

メディア処理用組み込み OS

藤井 茂樹 中谷 信太郎 松本 正治 平井 誠 清原 督三

松下電器産業(株) マルチメディア開発センター

メディア処理に特化した構成を取りつつも、プログラムを変更することにより様々なメディア処理を実現できるメディアプロセッサの開発が行われている。メディアプロセッサで複数の処理を実行するためには OS が必要であるが、性能、メモリサイズ等の点から汎用 OS を用いることはできない。本報告ではメディア処理に特化したメディアプロセッサ用組み込み OS について述べる。また、Audio/Video デコード処理のリアルタイム動作を実現する Media Core ProcessorTM にメディア処理用 OS を実装し、Audio/Video デコードの特徴を活かしたスレッド切り替え、スケジューリングを行うことによりリアルタイム性能を維持できることを示す。

Embedded Operating System for Media Processing

Shigeki Fujii Shintaro Nakatani Masaharu Matsumoto
Makoto Hirai Tokuzo Kiyohara

Multimedia Development Center, Matsushita Electric Industrial Co.,Ltd.
1006 Kadoma, Kadoma-shi, Osaka 571 JAPAN

Media processor which provides the flexibility for various multimedia applications is currently being developed. Although an Operating System is necessary even for media processor, common Operating System is not suitable because of the lack of real time performance and the requirement of large memory size. In this paper, we present Embedded Operating System for media processing. This Operating System is implemented on Media Core ProcessorTM which performs full-frame Audio/Video decoding. Performance estimation shows that this implementation keeps real time performance by utilizing Audio/Video decoding characteristics in thread switching and scheduling.

1 はじめに

マルチメディア処理における Audio/Video デコードなどの処理は、以前から専用 LSI が使われてきた。Audio あるいは Video のデコードを複数の専用 LSI を用いて処理を行い、ハードウェアロジックで Audio あるいは Video のある特定の規格のアルゴリズムを実現した。しかし各処理に対して専用のハードウェアを組むため、規格の変更や別のメディアへの展開は困難である。

一方、高性能な汎用 CPU においてもマルチメディア処理専用の命令を追加実装し [1]、ここで Audio/Video デコードなどの処理が行われている。各処理をプログラムで記述するため、規格の変更や別のメディアへの展開は容易である。しかし、現在高性能な CPU においてもフルフレームデコードの性能要求を満たすことが出来ない。また、この

セットを民生機器で応用しようとした場合に、高性能な CPU の他に Video や Audio の入出力処理を行う付加的なハードウェアが必要となり、複数チップで構成されハードウェアの量を増大させる。

そこで最近では、メディア処理に特化した構成を取りつつも、プログラムを変更することにより様々なメディア処理を実現できるメディアプロセッサ [2][3] の開発が行われている。今までの汎用 CPU の互換性ととられることなくメディア処理に適したアーキテクチャをとれるため、コンパクトで性能の良いプロセッサとなる。このプロセッサで複数の処理を実行するためには、OS が必要であるが、性能、メモリサイズ等の点から汎用 OS を用いることはできない。

本報告では、Audio/Video のリアルタイム同時デコードを実現するメディアプロセッサ用組み込み

OSについて述べる。まず、第2章では、メディア処理用組み込みOSの特徴について、第3章では、Audio/Videoデコード処理の特徴について、第4章では、メディアプロセッサの一種であるMedia Core ProcessorTM(MCP)[4]について、第5章では、MCP組み込みOSについて、第6章においてその評価について述べる。

2 組み込みOSの特徴

メディア処理用組み込みOSは、各種機器に組み込まれてその制御を行うコンピュータシステム(機器組み込み制御システム)[5]で用いられるOSと同じように、汎用OSとは違う特徴があり、さらに以下のような特徴がある。

2.1 リアルタイム

メディア処理では、Videoデコード処理やグラフィクス処理などによる画像出力や、Audioデコード処理による音声出力を行う。画像出力であれば、垂直同期信号のたびに画像を表示させなければ画像が不自然になる。また音声出力であれば、サンプリング周波数に合わせて音を出力しなければノイズとなって聞こえてしまう。したがって、OSは画像、音声を同時に処理する場合にリアルタイム性を損なわないスケジューリングを行う。また、コンテキストスイッチの時間や割り込み応答時間を短くする。

2.2 OSの効率

メディアプロセッサはメディア処理をリアルタイムで実行できる性能を持っている。よってOSの実装においてもOSの汎用性よりもメディア処理の性能に重点を置く。例えば処理するメディアの数に応じてOS内のリソースの量を静的に決定する。逆に処理するメディアの特性を活かすようにOSを設計して、OSの処理(アイドルスレッドの時間も含む)を最小にする。

2.3 OSのサイズ

メディアプロセッサの多くはプログラムを内蔵する命令メモリをチップ内に全て持ち、そのサイズは数十Kbyteほどである。したがって、OS以外のメインの処理プログラムを考慮するとOSに与えられるサイズは、高々数Kbyteである。した

がって、メディア処理に必要な機能のみでOSを構成する。

3 Audio/Videoデコード処理

MPEG2ビデオ、Dolby DigitalTM(AC3)オーディオおよびサブピクチャ(圧縮された2次元グラフィックス)を含む符号化されたDVD規格のビットストリームをデコードする場合の処理の流れを図1に示す。入力されたビットストリームに対してシステムレベルのストリーム解析を行い、ビデオ、オーディオ、およびサブピクチャの3種類の符号化されたストリームに分離する。各々のストリームに対して、独立にデコードし(図1の3つのデコードの流れ参照)、出力画像と音声を得る。ビデオ出力処理部では、出力画像に対してフォーマット変換(4:2:0→4:2:2)とサイズ変換を行い、さらに、サブピクチャおよびOSD(On Screen Display)データとのブレンド処理を行う。また、Audio/Video同期やトリックプレイ(早送り再生、スロー再生など)を実現するために、上記各処理の全体制御を行う。

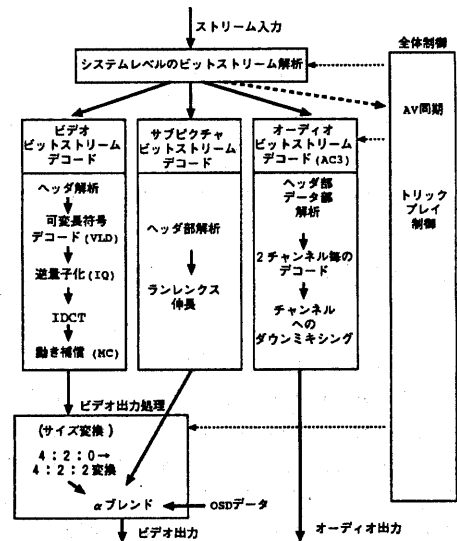


図1: DVDのAudio/Videoデコード処理の流れ

また演算性能に大きく影響するエレメンタリレベルのAudio/Videoデコード処理の詳細は以下のようなになる。

- Videoデコード

1. ヘッド解析: ビデオストリームはピクチャサイズや量子化係数などの画面情報を含んでいる。
2. ハフマンデコード: ビデオストリームにおける可変長符号を固定長のデータに変換する。
3. 逆量子化処理 (IQ): デコードされた数値をオリジナルの数範囲に変換する。
4. 逆離散コサイン変換 (IDCT) 演算: IDCT 演算は周波数領域から空間領域のデータに変換する。MPEG 演算においては 8×8 の行列演算となる。
5. 動き補償処理: Step4 で得られた差分データに参照画像を足し込むことによって復元画像を得る処理である。

- Audio デコード (Dolby Digital)

1. ストリームパージング: オーディオストリームにおけるヘッド解析を行う。
2. ビットアロケーション: 復元する際に割り付けるデータのビット数を決定するためにスペクトラム包絡線の解析を行う。
3. IMDCT & Windowing: IMDCT は周波数領域のデータを時間領域に変換する。
4. ダウンミキシング: デコードされたチャネルデータ (例えば5チャンネル) を2チャンネルにミキシングを行う。

上に挙げた処理のアルゴリズムにはそれぞれに多くの特徴が存在する。ビットストリームの処理は演算量は少なく処理は逐次化されてしまう。一方、IQ、IDCT のような行列演算では並列性を取り出すことは容易である。システム全体の観点からいけば、並列処理に使用する演算性能向上の方法は処理が逐次化される部分に関しては有効には働かない。さらに並行実行しているリアルタイム処理、例えば Audio/Video のシステムレベルの処理や I/O 処理には頻繁なコンテキストスイッチが必要となり単一プロセッサ構成の場合には、並列処理部分の性能低下の原因となる。

4 Media Core Processor

4.1 システムアーキテクチャ

Media Core ProcessorTM(MCP) は民生用マルチメディアに対するトータルソリューションを提供するチップである。民生用ではチップセットのコストや低消費電力を含めて低コストでの実現

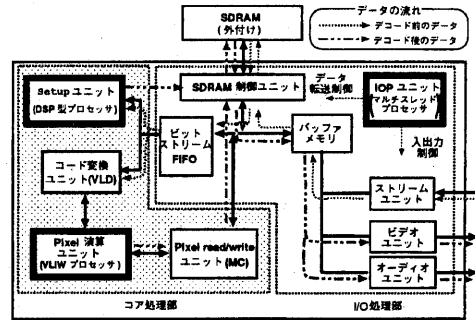


図 2: MCP のハードウェア構成

が重要な要因となる。また開発期間の短縮も高い優先度を持つ。

これらの要求に対して MCP では、逐次処理、並列処理、非同期イベント処理の3種類の性質が異なる処理に対して、その処理形態やアルゴリズムに適したプロセッサを用意し、その非均質なマルチプロセッサ構成において、プロセッサの稼働率を向上させるための最適な処理量の配分を行っている。この方法により動作周波数の増加を抑えながらマルチメディア処理に対して柔軟かつ効率的な対応を実現している。

図2に MCP のハードウェア構成を示す。MCP は、コア処理部と I/O 処理部で構成されている。コア処理部は Audio/Video のビットストリームデコードを主に行い、I/O 処理部は非同期イベント処理を含む間欠的な処理を主に扱う。これより、多種の非同期イベント発生によりコンテキストスイッチを必要とする処理の大部分は、I/O 処理部で処理されるため、コア処理部は、Audio/Video のデコード処理に専念することができる。MCP では、Audio/Video のデコード結果や入力したビットストリームを格納するために、外部にクロック同期型の DRAM (SDRAM) を使用する。

4.2 コア処理部

コア処理部は DSP 型プロセッサ (Setup ユニット)、可変長符号の切り出しを行う専用回路 (コード変換ユニット:VLD)、IDCT 演算を行う VLIW 型プロセッサ (Pixel 演算ユニット)、および MC の処理を行う Pixel read/write ユニットの4つのユニットで構成されている。

コア処理部では Audio/Video のエレメンタリレベルのデコードを行う。このデコード処理のなか

で、演算量が最も多いのはビデオの IDCT、MC 演算である。ただし演算に必要とされる精度は 16bit 程度と低い。また、MC では、参照画像の読みだしや生成した画像を書き戻すための広いメモリバンド幅 (60MB/s) も必要である。そこで IQ、IDCT 演算をピクセル単位で並列実行できる Pixel 演算ユニットで行い、MC はメモリアクセスの高速化のため専用ハードウェアを用いる。

ストリーム解析などの逐次処理は、演算量が少なく、かつ並列実行できないので、Setup ユニットにおいて実行する。ストリーム解析では、bit 判定命令の出現頻度が高いため、条件付命令をサポートすることにより分岐ペナルティを削減した。

オーディオデコードの IMDCT は並列処理であるが、ビデオに比べて演算量が少なく、かつ 24bit 以上の演算精度が必要とされるため Setup ユニットにより行う。さらに、ビデオの IDCT 演算を実行中に、Setup ユニットでオーディオデコードの処理を行えるように、VLD を行う専用ハードウェア (コード変換ユニット) を設ける。

また、Setup ユニットでは、スレッド管理、IO 処理部との通信、割込み処理を行う OS が動作する。

4.3 I/O 処理部

I/O 処理部は、入出力処理を制御する I/O Processor (IOP) ユニット、外部とのデータの入出力を行う周辺ユニット (ストリームユニット、ビデオユニット、オーディオユニット、SDRAM 制御ユニット)、および SDRAM と各ユニットとの間でデータ転送を行う場合に一時記憶用のバッファとして使用する内部メモリ (バッファメモリ、ビットストリーム FIFO) により構成される。IOP ユニットは、各処理タスクをオーバーヘッドなしに時分割多重実行できるマルチスレッド型のプロセッサであり、各処理タスクは、一定時間ごとに起動され、割り当てられた時間分の処理を実行するため、実時間動作を保証できる。これにより、ストリーム入力処理、データバッファ間の転送処理、Audio/Video 出力処理、OSD (On Screen Display) 処理など間欠的な非同期イベントを高効率で実行できる。

4.4 デコード基本動作

前述したように MCP で Audio/Video のデコード処理を行う場合の処理分担は以下ようになる。

- コア処理部 Setup ユニット Video デコードのヘッダー解析と Audio のデコード処理と OS を実行する。
- コア処理部 Pixel 演算ユニット Video デコードの IQ、IDCT 演算をピクセル単位で並列実行する。
- コア処理部 VLD 専用ハードウェアで Video デコードの可変長符号の切り出しを行う。
- コア処理部 MC 専用ハードウェアで Video デコードの参照画像の読みだしや生成した画像を書き戻す処理をする。
- I/O 処理部 マルチスレッド型のプロセッサで非同期イベントを処理を含む Audio/Video の入出力処理を時分割多重実行する。

Video デコード処理はマクロブロック (NTSC であれば 1 フレームで 45×30 個のマクロブロック) という単位で行われる。図 3 にビデオの 1 マクロブロック分のデコードを行った場合のコア処理部のパイプライン動作を示す。Setup ユニットは、最初にコード変換ユニット (VLD) との協調動作によりストリーム解析を行う。コード変換ユニット (VLD)、Pixel 演算ユニット (IDCT)、Pixel read/write (MC) ユニットの 3 ユニットは、Setup ユニットでのストリーム解析が完了した後に、同期を取りながらブロック単位 (Y0、Y1、Y2、Y3、Cb、Cr) でパイプライン動作する。

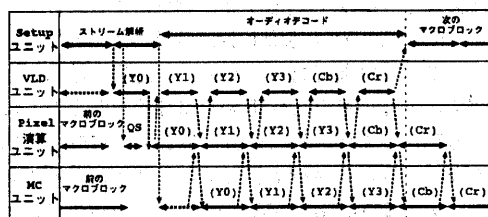


図 3: コア処理部のパイプライン動作

5 MCP 組み込み OS

Audio/Video デコード処理を MCP で動作させたときの組み込み OS について述べる。OS はコア処理部の Setup ユニットで動作し、スレッド管理、イベント管理、I/O 処理部との通信管理を行う。なお、メディア処理では性能を優先させるため仮想記憶の必要はなく、さらにメモリ割り当てを静的に行う。よって OS でメモリ管理は行って

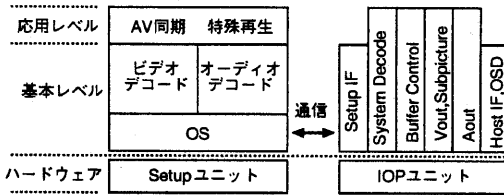


図 4: 全体ソフトウェア構成

いない。図 4 に全体ソフトウェア構成での OS の位置付けを示す。

5.1 スレッド管理

メディア処理の各処理を独立して記述できるようにするためにスレッドを設ける。スレッドはコンテキストと状態と NextPC を持つ。状態は、running, ready, wait の 3 つの状態がある。wait 状態は、各メディア処理の特性を活かすようなメディア固有の wait 要因を持つ。NextPC は、スレッドが wait 状態のときに次に実行を開始する PC (Program Counter) を保持する。Audio/Video のデコード処理の場合、スレッドを Audio デコード処理 (Audio スレッド)、Video デコード処理 (Video スレッド)、idle 状態 (idle スレッド)、OS メイン (OS スレッド) にそれぞれ割り当てる。idle スレッドは、実行すべき処理が無いときに実行する。

スレッドの切り替えは、Video デコードでは垂直同期ごとにデコード結果の画像を出力しなければならないという制約があるためイベントドリブン方式を用いる。これは Video の 1 フレームのデコード終了後に、Video スレッドを垂直同期待ちを要因とする wait 状態にし、Audio スレッドを起動する。その後、垂直同期時にイベントを発生させて wait 要因を解除し、Video スレッドを起動する。さらにデコード基本動作で述べたように、Video デコード処理はコア処理部でパイプライン動作を行う。Pixel ユニット等が動作しているときは、Video スレッドは実行されていないので、Audio スレッドを実行することができる。そこで、Video デコードのマクロブロックのヘッダー解析の終了後に、Video スレッドをマクロブロックのデコード終了待ちを要因とする wait 状態にし、Audio スレッドを起動する。その後、Pixel read/write ユニット (MC) の処理が終了と終了イベントを発生させて wait 要因を解除し、Video スレッドを起動する。このようにすれば、Setup ユニットで処理効率を

上げることができる。

各スレッドの wait 要因は、Video スレッドの場合は、

- マクロブロックのデコード終了待ち
- ビットストリーム入力待ち
- 垂直同期信号待ち
- IO 処理部との通信終了待ち

Audio スレッドの場合は、

- ビットストリーム入力待ち
- IO 処理部との通信終了待ち

となる。各 wait 要因が 1 つ以上あれば wait 状態になる。

5.2 イベント管理

イベントが発生すると直ちに OS スレッドへと実行が遷移する。ただし、OS スレッドの実行中であれば現実行が終わった後に、再び OS スレッドを実行する。なお OS スレッドの実行時間は有限時間内に終了することが保証されている。

イベントの発生により wait 状態であるスレッドの wait 要因が解除される。wait 要因が全てなくなれば、そのスレッドは ready 状態になる。この時 ready 状態のスレッドの中で、最も優先順位の高いスレッドを run 状態にし実行する。優先順位の低いスレッドは suspend され、ready 状態になる。このように固定優先順位によるスケジューリングを行うため、優先順位の再計算といった処理は必要ない。

Audio/Video デコード処理では、OS スレッド、Video スレッド、Audio スレッド、idle スレッドの順に優先順位が高い。Audio スレッドが Video スレッドより優先順位が低い理由は、Video デコード処理で Pixel 処理ユニットと Pixel read/write ユニットが動作している間は、Video スレッドが必ず wait 状態であるので、Audio スレッドが ready 状態であれば必ず実行されるからである。この優先順位の設定により、マクロブロック処理の終了イベントが発生すると Video スレッドの wait 要因が解除されて Video スレッドが ready 状態になり、優先順位の低い Audio スレッドが run 状態であればこれを suspend して ready 状態にし、Video スレッドを run 状態にする。

5.3 通信管理

IO 処理部に対して SDRAM との転送要求、出力要求、外部レジスタへの読み書き等を行うために、通信キューを設ける。通信を行う場合、必ず最初に通信キューの使用権の獲得を試み、獲得できればメッセージを通信キューに置く。メッセージはヘッダーとデータ(有限長)からなり、送信メッセージと受信メッセージがある。送信の場合はデータの書かれたメッセージを通信キューに置いた後、次の処理を行う。受信の場合は通信キューにメッセージを置いた後に、通信終了待ちを要因とする wait 状態になり、終了イベントを受けた後にデータを読み出す。通信キューの使用権を獲得できない場合も、通信終了待ちを要因とする wait 状態になり、終了イベントが受けた後に再び使用権の獲得を試みる。

5.4 IO 管理

入力ビットストリームやデータの出力は IO 処理部で実行され、本 OS からはドライバー層として見える。ビットストリームを読み込む場合に、入力データが無い場合は入力データ待ちのイベントが発生する。

6 評価

MCP に Audio/Video デコード処理プログラムと本 OS を ROM として内蔵する 1chipAV デコーダ (54MHz 動作) を開発した。MPEG2 の AV デコードに加え、OSD(On Screen Display) に使用する 2次元グラフィックス機能などを有する。また、NTSC/PAL エンコーダ、DVD 用デスクランブラ、Video 用 DA コンバータなどを内蔵し、SDRAM を外付けすることにより、1chip で DVD の再生機能を実現する。Setup ユニットの命令 ROM は全部で 80Kbyte あり、そのうち OS は 4.5Kbyte のみである。また、Audio/Video デコードを行いロジックアナライザを用いて各処理サイクルを測定した。入力ストリームを DVD 規格の最大ビットレートに近い 9.5Mbps とし、MPEG2, Dolby Digital(AC-3) をデコード処理する。図 5 に Setup ユニットにおける演算量と OS 処理の演算量を示す。

Audio/Video デコード処理において、スレッド切り替えをマクロブロック単位に行ったため、切り替え回数は増え、マクロブロック処理における

スレッド切り替え時間は、全体処理の内では 3.9% となった。しかし、この時間はマクロブロックの切り替えによって得られる Audio デコードの処理時間の 28.9% に比べ十分小さい。さらに idle スレッドの時間は 1.2% のみであり、Setup ユニットの効率は良い。また、OS 処理時間は全体の 6.5% であるが Audio/Video デコード処理以外に 4% のマージンをとることができ、リアルタイム性能を維持している。

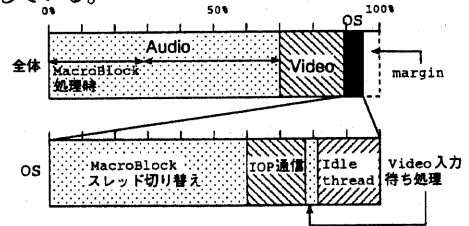


図 5: Setup における演算量

7 おわりに

本報告ではメディア処理に特化したメディアプロセッサの組み込み OS について述べた。また、Audio/Video デコード処理のリアルタイム動作を実現する Media Core ProcessorTM にメディア処理用 OS を実装し、Audio/Video デコードの特徴を活かしたスレッド切り替え、スケジューリングを行うことによりリアルタイム性能を維持できることを示した。

参考文献

- [1] Alex P. et al. "MMX Technology extension to the INTEL architecture," IEEE MICRO, Vol.17, No.2, pp.42-50 (1996).
- [2] 吉田豊彦 "VLIW 型メディアプロセッサ" 情報処理, Vol.38, No.6, pp.499-506 (1997)
- [3] Paul K. "Hardware-Software Interactions on Mpac," IEEE MICRO, Vol.16, No.4, pp.20-26 (1997).
- [4] 落合利之, 吉岡 康介, 木村 浩三 他 "1chipAV デコードを実現する非均質マルチプロセッサアーキテクチャ" 情報処理学会研究報告, 97-ARC-127, pp.871-878 (1997)
- [5] 中本幸一, 高田広章, 田丸喜一郎 "組込みシステム技術の現状と動向" 情報処理, Vol.38, No.10, pp.871-878 (1997)