電源ノイズ削減のためのマルチコアプロセッサ向けクロックゲーティング機構の提案

川部 純[1,a)]　武内 良典[1,b)]　劉 載勲[1,c)]　今井 正治[1,d)]

概要：タスクの並列処理により高速化を達成するマルチコアプロセッサは消費電流の変化量が大きく，消費電流の変化が電源ノイズを大きくさせる原因となっている．電源ノイズはチップの誤動作を引き起こしプロセッサの信頼性を低下させる．本稿では消費電力の削減手法であるクロックゲーティングをコアに対して適用し，プロセッサの電流変化量を動的に抑制する機構を提案する．提案する機構により，プロセッサの性能低下を抑えつつ電流変化量を一定以下に保証する．実験により 4 コアマルチプロセッサの性能低下を最大 7%に抑えることを確認した．提案するクロックゲーティング機構により，性能低下が起きない条件下で，電流変化量を 2 ビット実装の場合 41.2%，3 ビット実装の場合 37.3%，4 ビット実装の場合 35.1%削減することを確認した．

An Efficient Clock-Gating Mechanism for Multi-Core Processor to reduce Power Supply Noise

Kawabe Jun[1,a)]　Takeuchi Yoshinori[1,b)]　Yu Jaehoon[1,c)]　Imai Masaharu[1,d)]

Abstract: Multi-core processors achieve high performance by parallel processing of tasks. However, the large amount of current change of multi-core processors makes power supply noise large. Power supply noise causes low reliability of the processor. This paper proposes a mechanism which dynamically suppresses the amount of current change of the processor by controlling the execution of cores using clock-gating. The mechanism guarantees the amount of current change under certain level keeping performance. Evaluation results show that the performance degradation of 4 core multi-processor was within 7% by proposed mechanism, and amount of current change was reduced 41.2% by 2-bit implementation, 37.3% by 3-bit implementation, 35.1% by 4-bit implementation, respectively.

1. 序論

システム高速化の手段としてマルチコアプロセッサを用いることが多くなってきている．マルチコアプロセッサでは複数のコアでタスクを並列実行することで高速化を達成するが，タスク実行の並列度の増加に伴い，プロセッサの消費電流の変動量が大きくなりやすい．電源配線上には寄生インダクタンスが存在する．インダクタの両端の電位差はインダクタを流れる電流の単位時間当たりの変化量に比例するため，大きな電流変化は大きな電圧降下を引き起こす．これを電源ノイズと呼ぶ．電源ノイズによるプロセッサ供給電圧の低下は素子遅延増加，ロジックの誤動作等の原因となり，プロセッサの信頼性を低下させる．近年ではプロセス技術の進歩によるプロセッサの低電圧動作や動作周波数の増大により，プロセッサの電流変動に伴う電源ノイズの影響が大きくなりやすい．電源ノイズは一般的にデカップリングコンデンサと呼ばれるコンデンサの挿入で対策される．電荷を保持するコンデンサの性質によりプロセッサ電圧の低下が抑制される．しかし設計時には最悪時の電源ノイズを想定する必要があるため，予想される電源ノイズが大きいほど大きな容量のデカップリングコンデン

1 大阪大学 大学院情報科学研究科
Dept. Information Systems Engineering, Graduate School of Information Science and Technology, Osaka University
a) j-kawabe@ist.osaka-u.ac.jp
b) takeuchi@ist.osaka-u.ac.jp
c) yu.jaehoon@ist.osaka-u.ac.jp
d) imai@ist.osaka-u.ac.jp

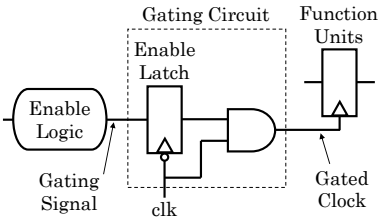

図 1: クロックゲーティング回路

サが必要となる．デカップリングコンデンサがチップ面積の 10%を占める例が報告されており [1]，回路面積を大きくさせる原因となっている．

本稿ではアーキテクチャによりプロセッサの電流変化を抑え電源ノイズを削減する手法を提案する．近年の集積システムは集積密度が増加しており，全ての機能ユニットを動かすのではなく，不要な機能ユニットを停止し効率化を図ることが主流となっている [2]．クロックゲーティングはシステムの動作に不要な回路へのクロック供給を制御することでスイッチング電流を削減する技術である．図 1 にクロックゲーティング回路を示す．機能ユニットへのクロック供給をイネーブルロジックにより決定しラッチと AND 素子を用いてゲーティング済みクロックを生成する．クロックゲーティングは仕組みが簡単であるため広く使われているが，システムの動作不要部分を随時的確に決定することは難しい．また粒度を細かく制御するためにはクロックゲーティングのための回路も大きくなってしまう．本研究では，クロックゲーティングをマルチプロセッサのコアに対して適用し，マルチコアプロセッサの電流変化量が一定以下となるようにクロックゲーティングを行う．

本稿の構成は以下の通りである．2 節で本研究で対象とするマルチコアプロセッサについて述べ，記号を定義する．3 節で提案するクロックゲーティング機構を説明する．4 節でクロックゲーティング機構の効果を実験的に評価し，5 節で結論を述べる．

2. 対象アーキテクチャと記号の定義

2.1 対象アーキテクチャ

本研究ではメモリ共有方式が NORA（No Remote Memory Access）[3] であるマルチコアプロセッサを仮定し，命令の実行に伴うメモリアクセスの競合は発生しないとする．ハードウェア・インターロック機構は持たず，パイプラインハザードは分岐命令によるパイプラインのフラッシュ以外では起きないとする．特定のステージの実行に複数サイクルかかるマルチサイクル命令は持たず，全ての命令が特定のパイプライン段数で実行完了するとする．命令メモリ，データメモリ，命令レジスタは読み書きする内容に依存せず同じ電流遷移を伴うとする．命令を実行しない機能ユニットや値の書き込みが行われないレジスタに流れる電流は時間によらず一定であるとする．

表 1: 定数の定義

記号	定義
C	コア数
S	パイプライン段数
A	命令の集合
F	機能ユニットの集合

表 2: 関数の仕様

関数	機能
$\mathrm{Current}(f)$	機能ユニット $f \in F$ の消費電流
$\mathrm{Unit}(a, j)$	命令 $a \in A$ が j ステージ目で使用する機能ユニットの集合

2.2 記号の定義

本稿で使用する定数の定義を表 1 に示す．マルチコアプロセッサのコア数を C，パイプライン段数を S で表す．命令（ADD，SUB など）の集合を A，機能ユニット（ALU，レジスタなど）の集合を F で表す．本稿で使用する関数の仕様を表 2 に示す．$\mathrm{Current}(f)$ は機能ユニット $f \in F$ の消費電流を返す関数である．$\mathrm{Unit}(a, j)$ は命令 $a \in A$ がパイプラインの j ステージ目で使用する機能ユニットの集合を返す関数である．I_{aj} を命令 a の j $(0 \leq j \leq S-1)$ ステージ目の消費電流とするとき，I_{aj} は以下の式で表される．

$$I_{aj} = \sum_{f \in \mathrm{Unit}(a,j)} \mathrm{Current}(f) \tag{1}$$

あるクロックサイクル（以降「サイクル」と表記する．）においてコア i $(0 \leq i \leq C-1)$ で実行される命令の集合を $D_i = \{d_{i0}, d_{i1}, ..., d_{i(S-1)}\}$ $(D_i \subset A)$ とする．d_{ij} はコア i で j ステージ目が実行される命令を表す．このときコア i の消費電流 Q_i は，パイプライン実行される命令の各ステージで消費される電流の和であるから，以下の式で表される．

$$Q_i = \sum_{0 \leq j \leq S-1} I_{d_{ij}j} \tag{2}$$

3. クロックゲーティング機構

図 2 に提案するクロックゲーティング機構の構成を示す．破線で囲まれた部分がマルチコアプロセッサに追加するクロックゲーティング機構である．クロックゲーティング機構は（1）コアの消費電流の計算回路，（2）実行/停止コアの決定回路，（3）ゲーティング回路から構成される．（1）は次のサイクルにおいてコアでパイプライン実行される命令を基に各コアの消費電流を計算する．図 2 の IR は命令レジスタを表す．（2）は次のサイクルにおけるマルチコアプロセッサ全体の消費電流を計算し，前のサイクルの消費電流との差を計算する．電流差は予め設定される制約電流値と比較され，制約を超えている場合どのコアを停止するかを決定する．（3）は図 1 の破線で囲まれた部分に相当し，ゲーティング済みクロックを生成する．

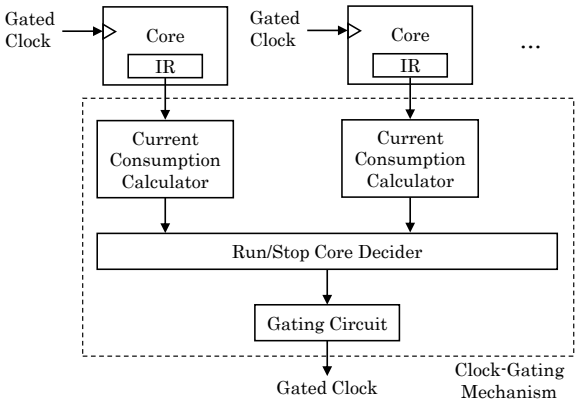

図 2: クロックゲーティング機構の構成

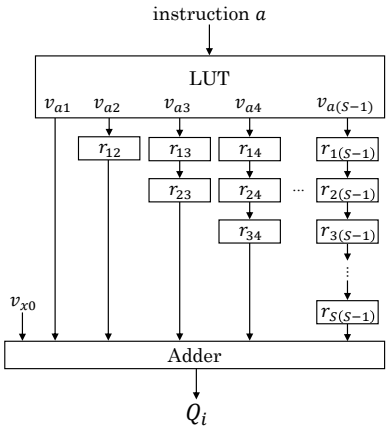

図 3: コアの消費電流の計算回路

3.1 消費電流の計算回路

図 3 にコアの消費電流の計算回路を示す．回路の入力は前のサイクルでフェッチされた命令である．回路の出力は次のサイクルのコアの消費電流である．ルックアップテーブル（LUT）は全ての命令の各ステージごとの消費電流（以降「命令の消費電流」と表記する.）を保持しており，命令 a の j ステージ目の消費電流 v_{aj} を出力する．v_{x0} は命令フェッチ時の消費電流を表す．命令フェッチ時の消費電流は命令の種類に依存せず一定とし加算器に別途入力している．チェーン状に接続されたレジスタ r_{tj} は，命令 a がフェッチされた時刻を t_0 とする時，時刻 $t_0 - t$ にフェッチされた命令の j ステージ目の消費電流を記憶する．クロックごとに保持する電流値をシフトすることで，パイプライン実行を考慮した次のサイクルのコアの消費電流を計算できる．各ステージの消費電流は加算器に入力され最終的にコア i の消費電流 Q_i を出力する．

3.2 実行/停止コアの決定法

実行コア，停止コアの決定法は非常に重要である．停止コアに割り当てられているタスクは実行されないためシステム全体の処理性能に直接影響を与える．本節では性能低下を抑えた停止コアの選択法について説明する．

	コア0 (7μA)	コア1 (9μA)	コア2 (8μA)	コア3 (9μA)	合計 [μA]	電流差 [μA]	制約超過?
1	○	×	×	×	7	3	NO
2	×	×	○	×	8	4	NO
3	×	×	×	○	9	5	NO
4	×	○	×	×	9	5	NO
5	○	×	○	×	15	11	NO
6	○	×	×	○	16	12	NO
7	○	○	×	×	16	12	NO
8	×	×	○	○	17	13	NO
9	×	○	○	×	17	13	NO
10	×	○	×	○	18	14	NO
11	○	×	○	○	24	20	NO
12	○	○	○	×	24	20	NO
13	○	○	×	○	25	21	YES
14	×	○	○	○	26	22	YES
15	○	○	○	○	33	29	YES

前のサイクルの消費電流: 4μA, 制約電流: 20μA

最適解では制約を超えない全ての組み合わせを探索

最大値選択法で探索する組み合わせ

(a) 最適解及び最大値選択法

	コア0 (7μA)	コア1 (9μA)	コア2 (8μA)	コア3 (9μA)	合計 [μA]	電流差 [μA]	制約超過?
1	○	○	×	×	16	12	NO
2	○	○	○	×	24	20	NO
3	○	○	○	○	33	29	YES

前のサイクルの消費電流: 4μA, 制約電流: 20μA
(コア0から選択開始する場合)

3コア実行すると制約下で消費電流を最大化できるため，この組み合わせを選択

(b) 最大値選択法の近似法

図 4: あるサイクルの実行/停止コアの組み合わせ例

3.2.1 性能低下を最小化する最適解

実行するタスク（命令列）が予め分かる場合，各サイクルにおけるコアの実行/停止の全組み合わせを再帰的に計算することで実行サイクル数を最小化するコアの実行/停止の組み合わせ（最適解）を求めることができる．図 4 にあるサイクルにおけるコアの実行/停止の組み合わせの例を示す．4 つのコアがそれぞれ次のサイクルで 7μA，9μA，8μA，9μA 消費するとする．前のサイクルの消費電流を 4μA，電流変化の制約値を 20μA とする．全てのコアを実行すると 33μA 消費するため前のサイクルとの電流差が制約を超える．図 4a に示すように，電流変化が制約を超えない 12 個の組み合わせそれぞれを採用した場合について再帰的に計算し最適解を求める．しかし組み合わせが膨大であるためタスク数（命令数）が多くなると現実的な時間で求めることができない．

3.2.2 最大値選択法

最適解は現実的な時間で求めることができないため，各サイクルにおいて制約以下で消費電流を最大化する組み合わせのみ計算する．この方法を「最大値選択法」と呼ぶ．図 4a の例では電流変化が制約を超えず且つ消費電流が最大となる組み合わせが 2 つ存在する．それぞれを採用した場合について再帰的に計算を行う．全ての組み合わせを計算しないため最適解に比べ計算量が大幅に削減される．

3.2.3 最大値選択法の近似法

最大値選択法は全ての組み合わせについて消費電流を計算する必要があるため実装コストが大きい．そこで調べる

組み合わせを限定して各サイクルの消費電流が大きくなるような選択法を提案する．提案手法はあるコアから連続する 2 コアを動かした場合，連続する 3 コアを動かした場合，...，全てのコアを動かした場合について消費電流を計算し，制約以下で最も消費電流が大きくなるような組み合わせを選択する．図 4b に提案手法を用いる場合の例を示す．コア 0 から連続する 2 コア，3 コア，4 コアを動かした場合の消費電流を計算し前のサイクルの消費電流との差を計算する．例では制約以下で消費電流が最も多く流れる組み合わせ「コア 0，1，2 を実行，コア 3 を停止」を採用する．この方法は必ずしも消費電流を制約以下で最大化する組み合わせを選択できないが，消費電流を計算する組み合わせが少ないため実装コストが小さい．図 4b の例ではコア 0 から連続するコアについて計算したが，次のサイクルではコア 1 から，その次はコア 2 から，というように計算を開始するコアを順番に変えてコアの実行率の偏りを無くす．

3.3 実行/停止コアの決定回路

3.2.3 項で説明した実行/停止コアの決定法を図 5 に示す回路で実現する．図 5 は 4 コアの場合の回路である．右上のカウンタは実行コアとして選択を開始するコアを順番に更新しコアの実行率の偏りを無くす役割をもつ．クロック入力の度に 0,1,2,3,0,1,2,$\cdots$ のように更新される．3 つの加算器はそれぞれ 2 コア実行する場合，3 コア実行する場合，全てのコアを実行する場合の消費電流を計算する．加算器に入力されるコアの消費電流はカウンタの値に応じて変える必要があるため，カウンタの出力を図 5 上部のセレクタの選択信号に入力している．各加算器の結果は減算器に入力され，前のサイクルの消費電流 I_{prev} との差を計算する．電流差は比較器で制約電流値 I_{lim} と比較される．比較器は消費電流の変化が制約を超えている場合は 0 を出力し，制約を超えていない場合は 1 を出力する．組み合わせ回路は比較結果とカウンタの値を基にゲーティング信号を出力する．

3.4 消費電流の量子化

提案するクロックゲーティング機構では消費電流を動的に計算するために図 3 の LUT に命令の消費電流を保持する．命令の消費電流の最大値を $\mathrm{IP}_{\mathrm{max}}$ とすると，$\mathrm{IP}_{\mathrm{max}}$ は以下の式で表される．

$$\mathrm{IP}_{\mathrm{max}} = \max_{0 \le j \le S-1,\ a \in A} I_{aj} \tag{3}$$

LUT の大きさは $\mathrm{IP}_{\mathrm{max}}$ を表現するために必要なビット数に大きく依存し，短いビット数で表現することが望ましい．次項より $\mathrm{IP}_{\mathrm{max}}$，$I_{aj}$ を表現するためのビット数について考察する．

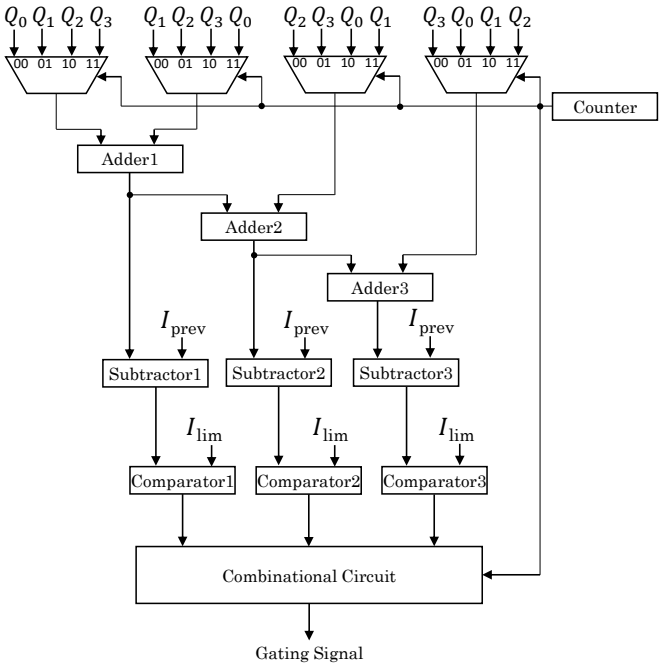

図 5: 実行/停止コアの決定回路

3.4.1 一般的な量子化手法と問題点

量子化とは連続値をとる定義域から離散値をとる定義域に写像することである．時刻 t $(0 \le t < T)$ にとる値を d_t とし，その値域を $0 \le d_t \le h$ とする．値域の最大値 h を $2^w - 1$ に割り当てると量子化幅 r は次の式で表される．

$$r = \frac{h}{2^w - 1} \tag{4}$$

d_t の量子化後のビット表現 v_t は，以下の条件を満たす値となる．

$$v_t r - \frac{r}{2} \ < \ d_t \ \le \ v_t r + \frac{r}{2} \tag{5}$$

したがって d_t の量子化後の値は $v_t r$ で表される．ε を平均二乗誤差とすると，次の式で表される．

$$\varepsilon = \sqrt{\frac{\sum_{t=0}^{T-1}(d_t - v_t r)^2}{T}} \tag{6}$$

この量子化手法を本研究の消費電流の量子化にそのまま適用すると以下の 2 つの問題が発生する．

A. 命令の消費電流が実際よりも小さく近似された場合に電流変化の大きさを保証できない
B. 量子化誤差が大きくなる場合がある

問題 A は式 5 より量子化後の値が元の値よりも小さく近似される場合があるため発生する．制約を満たすようにクロックゲーティングしても実際には制約を超える可能性がある．

問題 B は量子化するデータによっては量子化誤差が大きくなる場合があるため発生する．前述の量子化方法では量子化するデータの最大値 h をビット表現 $2^w - 1$ に割り当てた．これにより量子化ビット数 w に対して量子化幅を最

Algorithm 1 量子化アルゴリズム

Input: 命令の消費電流 d_t^*，量子化ビット数 w，パイプライン段数 S，コア数 C
Output: 命令の消費電流の量子化値 v_t^*

```
h ← max d_t^*
r_min ← ⌈h/(2^w − 1)⌉
r_max ← ⌊h/(2^(w−1) − 1)⌋
ε ← ∞
for r = r_min to r_max do
    k ← 0
    for t = 0 to SC − 1 do
        v'_t ← ⌈d_t/r⌉
        k ← k + (d_t − v'_t r)^2
    end for
    ε' ← √(k/(SC))
    if ε' < ε then
        ε ← ε'
        v_t^* ← v'^*_t
    end if
end for
return v_t^*
```

も細かく設定できる．しかし，量子化するデータが予め決まっている場合，量子化幅をあえて荒く設定する方が誤差が小さくなる場合がある．

3.4.2 量子化アルゴリズム

本項では 3.4.1 項で説明した問題を改善した量子化アルゴリズムを説明する．

問題 A は式 5 を以下のように改善することで解決する．d_t は量子化するデータ，v_t は量子化後のビット表現，r は量子化幅を表す．

$$(v-1)r \ < \ d_t \ \leq \ vr \tag{7}$$

量子化後の電流値が元の値と比べて大きいか等しくなるように近似されるため問題 A を解決できる．

問題 B は，量子化幅を変化させて最も平均二乗誤差が小さくなるような量子化幅で量子化することで解決する．$r_{\min}$ を量子化ビット数 w に対して設定できる最も細かい量子化幅，$r_{\max}$ を最も荒い量子化幅とすると，それぞれ以下の式で表される．h は量子化するデータの最大値を表す．

$$r_{\min} = \left\lceil \frac{h}{2^w - 1} \right\rceil, \quad r_{\max} = \left\lfloor \frac{h}{2^{w-1} - 1} \right\rfloor \tag{8}$$

以上の改善を行った量子化アルゴリズムを Algorithm 1 に示す．d_i^* は命令の消費電流 $\{d_0, d_1, \cdots, d_{SC-1}\}$ を表す．まず命令の消費電流の最大値 h と量子化幅を変化させる下限 $r_{\min}$，上限 $r_{\max}$ を求める．量子化幅 r を $r_{\min}$ から $r_{\max}$ まで変化させ，それぞれについて量子化を行い最小二乗誤差を求める．最小二乗誤差が最も小さくなれば誤差を ε に，量子化結果を v_i^* に記憶する．全量子化幅について計算したら最も誤差が小さい量子化結果 v_i^* を返す．量子化結果 $v_i \in v_i^*$ は $0 \leq v_i \leq 2^w - 1$ となり w ビットで表現できる．

4. 評価実験

本節では提案するクロックゲーティング機構を用いた場合の性能低下について評価し，最終的にどの程度，電流変化量を抑えることができたのかを評価する．

4.1 評価環境

本評価ではプロセッサのコアとして Brownie STD 32 [4] を用いた．Brownie STD 32 は 32 ビットデータバス，32 ビット命令バス，4 段パイプライン，ハーバードアーキテクチャを持つ RISC プロセッサである．プロセッサの動作電圧は 1.8V である．実験するシステムは 4 つの Brownie STD 32 プロセッサコアとコアの実行を制御するクロックゲーティング機構で構成される．クロックゲーティング機構は命令の消費電流を 10 ビット，4 ビット，3 ビット，2 ビットで量子化した場合の 4 種類を実装した．10 ビット実装は，測定した命令の消費電流を整数に丸めた値をそのまま用いた実装である．システムの実装後 Design Compiler を用いてクロックゲーティング機構の面積や消費電力を測定した．用いたスタンダードセルライブラリは TSMC 0.18μm の Typical で，遅延や面積を最小化して測定を行った．

評価のために，DSPstone [5] からランダムに選んだ 100 個のプログラムをタスクとして与えた．これらのプログラムは ASIP Meister [6] によって生成された Brownie STD 32 のコンパイラを用いてコンパイルした．全てのタスクの実行にかかった実行サイクル数を性能低下の評価に用いた．

4.2 実験結果

4.2.1 プロセッサの性能低下の評価

図 6 と表 3 に制約電流値を変化させたときの実行サイクル数を示す．この制約電流値には混乱を防ぐ目的でクロックゲーティング機構の消費電流を含めていない．1 コアの最大電流変化量を考慮して制約電流値の下限を 700μA に設定した．最大値選択法は 3.2.2 項で説明した方法で事前に計算して得られた結果で，それ以外は提案手法である最大値選択法の近似法を用いた場合の結果である．クロックゲーティング機構を用いない場合のサイクル数は図 6 より 2,100 サイクルである．制約電流値が小さくなるほどコアが停止される確率が高くなり，いずれの場合も実行サイクル数が増加していることが分かる．制約電流値が 700μA のとき性能低下が最も大きく，最大値選択法では 0.43%，10 ビット実装では 1.90%，4 ビット実装では 3.14%，3 ビット実装では 3.24%，2 ビット実装では 7.00%サイクル数が増えた．

4.2.2 電流変化量の削減率の評価

提案するクロックゲーティング機構の面積と消費電力の

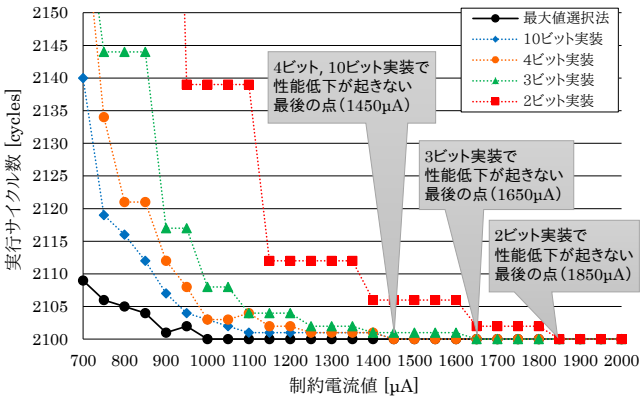

図 6: 性能低下の実験結果

表 3: 性能低下の実験結果

制約電流値 [μA]	最大値選択法 [cycles]	最大値選択法の近似法 [cycles]			
		10 ビット	4 ビット	3 ビット	2 ビット
700	2,109 (+0.43%)	2,140 (+1.90%)	2,166 (+3.14%)	2,168 (+3.24%)	2,247 (+7.00%)
750	2,106 (+0.29%)	2,119 (+0.90%)	2,134 (+1.62%)	2,144 (+2.10%)	2,247 (+7.00%)
800	2,105 (+0.24%)	2,116 (+0.76%)	2,121 (+1.00%)	2,144 (+2.10%)	2,247 (+7.00%)
850	2,104 (+0.19%)	2,112 (+0.57%)	2,121 (+1.00%)	2,144 (+2.10%)	2,247 (+7.00%)
900	2,101 (+0.05%)	2,107 (+0.33%)	2,112 (+0.57%)	2,117 (+0.81%)	2,247 (+7.00%)
950	2,102 (+0.10%)	2,104 (+0.19%)	2,108 (+0.38%)	2,117 (+0.81%)	2,139 (+1.86%)

測定結果を表 4 に示す．消費電力とプロセッサの動作電圧を基に計算した消費電流も合わせて掲載している．表 4 より最小の量子化ビット数である 2 ビットで実装した場合，10 ビット実装の場合から面積を 68.7%，消費電力を 69.8%削減できた．量子化ビット数が 3 ビット，4 ビットと大きくなるにつれて面積や消費電力が増大した．

提案するクロックゲーティング機構を用いない場合の最大電流変化量は，事前の測定により 5,824μA であった．図 6 より，性能低下が起こらない（実行サイクル数が増えない）条件下では，マルチプロセッサの電流変化量を 2 ビット実装の場合 1,850μA，3 ビット実装の場合 1,650μA，4 ビット実装，10 ビット実装の場合 1,450μA まで削減できた．しかし実際にはクロックゲーティング機構の消費電流も考慮する必要がある．表 4 に示した消費電流をそれぞれ加算して，提案するクロックゲーティング機構を用いた場合の最大電流変化量は，2 ビット実装の場合 3,427μA (-41.2%)，3 ビットの場合 3,653μA (-37.3%)，4 ビットの場合 3,780μA (-35.1%) まで削減することができた．10 ビット実装の場合は，クロックゲーティング機構の消費電流が 5,220μA と大きく電流変化量を削減できなかった．

表 4: クロックゲーティング機構の面積と消費電力

	面積 [μm^2]	消費電力 [mW]	消費電流 [μA]
10 ビット実装	20,470	9.396	5,220
4 ビット実装	9,573 (-53.2%)	4.194 (-55.4%)	2,330
3 ビット実装	7,940 (-61.2%)	3.606 (-61.6%)	2,003
2 ビット実装	6,596 (-68.7%)	2.838 (-69.8%)	1,577

5. 結論

本論文では電源ノイズの原因となるプロセッサの電流変化を一定以下の大きさに抑えるためのクロックゲーティング機構を提案した．提案手法では実行/停止コアの決定アルゴリズムを工夫することで回路の実装コストを抑えつつプロセッサの性能低下を抑えた．評価実験によって，クロックゲーティングに伴うプロセッサの性能低下を最大 7%に抑えることを確認した．また提案するクロックゲーティング機構を用いることで，性能低下が起きない（実行サイクル数が増加しない）条件下で，電流変化量を 2 ビット実装の場合 41.2%，3 ビット実装の場合 37.3%，4 ビット実装の場合 35.1%削減することを確認した．

ACKNOWLEDGEMENT

本研究は JSPS 科研費 JP26330064 の助成を受けたものである．

参考文献

[1] C.-H. Lu, H.-M. Chen, and C.-N. J. Liu, “Effective decap insertion in area-array soc floorplan design,” *ACM Transactions Design Automation of Electronic Systems*, vol. 13, no. 4, pp. 66:1–66:20, Oct. 2008.

[2] J. Henkel, S. Pagani, H. Khdr, F. Kriebel, S. Rehman, and M. Shafique, “Towards performance and reliability-efficient computing in the dark silicon era,” in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 1–6.

[3] M. Sueishi, M. Kitakami, and H. Ito, “Fault-tolerant message switching based on wormhole switching and backtracking,” in *Dependable Computing, 2004. Proceedings. 10th IEEE Pacific Rim International Symposium on*, March 2004, pp. 183–190.

[4] H. Iwato, T. Hieda, H. Tanaka, J. Sato, K. Sakanushi, Y. Takeuchi, and M. Imai, “A highly extensible base processor for short-term asip design,” *IPSJ SIG Technical Reports*, vol. 2007, no. 114, pp. 133–138, 2007 (in Japanese).

[5] V. Zivojnovic, J. Mart, C. Schlager, and H. Meyr, “DSPstone: A DSP-oriented benchmarking methodology,” in *Proceedings of the International Conference on Signal Processing Applications and Technology*, 1994.

[6] M. Imai, Y. Takeuchi, K. Sakanushi, and N. Ishiura, “Advantage and possibility of application-domain specific instruction-set processor (ASIP),” *IPSJ Transactions on System LSI Design Methodology*, vol. 3, pp. 161–178, 2010.