Vol.2010-ARC-187 No.1
Vol.2010-EMB-15 No.1
2010/1/28

細粒度パワーゲーティングを適用した汎用マイクロプロセッサ Geyser-1

池淵 大輔[†1] 関 直臣[†1] 小島 悠[†1]
釜田 雅大[†1] Lei Zhao[†1] 天野 英晴[†1]
白井 利明[†2] 小山 慧[†2] 橋田 達徳[†2]
馬橋 雄祐[†2] 増田 大樹[†2] 宇佐美 公良[†2]
武田 清大[†4] 並木 美太郎[†3]
近藤 正章[†5] 中村 宏[†4]

我々は、時間的にも空間的にも細粒度にパワーゲーティングを行うことでリーク電力を削減する手法を提案し、この手法を適用した MIPS R3000 互換のマイクロプロセッサ Geyser-1 を設計・試作した。本稿ではその設計手法、実装方式、ならびに実チップを用いた評価結果を報告する。

A general purpose microprocessor with fine-grained power gating

Daisuke Ikebuchi,[†1] Naomi Seki,[†1] Yu Kojima,[†1]
Masahiro Kamata,[†1] Lei Zhao,[†1] Hideharu Amano,[†1]
Toshiaki Shirai,[†2] Satoshi Koyama,[†2]
Tatsunori Hashida,[†2] Yusuke Umahashi,[†2]
Daiki Masuda,[†2] Kimiyoshi Usami,[†2] Seidai Takeda,[†4]
Mitaro Namiki,[†3] Masaaki Kondo[†5]
and Hiroshi Nakamura[†4]

We propose a technique of runtime power gating which enables fine-grained power supply control both temporally and spatially. We have applied this technique to MIPS R3000 architecture and developed a prototype processor called Geyser-1. In this paper, we show its design methodology, implementation, and evaluation results.

1. はじめに

LSI チップの漏れ電流は、90nm 以降、プロセスの進展に伴って急速に増大し、この削減手法は重要な問題となっている。汎用 CPU の設計においても、動的電力の削減手法が既に十分研究されて実用化されていることもあり、漏れ電力の削減手法の研究は重要な課題となっている。従来、チップ面積の大きな割り合いを占めるキャッシュの漏れ電流を削減する手法は広く研究され、様々な手法が提案されている。[1)] 一方でプロセッサコア部の漏れ電流削減手法はあまり検討されていない。漏れ電流を減らすためには、スレッショルドレベルの高いトランジスタを用いれば良いが、このようなトランジスタは動作速度が遅いため、性能が重要視されるプロセッサコア部に適用するのは難しい。

性能低下を伴わず、漏れ電流を削減する手法にパワーゲーティングがある。この手法では、漏れ電流の少ない高スレッショルドのトランジスタをパワースイッチとして利用し、利用されていない部分の電源を切り離してスリープ状態にすることにより漏れ電流を削減する。対象回路を通常のスレッショルドレベルの高速トランジスタで構成できるため、速度の低下はほとんどない。しかし、従来のパワーゲーティング手法では、対象回路をスリープ状態から動作状態に復帰（ウェイクアップ）させるのに長い時間を要していた。このため、長い期間スリープ状態を保つことが期待される領域、例えばアクセラレータ、I/O モジュール、メモリなど空間的にも大きい領域が対象であり、定常的に動作するプロセッサコア部にパワーゲーティング手法を適用することは困難であった。しかし、プロセッサコア部においても、演算ユニットの稼働率はそれほど高いわけではない。高速性が要求されるプロセッサコア部ではリーク電流の大きい高速トランジスタが用いられているため、これらの部分にパワーゲーティング手法を適用できればその効果は大きいと考えられる。

この点に着目し、我々は JST-CREST 研究課題「革新的電源制御による次世代超低電力高性能システム LSI の研究」において、高速にスリープおよび回復操作を行うことのできるパワーゲーティング手法の研究を進めている[2)]。時間的に細粒度にパワーゲーティングす

†1 慶應義塾大学
Faculty of Science and Technology, Keio University
†2 芝浦工業大学
Shibaura Institute of Technology
†3 東京農工大学
Tokyo University of Agriculture and Technology
†4 東京大学大学院
Graduate School of Information Science and Technology, The University of Tokyo
†5 電気通信大学
Tokyo University of Electro-Communications

Vol.2010-ARC-187 No.1
Vol.2010-EMB-15 No.1
2010/1/28

ることが可能となれば、空間的にも小さい領域をスリープ状態にできる機会が大きくなる。我々はこの時間的にも空間的にも粒度の小さい細粒度パワーゲーティング（細粒度 PG）の設計手法の確立を目指しており、既に乗算器に適応することに成功している[3]。さらにこの手法をプロセッサコア部に適用し、アーキテクチャ、コンパイラおよび OS レベルでパワーゲーティングを制御することで漏れ電流の削減を目指す汎用マイクロプロセッサ Geyser の実現を試みている[4]。Geyser では、命令フェッチ時に命令をプリデコードすることで必要な演算ユニットのみをウェイクアップし、演算の実行後は速やかにスリープ状態に戻す。パワースイッチを用いたパワーゲーティングでは、スリープとウェイクアップに電力消費が伴うため、スリープとウェイクアップを頻繁に繰り返すとむしろ電力を増加させてしまう。このため、演算ユニットの使用頻度をコンパイラが判断し、スリープすべきか否かを指定できる特殊ビットを持つ演算命令を用意している。また、スリープすべきか否かは漏れ電流の値に依存し、その値は実行状況で変化する温度に強く依存する。そのため、OS が温度などの実行状況を把握することで、状況に応じてスリープポリシーを切り替える機能も用意している。このように、システム設計階層間の協調に基づいてパワーゲーティング方式を制御することでリーク電力を抑え込むことを目指している。

これまで細粒度 PG を汎用 CPU に適用した例はなく、その効果を実証するためにはスリープ動作時における漏れ電流やウェイクアップ時間などを実チップにより測定する必要がある。そこで我々は ASPLA 90nm プロセスを利用し、プロトタイプチップ Geyser-0[5] を開発した。Geyser-0 は、パワーゲーティングを施した MIPS R3000 コアと通常の R3000 コアがキャッシュ、TLB を共有する構成を持ち、OS の動作の元で細粒度パワーゲーティングの効果を測定することができるようになっていた。しかし、Geyser-0 は構成上、電源を三系統にしたため給電系が複雑になり、これが原因で生じたレイアウト上の問題点により動作せず、目的を果たすことができなかった。

そこで、Geyser-0 の構成のうち、細粒度 PG を施した MIPS R3000 CPU コアのみを実装し、細粒度 PG の効果の測定に目的を絞ったチップ Geyser-1 を実装した。本報告では、Geyser-1 の設計方針と実装、ならびに実チップでの測定結果を述べる。

2. Geyser-1 の設計方針

2.1 細粒度パワーゲーティングの実現手法

従来のパワーゲーティングはコアやモジュールなどチップ上の大きな面積を単位としていた。すなわち、通常のゲートを用いて設計したコアまたはモジュールの周囲に仮想 VDD 用の電源リングを設け、VDD との間に電源遮断用のスレッショルドレベルの高いスリープトランジスタを置いて電源の遮断を行っていた[6][7]。この方法はパワーゲーティング用の特殊なゲートを備える必要はないが、大きな単位で電源をオン／オフするため、電源がオフであるスリープモードと電源がオンの動作モード間の遷移にはマイクロ秒単位の時間を要した。

これに対して、我々は図 1 に示すように、それぞれのゲート毎に仮想グラウンドを設けてこれらを一定数接続し、グラウンドとの間にスリープトランジスタを設ける。この方法は仮想グランドを持ったパワーゲーティング用のゲートを必要とするが、スリープトランジスタ数を調整することで、スリープモードに遷移するシャットダウン時間と動作モードに遷移するウェイクアップ時間をナノ秒単位に短縮することができる。この技術を用いれば、演算器レベルの細かい単位で、パワーゲーティングが可能になる。我々はこの技術を今までのパワーゲーティングと区別して細粒度パワーゲーティング (細粒度 PG) と呼ぶ。

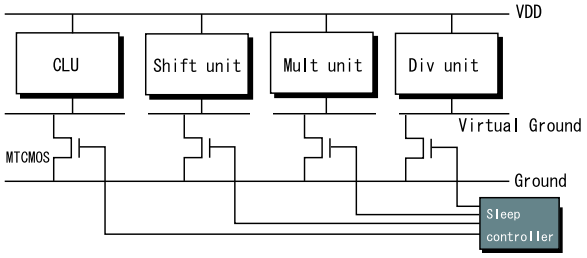

図 1 パワーゲーティング回路の概要

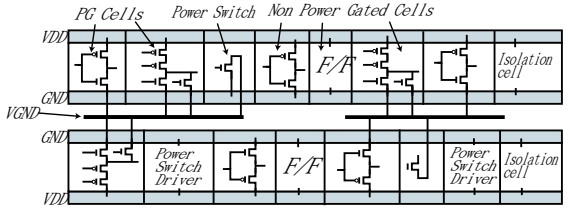

図 2 細粒度 PG の仮想グランドアーキテクチャ

図 2 は細粒度 PG における実際のセル配置を示している。従来のパワーゲーティングとは異なり、一つの仮想グランドラインを数個程度のセルが共有している。一つグラウンドラインとスリープトランジスタの集合で一つのユニットが構成される。ユニットは、単一のスリープ制御信号でスリープモードと動作モードの切り替えが制御される。高速なモード切り

Vol.2010-ARC-187 No.1
Vol.2010-EMB-15 No.1
2010/1/28

替えのためにはスリープトランジスタの数と配置を適切に行う必要があるが、これについては Pinnacle[3] および Geyser-0[5] の設計、実装により、その回路上の設計手法がほぼ確立している。

2.2 スリープ対象ユニットとスリープ制御

対象とするプロセッサとして、標準的な 5 段パイプライン構成 (命令フェッチ:IF、命令デコード:ID、実行:EX、書き戻し: WB) を持つ RISC 型の 32 ビット CPU MIPS R3000 を選択した。これは、組み込み分野での実績があること、および、構造が単純でありパワーゲーティング手法の効果を検証しやすいためである。

Geyser-1 では、この CPU コアの実行 (EX) ステージの演算ユニットを対象として、細粒度パワーゲーティングを実装した。実行ステージの演算ユニットは、予備評価によると CPU コアの 6 割の面積を占め、命令によって利用されるかどうかが判別可能である[8]。EX ステージでは、シフタ、乗算器、除算器、それ以外の命令を実行する CLU(Common arithmetic Logic Unit) をそれぞれ細粒度動的パワーゲーティングの対象ユニットとした。

CPU コアの他の部分、例えば命令デコード (ID) ステージのレジスタファイルなどもパワーゲーティングが有効となる可能性はあるが、動作頻度が高い点、およびデータ保持が必要である点から実装されていない。命令フェッチ (IF) ステージはキャッシュミス以外は、毎クロック動作する点、書き戻し (WB) ステージは、面積がほとんどない点からパワーゲーティングの対象とはしていない。

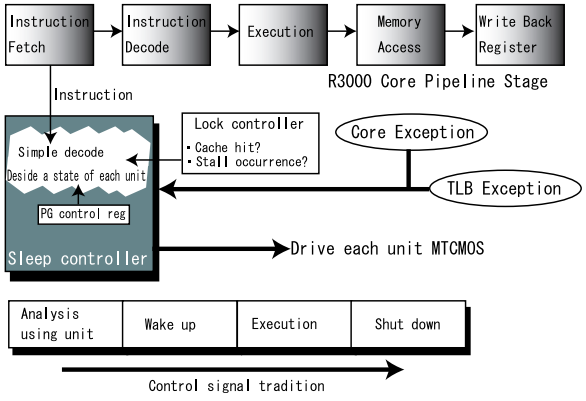

図 3 実行時スリープ制御

パワーゲーティングの制御は、スリープコントローラからの信号によって行われる。図 3 にスリープコントローラとパイプラインの関係を示す。各演算ユニットは演算完了後、すぐに自動的にスリープモードに遷移する。命令フェッチが IF ステージで行われると、スリープコントローラはこの命令を先見してどのユニットが使われるかを判断する。これは、ID ステージで命令を判定するとウェイクアップが間に合わず、EX ステージで演算の開始が間に合わない可能性があるためである。スリープコントローラではこのための専用の簡易命令デコーダを持っており、高速に次の命令で使うユニットを特定して、ウェイクアップ信号を送る。このことにより、ID ステージを命令が通過している間で、各ユニットはウェイクアップを行うことができる。

2.3 ブレイクイーブンポイントを考慮したスリープ制御

パワーゲーティングにおいては、モード遷移時に一定のエネルギーを消費する。また、スリープモードに入ってもリーク電流を遮断するまでには一定の時間がかかる。このため、モード遷移の頻度が大きいと、逆にエネルギーロスによって消費電力が大きくなってしまう可能性がある。ここで、スリープモードに遷移してから、エネルギーロスよりも削減したエネルギーの方が大きくなるまでの時間を、ブレイクイーブンタイム (BET: Break-Even Time) と呼ぶ。BET はパワーゲーティングの効果としての損益分岐点ともいうことができ、この BET より長くスリープ状態を継続できれば消費電力を削減することができる。

演算器の中で利用頻度が高いものに対して前節のスリープ制御をそのまま適用すると、スリープ状態は BET より短くなり、モード遷移が頻発してエネルギーロスを大きくしてしまう。また、BET は温度が高いほど短くなることが分かっている。これは、高温下ではスリープ時に対してアクティブ時のリーク電流の比率が大きくなるためである。したがって BET の値は実行状況に依存する。

このように、理想的なスリープ制御は実行プログラムの特徴や実行時の状況に応じて変わるため、前節で述べたハードウェアによる一律なスリープ制御は最善ではない。そこで、実行プログラムの特徴や実行時の状況に応じてスリープ制御を変えるために、PG 制御情報付き命令と、PG 制御レジスタの二つの機構を採用する。前者は実行プログラムの特徴に応じてコンパイラがスリープ制御するためものであり、後者は OS が実行時の状況に応じてスリープ制御するためのものである。

2.3.1 PG 制御情報付き命令

各ユニットの実際の利用状況は、正確には動作時にならないと分からないが、コンパイル時にもある程度の予測が可能である。そこで、エネルギーロスが大きいことがコンパイル時に明らかな場合にスリープモードへの移行を抑制するために、PG 制御情報を命令に付加する。図 4 は Geyser-1 用の PG 制御情報付き命令の構造を示している。

R3000 の ISA では、レジスタ演算命令の上位 6 ビット (ROP ビット) は 0 であり、下位 6 ビットで演算の種類を表す。この ROP ビットを 100111(2) に設定すれば利用した演算ユ

Vol.2010-ARC-187 No.1
Vol.2010-EMB-15 No.1
2010/1/28

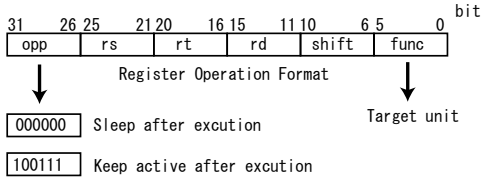

図 4 PG 制御命令

ニットの演算後に自動的にスリープする制御をキャンセルすることにした。例えば、シフト演算を行う命令で ROP ビットが 100111(2) であれば、演算終了後 Shift unit をスリープさせず動作モードのままとし、この動作モードは、次のシフト命令が実行されるまで継続される。

この命令拡張により、スリープさせるとエネルギーロスとなる命令をコンパイル時に発見しこの ROP ビットの付加情報を利用することで、該当命令のスリープをキャンセルしエネルギーロスを小さくすることができる。

2.3.2 PG 制御レジスタ

BET の温度依存性に対処するために、使わないユニットを動的にスリープ制御するか、あるいはスリープ制御を全く行わないかを OS が選択できる機能を持たせる。このために PG 制御レジスタを CP0 内に設け、コプロセッサ命令によりその値を設定できるようにした。また、BET に大きな影響を与えるリーク電流の値を、動作時に直接測定するためにリークモニタ[9] を実装する。これにより、OS は動作時の状況によりスリープ制御を切り替えることが可能となる。

3. Geyser-1 の実装

Geyser-1 は、細粒度 PG の効果を実チップで示すことに目的を絞り、TLB とキャッシュを装備しない CPU コアのみとした。また、最終段階で人手によるレイアウトをしなくても良いようにデザインフローを見直し、配置配線についても周波数方向に最適化せず、まず動作するチップを作ることを主目的とした。

このため動作周波数は 60MHz にとどまった。これは、キャッシュを外部に装備する構成にしたため、I/O を介したアクセス遅延が増大したこと、利用した富士通 e-shuttle 65nm プロセスが高速性よりも低電力性に特化したものであったことが主な原因である。

Geyser-1 は Verilog XL で記述し、合成には Synopsys　Design Compiler 2007.03-SP4 を、レイアウトには Synopsys Astro 2007.03-SP7 を用いた。セルライブラリは、細粒度パワーゲーティングを用いない部分については CS202SZ 65nm 12-metal-layer CMOS を用い、コア部分の電源電圧は 1.2V、I/O は 3.3V である。細粒度 PG 用には VGND を分離した独自のセルライブラリ (PG セル) を作成し、細粒度パワーゲーティングを適用する部分は、配置配線の段階でスタンダードセルをこのセルと入れ替えた。本来はライブラリ中のセルの全てに対応する PG セルを作るのが理想だが、今回は設計時間の関係で、一部のセルのみを作成した。レイアウト後のネットリストに対してスリープトランジスタを挿入する必要があるが、これはシーケンスデザイン社[⋆1]の Cool Power 2007.3.8.5. を用いた。スリープトランジスタ挿入後のネットリストを再び Astro で読み込み最終レイアウトを生成した。

表 1 細粒度 PG の面積オーバーヘッド

	Total(μm^2)	PS(μm^2)	ISO(μm^2)	Overhead
CLU	3594.8	296.4	79.2	10.8%
SHIFT	3294.8	298.8	76.8	10.5%
MULT	27782.8	1762.0	153.6	7.3%
DIV	28263.6	1301.2	153.6	5.4%
others	66045.6	-	-	-

表 1 は、細粒度 PG の対象モジュールにおける面積オーバーヘッドを示す。others は細粒度 PG の対象外の部分を示す。PS はパワースイッチの面積を示し、ISO は、アイソレーションセルと呼ぶ出力用のセルで、スリープ状態になったモジュールから電気的に浮いたレベルが出力されないようにするために用いられる。今回の実装では全体の面積の 43% が乗算器と除算器で占められており、この部分が細粒度 PG の対象となりリーク電力が削減されるのは大きい。スリープトランジスタとアイソレーションセルのオーバーヘッドは 5.4% - 10.8%であり、さほど大きくないことがわかる。図 5 に Geyser-1 のレイアウトを示す。チップサイズは 2.1mm ×4.2mm である。黒い四角形が細粒度 PG の対象モジュールで、中心部に離して配置した。周辺の小さな 4 つの四角形は、動作時のリーク電流値を測定するリークモニタ[9] である。

4. 実チップの評価

実チップの評価用に開発した Virtex-4/LX FPGA ボードにチップを搭載したマザーボードを接続し、評価を行った。外部キャッシュに代わって FPGA 上の BRAM 上に命令とデータを置いてテストプログラムが実行可能な環境を構築した。まず、簡単なプログラムで動作周波数を測定し、細粒度 PG が 60MHz クロックで正常に動作することを確認した。

次に、スリープ時間を変化させながら乗算器の電力を測定した。この結果を図 6 に示す。動作周波数は 50MHz、温度は 25°C とした。グラフより、この場合ブレークイーブンポイントは 44 クロックサイクルであることが判る。すなわち乗算命令の間隔がこれを下回る場

⋆1 2009 年にアパッチ社に吸収された

Vol.2010-ARC-187 No.1
Vol.2010-EMB-15 No.1
2010/1/28

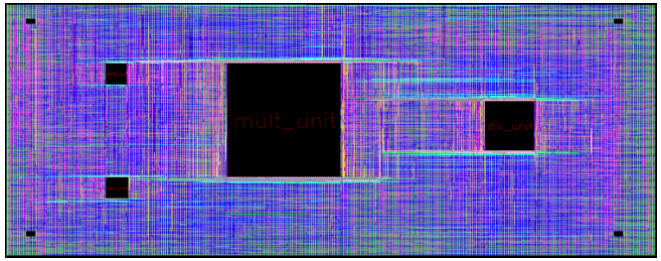

図 5 Geyser-1 のレイアウト

合は、スリープさせない方が有利である。

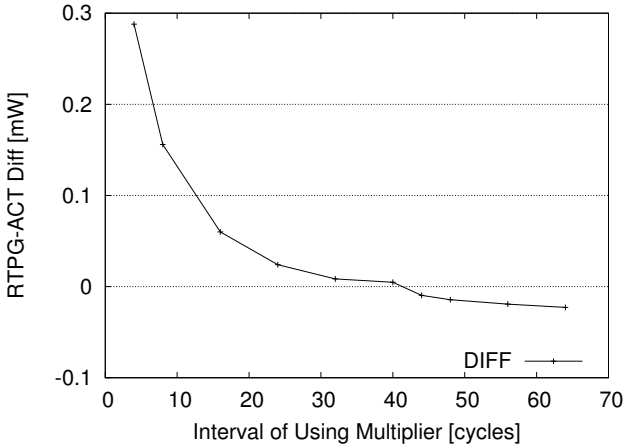

図 6 スリープ時間と消費電力 (乗算器)

4.1 細粒度 PG の効果

細粒度 PG モジュールを全てスリープさせた場合と、アクティブ状態にした場合の漏れ電流を測定した。ダイナミック電流をゼロとするために、クロックは停止した。この結果を図 7 に示す。細粒度 PG による漏れ電流の削減は 25°C では 7%に留まるが、温度が上がると 20%程度まで増大する。

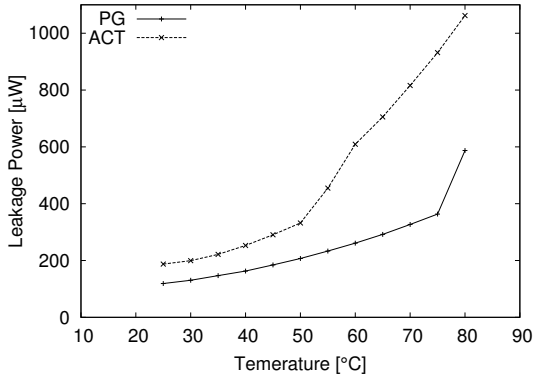

図 7 漏れ電力の削減

4.2 ベンチマークプログラムを用いた評価

組み込みプログラムのベンチマーク集 MiBench[10] の mathematics package より Quick Sort (QSORT)、network package より Dijkstra、またメディア処理を代表して、JPEG エンコーダの一部である 2 次元 DCT (Discrete Cosine Transform) のプログラムを Geyser-1 上で実行させ、消費電力を測定した。実用的なプログラムを動かす場合、FPGA 内の Block RAM からの遅延の影響で動作周波数が制限される。このため、10MHz のクロック周波数で測定を行った。

図 8、 9、 10 に、この 3 つのベンチマークプログラムの消費電力を示す。細粒度 PG の効果は乗算と除算を利用しない Dijkstra がもっとも大きく、全電力の 8%から 24% に当たる。この値は図 7 の値よりも大きいが、これは、細粒度 PG により動的電力も削減されているためである。Geyser-1 ではオペランドアイソレーションの実装が完全ではないため、演算器の入力変化により消費電力が増大する部分が一部ある。QSORT は、乗算器と除算器を利用するが頻度が高くないため、4%から 29%程度の削減が可能である。DCT は乗算器を頻繁に利用するため、3%から 17%の効果に留まっている。

上記の結果により細粒度 PG が CPU の消費電力を削減するのに効果があることがわかった。

5. まとめと今後の課題

我々は、細粒度 PG を適用した MIPS R3000 互換の汎用マイクロプロセッサ Geyser-1 を試作し、最大 60MHz で動作すること、および、リーク電力を効果的に削減できること

Vol.2010-ARC-187 No.1
Vol.2010-EMB-15 No.1
2010/1/28

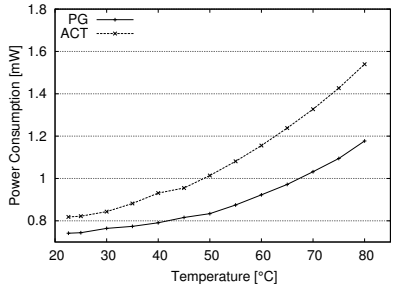

図 8 Dijkstra の消費電力

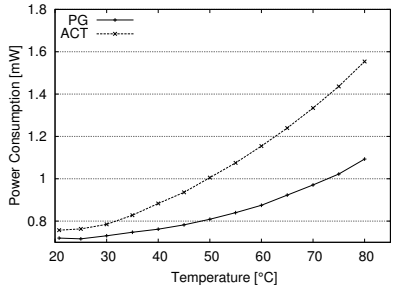

図 9 QSORT の消費電力

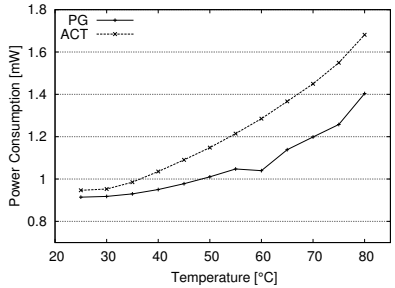

図 10 DCT の消費電力

を、実チップを用いて実証することに成功した。

Geyser-1 の動作周波数が低い主な原因は、キャッシュを外付けにしたためである。そこで、キャッシュと TLB を内蔵し、120MHz での動作と OS の稼働を目標とした Geyser-2 を 2009 年 11 月にテープアウトし、現在製造中である。Geyser-2 を用いて OS、コンパイラを含むリーク電力の削減を試みる予定である。

謝 辞 本研究は科学技術振興機構 (JST) の戦略的創造研究推進事業 (CREST) における研究領域「情報システムの超低消費電力化を目指した技術革新と統合化技術」の研究課題「革新的電源制御による次世代超低電力高性能システム LSI の研究」による．本研究は東京大学大規模集積システム設計教育研究センターを通し，シノプシス株式会社・日本ケイデンス株式会社の協力で行われた．

参 考 文 献

1) Weiping Liao, Basile J.M., L.H.: Microarchitecture-level leakage reduction with data retention, *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, Vol.13, pp. 1324–1328 (2005).
2) 中村宏，天野英晴，宇佐美公良，並木美太郎， 今井雅，近藤正章：革新的電源制御による超低消費電力高性能システム LSI の構想，情報処理学会研究報，Vol.ARC-173(14), pp.79–84 (2007).
3) 香嶋俊裕，武田清大，白井利明，大久保直昭，宇佐美公良：走行時パワーゲーティングを適用した低消費電力乗算器のアーキテクチャ設計， *VLD*, Vol.73, pp.7–12 (2006).
4) N.Seki and et.al.: A Fine Grain Dynamic Sleep Control Scheme in MIPS R3000, *ICCD* (Oct. 2008).
5) 関直臣他：MIPS R3000 プロセッサにおける細粒度動的スリープ制御の実装と評価，先進的計算基盤システムシンポジウム SACSIS2008 予稿集，pp.65–80 (2008).
6) M.Ishikawa and et. al.: A 4500 MIPS/W, 86μA Resume-Standby, 11μA Ultra-Standby Application Processor for 3G Cellular Phones, *IEICE Trans. on Electronics*, Vol.E88-C No.4, pp.528–535 (2005).
7) Y.Kanno: Hierarchical Power Distribution with 20 Power Domains in 90-nm Low-Power Multi-CPU Processor, *ISSCC2006*, pp.540–541 (2006).
8) 関直臣他：MIPS R3000 プロセッサにおける細粒度動的スリープ方式の提案，情報処理学会研究報，Vol.ARC-173(9), pp.49–54 (2007).
9) 小山慧，武田清大，宇佐美公良：MTCMOS 回路を利用したオンチップ・リークモニタの設計と評価，信学技報， Vol.VLD2007-146, pp.13–18 (2008).
10) M.R.Guthaus, J.S.Ringenberg and et. al.: MiBench: A free, commercially representative embedded benchmark suite, *Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop on*, pp.3–14 (2001).