

組込みシステム向け DVFS アルゴリズムのリアルタイム OS への 実装および比較評価

Implementations and Comparative Evaluation of DVFS algorithm on the Embedded RTOS

岩田 淳*
Atsushi Iwata

高瀬 英希*
Hideki Takase

高木 一義*
Kazuyoshi Takagi

高木 直史*
Naofumi Takagi

1. はじめに

組込みシステムの消費エネルギーを削減することは、近年の重要な課題となっている。組込みシステムの消費エネルギー削減の有効な技術として、動的電圧・周波数制御 (Dynamic Voltage and Frequency Scaling, DVFS) が注目されている。DVFS とは、プロセッサの供給電圧および動作周波数を適切に制御してタスクを実行することによって、消費エネルギーの削減を狙う技術のことである。

これまでに、様々な DVFS アルゴリズムが提案されてきた。しかし、既存の DVFS アルゴリズムの多くは、理想的な環境を前提とした提案および評価にとどまっており、実際の組込みシステムの環境は考慮されていない。

そこで、本研究では、既存の DVFS アルゴリズムを実際の組込みシステムの環境を考慮した上でリアルタイム OS 上に実装し、それらの有効性を比較評価する。実装および評価対象の DVFS アルゴリズムは、組込みシステムで一般に用いられ、固定優先度方式のうちで最適なタスクスケジューリング方式である Rate-Monotonic Scheduling (RMS) 方式 [1] における DVFS アルゴリズムを用いる。比較評価にあたっては、メモリ量、タスク集合実行時のアルゴリズムの実行時間、および、タスク集合実行時の消費エネルギーを用いて、組込みシステムにおける DVFS アルゴリズムの有効性を定量的に議論する。

2. 動的電圧・周波数制御 (DVFS)

組込みシステムの消費エネルギーは、システム全体の中でプロセッサの占める割合が大きい。プロセッサに一般的に用いられている CMOS 回路の消費エネルギーは、供給電圧の 2 乗に比例すると近似できる [2]。ゆえに、DVFS によってプロセッサの供給電圧と動作周波数を下げてタスクを実行することで、消費エネルギーの削減が可能となる。ただし、組込みシステムにおいては、全てのタスクの実行をそれぞれの決められたデッドラインまでに完了するというデッドライン制約を保証する必要がある。デッドライン制約を保証する範囲内でタスクの実行を伸ばすことができる時間のことをスラック時間という。

図 1 におけるタスク τ で、DVFS の適用例を説明する。周波数が f の場合、時間 T で実行完了し、スラック時間 S が生じるとする。 τ に対して実行時間が $T+S$ となるように周波数を f から $f \frac{T}{T+S}$ に引き下げ、それに合わせてプロセッサへの供給電圧も引き下げることができ

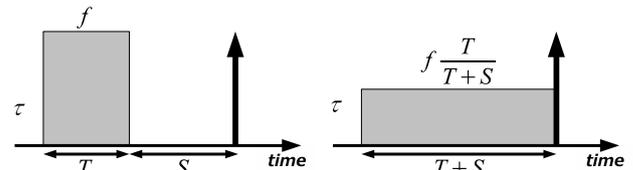


図 1: DVFS 適用前および適用後

る。DVFS の適用によってプロセッサの消費エネルギーは、もとの消費エネルギーの $\left(\frac{T}{T+S}\right)^2$ 倍となり、消費エネルギーが削減できる。

本研究では、評価対象の DVFS アルゴリズムとして、Work-Demand Analysis (WDA)[3]、Effective-WDA[4]、Leveling Frequency with Slack Time (LFST)[5] および LFST and NTA (LF-NTA)[5] を取り上げる。

WDA は、タスクがディスパッチされる時に、スラック時間を算出して DVFS を適用する。スラック時間は、ディスパッチされるタスクの残り最悪実行時間、そのタスクより優先度の高いタスク群からの干渉時間、優先度の低いタスク群からの干渉時間、および、そのタスクのデッドラインから求められる。WDA では、算出されたスラック時間はただちに全て活用して動作周波数が決定される。

Effective-WDA は、WDA におけるスラック時間算出手法の改善を狙ったアルゴリズムである。具体的には、まず、高優先度タスク群からの干渉時間の算出値の悲観性を改善している。さらに、低優先度タスク群からの干渉時間を算出する際に再帰計算を排除している。

システムの消費エネルギーは、各タスクの動作周波数が平準化されるようにスラック時間を配分することで最適になることが知られている。LFST は、これに着目し、Effective-WDA において算出された周波数とタスクの平均実行時間により算出された平準化周波数を選択するアルゴリズムである。

LFST では、平準化周波数を選択した場合、全てのスラック時間を有効に活用できないことがある。LF-NTA は、Stretching-to-NTA[6] 手法の考え方を導入することでこの問題を改善する。

3. リアルタイム OS への実装

本研究では、DVFS アルゴリズムを組込みシステムのシングルプロセッサ向けのリアルタイム OS である TOPPERS/ASP 上へ実装する。TOPPERS/ASP は、数多くの組込みシステムに採用されているリアルタイム OS である μ ITRON 仕様に準拠した、オープンソースのリ

*京都大学, Kyoto University

表 1: OMAP3530 の電圧および周波数の設定値

周波数 [MHz]	電圧 [V]
720	1.35
550	1.27
500	1.2
250	1.06
125	0.985

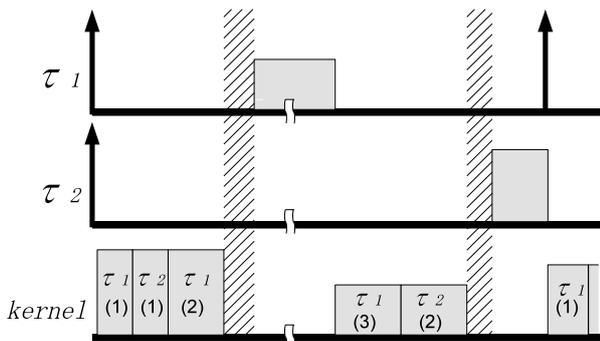


図 2: DVFS の実行時間および切替時間オーバーヘッド

アルタイム OS である。

第 2 章で説明した DVFS アルゴリズムは、アルゴリズムの実行時間や電圧および周波数の切替時間を考慮しない理想的な環境を前提としている。しかし、実際の組み込みシステムではそれらのオーバーヘッドがある。このため、既存の DVFS アルゴリズムをそのままリアルタイム OS 上に実装してしまうと、デッドライン制約は保証できない。また、設定できる周波数および電圧は、連続値、または、隣り合う数値の差が等しくなるように並べられた離散値とすることが前提となっている。

本章では、TOPPERS/ASP 上に DVFS アルゴリズムを実装する際に、考慮すべき点とそれに対応した実装手法を説明する。

3.1. 電圧および周波数の設定値

実際のプロセッサの供給電圧および動作周波数は、有限個の設定値にしか切り替えることができない。例として、表 1 は、DVFS に対応したマイコンボード BeagleBoard[8] に搭載されたプロセッサ OMAP3530[9] で切り替えることができる電圧および周波数の設定値を示している。

本研究では、DVFS アルゴリズムで算出した周波数以上で、かつ、設定値の周波数の中で最も値が小さいものを選択するように実装することで、この問題を解決する。

3.2. DVFS アルゴリズムの処理にかかる実行時間

DVFS アルゴリズムの処理に、実行時間のオーバーヘッドが生じる。具体的には、タスクのリリース時、ディスパッチ時および実行完了時において、周波数算出やアルゴリズムに必要な管理情報の更新といった処理が行われる。図 2 の (1)、(2) および (3) のカーネルの処理はそれぞれタスクのリリース時の処理、ディスパッチ時の処理および実行完了時の処理を示している。これらの処理は、オーバーヘッドとして実行サイクル数が必要となる。例として、タスクのディスパッチ時の実行サイクル数が 2000

とすると、表 1 の周波数の最小値での実行した際の実行時間は $16\mu\text{s}$ と計算できる。

これらの実行時間オーバーヘッドを考慮するために、まず、各処理の実行サイクル数をシミュレータで測定する。計測した実行サイクル数の最大値を用いて、各タイミングにおける実行時間のオーバーヘッドを以下のように考慮して、デッドライン制約を保証した実装とする。

(1) タスクのリリース時の実行時間オーバーヘッド

実行サイクル数を周波数設定値の最小値で実行した際の実行時間に換算した時間と高優先度タスクのリリース回数の積を、高優先度タスク群からの干渉時間に加算する。

(2) タスクのディスパッチ時の実行時間オーバーヘッド

実行サイクル数を周波数設定値の最小値で実行した際の実行時間に換算した時間と高優先度タスクのディスパッチされる最大回数の積を、高優先度タスク群からの干渉時間に加算する。また、実行サイクル数を現在の周波数で実行した際の実行時間に換算した時間をスラック時間から減算する。

(3) タスクの実行完了時の実行時間オーバーヘッド

実行サイクル数を周波数設定値の最大値で実行した際の実行時間に換算した時間を、タスクの最悪実行時間に加算する。

3.3. 電圧および周波数の切替時間オーバーヘッド

DVFS を適用し、電圧および周波数を切り替える際に、その切替時間のためのオーバーヘッドが生じる。図 2 の斜線部分が、この切替時間のオーバーヘッドを表している。

文献 [7] によれば、プロセッサへの供給電圧を V_{DD1} から V_{DD2} に切り替える際に生じる DC-DC コンバータの切替時間 t_{TRAN} は

$$t_{TRAN} = \frac{2 \cdot C}{I_{MAX}} \cdot |V_{DD2} - V_{DD1}| \quad (1)$$

となる。ここで、 C は DC-DC コンバータのもつコンデンサの電荷容量、 I_{MAX} は DC-DC コンバータの出力電流の最大値である。例として、 $C = 100\mu\text{F}$ 、 $I_{MAX} = 1\text{A}$ の DC-DC コンバータが表 1 の電圧の設定値の最大値から最小値に切り替えるのにかかる時間は $73\mu\text{s}$ と計算できる。

切替時間オーバーヘッドへの対応として、事前に切替前後の周波数に対する切替時間を格納したテーブルを用意する。周波数の切替が行われる際に、そのテーブルを参照して切替時間の最大値をスラック時間から減算し、さらに、その値から全切替時間中での最大値を減算する。これによって、対象タスクによる切替時間と次にディスパッチされるタスクの切替時間を考慮でき、デッドライン制約が保証できる。

4. 比較評価

4.1. 評価環境

本研究における評価環境として、ARM920T プロセッサのソフトウェア・シミュレータである SkyEye[10] を用いた。DVFS アルゴリズムが適用できるよう、プログラム実行中に周波数が変更できるように SkyEye を機能拡張した。SkyEye において設定できる周波数と電圧の設定値は、表 1 に示した 5 段階の値を用いた。電圧の切替

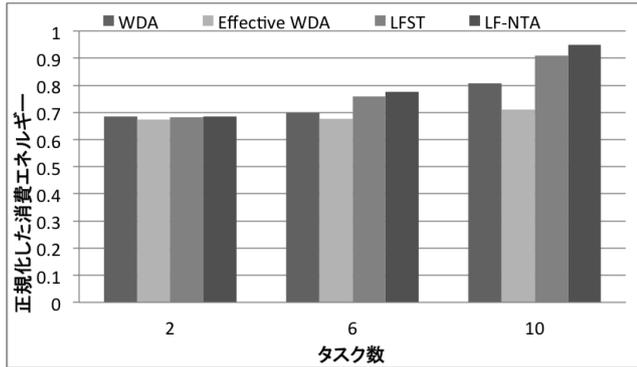


図 3: タスク数に対する消費エネルギー

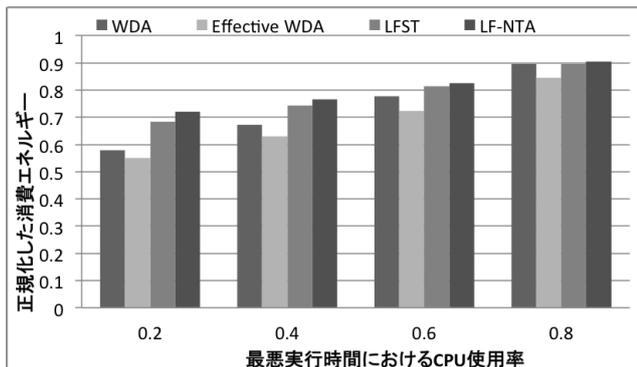


図 4: 最悪実行時間における CPU 使用率に対する消費エネルギー

時間オーバーヘッドは式 1 で、 $C = 100\mu\text{F}$ 、 $I_{MAX} = 1\text{A}$ とした際の値を用いた。

TOPPERS/ASP カーネルに DVFS を実装しないものと、2 章で挙げた 4 個の DVFS アルゴリズムを実装したものにおけるタスク集合のハイパーピリオドまでの消費エネルギーを算出した。消費エネルギーは、プロセッサおよび DC-DC コンバータのそれぞれの消費エネルギーの和として求めた。前者は、供給電圧の 2 乗と実行サイクル数の積に比例するとして求めた。後者は、電圧切替時に消費されるものであり、文献 [7] のモデルより求めた。なお、システム休止時の消費エネルギーは無視できるものとした。また、各 DVFS アルゴリズムの実装についてのメモリ量も測定した。

本評価では、タスク集合は自作したスクリプトによって生成されるタスク集合を使用した。スクリプトに与える条件は、以下の通りである。

- タスク数：2, 6, 10
- 最悪実行時間における CPU 使用率：0.2, 0.4, 0.6, 0.8
- 各タスクの周期 period：[1,100]ms のランダム値
- 各タスクの最悪実行時間 wcet：[1,period×1000] μs のランダム値

ここで、各タスクの実行時間は、[1,wcet] μs でリリース毎に変動するものとする。各タスク数、最悪実行時間における CPU 使用率に対し、50 個のタスク集合を生成し、それらに対して実験を行った。

表 2: タスク数に対するリリース時の実行サイクル数

DVFS アルゴリズム	タスク数 2 個	タスク数 6 個	タスク数 10 個
WDA	508	542	576
Effective WDA	510	543	576
LFST	546	625	695
LF-NTA	546	627	704

表 3: タスク数に対するディスパッチ時の実行サイクル数

DVFS アルゴリズム	タスク数 2 個	タスク数 6 個	タスク数 10 個
WDA	846	1359	2000
Effective WDA	736	859	972
LFST	978	1530	2118
LF-NTA	1033	1808	2457

表 4: タスク数に対する実行完了時の実行サイクル数

DVFS アルゴリズム	タスク数 2 個	タスク数 6 個	タスク数 10 個
WDA	534	2119	6117
Effective WDA	564	2619	4513
LFST	821	2885	4788
LF-NTA	826	3116	4858

4.2. 評価結果

タスク数および最悪実行時間における CPU 使用率に対して、DVFS アルゴリズムの機能がないものを 1 とし正規化した消費エネルギーを、それぞれ図 3 および図 4 に示す。図 3 では、評価対象の DVFS アルゴリズムの多くはタスクの個数が多いほど削減効果が小さくなるのに対し、Effective-WDA にはそのような傾向は見られなかった。図 4 では、最悪実行時間における CPU 使用率が高いほど、どの DVFS アルゴリズムも消費エネルギー削減効果が小さくなった。そして、Effective-WDA はどの条件でも、最も消費エネルギー削減効果が大きかった。

タスクの個数に対する、タスクのリリース時、ディスパッチ時および実行完了時における実行サイクル数をそれぞれ表 2、表 3 および表 4 に示す。タスクの個数が多いほど各タイミングの実行サイクル数は大きくなり、結果として実行時間オーバーヘッドが大きくなった。Effective-WDA はディスパッチ時の実行サイクル数が他の DVFS アルゴリズムより小さく、さらに、タスクの個数が増えることによる影響も小さい。

タスク数 6 のタスク集合について、DVFS なしのもの、WDA のもの、Effective WDA のもの、LFST のもの、LF-NTA のものの ROM 量および RAM 量を表 5 および表 6 に示す。なお、表 5 および表 6 の値は、TOPPERS/ASP だけでなく、タスク集合で構成されるアプリケーションのメモリ量も含んでいる。ROM および RAM いずれも、WDA および Effective-WDA の増加率は、LFST および LF-NTA より小さくなった。

4.3. 考察

ディスパッチ時の処理の実行時間オーバーヘッドに着目する。3.2 節で述べた通り、ディスパッチ時のオーバーヘッドを考慮するため、計測した実行サイクル数を最小の周

表 5: タスク数が6のROMの比較評価結果

DVFS アルゴリズム	ROM[byte]	ROM 増加率 [%]
機能なし	55000	0
WDA	57968	5.40
Effective WDA	57984	5.43
LFST	59464	8.12
LF-NTA	60142	9.35

表 6: タスク数が6のRAMの比較評価結果

DVFS アルゴリズム	RAM[byte]	RAM 増加率 [%]
機能なし	34048	0
WDA	34128	0.23
Effective WDA	34128	0.23
LFST	34176	0.38
LF-NTA	34176	0.38

波数設定値で動作させた際の実行時間に換算し、その時間を高優先度タスク群のディスパッチされうる最大の回数分を、干渉時間に加算するように実装した。つまり、干渉時間が大きいほどスラック時間が小さくなり、アルゴリズムにより算出される周波数値は大きくなり、消費エネルギー削減効果は小さくなる。そのため、ディスパッチ時の実行サイクル数は消費エネルギー削減効果に大きな影響を与える。

LFST および LF-NTA は、文献 [5] の評価では、Effective-WDA よりも消費エネルギー削減効果が大きかった。しかし、本評価では、タスク集合の条件に関わらず削減効果が小さくなった。これは、LFST および LF-NTA のディスパッチ時の実行サイクル数が、Effective-WDA のものより大きいことに起因する。このため、理想的な環境の消費エネルギー削減効果を実現できなかった。また、タスク数 10 個のときの WDA と LFST を比較すると、ディスパッチ時の実行サイクル数の差に対し、消費エネルギーの削減効率の差が大きい。これは、タスク数が多いほど、LFST で算出される平準化周波数が悲観的な値となってしまうことに起因する。

Effective-WDA は、他の DVFS アルゴリズムよりディスパッチ時の実行サイクル数が小さいため、消費エネルギー削減効果が最も大きい。また、メモリ量の増加率も WDA と並んで、実際に実装するのに影響のないレベルである。以上のことから、実際の組込みシステムの環境を考慮したとき、Effective-WDA は本研究で取り上げた DVFS アルゴリズムのうちで最も有効であると結論付けられた。

5. まとめ

本研究では、RMS 方式の DVFS アルゴリズムを実際の組込みシステムの環境を考慮した上でリアルタイム OS に実装し、比較評価した。DVFS アルゴリズムには、WDA、Effective-WDA、LFST および LF-NTA を取り上げた。評価指標として、消費エネルギー、アルゴリズムの実行サイクル数およびメモリ量を測定し、その結果、Effective-WDA が実際の組込みシステムの環境を考慮した DVFS アルゴリズムとして適していると結論付けられた。

今後の方針として、DVFS に対応した実際のプロセッサ上で DVFS アルゴリズムを比較評価することが挙げられる。

謝辞 本研究を進めるにあたり貴重なご助言をいただいた名古屋大学の曾剛講師、並びに、名古屋大学の松原豊特任助教に深く感謝致します。

参考文献

- [1] 白川洋充, 竹垣盛一: リアルタイムシステムとその応用, 朝倉書店 (2001) .
- [2] Hu, S. X. and Gang, Q.: Fundamental of Power-aware Scheduling, *Designing Embedded Processors: A Low Power Perspective*, Springer, chap. 9, pp. 219–230 (2007).
- [3] Kim, W., Kim, J. and Min, S. L.: Dynamic Voltage Scaling Algorithm for Fixed-Priority Real-Time Systems Using Work-Demand Analysis, *Proceedings of International Symposium On Low Power Electronics and Design*, pp. 396–401 (2003).
- [4] 三輪遼平, 高瀬英希, 曾剛, 富山宏之, 高田広章: 組込みシステムにおける低消費エネルギー志向の効率的なスラック時間の導出, 情報処理学会研究報告, Vol. 2010, No. 11, pp.1–8 (2010) .
- [5] 三輪遼平, 高瀬英希, 曾剛, 高田広章: 組込みシステムにおける消費エネルギーの削減のためのスラック時間の活用, 情報処理学会研究報告, Vol. 2012, No. 13, pp. 1-6 (2012) .
- [6] Shin, Y., Choi, K. and Sakurai, T.: Power optimization of real-time embedded systems on variable speed processors, *Proceedings of the International Conference on Computer Aided Design*, pp.365–368 (2000).
- [7] Burd, T. D. and Brodersen, R. W.: Design Issues for Dynamic Voltage Scaling, *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 9–14 (2000).
- [8] Beagleboard.org, BeagleBoard System Reference Manual Rev C4 [online], http://beagleboard.org/static/BBSRM_latest.pdf.
- [9] Texas Instruments, OMAP3530 data sheet [online], <http://www.ti.com/lit/ds/symlink/omap3530.pdf>.
- [10] Kang, S., Wang, H., Chen, Y., Wang, X. and Dai, Y.: Skyeye: An Instruction Simulator with Energy Awareness, *Proceedings of the International Conference on Embedded Software and Systems*, pp. 456–461 (2004).