

# ストリーム信号処理用マルチスレッドプロセッサの消費電力低減機構

榎田敏克<sup>†</sup> 樽家昌也<sup>‡</sup> 小川 陸<sup>‡</sup> 大根田拓<sup>‡</sup> 菅野伸一<sup>†</sup>

<sup>†</sup>(株) 東芝 研究開発センター <sup>‡</sup>(株) 東芝 セミコンダクター社

## 1 はじめに

現在、我々はソフトウェア無線機や動画処理等のストリーム信号処理を対象としたマルチスレッドプロセッサの研究を行っている。このような信号処理は、携帯機器でも数多く必要とされるようになってきているが、現状では消費電力等の観点から専用ハードウェアに頼る部分が多くなっている。しかし、様々な規格へ柔軟に対応するにはプロセッサでの実現が必要不可欠である。

一方で携帯機器での利用を考慮すると、消費電力の低減が実用化に向けての大きな課題となる。この課題を解決するため、プロセッサに要求される処理能力をリアルタイムに算出し、動作周波数と電源電圧を最適化することで消費電力の低減を図る機構を検討している。

本報告では我々が提案する消費電力低減機構とシミュレーションによる評価結果について報告する。

## 2 ストリーム信号処理用マルチスレッドプロセッサ

我々が提案するストリーム信号処理用マルチスレッドプロセッサでは様々なサンプリングレートの処理を取り扱うため、一連の処理を複数のスレッドへ分割して実行する。各スレッドの切替は頻繁に発生するため、リアルタイム OS などのソフトウェアによるスレッド切替ではスレッド切替えロスが大きく、処理能力不足が懸念される。この問題を解決するため、ハードウェア制御によるスレッド切替えとスケジューリング機能を備えている。

本アーキテクチャが対象とするストリーム信号処理では、入出力の標準化レートを一定に保持することができれば遅延時間についてはロバストである。そこで図 1 へ示すように各スレッド間のデータ授受に FIFO を利用し、ある程度のデータが蓄積してからスレッドの実行を開始することで、スレッドの切替え自体を削減する。スレッドは、各スレッド間に挿入した FIFO の利用状態をスケジューラが監視し、最も連続してデータを処理できるスレッドをプロセッサが実行する。また、プロセッサはスケジューラの指示を受けると短時間で実行するスレッドを切替える。本アーキテクチャの詳細については文献 [1, 2, 3] を参照されたい。



図 1: FIFO を用いたスレッドの処理フロー

## 3 消費電力低減機構

### 3.1 消費電力の削減

CMOS 半導体のスイッチングに起因する消費電力  $P$  とゲート遅延  $t$  は式 1 と式 2 で表すことができる。式中の  $C$  は CMOS の負荷容量、 $f$  は動作周波数、 $V_{dd}$  は電源電圧、 $V_{th}$  はトランジスタの閾値電圧である。また、 $t$  はキャリアの移動度に依存し、1~2 の値をとる。式 1 と式 2 から動作周波数を高くするためには電源電圧を高くする必要があり、電源電圧と動作周波数が大きくなることで消費電力が上昇することがわかる。つまり、プロセッサの動作周波数を高くすることで消費電力が大幅に増加する。

$$P \propto C \times V_{dd}^2 \times f \quad (1)$$

$$t \propto \frac{V_{dd}}{(V_{dd} - V_{th})} \quad (2)$$

### Dynamically power reducing method for stream signal processing

Toshikatsu Hida<sup>†</sup>, Masaya Tarui<sup>‡</sup>, Riku Ogawa<sup>‡</sup>, Taku Ohneda<sup>‡</sup> and Shin-ichi Kanno<sup>†</sup>

<sup>†</sup>Corporate Research & Development Center, TOSHIBA Corp.

<sup>‡</sup>Semiconductor Company, TOSHIBA Corp.

ストリーム信号処理では入出力レートが固定であるためプロセッサの処理能力が高過ぎても意味がなく、必要以上の処理能力は無駄な電力を消費するだけである。そこでプロセッサの負荷が小さい場合に、動作周波数を入出力レートが保証できる値まで調節することで無駄な電力を削減することができる。

そして、動作周波数にあわせて電源電圧を下げることでさらに消費電力を下げるができる。これは動的電源電圧変更 (Dynamic Voltage Scaling: DVS) と呼び、商用プロセッサ [4, 5] でも採用している。

### 3.2 消費電力の削減

DVS はプロセッサが頻繁にアイドル状態になる低負荷時に、動作周波数と電源電圧を下げることで消費電力の削減を狙う技術である。プロセッサがストールする割合を監視するだけで容易に実現可能であるが、プロセッサがストールせずに処理を実行する場合に、処理能力が足りない状態か最適な状態かが判別できない。このためプロセッサのストール状態を完全に取り除くことが難しく、消費電力削減の限界がある。

一方、我々が提案するマルチスレッドプロセッサではスレッド間のデータ授受に FIFO を使用している。この FIFO の使用状況を監視することで負荷の大きさを検出する。

例えば図 2 へ示すソフトウェア無線機の受信処理の部分を考える。図中のフィルタと復調、誤り訂正、CODEC はスレッドの処理内容を表し、A/D 変換から FIFO A へ一定の標準化レートでデータが入力される。また、処理が完了したデータが FIFO E から一定の標準化レートで出力されるため、この処理に見合うだけの処理能力を持つ必要がある。

ここで、プロセッサの処理能力が負荷の大きさよりも高い場合、A/D 変換からの入力データはすぐに処理が完了してしまい、FIFO E や FIFO D など出力に近い FIFO にしかデータがない状態になる。また逆に処理能力が低い場合には、A/D 変換からの入力データを格納する FIFO A が満杯になりオーバーランを起こすか、FIFO E が空になり出力するデータがないためにアンダーランを起こしてしまう。この場合、FIFO A や FIFO B など入力側の FIFO にしかデータがない状態になる。

そして、処理能力が負荷の大きさに最適な場合では、FIFO 量の大きな偏りがなくなり均等に使用される。つまり、FIFO の使用量を監視することでプロセッサの処理能力が負荷の大きさに対して大きいか小さいかを容易に検出可能である。この情報を基に動作周波数と電源電圧を動的に決定することで、負荷の大きさに適した消費電力でプロセッサを動作させることができる。

この方法では、処理するデータの内容によって必要な処理能力が変動する MPEG などのアプリケーションで特に効果が高い。MPEG のエンコード処理では画像に動きがない場合には処理能力が必要なく、画像に激しい動きがある場合に高い処理能力が必要になる。従来、このようなアプリケーションでは最も処理能力が必要な場合にあわせて動作周波数を決定せざるをえなかったが、FIFO の使用量を監視することで負荷の大きさに相応しい動作周波数を容易に推定できる。その結果、プロセッサの消費電力を負荷の大きさに対して最適化することができる。

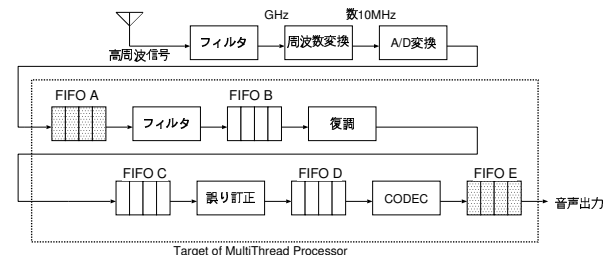


図 2: ソフトウェア無線機の受信処理

### 3.3 全体の構成と機能

我々が提案する消費電力低減機構の構成を図3へ示し、各構成要素の機能と簡単な動作を以下へまとめる。

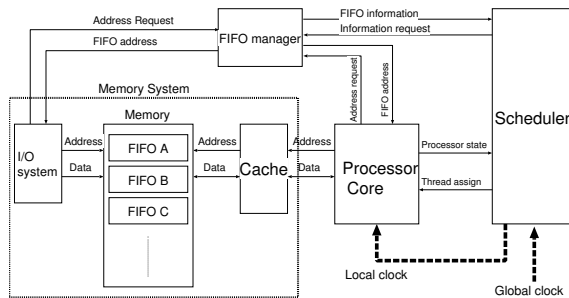


図3: 消費電力低減機構の構成

**Processor Core** スケジューラから割り当てられたスレッドを実行する機能を持つ。FIFOへのアクセスはFIFO managerからFIFOのメモリアドレスを取得し、そのメモリアドレスへ直接アクセスを行う。プロセッサコアはスケジューラが管理するLocal Clockで動作する。

**Scheduler** FIFO情報を基に複数のスレッドの中から実行するスレッドを決定し、プロセッサコアへスレッドを割り当てる。また同様にFIFO情報からプロセッサコアの動作周波数を決定しLocal Clockの制御を行う。

**FIFO manager** メモリ上に実装したFIFOのメモリアドレスとFIFOの容量を管理する。プロセッサコアやスケジューラ、外部I/OからFIFOへのアクセス要求が発生した場合、対応するFIFOのメモリ番地を要求元へ出力する。

**メモリシステム** キャッシュ、メインメモリ、I/O systemから構成される。I/O systemがFIFOへアクセスする場合、FIFO managerからFIFOのメモリアドレスを取得し、そのメモリアドレスへ直接アクセスを行う。

## 4 評価

### 4.1 評価条件

評価には3.3節で示したモデルのシミュレータを作成し、図2のソフトウェア無線機の例を用いた。シミュレーションは負荷の大きさが変動しない場合と変動する場合の2通りを1秒間行い、負荷が変動する場合のシミュレーションでは0.04~0.07秒の間で負荷が3倍になるとした。また、プロセッサのストール状態を監視して動作周波数と電源電圧を調節する方法を比較対象として用いた。

今回作成したシミュレータは、一般的なプロセッサシミュレータのようにアプリケーションのバイナリを解釈して実行する厳密なものではなく、FIFOから特定の大きさの入力データを受け取り、規定のクロック数が経過した後に特定の大きさのデータをFIFOへ出力するといった単純な動作をするものである。各スレッドの動作を表1へ示し、A/D変換からの入力レートと音声データの出力レート、プロセッサの条件は表2へ示す。

3章で述べた負荷の検出方法とその時の動作周波数の制御方法の実装について以下へまとめる。

表1: スレッドの動作

スレッド名	入力 (byte)	出力 (byte)	クロック数
フィルタ	1	1	100
復調	16	1	20
誤り訂正	9	4	100
CODEC	1	1	20

表2: 評価条件

入出力レート	1.44MHz/40KHz(入力/出力)
FIFO容量	1000byte
動作周波数	576MHzのm/10倍
電源電圧	1.5Vのm/10倍

**FIFOの監視** FIFO AとFIFO Eの容量を監視して動作周波数を決定する。FIFO Aの容量が60%以上で要求される処理能力が高いと判断し、40%以下で要求される処理能力が低いと判断する。また、FIFO Eの容量が10%以下である場合には処理能力が低いと判断する。もし要求される処理能力が高いと判断した場合には動作周波数を上昇させ、低いと判断した場合には動作周波数を下げる。

**ストールの監視** プロセッサコアがストールする割合を監視し、ストールが10%以下になるまで動作周波数を1段階ずつ下げる。10%以下になると動作周波数をその状態で維持し、ストールが0%になった場合に動作周波数を上昇させる。

### 4.2 結果

負荷が変動しない場合と変動する場合の動作周波数の変化を図4へ示し、そのときの消費電力の比較を表3へ示す。負荷が変動しない場合において、FIFOを監視した時の平均動作周波数が164MHzであった。動作周波数をこの値に固定して処理を実行した場合、入出力レートを保証しつつプロセッサがストールしない理想的な状態である。そこでこの場合の消費電力を目標値とし、負荷が変動しない場合の消費電力を目標値で正規化した。負荷が変動する場合の消費電力は、最大動作周波数で動作した場合において、ストール時にも動作時と同じ電力を消費すると仮定した値と、ストール時に電力を消費しないと仮定した値で正規化している。

図4から、負荷が変動しない場合と変動する場合の両方でFIFOを監視した時の方が低い動作周波数の変移を見せている。これはFIFOを監視する場合にはプロセッサをストールさせることなく処理を実行できるためである。この結果、表3のようにFIFOを監視した場合の方が消費電力を抑える効果が非常に高い結果となった。

負荷が変動しない場合において、FIFOを監視した時の消費電力は理想的な状態の消費電力よりもわずかに4%増という結果となった。また、負荷が変動する場合においても60%~80%の消費電力を削減することができた。以上よりFIFOを監視して動作周波数と電源電圧を最適化する消費電力低減機構の有効性が確認できた。

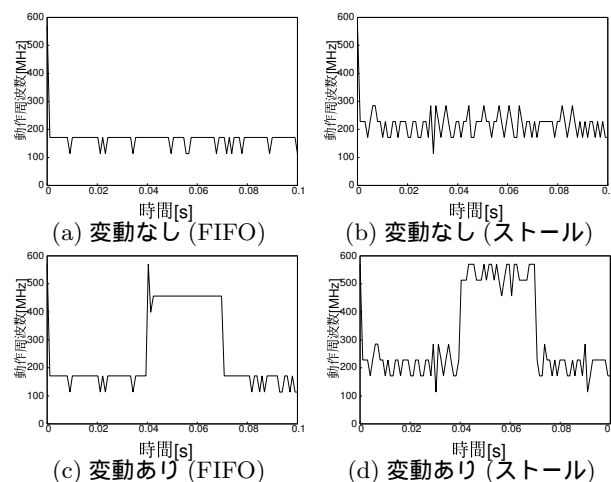


図4: 動作周波数の変化

表3: 消費電力の比較

	FIFOの監視	ストールの監視
負荷の変動なし	1.04	2.41
負荷の変動あり	0.19	0.32
ストール時に電力消費なし	0.42	0.73

## 5 まとめ

携帯機器上でストリーム信号処理を実現する場合に課題となる消費電力を削減するため、スレッド間へ挿入したFIFOの使用状況からリアルタイムにプロセッサの負荷の大きさを算出し、動作周波数と電源電圧を最適化する機構を提案した。

ソフトウェア無線機をモチーフとしたシミュレーションを行った結果、負荷が変動する場合と変動しない場合の両方で大きな消費電力低減効果を確認した。

### 参考文献

- [1] 菅野, 樽家, 大根田, 小川, 宮本: “ストリーム信号処理用マルチスレッドプロセッサのアーキテクチャ”, 電子情報通信学会, CPSY2003-48(2004).
- [2] 小川, 菅野, 樽家, 大根田, 宮本: “リアルタイム処理用マルチスレッドプロセッサのシミュレータの実装と評価”, 電子情報通信学会 2004年総合大会, D-6-3, Mar.2004.
- [3] 大根田, 小川, 樽家, 菅野: “リアルタイム処理用マルチスレッドプロセッサの性能評価”, 電子情報通信学会 2004年総合大会, D-6-4, Mar.2004.
- [4] Kerwell, K.: “Pentium M Hits the Street”, MICROPROCESSOR REPORT, Vol.17, No.3, Mar.2003.
- [5] Transmeta: “Crusoe Processor Product Brief Model TM5800”, 2003.