



Title	ビデオ画像の生成と圧縮およびATM音声伝送のためのVLSI化設計に関する研究
Author(s)	正城, 敏博
Citation	大阪大学, 1997, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3129097
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

ビデオ画像の生成と圧縮および
ATM 音声伝送のための
VLSI 化設計に関する研究

1997年

正 城 敏 博

ビデオ画像の生成と圧縮および
ATM 音声伝送のための
VLSI 化設計に関する研究

1997年

正 城 敏 博

内容梗概

本論文は、著者が平成 5 年から平成 8 年にかけて、大阪大学大学院工学研究科博士前期課程ならびに博士後期課程に在学中に行った、ビデオ画像生成用多重スレッドプロセッサの VLSI 化設計、HDTV レベル MPEG2 デコーダの VLSI 化設計、およびマルチメディア ATM ネットワークにおける音声伝送のための VLSI 化設計に関する研究成果をまとめたものである。

近年、広帯域 ISDN(B-ISDN) 上に展開される ATM 通信技術の発展により、従来の文字情報やデータだけではなく、画像や音声をも統合的に実時間処理するマルチメディア情報通信の有効な手段として ATM ネットワークを利用する動きが盛んになってきている。計算機のデータ伝送等、情報伝達系を中心に発展してきた ATM ネットワークは、CATV (cable television) 等のデジタル有線放送系や既存の電話網である音声用狭帯域ネットワークとは個別のネットワークとして活用されてきた。今後、より実用的な通信インフラストラクチャの構築のためには、高速データ、画像、音声などのマルチメディア情報を一元的に扱うことのできる新たな統合的ネットワーク手法を確立する必要がある。

マルチメディア ATM ネットワークを構築する有効なアプローチとして、新たに必要となる諸機能を既に実用段階に入っているデータ伝送系の ATM ネットワークに付加するという手法が考えられる。マルチメディア情報の実時間処理を行うために実現すべき要素技術として、送信側における画像生成、画像圧縮、音声符号化等の技術、伝送路における多種メディアの高品質/高信頼/高効率な処理技術、受信側における画像圧縮符号や音声符号に対する実時間復号化技術、さらには、これらの処理を行うための VLSI 設計技術が挙げられる。

本研究では、これらの要素技術のうち、画像や音声の実時間処理を行うために重要な、(i) ビデオ画像生成用途に適した多重スレッドプロセッサの VLSI 化設計、(ii) ビデオ動画像圧縮符号の国際標準 MPEG2 アルゴリズムの HDTV レベルデコーダの VLSI 化設計、および (iii) 音声伝送に適したマルチメディア ATM ネットワークの構成と音声用 ATM セルの処理を行うルートノードの VLSI 化設計、について考察する。

本論文は、全 6 章から構成される。第 1 章に序論を述べ、第 2 章ではビデオ画像生成用多重スレッドプロセッサの VLSI 化設計について考察し、第 3 章では HDTV レベル MPEG2 デコーダの VLSI 化設計について述べる。第 4 章では音声伝送を効率的にする VCI 共有セルを

用いたマルチメディア ATM ネットワーク手法について考察し、第 5 章では、第 4 章で考案したネットワーク手法を構築するために必要なリルートノードの VLSI 化設計について述べる。第 6 章に以上の章の結論を述べる。

第 1 章では、従来の ATM ネットワークにおけるマルチメディア情報の取り扱いや、ビデオ画像生成、ビデオ動画像圧縮手法について述べ、本研究の背景と目的を明らかにするとともに研究内容と成果について概説する。

第 2 章では、高品位なビデオ画像の高速生成の要求に対して、多重スレッド方式の特徴について述べ、さらに多重スレッドプロセッサのアーキテクチャと VLSI 化設計手法に関して考察する。高位合成システムを活用することによって、高機能な制御機構を有する多重スレッドプロセッサの実装がどのように実現し、かつ工程期間がどれ位短縮できるか、について主眼を置いて述べる。

第 3 章では、ビデオ動画像圧縮符号化アルゴリズムの国際標準 MPEG2 の規定のうち、高品位画像を扱ういわゆる MP@HL(Main Profile at High Level) 画像を実時間で復号化するための 1 チップ VLSI 化設計について記述する。従来の DSP コアと専用演算器を用いた構成ではなく、MPEG2 の復号化の各処理過程を実行するために新たに考案した専用アーキテクチャを活用することによって、1 チップデコーダが実現可能であることを示す。

第 4 章では、ATM ネットワークと従来の音声セル化手法に注目し、音声伝送効率の改善を目的としたマルチメディア ATM ネットワーク構成について考察する。ネットワーク内で動的に音声用セルを多重化したり、あるいは組み替えたりすることによって、伝送効率を高めることを目的とする。この目的を実現するために、新たに VCI 共有セル化手法を考案し、このセル化手法を通じて達成される伝送効率を評価する。VCI 共有セルは、標準 ATM セルとの親和性を損なうことなく、VCI 共有セルと標準 ATM セルの区別をネットワーク内の一部で認識するだけで音声伝送効率が向上できることを示す。

第 5 章では、第 4 章で考案したマルチメディア ATM ネットワーク内において、VCI 共有セルの多重化や組み替え処理、ならびに標準 ATM セルのヘッダ変換処理を可能とするリルートノードの VLSI 化設計に関して考察する。リルートノードは VCI 共有セルに関する固有な処理を単純な機構のハードウェアを用いて実行し、他のネットワーク要素の変更なしに音声用 ATM セルの効率的な伝送を実現する。

第 6 章では、本研究で得られた成果を要約し、今後に残された課題について述べ、結論とする。

関連発表論文

I. 学会論文誌発表論文

- (1) T. Masaki, Y. Nakatani, T. Onoye, and K. Murakami: "Voice and telephony over ATM for multimedia network using shared VCI cell," to appear in *Journal of Circuits, Systems and Computers*.
- (2) T. Onoye, T. Masaki, Y. Morimoto, Y. Sato, I. Shirakawa, and K. Matsumura: "Single chip implementation of MPEG2 decoder for HDTV level pictures," *IEICE Transactions on Fundamentals*, vol. E79-A, no. 3, pp. 330–338 (March 1996).
- (3) T. Masaki, Y. Morimoto, T. Onoye, and I. Shirakawa: "VLSI implementation of inverse discrete cosine transformer and motion compensator for MPEG2 HDTV video decoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 5, pp. 387–395 (October 1995).
- (4) T. Onoye, T. Masaki, I. Shirakawa, H. Hirata, K. Kimura, S. Asahara, and T. Sagishima: "High-level synthesis of a multithreaded processor for image generation," *IEICE Transactions on Fundamentals*, vol. E78-A, no. 3, pp. 322–330 (March 1995).

II. 研究会等発表論文 (査読付)

- (1) T. Onoye, T. Masaki, Y. Morimoto, Y. Sato, and I. Shirakawa: "HDTV level MPEG2 video decoder VLSI," in *Proc. TENCON on Microelectronics and VLSI*, pp. 468–471 (November 1995).
- (2) T. Masaki, Y. Morimoto, Y. Sato, T. Onoye, and I. Shirakawa: "Single-chip implementation of MPEG2 decoder dedicated to MP@HL," in *Proc. Int'l Conference on VLSI and CAD*, pp. 25–28 (October 1995).

- (3) T. Masaki, Y. Morimoto, Y. Sato, T. Onoye, and I. Shirakawa: "Single-chip VLSI decoder for MPEG2 MP@HL," in *Proc. Synthesis and System Integration of Mixed Technologies*, pp. 211–218 (August 1995).
- (4) T. Masaki, Y. Morimoto, T. Onoye, and I. Shirakawa: "Specific functional macrocells for MPEG2 single-chip HDTV decoder," in *Proc. Joint Technical Conference on Circuits/Systems, Computers and Communications*, pp. 499–502 (July 1995).
- (5) 森本康夫, 正城敏博, 尾上孝雄, 白川功: "HDTV レベル MPEG2 デコーダの VLSI 化設計," 電子情報通信学会回路とシステム軽井沢ワークショップ論文集, pp. 203–208 (1995 年 4 月).
- (6) T. Onoye, Y. Morimoto, T. Masaki, and I. Shirakawa: "Design of inverse DCT unit and motion compensator for MPEG2 HDTV decoding," in *Proc. Asia-Pacific Conference on Circuits and Systems*, pp. 608–613 (December 1994).
- (7) T. Onoye, T. Masaki, S. Asahara, T. Sagishima, I. Shirakawa, S. Tsukiyama, and S. Shinoda: "Design of multithreaded processor dedicated to image generation: high-level synthesis," in *Proc. Joint Technical Conference on Circuits/Systems, Computers and Communications*, pp. 689–694 (July 1994).
- (8) T. Onoye, T. Masaki, H. Hirata, K. Kimura, S. Asahara, T. Sagishima, I. Shirakawa, S. Tsukiyama, and S. Shinoda: "High-level synthesis of a multithreaded processor for image generation: design and simulation," in *Proc. European Simulation Multiconference*, pp. 948–953 (June 1994).
- (9) T. Onoye, T. Masaki, H. Hirata, K. Kimura, S. Asahara, T. Sagishima, I. Shirakawa, S. Tsukiyama, and S. Shinoda: "High-level synthesis of multithreaded processor based image generator," in *Proc. IEEE Int'l Symposium on Industrial Electronics*, pp. 47–52 (May 1994).
- (10) 正城敏博, 尾上孝雄, 平田博章, 木村浩三, 浅原重夫, 鷺島敬之, 白川功: "画像生成用多重スレッドプロセッサの高位合成手法による設計," 電子情報通信学会回路とシステム軽井沢ワークショップ論文集, pp. 61–66 (1994 年 4 月).

III. その他研究会等発表論文

- (1) 正城 敏博, 中谷 泰寛, 尾上 孝雄, 村上 孝三, 白川 功: “VCI 共有セルを用いた ATM 通信用リ라우ティングノードチップ VLSI の設計,” 電子情報通信学会 LSI 設計技術の未来を考える琵琶湖ワークショップ, pp. 56-60 (1996 年 11 月).
- (2) 中谷 泰寛, 正城 敏博, 尾上 孝雄, 村上 孝三: “VCI 共有セルを用いたマルチメディア ATM 通信手法と VLSI 設計,” 電子情報通信学会技術研究報告, DSP96-90, vol. 96, no. 301, pp. 39-46 (1996 年 10 月).
- (3) 正城 敏博, 中谷 泰寛, 尾上 孝雄, 村上 孝三: “VCI 共有セルを用いた ATM 音声通信手法,” 電子情報通信学会ソサイエティ大会, SB-10-2 (1996 年 9 月).
- (4) 佐藤 洋, 森本康夫, 正城敏博, 尾上孝雄, 白川功: “1 チップ MPEG2 デコーダの設計と動き補償器の VLSI 実装,” 情報処理学会 DA シンポジウム, pp. 47-52 (1996 年 8 月).
- (5) 尾上孝雄, 森本康夫, 正城敏博, 白川功: “MPEG2 HDTV 対応可変長符号復号化器, 逆量子化器のアーキテクチャ設計,” 電子情報通信学会総合大会, A-189 (1995 年 3 月).
- (6) 正城敏博, 森本康夫, 尾上孝雄, 白川功: “HDTV レベル MPEG2 復号プロセッサアーキテクチャの VLSI 化設計,” 電子情報通信学会技術研究報告, CAS94-74, vol. 94, no. 475, pp. 9-16 (1995 年 1 月).
- (7) 正城敏博, 森本康夫, 尾上孝雄, 白川功: “MPEG2 HDTV 動画復号化プロセッサ・コアの VLSI 設計,” 電子情報通信学会デジタル信号処理シンポジウム, pp. 235-240 (1994 年 11 月).
- (8) 尾上孝雄, 森本康夫, 正城敏博, 白川功: “MPEG2 HDTV デコーダ・コアのアーキテクチャ設計,” 電子情報通信学会秋期大会, A-93 (1994 年 9 月).
- (9) 正城敏博, 尾上孝雄, 平田博章, 木村浩三, 浅原重夫, 鷺島敬之, 白川功: “多重スレッドアーキテクチャを採用した画像生成用プロセッサの設計 ~高位論理合成システムによる実現~, ” 電子情報通信学会技術研究報告, CAS93-99, vol. 93, no. 432, pp. 67-74 (1994 年 1 月).
- (10) 尾上孝雄, 正城敏博, 平田博章, 木村浩三, 浅原重夫, 鷺島敬之, 白川功: “多重スレッドアーキテクチャを採用した画像生成用プロセッサの設計 ~詳細アーキテクチャの設計~, ” 電子情報通信学会技術研究報告, CAS93-98, vol. 93, no. 432, pp. 61-66 (1994 年 1 月).

- (11) 尾上孝雄, 正城敏博, 古賀拓也, 藤井貴晴, 平田博章, 木村浩三, 白川功: “高位論理合成システムによる画像生成用プロセッサの設計,” 電子情報通信学会秋季大会, A-65 (1993 年 9 月).

目次

第1章 序論	1
1.1 ビデオ画像生成	2
1.2 画像圧縮符号の復号化	3
1.3 ATM ネットワークにおける音声伝送	5
第2章 ビデオ画像生成用多重スレッドプロセッサ	7
2.1 緒言	7
2.2 ビデオ画像生成用アルゴリズムの高速化手法	8
2.3 高位合成システム PARTHENON	9
2.4 シミュレーションによる多重スレッドプロセッサの構成決定	11
2.4.1 ビデオ画像生成用途に適した多重スレッドプロセッサ構成決定手法	11
2.4.2 アプリケーションの解析	12
2.4.3 性能評価	15
2.4.4 多重スレッドプロセッサのハードウェア構成	16
2.5 多重スレッドプロセッサの詳細設計ならびに記述	16
2.5.1 命令フェッチユニット	17
2.5.2 命令解読ユニット	17
2.5.3 命令選択ユニット	20
2.5.4 実行ユニット	20
2.5.5 レジスタユニット	22
2.6 実装結果	22
2.7 結言	23
第3章 HDTV レベル MPEG2 デコーダ	25
3.1 緒言	25
3.2 MPEG2 圧縮符号化アルゴリズム	25

3.2.1	動き予測	26
3.2.2	離散コサイン変換と量子化	28
3.2.3	可変長符号化	30
3.3	デコーダアーキテクチャの設計	31
3.3.1	汎用プロセッサによる復号化	31
3.3.2	プログラマブル DSP による復号化	32
3.3.3	専用機能ユニットによる復号化	33
3.3.4	MP@HL に適したデコーダアーキテクチャ	33
3.4	HDTV レベル MPEG2 デコーダの設計	34
3.4.1	可変長符号復号化器	34
3.4.2	逆量子化器	37
3.4.3	逆離散コサイン変換器	38
3.4.4	動き補償器	45
3.4.5	マクロブロックレベル・パイプラインバッファ	49
3.4.6	フレームメモリ構成とインタフェース	51
3.4.7	シーケンス制御部	54
3.5	実装結果	55
3.6	結言	56
第4章	VCI 共有セルを用いたマルチメディア ATM ネットワークにおける音声通信手法	59
4.1	緒言	59
4.2	マルチメディア ATM ネットワーク構成	59
4.2.1	音声と ATM セル	60
4.2.2	VTOA を考慮したマルチメディア ATM ネットワーク構成	61
4.3	VCI 共有セル化手法	63
4.3.1	従来の音声セル化手法	63
4.3.2	標準 ATM の仮想チャネル	65
4.3.3	VCI 共有セルの構成	65
4.3.4	VCI 共有セルの VCI 割り当て	67
4.3.5	VCI 共有セル長と伝送効率の評価	68
4.4	結言	69
第5章	マルチメディア ATM ネットワークにおけるリルートノード	71
5.1	緒言	71

5.2	リルートノードアーキテクチャの設計	71
5.2.1	アドレス生成部	72
5.2.2	ヘッダテーブル	73
5.2.3	標準セルバッファ	73
5.2.4	フェローセルバッファ	73
5.2.5	セル組み替え制御部	73
5.3	バッファサイズシミュレーション	75
5.4	実装結果	77
5.5	結言	77
第6章	結論	79
	謝辞	81
	参考文献	83

第1章

序論

近年、広帯域 ISDN(Broadband-Integrated Services Digital Network) 上に展開される ATM (Asynchronous Transfer Mode) 通信技術の発展により、従来の文字情報やデータだけではなく、画像や音声をも統合的に実時間処理するマルチメディア情報通信の有力な手段として ATM ネットワークを利用する動きが盛んになってきている。計算機のデータ伝送等、情報伝達系を中心に発展し、既に実用段階に入ってきた ATM ネットワークは、CATV(cable television) 等のデジタル有線放送系や既存の電話網である音声用狭帯域ネットワークとは個別のネットワークとして活用されてきた。今後、より実用的な通信インフラストラクチャの構築のためには、高速データ、画像、音声などのマルチメディア情報を一元的に扱うことのできる新たな統合的ネットワーク手法を確立する必要がある。

マルチメディア ATM ネットワークを構築する有力なアプローチとして、新たに必要となる諸機能を既に実用段階に入っているデータ伝送系の ATM ネットワークに付加するという手法が考えられる。マルチメディア情報の実時間処理を行うために実現すべき要素技術として、送信側における画像生成、画像圧縮、音声符号化等の技術、伝送路における多種メディアの高品質/高信頼/高効率な処理技術、受信側における画像圧縮符号や音声符号に対する実時間復号化技術、さらには、これらの処理を行うための VLSI 設計技術が挙げられる。

特に VLSI 設計技術に関しては、近年、ASIC(Application Specific Integrated Circuit) 設計の分野において、高度な CAD(Computer-Aided Design) ツールの出現により、大規模かつ複雑な機能を自動的に合成することが可能となっている。特に、ハードウェア記述言語を用いて仕様記述を行い、この記述からマスクパターンを自動生成する高位合成ツールが急速に進展を遂げ、今や実用段階に入っている。このような高位合成ツールの出現により、設計者は論理設計やレイアウト設計に関する細かい知識を持ち合わせなくても、高位のアーキテクチャ/機能設計に専念しながらシステムを実現することができるようになった。

ここで注意すべきことは、記述を行うことの利点として、詳細設計の自動化という点だけで

はなく、設計仕様の明確化が挙げられ、仕様変更が起こった際には、それに伴う設計変更が容易であり、かつそれに要する時間を大幅に削減することができるということである。これらの VLSI 設計環境の充実に伴って、複雑な機能を持つ新しいアーキテクチャを容易に実現することができるようになった。

本論文では、このような状況をふまえ、前述の要素技術のうち、画像や音声の実時間処理を行うために重要な、(i) ビデオ画像生成用途に適した多重スレッドプロセッサの VLSI 化設計、(ii) ビデオ動画像圧縮符号の国際標準 MPEG2 アルゴリズムの HDTV(High Definition Television) レベルデコーダの VLSI 化設計、および (iii) 音声伝送に適したマルチメディア ATM ネットワークの構成と音声用 ATM セルの処理を行うリルートノードの VLSI 化設計、について考察する。

1.1 ビデオ画像生成

CG(Computer Graphics) の分野においては、従来の単純なモデルと比較し、より現実感のある高品位なビデオ画像を高速に生成することが要求されている。具体的な用途としては、娯楽映像ソフトウェアや VR(Virtual Reality) 等の分野があり、これらの技術の実現が各分野の発展に果たす役割は甚大である。

高品位なビデオ画像を計算機上で生成するアルゴリズムとしては、視線探索法^[1]や Radiosity 法^[2]等が知られている。

視線探索法は、特に写り込みや反射、透過等の表現モデルとして優れている。ビデオ画像の生成手続きとしては、スクリーン上のピクセル単位に、そのピクセルを通して入射する光線を逆追跡し、光線を放出する物体表面を交差判定により同定した上で輝度計算を行う。

Radiosity 法は、特に室内などの間接光の表現に優れたモデルである。物体間の相互反射を考慮するために、まず、物体表面をパッチと呼ばれる一定の単位に分割する。この各パッチに受けた光のエネルギーを拡散反射により放出し、他のパッチに与える新たな光のエネルギーを計算する。この連鎖的な処理を、全てのパッチが新たに受けるエネルギーの値が閾値を下回るまで演算する。最後に視点から見える物体を交差判定により同定し、描画を行う。

このようなビデオ画像生成アルゴリズムを用いて HDTV 等の画素数が極めて多い画像に対しても高速に描画するためには、演算処理を行うプロセッサ内において 1 サイクルで複数の命令を発行可能とした多重命令発行方式を用いることが考えられる。しかしながら、ビデオ画像生成アルゴリズムには、交差判定実行時に分岐方向が決定される条件分岐や、再起呼び出しされるサブルーチン呼び出し、形状データを扱うために必要となる間接データアクセス等が頻繁にあらわれるため、近隣の命令間に依存関係が発生する場合が多い。

この細粒度並列性が低いという性質のために、通常プロセッサの高速化手法などで用いられる複数命令の同時発行機構を用いた Superscalar 方式や、パイプラインの多段化等の技術を用いた Superpipeline 方式を利用しても、プロセッサの持つハードウェア資源を最大限に利用することができないばかりでなく、十分にビデオ画像生成アルゴリズムを高速に実行することができない。

これに対して、本論文では、ビデオ画像生成アルゴリズムが粗粒度並列性を有し、互いに独立に実行可能な処理単位(スレッド)に分解できることを利用し、画像生成処理の高速化を図る。ビデオ画像生成アルゴリズムのスレッドは、例えば視線探索法では、画面のピクセル毎の輝度計算を指し、Radiosity 法ではパッチ毎に受ける光のエネルギー計算を指す。本論文では、画像生成を高速に処理するシステム向けのプロセッサとして、複数のスレッドを同時に実行することができ、かつ、ハードウェア量の削減を達成することのできる多重スレッドアーキテクチャを採用し、その実装を行う。多重スレッドプロセッサは1チップ内に複数の命令解読部を持ち、各解読部が1組の実行ユニットを共有する構成を取る。このため、共有したユニットの調停等の極めて複雑な制御が必要となり、これまでにその実現に成功した例は報告されていなかった。

本論文では、ハードウェアの機能を動作レベルで記述/検証し、動作記述からハードウェアを自動合成することのできる高位合成システムを用いることで、多重スレッドプロセッサのVLSI化が可能であることを示す。

1.2 画像圧縮符号の復号化

マルチメディア情報のデジタル化が必要不可欠となる中、音声や静止画はまだしも動画像においてはその情報量は莫大であるため、伝送/処理に必要となる帯域/記憶容量が膨大なものとなっている。このため、衛星放送、デジタル地上波放送、テレビ会議、テレビ電話等の様々なマルチメディア応用分野の開発のためには、これらのマルチメディア情報を効率よく伝送/処理するための圧縮技法が必要不可欠なものとなっている。このような経緯から、MPEG(Moving Picture Experts Group)が組織され、動画像とそれに伴う音声の圧縮符号化技法の標準化を検討してきた。

MPEGは、まず、ISO/IEC 11172(通称MPEG1^[3])の標準化を完了し、特に、動画像に関する標準であるMPEG1 videoでは、VCR(Video Cassette Recorder)レベル(352×288画素、毎秒30フレーム)までの画質の動画像を1.5Mbps以下のビットレートで符号化することを規定した。MPEG1は、CD-ROMやハードディスクなどの蓄積メディアや、LAN上でのビデオメール、テレビ会議をその主な対象としている。

以後MPEGは、さらに高品質な動画像に対する高度な圧縮符号化を行うためのMPEG2^[4]

の標準化を完成した。当初 MPEG2 では、NTSC や PAL 等の現行テレビレベルの動画像を約 5Mbps で符号化し、その最大ビットレートを 10Mbps とすることを目的としていたが、かねてから強く要求のあった HDTV レベルの高品質な動画像への対応をも可能とするべくビットレートの上限を大幅に引き上げた。これにより、HDTV レベルの動画像を 40 ~ 80Mbps 程度に圧縮符号化することが可能となった。これまでに、MPEG2 は、DVD(Digital Video Disc) や、米国の次世代地上波 HDTV 放送である ATV(Advanced Television)、双方向マルチメディアネットワークサービスの国際標準 DAVIC(Digital Audio-Visual Council) 等への採用が決定されている。

MPEG2 は、各種の応用分野に対応させるために一種類のシンタックスに限定せず、複数のビットレート、解像度、フレームレート、画像品質を想定している。Table 1.1 に示すような、「プロファイル」と「レベル」と呼ばれるシンタックスの階層化を行い、各種応用分野に対応させている。プロファイルは、圧縮方法の他に予測モードや色差フォーマットを規定するもので、レベルは縦横の画素数や、フレームレート等の上限を規定している。

Table 1.1 MPEG2 のプロファイルとレベル

		High Compression			
Profile Level	Simple	Main	SNR Scalable	Spatial Scalable	High
High		MP@HL (ATV HDTV)			HP@HL
High1440		MP@H1440		SSP@H1440 (ED HDTV)	HP@H1440
Main	SP@ML (DigitalCATV)	MP@ML (Direc TV)	SNP@ML		HP@ML
Low		MP@LL	SNP@LL		

この標準化により、画像情報を扱う分野において、MPEG2 の符号化、復号化処理が可能なシステムの応用範囲がますます広がっている。特に、復号化システムは符号化システムと比較してその需要が増大している。符号化処理は処理過程にいくつかの選択枝を持ち、処理システム全体に柔軟性を要求される。また、特に蓄積メディアを対象とした符号化処理においては、実時間処理を行うことよりも、いかに符号の圧縮率を高め、かつ、元の画像品質を維持することができるかが重要な課題となる。これに対して、復号化処理は処理過程が限定されており、要求される処理能力も比較的小さいが、実時間処理を行うシステムを構築することが最も重要な課題となっている。またシステム全体を小型化し、消費電力を削減するために復号化 VLSI を 1 チップに集積することが必要不可欠である。

これまでに発表された MPEG2 デコーダ^[5,6]の多くは、現行テレビレベルである MP@ML (Main Profile at Main Level) を対象としている。しかしながら、MPEG2 の規定のうち HDTV レベルの高解像度動画像を扱う、いわゆる MP@HL(Main Profile at High Level) を対象としたデコーダは実現されていない。

そこで、本論文では MPEG2 MP@HL 動画像の復号化を実時間で処理することができるデコーダを 1 チップで VLSI 化実装することを目的として設計を行う。MP@HL 動画像は MP@ML 動画像と比較して 6 倍以上の処理能力を必要とするため、MP@HL デコーダの VLSI 化には、従来のアーキテクチャを踏襲するばかりではなく、さらに高性能なアーキテクチャの考案が必要となる。従来の DSP コアと専用演算器を用いた構成ではなく、MPEG2 の復号化の各処理過程を実行するために新たに考案した専用アーキテクチャを開発することによって、HDTV レベル MPEG2 デコーダの 1 チップ VLSI 化が実現可能となる。

1.3 ATM ネットワークにおける音声伝送

ATM ネットワークは、元来、マルチメディア情報を扱うための主要な技術として ITU-T (International Telecommunication Union - Telecommunication standardization sector) 等で提唱されてきた。ATM ネットワークでは、情報を固定長のブロックに分割し、ATM セルを用いて非同期に伝送する。ネットワークに ATM 方式を採用することにより、ネットワークの帯域割当を柔軟に行うことができるため、様々なトラフィック特性に適合した伝達サービスを提供することが可能となる。さらに、ATM ネットワークでは、いわゆる統計多重効果により、ネットワークが供給する帯域を効率よく利用することができる。また、ATM セルフォーマットが固定であり、ATM セルのヘッダのみで伝送/交換処理が行えるため、ハードウェアによる高速スイッチングを実現することができる。

しかしながら、これまで ATM ネットワークは LAN 等のデータ伝送系で活用されてきている。このような状況に対し、近年、ATM の特性を活用したマルチメディア ATM ネットワークの実現に対する要求が高まっている。FTTH(Fiber To The Home) 構想や FTTC(Fiber To The Curb) 構想等では、一般家庭またはその付近まで光ファイバケーブルを敷設し、各家庭から ATM 方式を用いた通信ネットワークの実現を目的としている。このようなマルチメディア ATM ネットワークを構築するための主要な技術的課題は、音声、画像、高速データ等の多様なトラフィックをいかにして一元的に扱うかということである。

既に、高速データ通信やテレビ会議等の画像通信に ATM 方式を利用する研究は数多く行われている。一方、音声通信に ATM ネットワークを用いることは、既存の STM 方式を用いた N-ISDN と比較して高価であったが、VLSI 技術の進歩により、ATM ネットワークによる音声

通信 (Voice and Telephony Over ATM: VTOA) が実用的に可能となってきている^[7]. このような状況をふまえて, ATM Forum 等では, 既存の STM ネットワークと ATM ネットワークの相互接続点に必要な機能や処理等を中心に議論が行われている^[8-11].

従来から, 音声データを ATM ネットワークを用いて伝送する際の最も重要な技術課題としては, 端末から端末までの伝送遅延の削減が挙げられている. 伝送遅延の主な原因は, 音声データを ATM セルのペイロード (48 バイト) に蓄積するために要するセル化遅延である. セル化遅延は, 動画像や高速データ等のビットレートの高いトラヒックでは問題にならないのに対して, 音声等の低ビットレートのトラヒック固有に発生する課題である. これまで, セル化遅延に対処するセル化手法として, パーシャルフィルセル, ショートセル, コンポジットセル等が提案されている^[12-14]. しかしながら, FTTH 構想の実現下のように, 本研究が対象するような終端まで ATM 化されたネットワークにおいて, 標準 ATM ネットワークとの親和性と高伝送効率の双方を満たすセル化手法は提案されていない.

本論文では, 各終端から ATM 方式を用いて伝送するネットワークにおいても音声の伝送効率を高めるために, あるスイッチから次段のスイッチまで同一の経路をたどる複数の音声用セルを, ネットワーク内で動的に多重化することのできるリルート機能を導入したネットワーク構成と, 新たな音声用セル化手法である VCI 共有セル化手法を考案する. また, リルート機能を実現するリルートノードのハードウェア構成を考案し, VLSI 化実装する. リルートノード内のみで標準 ATM セルと VCI 共有セルの区別を認識し, 各セルに対して必要な処理をするだけで, 他のネットワーク構成要素に影響を与えることなく, マルチメディア ATM ネットワーク内の音声伝送効率を高めることができる.

以下, 第2章ではビデオ画像生成用多重スレッドプロセッサの VLSI 化設計について考察し, 第3章では HDTV レベル MPEG2 デコーダの VLSI 化設計について述べる. 第4章では音声伝送を効率的にする VCI 共有セルを用いたマルチメディア ATM ネットワーク手法について考察し, 第5章では, 第4章で考案したネットワーク手法を構築するために必要なリルートノードの VLSI 化設計について述べる. 第6章に以上の章の結論を述べる.

第 2 章

ビデオ画像生成用多重スレッドプロセッサ

2.1 緒言

本章では、多重スレッドアーキテクチャ^[15]を採用し、高品位な画像を高速に生成するプロセッサを、高位合成システム PARTHENON^[16]を用いて設計する手法について記述する。

多重スレッドアーキテクチャは、ビデオ画像生成アルゴリズムに内在する粗粒度 (命令レベル) 並列性を利用して高いスループットを得ようとするものであり、複数の命令流を同時に実行可能とした資源共有型のプロセッサアーキテクチャである。このため、従来の Superscalar や VLIW などの、細粒度 (命令レベル) 並列性の利用による並列アーキテクチャが不得意としたアルゴリズムに対しても、性能向上を期待することができる。しかしながら、命令フェッチユニットや実行ユニットが複数の命令流間で共有されることに起因した、種々の複雑な制御機構が必要となり、現在までにその実現に成功した例はなかった。

本章では、まず、複雑な構造を持つ多重スレッドプロセッサの諸機能を、ハードウェア記述言語による単純な動作記述や、所有する部品の機能記述として表現する手法を考察し、次いで、多重スレッドプロセッサを高位合成システムによって実現する設計手法について述べる。設計に使用する高位合成システム PARTHENON は動作記述言語 SFL (Structured Function description Language) を中心に、動作検証、論理合成まで一貫した VLSI 設計システムである。ハードウェアの機能を動作レベルで記述・検証し、動作記述からハードウェアを自動合成するために、VLSI 設計工数の大幅な短縮を図ることができる。本章では SFL を用いて多重スレッドプロセッサの各部の機能を記述し、PARTHENON を用いて動作検証と論理合成を行う。これにより、従来困難とされていた、多重スレッドアーキテクチャによる高性能プロセッサの VLSI 化が可能となることを示す。

2.2 ビデオ画像生成用アルゴリズムの高速化手法

計算機によるビデオ画像生成の分野では、視線探索法^[1]や Radiosity 法^[2]など、高品位画像を生成するさまざまなアルゴリズムが提案されてきた。しかしながら、これらのアルゴリズムを実行する際には膨大な演算が必要となり、全ての演算を1つのプロセッサで実行しようとすると、画像生成速度の面で満足な性能が得られない。多重命令発行 (Superscalar, VLIW) により細粒度並列性を利用して処理性能の向上を図る場合、近隣命令間の依存性のために、並列実行を行うには限界がある。

一方、これらのビデオ画像生成アルゴリズムは粗粒度並列性が高く、互いに独立に実行可能な処理単位 (スレッド) に分解できる。ここでスレッドとは、例えば、視線探索法では画素単位の輝度計算に、Radiosity 法では描画する物体表面の一定単位 (パッチ) が受ける光エネルギー計算に、それぞれ相当している (Fig. 2.1)。

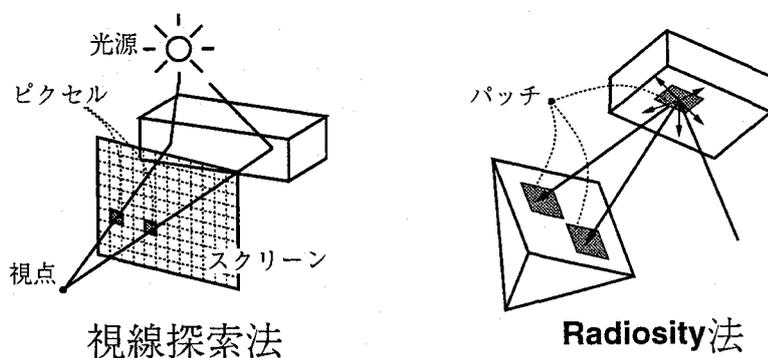


Fig. 2.1 ビデオ画像生成アルゴリズムの概要と粗粒度並列性

この粗粒度並列性を利用し、マルチプロセッサシステム^[17] (Fig. 2.2(a)) において各スレッドをそれぞれの要素プロセッサを用いて処理を行った場合では、その処理時間という面では改善されるが、ハードウェア量が膨大となるうえ、遠距離メモリアクセスなどでプロセッサが一時的に停止してしまい、本来の性能を十分に引き出していないというのが現状である。

マルチプロセッサシステムにおいては、各プロセッサ内の機能ユニットの稼働率が低いため、複数個の要素プロセッサで一つの機能ユニット群を共有するというアーキテクチャが考えられ、このアーキテクチャを多重スレッドアーキテクチャと呼ぶ^[15] (Fig. 2.2(b)). 多重スレッドアーキテクチャを用いた場合、ハードウェア量を減少させながら、並列度に見合ったプロセッサのスループットの向上を図ることができる。さらに、多重スレッドアーキテクチャでは、互いにデータ依存関係のない複数スレッドが、命令テキストを共有して並列実行されるため、あるスレッドが必要とするデータを、既に他のスレッドがキャッシュ上にロードしている可能性があるため、この場合にはキャッシュミスは発生しない。すなわち、プロセッサまたはシステム全体と

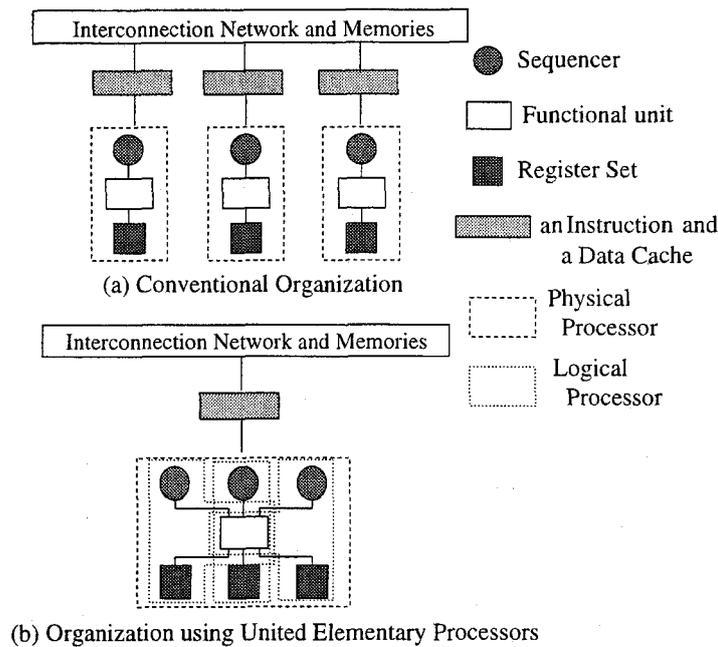


Fig. 2.2 マルチプロセッサシステムの構成

いった大局的な見地からは、単一スレッドでの実行の場合よりも命令キャッシュのヒット率を向上させることができる^[15]。このため、遠距離メモリアクセスの発生回数自体を減少させることができる。

しかしながら、2.5 節に示すように、命令フェッチユニットのバッファ制御や、割り込み処理等のために、状態遷移を行う命令解読ユニットは、それ自体だけでも制御が複雑なうえ、スレッドが交錯する各機能ユニットに対しても信号・データの受渡しを正しく設計しなければならない。したがって、従来の設計概念では、設計自体に膨大な時間を必要とするだけでなく、その修正にも多大な労力を必要とする。このような理由で、これまで多重スレッドプロセッサの実現例は報告されていなかった。そこで、詳細設計段階での仕様変更が多いことをも考慮に入れて、本プロセッサを高位合成システムを用いて設計・検証を行うという方法を採用した。

2.3 高位合成システム PARTHENON

高位合成による VLSI 設計では、ハードウェア記述言語を用いて回路の動作・機能を記述し、シミュレータにより機能検証を行うことができる。このため、初期の段階から入力誤りや設計誤りを発見することができ、動作記述に変更を加えるだけで容易に修正を行うことができる。

高位合成システム PARTHENON は、階層的な動作記述言語 SFL を用いて、回路を論理合成する。SFL は VHDL, Verilog HDL, UDL/I のようなハードウェア記述言語と比較して次のような特徴がある^[18,19]。

- 設計対象を単相同期回路に限定している。
- 接続記述を禁止し、動作のみを記述する。
- module と呼ばれる機能ユニットに動作を記述し、submodule を呼び出すことにより、階層的な記述を可能としている。

PARTHENON は SFL を中心に動作シミュレータ SECONDS, 論理回路シンセサイザ SFLEXP, オプティマイザ OPTMAP で構成されている。PARTHENON を用いた設計の流れは Fig. 2.3 のようになる。

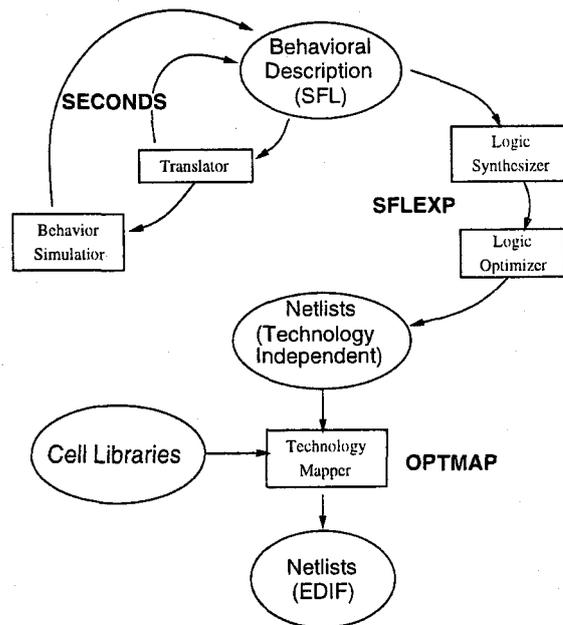


Fig. 2.3 PARTHENON による設計フロー

- 1° 設計仕様に基づき SFL で回路の動作を記述し、SECONDS で SFL のシンタクスチェックと設計したハードウェアの機能シミュレーションを行う。SECONDS は動作記述の階層構造を保持したまま、ネット展開を行わずにハードウェアの動作そのもの (出力端子, レジスタ, 内部状態など) をシミュレートする。従って, 構成要素におけるゲートの遅延などは考慮されないが, 高速なシミュレーションを行うことができる。
- 2° 1° で記述した回路に対して SFLEXP を用いて論理合成を行う。まず記述から状態の割り当て, 制御レジスタ, セレクタ, バスの挿入を行い, 論理の真理値表や接続情報を生成する。これをいったん 2 段の積和形表現に変換し, 論理圧縮を行った後, 多段論理回路へと論理合成する。SFL では, セルライブラリを利用して実現することができる特定モ

ジュールに対しては、SFLEXP が論理合成を省略するように指定する。ただし、この場合でも、論理合成を行わなかったモジュールの機能を記述することにより、動作シミュレーションが実行できる。

- 3° OPTMAP により、セルライブラリのゲート遅延や負荷、制約を考慮しながら実部品にテクノロジマッピングして、タイミングを考慮した回路最適化を行う。

PARTHENON はネットリストを EDIF 形式で出力ができるため、レイアウト機能を備える COMPASS Design Navigator などの CAD ツールへのデータの受渡しが可能である。

2.4 シミュレーションによる多重スレッドプロセッサの構成決定

多重スレッドプロセッサの構成ユニットはそれぞれ、各スレッドのプリフェッチバッファを含む命令フェッチユニット、各スレッド毎に有し、命令解読、依存解析、割り込み検出を行う命令解読ユニット、各スレッドからの命令を、機能ユニット群へ動的に調停を行うスケジューラユニット、整数演算、ロード/ストア、浮動小数点加算、浮動小数点乗算等の演算を行う実行ユニット、ならびに各スレッド毎に有するレジスタユニットである。本プロセッサでサポートする命令セットは SPARC^[20] 互換とする。ただし、拡張精度の浮動小数点演算は実際には使用しないためこれをサポートせず、新たにスレッド間通信レジスタ用命令やスレッドの状態を変化させる命令を付加する。なお、割り込みについては、アプリケーションプログラムの実行に必要なもののみをサポートする。本節では、多重スレッドプロセッサの各ユニットのより詳細な構成を決定するために、Radiosity 法のアプリケーションプログラムの実行に適したハードウェア構成を考察する^[21]。

2.4.1 ビデオ画像生成用途に適した多重スレッドプロセッサ構成決定手法

多重スレッドプロセッサをビデオ画像生成用途に適したアーキテクチャに最適化するためには、ビデオ画像生成用アプリケーションプログラムの振る舞いの解析と、プロセッサモデルの命令実行シミュレーションによる評価が必要である。アプリケーションプログラムの解析に基づき、プロセッサ内部の機能ユニットの種類や数、各機能ユニットのパイプライン段数、分岐機構、ならびにキャッシュの構成を決定することができる。さらに、命令実行シミュレーションにより、設計したプロセッサモデルの、実際のアプリケーションプログラムに対する有効性を評価することができる。本節で採用したビデオ画像生成用途に適した多重スレッドプロセッサ構成決定手法は以下の通りである。

- 1° C 言語で記述された Radiosity 法のプログラムについて、SPARC ver. 7^[20] の命令セットで実行命令のトレースを行い、その結果から動的な命令出現頻度、命令間の依存関係、分岐命令の出現頻度等の解析を行う。
- 2° 解析結果に基づいて、プロセッサ内部の機能ユニットの種類や、その個数、機能ユニットのパイプライン段数、および分岐機構を決定する。
- 3° 多重スレッドアーキテクチャのビデオ画像生成アルゴリズムに対する有効性を調べるために、命令実行シミュレータ上で Radiosity 法のプログラムを実行し、性能評価を行う。有効性が満足できない場合は 2° に戻り、十分であれば終了する。

2.4.2 アプリケーションの解析

2.4.2.1 命令の出現頻度と機能ユニットの構成

Radiosity 法のプログラム実行時の命令の動的出現頻度を Table 2.1 に示す。

Table 2.1 命令の動的出現頻度

命令の範疇	出現頻度 (%)
ロード/ストア	33.15
整数演算	33.89
浮動小数点加算 (add, sub, cmp, abs, neg, cnv)	8.83
浮動小数点乗算 (mul, div)	6.67
分岐	12.62
その他	4.85

同じ演算器構成で処理可能な命令を同一の範疇として分類する。ここで、浮動小数点除算命令は浮動小数点乗算器に採用するアルゴリズムを用いることとし、浮動小数点乗算命令と同じ範疇とした。ビデオ画像生成アルゴリズムのアプリケーションプログラムは、ロード/ストア命令と整数演算命令の動的出現頻度が他の命令と比較して高く、浮動小数点演算命令は比較的出現頻度が低い。

次に、命令の出現頻度の時間変化を見るために、命令群毎の出現頻度の変化を Fig. 2.4 に示す。Fig. 2.4 において、横軸は 30 のパッチから構成された 1 ブロック分の Radiosity 計算にかかる時間を 100 として表し、縦軸に各々の命令群の出現頻度を表している。処理の最初の段階では、データの初期化が行われているためロード/ストア命令が多いが、浮動小数点演算命令や分岐命令の出現頻度は低い。光のエネルギー計算や交差判定を行う、Fig. 2.4 中の時間 20 以降の

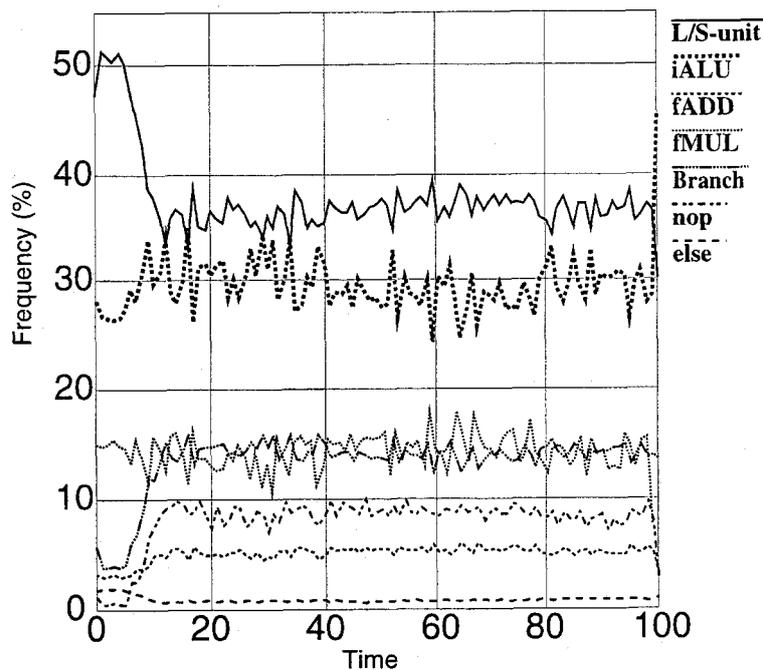


Fig. 2.4 命令の出現頻度変化

段階では、浮動小数点演算命令や分岐命令の出現頻度が高く、各々の命令群の出現頻度の変化は少なくほぼ一定となり、Table 2.1 の値と一致する。従って、Table 2.1 の命令出現頻度が定常的に保たれていると考えて最適なアーキテクチャを考案する。

Table 2.1 より、ロード/ストア命令と整数演算命令の出現頻度は、それぞれ 33.15%、33.89% と高いため、ロード/ストアユニットと整数演算ユニットを独立に設けることとする。浮動小数点演算命令は 15.5% と比較的出現頻度が少ないが、他の演算と比較して処理に要する実行サイクル数が多く、また出現頻度に偏りがあるため、浮動小数点加算ユニットと浮動小数点乗算ユニットが必要となる。

2.4.2.2 命令間の依存関係とパイプライン

依存関係を有する命令間の距離が離れている場合は、プロセッサの性能低下の要因とはなりにくい。このため、複数のスレッドが同時に命令を発行する機会が多くなり、機能ユニット数の増加、パイプライン段数の多段化による性能向上を実現することができる。Fig. 2.5 に、浮動小数点演算命令に関して、命令フォーマット中のコンディションコード、ならびにレジスタの依存関係を有する命令間の距離を示す。

浮動小数点命令のコンディションコードによる依存関係は非常に短い距離のみで発生する。また、レジスタの依存関係においては、直前の命令の実行結果に依存しているものが 26.2% と最も多く、4 命令以内で 50% 以上、10 命令以内で 70% に依存関係を有している。このた

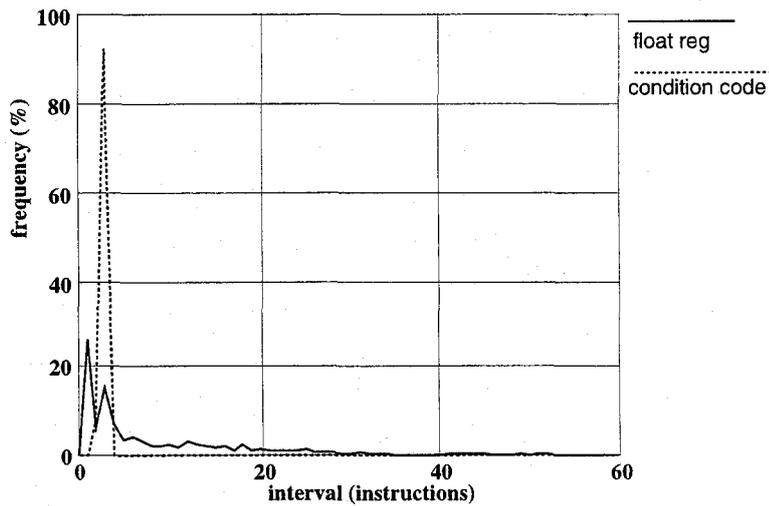


Fig. 2.5 浮動小数点命令の依存関係の距離

め、浮動小数点演算ユニットのパイプライン段数を4段以上とすると、従来のプロセッサ構成ではこれらの依存関係のため性能低下が発生する。しかしながら、多重スレッドプロセッサでは、互いに依存関係のないスレッドの命令が同時に複数発行されるため、パイプラインを有効に利用することが可能となり性能低下の要因とはならない。以上の点と各演算の処理の複雑さを考慮して決定した各機能ユニットのパイプライン段数、すなわち各命令の実行サイクル数をTable 2.2に示す。なお、浮動小数点除算命令は浮動小数点乗算器に採用するアルゴリズムを用いて演算するため、28サイクルで1命令の処理を行う。

Table 2.2 機能ユニットの実行サイクル数

機能ユニット	命令	サイクル数
ロード/ストアユニット	ロード/ストア	3
整数演算ユニット	加減算, 論理演算, 比較演算	1
浮動小数点加算ユニット	加減算, 比較演算, 絶対値, 型変換	3
浮動小数点乗算ユニット	乗算	3
分岐命令演算ユニット	除算	28
	分岐	2

2.4.2.3 分岐命令とその実行

分岐命令の実行は命令の実行系列を不連続にするためパイプラインが停止し、性能低下の要因となる。従って、分岐命令の振る舞いは、命令キャッシュの構成や、命令フェッチ、分岐処理の制御、およびパイプライン構成に多大な影響をおよぼす。このため、従来のプロセッサでは

分岐方向の予測や命令の先読み、命令の投機的実行等、性能低下を確率的に抑制する手段が用いられることが多い。しかしながら、Radiosity 法のプログラムでは、分岐方向が実行時に決定するために分岐予測が極めて難しい。また、依存関係のない複数のスレッドを同時に実行するため、あるスレッドの処理性能向上のために投機的に発行した命令が誤った分岐予測に基づいていた場合に、他のスレッドの実行を妨げ性能低下を引き起こす要因となる。このため、分岐命令は各スレッド内の命令解読ユニットで実行することとし、投機的な命令発行は行わない。

2.4.3 性能評価

提案した実行ユニットの基本構成(ロード/ストアユニット, 整数演算ユニット, 浮動小数点加算ユニット, 浮動小数点乗算ユニット: model A)の他に, 整数演算ユニットを付加した構成(model B), ロード/ストアユニットを付加した構成(model C), 整数演算ユニット, ロード/ストアユニットともに付加した構成(model D)について, ソフトウェアシミュレーションによる性能評価を行った。各構成においてスレッド数を変化させたときの, 1 スレッド基本構成に対する性能向上比を Fig. 2.6 に示す。

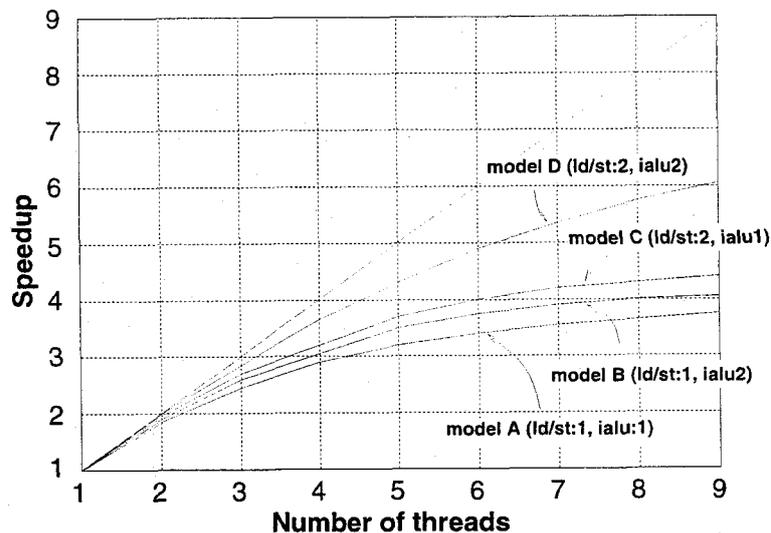


Fig. 2.6 スレッド数による性能評価

シミュレーション結果より, 基本構成のスレッド数を増加させると, 性能向上は得られるが投資したハードウェア量に見合った性能向上比が得にくくなるのが分かる。これは, スレッド数が4以上となると, 異なるスレッド間の各命令を実行ユニットに発行する際, 機能ユニットが4個であることから競合の発生する頻度が増加するためである。このため, スレッド数の多い構成に対しては, 競合の発生するユニットを増加させることで性能向上を果たすことができる。

2.4.4 多重スレッドプロセッサのハードウェア構成

Fig. 2.6 の結果と 1 チップに実装可能なハードウェア量の評価から決定された、ビデオ画像生成用多重スレッドプロセッサの構成を Fig. 2.7 に示す。

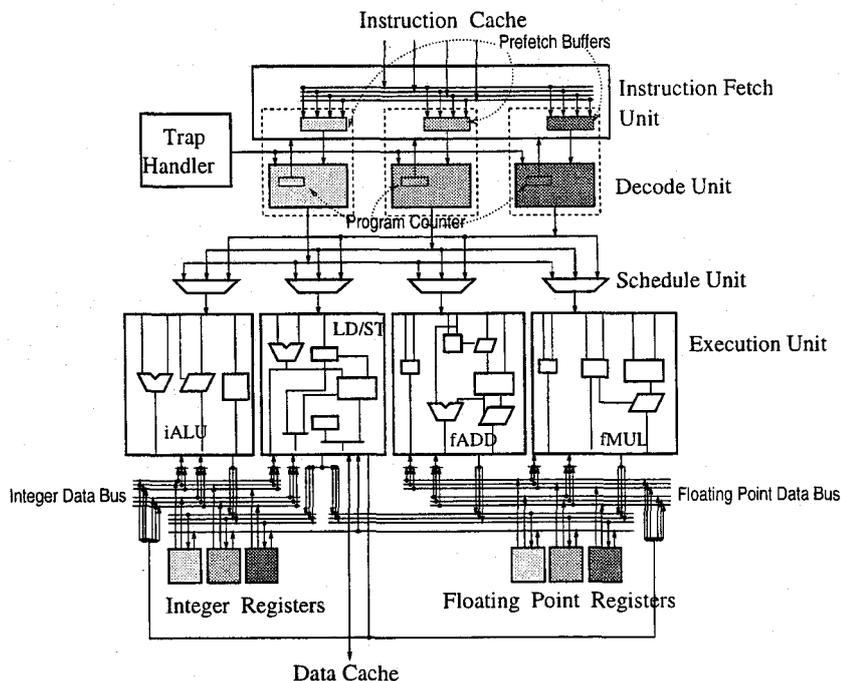


Fig. 2.7 ビデオ画像生成用途向け多重スレッドプロセッサのハードウェア構成

1 チップ内には 3 論理プロセッサを、実行ユニットには独立に実行可能な、ロード/ストアユニット、整数演算ユニット、浮動小数点加算ユニット、浮動小数点乗算ユニットの 4 機能ユニットを実装する。分岐命令は各スレッドの命令解読ユニットで実行する。パイプライン段数はオリジナルの SPARC プロセッサに多重スレッドアーキテクチャ特有の命令選択ステージを加えた、命令フェッチ、命令解読、命令選択、命令実行、書き戻しの 5 段とする。Fig. 2.7 の構成は、Fig. 2.6 より 1 スレッドの基本構成プロセッサに対して 2.5 倍の処理能力を有することがわかる。

2.5 多重スレッドプロセッサの詳細設計ならびに記述

本節では、多重スレッドプロセッサの各部の詳細設計、ならびに、実際に SFL の特性を考慮して記述した結果について述べる。多重スレッドプロセッサの各ユニットを SFL の “module” に対応させ、その動作の記述を行う。

2.5.1 命令フェッチユニット

Fig. 2.8 に示すように、命令フェッチユニットは、各スレッドからの命令フェッチ要求を調停し、実際にメモリアクセスを行う命令フェッチ部、およびスレッド毎に存在し、一度にフェッチした複数の命令を蓄えるプリフェッチバッファから構成される。命令フェッチ部では、各スレッドからのフェッチ要求に対し、順次キャッシュへのアクセス権を移していくが、分岐の発生などでプリフェッチバッファ内の命令数が少なくなったスレッドに対しては、この優先度を上げる。アクセス権を得たスレッドは命令キャッシュから一度に4命令を取得し、空状態にあるプリフェッチバッファに格納する。プリフェッチバッファの命令は、解読ユニットからのプログラムカウンタ値に応じて解読ユニットに送付される。

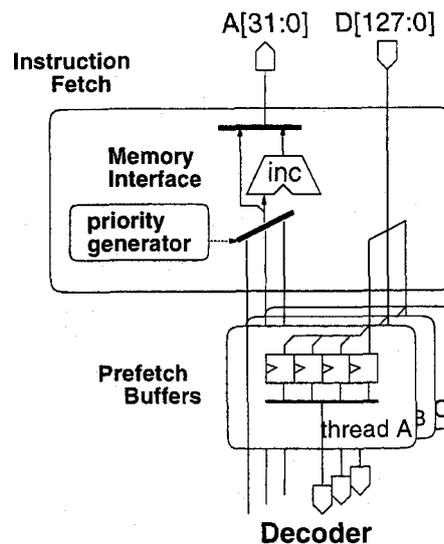


Fig. 2.8 命令フェッチユニットのブロック図

SFL では、プリフェッチバッファの書き込み/無効化の制御を行うプリフェッチバッファ制御モジュールと、各スレッドからのフェッチ要求を調停し、実際のフェッチ動作を行う命令フェッチモジュールとに分けて記述を行う。多重スレッドプロセッサでは、単純にプログラムカウンタの値を用いて命令フェッチを行うのではなく、プリフェッチバッファの内部状態を考慮して命令をフェッチする。また、分岐命令やキャッシュミスの影響で複雑なバッファ制御が必要となるが、これらの制御を SFL により記述する。

2.5.2 命令解読ユニット

命令解読ユニットは本来の解読機能を司る解読部の他に、演算に使用するオペランドの依存関係の有無を判定する依存解析部、割り込み検出部、命令シーケンス制御部のサブモジュールに分割して階層的に記述を行う (Fig. 2.9)。

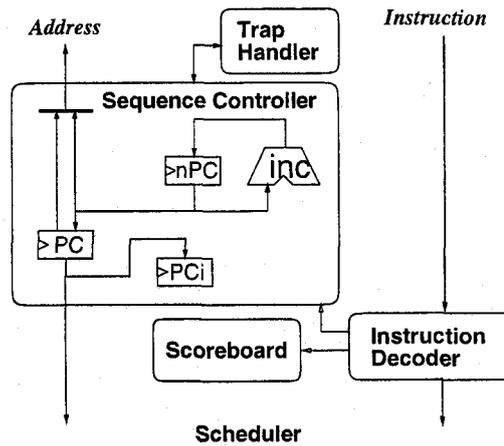


Fig. 2.9 命令解読ユニットのブロック図

2.5.2.1 解読部

プリフェッチバッファからの命令コードをデコードし、実行すべき機能ユニットへ出力する命令コードを生成する。オペランドのレジスタ番号や状態コードの参照など、命令解読ユニット内の各部が必要とするデータを生成し、伝達することも行っている。従って、解読部の機能を記述するモジュールは、入力データをもとに各機能ユニットに送付する命令コードや信号を生成する組合せ回路により構成する。SFLには仮想内部端子“tmp”を用意し、この端子へのデータ転送は、転送元と同一であるとみなし、論理合成の段階を経て最終的には回路上に要素として存在しなくなる。仮想内部端子を利用することにより、設計者にとって機能記述の容易さを保ちつつ、論理圧縮により回路の合成を行うことができる。

2.5.2.2 依存解析部

先行命令とのデータ依存関係の有無を検出したり、スレッドやレジスタウィンドウの切り替え時に先行命令の実行終了の確認を行ったりするためにスコアボードを実現する (Fig. 2.10)。

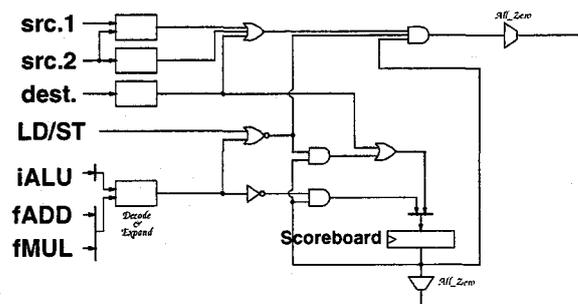


Fig. 2.10 依存解析部の構成

記述は、読み出し/書き込み/書き込み予約の各レジスタ番号をデコードし、依存関係を調べ

る組合せ回路が大部分を占める。他に、各レジスタ毎のスコアボードレジスタを設けており、解読部から命令が発行される際、書き込みレジスタに対応するスコアボードを設定する。

2.5.2.3 割り込み検出部

外部割り込み、ならびに内部割り込みを検出し、割り込みが発生した事実と最も優先順位の高い割り込みの処理アドレスとを命令シーケンス制御部に伝達する (Fig. 2.11)。記述は、上記の機能を実現する簡単な組合せ回路である。

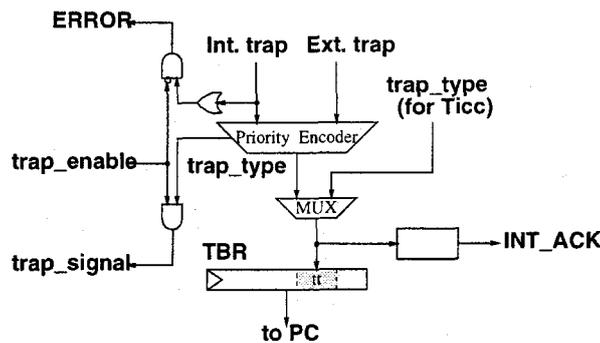


Fig. 2.11 割り込み検出部の構成

2.5.2.4 命令シーケンス制御部

解読部全体を制御する部分で、プログラムカウンタの制御、分岐命令の実行、ユニット外部とのインタフェースの各機能を受け持つ。本プロセッサでは、分岐命令のアドレス生成を命令パイプラインの解読段階で行うが、レジスタオペランドに相對した分岐など、処理に複数サイクルを要する命令に関しては、後続命令の解読を抑止し、解読部の内部状態を遷移させることにより実行する。このような分岐命令や割り込み処理と対応して、内部状態の遷移制御と各状態に応じた制御を行うために SFL の状態遷移記述を用いる。採用した命令の中で、解読部での実行に複数サイクルを要するものは、(1) JMPL (Jump and Link), (2) RETT (Return from Trap), (3) Ticc (Trap on integer condition codes) の 3 種類である。割り込み可能な状態において、割り込みを検出した場合には、割り込み処理を行う状態に遷移する (Fig. 2.12)。

各状態ではそれぞれの場合のプログラムカウンタの制御、プリフェッチバッファへのフェッチ要求、発行命令の有効化/無効化等を行うが、状態レジスタの最適な割当をはじめ、バス制御や冗長な状態の削除を PARTHENON の論理合成が受け持つため、複雑な回路をスキマティックに設計することなく、SFL による簡単な記述だけで論理合成を行いネットリストを生成することができる。

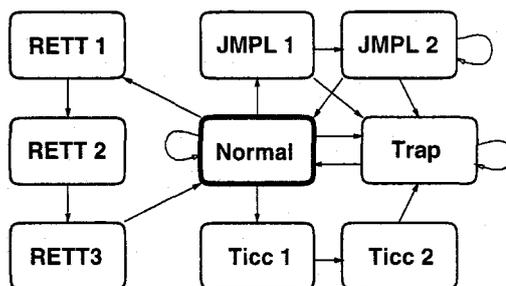


Fig. 2.12 命令シーケンス制御部の状態遷移図

2.5.3 命令選択ユニット

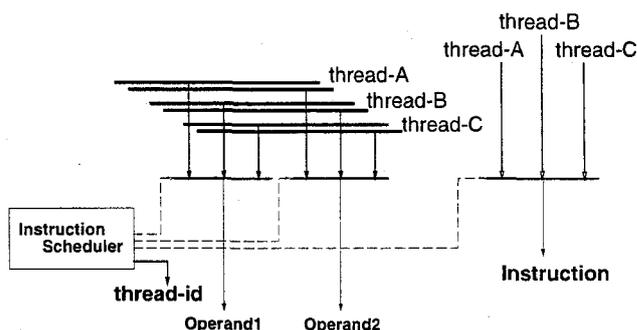


Fig. 2.13 命令選択ユニットのブロック図

各機能ユニット毎に設けられた命令選択ユニットは、複数スレッドからの機能ユニット使用要求に対して、調停を行う (Fig. 2.13). 調停は、巡回する優先度に従って行い、調停によって選択された一つの命令を機能ユニットに送り、各解読ユニットに選択/非選択信号を送る. SFL による記述では、Fig. 2.13 に見られるバスは単なる代入文として記述するのみであり、論理合成により自動的にデータバスを生成する.

2.5.4 実行ユニット

実行ユニットは、整数演算、ロード/ストア、浮動小数点加算、浮動小数点乗算の4機能ユニットからなる (Fig. 2.14). 個々のユニットは独立して動作し、個別の命令を実行することができる. また、命令選択ユニットにより、機能ユニットの使用要求を調整しているため、あるスレッドが解読ステージ等でパイプラインストールを起こした場合でも、実行ユニットにその影響は現れない. これは、プロセッサ全体では各スレッドのパイプラインストールを隠蔽していることになり、各機能ユニットの稼働率は向上することに起因する. それぞれのユニットは、レジスタ書き込みを行う直前のサイクルで、レジスタユニットへ書き込み要求信号を出す. レ

ジスタユニット内ではこの要求を調停する。これに敗れた機能ユニットはパイプラインストールを起こし、次サイクルに再び書き込み要求を出す。

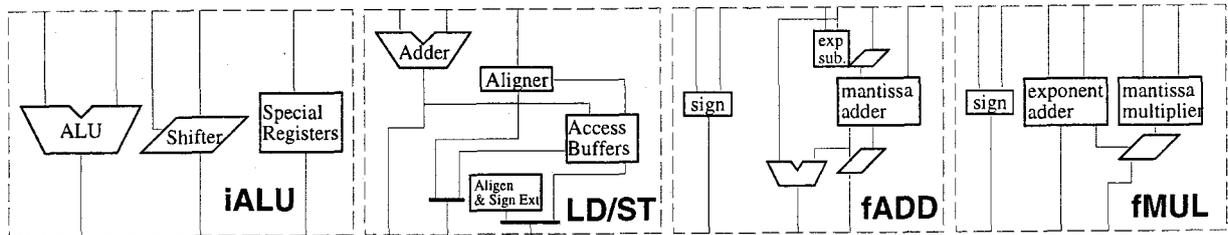


Fig. 2.14 実行ユニットのブロック図

機能ユニットに外部のより高性能なマクロセルやデータパスを利用する場合、論理合成の対象から外して設計を行う。その場合、論理合成の対象から外す機能ユニットに対しても機能記述を行うことで、動作シミュレーションにかけることができる。

2.5.4.1 整数演算ユニット

整数算術論理演算、シフト演算、制御レジスタ命令、レジスタウィンドウ制御を1サイクルで処理する。セルライブラリのデータパスを使用し、その入出力を最外郭モジュールで制御するため、SFLではALU、バレルシフタ、制御レジスタを論理合成の対象から外して設計を行う。この他の、データパスおよびレジスタウィンドウの制御部は論理合成によって実現する。

2.5.4.2 ロード/ストアユニット

データキャッシュメモリにアクセスする命令を処理する。3段のパイプラインで構成されており、倍精度のロード/ストア命令や単精度のストア命令の実行時には内部命令を発行する。記述はサブモジュールとして、パイプライン段階に対応した3つのモジュールを備えており、それぞれアドレス計算、メモリアクセス、データ整列等を行う機能を記述している。キャッシュミスを起こした場合には、ロード/ストアユニットは状態遷移を行い、外部信号がキャッシュヒットを知らせるまでパイプラインをストールさせる。

2.5.4.3 浮動小数点加算/浮動小数点乗算ユニット

これらのユニットは、3段のパイプライン構成を取っており、除算、開平、倍精度への型変換以外の演算は、全て3サイクルで演算を終了する。すなわち、ほとんど全ての浮動小数点演算を、毎サイクル連続して処理することができる。また、出現頻度が高い浮動小数点乗算専用に乗算器を設けており、他の浮動小数点演算と同時に処理を行うことができるため、性能の向

上を達成できる。実際には、本ユニットのほとんど全ての部分はデータパスライブラリを用いて実現する。

2.5.5 レジスタユニット

レジスタ部は、各スレッドスロット毎に設ける。これは整数レジスタ、浮動小数点レジスタ、書き込みアービタからなる。整数レジスタについては、レジスタウインドウ構造 (Fig. 2.15) を有しており^[20]、3つのウインドウを持つ。整数、浮動小数点レジスタとも一度に使用できるレジスタの個数は32である。書き込みアービタは複数の機能ユニットからの書き込み要求を調

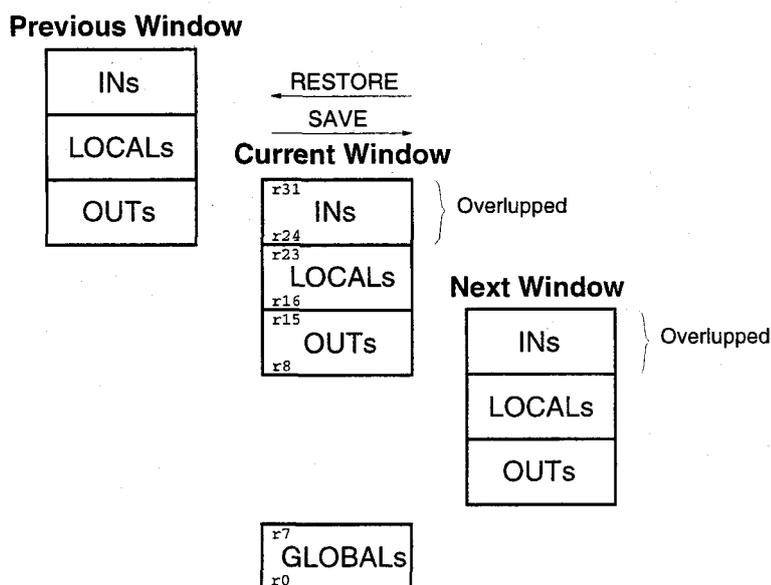


Fig. 2.15 整数レジスタウインドウ

停する。最も頻度が高いロード/ストアユニットからの要求を優先し、整数演算、浮動小数点加算/乗算ユニットは、要求が重なった場合には書き込みを延期する。レジスタファイル部分はマルチポートレジスタとして論理合成の対象から外し、ウインドウの制御、機能ユニットからの書き込み要求の調停等の論理を、合成対象として記述する。

2.6 実装結果

設計した多重スレッドプロセッサの機能検証、ならびに論理合成を PARTHENON を用いて行った。各モジュールに対して動作シミュレータ SECONDS 上で入力を設定し、機能シミュレーションを行った。解読部の状態遷移、割り込み検出等の動作を設計通りに記述できることを確認した。ここで、プロセッサ外部とのインタフェースに関しては、仮想的なキャッシュメモリまでを SFL で設計し、機能シミュレーションを行った。

全体の工程数は、PARTHENON、多重スレッドアーキテクチャの習得の段階を含め、論理合成まで 11 人月であった。特に、シミュレーション段階で様々な設計誤りを発見したり、高性能化のための仕様を変更したが、ここでの修正は SFL に変更を加えるだけであったため、短期間で実現できた。

各モジュールの動作記述ならびに $1.0\mu\text{m}$ スタンドセル、データパスライブラリを使用して論理合成した結果を Table 2.3 に示す。命令選択ユニット等の複数個を必要とするユニットに関しては、1 スレッド当たりに必要な記述量および面積を示している。

Table 2.3 実験結果

ユニット名	記述量 (Steps)	面積 (mm^2)
命令フェッチ	700	0.67
命令解読	2764	20.53
命令選択	374	0.66
レジスタ	388	15.34
整数演算	1159	3.12
ロード/ストア	1721	4.58
浮動小数点加算	1590	5.47
浮動小数点乗算	569	12.03

2.7 結言

ビデオ画像生成システムの性能向上とコスト削減を達成することができる多重スレッドプロセッサを、高位合成システム PARTHENON を用いて設計した。高位合成システムの利用により、複雑な構造を持つ多重スレッドプロセッサを短期間で設計誤りのなく構築できることを示した。

第3章

HDTV レベル MPEG2 デコーダ

3.1 緒言

本章では、ビデオ動画像圧縮符号化アルゴリズムの国際標準 MPEG2 の規定のうち、高品位画像を扱ういわゆる MP@HL 動画像を実時間で復号化するための1チップ VLSI 化設計について記述する。従来の DSP コアと専用演算器を用いた構成ではなく、MPEG2 の復号化の各処理過程を実行するために新たに考案した専用アーキテクチャを活用することによって、1チップデコーダが実現可能であることを示す。具体的には、復号化の過程において多大な計算量を必要とする逆離散コサイン変換、動き補償をはじめ、並列処理による高速化が極めて困難な可変長符号復号化や、逆量子化等の各処理についても専用機能ユニットを構築する。そして、各パイプラインステージ間に、設計した専用機能ユニットの処理能力を最大限に引き出すことのできるバッファ容量と構成について考慮し、これを構築する。さらに、データ転送の帯域を大幅に増加させることのできるフレームメモリ構成と、読み出した画素データの再利用を可能とするバッファを保持したフレームメモリインタフェースを構築することで、メモリアクセス数を大幅に削減する。各専用機能ユニット、及びパイプラインバッファの機能を、トップダウン VLSI 設計システム *COMPASS Design Navigator* を用いて設計することで、HDTV レベルの動画像を実時間で処理する能力を持つ MPEG2 デコーダを実装する。

3.2 MPEG2 圧縮符号化アルゴリズム

ビデオ動画像は、空間的に広がりを持つ2次元の静止画像が、時間的に連続しているものとみなすことができる。このため、動画像情報を圧縮符号化する手法としては、まず、空間的な情報を圧縮する手法と、時間的な情報を圧縮する手法が考えられる。前者については従来から静止画像の圧縮符号化として研究されており、多くの優れた手法が提案されている。後者の時間的な情報を圧縮する技法は、動画像において連続する画像が互いに相関関係が高いという性質

を利用している。このように動画像の圧縮には、空間軸、時間軸の双方の情報を圧縮するハイブリッド圧縮法が有効であると考えられており、MPEG2はこのハイブリッド圧縮法の1手法として位置づけられる。

MPEG2の画像圧縮技術は以下の3種の圧縮技法に分類することができる。

1. 時間的冗長性を排除する動き予測 (ME: Motion Estimation)
2. 空間的冗長性を排除する離散コサイン変換 (DCT: Discrete Cosine Transform) と量子化 (Q: Quantisation)
3. 符号の出現確率を利用して平均ビット長を短縮する可変長符号化 (VLC: Variable Length Coding)

これらの過程を経て生成された符号は以下の過程を経て元の画像情報に復号化される。

1. 可変長符号復号化 (VLD: Variable Length Decoding)
2. 逆量子化 (IQ: Inverse Quantisation)
3. 逆離散コサイン変換 (IDCT: Inverse Discrete Cosine Transform)
4. 動き補償 (MC: Motion Compensation)

Fig. 3.1 に、MPEG2 符号化/復号化アルゴリズムの流れを示す。

ME/MCの過程は、 16×16 画素サイズのマクロブロック単位で処理を行うが、それ以外の過程は 8×8 画素のブロック単位で行う。MP@HL(4:2:0 フォーマット)では、1マクロブロック当たり、輝度信号4ブロックと色差信号2ブロックの合計6ブロックから成り立っている。このため1マクロブロックでは、256の輝度信号と128の色差信号の計384の画素からなる。以下、MPEG2の最も重要な圧縮技法となっている3種類の処理過程について、その圧縮の原理と手法について述べる。

3.2.1 動き予測

ビデオ動画像を、時間的に変化する静止画の系列であると見なしたとき、近隣の静止画の間では相関関係が高くなるという性質がある。これを時間的冗長性といい、動き予測を用いてこの冗長性を排除する。

MPEG2の動き予測は、 16×16 画素サイズのマクロブロック単位毎に、いわゆるブロックマッチング法で行う (Fig. 3.2)。ブロックマッチング法は予測画像において、各マクロブロック

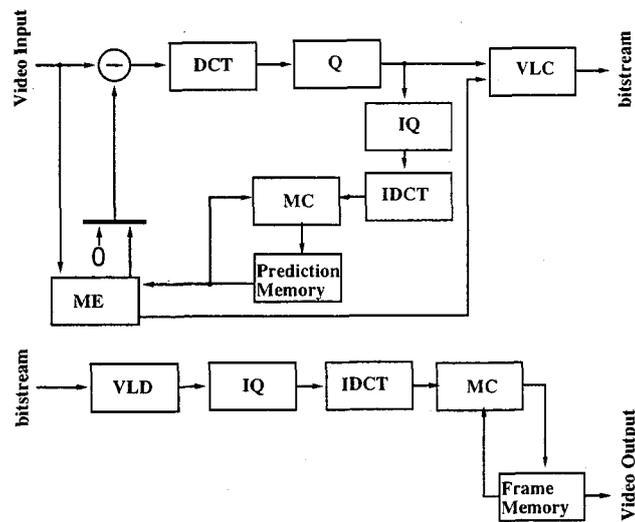


Fig. 3.1 MPEG2 アルゴリズムによる符号化/復号化

毎に、参照画像の中から最も差分の少ないマクロブロックを探し出し、マクロブロックの位置関係を動きベクトルとして検出するものである。予測の結果得られる予測マクロブロックと参照マクロブロック間の差分を予測誤差と呼ぶ。予測画像に対しては、この予測誤差と動きベクトルを符号化するだけでよいため、全体として画像情報量が圧縮される。

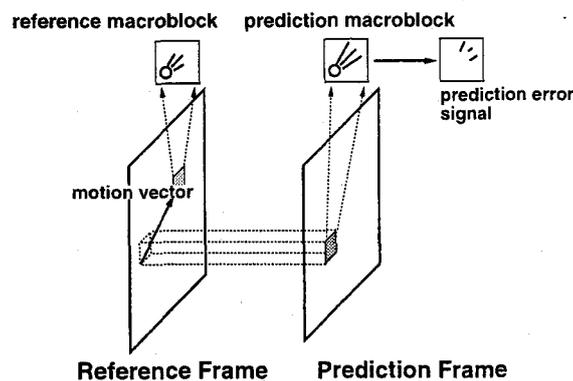


Fig. 3.2 ブロックマッチング法による動き補償

MPEG2 では、時間軸上の前方向の参照画像からの予測だけでなく、後方向や、前後双方向の参照画像からも予測を行うため、画面の切替え時においても情報を圧縮することができ、また予測精度を向上させることもできる。動き予測により画像情報量は 1/2 程度に圧縮される。動き予測により符号化された画像データは効率的に情報量が圧縮されているが 1 画面分だけでは

完結した画像として復元することができない。一方、蓄積や放送メディアではランダムアクセスの必要性が存在することから、MPEG2 ではランダムアクセスの単位として、GOP (Group of Pictures) を定めている。

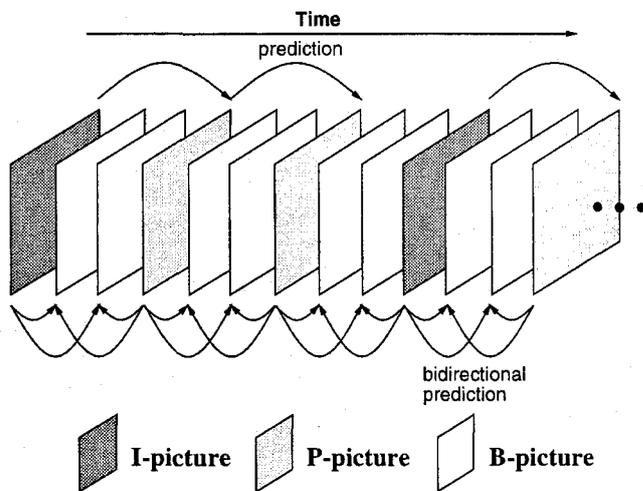


Fig. 3.3 ピクチャ・タイプ

GOP は、I ピクチャ(イントラ符号化画像)、P ピクチャ(順方向予測画像)、B ピクチャ(双方向予測画像)の3種類の画像タイプを持つことができる (Fig. 3.3)。GOP は全く予測を行わずに符号化された画像である I ピクチャを最初に 1 画面だけ保持している。I ピクチャは予測を行っていないため、他の画像とは独立して復号化することができる。この画像を参照画像にして GOP 単位で他の予測画像を復号化する。P ピクチャは最も近い過去の I または P ピクチャを基に動き予測を用いて符号化された画像である。復号化された P ピクチャもまた他の予測画像の参照画像となる。B ピクチャは最も近い前後の I、P ピクチャ、またはその両方を参照画像として、それぞれ動き予測を行い符号化された画像である。B ピクチャは最も高い圧縮率を得ることができるが、他の画像の参照画像としては使用されない。

このように、動き予測の処理は I ピクチャでは行わず、P、B ピクチャのみで行う。また、P、B ピクチャの中のマクロブロック単位では、動き予測を行わずに符号化するイントラ符号化 (intra coding) を選択することができる。通常の動き予測を行う符号化を非イントラ符号化 (non-intra coding) と呼ぶ。I ピクチャは全てイントラ符号化である。

3.2.2 離散コサイン変換と量子化

通常、自然画像 1 画面内での近隣の画素は緩やかに変化しているために相関関係が高い。この性質は空間的冗長性と呼ばれており、離散コサイン変換はこの冗長性を排除する最も有力な

圧縮技法の一つとして知られている。離散コサイン変換は、画素情報を空間周波数領域の信号である DCT 係数に変換する。Fig. 3.4 に離散コサイン変換を行った場合の DCT 係数の分布例を示す。通常、自然画像の場合、変換前の画像では画素情報がブロック内に一様に分布しているのに対し、変換後は低周波数成分の DCT 係数にエネルギーが集中する。このため量子化によって、エネルギーが小さく視覚的に影響の小さい高周波数成分の情報を落とすことで、画像情報が圧縮される。MPEG2 では、離散コサイン変換や量子化を 8×8 画素のブロック単位で行っている。

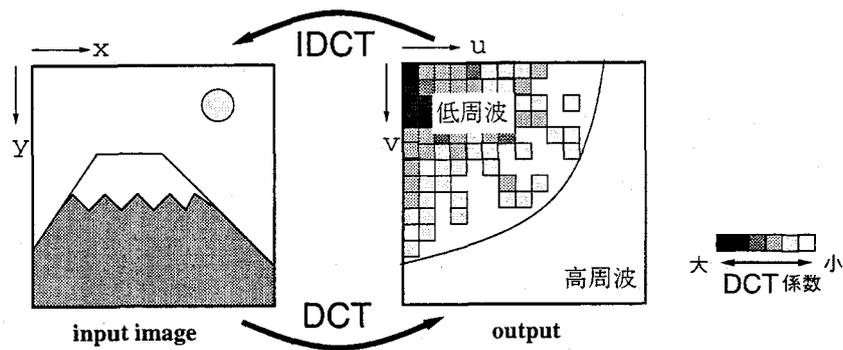


Fig. 3.4 離散コサイン変換の変換例

量子化の過程を Fig. 3.5 に模式的に示す。量子化の過程では、 8×8 ブロックの DCT 係数 (A_{ij}) を量子化ステップ (Q_{ij} : 量子化特性値 \times 量子化マトリックス) で除算し、ブロックの各値を量子化代表値 (B_{ij} : 量子化 DCT 係数) に変換することで画像情報量の圧縮を行っている (ただし $(i, j = 0, 1, \dots, 7)$)。この際、高周波数成分の DCT 係数は “0” になることが多く、さらに圧縮を図ることができる。

量子化マトリックスは、ブロック内の量子化ステップサイズを周波数成分毎に規定するために用いられ、2 種類の初期設定値と符号化側で設定したマトリックスを使用することができる。量子化特性値は、量子化マトリックス全体のステップサイズを規定するために用いられる。量子化特性値を変更して符号化することで、ピクチャ毎の量子化精度を可変にすることができる。量子化マトリックスに使われる 2 種類の初期設定値を Fig. 3.6 に示した。Fig. 3.6(a) は、イントラブロック用、Fig. 3.6(b) は、非イントラブロック用の量子化マトリックスである。このように MPEG2 の量子化は、視覚特性を利用して高周波数成分の精度をさらに落とし、圧縮効率を高めることができる。

以上のような量子化の制御により符号化側では、量子化による画像の劣化を最小限にとどめたり、情報量をできる限り小さくしたりといった符号化特性の選択が可能となる。

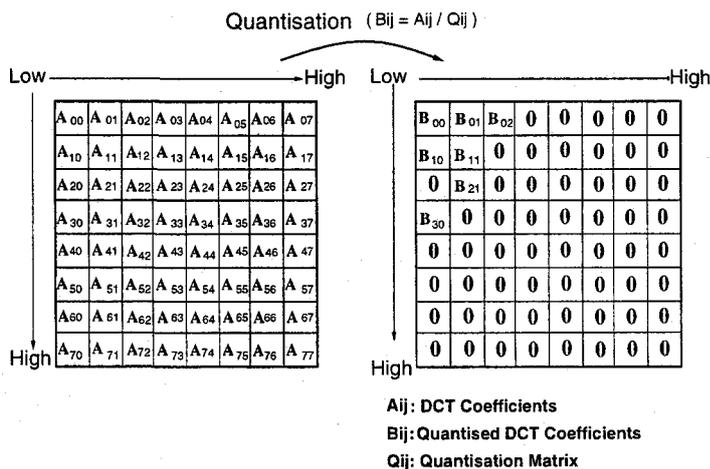


Fig. 3.5 量子化による DCT 係数の変換

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

(a) for intra blocks

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

(b) for non-intra blocks

Fig. 3.6 量子化マトリックスの初期設定値

一連の離散コサイン変換，量子化の処理を入力画像そのものや予測誤差に対して行い，可変長符号化の過程に送られる．これらの処理によって画像情報量は 1/10 ~ 1/20 に圧縮される．

3.2.3 可変長符号化

可変長符号化では，エントロピー符号化の一種であるハフマン符号化を行っている．ハフマン符号化は，符号の出現確率を基に符号の平均語長を最小にすることができる最適な木符号である．MPEG2 の可変長符号化では，動きベクトルや量子化 DCT 係数の各想定発生確率に従って規定された符号表を用いて符号化し，平均符号ビット長を短縮することで情報の圧縮を達成する．可変長符号化によって画像情報量は 2/3 ~ 1/2 に圧縮される．

特に DCT 係数の可変長符号化では，量子化 DCT 係数の分布特性を利用してさらに情報量

を圧縮している。DCT 係数は量子化を行うと高周波数成分の値が“0”となる頻度が極めて高い。そこで、Fig. 3.7 に示すようにジグザグに走査して量子化 DCT 係数を取り出していくと、高周波数成分の領域では“0”の連続(ラン)が出現する。MPEG2 では、この先行するゼロ係数の個数(ランレングス)とこれに続く非ゼロの量子化係数の値(レベル)をまとめて符号化を行う、いわゆるラン・レベル 2 次元可変長符号化を採用している。

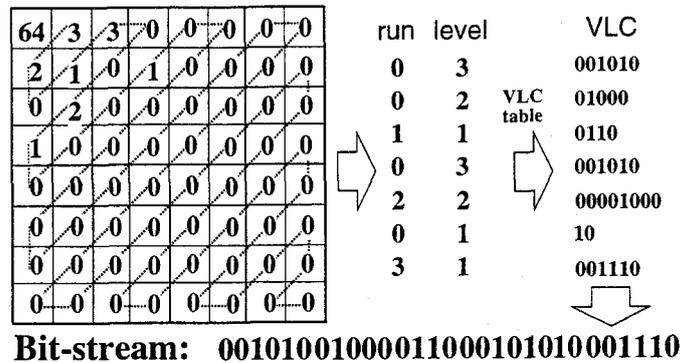


Fig. 3.7 ラン・レベル符号化

MPEG2 では動き補償、離散コサイン変換、量子化、可変長符号化といった圧縮技術を用いたハイブリッド符号化により、全体では動画像のデジタル情報を 1/30 ~ 1/80 程度に圧縮している。

3.3 デコーダアーキテクチャの設計

MPEG2 の符号化データを復号化する手法としては、

- (1) 汎用プロセッサを用いてソフトウェアで行う手法、
- (2) プログラマブルな DSP(Digital Signal Processor) を用いて行う手法、
- (3) 専用機能ユニットを用いて行う手法、

が考えられる。

3.3.1 汎用プロセッサによる復号化

(1) はソフトウェアで復号化するために、特に新たなプロセッサ・アーキテクチャを設計する必要がない。MPEG2 の符号化を行う場合や他の画像圧縮アルゴリズムを復号化する場合でも

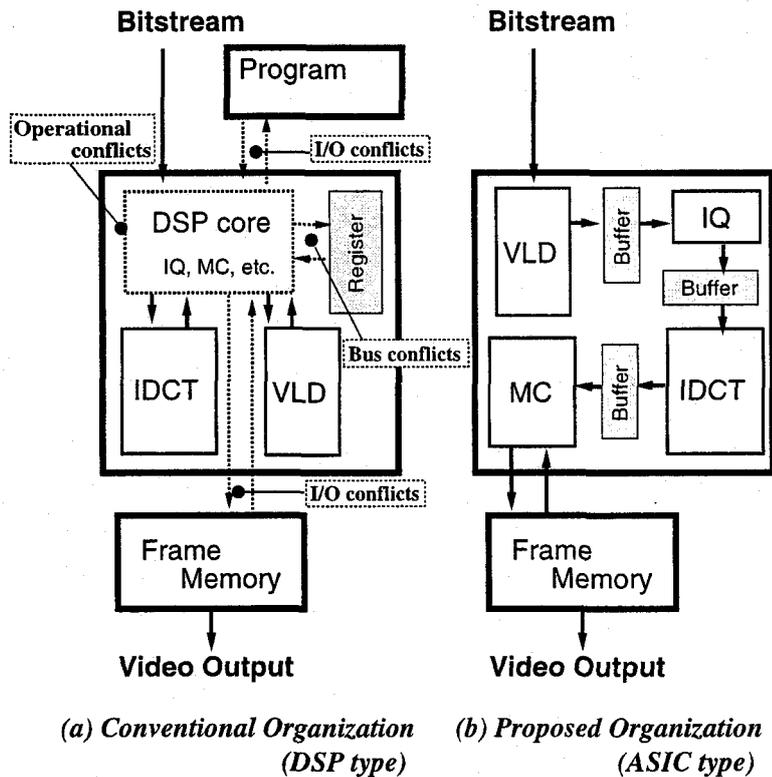


Fig. 3.8 MPEG2 デコーダの実装手法

ソフトウェアを変更/作成するだけで良い。しかしながら現在の汎用プロセッサの処理性能や通常のメモリアクセス機構の制限がボトルネックとなり、MPEG2 における MP@ML 程度の解像度の動画像を実時間で復元する性能にも至っていない。

3.3.2 プログラマブル DSP による復号化

(2) の手法で実現した場合の構成のブロック図を Fig. 3.8(a) に示す。DSP は、画像データアクセスを高速化するための入出力データ・メモリアドレス制御機構や、DSP コアと並列に利用可能な演算ユニットを保持することで、処理全体の高速化を図っている。また制御ソフトウェアやアルゴリズムを変更することで、様々な画像圧縮アルゴリズムや符号化にも比較的容易に対応させることができる。実際、多くの MPEG1/MPEG2 処理用 VLSI プロセッサが、この手法を用いて設計されている^[5, 6, 22, 23]。これらの DSP の多くは演算ユニット部である DSP コアの他に、離散コサイン変換等のように、演算量が多いが定型的な演算を行えばよい処理に対して、専用機能ユニットを設けた構成をとっている。しかしながら、設計した DSP の性能を十分に引き出せるコンパイラの構築などが非常に困難であるとともに、DSP コアでの各処理のコンフ

リクト，処理間でのメモリ入出力のコンフリクト，さらにはレジスタ入出力時等の内部バスのコンフリクト等による性能低下が引き起こされる (Fig. 3.8(a)). 従って，MPEG2 の MP@ML を実時間で復号化は可能であっても，MP@HL を実時間で復号化する処理性能を持ち合わせていない。

3.3.3 専用機能ユニットによる復号化

(3) による手法では復号化の各過程に対し，専用の機能ユニットをマクロセルで実現し処理を行う (Fig. 3.8(b)). この手法を用いて設計したデコーダは，他の画像圧縮アルゴリズムや符号化に対応させることは難しい。しかしながらこの手法は，メモリ入出力のコンフリクトや機能ユニットのコンフリクトが軽減されるため処理能力が優れており，復号化のように処理内容が限定されている場合では，最も処理性能が高い。また大量の需要が見込まれる応用分野のデコーダとして適した実現方法である。

3.3.4 MP@HL に適したデコーダアーキテクチャ

Table 3.1 に，MPEG2 のメインプロファイルにおける各レベルの制約を示す。MP@HL の動

Table 3.1 メインプロファイルの各レベルにおける制約

picture "level"		MP@LL	MP@ML	MP@H1440	MP@HL
maximum	(pels/line)	352	720	1440	1920
sampling	(lines/frame)	288	576	1152	1152
density	(frames/s)	30	30	60	60
luminance pel rate (samples/s)		3,041,280	10,368,000	47,001,600	62,668,800
macroblock rate (macroblocks/s)		11,880	40,500	183,600	244,800
allowable time (μ s)		84.2	24.7	5.45	4.08

画像符号は MP@ML 復号化処理の約 6 倍の処理能力を要求するため，既存の MP@ML の処理性能しか持たないプロセッサでは，1 チップによる実時間での復号化を行うことができない。また，本デコーダの復号対象は MPEG1/MPEG2 のビットストリームに限定しているためプログラマブルなプロセッサを構成しなくてもよい。そこで (3) の手法，すなわち復号化の過程を全て専用機能ユニットにより処理するアーキテクチャを設計することで，HDTV レベルの膨大な演算量を 1 チップで処理可能なデコーダの実現を図る。具体的には可変長符号復号化，逆量子化，逆離散コサイン変換，動き補償に対して専用機能ユニットを実現し，各処理をマクロ

ロックレベルのパイプラインステージに対応させる。そして、各機能ユニットを適切な容量と入出力機能を持ったバッファで接続し、パイプラインを構成する。このため、それぞれの機能ユニットの1マクロブロック当たりの処理時間を、Table 3.1 に示す MP@HL の許容時間内におさめることで、HDTV レベルの動画像を実時間で復号化するデコーダを構成することができる。また、本デコーダの処理能力を著しく低下させる要因となるメモリアクセスの回数を削減する構成についても考慮している。

3.4 HDTV レベル MPEG2 デコーダの設計

3.4.1 可変長符号復号化器

3.4.1.1 可変長符号復号化の従来手法

従来の可変長符号復号化器は、ルックアップテーブルを利用した手法 (Fig. 3.9) が挙げられる^[24, 25]。可変長符号復号化器は、ルックアップテーブル (PLA または ROM) と、語長検出機

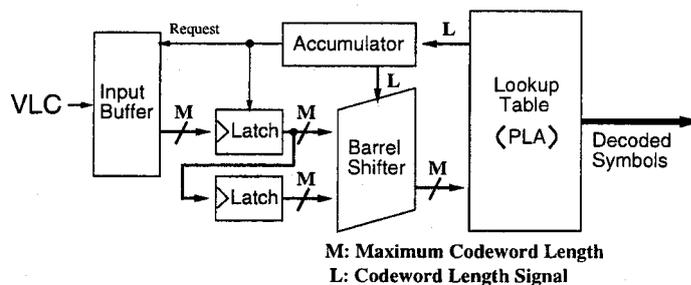


Fig. 3.9 従来の可変長符号復号化器の構成

構を組み合わせた回路によって構成することができる。ルックアップテーブルは、可変長符号をアドレスの入力とし、復号化されたデータを出力とする。アドレスは最大符号長の可変長符号を入力することができるように構成する。また、語長検出機構により復号化した可変長符号の符号長を検出し、シフタを用いて次の可変長符号の入力を準備する。PLA による可変長符号復号化器は、全ての可変長符号に対する符号のルックアップテーブルを保持しなければならないため、復号化しなければならない可変長符号の種類が膨大になると、ルックアップテーブルの回路規模が大きくなるばかりでなく、符号の読み出し時間の遅延も長くなる。これに伴って、復号化した符号の符号長検出から次の可変長符号をシフトするまでの経路が、クリティカルパスとなり、動作周波数の低下を引き起こす。

以上のように従来のハフマン符号復号化器は MP@HL デコーダの専用機能ユニットには適さない課題が残されている。

3.4.1.2 MP@HL 対応・可変長符号復号化器の設計

MPEG2 のラン・レベル符号化では 1 符号で複数の係数の情報を含むものから、符号と係数の個数が 1 対 1 に対応するものまで存在する。このため、1 マクロブロックの量子化 DCT 係数を符号化した際の符号長は、符号化が行われる画像データに大きく依存する。最悪の場合には 1 マクロブロック当たり 384 個ものラン・レベル符号をとる恐れがあり、可変長符号復号化器としてはこの場合でも正常動作を保証しなければならず、極めて多くの演算量が必要となる。MPEG2 の可変長符号復号化で行う量子化 DCT 係数に関する復号化処理は、入力ビットストリーム中の可変長符号からラン・レベル符号の復号化と、ラン・レベル符号からの量子化 DCT 係数の復号化に分類することができる。前者が、従来のハフマン符号復号化器と同様の機能を実現するものである。

本過程では復号化の対象となる符号が可変長であるため、前の符号の解釈が終了するまで次の符号の開始位置が定まらず、並列化による高速化は極めて困難である。このような観点から、これまで可変長符号復号化に対する様々なアーキテクチャが提案されている^[26,27,28]。

本節では、ルックアップテーブルを用いたアーキテクチャを基本構成に利用しながら、ルックアップテーブルの回路規模の縮小と符号生成時間の高速化を達成し、さらに、可変長符号復号化器のパイプライン化を実現することで符号長検出にともなうクリティカルパスを改善する構成を考案した。本節で構築する可変長符号復号化器の構成を Fig. 3.10 に示す。これまでシフタの入力から符号の復号化まで 1 段で行われていた処理を、3 段のパイプラインステージに分けて処理する。

第 1 ステージでは、まず符号中に連続する“0”列の検出を #0 検出器で行い、次の可変長符号の準備のみを行う。#0 検出器では、可変長符号を *Class I* と *Class II* に分類して処理する。*Class I* の可変長符号は、Fig. 3.11(a) に示すように先行する“0”の個数から符号のビット長を比較的容易に求めることができる。*Class II* は、Fig. 3.11(b) に示すように、連続する“0”とそれに続く数ビットで符号長を求めることができる。この特性を利用して第 1 ステージでは、次符号の開始位置の検出だけを入力ビットストリーム中の可変長符号から行う。設計した #0 検出器により次の可変長符号の準備を、実際のラン・レベルの符号の生成とは分離して行えるため、ルックアップテーブルの出力を待たずに次符号の処理を開始することができる。この構成により可変長符号の符号長検出に関するクリティカルパスが改善される。従来の構成でラン・レベル符号の復号化を行った後で次符号を用意するために 18.9ns あったクリティカルパスが、9.6ns に改善された。

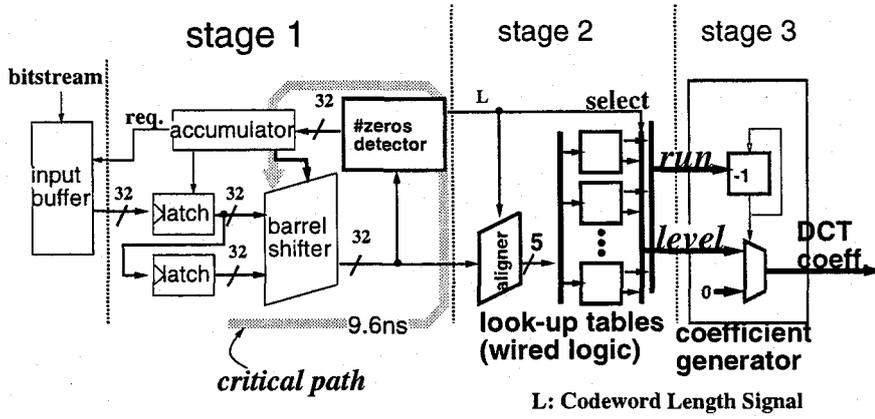


Fig. 3.10 可変長符号復号化器の構成

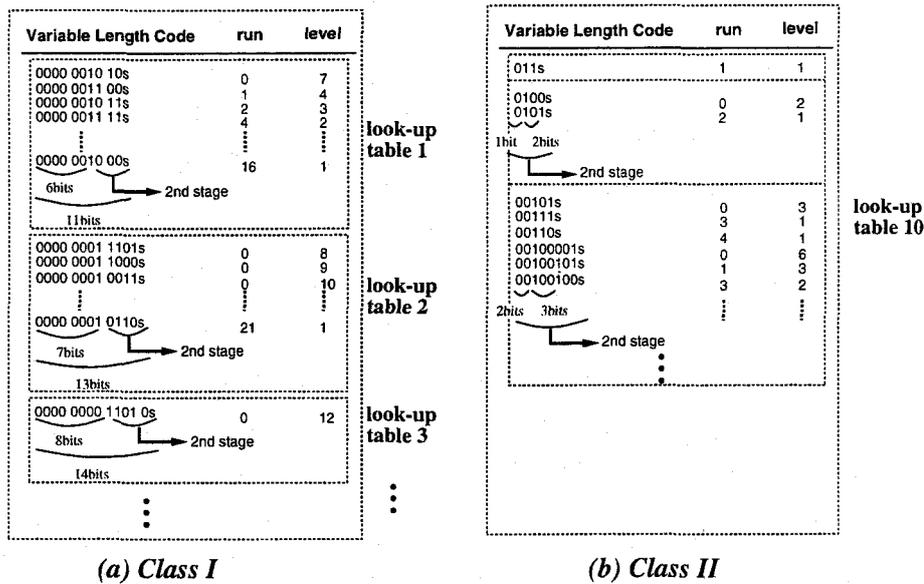


Fig. 3.11 MPEG2 の可変長符号ルックアップテーブル (抜粋)

第2ステージでは、第1ステージで得られた可変長符号と符号長情報からラン符号とレベル符号を生成する。Fig. 3.11に示すように、符号長毎にルックアップテーブルを分離して保有することにより、各ルックアップテーブルにおける入力ビット数を削減している。このため、従来のような可変長符号によるROMのダイレクトアクセスやPLAにより実現するのではなく、専用ロジック回路による実現が可能となり、面積の削減と、ラン・レベル復号化の遅延時間を改善した。また、MPEG2では量子化DCT係数用の可変長符号テーブルとして、非イントラ・マクロブロック(MPEG1互換)テーブルとイントラ・マクロブロック専用テーブルの2種類が規定されているが、上記のような“0”列の検出方式とルックアップテーブルの分離により、この2種類のルックアップテーブルの処理の半分以上を共有することができる。

第3ステージでは、第2ステージで得られたラン符号とレベル符号を元に、係数生成器が量子化DCT係数を生成する。

設計した可変長符号復号化器は、入力した可変長符号から1マクロブロック分の量子化DCT係数を384サイクルで出力することができる。

3.4.2 逆量子化器

3.4.2.1 逆量子化の従来手法

従来MPEG1/MPEG2の逆量子化過程は、内蔵するDSPコアを使用して処理を行うプロセッサが多かった^[6, 22, 23]。しかしながらこのような構成では、MP@HLのような大量の画素データを扱う場合、メモリアクセスやDSPコアに接続する内部バスがボトルネックとなるばかりでなく、他のMPEG2の処理過程も同じDSPコアで行うため、逆量子化に許される演算時間内に処理を終えることは極めて難しい。

3.4.2.2 MP@HL 対応・逆量子化器の設計

MPEG2の逆量子化の過程ではまず、量子化されたDCT係数の各周波数成分と、量子化特性値、及び量子化マトリックスとの乗算を行う。この他に飽和处理、逆離散コサイン変換ミスマッチ問題に対する処理を行うことで、元のDCT係数を得る。以上の処理を行う逆量子化器の構成をFig. 3.12に示す。

逆量子化器は、まず、符号化データから量子化特性値を復号化し、量子化データとの間で1係数ずつ乗算を行う。得られた演算結果は、量子化マトリックス中の対応する値と乗算を行う。量子化マトリックスは、初期設定値を保持するROMと、符号化側で設定した値を持つRAMのいずれかの出力を選択することができる。この際、MPEG2の規定する2種類の逆スキャン方式(Fig. 3.13(a), (b))に従って、量子化マトリックスの値を出力するように逆スキャン制御

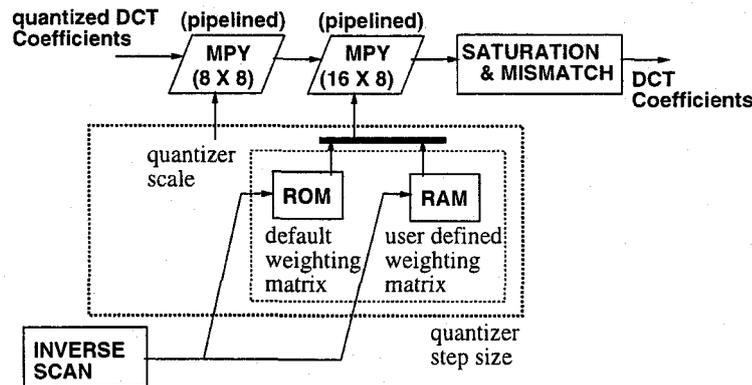
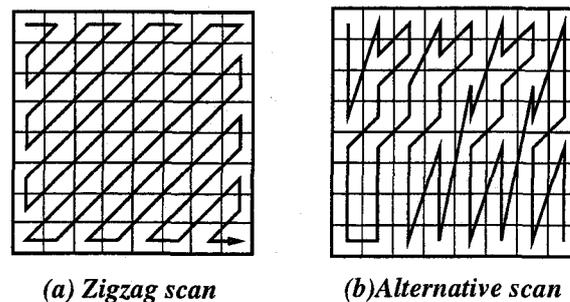


Fig. 3.12 逆量子化器の構成

部がアドレスを生成する。最後に乗算結果に対して、飽和処理、ミスマッチ制御を行い、DCT係数を出力する。乗算器にはパイプライン型並列乗算器を用いており、サイクル毎に乗算結果を出力することができる。このように、本逆量子化器では1係数を1サイクル毎に出力する構成のため、384サイクル毎に1マクロブロックが処理可能となる。



(a) Zigzag scan

(b) Alternative scan

Fig. 3.13 逆スキヤンの走査順序

逆量子化の過程に対して専用機能ユニットを設計することにより、他の復号化処理の演算とハードウェアの競合を引き起こすことなく処理を行うことができるため、HDTVレベルの演算量にも対応することができる。

3.4.3 逆離散コサイン変換器

3.4.3.1 2次元逆離散コサイン変換

2次元離散コサイン変換は画像圧縮技術の中で、空間的冗長性を減少させる最も効率的な変換の一つであり、画素領域のデータを空間周波数領域のDCT係数に変換するものである。逆

離散コサイン変換はこれまで H.261^[29]をはじめ, JPEG^[30](Joint Photographic coding Experts Group), MPEG1^[3]等に採用されており, その専用ハードウェアが実現されている^[22, 23, 27, 31].

MPEG2 では, 8×8 画素のブロック単位で逆離散コサイン変換を行っているが, 一般に $N \times N$ -2 次元逆離散コサイン変換は,

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) F(u, v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad (3.1)$$

と表せる. ここで $f(x, y)$ は画素のデータで, x, y ($x, y = 0, 1, \dots, N-1$) は画素領域での空間座標を表す. また $F(u, v)$ は DCT 係数で, u, v ($u, v = 0, 1, \dots, N-1$) は空間周波数領域での座標を表す. また,

$$C(u), C(v) = \begin{cases} 1/\sqrt{2} & (u, v = 0) \\ 1 & (\text{otherwise}) \end{cases}$$

と定義される. 式 (3.1) の 2 次元逆離散コサイン変換は,

$$f(x, y) = \sqrt{\frac{2}{N}} \sum_{v=0}^{N-1} C(v) \left\{ \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} C(u) F(u, v) \cos \frac{(2x+1)u\pi}{2N} \right\} \cos \frac{(2y+1)v\pi}{2N} \quad (3.2)$$

となり,

$$F'(x, v) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} C(u) F(u, v) \cos \frac{(2x+1)u\pi}{2N} \quad (3.3-a)$$

$$f(x, y) = \sqrt{\frac{2}{N}} \sum_{v=0}^{N-1} C(v) F'(x, v) \cos \frac{(2y+1)v\pi}{2N} \quad (3.3-b)$$

のように 2 つの $N \times 1$ -1 次元逆離散コサイン変換に分割できることが知られている. 式 (3.3-a) は, DCT 係数 $F(u, v)$ ブロックの各行に対し, 1 次元逆離散コサイン変換を行うもので, 中間データ $F'(x, v)$ に変換する. これに対して式 (3.3-b) は, 中間データブロックの各列に対し, 再度 1 次元逆離散コサイン変換を行うことにより, 画素データ $f(x, y)$ を復号化することができる. つまり, 2 次元 IDCT は, Fig. 3.14(a) に示す構成で実現することができる. ここで, 2 度の 1 次元逆離散コサイン変換は同一の変換であるため, 1 次元 IDCT ユニットの高速化し時分割利用することで, Fig. 3.14(b) に示す構成で実現可能である. Fig. 3.14(b) における 1 次元 IDCT ユニットの, Fig. 3.14(a) と比較して高速化により面積が増大し, また, 転置メモリの必要容量も増加する. しかしながら, 1 次元 IDCT ユニットの 1 ユニットのみのため, 実装結果より Fig. 3.14(b) の構成は Fig. 3.14(a) の構成と比較して面積を 80% に抑えることができる.

式 (3.3) より 8×1 -1 次元逆離散コサイン変換は,

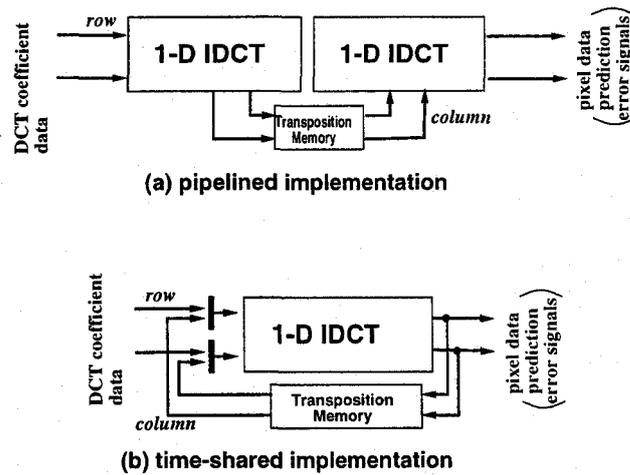


Fig. 3.14 2次元 IDCT の構成

$$x_v = \sum_{u=0}^7 C(u) X_u \cos \frac{(2v+1)u\pi}{16}; \quad (v = 0, 1, \dots, 7) \quad (3.4)$$

と表すことができる。しかしながら、式(3.4)のままでは1次元逆離散コサイン変換あたりの乗算回数が64回と多過ぎるため、VLSI化には適さない。このため1次元逆離散コサイン変換は、[22, 23, 27, 31]でも実現されているように、Chenのアルゴリズム^[32](バタフライ演算)と分散演算^[33]を採用して実現されたものが多い。

3.4.3.2 Chenのアルゴリズム

Chenのアルゴリズムは以下の方式により乗算の数を半減できることが知られており、式(3.4)で表される 8×1 -1次元逆離散コサイン変換は、

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{2} P X_o + \frac{1}{2} Q X_e \quad (3.5)$$

$$\begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \end{bmatrix} = \frac{1}{2} P X_o - \frac{1}{2} Q X_e \quad (3.6)$$

$$\begin{aligned}
 \mathbf{PX}_o &\triangleq \begin{bmatrix} x_{o0} \\ x_{o1} \\ x_{o2} \\ x_{o3} \end{bmatrix} = \begin{bmatrix} A & B & A & C \\ A & C & -A & -B \\ A & -C & -A & B \\ A & -B & A & -C \end{bmatrix} \begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{bmatrix} \\
 \mathbf{QX}_e &\triangleq \begin{bmatrix} x_{e0} \\ x_{e1} \\ x_{e2} \\ x_{e3} \end{bmatrix} = \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} \begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{bmatrix}
 \end{aligned}$$

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

となる。式(3.5), (3.6)は, \mathbf{PX}_o と \mathbf{QX}_e を加算するか減算するかの相違しかない。 \mathbf{PX}_o と \mathbf{QX}_e の結果は, それぞれ16回の乗算のみで得られるため, それぞれの出力を加算器, 減算器に入力することで, ハードウェア量を減少させるとともに, 1次元逆離散コサイン変換の乗算回数を32回に削減する。

3.4.3.3 分散演算

\mathbf{PX}_o と \mathbf{QX}_e の乗算回数をさらに減少させるものとして分散演算がある。逆離散コサイン変換における演算は \mathbf{PX}_o , \mathbf{QX}_e に見られるように, 入力データと定数との乗算と, その累算を行う積和演算である。分散演算(DA: Distributed Arithmetic)はこのように積算係数が固定されている積和演算において有効である。

\mathbf{PX}_o の $x_{oi}(i=0, 1, 2, 3)$, \mathbf{QX}_e の $x_{ej}(j=0, 1, 2, 3)$ の各行列の積和演算は,

$$x_{oi} = \sum_{u=0}^3 p_{iu} \cdot X_{2u+1}, \quad (3.7-a)$$

$$x_{ej} = \sum_{u=0}^3 q_{ju} \cdot X_{2u}, \quad (3.7-b)$$

のように表される。ただし, p_{iu}, q_{ju} は行列中の乗算係数である。ここで, 入力 X_ℓ は16ビットの2の補数表現であり, $X_\ell = -b_{\ell 0} + \sum_{n=1}^{15} b_{\ell n} \cdot 2^{-n} (\ell = 2u, 2u+1)$ と表せる。式(3.7)は,

$$x_{oi} = \sum_{u=0}^3 p_{iu} \cdot (-b_{(2u+1)0}) + \sum_{n=1}^{15} \left[\sum_{u=0}^3 p_{iu} \cdot b_{(2u+1)n} \right] \cdot 2^{-n}, \quad (3.8-a)$$

$$x_{ej} = \sum_{u=0}^3 q_{ju} \cdot (-b_{2u,0}) + \sum_{n=1}^{15} \left[\sum_{u=0}^3 q_{ju} \cdot b_{2u,n} \right] \cdot 2^{-n}, \quad (3.8-b)$$

となる。しかしながら、このままでは式 (3.8-a)/(3.8-b) を計算するのに積和演算を 16 回繰り返さなければならない。つまり (3.8-a)/(3.8-b) の演算には、

$$\sum_{u=0}^3 p_{iu} \cdot (-b_{(2u+1)0}), \quad \sum_{u=0}^3 p_{iu} \cdot b_{(2u+1)n} \quad (n = 1, 2, \dots, 15), \quad (3.9-a)$$

$$\sum_{u=0}^3 q_{ju} \cdot (-b_{2u,0}), \quad \sum_{u=0}^3 q_{ju} \cdot b_{2u,n} \quad (n = 1, 2, \dots, 15), \quad (3.9-b)$$

の乗算結果が必要となる。ここで、 p_{iu}, q_{ju} は定数であるために、(3.9-a)/(3.9-b) はそれぞれ、 b_{en} のみに依存する 16 個の異なる値しかとらない。従ってこれらの部分積の演算は、演算時間を短縮するために (3.9-a), (3.9-b) それぞれに対する 2 個の ROM で実現することができる。ROM には、あらかじめ乗算を行った結果をアドレス ($b_{1n}, b_{3n}, b_{5n}, b_{7n}$) または ($b_{0n}, b_{2n}, b_{4n}, b_{6n}$) から出力することができるように設定すれば良い。

以上により、2 個の ROM と、部分積を入力各ビット毎に 16 回順次累算していく 2 個のアクキュレータとにより式 (3.8-a)/(3.8-b) を演算する回路が実現できる (Fig. 3.15(a)). Fig. 3.15(a) では、式 (3.8-a)/(3.8-b) を演算する ROM とアクキュレータのうちの 1 組の積和乗算器 (例えば式 (3.8-a) に対応する部分) のみを表している。この構成では、積和演算器を並列乗算器を用いず、ROM、シフタ、累算器だけで実現できるためハードウェア量を大幅に削減するばかりでなく、積和乗算器の並列化によりさらに高速化を実現することができる。

しかしながら、Fig. 3.15(a) の構成では式 (3.8-a)/(3.8-b) を 1 回演算するのに、入力ビット毎に 16 回繰り返し積和演算を行わなければならない。このため、まず、 PX_o と QX_e の各 8 個の要素を並列に演算するために積和演算器を 8 ユニット実装しパイプライン化した構成をとったとしても、1 マクロブロックの処理に 1,536 サイクルを要する。このままでは、HDTV レベルの画像の実時間処理を行うために許される 1 マクロブロック当たりの演算時間には逆離散コサイン変換の処理を終えることが不可能となる。

3.4.3.4 MP@HL 対応・逆離散コサイン変換器の設計

マルチビット化

Fig. 3.15(a) のような構成の積和演算器では、MP@HL の動画像を実時間で処理することはできないため、式 (3.8) を 4 ビット分の並列入力を可能にするような次式のように拡張した (Fig. 3.15(b)).

$$x_{oi} = \sum_{u=0}^3 p_{iu} \cdot (-b_{(2u+1)0}) + \sum_{n=1}^3 \left[\sum_{u=0}^3 p_{iu} \cdot b_{(2u+1)4n} \right] \cdot 2^{-4n}$$

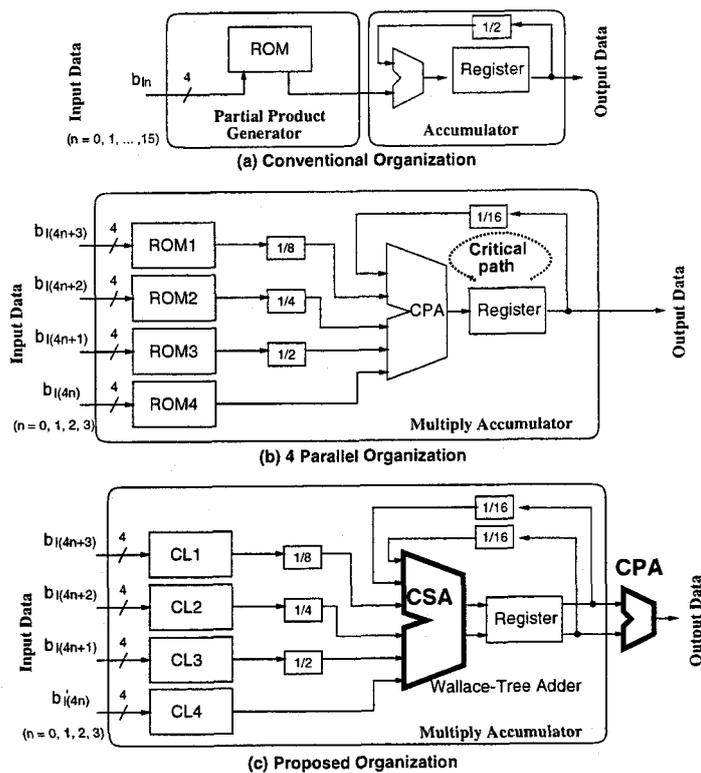


Fig. 3.15 積和演算器の構成

$$\begin{aligned}
 & + \sum_{n=0}^3 \left[\sum_{u=0}^3 p_{iu} \cdot b_{(2u+1)(4n+1)} \right] \cdot 2^{-(4n+1)} \\
 & + \sum_{n=0}^3 \left[\sum_{u=0}^3 p_{iu} \cdot b_{(2u+1)(4n+2)} \right] \cdot 2^{-(4n+2)} \\
 & + \sum_{n=0}^3 \left[\sum_{u=0}^3 p_{iu} \cdot b_{(2u+1)(4n+3)} \right] \cdot 2^{-(4n+3)}, \tag{3.10-a}
 \end{aligned}$$

$$\begin{aligned}
 x_{e_j} = & \sum_{u=0}^3 q_{ju} \cdot (-b_{2u \cdot 0}) \\
 & + \sum_{n=1}^3 \left[\sum_{u=0}^3 q_{ju} \cdot b_{2u \cdot 4n} \right] \cdot 2^{-4n} \\
 & + \sum_{n=0}^3 \left[\sum_{u=0}^3 q_{ju} \cdot b_{2u(4n+1)} \right] \cdot 2^{-(4n+1)} \\
 & + \sum_{n=0}^3 \left[\sum_{u=0}^3 q_{ju} \cdot b_{2u(4n+2)} \right] \cdot 2^{-(4n+2)} \\
 & + \sum_{n=0}^3 \left[\sum_{u=0}^3 q_{ju} \cdot b_{2u(4n+3)} \right] \cdot 2^{-(4n+3)}. \tag{3.10-b}
 \end{aligned}$$

式(3.10)の構成では、各積和演算を4サイクルで終わることができ、Fig. 3.15(a)の構成と比較して、1/4のサイクルで演算を終えることができる。このため、Fig. 3.15(b)の回路を8ユニット用いた場合、1マクロブロックを384サイクルで処理することができる。

しかしながらこの構成の場合、ROMから出力される4部分積と1累算結果を1サイクルで加算しなければならない。これを演算する5入力加算器を実装すると、この回路が桁上げの伝播などでクリティカルパスとなり、動作周波数を向上させることができない。このためFig. 3.15(b)の構成でも依然としてMP@HLの動画像に対する逆離散コサイン変換を実時間で処理することができない。

クリティカルパスの改善

Fig. 3.16に示す6入力(4部分積と、2累算ループ出力)Wallace-Tree型のパイプライン式CSA(Carry Save Adder)を用いることにより、最終的な積算結果を求めるCPA(Carry Propagate Adder)を累算ループの外部に置く構成(Fig. 3.15(c))を考案した。この6入力CSA Treeは桁上げの伝播を行わないため、毎サイクル4個の部分積の累算を高速に行うことができる。またCPAについてもパイプライン化を行い、動作周波数の向上を達成した。Fig. 3.15(c)の構成により、Fig. 3.15(b)の短所を克服することができた。また、部分積生成部はROMを用いずに組合せ回路で実現することにより、省面積化と部分積生成の遅延時間の減少を図った。

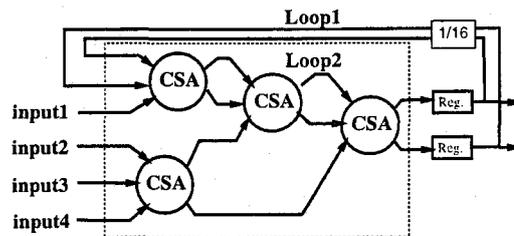


Fig. 3.16 6入力Wallace-Tree加算器

逆離散コサイン変換器の全体構成

Fig. 3.17に逆離散コサイン変換器の構成を示す。Fig. 3.15(c)の積和演算器2個を用いて、最終の加算/減算を行うことにより、一度に PX_o と QX_e の2係数(x_{o_i}, x_{e_i})—つまり式(3.5)、(3.6)の $\{(x_0, x_7), (x_1, x_6), (x_2, x_5), (x_3, x_4)\}$ のいずれかの組—を演算することができる。本逆離散コサイン変換器では、積和演算器対を4組用いており、シフトレジスタを用いて各積和演算器対への入力を1サイクルずつ遅らせて行わせている。このため、Fig. 3.15(c)中にある積算ループ外のCPAと最終の加減算器を共有することができ、サイクル毎に2係数が演算を可能とした。よって、 $8 \times 1-1$ 次元逆離散コサイン変換を4サイクル毎に終了でき、式(3.3-a)/(3.3-b)を各々32サイクルで終わることができる。Fig. 3.15(c)の構成により、従来、1マクロブロッ

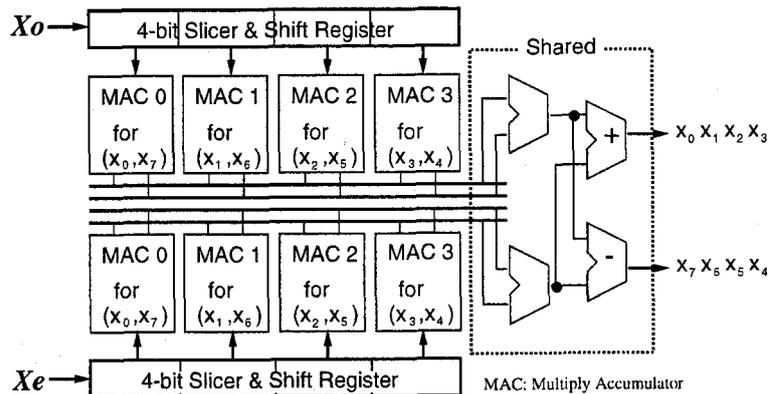


Fig. 3.17 逆離散コサイン変換器の構成

クの処理に 1,536 サイクルを要していた積和演算器を、実時間処理に十分な高周波数で動作し、かつ、384 サイクルで処理する機能ユニットとして実現した。

3.4.4 動き補償器

3.4.4.1 動き補償の従来手法

従来、動き補償は RISC 型の制御回路と DSP コアを用いて演算され、他の逆量子化や可変長符号復号化の過程と演算部を共有していることが多かった^[6, 22, 23]。しかしながらこのような設計では、他の処理に演算にも時間を割かなければならないために動き補償処理の性能が制限され、MP@ML の 6 倍もの処理性能を要求する MP@HL データを実時間で処理できない。特に動き補償では、参照画像中の画素データをフレームメモリから読み出し、演算を行った後再度メモリに書き込むといった作業を頻繁に繰り返すため、メモリインタフェースや内部データバスが高速化の大きな障害となる。

例えば双方向予測を行う場合、前方向の予測に対する参照画素の演算を行った結果をレジスタや外部メモリに書き出し、さらに後方向の予測に対する同様の演算を行った後、平均をとる。このような処理形態では、明らかにメモリインタフェースや内部バスのコンフリクトにより処理性能が低下する。

3.4.4.2 MP@HL 対応・動き補償器の設計

本節では従来のアーキテクチャの短所を克服するために、動き補償の処理に必要な全ての演算に対して専用ハードウェアを設計し、フレームメモリへのアクセス回数を大幅に軽減するアーキテクチャを構築した。動き補償の処理には、符号化データからの動きベクトル解読、デュア

ルプライム予測, 双方向予測, 半画素単位の予測等があげられる.

設計した動き補償器の構成図を Fig. 3.18 に示す. 動き補償器は, 動きベクトル復号部と予測画像生成部とからなる. 動きベクトル生成部では, MPEG2 が規定する様々な予測方式に対する符号から動きベクトルの再構築, 及び参照マクロブロックのアドレスの生成を行う. また, 予測画像生成部では, 参照ブロック中の画素を基に半画素単位の予測, 双方向予測を行い, 逆離散コサイン変換器からの予測誤差を足し合わせることで, 元の画像を復元する. 予測画像生成部は, 双方向予測を行うピクチャに対しても同時に 2 画素を生成することができるよう並列機構を導入した.

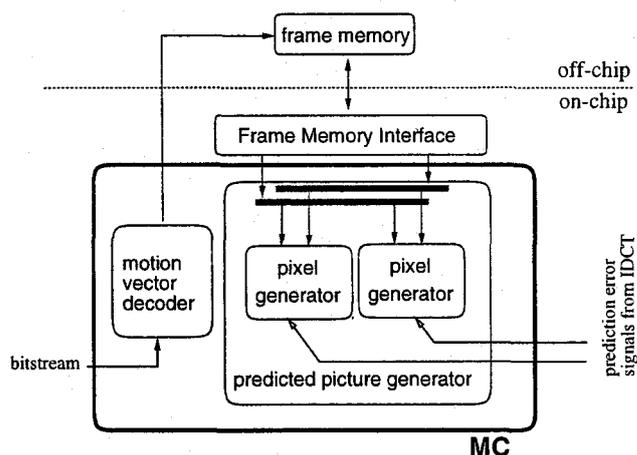


Fig. 3.18 動き補償器の構成

ベクトル復号部

ベクトル復号部は, 参照画像の参照マクロブロックを指定する符号化された動きベクトル (*vector*: 式 (3.12)) を復号化するものである. MPEG2 の動きベクトルは, 先行するマクロブロックの動きベクトル (*PMV*) との差異 (*delta*: 式 (3.11)) を基に式 (3.12) によって復号化することができる.

$$|\delta| = \begin{cases} |motion_code| & (f_code = 1 \text{ or } motion_code = 0), \\ (|motion_code| - 1) \times 2^{f_code-1} & \\ + motion_residual + 1 & (\text{otherwise}). \end{cases} \quad (3.11)$$

$$vector = \begin{cases} \delta + PMV + 32 \cdot 2^{f_code-1} & (\delta + PMV < -16 \cdot 2^{f_code-1}), \\ \delta + PMV - 32 \cdot 2^{f_code-1} & (\delta + PMV \geq 16 \cdot 2^{f_code-1}), \\ \delta + PMV & (\text{otherwise}). \end{cases} \quad (3.12)$$

ここで *motion_code*, *f_code*, *motion_residual* はそれぞれ、ベクトルの値、ベクトルの範囲、丸め誤差を表している。また *delta* の符号は *motion_code* と同じである。

式 (3.11), (3.12) の機能を実現した動きベクトル復号部の構成図を, Fig. 3.19 に示す。delta ユニットの符号化データの入力より, 式 (3.11) の *delta* を生成する。これを PMV レジスタの値と加算した後で, **range** ユニットで *f_code* による補正を行う。また, **scaler** ユニットでフレーム構造でのフィールド予測に対する処理を行っている。垂直方向の画素数が半分であるフレーム構造でのフィールド予測を行う際に, PMV レジスタの値の 1/2 倍を加算し, 演算した値の 2 倍を PMV レジスタに保存するという処理を行う。デュアルプライム予測を行っている場合はさらに Fig. 3.19 の出力をデュアルプライム演算部で演算する。

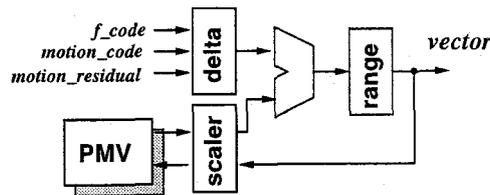


Fig. 3.19 ベクトル復号部の構成

以上の機能により, ベクトル復号部は参照画像の参照マクロブロックを指定する符号化された動きベクトルを復号化し, 参照ピクチャ中にある参照マクロブロックのフレームメモリ/画像キャッシュメモリアドレスの指定を可能とする。

予測画像生成部

予測画像生成部は, 2 個の画素生成部からなる。各画素生成部は以下のような機能を持つ。

1. 画像キャッシュメモリの処理を軽減するために半画素単位の予測を演算することができる。
2. 画像キャッシュメモリのサイズと予測画素生成時間を減少させるために双方向予測の演算を並列に実行することができる。
3. MP@HL の動画像を実時間で復元するために, 同時に 2 画素を出力する。

画素生成部は以上の機能を 3 段のパイプライン構成により実現している (Fig. 3.20)。

第 1 段: 時間軸上の前方向の予測に対して, 動きベクトルの示す参照マクロブロックの画素/半画素の生成を行う。同時に後方向の予測に対しても同様の処理を行う。

第 2 段: 1 段目で得られた結果の加算器への入力を制御することで, 前方向のみ, 後方向のみ, 双方向のうちの一つの予測値を生成する。

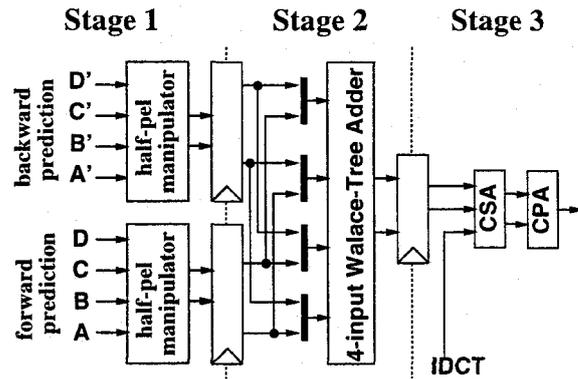


Fig. 3.20 パイプライン画素生成部の構成

第3段: 2段目の出力となる参照画素と逆離散コサイン変換器からの予測誤差を足し合わせる
ことにより、元の予測画素を生成する。

Fig. 3.20 中の半画素演算器の構成を、Fig. 3.21(b) に示す。MPEG2 の半画素予測では、参照画素の候補が Fig. 3.21(a) の画素 A のみでなく、中間値である画素 1, 2, 3 となり得る。画素 1, 2, 3 は、それぞれ A と B の平均、A と C の平均、A, B, C, 及び D の平均により求めることができる。そこで半画素演算器は、4点の中で最も演算量の多い画素 3 を処理する 4 入力加算器に対し、入力を (A, A, A, A), (A, B, A, B), (A, A, C, C), (A, B, C, D) と切り替え、1 個のユニットで全ての画素値 A, 1, 2, 3 を処理することができるよう構築した。また、Fig. 3.20 のように用いられる加算器には、最終段を除いて CSA を使用し動作周波数の向上を図った。

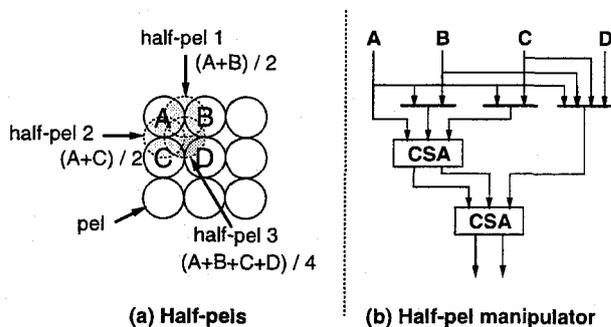


Fig. 3.21 半画素予測と半画素演算器の構成

従来、DSP コアを用いた手法では、1 画素の演算に (1) 前方向の参照画素の読み出し、(2) 後方向の参照画素の読み出し、(3) 予測誤差との加算の 3 ステップを要したため、256 個の輝度信号と 128 個の色差信号を持つ 1 マクロブロックの処理は、少なくとも $384 \times 3 = 1152$ サ

イクル必要である。本動き補償器は、半画素演算器を持ち、毎サイクル画素を復号化することができる画素生成部を2個内蔵するため、192サイクルで1マクロブロックの処理を終えることができる。

以上により、設計した動き補償器はMP@HLの動画像に対し、半画素単位での双方向予測に関しても、動き補償を実時間で処理することができる。

3.4.5 マクロブロックレベル・パイプラインバッファ

3.4.5.1 パイプライン・バッファの全体構成

本節では、各機能ユニット間に必要なパイプラインバッファの機能と容量を分析し、設計した機能ユニットの性能を最大限に利用することができるアーキテクチャを構築する。

設計した各機能ユニットの処理能力はMP@HLの実時間可能な1マクロブロック当たり $4.08\mu\text{s}$ 以下に抑えている。このため、各専用機能ユニットをパイプラインステージに割り当て、機能ユニットの処理を最大限に引き出すことのできるパイプラインバッファで接続することで、マクロブロックレベルでのパイプライン処理が可能となり (Fig. 3.22), MP@HLの実時間処理が達成される。

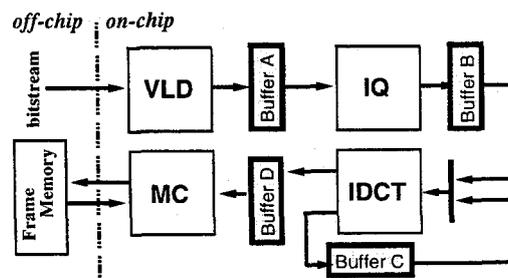


Fig. 3.22 マクロブロックレベル・パイプラインの構成

MPEG2の復号化用プロセッサにおいて、Fig. 3.22のようにマクロブロックレベル・パイプラインを構成した場合、命令レベルのそれと比較して以下のような留意すべき相違がある。

- (1) 可変長符号復号化器において、サイクル毎の入出力が一定でない。
- (2) 各機能ユニットの1サイクル当たりの入出力データ数が異なる。
- (3) データのアクセス順序(走査順)が各機能ユニットの入力/出力により異なる。
- (4) パイプラインステージの機能ユニット自体がパイプライン化されている。

(1) は、MPEG2 の符号化データが可変長符号であることに起因する。また (2), (3) の例としては、可変長符号復号化器におけるジグザグスキャンや、逆離散コサイン変換器におけるブロックの行 (列) 毎に 2 係数ずつを入力することが挙げられる。(4) は、逆離散コサイン変換器の Buffer C の設計において考慮する必要がある。

以上より、パイプラインバッファとして通常の FIFO(First-In First-Out) やレジスタ、RAM で設計を行っても、機能ユニットの性能を低下させるばかりではなく、オーバフロー/アンダフローが発生するため適切な復号化の妨げとなる。このため、各機能ユニットが MP@HL 動画像符号の復号化能力を保持しているにも関わらず、デコーダ全体としてこの処理能力を達成できなくなる。これらの点から各ステージ間のパイプラインバッファは、前後の機能ユニットのパイプライン段数や入力/出力の走査順、サイクル当たりのデータ数を個別に考慮して設計する必要がある。

例として、Fig. 3.22における Buffer B および Buffer C のブロック毎の書き込み/読み出し順序を Fig. 3.23に、これを元に設計した各バッファのタイムチャートを Fig. 3.24 に示す。

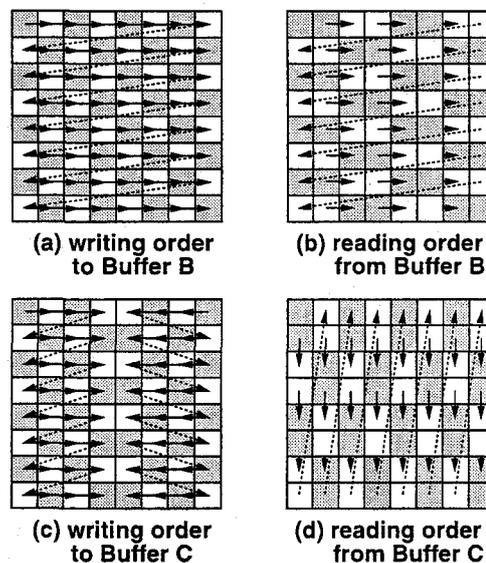


Fig. 3.23 Buffer B および C の書き込み/読み出し順序

これより、Buffer B は、64-word 3 ポートレジスタ、Buffer C は、32-word 3 ポートレジスタ 4 個から実装可能であることが分かる。

設計したマクロブロックレベル・パイプラインバッファ Buffer A, B, C, D により、各機能ユニットが 1 ブロックあたり 64 サイクルで処理することができるマクロブロックレベルパイプラインを構成することができる。設計した機能ユニットの処理性能を最大限に引き出すことができるため、デコーダ全体として、MP@HL 動画像を実時間処理を可能とする。

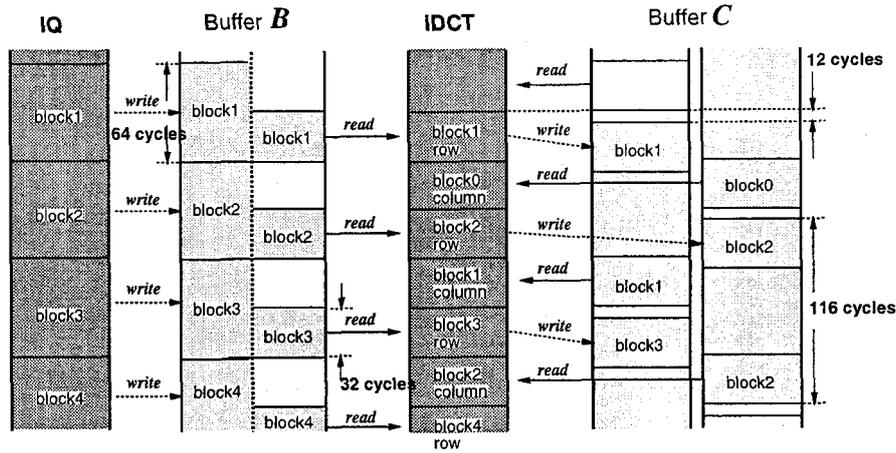


Fig. 3.24 Buffer B および C の書き込み/読み出しタイムチャート

3.4.6 フレームメモリ構成とインタフェース

動き補償器は参照画像の取得に加え、生成画像の格納に際してもフレームメモリを頻繁にアクセスする。特に、MP@HL 動画像では解像度が高いため、単位時間あたりのアクセス数が莫大となり、性能向上のボトルネックとなっている。このボトルネックを解決するために、本デコーダで採用したフレームメモリの構成を Fig. 3.25 に示す。

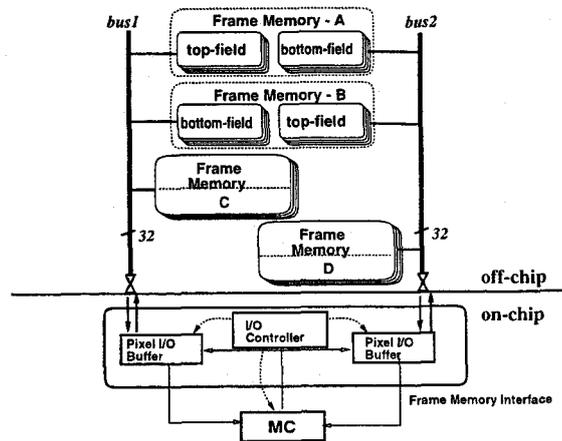


Fig. 3.25 フレームメモリとインタフェースの構成

本構成では以下のような特徴を持つ。

- MPEG2 のサポートする全てのピクチャ符号化構造 (I, P, B-picture) に対応するために、

合計4枚のフレームメモリ A, B, C, Dを持つ。Fig. 3.25 のフレームメモリ A, B はイントラ符号化等、フレームメモリ C, D は双方向予測等に使用される。

- 動き補償器で2画面同時参照を行えるようにするため、画素データの転送は2本の双方向データバスを介して行い、ピクセル I/O バッファを用いて動き補償器に適切な入出力を行う。
- インタレース画像に対応するために、フレームメモリ A ならびに B はトップ/ボトムフィールドに分割し、それぞれ2本の双方向バスに接続する。
- 一度に4画素のデータを転送するため、各フレームメモリは4バンクに分割し、双方向バスを32-bit 構造とする。

フレームメモリインタフェース部はベクトルデコーダ、ピクセル I/O バッファ、ならびに I/O コントローラからなる。ベクトルデコーダでは、動き補償器の出力である動きベクトルからフレームメモリのアドレスを生成する働きを持ち、2本の双方向バスにアドレスを送出することにより、適切に参照画像を取得する。

ピクセル I/O バッファの構成を Fig. 3.26 に示す。ピクセル I/O バッファはフレームメモリからの参照画素データを蓄積し、IDCT ユニットの予測誤差データの出力に合わせて、予測画像生成部に画素を供給するもので、前方向予測用と後方向予測用の2ピクセルバッファからなる。各ピクセルバッファは、Fig. 3.27 に示すような順に画素データをフェッチする。まず、半

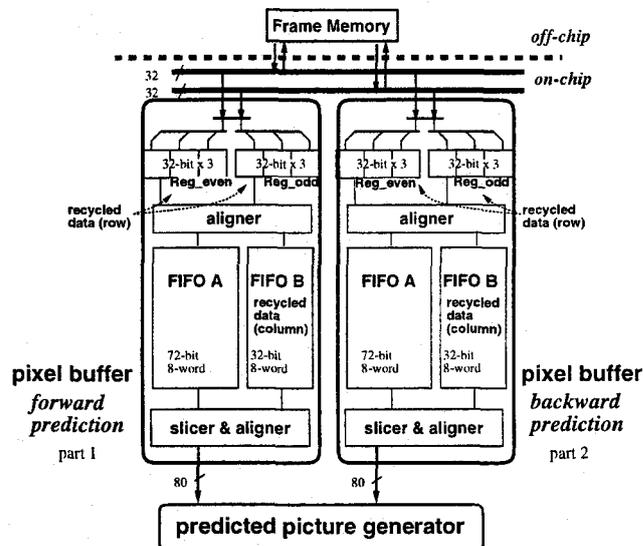


Fig. 3.26 ピクセル I/O バッファの構成

画素予測に対応するため拡張ブロック (9 × 9 画素のブロック) 毎にフレームメモリから読み出し、参照順序にしたがって並べ替えた上、レジスタと FIFO に蓄積、予測画像生成部に供給する。ここで、Fig. 3.27 中の “recycled data” と記した画素データは Fig. 3.26 中の対応するレジスタまたは FIFO に保持することができるため、フレームメモリのアクセス数を削減することができる。

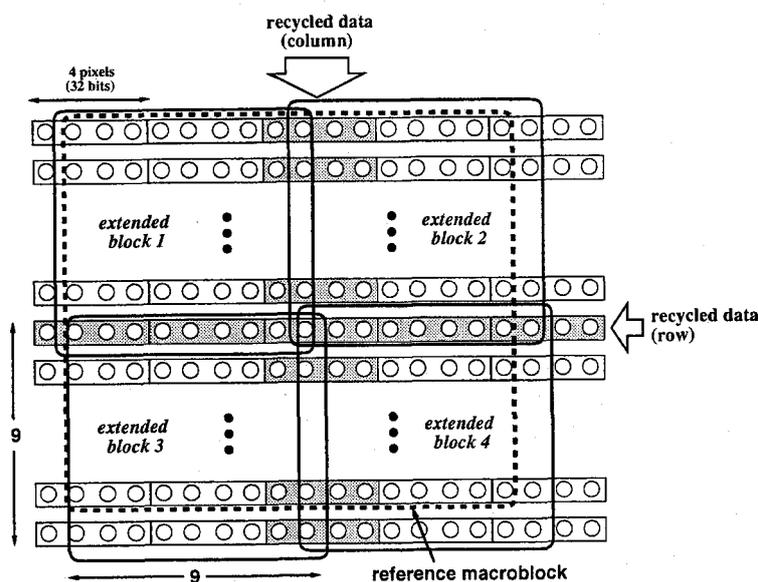


Fig. 3.27 フレームメモリからのブロックデータフェッチ順序

複雑なフレームメモリアクセスの例として、Fig. 3.28(a) に前方向予測のフィールド予測時、ならびに、Fig. 3.28(b) にインタレースピクチャの双方向予測時の参照手順を、さらに、Fig. 3.29 に Fig. 3.28(b) の予測モードでのデータバスタイミングチャートを示す。このようなフレームメモリアクセス要求に対しても Fig. 3.25 の構成により、参照データの転送が可能であることが分かる。

結果として、参照マクロブロックの読み出しならびに動き補償器の出力結果の書き戻しは、256 サイクルで可能となる。フレームメモリとデータバスは、66MHz で動作可能であるため、MPEG2 MP@HL の全ての予測モードに対して、データ転送が可能となる。MP@HL の画像が、 $1920 \times 1152 \times 8\text{-bit}$ の輝度信号と $960 \times 576 \times 8\text{-bit} \times 2$ の色差信号からなることから、フレームメモリ A, B, C, D は、それぞれ 16M SDRAM または、16M VRAM 4 個ずつで構成可能である。

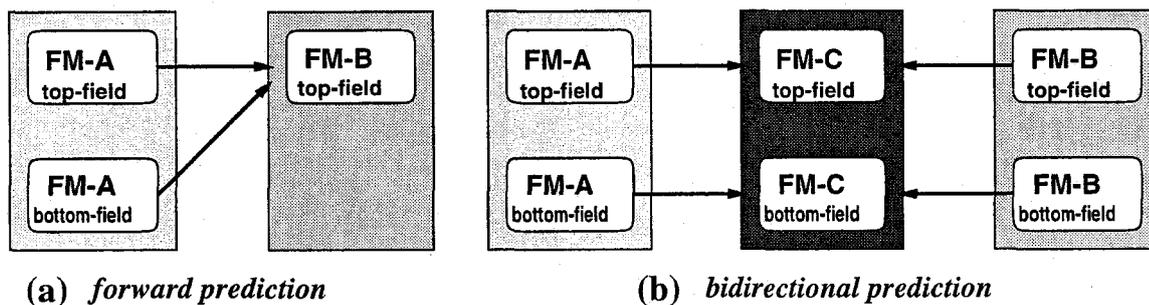


Fig. 3.28 予測モード別フレームメモリアクセス

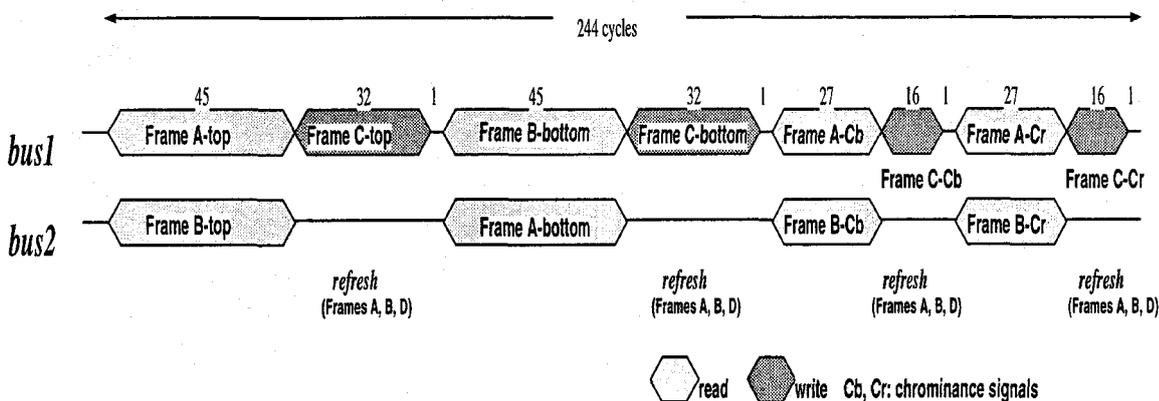


Fig. 3.29 インターレスピクチャ双方向予測時のバスタイムチャート

3.4.7 シーケンス制御部

チップ全体の制御部では各マクロブロックのデコードプロセスを制御し、各演算器間の同期を司る。Fig. 3.30 にシーケンス制御部の状態遷移を示す。可変長符号復号化器に併設するヘッダ検出、ヘッダ解読部で、ヘッダならびに画面情報などを解読し、マクロブロックヘッダを発見すると、マクロブロックデコード処理にモードを移す。マクロブロックデコードモードでは、異常なき場合、速やかに復号化を行うが、マクロブロックデコードの終了もしくは、異常発見時には残りの情報をスキップし、いち早くヘッダ検出を行う。DSP タイプに比べ、マクロブロックレベルのパイプラインを利用したデコーダではデータの流りが一定しているため制御は比較的簡単に行うことができる。

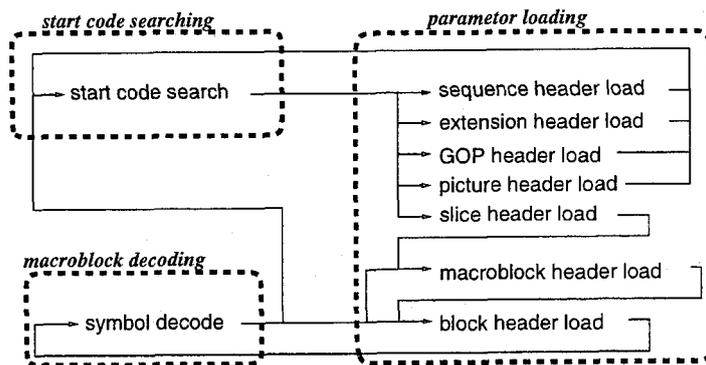


Fig. 3.30 シーケンス制御部の状態遷移

3.5 実装結果

設計した MPEG2 MP@HL デコーダをトップダウン VLSI 設計システム COMPASS Design Navigator を用いて VLSI 化した。0.6 μm 3 層 CMOS データパス/スタンダードセルを用いた実装結果を Table 3.2, Table 3.3 に示す。また, Fig. 3.31 にレイアウトを示す。

Table 3.2 機能ユニットの実装結果

Unit	Area (mm^2)	# Tr.
VLD	2.2	36,786
IQ	1.6	13,100
IDCT	6.4	43,350
MC	2.2	18,611
Buffers A~D	6.2	156,464
Pixel Buffers, Frame Memory I/F	5.3	94,419
Sequence Ctrl.	7.5	88,852

MPEG2 の MP@HL (Main Profile at High Level) の復号化を実時間で行うには, 1 マクロブロックあたり 4.08 μs 以内で処理しなければならない。設計した可変長符号復号化器, 逆量子化器, 逆離散コサイン変換器, 動き補償器は 1 マクロブロックを 3.84 μs 以下で処理することができるため, HDTV レベル (1920 \times 1152) の動画データを実時間で復号化する能力を持つ。

Table 3.3 MPEG2 MP@HL デコーダ緒元

テクノロジー	0.6 μ m CMOS triple-level Al
コアサイズ	8.8 × 9.2 mm ²
トランジスタ数	454K
動作周波数	100MHz (フレームメモリインタフェース: 66MHz)
消費電力	1.2W (at 3.3V)
サポート画像	MPEG2 MP@HL (I, P, B-picture)

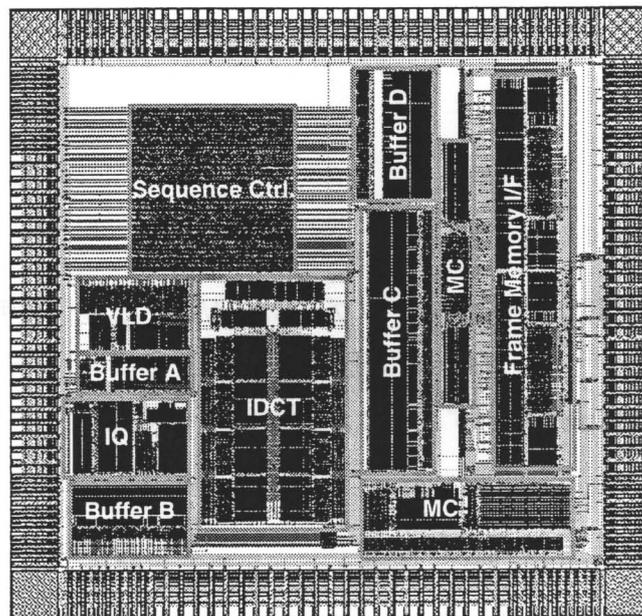


Fig. 3.31 MPEG2 MP@HL デコーダのレイアウト

3.6 結言

本章では、HDTV レベルに対処した MPEG2 デコーダの設計と実装について述べた。設計したデコーダは、マクロブロックレベルのパイプラインを構成し、各パイプラインステージに対応させる MPEG2 の処理に対して専用機能ユニットを設計することで、MP@HL 動画像を実時間処理することができることを示した。専用機能ユニットとして可変長符号復号化器、逆量子化器、逆離散コサイン変換器、及び動き補償器を構築し、それぞれが MP@HL の 1 マクロブロックの処理に許される時間内で処理を終えるよう設計した。可変長符号復号化器では、可変長符号の符号長の検出と復号化とを分離して行うことで高速化を実現した。逆量子化器では、パイプライン型並列乗算器を用いることで高速化を実現した。逆離散コサイン変換器は分散演算において、同時に複数の係数を処理し、動作周波数を向上させることにより高速化を達成し

た。また、動き補償器に半画素演算器等の専用機能ユニットを導入し、フレームメモリインタフェースにフレームメモリのアクセス回数を減少させる機構を設けることで、予測画像生成の処理速度向上を実現した。また、設計した各機能ユニットの処理能力を最大限に引き出すことのできるパイプラインバッファを構築した。これらの機能ユニットとパイプラインバッファにより、MPEG2 の HDTV レベル動画像を実時間処理することができるデコーダを実現可能であることを示した。

第4章

VCI 共有セルを用いたマルチメディア ATM ネットワークにおける音声通信手法

4.1 緒言

本章では音声データを効率よく伝送し、かつ、画像やデータ伝送を行うネットワークと統合することができる新しいマルチメディア ATM ネットワーク手法を考案する。このマルチメディア ATM ネットワークは、各段のスイッチに新たにリルート機能を導入し、ネットワーク内で動的に音声用セルを多重化したり、あるいは組み替えたりすることによって、伝送効率を向上可能であることを示す。考案するネットワーク手法において、スイッチ機能は標準の ATM スイッチにより行い、リルート機能はリルートノードを新たに設計しスイッチの入力部に接続することで実現する。リルート機能を実現するために、新たに VCI(Virtual Channel Identifier) 共有セル化手法を考案し、このセル化手法を通じて達成される伝送効率を評価する。VCI 共有セルは標準 ATM セルとの親和性に優れているため、リルートノードのみが VCI 共有セルと標準 ATM セルの区別を認識し、各セルに対して必要な処理をするだけで、他のネットワーク構成要素に影響を与えることなく、マルチメディア ATM ネットワーク内の音声伝送効率を向上させることを示す。

4.2 マルチメディア ATM ネットワーク構成

本研究で議論するマルチメディア ATM ネットワークの概略図を Fig. 4.1 に示す。ネットワークは、FTTH 構想に基づいており、動画、高速データ、音声などの情報をすべて ATM セルを用いて伝送するために、各加入者を ATM 接続している。伝送される情報は、まず端末 (TE: Terminal Equipment) 中の CLAD(Cell Assembly and Disassembly) で ATM セルに組み立てられる。次に生成されたセルは ATM マルチプレクサ (MUX) を経て集線され、それぞれのセル

が保持する情報の種類や特性によらず、マルチメディア ATM ネットワークを介して目的地に送られる。

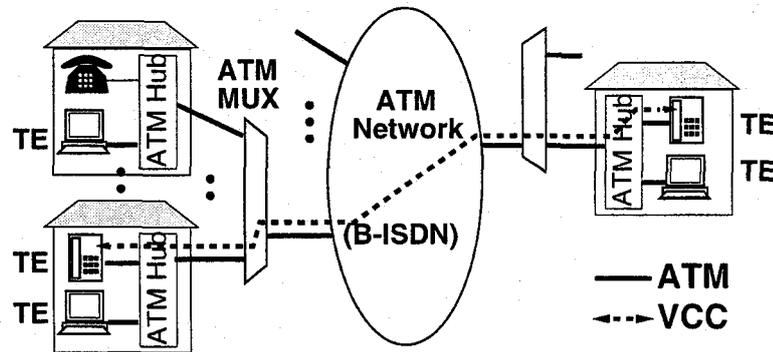


Fig. 4.1 マルチメディア ATM ネットワークの概略

これまでに、計算機のデータ伝送等、情報伝達系を中心に発展してきた ATM ネットワークは、CATV 等のデジタル有線放送系や既存の電話網である音声用狭帯域ネットワークとは個別のネットワークとして活用されてきた。これに対し、Fig. 4.1 に示す構成は、高速データ、画像、音声などのマルチメディア情報を一元的に扱うことのできる統合的なネットワーク手法を実現するものであり、より実用的な通信インフラストラクチャの構築を達成するものである。

4.2.1 音声と ATM セル

マルチメディア ATM ネットワークにおいて、64kbps の音声データを ATM セルに割り当てる最も簡単なセル化手法は、各呼毎に 1 セルを用いる手法である。これを 1 対 1 マッピングと呼ぶ。1 対 1 マッピングでは、64kbps のコネクション毎に、ATM ネットワークの仮想チャネルコネクション (VCC: Virtual Channel Connection) が使われる。しかしこの手法では、音声データを ATM セルのペイロードに蓄積するために生じるセル化遅延が大きい。64kbps の音声データの場合、セル化遅延は約 6ms になる。セル化遅延によって生じるいわゆるエコー問題等の通信品質の低下を避けるために、CLAD で生じる遅延は 1ms 以内に抑える必要がある^[8]。このセル化遅延を削減するためにペイロードに保持する情報量を制限し、ペイロードを一部使用せずに ATM セルを伝送すると、伝送効率が悪化する。

セル化遅延とそれに伴う伝送効率低下の問題を解消するために、複数の呼の音声データを 1 セルに収容する多対 1 マッピングによるセル化手法が提案されている。従来の ATM ネットワークのように、狭帯域ネットワークとの接続点である IWU (Interworking Unit) 内の CLAD でセル化が行われている場合、同じ ATM 終端点へ向かう複数の呼が存在し、同一の VCC で伝送す

ることが可能であった。しかしながら Fig. 4.1 のネットワークにおいては、各端末でセル化が行われるため、CLAD 内の呼は 1 種類存在するだけとなり、同一の ATM 終端点を持つ呼は存在しない。したがって、従来手法のように、セル化時に多対 1 マッピングを行うことができず、マルチメディア ATM ネットワーク内の音声伝送効率が低下する。本論文では、Fig. 4.1 のマルチメディア ATM ネットワークにおいても高い音声伝送効率を得るために、別々の経路を伝送される複数の呼の音声データをネットワーク内で動的に 1 個の ATM セルに多重化し、またそれらを組み替える手法を考案する。

4.2.2 VTOA を考慮したマルチメディア ATM ネットワーク構成

Fig. 4.2 は、多対 1 マッピングと音声用セルの組み替えを考慮したマルチメディア ATM ネットワークのモデルである。このモデルは、1 個の 8×8 のトランジットスイッチ (TS) と 8 個の 8×8 のローカルスイッチ (LS_i ($i = 0, 1, \dots, 7$)) からなり、各入出力線にはリルートノード (RT または RL) が接続されている。リルートノードは、音声用セルの多重化、およびその組み替えを行うために本章で新しく導入する VLSI であり、従来のネットワークでは HCV (header converter) が配置してある位置に設置する。各 ATM スイッチは、155.52Mbps のハイウェイを 8 本持ち、8 ビット並列、19.44MHz で動作する。このような ATM スイッチは、近年の VLSI 技術の進歩により比較的安価に入手可能である。LS_i は、加入者端末内の CLAD から伝送される ATM セルを 1 本のハイウェイに集線し、スイッチングする。TS は、トラフィック密度の高い部分におかれ、LS_i から送られてくる ATM セルをスイッチングする。加入者は複数の端末 (TE) を持つことができ、音声や、画像、高速データを別々に扱うことができる。

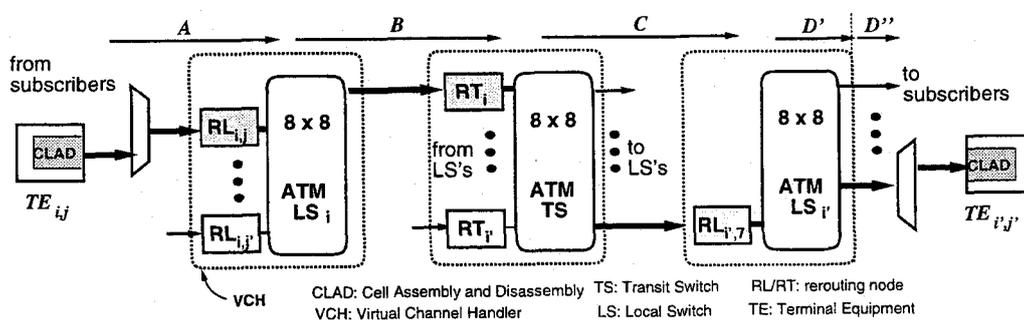


Fig. 4.2 マルチメディア ATM ネットワークモデル

考案するネットワークアーキテクチャを説明するために、ネットワークの各部分を記述する記号を示す。LSin_{i,j}とLSout_{i,j} ($i, j = 0, 1, \dots, 7$) は、それぞれ LS の入出力線を表す。i, j はそれぞれスイッチ番号とハイウェイ番号を表す。同様に TS は、TSin_iとTSout_i ($i = 0, 1, \dots, 7$) で表

される8本の入出力線をもつ。ここで、 i はハイウェイ番号である。RT $_i$ は、TSのリルートノードを表し、LSout $_{i,j}$ とTSin $_i$ の間に位置する。RLはLSのリルートノードを表す。RL $_{i,j}$ ($j \neq 7$)の一端はLSin $_{i,j}$ とLSout $_{i,j}$ に接続されており、他端は、複数の呼を集めるATM MUXに接続される。一方RL $_{i,7}$ はTSout $_i$ とLSin $_{i,7}$ の間に接続されている。TE $_{i,j}(\alpha)$ はRL $_{i,j}$ に接続されているTEを表す。ここで α は個々のTEを表し、音声データを扱う端末の他に、他の画像情報等を扱うTEも存在する。加入者はそれぞれ複数の端末を持つ。 $X \in \{A, B, C, D', D''\}$ は、Fig. 4.2中のVCリンクA, B, C, D', D''のうちの1つを表す。ただし、D', D''は、標準のATMプロトコルにおけるVCリンクとは異なる。

音声用セルは、リルートノード内でのみ認識される。音声用セルの多重化、および組み替えの概念図をそれぞれ、Fig. 4.3, Fig. 4.4に示す。リルートノードは到着した音声用セルを個々

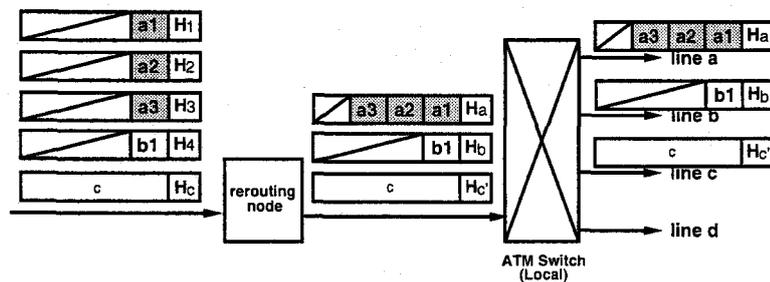


Fig. 4.3 音声用セルの多重化 (RL)

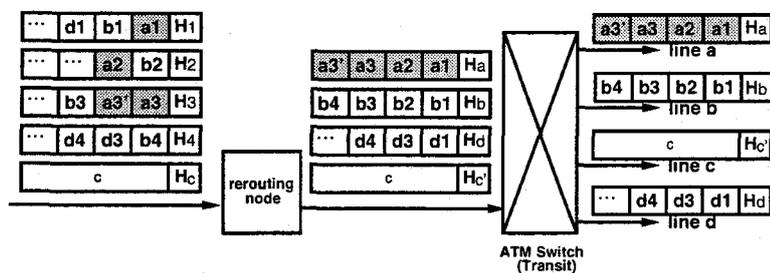


Fig. 4.4 音声用セルの組み替え (RT)

の呼に分割し、同じ出力方路へ向かう呼をATMセルに多重化する。例えばFig. 4.4において、リルートノードに到着した音声用セル中の呼のうち、スイッチの出力方路line aを通る呼 a_1, a_2, a_3, a_3' は1ATMセルに収容する。他の出力方路line b, dを通る呼についても同様の処理を行う。また、標準ATMセルCについてはATMセルのヘッダ変換のみが行われ、ペイロードに

に対する処理は行われない。音声用セルは標準 ATM のプロトコルに従っているため、リルートノード外においては、標準 ATM セルと音声用セルを区別せずに処理することができる。

次に考案するネットワークの收容能力とその拡張性について述べる。1,000 加入者がスイッチの 1 入力線に收容され、70 加入者が同時に発呼する場合を考える。ハイウェイの転送容量は 155.52Mbps であるため、それぞれの加入者には、音声、画像、高速データを送信するために平均約 2Mbps の帯域が割り当てられ、音声等を伝送するには十分となる。この時、Fig. 4.2 のマルチメディア ATM ネットワーク全体で $1,000 \times 7$ 入力線 $\times 8$ スイッチ、つまり 56,000 回線が收容可能となる。Fig. 4.2 のネットワークは 3 ステージ (LS-TS-LS) 構成であるが、この構成のスケラビリティを利用して多段化することでネットワークの規模を拡大することが可能である。また、TS のスイッチ容量の拡張やハイウェイの転送レートを引き上げることで、加入者数を増加させることが可能である。

ATM のハードウェアスイッチングによる高速性を損なわないようにするために、リルートノードでのセルはハードウェアで処理する。これらのリルート機能を実現するために、複数の呼を收容する音声用 ATM セルとして、VCI 共有セルを導入する。次節で、VCI 共有セルの構造とリルートノードにおける VCI 共有セルの処理について述べる。

4.3 VCI 共有セル化手法

4.3.1 従来の音声セル化手法

従来の音声のセル化手法の研究は、狭帯域ネットワークやプライベート ATM ネットワークに接続された ATM ネットワークについてなされていた。Fig. 4.5 に狭帯域ネットワークのサービスに対する ATM バックボーンネットワークの概念図を示す。ATM ネットワークは狭帯域ネットワークのバックボーンとして用いられる。N-ISDN と B-ISDN の接続点に必要な機能は IWF (Interworking Function) を用いて行われる。IWF は ATM 端末や VCC の終端点となっておりと同時に複数の呼の情報が集まる点でもある。

セル化遅延を削減するために、従来、パーシャルフィルセル、ショートセル、コンポジットセル、レイヤードセル等が提案されている。各セルの概略図を Fig. 4.6 に示す。

パーシャルフィルセル (Fig. 4.6(a)) は、N-ISDN の 1 コネクション中の数バイトのデータからセルを構成する。この手法は容易にセル化遅延を減少することができるが、帯域を有効に利用することができない。

ショートセル (Fig. 4.6(b)) は、パーシャルフィルセルから、利用していないペイロードの部分を切り離れたものである。標準 ATM セルよりペイロードが短いためセル化遅延は小さくなり、かつパーシャルフィルセルよりは高い伝送効率を得られる。

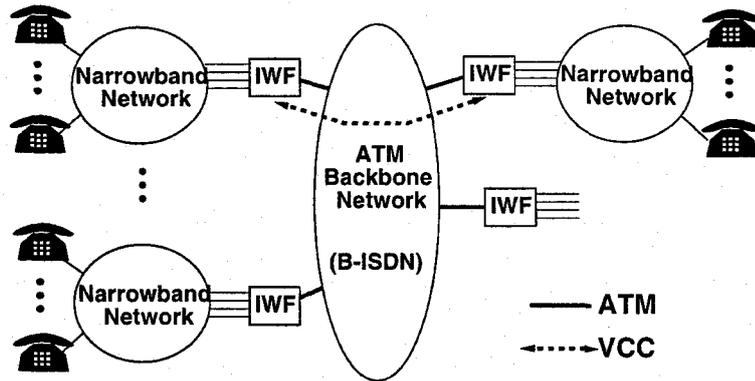


Fig. 4.5 バックボーンとしての ATM ネットワーク

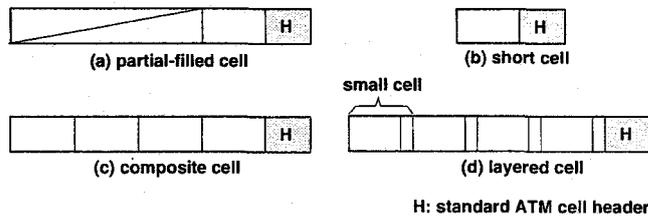


Fig. 4.6 従来の音声セル化手法によるセル構造

コンポジットセル (Fig. 4.6(c)) は、多対1 マッピングの概念に基づいている。同一の ATM 終端点へ向かう複数の呼を同じ VCC に割り当て、1ATM セルに多重化する。マッピングされる位置の情報はネットワーク内で保持する。このセル化手法により、セル化遅延の削減と高い伝送効率が達成される。

レイヤードセル (Fig. 4.6(d)) は、ペイロードが複数の小さなセルから構成されており、統計多重効果や無音圧縮が利用可能な多対1 マッピング手法である。レイヤードセルは、それ自身のヘッダと複数の小さなセルからなる。レイヤードセルに割り当てられたそれぞれの呼は、小さなセルが持つヘッダにより識別される。

以上のセル化手法をマルチメディア ATM ネットワークに適用することを考える。パーシャルフィルセルは伝送効率が低く、特にトラフィック密度が高いネットワークにおいて、他の呼や情報の伝送の妨げになる。ショートセルは、標準 ATM セルとセル長が異なるため、画像や高速データを伝送する標準の ATM ネットワークとの共存は難しい。また Fig. 4.1 のネットワークに示すように、各呼はすべて異なる VCC を持つので、同一の VCC に複数の呼を割り当てるコンポジットセルやレイヤードセルは利用することができない。

そこで、Fig. 4.2 のネットワークモデルにおいて、音声を効率よく伝送するための新しいセ

ル化手法が必要となる。

4.3.2 標準 ATM の仮想チャネル

本章で考案する VCI 共有セルは標準の HCV の機能を利用する。標準の ATM ネットワークにおいて、HCV は Fig. 4.2 の RT や RL のように ATM スイッチの前に設置されている。VCI 共有セルの構成および処理方法を説明するために、まず標準 ATM セルの VC ルーティングの手法について述べる。

B-ISDN の通信は仮想チャネル (VC) レベル、仮想パス (VP) レベル、トランスマッションパスレベル、デジタルセクションレベル、リジェネレータセクションレベルで行われる。VC レベルは、Fig. 4.7 の VCC (仮想チャネルコネクション)、VC リンク、VC の終端点、VC の接続点からなる。VC の終端点は TE (Terminal Equipment)、接続点は VCH (Virtual Channel Handler) であり通常スイッチを指す。仮想チャネルコネクションは終端間のコネクションと定義できる。VC リンクは、接続点間あるいは終端点から接続点間でのリンクとして定義され、VCI は仮想チャネルコネクションごとに割り当てられ、コネクションの識別子として標準 ATM セルヘッダ内に格納される。VC ルーティングは、VCH が、入力された VCI を対応する他の VCI に変換することで行われる。例えば、Fig. 4.7 の VCI_A と VCI_B は、常に同じ値とは限らないが、両方とも同一の仮想チャネルコネクションの識別子を表している。

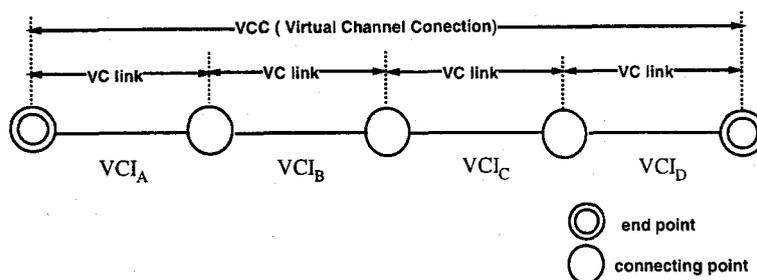


Fig. 4.7 VC レベル

4.3.3 VCI 共有セルの構成

Fig. 4.2 のネットワークモデルにおいて音声を効率よく伝送するために、スイッチの同一出力方路へ向かう複数の呼を多重化することが可能なセル化手法、VCI 共有セル化手法を導入する。VCI 共有セルのペイロードは同一出力方路へ向かう複数の呼のデータ (k バイト) と各呼に対応したヘッダからなる。このヘッダと呼の情報をまとめてフェローセルと呼ぶ。

本章で考案するマルチメディア ATM ネットワークで使用する 標準 ATM セル, ソリタリセル (solitary cell), VCI 共有セルの構成をそれぞれ, Fig. 4.8(a), (b), (c) に示す. 標準 ATM セルは音声を除くマルチメディア情報の伝送に用いられる. VCI 共有セルはヘッダと, 伝送する各呼の情報を保持したフェローセルとからなる. ソリタリセルは, CLAD からローカルスイッチに至る, 多重化されるまでの経路で音声用セルとして用いられる.

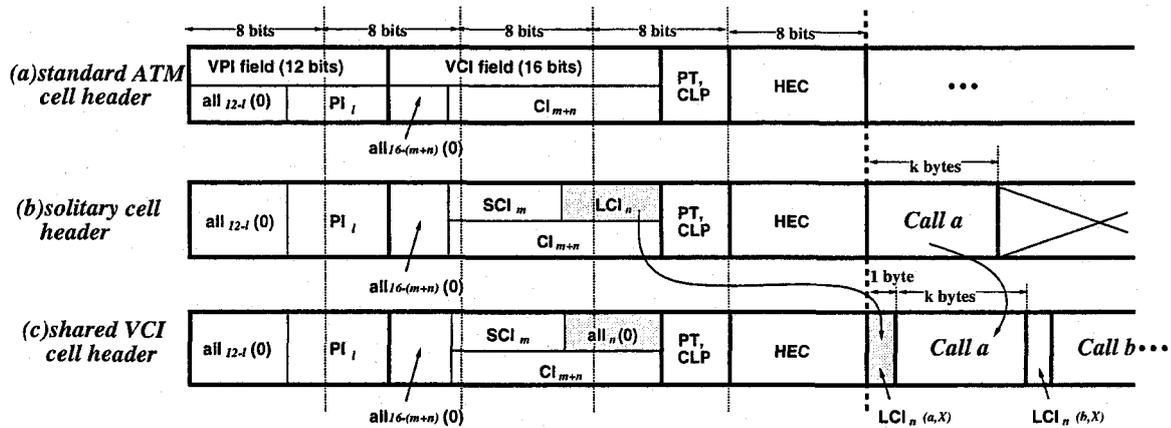


Fig. 4.8 マルチメディア ATM ネットワーク用 ATM セル

Fig. 4.8 に示すヘッダをマルチメディア ATM ネットワークでは, 以下のように規定する. 標準 ATM ネットワークノードインタフェース (NNI) において, 28 ビットをルーティングのために用いることができる. このうち 12 ビットが VPI (Virtual Path Identifier), 16 ビットが VCI に割り当てられているが, すべてのビットがネットワーク内で用いられるとは限らない. 本論文では, VPI と VCI フィールドのビット列をそれぞれ PI_l , CI_{m+n} で表す. $PI_l(a, X)$ は Fig. 4.2 の VC リンク X 中でのコネクション a の VPI を表す. ここで l は VPI のビット長を表し, $1 \leq l \leq 12$ である. $CI_{m+n}(a, X)$ は, VC リンク X 中でのコネクション a の VCI を表す. $m+n$ はビット長を表し, $1 \leq m+n \leq 16$ である. 以下では簡単化のために, 1 つの物理伝送路における PI_l の数を 1 つに制限するが, 一般性は失わない. さらに, ソリタリセルと VCI 共有セルの VCI フィールドは共有 VCI 部 (SCI_m) と下位 VCI 部 (LCI_n) に分ける. $SCI_m(a, X)$ は m ビット長の VC リンク X 中でのコネクション a の共有 VCI を表す. $LCI_n(a, X)$ は n ビットであり, VC リンク X 中でのコネクション a の下位 VCI を表す. ここで, $0 \leq m \leq 16$, $1 \leq n \leq 8$ である. したがって, $CI_{m+n}(a, X)$ は, $\langle SCI_m(a, X), LCI_n(a, X) \rangle$ と等しい. ただし, $\langle \sigma_1, \sigma_2 \rangle$ はビット列 σ_1 と σ_2 の連結を示す. 連結は, 連想的であるので $\langle \langle \sigma_1, \sigma_2 \rangle, \sigma_3 \rangle$ は $\langle \sigma_1, \sigma_2, \sigma_3 \rangle$ を表す. CI_{m+n} の最上位ビットは, 音声用セル (VCI 共有セル, およびソリタリセル) の識別子として用いる. いかえると, マルチメディア ATM ネットワークは音声伝送のために VC を確保していること

になる。音声用セルの識別子のビット数は1ビットに限定しない。

各呼が個別のVCC上の情報であることから、Fig. 4.2のVCリンクB,C,D'上で多重化した際、各呼を識別するためにフェローセルのヘッダは最大16ビットのVCI情報(CI_{m+n})を持つ必要がある。しかし2バイトのヘッダを各フェローセルが保持すると伝送効率が低く抑えられてしまう。このため、音声伝送に用いられる各VCIを m ビットの共有VCI部(SCI_m)と n ビットの下位VCI部(LCL_n)に分割し、仮想チャネル設定の際、スイッチの同一出力方路へ向かう呼、つまり同一のVCI共有セルに多重化される呼の共有VCI部が、共通の値を持つよう設定する。設定されたVCIの共有VCI部の情報はVCI共有セルヘッダに保持し、フェローセルヘッダには下位VCI部の情報のみ格納する。このとき、VCI共有セルヘッダにおいて下位VCI部を保持する必要はないので LCL_n には0が格納される。各呼のVCIはVCI共有セルヘッダとフェローセルヘッダの情報から再構成することができるため、スイッチの同一出力方路へ向かう複数の呼をVCI情報を損なうことなく、また伝送効率の劣化を抑えてVCI共有セルに収容することができる。このようなセル構成により、音声データのセル化遅延の削減、および高い伝送効率が実現できる。注意すべきことは、VCI共有セルはそれぞれの呼を識別するために新たな識別子をヘッダに持つ必要はなく、元のソリタリセルのヘッダ情報から過不足なく、VCI共有セルを構成可能であることである。また、VCI共有セルヘッダのVPI, PT (Payload Type), CLP(Cell Loss Priority), HEC(Header Error Control)については、標準ATMのヘッダに従う。

一方、加入者端末からローカルスイッチまでのVCリンク(Fig. 4.2中のVCリンクA)においては、音声用セルはリルートノード($RL_{i,j}$)まで到達していないため、VCI共有セルへの多重化は適用できない。CLADにおけるセル化遅延と、加入者線のトラフィック密度が低いことを考慮して、各呼を k バイトの音声データ毎にセル化する。このセルをソリタリセルとする。ソリタリセルの構成をFig. 4.8(b)に示す。ここで、ソリタリセルに格納する音声データ長(k バイト)をフェローセルのペイロード長に等しくすることで、リルートノードにおけるセルの処理が容易になる。ソリタリセルヘッダのVCIフィールドも $SCI_m(m$ ビット)と $LCL_n(n$ ビット)に分割されており、セルに収容されている呼のVCIの共有VCI部と下位VCI部となる。他のフィールドに関しては、標準ATMヘッダと同様とする。同様に、Fig. 4.2のVCリンクD''の異なる呼は各コネクション毎に分割するため、まとめることはできない。

4.3.4 VCI共有セルのVCI割り当て

音声のコネクションに対して新しいVCCを設定する際、VCIフィールドはVCI共有セルの組み替えが可能となるように割り当てる。新たにコネクションを接続する呼 a と同じ出力方路へ向かう呼 b が存在するとき、 a の共有VCI部($SCI_m(a, X)$)には、先に設定されている b の共

有 VCI 部 ($SCI_m(b, X)$) と同じビット列を割り当てる. a の下位 VCI 部 ($LCI_n(a, X)$) には, 同じ共有 VCI 部 ($SCI_m(a, X)$) を持つ他の呼と違うビット列を割り当てる. VCI 共有セルヘッダにおいて, LCI_n フィールドは 0 とする.

ここで例として, Fig. 4.2 中の $TE_{i,j}(\alpha)$ から $TE_{i',j'}(\alpha')$ 間の呼 a と $TE_{i,j}(\beta)$ から $TE_{i'',j''}(\beta')$ 間の呼 b の VCI を割り当てる場合を考える ($i \neq i', i \neq i'',$ かつ $i' \neq i''$). 呼 a と b は同じリルートノード RT_i を通るため, $SCI_m(a, B)$ は, $SCI_m(b, B)$ と等しい. 一方, $CI_{m+n}(a, A)$ と $LCI_n(a, B)$ は, それぞれ $CI_{m+n}(b, A)$ と $LCI_n(b, B)$ とは等しくない. 他のリルートノードでは, 変換した VCI を用いて同様にセルの組み替えが行われる. ここで, VCI 共有セルのヘッダは, 下位 n ビットがすべて 0 である VCI ($SCI_m, all_n(0)$) を持つことに注意する. ただし, $all_w(d)$ は, w 個の d からなるビット列を表す. 例えば, $all_5(0) = 00000$ である. この VCI は, 他の呼の VCI と一致しない. つまりネットワーク内の特定の VC リンク間でのみ動的に存在する一時的な VCI である.

4.3.5 VCI 共有セル長と伝送効率の評価

考案する VCI 共有セル化手法によって, ネットワーク内のセル数を削減することができる. しかしながら, その減少率はフェローセルの長さによって変化する. フェローセルの長さ 4, 6, 8, 12 バイト (ヘッダ 1 バイト) と変化させた時のネットワークの伝送効率を評価する. これらは 48 バイトのペイロードに過不足なく収容することができる長さであるが, 2 あるいは 24 バイトは, フェローセル長としては明らかに不適切であるので考慮しない. また, リルートノードがハードウェアのみで VCI 共有セルを処理することを考慮して, フェローセル長は固定とする.

C 言語で記述したリルートノードの機能シミュレータを用いて, 各フェローセル長に対するリルートノードの性能評価を行った. 結果を Table 4.1 に示す.

Table 4.1 リルートノードのシミュレーション結果

フェローセル長 (バイト)	12	8	6	4
呼数/セル	4	6	8	12
ヘッダ/ペイロード (%)	8.3	12.5	16.7	25.0
セル化遅延 (μs)	1,375	875	625	375
入力セル数/秒	51,212	80,480	112,678	187,799
出力セル数/秒	14,073	15,816	17,805	22,365

入力/出力セル数はある一定のトラヒック時において, 1 秒あたりにローカルスイッチのリルートノードに入力/出力するセル数を示す. シミュレーション結果の解析より, フェローセル

の長さを短くすると、リルートノードでの入力セル数に対する出力セル数の削減率が高くなり、セル化遅延も短くなるが、伝送効率は悪化し、セル数自体も増加する。本論文ではフェローセルの長さを1バイトのヘッダを含めて8バイトとする。このフェローセル長はCLADでのセル化遅延が1msを越えない中では最も出力セル数が少なく、かつ、セル化遅延の点で優れている。この時、Table 4.1より、リルートノードに入力するVCI共有セル数の約80%を削減できることがわかる。

4.4 結言

音声データのセル化遅延を考慮した上で、音声伝送効率をよく伝送するマルチメディア ATM ネットワーク手法について記述した。マルチメディア ATM ネットワークは、スイッチ毎にリルートノードを配置することを特徴とし、ネットワーク内で動的に音声用セルを多重化したり、あるいは組み替えたりすることで伝送効率の向上を実現した。この処理を実現するために、新たにVCI共有セル化手法を考案した上、考案したセル化手法を通じて達成される伝送効率を評価し、セル数を約80%削減可能であることを示した。VCI共有セルは、標準ATMセルとの親和性を損わないように設計され、VCI共有セルと標準ATMセルの区別をネットワーク内のリルートノードで認識するだけで、音声伝送効率を向上可能なことを示した。

第5章

マルチメディア ATM ネットワークにおけるリルートノード

5.1 緒言

本章では、第4章で考案したマルチメディア ATM ネットワーク内において、VCI 共有セルの多重化や組み替え処理、ならびに標準 ATM セルのヘッダ変換処理を実現するリルートノードの VLSI 化設計に関して記述する。リルートノードは、VCI 共有セルに関する固有な処理を、バッファのアドレス制御と CAM(Content Addressable Memory) の基本機能を活用した簡易な機構を用いて実行することができ、マルチメディア ATM ネットワーク内の他の構成要素に影響を与えることなく、音声伝送効率が向上できることを示す。設計したリルートノードの VLSI 化設計を *COMPASS Design Navigator* を用いて行った実装結果を示す。

5.2 リルートノードアーキテクチャの設計

標準 ATM セルヘッダの変換および VCI 共有セルの多重化/組み替えを行うリルートノードのハードウェアアーキテクチャについて述べる。標準 ATM ネットワークでは、HCV(Header Converter) が ATM スイッチの前に置かれ、セルヘッダを次の VC リンクで使用するヘッダに変換する。VCI 共有セルの多重化/組み替えを行うリルート機能は、このヘッダ変換の機能を利用している。このため、ATM セルヘッダの変換とリルート機能を1チップ (Fig. 4.2 における RT, または RL) に実装する。リルートノード VLSI は Fig. 5.1 に示すように、アドレス生成部 (address generator), ヘッダテーブル (new header table), 標準セルバッファ (standard cell buffer), セル組み替え制御部 (reassignment controller), フェローセルバッファ (fellow cell buffer) からなる。以下、Fig. 4.2 の VC リンク B, C の間に位置するリルートノード RT_i を例に、各部の詳細を述べる。

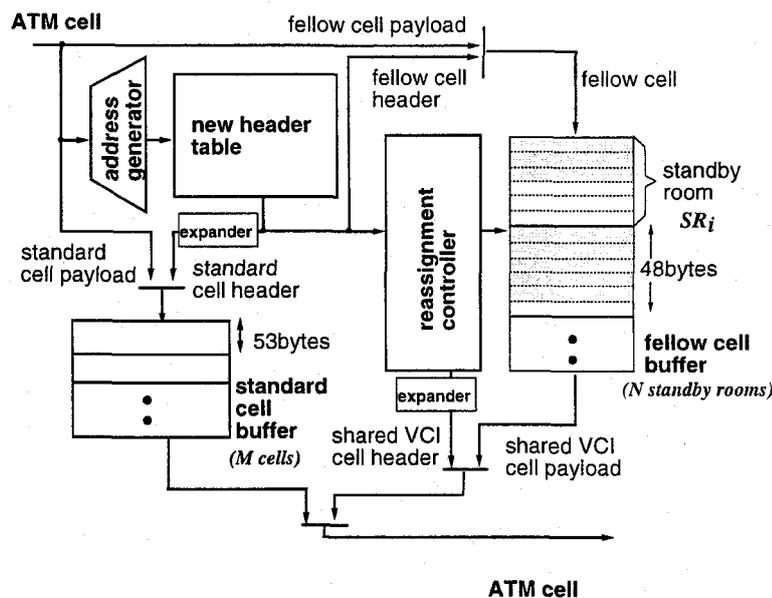


Fig. 5.1 リルートノードの構成

5.2.1 アドレス生成部

アドレス生成部は、入力セルのヘッダ中にある28ビットからなるルーティング情報 $\langle all_{12-l}(0), PI_l, all_{16-(m+n)}(0), CI_{m+n} \rangle$ から、内蔵する制御レジスタの情報を用いて、ネットワークに使用されていない領域を除いた13ビットのヘッダテーブルのアドレス $\langle all_{13-(l+m+n)}(0), PI_l, CI_{m+n} \rangle$ を生成する。生成されたアドレスを圧縮ルートデータと呼ぶ。アドレス生成部ではリルートノードに新しいセルが到着した時、VCI共有セルの識別子であるVCIフィールド CI_{m+n} の上位ビットを調べ、VCI共有セルかどうかを判断する。識別子の位置は制御レジスタが保持しているため、VCI共有セルはハードウェアによる識別が可能である。

VCI共有セルが到着した場合、アドレス生成部は、VCI共有セルヘッダの共有VCI部、 SCI_m とフェローセルヘッダが持つ下位VCI部、 LCI_n から、セル中の呼のVCIを再構成する。アドレス生成部は、この再構成されたVCI(16ビット)と共有VCIセルヘッダ内のVPI(12ビット)から28ビットのルーティング情報を生成し、これを入力として13ビットの圧縮ルートデータを出力する。例えば、VCI共有セルが呼 a, b を含んでいるとする。まず、アドレス生成部には、呼 a のVPI/VCIを再構築するために $\langle all_{12-l}(0), PI_l(a, B), all_{16-(m+n)}(0), SCI_m(a, B), LCI_n(a, B) \rangle$ が入力される。次に呼 b のVPI/VCIとして $\langle all_{12-l}(0), PI_l(a, B), all_{16-(m+n)}(0), SCI_m(a, B), LCI_n(b, B) \rangle$ が入力される。これは、 $LCI_n(b, B)$ 以外は呼 a と同じビット列である。1個のVCI共有セルには最大6個のフェローセルが含まれるため、1セルあたり最大6度ヘッダテーブルを参照する。標準ATMセルが到着した場合は、標準ATMセルのヘッダから28ビットのルー

ティング情報を取り出し、1度だけ圧縮ルートデータを生成する。

5.2.2 ヘッダテーブル

ヘッダテーブルは、アドレス生成部が生成した圧縮ルートデータから、次のVCリンクで用いる新しい圧縮ルートデータを出力する。呼 a に対する VCC が接続されているとき、 $\langle \text{all}_{13-(l+m+n)}(0), \text{PI}_l(a, B), \text{SCI}_m(a, B), \text{LCI}_n(a, B) \rangle$ が示すアドレスの指す内容は $\langle \text{all}_{13-(l+m+n)}(0), \text{PI}_l(a, C), \text{SCI}_m(a, C), \text{LCI}_n(a, C) \rangle$ に設定されている。それぞれは、VC リンク B, C における同一の VCC ルーティング情報を示す。ヘッダテーブルは、4 個の 16 ビット \times 2,048 ワード RAM から構成する。したがって、最大 2^{13} 個の VCC エントリを扱うことができる。圧縮ルートデータが 13 ビットであるため、RAM の 16 ビット中下位 3 ビットは使用しない。

5.2.3 標準セルバッファ

標準セルバッファは、変換されたヘッダを持つ M 個の標準 ATM セルを格納するために用いる FIFO である。標準セルバッファでは、ヘッダテーブルから出力される圧縮ルートデータを標準 ATM セルヘッダに拡張し、ATM セルを構成するためにそのペイロードと連結する。連結された ATM セルは標準セルバッファに格納し、VCI 共有セルが出力を要求していない時にリルートノードから出力する。

5.2.4 フェローセルバッファ

フェローセルバッファは、変換されたヘッダの下位 VCI 部をヘッダとするフェローセルを格納する。フェローセルバッファは N 個のスタンバイルーム ($\text{SR}_0, \text{SR}_1, \dots, \text{SR}_{N-1}$) に分割しており、個々に VCI 共有セルのペイロードを格納する。スタンバイルームには、最大 6 個のフェローセルを格納することが可能である。

5.2.5 セル組み替え制御部

セル組み替え制御部は、VCI 共有セルの多重化/組み替え、および出力を行うために、フェローセルバッファの書き込み、および読み出し制御を行う。処理中のフェローセルのバッファ書き込みアドレスは、圧縮ルートデータから生成する。組み替えた後の VCI 共有セルを出力するための読み出しアドレスはタイムスタンプを用いて生成する。

セル組み替え制御部 (Fig. 5.2) は、 N 個の 13 ビット圧縮ルートレジスタ (packed route register, $(pr_0, pr_1, \dots, pr_{N-1})$), N 個の 1 ビット有効データレジスタ (valid register, $(v_0, v_1, \dots, v_{N-1})$), N 個の tc ビット タイムスタンプレジスタ (time stamp register, $(tm_0, tm_1, \dots, tm_{N-1})$), N 個

の3ビット呼数レジスタ (#calls register, $(c_0, c_1, \dots, c_{N-1})$), 1個の13ビットマスクレジスタ (mask register), 読み出し/書き込みアドレス生成部 (read/write address generator), タイムカウンタ (time counter) からなる. 圧縮ルートレジスタとタイムスタンプレジスタはCAM(Content Addressable Memory)の原理に基づいて構成している. pr_i と c_i , tm_i は信号線 h_i により接続しており, また tm_i は信号線 to_i に接続している. N 個のレジスタ v_i , pr_i , c_i , tm_i はスタンバイルーム SR_i に1対1に対応している.

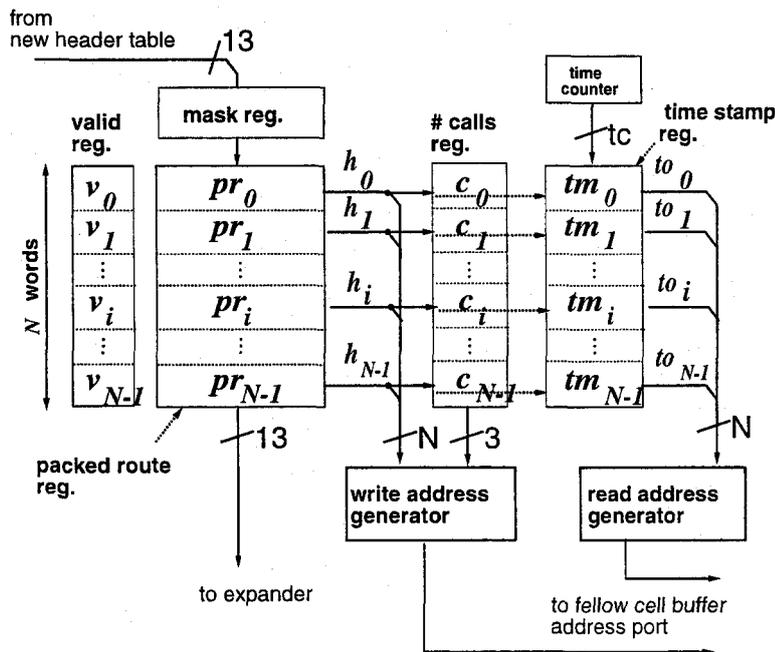


Fig. 5.2 セル組み替え制御部の構成

リルートノードに到着したVCI共有セルはペイロード部とヘッダ部に分けられ, それぞれ組み替えられた後, SR_i と pr_i に格納される. ただし pr_i にはヘッダを表す圧縮ルートデータを格納し, c_i にはスタンバイルーム SR_i のフェローセル数を格納する. 例えば, 4フェローセルが SR_i に保持されている時は, c_i は $\text{bin}_3(4)$ が格納されている. ただし, $\text{bin}_w(i)$ は, i の w ビットの2進表記を表す. ここで, i は, $0 \leq i \leq 2^w - 1$ の整数である. 例えば, $\text{bin}_5(9)$ は, 01001となる. 有効データレジスタ v_i は, SR_i に有効なフェローセルが格納されている時, 1となる. LCI_n の位置を示すためにビット列 $\langle \text{all}_{13-n}(0), \text{all}_n(1) \rangle$ がマスクレジスタに格納されている.

書き込みアドレス生成部は, 組み替え処理を行っている呼 a のフェローセルを格納すべきアドレスを生成する. まず, 呼 a の下位VCI部 LCI_n を除いた圧縮ルートデータと同一の圧縮ルートデータを保持するフェローセルがスタンバイルームに格納されているかどうかを, 圧縮ルートレジスタのCAM機能を用いて調べる. SR_i に格納されている時 h_i は1となり, 呼 a のフェロー

セルを SR_i に格納する。 SR_i がすでに 6 個のフェローセルを格納している場合や、呼 a の圧縮ルートデータと一致するものが存在しないとき、 h_i は全て 0 となり、新しい $SR_j (v_j = 0, i \neq j)$ を呼 a のフェローセルのために割り当てる。以上の制御により、書き込みアドレス生成部は h_i および c_i 、または使用していない SR_i の先頭アドレスから容易に書き込みアドレスを計算することができる。

読み出しアドレス生成部は、出力要求のある VCI 共有セルのアドレスを生成する。VCI 共有セルは各スタンバイルームが 6 フェローセルで充足しているかに関わらず、一定時間 TM 後に出力する。 TM の単位時間は、それぞれのセルが到着する間隔 (1 セル時間) である。1 セル時間 ($TM = 1$) は通常 53 サイクルである。 SR_i の最初のフェローセルが到着したとき、タイムカウンタは bin_{tc} (現在の時刻 + TM) の値を tm_i に格納する。1 セル時間毎に現在の時刻をタイムスタンプレジスタに入力し、一致する tm_i が存在するかどうか調べる。現在の時刻と一致する tm_i が存在するとき、 to_i に 1 がセットされ、読み出しアドレス生成部は SR_i に格納されている VCI 共有セルの読み出しアドレスを生成する。

設計したリルートノードにより、一度、仮想チャネルコネクションがネットワーク上で設定されると、コネクション上の VCI 共有セルはハードウェアによる処理のみで新しいヘッダに変換することができる。したがって、VCI 共有セルは ATM スイッチの高速スイッチング特性を損なわない。さらに、このリルートノードは標準 ATM セルのヘッダの変換も行うことができるため、従来の HCV をリルートノードに置き換えるだけで、提案するマルチメディア ATM ネットワーク機能を実現することができる。

5.3 バッファサイズシミュレーション

リルートノードでは、VCI 共有セルの出力優先度は標準 ATM セルよりも高く設定している。このため、標準セルバッファに保持することの可能なセル数 M とフェローセルバッファに保持することの可能なセル数 N を、リルートノード内のセルの棄却ができるだけ起こらないように設計しなければならない。両方のバッファが必要とするサイズは、タイムアウトの間隔 TM 、セルの到着率、VCI 共有セルが全体のセルに占める割合によって変化する。適切なバッファサイズを求めるために、セル組み替え制御部および各バッファのシミュレーションを行った。C 言語で記述したシミュレータにおいて、タイムアウト間隔 TM 、セル到着率、VCI 共有セルの占有率の 3 パラメータを変化させ、最大バッファサイズ M と N を求めた。それぞれのバッファサイズは、到着する VCI 共有セルがすべて 6 個のフェローセルを含む場合についてシミュレートした。

シミュレーション結果の解析より、フェローセルバッファの最大サイズ N は TM に比例する

という結果が得られた。一方、 M は変化させた3要因のすべてにより変化する。Fig. 5.3(a), (b), (c)に、タイムアウト間隔をそれぞれ $TM = 16, 32, 64$ としたときの、標準セルバッファのオーバーフローを発生しないために必要なバッファ容量を示している。

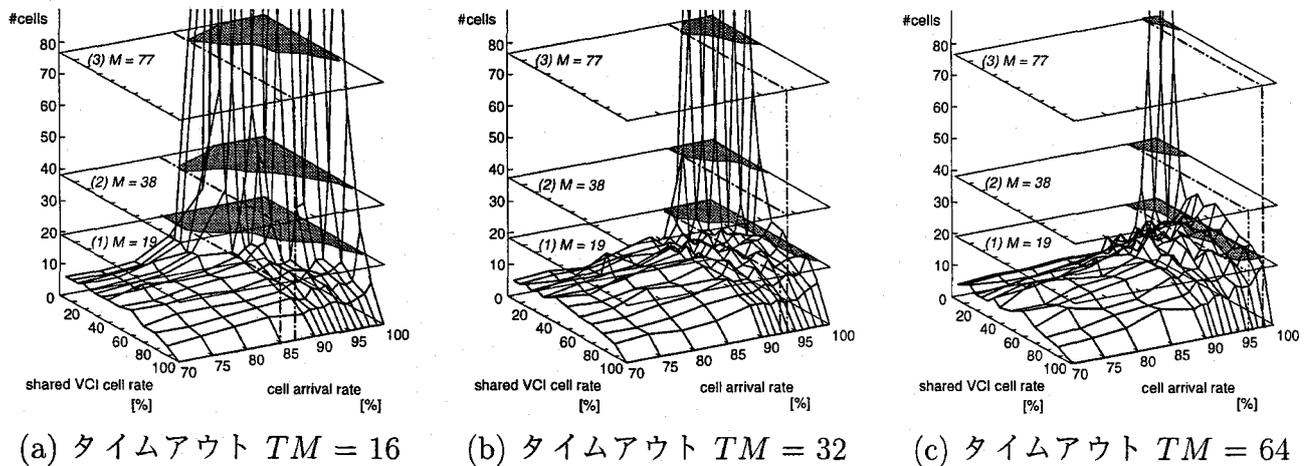


Fig. 5.3 標準セルバッファサイズのシミュレーション結果

セル到着率が高く、VCI共有セルの占有率が低いとき、必要となる標準バッファサイズ M は急激に増加する。この現象は、 TM がVCI共有セルの到着率に対して低すぎる時に6フェローセルで充足されていない状態で送出されるVCI共有セルが増加するためである。到着するVCI共有セルは最大6個のVCI共有セルに分裂する可能性がある。

グラフ中の平面(1), (2), (3)は、それぞれ標準セルバッファサイズ $M = 19, 38, 77$ を示し、これらのバッファはそれぞれ8ビット×1,024ワードRAM, 8ビット×2,048ワードRAM, 8ビット×4,096ワードRAMで構成することができる。平面の灰色部は、先に述べたセルの増加現象により、バッファのオーバーフローが発生し、セルの棄却が生じる領域を示している。

タイムアウト TM が16であるFig. 5.3(a), (1), (2), (3)の3構成では、セル棄却領域がセル到着率85%でさえも生じる。このセル到着率の時、次段のATMスイッチが発生するセル棄却の確率が非常に少ないのに対し、リルートノードを付加することで新たなセル棄却の要因を作ることとなる。しかしながら、セル到着率が95%を越える領域では、ATMスイッチにおいてもセル棄却が頻繁に発生するため、Fig. 5.3(b), (c)の構成のリルートノードを付加することで新たに発生するセル棄却は比較的少ないといえる。また、Fig. 5.3(b), (c)において、平面(1)($M = 19$)の構成ではバッファサイズに十分な余裕がないため、広い範囲でセルの棄却が発生する場合がある。他の4候補(Fig. 5.3(b), (c)の平面(2), (3)の構成)のうち、 TM や M の増加によりバッファ面積とセル化遅延が増加することを考慮して、Fig. 5.3(b)の平面(2)を採用する。標準セルバッファ、フェローセルバッファともに、8ビット×2,048ワードの2ポート

RAMにより実装することができ、それぞれ標準セルを38セル、VCI共有セルを42セル保持することができる。バッファによるリルートノードの付加遅延は、タイムアウト間隔 $TM = 32$ とした時 $0.09ms$ であるため、セル化遅延や、スイッチの平均伝送遅延が $0.5ms$ 程度^[12]であることと比較して十分小さい。

5.4 実装結果

COMPASS Design Navigator を用いて、設計したリルートノードを1チップに実装した。Table 5.1 にリルートノードの諸元を示す。Fig. 5.4 は $0.6\mu m$ CMOS スタンダードセル、データパスライブラリを用いて得られたレイアウトである。

Table 5.1 リルートノードの諸元

テクノロジー	0.6 μm 3層 CMOS	
	サイズ	トランジスタ数
リルートノード	9.50 × 5.21mm	1,358,619
ヘッダテーブル	32.10mm ²	1,002,440
標準セルバッファ	3.78mm ²	126,775
フェローセルバッファ	3.78mm ²	126,775
セル組み替え制御部	4.97mm ²	57,473
リルートノード制御部	2.23mm ²	27,256
最大動作周波数	32MHz	

ヘッダテーブル、標準セルバッファ、フェローセルバッファは2ポートRAMにより実装した。一方、セル組み替え制御部はデータパスライブラリにより実装した。アドレス生成部や他の制御部はVHDLで記述し、スタンダードセルライブラリにより合成した。

HCVと比較してVCI共有セルを処理するために新たに付加された機能ブロックはフェローセルバッファ、セル組み替え制御部、リルートノード制御部である。標準ATMセルのHCVをリルートノードで置き換えた結果、これらのブロックによる面積増加は約54.2%である。155.52Mbpsの伝送レートのATMを8ビット並列で処理するために必要な動作周波数は19.44MHzであることから、設計したリルートノードはこれを処理することが可能となった。

5.5 結言

遅延時間を考慮した上においても、音声を効率よく伝送することのできるマルチメディアATMネットワークで用いる、リルートノードのVLSI化設計について述べた。VCI共有セルの

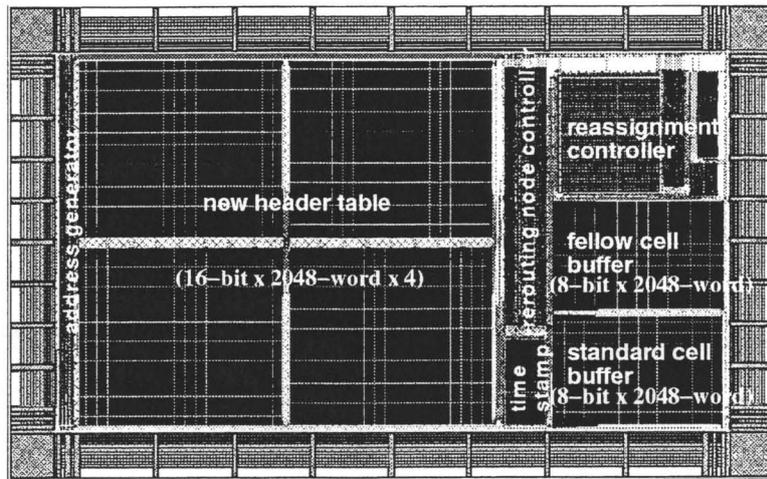


Fig. 5.4 リルートノードのレイアウト

多重化や組み替え処理，ならびに標準 ATM セルのヘッダ変換処理を可能とするリルートノードは，主に 2 ポート RAM と，CAM の基本機能を用いた制御部の簡易な機構により，1 チップに実装可能であることを示した．音声伝送効率を向上させるマルチメディア ATM ネットワークはネットワーク内の ATM スイッチにリルートノードを配置することで構成可能であることを示した．

第6章

結論

本論文では、マルチメディア ATM ネットワークを構築するための課題である画像と音声の実時間処理に関する主要な要素技術として、ビデオ画像生成用途に適した多重スレッドプロセッサの VLSI 化設計、ビデオ動画像圧縮符号の国際標準 MPEG2 HDTV レベルデコーダの VLSI 化設計、および音声伝送に適したマルチメディア ATM ネットワークの構成と音声用 ATM セルの処理を行うリルートノードに着目し、各々の VLSI 化設計を行った。

本研究で得られた主要な結果を以下に要約する。

- ビデオ画像生成用アルゴリズムを高速に処理するシステムの要素プロセッサとして、多重スレッドプロセッサの VLSI 化実装を行った。多重スレッドプロセッサは、ビデオ画像生成アルゴリズムに内在する粗粒度並列性を利用し処理の高速化を図るとともに、機能ユニット等を共有しハードウェア量の削減を実現したアーキテクチャであるため、コスト性能比に優れている。高位合成システムを用いて動作記述からハードウェアを自動合成する手法を用いたことにより、共有化したユニットに対する制御が複雑で、これまで実現が困難であった多重スレッドプロセッサの VLSI 化を達成した。
- 画像圧縮符号化の国際標準 MPEG2 の MP@HL デコーダの 1 チップ VLSI 化設計を行った。従来実現されている MP@ML デコーダの約 6 倍の処理能力が必要な MP@HL デコーダに対し、MPEG2 の復号化の各処理過程を実行するために新たに考案した専用アーキテクチャを活用することによって、処理性能の要求を達成した。設計した各機能ユニットをマクロブロックレベル・パイプラインバッファで接続し、また、高速処理のボトルネックとなっていたフレームメモリアクセス数を削減することで、1 チップ HDTV レベル MPEG2 デコーダの VLSI 化を実現した。
- 音声伝送効率の改善を目的とした FTTH 構想に基づくマルチメディア ATM ネットワーク構成を考案した。ネットワーク内で動的に音声用セルを多重化したり、あるいは組み替

えたりすることによって伝送効率の向上を図った。この処理を実現するために、新たに VCI 共有セル化手法を考案し、考案したセル化手法を通じて達成される伝送効率を評価した。設計した VCI 共有セルは、標準 ATM セルとの親和性を損なうことがなく、さらに、VCI 共有セルと標準 ATM セルの区別をネットワーク中のリルートノード内において認識するだけで、ネットワーク全体の音声伝送効率が向上できることを示した。

- マルチメディア ATM ネットワーク内において、VCI 共有セルの多重化や組み替え処理、ならびに標準 ATM セルのヘッダ変換処理を可能とするリルートノードの VLSI 化設計を行った。リルートノードは VCI 共有セルに関する固有な処理を、単純な機構のハードウェアを用いて処理することで、ATM の高速ハードウェアスイッチングの特性を維持することを可能とした。リルートノードの導入により、他のネットワーク要素の変更なしに音声用 ATM セルの効率的な伝送を達成するマルチメディア ATM ネットワークが実現可能であることを示した。

今後の課題としては、本研究で設計/考案した VLSI やネットワーク構成をもとに、マルチメディア ATM ネットワークにおけるさらに発展した情報通信サービスの実現に向け、(i) マルチメディア ATM ネットワークのノードシステムの開発、(ii) ネットワーク規模での動的なリルート機能の実現、(iii) 画像/音声伝送の同期方式の考案、に関する研究を行う必要がある。また、各システムの小型化、携帯化、高機能化を目指し、VLSI の面積の縮小化、消費電力削減、各処理の高速化を達成することも重要な課題である。

謝辞

本研究の全過程を通じて、終始御懇切な御指導と御鞭撻を賜りました、大阪大学大学院工学研究科情報システム工学専攻白川功教授に謹んで深謝の意を捧げます。

大阪大学大型計算機センター村上孝三教授には御懇到な御指導、御助言を賜り、心から深謝の意を捧げます。

大阪大学大学院工学研究科情報システム工学専攻石浦菜岐佐助教授には適切な御指導、ご助言をいただき、心から感謝申し上げます。

本論文を執筆するにあたり有益な御教示、御助言を賜りました、大阪大学大学院工学研究科情報システム工学専攻寺田浩詔教授、大阪大学大学院工学研究科電子情報エネルギー工学専攻岸野文郎教授に厚く感謝申し上げます。

大阪大学大学院博士前期、後期両課程に在学中に多くの御指導、御教示を賜りました大阪大学大学院工学研究科情報システム工学専攻藤岡弘教授、西尾章治郎教授、薦田憲久教授、鈴木胖教授に厚く御礼申し上げます。

本研究の全過程を進めるにあたり、直接の御指導、御助言、御討論頂き、また様々な面でお世話になりました同情報システム工学専攻の尾上孝雄助手に衷心より感謝申し上げます。

第2章における研究を行うにあたり、適切な御指導、御討論をいただいた大阪大学大学院在学中の木村浩三氏、京都工芸繊維大学平田博章助手に深甚なる謝意を表します。また、ともに設計に携わり有益な御討論をいただいた日本電気株式会社古賀拓也氏、藤井貴晴氏に深く感謝致します。

第3章について、ともに研究を進め御討論、御協力を頂いた三菱電機株式会社森本康夫氏、株式会社東芝田村尚之氏、白川研究室大学院生佐藤洋氏に感謝の意を表します。

第4章、第5章における研究を行うにあたり、ともに研究を進めご協力いただいた白川研究室大学院生中谷泰寛氏、カリフォルニア大学バークレイ校学生 Tom Chiou 氏に感謝申し上げます。

最後に、大阪大学大学院工学研究科情報システム工学専攻重弘裕二助手、秘書馬場有紀子氏、研究員 Itthichai Arungsrisangchai 氏、藤嶋秀幸氏、井堀雅淳氏、同大学院在学中の長尾明氏、宇野裕史氏、岡田圭介氏、近藤仁志氏、矢野政顕氏、山口雅之氏、大学院生奥畑宏之氏、熊谷

圭司氏，藤田玄氏，森川俊氏，宮野鼻晃士氏，山本哲三朗氏，宋宝玉嬢をはじめとする白川研究室の諸氏には種々の面で御協力頂き，厚く御礼申し上げます。

参考文献

- [1] T. Whitted: "An improved illumination model for shaded display," *Commun. ACM*, vol. 23, no. 6, pp. 343–349 (June 1980).
- [2] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile: "Modeling the interaction of light between diffuse surfaces," in *Proc. SIGGRAPH*, vol. 18, 3, pp. 213–222 (July 1984).
- [3] ISO/IEC 11172-2: "Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s," International Standard (1993).
- [4] ITU-T Rec. H.262, ISO/IEC 13818-2: "Information technology - generic coding of moving pictures and associated audio," International Standard (March 1994).
- [5] T. Demura, T. Oto, K. Kitagaki, S. Ishiwata, G. Otomo, S. Michinaka, S. Suzuki, N. Goto, M. Matsui, H. Hara, T. Nagamatsu, K. Seta, T. Shimazawa, K. Maeguchi, T. Odaka, Y. Uetani, T. Oku, T. Yamakage, and T. Sakurai: "A single-chip MPEG2 video decoder LSI," in *ISSCC Digest of Technical Papers*, pp. 72–73 (February 1994).
- [6] M. Toyokura, M. Saishi, S. Kurohmaru, K. Yamauchi, H. Imanishi, T. Ougi, A. Watabe, Y. Matsumoto, T. Morishige, H. Kodama, E. Miyagoshi, K. Okamoto, M. Gion, T. Mine-maru, A. Ohtani, T. Araki, K. Aono, H. Takeno, T. Akiyama, and B. Wilson: "A video DSP with a macroblock-level-pipeline and a SIMD type vector-pipeline architecture for MPEG2 CODEC," in *ISSCC Digest of Technical Papers*, pp. 74–75 (February 1994).
- [7] T. Koinuma and N. Miyaho: "ATM in B-ISDN communication systems and VLSI realization," *IEEE J. Solid-State Circuits*, vol. 30, no. 4, pp. 341–347 (April 1995).
- [8] "Voice and telephony over ATM -ATM Trunking for narrowband services," ATM Forum, 95-0446R5 (February 1996).

- [9] H. Iwata, H. Yamada, T. Sampei, and S. Kikuchi: "An ATM subscriber line interface circuit for voice and telephony over ATM (VTOA)," in *Proc. GLOBECOM*, pp. 270-274 (November 1995).
- [10] R. Mauger and S. Brueckheimer: "The role of ATM in 64 kb/s switching and transmission networks," in *Proc. ISS*, vol. 2, pp. 87-91 (April 1995).
- [11] H. Ohnishi, S. Suzuki, H. Nakayama, and H. Tanaka: "All band switching node architecture for flexible and cost-effective evolution toward B-ISDN," in *Proc. ISS*, vol. 1, pp. 57-61 (April 1995).
- [12] I. Gard and L.-G. Petersen: "Supporting STM traffic with ATM - a switch implementation," in *Proc. ISS*, vol. 1, pp. 62-66 (April 1995).
- [13] H. Nakamura, M. Onuki, and A. Nakajima: "Using ATM to carry very low bit-rate mobile voice signals," in *Proc. ICUPC*, pp. 863-867 (November 1995).
- [14] R. A. Spanke and J. M. Adrian: "ATM composite cell switching for DS0 digital switches," in *Proc. ISS*, vol. 1, pp. 268-272 (April 1995).
- [15] 平田博章, 木村浩三, 永峰聡, 西澤貞次, 鷺島敬之: "多重スレッド・多重命令発行を用いる要素プロセッサ・アーキテクチャ," *情報処理学会論文誌*, vol. 34, no. 4, pp. 595-605 (April 1993).
- [16] Y. Nakamura, K. Oguri, A. Nagoya, M. Yukishita, and R. Nomura: "High-level synthesis design at NTT Systems Labs," *Trans. IEICE*, vol. E76-D, no. 9, pp. 1047-1054 (September 1993).
- [17] H. Nishimura, H. Ohno, T. Kawata, I. Shirakawa, and K. Omura: "LINKS-1: A parallel pipelined multimicrocomputer system for image generation," in *Proc. Annual Int. Symp. Computer Architecture*, pp. 387-394 (June 1983).
- [18] 小栗 清, 中村 行宏, 野村 亮, 名古屋 彰: "主要なハードウェア記述言語の特徴と標準化状況 SFL," *情報処理学会誌*, vol. 33, no. 11, pp. 1256-1262 (November 1992).
- [19] 神原弘之, 安浦寛人, P. Kukkal, H. Kobayashi, 野地保, 小栗清: "ハードウェア記述言語の比較," *情報処理学会誌*, vol. 33, no. 11, pp. 1269-1283 (November 1992).
- [20] *SPARC RISC User's Guide*, Cypress Semiconductor (February 1990).

- [21] 木村浩三, 平田博章, 清原督三, 浅原重夫, 鷺島敬之, 尾上孝雄, 白川功: “画像生成用マルチスレッド型プロセッサのマイクロ・アーキテクチャ,” 情処研報, 94-ARC-104, vol. 94, no. 13, pp.97-104 (January 1994).
- [22] K. Aono, M. Toyokura, T. Araki, A. Ohtani, H. Kodama, and K. Okamoto: “A video digital signal processor with a vector-pipeline architecture,” *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp.1886-1894 (December 1992).
- [23] T. Inoue, J. Goto, M. Yamashina, K. Suzuki, M. Nomura, Y. Koseki, T. Kimura, T. Atsumo, M. Motomura, B. S. Shih, T. Horiuchi, N. Hamatake, K. Kumagai, T. Enomoto, H. Yamada, and M. Takada: “A 300-MHz 16-b BiCMOS video signal processor,” *IEEE J. Solid-State Circuits*, vol. 28, no. 12, pp.1321-1330 (December 1993).
- [24] S.-F. Chang and D. G. Messerschmitt: “Designing high-throughput VLC decoder, part I-Concurrent VLSI Architectures,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 2, no. 2, pp.187-196 (June 1992).
- [25] H.-D. Lin and D. G. Messerschmitt: “Designing high-throughput VLC decoder, part II-Parallel decoding methods,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 2, no. 2, pp.197-206 (June 1992).
- [26] 大井康, 谷口敦, 出村茂樹: “適応木探索法を用いた可変長符号の復号化 LSI の開発,” 信学技報, ICD94-87, pp.17-24 (August 1994).
- [27] M. Matsui, H. Hara, K. Seta, Y. Uetani, L.-S. Kim, T. Nagamatsu, T. Shimazawa, S. Mita, G. Otomo, T. Oto, Y. Watanabe, F. Sano, A. Chiba, K. Matsuda, and T. Sakurai: “200MHz video compression macrocells using low-swing differential logic,” in *ISSCC Digest of Technical Papers*, pp.76-77 (February 1994).
- [28] E. Komoto and M. Seguchi: “A 110MHz MPEG2 variable length decoder LSI,” in *Symp. VLSI Circuits Digest of Technical Papers*, pp.71-72 (June 1994).
- [29] ITU-T Rec. H.261: “Video codec for audio visual services at $p \times 64$ kbit/s,” (March 1993).
- [30] ISO/IEC 10918-1: “Digital compression and coding of continuous-tone still images,” (1993).

- [31] S. Uramoto, Y. Inoue, A. Takabatake, J. Takeda, Y. Yamashita, H. Terane, and M. Yoshimoto: "A 100MHz 2-D discrete cosine transform core processor," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp.492-499 (April 1992).
- [32] W. H. Chen, C. H. Smith, and S. C. Fralick: "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Communications*, vol. 25, no. 9, pp.1004-1009 (September 1977).
- [33] A. Peled and B. Liu: "A new hardware realization of digital filters," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 22, no. 6, pp. 456-462 (December 1974).