

## CMP アーキテクチャを導入した マルチメディア処理向け VLIW プロセッサ

都市 雅彦<sup>†</sup> 須賀 敦浩<sup>†</sup> 早川 文彦<sup>†</sup> 多湖 真一郎<sup>†</sup> 今井 賢<sup>†</sup> 田中 篤志<sup>†</sup>

<sup>†</sup>富士通研究所 〒211-8588 神奈川県川崎市中原区上小田中 4-1-4

E-mail: <sup>†</sup>toichi@jp.fujitsu.com, <sup>†</sup>suga.atsumuro@jp.fujitsu.com, <sup>†</sup>hayakawa.fumihi@jp.fujitsu.com,

<sup>†</sup>s-tago@jp.fujitsu.com, <sup>†</sup>imai.satoshi-02@jp.fujitsu.com, <sup>†</sup>tanaka.atsushi@jp.fujitsu.com,

あらまし CMP アーキテクチャ向けに、粗粒度データ並列性に注目した MPEG2 ソフトウェアデコーダを開発した。このデコーダは、実行するプロセッサコア数が増減しても、プロセッサコアへの処理割付けなど一部のコード修正のみでそのまま再利用でき、プロセッサコア数に対しスケラビリティのあるものとなっている。

我々の開発した CMP アーキテクチャを採用した 4 コアプロセッサ FR1000 にソフトウェアデコーダを搭載したところ、1 コア使用時と比較して約 3.2 倍の性能向上が得られ、1 コアで目標性能を実現するよりも消費電力を削減できる結果が得られた。

キーワード 組み込み、マイクロプロセッサ、チップマルチプロセッサ、マルチコアプロセッサ、低消費電力

### A VLIW Single-Chip Multi-Processor for Multimedia processing

Masahiko TOICHI<sup>†</sup> Atsumuro SUGA<sup>†</sup> Fumihiko HAYAKAWA<sup>†</sup>

Shinichiro TAGO<sup>†</sup> Satoshi IMAI<sup>†</sup> and Atsumuro TANAKA<sup>†</sup>

<sup>†</sup>Fujitsu Laboratories 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa 211-8588 Japan

E-mail: <sup>†</sup>toichi@jp.fujitsu.com, <sup>†</sup>suga.atsumuro@jp.fujitsu.com, <sup>†</sup>hayakawa.fumihi@jp.fujitsu.com, <sup>†</sup>s-tago@jp.fujitsu.com, <sup>†</sup>imai.satoshi-02@jp.fujitsu.com, <sup>†</sup>tanaka.atsushi@jp.fujitsu.com

**Abstract** We have developed a scalable MPEG2 software decoder for CMP architecture, it is optimized for coarse grain level data parallelism. The decoder has scalability for processor numbers, because it can work on processors of various numbers without redesign.

We implemented the decoder on a single-chip multi-core processor FR1000 that integrates four 8-way VLIW FR-V processor cores. As a result, the decoder with 4 processors shows 3.2 times higher performance than that with 1 processor, and reduces power consumption less than 1 processor.

**Keyword** embedded, microprocessor, chip multi-processor, multi-core processor, low-power.

#### 1. はじめに

プロセッサの性能向上に伴いソフトウェアで実現可能な機能範囲は広がっており、従来では専用ロジックを必要とした動画の再生処理もソフトウェアで行えるようになってきている。速度性能や消費電力などの面で最先端の性能を必要とする製品分野では ASSP などの専用ロジックを用いたハードウェアが適しているが、ミドルからローエンドの製品分野ではソフトウェアで機能実装を行うことで機能の変更や追加が容易になり幅広い展開が可能となる。

このような流れの中でより高い性能をもつ製品分野へのプロセッサの利用が検討されており、これを実現するプロセッサへの高性能化の要求が高まっている。しかし、従来のプロセッサの性能向上は周波数増加に頼ったものであり、消費電力の問題などから限界が指摘されている。

これに対する解として複数のプロセッサコアを一つのチップに搭載した CMP 構成をとるプロセッサが開発されている<sup>[4][5]</sup>。我々も CMP アーキテクチャ採用したプロセッサとして VLIW コアを 4 つ搭載した

FR1000を開発した<sup>[6]</sup>。CMPでは、処理の量にあわせてコア数を増やすことで、容易に性能の向上が得られる。また、CMPではコア数を増やすことにより、一つ一つのコアに求められる性能が低減され、これにより周波数を下げることができる。周波数を下げると動作電圧も下げることができ、より消費電力を抑えることが可能である。このように速度性能、消費電力などの製品要求にたいし、CMPアーキテクチャはコア数の増減などで対応可能である。

しかしCMP向けソフトウェアでは、プロセッサの性能を引き出すために、コア数に応じてプログラムを分割する必要がある、開発が困難といわれている。

デジタル機器の高度化・多機能化により開発期間とコストの増大が問題を解決するが手段として、開発プラットフォームを定め、ソフトウェアの機能を階層化して定義し、インタフェース(API)を定めることにより、ソフトウェアの再利用する方法が提案されてきた。

[1][2][3]

しかし、異なるコア数のプロセッサへのプログラム分割においては、このような階層化とインタフェースの定義だけではプログラムの再利用は難しい。このため、CMP向けソフトウェアとして、再設計をすることなくプロセッサコアの数に応じてスケラブルに拡張可能なソフトウェアを開発する手法がもとめられている。

本稿では、コア数の変化に対しスケラブルに拡張可能なソフトウェアの事例として、CMP向けMPEG2ソフトウェアデコーダの構成とFR1000への実装について報告する。

## 2. CMP向けMPEG2デコーダの構成

CMP向けソフトウェアは複数のプロセッサコアに処理を割り当てるために、処理を分割することになる。本章では、分割数に対してスケラブルに拡張可能なCMP向けソフトウェア構成を述べるとともに、MPEG2デコード処理での処理分割方法について述べる

### 2.1. スケラブルに拡張可能なソフトウェア構成

ソフトウェアを分割する際に分割の方法は、処理に注目した分割方法とデータに注目した分割方法に分類することができる。処理に注目したソフトウェアの分割では、それぞれのプロセッサコアで別々の処理関数やタスクが並列に動作することになる。一方、データに注目した分割では、それぞれのプロセッサコアで同じ処理が行われ、依存関係の無いデータに対して同じ処理が行われる。このため分割したそれぞれの処理で用いられるデータは並列処理可能なように独立したデータになっている必要がある。

処理に注目した分割の場合、ある程度以上数のタス

クへ分割するためにはソフトウェアの再設計を必要とし、同じソフトウェア構成のまま多くのプロセッサコアを持つCMPへの適用は困難となる。一方データに注目した分割では、それぞれのプロセッサコアに対し同じ処理を割り当てることができる。データの並列性が十分あれば、ソフトウェアの再設計が必要なくコア数の異なるCMPへ拡張することが可能となる。

設計コストを削減するためには、一度開発したソフトウェアが再設計をせず、さまざまなコア数へ対応できると望ましい。このため、分割のスケラビリティを考慮した場合、プログラムの分割はデータ並列に注目して行うことが望ましいと考えられる。

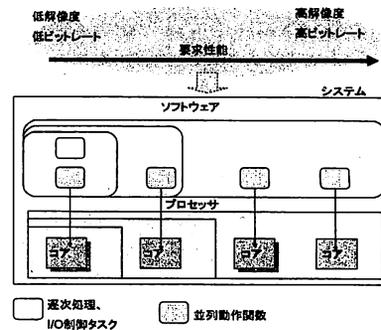


図 1: CMP向けソフトウェアアーキテクチャ

### 2.2. MPEG2デコード処理の粗粒度並列性

MPEG2のデコード処理の特徴から具体的な分割方法を考える。

MPEG2デコード処理では、処理に注目した分割として、VLD, IDCT, MCなどの各マクロブロックに対して行われる処理をパイプライン並列処理することが可能である。しかし前節で述べたように、分割数ごとにプログラムの書き直しが必要となるため、分割のスケラビリティの観点から有効ではない。そのため、この方法は処理の分割のとしては採用しないこととした。

動画処理はその特性から、データに注目した並列性として、ピクチャ間の並列性と画素間の並列性を見出すことができる。MPEG2において、GOPレイヤ、ピクチャーレイヤでの並列性がピクチャ間の並列性に対応し、スライスレイヤ、マクロブロックレイヤでの並列性が画素間の並列性に対応する。

この2つの並列性について考察する。

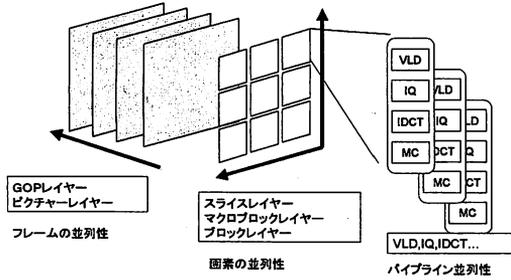


図 2 : MPEG2 復号処理の並列性

(1) ピクチャ間の並列性を活かした並列処理

MPEG などの動画圧縮方式はピクチャ間のデータ依存性を利用して圧縮率を高めるアルゴリズムになっている。ただし MPEG2 では、B ピクチャは他の画像から参照されることが無いため、B ピクチャ間でデータ依存が無い。そのためピクチャ間の並列性を生かした並列処理が可能である。また、複数の画像からなる GOP 間にもデータの参照関係が無いため、プロセッサコアにデコード処理を GOP 単位で割り当てる並列処理が可能である。

しかしながら、ピクチャごとの処理量は均等ではなく負荷バランスをとりにくい。フレームごとの実行サイクルを調査した結果を図 3 のグラフに示した。最大で 2 倍以上の実行サイクルの差が発生している。フレームごとの実行サイクルのばらつきはストリームに大きく依存するものであり、事前の予測は困難である。

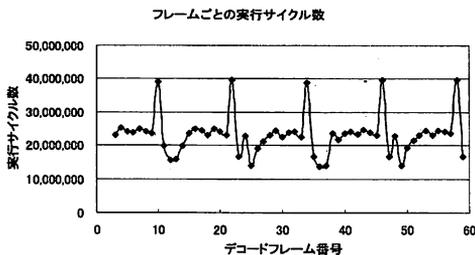


図 3 : フレーム毎の実行サイクル数

(2) ピクチャ間の並列性を活かし並列処理

MPEG2 では同一フレーム内のスライスの間でデータ依存が無いため、スライスレイヤでの並列化が

可能である。最大で 1 フレームに含まれるスライス数と同じ数の並列化が可能で、より処理能力を必要とする高解像度の画像ほど高い並列性を引き出すことが可能である。

処理負荷のばらつきなどを考慮して、今回の実装では、MPEG2 デコード処理ソフトウェアをスライス層で分割することとした。分割の様子を図 5 に示す。



図 4 : MPEG2 デコード処理の並列性

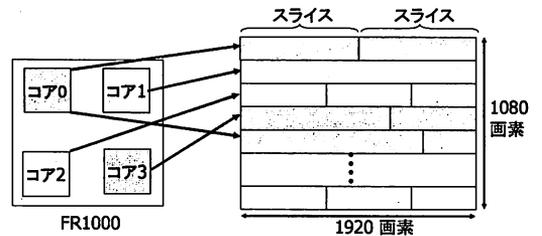


図 5 : スライス層分割

2.3. 動作率向上のための負荷割り当て

CMP 向け並列処理プログラムにおいて性能向上率は次のように表現することができる。

$$\text{性能向上率} = \frac{\text{使用コア数} \times \text{平均動作率}}{\text{並列化による総処理サイクルの増加比}}$$

平均動作率とはソフトウェア動作時間のうち、各プロセッサコアが動作している時間の平均値である。逐次処理が必要な箇所や処理負荷が均等でなく、プロセッサコアでの処理がアイドル状態で停止することが多いと値は低くなる。0~1 の値をとり、負荷が均等でア

アイドル状態がない理想的な状態では平均動作率は1となる。

並列化による総演算数の増加比はそれぞれのコアの処理サイクルの合計値が、シングルコアでの処理サイクルと比較してどれだけ増加しているかを表す。1以上の値を取り、ソフトウェアの並列化によるオーバーヘッドが一切無い理想的な状態では並列化による総処理サイクルの増加比は1である。

性能向上率を高めるためには、平均動作率を高め、並列化による総処理サイクルの増加を抑えることが必要となる。

MPEG2 デコード処理において、動きの激しい部分などでは処理負荷が重くなるため、静的に各コアで処理を行うスライス数を均等に割り当てるだけでは各コアの処理量を均等にならない。各コアに対する処理量を均等化するため、各コアに割り当てるスライス数を動的に変化させ、負荷バランスを均等化することとした。

### 3. 実装環境

MPEG2 ソフトウェアデコーダを FR1000 へ実装対象である FR1000 のアーキテクチャおよび並列プログラム開発環境の紹介を行う。

#### 1) FR1000 アーキテクチャ

FR1000 は VLIW 方式のプロセッサコアを 4 つ内蔵したチップマルチプロセッサである。それぞれのコアは、1 つの命令で 8 演算を同時処理可能な SIMD 方式の命令を持ち、最大 8 命令まで同時実行可能である。プロセッサ間転送用の専用バスを持ち、各コアで動作するコア間の通信を高速に行うことができる。

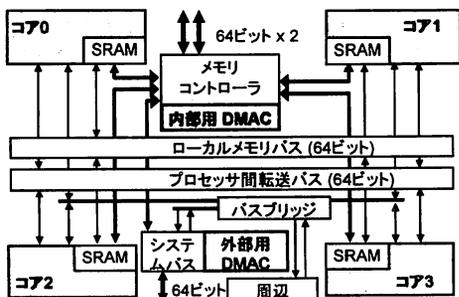


図 6 : FR1000 の構成図

#### 2) 並列プログラム開発環境

プロセッサ間サービスコールを規定し  $\mu$  I T R O N 仕様 OS をマルチプロセッサ用に拡張した<sup>[7]</sup>。

プロセッサ内のタスク間通信は従来の  $\mu$  I T R O N のサービスコールを用い、プロセッサ間のタスク間通信はプロセッサ間のサービスコールを用いて通信を行う。

4 コアでの MPEG2 デコード処理のタスク構成は図 7 のようになる。

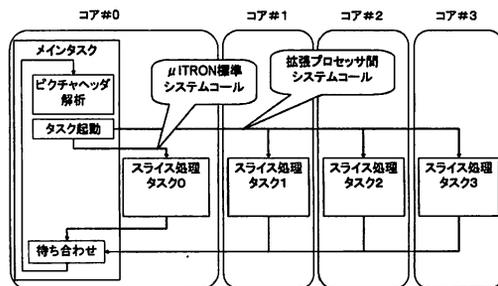


図 7 : MPEG2 復号処理タスク構成

タスク起動および待ち合わせ部の簡単な修正で、1 ~ 3 コアでのプログラムも容易に作成可能である。さらに、将来的に 4 コアを超えるコア数を持つプロセッサを開発した場合も容易に対応できる。

### 4. 性能評価

#### 4.1. 速度性能の評価

2 章の方針に基づいて実装したプログラムの評価を行った。デコードするストリームは MPEG2 MP@HL で画像サイズは 1920×1088、ビットレートは 22Mbps である。測定の結果、FR1000 では、MPEG2 MP@HL をデコードするのに必要な周波数は約 390MHz であった。1 つのコアで MPEG2 MP@HL をデコードするのに、約 1200MHz 以上の周波数が必要であり、4 つのコアで並列処理を行うことにより、1 コアに比べて約 3.2 倍の性能向上が得られた。

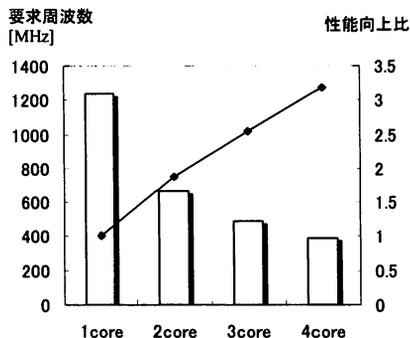


図 8 : MPEG2 MP@HL 性能評価結果

図 9 のグラフはあるフレームでのプロセッサコアごとのスライス処理時間を示している。静的なスケジューリングでは、最も処理負荷が重いコアではデコード処理に約 28ms の時間がかかっている。一方処理が軽いコアでは約 19ms で処理が終わっており、多くの時間 IDLE 状態となっている。

動的なスケジューリングを行うことにより負荷バランスが均等化され、23~25ms の範囲で処理が完了している。それぞれのプロセッサコアの平均動作率は約 91% であった。

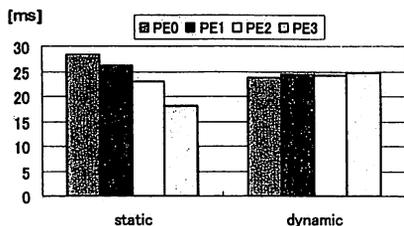


図 9 : 動的スケジューリングの効果

#### 4.2. 消費電力の評価

図 10 はそれぞれの周波数で、電圧値を正常動作する最低の電圧に設定し、電力を測定したものである。シングルコアで周波数を上げる場合は、周波数ごとに電圧が上がり、周波数一定(250MHz)でコア数を増やしていく場合では電圧は一定になっている。

同一性能で比較した場合、周波数をあげた性能向上と比較して、コア数を増やした性能向上は約半分の消費電力で実現できる。

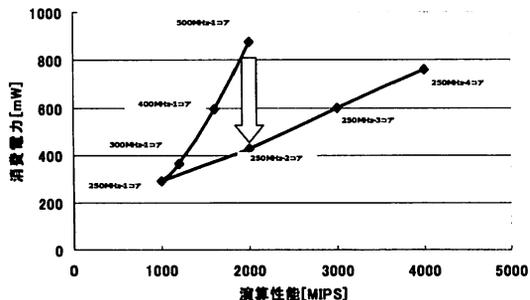


図 10 周波数とコア数を変えた場合の消費電力

以上の結果より、マルチコアはコア数を増やすことによりスケラブルに性能向上が可能であり、周波数をあげる場合に比べて低消費電力であることがわかる。

コア数や周波数は、速度性能・消費電力などの製品要求から求められる。CMP アーキテクチャでは、スケラブルなソフトウェアを構築することにより、コア数の増減に対して同じプログラムを使用することができ、設計コストを抑えて様々なシステムを構築可能となる。

#### 5. まとめ

スケラブルな拡張性を考慮した CMP 向け MPEG 2 ソフトウェアデコーダを開発し、VLIW プロセッサコアを 4 つ内蔵した 1 チップマルチコアプロセッサ FR1000 へ実装した。

プログラムはタスクの起動待ち合わせ処理部の修正を行うことで使用コア数の変更が可能であり、わずかな修正で 1~4 コアに対応可能であることを確認した、さらにコア数が増えた場合にも同様のコード変更量で対応可能と考えられる。また、性能はコア数が増えるごとに比例して向上しており、1 コア使用時と比較して 4 コア使用時には約 3.2 倍の性能向上が得られた。

シングルコアで周波数を変えた場合と周波数一定でコア数を変えた場合の電力測定を行い、同一性能で比較するとコア数を増やした性能向上が約半分の消費電力で実現可能であった。マルチコアは処理を分散し、一つ一つのコア数の周波数を下げるにより低電力化が可能であることがわかる。

CMP アーキテクチャは様々な性能要求に対し、コア数に応じた性能向上を低い電力で実現可能である。スケラブルな拡張を考慮したソフトウェアはコア数の変更に対しても、同じソフトウェアを使用することが

可能であり、開発コストを抑えてさまざまなシステムを提供可能である。

## 文 献

- [1] <http://www.semicon.panasonic.co.jp/micom/MicomFamily/uniphier.html>, "UniPhier トップページ", 松下電器
- [2] <http://www.necel.com/platformovia/ja/index.html>, "platformOVIA", NEC エレクトロニクス
- [3] [http://japan.renesas.com/fmwk.jsp?cnt=press\\_release\\_20051025.htm&fp=/company\\_info/news\\_and\\_events/press\\_releases](http://japan.renesas.com/fmwk.jsp?cnt=press_release_20051025.htm&fp=/company_info/news_and_events/press_releases), "システム開発の統合ソリューションを提供する「EXREAL Platform」", 株式会社ルネサス テクノロジ
- [4] M. Edahiro, S. Matsushita, M. Yamashita, N. Nishi, "A Single-Chip Multiprocessor for SmartTerminals," IEEE MICRO, pp.12–20, July, 2000.
- [5] D.Pham et al., "The Design and Implementation of a First-Generation CELL Processor," ISSCC Dig. Tech. Papers, pp.184–185, Feb 2005
- [6] T. Shiota et al., "A 51.2GOPS 1.0GB/s-DMA single-chip multi-processor integrating quadruple 8-way VLIW processors," ISSCC Dig. Tech. Papers, pp.18–19, Feb 2005.
- [7] 鈴木貴久, 上方輝彦, "マルチプロセッサ向け  $\mu$ ITRON OS の開発", 第 157 回計算機アーキテクチャ研究発表会, Dec., 2005