Vol.2009-ARC-183 No.34
Vol.2009-OS-111 No.34
2009/4/24

省電力MIPSプロセッサコア評価のための計算機システムのFPGAによる試作

木村一樹[†1] 砂田徹也[†1] 長井智英[†2]
関直臣[†3] 近藤正章[†4] 天野英晴[†3]
宇佐美公良[†5] 中村宏[†2] 並木美太郎[†1]

本研究では，演算ユニットごとに動的なパワーゲーティング技術を施したMIPS R3000ベースのプロセッサコア，Geyser-0について，そのOS開発プラットフォームを実現するため，FPGA上でメモリコントローラと各種の入出力装置を有する計算機システムを試作した．また処理性能及びパワーゲーティングによる電力削減効果の評価を行うため，専用のパフォーマンスカウンタを設計した．これを用いた評価の結果，試作した計算機システムはシミュレーションの約400倍の実行速度を達成し，スリープ時の演算ユニットの消費電力を誤差率約20[%]の精度で推算した．

Prototyping of a Computer System to Evaluate Power-saving MIPS Processor Core Using FPGA

Kazuki Kimura,[†1] Tetsuya Sunata,[†1]
Tomohide Nagai,[†2] Naomi Seki,[†3] Masaaki Kondo,[†4]
Hideharu Amano,[†3] Kimiyoshi Usami,[†5]
Hiroshi Nakamura[†2] and Mitaro Namiki[†1]

This paper describes 'Geyser-0', the processor core based on MIPS R3000 architecture with a fine grain power gating technique that is designed for the research of power saving processor. To implement a platform for OS development, a computer system with Geyser-0 core is prototyped using FPGA. For evaluating performance and efficiency of power gating, a performance counter module is designed and implemented. Then we evaluated performance of the computer system and its function of power estimation. As the result, the prototyped computer system marked about 400 times faster than the simulation, and estimated power consumption of a computing unit in sleep period with an error rate of about 20[%].

1. はじめに

現代の高度情報化社会を支えるシステムLSIは，これまで動作クロックの高速化と製造プロセスの微細化によって著しい性能向上を達成してきた．しかしクロックの高速化に従い，これらLSIを構成するトランジスタを駆動するためのダイナミック電力の増大と，それに伴うLSIの発熱が問題となった．一方，プロセスの微細化技術はLSIの動作電圧の低減をもたらし，ダイナミック電力の増大を抑制することに寄与した．ところが，プロセスの微細化は同時に漏れ電流を増大させリーク電力を顕在化させる要因となった．近年ではLSIの消費電力のうちリーク電力の占める割合が支配的となってきており，これを抑える技術の確立が求められている．

LSIの消費電力の問題に対しては，ハードウェア，ソフトウェアの各分野で様々な取り組みが行われてきた．半導体レベルでは，ボディバイアスやVDDコントロールなどの技術が，回路レベルでは機能ブロックごとのクロックゲーティングやパワーゲーティングなどの技術が提案された．またソフトウェアにおいてもメモリ管理機構やスケジューリングの工夫によって消費電力の削減を図る取り組みが行われている[4]ほか，ソフトウェアとハードウェアの連携によって省電力化を実現する研究も行われている．

例えばDonald[2]らは，マルチコアプロセッサにおけるコアの製造バラつきに着目し，消費電力の少ないコアから優先的にタスク割り当てを行う技術を提案した．またRamamurthy[3]らは，性能を考慮しつつメモリ配置を工夫し，メモリデバイスの電源管理を行う技術を提案した．これら既存の省電力研究においては，処理性能と消費電力のトレードオフが課題とされた．

そこで，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「情報

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

Vol.2009-ARC-183 No.34
Vol.2009-OS-111 No.34
2009/4/24

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」[1] の一プロジェクトでは，演算ユニットごとに命令サイクル単位でパワーゲーティングを施し，性能を劣化させずにリーク電力の削減を図る技術を施した MIPS R3000 アーキテクチャベースのプロセッサコア，Geyser-0 を試作した．またこのプロジェクトにおいて東京農工大学並木研究室では，DVFS などシステムソフトウェアレベルの省電力技術を研究するとともに，Geyser-0 用のマルチタスク OS である Geyser-0 OS の試作を行った[7]．

本稿では，Geyser-0 及び Geyser-0 OS について概説するとともに，Geyser-0 用 OS の開発環境を実現するために行った，Geyser-0 を用いた FPGA での計算機システムの試作，またその中でも特に試作計算機システム上での電力削減効果の推算方法の設計について記述する．

2. Geyser-0 概要

2.1 Geyser-0 の構造とパワーゲーティング

Geyser-0 は MIPS R3000 アーキテクチャをベースとしたプロセッサコアであり，ALU，SHIFT，MULT，DIV の各演算ユニットとシステム制御コプロセッサ CP0 に対して命令サイクルごとの細かい粒度で動的にパワーゲーティングを行う機能を持つ（図 1）．

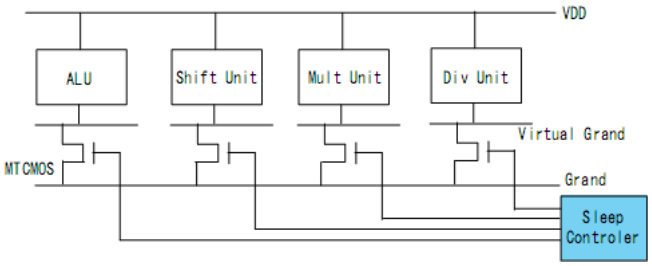

図 1 細粒度パワーゲーティングの仕組み[7]

Geyser-0 の実装は，本研究プロジェクトの中でアーキテクチャ及び回路設計のグループによって行われた[5][6]．Quick Sort や Dijkstra などいくつかのベンチマークによる評価の結果，Geyser-0 はパワーゲーティングの効果により，コア全体で平均約 47%の消費電力の削減を達成した[5]．

2.2 組込み OS 開間による評価

並木研究室では，同研究室で開発された組込み向け OS「開聞（かいもん）」の Geyser-0 向け移植版であるマルチタスク OS「Geyser-0 OS」を試作した[7]．この OS はタスクスケジューラ，システムコール，例外処理，割込み管理などの機能を備える．Geyser-0 OS 上でいくつかのベンチマークプログラムをマルチタスクで動作させた結果，動的パワーゲーティングにより演算器の消費電力を半減（図 2）し，またコア全体で平均約 37%のリーク電力削減効果が確認された（図 3）．

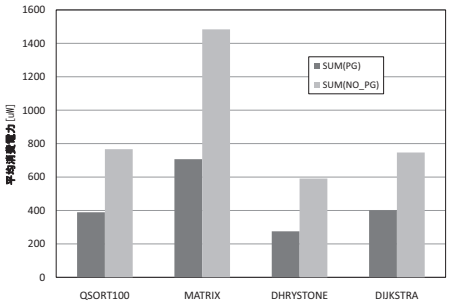

図 2 25 ℃での演算器の総消費電力

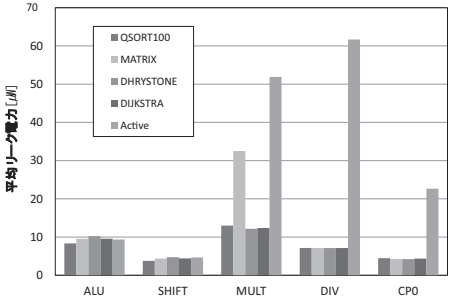

図 3 25 ℃での各演算器のリーク電力

3. 課題と目標

前章で述べた評価においては，シミュレーション時間の長さ，メモリと入出力装置の実装が課題として残された．そこで本研究ではこれらの課題を解決するため，Geyser-0 を用いた FPGA による計算機システムの試作を行う．

3.1 シミュレーション時間

先行研究[7] では，Geyser-0 OS の開発及びベンチマークの実行による電力評価は，シミュレータを使用した RTL/ゲートレベルシミュレーションにより行われた．これらのシミュレーションに要した時間は，表 1 に示すように，プログラムのサイクル数から予測される実行時間に対し，RTL で数万倍，ゲートレベルでは数千万倍に達した．このことは，より長い時間での電力評価や大規模なプログラム開発において障壁となる．OS 開発の立場では，開発効率の視点において，実時間で実行可能な環境が要求される．

Vol.2009-ARC-183 No.34
Vol.2009-OS-111 No.34
2009/4/24

表 1 各アプリケーションごとの予想実時間 (単位: msec)[7)]

プログラム (+KERNEL)	RTL	ゲートレベル	予想実時間
QSORT100*2	$36 * 10^3$	$18300 * 10^3$	1.21
MATRIX*2	$28 * 10^3$	$19800 * 10^3$	0.88
DHRYSTONE*2	$33 * 10^3$	$20400 * 10^3$	1.16
DIJKSTRA*2	$36 * 10^3$	$19200 * 10^3$	1.12

3.2 メモリと入出力装置

シミュレーションでは，メモリはすべて RTL 記述によるレジスタアレイとして実現されていた．これはアクセスコストが極めて小さく，キャッシュメモリなど本来ならばより高速であるはずの記憶領域と差別化されないため，OS 開発においてはメモリ管理機構の設計や評価において実際的な計算機とは異なる環境となってしまう．その反面，論理合成などの負荷が肥大化することからメモリ空間を多大に確保することが難しく，Linux のポーティングなど大規模なプログラムの開発には不向きでもある．また入出力装置もないため，ユーザインタフェースや実行時におけるデータの送受信機能などの実現も難しい．OS 開発環境として実現するために，本計算機システムではこれらの課題を解決する必要がある．

3.3 電力削減効果の評価方法

FPGA は，デバイス内部のロジックセルの LUT(Look Up Table) や配線情報をコンフィギュレーションすることにより，任意の論理回路を構成する．したがって Geyser-0 の最大の特徴であるパワーゲーティングの働きによって各演算ユニットにスリープをさせる信号が伝達されても，実際に電源供給が遮断されることは無い．つまり，消費電力を実測することは構造上不可能である．このため，試作計算機システムを用いて消費電力削減量の評価を行うための方法を検討する必要がある．

3.4 目　　標

本研究では，FPGA 上に Geyser-0 コアを移植した上で各種のメモリコントローラ，入出力モジュールを接続し，FPGA ボード上で高速実行可能かつ実際的な計算機システムを試作する．

また試作する計算機システムでは，パワーゲーティングにより演算ユニットがスリープした状況を観測し，その情報を元に，シミュレーションで算出された消費電力値から推算するという方法によって消費電力の削減量を評価する．本研究ではこのための具体的な推算方法とスリープ情報の記録方法を検討し，評価機構を設計する．

4. 計算機システムの設計

4.1 全 体 構 成

試作する計算機システムでは，Geyser-0 をコアとし，各種のメモリコントローラ及び評価機構（パフォーマンスカウンタ）を含む入出力装置をバス接続する．

全体の構成を図 4 に示す．これらのモジュールの説明を，次項以降で示す．

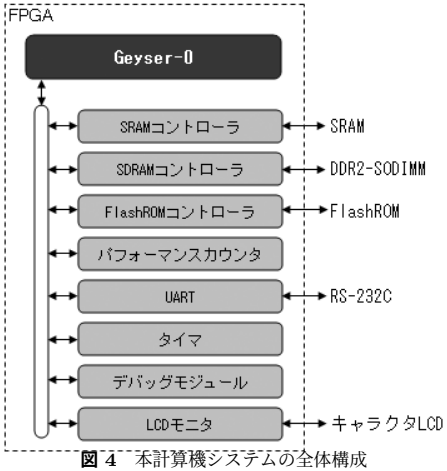

図 4 本計算機システムの全体構成

4.2 パフォーマンスカウンタ

一般にプロセッサには，演算能力などの処理性能を解析するためのパフォーマンスカウンタが実装されている．これらのカウンタを利用することで，ソフトウェア開発者は OS などのプログラムを実行した際にボトルネックとなる部分をプロファイリングしたり，設計パラメータと性能とのトレードオフポイントを観測したりできるようになり，実行環境におけるプログラムの最適化を図ることができる．

また本計算機システムでは，パワーゲーティングによる電力削減効果を推算するための機能の実現が要求される．この推算機能によって，ソフトウェア開発者はプログラムがパワーゲーティングの働きによりどの程度の電力削減効果を得ているかということを知ることができる．

Vol.2009-ARC-183 No.34
Vol.2009-OS-111 No.34
2009/4/24

4.2.1 消費電力の推算方法

パワーゲーティング対象である演算ユニットの消費電力は，シミュレーションでも FPGA での実装でも実測することはできない．このためシミュレーションでの評価では本項で次に示すような方法によって消費電力を求めた．

ある観測区間において，観測対象の演算ユニットの消費電力は，パワーゲーティングが行われていないアクティブな状態のときの消費電力と，パワーゲーティングが行われているスリープ状態のときの消費電力とに分けられる（図 5）．つまり，前者の電力をを P_{Aall}，後者を P_{Sall} で表すとすると，演算ユニットの消費電力は

$$P = P_{Aall} + P_{Sall} \quad (1)$$

となる．このうちアクティブ時の消費電力は先行研究[7] において演算ユニットごとに取得済みであり，観測したサイクル数を乗じるだけで観測区間における全アクティブ時消費電力 P_{Aall} 算出することができる．

一方，スリープ時の消費電力は，演算ユニットのスリープ状態への遷移による過渡現象の影響を考慮しなければならない．この過渡現象は図 6 に示すように消費電力のオーバヘッドを発生させるもので，それを含めたスリープ時の消費電力はスリープサイクル数 N の長さに依存する．N が長いほどオーバヘッドとなる消費電力は収束する方向へ向かう．したがって各スリープ時の消費電力を P_S とすると，P_S は状態遷移に伴う過渡現象によるオーバヘッドを含めて N の関数 $P_S(N)$ として表すことができる．$P_S(N)$ も先行研究[7] において各ユニットごとに取得済みである．この研究においては N が 1 サイクルから 1000 サイクルのときまでの各ユニットの $P_S(N)$ が取得された．これは，1000 サイクルを上回る長さの $P_S(N)$ は過渡現象の収束により，$P_S(1000)$ とみなすためである．

図 5 パワーゲーティングによる電力削減の模式図

観測区間において，N サイクルのスリープ状態があった回数を M_N 回とすると，その区間におけるスリープ状態の消費電力の総和は

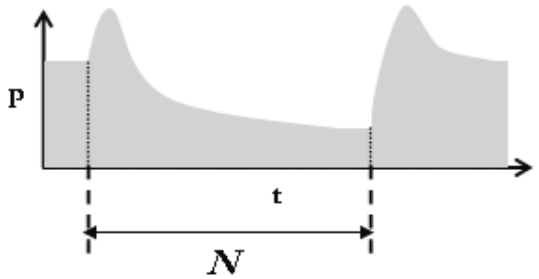

図 6 パワーゲーティングによるスリープ状態遷移時の過渡現象

$$P_{Sall} = \sum_N (P_S(N) * M_N) \quad (2)$$

として表すことができる．このとき $P_S(N)$ は既知であるので，実行時においては各スリープサイクル数に対する回数 M_N を取得する必要がある．そこで本計算機システムにおける消費電力の評価機構には M_N を観測し記録するための機構を搭載する必要がある．

取得した M_N からスリープ時の消費電力を求めることで，観測区間における観測対象ユニットの全消費電力を算出することができる．また観測区間のすべてにおいてユニットがパワーゲーティングされずアクティブだった場合の消費電力を計算し，パワーゲーティングを行った場合の消費電力との差分を求めることで，電力削減量を定量的に推算することができる．

4.2.2 ハードウェアカウンタ

パワーゲーティングによる電力削減効果の推算用として，先述した N サイクルスリープ回数 M_N を記録するためのカウンタを実装する．

M_N の記録方法について説明する．取得するべきパラメータが数列であるため，M_N の記録機構は単純なレジスタとしてではなくレジスタアレイまたはメモリの形で実装なければならない．そこで図 7 に示すような，N をインデックスとする記録機構を考える．各ユニットにおいて，一回のスリープサイクルが終わったタイミングで，インデックス N のエントリに，N サイクルスリープした回数 M_N を記録する．

M_N を記録するべき N の値は 1 から 1000 サイクルまで存在しうるが，それをパワーゲーティング対象の 5 ユニットすべての分に用意することとなると，実装規模の点で制約を受ける恐れもある．そのため，ある程度の粒度でスリープサイクル数 N の計測解像度を下げ

Vol.2009-ARC-183 No.34
Vol.2009-OS-111 No.34
2009/4/24

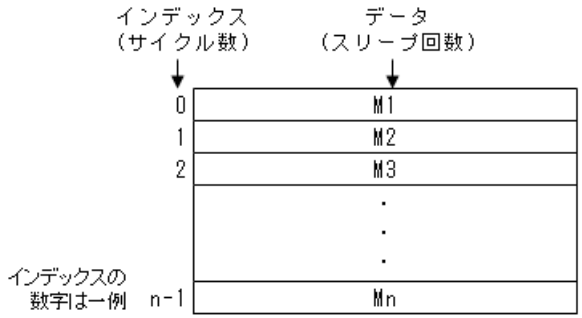

図 7 N サイクルスリープ回数 M_N の記録機構

ることも検討する．つまり，サイクル数 N の数に対して区間を設定し，その間に入るサイクル数を同一の N として計測する．この区間の設定については，標準数列を用いることで，計測結果が短いスリープサイクル数に偏った場合も，長いスリープサイクル数に偏った場合も，計測誤差を一定範囲に収めることができると見込んでいる．

解像度を下げる場合の誤差に加え，記録機構の設計上のもう一つの課題として，カウンタの飽和対策が挙げられる．これは，プログラムの繰り返し演算などでスリープサイクル数が偏った場合に，スリープ回数 M_N がエントリから飽和する可能性がある，ということである．エントリのデータ幅にある程度の大きさがあればこういった状況に陥る可能性は低いと考えられる．そのため，飽和を検出するか，飽和が予測されたときに例外を発生させる機構を実装し，例外処理によって飽和状態の処理を行う設計を検討する．

また OS 開発での処理性能の測定に用いるため，次のようなカウンタを実装する．

- 経過サイクル数
- メモリアクセス状況
- キャッシュミスヒット回数
- TLB ミスット回数

4.2.3 API

本計算機システムは OS 開発に供用することを目的として試作している．したがって先述したカウンタは，OS/プログラム開発者が OS 中のシステムコールやプログラム中に埋め込んだコードによって参照，操作されなければならない．そのようなプログラミングインタフェースとしては PAPI(Performance API) がよく知られている．本計算機システムでもこ

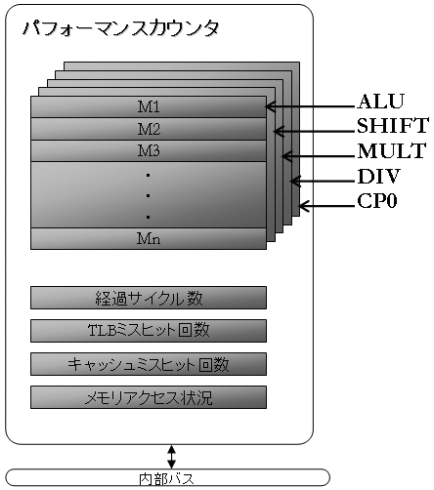

図 8 パフォーマンスカウンタ

れを参考に，カウンタのハードウェアを抽象化し，プログラム中に挿入することでカウンタの参照や操作を可能とする関数群を用意する．

プログラム中でカウンタを使用するとき，一般的には観測対象のコードの直前でカウンタの初期化を行ってから観測をはじめ，観測対象が終了したらカウンタの動作を止め値を出力する，といった使用法がとられることが多いと考えられる．しかし複数あるカウンタで同時に測定を行う場合，これらのコーディングをカウンタごとに行うことはプログラム開発者にとって負担である．そこで，同時に測定に使いたいカウンタをモニタ対象としてまとめ，一括した操作を可能とする．

API では具体的には，次のような機能を持つ関数を提供する．

- モニタ対象のカウンタの値を初期化
- 指定したカウンタをモニタ対象に追加
- 指定したカウンタをモニタ対象から除外
- カウンタの記録間隔を指定
- モニタ対象のカウンタの測定を開始
- モニタ対象のカウンタの測定を停止
- モニタ対象のカウンタの名称を出力

- モニタ対象のカウンタが保持している値を出力

4.3 メモリコントローラ

FPGA ボード上で FPGA と接続された各種のメモリデバイスを FPGA 内の Geyser-0 に接続し，RAM，ROM 領域として用いるため，それぞれにメモリコントローラを作成し，Geyser-0 とバス接続する．

4.3.1 SRAM コントローラ

本計算機システムでは RAM 領域として FPGA ボード上の SRAM を用いる．Geyser-0 によるロード/ストアのほかプログラムを配置して命令のフェッチを行うため，一回のバストランザクションで 1 ワードの読み込みを完結する必要がある．

4.3.2 FlashROM コントローラ

本計算機システムでは FPGA ボード上の FlashROM をブートデバイスとして用い，リセット例外によるブートローダプログラムの実行，プログラムイメージの格納を行う．このため FlashROM に対しては Geyser-0 がロード/ストアのほか命令フェッチを行えるよう，SRAM と同様にバストランザクション一回で読み込み動作を完結させる．

また FlashROM デバイスは一般的にコマンド入力という手法によって動作の設定を行う．本計算機システムでは電源投入直後のリセット例外発生時に命令フェッチを行う必要があるため，それに対応したリード動作を行える状態にするため初期的なコマンド入力を FlashROM コントローラにハードウェア的に組み込む．

4.4 入出力装置

本計算機システムでは，ユーザインタフェースの実現，データの入出力や Geyser-0 のペリフェラルとして，次に挙げる周辺モジュールを実装する．いずれのモジュールも Geyser-0 とはバス接続し，ロード/ストア命令によってアクセス可能とする．

- UART
  Geyser-0 を用いた本計算機システムでは，ソフトウェア開発時のプログラムイメージの転送や実行時のデバッグプリント，ユーザのキー入力や実行結果の表示などのための CUI を提供する必要がある．そのため FPGA ボード上のシリアル入出力端子を，URAT モジュールによって Geyser-0 と接続する．
- タイマ
  本計算機システムでは，マルチタスク OS を移植した際にタイムスライスによるタスクスイッチの機能をサポートするため，一定時間おきに割込みを発生させるインターバルタイマを用意する．このタイマは，サイクル数を計測するためのカウントレジスタと，コンペアレジスタを持ち，あらかじめコンペアレジスタに設定したカウント数にカウントレジスタの値が一致すると割込みを発生する．
- デバッグモジュール
  デバッグのため，任意のプログラムカウンタ値での実行停止（ブレーク）や，一サイクルごとの実行（シングルステップ実行）などの機能を提供する．
- LCD モニタ
  デバッグのため，FPGA ボード上のキャラクタ LCD にバスやレジスタの値を表示する．

4.5 ブートローダ

本計算機システムでは FlashROM 内に設定したリセットベクタにブートローダを配置する．

4.5.1 ブートシーケンス

ブートローダは，リセットベクタに配置されることにより計算機システムの電源投入直後などリセット例外の発生時に実行され，ハードウェアの初期化，プログラムの展開などを行う．展開・起動するプログラムの選択のため，簡単なユーザインタフェースを提供する．これらブートローダの行う処理をまとめると，次の通りである．

(1) スタックポインタ，グローバルポインタの初期化
(2) シリアルコンソールへのブート可能なプログラムのメニュー表示
(3) シリアルコンソールからの入力の受け付け
(4) 選択されたプログラムの ROM から RAM への展開
(5) RAM のプログラム展開先へ PC をジャンプ

FlashROM 内のリセットベクタに配置されたブートローダの実行から，ブートメニューで選択されたプログラムイメージの RAM への展開，RAM へのジャンプまでの一連の動作を，図 9 に図示する．

4.5.2 関　　数

これらの機能を実現するために次の関数を用意する．

- putch 関数
  引数で指定した文字コードの文字を UART モジュールに書き込むことで，コンソール出力を行う．
- getch 関数
  コンソールに入力されたことで UART モジュールが受信した文字を読み出し，戻り値に返す．

Vol.2009-ARC-183 No.34
Vol.2009-OS-111 No.34
2009/4/24

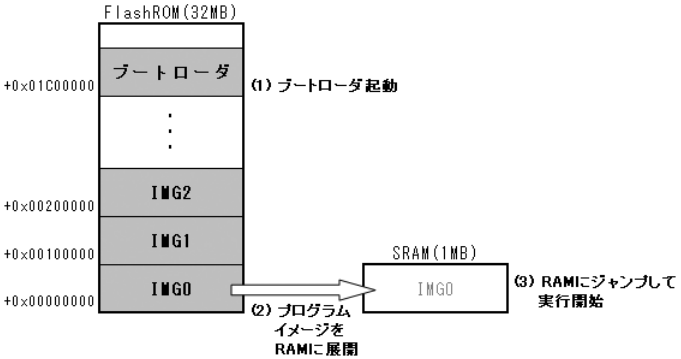

図 9 ブートシーケンス

- putstr 関数
  引数 str で指定されたアドレスから始まる文字列をコンソールに出力する．内部的には，終端文字が出現するまで一文字ずつ putch 関数を繰り返しコールする．
- memcopy 関数
  引数で指定したソースアドレス，データ長，ディスティネーションアドレスにしたがって，メモリ内のデータのコピーを行う..
- jump_to_sram 関数
  プログラムカウンタを SRAM の先頭アドレスへジャンプする．

5. 実　　装

計算機システムの実装には，Xilinx 社の FPGA，Virtex5 を搭載した FPGA ボード，ML501 を用いた．メモリデバイスには ML501 オンボードの SRAM(1MB), FlashROM(32MB) をそれぞれ RAM，ROM として用い，それらのためのメモリコントローラを FPGA 上に作成し，Geyser-0 とバス接続した．また入出力装置としてボード上の RS-232C 端子やキャラクタ LCD を接続した．

パフォーマンスカウンタでは，M_N の記録機構として 32 ビット幅，500 エントリのレジスタアレイを用意した．処理性能の測定用としては経過サイクル数のカウンタを実装した．

SRAM をメインメモリとして用いるとともに，FlashROM にはブートプログラムを配置し，リセットベクタに設定した．また FlashROM 内の別の領域にはブートするためのプログラムを書き込み，ブートプログラムによる CUI のメニューで選択されたプログラムがメインメモリに展開され，続いて展開された領域の先頭にジャンプすることで実行が開始される実装とした．またブートローダでは，SREC 形式ファイルの転送及び実行にも対応した．

Geyser-0 の動作クロック速度は，FlashROM の動作速度と FPGA ボード上のクロックリソースの制約から，16.5MHz とした．

6. 評価と考察

試作した計算機システムの評価結果を示す．

6.1 実 行 速 度

試作した計算機システムのプログラム実行速度を，RTL シミュレーションと比較した．実行速度の評価には，1000 × 1000 正方行列の乗算プログラムを用いた．またシミュレーション環境には Cadence 社の NC-verilog を使用し，RTL シミュレーションを行った．結果は表 2 の通りである．

表 2 プログラム実行速度の比較

	RTL sim.	本計算機
サイクル数	3.20×10^7	1.60×10^8
実行時間 [s]	3.77×10^3	9.71

シミュレーション環境では HDL で記述したレジスタ群をメモリとしておりアクセスコストが小さいが，試作した計算機システムはメモリアクセスに数サイクルかかるためサイクル数では不利である．しかし実時間で実行できる利点が大きく，シミュレーション環境の約 400 倍の実行速度を得た．この結果より，試作した計算機システムは OS 開発環境という点で，シミュレーションに比べ高い実用性を達成したといえる．

6.2 消費電力の推算機能

設計した消費電力の推算機能を試作した計算機システム上に実装し，動作の検証を行った．

観測対象のユニットは ALU，観測したプログラムは 1000 × 1000Matrix 乗算二回である．

M_N の記録機構によって取得された計測結果を表 3 に示す．

M_N の計測結果より推算したスリープ時リーク電力値と，その値のシミュレーションとの比較を，表 4 に示す．

消費電力推算機構では，平均約 20[%] の誤差率でスリープ時電力を推算した．シミュレーション環境ではメモリとしてレジスタアレイを使用しているためアクセスレイテンシがなく

Vol.2009-ARC-183 No.34
Vol.2009-OS-111 No.34
2009/4/24

表 3 1000 × 1000Matrix 乗算二回実行時の ALU スリープ状況

スリープサイクル数 N	N サイクルスリープの回数 M_N
1	2000000
5	2006
9	501
10	81130
11	6868
13	7031
32	929

表 4 1000 × 1000Matrix 乗算二回実行時の ALU スリープ時リーク電力

温度 [℃]	シミュレーション [uW]	本計算機 [uW]	誤差 [uW]	誤差率 [%]
25	0.77	0.57	0.20	26.1
65	1.09	0.85	0.24	22.3
100	1.69	1.36	0.34	19.8
125	2.38	1.95	0.44	18.3

5 サイクルでロード/ストアが可能だが，試作した計算機システムではメモリデバイスのレイテンシのため 7 サイクル以上を要するこの差がスリープ状況の計測結果に現れることで，電力推算の誤差として伝播している可能性が強いと考えられる．よってメモリアクセスコストの違いを考慮することでさらに誤差率を改善することができると予想される．

7. おわりに

本稿では，省電力 MIPS プロセッサ Geyser-0 を用いた計算機システムの FPGA による試作と，Geyser-0 のパワーゲーティングによる消費電力削減量の評価機構の設計について記述した．

7.1 まとめ

計算機システムの試作では，FPGA 上で Geyser-0 と各種メモリコントローラ，入出力装置を接続し，FPGA ボードのオンボードデバイスを用いて実際に計算機システムを実現した．実行速度の評価の結果，シミュレーション環境に比べ約 400 倍の高速化を達成し，OS 開発環境としての実用性を確認した．また消費電力削減量の評価機構では，具体的な推算方法を検討した上で，パフォーマンスカウンタ，プログラミングインタフェースを設計するとともに，設計課題を明確化した．さらに機能の一部を実装し動作検証を行った結果，平均約 20[%] の誤差率でスリープ時電力を推算した．

7.2 今後の課題

計算機システムについては，さらに大規模なプログラムの実行を可能とするため，SDRAM コントローラの実装，動作周波数の高速化，入出力機能の強化などを行う．また XMODEM などプログラム転送のためのプロトコルの実装を行う．

電力評価機構については，シミュレーションとの整合性を検証して推算精度を高めるとともに，設計課題の解決，未実装カウンタとライブラリの実装を行う．

謝辞 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 (VDEC) を通し, 株式会社半導体理工学研究センター, 富士通株式会社, 松下電器産業株式会社, NEC エレクトロニクス株式会社, 株式会社ルネサステクノロジ, 株式会社東芝の協力で行われたものである．

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．

参考文献

1) 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告 ARC-173, pp.79-84 (Jun 2007).
2) James Donald, Margaret Martonosi, 'Power Efficiency for Variation-Tolerant Multicore Processors', International Symposium on Low Power Electronics and Design (Proc. ISLPED'06) SESSION10, No.3, pp.304-309, (Oct 2006).
3) Pratap Ramamurthy et al., 'Performance-directed Energy Management using BOS', ACM SIGOPS Operating Systems Review, Vol.41, pp.66-77, (2007).
4) Bernhard Egger, Jaejin Lee, Heonshik Shinl, 'Scratchpad Memory Management for Portable Systems with a Memory Management Unit', EMSOFT '06: Proceedings of the 6th ACM & IEEE International conference on Embedded software, pp321-330, (2006).
5) 関直臣, Lei Zhao, 徐慧, 池淵大輔, 小島悠, 長谷川揚平, 天野英晴, 香嶋俊裕, 武田清大, 白井利明, 中田光貴, 宇佐美公良, 砂田徹也, 金井遵, 並木美太郎, 近藤正章, 中村宏, 「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」, 情報処理学会研究報告 2008-ARC-176, pp.71-76, (Jan 2008).
6) 白井利明, 香嶋俊裕, 武田清大, 中田光貴, 宇佐美公良, 長谷川揚平, 関直臣, 天野英晴, " ランタイムパワーゲーティングを適用した MIPS R3000 プロセッサの実装設計と評価 ", 電子情報通信学会技術研究報告. VLD, VLSI 設計技術, Vol.107, No.414(20080109) pp. 43-48.
7) 砂田徹也 他「省電力 MIPS プロセッサにおける OS の試作とシミュレーションによる電力評価」, 情報処理学会「システムソフトウェアとオペレーティング・システム」第 108 回研究報告, Vol.2008-OS-108, pp.163-170 (Apr 2008)