システムＬＳＩ設計技術　104－15
(2002. 1. 24)

MPEG-4 コア・プロファイル・コーデックの VLSI アーキテクチャ

中川　貴史† 濱中　慎介† 肖　云和† 藤田　玄† 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

あらまし　本文では，MPEG-4 コア・プロファイル・コーデックの VLSI アーキテクチャに関して述べる．MPEG-4 のシンプル・プロファイル・コーデックに関してはすでに多くの報告が行われているが，形状符号化が可能であり，要求性能がより高いコア・プロファイル・コーデックの報告例は少ない．本文では，コア・プロファイル・コーデックの設計にあたり，各処理過程を行うモジュールのバス幅や演算能力を入力とする性能評価システムを構築する．これにより，コア・プロファイルの要求性能を満たしつつ，最小のバス・演算器構成を効率よく求めることが可能となる．また，形状符号化に関連する各モジュールのアーキテクチャも提案する．

キーワード　MPEG-4, 動画像符号化, VLSI

VLSI Architecture for MPEG-4 Core Profile Video Codec

Takashi NAKAGAWA†, Shinsuke HAMANAKA†, Xiao YUN HE†, Gen FUJITA†, and Isao SHIRAKAWA†

† Department of Information Systems Engineering, Osaka University, 2-1 Yamada-Oka, Suita, Osaka, 565-0871 Japan

Abstract　In this paper a VLSI architecture of the MPEG-4 Core Profile video codec is described. Although a great number of authors have reported on the MPEG-4 Simple Profile codec, there rarely exists the one of such high performance as to be capable of the shape codec. Thus, first a performance verification model is formed with the use of parameters of bath widths and calculation performances of functional modules, each constructed as a specific processing unit of the total codec, and then these parameters are attempted to be optimized to make the MPEG-4 Core Profile codec attain the maximum possbile performance. The shape codec core, motion estimation core, and motion compensation core architectures are also described dedicatedly for the object base coding.

Key words　MPEG-4, video codec, VLSI

1. はじめに

MPEG-4 [1] は低ビットレート (38.4kbps～2Mbps) を対象として開発された動画像符号化の国際標準であり，1999 年に Version 1 が勧告されている．MPEG-4 は従来の動画像圧縮アルゴリズムに比べて，オブジェクトごとに符号化/ 復号化が実行できることやエラー耐性の増強が図られていることなどが大きな特長である．

W-CDMA が採用している Ezweb での動画配信には MPEG-4 シンプル・プロファイルが用いられており，このシンプル・プロファイルのコーデック用の VLSI に関しては多くの報告がなされている [3]. MPEG-4 の最大の特長であるオブジェクト符号化を実行するためにはコア・プロファイルに対応する必要があるが，復号化 VLSI に関しては報告例 [4] があるものの，コーデックに関する報告例はない．

MPEG-4 の実装には，ソフトウェアによる実装とハードウェアによる実装が考えられる．しかし，移動体通信においては低消費電力化の観点からハードウェア実装が有利であり，さらにコア・プロファイルによって実時間通信を行う場合，組込み向け汎用プロセッサの処理能力には限界があるため，専用ハードウェアが不可欠となる．

専用ハードウェアの設計はコストが大きいが，シンプル・プロファイルの設計資産を有効に活用することができれば，設計期間の短縮につながる．しかしながら，シンプル・プロファイルとコア・プロファイルとの要求性能比に応じて各モジュールの性能を向上させただけでは，新たなボトルネックが生じ，要求性能を満たさない可能性がある．十分な性能マージンを考慮して設計を行えば，このような事態は防止できるが，冗長な性能が消費電力の向上につながり，専用ハードウェア化の利点を失ってしまう．

このような問題を解決するには，アーキテクチャ設計段階において，各処理を行う専用モジュールの構成からコーデック全体の性能を正確に見積もり，最適な構成を得ることが重要となる．

本研究では，MPEG-4 の既存の設計モジュールの再利用を前提に，各専用モジュールの演算器数やバス幅から面積，処理能力が最適となるアーキテクチャを検討するための評価システムを構築する．これを用いて最適な構成の MPEG-4 コア・プロファイル対応のビデオコーデックのアーキテクチャを提案する．

以下 2 章では本研究で提案する MPEG-4 コア・プロファイル対応のビデオコーデックのアーキテクチャについて概要を述べ，続く 3 章では本研究で構築した評価モデルとそのシミュレーション結果を述べる．次の 4～6 章では評価モデルによって得られた結果に基づいた主要モジュールの詳細なアーキテクチャを説明する．

2. アーキテクチャ概要

本研究で提案するコア・プロファイル対応の MPEG-4 ビデオコーデックコアのアーキテクチャを以下の図 1 に示す．各ブロックは機能モジュールを表し，表 1 に各モジュールの役割を示す．

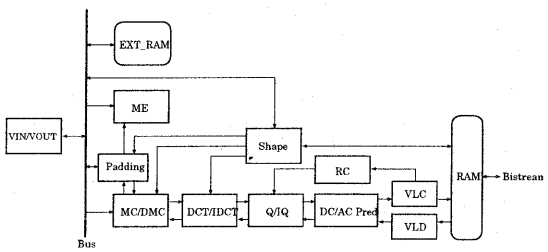


図 1　MPEG-4 ビデオコーデックコアのアーキテクチャ

表 1　モジュールの機能

名称	機能
VIN/VOUT	ビデオ入力/出力
EXT_RAM	外部メモリ
Shape	形状情報符号化/復号化
ME	動き検出
MC/DMC	動き補償
Padding	パディング
DCT/IDCT	離散コサイン変換/逆離散コサイン変換
Q/IQ	量子化/逆量子化
DC/AC Pred	DC/AC 予測
VLC	可変長符号化
VLD	可変長復号化
RAM	ストリームバッファ
RC	レートコントロール

符号化時はまず，画像データおよびその形状情報が VIN/VOUT より入力され，EXT_RAM に書き込まれる．これを用い，Shape による形状情報，ME によって算出された動きベクトルを元に，MC および DCT によって差分画像が周波数領域に変換され，その後 Q，DC/AC Pred によって冗長情報の削減が行われた後，VLC によって可変長符号化を行い，ビットストリームを出力する．最後に，量子化後のデータを，IQ において逆量子化，IDCT において逆離散コサイン変換，DMC によって動き補償を行い，次フレームの参照画像として EXT_RAM に書き込み，次フレームの予測に用いる．ME 以下の処理を

総称して以後テクスチャ符号化処理と呼ぶ．

復号化を行う場合は，ストリームバッファに書き込まれたビットストリームデータに対して，VLD において可変長復号化が行われた後，上記と同様の過程を経て EXT_RAM に復元画像として書き込まれ，VIN/VOUT を通じて出力される．

提案アーキテクチャは，これら一連の MB (マクロブロック) の処理を全て専用ハードウェアによるパイプライン処理にて行うことが大きな特徴である．一部をソフトウェア化した場合と比べ，柔軟性は劣るが，外部ホストプロセッサに負荷がかからないため，多様なシステムへの組込みが容易となる．

このようなアーキテクチャを，要求性能を満たしつつ小面積化するには，各モジュールの演算性能，外部バスやモジュール間のバス帯域等のパラメータを最適に設定する必要がある．本研究では，次章に述べる評価モデルを構築し，複数の動画シーケンスによるシミュレーションを行うことによって最適なパラメータ設定を行う．

3. MPEG-4 評価モデル

3.1 概　　要

本研究により構築した，提案コア・プロファイルアーキテクチャの評価モデルの概要を図 2 に示す．

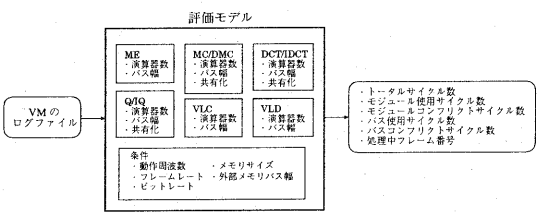


図 2　MPEG-4 評価モデルの概要

各モジュールの演算器数，バス幅等のパラメータを設定し，入力として MPEG-4 リファレンスソフトウェア [2] により画像シーケンスを復号化した際に得られるログファイルを与えることによって，処理全体のサイクル数，各モジュールの使用率，各モジュール間のコンフリクト率，フレームメモリバスの占有率やコンフリクト率，シミュレーション終了時点での処理フレーム番号 (VIN, ME, Shape, テクスチャ別) が出力される．ログファイルには MB スキップや可変長符号化のシンボル数の情報が含まれており，高精度な性能評価が可能である．また，DCT/IDCT などのモジュールを共有した場合の評価も容易に行うことができる．

本評価モデルにおいて，各モジュールは以下のステップによりそれぞれ独立して動作し，最終的なサイクル数を導出する．

（1） 以下の条件が整えば入力動作を行う．入力に必要なサイクル数はバス幅等のパラメータにより求まる．

（a） 外部メモリデータを用いる場合は，外部バスが他のモジュールに使用されていないこと．

（b） 他モジュールから入力データが必要な場合は，そのデータが準備されていること．

（2） 各モジュール固有の演算処理を行う．演算処理に必要なサイクル数は演算器数等のパラメータにより求まる．

（3） 以下の条件が整えば出力動作を行う．出力に必要なサイクル数はバス幅等のパラメータにより求まる．

（a） 外部メモリデータを用いる場合は，外部バスが他のモジュールに使用されていないこと．

（b） 他モジュールへ出力する場合は，そのモジュールが入力可能な状態であること．

（4） 以上を MB あるいはブロックを単位として繰り返す．

提案するアーキテクチャは対象としてコア・プロファイル レベル 2，携帯用途を想定しているため，表 2 に示す諸元を用いてシミュレーションを行う．

表 2　シミュレーション諸元

条件	値
動作周波数	33MHz
サンプル画像	Foreman 他
フレームレート	60fps
ビットレート	1Mbps
符号化順序	I, B, B, P, B, B, P, B, ⋯ (30 フレーム毎に強制的に I フレームを挿入)

3.2 評 価 結 果

シミュレーションを行うにあたり，設計資産であるシンプル・プロファイル相当の動画像圧縮アルゴリズム H.263 の VLSI アーキテクチャ構成 [5] をデフォルトのパラメータとして用いる．当然ながらこれでは処理速度が不足するため，性能の向上が必要となる．全てのパラメータを同時に調整しては結果が収束しないおそれがあるため，1) ME に十分な性能を与える．2) テクスチャの各モジュールの性能を演算の大きなモジュールから順に最適化．3)ME および外部バスの性能を最適化 という 3 ステップにより，パラメータの調整を行う．

3.2.1 結果1 (デフォルト値)

デフォルト値によりシミュレーションを行った結果を表3に示す．各数値は，それぞれの画像に対しての各モジュールの使用サイクル数とコンフリクトサイクル数である．使用サイクル数とは，当該モジュールが動作しているサイクル数，コンフリクトサイクル数とは，当該モジュールが動作中のため，他の本来動作すべき他のモジュールが停止してしまうサイクル数を表す．括弧内の数字は各モジュールごとの使用率およびコンフリクトの割合を示す．

また，表4には処理を行なったフレーム数およびその他のモジュールによる停止サイクル数を示す．表中のV, T, M, Sはそれぞれビデオ入力，テクスチャ符号化，ME，形状符号化を表す．T停止における項目Vは，ビデオ入力が終了していないためにテクスチャ処理が停止しているサイクル数を表す．T, M, Sのフレーム番号がVのフレーム番号から離れた値である場合，T, M, Sの処理性能が不十分であることを示す．逆に，T停止, M停止, S停止のサイクル数が大きいと，それぞれオーバースペックであることを意味する．表4からはT, Mの処理能力が不足していることがわかるため，前節に示した手順により，パラメータの最適化を図る．

表3　結果1 (デフォルトのモジュール使用率)

Name	Module Usage 千単位 (使用率)	Module Conflict 千単位 (使用率)
MC	74,963 (45%)	72,043 (43%)
DCT	109,486 (66%)	48,939 (29%)
Q	13,686 (8%)	214 (0%)
IQ	333 (0%)	9,545 (6%)
VLC	619 (0%)	214 (0%)
IDCT	43,329 (26%)	27,218 (16%)
DMC	26,069 (16%)	3,520 (2%)
ME	156,442 (94%)	1,830 (1%)
SHAPE	160,277 (97%)	115 (0%)
Total cycle		165,983

表4　結果1 (デフォルト値)

フレーム番号				停止サイクル数 (千単位)								
					T停止			M停止			S停止	
V	T	M	S	V	M	S	V	T	S	V	T	M
302	89	91	291	0	2,781	0	0	2,859	0	0	0	0

3.2.2 テクスチャパラメータ最適化(DCT/IDCTを例に)

本節ではテクスチャの各モジュールパラメータ調整の一例としてDCT/IDCTのパラメータ調整に関して述べる．DCT/IDCTの処理能力を向上には2通りのアプローチがある．1つは，DCT/IDCTを別のモジュールとして設計する方法，もう1つは，DCT/IDCTの演算器数を増やす方法である．表5にDCT/IDCTの演算器数およびDCT/IDCTを共有，非共有にしたときの結果を示す．

表5　結果2 (DCT/IDCTの演算器数および共有化による性能評価)

演算器	フレーム番号				停止サイクル数 (千単位)								
					T停止			M停止			S停止		
	V	T	M	S	V	M	S	V	T	S	V	T	M
共有化のとき													
1	298	89	91	285	0	436	0	0	94,867	0	0	0	0
2	162	89	91	155	0	436	0	0	23,783	0	0	0	0
4	115	89	91	108	0	6,818	0	0	0	0	0	0	0
8	114	89	91	107	0	11,188	0	0	0	0	0	0	0
16	114	89	91	107	0	11,188	0	0	0	0	0	0	0
非共有のとき													
1	264	89	91	253	0	439	0	0	77,545	0	0	0	0
2	147	89	91	140	0	436	0	0	15,765	0	0	0	0
4	114	89	91	107	0	9,388	0	0	0	0	0	0	0
8	113	89	91	106	0	12,951	0	0	0	0	0	0	0
16	113	89	91	106	0	12,951	0	0	0	0	0	0	0

表5より，演算器数が4以上の場合，DCT/IDCTを共有化した場合と共有化しない場合において，処理能力に差がないことがわかる．また，これ以上演算器数を増やしても性能の向上が見られないため，演算器数4かつ共有させたケースが最適であることがわかる．

3.2.3 外部メモリのバス幅およびMEの最適化

前節のように，テクスチャの各処理部の最適なパラメータを順に算出する．しかし，MEの演算能力を最大に設定しているにも関わらず，要求性能を満足する性能が得ることはできない．原因としては，外部メモリの帯域不足が考えられるため，バス幅を増加させることによって要求性能を達成する．表6にそのシミュレーション結果を示す．

表6　結果3 (外部メモリのバス幅による変化)

バス幅	フレーム番号				停止サイクル数 (千単位)								
					T停止			M停止			S停止		
	V	T	M	S	V	M	S	V	T	S	V	T	M
2	115	89	91	109	0	2,611	0	0	151	0	0	0	0
4	97	89	91	96	0	383	0	0	843	0	830	0	0
8	92	89	91	91	0	334	47	2,281	1,761	0	1,326	0	0
16	92	89	91	91	0	1,203	1,021	5,581	822	0	1,522	0	0
32	92	89	91	91	0	1,603	1,466	7,086	642	0	1,595	0	0
64	92	89	91	91	0	1,794	1,683	7,836	621	0	1,627	0	0

表6より，外部メモリの帯域を8ワード/サイクルとすれば要求性能を満たすことが分かる．現状では，MEの演算器数は256と過剰に設定しているため，最後にMEのパラメータ調整を行う．表7にそのシミュレーション結果を示す．

表 7　結果 3 (ME の演算器数による変化)

演算器	フレーム番号				停止サイクル数 (千単位)								
					T 停止			M 停止			S 停止		
	V	T	M	S	V	M	S	V	T	S	V	T	M
8	260	89	91	259	0	93,236	0	0	0	0	2,973	0	0
16	159	89	91	158	0	37,120	0	0	0	0	2,019	0	0
32	112	89	91	111	0	11,222	0	0	0	0	1,529	0	0
64	93	89	91	92	0	544	0	80	401	0	1,331	0	0
128	92	89	91	91	0	334	47	2,281	1,761	0	1,326	0	0
256	92	89	91	91	0	334	47	2,281	1,761	0	1,326	0	0

表 8　評価モデルのパラメータ

内容	デフォルト値	最適値
外部メモリのバス幅	2	8
ME の演算器数	8	64
MC と外部メモリ間のバス幅	2	8
MC の演算器数	2	1
MC と DCT 間のバス幅	1	1
DCT の演算器数	1	4
DCT と Q 間のバス幅	1	1
Q における 1 画素処理に必要なサイクル数	1	1
Q と IQ 間のバス幅	1	1
Q における 1 画素処理に必要なサイクル数	1	1
IQ と IDCT 間のバス幅	1	1
IDCT の演算器数	1	4
IDCT と DMC 間のバス幅	1	1
DMC の演算器数	2	1
MC/DMC 共有化フラグ	共有	共有
DCT/IDCT 共有化フラグ	共有	共有
Q/IQ 共有化フラグ	非共有	共有

表 7 の結果より，ME の演算器数は 64 が最適であることがわかる．これにより，提案アーキテクチャのコア・プロファイル レベル 2 を満足する最適なパラメータが求められた．表 8 にその一部を示す．

4. 動き検出器のアーキテクチャ

動き検出に関しては，現在までに様々な高速アルゴリズム手法が提案されてきたが，本研究では専用ハードウェア化に最適な Full Search アルゴリズムを基盤とする，処理効率の高い動き検出器を提案する．本動き検出器の特長は以下の 2 点である．

- 検索範囲を適応的に変更することが可能
- オブジェクトの形状を利用した高速な動き検出処理が可能

Full Search アルゴリズムを用いた 2 次元 PE (Processing Element) アレイで構成される従来の動き検出器 [6] は，探索範囲が固定されており，例えば PE 数 64 の構成では X，Y 方向とも -4〜$+3$ 画素の探索を並列に実行する．しかしながら CIF では解像度に見合った，さらに広い探索範囲が必要となる．よって本動き検出器では，PE の探索範囲を適応的に変更できる PE アレイを提案する．このアレイ構造により，従来の動き検出器と同数の PE で見かけ上の検索範囲を広めることが可能である．

また本動き検出器ではオブジェクトの形状に特化した動き検出処理が可能である．2 次元 PE アレイによって構成される従来の動き検出器を用いてオブジェクト境界の MB の処理を行うと，オブジェクト外に相当する画素分だけ PE の稼働率が低下する．本手法では，オブジェクト内の画素を優先して PE へ入力することにより，PE の稼働率を高め，処理時間を削減することが可能である．ただし，オブジェクトの複雑な形状に対応するためには膨大な参照メモリの帯域が必要となってしまうため，本手法では図 3 に示すようにオブジェクトを完全に包含し，かつ X 方向の画素数が閾値以上となる矩形領域に対して動き検出を行うことによって，必要となるメモリ帯域を最小限に留めつつ境界 MB に対する処理の高速化を図る．

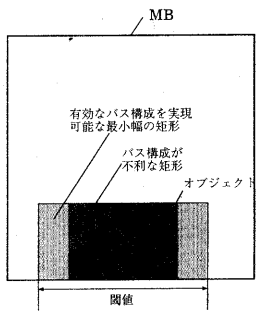


図 3　矩形の定義

評価モデルにより求めたパラメータをもとに，動き検出器の最適な構成を提案する．図 4 に動き検出器の全体構成を示す．

提案する動き検出器は，形状検出器，参照画像を保持するメモリ，差分絶対値和を求める PE，参照画像や現画像入力用のバス，動きベクトルを求めるアキュムレータにより構成される．

本動き検出器は検索対象の MB に対して基本的に -4〜$+3$ 画素の検索範囲を想定し，64 の PE によって構成される．提案する PE は図 5 で示すように，8×8 から 16×4，もしくは 4×16 に構成を切替えることが可能であり，これに伴って最大 -8〜$+7$ 画素の検索範囲に対する探索が可能となる．

2 次元 PE アレイを効率よく動作させるため，参照画像の入力は 4 画素分同時に入力可能となっている．オブジェクトの境界 MB の探索時にはこの参照画像入力の帯域を有効に活用することにより，上記の矩形領域に対する探索を可能としている．このようなオブジェクト境界の MB に対する処理の処理時間削減効果を検証するため，シミュレーションによ

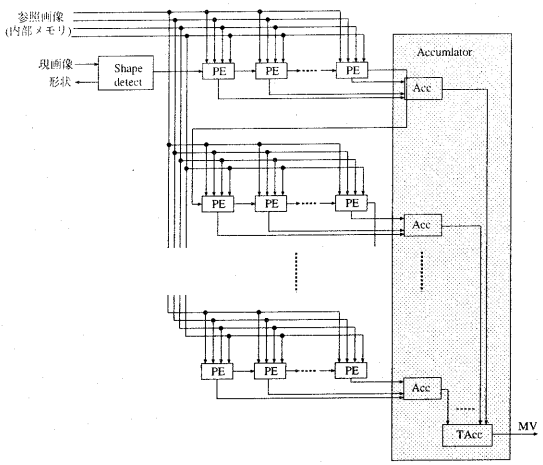


図 4　動き検出器の全体構成

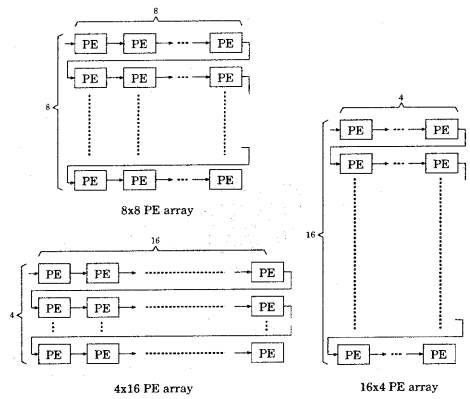


図 5　PE アレイの構成

り処理時間の比較を行った．

表 9　各動き検出対象による処理時間 (Weather)

動き検出対象	閾値	サイクル数	削減率
Only object	N/A	6621	0.73
Rectangle	0	7106	0.79
Rectangle	4	7144	0.79
Rectangle	**8**	**7278**	**0.81**
Rectangle	12	7723	0.85
Full Search	N/A	9045	1.00

表 9 は，シミュレーションで得られた MB 処理サイクル数である．“Only object” はオブジェクト内の画素に対してのみ動き検出を行った場合であり，“Rectangle” はオブジェクトを包含する矩形領域に対して動き検出を行った場合の結果である．対応する閾値は矩形領域の X 方向の最小画素数を表す．また削減率は “Full Search” の処理時間に対する各提案手法の処理時間の割合を示す．

シミュレーション結果より，矩形幅の閾値が 8 の場合は閾値が 0 や 4 の場合の削減率とほぼ変わらないことがわかる．閾値が 0 や 4 のケース，あるいは Only object のケースでは参照画素入力の帯域がさらに必要となるため，本研究では，既存の 4 画素分の画像用バスにより実現可能な閾値 8 を採用する．

5. 動き補償とパディング処理部のアーキテクチャ

評価モデルより算出したパラメータをもとに，動き補償器のアーキテクチャを提案する．本動き補償器の特長は以下の 2 点である．

- 動き補償を最小単位で行うことによる内部バッファの削減
- パディング処理における帯域幅の削減

動き補償ではブロック単位の処理が可能であるが，パディングは仕様上，MB 単位の処理が必要である．本研究では動き補償はブロック単位，パディングは MB 単位という最小構成を採用することによって内部バッファの削減を図る．

パディングに関して，通常は復元画像を外部メモリへ書き戻す際にパディング処理を行うが，本提案アーキテクチャではパディング処理を分割し，拡張パディングに関しては，その情報が次フレームの参照画像として使用される際に補間を行うことにより，外部メモリとの帯域を削減している．このようは処理を実現するため，提案する動き補償器は，参照画像は符号・復号時の双方ともパディング処理部を介して外部メモリから読み込む構成とした．図 6 に動き補償器とパディング処理部の全体構成を示す．

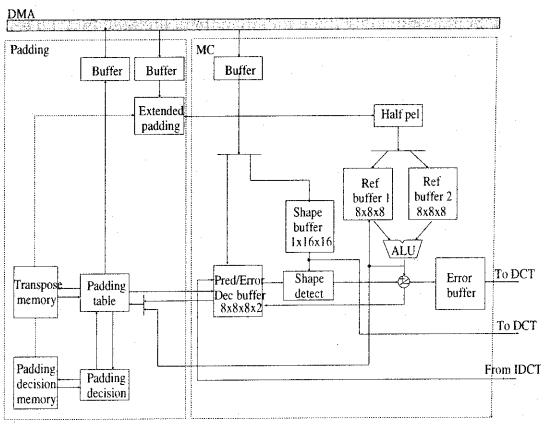


図 6　動き補償器とパディング処理部の全体構成

評価モデルによるシミュレーション結果により，本

動き補償器では2ブロック分の参照バッファを用いてブロック単位で処理を行う.

6. 形状符号化/復号化部のアーキテクチャ

形状情報も画素情報と同様に符号化/復号化処理を行う. 形状情報の符号化処理は, 図7に示すように, Mode Decision (以下 MD), BME, BMC, Size Control (以下 SC), CAE の5つの処理によって実行される. 入力された形状情報は, まず, MD によって符号化方法を決定し, 符号化を行う場合は, BME で動き検出を行い, BMC において動き補償を行う. 次に SC にてサブサンプリングのサイズを決定し, CAE にて算術符号化を行い, ビットストリームとして出力する.

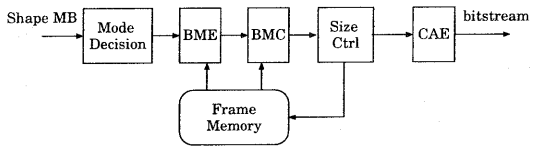
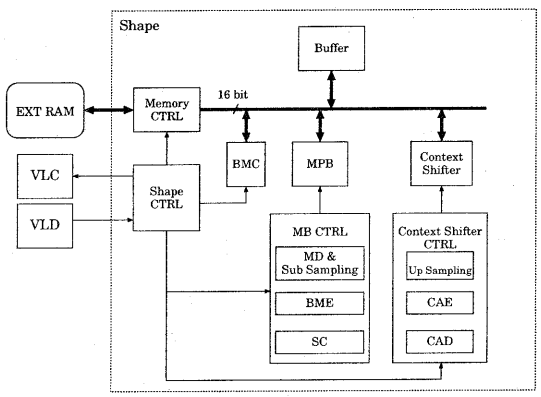


図7 形状符号化処理

形状情報の復号化処理は, 符号化時と逆の演算を行うことによって可能である. 具体的には, VLD からの入力を算術復号化してさらにアップサンプリングすることによって元のデータが復元される.

提案する形状情報符号化/復号化部の全体構成を以下の図8に示す. 本アーキテクチャは後述するように各処理の大半を2種類の回路によって共有化しており, 小面積を実現している.



VLC: Variable Length Coding
VLD: Variable Length Decoding
BME: Binary Shape Motion Estimation
BMC: Binary Shape Motion Comensation
SC: Size Control
CAE: Context-based Arithmetic Coding
CAD: Context-based Arithmetic Decoding
MPB: Multi-Process Block Circuilt

図8 形状情報符号化/復号化部の全体構成

MD および BME, SC の一部(サブサンプリング, サイズ決定)における演算は大半が累積演算で占められている. これら3つの処理を, MPB (Multi-Process Block Circuilt) により共有化し, MB CTRL によって各処理の制御を行う.

また, CAE と SC の一部(アップサンプリング)では以下に示す共通点がある.

- 画素毎に処理を行い, 近傍の画素を参照する.
- 同じデータ構造を持っており, 同じ画素を左にシフトする演算がある.
- CAE とアップサンプリングのコンテクスト導出にはよく似た演算を行う.

これらの共通点を利用し, CAE, SC の処理を Context Shifter により共有化を行っている.

7. ま と め

本研究では, MPEG-4 の既存の設計モジュールの再利用を前提として, 各専用モジュールの演算器数やバス幅から, 面積, 処理能力が最適となるアーキテクチャを検討するための評価システムを構築し, これを用いて最適な構成の MPEG-4 コア・プロファイル対応のビデオコーデックのアーキテクチャを提案した.

今後の課題としては, 評価システムに復号化処理を取り入れることによる Full Duplex コーデック可能なアーキテクチャ構成の評価, 処理単位の評価を MB だけでなく, フレームやスライス単位によって行うことなどが挙げられる.

文 献

[1] ISO/IEC 14496-2: "Generic coding of audio-visual objects Part2: Visual," International Standard, Jan. 1999.

[2] ISO/IEC 14496-5: "Generic coding of audio-visual objects Part5: Reference Software," International Standard, Jan. 2000.

[3] Chi-Weon Yoon, Ramchan Woo, Jeonghoon Kook, Se-Joong Lee, Kangmin Lee, Young-Don Bae, In-Cheol Park and Hoi-Jun Yoo: "A 80/20MHz 160mW multimedia processor integrated with embedded DRAM MPEG-4 accelerator and 3D rendering engine for mobile applications," in *IEEE ISSCC Digest of Tech. Papers*, pp. 142–143, Feb. 2001.

[4] Takashi Hashimoto, Shun-ichi Kuromaru, Masatoshi Matsuo, Yasuo Kohashi, Toshihiro Mori-iwa, Kenichi ishida, Satoshi Kajita, Masahiro Ohashi, Masayoshi Toujima, Tsuyoshi Nakamura, Mana Hamada, Tomonori Yonezawa, Takahiro Kondo, Kohkichi Hashimoto, Yuji Sugisawa, Hiroki Otsuki, Miki Arita, Hiromasa Nakajima, Hitoshi Fujimoto, Junji Michiyama, Yasuo Iizuka, Hiroyuki Komori, Shintaro Nakatani, Hiroaki Toida, Toshiya Takahashi, Hiroyuki ito, and Takeshi Yukitake: "A 90mW MPEG-4 video codec LSI with the capability for core profile," in *IEEE ISSCC*

Digest of Tech. Papers, pp. 140–141, Feb. 2001.

[5] Morgan Hirosuke Miki, Gen Fujita, Takao Onoye, and Isao Shirakawa: "Low power implementation of H.263 codec core dedicated to mobilecomputing," *IEICE Trans. Fundamentals*, vol. J81-A, no.10, pp.1352-1361, Oct. 1998.

[6] Jun-Fu Shen, Tu-Chih Wang, and Liang-Gee Chen: "A novel low-power full-search block-matching motion-estimation design for H.263+," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 7, pp. 890–897, Jul. 2001.