

リアルタイム電圧最適化による マルチタスク処理の消費エネルギー最小化

塩見 準^{1,a)} 石原 亨² 小野寺 秀俊¹

概要：電源電圧 (V_{DD}) としきい値電圧 (V_{th}) の動的制御は、プロセッサの消費エネルギーを削減する有効な手法の 1 つである。本稿では、リアルタイム組み込みシステムの消費エネルギーを最小化する電圧制御手法を提案する。プロセッサに与えられたデッドラインのもと、処理タスクに応じてプロセッサの V_{DD} , V_{th} およびタスクの実行時間をリアルタイムに最適化することで、デッドラインを守りながらマルチタスク処理時に発生する消費エネルギーを最小化する。本稿では、プロセッサの活性化率がタスクに依存して変化する事実に注目し、タスクの実行サイクル数だけでなく活性化率を考慮してタスクの実行時間を最適化することで、消費エネルギーを効果的に削減できることを示す。タスクの活性化率を考慮しない従来のスケジューリング手法と比較して、平均 24.3% 消費エネルギーを削減できることを示す。

1. はじめに

LSI のエネルギー効率改善は、高度に発展した今日の情報化社会における最重要課題の 1 つである。エネルギー効率の改善は、二酸化炭素排出量の削減や情報通信機器のバッテリー駆動時間改善などに繋がる。多くの LSI は自動車などのリアルタイムシステムに実装される。リアルタイムシステムに組み込まれた LSI は設定された時間以内にタスクを処理する必要がある。デッドラインを守りながら省エネルギー動作を実現することが強く求められている。

電源電圧 (V_{DD}) としきい値電圧 (V_{th}) の動的制御は、集積回路の消費エネルギーを削減する最も有効な手法の 1 つである [1–4]。回路の総消費エネルギーは容量成分の充放電に起因する動的消費エネルギーとサブスレッショルドリーク電流に起因する静的消費エネルギーの和で表現される。 V_{DD} を下げることで 2 乗のオーダーで動的消費エネルギーを削減でき、 V_{th} を増大させることで静的消費エネルギーを指数関数的に削減できる。回路の動作速度 (i.e., クロック周波数) は V_{DD} や V_{th} の強い関数であるため、特定の動作周波数を実現する V_{DD} と V_{th} の組が無数に存在する。その結果、与えられた動作周波数を満たしながら回路の総消費エネルギーを満たす V_{DD} と V_{th} の組がただ一つ存在する。プロセッサが最適な V_{DD} と V_{th} の組で動作しない場合、大きな消費エネルギーのオーバーヘッドが発生する [4]。

本稿では、リアルタイム組み込みシステムの消費エネルギー

削減に向けたランタイム電圧制御技術を提案する。入力データ列をソートするタスクのように、タスク処理に必要な最悪ケースの実行サイクル数を容易に見積もれる一方、入力データに応じた実際の実行サイクル数を見積もることは困難である。もし最悪ケースより早めにプロセッサがタスク処理を完了すると、発生したタイミングスラックを用いて後続タスクに対してさらに電圧スケールリングを適用できる。以上の事実は、先行タスクの実行時間に応じて、後続タスクに割り当てる V_{DD} , V_{th} および実行時間を適応的に最適化することが重要であることを示している。本稿ではまず、単純な解析モデルに基づき、後続タスクの実行時間を実行時に最適化する手法を提案する。従来のスケジューリング手法では、実行サイクル数に比例した実行時間を後続タスクに割り当てる。本稿では、タスクの実行サイクル数と回路の活性化率に応じてタスクの実行時間を最適化することで、従来手法よりさらに消費エネルギーを改善できることを示す。ここで、活性化率は回路中のノードが 0 から 1 に遷移する平均の確率であり、回路の活性化率は実行タスクに応じて変動する。例えば頻繁にストールするタスクと演算器を頻繁に活性化するタスクでは、回路の活性化率に大きな差が発生する。本稿では、2 つのタスクが同じ実行サイクル数を有していても、活性化率に応じて異なる実行時間 (i.e., クロック周波数) を割り当てることで消費エネルギーを削減できることを示す。

V_{DD} のみ制御する従来の Dynamic Voltage and Frequency Scaling (DVFS) では、動作速度と V_{DD} が 1 対 1 対応する。他方、 V_{DD} と V_{th} の同時最適化を行う場合、最適な V_{DD} と V_{th} の組が動作速度だけでなく、回路の活性化率とチップ温度に強く依存しており [5]、最適化問題の複雑

¹ 京都大学大学院情報学研究科

² 名古屋大学大学院情報学研究科

^{a)} shiomi-jun@i.kyoto-u.ac.jp

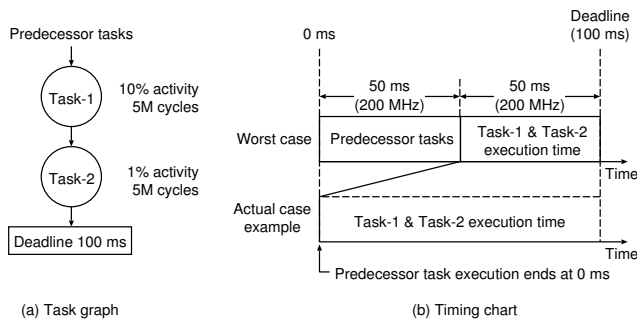


図 1 リアルタイム組み込みシステムで処理されるタスクの例。

さを増大させている。本稿では、上記パラメータが動的に変化する状況において、先行タスクに起因するタイミングスラックを活用し、後続タスクに割り当てる V_{DD} , V_{th} および実行時間を実行時に最適化することで消費エネルギーを最小化する。

本稿の成果: リアルタイム組み込みシステムの消費エネルギー最小化に向けた、単純なランタイム電圧最適化技術を提案する。対象とするシステムでは、異なる活性化率を有する複数のタスクが特定のデッドラインのもと処理される。各タスクでワーストケース実行時間より早く処理が終了する可能性がある。タイミングスラックを活用し、提案手法は単純な性能予測モデルに基づき後続タスクに割り当てる V_{DD} , V_{th} および実行時間を実行時に最適化する。

本稿の構成を以下に示す。第 2 章では動的電圧制御の重要性を示す Motivational Example および関連研究を述べる。第 3 章では、リアルタイムにプロセッサの V_{DD} , V_{th} およびクロック周波数を最適化する手法を提案する。第 4 章で提案手法の検証を行う。第 5 章で結論を述べる。

2. リアルタイム電圧制御による消費エネルギー削減

2.1 Motivational Example

図 1 (a) に示すアプリケーションを実行するプロセッサの消費エネルギーを電圧制御により削減する状況を考える。プロセッサは複数のタスクで構成されるアプリケーションを実行している。アプリケーションは“Predecessor tasks”と書かれたタスクと活性化率が 10 倍異なる 2 個のタスクを処理する。タスク毎に活性化率が変化する現象は一般的なプロセッサで発生する。例えば、Task-2 がオフチップメモリを頻繁にアクセスするタスクであると仮定する。プロセッサは Task-2 処理時に、アクセス時間の長いオフチップメモリにアクセスするため頻繁にストールする。その結果、Task-2 の活性化率が Task-1 より小さくなる。

本稿では、しきい値電圧が 0.3 V であるプロセッサの性能が、定格電圧 1.0 V において以下であると仮定する。

(P1) 200 MHz の動作速度

(P2) 90 mW の動的消費電力 (活性化率 10%)

(P3) 10 mW の静的消費電力

プロセッサの性能モデルの詳細を 4.1 節で述べる。

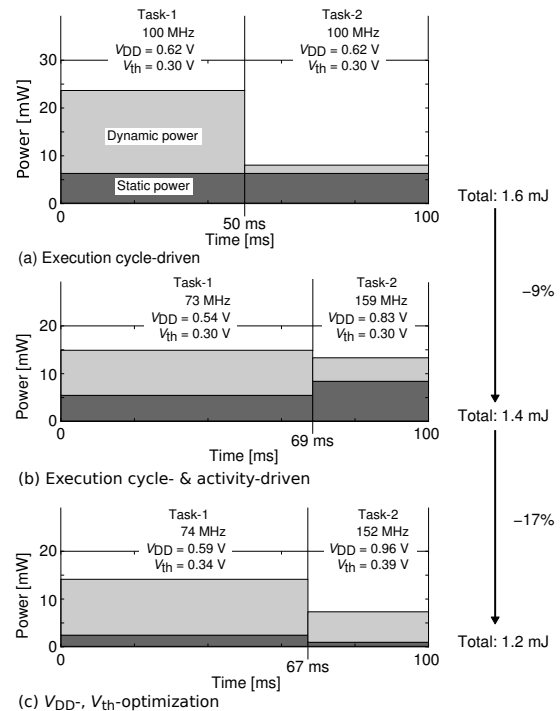


図 2 電圧制御による消費エネルギー削減。“Dynamic power”: 動的消費電力。“Static power”: 静的消費電力。

図 1 (b) に、プロセッサのタイミングチャートを示す。“Worst case”に、タスクが最悪ケースの実行サイクル数で処理される場合のタイミングチャートを示す。実際のタスク処理において、タスクは最悪ケースより少ないクロックサイクル数で処理される可能性がある。プロセッサが最悪ケースより早くタスク処理を完了する状況を考える。“Predecessor tasks”と書かれた先行タスクの処理が最悪のケースより早く終了し、プロセッサが時刻 0 ms において後続する 2 個のタスク処理に取りかけられる状況を“Actual case example”に示す。Task-1 および Task-2 に割り当て可能な時間が 50 ms から 100 ms へ倍増するため、実行時に電圧スケーリングを適用することでデッドラインを守りながら消費エネルギーをさらに削減できる。本稿では以下の 3 種類の方法を用いて消費エネルギー削減を行う。

(a) タスクの実行サイクル数に比例して実行時間を割り当て、 V_{DD} を最適化

(b) タスクの活性化率に応じて実行時間と V_{DD} を最適化

(c) V_{DD} , V_{th} , 実行時間をすべて最適化

2 つのタスクの実行サイクル数が同じであるため、ともに実行時間を倍増させることが最も自然なスケーリング方式である。従来のスケーリング方式では一般的に実行サイクル数に比例した時間割り当てが行われる。回路の動的消費エネルギーは V_{DD} の二乗に比例する一方、動作速度はオーバードライブ電圧 ($V_{DD} - V_{th}$) におよそ線形に比例する。したがって、図 1 (b) の例において、電源電圧を下げることで動的消費エネルギーをおよそ 4 分の 1 まで下げることができる。図 2 (a) は V_{DD} を制御し消費エネルギーを削減した結果である。Task-1 および Task-2 の活性化率が

互いに異なるため、タスク処理に必要な消費エネルギーが大きく異なる。例えば、図2(a)の例ではTask-1の消費エネルギーがTask-2の消費エネルギーの約3倍大きい。したがって、2つのタスクの実行サイクル数が同じ場合でも、Task-1にTask-2より長い実行時間を割り当てることで消費エネルギーを削減できる。

タスクの実行時間を最適化した結果を図2(b)に示す。実行時間と V_{DD} の最適化により、消費エネルギーを9%削減できる。しかし、 V_{DD} の制御は静的消費エネルギーを十分に削減できない。特にTask-2に関し、静的消費エネルギーが消費エネルギーの半分以上を占めている。 V_{th} の制御により静的消費エネルギーを指数関数的に削減できるため、 V_{DD} だけでなく、 V_{th} の同時最適化が省エネルギー動作に向けて重要な役割を果たす。

V_{DD} , V_{th} そして実行時間をタスク毎に最適化した結果を図2(c)に示す。 V_{th} 制御により、総消費エネルギーを17%削減可能である。リアルタイム組み込みシステムでは、活性化率およびタスク実行サイクルが動的に変化するため、以上のMotivational exampleは、 V_{DD} , V_{th} そして実行時間をタスクに応じて動的に最適化することが重要であることを示しているが、上記3パラメータを実行時に最適化する手法は自明ではない。また、チップ温度に応じて、静的な消費エネルギーが指数関数的に変化する。したがって、チップ温度のような環境パラメータが動的に変化する状況においても上記3パラメータを実行時に最適化する手法が必要である。

2.2 関連研究

リアルタイムシステムの省エネルギー動作に向けた V_{DD} 制御技術に関する初期の研究の1つはLeeとSakuraiによる[6]。文献[6]では、リアルタイム処理が求められる動画処理プロセッサに対し、DVFSを適用することで消費エネルギーを削減する技術が提案されている。動画処理プロセッサでは、フレームレートにより決定されるデッドラインを守るよう画像処理タスクが実行される。タスクがデッドラインより早く終了した場合、タイミングスラックを用いて次フレームの処理を行う。タイミングスラックの分だけクロック周波数および V_{DD} を下げることでデッドラインを守りながら消費エネルギーを削減できる。文献[7]は、複数のタスクから構成されるアプリケーションを対象としており、実行時に発生するタイミングスラックを活用し、クロック周波数や V_{DD} を最適化するスケジューラを提案している。提案スケジューラによりデッドラインを超過することなくリアルタイムシステムの消費エネルギーを削減できることを示している。しかし、以上の技術は V_{DD} および V_{th} の同時最適化は行っておらず、活性化率のタスク依存性も注目していない。

V_{DD} と V_{th} の動的制御技術は20年以上前から重点的に研究されている[1-3]。文献[1]では異種のプロセッサが相互結合した分散リアルタイムシステムの V_{DD} , V_{th} および

タスク実行時間を最適化し、システムの総消費エネルギーを削減する手法が提案されている。しかし、文献[1]では複雑な数値解析をアプリケーション実行前に行う必要がある。先行タスクにより発生するタイミングスラックや、チップ温度、活性化率が動的に変化する状況では、文献[1]の手法を用いて実行時に V_{DD} , V_{th} およびタスク実行時間を最適化することは困難である。

本稿では、既存研究と異なり、単純なモデル式に基づき、ランタイムに V_{DD} , V_{th} およびタスク実行時間を最適化する手法を提案する。

3. リアルタイム組み込みシステムの消費エネルギーを最小化する実行時電圧制御手法

本章では、タスクに割り当てる V_{DD} , V_{th} および実行時間を実行時に最適化し、リアルタイム組み込みシステムの消費エネルギーを最小化する電圧制御アルゴリズムを提案する。3.1節で提案手法の概要を述べる。文献[1]で提案されているEnergy gradientの概念が V_{DD} , V_{th} および実行時間の最適化に重要な役割を果たす。問題定義を3.3節で述べる前に、3.2節でEnergy gradientを閉形式で近似する解析モデルを提案する。Energy gradientモデルの導出後、3.3で問題定義を行い、3.4節で提案制御手法を述べる。

3.1 提案手法の概要

図3に対象とするリアルタイム組み込みシステムを示す。対象システムでは、プロセッサは J 個のタスクをデッドライン T_d までに処理しなければならない。最悪ケースのクロックサイクル数でタスク処理される場合のタイミングチャートを図3の“Worst case”に示す。Task- i ($i = 1, 2, 3, \dots, J$)には最悪ケースの実行クロックサイクル数 $N_{w,i}$ が存在する。 $N_{w,i}$ はコンパイル時等、タスク処理以前に推定可能であると仮定する。一般的に、タスク処理に要した実際のクロックサイクル数 $N_{a,i}$ を事前に求めることは困難である。図3の“Actual case”にプロセッサがTask- $(i-1)$ の処理を完了し、Task- i の処理に取り掛かる直前の状況を示す。最悪ケースより早めにタスク処理が終了する場合、タイミングスラックを活用し、“Voltage scaling for subsequent tasks”に示すように、後続タスク(Task- i 以降)に電圧スケール技術を実行時に適用する。以下に示すように、本稿では処理タスクごとに異なる V_{DD} , V_{th} , 実行時間を割り当て、消費エネルギーの最小化を実行時に行う。

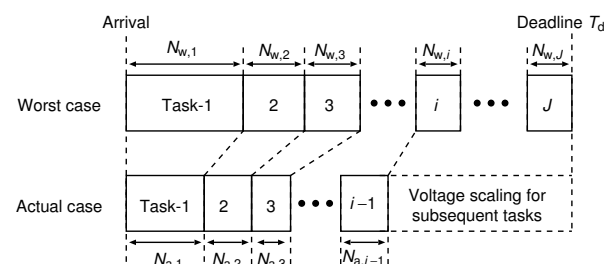


図3 対象とするリアルタイム組み込みシステム。

- (1) 先行タスク処理に要した実際の実行時間から、後続タスクに割り当て可能な実行時間を計算
- (2) 後続タスクに割り当てる実行時間を最適化
- (3) 最適化された実行時間のもと、消費エネルギーを最小化する V_{DD} , V_{th} を各タスクへ割り当て

以上のフローの中で、ステップ (2) が最も困難な課題である。本稿では次節で述べる Energy gradient がタスク実行時間の最適化に重要な役割を果たす。次節でまず Energy gradient の近似解析モデルの提案を行う。

3.2 実行時電圧制御を可能にする Energy Gradient 近似モデル

3.2.1 Energy gradient

与えられた時間制約以内にタスクをプロセッサがタスクを処理しなければならない状況を考える。一般的に、時間制約が緩和されるとプロセッサのクロック周波数 (f) を低く設定できる。 f の低下にともない、 V_{DD} と V_{th} を最適化することでタスク処理に必要な消費エネルギーを削減できる。与えられたタスク (以降、Task- i とする) に対し、Energy gradient (G_i) を次のように定義する [1]。

- Energy gradient G_i : Task- i 処理のために割り当てた時間が微量緩和されたときに、 f , V_{DD} および V_{th} の最適化で削減可能な Task- i の消費エネルギーの割合、処理時間を ΔT_i 緩和することで削減可能な消費エネルギーを $-\Delta e_i$ と表記すると、 G_i は (1) で定式化される：

$$G_i \simeq -\frac{\Delta e_i}{\Delta T_i}. \quad (1)$$

Energy gradient を用いることで、タスク処理時間を最適化できる [1]。例えば、図 4 (a) に示す 2 個のタスクからなるシステムを考える。デッドライン T_d のもと、タスクの処理時間 T_1, T_2 ($= T_d - T_1$) を最適化し、消費エネルギーを最小化する。Energy gradient の定義に基づくと、 T_1 を ΔT だけ緩和すると、Task-1 の処理に必要な消費エネルギーを $-G_1 \Delta T$ だけ削減できる。一方、 T_2 が ΔT だけ厳しくなるため、Task-2 の処理に必要な消費エネルギーが $G_2 \Delta T$ だけ増大する。したがって、実行時間調整により削減可能な総消費エネルギーは $-(G_1 - G_2) \Delta T$ である。 $G_1 > G_2$ が成立する場合、時間制約の調整により総消費エネルギーを削減できる。図 4 (b) に示すように、タスクに

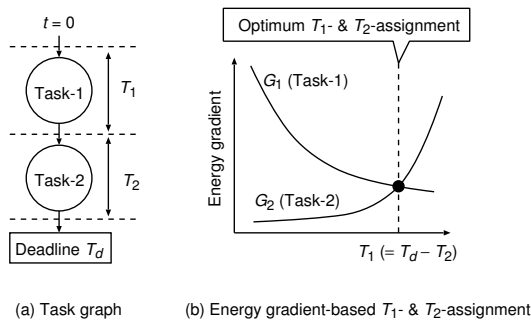


図 4 Energy gradient によるタスク処理時間最適化。

割り当てる遅延を大きくすると単調に Energy gradient が減少する [1]。以上の事実をまとめると、消費エネルギー最小化のための必要十分条件は $G_1 = G_2$ である。したがって、2 個のタスクの Energy gradient が同じ値になるよう遅延を割り当てると、消費エネルギーが最小化される。

本稿では、同様のアプローチを用いてタスクに割り当てる実行時間を最適化する。しかし、文献 [1] では、Energy gradient を用いたクロック周波数最適化が複雑な数値計算を経て実現されているため、実行時最適化に適していない。次節では、LSI の性能モデルを用い、Energy gradient の単純な近似モデルを導出し、実行時最適化を実現する。

3.2.2 Energy gradient 近似モデルの導出

V_{DD} が V_{th} より十分大きい場合、プロセッサの動作速度 (f_i) はオーバードライブ電圧 (i.e., $V_{DD} - V_{th}$) に概ね線形に比例する：

$$f_i \propto V_{DD,i} - V_{th,i}. \quad (2)$$

ここで、 $V_{DD,i}$ と $V_{th,i}$ は与えられた実行時間 T_i のもと、Task- i の消費エネルギーを最小化するよう最適化された電源電圧としきい値電圧である。Task- i の処理に必要なクロックサイクル数が N_i であると仮定すると、 T_i と N_i の間で以下の関係が成立する：

$$T_i \propto \frac{N_i}{V_{DD,i} - V_{th,i}}. \quad (3)$$

Task- i の処理に必要な消費エネルギー ($e_{t,i}$) は、動的消費エネルギー ($e_{d,i}$) と静的消費エネルギー ($e_{s,i}$) の和でモデル化できる：

$$e_{t,i} = e_{d,i} + e_{s,i} = \frac{e_{d,i} + e_{s,i}}{e_{d,i}} e_{d,i} \left(= \frac{e_{d,i}}{R_d} \right). \quad (4)$$

ここで、 R_d は $e_{t,i}$ に対する $e_{d,i}$ の割合である。与えられた時間制約 T_i のもと、消費エネルギーを最小化するよう $V_{DD,i}$ と $V_{th,i}$ を最適化すると、 R_d が 30% 付近の定数になる [5, 8]。 $e_{d,i}$ はプロセッサの活性化率 (a_i)、 N_i および $V_{DD,i}^2$ に比例するため、以上の事実をまとめると以下が成立する：

$$e_{t,i} \propto \frac{a_i N_i}{R_d} V_{DD,i}^2 \propto a_i N_i V_{DD,i}^2. \quad (5)$$

次に、時間制約 T_i がわずかに ΔT_i だけ緩和され、 $V_{DD,i}$ と $V_{th,i}$ が再び最適化された状況を考える。ここで、時間制約緩和前と比較した $V_{DD,i}$ と $V_{th,i}$ の変化量をそれぞれ $\Delta V_{DD,i}$, $\Delta V_{th,i}$ とする。消費エネルギー削減量 ($\Delta e_{t,i}$) と ΔT_i は、 $V_{DD,i}$ と $V_{th,i}$ に関して (3) と (5) を全微分することでモデル化できる：

$$\Delta e_{t,i} \propto a_i N_i V_{DD,i} \Delta V_{DD,i}, \quad (6)$$

$$\Delta T_i \propto \frac{N_i \Delta V_{DD,i}}{(V_{DD,i} - V_{th,i})^2} \left(1 - \frac{\Delta V_{th,i}}{\Delta V_{DD,i}} \right). \quad (7)$$

$V_{DD,i} \gg V_{th,i}$ が成立する場合、 $\Delta V_{th,i}$ は $\Delta V_{DD,i}$ より十分小さくなる [8]。したがって、(7) にあらわれる $\frac{\Delta V_{th,i}}{\Delta V_{DD,i}}$ の

項を無視し, (6) の $(V_{DD,i} - V_{th,i})$ を $V_{DD,i}$ で近似すると, (1), (3), (6), (7) から以下の Energy gradient モデルを導出できる:

$$G_i \simeq -\frac{\Delta e_{t,i}}{\Delta T_i} = C a_i \left(\frac{N_i}{T_i} \right)^3. \quad (8)$$

ここで, C は製造プロセスや回路構造, チップ温度等に依存する定数である. 本稿では, (8) に示したモデルに基づき, スーパーレッシュヨルド領域 ($V_{DD} \gg V_{th}$) のみならず, 近年省エネルギー動作を実現する電圧領域として注目されているニアスレッシュヨルド領域 (V_{DD} が V_{th} よりわずかに大きい電圧領域) で動作するプロセッサのクロック周波数最適化を行う.

Energy gradient の近似モデルを構築する際のポイントは, 最適な実行時間割り当てから実行割り当てを微量量変更したときの消費エネルギーのオーバーヘッドが比較的小さいことである. 例えば, 図 2 (c) に示した例で, Task-1, Task-2 に割り当てる実行時間が最適値から $\pm 10\%$ 変動する状況を考える. 実行時間割り当てが最適値から離れるため消費エネルギーのオーバーヘッドが発生するが, その値は高々 2.5% である.

3.3 問題定義

3.1 節の図 3 に示すように, J 個のタスクからなるリアルタイム組込みシステムを対象とする. 各タスクが処理されると, 先行タスクの処理に要した実際の処理時間を基に, 後続タスクに割り当てる V_{DD} , V_{th} およびクロック周波数を最適化する. Task- $(i-1)$ までの処理が終わり, プロセッサが Task- i 以降の処理を行う際の最適化問題は以下のように定式化される.

$$\begin{aligned} \min \quad & \sum_{j=i}^J (e_{t,j} + e_{tr}), \\ \text{s.t.} \quad & \sum_{j=i}^J T_{w,j} = T_d - \sum_{k=1}^{i-1} T_{a,k}, \\ & T_{w,j} = N_{w,j} d_j + d_{tr}, \\ & V_{DD,j}, V_{th,j} \in \mathbb{R}. \end{aligned} \quad (9)$$

ここで, $V_{DD,j}$ および $V_{th,j}$ はそれぞれ Task- j に割り当てる電源電圧およびしきい値電圧である. $e_{t,j}$ は Task- j の処理により発生する消費エネルギーである. d_j は Task- j 処理時のプロセッサのクリティカルパス遅延である. $T_{a,k}$ と $T_{w,j}$ はそれぞれ Task- k の実際の処理時間および Task- j の最悪の処理時間である. ここで, i 番目以降のタスクの実際の処理サイクル数を見積もることは困難であることに注意. 結果として, リアルタイムシステムにおいて遅延制約を満たすため i 番目以降のタスクは最悪ケースの実行サイクル数で処理される状況を想定しなければならない. e_{tr} と d_{tr} は, それぞれタスク間で電圧を切り替える際に必要な消費エネルギーおよび遅延のオーバーヘッドである. 本稿では, 電圧切り替時においてプロセッサは動作停止状態であり, タスク処理を行わない状況を想定する. $e_{t,j}$ は動的消費エネルギーと ($e_{d,j}$) と静的消費エネルギー ($e_{s,j}$) の和である. それぞれの消費エネルギーモデルを以下に示す:

$$e_{t,j} = e_{d,j} + e_{s,j}, \quad (10)$$

$$e_{d,j} = k_1 a_j N_{w,j} V_{DD,j}^2, \quad (11)$$

$$e_{s,j} = k_2 T_{w,j} V_{DD,j} \exp \left(-\frac{V_{th,j} + \kappa \Delta T_c}{n_s} \right). \quad (12)$$

ここで, n_s は理想係数と熱電圧の積である. したがって n_s はチップ温度 (T_c) に比例する. a_j は Task- j の活性化率である. k_1 と k_2 は回路構造や製造プロセスに依存するフィッティング係数である. ΔT_c は, 室温と比較した時のチップ温度の変化量である. $\kappa \Delta T_c$ は温度変化に起因するしきい値電圧の変化量を意味する. 電源電圧がしきい値電圧より十分に高い場合, α 乗則を用いて精度良くクリティカルパス遅延 d_j を近似できる [9]:

$$d_j = \frac{k_3 V_{DD,j}}{(V_{DD,j} - V_{th,j} - \kappa \Delta T_c)^{\alpha_j}}. \quad (13)$$

ここで, α_j と k_3 はフィッティング係数である. α_j は通常 1 と 2 の間の値を持つ.

トランジスタのバックゲート電圧を調節することでプロセッサのしきい値電圧を調節することができる. 文献 [10] で, 単一電源電圧で動作する低消費電力基板バイアス生成回路が提案されている. 本稿では, 上記に類する基板バイアス生成回路を活用し, プロセッサの消費電力と比較して無視できるオーバーヘッドでしきい値電圧調整ができると仮定する. また, 回路シミュレーションや出荷前のテスト等の方法で, 以下のパラメータが事前に正確に求められることを仮定する.

- 回路の物理特性: k_1, k_2, k_3, κ , 理想係数
- タスクの特性: $a_i, N_{w,i}$

また, [11] 等の方法で, 消費エネルギーオーバーヘッドなくタスク処理と独立して T_c を定期的かつ正確に測定可能であることを仮定する.

3.4 Energy Gradient に基づく電圧最適化手法

3.4.1 フローチャート

提案手法のフローチャートを図 5 に示す. $N_{w,i}, a_i, J, T_d$ からなるタスクに関連する情報が入力値として入力される. 提案手法では, 以下の 3 つのステップを実行することで消費エネルギーの最小化を実現する. Step1 では, 提案する G_i を用いて $T_{w,i}$ が最適化され, 結果として $T_{w,i}^*$ を出

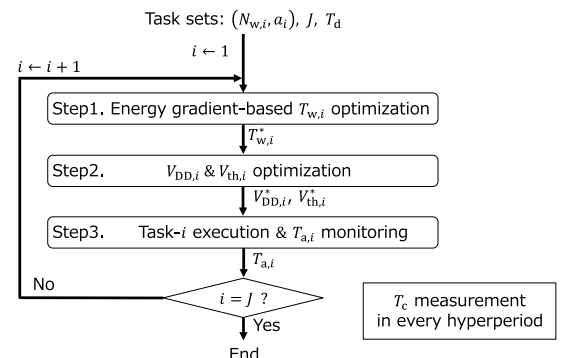


図 5 提案手法のフローチャート.

力する。ここで、図5にあらわれる添字“*”は最適化されたパラメータであることを示す。与えられた遅延制約 $T_{w,i}^*$ のもと、 $V_{DD,i}$ と $V_{th,i}$ が Step2 で最適化される。Step3 で Task- i が処理された後、後続タスクの最適化のために $T_{a,i}$ がモニタされる。チップ温度の動的変動を評価するために、チップ温度 T_c がタスク処理と独立して測定される。Step1 および Step2 の詳細をそれぞれ 3.4.2 節および 3.4.3 で述べる。

3.4.2 Energy Gradient に基づくタイミング最適化

3.2 節で述べた通り、2 個のタスクの Energy gradient が同じ値を持つとき、2 個のタスクの消費エネルギーを最小化できる。一般的に、処理タスクが 3 個以上の場合でも同様の事実が成立し、すべてのタスクの Energy gradient が同じ値を持つことが消費エネルギー最小化のための必要十分条件である。したがって、 $j, l \in \{i, i+1, \dots, J\}$ とすると、以下が最適化問題 (9) と等価な問題である：

$$G_j = G_l. \quad (14)$$

(8) を (14) に代入すると、次式が成立する：

$$C_j' T_{w,j} = C_l' T_{w,l}. \quad (15)$$

ここで、 $C_j' = a_j^{-\frac{1}{3}} N_{w,j}^{-1}$ である。(9) の制約条件から、次式が成立しなければならない：

$$\sum_{j=i}^J T_{w,j} = T_d - \sum_{k=1}^{i-1} T_{a,k}. \quad (16)$$

(15) と (16) は $(J-i+1)$ 個の変数 $T_{w,i}, T_{w,i+1}, \dots, T_{w,J}$ からなる連立 1 次方程式であるため、解 $T_{w,i}^*$ を閉形式で簡単に求めることができる：

$$T_{w,i}^* = \frac{T_d - \sum_{k=1}^{i-1} T_{a,k}}{C_j' \sum_{n=i}^J \frac{1}{C_n'}}. \quad (17)$$

一般的に、タスクの実行時間の上限/下限はプロセッサの動作速度の下限/上限により決定される。(17) で得られた Task- n に関する解 $T_{w,n}^*$ がプロセッサの動作速度の限界を超える場合、 $T_{w,n}^*$ の値は上限または下限に変更される。(16) のリアルタイムタイミング制約を満たすため、Task- n を除外して (15), (16) から最適な時間割り当てが再度求められる。

3.4.3 電源電圧としきい値電圧の最適化

与えられた遅延制約 $T_{w,j}^*$ のもと、Task- j の消費エネルギーを最小化する $V_{DD,j}$ と $V_{th,j}$ が存在する。タスクの活性化率 a_j と T_c が明らかになると、文献 [5] の方法により閉形式で最適な電圧組を近似的に求められる。文献 [5] の方法と等価な表記を次式に示す：

$$V_{DD,j}^* = \frac{V_{th,j}^* + \chi - K_1}{1 - K_1}, \quad (18)$$

$$V_{th,j}^* = -n_s \ln \left(\frac{a_j N_{w,j} n_s K_2}{T_{w,j}} \right) - \kappa \Delta T_c. \quad (19)$$

ここで、 $K_1 = \frac{\chi}{\alpha}$, $K_2 = \frac{2k_1}{k_2} \cdot \frac{\alpha}{\alpha - \chi}$, $\chi = \sqrt{\frac{k_3 N_{w,j}}{T_{w,j} - d_{tr}}}$ であ

る。また、前述のとおり n_s は T_c に比例するパラメータである。なお、(18), (19) により決定した $V_{DD,j}^*$ と $V_{th,j}^*$ において、 $T_{w,j}^*$ と $N_{w,j}$ により決定されるクロック周波数で回路動作させると、近似誤差に起因してタイミング故障が発生する可能性がある。本稿ではタイミング故障が発生した場合、文献 [12, 13] で示された方法のように、クリティカルパスレプリカの伝搬遅延を評価しながら基板バイアス生成回路で V_{th} を微調整し対応する。なお、本稿ではこの際に発生する遅延オーバーヘッドは電圧切り替え時間の中に含まれるものと仮定する。

4. 提案手法の検証

4.1 セットアップ

本章では、(10), (11), (12), (13) で示した性能モデルで動作する仮想的なプロセッサを想定する。各パラメータの値を表1に示す。 d_{tr} と e_{tr} の値は文献 [14] に基づく。表1に示すパラメータのもとでは、プロセッサは 2.1 節の (P1), (P2), (P3) に示す性能を実現する。また、プロセッサの最大動作速度、最小動作速度としてそれぞれ 220 MHz, 40 MHz を仮定する。上記動作速度はオーバードライブ電圧がそれぞれおよそ 0.8 V, 0.15 V の電圧条件に対応する。また、 V_{DD} , V_{th} を 10 mV 刻みで変更できると仮定する。

本稿の処理対象は以下に示す 4 個のタスクからなるアプリケーションである。

- デッドライン: 500 ms
- 表2に示す実行サイクル数、活性化率を有している
- タスクが処理される順番はランダムである

上記タスクから構成されるアプリケーションを 100 回評価し、提案手法による消費エネルギーの平均/最大/最小削減率を評価する。なお、プロセッサが 4 個すべてのタスクをデッドラインより早く処理した場合は動作を終了し、パワーゲーティング等の技術によりアイドル期間の消費エネルギーのオーバーヘッドが発生しないと仮定する。

本稿では、以下の 4 通りの電圧制御シナリオにより提案手法の効果を確認する。

表 1 仮想プロセッサのパラメータ。

パラメータ	値
α	1.5
理想係数	1.5
k_1	4.5 nJV ⁻²
k_2	22.9 Js ⁻¹ V ⁻¹
k_3	2.93 nsV ^{$\alpha-1$}
κ	-1 mVK ⁻¹
d_{tr}	150 μ s
e_{tr}	4 μ J

表 2 タスクの特性。

タスク番号	1	2	3	4
$N_{w,i}$	5×10^6	5×10^6	35×10^6	35×10^6
$N_{a,i}$	区間 $[0.4N_{w,i}, N_{w,i}]$ 上の一様分布			
a_i	10%	5%	0.2%	0.1%

表 3 各電圧制御シナリオにおける消費エネルギー最適化結果. *に対する注意: 削減率が正の値の場合, 提案手法導入により消費エネルギーが削減される. 負の値の場合, 提案手法により消費エネルギーのオーバーヘッドが発生する.

シナリオ		Proposed	Brute force	Uniform	DVFS
消費エネルギー [mJ]	最大	1.87	1.82	3.03	2.00
	最小	0.95	0.93	1.17	1.04
	平均	1.42	1.39	1.96	1.54
Proposed 導入による消費エネルギー削減率*	最大	-	-0.4%	45.3%	14.8%
	最小	-	-4.0%	-12.8%	5.2%
	平均	-	-2.1%	24.3%	7.8%

- **(Proposed):** 提案手法
- **(Brute force):** (9) を総当たりで解く方法. 提案手法に対する厳密解に対応.
- **(Uniform):** タスクの実行時間を実行サイクル数に比例して決定する方法. 実行時間決定後, 消費エネルギーを最小化するように V_{DD} , V_{th} を最適化.
- **(DVFS):** (9) を総当たりで解く方法. ただし, V_{DD} と実行時間のみ調整可能とする.

シナリオ (DVFS) では, V_{th} を 50 mV 刻みに調節して最も消費エネルギーが小さくなる電圧であった 0.4 V を V_{th} として用いる. タスクの活性化率を区別しない場合, シナリオ (Uniform) が消費エネルギーを最適化する解になる. 本稿では (Uniform) を既存研究の代表とする. このような仮定は [15] 等の文献で行われている.

4.2 実験結果

室温 (300 K) において各手法の評価を行った結果を表 3 に示す. “消費エネルギー [mJ]” は, 各シナリオで達成される消費エネルギーの最大/最小/平均を示しており, “Proposed 導入による消費エネルギー削減率*” は, 提案手法導入により達成可能な消費エネルギー削減率である. 削減率が正の値の場合, 当該シナリオと比較し, 提案手法により消費エネルギーを削減できることを意味する. 他方, 削減率が負の値の場合, 当該シナリオと比較し, 提案手法により消費エネルギーが増大することを意味する. “Brute force” と “Proposed” を比較すると, 消費エネルギーに換算して 4.0% 以内の精度で最適な V_{DD} , V_{th} , 実行時間を推定可能である. タスクの実行時間を実行サイクル数に比例して割り当てる “Uniform” と比較すると, 提案手法により最大 45.3%, 平均 24.3% の消費エネルギーを削減できる. これは 2.1 節で述べたように, 消費エネルギーを最小にする V_{DD} , V_{th} , 実行時間がタスクの活性化率に依存するためである. 他方, “Uniform” が提案手法より 12.8% 省エネルギーになる例も見られた. (9) に示すように, 提案手法はすべてのタスクが最悪ケースで実行されたときを想定して V_{DD} , V_{th} , 実行時間の最適化を行う. しかし, タスク処理に必要な実行サイクル数は最悪値と同じになるとは限らない. 以上が結果的に “Uniform” が提案手法に対して優位性を示す例が見られた原因と考えられる. “DVFS” との比較結果によると, 提案手法により最大 14.8%, 平均 7.8% 消

費エネルギーを削減できる. 2.1 節で示したように, 静的消費エネルギーの動的調節が省エネルギー化に重要な役割を果たすためである.

前述のシナリオから, チップ温度が室温から 30 K 上昇した状況を考える. チップ温度上昇により, サブスレッショルドリーク電流が指数関数的に増大し, 静的消費エネルギーが増加する. 結果としてタスクに割り当てる最適な V_{DD} , V_{th} , クロック周波数が変化する. 本稿では, 以下のシナリオを用いて, 動的な V_{DD} , V_{th} , クロック周波数の最適化の重要性を示す.

- **(Proposed 330 K):** 提案手法
- **(Proposed 300 K):** 300 K の状態において最適化された V_{DD} , V_{th} , クロック周波数をそのまま使用
- **(DVFS 330 K):** チップ温度の変化を考慮して V_{DD} , クロック周波数を最適化

シナリオ (DVFS 330 K) に関し, 300 K の際の実験と同じしきい値電圧 (0.4 V) で稼働する状況を仮定する.

表 4 に評価結果を示す. シナリオ “Proposed 300 K” と比較して, チップ温度の変化に応じて V_{DD} , V_{th} , タスク実行時間を動的に最適化することで静的消費エネルギーを削減し, 平均 21.1%, 最大 24.3% 消費エネルギーを削減できる. また, シナリオ “DVFS 330 K” では V_{DD} と実行時間のみ変更するため, 温度上昇により増大した静的消費エネルギーを効果的に削減できない. 結果として, 提案手法により最大 43.1%, 平均 31.6% 消費エネルギーを削減できる. 以上の事実は, チップ温度の動的な変化に応じて V_{DD} , V_{th} , タスク実行時間を最適化することが省エネルギー動作に重要であることを示している.

5. 結論

電源電圧 (V_{DD}) としきい値電圧 (V_{th}) の動的調整は消費エネルギーを効果的に削減できる最も有力な手法の 1 つである. 本稿では, リアルタイム組み込みシステムの消費エネルギー削減を目指し, 複数のタスクから構成されるアプリケーションを処理する際, タスク毎に設定する V_{DD} , V_{th} および実行時間を最適化する単純な電圧制御手法を提案した. 一般的に, タスクの処理は最悪ケースで予想される実行時間より早く終了する. また, 活性化率が異なるが実行サイクル数が同じ 2 つのタスクが連続実行される場合, 消費エネルギーを最小化する最適な実行時間が異なる. 以上

表 4 チップ温度が 30 K 上昇したときの消費エネルギー最適化結果. *に対する注意: 削減率が正の値の場合, 提案手法導入により消費エネルギーが削減される. 負の値の場合, 提案手法により消費エネルギーのオーバーヘッドが発生する.

シナリオ		Proposed 330 K	Proposed 300 K	DVFS 330 K
消費エネルギー [mJ]	最大	2.25	2.83	3.23
	最小	1.15	1.49	1.74
	平均	1.72	2.18	2.52
Proposed 330 K 導入による消費エネルギー削減率*	最大	-	24.3%	43.1%
	最小	-	18.5%	24.4%
	平均	-	21.1%	31.6%

の事実に基づき, プロセッサがマルチタスク処理を行う際, タスク処理の度に発生するタイミングスラックを活用し, 後続タスクに対して V_{DD} , V_{th} および実行時間を動的に最適化することで消費エネルギーをさらに削減できるが, その最適化手法は自明ではない. 本稿では, 単純な Energy gradient モデルを提案し, 当該モデルに基づきタスクに割り当てる最適な実行時間を閉形式関数で導出した. 導出した最適なクロック周波数に基づき, 最適な V_{DD} , V_{th} を閉形式関数で表現した. LSI の性能モデルを用いた実験の結果, タスクの活性化率を区別しない従来手法と比較し電圧制御手法と比べて平均 24.3% の消費エネルギー削減に成功した. また, チップ温度の変化に応じて V_{DD} , V_{th} およびタスク実行時間を動的に最適化することで, チップ温度を考慮せずプロセッサを稼働させ続ける状況と比較し, 平均 21.1% 消費エネルギーを削減できることを確認し, 動的に V_{DD} , V_{th} およびタスク実行時間を最適化する重要性を示した. 実際の商用プロセステクノロジーを用いた提案手法の検証や, プロセッサのスケジューラへの組み込みが今後の課題である.

謝辞 本研究は JSPS 科研費 (18H06462) による支援によって行われた.

参考文献

- [1] L. Yan, J. Luo, and N. Jha, "Joint Dynamic Voltage Scaling and Adaptive Body Biasing for Heterogeneous Distributed Real-Time Embedded Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 7, pp. 1030–1041, July 2005.
- [2] S. Martin, K. Flautner, T. Mudge, and D. Blaauw, "Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Lower Power Microprocessors under Dynamic Workloads," in *International Conference on Computer Aided Design*, Nov 2002, pp. 721–725.
- [3] A. Basu, S.-C. Lin, V. Wason, A. Mehrotra, and K. Banerjee, "Simultaneous Optimization of Supply and Threshold Voltages for Low-Power and High-Performance Circuits in the Leakage Dominant Era," in *Design Automation Conference*, July 2004, pp. 884–887.
- [4] S. Hokimoto, T. Ishihara, and H. Onodera, "Minimum Energy Point Tracking Using Combined Dynamic Voltage Scaling and Adaptive Body Biasing," in *International System-on-Chip Conference*, Sept 2016, pp. 1–6.
- [5] K. Nose and T. Sakurai, "Optimization of VDD and VTH for Low-power and High Speed Applications," in

- Asia and South Pacific Design Automation Conference*, Jan 2000, pp. 469–474.
- [6] S. Lee and T. Sakurai, "Run-time Power Control Scheme Using Software Feedback Loop for Low-power Real-time Application," in *Asia and South Pacific Design Automation Conference*. ACM, 2000, pp. 381–386.
- [7] P. Pillai and K. G. Shin, "Real-time Dynamic Voltage Scaling for Low-power Embedded Operating Systems," in *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, ser. SOSP '01. New York, NY, USA: ACM, 2001, pp. 89–102. [Online]. Available: <http://doi.acm.org/10.1145/502034.502044>
- [8] T. Takeshita, T. Ishihara, and H. Onodera, "Guidelines for Effective and Simplified Dynamic Supply and Threshold Voltage Scaling," in *International Symposium on VLSI Design, Automation and Test*, April 2016, pp. 1–4.
- [9] T. Sakurai and A. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr 1990.
- [10] N. Kamae, A. K. M. M. Islam, A. Tsuchiya, and H. Onodera, "A Body Bias Generator with Wide Supply-Range down to Threshold Voltage for Within-die Variability Compensation," in *Asian Solid-State Circuits Conference*, Nov 2014, pp. 53–56.
- [11] A. K. M. M. Islam, J. Shiomi, T. Ishihara, and H. Onodera, "Wide-Supply-Range All-Digital Leakage Variation Sensor for On-Chip Process and Temperature Monitoring," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 11, pp. 2475–2490, Nov 2015.
- [12] J. Park and J. A. Abraham, "A Fast, Accurate and Simple Critical Path Monitor for Improving Energy-delay Product in DVS systems," in *International Symposium on Low Power Electronics and Design*, Aug 2011, pp. 391–396.
- [13] K. A. Bowman, C. Tokunaga, J. W. Tschanz, A. Raychowdhury, M. M. Khellah, B. M. Geuskens, S. L. L. Lu, P. A. Aseron, T. Karnik, and V. K. De, "All-Digital Circuit-Level Dynamic Variation Monitor for Silicon Debug and Adaptive Clock Control," *IEEE Transactions on Circuits and Systems*, vol. 58, no. 9, pp. 2017–2025, Sept 2011.
- [14] N. Minallah, Y. ji Wang, J. sheng Xing, W. Nisar, and A. raza Kazmi, "Towards dynamic voltage scaling in real-time systems- a survey," *International Journal of Computer Sciences and Engineering Systems (IJCSES)*, vol. 1, no. 2, pp. 93–103, 2007.
- [15] F. Yao, A. Demers, and S. Shenker, "A Scheduling Model for Reduced CPU Energy," in *Proceedings of IEEE 36th Annual Foundations of Computer Science*, Oct 1995, pp. 374–382.