frequency. Moreover, the package technique used for Geyser-1 also imposes limitations on the maximum frequency. As a result, the maximum clock frequency of Geyser-1 is set to be 60 MHz at the layout stage.

4.2 Implementation

Geyser-1 has been described by Verilog HDL, synthesized with Synopsys Design Compiler 2007.03-SP4, and layouted by using Synopsys Astro 2007.03-SP7. Fujitsu's 65 nm 12-metal-layout CMOS library CS202SZ (high-Vth process) is used as the standard cells library, in which core cells work at 1.2 V while I/O cells working at 3.3 V. As illustrated in Section 2, the fine-grained PG technique requires a set of customized PG-cells. We selected 106 cells from Fujitsu CS202SZ cell library and modified them to have separate VGND lines. These cells are used to build functional units during the placement and routing phase; and other parts of the processor use the common cells. In addition, we has designed power switches and isolation cells, which are also required by the fine-grained PG design flow.

Power switches are inserted in the post layout netlist by using Sequence Design's Cool Power 2007.3.8.5. Since the inserted power switches will increase the voltage of VGND (IR-drop problem), the performance of functional units in the active mode may be degraded. As mentioned above, the critical path of Geyser-1 sits in the IF/MEM stage, where the off-chip cache-access happens. Thus, by appropriately setting the IR-drop target according to the timing slack existing in the EX stage, the cycle time degradation of the whole processor can be avoided. Here, we determine the IR-drop target based on the timing analysis of each functional unit (With the proposed design flow, the IR-drop of a PG target can be managed easily, and we can set the IR-drop target of each functional unit independently. As a result, for the multi-cycle multiplier and divider, which have a large amount of timing slack, the IR-drop target is set as 200 mV; while for CLU and shifter, whose timing slack is tighter, the IR-drop is set as 100 mV. In both cases, no performance degradation will be incurred.

Table 1 shows the area of each functional unit. In the table, PS means the area of power switches, while ISO stands for the area of isolation cells. The overhead

Table 1 The area overhead of fine-grained PG.

	Total (μm^2)	PS (μm^2)	ISO (μm^2)	Overhead
CLU	3,752.8	296.4	79.2	10.3%
SHIFT	3,078.0	298.8	76.8	12.6%
MULT	23,863.6	1,762.0	153.6	8.5%
DIV	27,918.4	1,301.2	153.6	5.4%
others	46,304.4	-	-	-

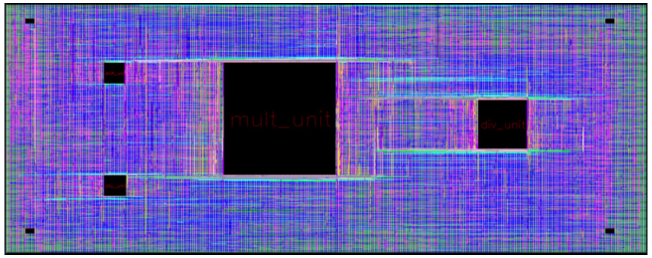

Fig. 6 Layout of Geyser-1.

of the fine-grained PG is 5.4%–12.6%, which are mainly caused by power switches and isolation cells.

The cells introduced by fine-grained PG (e.g., isolation cells and PG control circuit) will also cause additional leakage power consumption. According to the circuit-level simulation, the leakage overheads caused by isolation cells and PG control circuit (including the pre-decoder in the IF stage) are 0.44% and 3.03% of the whole processor.

Figure 6 shows the layout of Geyser-1. The chip size is 2.1 mm × 4.2 mm. As shown in the figure, the four black boxes in the middle of the layout are functional units which are implemented with fine-grained PG; and four small black boxes located near corners are leakage monitors.

5. Real-chip Evaluations

In order to evaluate the chip, we developed a dedicated board including an Virtex-4/LX FPGA board and socket for Geyser-1 chip. The power supply of Geyser-1 chip is completely separated from others for the purpose of accurate current-measurement. A thermal chamber is utilized to heat up the whole system and measure the power consumption of the chip at different temperatures. Since caches and TLBs are not included in Geyser-1, small and high speed memory for storing instructions and data are implemented with BRAMs in the FPGA.

5.1 Clock Frequency and Wakeup Latency

First, we evaluate the maximum clock frequency and the wakeup latency. For this purpose, we define two different working mode of the chip. The RTPG mode means the test program running on the processor does not include any non-PG instructions, and a functional units will be immediately powered off after its operation. On the other hand, the ACT mode indicates that all functional instructions are replaced by non-PG instructions. When working at such a mode, functional units will always stay in the active mode.

A simple benchmark program is executed. When working at 60 MHz, the prototype chip works correctly at both the ACT mode and the RTPG mode. Since we assumed the wakeup latency is one clock cycle (Section 3), the evaluation result proves that the wakeup latency of the fine-grained PG is less than 17 ns.

5.2 BET

Since BET is an important design factor of fine-grained PG, we evaluate BET of each functional unit on the real chip. Here, we take the multiplier as an example to illustrate our measurement strategy. The test program is a simple loop which consists of a multiplication instruction, several idle cycles, and a return to the loop entrance. The longer the idle interval is, the less frequently the mode-transition happens; hence, the better power-saving can be achieved by PG. With such a measurement strategy, we investigate BET by changing the length of idle intervals and comparing the power difference between RTPG mode and ACT mode. **Figure 7** shows obtained values of the multiplier. The horizontal axis presents the length of idle interval in the form of clock cycles; while the vertical axis is the power difference of the test program which is executed at ACT mode and RTPG mode respectively. The BET is the interval at which the power difference of ACT mode and RTPG mode becomes zero. As shown in Fig. 7, when working at 25℃, the BET of the multiplier is 880 ns. **Table 2** presents the BET of four functional units when working at 25℃, 65℃, and 100℃, respectively. As shown in the table, BET decreases exponentially as temperature increase. Note that, with BET-aware PG control schemes mentioned in Section 3,

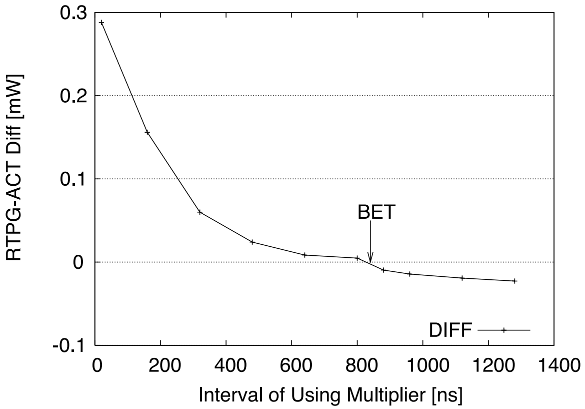

Fig. 7 BET of multiplier.

Table 2 BET of functional unis (ns).

	25˚C	65˚C	100˚C
CLU	1,080	280	120
SHIFT	1,340	320	140
MULT	880	240	100
DIV	640	180	80

the compiler will use the non-PG instructions if the interval of two instructions, which use the same functional unit, is less than the BET.

5.3 Leakage Power Reduction

In this subsection, we measure the leakage power when all functional units are staying in the active mode and the sleep mode respectively. After putting functional units into a given mode (sleep or active), we stop the clock signal, thus no dynamic but only leakage power of the processor can be measured. As shown in **Fig. 8**, the PG can reduce the leakage power of the whole processor core by 5% at 25˚C. It is worth noting that, when the temperature grows up, the effect of PG becomes more obvious.

5.4 Evaluations with Benchmark Programs

Evaluations with benchmark programs are also executed. We select two programs from MiBench [21]: Quick Sort (QSORT) from mathematics package and Dijkstra from the network package. In addition, we also use DCT (Discrete Cosine Transform) from JPEG encoder program as an example of media processing. Since the delay of the Block RAM inside the FPGA is large, the evaluation is performed at 10 MHz.

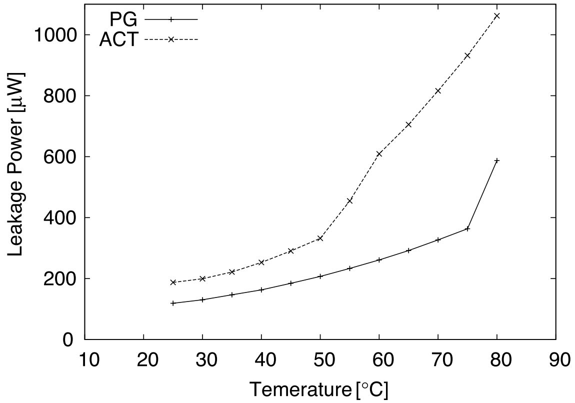

Fig. 8 Leakage power reduction of Geyser-1.

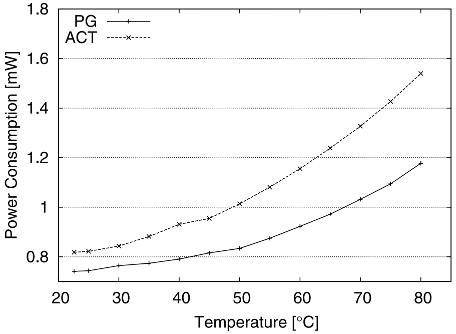

Fig. 9 Power for Dijkstra.

Figures 9, **10** and **11** show the power consumption of three benchmark programs working with PG and without PG. The best power reduction effect is achieved in Dijkstra which does not use the multiplier and divider. The total power consumption of the processor core can be reduced by 8% at 25˚C and 24% at 80˚C. Note that, power reduction effects are better than the values shown in Fig. 8. It is not strange because by putting functional units into the sleep mode,

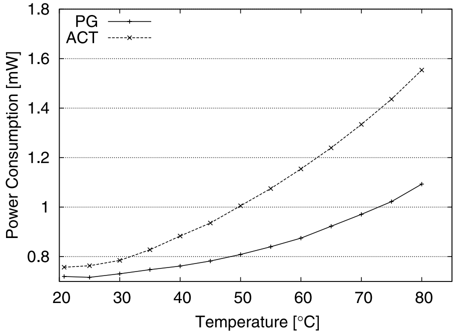

Fig. 10 Power for QSORT.

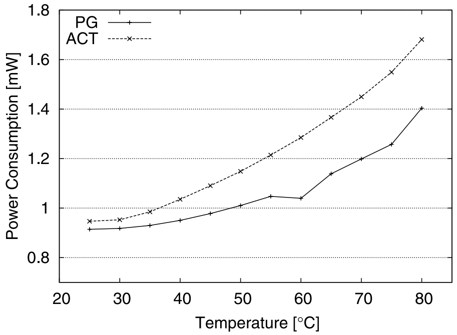

Fig. 11 Power for DCT.

redundant dynamic power consumptions, which caused by incomplete operand isolation, can also be eliminated. The power reduction of Quick Sort, which occasionally uses the multiplier and divider, is from 4% to 29%; and for DCT, which uses multiplier frequently, the power is reduced by 3% at 25°C and 17% at 80°C.

6. Conclusion and Future Work

Geyser-1, a prototype MIPS R3000 processor which implements the fine-grained power gating control on its functional units, has been presented. In this paper, we have discussed the design methodology of fine-grained power gating technique, the power gating control policy on microprocessor functional units, and the real-chip implementation and evaluation. To simplify the design process with the fine-grained power gating technique, a fully automated design flow have been also proposed. The real-chip evaluation of Geyser-1 shows the fine-grained power gating can reduce the power consumption of the the processor core by 5% at 25°C and 23% at 80°C.

Since Geyser-1 does not include on-chip caches, its maximum working frequency is restricted to 60 MHz. We have assumed the wakeup latency of fine-grained power gating is one clock cycle, thus, the real-chip evaluation can only verify that the wakeup latency is less than 17 ns, although circuit-level simulation shows it is less than 5 ns. Another problem is that the operating system is difficult to be ported without on-chip TLB. To address these problems, a new prototype chip – Geyser-2 [22], which is designed with on-chip caches and TLB, has been fabricated. A preliminary real-chip evaluation shows the fine-grained PG can work at 210 MHz without incurring any electric problems. Evaluating the leakage reduction effects of Geyser-2 with benchmark programs will be our next-step work.

Acknowledgments This research was performed by the author for STARC as part of the Japanese Ministry of Economy, Trade and Industry sponsored "Next-Generation Circuit Architecture Technical Development" program. The authors thank to VLSI Design and Education Center (VDEC) and Japan Science and Technology Agency (JST) CREST for their support.

References

1) Krishnamurthy, R., Mathew, S., Anders, M., Hsu, S., Kaul, H. and Borkar, S.: High-performance and low-voltage challenges for sub-45 nm microprocessor circuits, *6th International Conference On ASIC, 2005*, Vol.1, pp.283–286, IEEE (2006).

2) Joseph, R. and Martonosi, M.: Run-time power estimation in high performance microprocessors, *Proc. 2001 International Symposium on Low Power Electronics and Design*, pp.135–140, ACM (2001).

3) Jotwani, R., Sundaram, S., Kosonocky, S., Schaefer, A., Andrade, V., Constant, G., Novak, A. and Naffziger, S.: An x86-64 core implemented in 32 nm SOI CMOS, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp.106–107, IEEE (2010).

4) Kumar, R. and Hinton, G.: *Solid-State Circuits Conference-Digest of Technical Papers, 2009, ISSCC 2009, IEEE International*, pp.58–59, IEEE (2009).

5) Kanno, Y.: Hierarchical Power Distribution with 20 Power Domains in 90-nm

Low-Power Multi-CPU Processor, *ISSCC 2006*, pp.540–541 (2006).

6) Hu, Z., e.a.: Microarchitectural Techniques for Power Gating of Execution Units, *ISLPED*, pp.32–37 (2006).

7) Youssef, A., Anis, M. and Elmasry, M.: Dynamic standby prediction for leakage tolerant microprocessor functional units, *39th Annual IEEE/ACM International Symposium on Microarchitecture* (*MICRO'06*), IEEE Computer Society (2006).

8) Lungu, A., Buyuktosunoglu, A. and Sorin, D.: Dynamic power gating with quality guarantees, *Proc. 14th ACM/IEEE International Symposium on Low Power Electronics and Design*, pp.377–382, ACM New York, NY, USA (2009).

9) Zhang, W., Kandemir, M., Vijaykrishnan, N., Irwin, M. and De, V.: Compiler support for reducing leakage energy consumption, *Design, Automation and Test in Europe Conference and Exhibition, 2003*, pp.1146–1147, IEEE (2005).

10) Rele, S., Pande, S., Onder, S. and Gupta, R.: Optimizing static power dissipation by functional units in superscalar processors, *Compiler Construction*, pp.85–100, Springer (2002).

11) Roy, S., Ranganathan, N. and Katkoori, S.: A Framework for Power-Gating Functional Units in Embedded Microprocessors, *IEEE Trans. Very Large Scale Integration* (*VLSI*) *Systems*, Vol.17, No.11, pp.1640–1649 (2009).

12) Seki, N., Lei, Z., Kojima, Y., Ikebuchi, D., Hasegawa, Y., Ohkubo, N., Takeda, S., Kashima, T., Shirai, T., Usami, K., Sunata, T., Kanai, J., Mitara, M., Kondo, H., Nakamura, H. and Amano, H.: A Fine Grain Dynamic Sleep Control Scheme in MIPS R3000, *26th International Conference on Computer Design, 2008, ICCD2008, IEEE*, IEEE (2008).

13) Seki, N., Lei, Z., Kojima, Y., Ikebuchi, D., Hasegawa, Y., Ohkubo, N., Takeda, S., Kashima, T., Shirai, T., Usami, K., Sunata, T., Kanai, J., Mitara, M., Kondo, H., Nakamura, H. and Amano, H.: A Fine Grain Dynamic Sleep Control Scheme in MIPS R3000 (in Japanese), *IEICE Trans. Inf. Syst.* (*Japanese Edition*), Vol.J93, No.6, pp.920–930 (2010).

14) Ikebuchi, D., Seki, N., Kojima, Y., Kamata, M., Lei, Z., Amano, H., Shirai, T., Koyama, S., Hashida, T., Umahashi, Y., et al.: Geyser-1: A MIPS R3000 CPU core with fine grain runtime power gating, *Solid-State Circuits Conference, 2009, A-SSCC 2009, IEEE Asian*, pp.281–284, IEEE (2009).

15) Usami, K.N.: An Approach for Fine-grained Run-time Power Gating using Locally Extracted Sleep Signals, *24th International Conference on Computer Design, 2006, ICCD2006, IEEE*, IEEE (2008).

16) Usami, K., Shirai, T., Hashida, T., Masuda, H., Takeda, S., Nakata, M., Seki, N., Amano, H., Namiki, M., Imai, M., et al.: Design and implementation of fine-grain power gating with ground bounce suppression, *2009 22nd International Conference on VLSI Design*, pp.381–386, IEEE (2009).

17) Society, I.C.: IEEE Standard for Design and Verification of Low Power Integrated Circuits, *IEEESTD, 2009, 4809845* (2009).

18) Keating, M., Flynn, D., Aitken, R., Gibbons, A. and Shi, K.: *Low power methodology manual: for system-on-chip design*, Springer Verlag (2007).

19) Sequence Design, I.: Cool Power,
available from ⟨http://www.sequencedesign.com⟩.

20) Farquhar, E. and Bunce, P.: *THE MIPS PROGRAMMER'S HANDBOOK*, Morgan Kaufmann Publishers (1994).

21) Guthaus, M.R., Ringenberg, J.S., et al.: MiBench: A free, commercially representative embedded benchmark suite, *2001 IEEE International Workshop on Workload Characterization, 2001, WWC-4*, pp.3–14 (2001).

22) Lei, Z., Ikebuchi, D., Saito, Y., Kamata, M., Seki, N., Kojima, Y., Amano, H., Koyama, S., Hashida, T., Umahashi, Y., Masuda, D., Usami, K., Sunata, T., Kimura, K., Namiki, M., a.S., Nakamura, H. and Kondo, M.: Geyser-1 and Geyser-2: MIPS R3000 CPU Chips with Fine-grain Runtime Power Gating, *13rd IEEE Symposium on Low-Power and High-Speed Chips, 2010, CoolChips 2010, IEEE*, IEEE, pp.161–163 (2010).

(Received November 29, 2010)
(Revised March 4, 2011)
(Accepted April 24, 2011)
(Released August 10, 2011)

(Recommended by Associate Editor: *Makoto Nagata*)

Zhao Lei is a Ph.D. candidate at Keio University. His research interests include microprocessor achitecture and low power design. Currently, he is also a Research Fellow at University of Electro-Communications.

Daisuke Ikebuchi received his M.E. degree from Keio University, Yokohama, Japan, in 2010.

Kimiyoshi Usami received his B.S., M.S., and Ph.D. degrees from Waseda University, Tokyo, Japan, in 1982, 1984, and 2000, respectively. He is currently a Full Professor with Shibaura Institute of Technology, Kohtoh-ku, Tokyo. He was with Toshiba, Tokyo, and was involved in research and development in the field of low-power design of microprocessors and system-on-chips. From 1993 to 1995, he studied at Stanford University, Stanford, CA, as a Visiting Scholar. His current research interests include energy-aware computing and ultra-low voltage design. Dr. Usami is a member of IEEE, ACM, and IEICE in Japan.

Mitaro Namiki is a Professor in the Department of Computer Science, faculty of Technology, Tokyo University of Agriculture and Technology. He received Ph.D. from Tokyo University of Agriculture and Technology in 1992. His research interests include operating systems, programming languages, parallel processing, computer network.

Masaaki Kondo received his B.E. degree in College of Information Sciences and M.E. degree in Doctoral Program in Engineering from University of Tsukuba in 1998 and 2000 respectively, and Ph.D. degree in the Graduate School of Engineering from the University of Tokyo in 2003. He is currently an Associate Professor of Graduate School of Information Systems, University of Electro-Communications. His research interests include computer architecture, high-performance computing, and dependable computing. He is a member of IEEE, ACM, and IEICE.

Hiroshi Nakamura received his B.E., M.E., and Ph.D. degrees in Electrical Engineering from the University of Tokyo in 1985, 1987, and 1990, respectively. He was a Visiting Associate Professor at the University of California, Irvine from 1996 to 1997. He is currently a Professor of the Department of Information Physics and Computing at the University of Tokyo. His research interests include low-power VLSI design, power-aware computing, high-performance computer systems, and dependable computing. He is a senior member of IEEE and ACM.

Hideharu Amano received his Ph.D. from Keio University, Japan in 1986. He is now a Professor in the Department of Information and Computer Science, Keio University. His research interests include the area of parallel architectures and reconfigurable computing. He is a member of IEEE, IEICE and IPSJ.