

マルチメディアのための専用型動画像 符号化処理方式の研究

藤原 洋[†] 岡田 豊[†] 小林 孝之[†]
上符 浩男^{††} 丸山 優徳^{†††}

デジタル信号メディア、デジタル蓄積メディアの発展に伴い画像圧縮を始めとするマルチメディア符号化技術の研究が盛んになってきた。特に、研究の重点は、信号処理アルゴリズムから高解像度テレビやコンピュータの高精細ディスプレイに表示可能なように大容量データをリアルタイム符号化処理することのできる高速処理アーキテクチャへ移行してきている。ここでは、現行テレビから将来の超高解像度ディスプレイへの表示も考慮した専用型マルチメディア符号化処理アーキテクチャを提案し、併せてVLSI実装例を示すものである。

Research on Specific Architecture of Video Coding for Multi-Media Information

HIROSHI FUJIWARA,[†] YUTAKA OKADA,[†] TAKAYUKI KOBAYASHI,[†]
HIROO UWABU^{††} and MASANORI MARUYAMA^{†††}

Research on the multi-media coding technology, especially video coding, becomes more active. First of all, weight of the research has shifted from the basic algorithm to the high-speed processing architecture which can handle tremendous data of HDTV or high-resolution computer display at real-time. In this paper, we propose a specific architecture for multi-media coding architecture in order to process even Ultra-Definition Television Signal in near future.

1. ま え が き

近年、オーディオ信号、画像信号のデジタル信号処理の発展により、コンピュータ、家電機器を中心にマルチメディア化が急速に進展している。このマルチメディア化の鍵を握る技術として、オーディオ情報、画像情報を統一されたデジタルフォーマットで取り扱うとともに、おのおのの情報を特性に応じた最も効率的な手法によって符号化（圧縮）することが強く求められている。一般に、オーディオ情報の符号化は、アルゴリズムは複雑であるが単位時間当たりの情報処理量は少ない。また画像情報の符号化は、アルゴリズムは比較的単純であるが、単位時間当たりの情報処理量が極めて多い。したがって、この画像符号化処理の

リアルタイム高速処理が今後の情報処理の重要なテーマとして位置づけられると考えられる。

デジタル信号処理に基づく画像符号化（圧縮）技術は、従来からデジタル通信網によるテレビ会議/電話等への適用を目的に研究が進められてきた。しかしながら、ここへきて、近年の符号化アルゴリズムと半導体技術の進展に加え、光記録技術、光磁気記録技術の進歩によってコンピュータを始めとする情報処理機器が画像情報を扱うことが現実味を帯びてきている。

テレビ会議用画像符号化方式 CCITT H. 261¹⁾は、伝送速度 $P \times 64$ kbps ($P=1 \sim 30$)、最大 352×288 の有効解像度を持つ方式として 1990 年 12 月に CCITT 勧告 H. 261 として勧告されている。現在 ISO では、蓄積メディア用符号化方式として、低解像度用の MPEG1²⁾に引き続いて、伝送速度 5 Mbps～数十 Mbps、解像度 $720 \times 480 \sim$ HDTV 品質に及ぶ高解像度画像を対象とした符号化方式を MPEG 2 として、標準化作業を行っている。

このように、通信用とは異なる蓄積メディア用画像

[†] (株)グラフィックス・コミュニケーション・ラボラトリーズ

Graphics Communication Laboratories

^{††} 京セラ(株)中央研究所
KYOCERA CORPORATION

^{†††} (株)日立製作所中央研究所
Hitachi Ltd.

符号化処理に要求されることは、解像度で4~20倍程度、ランダムアクセス性、および逆転、早送り再生等の付加機能であるが、特に重要になってくるのは、数千~数万 MOPS (Million Operations Per Second) に及ぶ膨大な演算量を実行可能なリアルタイム処理アーキテクチャの確立である。

これまで情報処理のためのプロセッサとしてストアードプログラム方式の CISC や RISC あるいは、信号処理に適した VSP (Video Signal Processor) が考えられてきたが³⁾、これらのプロセッサアーキテクチャによって近未来に所定の要求性能を実現することは極めて困難である。また、シストリックアレイやトランスピュータ等の高度並列処理システムが提案されているが、大規模システムとなるため小型化、量産化には現時点では適さない。

以上のような背景の下に、本論文では、マルチメディア情報の中でも画像情報、特にカラー動画像信号の今後の高解像度化の発展トレンドをふまえ、その符号化・復号化アルゴリズムの特徴を分析するとともに、おのおののアルゴリズム要素、特に符号化演算の大半を占める動きベクトル検出部と、画質を大きく左右する符号量制御部の処理に最適なアーキテクチャを明確化する。そして最後に、これらのアーキテクチャに基づいて試作を行った VLSI チップの実装例について示す。

2. 画像符号化の発展トレンドとアルゴリズムの特徴

画像信号の持つ冗長成分を除去し、情報圧縮を実現するには、現在以下の三つのアプローチが主流となっている。

- ① 空間成分の冗長成分除去：直交変換，ベクトル量子化等
- ② 時間成分の冗長成分除去：フレーム間予測，動き補償等
- ③ データ表現形式の冗長成分除去：ハフマン符号化，算術符号化等

実際の CCITT, ISO 等の国際標準化においても、これら三つの方式のいずれか、および組み合わせによるハイブリッド符号化方式が取り入れられている。これら三つのほかにも、画像対象物体についての知識に基づく知的符号化、分析合成符号化⁴⁾、フラクタル符号化⁵⁾等が研究されているが、国際標準化の提案方式になるまでには至っていない。

そこで、国際標準化が進められている画像符号化方式と、それらの利用対象となると考えられる現在から近未来の蓄積・通信・放送の各メディアの開発動向との関連を図1に示す⁶⁾。本図において、最終利用者端での映像品質を受信品質、放送局から送信する時の映像品質を分配品質と呼んでいる。今後、画像符号化の対

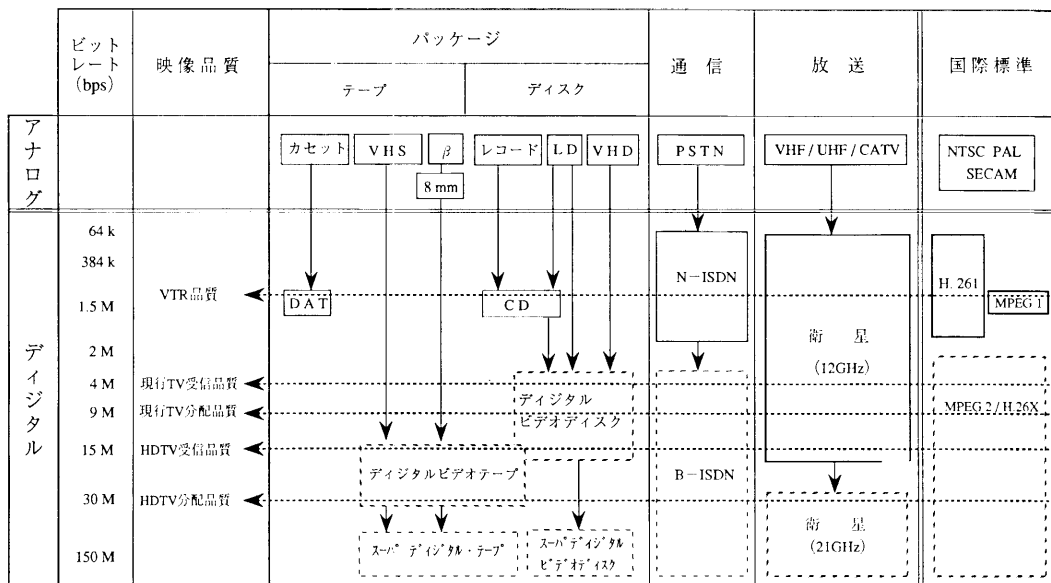
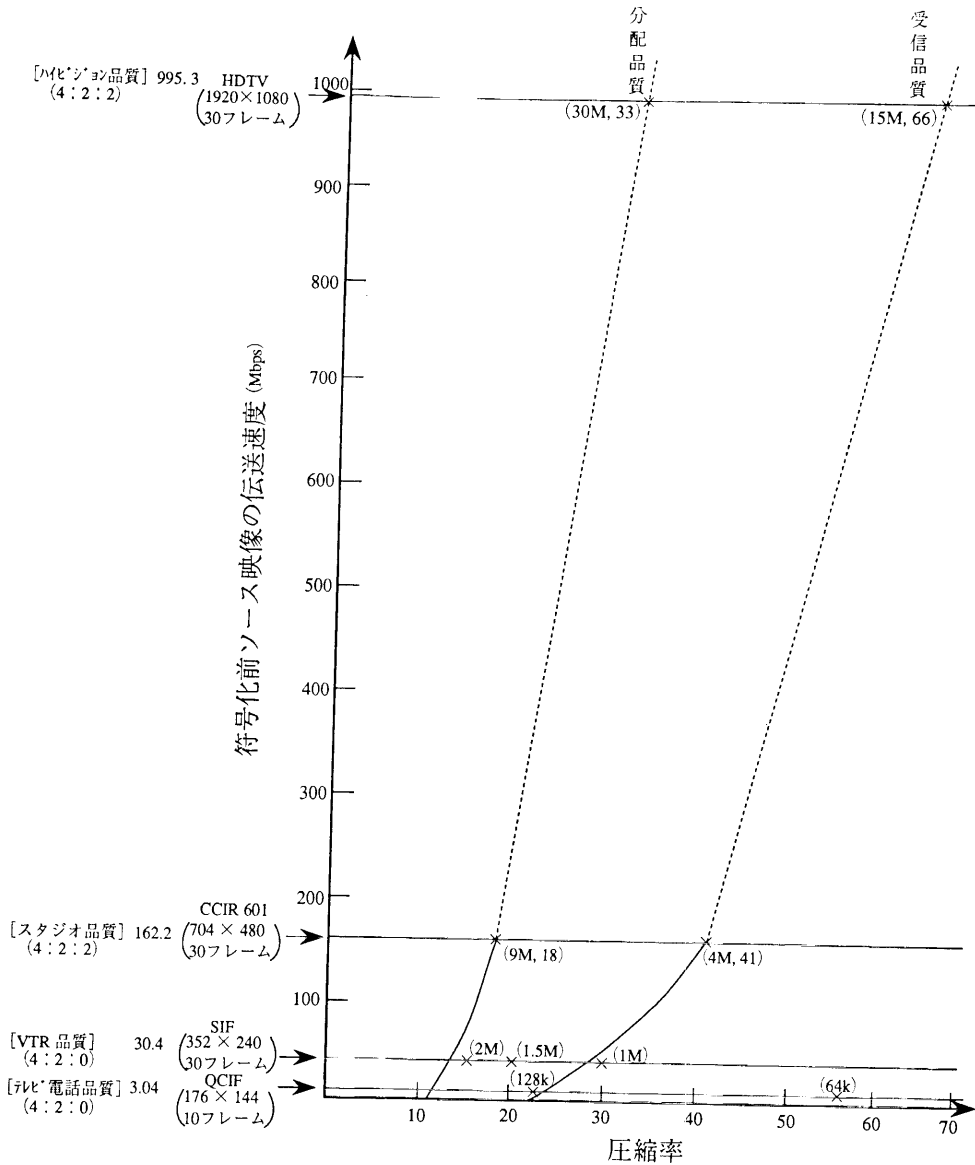


図1 映像品質、伝送速度と各メディア間の相関関係
Fig. 1 Relationship among video quality, bit rate, and media.



注) () 内の数値は、各圧縮率に対応した符号化後伝送速度と圧縮率を示す

図 2 ソース映像伝送速度と圧縮率と画像品質の関係

Fig. 2 Relationship between video source bit rate and video quality.

象画像が高解像度化、高ビットレート化するにつれて目標とする圧縮率を算定する必要があるが、圧縮率は高解像度化するに従って大きくとることができる傾向にある。現在までのところ、本件は QCIF, CIF, CCIR 601 品質、および HDTV の一部の映像を対象にシミュレーション画像の主観評価によって、実験的に確かめているが、今後は HDTV 品質およびそれ以

上の品質について確認する予定である。この傾向について図 2 に示す。

図 1 中で、最初にハイブリッド符号化方式の国際標準となった CCITT H. 261¹⁾、およびこれを受けて第 2 番目の国際標準となった ISO-MPEG 1²⁾の符号化ブロック図を図 3 に示す。ここで、H. 261 と MPEG 1 の本質的な違いはないが、H. 261 がフレーム間符号

表 1 ハイブリッド符号化による所要演算量 (単位: MOPS)
Table 1 Required performance of hybrid video codecs.

対象画像	画像フォーマット	CIF	SIF	CCIR 601	HDTV	UTDV 0	UDTV 1	UDTV 2	UDTV 3	
	符号化方式	H.261	MPEG 1	MPEG 2	MPEG 型	MPEG 型	MPEG 型	MPEG 型	MPEG 型	
対象画像	解像度	352×288	352×240	720×480	1920×1080	1920×1080	3840×2160	5760×3240	7680×4320	
	フレームレート	30	30	30	30	60	60	60	60	
	階調 (ビット)	8	8	8	8	10	10	12	12	
	前処理	100	50	200	1200	2400	9600	21600	38400	
所要演算量	符号化処理	動きベクトル検出	1560	7800	31200	187200	374400	1497600	3369600	5990400
		ループフィルタ	60	—	—	—	—	—	—	—
		DCT	380	320	1280	7680	15360	61440	138240	245760
		逆DCT	380	320	1280	7680	15360	61440	138240	245760
		可変長符号化	120	100	400	2400	4800	19200	43200	76800
		フレームレート制御	20	—	—	—	—	—	—	—
		量子化	10	10	40	240	480	1920	4320	7680
		逆量子化	10	10	40	240	480	1920	4320	7680
		ステップサイズ制御	20	20	80	480	960	3840	8640	15360
		復号化処理	可変長復号化	120	100	400	2400	4800	19200	43200
	逆量子化		10	10	40	240	480	1920	4320	7680
	逆DCT		380	320	1280	7680	15360	61440	138240	245760
	画面再構成		20	20	80	480	960	3840	8640	15360
	ループフィルタ		20	—	—	—	—	—	—	—
	後処理		420	350	1400	8400	16800	67200	151200	268800
	符号化合計		2660	8630	34520	207120	414240	1656960	3728160	6627840
	復号化合計		970	800	3200	19200	38400	153600	345600	614400
総計		3630	9430	37720	226320	452640	1810560	4073760	7242240	

張があることを前提に、実際上実用的な整数画素精度全領域探索と半画素精度の局所的探索を組み合わせたアルゴリズムを仮定することにする。MPEG 2以降のさらなる高解像度化については、単純な空間解像度および時間解像度の増加に合わせた演算量を算出することにしている。

演算量の単位としては MOPS を用いる。まず、ハイブリッド符号化の基本となる 2次元 DCT では、行変換、行列転置、列変換が行われ、1次元、2次元の DCT に分割することとし、1次元 8×8 DCT について各画素に対して 8 回のデータロード、係数ロード、剰算、および格納、7 回の加算、1 回のシフト、1 回の格納から成り、合計して 41 回の操作となる。これに対して 2次元分、30 フレーム/秒、64 画素/ブロック、2376 ブロック (輝度および色差ブロックの全ブロック数) を乗じて以下ようになる⁷⁾。

$$\bullet 41 \times 2 \times 64 \times 2376 \times 30 \div 380 \text{ MOPS}$$

また、最も処理量の多いのは、後述の動きベクトル演算で、16×16 画素のブロックに対する探索領域を 31×31 としたとき、探索回数は 256 回となり、データロードは、あらかじめキャッシュメモリにされているとしても、1 回の絶対値減算と蓄積加算、およびこれらの演算に対して 256 回ごとに比較演算が必要である。したがって 30 フレーム/秒、396 ブロック/フレームに対して MOPS 値を求めると下記となる。こ

こで、比較演算は絶対値減算と積和加算に対し、演算量が少ないため除外している。

$$\bullet 2 \times 30 \times 256 \times 256 \times 396 \div 1560 \text{ MOPS}$$

その他符号化部の処理として、プリフィルタ、ライン変換などの前処理、ループフィルタ、可変長符号化、フレームレート制御、量子化、量子化ステップサイズ制御に分類して検討した。また、復号化部の処理として可変長復号化、逆量子化、逆 DCT、画面再構成、ループフィルタ、ポストフィルタ、ライン変換などの後処理に分類して検討した。この結果を表 1 にまとめる。表 1 の中で、H. 261 と MPEG 1 以降の大きな差は、動きベクトル検出にあるが、同じ 16×16 画素ブロックに対する演算量は、空間領域として約 4 倍、また 2 フレームの前方予測に対し 1 フレームの双方予測フレームを挿入するとして約 1.5 倍の合計約 6 倍になるとして計算している。なお、可変長符号化、復号化処理は、ビットレートにも依存するが、H 261 で 1.5 Mbps としている。さらに、MPEG 2以降は、将来のノンインタレーステレビジョンへの発展を考慮し、簡単のためにフィールド構造等処理は省略して、単純に空間・時間解像度拡張を行っている。

4. システムアーキテクチャ

4.1 全体システム構成

第 2 章で述べた H. 261, MPEG 等の動画像信号の

ハイブリッド符号化アルゴリズムを実行する上での全体システム構成を図4に示す。本図の上半分は、符号化部、下半分は復号化部を表している。映像入力信号は、まずビデオ・メモリに入った後、フリッカノイズ等を除去するとともに、ライン数変換等の前処理を行った後、フレームメモリに記憶される。フレームメモリ内のデータは、前画面の予測メモリ内のデータと比較されるが、動きベクトル演算のような高速処理には、専用のキャッシュメモリの利用が有効である。時間相関がない場合は、DCT 演算によって空間的圧縮処理、量子化処理の後、可変長符号化されて送出される。図4において復号化部は、その逆の処理が行われる。本図にある各ブロックで行う処理は、動きベクトル演算のような単純な差分比較、量子化・逆量子化のような乗除算演算、あるいはDCTのような固定型積和演算等全く異なる性質を持っており、専用処理ブロックに分解し、各部を専用化することでハードウェアの実装効率が向上すると考えられる。特に、第3章で示した所要演算量から考えると、今後のHDTVからUDTVへ進む高解像度、高フレームレートなど高画質化の要求に対応するため、各演算ブロックごとに性能面での拡張性の確保と、VLSI に実装する場合の単位面積、消費電力当たりの処理性能を最大化することが望ましい。

第3章で述べたようにハイブリッド符号化アルゴリズムは、符号化と復号化の二つの部分に分けられる。ここで、一般にDCTと量子化を基本とするアルゴリズムは、演算処理量に著しい非平衡性がある。すなわち、復号化に比べ符号化演算量が極めて大きいし、また画質を大きく左右する。これに対し、復号化は比較的容易であり、高画質化してもVLSI化は容易である。表2に、各演算ブロックの処理の特徴と考えられる実現方式の例を示すが、以下にこれまでVLSI化が困難であると考えられてきた動きベクトル演算部と符号量制御部のアーキテクチャの検討とVLSI実装例について示す。

4.2 動きベクトル演算部

動きベクトル演算は、第3章で述べたように符号化処理の大部分を占める最も重要な部分である。しかしながら、その処理内容は極めて単純で、前画面における画素ブロックの中に、入力画素ブロック (a_{ij}) と最も一致度の高い前画面画素ブロック $(b_{i+k,j+l})$ を所定の探索領域の中から検出するものである。

そこで、この一致度を評価するための評価関数がいくつか提案されているが、ここでは、VLSI化が比較的容易な絶対値差分総和を用いることが最適と判断した。絶対値差分総和は、下式で表すことができる。

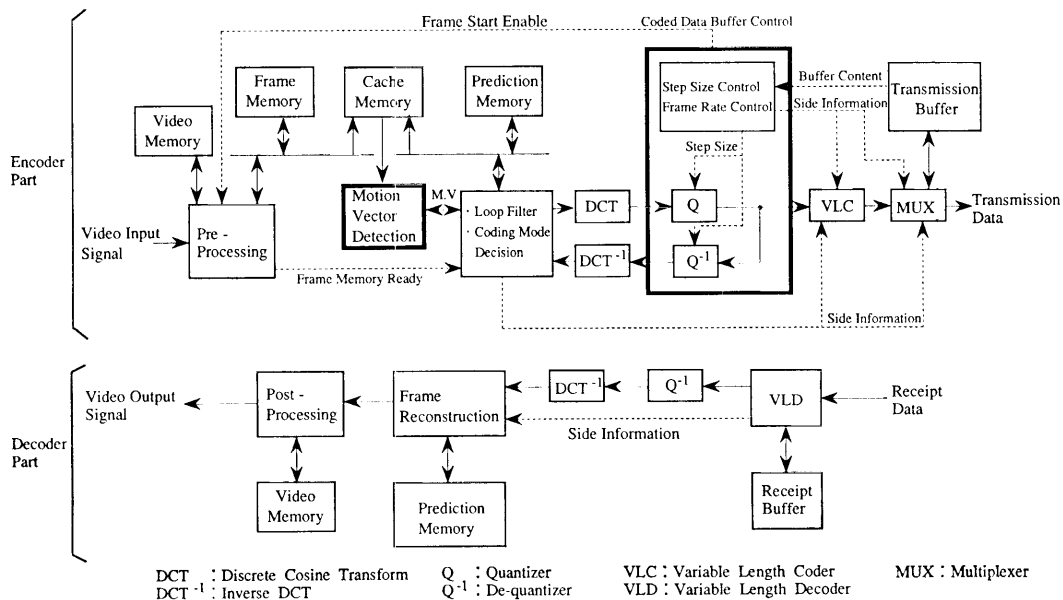


図4 ハイブリッド映像符号・復号器の全体ブロック図
Fig. 4 Total block diagram of hybrid video encoder/decoder.

表 2 ハイブリッド符号器の各演算ブロックの特徴と実現方式の例
Table 2 Special Feature and Implementation Examples of each block in Hybrid Video Encoder.

処理項目	処理の特徴	実現方式の例
前処理	・多段タップフィルタ (定型積和演算)	専用回路 ⁹⁾
動きベクトル検出	・全体の60~80%に相当する大量演算 (定型絶対値差分総和演算)	簡易手法の DSP による実現 ¹⁰⁾
ループフィルタ	・多段タップフィルタ (定型積和演算)	専用回路 ¹¹⁾ または DSP ¹⁰⁾
DCT/逆 DCT	・定数係数との乗算総和 (定型積和の演算)	専用回路 ⁷⁾ または DSP ¹⁰⁾
可変長符号化	・テーブル参照による符号変換 (定型テーブル検索)	専用回路 ⁷⁾ または DSP ¹⁰⁾
フレームレート制御	・バッファ容量に応じたオン/オフ制御	DSP ^{10), 12)}
量子化	・定型除算演算	DSP ^{10), 12)}
逆量子化	・定型乗算演算	DSP ¹⁰⁾
ステップサイズ制御	・バッファ容量に応じた積和演算 (定型積和演算)	DSP ^{10), 12)}

$$\sum_{i=0}^{15} \sum_{j=0}^{15} |a_{i,j} - b_{i+k, j+l}| \rightarrow \text{Min.}$$

$$(0 \leq k \leq 15, 0 \leq l \leq 15) \quad (1)$$

式(1)を用いる場合の入力画素ブロックと前画面画素ブロックの関係を表したものを図5に示す。図5の例では、入力画素ブロックを 16×16 画素、探索領域内

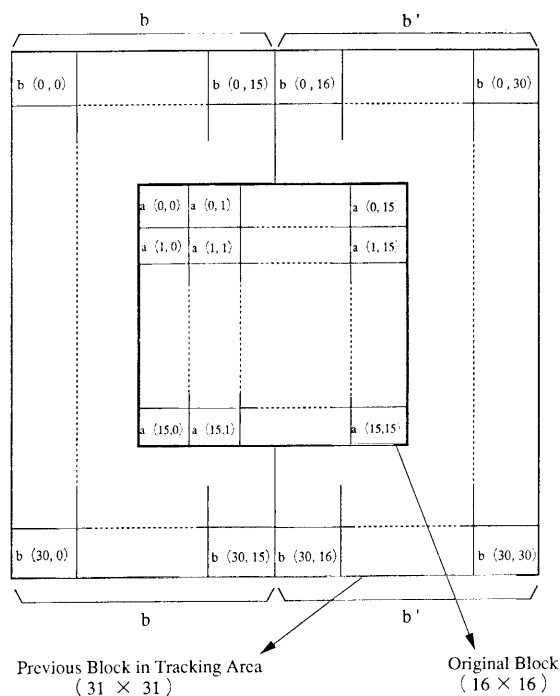


図 5 入力画素ブロックと前画面画素ブロックとの関係
Fig. 5 Relationship between an original block and a previous block in the tracking area.

の比較対象となる前画面画素ブロックを 31×31 画素としている。

このような動きベクトル演算モデルにおいて、膨大な演算量を約 10 分の 1 に削減する手法として 3 ステップサーチ⁸⁾等の簡易化手法が提案されているが、今後の高画質化要求に対応するには全領域探索が不可欠であると判断し、ここでは並列処理による全領域探索方式を採用することとした⁷⁾。図6は、本処理方式に基づくデータフロー図である。ここで、図中の b 、 b' は、おのおの、探索領域の左半面と右半面の画素ブロックである。このようなデータフローを実現する並列処理アーキテクチャは種々考えられるが、VLSI 化を目的とした粒度、並列度、結合度についての最適化を図るために 16 個の PE (Processing Element) による SIMD (Single Instruction-stream, Multiple Data-stream) 型のアーキテクチャを前提とする設計を行うこととした。ここで、各 PE の制御命令は、前述の絶対値差分総和であることから固定の論理回路で実現することとしている。

以上の前提条件の下に設計を行った全体ブロック図を図7に示す。以下、各ブロックの回路の実現方式についての検討結果を示す。

PE 部は、専用命令化されているため固定の論理回路で構成するが、初期の目標性能値として動画像通信規格 CCITT H. 261 で規定されている 1 フレーム当たり 352×288 画素 (396 個の画素ブロック) のフレームについて、1 チップで毎秒 15 フレーム/秒を実現することとした。

このことから、16×16 画素ブロックの処理に約 4,100 クロックを要することから 25 MHz 以上のク

Cycle time	Data sequences	PE 0	PE 1	PE 2	PE 14	PE 15
t	a b b'	$\sum \sum a(i, j) - b(k, l) $	$\sum \sum a(i, j) - b(k, l+1) $	$\sum \sum a(i, j) - b(k, l+2) $	$\sum \sum a(i, j) - b(k, l+14) $	$\sum \sum a(i, j) - b(k, l+15) $
0	$a(0,0), b(0,0)$	$a(0,0) - b(0,0)$				
1	$a(0,1), b(0,1)$	$a(0,1) - b(0,1)$	$a(0,0) - b(0,1)$			
2	$a(0,2), b(0,2)$	$a(0,2) - b(0,2)$	$a(0,1) - b(0,2)$	$a(0,0) - b(0,2)$		
14	$a(0,14), b(0,14)$	$a(0,14) - b(0,14)$	$a(0,13) - b(0,14)$		$a(0,0) - b(0,14)$	
15	$a(0,15), b(0,15)$	$a(0,15) - b(0,15)$	$a(0,14) - b(0,15)$	$a(0,13) - b(0,15)$	$a(0,1) - b(0,15)$	$a(0,0) - b(0,15)$
16+0	$a(1,0), b(1,0), b(0,16)$	$a(1,0) - b(1,0)$	$a(0,15) - b(0,16)$	$a(0,14) - b(0,16)$		$a(0,1) - b(0,16)$
16+1	$a(1,1), b(1,1), b(0,17)$	$a(1,1) - b(1,1)$	$a(1,0) - b(1,1)$	$a(0,15) - b(0,17)$	$a(1,0) - b(1,2)$	
16+15	$a(1,15), b(1,15), b(0,31)$					$a(0,15) - b(0,30)$ $a(0,0) - b(1,15)$
2×16+0	$a(2,0), b(2,0), b(1,16)$	$a(2,0) - b(2,0)$	$a(1,15) - b(1,16)$	$a(1,14) - b(1,16)$		$a(1,1) - b(1,16)$
2×16+1	$a(2,1), b(2,1), b(1,17)$	$a(2,1) - b(2,1)$	$a(2,0) - b(2,1)$	$a(1,15) - b(1,17)$	$a(2,0) - b(2,2)$	
15×16+0	$a(15,0), b(15,0), b(14,16)$	$a(15,0) - b(15,0)$	$a(14,15) - b(14,16)$			
15×16+16	$a(15,15), b(15,15), b(14,31)$	$a(15,15) - b(15,15)$	$a(15,14) - b(15,15)$	$a(15,13) - b(15,15)$		
16×16+0			$a(15,15) - b(15,16)$	$a(15,14) - b(15,16)$		
16×16+1	$b(15,16)$ $b(15,17)$			$a(15,15) - b(15,17)$		
16×16+14	$b(15,30)$					$a(15,15) - b(15,29)$
16×16+15	$b(15,31)$					$a(15,14) - b(15,29)$ $a(15,15) - b(15,30)$

図 6 全領域探索方式に基づく動きベクトル検出データフロー図
Fig. 6 A data flow diagram for a full-search block-matching motion estimator.

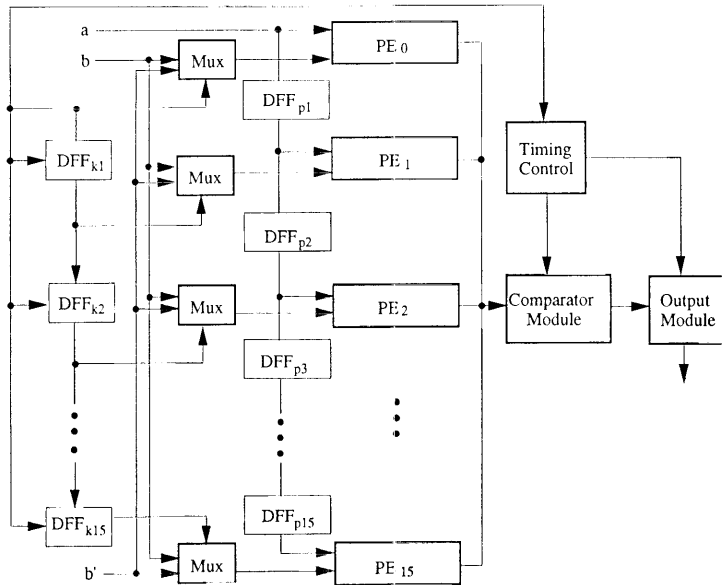


図 7 動きベクトル検出部の全体ブロック図
Fig. 7 Total block diagram of the motion vector detector.

ロックレートが必要となるが、標準的な CMOS プロセスでも容易に実現可能なアーキテクチャが求められる。絶対値差分総和演算は、FA (Full Adder: 全加算器) と ACC (Accumulator: 加算蓄積器) で構成することができるが、ここでのクリティカルパスは、キャリー発生時の FA および ACC での全ビット加

算時の演算遅延時間である。そこで、当部分に各演算フェーズでのパイプライン処理を導入することとした。このような考え方に基づく PE の内部構成を図 8 に示す。

最小値選択部の機能は、16 個の各 PE から 16 回出力される合計 256 の評価関数値から最小値とこれに対応するベクトル (X, Y の値) を求めることである。この実現方式としては、隣接する PE 出力結果を順次比較するトーナメント選択回路方式と、PE 出力結果を順次転送して最終段で比較するシフト転送選択回路方式について検討し、ここでは、図 9 に示すような PE と一組にしたセル化が容

易なシフト転送選択回路方式を採用することとした。図 7 のアーキテクチャに基づく拡張性を実現するには、時間軸の拡張性、すなわちフレームレートの高高速化と、空間軸の拡張性、すなわち探索領域の拡張との二つがある。まず、時間軸の拡張に対しては、例えば CIF フォーマットについて、おのおの奇数ブロック

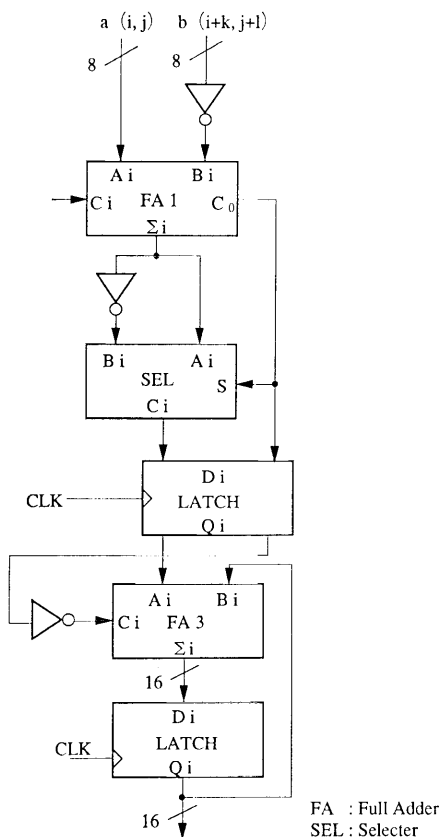


図 8 処理要素 (PE) の内部構成
Fig. 8 Configuration of the processing element.

と偶数ブロックの動きベクトル演算を担当する 2 個のチップを利用して演算することができる。これにより 2 チップでの 30 フレーム/秒のフレームレートを達成することができる。次に、空間軸の拡張に対しては、1 チップで探索可能な 31×31 画素から、 47×47 画素、 63×63 画素へと拡張するに従って、おのおの 4 個または 9 個のチップを用いることで対応可能なようにしている。図 10 にこの多チップ並列処理による探索領域拡張方式について示す。また、絶対値差分総和として求められる評価関数値は、たとえ最小値であってもある規準値以上に大きいと動きベクトルとして採用することは不適切である場合がある。これは、シーン・チェンジやズームング、パニングなどの場合であるが、そこで動き補償を行うかどうかの判定が必要となる。

そこで、この規準値の選び方が重要になってくるが、一例として CCITT SGXV ビデオ符号化専門家

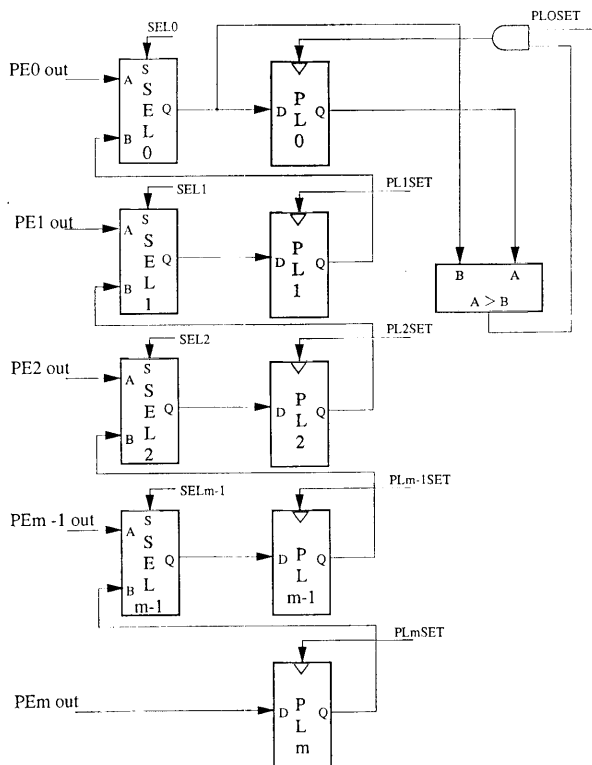


図 9 シフト転送選択回路方式による比較器の内部構成
Fig. 9 Configuration of the comparator based on parallel shift registers.

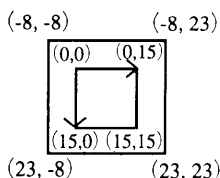
会で議論された RM 8 (Reference Model 8) の例を図 11 に示す。この RM 8 の例にも用いられているように、前画面における同位置との絶対値差分総和、すなわち SLPF 値をこの規準値の算出に利用することが一般的である。この SLPF 値が、動き補償のオン・オフ判定のために算出され、図 11 の判定曲線が用いられる。SLPF 値は、図 10 に示したように 1 個のチップが担当する探索領域のチップ数 4, 16, ... の場合は左上隅に、または、1, 9, 25, ... の場合は中央の値となる。したがって、本方式の場合、中間結果として算出されていることを活用して、外部からの制御信号で左上隅または中央の値を SLPF 値として取り出せるようにしている。この機能は、動き補償を行うか否かの符号化モードを決定する外部判定器にとって極めて有効である。

4.3 符号量制御部

符号量制御は、第 3 章で述べたように演算量はそれ

(a) 1 Chip solution

Tracking Area of the chip : 31×31
 $(-8, -8) - (-8, 23) - (23, 23) - (23, -8)$

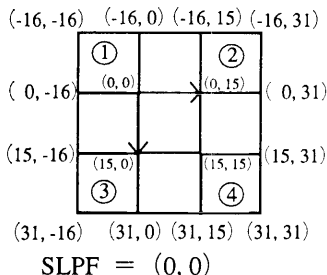


Original Block : 16×16
 $(0, 0) - (0, 15) - (15, 15) - (15, 0)$
 SLPF = $(0, 0)$
 ↓
 Center of the chip's tracking area

(b) 4 Chips solution

Tracking Area of

- Chip① : $(-16, -16) - (-16, 15) - (15, 15) - (15, -16)$
- Chip② : $(-16, 0) - (-16, 31) - (15, 31) - (15, 0)$
- Chip③ : $(0, -16) - (0, 15) - (31, 15) - (31, -16)$
- Chip④ : $(0, 0) - (0, 31) - (31, 31) - (31, 0)$

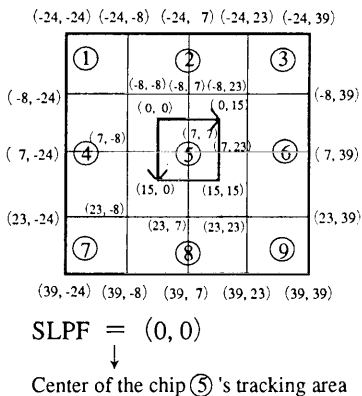


SLPF = $(0, 0)$
 ↓
 Edge of the chip ④'s tracking area

(c) 9 Chips solution

Tracking Area of

- Chip① : $(-24, -24) - (-24, 7) - (7, 7) - (7, -24)$
- Chip② : $(-24, -8) - (-24, 23) - (7, 23) - (7, -8)$
- Chip③ : $(-24, 7) - (-24, 39) - (7, 39) - (7, 7)$
- Chip④ : $(-8, -24) - (-8, 7) - (23, 7) - (23, -24)$
- Chip⑤ : $(-8, -8) - (-8, 23) - (23, 23) - (23, -8)$
- Chip⑥ : $(-8, 7) - (-8, 39) - (23, 39) - (23, 7)$
- Chip⑦ : $(7, -24) - (7, 7) - (39, 7) - (39, -24)$
- Chip⑧ : $(7, -8) - (7, 23) - (39, 23) - (39, -8)$
- Chip⑨ : $(7, 7) - (7, 39) - (39, 39) - (39, 7)$



SLPF = $(0, 0)$
 ↓
 Center of the chip ⑤'s tracking area

図 10 探索領域の拡張方式
 Fig. 10 Extension method of tracking area.

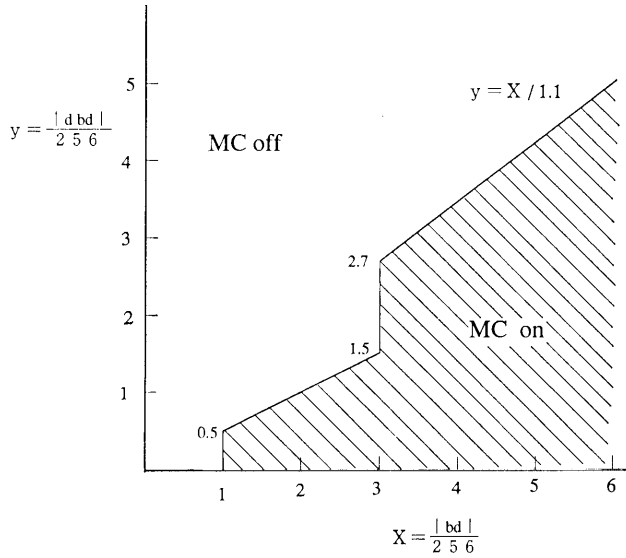
ほど多くないが、ハイブリッド符号化の各アルゴリズム要素ブロックの中で、画質を左右する最も複雑でかつ制御の自由度を大きくとるべきブロックである。ここに汎用 DSP を用いることも考えられるが、今後の高画質化・高速化に対応するためには処理性能上の問題がある。そこで、本研究においては、汎用 DSP の柔軟性と専用回路のもつ実装効率・高速性の両者の長

所を兼ね備えた統合型符号量制御演算アーキテクチャを新たに考案し、VLSI 実装を併せて行った¹³⁾。

符号量制御部の機能は、量子化ステップ (QSTEP) 制御とフレームレート制御、すなわち、QSTEP 制御だけでは符号量発生に見合うだけの送信レートに抑え切れない場合に駒落しを行うことの二つに集約することができる。

この二つの機能を実現するためのアーキテクチャ設計を行うために以下の基本方針を設定した。

- ① 1クロックで1演算を実行可能なこと。
- ② 最低限必要な ALU (Arithmetic Logic Unit) を備えること。
- ③ 同一アーキテクチャで他標準への対応が可能なこと。



bd : block difference , d bd : displaced block difference
 SLPF : Same Location value with Value on the Previous Frame
 $SLPF = \frac{|bd|}{2^5 \cdot 6}$

図 11 RM8 における MC/no MC 判定規準
 Fig. 11 MC/no MC decision criteria in RM8.

以上の基本方針に則り、2個の乗算器を核とする ALU と設計段階でプログラマブルな状態遷移制御器およびこれらの接続する統合化バスとで構成することとした。

図 12 に符号量制御部の全体ブロック図を示す。2個の乗算器のうち、ALU 1 は、フィードバックループ用の逆量子化演算に用い、他方の ALU 2 は、QSTEP 演算、フレームスキップ演算を行うために用いる。

しきい値制御部は、DCT 後の各変換係数のしきい値制御を行い、その結果を量子化後の丸めおよびオーバフロー制御に利用する。

オフセット制御部は、DCT 後の変換係数を入力とするオフセット演算を行い量子化演算のための前処理を行う。

表現形式判定部は、ALU 1 のローカルループ用の逆量子化演算結果に対する規格

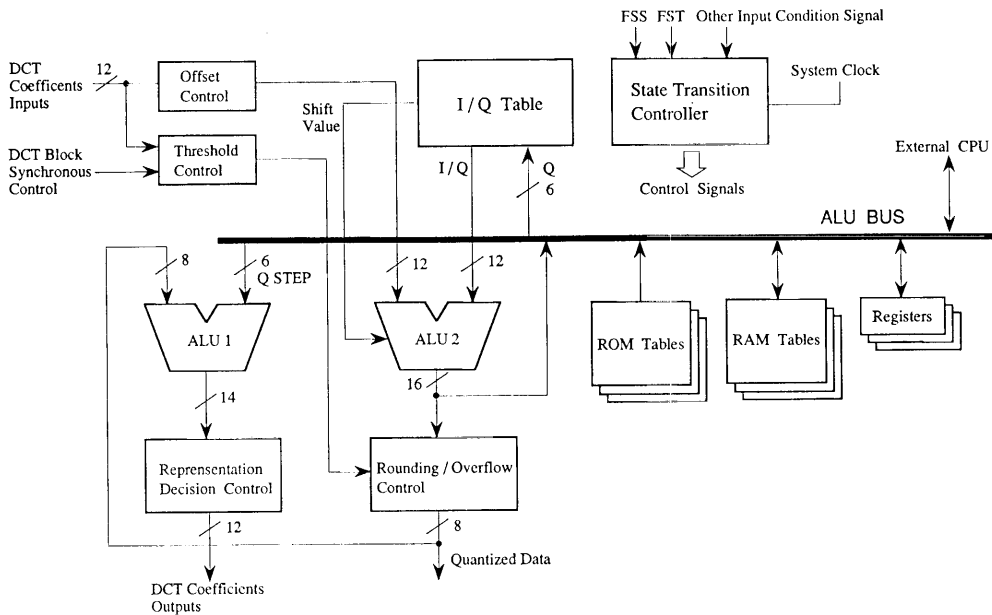


図 12 符号量制御部の全体ブロック図
 Fig. 12 Total block diagram of coded data buffer control.

化されたフォーマットへの統一を行うための部分である。

丸めおよびオーバーフロー制御部は、ALU2による量子化演算結果が規定値を越えることを防止するとともに、所定桁数に丸めるための処理を行う。

1/Q テーブル ROM は、量子化するかわち、QSTEP値による除算を実行する際に高速処理化するためあらかじめ逆数テーブルをROMに格納しておき、これとの乗算を行うことによって実効的に除算を行うためのものである。

状態遷移制御部は、1クロックで1命令を実行するものであり、回路構成に対する柔軟性を設計段階で実現しており、他標準への対応が状態遷移制御部のプログラム修正で可能である。また実装後の占有面積を極小化し、専用回路並みの高速性能を実現している。

ALU1, ALU2, 1/Q テーブルROM, 状態遷移制御部, レジスタ群, ROM テーブル群, RAM テーブル群は、統合ALUバスによって接続されており、起動後1クロックごとに状態遷移とともに制御信号を出力する状態遷移制御器によって全体の動作が規定されるようになっている。

次に、状態遷移制御部による基本動作について説明する。本基本動作によって本質的に重要な入出力信号を以下に示す。

(1) 入力信号

FST：ビデオ入出力系から送られる29.97Hzのビデオ同期信号

FSS：フレーム単位の符号化同期信号（フレームスキップ時は発生しない）

MBS：FSS内にCIFフォーマット時は396回出力されるマクロブロック単位の符号化同期信号

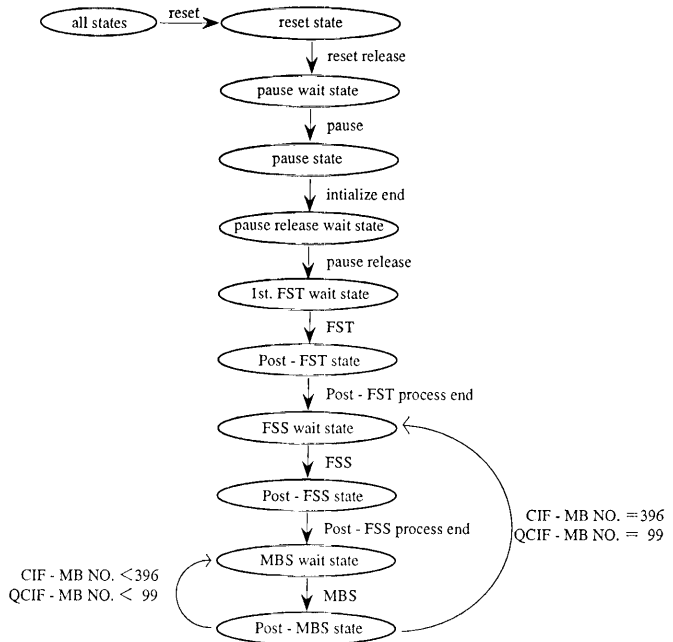
(2) 出力信号

FSEN：フレーム単位の符号化起動信号

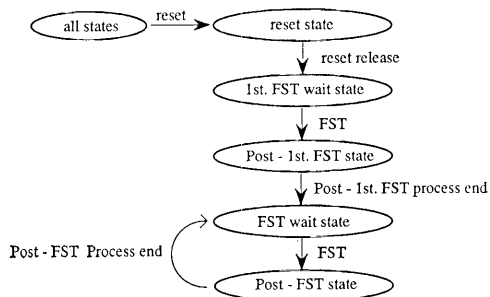
SCRQ：シーンチェンジ要求信号

FFIX：強制固定符号化指令信号

以上の入出力信号をもとに最も重要な状態遷移制御部のアーキテクチャ設計を行った。状態遷移制御部は、現状態と入力信号によって1クロックごとに次状態と制御出力信号が決定される有限オートマトンモデルに基づいて設計されている。ここでは、簡単のためにFST起動状態制御部とFSS起動状態遷移制御部の二つの部分に分けて実現することにした。図13とこれらの状態遷移制御部における基本状態遷移図を示す。表3に図13に示した各状態における処理内容に



(a) Basic State Transition based on FSS



(b) Basic State Transition based on FST

図13 符号量制御部の基本状態遷移図

Fig. 13 Basic state transition of coded data buffer control.

表 3 各状態における処理内容
Table 3 Process of each state.

状 態	処 理 内 容
Pause State	・システム制御用 CPU のソフトウェアによる初期化
Post-FST State	・TdSTP, MrD, TRFS の算出
Post-FSS State	・フレーム起動初期化, CIF/QCIF のチェック
Post-MBS State	・最初の MB の時のみ TdSTP の上限チェックと TdSTP をもとにした TRFS の算出 ・QSTEP の算出 ・SCRQ, FFIX 出力有無の判定 ・量子化/逆量子化演算
Post-1 st. FST State	・FURQ フラグオン ・TR インクリメント ・FSEN オン
Post-FST State	・FST 立上りで FSEN オンならオフ ・TR インクリメント ・FURQ フラグオンなら SCRQ オン, FURQ フラグオフ ・バッファ容量規定値以下なら FSEN オン

注) TdSTP: フレームスキップ数

MrD: 現マクロブロック符号量と標準マクロブロック符号量の差

TRFS: 次に FSEN オンすべき TR

TR: フレームのシーケンス番号

ついて示す。

ここで、符号量制御にとって最も重要な時間解像度を決定するフレームスキップ・パラメータである Td STP(Time Difference Step Size) と、空間解像度を決定する量子化幅パラメータである QSTEP (Quantization Step Size) の算出については、種々のアルゴリズムが考えられるが、ここでは、外部設定される画像に割り当てられた伝送速度、現在の送信バッファ容量、最大・最小フレームレート、画質優先・動動き優先・中間等のパラメータから TdSTP 算出用テーブル、および QSTEP 算出用テーブルを参照して計算されるようになっている。また、各テーブルは、数種の ROM およびダウンロード可能な RAM を用意し柔軟性を持たせている。

4.4 その他のブロックのアーキテクチャ

図 5 中に示したほかの部分については、DCT および DCT⁻¹ 部分は、あらかじめ規定された演算順序を実行するだけであるため高速化のために専用回路化することが妥当である¹⁴⁾。

VLC および VLD 部分は、同様に処理内容に適した専用回路化が考えられる^{15),16)}。また、前処理部⁷⁾、後処理部は単純なフィルタ処理であるため、専用の積和演算回路によることが適切である。さらに、符号化システム管理部¹¹⁾、復号化システム管理部¹⁷⁾は、基本的にはタイミング管理のための固定の順序回路によって構成することが適切である。

5. VLSI 実装

4章で述べたように、H. 261, MPEG 等の動画像符号化処理には、比較的単純な演算をパイプライン的に、またはシストリックアレイ的に高速に行う部分と、種々の状態に応じた符号化の制御を行う部分がある。処理内容に応じて VLSI の設計方法も異なってくる。本章では、前者の例として動きベクトル検出用チップ、後者の例として符号量制御用チップをとりあげ VLSI 実装について述べる。

5.1 動きベクトル検出 LSI

動きベクトル検出用チップは、16 個の PE を内蔵し、これらの並列パイプライン動作により動きベクトルを検出する。本チップの主な特徴を表 4 に示す。データブロックは 16 画素×16 画素、探索領域は 31 画素×31 画素で、動きベクトルを X 方向、Y 方向とも -8, +7 の範囲で検出する。この場合、1 マクロブロックに対して動きベクトルの候補は、16×16 通りあり、1 つの動きベクトルに対して、16×16 の差分演算を実行する。CIF 画像で 30 フレーム/秒の符号化を行う場合、マクロブロック当たりの演算時間は約 80 μs である。

本チップは、16 個の PE を内蔵しているため、1 PE 当たり 16×16×16=4096 演算となる。したがって、1 演算当たり約 20 ns すなわち 50 MHz である。本チップの動作周波数を 30 MHz と定め、LSI 2 個用い

表 4 動きベクトル検出用チップの特徴

Table 4 Features of the Motion Vector Detection Chip.

項目	特徴
ベクトル検出	絶対値差分総和判定による全検索
データブロック	16×16 画素
探索範囲	32×32 画素
演算出力	動きベクトル値 V_x, V_y
	動き補償後残差
	同一座標での残差

ることにより、CIF、30 フレーム/秒を処理可能とした。演算の結果、動きベクトル値 V_x, V_y 、動き補償後の残差値 (SLPF 値)、同一座標での残差値を一つの端子からシリアルに出力する。残差の精度は、16 ビットと 8 ビットで切り替え可能とした。

演算要素 PE は、同一の回路を 16 個並べて規則的なパイプライン構成で実現できる。本チップのマスク写真を図 14 に、諸元を表 5 に示す。全ゲート数は、20k ゲートで A12 層配線の 1 μm CMOS スタンダードセルにより設計した。面積効率を考慮し、各 PE はポ

表 5 動きベクトル検出用チップの諸元

Table 5 Specification of the Motion Vector Detection Chip.

動作周波数	30 MHz
ゲート数	20kゲート
プロセス	1 μm CMOS スタンダードセル 2 層配線
チップサイズ	6.7 mm×6.7 mm
パッケージ	84 ピン PLCC

リセル方式によりレイアウトした。

5.2 符号量制御用チップ

本チップの主な機能は、符号化のための量子化ステップとフレームスキップの最適制御である。特徴を表 6 に示す。本チップには、CPU インタフェースを設け、対象画像に応じた動作モードを外部から設定可能とした。

内部は、図 12 に示したように、乗算器を含む 2 系統のデータバス、データバスを制御する状態遷移回路、RAM テーブル、ROM テーブルなどから構成される。図 13 の流れに従い、マクロブロックごとに最適量子化ステップを算出し、その結果により量子化が実

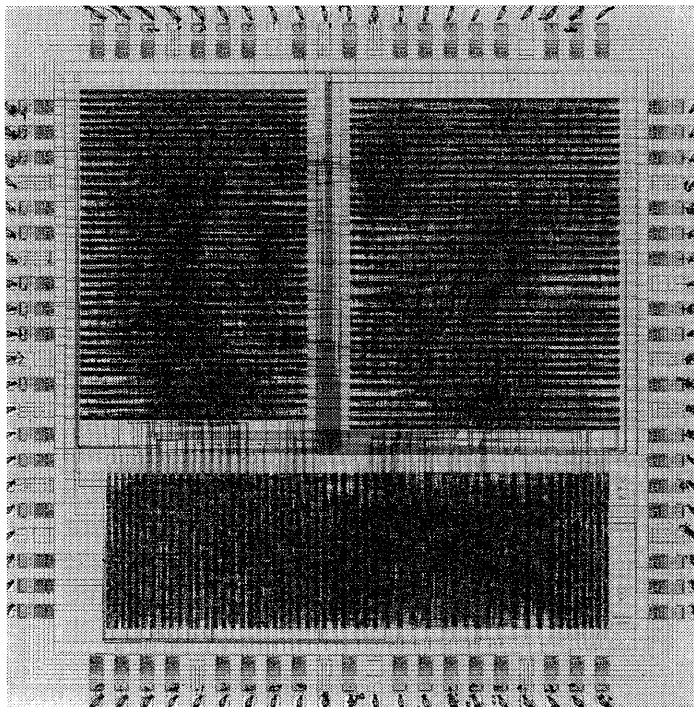


図 14 動きベクトル検出用チップのマスク写真

Fig. 14 Photograph of the Motion Vector Detection Chip.

表 6 符号量制御チップの特徴
Table 6 Features of the Buffer Control Chip.

項目	特徴
主な機能	量子化ステップとフレームスキップの最適制御 (1~30 フレーム/秒の可変フレームレート)
	各種動作モードの外部設定 (解像度優先/動き優先制御など)
	ジグザグ変換
演算方法	状態遷移によるデータパス制御
	6段パイプラインによる高速演算

表 7 符号量制御チップの諸元
Table 7 Specification of the Buffer Control Chip.

動作周波数	15 MHz (30 フレーム/秒動作時)
ゲート数	26kゲート
内蔵 RAM	14kビット
プロセス	1 μ m CMOS スタンダードセル2層配線
チップサイズ	9.8 mm \times 10.2 mm
パッケージ	136ピン QFP

行される。量子化は、2系統のデータパスを6段パイプラインで動作させることにより、各データごとに高速に連続して行われる。本チップのマスク写真を図15に、諸元を表7に示す。設計に際しては、データパス

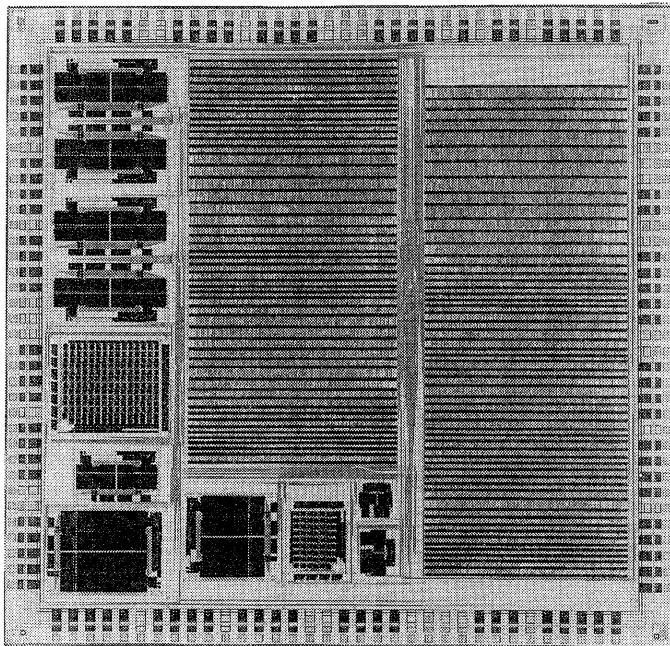


図 15 符号量制御チップのマスク写真
Fig. 15 Photograph of the Buffer Control Chip.

コンパイラ、状態遷移コンパイラ、RAM および ROM コンパイラを用いた。図13の状態遷移は、約300ステップで記述され、コンパイルの結果、約3000ゲートとなった。全ゲート数は、26kゲートで、A12層配線の1 μ m CMOS スタンダードセルにより実現した。

6. おわりに

以上、高解像度化がますます進展する画像符号化分野において、主流となっているハイブリッド符号化に焦点を当てて、各アルゴリズム演算要素に要求される処理性能を算出した。また、これに対応するために各アルゴリズム要素に最適な専用型マルチメディア符号化処理アーキテクチャを考案し、VLSI 実装して評価した。

評価の結果、最もクリティカルな動きベクトル検出部について1.0 μ m CMOS 技術によって動作クロック約30 MHzで最大フレームレート30フレーム/秒、解像度352 \times 288 (CIF) を達成した。これは、ほぼ1560 MOPS に相当する処理性能である。また、同等プロセスにおいて符号量制御プロセッサを約15 MHzで最大フレームレート30フレーム/秒、解像度352 \times 288 (CIF) を達成した。これは、ほぼ60 MOPS に相当する処理性能である。

これらの値は、DSP などの汎用プロセッサアーキテクチャと比較して同等のクロックレートでの性能比が50~100倍高いことを示している。この結果、本論文で述べた専用型アーキテクチャによるマルチメディア符号化システムを構成することが有効であり、今後0.5~0.35 μ m CMOS プロセスが一般化し、システムクロックが500 MHz程度になることを考えるとHDTV~UDTV1程度の高解像度に対してはリアルタイム符号化処理の実現見通しが得られたといえる。

今後は、さらに、高効率な並列処理アーキテクチャを検討し、さらなる超高解像度符号化処理へ向けての試作と評価を行う予定である。

参考文献

- 1) Video Codec for Audio Visual Services at P \times 64 kb/s, CCITT

- Recommendation H. 261 (1990).
- 2) MPEG, CD 11172: Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up To About 1.5 Mbps. (1991).
 - 3) Tamitani, I. et al.: A Real-time Video Signal Processor Suitable for Motion Picture Coding Application, *IEEE Trans. on Circuits and Systems*, Vol. 36, No. 10, pp. 1259-1266 (1989).
 - 4) 小山他: モデルに基づく分析合成符号化における顔の動き量の推定, 電子情報通信学会秋季全国大会, pp. D-1-231-D-1-232 (1988).
 - 5) Jacquin, A. E.: Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations, *IEEE Trans. Image Process.*, Vol. 1, No. 1, pp. 18-30 (Jan. 1992).
 - 6) 藤原: デジタル映像技術の動向とデジタル記録技術, テレビジョン学会誌, Vol. 46, No. 7, pp. 801-806 (1993).
 - 7) Fujiwara, H. et al.: An All-ASIC Implementation of a Low Bit-Rate Video Codec, *IEEE Transactions on Circuit and Systems*, Vol. 2, No. 2, pp. 123-134 (1992).
 - 8) Koga, T. et al.: Motion-Compensated Interframe Coding for Video Conferencing, NTC 81, Proc., pp. G 5.3.1-G 5.3.5 (Dec. 1981).
 - 9) 土屋, 西村: NTSC/CIF 相互変換 LSI の構成, NTT R & D, Vol. 40, No. 8, pp. 1043-1050 (1991).
 - 10) Yamauchi, H. et al.: Architecture and Implementation of a Highly Parallel Single Chip Video DSP, *IEEE Transactions on Circuit and Systems*, Vol. 2, pp. 207-220 (1992).
 - 11) Tayama, M. et al.: VLSI Implementation of System Manager Chip for p×64 kbps Video Encoder, *The Third International Workshop on 64 kbit/s Coding of Moving Video*, Rotterdam, the Netherlands (September 4-6, 1990).
 - 12) Endo, Y. et al.: p×64 Standard Video Codec with LSI Technology, ITU forum '91, Technical Symposium Integration, Interoperation, and Interconnection: the way to Global Services, Vol. 1, Part 2, pp. 303-307, Geneva (October 10-15, 1991).
 - 13) Uwabu, H. et al.: VLSI Implementation of A Buffer, Universal Quantizer and Frame Rate Control, SPIE Visual Communications and Image Processing '91: *Visual Communication*, Vol. 1605, pp. 928-937 (November 10-13, 1991).
 - 14) Maruyama, M. et al.: VLSI Architecture and Implementation of a Multi-Function Forward-Inverse Discrete Cosine Transform Processor, *SPIE Visual Communication and Image Processing 1990*, Vol. 1360, pp. 410-417 (October 1-4, 1990).
 - 15) Fujiwara, H. et al.: VLSI Implementation of a Flexible Variable-Length Decoding Processor for Real-Time Video, *The Third International Workshop on 64 kbit/s Coding of Moving Video*, Rotterdam, the Netherlands (September 4-6, 1990).
 - 16) Saito, R. et al.: VLSI Implementation on a Variable-Length Coding Processor for Real-Time Video, *IEEE Workshop on Visual Signal Processing and Communication*, pp. 87-90, Hsinchu, Taiwan (June 6-7, 1991).
 - 17) Fukuchi, H., Shomura, K. and Fujiwara, H.: VLSI Implementation of Decoding System Manager Chip for Motion Picture Decoder, *IEEE Workshop on Visual Signal Processing and Communication*, pp. 173-176, Hsinchu, Taiwan (June 6-7, 1991).

(平成 5 年 9 月 20 日受付)

(平成 6 年 3 月 17 日採録)



藤原 洋 (正会員)

昭和 29 年生。昭和 52 年京都大学理学部卒業。コンピュータメーカー、エンジニアリング会社を経て、昭和 60 年(株)アスキー入社、昭和 62 年(株)グラフィックス・コミュニケーション・テクノロジーズ出向、平成 3 年ジー・シー・テクノロジー(株)出向を経て、平成 5 年 4 月より(株)グラフィックス・コミュニケーション・ラボラトリー出向。現在、常務取締役研究開発本部長、およびジー・シー・テクノロジー(株)代表取締役社長(兼務)。コンピュータ通信システム、画像符号化技術の研究開発に従事。IEEE、電子情報通信学会、テレビジョン学会、システム制御学会、画像電子学会各会員。主な著書「画像符号化技術」(訳書、オーム社刊)。

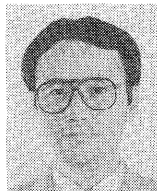


岡田 豊

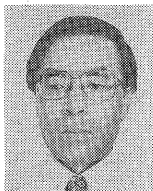
昭和 26 年生。昭和 50 年東京大学電気工学科卒業。(株)日立製作所入社、同社中央研究所にて、アナログ・デジタル混在 LSI、信号処理 LSI の研究に従事。昭和 60~61 年米国アリゾナ大学客員研究員。平成 3 年よりジー・シー・テクノロジー(株)出向、平成 5 年 4 月より(株)グラフィックス・コミュニケーション・ラボラトリー出向、現在に至る。画像コーデック用 LSI の研究、開発に従事。電子情報通信学会会員。

**小林 孝之**

昭和 32 年生。昭和 54 年埼玉大学
理工学部卒業。エレクトロニクス
メーカーを経て、平成 5 年 4 月(株)
アスキー入社、(株)グラフィック
ス・コミュニケーション・ラボラト
リーズ出向、現在に至る。画像符号化技術の研究、お
よび LSI、端末機器開発に従事。テレビジョン学会、
電子情報通信学会各会員。

**上符 浩男**

昭和 28 年生。昭和 51 年九州芸術
工科大学音響設計学科卒業。京セラ
(株)入社。昭和 62 年より(株)グラ
フィックス・コミュニケーション・
テクノロジー出向、画像通信の研究
開発に従事。現在、京セラ(株)中央研究所勤務。電子
情報通信学会会員。

**丸山 優徳**

昭和 22 年生。昭和 51 年東北大学
大学院修了。(株)日立製作所中央研
究所入社。映像電子管用電子ビーム
系の研究開発に従事。昭和 62 年 6
月より(株)グラフィックス・コミュ
ニケーション・テクノロジーに出向し、画像通信用
デバイスの研究開発に従事。平成 3 年 8 月より(株)日
立製作所中央研究所にて通信ネットワークの研究に従
事。現在、通信ネットワーク研究センタ、センタ長、
IEEE、電子情報通信学会、テレビジョン学会、SPIE
各会員。