

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

菅野 伸一 樽家 昌也 大根田 拓 小川 陸 宮本 幸昌

(株) 東芝 研究開発センター

あらまし 本報告では、ソフトウェア無線機等の多様なサンプリングレートの混在するストリーム信号処理を効率良く処理するための、レジスタウィンドウを用いたプロセッサアーキテクチャを提案する。イベント駆動によりスケジューリングを行うが、FIFOを用いてスレッド切り替え回数を削減することにより高い処理性能を実現する。また、シミュレーションによりソフトウェア無線機等のアプリケーションを念頭においた性能推定を行い、スレッド切り替え損失の削減効果の見通しが得られた。

キーワード ソフトウェア無線機, マルチスレッド

The architecture of multi-threaded processor for stream signal processing

KANNO, Shin-ichi TARUI, Masaya OHNEDA, Taku OGAWA, Riku MIYAMOTO, Yukimasa

Corporate Reserch and Development Center, TOSHIBA Corporation

abstract In this paper, we propose the multi-threaded processor architecture for stream signal processing. It schedules based on FIFO's which connect between thread to thread.

We estimate the performance of this architecture in some of typical application. It perform reduce performance loss caused context switch.

Key words Software Defined Radio, Multi-thread

1 はじめに

近年多様な無線通信方式が利用されていて、携帯電話だけでも様々な方式が利用されている。また、通信速度の高速化等の要求のため様々な通信方式が提案されてきている。そのため、同一のハードウェアで様々な方式に対応することを目的に、ソフトウェアで機能が変更できるソフトウェア無線機に対する期待が高まっている。

一方で、携帯端末などの無線機器の信号処理をソフトウェアで実現するためには、プロセッサには高速な信号処理に耐え得る処理能力が必要なことはもちろんであるが、一定のレートで入力させるストリーム状の信号を処理し続けられることを保証しなければならない。また、ソフトウェアの再利用性と保守性を考慮すると様々なハードウェアで実行できる仕組みが必要である。

そこで、本報告ではソフトウェア無線機を実現するための各処理段をスレッドに分割しハードウェアでスレッド切り替えを行うアーキテクチャを提案し、併せてスレッド切り替えに伴うプロセッサキャッシュの動作のシミュレーション結果について述べる。

2 無線機における信号処理

ソフトウェア無線機は、被処理データのサンプリングレートが高い高周波側の処理からサンプリングレートの低い低周波側の処理まで様々な処理が混在しているのが特徴である。

図1に典型的な無線機の例を挙げる。この例では高周波信号を一度中間周波数に変換し、その後直交復調を行う例である。様々な方式を扱う必要性を考慮すると、BPF2とLPF3,LPF4の特性はソフトウェアによ

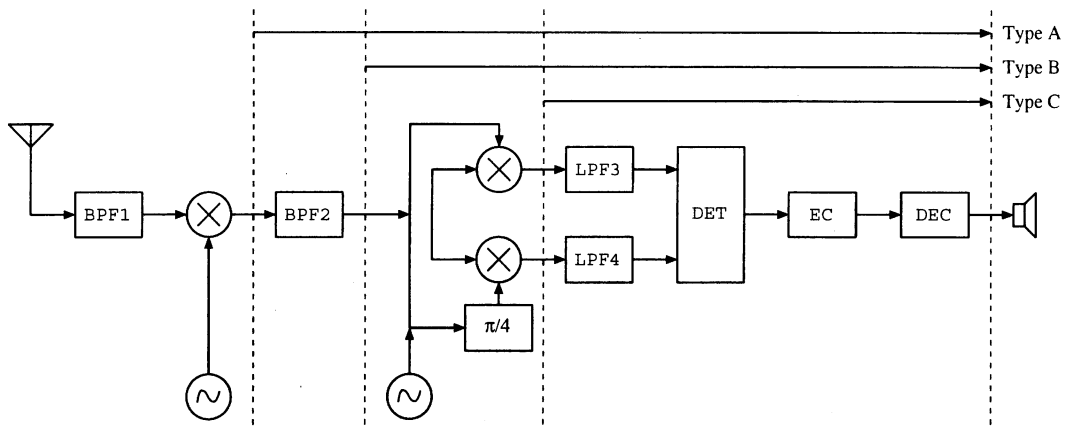


図 1: 無線機の場合

て変更できるのが望ましい。しかし、コストパフォーマンス柔軟性のトレードオフを考慮すると図 1に示すように、ソフトウェアの担当部分を 3 種類に分類することができる。

Type A フィルタリング前の中間周波をソフトウェアで処理するため最も柔軟性が高いが、処理能力を最も必要とするタイプ

Type B 中間周波処理をソフトウェア処理するタイプ

Type C ソフトウェア処理はベースバンドのみのタイプ

柔軟性を高くとるためにはソフトウェア処理を行う部分を増やすのが望ましい。しかし、近接周波数からの妨害対策のため、受信処理の A/D 変換のダイナミックレンジが問題となるため、Type C が現実的と思われる。

ソフトウェア無線機で行う信号処理の性質は以下の通りである。

- a) 一定のレートで入力されるデータを連続して処理する。
- b) 処理遅延時間への要求がそれほど厳しくない。
- c) 高周波に近い段は高レートであるが、低周波段は低レートであり各々異なるサンプリングレートで処理されている。

- d) 高周波に近い段は定型的な処理が多く、例外処理が皆無あるいは非常に少ない。データの内容によって処理が変わる可能性が高いのは、誤り訂正処理やコーデック周辺等である。

したがって、低レートの低周波段あるいはベースバンド処理のスレッドに、高い頻度で高周波段処理のスレッドが割り込む形になることが想定される。a) の条件が満足できれば b) の性質により図 2のように FIFO にある程度のデータを溜めてからまとめて処理を行うことによりスレッド切り替えを回数を減らし、スレッド切り替えの損失を減らすことが可能になる。

3 アーキテクチャ

ソフトウェア無線機における信号処理では、一定の時間の間にデータを処理しなければならないデッドラインが厳密に存在する。そこでデッドラインをベースにスケジューリングを行うことが考えられる [4]。しかし、ソフトウェア無線機の高周波段の処理を想定すると、サンプリングレートが高いことから相当に高速なスレッド切り替えが必要になる。また、定型的な処理が多いため一定期間に一定の処理能力を確保できれば良く、固定的なスケジューリングでスレッド切り替えを行うことでも十分な場合が多い。しかし、デバッグや仕様変更等に伴い一部のソフトウェアの変更を行おうとすると、全体のスケジューリングの調整が必要になる。

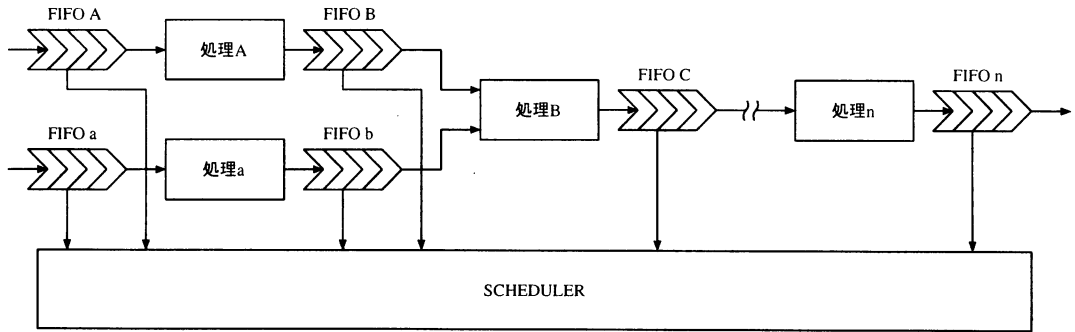


図 2: FIFO を用いたストリーム処理系

ソフトウェア無線機ではプロセッサの処理性能が不足がちなこと、このようなソフトウェアを多様なハードウェアで実行しようとする、個々にスケジューリングの調整を行うことが困難になることが想定される。

ソフトウェア無線機の場合、各処理段でサンプリングレートが異なるためイベント駆動によるスケジューリングを行うのが妥当と考えられる。ただし、非常に高速なレート of データを扱うため、ソフトウェアの介入するスレッド切り替えでは処理能力が不足することが懸念される。そこで、図 3 に示すようなプロセッサに複数のレジスタセットを持たせ、それぞれにスレッドを割り当て、スレッド毎に独立したプログラムカウンタとスタック空間を配置し、ハードウェアによってスレッド切り替えを行う方法を提案する。

この方法では、スレッド切り替えはレジスタウィンドウを切り替えることによって行う。スレッド間通信は、前章で述べたようなソフトウェア無線機では多少の遅延が許されるという性質を利用し、図 2 のように各スレッド間に FIFO を挿入し FIFO の状態を監視し、その状態によってスレッド切り替えを行う。FIFO の段数を増加すれば遅延時間は増加するが、スレッド切り替え回数を減らせるので高速な処理が可能になる。

FIFO はスレッド間通信の伝送量が少量であればハードウェアで直接実装することも可能であるが、多量のデータが必要な場合には汎用メモリーで実装することも必要になる場合がある。また、ソフトウェア無線機の各処理をスレッドに割り当てるとレジスタウィンドウの数がハードウェア実装だけでは不足するので、汎用メモリーの方に退避エリアを設け、仮想的に多量

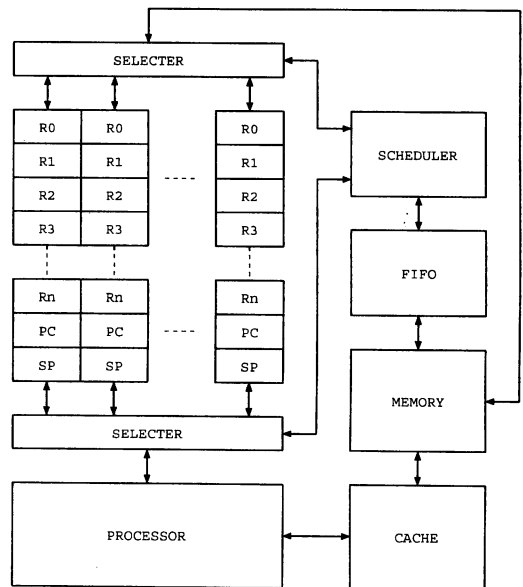


図 3: 提案するアーキテクチャ

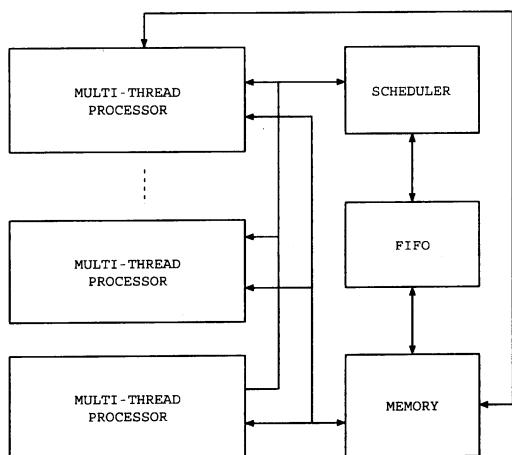


図 4: 複数プロセッサによる構成

のレジスターセットを持たせることができるようにしている。レジスターセットの切り替え時には、他スレッド実行中にアクティブでないレジスターセットとの間で入れ換えを行い、レジスターセット切り替え時間を隠蔽するものとする。

なお、複数のプロセッサにより構成する場合には、図 4 のようにスケジューラーとメモリーを共有する。レジスターセットを仮想化しているので、他のプロセッサで実行していたスレッドを他のプロセッサに移送して実行可能である。そのため、処理能力に余裕のあるプロセッサに処理を移動する操作を行い、負荷のバランスをとることが可能である。

4 スケジューリング

スケジューリング方式としては、イベントドリブン型を原則として以下の 2 通りの検討を行った。

図 5 にスケジューリングの様子を示す。(a) は各スレッドが実行可能になるタイミングと必要な処理量を示している。各スレッドが独立に実行されるならば、並列にこのようなタイミングで実行される。

4.1 固定優先度

FIFO の状態を監視し、データが到着した場合により優先度の高いスレッドを起動する方法である。

図 5(b) は、スレッド A が優先度が高く、以下 B, C の順にスケジューリングされた例である。図に示すようにスレッド切り替えが数多く起きている。一方で図 5(c) はその逆の例であり、こちらはスレッド切り替えが減少している。特に同一スレッドが連続実行されることからプロセッサのインストラクションキャッシュが当たりやすくなる。

4.2 可変優先度

FIFO の状態を監視し、もっともスレッド切り替え間隔が長くなるようなスレッドを選択し切り替える方式である。

スレッド入力側の FIFO に蓄積されたデータの量と、出力側の FIFO の空きデータの量を比較し、一定の閾値を越えたスレッドのみが起動対象になる。スレッド切り替え時には、その中で入力データの数が多いいスレッドに切り替える。

一度起動されたスレッドは入力 FIFO のデータ量が消費され、閾値を下回るまで連続実行する。このスレッドに他のスレッドが割り込むことは、例外処理等を実行する場合以外には原則として行わない。

図 5(d) はそのスケジューリング結果の例である。結果として図 5(c) と同一になっているが、こちらの場合には特にスレッド間で優先度を設けずにスケジューリングを行っている点が異なる。

この方式は、同一のハードウェア上に複数のストリーム処理が存在する場合に、それぞれの処理のスケジュール間の調整が自動的にとれる点がメリットとなる。

5 シミュレーション

本アーキテクチャを利用して以下に述べるような、W-CDMA の受信ベースバンド処理、一般的なストリーム信号処理の例として MP3 デコーダ、MPEG2 のエンコーダ/デコーダといったアプリケーションを使用してスレッド切り替えの様子、切り替えに伴うキャッシュメモリーの振舞いについてシミュレーションを行った [1, 2]。

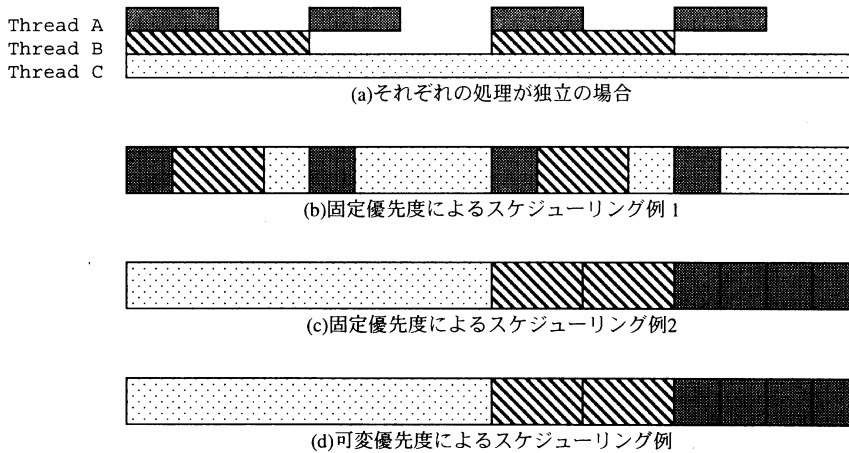


図 5: スケジューリングの様子

表 1: キャッシュの構成

	Cache Size	Line Size	Associativity
I	1kB	32B	1(directmap)
D	1kB	32B	4(LRU)

W-CDMA 受信処理 W-CDMA の受信処理は、同期、逆拡散、畳み込み符号やターボ符号による誤り訂正処理等から構成される。スレッド数が多く、なおかつ単純な処理の積み上げになっている。

MP3 デコーダ 既存の MP3 デコーダソフトウェアを、本アーキテクチャに適合するように改造を行った。

MPEG2 エンコーダ/デコーダ 既存の MPEG2 エンコーダ/デコーダソフトウェアを、本アーキテクチャに適合するように改造を行った。相互参照が多く、多くのスレッドに分割することが困難であったので、スレッド数が少なくなっている。

本シミュレーションでは、FIFO のサイズは十分大きくとるとして行う。シミュレーションに用いるプロセッサは当社製品の MeP[3] にレジスターウィンドウを付加したアーキテクチャで行っている。本シミュレーションに用いたキャッシュの仕様を表 1 に示す。

6 結果

シミュレーション結果を表 2 に示す。キャッシュ管理の方式として、スレッド毎に独立なキャッシュ領域を割り当てる方式と LRU による方式についてシミュレーションを行っているが、LRU 方式の方が好結果を得ていることから [2]、今回はキャッシュは LRU 方式で評価を行っている。

スレッドのサイズは、スレッドのコードサイズと利用するデータのサイズの合計である。W-CDMA はスレッド数が多いが、他のものは基にしたプログラムの構造の制約からスレッド数が少なくなっている。また、画像系のアプリケーションでは扱うデータが大きいことからデータのサイズが大きくなっている。

6.1 スレッドサイズ

レジスターウィンドウによるマルチスレッドのスレッドサイズは最低限レジスターセットのサイズ、それに加えてキャッシュメモリーに蓄積された当該スレッドに関連するデータのサイズとなる。もし、キャッシュの大きさに比べてスレッドサイズが十分小さい場合には、スレッド切り替えのペナルティは小さくなる。一方でスレッドサイズが大きい場合には、スレッド切り替えにともないキャッシュの内容の更新が必要になりそのペナルティは大きくなる。

表 2: シミュレーション結果

サンプル名称	スレッド数	キャッシュ種別	スレッドサイズ [kB]	キャッシュヒット率 [%]		
				固定優先度 1	固定優先度 2	可変優先度
W-CDMA	45	I	17	96.576	96.355	96.640
		D	291	99.074	99.340	99.343
MP3 デコーダ	13	I	14	95.740	95.653	95.736
		D	2	95.795	95.760	95.759
MPEG2 エンコーダ	8	I	375	98.832	98.839	98.851
		D	374	97.943	97.965	97.965
MPEG2 デコーダ	7	I	47	98.511	98.511	98.195
		D	67	98.263	98.263	98.263

6.2 スレッド切り替えのロス

本アーキテクチャでは、FIFOを用いて同一スレッドが連続実行できるようにしたため、スレッド切り替え回数が減少しているためキャッシュヒット率が高くなっている。

特にスレッド数が多く、またサンプリングレートがまちまちな W-CDMA では、可変優先度制御によるスレッド切り替え回数削減の効果が大きく、性能が高い方の固定優先度よりも若干のキャッシュヒット率の向上が確認できている。

しかし、他のアプリケーションの場合には可変優先度による効果は確認できなかった。これは、サンプリングレートが処理の開始から終了まで一定であるため、可変優先度にする効果が無かったためと推測できる。これは、固定優先度の両制御方式でも大きな差がないことから裏付けられる。

7 まとめ

本報告では、プロセッサにてソフトウェア無線機のような処理を実現するためのアーキテクチャ提案を行い、あわせてシミュレーションを行いその結果について述べた。

FIFO の利用量を基にイベント駆動によりスレッド切り替えを行い、スレッド入れ換え回数を減少するようにスケジューリングを行うことにより、ソフトウェア無線機のような高速に多数の処理を同時に行うようなストリーム処理において、キャッシュメモリーを含

めたスレッド切り替えのロスを隠蔽できることが確認できた。

今後は、ソフトウェア無線機の実現に向けてリコンパイルロジックの適用等の検討を行って行く予定である。

参考文献

- [1] 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., 発表予定
- [2] 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., 発表予定
- [3] 東芝: “Media embedded Processor”, <http://www.mepcore.com/>
- [4] G. C. Butazzo, F. Sensini: “Optimal Deadline Assignment for Scheduling Soft Aperiodic Tasks in Hard Real-Time Environments”, IEEE trans. Computers, pp. 1035-1052, Vol. 48, No. 10., Oct. 1999