

## MPEG-4 コアプロファイル符号化に対応した専用演算器を持つ DSP

石本 剛<sup>†</sup> 宮岡祐一郎<sup>†</sup> 戸川 望<sup>††,†††</sup> 柳澤 政生<sup>†</sup> 大附 辰夫<sup>†</sup>

<sup>†</sup> 早稲田大学理工学部電子・情報通信学科

<sup>††</sup> 北九州市立大学国際環境工学部情報メディア工学科

<sup>†††</sup> 早稲田大学理工学総合研究センター

〒 169-8555 東京都新宿区大久保 3-4-1

Tel: 03-3209-3211(5716), Fax: 03-3204-4875

E-mail: ishimoto@yanagi.comm.waseda.ac.jp

あらまし 本稿では MPEG-4 コアプロファイル符号化に対応した専用演算器を持つ DSP を提案する。提案 DSP はコアプロファイルを用いて QCIF 動画を 30fps で符号化可能である。本 DSP は専用ハードウェアとして形状符号化器、パディング器、量子化器、可変長符号化器を持つ。これらは MPEG-4 コアプロファイルで用いられる計算量を多く必要とする処理を実行する。専用ハードウェアは演算器形式とし、DSP から直接操作できる形とする。本 DSP では任意長のビットストリームのメモリアクセスを実現するため、ビットストリームロードユニットおよびビットストリームストアユニットを使用する。これらのユニットが形状符号化器や可変長符号化器から出力される符号のメモリへの書き出しを高速化し、符号化器のプロセッサへの組み込みを可能とする。これにより本 DSP は専用ハードウェアの性能とプロセッサとしての柔軟性の両立を実現する。本 DSP をチップ試作を通して評価した。試作チップは 0.35 $\mu\text{m}$  プロセスを用いて 40MHz で動作する。

キーワード DSP, MPEG-4, コアプロファイル, プロセッサ

## A DSP with Dedicated Functional Units for MPEG-4 Core Profile Encoding

Takeshi ISHIMOTO<sup>†</sup>, Yuichiro MIYAOKA<sup>†</sup>, Nozomu TOGAWA<sup>††,†††</sup>, Masao YANAGISAWA<sup>†</sup>, and Tatsuo OHTSUKI<sup>†</sup>

<sup>†</sup> Dept. of Electronics, Information and Communication Engineering, Waseda University

<sup>††</sup> Dept. of Information and Media Sciences, The University of Kitakyushu

<sup>†††</sup> Advanced Research Institute for Science and Engineering, Waseda University

3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan

Tel: +81-3-3209-3211(5716), Fax: +81-3-3204-4875

E-mail: ishimoto@yanagi.comm.waseda.ac.jp

**Abstract** This paper proposes a DSP with dedicated functional units for MPEG-4 core profile encoding. In the proposed DSP, we can have 30fps of QCIF for MPEG-4 core profile encoding, which no other LSIs for MPEG-4 have ever achieved. The proposed DSP has four dedicated functional units: a shape coding unit, a padding unit, a quantization unit, and a variable-length coding unit. These units execute processes which require much computational power for MPEG-4 core profile encoding. It also has a bitstream load unit and a bitstream store unit which realize a variable-length memory access. These units speed up reading and writing variable-length codes outputted from the shape coding unit and the variable-length coding unit, and make it possible to incorporate these coding units into the proposed DSP. Therefore, our DSP can achieve both the performance of dedicated hardwares and the flexibility of a processor. Our DSP has been implemented using 0.35  $\mu\text{m}$  CMOS technology and operates at 40MHz.

**Key words** DSP, MPEG-4, core profile, processor

## 1. はじめに

動画画像符号化の規格には主に MPEG-1, MPEG-2, MPEG-4 がある。この中で MPEG-1 はビデオ CD に、MPEG-2 は DVD やデジタル TV 等に用いられている。MPEG-4 は 1999 年に最初の国際標準化がなされたマルチメディア符号化規格である。MPEG-4 は移動体通信やインターネットを用いた動画配信等の用途を主なターゲットとしている。このため、MPEG-4 は高効率符号化、強力なエラー耐性、マルチメディア対応等の特徴を持っている。MPEG-4 の持つ特徴の中で、特に MPEG-1/2 との違いを明確にしているのがマルチメディアへの対応である。MPEG-1/2 では単一の動画画像・音声を対象としていたのに対して、MPEG-4 では CG、顔の動き、テキスト等を含めて統合的に扱うことができる。MPEG-4 ではシーン内のオブジェクトごとに符号化して復号時に合成するという、オブジェクトベースの符号化の概念を導入し双方向性を持たせることも可能となっている。この機能の実現のため、動画画像の符号化では任意形状の符号化に対応している。

MPEG-4 において現在最も使用されているプロファイルはシンプルプロファイルであり、NTT ドコモの FOMA で利用できる i モーション等でも使用されている。しかしシンプルプロファイルはエラー耐性ツールが追加されているものの、基本的には従来の MPEG-1/2 と同様の符号化である。オブジェクトベースの符号化という MPEG-4 特有の機能を活用できるのはコアプロファイル以上のプロファイルである。コアプロファイルでは 2 値形状情報を符号化することで、復号側で違和感のない合成が可能となっている。

ここ数年に多数の MPEG-4 対応 LSI が発表・発売されている。これらの多くはシンプルプロファイルのみに対応したものである。現在までに 1 チップかつ小面積でコアプロファイル符号化に対応した LSI は存在しない。コアプロファイルで用いる形状符号化向けのアーキテクチャを提案した文献 [2] はあるが、チップとしての実装やテキスト符号化との統合に至っていない。文献 [3] の LSI はコアプロファイル復号化に対応しているが、コアプロファイル符号化には対応しない。

本稿では形状符号化を含む MPEG-4 コアプロファイル符号化に対応した DSP を提案する。本 DSP では MPEG-4 符号化において多くの計算量を必要とする処理について専用ハードウェアを用い、演算器として扱うことで専用ハードウェアの高い性能と DSP として高い柔軟性を両立させる。本 DSP を VHDL で記述し、実チップ設計を通して面積・性能を評価した。

## 2. 既存研究と提案 DSP の特徴

MPEG-4 動画画像符号化 LSI の実現方法には主に、専用ハードウェアを中心とするアプローチ [3], [6], [8] と汎用プロセッサ上のソフトウェアによるアプローチ [4], [7], [10] の 2 通りがある。

専用ハードウェアを使用する場合、専用ハードウェアとそれらを制御する RISC プロセッサの組合せによる構成が一般的である。この構成では CIF 動画を 15fps でシンプルプロファイル符号化することが可能となっている [6]。文献 [3] は専用ハードウェアと DSP の構成で QCIF 15fps のシンプルプロファイル符号化および復号化、もしくはコアプロファイル復号化を実現している。文献 [2] は形状符号化を実行するハードウェアを提案しているが、テキスト符号化と組み合わせたシステムの提案には至っていない。

専用ハードウェアを中心とする構成では専用ハードウェアを並列に動作させて性能を満足することはできても、ソフトウェアのような柔軟性には欠ける。例えば、図 1 のような RISC+専用ハードウェアの構成において、図 2 のように専用ハードウェアをパイプライン動作させた場合を考える。フ

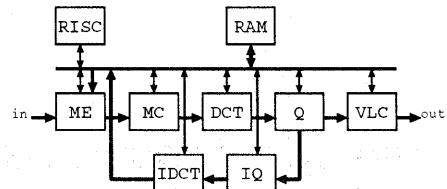


図 1 RISC+専用 HW アーキテクチャ例

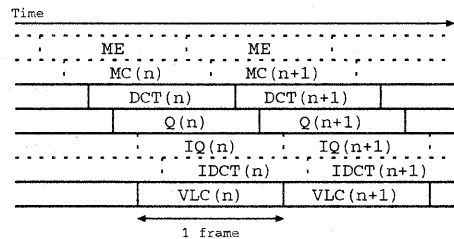


図 2 専用 HW パイプライン動作例

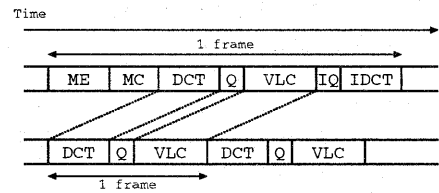


図 3 本 DSP での動作例

レーム間予測をしないことで動き予測や動き補償等の処理を削減しても、DCT等の処理がボトルネックとなりフレームレートの大幅な向上等は見込めない。

最近では強力なデータバスとクロック周波数の向上により、汎用 DSP を用いた MPEG-4 符号化処理の実装も可能となってきている。文献 [4] は 160MHz で動作する汎用 DSP を使用して CIF 動画を 15fps でシンプルプロファイル符号化している。文献 [10] は 133MHz で動作するアプリケーションプロセッサを使用して QCIF 動画を 15fps でシンプルプロファイル符号化している。ソフトウェアの処理は柔軟にクロックサイクルを割り振ることが可能なため、要求に応じた符号化が可能である。しかし、コアプロファイル符号化で用いられる処理はソフトウェアでの実行に過ぎず、専用ハードウェアを持たない DSP でコアプロファイルに対応したものは無い。

提案 DSP ではこれらを踏まえ、以下の 2 点を主な特徴とする。

- コアプロファイル符号化対応
  - DSP+演算器形式専用ハードウェアによる高い柔軟性
- コアプロファイル符号化で用いられる処理には形状符号化やパディング等、分岐やビット単位の処理が多く通常のソフトウェア処理では計算量を多く必要とするものがある。提案 DSP はコアプロファイル符号化で使用する形状符号化、パディング、量子化、可変長符号化の各処理に専用ハードウェアを用意することで必要な性能を得る。これらの専用ハードウェアは DSP から直接操作できる演算器の形式とする。形状符号化器や可変長符号化器からは可変長の符号が出力される。しかし、ソフトウェアのみでこのような任意長のビット列をメモリに書き出すにはビット単位の操作を必要とし、高速化が困難である。提案 DSP は任意長のビット列を読み書きするためのユニットをメモリアクセス部に付加し、可変長の符号をメモリに書き出せるようにする。これにより可変長の符号を出力する専用ハードウェアを持つことが可能となる。

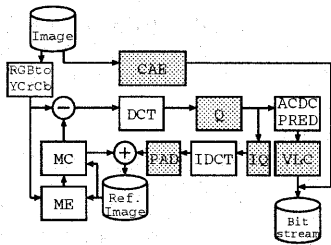


図4 MPEG-4符号化ブロック図

提案DSPで使用する演算器形式の専用ハードウェアは並列に動作することはできないが、ソフトウェア動作と同様に柔軟にクロックサイクルを割り振ることが可能である。このため図3のようにフレーム間予測を省いてフレームレートを2倍にするといったことも可能である。

### 3. ハードウェアアーキテクチャ

#### 3.1 DSP 構成

MPEG-4 コアプロファイル符号化処理のブロック図を図4に示す。DSPはビット単位の操作や多数の分岐、除算を含む処理を苦手とし、加減算や積和演算を中心とした処理を得意とする。MPEG-4 コアプロファイル符号化に用いる処理の中で形状符号化(CAE)は2値形状情報を符号化する処理で、算術符号化を用いるためソフトウェアでは多くの分岐を含み、ビット単位の操作を多数必要とする。パディング(PAD)は画素単位の処理であるが形状情報に応じて隣接画素の値のコピーや平均を繰り返すため、ソフトウェアで実現すると分岐が多く逐次的な実行となる。量子化(Q)の主な演算は除算であり、除算器を持たないDSPでは実現が困難である。可変長符号化(VLC)は対象によって符号表の参照回数や出力ビット数が大きく変わり、ソフトウェア実行では命令数を見積もるのが困難である。これらの処理を専用ハードウェアが実行することで、高速かつ確定的な命令数で処理を完了することができるようになる。一方、動き予測(ME)やDCTを含む他の処理は、加減算や積和演算を中心としたDSPの得意とする処理である。これらはALUで実行することとし、各処理を効率良く実行できるALU構成とすることで性能の向上を図る。

提案DSPの構成を図5に、基本仕様を表1に示す。形状符号化器(CAE)、パディング器(PAD)、量子化器(Q/IQ)、可変長符号化器(VLC)は演算器形式の専用ハードウェアとしてDSPからALUと同様に扱う。ビットストリームロードユニットとビットストリームストアユニットは任意長のビット列の毎サイクルのロードやストアを可能とする。形状符号化器と可変長符号化器は可変長の符号を出力するが、このような任意長のビット列のロードやストアは一般のプロセッサが不得意とするビット単位の操作を多く必要とする。これらのユニットにより形状符号化器や可変長符号化器からの出力を効率良くメモリに書き出すことができる。

専用ハードウェアは前述のように演算器の形をとり、入力は1つまたは2つ、出力は1つである。全ての専用ハードウェアは内部にレジスタを持つ。一定の処理が終了するまで、同じハードウェアを使用する別の処理による割り込みはできないが、他のハードウェアを使用する処理であれば割り込みも可能である。データメモリ(DMEM)とアドレッシングユニットは2つずつ持つ。演算操作と並列にメモリアクセスをすることで、データ転送量の多い処理においてメモリアクセスがボトルネックにならないようにする。

以下に本DSPを構成する各ハードウェアの概要を示す。

#### 3.2 専用演算器

演算器形式の専用ハードウェアを専用演算器と呼ぶ。以

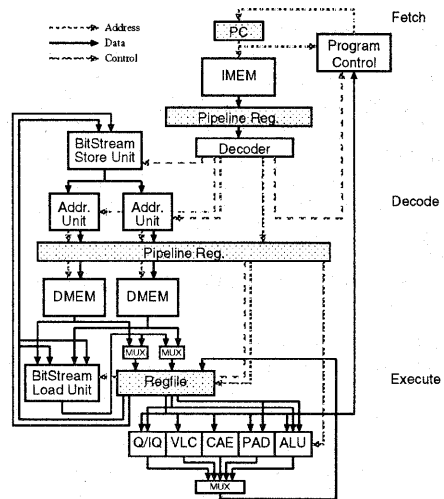


図5 DSP 構成

表1 DSP 基本仕様

パイプライン段数	3 (fetch/decode/execute)
命令構成	演算命令 × 1 メモリアクセス命令 × 2
汎用レジスタ	32ビット × 32word
メモリアクセス幅	54ビット(命令) 8/16/32ビット × 2(データ)

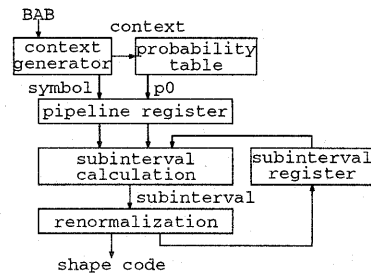


図6 形状符号化器

下に各専用演算器の概要を示す。

##### 3.2.1 形状符号化器(CAE)

MPEG-4 コアプロファイルにおける形状符号化では現フレームの近傍10画素(intraCAE)、または現フレームの近傍4画素と前フレームの近傍5画素の合計9画素(interCAE)をコンテキストとしたコンテキストベース算術符号化(CAE: Context based Arithmetic Encoding)が用いられる。本DSPの形状符号化器は、後述するビットストリームストアユニットと協調して動作させることで可変長符号のメモリへの書き出しを高速化し、1画素あたり1命令で算術符号化を実行する。

形状符号化器のブロック図を図6に示す。形状符号化器では16×16画素の形状符号化をBAB(Binary Alpha Block)の読み込みを含めて285命令で処理できる。形状符号化器は最初に2ライン分の符号化に必要なBABを読み込み、算術符号化を開始する。2ラインの符号化が終わるごとに新たに必要となるBABを読み込み算術符号化を続行する。形状符号化器は内部を2段のパイプライン構成とし、第1ステージでコンテキストの生成と確率モデルの参照を、第2ステージで部分区間の計算と符号の出力をする。本符号化器では'0'の連続を抑制するbit stuffingに対応していないため、後処理としてbit stuffing処理を施す必要がある。

### 3.2.2 パディング器 (PAD)

パディングには境界マクロブロックに対する繰り返しパディング、境界マクロブロックに接する外部のマクロブロックに対する拡張パディング、更に外部のマクロブロックに対する一様パディングの3種がある。パディング器では最も複雑な処理である繰り返しパディングを対象とする。

パディング器の設計は文献 [1] の手法を採用する。パディング器ではBABを参照しながらマクロブロックを横方向および縦方向に往復して走査し、繰り返しパディングを施す。入力のパディングの対象となる画素情報とBABであり、出力はパディングを施した途中経過あるいは結果の画素情報である。

### 3.2.3 量子化器 (Q/IQ)

量子化器の入力はDCT係数(量子化)または量子化DCT係数(逆量子化)と量子化ステップであり、出力は量子化DCT係数(量子化)または逆量子化DCT係数(逆量子化)である。

MPEG-4の量子化は1つのDCT係数に対して $8 \times 8$ の量子化マトリクス係数による除算と量子化ステップによる除算の2つの除算を用いる。量子化器では1係数あたり1命令のスループットで量子化するため、内部を2段のパイプライン構成として各ステージにそれぞれ除算器を設ける。内部に持つ量子化マトリクスを利用する逆量子化にも量子化器を用いる。

### 3.2.4 可変長符号化器 (VLC)

可変長符号化処理は $8 \times 8$ のブロックをジグザグスキャン、縦方向優先スキャン、横方向優先スキャンのいずれかの方法で走査し、非ゼロの量子化DCT係数に対し最大3回の符号表参照により可変長符号を得る。可変長符号化をソフトウェアで実現する場合は、係数がゼロであるか否かや符号表の参照回数等によって多数の分岐を必要とするため効率的な処理ができない。出力が可変長符号のビットストリームであることもソフトウェア実行性能を低くする原因となる。

本DSPの可変長符号化器は、ソフトウェア上での分岐を解消するため1係数につき4命令固定で処理する。形状符号化器の入力は量子化DCT係数であり出力は可変長符号である。形状符号化器と同様にビットストリームストアユニットと協調して動作させることでメモリへの書き出しを高速化する。量子化DCT係数のスキャンはメモリからの読み出しの際に専用のアドレッシングモードを用いることで実現する。

## 3.3 ALU

専用演算器を用意しない処理は全てALUを用いる。ALUもMPEG-4の各処理を効率的に実行できるよう強化した構成とする。ALUは乗算器バス、加算器バス、シフト、梱包開梱器、論理演算器、特殊演算器で構成される。ALUは32ビットを基本演算語長とし16ビット2梱包や8ビット4梱包でのSIMD演算もサポートする。以下にALUの各構成要素について示す。

### 3.3.1 乗算器バス

乗算器バスは乗算や積和演算を主に実行する。ALUで実行する処理のうち、DCT/IDCTは乗算を含む処理である。乗算器バスはこれらの処理を効率化するために、4つの16ビット乗算器と得られた結果を足し合わせる加算器を持つ。

DCT/IDCT高速化アルゴリズムではしばしば以下のような計算が用いられる。 $f_i$  と  $f_j$  はDCT演算で用いられる定数である。

$$x = a * f_i + b * f_j \quad (1)$$

$$y = b * f_i - a * f_j \quad (2)$$

この計算を1命令で実行するため、DCT演算で用いる定数と4つの乗算器を乗算器バス内部に持ちDCT専用命令で使用する。DCT専用命令を用いると256回の乗算と416回の加算が必要な $8 \times 8$  DCTをメモリアクセスを含

めて248命令で処理できる。RGBtoYCrCb変換でもこれらの乗算器を使用して1画素につき3命令で処理する。乗算器バスは通常の演算としてSIMDに対応した乗算や積和演算も受け持つ。

### 3.3.2 加算器バス

加算器バスはSIMDに対応した加減算、最大値最小値演算、絶対値演算、平均値演算の他、動き予測におけるブロックマッチングのための差分絶対和演算を実行する。差分絶対和演算は、2つの8ビット4梱包の値から各フィールドごとに差分の絶対値を計算し、4つの結果全てと更にもう1つの値を加算して出力する演算である。

### 3.3.3 シフト

シフトはSIMDに対応したシフト演算を実行する。シフト量は可変で、シフト方向、算術シフト/論理シフト、SIMDの梱包数に応じて1サイクルで実行する。

### 3.3.4 その他

梱包開梱器はSIMD演算のための梱包、開梱、組み換えを実行する。32ビット1梱包と16ビット2梱包、16ビット2梱包と8ビット4梱包の相互間の変換、同じ梱包数内での要素の組み換え、符号拡張等を実行する。

論理演算器は一般的な論理演算を実行する。論理演算はビットごとに独立した処理でありSIMD対応は特にしない。

特殊演算器はビット単位のインタレース化や最上位の'1'や'0'のビット位置を返す演算を実行する。前者は形状符号のサブサンプリング等に後者は可変長符号化に使用する。

## 3.4 メモリアクセスユニット

本DSPは2系統のデータメモリを持つ。データメモリは命令中に2つあるメモリアクセス用のフィールドにそれぞれ対応する。データメモリへのアクセスに関するユニットとして、データメモリの他にアドレッシングユニット、ビットストリームロードユニット、ビットストリームストアユニットがある。アドレッシングユニットは2つありデータメモリと1対1で対応する。ビットストリームロードユニットとビットストリームストアユニットは1つずつである。これらは2つのデータメモリで共有する。

### 3.4.1 データメモリ (DMEM)

データメモリはそれぞれ32ビット幅としSIMDの梱包数に応じたメモリアクセスを可能とするため8ビット単位での書き込み制御ができるものとする。メモリアクセス幅に応じたデータの整形はアドレッシングユニットを用いる。

### 3.4.2 アドレッシングユニット

2つのアドレッシングユニットはそれぞれ4つのアドレスレジスタと、各アドレスレジスタに対応するインデックスレジスタを持つ。アドレッシングは全てポスト修飾アドレッシングで、インクリメント、デクリメント、インデックスレジスタの加減算、ビットリバースモードでのインデックスレジスタの加算の他、3種類のブロック走査を実現する加減算のモードがある。ブロック走査のためのモードでは、アドレッシングユニット内にカウンタとインデックスの表を持ち、各サイクルごとに更新することで走査する。

アドレッシングユニットはストアするデータをアクセス幅とアドレスに応じて正しいメモリ位置に書き込まれるよう整形する機能を持つ。

### 3.4.3 ビットストリームロードユニット

ビットストリームロードユニットは0-24、32ビットの任意のビット数のデータをロードするためのユニットである。ビットストリームロードユニットは内部に64ビットのレジスタとバレルシフトを持つ。あらかじめ内部のレジスタにメモリからデータをロードし、要求に応じたビット数のデータを出力する。必要に応じてメモリから32ビット分のデータをロードする。

ビットストリームロードユニットと後述するビットスト

xx...xx1001.....101 bit length

bit length

図7 ビットストリームデータ形式

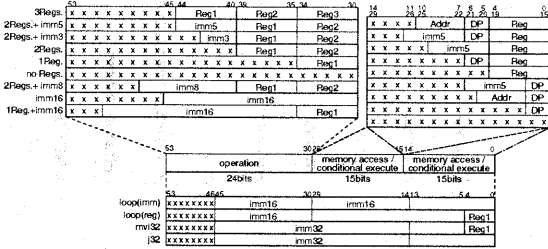


図8 命令フィールド構成

ストリアユニットは図7に示すビットストリームデータ形式を使用する。この形式は任意のビット長のデータを表す形式で、32ビットを分割し、上位24ビットに右詰めでデータを、下位8ビットにそのビット長を格納する。ビットストリームロードユニットでデータをロードするとビットストリームデータ形式でレジスタに格納される。

### 3.4.4 ビットストリームストアユニット

ビットストリームストアユニットはビットストリームデータ形式のデータをメモリに格納するためのユニットである。形状符号化器や可変長符号化器はビットストリームデータ形式で可変長符号を出力する。

ビットストリームストアユニットはビットストリームロードユニットと同様に64ビットのレジスタとパレルシフトを持つ。このユニットにビットストリームデータ形式のデータを入力し32ビット分が入力されるごとにメモリ上に書き出すことでビットストリームの出力を可能としている。

## 4. 命令セット

本DSPの命令長は54ビットで固定である。命令フィールドの構成を図8に示す。命令は基本的に1つの演算命令と2つのメモリアクセス命令から構成される。以下に本DSPの命令セットについて示す。

### 4.1 演算命令

演算命令は24ビットで演算器に関する操作とプログラム制御をする。演算命令は最大で3つのレジスタを指定する。積和演算等3つのレジスタから値を読み出す命令は1つのレジスタを書き込みにも使用する。専用演算器を使用する命令もALUを使用する命令と同じ形式を取る。

プログラム制御には分岐先を即値でPC相対指定するジャンプ命令とレジスタで絶対指定するジャンプ・レジスタ命令がある。どちらも制御の単純化のため遅延分岐とする。

即値はシフト命令、一部の専用演算器命令、即値をオペランドに取る加算命令および乗算命令、即値代入命令、ジャンプ命令で使用する。命令長の制限により加算命令と乗算命令では即値長は8ビットとし、即値代入命令とジャンプ命令では16ビットとする。即値代入命令とジャンプ命令は32ビットの即値を指定する場合、後述する54ビット長命令を使用する。

演算命令フィールドを使用する命令はALU命令が82命令、専用演算器命令が85命令、プログラムコントロールを含むその他の命令が7命令で合計174命令である。

### 4.2 メモリアクセス命令

メモリアクセス命令のフィールドはそれぞれ15ビットでメモリアクセスに関する操作または条件実行の指定をする。

メモリアクセスに関する命令は通常のアクセスでは汎用レジスタの番号とアドレスレジスタの番号、アドレッシングモードを指定する。即値を用いたアドレッシングは無く、あ

表2 試作チップ諸元

プロセス	ROHM 0.35 μm CMOS 3層メタル
回路規模	10万ゲート(2入方nand換算, SRAM除く)
動作周波数	40.9MHz(24.43ns)
電源電圧	3.3V
面積	4.9 × 4.9mm <sup>2</sup> (コア領域 3.8 × 3.8mm <sup>2</sup> )
信号ピン数	109
パッケージ	160pin QFP

らかじめ汎用レジスタに代入した上でアドレスレジスタやインデックスレジスタに転送する必要がある。ビットストリームロード命令やビットストリームストア命令ではアドレッシングはインクリメントのみとする。

条件実行命令は汎用レジスタ1つと条件を指定する。条件実行命令により分岐を減らして性能を向上することができ、一連の処理を確定的な命令数で実行できる。指定できる条件は、レジスタから読み出した数値をxとして $x=0, x \neq 0, x > 0, x < 0, x \geq 0, x \leq 0$ のいずれかである。条件実行命令が命令中にある場合、レジスタへの書き込みやPCの更新は条件が成立したときのみとなる。条件分岐は条件実行命令とジャンプ命令の組合せで実現する。また、条件実行命令はSIMDに対応する。レジスタから読み出した値を16ビット2梱包や8ビット4梱包と比較して、対応するフィールドのレジスタへの書き込みを制御する。

メモリアクセス命令フィールドを使用する命令は通常のメモリアクセス命令が14命令、ビットストリームメモリアクセス命令が20命令、条件実行命令が3命令で合計37命令である。

### 4.3 54ビット長命令

即値が長く演算命令のフィールドに入り切らない場合にメモリアクセス命令フィールドと統合して54ビット長命令として単独で発行する。54ビット長命令には32ビットの即値を用いる命令とループ命令がある。

32ビットの即値を用いる命令には32ビット即値代入命令と32ビット即値ジャンプ命令がある。これらの命令はメモリアクセス命令と並列に発行することができない。32ビットの即値命令は、メモリアクセス命令と並列に発行できる16ビットの即値命令と状況に応じて使い分ける。

ループ命令はハードウェアループを開始する命令である。ループ命令は1イタレーションの命令数を即値で指定し、ループの回数を即値またはレジスタから読み出して指定する。ループにはハードウェアループを使用し、オーバーヘッドを伴わない繰り返し処理ができる。ループの深さは4段までである。

## 5. LSI 試作

本DSPをVHDLで記述し配置配線をして、東京大学大規模集積システム設計教育センター(VDEC)のチップ試作サービスに試作のためのデータを提出した。試作チップは2003年3月に納品予定である。試作では使用可能なIOピン数とチップ面積の制約から命令メモリを外部に、データメモリを内部に置く。データメモリにはそれぞれ8ビット256ワードのSRAMマクロを4つずつ用いた。

VHDLの記述量は約12700行となった。VHDL記述をSynopsys Design Compilerで論理合成しネットリストを得た。使用ライブラリはROHM 0.35μmプロセス向けPassportライブラリである。23nsの遅延制約を与えて論理合成した結果、クリティカルパス遅延は制約を満たし、メモリとIOを除くセル面積は5.09mm<sup>2</sup>となった。

論理合成によって得られたネットリストを元にAvant! Apolloを使用して配置配線をした。試作チップの諸元を表2に示す。

表3 QCIF 15fps の符号化に必要なサイクル数

処理		Mcycles/s
(a)	動き予測	3.918
	差分画像作成	0.571
	DCT	2.210
	量子化	0.615
	係数予測	0.135
	可変長符号化	2.444
	逆量子化	0.633
	IDCT	2.807
	動き補償	0.421
	フレーム再構成	0.571
(b)	形状符号化	1.304
	パディング	0.657
合計	simple (a)	14.325
	core (a)+(b)	16.286

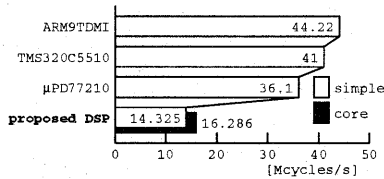


図9 QCIF15fps 符号化の所要サイクル数

## 6. 性能評価

本DSPでMPEG-4コアプロファイル符号化に必要な各処理の命令列を記述しクロックサイクル数を見積もった。QCIF画像を15fpsで符号化するのに必要なクロックサイクル数を表3に示す。動き予測には3step探索手法を用いた。本DSPを40MHzで動作させてQCIF画像をMPEG-4符号化した場合、シンプルプロファイル符号化で37.9fps、コアプロファイル符号化で33.7fpsの処理性能となる。

他のRISCプロセッサやDSPとの比較を図9に示す。比較対象としてARM9TDMIコア[7]、TMS320C5510[9]、μPD77210[4]を用いた。本DSPはこれらのプロセッサと比較して半分以下のクロックサイクル数でシンプルプロファイル符号化が可能となっている。

MPEG-4コアプロファイル符号化の主な処理について本DSPでの性能とソフトウェア実行での性能を比較した。ソフトウェア実行環境はSun Blade1000 (CPU: UltraSPARC III 750MHz, RAM: 1GB, gcc-2.95.3) と Dell Dimension4100 (CPU: Pentium III 866MHz, RAM: 512MB, Visual C++ 6.0) で最適化オプションは共にO2とした。動き予測を除きMPEG-4参照ソフトウェア[5]をコンパイルして実行し実行時間を測定した。動き予測は参照ソフトウェアと本DSPのサイクル数見積もりで探索手法が異なるため、3step探索手法をC言語で実装して使用した。QCIF画像15フレーム分の処理に必要な実行時間を図10に示す。Blade1000とDimension4100では可変長符号化と形状符号化がそれぞれ大きな割合を占めており全体の性能を低下させる要因となっている。符号化処理そのものの複雑さに加え汎用プロセッサがビットストリームの取り扱いに適さないことが原因である。本DSPではそれぞれ専用の符号化器を用意しビットストリームストアユニットを用いてメモリに書き出すことにより実行時間の大幅な削減を実現している。

## 7. むすび

MPEG-4コアプロファイル符号化向けDSPを提案した。試作チップが納品され次第テストを実施して動作を確認し消費電力等を測定する予定である。

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てローム(株)および凸版印刷(株)の協力で行われたものである。本研究はVDEC IPコアプロジェクトの一貫として実施されたものである。

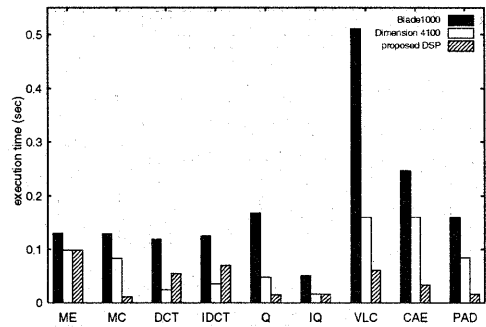


図10 QCIF15 フレームの符号化に必要な実行時間

## 文献

- [1] M. Berekovic, H. J. Stolberg, M. B. Kulaczewski, P. Pirsch, H. Möller, H. Runge, J. Kneip, and B. Starbarnack, "Instruction set extensions for MPEG-4 video," *Journal of VLSI Signal Processing Systems*, Vol. 23, No. 1, pp. 27-50, 1999.
- [2] H. C. Chang, Y. C. Chang, Y. C. Wang, W. M. Chao, and L. G. Chen, "VLSI architecture design of MPEG-4 shape coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 9, pp.741-751, 2002.
- [3] T. Hashimoto, S. Kuromaru, M. Matsuo, Y. Kohashi, T. Mori-iwa, K. Ishida, S. Kajita, M. Ohashi, M. Toujima, T. Nakamura, M. Hamada, T. Yonezawa, T. Kondo, K. Hashimoto, Y. Sugisawa, H. Otsuki, M. Arita, H. Nakajima, H. Fujimoto, J. Michiyama, Y. Iizuka, H. Komori, S. Nakatani, H. Toida, T. Takahashi, H. Ito, and T. Yukiitake, "A 90mW MPEG4 video codec LSI with the capability for core profile," in *ISSCC Digest of Technical Papers*, pp. 140-141, 2001.
- [4] A. Hatabu, T. Miyazaki, and I. Kuroda, "QVGA/CIF resolution MPEG-4 video codec based on a low-power and general-purpose DSP," in *Proc. IEEE Workshop on Signal Processing Systems*, pp. 15-20, 2002.
- [5] ISO/IEC 14496-5: 2001, "Information technology—coding of audio-visual objects—Part 5: reference software," 2001.
- [6] H. Nakayama, T. Yoshitake, H. Komazaki, Y. Watanabe, H. Araki, K. Morioka, J. Li, L. Peilin, S. Lee, H. Kubosawa, and Y. Otake, "An MPEG-4 video LSI with an error-resilient codec core based on a fast motion estimation algorithm," in *ISSCC Digest of Technical Papers*, pp. 368-369, 2002.
- [7] R. S. V. Prasad and K. Ramkishor, "Efficient implementation of MPEG-4 video encoder on RISC core," in *International Conference on Consumer Electronics 2002 Digest of Technical Papers*, pp. 278-279, 2002.
- [8] M. Takahashi, T. Nishikawa, M. Hamada, T. Takayanagi, H. Arakida, N. Machida, H. Yamamoto, T. Fujiyoshi, Y. Ohashi, O. Yamagishi, T. Samata, A. Asano, T. Terazawa, K. Ohmori, Y. Watanabe, H. Nakamura, S. Minami, T. Kuroda, and T. Furuyama, "A 60-MHz 240-mW MPEG-4 videophone LSI with 16-Mb embedded DRAM," in *IEEE Journal of Solid-State Circuits*, pp. 1713-1721, 2000.
- [9] Texas Instruments, *OMAP: Enabling Multimedia Applications in 3G Wireless Terminals*, <http://www-s.ti.com/sc/psheets/swpa001/swpa001.pdf>, 2000.
- [10] T. Yamada, N. Irie, J. Nishimoto, Y. Kondoh, T. Nakazawa, K. Yamada, K. Tatezawa, T. Irita, S. Tamaki, H. Yagi, M. Furuyama, K. Ogura, H. Watanabe, R. Satomura, K. Hirose, F. Arakawa, T. Hattori, I. Kudo, I. Kawasaki, and K. Uchiyama, "A 133MHz 170mW 10μA standby application processor for 3G cellular phones," in *ISSCC Digest of Technical Papers*, pp. 370-371, 2002.