

マイクロプロセッサのエネルギー消費特性抽出とソフトウェア デバッグを用いた消費エネルギー見積もり

李東勲¹ 石原亨² 室山真徳² 安浦寛人² Farzan FALLAH³

¹九州大学工学部電気情報工学科 〒812-8581 福岡県福岡市東区箱崎 6-10-1

²九州大学システムLSI研究センター 〒814-0001 福岡県福岡市早良区百道浜 3-8-33

³米国富士通研究所先端CAD技術部 〒94085 1240 East Arques Avenue M/S345, Sunnyvale, CA, USA

E-mail: donghoon@c.csce.kyushu-u.ac.jp, ²{ishihara,muroyama,yasuura}@slrc.kyushu-u.ac.jp,

³Farzan.Fallah@us.fujitsu.com

あらまし マイクロプロセッサの消費エネルギーを高い抽象度でキャラクタライズする手法を提案する。消費エネルギーのモデルには線形式を用いる。本稿におけるキャラクタライズとは、線形式のパラメータ係数をゲートレベルのエネルギー見積もり値に基づく重回帰分析により決定することを意味する。キャラクタライズは対象となるデバイス(マイクロプロセッサと使用されるプロセステクノロジー)ごとに行う。提案手法で導出した線形式を用いることによりマイクロプロセッサ上でソフトウェアを実行する際に消費されるエネルギーを数千から数万命令ごとの粒度で正確かつ高速に見積もることができる。M32R-II プロセッサと SDRAM を主記憶としたシステムを対象として実験を行った結果、平均誤差 8 %、最大誤差 38 % で消費エネルギーの見積もりが行えることを確認した。

キーワード 消費エネルギー、キャラクタライゼーション、マイクロプロセッサ

An Energy Characterization Technique for Fast and Accurate Software Power Estimation

Lee DONGHOON¹ Tohru ISHIIHARA² Masanori MUROYAMA² Hiroto YASUURA²
and Farzan FALLAH³

¹Dept. of EE and CS, School of Eng., Kyushu Univ. 6-10-1 Hakozaki, Higashiku-ku, Fukuoka, 814-0001 Japan

²System LSI Research Center, Kyushu University 3-8-33 Momochihama, Sawara-ku, Fukuoka, 814-0001 Japan

³Advance CAD Technology, Fujitsu Labs. America, 1240 E. Arques Ave. M/S345, Sunnyvale, CA, 94085 USA

E-mail: ¹donghoon@c.csce.kyushu-u.ac.jp, ²{ishihara,muroyama,yasuura}@slrc.kyushu-u.ac.jp,

³Farzan.Fallah@us.fujitsu.com

Abstract In this paper, we propose a characterization framework which helps designers in developing an accurate energy consumption model for a target processor. We use a linear model for energy consumption and we find the coefficients of the model using Linear Programming (LP). We use our approach for estimating the energy consumption of an M32R-II processor system including CPU core, on-chip caches and an off-chip SDRAM. Experimental results demonstrated that the error of our technique is on an average 8% and worst case 38% compared to the gate-level estimation results. Once the linear equation has been developed, the power consumption of a target program can be estimated with the speed of 35,000 instructions per second.

Keyword Energy Consumption, Characterization, Microprocessor

1. はじめに

携帯型情報機器の普及と共に組込みシステムの低消費電力化に対する要求はますます強まっている。しかし、組込みソフトウェア設計の分野では低消費電力化に対する意識はそれほど高くない。消費電力はハードウェアで削減するものだという意識があったため

であると考えられる。

筆者らは組込みソフトウェア分野の技術者が設計段階で消費エネルギーを意識できる環境の提供を目標として、ソフトウェアデバッグを用いてプロセッサシステムの消費エネルギーを解析する研究に取り組んでいる。本稿では、ソフトウェアデバッグから容易に抽

出可能なパラメータを用いてプロセッサシステムの消費エネルギーをキャラクタライズする手法を提案する。モデルには線形近似モデルを使用する。ゲートレベルの電力見積もり値を基準値としてキャラクタライズを行う。プロセッサコア、キャッシュメモリおよび外部主記憶のエネルギー消費を見積もり対象とする。

デジタルシステムの電力見積もり手法は今日までに数多く提案されている。ソフトウェアの振る舞いが消費エネルギーに与える影響を見積もる最も正確かつ高速な方法は実チップの電力を測定することである。しかし、電力測定では、プロセッサ内部のホットスポット解析や、数 μ 秒オーダーの間に消費されるエネルギーを見積もることは難しい。一方、設計初期段階の見積もり手法の多くはゲートレベルやRTLのシミュレーション[1]をベースとしているため、大規模ソフトウェアの解析には膨大な時間を要する。また、ハードウェアシミュレータの使用はソフトウェア開発者にとって敷居が高い方法であると考えられる。命令セットシミュレータ（以下 ISS）を用いた電力見積もり手法も数多く提案されているが[12][15-19]、電力見積もり機能を持つ ISS は一部のプロセッサに限られているため、この機能のサポートが無い場合は、専用のシミュレータと電力モデルを開発する必要がある。また、現時点では電力見積もり機能を持つ組み込みプロセッサ向けソフトウェアデバッガは少ない。

筆者らの提案する手法は対象とするプロセッサシステムに対して半自動でキャラクタライズを行うことを特徴とする。生成された線形モデルは、専用に開発されたサイクル精度 ISS や GNU のソフトウェアデバッガと組み合わせて使用することができる。

本稿の構成は次の通りである。2章で関連研究を紹介し、3章で提案手法の詳細を説明する。4章では実験結果を示し、5章で本稿をまとめる。

2. 関連研究

文献[2,3]では実チップの電力測定手法が提案されている。これらの手法はホストマシンから制御可能なマルチメータを使用する。文献[2]の PowerScope はメモリや CPU などの部分システムの電流を個別に測定できないという問題がある。Itsy [3] では上記の問題が解決されているが、短時間の電流変化を測定できない。従って、上記の手法を用いる場合、プロセッサ上でプログラムを連続実行して、平均電流を測定することになる。文献[4]ではサイクルごとの電力を測定する手法が提案されている。しかし、チップ内部のハードウェアモジュールを個別に解析することは困難である。

ISS を用いた見積もり手法も数多く提案されている。文献[5,6]では、同じ種類の命令を連続実行させた時の実チップの電流測定値を用いて各命令の消費エネルギ

ーをモデル化する手法が提案されている。文献[7]では、上記の手法の拡張が行われている。命令アドレスやデータアドレス、オペランドの値およびレジスタファイルのアドレスを考慮に入れることにより見積もり精度が改善されることが報告されている。主な問題は、すべての命令および異なる命令間の影響を見積もるために大量のテストベンチを使用して繰り返し電流測定を行う必要がある点である。文献[8,9]では一部の命令のみを使ってキャラクタライズを行うことにより測定の効率化を行っている。しかし、何れの手法もチップ内部の消費エネルギーの内訳を解析することは難しい。

文献[10,11]では命令の実行回数をパラメータとした単純な線形近似により 10%未満の誤差でプロセッサコアの電力見積もりを行えることが示されている。文献[12]では、サイクル精度のシミュレータ (Trimaran [13]) を使用して VLIW プロセッサのパイプラインステージ毎の消費エネルギーを見積もる手法が提案されている。この手法は特定のプロセッサを対象とした手法であるため、ターゲットが変わればシミュレータおよび電力モデルを作り直す必要がある。文献[14]はRTL記述からサイクル毎に各命令の平均エネルギーをキャラクタライズする手法を提案している。しかし、命令毎の平均エネルギーからプログラムの実行時に消費されるエネルギーを見積もる方法は示されていない。上記以外にも線形モデルを用いてプロセッサシステムの消費エネルギーを見積もる手法は多数提案されているが[15-19]、キャラクタライズの手法に関して議論した研究は少ない。文献[20]では、より高精度な消費エネルギーの見積もり値（例えば SPICE で見積もった値）を基準値とした重回帰分析により、精度の良い近似が行えるという一般的な議論が行われている。しかし、パラメータの決定方法やキャラクタライズに用いるテストベンチのあり方に関する議論はない。

筆者らが事前に行った実験では、無作為に選択したテストベンチを使用してキャラクタライズを行うと 100%以上の誤差が生じることが確認されている。本稿では、キャラクタライズのためのいくつかのガイドラインを示し、精度の良い線形近似式を得る方法を提案する。係数の決定には線形計画法を用いる。

3. エネルギー消費モデル

3.1. 一般的な考え方

式(1)に示す線形近似式を用いてプロセッサシステムの消費エネルギーを見積もる手法が数多く提案されている [1][5-12][14-20]。

$$E_{\text{estimate}} = \sum_{i=0}^N c_i \cdot P_i \quad (1)$$

ここで、 E_{estimate} 、 P_i 、 c_i 、および N は、それぞれ消

費エネルギーの見積もり値、線形モデルのパラメータ、各パラメータの係数、およびパラメータの数を表す。文献[6]のWattchでは、ターゲットとするプロセッサを構成するマクロモジュールの活性化率をパラメータとし、そのモジュールが活性化されたときの消費エネルギーをパラメータの係数として用いる。文献[10]のJouleTrackでは命令の実行回数をパラメータとし、パラメータ係数の値は消費電力の実測値に基づく線形回帰分析により求めた値を使用する。パラメータ P_i の種類とその係数 c_i の値が決定されれば後はパラメータ P_i の値を計測することにより、消費エネルギー $E_{estimate}$ を高速に求めることが出来る。Wattch[6]は、マクロモジュール毎の消費エネルギーをテーブルとして持っており、サイクル精度シミュレータを用いて見積もったモジュールの稼働率とテーブルの値を掛け合わせる事により消費エネルギーを計算する。この場合、ターゲットデバイスに対応したサイクル制度のハードウェアシミュレータが必要になる。文献[10]では、命令の実行回数をパラメータとした単純な近似により、StrongARMプロセッサの消費エネルギーを8%未満の誤差で見積もる方法を提案している。しかし、オフチップメモリの消費エネルギーを考慮した場合、1パラメータでは正確な見積もりが出来ない。また、文献[20]では、回帰分析によりLSIチップの消費エネルギーをキャラクタライズする一般的な考え方が示されているが、パラメータや係数の決定方法、およびキャラクタライズに適したテストベンチに関する議論が行われていない。

本稿では、ソフトウェア開発の観点で消費エネルギー解析を行う一つの手段として、ソフトウェアデバッガから容易に抽出可能なパラメータのみを用いてプロセッサおよび外部主記憶のエネルギー消費をキャラクタライズする方法を提案する。係数の決定には線形計画法を用いる。ソフトウェアデバッガを利用する理由は以下の通りである。

1. ソフトウェア開発者が利用しやすいこと
2. 容易にリターゲット可能であること
3. ハードウェアシミュレータと比較して一般に高速であること

3.2. キャラクタライズのアウトライン

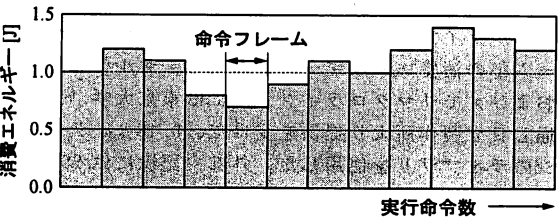


図 1 テストベンチの分割

筆者らは図 1 に示す通り、プログラムを一定命令数ごとに分割しキャラクタライズの際のサンプル点(以下、命令フレーム)として用いる。命令フレーム毎にゲートレベルの電力見積もりを行い基準値とする。

以下のステップで消費エネルギーのキャラクタライズを行う。

1. パラメータの決定
2. テストベンチを命令フレームへ分割
3. 命令フレーム毎に命令トレースを生成
4. 命令フレーム毎にパラメータ値を抽出
5. パラメータ毎に標準偏差と相関係数を評価
6. 標準偏差と相関係数の値が基準を満たすまでテストベンチまたは命令フレームを変更
7. 命令フレーム毎に消費エネルギーの基準値を計測する(ゲートレベルの見積もり値を使う)
8. 線形計画法を用いて係数を決定する。

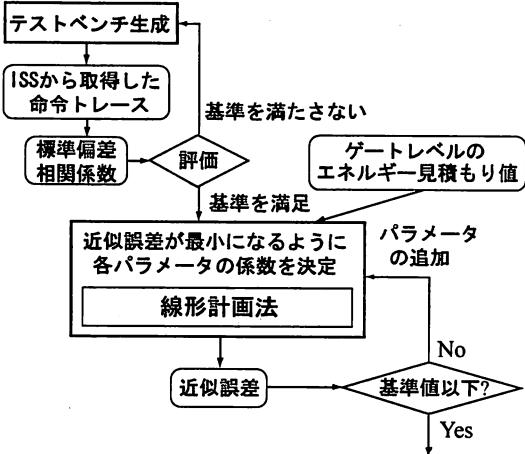


図 2 キャラクタライズの概要

3.3. パラメータとテストベンチの決定

パラメータにはソフトウェアデバッガから容易に抽出でき、かつ消費エネルギーに影響を与えるものを選択することが重要である。多くのプロセッサシステムでは、メモリアクセスに要するエネルギーが非常に大きな割合を占めるため、キャッシュメモリやスラッチパッドメモリ、外部メモリへのアクセス回数をパラメータとして使用することは必要不可欠である。それ以外にCPUがストールを起こす原因をパラメータとして使用する。筆者らは以下のパラメータを使用してキャラクタライズを行う。

- 命令毎の実行回数
- 命令キャッシュミス回数
- データキャッシュミス回数
- 命令キャッシュとデータキャッシュを同時にミスヒットする回数
- ロードまたはストア命令が連続して実行される

回数（バーストアクセス）

- 分岐命令のうち分岐が不成立であった回数
- 分岐不成立と命令キャッシュミスが同時に発生する回数
- リードアフターライト(RAW)ハザードの発生回数
- データキャッシュミスとRAWハザードが同時に発生する回数
- 乗算命令や除算命令などのマルチサイクル命令がキャッシュミスを引き起こす回数

キャラクタライズの際に最も重要であるのがテストベンチの選択である。事前に行った実験では、無作為に選択したテストベンチを使用してキャラクタライズを行うと、100%以上の誤差を生じることを確認した。テストベンチは以下の二つの基準に基づいて生成する。

1. 別々のテストベンチから抽出した同じパラメータ値の標準偏差がすべて基準値以上であること。命令フレーム（図1参照）が変わってもあるパラメータの値が変化しない場合には、そのパラメータが消費エネルギーに与える影響を見積もることが出来ない。従って、キャラクタライズを行う際にはパラメータの値が命令フレーム毎にくばらついていることが望ましい。
2. あるパラメータの値と別のパラメータの値との相関係数がすべて基準値未満であること。パラメータ同士に強い相関がある場合には、どちらかの影響が他方のパラメータによって隠されてしまう。従って、各パラメータの値がなるべく独立に変化するようなテストベンチおよび命令フレームを作成する必要がある。

現在は上記の基準に基づいてテストベンチを手作業で作成しているが、将来は命令セットやキャッシュメモリの構成などからキャラクタライズ用のテストベンチを自動生成することを目指している。

3.4. 係数の決定方法

係数の決定には線形計画法を用いる。問題の定式化に用いる変数を以下に定義する。

- i : 時間フレーム(Time Frame)の番号
- j : パラメータの番号
- N : パラメータの数
- M : 時間フレームの数
- E_i : ゲートレベルシミュレーションで求めた i 番目の時間フレームに対応する消費エネルギー値
- E'_i : 式(1)に示す線形近似式を用いて見積もった消費エネルギーの見積もり値
- Y_i : 見積もり誤差の絶対値 $|E'_i - E_i|$
- P_{ij} : 命令セットシミュレータを用いて抽出したパラメータの値

- c_j : パラメータの係数（決定変数）

線形近似モデルを用いて消費エネルギーをキャラクタライズする問題は以下のように定義できる。

“ゲートレベルシミュレーションにより取得した E_i の値と命令セットシミュレータを用いて取得した P_{ij} が入力として与えられたときに見積もり誤差 Y_{obj} が最小になるように係数 c_j の集合を決定する問題”

Minimize

$$Y_{obj} = \sum_{i=0}^M Y_i$$

Subject to

$$-Y_i \leq \left(\sum_{j=0}^m c_j \cdot P_{ij} \right) - E_i \leq Y_i \quad (i = 0..M-1)$$

$$c_j, P_{ij}, E_i, Y_i \geq 0 \quad (i = 0..M-1, j = 0..N-1)$$

4. 実験と考察

4.1. 実験環境

実験には M32R-II プロセッサを使用した。M32R-II は Renesas Technology 社から東京大学 VDEC を通じて提供された、32 ビット RISC マイコンである。CPU は 5 段パイプライン構造を持っている。メモリは 8KB 2way セットアソシアティブの命令/データキャッシュメモリ、32KB の SRAM を内蔵する。メモリアドレス管理は TLB (Translation Look-aside Buffer) アドレス変換方式を使用している。TLB は 16 エントリである。生成した線形式を評価するために 5 種類のベンチマーク (JPEG エンコーダ、MPEG2 エンコーダ、compress、FFT、DCT) を用いた。コンパイル時に最適化オプション -O3 を付けたものと付加しないものをそれぞれ用いた。

4.2. キャラクタライズの詳細フロー

キャラクタライズの具体的な流れを図3に示す。まず GNU C コンパイラを用いてテストベンチから目的コードを生成する。次にプロセッサのゲートレベルシミュレーションモデル上で目的コードを実行する。ゲートレベルのシミュレーションには Cadence 社の Verilog シミュレータを用いた。ゲートレベルのシミュレーションにより得られたゲートのスイッチング情報 (SAIF: Switching Activity Interchange Format) とセルライブラリの消費エネルギー情報から命令フレーム毎の消費エネルギーを計算する。セルライブラリおよびメモリマクロライブラリには東京大学 VDEC を通じ日立製作所より提供を受けた、0.18 μm テクノロジーのライブラリを使用した。外部主記憶には SDRAM を仮定した。SDRAM の消費エネルギーモデルは、Micron テクノロジー社が公開している電力計算スクリプト [21] を使用した。プロセッサの消費エネルギーの計算には

SYNOPSYS 社の PowerCompiler を用いた。同時に各命令フレームに対応する各パラメータの値を命令セットシミュレータにより抽出する。命令セットシミュレータには M32R-II プロセッサ向けの GNU デバッガ (m32r-linux-gdb または m32r-linux-run) を用いた。係数を決定する際の LP ソルバには ILOG 社の CPLEX を使用した。

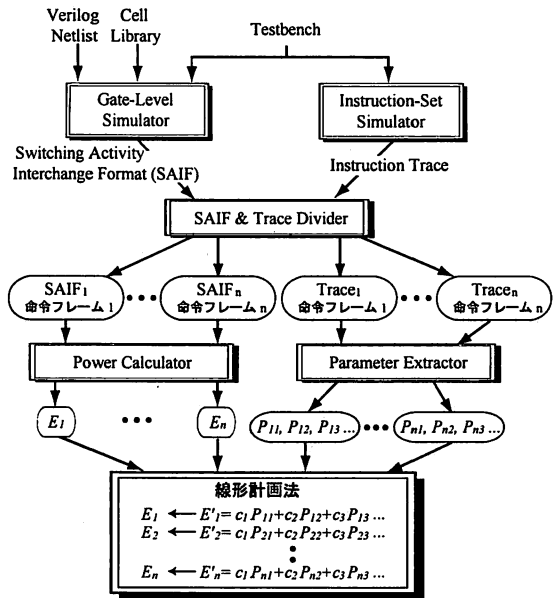


図 3 キャラクタライズの具体的な流れ

4. 3. 実験結果

キャラクタライズには各命令の実行回数やキャッシュミス回数など合計 88 個のパラメータを使用し、3.2 節で述べたガイドラインに基づいて生成したテストベンチを用いた。このテストベンチは、各命令を一定回数実行するサブルーチン、キャッシュミスを引き起こすサブルーチンなどから構成される。実行命令数

は 500,000 命令 (5000 命令×100 命令フレーム) である。キャラクタライズに要した時間は Intel 社製 Xeon™ CPU 3.80GHz の計算機を使用して 12,420 秒 (約 3 時間半) であった。キャラクタライズが終了するとソフトウェアデバッガの実行速度 (約 35,000 命令/秒) で消費エネルギーの見積もりを行うことが出来る。

実験結果を表 1 に示した。語尾に opt が付いたベンチマークプログラムは -O3 オプションを付けてコンパイルしたものである。実験の結果から compress_opt を除くプログラムでは 10% 未満の平均誤差で見積もりが行えることを確認した。図 4 に JPEG エンコードの消費エネルギー見積もり例を示した。消費エネルギーの傾向が正確に見積もられていることが確認できる。図 5 には最も誤差の大きかった compress_opt の結果を示した。誤差の原因解析は今後の課題である。

表 1 消費エネルギー見積もり誤差

ベンチマーク プログラム	誤差 (%)		誤差の標準偏差
	平均	最大	
JPEG	4.26	29.29	5.08
JPEG_opt	9.07	29.04	4.91
MPEG2	7.42	29.17	3.88
MPEG2_opt	7.89	29.51	3.99
compress	4.95	11.32	1.83
compress_opt	22.04	38.32	12.00
FFT	6.52	13.01	2.97
FFT_opt	7.06	12.80	3.01
DCT	4.76	7.54	0.46
DCT_opt	5.13	7.29	0.39
全体	7.91	38.32	6.74

5. おわりに

本稿ではマイクロプロセッサシステムの消費エネルギーをキャラクタライズする一手法を示した。今後は線形近似以外の方法も含めてモデルの精度向上に関する検討を行う予定である。キャラクタライズフローの自動化も今後の課題の一つである。

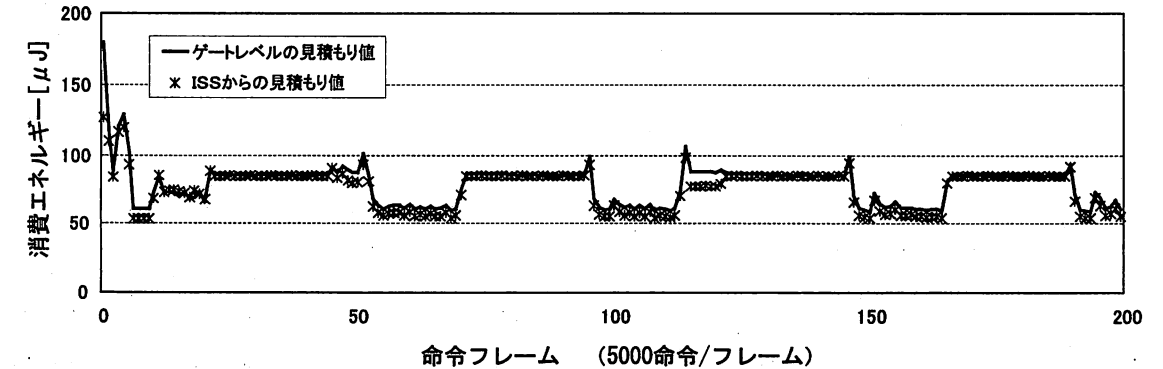


図 4 JPEG エンコードの実験結果

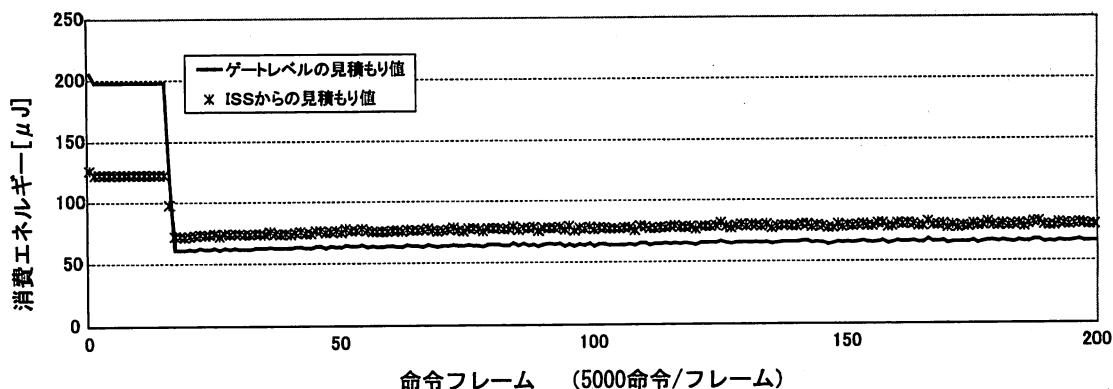


図 5 最適化オプションを付けてコンパイルした compress の実験結果

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を通し、ケイデンス株式会社、シノプシス株式会社、株式会社ルネサステクノロジ、株式会社日立製作所、及び、大日本印刷株式会社の協力で行われたものである。本研究の一部は、科学技術振興事業団 (JST) の戦略的推進事業 (CREST) 「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」によるものである。

参考文献

- [1] W. Ye, N. Vijaykrishnan, M. Kandemir and M.J. Irwin, "The Design and Use of SimplePower: A Cycle-Accurate Energy Estimation Tool", Proc. of 37th DAC, pp.340-345, June 2000.
- [2] J. Flinn and M. Satyanarayanan, "Powerscope: a Tool for Profiling the Energy Usage of Mobile Applications", in Proc. of the 2nd IEEE workshop on Mobile Computing Systems and Applications, pp.2-10, February 1999.
- [3] W. R. Hamburgen, D. A. Wallach, M. A. Viredaz, L. S. Brakmo, C. A. Waldspurger, J. F. Bartlett, T. Mann, and K. I. Farkas, "Itsy: Stretching the Bounds of Mobile Computing", IEEE Computer, vol. 34, pp.28-37, April 2001.
- [4] N. Chang, K. Kim, and H. G. Lee, "Cycle-Accurate Energy Consumption Measurement and Analysis: Case Study of ARM7TDMI", In Proc. of ISLPED, pp.185-190, August 2000.
- [5] V. Tiwari, S. Malik, and A. Wolfe, "Power Analysis of Embedded Software: A First Step towards Software Power Minimization", in Proc. of ICCAD, pp.384-390, November 1994.
- [6] D. Brooks, V. Tiwari, and M. Matonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimization", in Proc. of ISCA, pp.83-94, June, 2000.
- [7] M. T. C. Lee, V. Tiwari, S. Malik and M. Fujita, "Power Analysis and Low-Power Scheduling Techniques for Embedded DSP Software", in Proc. of the 8th ISSS, pp.110-115, Sept. 1995.
- [8] C. Brandolesse, W. Fornaciari, F. Salice, and D. Sciuto, "An Instruction-level Functionality-based Energy Estimation Model for 32-bit Microprocessors", in Proc. of DAC, pp.346-351, June 2000.
- [9] A. Sama, M. Balakrishnan, and J. F. M. Theeuwens, "Speeding Up Power Estimation of Embedded Software", in Proc. of ISLPED, pp.191-196, August 2000.
- [10] T. Sinha, and A. P. Chandrakasan, "JouleTrack - A Web Based Tool for Software Energy Profiling", in Proc. of DAC, pp.220-205, June 2001.
- [11] A. Sinha, N. Ickes, A. P. Chandrakasan, "Instruction level and operating system profiling for energy exposed software", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.11, no.6, pp.1044-1057, Dec. 2003.
- [12] M. Sami, D. Sciuto, C. Silvano and V. Zaccaria, "Instruction-Level Power Estimation for Embedded VLIW Cores" in Proc. of 8th Int'l Workshop on Hardware/Software Co-design, pp.34-38, May 2000.
- [13] Trimaran: <http://www.trimaran.org>.
- [14] C. T. Hsieh, L. S. Chen, M. Pedram, "Microprocessor Power Analysis by Labeled Simulation," in Proc. of the Conference on Design Automation and Test in Europe, pp.182-189, March 2001.
- [15] J. T. Russell and M. F. Jacome, "Software power estimation and optimization for high performance, 32-bit embedded processors," in Proc. of ICCD, pp.328-333, October 1998.
- [16] S. Steinke, M. Knauer, L. Wehmeyer, and P. Marwedel, "An Accurate and Fine Grain Instruction-Level Energy Model Supporting Software Optimizations", in Proc. of Int'l Workshop on Power And Timing Modeling, Optimization and Simulation, pp.3.2.1-3.2.10, September 2001.
- [17] T. Li and L. K. John, "Run-time Modeling and Estimation of Operating System Power Consumption", in Proc. of Int'l Conference on Measurements and Modeling of Computer Systems, pp.160-171, June 2003.
- [18] G. Contreras and M. Martonosi, "Power Reduction for Intel XScale® Processors Using Performance Monitoring Unit Events", in Proc. of ISLPED, pp.221-226, August 2005.
- [19] W. L. Bircher, M. Valluri, J. Law, and L. K. John, "Runtime Identification of Microprocessor Energy Saving Opportunities", in Proc. of ISLPED, pp.275-280, August 2005.
- [20] T. K. Tan, A. Raghunathan, G. Lakshminarayana, N. K. Jha, "High-level Software Energy Macro-modeling", in Proc. of DAC, pp.605-610, June 2001.
- [21] The Micron System Power Calculator, <http://www.micron.com/products/dram/syscalc.html>