計算機アーキテクチャ 131－5
（1998. 9. 21）

動画像復号化と3次元グラフィクスで共用可能なメディアプロセッサ向き演算モジュールの設計

藤嶋 秀幸[†,††] 竹本 裕介[†] 米田 友和[†] 尾上 孝雄[†] 白川 功[†]

[†]大阪大学大学院 工学研究科 情報システム工学専攻
〒565-0871 大阪府吹田市山田丘 2-1
Phone: 06-879-7808, Fax: 06-875-5902
E-mail: {*fujisima, takemoto, tomokazu,*
onoe, sirakawa}@ise.eng.osaka-u.ac.jp

[††]九州松下電器株式会社 技術本部 開発研究所
〒814-0001 福岡市早良区百道浜 2 丁目 4 番 16 号
Phone: 092-852-1628, Fax: 092-852-1686

本稿ではMPEG-4で規定されている自然画像（動画像）復号化と人工画像（3次元コンピュータグラフィクス生成画像）を処理するためのメディアプロセッサの中で使用する共用可能なハードウエア資源について考察する．すなわち，IDCTと幾何変換で共用可能な演算モジュールおよび動き補償とテクスチャマッピングで共用可能な演算モジュールを0.35μm 4層CMOSテクノロジにより，それぞれ約44万および約11万トランジスタを用いて実装し，両演算モジュールとも20MHzで動作可能とする．

MPEG-4，メディアプロセッサ，IDCT，幾何変換，テクスチャマッピング，動き補償

Function Modules for Video Decoding and Computer Graphics

Hideyuki FUJISHIMA[†,††], Yusuke TAKEMOTO[†], Tomokazu YONEDA[†],
Takao ONOYE[†], and Isao SHIRAKAWA[†]

[†]Dept. Information Systems Eng.,
Osaka University
2-1 Yamada-Oka, Suita, Osaka, 565-0871 Japan
Phone: +81-6-879-7807, Fax: +81-6-875-5902
E-mail: {*fujisima, takemoto, tomokazu,*
onoe, sirakawa}@ise.eng.osaka-u.ac.jp

[††]Research & Development Laboratory
Kyushu Matsushita Electric Co., Ltd.
4-16-2 Momochihama Sawara-Ku, Fukuoka, 814-0001 Japan
Phone: +81-92-852-1628, Fax: +81-92-852-1686

The present paper describes two hybrid VLSI modules dedicated to MPEG-4 natural/synthetic video decoding, which are to be incorporated into an MPEG-4 based media-processor. One is Matrix-Vector Multiplier, which can perform the matrix-vector multiplication both in the inverse discrete transform and in the geometrical transformation of three dimensional computer graphics. Another is Image Mapping Module, which can be used commonly for the motion compensation of natural video objects and for the texture mapping of synthetic video objects. The implementation result shows that these modules have been integrated with the use of 440k and 110k transistors, respectively. Both modules can operate at 20MHz or less.

MPEG-4, media-processor, inverse discrete cosine transform, geometrical transformation, texture mapping, motion compensation

1 はじめに

ISO の MPEG (Moving Picture Expert Group) では，次世代マルチメディア符号化標準として MPEG-4 の標準化を行なっている [1]．従来の MPEG の画像符号化方式と比較して，MPEG-4 が大きく異なる点は，画像をオブジェクト単位に分割し，それぞれに適した符号化方式が適用できることであり，さらにそのオブジェクトとして自然画像（ピクセルベース画像）と人工画像（3 次元グラフィクス生成画像）を定義していることである．この異なる 2 種類の画像オブジェクトの処理過程の概要を Fig. 1 に示す．

この一連の処理は演算量が膨大なため，汎用プロセッサだけで処理を行なう場合，各命令のフェッチやレジスタへのアクセスが頻繁になり，動作周波数を高くする必要があり，その結果，プロセッサの電力消費が増大してしまう．

一方，専用ハードウエアにより処理を行なう場合には，表示する自然画像と人工画像の数の比率が一定でないため，自然画像/人工画像処理に対してそれぞれ専用のハードウェアを用意するとその比率によってはハードウエア資源を有効に活用することができないという問題が生じる．

本文では，これらの問題を解決するために，新しいメディアプロセッサのアーキテクチャを提案する．このメディアプロセッサは演算量の多い自然画像の復号化と人工画像の生成処理に特化した複数の専用モジュールとプロセッサコアによる構成をとっており，自然画像と人工画像のどちらが出現しても柔軟に対応できるように，2 種類の処理を実行する専用モジュールを備える．このようなアーキテクチャの採用によって，メディアプロセッサ全体の面積の増加を防ぐとともに，動作周波数の低減による電力削減を図る．

具体的には，まず，自然/人工画像処理の中で演算量が大きい逆離散コサイン変換（IDCT）と幾何変換が，共に 4×4 行列とベクトルの乗算で実行できることに着目して，この 2 つの処理機構の共有化を図る [2, 3]．ついで，自然画像復号化における動き補償 (Motion Compensation: MC) と人工画像処理におけるテクスチャマッピング (Texture Mapping:TM) が，共に膨大なメモリアクセスを伴うことに着目して，この 2 つの処理過程で共用可能な専用モジュールを考案し，その VLSI 化設計を行う [4]．

さらに，VDEC によるチップ試作サービスを利用して，上記の行列ベクトル乗算器の試作を行なっているので，併せて報告する．

2 メディアプロセッサのアーキテクチャ概要

Fig. 1で示す 2 種類の処理を実行するために，Fig. 2に示す構成を考える．ここでは汎用プロセッサとして ARM プロセッサを採用し，そのコプロセッサバスを利用して各演算器の制御を行なう．各画像データの読み込み用に 2 チャネルの DMA コントローラを置く．演算モジュールとしては可変長復号化モジュール（VLD），逆量子化モジュール（IQ），変換モジュール（TRANS），画像マッピングモジュール（IM），陰影モジュール（SH）および画像配置モジュール（C）からなる．

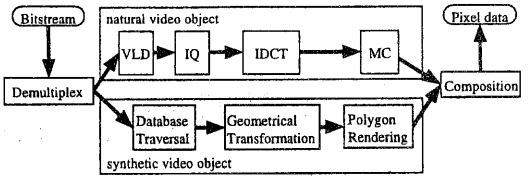


Fig. 1　動画像と合成画像の処理フロー

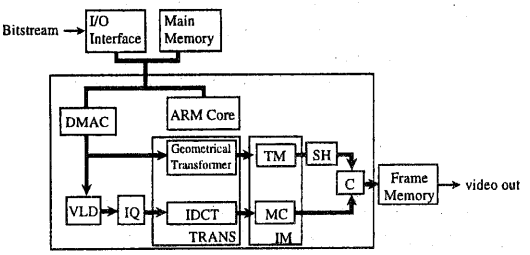


Fig. 2　メディアプロセッサのアーキテクチャ

まず，ビットストリームは汎用プロセッサコアにより逆多重化され，抽出されたデータは DMA コントローラにより各演算パスにそれぞれ入力される．ここで，データとは自然画像データと人工画像データであり，先述のフローの処理をそれぞれ施すことにより表示されるものである．

自然画像の場合には，VLD により可変長符号が復号化され，その出力は IQ へと直接入力される．IQ の中ではデータの並べ換えや AC/DC 予測処理 [1] が施されてから逆量子化の処理が行なわれる．TRANS ではこのようにして求められた DCT 係数に対して IDCT が行なわれる．IDCT を施された動画像のデータは前画像の情報を加えるため，その値に対して動き補償（MC）を行なう必要がある．この動き補償は IM で行なわれる．IM により動き補償を行なわれた画像データはメモリ上に展開され，画像配置モジュールにより画像メモリ上に配置される．

一方，3 次元コンピュータグラフィクスの場合には，汎用プロセッサコアによってメモリ上のデータベースに格納されたデータのトラバーサルが行なわれ，DMA により TRANS に対してデータ入力が行なわれる．ここでは TRANS は 3 次元コンピュータグラフィクスのための浮動小数点数を使った行列ベクトル乗算を行なう．これにより，

3次元空間上に定義された物体の位置関係を定義する幾何変換を行なう．次の工程でこの幾何変換を施された物体に対して表面の色を決定する．3次元CGの場合には，テクスチャマッピングおよびシェーディングを施した後，隠面消去を行ないながら画像配置モジュールが画像をメモリ上に展開する．

このように異なる処理にハードウエアを共有化することにより効率良く資源を活用するとともに，できるだけ画像の構成に必要なハードウエアを専用化し，バスのトラフィックを減少させることができる．この処理過程で比較的多くの計算量を必要とし共有化が可能な，行列ベクトル乗算器と画像マッピングモジュールの設計について以下に述べる．

3 行列ベクトル乗算器（MVM）

この節ではIDCTと幾何変換を実行するためのMVMについて述べる．IDCTは式(1)および(2)，幾何変換は式(3)によって計算することができる．すなわちIDCTと幾何変換は4×4行列とベクトルの乗算で表すことができ，MVMはこの4×4行列と入力ベクトルとの乗算を行う専用ハードウエアである．

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} A & B & A & C \\ A & C & -A & -B \\ A & -C & -A & B \\ A & -B & A & -C \end{bmatrix}\begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix}\begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{bmatrix}, \quad (1)$$

$$\begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} A & B & A & C \\ A & C & -A & -B \\ A & -C & -A & B \\ A & -B & A & -C \end{bmatrix}\begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix}\begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{bmatrix}, \quad (2)$$

$$A = \cos\frac{\pi}{4},\quad B = \cos\frac{\pi}{8},\quad C = \sin\frac{\pi}{8},$$
$$D = \cos\frac{\pi}{16},\quad E = \cos\frac{3\pi}{16},\quad F = \sin\frac{3\pi}{16},$$
$$G = \sin\frac{\pi}{16},$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (3)$$

3.1 MVMのアーキテクチャ

Fig. 3にMVMのブロック図を示す．IDCTは16ビットのDCT係数と12ビットの行列パラメータの乗算であり，幾何変換は32ビット浮動小数点数の頂点座標値と行列パラメータの乗算である．これらの乗算をMVMで実行できるよう，入力レジスタを32ビットで，行列パラメータレジスタを24ビットで構成する．IDCT及び幾何変換は以下のように実行する．

IDCT: 入力ベクトルレジスタを32ビットで構成するため，2つのDCT係数を入力することができる．また，行列パラメータレジスタを24ビットで構成するため，同様に2つの行列パラメータを入力することができる．部分積生成部でDCT係数と行列パラメータに対して，ブースアルゴリズムを用いて部分積を生成し，部分積加算器で部分積を加算する．アキュムレータで4個の積を順次加算し，まるめ処理部で加算結果のまるめを行い，加減算器で式(1)，(2)に従って加減算を行う．

以上の処理を通じて8×1-1次元IDCTが完了する．

幾何変換: 入力ベクトルレジスタには頂点座標を入力し，行列パラメータレジスタには行列パラメータを入力する．これらの積を求める際に8ビットの指数と24ビットの仮数に分けて行う．積の指数は指数加算器で加算して求め，仮数は部分積生成部で生成した部分積を部分積加算器で加算して求めて，1つの乗算が完了する．指数減算器とシフタで浮動小数点数の加算の際に必要な桁合わせを行う．アキュムレータで桁合わせをした積を順次加算し，積和演算結果を正規化処理部で正規化し，まるめ処理部で仮数をまるめる．以上の処理をベクトルの各成分に施すことにより幾何変換が完了する．

3.2 MVMの要求処理能力

本MVMは，21サイクルのレイテンシと16サイクルのスループットで行列ベクトル乗算が実行できるが，実際には極めて膨大なデータを処理するため，レイテンシは無視することができ，MVMの性能はスループットによって評価できる．

いま，MPEG-4の画像オブジェクトを処理するため，自然画像オブジェクトに対して，30フレーム/秒で，1枚のCIF画像と4枚のQCIF画像を生成すること，つまり1フレームあたり792 MB (Macro Block) のIDCTを行うことが要請されるものとし，人工画像オブジェクトに対しては，200万頂点/秒で幾何変換することが要請されるものとする．MVMは，1 MBのIDCTを1,536サイクルで実行することができるので，自然画像オブジェクトに対して必要なMVMのサイクル数は，

$$30 \times 792 \times 1,536 = 36,495,360 \leq 36.5M,$$

となる．一方，MVMは1頂点の幾何変換を16サイクルで行うことができるので，人工画像オブジェクトに必要なサイクル数は，

$$2,000,000 \times 16 = 32,000,000 \leq 32.0M,$$

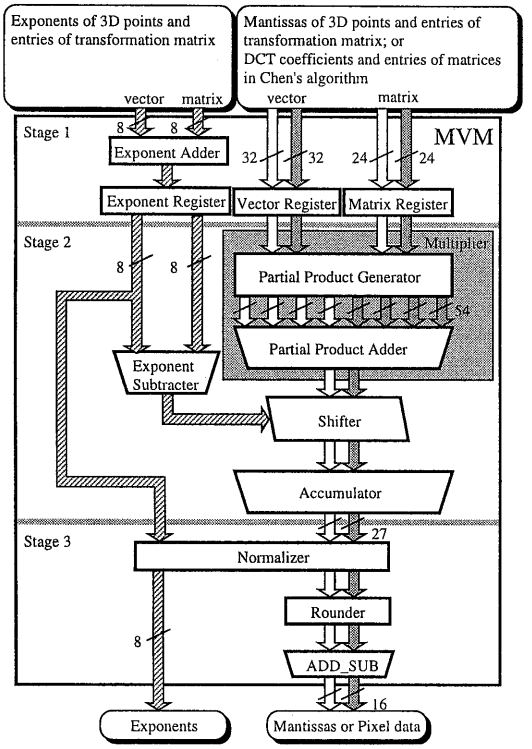


Fig. 3　MVM のブロック図

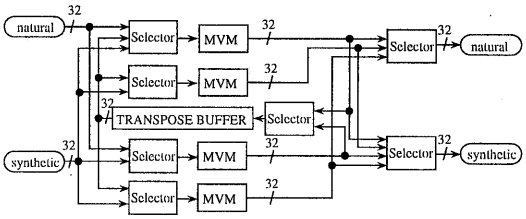
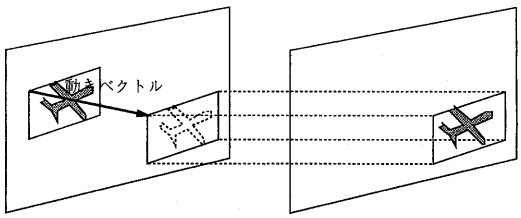


Fig. 4　MVM を用いた変換の構成



参照画像　　現在再生する画像

Fig. 5　動き補償.

となる．したがって MVM が自然画像オブジェクトと人工画像オブジェクトを同時に行うために必要なクロック数は約 68.5M サイクル必要となる．

上記の性能を達成するため，複数の MVM を用いて低い周波数で動作させる方法を採用する．MVM を 17.1*MHz* で動作させることができれば，4 個の MVM を用いることにより目標性能を達成することができる．

3.3　変換モジュール（TRANS）の構成

Fig. 4に示すように，上記の MVM を接続する．このような接続により表示すべき内容に応じて MVM の割当を変更しハードウエアを有効に利用することが可能となる．例えば，通常は 2 個の MVM を IDCT に割り当て，残りの 2 個を幾何変換に割り当てるものとするが，自然画像オブジェクトが人工画像オブジェクトより多くなると，それに応じて MVM を IDCT に割り当てる．

4　画像マッピングモジュール

つぎに，自然画像のための動き補償と人工画像のためのテクスチャマッピングを行なう画像マッピングモジュールについて述べる．まず，動き補償とテクスチャマッピングの概要について述べる．

4.1　動き補償

MPEG では，動画像が持つ時間的相関性を利用して，Fig. 5 に示すように，ブロック単位で過去のフレーム (参照画像) から現在のフレームを作成する動き補償を採用している．スクリーン上のブロック内の全ての画素に対し，動きベクトルを用いて参照画像内の画素の座標を求めなければならないが，参照画像内の画素の座標値は半画素精度で与えられるため，対応する値を求めるには，Fig. 6 に示すように，近隣 4 個の画素値を用いてハーフペル処理を行なわなければならない．このようにして生成された画素値を予測誤差に加算することにより，スクリーン上の画素値を求めることができる．

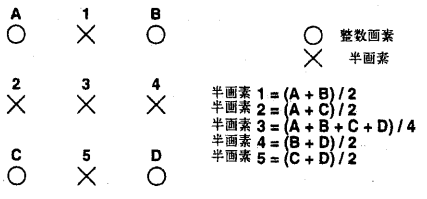


Fig. 6　ハーフペル処理.

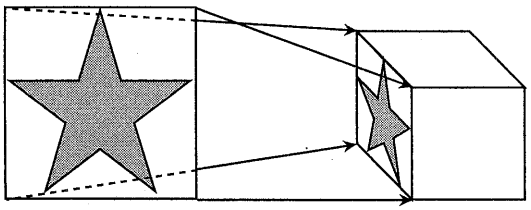


Fig. 7　テクスチャマッピング.

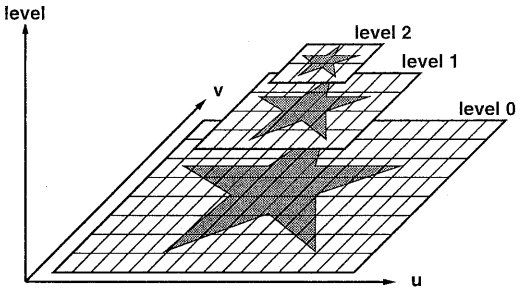


Fig. 8　ミップマップ.

4.2　テクスチャマッピング

多くの物体表面は，複雑なテクスチャ（図柄や模様）を持っている．これを比較的簡単に表現するために，テクスチャマッピングと呼ばれる手法が多く用いられている．テクスチャマッピングは，Fig. 7 のように，2 次元のテクスチャをスクリーン内の 3 次元の物体表面上に貼り付ける手法であり，テクスチャの複雑さに依存せずに高速に画像を生成することができる．さらに，テクスチャの形状変換を行なう際の計算量削減のため，ここではミップマップ法 [5, 6, 7] と呼ばれる手法を用いる．ミップマップ法では，スクリーン上の 1 画素値を求める際，テクスチャ内の対応する全画素値の平均を表示の度に計算するのではなく，Fig. 8 のように，あらかじめ数種類の解像度のレベルをもつ画像を準備しておき，このレベルから 1 画素の大きさに相当するテクスチャ領域を含むレベルを選択して利用することにより画素値を得る．

ここで，立体の表面は三角形のポリゴンによって近似するものとし，テクスチャマッピングの開始時に，スクリーン上のポリゴンの 3 頂点座標と，それぞれの頂点に対応するテクスチャ上の 3 頂点座標が与えられるものとする．まず，スクリーン上のポリゴン内のすべての画素の座標を求め，各画素座標に対応するテクスチャ上の領域を求める．次に，求めた領域を正方形で近似し，その領域に対応するミップマップ上の領域が 1 画素の大きさと近いミップマップの 2 つのレベルを選択する．各レベルでは，4 個の画素値を用いてバイリニア補間により対応する画素値を求める．最後に，補間された 2 レベルの画素値に，線形補間を施すことにより，表示すべき画素値が計算できる．Fig. 9 に 1 画素値を生成する過程を示す．

以上述べたように，動き補償とミップマップ法によるテクスチャマッピングは，共にスクリーン上の画素値を求めるために，テクスチャ座標あるいは参照画像の画素値を補間していることがわかる．このことに着目し，この 2 種類の処理を実行する専用モジュールのアーキテクチャを次節で述べる．

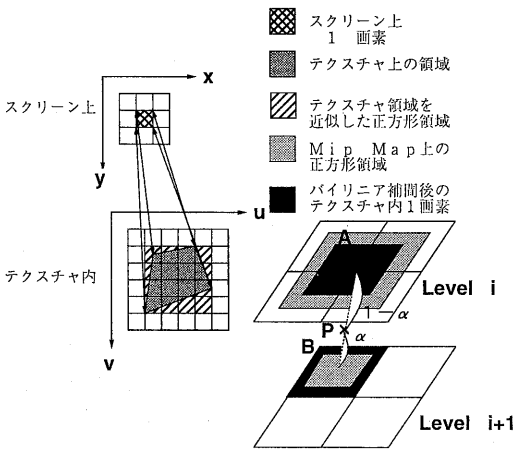


Fig. 9　ミップマップ法による画素値の生成.

4.3　画像マッピングモジュールのアーキテクチャ

Fig. 10に示すように，テクスチャマッピングと動き補償は類似した処理過程をもつ．網かけ枠で囲まれた処理内容は同一ハードウェアで実現可能である．この一連の処理を実行する画像マッピングモジュールのブロック図を Fig. 11 に示す．ここで，テクスチャ画像と参照画像を外部メモリに格納しているものとする．

テクスチャマッピングの処理は，動き補償の処理より複雑である．そこで，この専用モジュールでは，テクスチャマッピングを処理するアーキテクチャ上で動き補償を実現する．

このマッピングモジュールは 6 つの機能ユニットから構成される．Table 1 に各ユニットの両処理における機能をまとめている．このように，共通の機能ユニットを用いて動き補償とテクスチャマッピングを実行することができる．

Table 1　画像マッピングモジュールの各ユニットの機能

ユニット	動き補償	テクスチャマッピング
DDA	これから描画するブロックの左上端の点を基準にブロック内に含まれる点を順次出力する.	入力された3点で定義される三角形上の点を順次出力する.
Screen-Texture Transformer	動きベクトルにしたがって，参照画像上の点を求める.	スクリーン座標上の三点と，それぞれに対応するテクスチャ画像上の3点の関係からスクリーン座標とテクスチャ画像の関係を求めテクスチャ画面上の点を求める.
Level Finder	動き補償では不用.	スクリーン座標と参照画像での領域の大きさの違いからミップマップレベルの値を計算する.
Address Generator	座標とブロックの種類からアドレスを生成する.	座標とレベルからアドレスを生成する.
Mean Calculator	動きベクトルの半画素成分にしたがってハーフペル処理を行なう.	選択されたレベル内でのバイリニア補間を行なう.
Linear Intrerporator	予測誤差を加算する.	2つのレベル間で求められた値を補間する.

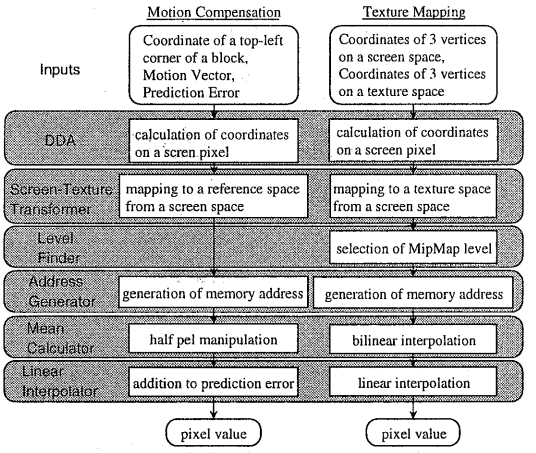


Fig. 10　動き補償とテクスチャマッピングの処理フローと機能ユニットの関係

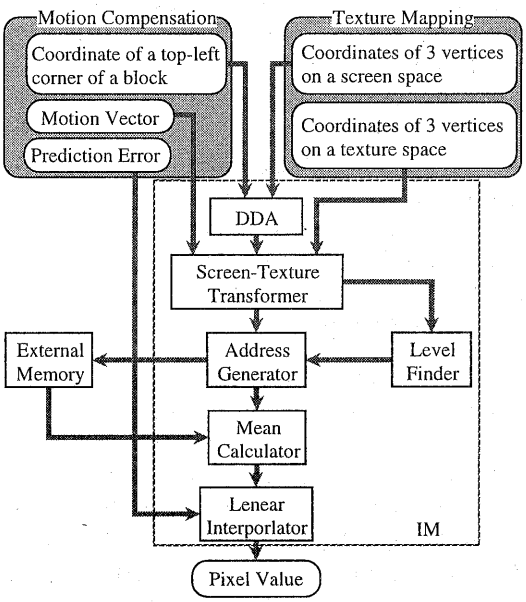


Fig. 11　画像マッピングモジュールのブロック図

5　実装結果

これまでに述べた2つの演算モジュールの実装結果を示す.

4個のMVMと1個のIDCT用転置バッファをVerilog-HDLで記述し，*COMPASS Design Navigator*により構築した実装結果をTable 2に，得られたレイアウトパターンをFig. 12に示す．クリティカルパスは47.7 nsであるので，動作周波数20MHzが可能となり，所望の処理能力が達成でき，さらに消費電力は20MHz動作時に412 mWとなる.

ここで，4個のMVMで実装した場合の消費電力削減効果について述べる．消費電力は動作周波数とトランジスタ数に比例するが，動作周波数を上げると，クリティカルパスを短縮するためのレジスタによる消費電力を考慮しなければならない．したがって，4個のMVMで実装すると1個のMVMで実装する場合と比較してトランジスタ数は多くなるが，回路全体の動作周波数を下げることができるため，4個を用いても電力消費は変化しない．さらにクロック周期が長くなるためレジスタを付加する必要がなくなり，1個のMVMを用いて実装するよりも633ビット分のレジスタの消費電力，約73mWを削減できる.

Table 2　変換モジュールの実装結果

テクノロジ	0.35μm 4層メタル CMOS スタンダードセル
面積	$7.62mm \times 1.93mm$
トランジスタ数	439,086
クリティカルパス	47.7ns
消費電力	20.6 mW/MHz

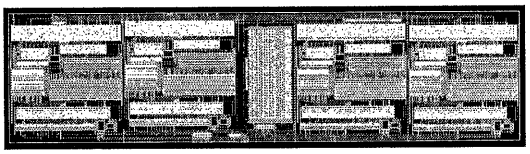


Fig. 12　変換モジュールのレイアウトパターン

4節で述べた，画像マッピングモジュールを 0.35 μm 4層 メタル CMOS スタンダードセルを用いて実装した結果を Table 3 に，得られたレイアウトパターンを Fig. 13 に示す．IDCT と幾何変換を同時に実行する MVM [2, 3] は 20 MHz で動作するが，本演算モジュールのクリティカルパスは，30.36 ns であるため，20 MHz で十分動作させることが可能となる．テクスチャマッピングでは，2 レベルのバイリニア補間を用いて線形補間を行なうので，2 サイクルに 1 画素値を出力する．したがって，20 MHz で動作させた場合，1 ポリゴンを平均 10 ピクセルとすると，その処理性能は，

$$10,000,000 \text{ 画素／秒} = 1,000,000 \text{ ポリゴン／秒},$$

となる．一方，動き補償の場合には，1 クロックに 1 画素を生成することができるので，1 マクロブロックを生成するためには 384 サイクルを必要とし，その処理性能は，

$$20,000,000 \text{ 画素／秒} = 52,083MB\text{／秒}$$

となる．これは，CIF (352 × 288) 画像を 30 フレーム／秒で表示する場合，4 枚分の画像の同時生成に相当する．したがって，時分割で合成画像と自然画像の処理を行なうと，処理の比率が一定の場合は，合成画像は 5,000,000 ポリゴン／秒で，自然画像は 2 枚分の CIF 画像を 30 フレーム／秒 で処理することができる．

さらに，変換モジュールについては，VDEC によるチップ試作サービスを利用して，2 個の MVM と 1 個の転置バッファをローム社の 0.6μm プロセスを用いて実装した．転置バッファにはメモリマクロを用い，それ以外の部分ではトランジスタ 227,824 個で $6.01mm \times 4.00mm$ の領域にレイアウトして，クリティカルパスが 86.1ns となった．

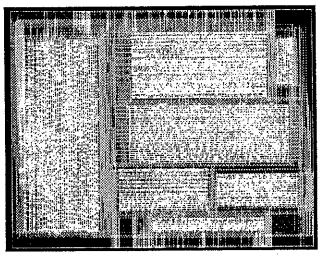


Fig. 13　画像マッピングモジュールのレイアウトパターン

Table 3　画像マッピングモジュールの実装結果

テクノロジ	0.35μm 4層メタル CMOS スタンダードセル
面積	1.85 mm×1.45 mm
トランジスタ数	112,656
クリティカルパス	30.36 ns

今後はこのチップを利用してシステムの構築を行なう予定である．

6　まとめ

自然画像と人工画像のハイブリッド処理を目的としたメディアプロセッサに向けて，IDCT と幾何変換のいずれにも必須の行列ベクトル乗算器 (MVM) のアーキテクチャを提案した．MVM は動作周波数 20MHz で動作させることができ，所望の性能を達成し，かつ消費電力の削減を行った．また，MVM を 4 個用いることにより，自然画像と人工画像のオブジェクトの出現比率に応じて，MVM の IDCT と幾何変換への割り当て比率を変えることができ，柔軟なハイブリッド処理が可能となる．

さらに，画像マッピングモジュールのアーキテクチャについても提案した．テクスチャマッピングと動き補償が共に画素参照，平均化処理等を主に実行していることに着目し，2 つの処理で共用可能な構造を考案した．このモジュールを用いることにより，テクスチャマッピングと動き補償の時分割処理が可能となり，省面積化を実現することができた．

今後の課題としては，メディアプロセッサの残された演算モジュールの設計が挙げられる．

7　謝辞

本チップ試作は東京大学大規模集積システム設計教育研究センターを通しローム（株）および凸版印刷（株）の協

力で行なわれたものである.

参考文献

[1] MPEG, "Coding of audio-visual objects : visual-committee draft," ISO/IEC JTC1/SC29/WG11 N2202, Mar. 1998.

[2] H. Fujishima, Y. Takemoto, T. Onoye, I. Shirakawa, S. Sakaguchi, "A unified media-processor architecture for video coding and computer graphics," in *Proc. Int. Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging*, pp. 253–256, Sep. 1997.

[3] 竹本 裕介, 藤嶋 秀幸, 尾上 孝雄, 白川 功 "動画像復号化と 3 次元コンピュータグラフィクス向き行列ベクトル乗算器のアーキテクチャ," 第 11 回 回路とシステム (軽井沢) ワークショップ, pp. 451–456, Apr. 1998.

[4] 竹本裕介, 米田友和, 藤嶋秀幸, 尾上孝雄, 白川功 "テクスチャマッピングおよび動き補償用共有回路の VLSI 化設計", 信学技報, VLD98-33 (1998-07)

[5] Paul S.Heckbert, "Survey of Texture Mapping," *IEEE CG&A*, Nov. 1986.

[6] Lance Williams, "Pyramidal Parametrics," *in Proc. SIGGRAPH 83*, Vol.17, pp.1–11, Jul. 1983.

[7] "Summed-Area Tables for Texture Mapping," *in Proc. SIGGRAPH 84*, Vol.18, pp.207–212, Jul. 1984.