

## バッファの使用量に着目したプロセッサ周波数変更による低消費電力化手法

熊谷晋† 杉野栄二‡ 瀬川典久‡ 澤本潤‡

†岩手県立大学ソフトウェア情報学研究科

‡岩手県立大学ソフトウェア情報学部

近年、携帯電話を始め多くの電子機器が持ち運ばれるようになった。持ち運び可能な電子機器は電池で動作することが多く、電池がどの程度持つかが重要である。本研究ではまず、電池で動作するポータブル電子機器を想定し電池を長持ちさせるための周波数変更手法について考察を行った。さらに、音声ファイル再生時のオーディオデータバッファの使用量に着目したプロセッサの周波数変更を行う手法を提案し、その手法を音楽再生が可能な評価ボード上に実装した。その結果、周波数を最適な値に変更することが可能となった。

### Dynamic Frequency Scaling for Low Power Computing Based on Usage Rate of Buffer

Shin Kumagai† Eiji Sugino‡ Norihisa Segawa‡ Jun Sawamoto‡

† Graduate School of Software and Information Science, Iwate Prefectural University

‡ Software and Information Science, Iwate Prefectural University

This paper proposes method of Dynamic Frequency Scaling based on usage rate of audio data buffer. First, we consider methods for long battery life and propose Dynamic Frequency Scaling for Audio Playback. We implemented the method for test board and evaluated.

#### 1. はじめに

近年、ポータブルミュージックプレイヤーなどのポータブル電子機器が数多く出回っている。これらの機器は電池で動作することが多く、バッテリ

がどの程度持つかが重要となっている。最近では消費電力の増大から、PCやサーバのような非ポータブル機器においても低消費電力化のための研究がされている。これらの低消費電力化手法を実現

するために、ソフトウェアの制御でプロセッサに供給する電圧や周波数を変更可能なハードウェアを用いるのが一般的である。

電圧や周波数を動的に変更する DVFS(Dynamic Voltage and Frequency Scaling)の研究は数多く行われているが、シミュレーションでの評価や、PC のような汎用に利用するコンピュータ用実装している例が多い。また、事前にプログラムに対してプロフィールを作り、その情報を基に周波数の制御を行うものが多い。

そこで、本研究では電池で動作するポータブルミュージックプレイヤーのような、使用目的が限られており、ストリーム処理を行う機器を想定し、バッファのアクセス状況からプロセスの周波数を決定し、低消費電力化するための手法を考察、提案する。また、実際に音楽再生が可能な評価ボード上に実装し評価を行う。

## 2. 消費電力と電池

一般的に、プロセッサで消費される電力は電圧の2乗に比例し、周波数と比例する。周波数を下げることが出来れば電圧を下げることも可能となる。よって、プロセッサで使用される電力と性能の間にはトレードオフの関係があるといえる。しかし電池は高出力時に放電効率が悪くなるため、低出力利用時と比べ、放出できる電池容量が少なくなるという特徴がある。そのため同じ処理をする場合、デッドラインを守りつつなるべく低い一定の周波数を保つのが好ましいとされている。MINBAT[1]では Li イオン電池の放電特性を考慮し、同一周波数で動作させることが可能なタスクをタスクブロック単位で管理することにより、電池の消費量を減らすことが出来るとしている。しかし評価が動的に周波数を変更せずに一定のクロックで実行した場合との比較でされており、また、実装を行ってデータをとっていないため、電池の

放電特性によってどの程度の電池の有効利用が可能となっているかについては言及されていない。本研究ではまず、消費電力の少ない組み込み用の評価ボードで放電特性を考慮した周波数切り替えがどの程度有効であるかについて考察するために予備実験を行った。

### 2.1. 予備実験

#### 2.1.1 実験目的

同じ処理を同じ時間で処理する場合に、同一周波数での実行時と、高い周波数と低い周波数を交互に実行した場合で、どれだけ電池の持ちに差が出るかを調べる。

#### 2.1.2 実験内容

評価ボード上で決まった回数を単純にループし、シリアルポートに文字を出力させる処理を繰り返し行い、電池の持続時間と、電池の出力電圧について調べた。一定の周波数で実行するプログラムは周波数を 18MHz に設定し、2つの周波数で交互に実行させるプログラムは周波数を 12MHz と 24MHz に設定した。一定時間間隔でのループ処理量を等しくするために、18MHz のプログラムではループを 15 万回、12MHz で 10 万回、20MHz で 20 万回とした。

#### 2.1.3 実験環境

実験には、ファインアーク(株)の FS-500 を使用した。FS-500 は独自開発の DSP コアと CPU コアをワンチップ化したマルチコア CPU を搭載しており、CPU コアと DSP コアに処理を分担することにより、低クロックで音楽再生のような処理を行うことが出来、低消費電力化が可能である。ただし本論文のプログラムでは DSP コアを使用していない。

電池は日立マクセル(株)の 6LF22 を使用した。

6LF22 はアルカリ電池セルを 6 つ直列にした 9V 電池である。電池は各実験に新品を利用した。実験は図 1 のようにして行った。

#### 2.1.4 予備実験結果

予備実験の結果、12MHz と 24MHz の周波数を交代させて実行した場合の電池の持続時間は、29995 秒(8 時間 19 分 55 秒)、18MHz で実行した場合の電池の持続時間は 30325 秒(30325 秒)であった。図 2 にそれぞれの実行時の時間経過に伴う電圧変化を示す。動作開始時の電圧は 12MHz と 18MHz 動作時が 9.24V、18MHz 動作時が 9.27V で、動作はどちらも 4V で終了した。12MHz と 24MHz で交互に実行した場合の方が実行開始からわずかに高い電圧を維持し、実行時間が長くなった。

#### 2.1.5 予備実験考察

実験の結果、一定の周波数を保つより、12MHz と 24MHz を交互に繰り返した方が電池の持続時間が長くなった。この原因を調べるために、各周波数で動作させた場合のボード全体の電流の平均値を計測した(表 1)。計測時には電力源として、12V の AC アダプタを使用し、電力計は AC アダプタとボードの間に接続した。12MHz と 24MHz

を交互に動作させた場合(実測値)の電流が 12MHz と 24MHz をそれぞれ単独で動作させた場合の電流の平均(仮想値)よりも低い事が分かった。そこで、周波数切り替えのみを繰り返し行い、その時の電流を計測した。周波数切り替えを繰り返し行った場合の電流は 26.8mA 程度でループ時の電流より低い事が分かった。そのため、交互に動作させた場合は単独で動作させた場合と比べ電流が低かったと考えられる。

次に、12V の AC アダプタを使用した場合の電流値と予備実験での電池持続時間を使い、擬似的に予備実験でどれだけ電力を消費したかを求めた。

・ 12MHz と 24MHz を交互

$$34.06 \times 30325 / 3600 = 286.057 (mAh)$$

・ 18MHz で一定

$$34.24 \times 29995 / 3600 = 285.286 (mAh)$$

以上より、2つの実験で利用できた電池の容量に大きな差はない事がわかった。理由として、電流が小さく電流の差が数 mA 程度しかなかったこと、アルカリ電池は電流と放電容量の関係がリニアに

表 1 周波数と電流の関係

周波数	電流 (mA)
12MHz	32.06
18MHz	34.24
24MHz	37.17
12MHzと24MHz(仮定)	34.62
12MHzと24MHz(実測)	34.06
周波数変更時	26.8

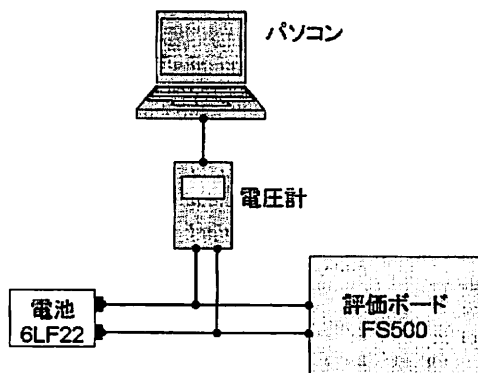


図 1 実験環境

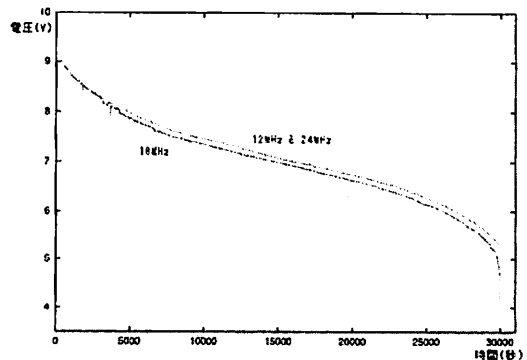


図 2 ループ処理に伴う電圧の変化

近いため、ほとんど効果がなかったと考えられる。

### 3. 周波数制御手法

予備実験の結果から、消費電力の少ない機器では電池の放電特性を考慮した周波数とタスクスケジューリング、つまり、消費電力の大小による電池の消費量の考慮は必要ないと判断した。本研究では、消費される電力のみを考慮した周波数切り替えを行う。

本研究では、オーディオ再生タスクの処理をするのに必要な周波数の決定に、オーディオデータ用のバッファのアクセス情報を用いる。

個別のタスクで必要な周波数から、複数のタスクをスケジュールするにもバッファの空き領域等を考慮する必要があると考えられる。手法については今後設計、実装を行う予定である。

#### 3.1. クロック決定手法

オーディオデータバッファへの書き込み頻度から周波数を低く適切な値に設定することが可能か検証するために以下の方法を FS-500 に実装した。

メモリ領域上にオーディオデータバッファ領域を 8KB 用意し、リングバッファとした。このバッファにオーディオデータを書き込む時に、バッファが満杯になっている場合は、バッファに空きができるまでループ処理によって停止させ、何回ループしたかを数えておく。タイマー割り込みによって一定の時間が経過したら、何回無駄にループしたかを調べ、その値から次の周波数を決定するようにした。周波数の決定のために、ループ条件評価などのループ処理とカウンタのインクリメントに必要なクロックを前もって調べておいた。ループ処理とカウンタのインクリメントに必要な処理は 7 クロックとして計算した。次に設定するクロックは次の式で求めた。

$$(\text{現在のクロック}) - (\text{ループ回数}) \times 7$$

この式で求めた周波数以上の周波数になるように設定する。クロックは入力クロックの通倍で設定するため、変更される周波数は、式で求めた周波数と同じかそれよりも上になる。

図 3 に FS-500 に上で述べた手法を実装した場合のクロックの変化と電流の変化を示す。クロックは初期値として、処理に十分と考えられる周波数、61.44MHz とした。また、カウンタの値から計算し、周波数を変更するための処理は 2 秒に一度タイマ割り込みを発生させて行った。これは電流計が電流値をログに残すのが 1 秒に 1 回であるためである。再生した音声は 1 秒程度の wave 形式のファイルで、前もって ROM 上に記録させておいたものを RAM 上に用意しておいたバッファに転送し、オーディオインタフェースからの割り込み処理によってバッファからオーディオインタフェースに転送される。音声は最後までいくとまたはじめから再生されるようにした。

上で述べた手法により、手動で音が切れない周波数に設定した時と同じ周波数まで落とすことが出来た。ただし、今回の実験では、バッファにデータを供給する処理量が一定である点、周波数を下げるのみの処理で上げる処理がない点で不十分である。今後は、処理量が増減した場合にもバッファが空になる前に割り込みを発生させるなどして周波数をあげる処理が必要になると考えられる。

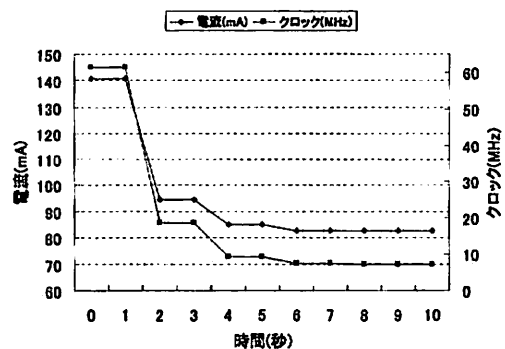


図 3 クロックと電流の変化

## 4. 関連研究

中村ら[2]はフィードバック制御により、低いデッドラインミス率を目標とし、周期的にデッドラインミス率を取得し、その情報を元に周波数を切り替えている。タスクスイッチ毎に周波数制御にかかわる処理を行わず、周期的に行うため、周波数制御に関わる処理がタスクの数に左右されず一定で効率が良い。しかしこの手法ではデッドラインミスが起こるまで周波数が下げられてしまうため、本研究のような音声再生プロセスにおいては向いていないと考えられる。

宮川ら[3]は、汎用 OS においてタスクの性質からインタラクティブなタスクであるか、バッチプロセスであるかを予測し、優先度、周波数の制御を行っている。リアルタイム性を必要とするシステムでの動作は想定していない。

Pillai ら[4]は、DVFS をリアルタイムシステムに適用した手法をいくつか提案し実装している。実験の結果として、20~40%程度の低消費電力化が可能であったとしている。実装はノートパソコン上でされ、linux を用いて行っている。本研究のようなメモリ量のごく限られている環境においても実装可能であるかどうか検証の必要がある。

## 5. まとめと今後の課題

本論文では、実際に利用されるボードにおいて、同じ処理量をこなす場合に、クロックが頻繁に入れ替わる場合と、一定の場合での電池の持続時間について調べ、周波数制御手法について考察した。また、バッファへのアクセスを監視することによってプロセッサの周波数を適切に変更できることを確認した。今後は複数のタスクを動かし、デッドラインを守りつつ周波数を出来る限り下げたためのタスクスケジューラの実装を行い、評価を行う予定である。

## 謝辞

この論文を作成するにあたって、研究用にボードを貸して頂いたファイナーク(株)に感謝する。本研究ではボードの特性を考慮しておらず、プログラムも最適化していないため本来のパフォーマンスを発揮していません。

## 参考文献

- [1] 携帯機器の2次電池の最長利用を目的とするリアルタイムスケジューリングアルゴリズム、中本幸一、辻野嘉宏、都倉信樹、電子情報通信学会論文誌(VOL.J83-D-I)
- [2] リアルタイム性を考慮したフィードバック制御による動的周波数制御手法、中村哲朗、加藤真平、小林秀典、山崎信行、IPSJ-SLDM06124015
- [3] 低消費電力のためのスケジューリングアルゴリズム、宮川大輔、石川裕、IPSJ-OS06102003
- [4] Padmanabhan Pillai, Kang G.Shin Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems, {ACM} Symposium on Operating Systems Principle(2001)