情報処理学会論文誌 Vol.59 No.3 1061–1076 (Mar. 2018)

DVFS使用下における余剰時間を利用した最上位キャッシュ切替えによるキャッシュ消費エネルギーの削減

齋藤 郁[1,a)] 小林 良太郎[2,b)] 嶋田 創[3,c)]

受付日 2017年6月26日, 採録日 2017年12月8日

概要：IoT デバイスやスマートデバイスに搭載される System-on-a-Chip（SoC）アーキテクチャの中で，CPU に着目すると，内包されているキャッシュメモリの消費電力割合が大きい．回路の省電力化には，最低限の電圧を供給するような電源管理がよく用いられるが，キャッシュはプロセスルールの微細化の影響で供給電圧を大幅に下げることが困難であり，電源管理による省電力化の効果が小さい．そこで本論文では，キャッシュ省電力化の実現のため，CPU の省電力化手法である動的電圧周波数制御（DVFS）における，プロセッサコアの周波数制御に着目した．DVFS は，CPU の要求性能が低い場合に，主にプロセッサコアの周波数と電圧を低下させて電力を削減する．プロセッサコアの周波数が低下している場合，CPU 全体の動作に影響を与えずに，低速・低電力なキャッシュにアクセスできると考えた．提案機構では，DVFS によって変化するコアの周波数に合わせて，速度と電力が異なる 2 つの L0 キャッシュから最適な設定を自動的に選択して使用する．これにより，CPU 全体の要求性能に対して最低限の消費エネルギーを持つ L0 キャッシュを利用できる．評価の結果，提案機構は SPECint2006 で最大 23.5%，SPECfp2006 で最大 27.2%の消費エネルギーを削減可能という結果を得た．

キーワード：キャッシュメモリ，CPU，DVFS，消費エネルギー削減

Cache Energy Reduction by Dynamically Switching The Highest-level Caches during Surplus Time Due to DVFS

Kaoru Saito[1,a)] Ryotaro Kobayashi[2,b)] Hajime Shimada[3,c)]

Received: June 26, 2017, Accepted: December 8, 2017

Abstract: IoT devices and smart devices equip a System-on-a-Chip (SoC) architecture. An SoC architecture contains CPUs and its cache memory occupies a large part of CPU energy consumption. To reduce the power consumption of electronic circuits, power management is used. However, it is a challenge to apply power management to the cache because the SRAM cell, which is the principal component of the cache, presents some difficulty in reducing its power voltage requirements. Dynamic Voltage and Frequency Scaling (DVFS) is commonly used to reduce CPU energy consumption. In this paper, we propose a cache energy reduction mechanism focusing on the frequency control of the processor core in DVFS. When the required performance is low, DVFS decreases the processor core supply voltage and clock frequency to reduce its energy consumption. Our proposed mechanism implements switchable L0 caches, low performance (low speed and low access energy) L0, high performance (high speed and high access energy) L0, switching them or accessing in sequential order depending on DVFS activity. The proposal consumes the minimum L0 cache access energy necessary to maintain CPU performance. The evaluation result shows that the proposed mechanism reduces the cache access energy by 23.5% and 27.2% for SPECint2006 and SPECfp2006, respectively.

Keywords: cache memory, CPU, DVFS, energy reduction

[1] 豊橋技術科学大学
Toyohashi University of Technology, Toyohashi, Aichi 441–8580, Japan
[2] 工学院大学
Kogakuin University, Shinjuku, Tokyo 163–8677, Japan
[3] 名古屋大学
Nagoya University, Nagoya, Aichi 464–0814, Japan
[a)] saito2018@ppl.cs.tut.ac.jp
[b)] ryo.kobayashi@cc.kogakuin.ac.jp
[c)] shimada@itc.nagoya-u.ac.jp

1. はじめに

近年，様々なモノをインターネットに接続し，相互にデータをやりとりすることによってシステムを構築するInternet of Things（IoT）が重要視され始めている．IoTシステムは，洗濯機やテレビなどの家電からカーナビなどの車載装置，スマートフォンやスマートウォッチなどのスマートデバイスまで様々なデバイスによって構成される．

IoTデバイスのうち，家電製品であれば，家庭用電源からの安定した電源供給が可能である．また，車載装置も車載バッテリを電源とするため，動作時間の確保は比較的容易となっている．

一方，バッテリによって動作するデバイスについては，動作時間の確保が問題となる．昨今，爆発的に普及しているスマートデバイスは，大画面化の傾向により大容量バッテリが搭載可能となっているが，提供されるサービスの高度化と性能向上のため，機器の電力量はいまだ問題になりやすい．スマートデバイスの中でも，スマートウォッチのような小型なデバイスについては，大容量バッテリを搭載できない．実際に，Apple Watch Series 2に搭載されているバッテリは273 mAhで連続待ち受け時間は18時間となっている[1]．2020年までにIoTデバイスの数は310億台に達するともいわれており，省電力化への需要は高い[2]．

IoTデバイスの省電力化対象の1つとして，各デバイスに搭載されるSystem-on-a-Chip（SoC）アーキテクチャがあげられる．SoCアーキテクチャは，処理の中心となるCPUはもちろん，メモリ，ビデオチップ，I/Oなどの機能を1つのチップ上に統合したものとなっている．SoCアーキテクチャ上の各機能それぞれに省電力化の余地が存在するが，本研究ではCPUに着目する．

CPUの省電力化手法の1つとして，動的電圧周波数制御（DVFS）が存在する．これは，主にプロセッサコアの省電力化に用いられている手法である．DVFSが有効になっているCPUでは，要求性能および負荷状況に応じて，主にプロセッサコアの動作周波数と供給電圧が動的に制御される．たとえば，CPU全体の動作に余裕がある場合に，プロセッサコアの周波数と電圧を低下させて処理を実行することで，CPUの省電力化を実現することができる．

一方，CPU全体の消費電力を確認するとキャッシュメモリが大きな割合を占めている．例として，CPU全体の電力に対してキャッシュメモリが消費する割合は，StrongARM SA-110では43%[3]，ARM920Tでは44%[4]となっている．このことから，キャッシュメモリの省電力化がCPUの省電力化に有効であり，バッテリによって動作するデバイスの動作時間延長を実現する手段の1つであるといえる．

本研究では，DVFSを利用することで，アーキテクチャ技術による新たなキャッシュ動的エネルギーの削減手法を提案し，キャッシュメモリの省電力化を実現する．DVFSが有効になっているCPUは，処理の負荷状況に応じてコアの動作周波数が変動し，それに合わせてサイクルタイムが変動する．本研究ではこの点に着目し，サイクルタイムに応じて最適な性能のL0キャッシュを使用するように設定を切り替えることによって，キャッシュ動的エネルギーを削減し，キャッシュを省電力化する．

キャッシュの消費エネルギーを論じる際，アクセスにともなう消費エネルギーである動的エネルギーについてだけでなく，待機時に発生する静的エネルギーについても言及されることが多い．静的エネルギーに関しては，アーキテクチャ技術による削減と，デバイス技術による削減の両方について研究事例が存在する．特にデバイス技術においては，FinFETや垂直STT-MRAMなどにより大幅な静的エネルギーの削減が実現しつつある．

FinFETに関しては，Intelが2011年に，従来のプロセスルール32 nmのCMOSから静的電力を1/10まで削減した22 nm FinFETを発表している[5]．加えて，2017年には同じくIntelが，22 nmのFinFETの静的電力をさらに1/100まで削減した22 nm FinFET Low Power（22FFL）を発表している[6]．このように，デバイス技術の発展による静的電力の削減が進んでいることから，今後，キャッシュ全体の消費エネルギー（動的＋静的）に対する静的エネルギーの割合が小さくなっていくと考えられる．

以上のような背景から，本研究では，静的エネルギーの削減をデバイス技術の発展に任せ，アーキテクチャ技術の改良による動的電力の削減によって，キャッシュの消費エネルギー削減を試みる．

以下，2章ではキャッシュメモリの電力削減に関連する研究について，3章では提案機構に用いるアイディアの要説とDVFSについて，4章では提案機構の詳細について，5章ではその評価について，6章では提案機構が有効に働く構成の検討について，7章では全体のまとめをそれぞれ述べる．

2. 関連研究

キャッシュメモリの省電力化に関する既存の研究は多く存在し，キャッシュメモリを構成する半導体素子の改良によるものと，制御方式の改良によるものとに大別できる．

キャッシュメモリを構成する半導体素子の改良に関する研究として，FinFET[9]や垂直STT-MRAM[10]があげられる．

FinFETは，3次元構造を持つトランジスタであり，広範なゲート面積によって高速性と省電力性の両立を実現している．従来のCMOSと同等の性能を持ちながら，動的エネルギー，静的エネルギーともに削減しており，特に静的エネルギーは，従来より大幅に削減されている[5], [6]．キャッシュメモリに限らず，CPUなどの記憶素子以外の回路を構成する際に用いることで，それらの省電力化も可

能となる．

垂直 STT-MRAM は，動作速度が速く，無限大に近い書き込み耐性を持つ不揮発性メモリである．不揮発性メモリの特性を活かし，タスクの実行中にパワーゲーティングを行うことで，保持したデータの破損を防ぎつつ省電力化を可能としている．ただし，L1 キャッシュとして使用する場合には，性能面で既存の MOSFET に劣るという欠点がある．L2 キャッシュ以降に適用する場合であれば十分な性能を維持しており，実際に適用した場合には，消費エネルギーを MOSFET の 80%程度に抑えることができる．

本研究で提案する機構は，L0 キャッシュのような最上位キャッシュに対しての適用を想定している．また，提案機構には，周波数の低下による動作時間の延長によって，静的エネルギーが増加するという特性がある．したがって，本提案機構と改良素子を組み合わせる場合，L0 キャッシュに適用可能で，静的エネルギーを大幅に削減可能なものであれば，さらなる消費エネルギー削減が期待できる．そのような条件を満たす素子として FinFET があげられる．

一方，STT-MRAM は，静的エネルギーの割合を削減できるが，L2 キャッシュなどの下位のキャッシュに対して適用されるため，提案機構のさらなる消費エネルギー削減は期待できない．

なお，改良素子との組合せによる効果の検証は今後の課題とする．本論文では，従来の CMOS の使用を前提として評価を行う．

キャッシュメモリの制御方式の改良に関する研究は数多く存在する [11], [12], [13], [14]．それらの中でも，本研究に関連が深いものとして，動的にキャッシュ容量を変更する制御方式が提案されている [13], [14]．

その 1 つである DRI キャッシュ [13] は，一定サイクル数ごとのキャッシュミス回数を確認し，ミス回数に応じて，動的にキャッシュ容量を変更する．ミス回数が一定以下であれば，キャッシュ容量の半分に相当する領域をシャットダウンすることで容量を削減する．その一方で，ミス回数が一定以上の場合，シャットダウンしていた領域を通常モードに復帰させ，キャッシュ容量を増加させることで，性能の低下を防ぐ．

DRI キャッシュでは，キャッシュの一部領域をシャットダウンする際，その領域に格納されていたデータは破棄される．そのため，データキャッシュに適用する場合には，データの一貫性を保つために，データを下位の記憶装置に書き戻す処理が必要となり，シャットダウンのたびにオーバヘッドが発生することになる．

一方，可変レベルキャッシュ [14] は，一定サイクル数ごとのキャッシュミス回数を確認し，ミス回数が一定以下であれば，キャッシュ容量の半分をスリープモードに移行させる．

スリープモードとは，キャッシュメモリが保持するデータを破損しない程度に電源電圧を低下させることで，リークエネルギーを削減する状態である．また，スリープモードへの移行はインデックスにマスクをかけることによる，一度にアクセス可能なキャッシュ領域の制限によって実現しているため，動的エネルギーの削減にもつながる．

スリープモードに移行したキャッシュ領域へのアクセスでは，通常のキャッシュアクセスレイテンシに加えて，スリープ状態から通常モードへの復帰のためのレイテンシが発生する．これを可能な限り抑制するため，スリープモードに移行した領域を，移行前の 1 つ下の階層のエクスクルーシブキャッシュとして扱う．例として，256 kB の L2 キャッシュの半分をスリープモードに移行させる場合，次の 2 つとして扱う：128 kB の L2 キャッシュ，128 kB の L3 キャッシュ（エクスクルーシブキャッシュ）．

このように可変レベルキャッシュは，要求性能に合わせてキャッシュの一部分をスリープモードに移行させることによって，省電力化を実現する．

可変レベルキャッシュは DRI キャッシュと異なり，通常モードからスリープモードへ移行させる際に，下位の記憶装置へデータを書き戻す必要がない一方で，スリープモードから通常モードに移行させる際に，すべてのデータを下位の記憶装置に書き戻す必要がある．これは，スリープモードから復帰した直後は，キャッシュ内のデータ配置が通常モードでの本来の配置と異なっているためである．この状態で動作し続けると，参照時にデータが存在していても，本来のインデックスと異なる位置に存在していることになるため，正しく参照することができなくなる．参照できなくなったデータが最新だった場合，下位の記憶装置から古いデータを読み込むことになり，データの一貫性が保てなくなる．そのため，通常モードへの移行の際には，すべてのデータを下位の記憶装置に書き戻し，参照された際に最新のデータが読み込まれるとともに，正しいインデックスに再格納されるようにする．

以上のような既存の研究は，一般的な 2 階層キャッシュに搭載された，L1 や L2 に手を加える内容となっている．一方，提案機構では，L1 と L2 で構成された 2 階層キャッシュに対して，省電力化のためのアイディアを盛り込んだ L0 を加えている．

3. 基本方針

3.1 提案するアイディアの要説

本研究における提案機構では，CPU の省電力化手法の 1 つである DVFS と，キャッシュの省電力化に用いられている L0 キャッシュの技術を利用する．

DVFS 適用下では，主に CPU 内部のプロセッサコアの動作周波数と供給電圧が，CPU の要求性能および負荷状況に合わせて動的に制御される．たとえば，CPU の動作に余裕があり，処理される各タスク間に処理を行わずに待

機する余剰時間が存在する場合，プロセッサコアの周波数と電圧を低下させて処理を実行することで，動作にともなう電力を削減できる．この際，キャッシュメモリに供給される電圧もある程度スケーリング可能であるが，設計上の問題により，供給電圧を大幅に低下させることができず，それによる省電力化の効果は小さい．この設計上の問題と供給電圧の関係については，3.2 節において改めて触れる．

プロセッサコアに話を戻す．コアの周波数が低下すると，サイクルタイムが増加する．サイクルタイムが増加すると，レイテンシを増加させることなく低速なキャッシュにアクセス可能となる．一般的にキャッシュは，低速な設計であれば，アクセスにともなう消費エネルギーを小さくすることができる．これらのことから我々は，コアの動作周波数が低下すると，CPU 全体の処理に悪影響を与えることなく消費エネルギーが小さなキャッシュにアクセスできると考えた．

一般的なキャッシュメモリは，L1 と L2 の 2 階層で構成される．これに対し，L1 キャッシュの上位に L0 キャッシュを用いる手法が多く研究されている [3], [15], [16]．これらの研究事例では，一般的な階層キャッシュにおいて最上位キャッシュとして扱われる L1 キャッシュの上位に，L0 キャッシュを導入している．

L0 キャッシュは，L1 キャッシュより小容量であるため，1 回のアクセスに要する消費エネルギーが小さい．そして，ある一連の命令処理中に，同じデータが高頻度にアクセスされるようなデータの局所性が存在する場合に有効に働く．

高頻度にアクセスされるデータ群が，L0 キャッシュに書き込みきれる容量であるならば，L0 キャッシュが実用上差し支えないヒット率を持つことになる．L0 キャッシュが十分なヒット率を持つ場合，L1 キャッシュや L2 キャッシュへのアクセス回数が減少する．このような条件を満たす環境であれば，L1 や L2 へのアクセスで発生するエネルギーを抑制することができ，最低限の消費エネルギーでキャッシュアクセスを完了できる．

一方，データの局所性がほとんどなく，異なるデータが次々にアクセスされるような命令処理が続く場合，L0 のみでキャッシュアクセスが完了せず，L1 以下のキャッシュへアクセスが頻発する．このような，L0 のヒット率が低い環境では，L0 へのアクセス分のエネルギーが無駄に消費されてしまうことになり，L1 と L2 の 2 階層キャッシュと比較して消費エネルギーが増加してしまう．

したがって，L0 を導入する場合には，想定される CPU の処理内容に対して，L0 が十分な容量を持っているか確認することが求められる．

以上の DVFS と L0 キャッシュを用いて，提案機構を構成する．提案機構では，容量が同一であるが，性能の異なる 2 種類の L0 キャッシュを用意する．そして，最上位キャッシュとして使用する L0 キャッシュを DVFS によるコアの動作周波数の変化に連動して切り替える．

先にあげた研究事例 [3], [15], [16] をはじめとして，従来の L0 キャッシュに関する研究では，CPU へ単一の L0 キャッシュを搭載している．一方，本研究では，CPU に容量が同一だが性能が異なる 2 種類の L0 キャッシュを同時に搭載し，コアの周波数に応じてそれらを切り替える手法を新たに提案している．

本論文では，アクセス速度を L0 キャッシュ性能の基準として論じる．アクセス速度以外の性能指標については，アクセス速度が高速で面積を一定にするならば消費エネルギーが大きくなるといった，一般的な性能のトレードオフに従う．つまり，性能が高いと表現した場合，アクセスが高速だが消費エネルギーが大きい L0 キャッシュを指し，性能が低いと表現した場合，アクセスが低速だが消費エネルギーが小さい L0 キャッシュを指す．なお，単体の L0 キャッシュのヒット率に関しては，議論の必要に応じてヒット率という表現で言及するため，L0 キャッシュの性能という表現の範疇に含まないものとする．

以上をふまえて提案機構では，コアの動作周波数が高い範囲では，性能の高い L0 キャッシュ（高速だがアクセスにともなう消費エネルギーが大きい L0 キャッシュ）を使用し，コアの周波数が十分に低下した範囲では，性能の低い L0 キャッシュ（低速だがアクセスにともなう消費エネルギーが小さい L0 キャッシュ）を使用する．これにより，コアの周波数が低下した場合に，既存の L0 キャッシュ単体使用時と比較してさらなる省電力化を実現する．

DVFS は本研究の実現に不可欠な技術であるため，次節でその詳細を説明する．

3.2 動的電圧周波数制御（DVFS）

先述のとおり DVFS とは，CPU の省電力化のために，一般的に用いられている手法の 1 つである [7]．この手法は，CPU が順次処理するタスク間に，処理を行わず待機することになる余剰時間が存在する場合に適用可能である．DVFS では，その余剰時間を利用して処理を行うように，CPU の供給電圧・動作周波数を低下させることで，CPU の動作にともなう消費電力を削減する．つまり，各タスク間に存在する利用可能な余剰時間が長いほど，DVFS による省電力化の効果が大きいということになる．

DVFS による制御の際，CPU 内の各モジュールの電圧を制御可能であるが，主にプロセッサコアの電圧が制御される．したがって，DVFS は主にプロセッサコアの省電力化によって CPU の省電力化を実現している．

図 1 に DVFS によるプロセッサコア省電力化の概要図を示す．淡色の矩形が最大周波数でタスクを処理した場合の処理時間と供給電圧の関係を示している．淡色の矩形どうしの間を確認すると，あるタスクの終了後，次のタスクが実行されるまでの間に処理を行わず待機している時間が

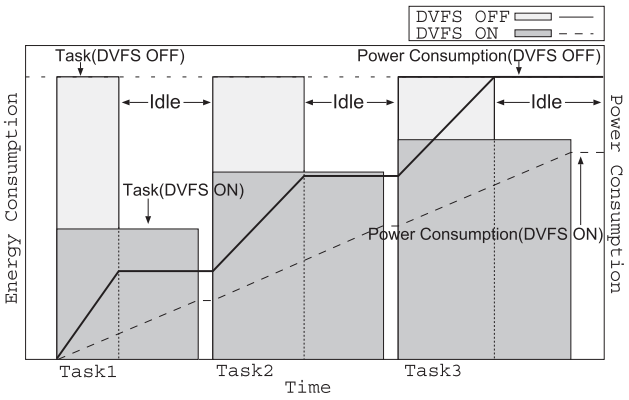


図 1 DVFS によるプロセッサコアの消費電力削減
Fig. 1 Summary of how DVFS reduces processor core energy.

存在することが分かる．たとえば，Task1 の淡色の矩形と，Task2 の淡色の矩形間を確認すると，利用可能な余剰時間（Idle）が存在している．DVFS による制御を有効にしている場合，この Idle を利用する．すなわち，次のタスクの実行開始をデッドラインとし，その開始に影響を及ぼさない範囲で処理に要する電圧と周波数を低下させることで，濃色の矩形のように処理を実行する．例として，Task1 の淡色の矩形（Task（DVFS OFF））から，濃色の矩形（Task（DVFS ON））のように，デッドライン（Task2 の開始）に間に合う範囲で，利用可能な余剰時間（Idle）を有効活用し，低消費エネルギーで処理を実行する．

このとき，プロセッサコアの消費エネルギーと供給電圧の関係は次式で表すことができる [8]．

$$E = \int_0^T (\alpha C V_{dd}^2 f + V_{dd} I_Q) dt \tag{1}$$

ここで，α はスイッチング確率，C は回路の静電容量，V_{dd} は回路の供給電圧，f は動作周波数，I_Q はリーク電流である．

C，V_{dd}，f，I_Q が時間依存しない定数であるとすると，式 (1) は次のように書き換えることができる．

$$E = \alpha C V_{dd}^2 f T + V_{dd} I_Q T \tag{2}$$

ここで，時間 T は，周波数 f，実行サイクル数 N_c を用いて以下のように定義する．

$$T = \frac{N_c}{f} \tag{3}$$

式 (3) を用いて，式 (2) を変形すると以下のようになる．

$$\begin{aligned} P &= \alpha C V_{dd}^2 f \frac{N_c}{f} + V_{dd} I_Q \frac{N_c}{f} \\ &= \alpha C V_{dd}^2 N_c + V_{dd} I_Q \frac{N_c}{f} \end{aligned} \tag{4}$$

式 (4) より，消費電力は電圧の二乗に比例することが分かる．したがって，供給電圧を低下させ，濃色の矩形のようにタスクを実行することで，図 1 中の実線から破線のようにプロセッサコアの動作にともなう消費電力を削減することが可能となる．

2000 年代初頭から，Intel 系 CPU では SpeedStep [17] として，AMD 系 CPU では PowerNow!や Cool'n'Quiet [18] といった名称で DVFS に基づいた機能を導入し始め，その後広く使われるようになった．このように DVFS は，CPU の省電力化手法として一般に用いられている．

先述のとおり，DVFS は CPU 内の各モジュールに適用できるが，供給電圧の低下という観点において，キャッシュメモリに対しては効果が小さい．

キャッシュメモリは，主に CMOS 回路による SRAM セルで構成されている．CMOS のような半導体は，数十年にわたってプロセスルールの微細化が進められてきた．この動きによって，半導体は高速な動作を実現できるようになったが，製造時のバラつきが発生しやすくなった．これにより，ゲート閾値電圧のバラつきやソフトエラーの発生率上昇という問題が顕在化し，キャッシュに対する供給電圧は余裕を持った値に設定する必要が出てきた．いい換えるならば，大幅な供給電圧の低下はキャッシュの動作異常を引き起こしやすくなっている．

このような背景から本論文では，キャッシュメモリの周波数は大きく変化させることができるが，電圧は大きく変化させることができないと仮定する．

一方，DVFS による制御が有効になっている場合，プロセッサコアの供給電圧とともに動作周波数が低下していることは先に述べた．プロセッサコアの動作周波数が低下している場合，キャッシュメモリへのアクセス時間にも余裕が生じる．キャッシュメモリは設計上，容量が同一であるなら，低速であるほど消費電力が小さくなり，高速であるほど消費電力が大きくなる．したがって，プロセッサコアの周波数が高い場合，高速・高消費電力な L0 キャッシュに切り替え，コアの周波数が低い場合，低速・低消費電力な L0 キャッシュに切り替える．これにより，CPU 全体の処理に影響を及ぼさずに L0 キャッシュへのアクセスで発生するエネルギーを削減できると考えた．

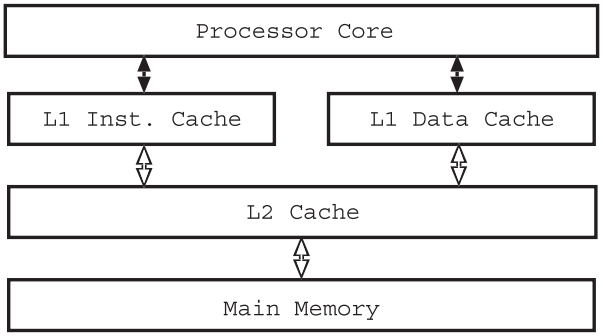
以上をふまえ，提案機構の実装について 4.1 節で説明する．

4. 提案機構

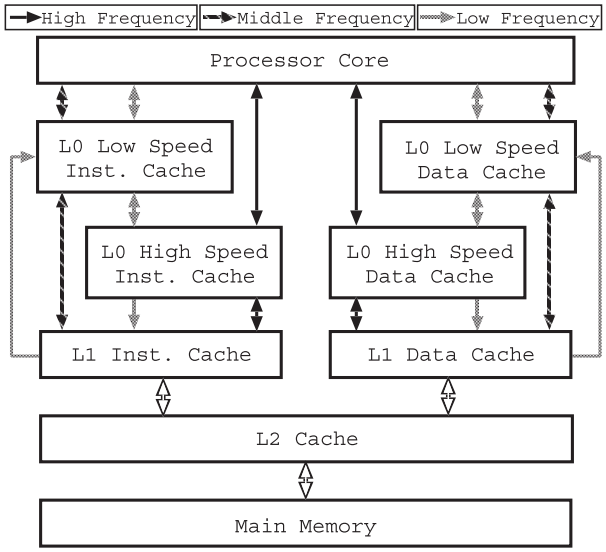
4.1 実装

図 2 に一般的な 2 階層キャッシュと提案機構の構成を示す．一般的な 2 階層キャッシュは，図 2 (a) のように L1 と L2 の 2 つで構成される．前述のとおり，この 2 階層に加えて，L1 とコアの間に L0 を搭載し，省電力化を図る技術が存在する．

提案機構では，L0 キャッシュをさらに発展させ，図 2 (b) のようにし，容量が同一だが性能が異なる 2 種類の L0 キャッシュを搭載する．本機構では，DVFS による周波数変動に合わせて，L0 キャッシュへのアクセスが 1 サイク



(a) 既存の 2 階層キャッシュ



(b) 提案機構

図 2 キャッシュの構成
Fig. 2 Cache configurations.

ルで完結するよう 2 種類の L0 を切り替えて使用する．これらの L0 キャッシュは，その性能の違いから，L0 High Speed（L0HS），L0 Low Speed（L0LS）と呼称する．また，設定によっては，2 種類の L0 を同時に使用する．このとき，これらを 1 つの L0 と見立て，L0MIX と呼称する．L0MIX では，2 種類の L0 が互いに共通なデータを持たないエクスクルーシブキャッシュとして動作する．これにより，L0 の実質的な容量が単体使用時の 2 倍になり，ヒット率の向上が見込める．

提案機構では，プロセッサコアの周波数を 3 つの範囲に分けて定義する．そして，3 つの周波数範囲に，3 種類の L0 キャッシュを次のように対応させる．2.0 GHz–1.4 GHz の範囲を High Frequency とし，L0 キャッシュとして L0HS を使用する．1.3 GHz–0.9 GHz の範囲を Middle Frequency とし，L0LS を使用する．0.8 GHz–0.2 GHz の範囲を Low Frequency とし，L0MIX を使用する．以上の提案機構における周波数範囲とキャッシュ設定の対応を**表 1** にまとめる．

図 3 に 3 つの周波数範囲それぞれに，各 L0 キャッシュを

表 1 提案キャッシュ制御情報
Table 1 Proposed cache control.

Frequency Range	Frequency [GHz]	L0-config
High Frequency	2.0–1.4	L0HS
Middle Frequency	1.3–0.9	L0LS
Low Frequency	0.8–0.2	L0MIX

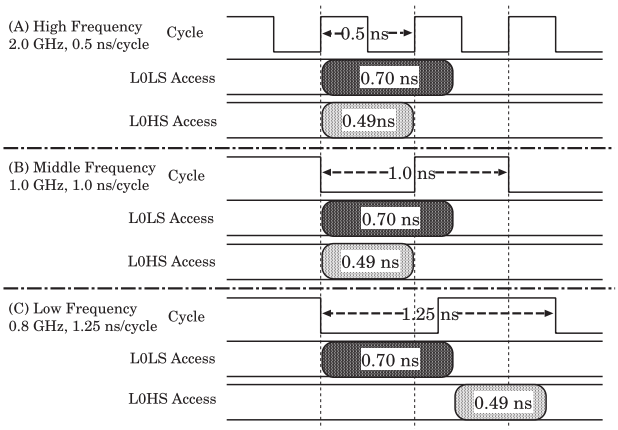


図 3 コアの周波数変化にともなう L0 アクセスサイクル数の変化
Fig. 3 Processor Core Frequency and L0 access cycles.

対応させる際の考え方の例を示す．ここで，L0HS と L0LS のアクセスに要する時間は，それぞれ 0.49 ns，0.70 ns であるとする．

図 3 (A) では，High Frequency の例として，コアが 2.0 GHz で動作している場合を示している．ここでは，1 サイクルタイムは 0.50 ns となっているため，L0HS は 1 サイクルでアクセスが完了するが，L0LS は完了までに 2 サイクルを要してしまう．この場合，L0 キャッシュへのアクセスを 1 サイクルに抑えるため，L0HS を使用する．

次に，図 3 (B) では，Middle Frequency の例として，コアが 1.0 GHz で動作している場合を示している．このとき，1 サイクルタイムは，1.0 ns となっている．L0HS，L0LS ともに 1 サイクルでアクセスできるが，省電力化のため，消費エネルギーが小さい L0LS を使用する．

最後に，Low Frequency の例である図 3 (C) では，コアが 0.8 GHz で動作している．ここで，1 サイクルタイムは 1.25 ns である．これは，L0HS と L0LS のアクセス時間の合計である 1.19 ns より長い．この場合，L0LS でミスした後，L0HS にアクセスしても 1 サイクルでアクセスを完了できるため，上述のとおり 2 つの L0 を L0MIX として同時に使用する．

キャッシュ設定の切替えパターンは，① L0HS → L0LS，② L0LS → L0MIX，③ L0MIX → L0LS，④ L0LS → L0HS の 4 通り存在する．そのうち，キャッシュが保持するデータの一貫性を保ったまま切替えができるのは，② の場合のみである[*1]．② 以外の切替えパターンでは，単純に切替え

[*1] ③ の場合，L0LS のみ一貫性が保たれる．

を行うとキャッシュの一貫性保持ができなくなる．そのため，一貫性を保つ処理として，2 種の L0 を切り替える際に，L0 内のすべてのデータを走査し，ダーティなデータが存在するかどうかを確認したうえで，必要に応じて下位のキャッシュへバックアップを行う．この処理のためにオーバヘッドが発生することになるが，L0 の容量はきわめて小さいため，CPU 全体の処理とエネルギーに対する影響は少ない[*2]．

DVFS による周波数制御は，一連の処理実行中に，CPU の負荷状況に応じて動的に行われる．そのため，DVFS による周波数制御には，あらかじめ負荷状況を把握し，処理実行中の各区間において，どの程度の周波数で実行可能であるかスケジューリングすることが必要となる．

一般的なリアルタイム性を持つ組込みシステムにおけるスケジューリングでは，最悪実行時間解析に基づいて処理のスケジューリングが行われる．最悪実行時間の解析方法として，いくつかの考えられる入力パターンを与えながら処理時間を計測し，最も長い処理時間に安全率を加えたものを最悪実行時間とする測定ベース解析手法がとられる [19]．

本提案機構の評価では，この測定ベース解析が理想的に実行できる状態を想定し，DVFS による制御を行う．すなわち，あらかじめベンチマーク上で一連の命令処理に要する総サイクル数を測定しておき，最悪実行時間としてスケジューリングに利用する．

具体的には，100 万命令の実行ごとに，周波数切替えの判断として，これまでの命令実行時間（A）と次の 100 万命令の実行完了に要する時間（B）をあらかじめ測定したサイクル数から算出し，デッドラインと比較する．B を算出する際，現在の周波数，1 段階上げた周波数，1 段階下げた周波数の 3 つのパターンで算出する．

なお，1 段階周波数を上げた場合と，下げた場合の B を算出するとき，周波数およびキャッシュ設定の切替えによって発生するオーバヘッドを加味する．たとえば，現在の周波数が 1.8 GHz であった場合，1.8 GHz，1.9 GHz，1.7 GHz のそれぞれにおいて，実行サイクル数および切替えオーバヘッドより B を計算する．

A と B の合計とデッドラインを比較した際，デッドライン制約を満たしつつ周波数が最低となる設定を選択する．

4.2 キャッシュアクセスの動作

図 4 に提案機構のキャッシュアクセス動作を示す．提案機構は，4.1 節に記述したとおり，3 つの周波数範囲によって L0 キャッシュのアクセス先や順序が異なる．

High Frequency でのデータ・命令参照時は，最初に L0HS にアクセスする．L0HS に目的のデータ・命令が存在し，

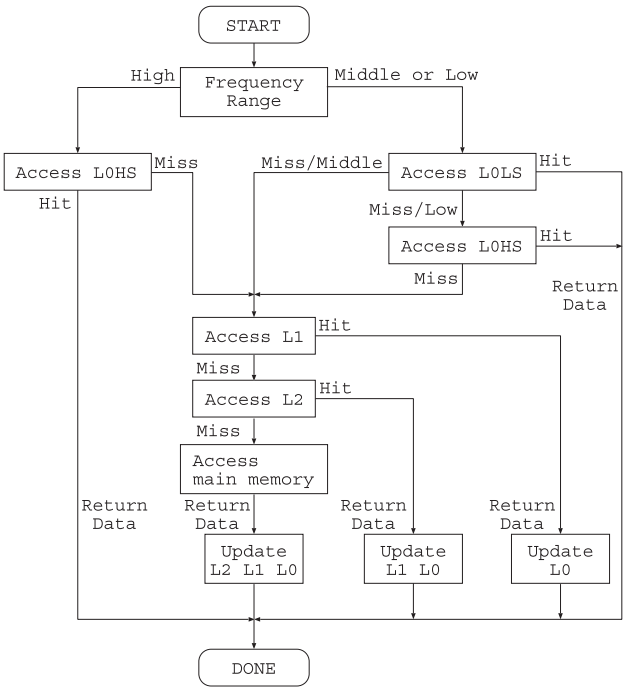


図 4 提案キャッシュアクセスのフローチャート
Fig. 4 Flowchart of proposed cache access.

Hit となった場合，キャッシュアクセスを終了する．L0HS に目的のデータ・命令が存在せず，Miss となった場合，L1 以降のキャッシュにアクセスする．その後，L1 で Hit した場合，Hit したデータ・命令を L0HS に書き込み，アクセスを終了する．L1 で Miss となった場合，L2 にアクセスする．L2 で Hit した場合，Hit したデータ・命令を L1，L0HS に書き込み，アクセスを終了する．L2 でも Miss となった場合，メインメモリにアクセスし，参照されたデータ・命令を L2，L1，L0HS に書き込んでアクセスを終了する．

なお，L0 への書き込みは各周波数範囲において動作中の L0 に対してのみ行う．そのため，High Frequency の場合では，L0HS のみへ書き込みを行っている．

Middle Frequency でのデータ・命令参照時は，最初に L0LS にアクセスする．それ以降の動作は High Frequency の場合における L0HS が L0LS に置き換わったものになる．

Low Frequency でのデータ・命令参照時も，最初に L0LS にアクセスする．L0LS で Hit となった場合，アクセス終了となるが，Miss した場合，L0HS にアクセスする[*3]．L0HS で Hit した場合，アクセスは終了となる．Miss の場合，High/Middle Frequency の場合と同様に L1 以降へアクセスする．

Low Frequency では，L1 以降で Hit した後にそのデータ・命令を L0 に書き込む場合，L0LS に書き込む．L0LS への書き込みの際に，既存のデータ・命令が追い出される

[*2] 本論文では，一貫性を保つために必要な処理と，その際に発生するエネルギーを考慮した評価を行っている．

[*3] L0HS は L0LS と同じ容量だが，Low Frequency では，2 つの L0 が同じデータを保持しないように動作するため，内容を確認する意味がある．

場合，それをL0HSに書き込む．これによって，2つのL0は，同じデータを保持しないように動作する．

5. 評価

5.1 評価環境

本研究の評価では，プロセッサシミュレータであるSimCore/Alpha Functional Simulator Version 2.0（SimCore）[20]と，キャッシュ分析モデルであるCACTI [21]を使用した．

SimCoreは，プロセッサアーキテクチャの研究と教育を目的として開発されており，C++によるコンパクトな実装で，多くのプラットフォームに対応している [22]．シミュレーション精度を低下させずに高速化するという設計方針を掲げており，論文 [23] によれば，SimpleScalarツールセット [27] のsim-fastと比較して，シミュレーション速度が19%改善されている．実際に，評価にSimCoreを使用している研究 [24], [25], [26] が複数存在する．

CACTIとは，HP研究所 [28] が公開しているキャッシュ分析モデルであり，プロセスルールやキャッシュサイズといった入力情報から，キャッシュのアクセス時間，面積，電力を出力する．HP研究所が公開しているテクニカルレポート [29] によると，CACTIの出力値と，シノプシスの回路シミュレーションツールHSPICE [30] との出力値の誤差は，平均で12%以内となっている．加えて，CACTIのモデルの計算量はHSPICEよりはるかに少なく，計算速度が高速である．実際に，評価にCACTIを使用している研究 [31], [32], [33], [34] が複数存在する．

評価にあたり，プロセッサシミュレータであるSimCoreに提案機構である階層キャッシュと周波数切替え機構を実装した．提案機構を実装したシミュレータ上のCPUは，プロセッサコアの最大周波数を2.0 GHzとし，DVFSによって周波数を0.1 GHzずつ，供給電圧を0.01 Vずつスケーリング可能であると想定している．各L0キャッシュの周波数と電圧値の対応は次のとおりである．

- L0HS：2.0 [GHz]/0.90 [V]～1.4 [GHz]/0.84 [V]
  - 1.3 [GHz] 以下は0.83 [V] 固定（L0MIX時も0.83 [V]）
- L0LS：1.3 [GHz]/0.83 [V]～0.9 [GHz]/0.79 [V]
  - 0.8 [GHz] 以下は0.78 [V] 固定（L0MIX時も0.78 [V]）

実装後のSimCoreを用いて，SPECint2006より8種類のベンチマークを用いて，10億命令を実行した際のキャッシュアクセス状況を測定した．使用したベンチマークは，bzip2, gobmk, h264ref, hmmer, mcf, omnetpp, perlbench, sjengの8種類である．なお，処理開始後の20億命令は，初期化フェーズとして評価から除外し，その後の10億命令を対象にしている．加えて，CACTIを用いて各キャッシュの1アクセスに要するエネルギーと時間を算出し，それぞれの結果より，シミュレーション時にキャッシュメモリが消費するアクセスエネルギーを計算した．

評価では，提案機構の消費エネルギーと，既存のL0キャッシュを用いた従来機構の消費エネルギーを測定し，提案機構の消費エネルギー削減率を計算する．比較用の従来機構（以下，Base）は，L0，L1，L2の3階層からなる．BaseのL0容量は，性能とヒット率の面で公平性を保つため，提案機構のL0MIXと同じ容量に設定するとともに，2.0 GHzでCPUが動作している際に，1サイクルでアクセスが完了するようにパラメータを調整している．L0Baseの周波数と電圧値の対応は次のとおりである．

- L0Base：2.0 [GHz]/0.90 [V]～1.4 [GHz]/0.84 [V]
  - 1.3 [GHz] 以下は0.83 [V] 固定

エネルギー削減率の計算では，L0とL1へのアクセスエネルギーの総量を求めており，L2へのアクセスエネルギーを含んでいない．これは，提案機構とBaseとの間のエネルギー差は，それぞれのL0のヒット率と，L0でミスした際に発生するL1へのアクセス回数の差で生じるためである．また，提案機構，BaseともにL1とL2は容量などの設計を同一とするため，L1でのヒット率には有意な差が生じない．このため，L2へのアクセス回数にも差が生じないので，L2へのアクセスエネルギーは提案機構，Baseともに評価に含めない．

キャッシュメモリは，実行する処理に応じて，最適となる容量が異なる．キャッシュ容量が処理内容に対して十分であるかどうかに応じて，容量や消費エネルギーの効率が変わってくる．一般的には，製品として既存のCPU内のキャッシュの容量を自由に変更することはできないため，使用頻度の高い変数を隣接するメモリセルに配置するなど，キャッシュの動作に最適化したコーディングが求められる．本研究の評価では，このような要素による提案機構への影響を最小限に抑えるため，8つのベンチマークの実行結果の平均から，最適なL0キャッシュ容量を決定した．

提案機構に用いるL0容量を決定するため，L0の容量を変化させながらベンチマークを実行し，面積と消費エネルギーの関係を求めた．その結果を**図 5** に示す．図を見ると分かるとおり，L0データキャッシュ4 kB，命令キャッシュ2 kBでエネルギーが最低になっている．そして，この容量より大きい範囲では消費エネルギーが増加していることから，すでにヒット率が飽和しており，その後のL0の容量の増加に合わせて大きくなったL0アクセスエネルギー分損失が発生していると読み取れる．

図5の結果より，提案機構に用いるL0キャッシュの容量を**表 2** のように定めた．あわせて本評価では，スマートフォンやタブレットPCのような高性能なスマートデバイスに搭載されるSoCを想定している．そのような用途で使用されているSoCのキャッシュ容量を参考に，L1はデータ・命令ともに64 kB，L2は1024 kBとしている．

BaseのL0キャッシュ容量を**表 3** に示す．BaseのL1，L2の容量は提案機構と同様である．

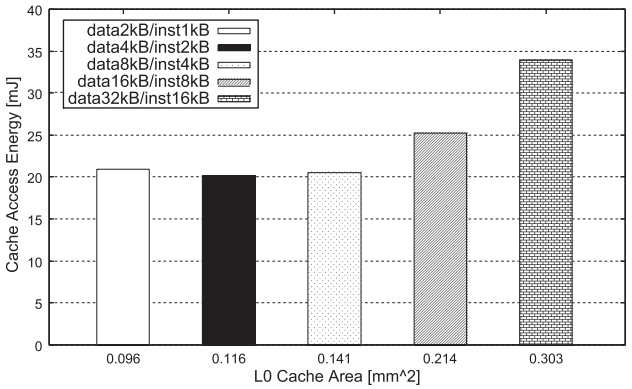


図 5 L0 キャッシュ面積と消費エネルギーの関係
Fig. 5 Relation between cache energy and L0 cache area.

表 2 提案機構の L0 容量と 1 サイクルアクセス可能な最大周波数
Table 2 L0 capacity in the proposed mechanism and the 1 cycle accessible maximum frequency.

	Cache Capacity	Max Freq.
L0 High Speed（L0HS）	Data 2 kB / Inst. 1 kB	2.0 GHz
L0 Low Speed（L0LS）	Data 2 kB / Inst. 1 kB	1.3 GHz
L0MIX（L0LS + L0HS）	Data 4 kB / Inst. 2 kB（Effective capacity）	0.8 GHz

表 3 比較用従来 L0 容量と 1 サイクルアクセス可能な最大周波数
Table 3 L0Base capacity and 1 cycle accessible maximal frequency.

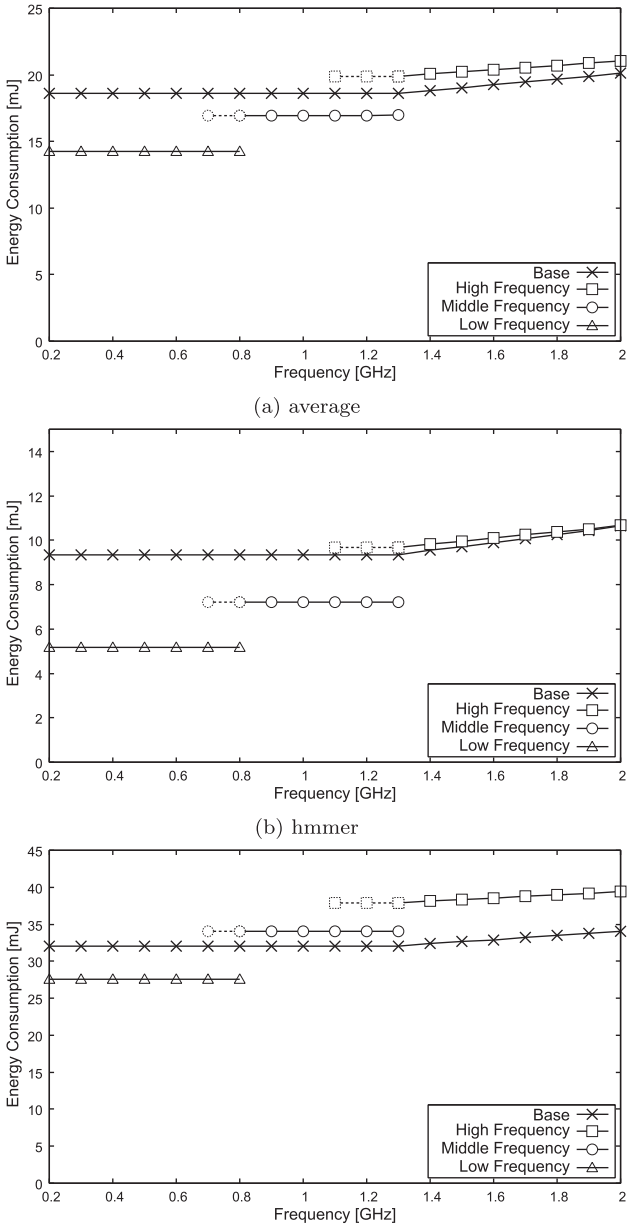
	Cache Capacity	Max Freq.
L0Base	Data 4 kB / Inst. 2 kB	2.0 GHz

前述のとおり，各キャッシュ設定には対応する周波数範囲が決まっている．先ほど定めた L0 キャッシュ容量を基に，CACTI を用いて，各周波数範囲で 1 サイクルアクセス可能なアクセス時間を持つ L0 キャッシュのエネルギーを算出した．その結果は次のようになっている．

- L0HS Inst.：3.39 [pJ/access], 0.30 [ns/access]
- L0HS Data：6.80 [pJ/access], 0.49 [ns/access]
- L0LS Inst.：1.55 [pJ/access], 0.60 [ns/access]
- L0LS Data：3.56 [pJ/access], 0.70 [ns/access]
- L0Base Inst.：4.99 [pJ/access], 0.28 [ns/access]
- L0Base Data：8.39 [pJ/access], 0.32 [ns/access]

評価は，次の 2 つの条件で実施する．1 つ目は DVFS による周波数の動的制御が発生しない環境での評価である．この場合，提案機構における各キャッシュ設定のエネルギーと周波数の関係を示すことになる．そして，その結果から，2 つ目の評価によって得られる結果の傾向を予測する．

2 つ目は，DVFS による周波数の動的制御が発生する環境で行う評価である．こちらでは，処理実行中の利用可能な余剰時間の変化を CPU の要求性能の変化と見立て，余剰時間を一定割合で変化させながらベンチマークを実行する．これによって，CPU の要求性能の変化に応じた消費エネルギーの傾向を評価する．



(a) average

(b) hmmer

(c) perlbench

図 6 周波数を固定して実行した場合の周波数ごとの消費エネルギー
Fig. 6 Overall energy under fixed frequency.

5.2 動的制御なしの評価

提案機構において，DVFS による周波数の動的制御を行わない場合の評価を示す．この場合，動的制御を行わないので，開始から終了までキャッシュ設定は対応する周波数ごとに一定である．この条件の結果より，5.3 節で行う周波数を動的制御した場合における，CPU の要求性能の変化に対するキャッシュ消費エネルギーの傾向を予測する．

図 6 に周波数・キャッシュ設定ごとの，8 つのベンチマークを実行した平均の結果と，代表的な 2 つのベンチマークの結果を示す．グラフでは，周波数範囲ごとに異な

る 3 つのキャッシュ設定で動作する提案機構の消費エネルギーと，周波数にかかわらず同じ設定で動作し続ける Base の消費エネルギーを示している．提案機構における周波数範囲と L0 キャッシュの設定の対応は，表 1 に従っている．

グラフ中では，High Frequency 範囲と Middle Frequency 範囲の結果の一部が点線で示されている．これは，それぞれ低消費エネルギーなキャッシュ設定に移行可能な周波数範囲に差し掛かった際に，設定を切り替えずに動作し続けるとした場合の消費エネルギーを示している．提案機構の消費エネルギーの状況を確認すると，移行可能な周波数範囲に差し掛かった段階で，より低消費エネルギーなキャッシュ設定を利用した方が総消費エネルギーが低下している．このような特性があるので，周波数の動的制御を行う場合，これらの周波数範囲に差し掛かった段階で，動的にキャッシュ設定を切り替えた方が有利になることが分かる．

Base と提案機構の消費エネルギーを図 6 (a) に示した平均の結果より比較する．High Frequency 範囲では，Base に対して提案機構の消費エネルギーが大きいことが確認できる．数値としては，1.7 GHz の段階で提案機構の方が Base より 5.5%大きい．これは，L0Base と L0HS の容量差により，提案機構では Base と比較して，L1 へのアクセス回数が増加することから生じている．

Middle Frequency 範囲では，提案機構の L0LS の容量は High Frequency 範囲の L0HS と同様であるが，1 アクセスあたりの消費エネルギーは約半分となっている．これによって，L1 へのアクセス回数の面で Base に劣る点は同様だが，総消費エネルギーでは提案機構の方が有利となる．1.0 GHz での消費エネルギーを比較すると，Base に対して，8.9%の削減が確認できる．

Low Frequency 範囲では，L0LS と L0HS を L0MIX として同時に使用するため，提案機構の実質的な L0 の容量は L0Base と同等になる．これにより，L0MIX におけるヒット率が L0Base と同等になり，L1 へのアクセスを抑えられるだけでなく，L0LS でアクセスが完結する頻度が高いほど，Base と比較して大幅な消費エネルギー削減が可能となる．ここで数値としては，23.5%のエネルギーが削減されている．

以上の結果より，CPU の要求性能の変化に応じて周波数を動的制御する場合，8 つのベンチマークの平均の消費エネルギーは，次のような傾向になると予測できる．

- CPU の要求性能が高く，全体の処理時間に対して，プロセッサコアを高い周波数で動作させる時間の割合が大きい場合，L0HS で動作する時間が長くなるため，提案機構の消費エネルギーは Base と同等か劣る．
- CPU の要求性能が低下し，全体の処理時間に対して，プロセッサコアの周波数を下げて動作させる時間の割合が大きくなるにつれて，L0LS や L0MIX を利用可能な時間が増加し，提案機構の消費エネルギーが小さくなっていく．

実際に 5.3 節において，CPU の要求性能を変化させながら，プロセッサコアの周波数を動的制御する場合の評価を行うことで，この予測が実際の傾向と一致するかを確かめる．

次に，代表的な 2 つのベンチマークの結果を確認する．図 6 (b) では，最も提案機構が有効に働いているベンチマークの例として hmmer の結果をあげた．hmmer における消費エネルギーの推移は，平均の結果と同様の傾向を示している．

図 6 (c) では，提案機構の効果が小さいベンチマークの例として perlbench の結果をあげた．perlbench では，High Frequency 範囲だけでなく，Middle Frequency 範囲でも提案機構の消費エネルギーが Base より大きい．これは，提案機構における L0LS のヒット率と L0Base とのヒット率の差によって生じている．

perlbench における命令キャッシュのヒット率は，L0Base が 91.10%，L0LS が 89.93%，データキャッシュのヒット率は，L0Base が 88.18%，L0LS が 78.10%となっている．このうち，データキャッシュのヒット率の差である 10.08%は，評価に使用した 8 つのベンチマーク中で最大となっている．つまり，本評価では，perlbench において，Base と比較した提案機構の相対的な L1 データキャッシュへのアクセス回数が最大となっている．L0LS へのアクセス後に L1 へアクセスする頻度が高いため，1 度のメモリ参照で両者のアクセスエネルギーを消費する頻度が高くなる．これは平均の結果でも同様の傾向が見られたが，perlbench においては，この頻度が L0LS による消費エネルギー削減効果を無効化してしまうほど高い．その結果，Middle Frequency 範囲でも，提案機構の消費エネルギーが Base に劣っている．

一方，hmmer と perlbench のどちらにおいても，Low Frequency 範囲における消費エネルギーは提案機構の方が小さくなっている．このことから，周波数が十分低下した環境での提案機構の有効性がうかがえる．

次の 5.3 節において，周波数を動的制御する場合の評価でも，この 2 つのベンチマークの消費エネルギーを確認する．

5.3 動的制御ありの評価

提案機構において，DVFS による周波数の動的制御を行う場合の評価を示す．この場合，処理の実行中に発生するコアの周波数の切替えにともない，キャッシュ設定の動的切替えが発生する．

この評価では，利用可能な余剰時間の長さ（図 1 における Idle）を CPU の要求性能の高さと見立て，それを変化させながらベンチマークを実行していく．たとえば，利用可能な余剰時間が長くなるほど，CPU の要求性能が低くなっているものとする．これにより，CPU の要求性能の

変化に対するキャッシュ消費エネルギーの傾向を得ることができる．得られた結果より，5.2 節において予測した傾向と一致するか確認する．

この評価における周波数およびキャッシュ設定の切替えは，4.1 節において記載した方法で行う．また，デッドラインについては，Base でのベンチマーク実行時間を基準とする．Base でベンチマークを実行した際の 100 万命令ごとの処理の完了時間を，その区間のデッドラインとしてあらかじめ測定しておく．このデッドラインをどれだけ伸長するかによって，利用可能な余剰時間の長さが変化することになる．加えて，余剰時間の長さを正規化して表現する．正規化する基準として，2.0 GHz と 0.2 GHz で Base において測定したデッドラインを用いる．周波数 2.0 GHz でのデッドラインに対する余剰時間を，余剰時間 0%と表現し，周波数 0.2 GHz でのデッドラインに対する余剰時間を，余剰時間 100%と表現する．

図 7 に CPU の要求性能の変化による，利用可能な余剰時間の変化に応じた評価の結果を示す．こちらも 5.2 節と同様に，8 つのベンチマークを実行した平均と，代表的な 2 つのベンチマークの結果を示している．グラフでは，横軸である利用可能な余剰時間の割合について，最大値を 30%としている．これは，余剰時間 20%前後で提案機構の消費エネルギー削減が飽和状態に至ったためである．

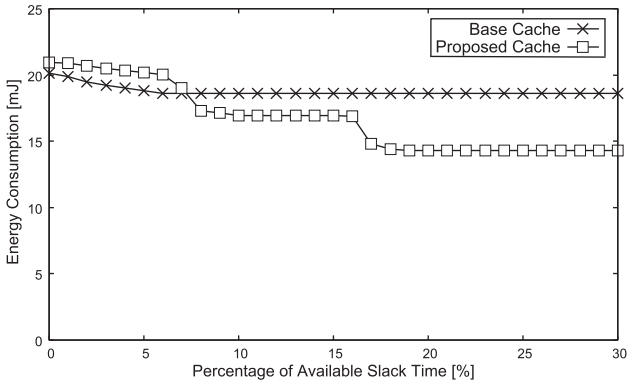
平均の結果を確認すると，余剰時間 7%前後までは，Base の消費エネルギーが提案機構より小さい．この範囲では，提案機構が High Frequency 範囲の設定で動作している割合が大きいため，5.2 節で確認したとおり，エネルギーの面で不利になっている．余剰時間 8%から，CPU の要求性能が低下し始め，提案機構が Middle Frequency 範囲の設定で動作する割合が大きくなることにより，消費エネルギーが Base より小さくなっていく．さらに，余剰時間 17%から，提案機構が Low Frequency 範囲の設定で動作する割合が大きくなり，さらなる消費エネルギー削減が確認できる．

具体的な数値を確認すると，余剰時間 4%において提案機構は Base より 6.8%消費エネルギーが大きいが，12%では 8.9%削減され，最終的に 20%では 23.2%削減されている．

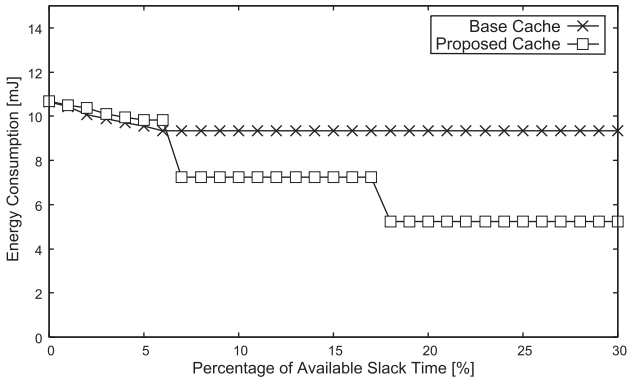
以上より提案機構では，CPU の要求性能が低下し，利用可能な余剰時間が増えるにつれて，消費エネルギーの小さいキャッシュ設定で動作する割合が大きくなっていく．また，それによって消費エネルギーの削減率が向上していく傾向が確認できる．この傾向は，5.2 節で予測した傾向と一致している．

5.2 節に引き続き，図 7 (b) に hmmer の結果を，図 7 (c) に perlbench の結果を示した．5.2 節と同様に hmmer の結果は，平均の結果と同様の傾向を示しており，余剰時間 7%から提案機構の消費エネルギーが Base の消費エネルギーより小さくなっている．一方，perlbench では，余剰時間の割合が増えるに応じて，提案機構の消費エネルギーが下がっているものの，Middle Frequency 範囲での動作が中心になる余剰時間 8%から 16%においても，依然として提案機構の消費エネルギーが Base より大きい．これは，5.2 節で確認したとおり，perlbench における L0LS と L0Base とのヒット率の差と，それによる L1 へのアクセス頻度の差が影響している．

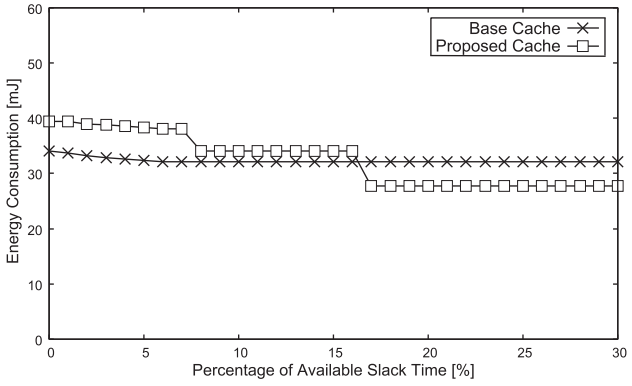
perlbench の結果から，Middle Frequency 範囲における提案機構のエネルギー削減効果が，ベンチマークの特性に依存して小さくなってしまう状況が存在することが確認できる．この問題を改善できれば，提案機構のさらなる省電力化が可能になると考えられる．



(a) average



(b) hmmer



(c) perlbench

図 7 周波数を動的に切り替えた場合の利用可能な余剰時間ごとの消費エネルギー

Fig. 7 Overall energy under DVFS environment.

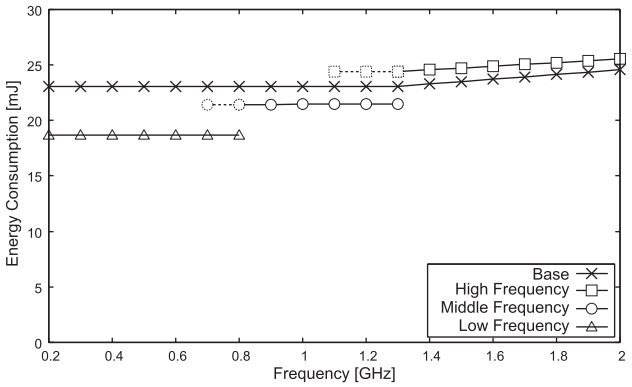


図 8 L2 を含めた動的制御なしの評価
Fig. 8 Evaluation including L2 by fixed frequency.

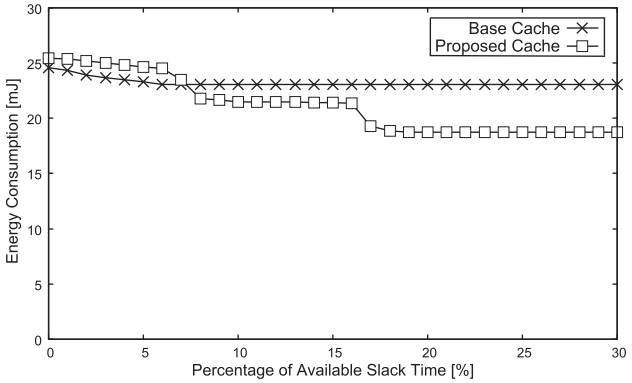


図 9 L2 を含めた動的制御ありの評価
Fig. 9 Evaluation including L2 by dynamically switching.

5.4 L2 のエネルギーを含めた場合の評価

図 8 と**図 9** に，5.2 節および 5.3 節と同様の評価を L2 のエネルギーを加えて行った結果を示す．図 8 および図 9 は，ベンチマーク平均の結果を示している．

図 8 は，5.2 節と同様に，キャッシュの動的制御を行わない場合の評価結果である．L2 を含めていない図 6 (a) と比較すると，グラフの傾向は変わっていないことが分かる．一方，L2 の動的エネルギーを含めた関係上，縦軸の最大値が 25 mJ から 30 mJ に増加している．最大の削減率を確認すると，図 6 (a) では，23.5%であったのに対し，図 8 では 18.9%となっている．

図 9 は，5.3 節と同様に，キャッシュの動的制御を行う場合の評価結果である．L2 を含めていない図 7 (a) と比較すると，こちらもグラフの傾向は変わっていないことが分かる．動的制御を行わない場合と同様に，縦軸の最大値は 25 mJ から 30 mJ に増加している．最大の削減率を確認すると，図 7 (a) では，23.2%であったのに対し，図 9 では 18.7%となっている．

以上より，L2 を含めた場合の評価結果は，エネルギーの削減率に若干の悪化が見られるものの，L2 を含めない場合と同様の傾向を示している．したがって，以降の評価も L2 を含めずに実施する．

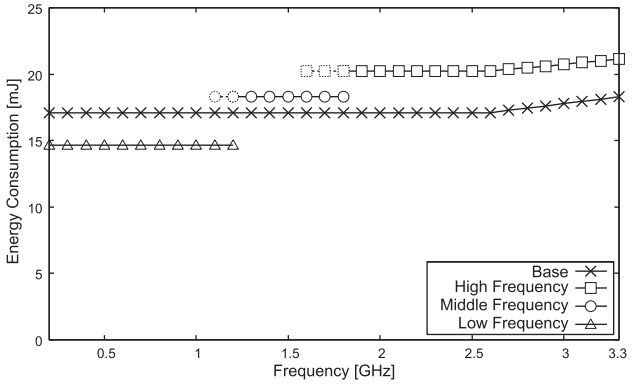


図 10 L0，L1 容量を半分にした場合の評価
Fig. 10 Evaluation when L0 and L1 capacity are halved.

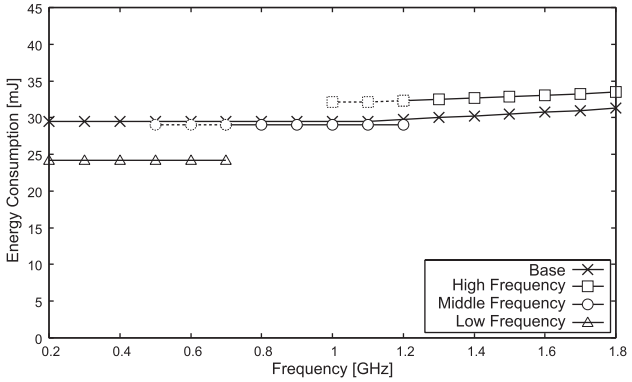


図 11 L0，L1 容量を 2 倍にした場合の評価
Fig. 11 Evaluation when L0 and L1 capacity are doubled.

5.5 L0，L1 容量を変化させた場合の評価

図 10 と**図 11** に，L0 と L1 の容量を変化させて，動的制御なしの評価を行った結果を示す．この評価では，5.2 節および 5.3 節での評価に用いた L0 および L1 の容量を変化させた場合の消費エネルギーと対応可能な周波数範囲の変化を確認する．図 10 および図 11 は，ベンチマーク平均の結果を示している．

5.2 節および 5.3 節の評価で用いた L0 と L1 の容量と面積は以下のとおりである．

- L0HS Inst.：1 [kB], 0.029 [mm^2]
- L0HS Data：2 [kB], 0.301 [mm^2]
- L0LS Inst.：1 [kB], 0.027 [mm^2]
- L0LS Data：2 [kB], 0.166 [mm^2]
- L0Base Inst.：2 [kB], 0.019 [mm^2]
- L0Base Data：4 [kB], 0.097 [mm^2]
- L1 Inst.：64 [kB], 1.158 [mm^2]
- L1 Data：64 [kB], 1.720 [mm^2]

図 10 は，上記の L0 と L1 の容量をすべて半分にした場合の結果である．このとき，各キャッシュ容量と面積は次のとおりである．

- L0HS Inst.：0.5 [kB], 0.015 [mm^2]
- L0HS Data：1 [kB], 0.152 [mm^2]
- L0LS Inst.：0.5 [kB], 0.015 [mm^2]

- L0LS Data：1 [kB], 0.152 [mm^2]
- L0Base Inst.：1 [kB], 0.010 [mm^2]
- L0Base Data：2 [kB], 0.086 [mm^2]
- L1 Inst.：32 [kB], 0.775 [mm^2]
- L1 Data：32 [kB], 2.324 [mm^2]

5.2 節で行った評価の結果である図 6 (a) と比較すると，各 L0 が 1 サイクルでアクセスを完了できる最大周波数が大きくなっている一方で，Middle Frequency での消費エネルギーが Base より大きくなっている．これは，L0 の容量が半分になっているためにアクセスに要する時間が短くなった一方で，L0 のヒット率低下によって，L1 へのアクセス回数が増加したことが原因と考えられる．

図 11 は，上記の L0 と L1 の容量をすべて 2 倍にした場合の結果である．このとき，各キャッシュ容量と面積は次のとおりである．

- L0HS Inst.：2 [kB], 0.035 [mm^2]
- L0HS Data：4 [kB], 0.330 [mm^2]
- L0LS Inst.：2 [kB], 0.029 [mm^2]
- L0LS Data：4 [kB], 0.197 [mm^2]
- L0Base Inst.：4 [kB], 0.019 [mm^2]
- L0Base Data：8 [kB], 0.122 [mm^2]
- L1 Inst.：128 [kB], 2.074 [mm^2]
- L1 Data：128 [kB], 8.403 [mm^2]

先ほどと同様に，図 6 (a) と比較すると，各 L0 が 1 サイクルでアクセスを完了できる最大周波数が小さくなっている．加えて，Middle Frequency と Base の消費エネルギーの差が小さくなっており，提案機構の消費エネルギー削減率が悪化している．

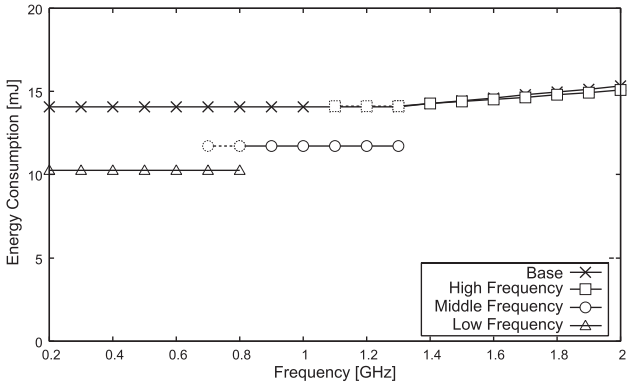
上記の結果から，提案機構が有効に働く容量は，5.2 節と 5.3 節で使用した容量であったことが分かる．提案機構を効果的に運用するには，5.1 節の図 5 で行った事前評価のように，使用する環境に合わせて最適な容量を設定する必要がある．

次に，5.2 節および 5.3 節の評価で使用したキャッシュの面積から，提案機構のハードウェアオーバヘッドを検討する．上述した，5.2 節および 5.3 節で使用した各キャッシュの面積を確認すると，提案機構は，L0LS，L0HS，L1 の合計で 3.401 mm^2 に対し，従来機構の面積は，L0Base と L1 の合計で 2.994 mm^2 となっている．したがって，提案機構の導入により，面積が 13.6%増加することになる．

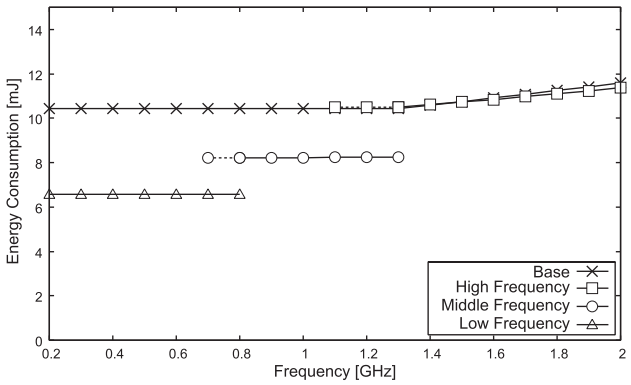
5.6 SPECfp2006 による評価

5.2 節および 5.3 節と同様の評価を SPECfp2006 ベンチマークを使用して行った結果を示す．評価では，SPECfp2006 より 5 種類のベンチマークを使用した：dealII，milc，namd，soplex，sphinx3.

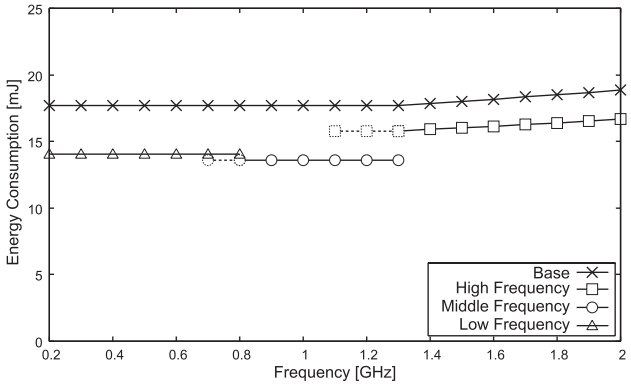
図 12 は，5.2 節の評価と同様にキャッシュの動的制御を行わない場合の結果である．グラフは，5 つのベンチマーク平均のほかに，提案機構が有効に働いているベンチマークと，有効に働いていないベンチマークの結果を示している．



(a) average



(b) soplex

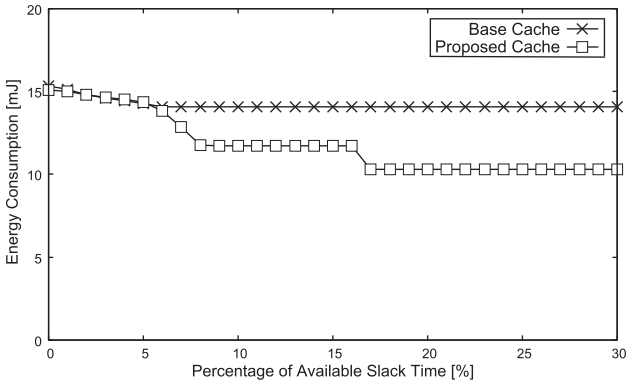


(c) dealII

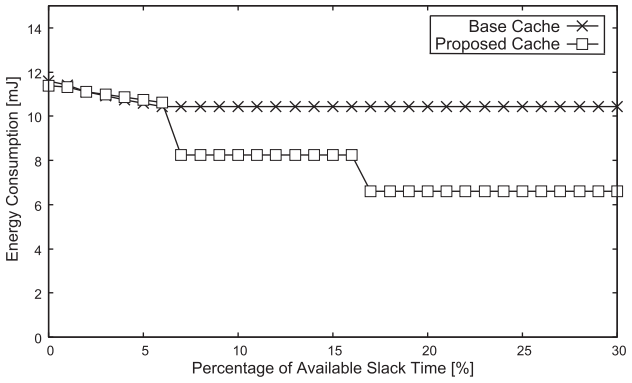
図 12 動的制御なしの評価（SPECfp2006）
Fig. 12 Evaluation by fixed frequency (SPECfp2006).

図 12 (a) は，平均の結果である．High Frequency では，提案機構は Base とほぼ同等の消費エネルギーとなっている．Middle Frequency 以降では，提案機構の消費エネルギーが Base より削減されている．数値を確認すると，Middle Frequency の 1.0 GHz で 16.9%，Low Frequency の範囲で 27.2%の削減率となっている．

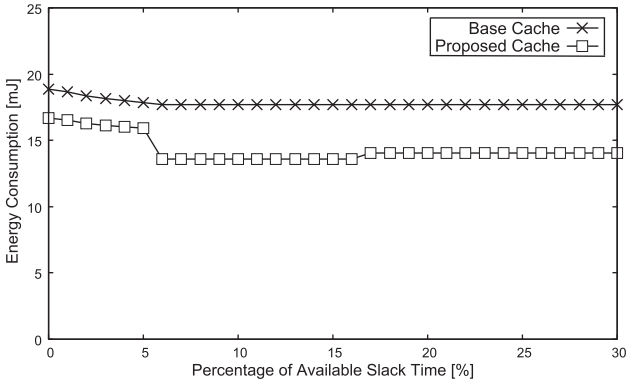
図 12 (b) は，soplex の結果である．これは，提案機構が有効に働いている例としてあげた．消費エネルギーの削減率は，1.0 GHz で 21.0%，Low Frequency の範囲で 36.8%となっている．



(a) average



(b) soplex



(c) dealII

図 13 動的制御ありの評価（SPECfp2006）
Fig. 13 Evaluation by dynamically switching (SPECfp2006).

図 12 (c) は，dealII の結果である．これは，提案機構が有効に働いていない例としてあげた．提案機構と Base の消費エネルギーを比較すると，つねに提案機構の方が小さくなっているが，Low Frequency の消費エネルギーが Middle Frequnecy の消費エネルギーより大きくなっているなど，提案機構の想定と異なっている．

dealII では，Base，High Frequency，Middle Frequency，Low Frequency のすべてで，最上位キャッシュとなる L0 のヒット率が，命令キャッシュは 100%，データキャッシュは 87.5%となっている．加えて，Low Frequency では，L0LS のミス後にアクセスされる L0HS のヒット率が，0%となっている．

これらのことから，dealII では，L0HS や L0LS より大きな容量の L0（L0Base，L0MIX）を使用していても，L1 キャッシュへのアクセス回数が低下しないことが分かる．結果的に，提案機構の L0LS および L0HS の容量で十分であるため，L0Base や L0MIX の容量が余剰になっており，容量に合わせて増加した動的エネルギーが無駄に消費される状態となっている．

図 13 は，5.3 節の評価と同様にキャッシュの動的制御を行う場合の結果である．図 12 と同様に，ベンチマーク平均，soplex，dealII の結果を記載している．

図 13 (a) は，平均の結果である．5.3 節での評価結果と同様に，CPU の要求性能が低下し，利用可能な余剰時間の割合が増えるにつれて，提案機構が消費エネルギーの小さい設定で動作する割合が増え，消費エネルギーが低下していく様子が確認できる．

具体的な数値を確認すると，余剰時間 4%において提案機構は Base より 0.5%消費エネルギーが大きいが，12%では 16.7%削減され，最終的に 20%では 26.9%削減されている．

図 13 (b) は，soplex の結果である．soplex は，平均と同様の傾向を示しており，余剰時間 7%から大幅な消費エネルギー削減が確認できる．

図 13 (c) は，dealII の結果である．動的制御なしの場合と同様に，つねに提案機構の消費エネルギーが Base より小さくなっている．余剰時間 6%から 16%にかけては，Middle Frequency 範囲の設定での動作が中心になっていると考えられる．一方，余剰時間 17%以降は，Low Frequency 範囲の設定での動作が中心になっていると考えられる．両者の消費エネルギーを比較すると，後者の方が大きくなっている．これは，図 12 (c) で確認した内容と一致している．

以上のことから，SPECfp2006 では，SPECint2006 と同様に，提案機構が有効に働くベンチマークと，有効に働かないベンチマークが存在することが分かる．

6. 提案機構が有効に働く構成の検討

5 章全体の評価結果から，提案機構が有効に働くキャッシュ構成について検討する．

5.1 節の図 5 で行った事前評価のように，本論文では，ベンチマーク平均でエネルギーと面積の効率が最適となるように L0Base と L0MIX の容量を選択していた．ここで，エネルギーと面積の効率が最適になるためには，処理の内容に対して，L0Base および L0MIX が必要最低限の容量であることが要求される．必要最低限の容量とは，処理中のある期間内で高頻度に参照されるデータ・命令（以下，“ワーキングセット” と表記）を格納できる容量を指す．

ワーキングセットより L0MIX の容量が大きすぎる場合，キャッシュの容量の一部がほとんど使われない一方で，消費エネルギーが大きくなる．また，L0HS や L0LS から L0MIX へ切り替わった際の，容量増加の恩恵も受けられなくなる．5.6 節における dealII の評価結果において，提

案機構が有効に働いていなかったのは，このためと考えられる．

ワーキングセットよりL0MIXの容量が小さすぎる場合，キャッシュのヒット率が低下し，L1以降のキャッシュへのアクセス回数が増加するため，キャッシュ全体としては消費エネルギーが大きくなる．この理由により提案機構が有効に働いていなかった例として，5.2節および5.3節におけるperlbenchの評価結果があげられる．

以上より，提案機構が有効に働くキャッシュ構成を実現するためには，ワーキングセットに対して必要最低限の容量にL0MIXを調整する必要があることが分かる．

7. まとめ

本研究では，バッテリによって動作するIoTデバイスの動作時間を延ばすために，CPUに搭載されるキャッシュメモリの消費エネルギー削減に着目した．提案機構では，CPUの省電力化に用いられる既存手法であるDVFSによる周波数・電圧制御を利用し，CPUの要求性能の変化に合わせて，L0キャッシュを切り替えた．具体的には，プロセッサコアの周波数低下によって動作速度が低下した場合に，動作速度が遅いが消費エネルギーが小さいL0キャッシュを使用するように設定を変更した．これにより，L0キャッシュのアクセスにともなう消費エネルギーの削減を実現した．数値としては，SPECint2006で最大23.5%，SPECfp2006で最大27.2%の消費エネルギー削減を確認している．

謝辞 本研究の一部は，JSPS科研費17K00076の支援により行った．

参考文献

[1] Apple：バッテリーに関する一般情報，入手先 ⟨https://www.apple.com/jp/watch/battery.html⟩（参照 2017-04-24）．

[2] 総務省：平成28年版情報通信白書第1部第2章IoT時代におけるICT産業動向分析，入手先 ⟨http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/html/nc120000.html⟩（参照 2017-05-19）．

[3] Kamble, M.B. and Ghose, K.: Analytical energy dissipation models for low-power caches, *ISLPED '97, Proc. 1997 international symposium on Low power electronics and design*, pp.143–148 (online), DOI: 10.1145/263272.263310 (1997).

[4] Grabher, P., Großschädl, J., Hoerder, S., Järvinen, K., et al.: An exploration of mechanisms for dynamic cryptographic instruction set extension, *Journal of Cryptographic Engineering*, Vol.2, pp.1–18 (online), DOI: 10.1007/s13389-011-0025-8 (2012).

[5] Intel: Intel 22nm 3-D Tri-Gate Transistor Technology (22nm-Announcement Presentation.pdf), available from ⟨https://newsroom.intel.com/wp-content/uploads/sites/11/2011/05/22nmAnnouncement_Presentation.pdf⟩ (accessed 2017-09-21).

[6] Intel: Intel 22 nm FinFET Low Power (22FFL) Technology: FinFET Technology for the Mainstream, available from ⟨https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2017/03/22-nm-finfet-fact-sheet.pdf⟩ (accessed 2017-09-21).

[7] Le Sueur, E. and Heiser, G.: Dynamic Voltage and Frequency Scaling: The Laws of Diminishing Returns, *Proc. 2010 International Conference on Power Aware Computing and Systems*, pp.1–8, (2010).

[8] Weste, N.H.E. and Eshraghian, K.: *Principles of CMOS VLSI Design: A Systems Perspective*, Addison Wesley Longman Publishing Co. (1985).

[9] Tang, A. and Jha, N.K.: Design space exploration of FinFET cache, *ACM Journal on Emerging Technologies in Computing Systems* (*JETC*), Vol.9, No.3, pp.20:1–20:16 (online), DOI: 10.1145/2491678 (2013).

[10] 北川英二，藤田 忍，伊藤順一：低消費電力のキャッシュメモリを可能にした垂直STT-MRAM，東芝レビュー，Vol.68, No.6, pp.44–47 (2013).

[11] Flautner, K., Kim, N.S., Martin, S., Blaauw, D., et al.: Drowsy caches: Simple techniques for reducing leakage power, *SIGARCH Comput. Archit. News*, Vol.30, No.2, pp.148–157 (online), DOI: 10.1145/545214.545232 (2002).

[12] Sakanaka, A. and Sato, T.: Reducing static energy of cache memories via prediction-table-less way prediction, *Proc. 13th International Workshop on Integrated Circuit and System Design, Power and Timing Modeling, Optimization and Simulation*: *PATMOS 2003*, Turin, Italy, pp.530–539 (online), DOI: 10.1007/978-3-540-39762-5_59 (2003).

[13] Yang, S., Powell, M.D., Falsafi, B., Roy, K. and Vijaykumar, T.N.: An integrated circuit/architecture approach to reducing leakage in deep-submicron high-performance I-caches, *Proc. HPCA 7th International Symposium on High-Performance Computer Architecture, Monterrey, 2001*, pp.147–157 (online), DOI: 10.1109/HPCA.2001.903259 (2001).

[14] Watanabe, K., Sasaki, T., Nakabayashi, T., Ohno, K. and Kondo, T.: Reducing dynamic energy of variable level cache, *International Journal of Computer and Electrical Engineering*, Vol.5, No.6, pp.581–586 (online), DOI: 10.7763/IJCEE.2013.V5.777 (2013).

[15] Su, C.L. and Despain, A.M.: Cache design trade-offs for power and performance optimization: A case study, *ISLPED'95*, *Proc. 1995 International Symposium on Low Power Design*, pp.63–68 (online), DOI: 10.1145/224081.224093 (1995).

[16] Hajj, N.B.I., Polychronopoulos, C. and Stamoulis, G.: Architectural and compiler support for energy reduction in the memory hierarchy of high performance microprocessors, *Proc. 1998 International Symposium on Low Power Electronics and Design* (IEEE Cat. No.98TH8379) (online), pp.70–75, DOI: 10.1145/280756.280788 (1998).

[17] Intel Corporation: Intel Turbo Boost Technology in Intel Core Microarchitecture (Nehalem) Based Processors, Whitepaper, Intel Corporation (2008).

[18] Advanced Micro Devices Inc.: AMD PowerTune Technology, Whitepaper, Advanced Micro Devices, Inc. (2012).

[19] 宮本寛史，飯山真一，冨山宏之，高田広章，中島 浩：動的計画法を用いたキャッシュフラッシュの最悪タイミングの探索手法，情報処理学会論文誌，Vol.46, No.SIG16 (ACS12), pp.85–94 (2005).

[20] SimCore/Alpha Functional Simulator Homepage, available from ⟨http://www.arch.cs.titech.ac.jp/~kise/

SimCore/functional.htm⟩ (accessed 2017-09-12).
[21] HP Labs: CACTI, available from ⟨http://www.hpl.hp.com/research/cacti/⟩ (accessed 2017-09-12).
[22] Kise Laboratory, available from ⟨http://www.arch.cs.titech.ac.jp/⟩ (accessed 2017-09-12).
[23] Kise, K., Katagiri, T., Honda, H. and Yuba, T.: The SimCore/Alpha Functional Simulator, *WCAE '04, Proc. 2004 Workshop on Computer Architecture Education: Held in Conjunction with the 31st International Symposium on Computer Architecture*, No.24, pp.128–135 (online), DOI: 10.1145/1275571.1275602 (2004).
[24] 小川周吾，菅原 豊，平木 敬：TLB を用いるキャッシュ利用状況推定の高精度化，情報処理学会研究報告計算機アーキテクチャ（ARC），No.79 (2007-ARC-174), pp.25–30 (2007).
[25] 二ノ宮康之，阿部公輝：パーセプトロン分岐予測器を用いた予測ミスする分岐命令の効率的分離，情報処理学会研究報告計算機アーキテクチャ（ARC），No.75 (2008-ARC-179), pp.55–60 (2008).
[26] 山本浩暉，小林良太郎，嶋田 創：キャッシュライン中の分岐命令数に着目した BTB の消費エネルギー削減（コンピュータシステム），電子情報通信学会技術研究報告 = IEICE Technical Report：信学技報，Vol.114, pp.89–94 (2005).
[27] SimpleScalar LLC to serve and project, available from ⟨http://www.simplescalar.com/⟩ (accessed 2017-09-12).
[28] HP Labs, available from ⟨http://www8.hp.com/us/en/hp-labs/index.html⟩ (accessed 2017-09-11).
[29] Muralimanohar, N., Balasubramonian, R. and Jouppi, N.P.: *CACTI 6.0: A Tool to Model Large Caches*, HP Laboratories (online), HPL-2009-85 (2009).
[30] Synopsys: HSPICE, available from ⟨https://www.synopsys.com/verification/ams-verification/circuit-simulation/hspice.html⟩ (accessed 2017-09-12).
[31] Kong, J. and Lee, K.: A DVFS-aware cache bypassing technique for multiple clock domain mobile SoCs, *IEICE Electronics Express*, Vol.14, No.11, pp.20170324–20170324 (online), DOI: 10.1587/elex.14.20170324 (2017).
[32] Kim, S., Lee, J., Kim, J. and Hong, S.: Residue cache: A low-energy low-area L2 cache architecture via compression and partial hits, *MICRO-44, Proc. 44th Annual IEEE/ACM International Symposium on Microarchitecture*, pp.420–429 (online), DOI: 10.1145/2155620.2155670 (2011).
[33] Cheng, H.-Y., Poremba, M., Shahidi, N., Stalev, I., et al.: EECache: A Comprehensive study on the architectural design for energy-efficient last-level caches in chip multiprocessors, *ACM Trans. Architecture and Code Optimization* (*TACO*), Vol.12, No.17, pp.17:1–17:22 (online), DOI: 10.1145/2756552 (2015).
[34] Lee, J., Hong, S. and Kim, S.: TLB index-based tagging for cache energy reduction, *ISLPED '11, Proc. 17th IEEE/ACM International Symposium on Low-power Electronics and Design*, pp.85–90 (online), ISBN: 978-1-61284-660-6 (2011).

齋藤 郁

2016 年豊橋技術科学大学工学部情報工学課程卒業．同年豊橋技術科学大学大学院工学研究科情報工学専攻博士前期課程入学．計算機アーキテクチャの研究に従事．

小林 良太郎 （正会員）

1995 年名古屋大学工学部電子情報学科卒業．1997 年名古屋大学大学院工学研究科電子情報学専攻博士課程前期課程修了．2000 年名古屋大学大学院工学研究科電子情報学専攻博士課程後期課程満了．工学博士．2000 年名古屋大学大学院工学研究科電子情報学専攻助手．2008 年豊橋技術科学大学工学部講師．2016 年豊橋技術科学大学大学院工学研究科准教授．2017 年工学院大学情報学部准教授．1999 年情報処理学会山下記念研究賞受賞．2002 年情報処理学会論文賞受賞．計算機アーキテクチャ，ネットワークセキュリティの研究に従事．本会シニア会員．

嶋田 創 （正会員）

1976 年生．1998 年名古屋大学工学部情報工学科卒業．2000 年名古屋大学大学院工学研究科情報工学専攻博士前期課程修了．2004 年名古屋大学工学博士．2004 年名古屋大学工学部電気系 COE 研究員．2005 年京都大学大学院情報学研究科特任助手．2006 年京都大学大学院情報学研究科助手．2009 年奈良先端大学院大学情報科学研究科准教授．2013 年名古屋大学情報基盤センター情報基盤ネットワーク研究部門准教授．低消費電力と高信頼性を兼ね備えた計算機アーキテクチャとネットワーク関連の研究に従事．電子情報通信学会，IEEE 各会員．本会シニア会員．