

メディアプロセッサのデータ転送方式

6K-3

細木 浩二, 東嶋 重樹, 鍛田 真*, 野尻 徹, 西岡 清和

(株) 日立製作所 システム開発研究所

(株) 日立製作所 デバイス開発センタ*

1 はじめに

動画処理などメディア処理の実時間実行という要求に対し、メディアプロセッサが注目を集めている。メディアプロセッサはC言語などの高級言語による開発が可能のため、新規機能の迅速な対応が可能と共に、画像処理や音声処理などの異なった機能を一つのプロセッサで実現出来る。しかし、数Mバイトにわたる領域のデータに対し、順次処理を進めるメディア処理では、局所参照性に優れたキャッシュメモリを有効に使用することは困難である。この問題を、メディア演算処理とデータ転送の並列実行を実現するアーキテクチャにより解決した。

この論文では、そのアーキテクチャと、デジタルビデオ向けのHD(1080i)フォーマット(MPEG-2フォーマットによる1920*1080画素、30フレーム毎秒のインタレース・モード)[1]のダウンデコードによる性能評価について述べる。

2 メディアプロセッサのアーキテクチャ

図1に我々が開発したメディアプロセッサのブロック図を示す。演算器は、VLIW[2]アーキテクチャ構造で、メディア演算命令を含む4命令を同時発行可能なVLIW CPUコアと、ハフマン符号及び復号を行う可変長符号コプロセッサからなる。VLIW CPU コアは、4命令のうち、2命令をメモリアクセスとして指定可能で、データキャッシュをアクセスする。データキャッシュは4バンクインタリーブ構造のライトバック方式を採り、4並列のキャッシュアクセスが可能である。更に、演算器と独立して、内部バスに接続されたモジュールやIO間のDMA転送を行うデータ転送エンジンを設けた。

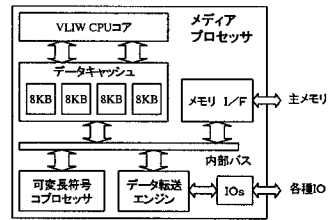


図1 メディアプロセッサのブロック図

3 データ転送方式

このデータ転送エンジンは、DMAによるメモリ・メモリ転送を行うDMAエンジンで、3種のコヒーレント[3]モード(表1)をもって、主メモリやIO間のデータ転送を実現すると共に、データキャッシュへのアクセスを可能とする。

表1 コヒーレント・モード

コヒーレント・モード	動作
Coherently Allocate	データキャッシュに対するキャッシュナリード・ライト。キャッシュミス時、データキャッシュ上にそのブロックをアロケートする。
Coherently No-Allocate	通常のスヌープと同様に振る舞い、データキャッシュヒットの場合、データキャッシュに対しリード・ライトを行う(ライト時は主メモリも更新)。データキャッシュミスの場合、主メモリに対してリード・ライトを実行し、データキャッシュのアロケートは行わない。
Non-Coherently	スヌープ処理を伴わないデータ転送

本方式により、Coherently Allocateモードによるデータキャッシュへのライトアクセスは、データキャッシュへのプリフェッチとして動作し、Coherently No-Allocateモードによるリードアクセスは、データキャッシュからのデータ吐出しを実現し、データの整合性を維持する。

一般に1画面あたり数Mバイトの画像データは、2次元配列のデータ形式をとり、マクロブロック等の単位で処理を実行する場合、その規則的な配置により、特に小容量で連想度の低いキャッシュメモリでは確実にブロックの衝突が発生し、スラッシングが激増する。この問題を考慮し、本データ転送エンジンではプログラマブルなアドレス生成器を設けた。これは、2次元配列データの1次元配列化、若しくは1次元配列から2次元配列へのアドレス変換(図2)を行うものである。まず、ソフトウェアに

A data transfer implementation for a media processor
Koji Hosogi, Shigeki Higashijima, Makoto Kuwata*,
Toru Nojiri, Kiyokazu Nishioka
System Development Laboratory, Hitachi, Ltd.
Device Development Center, Hitachi, Ltd.*

よりデータキャッシュ上の小容量な連続領域をバッファとして設定し、データ転送エンジンは、主メモリから2次元配列のデータを読み出し、連続した1次元データにマッピング後、データキャッシュ上に作成したバッファにコピーすることで、スラッシングのないプリフェッチを実現する。同様に、データキャッシュ内に格納された連続データ構造をとる画像データのコピーバックを、2次元配列の画像イメージに再マッピングすることも可能である。

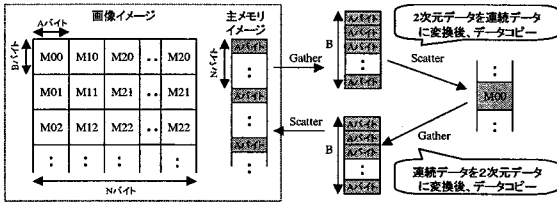


図2 データ転送エンジンのアドレス生成

4 ソフトウェア・モデル

次に、本アーキテクチャに適した、VLIW CPU コアのソフトウェア・モデルについて述べる。

これは、メディア処理に必要な数 M バイトといった大容量データを複数に分割し、データ転送エンジンは、ソフトウェアにてメモリアロケーション管理したデータキャッシュ内の小容量領域に、繰返しプリフェッチとコピーバックを行い、更にデータキャッシュをダブルバッファとして使用するソフトウェア・モデルである (図3)。

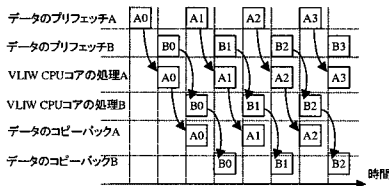


図3 ソフトウェア・パイプライン

VLIW CPU コアとデータ転送エンジンは、このデータキャッシュ上に作成された小容量領域を繰返しアクセスすることにより確実にキャッシュヒットする。ダブルバッファの使用とは、この小容量領域を2セット持ち、データ転送とVLIW CPU コアのアクセスを、異なったセット同士にて排他的に使用することにより、これらを並列実行し、データ転送のオーバヘッドを隠す方式である。

5 性能評価

本アーキテクチャを採用したメディアプロセッサの性能評価を行った。評価に使用したプログラムは、HD(1080i)フォーマットのビットストリーム画像を、縦横共に2分の1にダウンサンプリングし、従来のアナログテレビへ出力するダウンデコーダで、データ転送エンジンにて、ハフマン復号化、逆離散コサイン変換、動き補償、画像生成などに伴うデータ転送を行うモデルである。

この結果、640M サイクル毎秒要した実時間デコードのプログラムが、データ転送エンジンの使用により、263M サイクル毎秒で実現できた(図4) [4]。この時、データキャッシュミスは、初期設定時に65回発生し、他の全てのアクセスはキャッシュヒットする為、キャッシュヒット率100%を実現した。

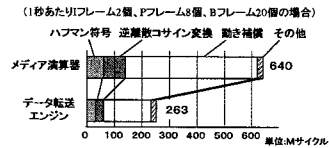


図4 HD(1080i)のダウンデコードに必要なサイクル数

6 まとめ

メディア処理のデータ転送を考慮したメディアプロセッサを開発し、評価用に用意した全てのHD(1080i)フォーマットの画像を、周波数270MHzにて実時間デコードできた。

最後に、本開発にてご支援頂いた、川口敦生氏、田代卓氏、廣井和重氏に深く感謝いたします。

参考文献

[1]: ISO/IEC: "Information technology - Generic coding of moving pictures and associated audio information", ISO/IEC 13818, 1996
 [2]: Mike Johnson: "Superscalar Microprocessor Design", P T R Prentice Hall, Inc. 1991
 [3]: Daniel E. Lenoski, Wolf-Dietrich Weber: "Scalable Shared-Memory Multiprocessing", Morgan Kaufmann Publishers, Inc., 1995
 [4]: 廣井他:"VLIW型メディアプロセッサを用いたMPEG-2ビデオデコーダ", 映像情報メディア学会技術報告, Vol.25, No.21 PP.25~30", 2001