

低消費電力化ソフトウェア技術

石原, 亨
九州大学 システムL S I 研究センター

富山, 宏之
名古屋大学 大学院情報科学研究科

<https://hdl.handle.net/2324/6254>

出版情報：組込みソフトウェアシンポジウム論文集. 2005 (12), pp.188-190, 2005-10. 情報処理学会ソフトウェア工学研究会

バージョン：

権利関係：ここに掲載した著作物の利用に関する注意 本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。

低消費電力化ソフトウェア技術

石原亨†

†九州大学 システムLSI研究センター

富山宏之‡

‡名古屋大学 大学院情報科学研究科

アブストラクト 消費電力の問題は組込みシステム設計において非常に深刻である。しかし、ソフトウェア設計における低消費電力化への意識はそれほど高くない。本稿ではソフトウェアによる低消費電力化技術をいくつか紹介し、その重要性に関して議論する。

Software Techniques for Low Power Embedded Systems

Tohru Ishihara†

†System LSI Research Center, Kyushu University

Hiroyuki Tomiyama‡

‡Graduate School of Information Science, Nagoya University

Abstract Power consumption has been a critical issue in the embedded system design. However, embedded software has been traditionally designed without taking care of the power consumption issues. This paper addresses several software techniques for reducing the power consumption and discusses their importance in the embedded system design.

1. はじめに

世界のIT産業の主力は、パーソナルコンピュータからデジタル情報家電にシフトしつつある。日本は、情報家電分野における世界市場のおよそ50%を占めているため、世界的なデジタル情報家電の隆盛と共に日本における半導体消費量は増大し、2004年には米国を抜いて世界最大の半導体消費国となった。

一方で集積回路の微細化に伴い、ハードウェア設計は複雑化し、ハードウェアの設計コスト、検証コスト、テストコストおよびマスクコストが急激に上昇している。アプリケーションごとにハードウェアを一から作り直すアプローチはコストの観点から見合わなくなりつつある。情報家電は製品寿命が短く、競争が激しい。このような状況で収益を上げるためには、付加価値の高い新製品を短期間で開発することが重要である。即ち、少量多品種生産に短納期、低コストで対応できる柔軟なアプローチとしてソフトウェアベースのアプローチに対する期待がますます高まっている。

デジタル情報家電に代表される組込みシステムは典型的には10mW~10W程度の電力を消費する。仮に、日本人が一人あたり一つの情報家電製品を使用したとすると、日本だけで最大130万キロワットにも及ぶ電力を消費することになる。これは、美浜原子力発電所1号機約4基分に相当する。デジタル情報家電の分野で日本が世界をリードし続けるためには、これらの製品を低い消費電力で実現する技術が鍵となる。

2. 電力消費のメカニズム

ソフトウェア自体は電力を消費しない。従って、低消費電力化ソフトウェア技術を議論するとき、ハードウェアが電力を消費するメカニズムを理解しておくことが重要である。本稿では、今日主流となっているCMOS集積回路をベースに議論する。CMOS集積回路の消費電力 P は式(1)で近似することができる[1][2]。

$$P = \sum_{g \in G} \{P_D(g) + P_L(g)\} \\ = \sum_{g \in G} \{SA(g) \cdot CL(g) \cdot V_{DD}(g)^2 + P_L(g)\} \dots \dots (1)$$

$P_D(g)$: ゲート g がスイッチする時に消費する電力

$P_L(g)$: ゲート g がスイッチしない時に消費する電力

$SA(g)$: ゲート g の単位時間当たりのスイッチング回数

$CL(g)$: ゲート g の負荷容量

$V_{DD}(g)$: ゲート g の電源電圧

$V_{th}(g)$: ゲート g の閾値電圧

$V_T(g)$: ゲート g の熱電圧(常温で25mV、温度に比例)

$$P_L(g) = I_L(g) \cdot V_{DD}(g), \quad I_L(g) \propto \exp\left(\frac{-V_{th}(g)}{\alpha \cdot V_T(g)}\right)$$

また、タスク i の実行時に消費されるエネルギーは、式(2)によって近似できる。

$$E = \int_0^{t_i} P dt \dots \dots \dots (2)$$

t_i : タスク i の実行時間

ゲートの負荷容量と熱電圧はソフトウェアから制御できない。また、ゲートの電源電圧と閾値電圧はソフトウェアの振る舞いに依存しないが、ソフトウェアから制御することが可能である。ソフトウェアによって電力を削減するためには、大きく分けて次の4つの方法が考えられる。

1. ゲートのスイッチング回数を削減する
2. ゲートの電源電圧を下げる（電源のシャットオフを含む）
3. ゲートの閾値電圧を上げる
4. タスクの実行時間(実行サイクル数)を減らす

ここで、ゲートのスイッチングとは、ゲート素子の論理値が0から1または1から0に変化することを意味する。CMOS集積回路はゲートがスイッチする時に消費する電力が支配的である点の特徴である。

3. ソフトウェア向け電力モデリング

ソフトウェアはハードウェアがなるべく電力を消費しないように振舞うことが重要である。そのためには、ソフトウェアの動作に関連付けてハードウェアの消費電力をモデル化することが必要になる。モデル化は、多くの場合、ハードウェアモジュールの消費電力をその動作モードとモード切り替えごとに定義することにより行われる。ハードウェアモジュールとは、ゲート素子や演算器あるいはキャッシュメモリなどを指す。動作モードとは、実行モード、待機モード、スリープモードなどである。消費電力または消費エネルギーの見積もりは、ターゲットアプリケーションを実行した時の各ハードウェアモジュールにおける各動作モードの実行時間とモードが切り替わる回数をシミュレーションまたは静的解析手法により見積もることによって行われる[3]。

文献[4]ではCISCプロセッサ486DX2を用いて、各命令を実行した際の電流を測定し、次のような結果を報告している。(a)命令により電流が異なる、(b)アドレッシングモードにより電流が異なる、(c)命令オペランド(メモリやレジスタ)の内容は影響を与えない、(d)前後の命令間の影響は無視できる(電流が15mA程度)。従って、CISCプロセッサでは命令ごとに消費電力を定義しておくことによって、コンパイラはコード生成を行う際、消費電力が小さくなるように命令を選択することができる。

4. コンパイル技術

実行時間を削減すれば、エネルギー消費も削減される。コンパイル技術に関しては、実行時間の削減以外の最適化技術は効果が小さいことが多い。しかし、ハードウェアを変更することなく低電力化が実現できるため、将来的には重要になると考える。これまでに提案された低電力指向コンパイル技術を紹介する。

Cold Scheduling[5]はスイッチング確率の最小化を

目的とした命令スケジューリング手法である。命令間のハミング距離(論理値が反転するビットの数)が最小になるように命令を並べ替える。キャッシュミス時に主記憶からキャッシュに命令を転送する際の電力を低減する命令スケジューリング手法も提案されている[6]。メモリバスでのスイッチング回数が最小となるように、命令を並び変える。DSPを対象とした最適化は文献[7]で提案されている。各命令の消費電力を測定しテーブルに持っておき、コード生成を行う際、消費電力が少なくなるように、命令を選択する。文献[8]では、レジスタ割当ての変更により電力を削減する手法が紹介されている。命令コード中のオペランド(レジスタアドレス)部分のスイッチングが減少するように、レジスタを変数に割り付ける。アドレスバスでのスイッチングを減少させることによって、オフチップ・ドライバやデコーダ回路の電力を削減する手法は、文献[9]で提案されている。信号処理アプリケーション(画像や音声など)はメモリアクセスが多く、規則的なアクセスパターンを示すため、コンパイル時に、メモリアドレスバスのスイッチング回数が最小になるように配列を主記憶に割り当てることができる。キャッシュミス回数を削減することにより主記憶へのアクセス回数を削減し、オフチップバスや主記憶で消費される電力を削減する手法は数多く提案されている[10-13]。一般にオフチップへのアクセスは非常に大きな電力を消費するため、キャッシュミス回数の削減は非常に効果的な低消費電力化手法の一つである。

5. アプリケーション設計技術

計算のアルゴリズムを変更することにより消費電力を削減することができる。文献[14]では、多項式の計算や、連立方程式の計算で使用される演算数を削減する手法が提案されている。計算の品質と消費エネルギーのトレードオフを図ることも可能である。例えば、画像処理のアプリケーションで、多少画質が低下しても構わないのであれば、画質への影響が小さい演算を省略することにより電力を削減することができる[15]。アプリケーションプログラムからプロセッサの電源電圧を制御することにより電力を削減する手法が文献[16]で紹介されている。また、文献[17]では頻繁に実行されるループ内で使用される別々の配列変数同士がキャッシュコンフリクトを起こさないように配列を主記憶上に分割して配置する手法が提案されている。アプリケーション設計技術は電力削減効果が非常に大きいものが多い。

6. OSによる電力管理技術

OSによる電力管理技術には大きく分けてシャットダウンとスローダウンがある。シャットダウンは電源供給を止めるのに対し、スローダウンはクロック周波数と電源電圧を低減する。いずれの場合も、システムの各コンポ

一ネットの使用時間、頻度を予想することが重要である。シャットダウンとスローダウンでは、理想的にはスローダウンの方がより低電力である[18]。シャットダウンの基本コンセプトは、アイドル時間がある一定値を超えたら、スリープ状態に遷移することである。しかし、スリープになるまでのアイドル状態では電力が無駄となっている。よって、高精度な予想が必要であり、その1つの方法は、アイドル状態の履歴に基づいて決定する方法である[19]。スローダウンの基本コンセプトは、あるタスクのデッドラインが t である時、そのタスクが時刻 t ぴったりに終了するようにクロック周波数と電源電圧を下げることである。CMOS 集積回路の遅延は電源電圧にほぼ比例するため、クロック周波数を半分にすれば電源電圧を半分にしてもプロセッサは正常に動作する。この考え方を利用したタスクスケジューリング手法は数多く提案されているが、最初のモデルは文献[20]で提案された。消費電力は電源電圧の 2 乗に比例するため、低電圧化は最も効果的な低電力化手法である。

7. おわりに

本稿では、ハードウェアが電力を消費しないようにするソフトウェア技術についていくつか紹介した。本稿で紹介した技術以外にも、リーク電流を削減するソフトウェア技術やチップの発熱を抑える技術なども数多く提案されている。今後はハードウェア設計コストの上昇に伴って、ソフトウェアによる製品の差別化がますます重要になり、結果的にソフトウェアによる低消費電力化は組込みシステム設計の一つの鍵になると考えられる。

謝辞

本研究の一部は、科学技術振興事業団(JST)の戦略的創造研究推進事業(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」および科学研究費補助金(No.16700058)の支援によるものである。

参考文献

- [1] N. Weste and K. Eshraghian, "Principles of CMOS VLSI design", Addison-Wesley, 1993.
- [2] L. Yan, J. Luo, and N. K. Jha, "Joint Dynamic Voltage Scaling and Adaptive Body Biasing for Heterogeneous Distributed Real-Time Embedded Systems", IEEE Trans. on CAD, vol.24, no.7, pp.1030-1041, July 2005.
- [3] T. K. Tan, A. Raghunathan, and N. K. Jha, "High-level Software Energy Macro-modeling" in Proc. DAC, pp. 605-610, Jun. 2001.
- [4] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: a first step towards software power minimization", IEEE Trans. on VLSI, vol. 2, no. 4, pp. 437-445, Dec. 1994.
- [5] C. L. Su, C. Y. Tsui, and A. M. Despain, "Low power architecture design and compilation technique for high-performance processors," in Proc. COMPCON, pp. 209-214, Feb. 1994.
- [6] H. Tomiyama, T. Ishihara, A. Inoue, and H. Yasuura, "Instruction scheduling to reduce switching activity of off-chip buses for low-power systems with caches," IEICE Trans. Fundamentals, vol. E81-A, no. 12, pp. 2621-2629, Dec. 1998.
- [7] M. T. C. Lee, V. Tiwari, S. Malik, and M. Fujita, "Power analysis and minimization techniques for embedded DSP software," IEEE Trans. on VLSI, vol. 5, no. 1, pp. 123-135, Mar. 1997.
- [8] H. Mehta, R. M. Owens, M. J. Irwin, R. Chen, D. Ghosh, "Techniques for low energy software," in Proc. ISLPED, Aug. 1997, pp. 72-75.
- [9] P. Panda and N. Dutt, "Low-power memory mapping through reducing address bus activity," IEEE Trans. on VLSI, vol. 7, no. 3, pp. 309-320, Sep. 1999.
- [10] S. McFarling, "Program Optimization for Instruction Caches", In Proc. of ASPLOS, pp.183-191, Apr. 1989.
- [11] H. Tomiyama and H. Yasuura, "Code placement techniques for cache miss rate reduction," ACM TODAES, vol. 2, no. 4, pp. 410-429, Oct. 1997.
- [12] P. Panda, N. Dutt, and A. Nicolau, "Memory data organization for improved cache performance in embedded processor applications", ACM TODAES, vol. 2, no. 4, pp. 384-409, Oct. 1997.
- [13] A. H. Hashemi, D. R. Kaeli, and B. Calder, "Efficient Procedure Mapping Using Cache Line Coloring", in Proc. PLDI, pp.171-182, Jun. 1997.
- [14] A. Hosangadi, F. Fallah, and R. Kastner, "Energy Efficient Hardware Synthesis of Polynomial Expressions", in Proc. Int'l Symp. on VLSI Design, pp.653-658, Jan. 2005.
- [15] A. Sinha, A. Wang, and A. P. Chandrakasan, "Algorithmic transforms for efficient energy scalable computation," in Proc. ISLPED, pp. 31-36, Aug. 2000.
- [16] S. Lee and T. Sakurai, "Run-time voltage hopping for low-power real-time systems," in Proc. ASPDAC, pp. 381-386, Jan. 2000.
- [17] C. Kulkarni, C. Ghez, M. Miranda, F. Catthoor, H. De Man, "Cache Conscious Data Layout Organization for Conflict Miss Reduction in Embedded Multimedia Applications," Proc. of DATE 2001, pp.686-691, March 2001.
- [18] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in Proc. ISLPED, pp. 197-202, Aug. 1998.
- [19] C. Hwang and A. Wu, "A predictive system shutdown method for energy saving of event-driven computation," in Proc. DAC, pp. 28-32, Nov. 1997.
- [20] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in Proc. IEEE Annual Foundation of Computer Science, 1995, pp. 374-382.