A Fine-grain Dynamic Sleep Control Scheme in MIPS R3000

Naomi Seki, Lei Zhao, Jo Kei, Daisuke Ikebuchi, Yu Kojima, Yohei Hasegawa, Hideharu Amano
Keio University,
Toshihiro Kashima, Seidai Takeda, Toshiaki Shirai, Mitustaka Nakata, Kimiyoshi Usami
Shibaura Institute of Technology,
Tetsuya Sunata, Jun Kanai, Mitaro Namiki
Tokyo University of Agriculture and Technology,
Masaaki Kondo
The University of Electro-Communications,
Hiroshi Nakamura
The University of Tokyo

Abstract—A fine-grain dynamic power gating is proposed for saving the leakage power in MIPS R3000 by sleep control and applied to a processor pipeline. An execution unit is divided into four small units: multiplier, divider, shifter and other (CLU). The power of each unit is cut off dynamically, based on the operation. We tape-outed the prototype chip Geyser-0, which provides an R3000 Core with the power reduction technique, 16 KB caches and Translation Lookaside Buffer (TLB) using 90 nm CMOS technology. The evaluation results of four benchmark programs implemented based on the proposed technique with 90 nm CMOS technology. The evaluation results of four benchmark programs for embedded applications show that 47% of the leakage power is reduced on average with 41% area overhead.

I. INTRODUCTION

Recently, power dissipation has emerged as a major concern of CPU design. Since dynamic power reduction techniques have been well researched and applied in real systems, leakage power has been a large factor in the total power dissipation, particularly in CMOS circuits with less than 90 nm technology. Power Gating (PG) is a technique for reducing leakage power by shutting off the power of idle modules. Unlike other leakage power reduction techniques, including dual thresholds and biased bases, PG can be easily handled from the architecture layer and software layer of processors. By combining techniques in various layers with power gating, sophisticated leakage power savings are possible.

Traditionally, leakage power-saving techniques for a processor have been primarily applied to cache memory [1][2], which tends to work only partially and so is a good target of power saving. In order to reduce the total leakage power of a processor, the logic in a CPU with a high degree of activity must be the next target. However, traditional PG targets, such as a large power domain corresponding to a total CPU core and entire IP modules, and the wake-up latency from the power-off, is on the order of microseconds [3]. Thus, the power shut-off control is accomplished by a long time interval [2][4].

For reducing the leakage power of CPU logic, we need a power gating technique for small CPU components so that the power for the unused component is quickly shut down based on execution instructions. Such a high-speed power gating technique for a small power domain is called a fine-grain dynamic PG [5]. Unlike traditional PGs, a fine-grain dynamic PG uses special standard cells, in which the virtual ground is separated from the real ground and a certain number of sleep transistors are inserted for quick power shut-down and wake-up. Since fine-grain dynamic PG requires energy overhead for power control and wake-up latency as well as area overhead, the circuit design and power shut-down and wake-up control need to be investigated.

Here, we propose an efficient method to apply fine-grain dynamic PG to the CPU with the cooperation of circuits, architecture and software. The largest part of CPU, the execution step, is divided into several components whose power is dynamically controlled according to the instructions being executed. The compiler added bit can also be used to control the shut-off. Since the decision to shut off power or not depends on the temperature, one of three modes can be selected by the software. A prototype chip called Geyser-0, a MIPS R3000 compatible processor, is designed and implemented based on the proposed technique with 90 nm CMOS technology. The evaluation results of the designed CPU show that about 47% leakage power can be reduced with the proposed technique.

II. FINE-GRAIN DYNAMIC POWER GATING

A. A fine-grain PG

In traditional PG techniques, a large semiconductor domain corresponding to the CPU itself or an entire IP is the target of the power shut-off. The VDD ring is provided around the target domain and is connected through sleep transistors with high Vth multi-threshold CMOS (MTCMOS) to the real VDD. By turning off the sleep transistors, the power for the entire domain is shut off. This mode is called the sleep mode, and the time from sleep mode to normal mode is called the wake-up time. Although this method has the benefit that the target domain can be designed with a common design method without power gating, the wake-up is on the order of microseconds. So, a simple sleep control method, which shuts off the power when the domain is not used for long time, has been used [2][3].

Unlike the traditional method, we use gates, each of which has its own virtual ground (VGND) line. VGND lines for several gates are connected through a single sleep transistor with a high threshold to the real GND, as shown in Fig. 1, to form a footer power switch.

As shown in Fig. 2, a certain number of gates are connected by the same VGND line and they shut off multiple sleep
transistors with a single sleep control signal. The VGND line is at a fixed position in the cell layout and locally shared by multiple cells. Since the existing ground rail in the standard cell is used as the real ground, non-power-gated cells such as flip-flops, clock buffers, repeaters, power switch drivers and isolation cells can be placed at any location in a row. Here, we refer to such a set of gates controlled with a single sleep control signal as a unit. A unit can be a computational module such as multipliers or adders; which are much smaller than the power domain in the traditional method. By providing an appropriate number of transistors, the wake-up can last merely a few nanoseconds. We call this technique as fine-grain PG.

The problem of fine-grain PG is providing a set of cells for power gating (PG cells) and establishing the design methodology to select the appropriate number of sleep transistors. First, we follow the conventional cell placement flow and then swap it with the same-sized PG cell. The PG cell is connected to a VGND pin instead of the real ground rail. After inserting properly sized power switches and power switch drivers, the signal wires and local VGND lines are routed. A research group of authors has established the design methodology [5][6], and now a real multiplier chip called Pinnacle, part of which sleeps depending on the multiplier instructions from the input data, is available.

By using fine-grain PG, a computational component can be shut off according to the execution instructions. For example, multipliers and dividers occupy a large area in the CPU layout, but are not used in every instruction. Such components can be woken up only when they are required by the fetched instruction. Here, such an aggressive dynamic control of PG is called the fine-grain dynamic PG.

However, such an aggressive control has a problem: that is, turning the sleep transistors on and off itself requires energy, and it takes a certain time to decrease the leakage power after turning components off. Thus, in order to reduce the leakage power, the time in sleep mode must be more than a certain break-even point (time). In other words, if the sleep time is less than the break-even point, the energy is increased by power gating instead of decreased. The main concern is how to control the sleep signals, which include temperature, so that the sleep time is less than the break-even point in a given circumstance.

B. Dynamic fine-grain power gating for a CPU pipeline

1) Target CPU and units: The selected target CPU of fine-grain dynamic PG is MIPS R3000 [7], a 32 bit RISC processor popularly used in embedded applications. It provides a standard five-stage pipeline structure consisting of Instruction Fetch (IF), Instruction Decode (ID), Execution (EX), Memory Access (MEM), and Write Back (WB). Here, we assume that the CPU is designed with a 90 nm CMOS process and works with a 200 MHz clock. As units of the fine-grain PG, computational units in the EX stage and co-processor 0 (CP0) are selected. The CP0 supports the operating system for processing the exception or the interrupt. So it only works in special conditions when the operating system is invoked. The CP0 is also selected because of its high possibility of sleep. Computational units occupy approximately 60% of the total CPU area, and it is easy to identify their usage based on the fetched instruction.

Here, we selected the following units as target units of dynamic fine-grain PG.

- **CLU (Common Arithmetic and Logic Unit)**
  General computational units including addition and subtraction. It can be in the sleep mode when branches, node occurrence probability (NOP) or memory access instructions without address calculation are fetched.
- **Shift unit**
  Since the barrel shifter occupies a considerable area but is not frequently used, it is selected as an individual unit.
- **Mult unit**
  The multiplier unit takes four clock cycles to complete 32-bit multiplication. If the upper 16 bits of either operand are all zeroes (0), only the upper part is in the sleep mode.
- **Div unit**
  The division unit takes 10 clock cycles to complete 32-bit division.
- **CP0**
  A coprocessor which handles exceptions and controls TLB entries is an interface with the operating system. It can be in sleep mode when the CPU runs in the user mode.

2) Fundamental control: Sleep signals for target units are generated in the sleep controller shown in Fig. 3.

![Fig. 3. Sleep control by the fetched instruction](image-url)
First, the fundamental operation for power gating is introduced.

All computational units except CLU are put in sleep mode automatically after finishing the operation in the unit. When an instruction is fetched in the IF stage, the sleep controller must check the fetched instruction and judge which unit is to be used. Since it takes a certain time to wake up (approximately 5 nsec is assumed here) the unit, the detection must be done in the IF stage when the processor works with a 200 MHz clock. For this purpose, a high-speed simple decoder, which only detects the unit to be used, is provided in the IF stage so that the unit is available when required.

In the IF stage so that the unit is available when required.

For this purpose, a high-speed simple decoder, which only detects the unit to be used, is provided in the IF stage so that the unit is available when required.

Figure 4 shows the flow of a high-speed detector for generating a wake-up signal from the MIPS instruction set. The detector checks the uppermost 6 bits of the code, and judges whether the instruction is a register-register operation (ROP). If so, the unit to be used is identified by the last 6 bits of the code. Otherwise, some immediate instructions and load/store instructions which use the CLU are detected.

Since the MIPS R3000 is an in-order processor, the pipeline is stalled during both an instruction and data cache miss-hit. Thus, all units including the CPU are in sleep mode when either cache miss-hit or a clock wake-up are detected before the cache becomes ready. The sleep controller also detects the stall from the multiplier or divider while waiting for data, and changes all units into the sleep mode.

The sleep control for CP0 operates in a completely different manner; that is, it is in the sleep mode when the CPU runs in user mode. The exception must be received in the sleep controller first; it sends CP0 the wake-up signal, and also sends a signal to the pipeline to stall 3 clock cycles. Since the content of registers in CP0 are lost by a power shutdown [8], registers which need to be saved are also implemented by the sleep controller.

C. Sophisticated power control methods

1) Instruction with PG direction: The fundamental method in the previous subsection can possibly increase the leakage power. For example, when multiple multiply operations are executed within an interval of few other instructions, the multiplication unit will be woken up as soon as it is shut off. In such cases, the sleep time is less than the break-even point, and the overhead power consumption increases the total power. Such a combination of instructions can be detected by the compiler, and we can save such overhead by keeping the power of the Mul unit after the first computation is finished. For this purpose, we introduce an instruction with PG (Power Gating) direction.

Figure 5 shows an example of instructions with PG direction. In the MIPS R3000 ISA, when the uppermost 6 bits (opcode) are all 0s, the operation is applied with the two registers shown in operand fields, and the type of operation is indicated with the last 6 bits. Instead of all 0s, we use "100111", which is not defined in the original ISA, as the uppermost 6 bits to indicate the PG direction. After executing such instructions, the functional unit is not shut off, but kept in active mode. The active mode remains until the next instruction whose function is the same but the opcode of all 0s is executed.

By replacing such instructions with instructions that degrade the leakage power, the overhead of power gating can be avoided.

2) PG policy register: The leakage power is sensitive to the temperature, that is, it increases drastically as the temperature rises. The break-even point is also influenced by the temperature, and tends to be short when the chip becomes hot. The power gating works most efficiently in such a situation. Conversely, if the chip is cool, a less aggressive power shut-off policy suppresses the overhead. Thus, the policy of power gating should be changed based on the temperature, which is measured by a thermal sensor.

Here, three options are available.

- Policy-1: The above fundamental policy, which shuts off every unit dynamically after using it or after a stall, including a cache miss-hit.
- Policy-2: Units are put in sleep mode only when a cache miss-hit is detected.
- Policy-3: Units are never put in sleep mode.

The PG policy register stores the above policies, and they can be written only in the kernel mode. The operating system controls the sleep mode depending on the data of the thermal sensor, as shown in Fig. 6. Note that a policy can be selected for each unit, because the break-even point is different for each unit. Two bits are needed to represent the policy for computational units, but only 1 bit indicates whether the sleep mode is used or is not needed for CP0.

III. PROTOTYPE CHIP GEYSER-0

In order to evaluate dynamic fine-grain power gating for the MIPS R3000 processor in a real chip, we designed a prototype chip called Geysers-0. A 2.5 mm × 5 mm die with 90 nm CMOS in six metal layers is used.

As shown in Fig. 7, Geysers-0 consists of two CPU cores: one is a core with the power gating mechanism (PG-CORE), and the other is a normal core without power gating for...
comparison (N-CORE). They share the instruction cache and data cache through TLB, which selectively connects them. The part enclosed in the dotted frame in Fig. 7 is the target of the proposed power gating. PG-CORE, N-CORE, the shared caches and TLB have their own I/O pins for the power supply, so that the input current can be measured independently.

Geyser-0 has an L1 instruction/data cache, each of which is 8 KB, and a 2-way set associative write-back cache with a 64 B block size. The TLB has a full associative address translation table with 16 entries. Figure 8 shows the flow of memory access in Geyser-0. The upper 20 bits of the virtual address are used as a virtual page number which is translated into a physical address by table reference, while the remaining 12 bits; offset in a page, are used as an index of the cache. Thus, referencing the TLB and cache directories is done simultaneously.

Although the leakage power in the cache and TLB is a substantial problem in a whole computer system, in order to concentrate the power gating for the CPU itself, no leakage power reduction techniques are applied to the memory system of Geyser-0. We will investigate power gating of a whole system as the next step of this study.

Since the I/O pins of Geyser-0 are strictly limited in, the address and data for the outside memory module is multiplexed. The memory interface logic is provided as an interface for the outside memory. For debugging, both the PG-CORE and N-CORE can access the outside memory by multiplexed. The memory interface logic is provided as an interface for the outside memory. For debugging, both the PG-CORE and N-CORE can access the outside memory by directly bypassing the TLB and cache system.

A. Design flow

Geyser-0 is described in Verilog XL, synthesized with Synopsys Design Compiler, and the layout is generated by using Synopsys Astro.

For fine-grain power gating, a limited set of standard cells supported by the Semiconductor Technology Academic Research Center (STARC) were modified manually to separate the VGNDs. The PG cells were replaced with standard cells after the layout. In this implementation, we could not modify all standard cells into PG cells because of the limitation of design time. Since some compound gates in the standard cell library are not included in the PG cells, the area increased by 17% compared to the case when the standard cells are fully used. However, this condition can be improved with some effort.

The optimum number of sleep transistors are inserted in the post layout netlist by the Sequence Design tool called Cool Power. The condition of optimization is as follows. When all gates which share a MTCMOS as a sleep transistor work together in active mode, the maximum level of VGND is less than 10% of VDD. The optimized netlist is read by Astro to generate the final GDS file for mask patterns.

IV. Method of Power Analysis

A. Dynamic power and leakage power in the active mode

Geyser-0 is now under fabrication, and real chips are not available yet. Here, evaluation results based on the real layout are shown. The dynamic power and leakage power in the active mode are evaluated with a standard method. That is, the switching probability is measured with a post-layout simulation, and the Spatial Archive Interchange Format (SAIF) file is generated. Synopsys’s PowerCompiler is used to compute the power consumption with the post-layout netlist, including the information on buffers, wires and capacitors. Here, the leakage power in the active mode is referred to as \( L_{wa} \).

B. Leakage power in the sleep mode

Unlike the active mode, the power consumption in the sleep mode is complicated since it includes the transient power for shut-off, stable leakage power when the sleep transistor is turned off, and the power for wake-up. These are all different in each unit and so the statistic information that “N cycles sleep happens M times” is needed for all Ns when an application program is executed. This information can be obtained from a register transfer level (RTL) simulation using real application programs.

Then, the power model for each unit is generated by measuring the leakage current in sleep mode. The high-speed circuit level simulator Hsim is used for the measurement. The average power consumption when a unit is in sleep mode in \( N \) cycles and then wakes up (\( L_{wsy} \)) is computed from the power model. Since it takes a large computation time for making a complete table for every \( N \), we measured cases of 2, 4, 6, 8, … 36, 38, 40, 50, 60, 80, 90, 100, 300, 500, 750, and 1000 cycles assuming a 200 MHz clock. The current for other cycles less than 1000 is computed by linear interpolation. The case of more than 1000 cycles, that is, more than 5 msec sleep, can be considered as a stable state. In other words, the data for 1000 cycles is used for such a case.

The average leakage power \( L_w \) in a unit is computed with the following expression.

\[
L_w = \sum_i (L_{wsy} \cdot \frac{M_i \cdot i}{T}) + L_{wa} \cdot (1 - \frac{\sum_i (M_i \cdot i)}{T})
\]

where it is assumed that \( i \) cycles of sleep occur \( M(i) \) times in the program, whose total number of execution clock cycles is \( T \).
C. Benchmark programs

As benchmark programs we used the following three programs from the MiBench suite [9]: Quick Sort from the mathematics package, Dijkstra from the network package, and Blowfish from the security package. As an example of media processing, a JPEG encoder program is also used.

All the programs are written in C language and compiled to MIPS code by using gcc (2.95.29) in Linux. In order to run the programs in the RTL simulation, the object code is inverse-assembled by objdump (2.16.1) in the gcc binary utility and translated into the data file for the Verilog-XL test bench.

V. Evaluations

A. Break-even point analysis

Figure 9 shows the leakage power when the CLU is shut off at 65°C.

The horizontal axis is the time from when the power is shut off, the left vertical axis is the power (µW), and the right vertical axis is the thin line representing the total energy consumption (fJ). Here, the power includes the power consumed in the sleep transistors. Immediately after the power is shut off, the power is increased sharply by the dynamic power needed in the sleep transistor. Then, it decreases rapidly, but the reduction ratio is decreased gradually. Finally, it is close to the stable power in the sleep mode. The tendency is the same in all other units, although the absolute values are different. Here, the crossing point of the total energy consumed in shutdown mode and that in active mode is the break-even point. Table I shows the measured break-even point of each unit. A clock cycle corresponding to 5 nsec of the 200 MHz clock is assumed.

<table>
<thead>
<tr>
<th>Temperature (°C)</th>
<th>CLU</th>
<th>Shift</th>
<th>Mult</th>
<th>Div</th>
<th>CP0</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>74</td>
<td>114</td>
<td>74</td>
<td>44</td>
<td>92</td>
</tr>
<tr>
<td>65</td>
<td>26</td>
<td>38</td>
<td>22</td>
<td>14</td>
<td>28</td>
</tr>
<tr>
<td>100</td>
<td>12</td>
<td>16</td>
<td>10</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>125</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>8</td>
</tr>
</tbody>
</table>

Since the leakage power in active mode is increased by high temperature, the break-even point becomes short when the chip is hot. Note that the break-even point of a large unit is not always long. For example, the small CLU has a longer break-even point than Mult. This is because the number and position of sleep transistors are different for each unit. Thus, the power model for each unit must be based on the circuit level simulation, and a simple mathematical model is difficult to apply.

B. Utilization ratio of each unit

The probability of power shut-off depends on the utilization of computational units. Figure 10 shows the utilization of four target computational units.

Both CLU and shift are frequently used, especially in Blowfish for code encryption. In contrast, utilization of Mult and Div is low in all programs. The JPEG encoder uses the Mult unit for DCT (Discrete Cosine Transform) calculation, and in QSORT, both the Mult unit and Div unit are only slightly used (0.2%). However, Dijkstra and Blowfish never use the Mult or Div units. Thus, the units that are in sleep mode depend on the application programs.

C. Power-reduction efficiency

The leakage power can be evaluated by using the power consumption in both models and the utilization ratio. The power model introduced in the previous section is used for calculation.

Figure 11 shows the leakage power of each application. For comparison, the leakage power without the power gating at 25°C is also shown in the graph. Here, the instructions with PG directions are not used, and the PG policy is fixed as Policy-1, the fundamental method.

In the computational units, 76% leakage power is reduced. In CP0, the leakage power in the active mode is 16.7 µW whereas in the sleep mode it is 6.4 µW. The CLU and SHIFT units use Policy-2, and the Mult unit and Div unit use Policy-1 in this evaluation. But, if the OS knows the chip temperatures is 25°C and sets the PG policy register, the CLU and SHIFT units must be set for Policy-3. By applying these policies, 1.6% (62.2 to 61.2 µW) of the total power is reduced at 25°C. In contrast, for 100°C, CLU and SHIFT unit must be set for Policy-2, and the Mult unit and Div unit for Policy-1. Compared with the policies of the optimized 25°C, 1.5% (239.7 to 236.1 µW) of the total power is reduced.

The total leakage power in the processor core including the register files and instruction decoder is 439 µW without
power gating, but it can be reduced to 232 μW by using the fine-grain dynamic power gating. That is, 47% of the leakage power is reduced. This comes from the utilization ratio of the Mult unit and Div unit, both of which occupy a large area and can be in sleep mode for a long time.

However, in the 90 nm process, the leakage power itself is still not dominant in the total power consumption. That is, 2.8 mW dynamic power is consumed in the total core. However, the leakage power becomes dominant in the advanced process beyond 45 nm, and the proposed techniques will be useful in the future.

D. Area overhead

Figure 12 shows the area of each unit in the PG-CORE with the floor planning.

It appears that the Mult unit and Div unit occupy 40% of the total core area. Table II shows the areas of PG-CORE and N-CORE with the overhead of power gating. The area overhead of PG-CORE compared with N-CORE is mainly caused by the sleeping transistors and the wire congestion for keeping the VGND lines. MAIN also includes the area for the sleep controller and surrounding logic. The overhead of “MAIN Control” is by the Sleep Controller. It seems unusual that there is no overhead in CP0, but the area utilization and wire density of CP0 are originally low. In such a case, there is enough space for inserting sleep transistors without increasing the area. Since the characteristics of the layout influences the overhead, the ratio varies widely. In total, 41% overhead is required for the proposed fine-grain dynamic power gating. However, the overhead can be improved by optimizations.

VI. Conclusions

Fine-grain dynamic power gating is applied to the MIPS R3000, and the prototype chip Geyser-0 is designed and implemented with a 90 nm CMOS process. Evaluation results show that leakage power can be reduced by 47% with 41% area overhead. Although the dynamic power is dominant in the current 90 nm process, this technology is useful for future processes in which the ratio of leakage power is increased.

Because Geyser-0 is the first prototype chip, much future work is needed for improvements. The area overhead can be further reduced by the improvement of circuits and design technologies. A compiler which can handle instructions with PG direction will be developed based on the evaluation results. Fine-grain power gating can be applied to the register [8] files, cache controller and a part of TLB. Furthermore, the effect of fine-grain dynamic power gating for super-scalar, VLIW and multi-core processors should be examined.

Acknowledgments

The authors thank VLSI Design and Education Center(VDEC), Synopsys, Cadence, Mentor, Sequence Design, STARC. This work is supported by Japan Science and Technology Agency (JST) CREST.

References


Table II

<table>
<thead>
<tr>
<th>PG-CORE</th>
<th>N-CORE</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>161866</td>
<td>118585</td>
</tr>
<tr>
<td>CLU</td>
<td>7764</td>
<td>4970</td>
</tr>
<tr>
<td>SHIFT</td>
<td>58516</td>
<td>10663</td>
</tr>
<tr>
<td>DIV</td>
<td>54863</td>
<td>32498</td>
</tr>
<tr>
<td>CP0</td>
<td>22861</td>
<td>22861</td>
</tr>
<tr>
<td>TOTAL</td>
<td>31270</td>
<td>222097</td>
</tr>
</tbody>
</table>